Technical Documentation Page

1. Report No. CA08-0580	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle Alternatives to Labor Intensive Tasks in Roadside Vegetation Maintenance	5. Report Date June 30, 2008
	6. Performing Organization Code AHMCT
7. Author(s): Aaron Arsenault, Jonah Teeter-Balin, Steven Velinsky, Wilderich White	8. Performing Organization Report No. UCD-ARR-08-06-30-01
9. Performing Organization Name and Address AHMCT Research Center UCD Dept of Mechanical & Aeronautical Engineering Davis, California 95616-5294	10. Work Unit No. (TRAIS)
	11. Contract or Grant IA 65A0210 Task Order 08-14
12. Sponsoring Agency Name and Address California Department of Transportation Sacramento, CA 95819	13. Type of Report and Period Covered Final Report January 2006 - June 2008
	14. Sponsoring Agency Code CALTRANS

15. Supplementary Notes

16. Abstract

This report describes an investigation into machine based technologies that are alternatives to the current methods of vegetation control in roadside maintenance. Two different lines of inquiry are described in separate report sections. The first section is an analysis and comparison of technologies that are potential alternatives to the application of herbicides in vegetation control. The alternatives of hot foam application, radiant heating, and high pressure water application are compared to mowing and herbicide application in maintenance of areas around posts and guardrails, mow strips, and paved surfaces. A concept scoring matrix resulted in the recommendation that the hot foam application and radiant heating technologies are candidates for further development. The second section describes the detailed development of an autonomous mower design. Mower control is achieved using dead reckoning information obtained from shaft mounted encoders which is fused with absolute positioning and heading information provided by a differential (WAAS corrected) GPS unit, to obtain accurate localization. In this study however, the GPS unit provides accurate drift free heading, eliminating the need for a gyroscope unit, and leading to an overall low cost high performance solution. Implementation of autonomous mowing systems has potential to achieve large savings in mowing operations.

17. Key Words Highway maintenance, roadside, autonomous mower, vegetation control, herbicide alternatives, GPS control	18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

20. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 245	22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized
(PF V2.1, 6/30/92)

DISCLAIMER STATEMENT

California AHMCT Program
University of California at Davis
California Department of Transportation

ALTERNATIVES TO LABOR INTENSIVE TASKS IN ROADSIDE VEGETATION MAINTENANCE

Aaron Arsenault, Jonah Teeter-Balin, Wilderich White &
Steven Velinsky, Principal Investigator

AHMCT Research Report
UCD-ARR-08-06-30-04

Final Report of Contract IA 65A0210 – Task Order 07-06

June 30, 2008

This report has been prepared in cooperation with the State of California, Business Transportation and Housing Agency, Department of Transportation, and is based on work supported by Contract Number IA 65A0210 – Task Order 07-06 through the Advanced Highway Maintenance and Construction Technology Research Center at the University of California at Davis.

ABSTRACT

This report describes an investigation into machine based technologies that are alternatives to the current methods of vegetation control in roadside maintenance. Two different lines of inquiry are described in separate report sections. The first section is an analysis and comparison of technologies that are potential alternatives to the application of herbicides in vegetation control. The alternatives of hot foam application, radiant heating, and high pressure water application are compared to mowing and herbicide application in maintenance of areas around posts and guardrails, mow strips, and paved surfaces. A concept scoring matrix resulted in the recommendation that the hot foam application and radiant heating technologies are candidates for further development. The second section describes the detailed development of an autonomous mower design. Mower control is achieved using dead reckoning information obtained from shaft mounted encoders which is fused with absolute positioning and heading information provided by a differential (WAAS corrected) GPS unit, to obtain accurate localization. In this study however, the GPS unit provides accurate drift free heading, eliminating the need for a gyroscope unit, and leading to an overall low cost high performance solution. Implementation of autonomous mowing systems has potential to achieve large savings in mowing operations.

EXECUTIVE SUMMARY

This report describes an investigation into machine based technologies that are alternatives to the current methods of vegetation control in roadside maintenance. Two different lines of inquiry are described in separate report sections identified as Part 1 and Part 2.

Part 1 is an analysis and comparison of technologies that are potential alternatives to the application of herbicides in vegetation control. Caltrans commissioned AHMCT to look into their vegetation maintenance control plan and recommend ways to improve worker safety, cost effectiveness, and environmental impact. This report specifically addresses herbicide use and the feasibility of replacing herbicides with alternative mechanical vegetation control technologies. The reduction of herbicides is important because Caltrans mandated that herbicide use should be reduced to 80% of 1994 levels by 2012. This decision is based on an environmental impact report completed in 1992. So far, Caltrans is having problems reducing herbicide use because of maintenance required to limit fire risk and the spread of noxious weeds.

Caltrans is responsible for a vast array of vegetation maintenance tasks. In order to narrow the scope of the report, only three vegetation maintenance scenarios are considered in which to compare all of the mechanical vegetation control technologies. These three maintenance scenarios are post & guardrail, mow strip, and urban vegetation. Each scenario has unique characteristics that will test the flexibility and applicability of each vegetation control method. Caltrans' two main mechanical vegetation control methods – herbicide spraying and mowing – are compared with the three alternative vegetation control methods – hot foam application, radiant heating, and high pressure water application – in each of the three scenarios. Calculations for the bare cost of operation and the daily coverage area are included in each control method description. These calculations are used as part of the overall comparison of technologies. In addition, calculations based on estimates for the theoretical performance of automated roadside versions of each alternative technology are also included. These theoretical control technologies allow for fair comparison of control methods.

A concept scoring matrix is utilized for each scenario to compare each of the alternative vegetation control technologies. Those control methods that score the highest are recommended for future research and development and possible implementation. The highest scorers were the theoretical radiant heating technology, the theoretical hot foam application technology, and the mowing technology. The unmodified alternative vegetation control methods and herbicide use did not score well in any of the scenarios. Mowing is also a suitable alternative to herbicide application. This analysis has shown that although the current level of many alternative technologies is not competitive with herbicide use, the development of specific alternative vegetation control technologies has potential. In addition, with the implementation of alternative technologies, Caltrans can continue to restrict herbicide application without reducing the level of vegetation maintenance. Based on the results of this analysis, it is recommended that radiant heating and hot foam application vegetation control technologies should be further developed for roadside use by Caltrans.

Part 2 details the proposal and development of an autonomous roadside mowing agent aimed at supporting Caltran's needs for alternative vegetation control technologies. To aid in this study, a fully functional proof-of-concept testbed is designed and built. The testbed makes use of a low cost sensor array selected for quick, efficient, and effective roadside mowing. In particular, dead reckoning information obtained from shaft mounted encoders is fused with absolute positioning and heading information provided by a differential (WAAS corrected) GPS unit, to obtain accurate localization. Most other studies which have taken this approach complement dead reckoning with a high performance, but expensive, rate gyroscope. In this study however, the GPS unit provides accurate drift free heading, eliminating the need for a gyroscope unit, and leading to an overall low cost high performance solution. The overall sensor fusion process utilizes an Extended Kalman filter (EKF).

A series of out door tests were performed to evaluate the success and practicality of the proposed agent. The mowing performance was evaluated based upon the consistency of consecutive mowing paths. The results indicated that a mowing consistency on the order of 4.0 in (10.0 cm) is attainable. Further analysis indicates that the slow moving drifts inherent in GPS positioning information do not affect the overall performance of autonomous mowing substantially. The results also indicate that the future development and fusion into Caltrans current mowing practices could significantly reduce Caltrans dependence on herbicide, improve worker and public safety, lead to enhanced levels of environmental quality, and decrease the annual statewide mowing budget.

TABLE OF CONTENTS

LIST OF FIGURES

DISCLAIMER/DISCLOSURE

The research reported herein was performed as part of the Advanced Highway Maintenance and Construction Technology (AHMCT) Research Center, within the Department of Mechanical and Aeronautical Engineering at the University of California – Davis, and the Division of Research and Innovation at the California Department of Transportation. It is evolutionary and voluntary. It is a cooperative venture of local, State and Federal governments and universities.

The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California, the Federal Highway Administration, or the University of California. This report does not constitute a standard, specification, or regulation.

PART 1: ALTERNATIVES TO MOWING AND HERBICIDES

CHAPTER 1 Introduction

1.1 Project Outline

The California State Department of Transportation (Caltrans) is charged with the maintenance of state highways, federal highways and rights-of-way within California, including vegetation maintenance and control. Since the development of the Caltrans vegetation maintenance policy in 1987, the current vegetation maintenance practices have relied mainly upon mowing and herbicide application. Herbicides are widely considered the least labor intensive and most cost efficient method for vegetation maintenance. Herbicides are also the least environmentally friendly and have been mandated for dramatic reduction in use. This has posed a problem for Caltrans because there has been little spending on herbicide alternatives, which has prevented the development of new vegetation control technologies. This problem is also compounded by the continued budget short falls of the state of California.

In order to remedy this situation, Caltrans commissioned the Advanced Highway Maintenance and Construction Technology Research Center (AHMCT) to study current and experimental vegetation control technologies including herbicide application. The purpose of this study is to determine the most cost effective methods for vegetation maintenance, while also keeping in mind environmental and worker safety issues. This research should help provide Caltrans with a comprehensive integrated vegetation control policy that will reduce costs, decrease manual labor, increase efficiency, increase effectiveness and protect the environment.

There are currently many alternative vegetation control technologies, either under development, in testing, or in limited use throughout the United States. This report will examine the current Caltrans vegetation control methods — mowing and herbicide spraying — as well as the alternative technologies to determine if cheaper, more efficient, labor saving solutions can be developed. This analysis will be engineering based, where values are assigned to all positive and negative aspects of each method and then ranked in a uniform manner. It is hopeful that these analyses will highlight the methods that have the most promise for vegetation maintenance.

Once the analyses are complete, the methods that show the most promise will be examined for adaptation and adoption by Caltrans for roadside vegetation maintenance. Recommendations will be made for the specifications of the equipment and the methods of treatment for optimal performance. Even though the end result of this project is only a report and not equipment, it is hopeful that Caltrans and AHMCT will use the results of this report and will further develop the most promising alternative technologies for roadside vegetation maintenance.

1.2 Project Sponsors

1.2.1 Caltrans

California is the most populous state in the United States and is home to the seventh largest economy in the world. In order to support this economy and bourgeoning population, people must commute to and from work, tourists must have access to all areas of the state, goods must be transported, and citizens must be able to travel freely in a safe and timely manner. The majority of this travel is accomplished through automotive use of California's vast highway network. Every day of the year, 65% of the miles traveled by Californians are driven on state and county thoroughfares.[1] The California Department of Transportation (Caltrans) is responsible for this highway system, which includes the California State Highway System and the sections of the Interstate Highway System within California. Caltrans is charged with the design, construction, maintenance, and operation of these roadways and roadsides which total 24,375 centerline kilometers [15,146 miles] of highway and over 930 square kilometers [230,000 acres] of right-of-way, roadsides and medians.[2]

In order to fulfill its mandated duties, Caltrans is given a yearly budget of $6.5 billion.[3] The portion of this budget spent on maintenance activities is $761 million,[3] and about 40% of this amount is spent on roadside vegetation maintenance and control.[3] Caltrans is funded both locally by the state of California and by the federal government. In the face of continued federal and state budget shortfalls, Caltrans has continually been called upon to reduce its budget even though total automobile traffic is increasing. Figure 1.1 shows the increase in the number of vehicle miles per lane mile of California State highways.

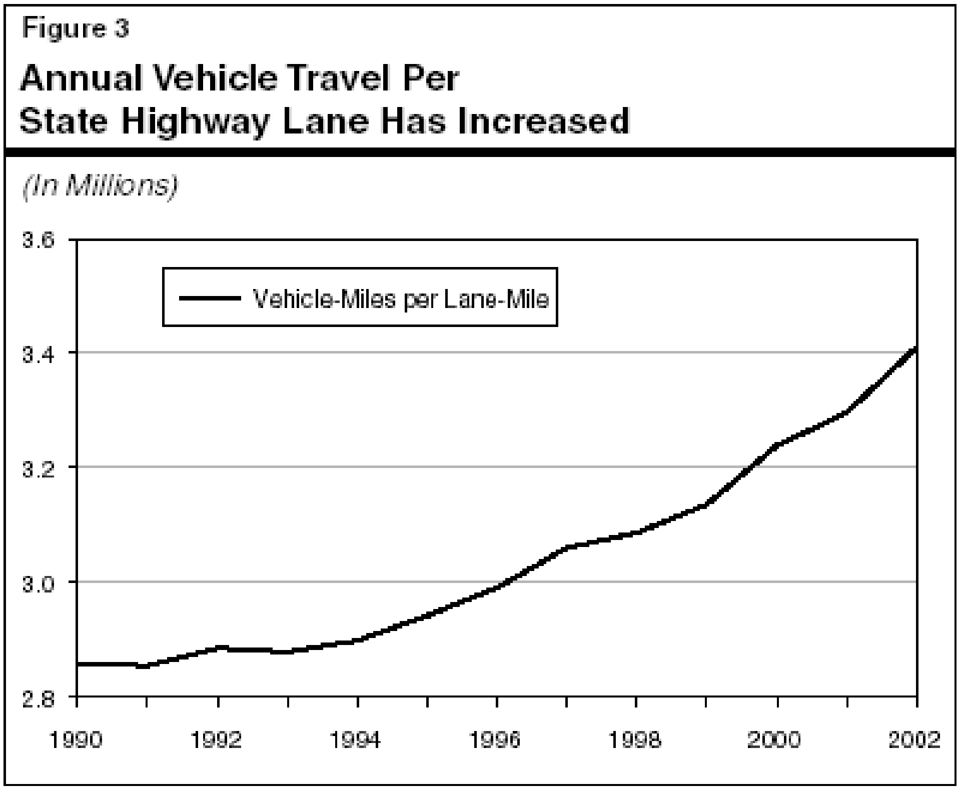


Figure 1.1 Annual Vehicle Travel Per State Highway Lane in California[3]

1.2.2 AHMCT

Caltrans uses a combination of methods, machines, and equipment to maintain the current California highway system and to construct new state highways. Although highway maintenance and construction has come a long way, some of the current methods are outdated, labor intensive, inefficient, environmentally harmful, or a combination of the preceding. As part of the effort to remain a leader in highway maintenance and construction, Caltrans in partnership with the University of California at Davis, sponsored the development of the Advanced Highway Maintenance and Construction Technology Research Center (AHMCT). This center helps Caltrans by providing access to relevant research, evaluating new and existing highway technologies, improving Caltrans public image, and training students and professionals in transportation technologies. AHMCT looks for ways to limit the dangers to highway workers and the general traveling public, improve the efficiency of highway maintenance and construction, improve the reliability of highway infrastructure, minimize congestion delays due to maintenance and construction tasks, and minimize the environmental impacts of maintenance and construction activities.[4]

The main focus of AHMCT is the development of new mechanical technologies that address various problems faced by Caltrans. AHMCT has developed numerous machines including an automated cone layer, many generations of automated crack sealers, advanced snowplows and the automated roadway debris vacuum (ARDVAC). These technologies remove workers from dangerous roadway work, while improving efficiency and minimizing the effects on traffic congestion and the environment with advanced technological solutions.

1.3 Caltrans Vegetation Maintenance Policy

1.3.1 Development of Caltrans Vegetation Maintenance Plan

A statewide committee convened to establish a vegetation control policy for Caltrans in 1987. This policy called for a clear narrow strip next to roadways to control the risk of fire, provide visibility, preserve pavement, and allow an area next to roadways in case of emergency. This policy was heavily dependant on herbicides for vegetation control in the clear strip because herbicides are considered extremely cost effective. This 1987 vegetation control policy was then subjected to a 1992 Environmental Impact Report (EIR) because of the growing concern over herbicide use and its impacts on the environment. This EIR concluded that vegetation in these clear strips should use integrated vegetation management principles to lower the use of herbicides in maintaining the clear strips. The 1992 EIR also pressed the desirability of reducing or eliminating the need to perform vegetation control along roadways. This forced Caltrans to implement alternatives to herbicides in controlling vegetation. As a result of the 1992 EIR, Caltrans set goals for reducing herbicide use by 50% of 1992 levels by 2000 and 80% by 2012.[2] Districts are also restricted to herbicides that are allowed under the new EIR. New herbicides can only be added after conforming to risk assessment procedures. The two main herbicides used by Caltrans are Diuron and Glyphosate, also known and marketed respectively as Diuron FL and Roundup.[1]

1.3.2 Vegetation Maintenance Considerations

The maintenance of the highway rights-of-way and the clear strip, also called the mow strips because they are often mowed, accounts for a large portion of Caltrans' maintenance operations. There are many considerations that must be taken into account when performing this vegetation maintenance. The prime considerations are safety, aesthetics, and compatibility with neighboring lands. It is important to keep native vegetation near the highways because it reduces driver fatigue, improves storm water quality, prevents dust and helps to maintain slope stability, which keeps the roadways intact. Vegetation should be removed if it blocks lines of sight, increases the risks of fire or accidents, or blocks roadside appurtenances such as road signs and guardrails.[5]

Fire risk is one of the main reasons that vegetation maintenance is performed. Hot mufflers, cigarette butts, and random sparks can set fires to roadside vegetation. As such, the mow strip widths are linked primarily to the risks and consequences of fire at that location. The lower the fire risk and fire consequences, the thinner the mow strips. The greater the fire risk and consequences, the larger the mow strip. The strip sizes vary from no vegetation control, 0 meters [0 feet], up to maximum control, 2.5 meters [8 feet], depending on the nature of the roadway, roadside vegetation, traffic conditions, and adjoining land.[5] Caltrans often trims and mows bushes and grass beyond the mow strip so that less fuel is available in some areas that have high fire risk.

Cost is also a big factor in determining the maintenance levels of roadsides. In order to reduce costs, Caltrans has expressed interest in reducing vegetation maintenance, including mow strip maintenance. The first method Caltrans utilizes is the planting of vegetation that is low, slow growing, and requires little maintenance or herbicide use.[5] This vegetation is usually native and would replace annual and perennial weeds that regularly grow on California roadsides. Another method is the use of hardscapes for roadsides that are built such that little maintenance is required. An example of this is a fully paved median or roadside. Vegetation is only able to grow in cracks that occur in the pavement over time so maintenance is only required in this case. Finally, Caltrans sometimes performs little or no maintenance on some areas. With tight budgets, maintenance yards may not maintain low priority areas so that they can focus resources on important projects. Low priority areas are those roadsides with little traffic and low fire risk.

Pesticides, of which herbicides are a part, are also a large consideration for Caltrans. Since the negative environmental effects of pesticides have been well documented, it has become necessary for Caltrans to cut back on their use. Caltrans has thus appropriated funds for research into alternatives to herbicides. This is one of the other reasons Caltrans has stressed the desirability in landscaping roadsides so that little or no maintenance is required. Pesticides will remain a tool for Caltrans because of the low cost and great effectiveness of this process until a new technology can be developed to take its place.

Finally, Caltrans has looked to reduce the amount of injuries sustained performing vegetation maintenance tasks, especially tasks involving manual labor. The injuries that occur on roadsides can be separated into two categories, those that occur due to the work performed and those that are caused by vehicular traffic. The injuries that traffic causes can be limited by shortening the amount of time workers spend on roadways. Those injuries that occur on roadsides can be limited by reducing the amount of manual labor performed. Caltrans seeks to automate tasks and design highway roadsides so that workers perform less manual labor and are exposed to fewer traffic hazards respectively.

1.4 Summary

In order for Caltrans to continue successful vegetation maintenance operations, the agency must improve on a variety of issues. Caltrans should strive to reduce the amount of herbicides and pesticides used, improve worker safety, increase operational efficiency, and lower costs. In order to achieve these goals, Caltrans has commissioned AHMCT to look at current Caltrans vegetation maintenance operations. This report is one part of that analysis and will compare the two main Caltrans means of maintenance, mowing and herbicide spray, to experimental and alternative technologies. This will be a strict engineering analysis, where values are assigned to all positive and negative categories in an attempt to ascertain the most appropriate mechanical methods for vegetation maintenance. An engineering analysis seeks to be as objective as possible given inherent human subjectivity.

One objective of this report is to find the best mechanical methods that would advance the interests of Caltrans in lowering costs, improving worker safety, etc. Another objective is to determine if any alternative vegetation control technologies are competitive with herbicide application. Due to the vastness of Caltrans vegetation maintenance operations, it was necessary to narrow down the scope of this report. Three scenarios — urban pavement, mow strip, guardrails and posts — were chosen as environments to analyze current and alternative vegetation control technologies.

This report is organized in the following manner. The first chapter is an introduction to this project, the project sponsors and Caltrans' vegetation maintenance policy. The second chapter is a description of the three vegetation control scenarios – guardrails & posts, mow strip, and urban vegetation maintenance. The performance of each vegetation control method is compared in each scenario. The third chapter is a description and analysis of the two current and three alternative vegetation control technologies selected for analysis. Cost and daily coverage area calculations are included for each method. Chapter Four includes the concept scoring analysis of the vegetation control methods in each scenario. Finally, Chapter Five includes recommendations and conclusions based on the results of the research and analysis of this report.

The following chapter describes the scenario selection criteria and details the unique aspects of each scenario. The three scenarios selected for analysis are guardrail & post, mow strip, and urban vegetation maintenance. The first two scenarios are self explanatory but urban vegetation requires some explanation. Urban vegetation maintenance involves the maintenance of vegetation that grows in cracks on road surface and between the roadway and other support structures. The specific aspects that make each scenario unique are discussed and the criteria that are most important for successful vegetation control maintenance are discussed. These scenarios are critical for developing the criteria used in the comparison of vegetation control methods.

CHAPTER 2 Vegetation Control Scenarios

This chapter introduces three vegetation scenarios that were selected for analysis. These scenarios were chosen because they present room for improvement and constitute a valid portion of overall vegetation maintenance operations. Details are presented about the selection of each scenario as well as the specific qualities that make each scenario unique. In addition, the vegetation control problems, which affect each scenario, are also outlined. These scenarios provide a basis for the analysis of alternative vegetation methods.

California is home to a highway system possessing maintenance needs as diverse as the ecosystems through which they wind. From the deserts in the south, the forests in the north, the Sierra Mountains in the east, the coastal regions in the west, and the Central Valley, the range in climates and terrain is immense. Although there is great diversity in the California landscape, there are certain vegetation control scenarios that persist throughout large portions of the state. These scenarios include mow strip maintenance, median vegetation control, tree maintenance, post maintenance, shrub encroachment maintenance, gore area maintenance, and drain clearing, to name several. The methods currently employed by Caltrans in each of these scenarios are also fairly constant, consisting mostly of mowing, herbicide application, and manual labor.

This report focuses on three vegetation control scenarios in an attempt to analyze current Caltrans vegetation control activities and compare them to alternative and experimental methods. An engineering analysis will be performed for each control method in each applicable scenario in order to find the most appropriate methods of vegetation control. This analysis may also be used to develop a template that can be applied to the selection of vegetation control methods for scenarios not covered.

There are several criteria used in the selection of scenarios. First, it is important to find problem areas, in which Caltrans desires to improve worker safety, manual labor, pesticide use, cost, and efficiency. It is also desirable to study scenarios that are common to a majority of maintenance districts. In this way, the best vegetation control methods found during this analysis can be implemented throughout the whole state. Caltrans receives the most benefit from the research with the broadest possible application. In addition, it is important to select scenarios for which multiple alternative technologies are currently available. Finally, the list of scenarios was presented to Caltrans personnel at the kick off meeting for this project. Their feedback, along with the selection criteria, helped in the selection of three vegetation maintenance scenarios: guardrails and posts, mow strip, and urban pavement. The following is a description of each scenario including the unique characteristics, safety concerns and current treatment methods.

2.1 Guardrails & Posts

The first scenario comprises the treatment of vegetation around posts, namely those that support signs and guardrails. Guardrails protect the traveling public, wildlife, overpass supports, and other median and roadside buildings. Road signs warn of dangerous conditions, guide traveling public, and display speed limits. It is important to keep the vegetation clear around guardrails for several purposes including lower fire risk, improved sign visibility, more desirable roadside aesthetics, and access for inspection and maintenance. Since guardrails and signs are present on every highway in California, an analysis of this scenario could prove valuable to the majority of maintenance districts in the state. Caltrans employees approved of this scenario in the kick-off meeting held September 12, 2005 in Sacramento. This scenario originally comprised guardrails only, but Caltrans personnel recommended that all posts be included because treatments and obstacles were nearly identical.[6] Figure 2.1 shows a guardrail protecting an overpass support.

Figure 2.1 Guardrail Around Overpass Support

The treatment of posts is a unique problem because posts hamper normal treatment methods. Workers are more exposed to injury during guardrail maintenance. Some of the common risks include cuts from the edges of the guardrail, leg injuries from protruding bolts, slipping over an edge or slope, objects ricocheting off of guardrails, and the danger of crossing the roadway to access service vehicles located on opposite side of road.[21] Currently, the main method for treating the vegetation around posts consists of herbicide spray, tractor mounted boom mowers, and manual weed eaters.[21] Other than the additional risks to Caltrans workers around guardrails, there are other issues. Mowing and herbicide application can miss patches of weeds because of guardrail obstruction. This leads to the necessity for spot treatments by manual foot maintenance. Maintenance crews use either weed eaters or hand sprayed herbicides to treat the remaining vegetation. New treatment methods have been developed or are under development right now that decrease manual labor and increase protection for highway workers. These methods will be analyzed and discussed in detail in the following chapter. Figure 2.2 shows another picture of a highway guardrail.

Figure 2.2 Highway Guardrail

2.2 Mow Strip

The mow strip is the next scenario under analysis. The mow strip is the area of land immediately adjacent to the roadway that is cleared of vegetation. The treatment area can range from about .7 m (2 ft) to approximately 3 m (8 ft) in width.[5] As mentioned previously, the mow strip is important because it provides a barrier between the roadway and the vegetation on the roadside. Cars have room to pull over without setting vegetation ablaze, and cigarettes have to travel further to ignite vegetation. Sometimes, cigarettes can get caught in sparse vegetation in the mow strip area, which can also reduce fire potential. This is also a scenario that applies to nearly every highway roadside in the state. The following picture shows a mow strip recently treated with chemical herbicides.

Figure 2.3 Mow Strip Recently Treated With Chemical Herbicides

The treatment of the mow strip area is performed with a combination of mowing and herbicide application. Caltrans has cut back on herbicide use in accordance with the findings and recommendations of the 1992 environmental impact report.[7] Caltrans has met the 50% herbicide reduction goal in 2000, but it seems that there may be some issues that will make meeting the 80% reduction goal in 2012 difficult. Herbicide use is creeping back up as districts and maintenance yards request higher allowances for noxious weed and fire control.[6] Herbicide reduction has taken a back seat, since noxious weeds and fires have become such a problem.

Since Caltrans uses the majority of its herbicides in maintaining this mow strip, an alternative to herbicides could help dramatically reduce usage. For instance, 61% of all herbicides used in Caltrans District 3 are used in the maintenance of mow strips. When considering vegetation maintenance only (C Family), mow strip herbicide use accounts for 83% of the total.[8] A healthy reduction in mow strip herbicide use would allow Caltrans to satisfy the mandated herbicide levels by 2012. A large herbicide reduction in this scenario would also allow for continued use in fire and invasive weed reduction, while still allowing Caltrans to meet the 80% herbicide reduction in 2012.

Safety is especially important in the maintenance of the mow strip. When working adjacent to the freeway, errant vehicles pose a big threat. Also debris gets kicked up from the roadway and strikes workers. Most mow strip maintenance is now performed from the safety of a truck or riding mower. Little maintenance is performed by hand. It is important that the application of alternative technologies limit worker exposure. In work zones that can leave the workers exposed, safety is very important in considering the effectiveness of the mow strip maintenance method. A mow strip, recently tractor mowed, is shown in Figure 2.4.

Figure 2.4 Mow Strip Recently Mowed

2.3 Urban Vegetation

The final vegetation control scenario involves the maintenance of vegetation that grows in the road surface. Cracks form in asphalt and concrete roadways for a variety of reasons including weathering, traffic, plant roots, and erosion of the roadway bed. These cracks trap dirt and moisture and provide ideal conditions for vegetation growth. In small cracks, little if any vegetation can be present. In more established cracks, growth can be as high as 1-2 meters. This vegetation is unsightly and can damage the aesthetics of the highway. Also, this vegetation can aid in the lengthening and widening of cracks. Finally, the vegetation in the cracks can provide fuel for fires as they are closer to traffic and tossed cigarette butts. These cracks are on every roadway and present a problem for most maintenance crews in the state. Caltrans personnel were also pleased with the selection of this difficult scenario. Vegetation growing up in a roadway crack is visible in Figure 2.5.

Figure 2.5 Urban Vegetation in Roadway Crack

Maintenance of this vegetation is troublesome and usually only occurs when cracks are to be sealed. The use of mowers is not permitted because the mower blades damage the asphalt. Blanket herbicide spray cannot be used because a majority of the spray would settle on the road surface. The herbicide residue could then be washed into the storm drains on subsequent rains, polluting the run off and violating water quality standards. Hand maintenance and manually applied high pressure air are used when sealing the cracks, but these methods require manual labor and increased traffic exposure. The weed growth is further exacerbated by street sweepers. The sweepers remove little dirt or vegetation from the cracks on the roadside matting down most of the vegetation. The sweeper water system used to clean the streets ends up watering the vegetation allowing it to grow further.[6] Regular urban vegetation maintenance should improve aesthetics, lower cost (from widened cracks), and reduce the manual labor required in preparation of crack sealing. In Figure 2.6, vegetation overgrows part of the roadway and sprouts in a longitudinal crack.

Figure 2.6 Urban Vegetation Overhanging Roadway and Sprouting in Crack

2.4 Summary

These three scenarios form the basis for the investigation into Caltrans vegetation maintenance practices. Each scenario is unique in its problems and preferred maintenance methods, but each can be analyzed to reduce costs, manual labor, roadside hazards, fire risk and herbicide use. The purpose of the following analysis will be to address current and alternative methods that are specific to each scenario. The methods that objectively fulfill stated requirements and perform the best under the analysis will be recommended. In addition, it is hopeful that this analysis will foster the development of a general approach that Caltrans can use for other vegetation maintenance scenarios. Finally, recommendations will be made for the implementation of the best performing technologies in each scenario. The following chapter discusses the current and alternative vegetation control methods. All of these methods are then compared with each other to determine the most effective solution for each scenario. The final goal is to determine which, if any, alternative methods can compete with existing Caltrans maintenance methods.

CHAPTER 3 Vegetation Control Methods

This chapter focuses on current and alternative mechanical methods for vegetation control. Currently, dozens of alternative vegetation control methods are available to various fields. These alternative methods also exist in various stages of development from prototype to retail equipment. This chapter specifically addresses the most viable alternative methods with an emphasis on mechanical technologies. The technology descriptions are important in analyzing the maintenance tasks of each scenario, in order to determine if an alternative method is better suited for a specific maintenance task than a traditional method. This chapter provides the basis for the next chapter on scenario analysis.

The following is a description of technologies that pertain to the treatment of existing vegetation. Within the following descriptions, however, some methods are left out of the discussion. Namely, the planning and landscaping solutions that minimize the amount of required vegetation maintenance are left out. Some of these methods include hardscaping of rights-of-way, weed mats, and the revegetation of native species. Although these methods are important to the overall vegetation maintenance effort, they are long term solutions that require significant capital investment, time, and planning. The purpose of this analysis is to address more immediate needs of Caltrans including fire risk, worker safety, cost, efficiency, public perception, and most importantly the reduction of herbicides. Therefore this report considers only maintenance operations and alternative maintenance methods and not planning, construction, or landscaping methods.

A more comprehensive look at the various vegetation control methods can be found in McPhee's Thesis.[9]

Cost and efficiency are both good ways to compare the following methods to each other. Unfortunately it is hard to quantify a total cost for each method. Caltrans has supplied some cost estimates for mowing and herbicide application, but the costs are not itemized. It is impossible to figure out the individual costs from Caltrans' figures. Therefore, calculations are inserted into each section to show a bare cost for each method. The following calculations are rough estimates of the basic costs of performing each maintenance activity as well as some theoretical calculations of automated alternative technologies suitable for roadway use. The constants were gathered from Caltrans sources and other vegetation sources listed above. Other sources include various pump and engine manufacturers for engine efficiencies and fuel consumption, and the consumer price index for the cost of gasoline and propane. These calculations do not take into account a variety of additional activities that inflate costs including road closures, shadow vehicles, worker injuries, damage to the environment, long term health problems related to chemical exposure, weather, initial equipment purchase, cleaning, and maintenance costs. This serves only as a rough estimate of the per hectare (acre) costs of various vegetation control methods, and is only used as a comparison between them.

3.1 Current Vegetation Control Methods

Caltrans currently utilizes two main methods of vegetation maintenance — mowing and herbicide application. These two methods account for the majority of all vegetation maintenance operations. Caltrans also relies on a number of other methods such as controlled burns, grazing and hand operations. However, these methods are used mainly in special circumstances and account for a small portion of overall maintenance. Hand maintenance is another method that Caltrans utilizes in controlling roadside vegetation. This method, however, is very dangerous because it exposes workers to traffic. This method also uses lots of man hours, is very costly at about $1,200 per hectare ($500 per acre),[9] and produces higher injury rates than mechanical maintenance methods.[6] Accordingly, Caltrans wants to limit the amount of hand maintenance performed. Hand maintenance, therefore, is not considered a viable method for this analysis. Alternately, those methods that reduce the amount of manual labor required receive beneficial rankings.

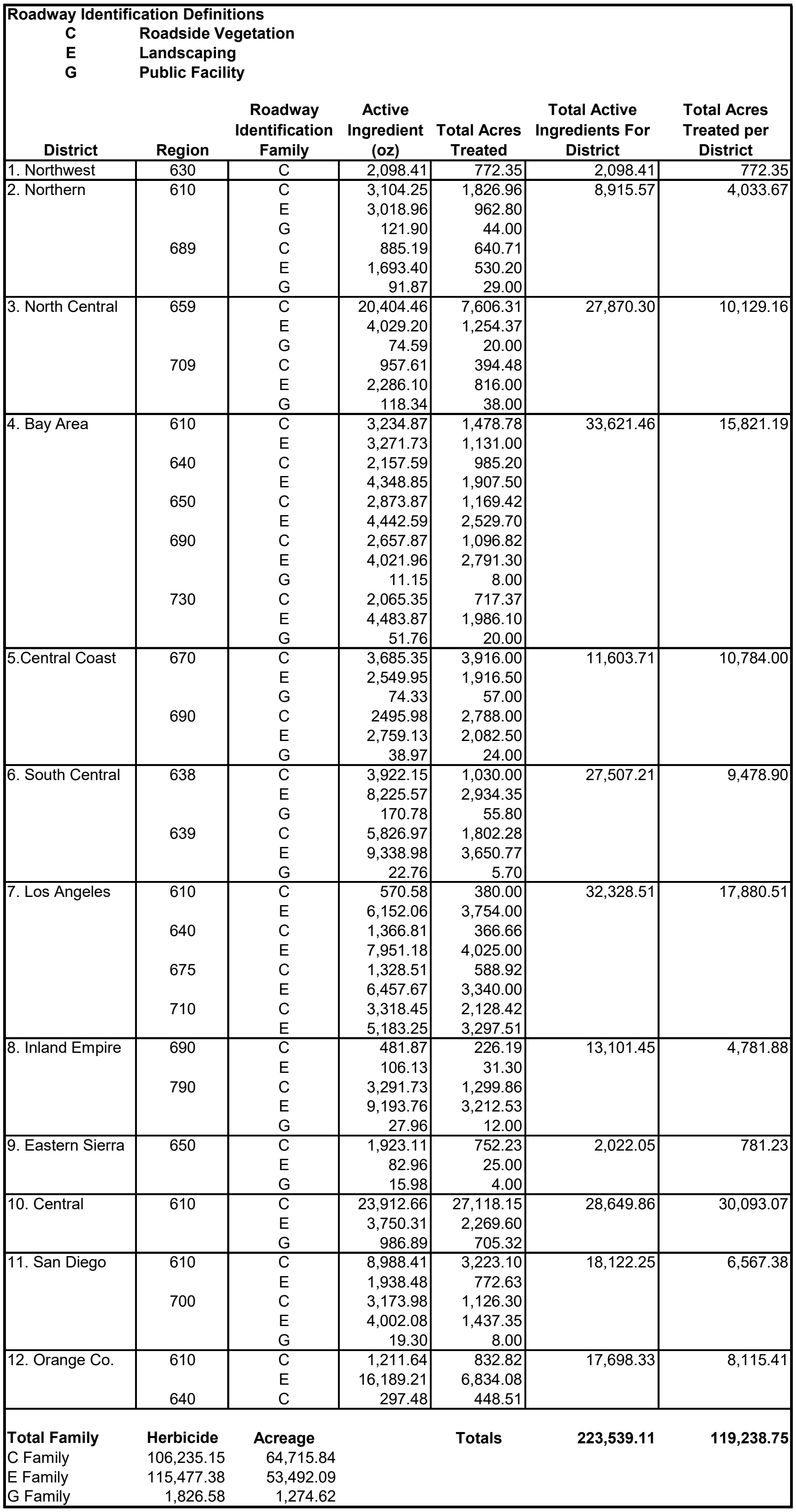
A growing portion of roadside vegetation does not receive any maintenance at all. Due to the backlog of work orders for roadways and roadsides, Caltrans maintenance yards are not able to keep up with all aspects of maintenance. Lower priority roads are occasionally neglected.[3] A backlog of maintenance tasks has also been growing considerably as Caltrans struggles under budget limitations. The cost and effect of neglecting roadside vegetation is also discussed below.

3.1.1 Herbicide Application

Herbicides are very important to vegetation maintenance throughout California. Caltrans mainly uses herbicides to control invasive weeds and lower fire risk along freeways and rights of way. Herbicides are used in the maintenance of gore strips, mow strips, guardrails, shrub encroachment, and posts. Herbicides attack plants in various ways to inhibit plant growth. Some herbicides are selective in that they only affect certain types or species of plants. Other herbicides are non-selective, and attack all types of vegetation. Caltrans uses automated 3,800 liter (1000 gallon) spray trucks to apply the herbicides. Depending on the spray rate these trucks can cover 200-800 hectares (500-2000 acres) per tank.[11] Figure 3.1 shows a picture of an herbicide spray truck in operation.

Figure 3.1 Herbicide Spray Truck

Herbicide usage has recently come under fire for its adverse environmental and health effects. In 1992, Caltrans set goals for curbing herbicide application in order to reduce the impact of these harmful side effects. Caltrans met the 50% reduction goal in 2000, but as previously mentioned, is not on track to meet the 80% reduction goal in 2012.[7] Caltrans used approximately 6,800 liters (1,800 gallons) of active ingredient, on 48,500 hectares (120,000 acres) of roadsides and rights-of-way in the 2004/2005 fiscal year.[8] This does not account for the total material sprayed along roadsides, only the active ingredients. The majority of spraying occurs in the Central Valley, the San Francisco Bay Area, and the greater Los Angeles area. Figure 3.2 shows the use of all herbicide active ingredients per District. Each District uses a different mixture of chemicals depending on the types of vegetation and the climate, but only the total active ingredient usage is shown.



Roadway Identification Definitions

C **Roadside Vegetation**
E **Landscaping**
G **Public Facility**

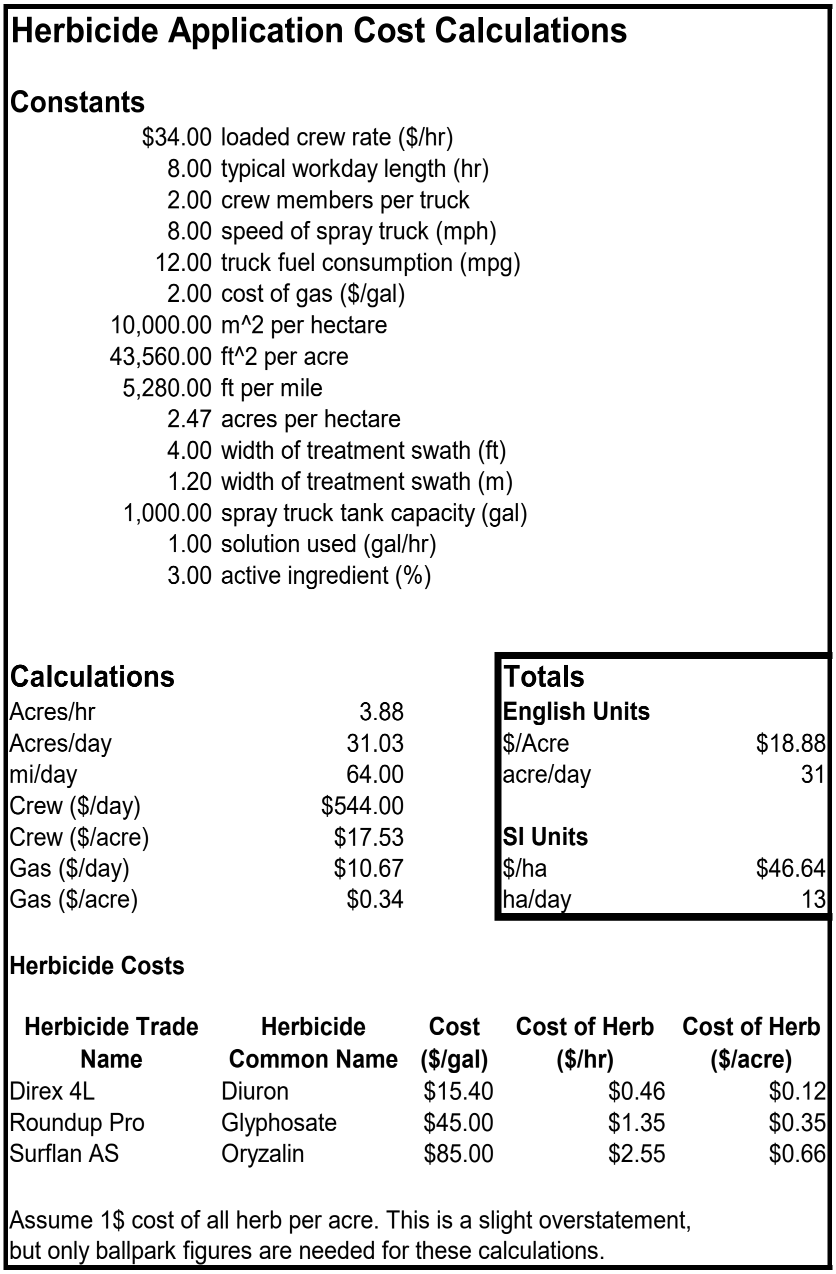
District	Region	Roadway Identification Family	Active Ingredient (oz)	Total Acres Treated	Total Active Ingredients For District	Total Acres Treated per District
1. Northwest	630	C	2,098.41	772.35	2,098.41	772.35
2. Northern	610	C	3,104.25	1,826.96	8,915.57	4,033.67
		E	3,018.96	962.80		
		G	121.90	44.00		
	689	C	885.19	640.71		
		E	1,693.40	530.20		
		G	91.87	29.00		
3. North Central	659	C	20,404.46	7,606.31	27,870.30	10,129.16
		E	4,029.20	1,254.37		
		G	74.59	20.00		
	709	C	957.61	394.48		
		E	2,286.10	816.00		
		G	118.34	38.00		
4. Bay Area	610	C	3,234.87	1,478.78	33,621.46	15,821.19
		E	3,271.73	1,131.00		
	640	C	2,157.59	985.20		
		E	4,348.85	1,907.50		
	650	C	2,873.87	1,169.42		
		E	4,442.59	2,529.70		
	690	C	2,657.87	1,096.82		
		E	4,021.96	2,791.30		
		G	11.15	8.00		
	730	C	2,065.35	717.37		
		E	4,483.87	1,986.10		
		G	51.76	20.00		
5.Central Coast	670	C	3,685.35	3,916.00	11,603.71	10,784.00
		E	2,549.95	1,916.50		
		G	74.33	57.00		
	690	C	2495.98	2,788.00		
		E	2,759.13	2,082.50		
		G	38.97	24.00		
6. South Central	638	C	3,922.15	1,030.00	27,507.21	9,478.90
		E	8,225.57	2,934.35		
		G	170.78	55.80		
	639	C	5,826.97	1,802.28		
		E	9,338.98	3,650.77		
		G	22.76	5.70		
7. Los Angeles	610	C	570.58	380.00	32,328.51	17,880.51
		E	6,152.06	3,754.00		
	640	C	1,366.81	366.66		
		E	7,951.18	4,025.00		
	675	C	1,328.51	588.92		
		E	6,457.67	3,340.00		
	710	C	3,318.45	2,128.42		
		E	5,183.25	3,297.51		
8. Inland Empire	690	C	481.87	226.19	13,101.45	4,781.88
		E	106.13	31.30		
	790	C	3,291.73	1,299.86		
		E	9,193.76	3,212.53		
		G	27.96	12.00		
9. Eastern Sierra	650	C	1,923.11	752.23	2,022.05	781.23
		E	82.96	25.00		
		G	15.98	4.00		
10. Central	610	C	23,912.66	27,118.15	28,649.86	30,093.07
		E	3,750.31	2,269.60		
		G	986.89	705.32		
11. San Diego	610	C	8,988.41	3,223.10	18,122.25	6,567.38
		E	1,938.48	772.63		
	700	C	3,173.98	1,126.30		
		E	4,002.08	1,437.35		
		G	19.30	8.00		
12. Orange Co.	610	C	1,211.64	832.82	17,698.33	8,115.41
		E	16,189.21	6,834.08		
	640	C	297.48	448.51		

Total Family	**Herbicide**	**Acreage**		**Totals**	**223,539.11**	**119,238.75**
C Family	106,235.15	64,715.84				
E Family	115,477.38	53,492.09				
G Family	1,826.58	1,274.62				

Figure 3.2 Caltrans Herbicide Usage Statistics for Fiscal Year 2004/2005[8]

Figure 3.3 shows the approximate bare cost for Caltrans mowing operations. Caltrans does have cost estimates that are significantly higher but, as mentioned previously, the costs include many unknown additional factors that would make comparison with alternative control methods difficult. The calculations for every calculation are simple multiplication, division, and unit conversion. For those units that require constant water or solution refills, calculations are made for various refill times.

The crew cost per day is the hourly crew cost times the number of workers times eight hours per day. The area per hour is the hourly speed of the application times the application width. The gasoline cost is the fuel price per gallon times the daily travel distance divided by the fuel economy of the vehicle. All of the other calculations are unit conversions or simple arithmetic. No complex formulas are used in the cost calculations.



Herbicide Application Cost Calculations

Constants

$34.00	loaded crew rate ($/hr)
8.00	typical workday length (hr)
2.00	crew members per truck
8.00	speed of spray truck (mph)
12.00	truck fuel consumption (mpg)
2.00	cost of gas ($/gal)
10,000.00	m^2 per hectare
43,560.00	ft^2 per acre
5,280.00	ft per mile
2.47	acres per hectare
4.00	width of treatment swath (ft)
1.20	width of treatment swath (m)
1,000.00	spray truck tank capacity (gal)
1.00	solution used (gal/hr)
3.00	active ingredient (%)

Calculations

Acres/hr	3.88
Acres/day	31.03
mi/day	64.00
Crew ($/day)	$544.00
Crew ($/acre)	$17.53
Gas ($/day)	$10.67
Gas ($/acre)	$0.34

Totals

English Units	
$/Acre	$18.88
acre/day	31
SI Units	
$/ha	$46.64
ha/day	13

Herbicide Costs

Herbicide Trade Name	Herbicide Common Name	Cost ($/gal)	Cost of Herb ($/hr)	Cost of Herb ($/acre)
Direx 4L	Diuron	$15.40	$0.46	$0.12
Roundup Pro	Glyphosate	$45.00	$1.35	$0.35
Surflan AS	Oryzalin	$85.00	$2.55	$0.66

Assume 1$ cost of all herb per acre. This is a slight overstatement, but only ballpark figures are needed for these calculations.

Figure 3.3 Herbicide Cost Calculations for Roadside Application

Benefits of Chemical Herbicide Application

The benefits of using herbicides are the cost and effectiveness of application. Herbicides are considered to be the cheapest, most cost efficient manner for vegetation control by Caltrans. The cost of herbicides is usually less than a $2 per hectare ($1 per acre). As such, the main cost in application is the labor cost. Including all the costs, herbicides are still the cheapest and most efficient method of vegetation control. This cost is addressed in more detail in the next chapter. Caltrans currently employs the use of herbicide spray trucks that hold a large reservoir of herbicide solution and dispense it from spray nozzles onto roadside vegetation. This is generally a safer method than hand application as highway workers are safer inside the spray trucks than if they were on foot on the roadside.

Herbicide application is also efficient. Herbicides are applied, via spray trucks, in a blanket spray onto the target vegetation. Spray truck speed depends on terrain and traffic flow but generally move about 13 k/hr (8mph).[11] Given a spray swath of 1.2 m (4 ft) a typical spray truck covers 13 ha (31 acres) per day. Two applications of herbicide are often efficient enough to effectively treat weeds for a season. The two treatments consist of an early pre-emergent herbicide and then of a post emergent herbicide later on. Additional treatments are sometimes required in areas with dense noxious weed growth, extreme fire risk, or poor public perception.[11]

Finally, herbicides come in many different formulations and have a wide array of applications. There are pre-emergent herbicides that prevent the growth of unwanted vegetation, and post-emergent herbicides that treat existing vegetation. Some chemicals specifically target certain types of vegetation, allowing desirable vegetation to grow. These are called selective herbicides. Furthermore, non-selective herbicides affect all vegetation. At last, herbicides can be residual or non-residual. Residual herbicides are used for pre-emergent treatment and for total control of an area (bare ground). Non-residual herbicides treat current plant growth but do not remain in the soil for long periods of time allowing growth of desired vegetation. Almost any vegetation problem can be treated with a combination of these herbicide types. Herbicides are a flexible, effective, and cost efficient method for vegetation control, but there is a price to pay for these benefits.

Drawbacks of Chemical Herbicide Application

There are many issues with herbicide use, including environmental, health, and logistical. Logistical problems are the least contentious of all. Herbicide spray trucks hamper traffic flow because they operate on the shoulder and require a lane closure. This may cause traffic delays. Herbicides are also sensitive to poor weather conditions. Because herbicides are sprayed, wind and rain can disrupt the spray and dilute the solution so that it is no longer effective. Even light wind and rain may limit the efficacy of the herbicide treatment.[11] In addition, safety equipment is required for application because of the health risks associated with herbicides. Spray equipment must be cleaned and maintained regularly to insure adequate safety. This increases the cost and cuts down of the efficiency of the method. Finally, herbicides leave treated vegetation black and brown upon death leaving unsightly remnants behind. These drawbacks to herbicide usage are rarely mentioned though because opposition to herbicide usage is focused on environmental and health effects.

Chemical herbicides attack vegetation by damaging the plant cell structures and disrupting internal processes necessary for life. These same chemicals may have a similar effect on desirable vegetation, animals, and humans. Many studies have documented the adverse consequences of herbicide use. Chemicals contained in prominent herbicides have been found in wells, lakes, streams, soil, air, animals, and humans in California. Caltrans acknowledges these risks and has conducted studies to determine the reach of hazardous chemicals. For instance, Caltrans monitored storm water in 1999 and 2000 in District 1 and found pre-emergent herbicides in the runoff.[10]

In a similar experiment conducted on air contamination, it was found that 14% to 78% of the herbicide Glyphosate drifted from the intended target.[1] Figures 3.4 through 3.8 outline the negative effects on the humans, animals, and the environment of common active ingredients used in California.

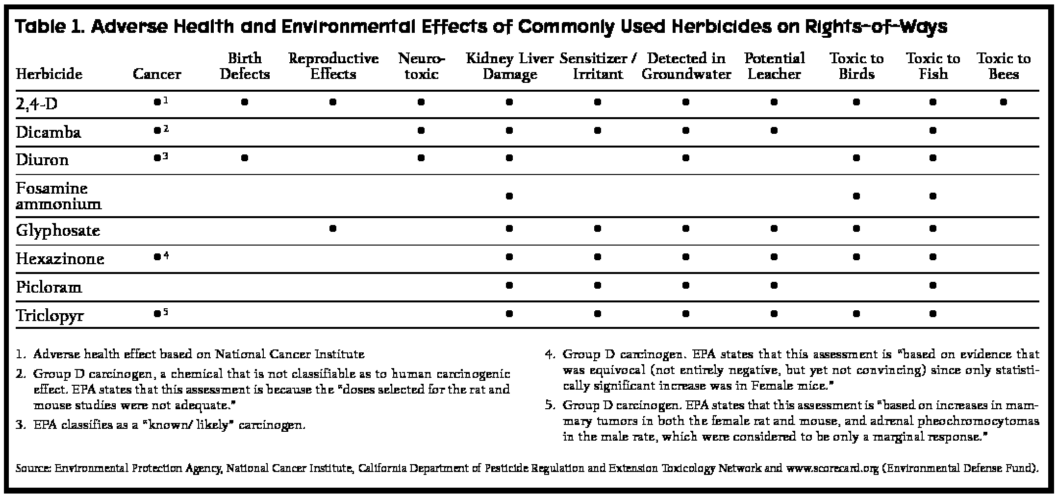


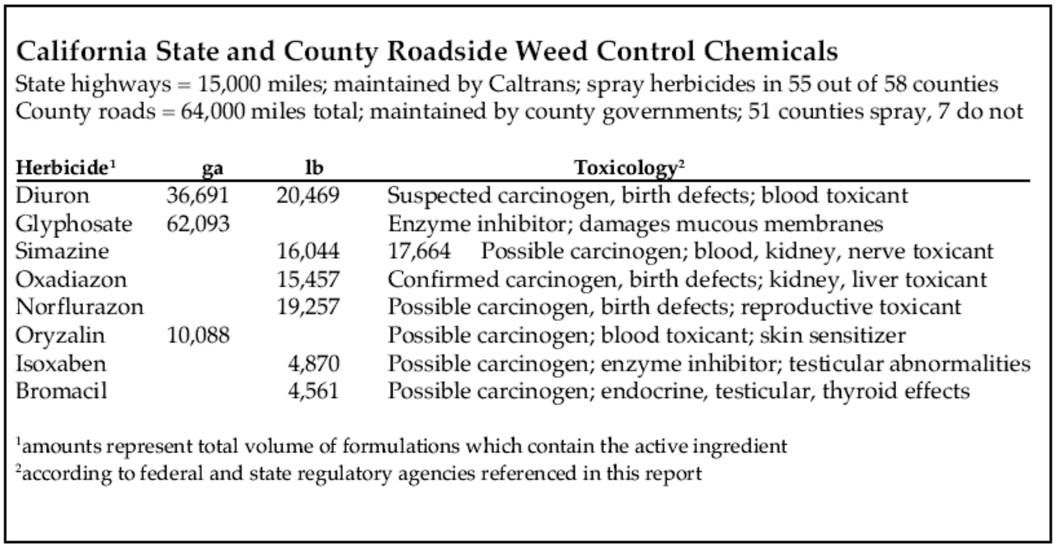
Table 1. Adverse Health and Environmental Effects of Commonly Used Herbicides on Rights-of-Ways

Herbicide	Cancer	Birth Defects	Reproductive Effects	Neuro-toxic	Kidney Liver Damage	Sensitizer / Irritant	Detected in Groundwater	Potential Leacher	Toxic to Birds	Toxic to Fish	Toxic to Bees
2,4-D	●[1]	●	●	●	●	●	●	●	●	●	●
Dicamba	●[2]			●	●	●	●	●		●	
Diuron	●[3]	●		●	●		●		●	●	
Fosamine ammonium					●				●	●	
Glyphosate			●		●	●	●	●	●	●	
Hexazinone	●[4]				●	●	●	●	●	●	
Picloram					●	●	●	●		●	
Triclopyr	●[5]				●	●	●	●	●	●	

1. Adverse health effect based on National Cancer Institute
2. Group D carcinogen, a chemical that is not classifiable as to human carcinogenic effect. EPA states that this assessment is because the "doses selected for the rat and mouse studies were not adequate."
3. EPA classifies as a "known/ likely" carcinogen.
4. Group D carcinogen. EPA states that this assessment is "based on evidence that was equivocal (not entirely negative, but yet not convincing) since only statistically significant increase was in Female mice."
5. Group D carcinogen. EPA states that this assessment is "based on increases in mammary tumors in both the female rat and mouse, and adrenal pheochromocytomas in the male rate, which were considered to be only a marginal response."

Source: Environmental Protection Agency, National Cancer Institute, California Department of Pesticide Regulation and Extension Toxicology Network and www.scorecard.org (Environmental Defense Fund).

Figure 3.4 Adverse Health Effects[22]



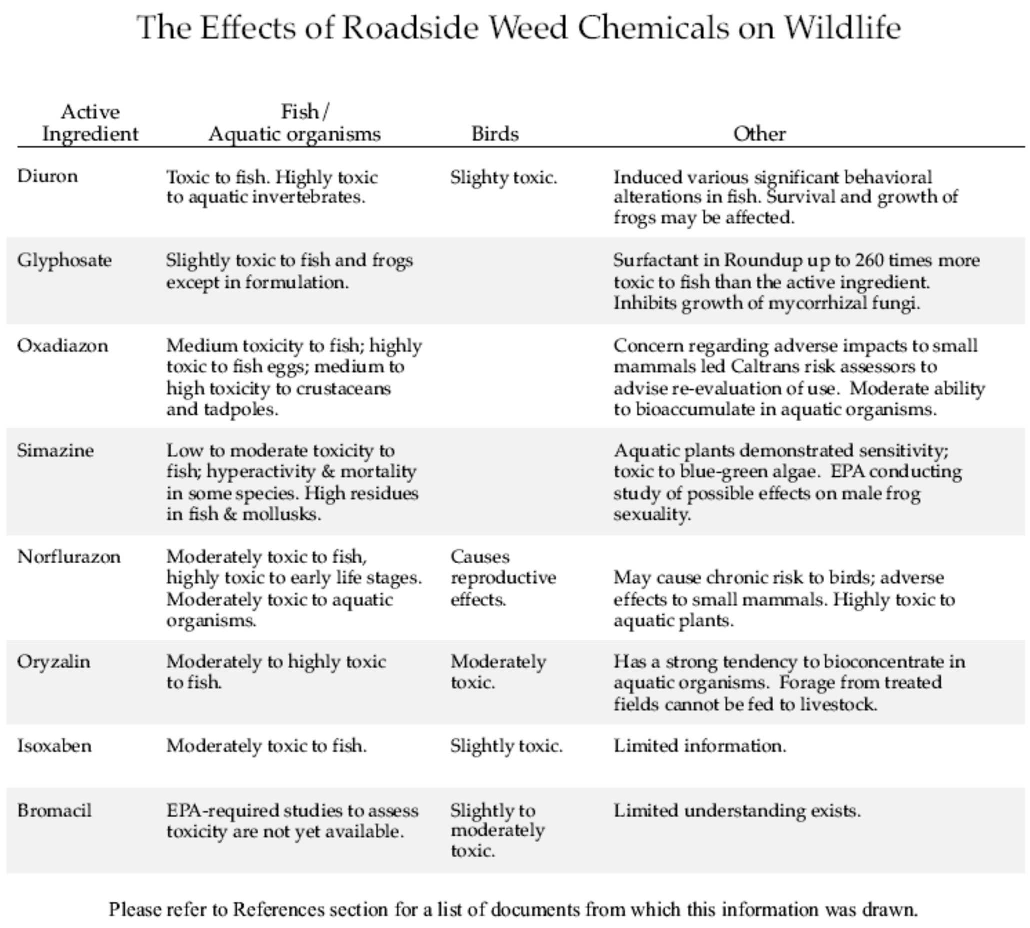
California State and County Roadside Weed Control Chemicals

State highways = 15,000 miles; maintained by Caltrans; spray herbicides in 55 out of 58 counties
County roads = 64,000 miles total; maintained by county governments; 51 counties spray, 7 do not

Herbicide[1]	ga	lb	Toxicology[2]
Diuron	36,691	20,469	Suspected carcinogen, birth defects; blood toxicant
Glyphosate	62,093		Enzyme inhibitor; damages mucous membranes
Simazine		16,044	17,664 Possible carcinogen; blood, kidney, nerve toxicant
Oxadiazon		15,457	Confirmed carcinogen, birth defects; kidney, liver toxicant
Norflurazon		19,257	Possible carcinogen, birth defects; reproductive toxicant
Oryzalin	10,088		Possible carcinogen; blood toxicant; skin sensitizer
Isoxaben		4,870	Possible carcinogen; enzyme inhibitor; testicular abnormalities
Bromacil		4,561	Possible carcinogen; endocrine, testicular, thyroid effects

[1] amounts represent total volume of formulations which contain the active ingredient
[2] according to federal and state regulatory agencies referenced in this report

Figure 3.5 Usage and Adverse Health Effects of Common Caltrans Chemicals[1]

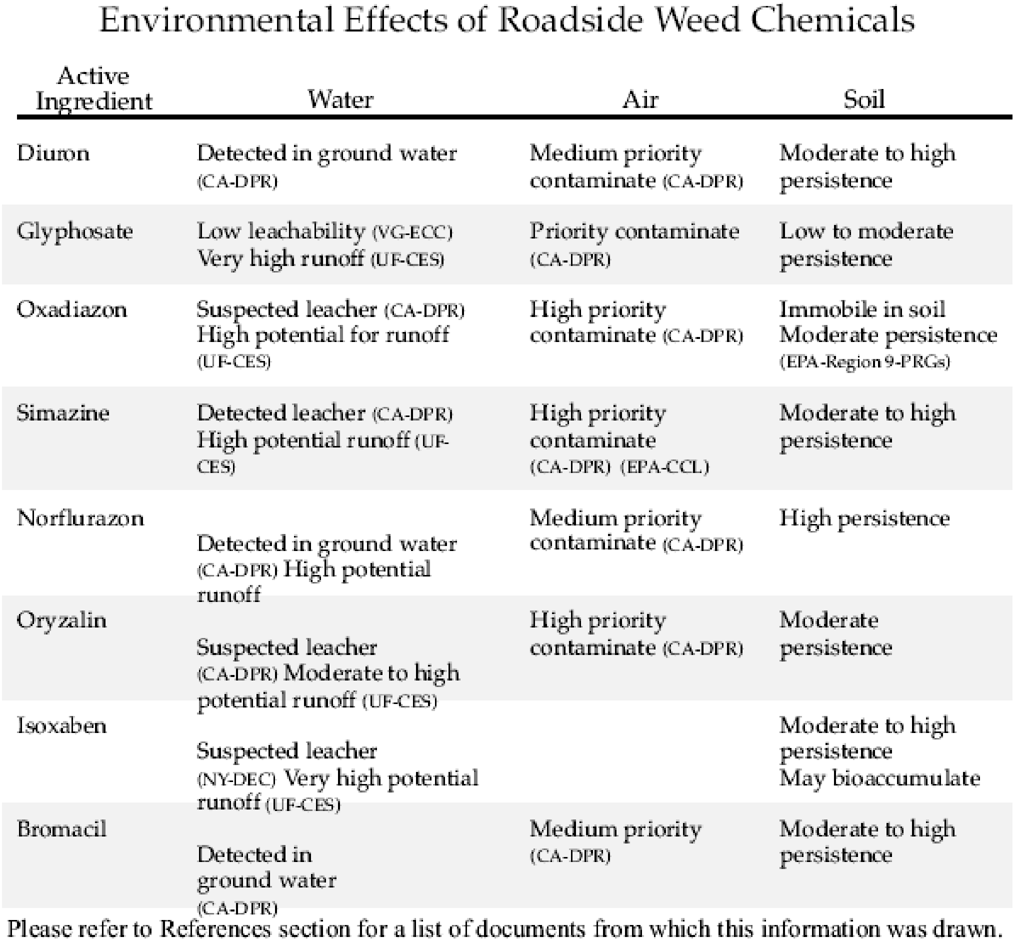


The Effects of Roadside Weed Chemicals on Wildlife

Active Ingredient	Fish / Aquatic organisms	Birds	Other
Diuron	Toxic to fish. Highly toxic to aquatic invertebrates.	Slighty toxic.	Induced various significant behavioral alterations in fish. Survival and growth of frogs may be affected.
Glyphosate	Slightly toxic to fish and frogs except in formulation.		Surfactant in Roundup up to 260 times more toxic to fish than the active ingredient. Inhibits growth of mycorrhizal fungi.
Oxadiazon	Medium toxicity to fish; highly toxic to fish eggs; medium to high toxicity to crustaceans and tadpoles.		Concern regarding adverse impacts to small mammals led Caltrans risk assessors to advise re-evaluation of use. Moderate ability to bioaccumulate in aquatic organisms.
Simazine	Low to moderate toxicity to fish; hyperactivity & mortality in some species. High residues in fish & mollusks.		Aquatic plants demonstrated sensitivity; toxic to blue-green algae. EPA conducting study of possible effects on male frog sexuality.
Norflurazon	Moderately toxic to fish, highly toxic to early life stages. Moderately toxic to aquatic organisms.	Causes reproductive effects.	May cause chronic risk to birds; adverse effects to small mammals. Highly toxic to aquatic plants.
Oryzalin	Moderately to highly toxic to fish.	Moderately toxic.	Has a strong tendency to bioconcentrate in aquatic organisms. Forage from treated fields cannot be fed to livestock.
Isoxaben	Moderately toxic to fish.	Slightly toxic.	Limited information.
Bromacil	EPA-required studies to assess toxicity are not yet available.	Slightly to moderately toxic.	Limited understanding exists.

Please refer to References section for a list of documents from which this information was drawn.

Figure 3.6 Effects of Roadside Weed Chemicals on Wildlife[1]

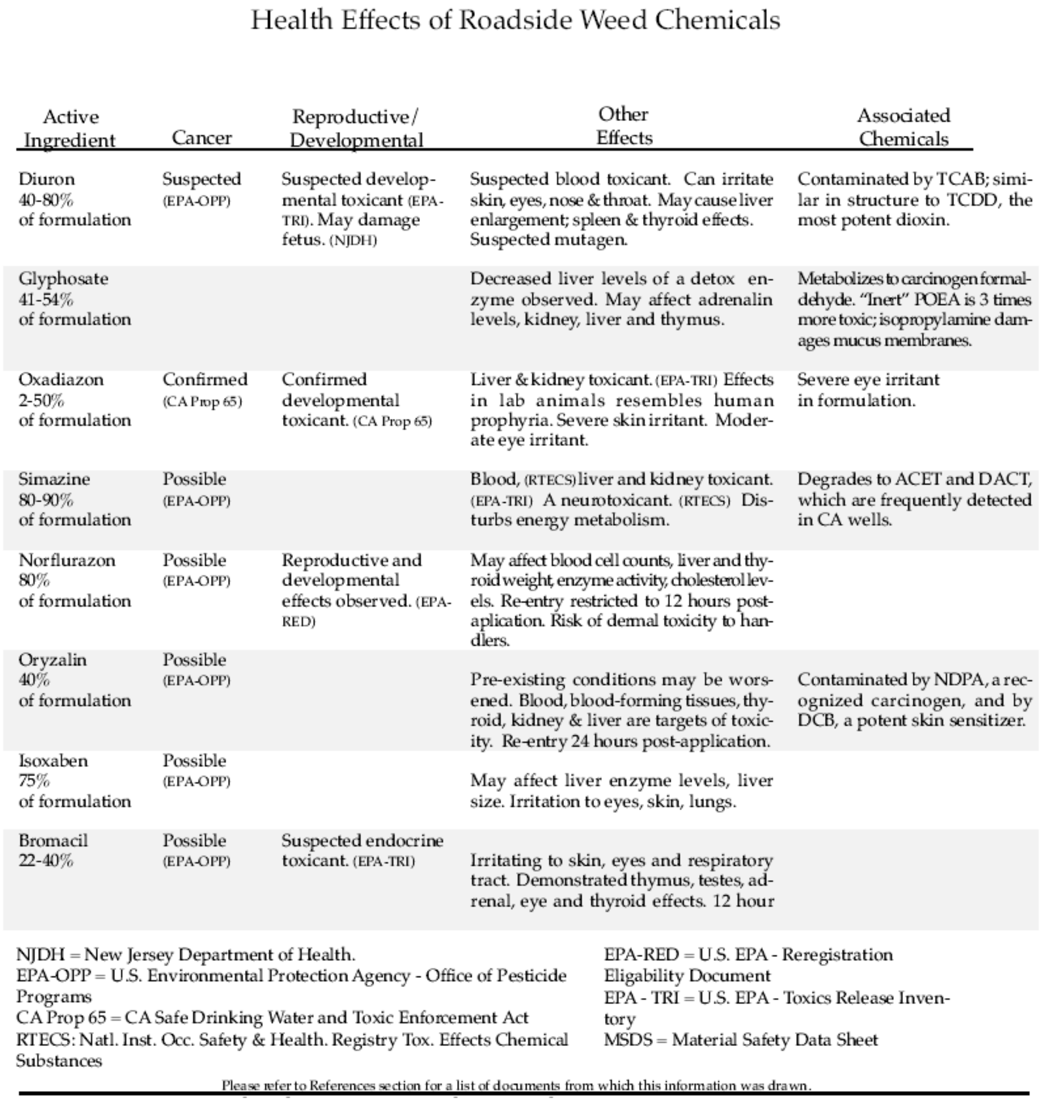


Environmental Effects of Roadside Weed Chemicals

Active Ingredient	Water	Air	Soil
Diuron	Detected in ground water (CA-DPR)	Medium priority contaminate (CA-DPR)	Moderate to high persistence
Glyphosate	Low leachability (VG-ECC) Very high runoff (UF-CES)	Priority contaminate (CA-DPR)	Low to moderate persistence
Oxadiazon	Suspected leacher (CA-DPR) High potential for runoff (UF-CES)	High priority contaminate (CA-DPR)	Immobile in soil Moderate persistence (EPA-Region 9-PRGs)
Simazine	Detected leacher (CA-DPR) High potential runoff (UF-CES)	High priority contaminate (CA-DPR) (EPA-CCL)	Moderate to high persistence
Norflurazon	Detected in ground water (CA-DPR) High potential runoff	Medium priority contaminate (CA-DPR)	High persistence
Oryzalin	Suspected leacher (CA-DPR) Moderate to high potential runoff (UF-CES)	High priority contaminate (CA-DPR)	Moderate persistence
Isoxaben	Suspected leacher (NY-DEC) Very high potential runoff (UF-CES)		Moderate to high persistence May bioaccumulate
Bromacil	Detected in ground water (CA-DPR)	Medium priority (CA-DPR)	Moderate to high persistence

Please refer to References section for a list of documents from which this information was drawn.

Figure 3.7 Environmental Effects of Roadside Weed Chemicals[1]



Health Effects of Roadside Weed Chemicals

Active Ingredient	Cancer	Reproductive/ Developmental	Other Effects	Associated Chemicals
Diuron 40-80% of formulation	Suspected (EPA-OPP)	Suspected developmental toxicant (EPA-TRI). May damage fetus. (NJDH)	Suspected blood toxicant. Can irritate skin, eyes, nose & throat. May cause liver enlargement; spleen & thyroid effects. Suspected mutagen.	Contaminated by TCAB; similar in structure to TCDD, the most potent dioxin.
Glyphosate 41-54% of formulation			Decreased liver levels of a detox enzyme observed. May affect adrenalin levels, kidney, liver and thymus.	Metabolizes to carcinogen formaldehyde. "Inert" POEA is 3 times more toxic; isopropylamine damages mucus membranes.
Oxadiazon 2-50% of formulation	Confirmed (CA Prop 65)	Confirmed developmental toxicant. (CA Prop 65)	Liver & kidney toxicant. (EPA-TRI) Effects in lab animals resembles human prophyria. Severe skin irritant. Moderate eye irritant.	Severe eye irritant in formulation.
Simazine 80-90% of formulation	Possible (EPA-OPP)		Blood, (RTECS) liver and kidney toxicant. (EPA-TRI) A neurotoxicant. (RTECS) Disturbs energy metabolism.	Degrades to ACET and DACT, which are frequently detected in CA wells.
Norflurazon 80% of formulation	Possible (EPA-OPP)	Reproductive and developmental effects observed. (EPA-RED)	May affect blood cell counts, liver and thyroid weight, enzyme activity, cholesterol levels. Re-entry restricted to 12 hours post-aplication. Risk of dermal toxicity to handlers.	
Oryzalin 40% of formulation	Possible (EPA-OPP)		Pre-existing conditions may be worsened. Blood, blood-forming tissues, thyroid, kidney & liver are targets of toxicity. Re-entry 24 hours post-application.	Contaminated by NDPA, a recognized carcinogen, and by DCB, a potent skin sensitizer.
Isoxaben 75% of formulation	Possible (EPA-OPP)		May affect liver enzyme levels, liver size. Irritation to eyes, skin, lungs.	
Bromacil 22-40%	Possible (EPA-OPP)	Suspected endocrine toxicant. (EPA-TRI)	Irritating to skin, eyes and respiratory tract. Demonstrated thymus, testes, adrenal, eye and thyroid effects. 12 hour	

NJDH = New Jersey Department of Health.
EPA-OPP = U.S. Environmental Protection Agency - Office of Pesticide Programs
CA Prop 65 = CA Safe Drinking Water and Toxic Enforcement Act
RTECS: Natl. Inst. Occ. Safety & Health. Registry Tox. Effects Chemical Substances
EPA-RED = U.S. EPA - Reregistration Eligability Document
EPA - TRI = U.S. EPA - Toxics Release Inventory
MSDS = Material Safety Data Sheet

Please refer to References section for a list of documents from which this information was drawn.

Figure 3.8 Health Effects of Roadside Weed Chemicals[1]

While not every herbicide is equally harmful to all life, there persists a general health concern with chemical herbicide application. Highway workers, CHP officers, and motorists are at the highest risks of exposure. In a study conducted in 1986 by University of California researcher Dr. Neil Maizlish, it was found that Caltrans workers had 68% increase in brain cancers and a 62% increase in blood cancers over the national average[1]. This increase could be due to increased exposure to herbicides, motor vehicle exhaust, a combination of the two, or other mitigating factors. The study was later rebuffed by Caltrans, however, because the cancer cases were concentrated in office workers. A shortfall of the study was that only the latest position held by the Caltrans employee was listed in the study. It is possible that many employees started initially on the roadway, were exposed to herbicides, and then moved up into office jobs as they advanced in their careers. In addition, management is usually composed of the older and more mature employees. Cancer usually forms after many years of low level exposure and thus the cancer would be more prevalent among older employees. This does not conclusively prove nor disprove the long term cancer risks of herbicide use. However, given the results of this first study, a more comprehensive analysis should be undertaken.

Alternate Methods of Chemical Herbicide Application

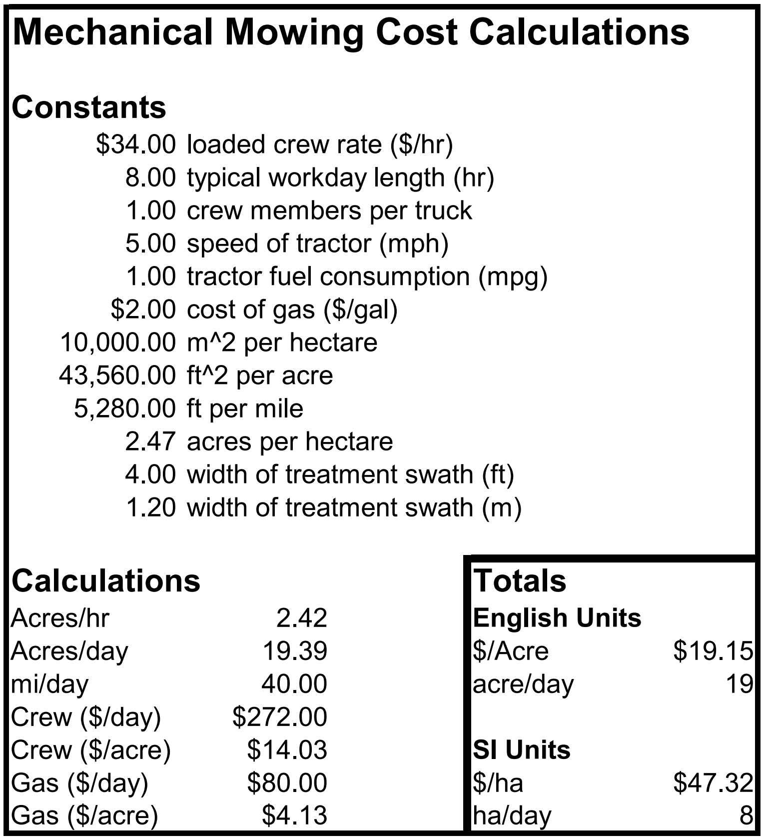
Herbicide application is a straight forward process. In most cases herbicide is sprayed from a truck through a nozzle in a blanket application process. Nevertheless, this process has some shortfalls. As mentioned above, herbicides can drift out of the target area. In addition, the tops of the leaves are more resistant to herbicides than the underside. The alternative methods that were developed seek to minimize the amount of herbicide through different application methods. This reduces the impact on the environment and the cost per application. Some of the available alternative methods are the Weed/Sweep© grit mower and the Sidewinder® with WeedSeeker® automated spot spray system.

3.1.2 Mechanical Mowing

Mechanical mowing of rights-of-way and medians is an important vegetation maintenance tool. For Caltrans, mowing is the only other major vegetation control method than herbicide application. Mechanical cutting is an integral component of Caltrans' vegetation control plan for a variety of reasons. The mowing of roadside vegetation reduces the risk of fire, improves aesthetics, improves visibility and reduces the growth and spread of noxious weeds. There are many types of mowers and cutters that perform the task of cutting existing vegetation, from hand held lawn mowers to industrial farm equipment. Not all of these are suitable for the roadside environment, however. Caltrans relies mainly on tractor mounted mowers for median and right-of-way vegetation control.[11] Tractors are versatile and can be fitted with a variety of mow attachments. There are many types of mowers including rotary blades, discs, flails, cords/cables, and cutter bars. Caltrans mainly uses blades and discs for most mowing operations. Figure 3.9 shows a tractor mounted boom mower. This is the standard mowing setup used by Caltrans.

Figure 3.9 Tractor Mounted Boom Mower

Caltrans regularly mows the medians and rights-of-way, especially the vegetation adjacent to the roadway. Caltrans usually mows each area twice a year, although some areas are treated more often due to political pressure and fire risk.[11] The controlled strip of vegetation adjacent to the roadway is often called the mow strip for this reason. As mentioned, herbicides are also used in the mow strip area to control vegetation. Mowing differs from herbicide use in a number of ways. Mainly, mowing leaves some vegetation remaining after treatment. The grasses, brush, and weeds are cut to between 10 and 20 cm (4 to 8 in) in height. The cut should not be any lower than 10 centimeters. Lower cutting heights can damage mowers and cause scalping of the ground, which can lead to adverse environmental effects.[5] Figure 3.10 shows the bare cost calculations for Caltrans mowing operations. The following is a discussion of the benefits and drawbacks of mechanical mowing.



Mechanical Mowing Cost Calculations

Constants

$34.00	loaded crew rate ($/hr)
8.00	typical workday length (hr)
1.00	crew members per truck
5.00	speed of tractor (mph)
1.00	tractor fuel consumption (mpg)
$2.00	cost of gas ($/gal)
10,000.00	m^2 per hectare
43,560.00	ft^2 per acre
5,280.00	ft per mile
2.47	acres per hectare
4.00	width of treatment swath (ft)
1.20	width of treatment swath (m)

Calculations

Acres/hr	2.42
Acres/day	19.39
mi/day	40.00
Crew ($/day)	$272.00
Crew ($/acre)	$14.03
Gas ($/day)	$80.00
Gas ($/acre)	$4.13

Totals

English Units	
$/Acre	$19.15
acre/day	19
SI Units	
$/ha	$47.32
ha/day	8

Figure 3.10 Mechanical Mowing Cost Calculations

Benefits of Mechanical Mowing

Mechanical mowing is beneficial for many reasons. First of all, mowers only require one person for operation, where as herbicide spray trucks require two and hand maintenance can take ten or more. Mowing does not use any chemicals other than fuel for power. Therefore it has little impact on the environment. Mowing is also an economical choice for vegetation control. On a per acre cost basis, mowing is cheaper than most vegetation control methods except for herbicide applications, with which it compares closely. Mowing operations also have few weather related limitations. Only high fire danger and wet weather, which may cause the mowers to become stuck, limit mowing activities.[11] Mower blades may emit sparks when they hit rocks, possibly igniting vegetation during times of high fire risk. In summary, mowing is a cheap vegetation control method with little environmental impact and few weather related restrictions.

Drawbacks of Mechanical Mowing

The following are the disadvantages with mechanical mowing. Mowing often requires a lane closure for operation, as the tractor requires a lane from which to operate. Most of the operations are performed in the daytime so traffic delays are possible in congested areas. Some mowing is done entirely in the median or right-of-way to avoid lane closures, but it is usually unavoidable. Another problem is that mowers leave grass cuttings behind on the controlled vegetation. These cuttings, called duff, dry out quickly and pose an additional risk of fire. Duff is responsible for some fires in areas with low or moderate fire risk.[11] In addition, Mowers travel at relatively slow rates, increasing the time and cost of a treatment. Caltrans landscaping mowers operate at a maximum of 8 km/h (5 mph).[12] Finally, mowing does not totally eliminate unwanted vegetation, nor does it prevent the growth of unwanted vegetation. Mowing only reduces the amount of vegetation on the roadside. Sometimes more than two treatments are necessary for proper maintenance, which drives up the cost of mowing operations. The downsides of mowing include lane closures, increased fire risk from duff, slow travel speeds, and the necessity of frequent treatments.

Alternative Mowing Technologies

There are quite a few different types of mowers as previously mentioned. Some alternative mower attachments may improve upon some methods of maintenance. For instance, many different types of guardrail mowers are currently available. These mowers attach to tractors and weave in and out of posts so as to cut vegetation, but protect the rails and the posts. John Deere, US Ditcher and Dondi USA Inc., and Alamo Industrial are a few of the companies that produce guardrail mowers. These alternatives are compared to other methods in the next chapter. Figure 3.11 is a picture of a tractor mounted guardrail mower.

Figure 3.11 Tractor Mounted Guardrail Mower

3.1.3 No Vegetation Maintenance

Over the last few years, the Federal government and the State of California have had many budget issues. Caltrans funding has been cut along with many other federal and state agencies in an attempt to balance the budget.[3] Because of these funding shortfalls, Caltrans has cut its labor force. Maintenance yards are currently operating with fewer workers than required to perform all of the yearly maintenance tasks.[6] This means that maintenance tasks are prioritized and the most pressing maintenance tasks are performed first. Sometimes, low priority vegetation control tasks are not completed. Although this is not a legitimate vegetation control technique, it is important to look at the pros and cons of not performing vegetation maintenance. Figure 3.12 shows an untreated vegetation landscape.

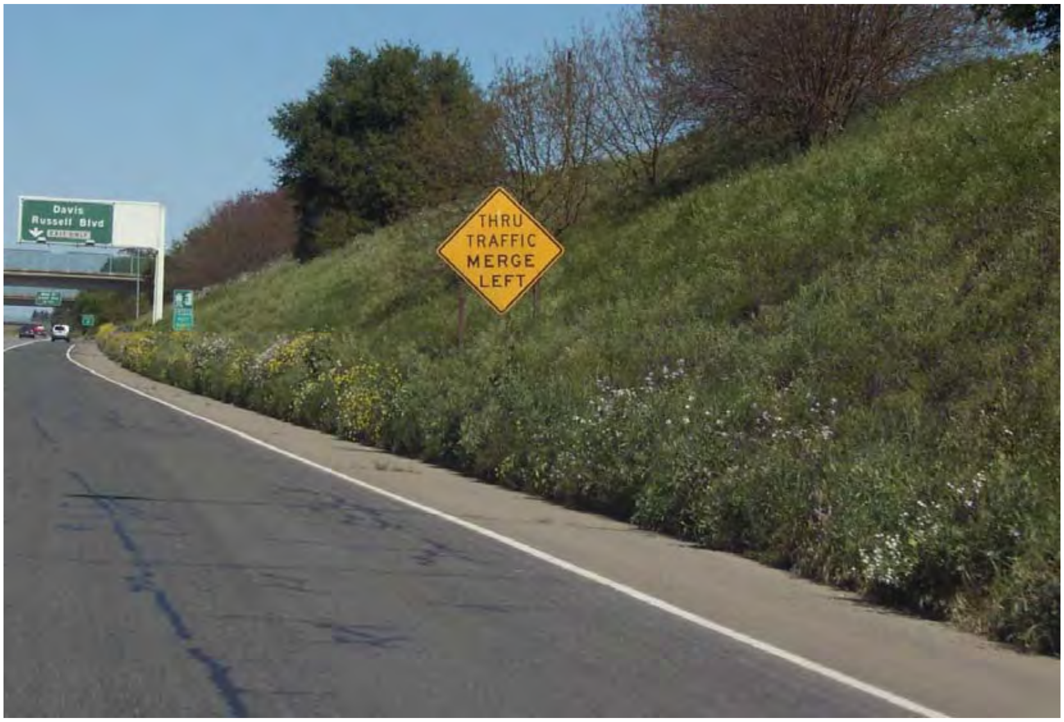


Figure 3.12 Untreated Vegetation Landscape

Benefits of No Vegetation Maintenance

The obvious benefit is that neglect of roadside vegetation costs Caltrans nothing. In addition there are no lane closures to slow traffic, no harmful chemical exposure, and no noise. Some people argue that natural vegetation is also more beautiful. There is a reason, however, for vegetation maintenance, and there are definite drawbacks to neglect.

Drawbacks of No Vegetation Maintenance

There are many reasons for vegetation maintenance, and the drawbacks of performing no maintenance should make this clear. First of all, there is much greater risk of fire as brush grows right up to the roadway. Also, invasive plants can grow out of control along the roadside. Invasive plants harm the agricultural production of California and also present higher fire risk as they grow higher and faster than native vegetation. Vegetation obscures road signs, undermines the integrity of the roadway, limits visibility, and blocks safety access for vehicles in case of emergency. If left to itself, vegetation causes more problems, and may cost more money in indirect costs than is saved by not performing maintenance.

Although there are many drawbacks to not performing vegetation control, there is some room for study. It is possible that levels of service between total neglect and current levels of control are applicable for low priority, low traffic roadways. A much more in depth cost analysis is required to determine the most efficient levels of vegetation control.

3.2 Alternative Vegetation Control Technologies

Alternative technologies are the vegetation control methods that are not widely used by Caltrans. Many of these technologies are in the prototype or development stages of production. Some of these technologies are used only in industrial farming, in Europe, or in other non roadside applications. The following technologies are identified as possible alternatives to current Caltrans maintenance methods, with an emphasis on replacing herbicide application. In the case where the control method is not developed for mechanical roadside use, estimates are also used to determine the cost, safety and effectiveness of an automated truck mounted operation. These vegetation control methods are compared in the following chapter to the current Caltrans control methods introduced above.

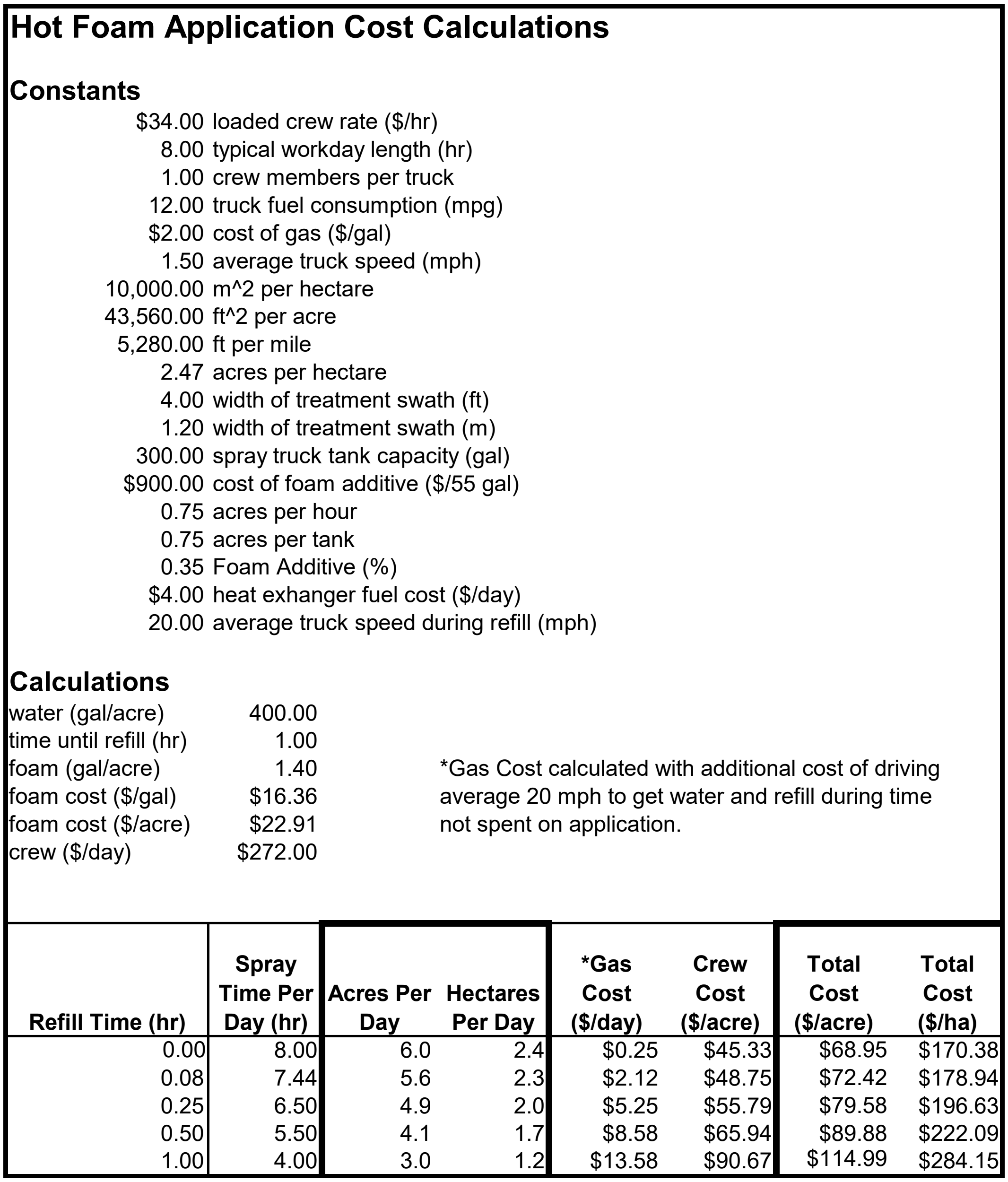
The following methods were selected for a variety of reasons. First of all, Caltrans has previously, or is currently testing the practicality of implementing these alternative methods. Second of all, these methods directly reduce the amount of herbicides used in vegetation control. Finally, these methods are mechanical alternatives to current vegetation control methods. Implementation of these methods may save Caltrans time, labor, money, and limit worker exposure to roadway hazards.

3.2.1 Hot Water, Hot Foam, Steam Application

The treatment of undesirable vegetation using hot water has been around for about 20 years. The concept of the application is very simple. Water, heated to temperatures near boiling, is applied to unwanted vegetation. The water melts the waxy coating of the vegetation leading to severe dehydration.[15] Several seconds of high intensity heat are enough to kill the vegetation. The effects of this method are nearly instantaneous and usually kill the plant within 24 hours.[13] The effects are similar to herbicide application in that the ground becomes completely bare after treatment. Figure 3.13 shows a Waipuna hot foam vegetation treatment system.

Figure 3.13 Waipuna Hot Foam Treatment Unit

The two main manufacturers of hot water applicators are Waipuna Inc. and Aqua Heat Technology, Inc. Both of the machines have similar operational designs, but each system is used for different purposes. The Aqua Heat machine is used primarily in orchards, whereas the Waipuna system is used on roadsides, parks, fields, and forests.[13] The Waipuna machine also uses a hot foam surfactant during application because hot foam traps heat around the plants. This allows for greater application speed as less water is required to sufficiently treat vegetation. In making calculations for the benefits and drawbacks and in making the scenario comparisons, only numbers for the Waipuna system are used. This is because Waipuna, Inc. has the most advanced technology, widely available information and has been tested on roadsides. Figure 3.14 shows the calculated cost and efficiency of the Waipuna system.



Hot Foam Application Cost Calculations

Constants

$34.00	loaded crew rate ($/hr)
8.00	typical workday length (hr)
1.00	crew members per truck
12.00	truck fuel consumption (mpg)
$2.00	cost of gas ($/gal)
1.50	average truck speed (mph)
10,000.00	m^2 per hectare
43,560.00	ft^2 per acre
5,280.00	ft per mile
2.47	acres per hectare
4.00	width of treatment swath (ft)
1.20	width of treatment swath (m)
300.00	spray truck tank capacity (gal)
$900.00	cost of foam additive ($/55 gal)
0.75	acres per hour
0.75	acres per tank
0.35	Foam Additive (%)
$4.00	heat exhanger fuel cost ($/day)
20.00	average truck speed during refill (mph)

Calculations

water (gal/acre)	400.00
time until refill (hr)	1.00
foam (gal/acre)	1.40
foam cost ($/gal)	$16.36
foam cost ($/acre)	$22.91
crew ($/day)	$272.00

*Gas Cost calculated with additional cost of driving average 20 mph to get water and refill during time not spent on application.

Refill Time (hr)	Spray Time Per Day (hr)	Acres Per Day	Hectares Per Day	*Gas Cost ($/day)	Crew Cost ($/acre)	Total Cost ($/acre)	Total Cost ($/ha)
0.00	8.00	6.0	2.4	$0.25	$45.33	$68.95	$170.38
0.08	7.44	5.6	2.3	$2.12	$48.75	$72.42	$178.94
0.25	6.50	4.9	2.0	$5.25	$55.79	$79.58	$196.63
0.50	5.50	4.1	1.7	$8.58	$65.94	$89.88	$222.09
1.00	4.00	3.0	1.2	$13.58	$90.67	$114.99	$284.15

Figure 3.14 Hot Foam Application Cost Calculations

The hot water application system is a fairly simple concept. Water is initially stored in a large holding tank, and then piped into an engine powered heat exchanger. Next, the super heated water is transferred through an insulated hose out to the applicator head which is positioned close to the vegetation for minimum heat loss. As mentioned, Waipuna Inc. has developed a foam surfactant that traps heat for longer periods of time on the vegetation. The additive is premixed with the water before treatment begins. Figure 3.15 shows a hot foam applicator in action.

Figure 3.15 Hot Foam Application Applied on Foot

Waipuna, Inc. currently sells and leases the most advanced hot foam applicator. The standard tank holds 1,140 liters (300 gallons) of water. Waipuna, Inc. also sells a hot foam additive that costs about $900 per 100 liters (55 gallons). Approximately 4 liters (1.1 gal) of foam surfactant concentrate are used per tank.[13] This single boiler system is currently leased for $700 a month, and may be bought outright for $28,500.[14] The following will cover the pros and cons of hot foam application. In addition, theoretical calculations of an automated roadway hot foam system are also discussed.

Benefits of Hot Foam Application

Hot water/foam treatments offer numerous benefits over herbicide and mowing applications. First of all, the water and additive solution, alkyl polyglycoside, is nontoxic and biodegrades in the ground within 28 days. The additive is comprised of corn and coconut sugars. Limited investigations show the additive to be nontoxic to life except if applied directly to the surface of water. The California Department of Pesticide Regulation does not consider the surfactant as a pesticide, so no registration is required.[14] Due to the additive's low toxicity, hot foam requires only minimal cleaning and safety expenditures compared to those associated with herbicides.

In addition, a major advantage of hot foam is its resistance to variable weather conditions. Application is more effective on hot clear days, but treatment is also available in windy conditions with light rain.[13] In contrast, herbicide applications are useful only in dry weather with a maximum of light wind. Sometimes, early herbicide applications are cancelled due to poor weather conditions. If herbicides cannot be applied, greater vegetation growth may lead to higher fire risk and greater noxious weed contamination.[6] The durability of this process and low toxicity of the surfactant are big advantages over herbicide use. The drawbacks may severely limit Caltrans' implementation of hot foam application though.

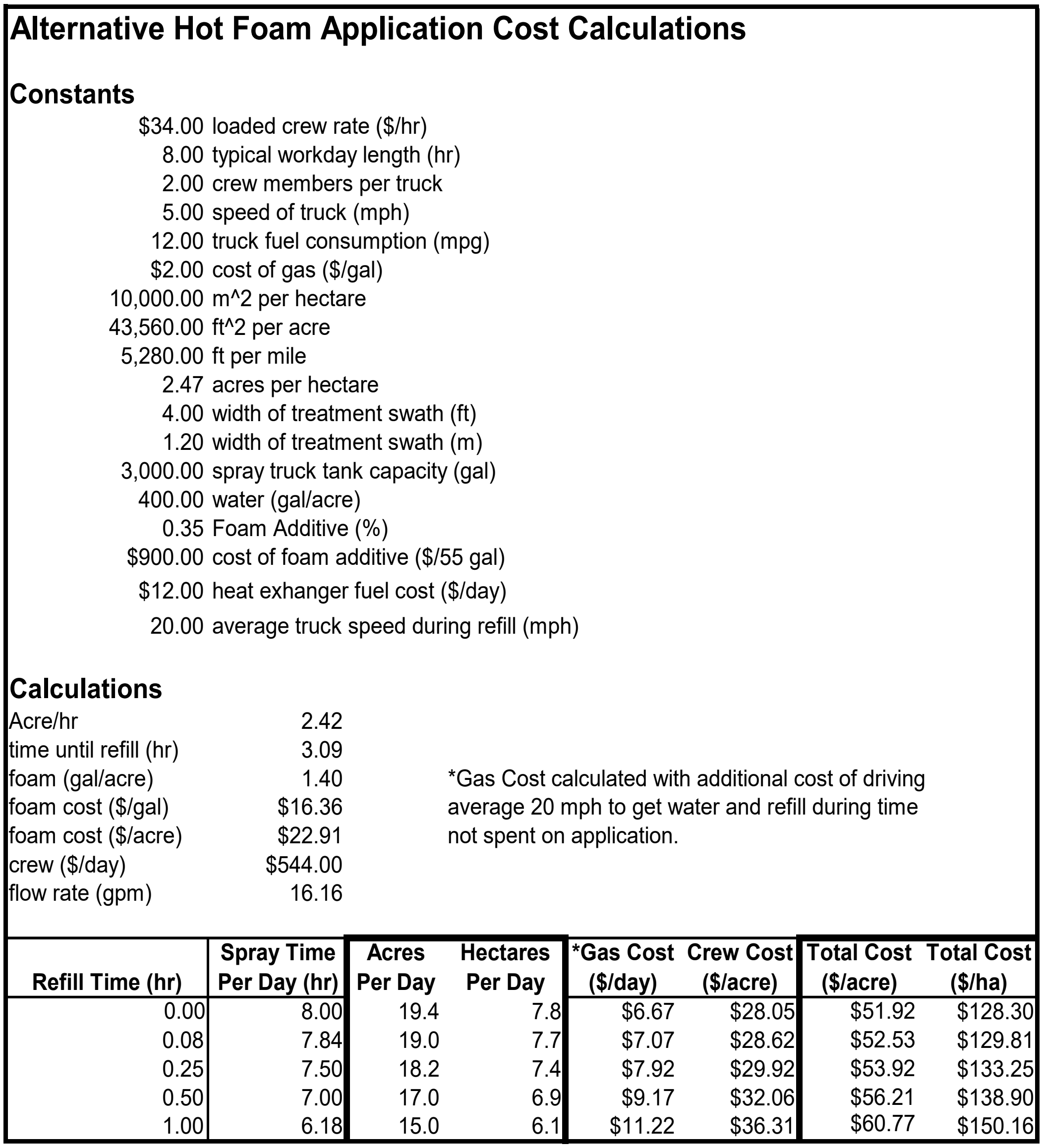
Drawbacks of Hot Foam Application

Cost, efficiency, and worker safety are all important aspects in determining the quality of any vegetation control method. Unfortunately, hot foam application performs poorly in all three areas. The second problem involves the Waipuna system's reliance on a central spray truck, which puts workers on foot to spray the undesired vegetation. Workers do not have the protection of a heavy vehicle while on foot. In addition, workers are exposed to accidents and injuries unrelated to traffic.[6] Workers on the roadside are much more prone to injury than those operating from the safety of a vehicle.

The efficiency of the hot foam system is probably the biggest drawback. The hot foam application is cumbersome. A typical application, with a 1.2 m (4 ft) spray swatch, covers 1.5-2 ha (4-5 acres) per day. This is far fewer than the 13 ha (31 acres) for a typical herbicide application. Another problem is the large use of water. The hot foam treatment requires 3700 l/ha (400 gal/acre) of water for a complete treatment.[13] This would require a refill 6 to 8 times per day. The number of refills and the time it takes to refill the tank negatively impact the amount of time spent treating vegetation and the daily treatment area. Figure 3.14 shows the relative amounts of time spent on treatment, and the effective daily treatment area for different refill times. It is interesting that even if no refills were required (water tank had sufficient water capacity for an entire day), the treatment still only covers 2.5 ha (6 acres). This is still very poor in comparison to herbicide and mow treatments. This system has many benefits, especially in regards to the environment, but the efficiency drawbacks severely limit the applicability of this vegetation control method.

Theoretical Analysis of Automated Hot Foam Application

An automated hot foam applicator, tailored to Caltrans vegetation maintenance needs may eliminate many of the drawbacks to currently available hot foam technology. Figure 3.16 shows the assumptions of the capabilities of a fully automated roadway hot foam spray truck, as well as the theoretical daily cover area and base cost.



Alternative Hot Foam Application Cost Calculations

Constants

$34.00	loaded crew rate ($/hr)
8.00	typical workday length (hr)
2.00	crew members per truck
5.00	speed of truck (mph)
12.00	truck fuel consumption (mpg)
$2.00	cost of gas ($/gal)
10,000.00	m^2 per hectare
43,560.00	ft^2 per acre
5,280.00	ft per mile
2.47	acres per hectare
4.00	width of treatment swath (ft)
1.20	width of treatment swath (m)
3,000.00	spray truck tank capacity (gal)
400.00	water (gal/acre)
0.35	Foam Additive (%)
$900.00	cost of foam additive ($/55 gal)
$12.00	heat exhanger fuel cost ($/day)
20.00	average truck speed during refill (mph)

Calculations

Acre/hr	2.42
time until refill (hr)	3.09
foam (gal/acre)	1.40
foam cost ($/gal)	$16.36
foam cost ($/acre)	$22.91
crew ($/day)	$544.00
flow rate (gpm)	16.16

*Gas Cost calculated with additional cost of driving average 20 mph to get water and refill during time not spent on application.

Refill Time (hr)	Spray Time Per Day (hr)	Acres Per Day	Hectares Per Day	*Gas Cost ($/day)	Crew Cost ($/acre)	Total Cost ($/acre)	Total Cost ($/ha)
0.00	8.00	19.4	7.8	$6.67	$28.05	$51.92	$128.30
0.08	7.84	19.0	7.7	$7.07	$28.62	$52.53	$129.81
0.25	7.50	18.2	7.4	$7.92	$29.92	$53.92	$133.25
0.50	7.00	17.0	6.9	$9.17	$32.06	$56.21	$138.90
1.00	6.18	15.0	6.1	$11.22	$36.31	$60.77	$150.16

Figure 3.16 Theoretical Hot Foam Application Cost Calculations

Assuming the automated roadside hot foam machine has a capacity of 11,000 liters (3000 gal) and travels at 8 km/h (5 mph) along the roadside, the cost, worker safety and efficiency improve dramatically. Even though this new vehicle carries two crew members instead of one as do herbicide spray trucks, the cost is more than offset by the increased speed and efficiency. This theoretical machine is roughly 30% less expensive to operate than the standard Waipuna machine. Even with all the benefits of this machine, it is still 2.5 times more expensive than mowing and herbicide application.

The efficiency of this theoretical machine is dramatically improved. This alternative method covers four times more ground per day than the Waipuna system. It also covers slightly less ground than mowing operations and only 30% less than herbicide applications. Another benefit of an automated truck is that the workers are protected inside the truck cab. It is a great benefit anytime workers are removed from the roadway.

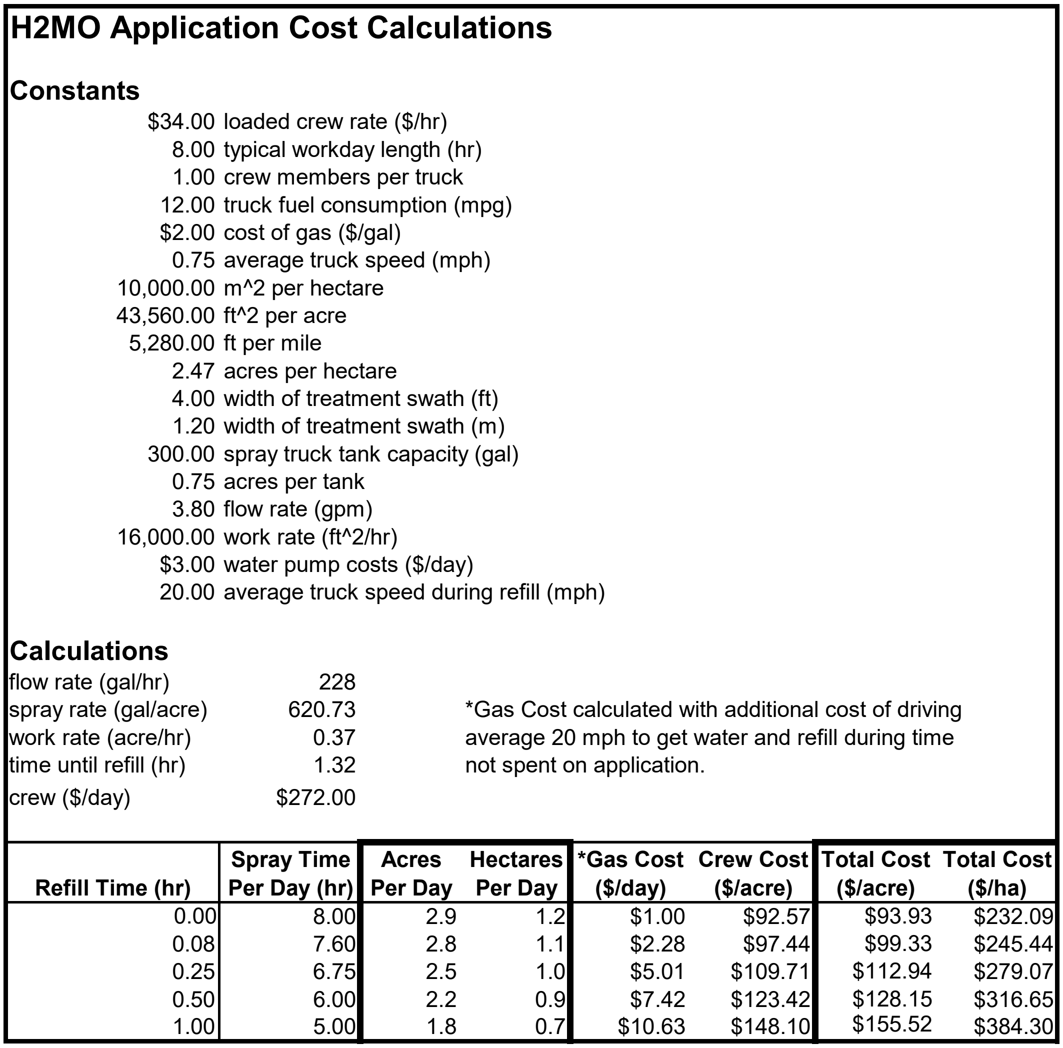
There are still a couple of concerns with this automated model. The water output flow rate is approximately 60 l/min (16 gpm). This is a very high output rate and a large boiler is required to cope with the volume of heated water. In addition, refill time and the availability of water may further increase the cost of this application process. The cost of the foam is also a factor. At nearly \$57 per ha (\$23 per acre), the foam dramatically increases the cost of operation. There are still many unknowns involved in the development of this design. This theoretical analysis is only an attempt to show the long term feasibility of hot foam application.

3.2.2 Hydro-Mechanical Obliteration

Hydro-mechanical obliteration is a control method that uses high pressure water to cut or mow vegetation. The operation of this device is simple and straightforward. Water is stored in a tank and then pumped through a compressor. The water reaches pressures of 21,000-48,000 kPa (3,000-7,000 psi).[16] The water is pumped through a hose and out of a specific high pressure nozzle attached to a hand held wand. The worker aims the wand at unwanted vegetation and clears it from the ground. The brush is either mulched back into the soil or pushed back off of cleared land by the water stream. The distance from the ground, vegetation, and spray nozzle determines the power of the spray. When held at a distance of 1 meter (3 ft) the nozzle clears most brush without disturbing the soil. Figure 3.17 shows a H2MO$^{(sm)}$ unit in operation.

Figure 3.17 H2MO$^{(sm)}$ Unit in Operation

The A-1 unit comes standard with a 1,140 liter (300 gallon) water tank. The system uses a 24 kW (32 hp) water compressor and can deliver up to 14 l/min (3.8 gpm) of pressurized water. With a single operator, up to 1,500 square meters (16,000 square feet) is covered in an hour. The treatment is slow and a worker can cover about one hectare (2.5 acres) per day. An hour and twenty minutes of non stop operation are possible at the maximum flow rate. The cost of purchasing this unit is $19,000.[16] Figure 3.18 shows the cost and efficiency of using a high pressure water system. The following discusses the pros, cons, and theoretical performance of an automated roadway hydro-mechanical obliteration machine.



H2MO Application Cost Calculations

Constants

$34.00	loaded crew rate ($/hr)
8.00	typical workday length (hr)
1.00	crew members per truck
12.00	truck fuel consumption (mpg)
$2.00	cost of gas ($/gal)
0.75	average truck speed (mph)
10,000.00	m^2 per hectare
43,560.00	ft^2 per acre
5,280.00	ft per mile
2.47	acres per hectare
4.00	width of treatment swath (ft)
1.20	width of treatment swath (m)
300.00	spray truck tank capacity (gal)
0.75	acres per tank
3.80	flow rate (gpm)
16,000.00	work rate (ft^2/hr)
$3.00	water pump costs ($/day)
20.00	average truck speed during refill (mph)

Calculations

flow rate (gal/hr)	228
spray rate (gal/acre)	620.73
work rate (acre/hr)	0.37
time until refill (hr)	1.32
crew ($/day)	$272.00

*Gas Cost calculated with additional cost of driving average 20 mph to get water and refill during time not spent on application.

Refill Time (hr)	Spray Time Per Day (hr)	Acres Per Day	Hectares Per Day	*Gas Cost ($/day)	Crew Cost ($/acre)	Total Cost ($/acre)	Total Cost ($/ha)
0.00	8.00	2.9	1.2	$1.00	$92.57	$93.93	$232.09
0.08	7.60	2.8	1.1	$2.28	$97.44	$99.33	$245.44
0.25	6.75	2.5	1.0	$5.01	$109.71	$112.94	$279.07
0.50	6.00	2.2	0.9	$7.42	$123.42	$128.15	$316.65
1.00	5.00	1.8	0.7	$10.63	$148.10	$155.52	$384.30

Figure 3.18 High Pressure Water Application Cost Calculations

Benefits of Hydro-Mechanical Obliteration

The advantages of hydro-mechanical obliteration include environmental friendliness and the lack of additives. There are no additives to the water as there are with herbicides or hot foam. This cuts back on cost and time. There are also very few environmental issues because water is the only substance used. The only significant environmental concern relates to erosion. It is possible to disturb soil if the high pressure spray is too close and so erosion is a risk. However, as long as the wand is held at the appropriate distance from the soil, erosion risk is minimal.[16] Storm water runoff laws do not appear to limit the use of clean water along roadsides. Finally, this method can be applied in any weather condition except for the extremely windy or wet conditions. The wind may disrupt the spray and wet conditions may increase the risk of erosion because the soil is looser after rains. As with the other alternative methods, the environmental friendliness is the largest benefit.

Drawbacks of Hydro-Mechanical Obliteration

The drawbacks include efficiency, water use, vegetation growth, duff, and worker safety. Hydro-mechanical efficiency is especially poor. As mentioned above, a typical treatment covers 1 hectare (2.5 acres) in a full work day. This is paltry in comparison to 13 hectares (31 acres) for herbicides and 8 hectares (19 acres) for mowing. Another problem with the efficiency is that it takes time to fill up the water tank. Sometimes three or four refills per day are required. In addition, it takes roughly 2,200 liters (600 gal) of water to treat one acre of land. This is 50% more than hot foam treatments. Another problem has to do with the treatment of vegetation. Since this method dispenses a large amount of water on the ground, plants always grow back.[16] Also, vegetation tends to grow back quickly so two or three applications are required for complete control. In addition, the treated vegetation may build up along the treated area. This brush may dry out quickly and present a fire hazard similar to duff.

The last downside to hydro-mechanical obliteration is the lack of safety for workers. First of all, workers are exposed when outside of vehicles. This method requires maintenance workers to be outside of the vehicle. As mentioned, workers are much more exposed to injury while working on the roadway as opposed to inside of a vehicle. The high pressure water is also hazardous. The water exits at extremely high velocities and was observed to chip asphalt when the water jet was held closer than 15 cm (6 in) to the concrete. The wand itself is hard to control under such extreme pressure.[16] This could lead to accidents and worker fatigue. Finally, it is recommended that a user wear full protective gear similar to a fire fighter to prevent injuries. This gear is cumbersome and would take more money, time, and energy to use. An automated machine would reduce or remove many, but not all, of the drawbacks to hydro-mechanical obliteration.

Theoretical Analysis of Automated Hydro-Mechanical Obliteration

Similar to the hot foam applicator, an automated high pressure water system tailored to Caltrans vegetation maintenance needs may eliminate many of the drawbacks to high pressure water vegetation control. Figure 3.19 shows the assumptions of the capabilities of a fully automated roadway high pressure spray truck, as well as the theoretical daily cover area and base cost.

Alternative High Pressure Water Application Cost Calculations

Constants

$34.00	loaded crew rate ($/hr)
8.00	typical workday length (hr)
2.00	crew members per truck
12.00	truck fuel consumption (mpg)
$2.00	cost of gas ($/gal)
5.00	average truck speed (mph)
10,000.00	m^2 per hectare
43,560.00	ft^2 per acre
5,280.00	ft per mile
2.47	acres per hectare
4.00	width of treatment swath (ft)
1.20	width of treatment swath (m)
3,000.00	spray truck tank capacity (gal)
620.00	flow rate (gal/acre)
$18.00	water pump costs ($/day)
20.00	average truck speed during refill (mph)

Calculations

work rate (acre/hr)	2.42
flow rate (gal/hr)	1503.03
capacity (acres/tank)	4.84
time until refill (hr)	2.00
crew ($/day)	$544.00
spray rate (gpm)	25.05

*Gas Cost calculated with additional cost of driving average 20 mph to get water and refill during time not spent on application.

Refill Time (hr)	Spray Time Per Day (hr)	Acres Per Day	Hectares Per Day	*Gas Cost ($/day)	Crew Cost ($/acre)	Total Cost ($/acre)	Total Cost ($/ha)
0.00	8.00	19.4	7.8	$6.67	$28.05	$29.32	$72.45
0.08	7.76	18.8	7.6	$7.27	$28.92	$30.26	$74.77
0.25	7.25	17.6	7.1	$8.54	$30.95	$32.46	$80.21
0.50	6.50	15.8	6.4	$10.42	$34.52	$36.33	$89.76
1.00	6.00	14.5	5.9	$11.67	$37.40	$39.44	$97.46

Figure 3.19 Untreated Vegetation Landscape

The theoretical machine moves at 8 km/h (5 mph), carries 3000 gallons of water, and has a passenger controlled water spray system. There are two operators, one to drive and one to operate the sprayer system. For a 1.2 meter (4 ft) treatment swath, this machine covers about 7 hectares (18 acres) per day. This machine only requires a water refill every two hours instead of a refill every 1.3 hours for the standard H2MO^{sm}. This method covers a similar amount of area to mowing and about 30% less than herbicide application. This is comparable to the theoretical hot foam applicator. Including the refill time, this method would cost about $80 per hectare ($35 per acre). This is only a quarter of the cost of the available high pressure water systems. This method is also only 75% more expensive than mowing and herbicide operations, fairly competitive for an alternative method. Finally, the workers are protected inside the cab when working from this automated vehicle. As mentioned many times, roadway injuries are significantly reduced by removing maintenance workers from the ground.

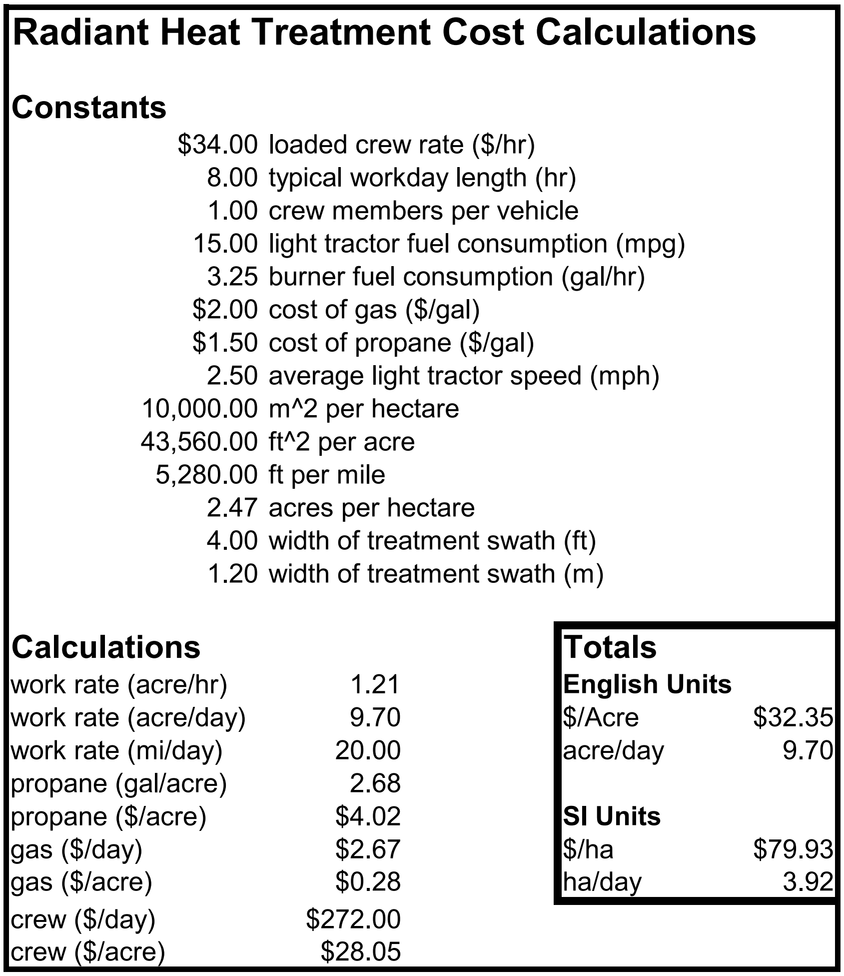
There are still some problems including the demand for water and the need for increased treatments on account of reduced effectives. This method would require a spray rate of 95 l/min (25 gpm). This is fairly high spray rate, although no heating is required so this spray rate is possible. This treatment also uses large quantities of water so plant growth on treated area is all but assured. Two to three yearly treatments may be necessary for complete treatment. Overall this theoretical machine is a huge improvement over its currently available counterpart. A machine like this protects workers, improves productivity, and lowers costs.

Thermal Weed Treatment

Heat treatments take a controlled approach to the age old method of using fire to control weeds. There are a many types of thermal units including hot water and hot foam treatments discussed above. The other treatments all involve some form of combustion heating. The three main methods use open flames, radiant heat, or ultraviolet heating. Open flame heating, known as flaming, uses propane or gasoline to ignite a torch. The torch is passed over vegetation to treat it. This method is not considered in this paper because of the extreme fire risk. Ultraviolet heating uses powerful ultraviolet light bulbs to stimulate the water molecules of plants in a method similar to microwave heating. There is not much information available about this method, and it appears to be in the early stages of development. Therefore, ultraviolet heating is also not considered in this paper. Radiant heat treatments, however, have much lower risks of fire, more information available, and are more applicable to vegetation control in California. This section will focus on the current radiant heat treatment technology. Figure 3.20 shows a boom attachment for a thermal radiant heat unit.

Figure 3.20 Radiant Heat Boom Attachment

Radiant heaters, also called infrared heaters, work in the following way. Fuel is combusted inside ceramic heating elements. The ceramic elements radiate infrared heat, which is directed at vegetation with the heat source only a few centimeters (.5 inches) above the vegetation. The heating elements can reach temperatures of up to 1000°C (1832°F).[19] The intense heat ruptures the protein cells and stops photosynthesis and kills the plant.[18] A German company produces the ECOflame® infrared heat treatment machine that is towed or pushed by a light duty tractor.[20] As shown in Figure 3.21, Operation of an ECOflame® machine would cost Caltrans approximately $80 per hectare ($32 per acre). With this method, Caltrans could cover about 4 hectares (10 acres) per day. Radiant heat units are popular in Europe but have not taken hold in the United States. The following is a discussion of the benefits and drawbacks of thermal weed treatment and radiant heating in particular.



Radiant Heat Treatment Cost Calculations

Constants

$34.00	loaded crew rate ($/hr)
8.00	typical workday length (hr)
1.00	crew members per vehicle
15.00	light tractor fuel consumption (mpg)
3.25	burner fuel consumption (gal/hr)
$2.00	cost of gas ($/gal)
$1.50	cost of propane ($/gal)
2.50	average light tractor speed (mph)
10,000.00	m^2 per hectare
43,560.00	ft^2 per acre
5,280.00	ft per mile
2.47	acres per hectare
4.00	width of treatment swath (ft)
1.20	width of treatment swath (m)

Calculations

work rate (acre/hr)	1.21
work rate (acre/day)	9.70
work rate (mi/day)	20.00
propane (gal/acre)	2.68
propane ($/acre)	$4.02
gas ($/day)	$2.67
gas ($/acre)	$0.28
crew ($/day)	$272.00
crew ($/acre)	$28.05

Totals

English Units	
$/Acre	$32.35
acre/day	9.70
SI Units	
$/ha	$79.93
ha/day	3.92

Figure 3.21 Radiant Heat Treatment Cost Calculations

Benefits of Thermal Weed Treatment

Thermal weed heaters possess significant benefits. First of all, the energy required for operation of the heating elements is fairly low. Most radiant heat machines burn propane gas, which burns cleaner and more efficiently than gasoline or diesel. This method is also the fastest of the alternative methods with a daily coverage area of 4 hectares (10 acres). A typical application covers 67% more than hot foam treatments and over 200% more than high pressure water treatments. In addition, this method can be used in situations where open flames are very dangerous.[18] Another benefit is that the intense heat kills seeds before they can germinate. Finally, the largest advantage of this machine is that it does not use chemicals and has low environmental impact. This is the largest advantage of every alternative technology to herbicide application.

Drawbacks of Thermal Weed Treatment

Thermal weed control possesses many of the same drawbacks as other alternative technologies. At $80 per hectare ($32 per acre), thermal weed control only costs about 50% more than mowing and herbicides. Despite the low cost, this technology covers far less area per day than traditional treatments. This increases the cost of application further because the workers are held longer for fixed area treatments, and are unable to perform other maintenance tasks. Another problem is that weeds tend to grow back sooner than with other methods. The effects of the treatment do not linger and two to three treatments per year are necessary for control. This operation may additionally have problems with extremely tall vegetation and dense vegetation. Finally, thermal treatments also possess some fire risk because of the intense heat. Although no open flames are used, dry vegetation may still ignite in some situations. These drawbacks can be substantial, but a theoretical operation may eliminate some of the efficiency issues.

Theoretical Analysis of Thermal Weed Treatment

In order to find a more appropriate comparison to current Caltrans technologies, a theoretical analysis of a possible roadside thermal weed treatment machine is examined. Figure 3.22 shows the costs and daily coverage of an automated radiant heat machine.

Alternative Radiant Heat Cost Calculations

Constants

$34.00	loaded crew rate ($/hr)
8.00	typical workday length (hr)
1.00	crew members per vehicle
1.00	tractor fuel consumption (mpg)
6.50	burner fuel consumption (gal/hr)
$2.00	cost of gas ($/gal)
$1.50	cost of propane ($/gal)
5.00	average light tractor speed (mph)
10,000.00	m^2 per hectare
43,560.00	ft^2 per acre
5,280.00	ft per mile
2.47	acres per hectare
4.00	width of treatment swath (ft)
1.20	width of treatment swath (m)

Calculations

work rate (acre/hr)	2.42
work rate (acre/day)	19.39
work rate (mi/day)	40.00
propane (gal/acre)	2.68
propane ($/acre)	$4.02
gas ($/day)	$80.00
gas ($/acre)	$4.13
crew ($/day)	$272.00
crew ($/acre)	$14.03

Totals

English Units	
$/Acre	$22.17
acre/day	19.39
SI Units	
$/ha	$54.79
ha/day	7.85

Figure 3.22 Alternative Radiant Heat Cost Calculations

This theoretical analysis assumes an infrared heating tractor attachment. The one person tractor moves at 8 km/h (5 mph). An alternative radiant heat machine would cost approximately $55 per hectare ($22 per acre) to operate. This is only a few dollars more expensive than mowing and herbicide application and is very competitive. Small improvements in operational speed would make this method cheaper and more efficient than herbicide application. In addition, with a tractor mounted operation, the same amount of area can be covered as with a mowing operation. If cost and efficiency is the most important measure, this method may compete well with current Caltrans operations. Again, as mentioned many times, protecting the workers in a cab is much more preferable to having them on the roadway. This is another benefit of an automated radiant heat system. Finally, this method has some downsides such as fire risk and treatment. These drawbacks are small though and it is possible that this method could prove to be a better choice both environmentally and in terms of cost than current Caltrans operations.

3.3 Summary

This chapter is a compilation and analysis of the information pertinent to the two standard and three alternative methods for roadside vegetation control. Both the pros and the cons of each method are discussed and analyzed. Bare calculations for each method are also performed for the comparative analysis undertaken in the proceeding chapter. In addition, assumptions are made as to the performance of alternative methods if they are adapted for Caltrans' roadside use. Using this information, theoretical cost and efficiency calculations are performed for each alternative method. The theoretical analysis is used in the next chapter as well in the concept scoring analysis. It is important to develop these theoretical calculations so that prospective alternative methods can compete on a level playing field developmentally with current Caltrans methods. The pros and cons of each method are clearer in this way.

Effort is taken to keep the above cost and efficiency calculations as conservative as possible. Vehicle speeds were limited to 8 km/h (5 mph) and water tank capacities were limited to the capacities of current Caltrans' fleet vehicles. Fuel costs are taken from nationwide averages and may not reflect actual purchase prices. In addition, fuel consumption is related to the consumption of Honda motors of equivalent size to those required for boiler and pump operations. Overall, the leading cost determinant appears to be labor cost with labor accounting for 60%-90% of the total cost of each method. It may be possible to reduce the total cost of alternative control methods by increasing operational speed and/or the effective treatment swath.

These methods are meant to serve as ballpark estimates only and do not include research, fleet acquisition, maintenance, cleaning, and travel expenses to and from work sites before work begins. The calculations are meant to be as accurate as possible, but with limited information, it is very difficult to make accurate calculations. In addition, since there are no operational alternative control method prototypes suitable for regular maintenance work, it was necessary to find a way to compare all methods equally. It is important that apples are compared with apples. This is not always the case when comparisons are drawn between herbicides, mowing, and alternative vegetation control methods. Taking this into account, the preceding calculations are suitable for the following comparison and make a valid comparison possible and informational.

The following chapter is a comparative analysis of all of the methods discussed and analyzed in this chapter. The goal of the analysis is to determine if any of the alternative methods are feasible replacements for herbicide use. The analysis uses concept-scoring matrices for each of the three scenarios – mow strip, guardrails & posts, and urban vegetation. The theoretical alternative method analysis is important because it may show that further development of alternative control technologies is warranted. An alternative method may prove suitable only for one specific scenario or multiple scenarios. Finally, the next chapter sets up a standard engineering comparison method for the comparison of technologies. It is hopeful that this comparison method may then be adjusted to address additional vegetation control methods, alternate scenarios, and even maintenance tasks outside of vegetation control maintenance.

CHAPTER 4 Vegetation Control Scenario Method Selection

This chapter involves the concept scoring of Caltrans' two main vegetation maintenance methods – mowing, herbicide application – and the three alternative vegetation control methods – hot foam application, radiant heating, and high pressure water treatment. The scoring also includes the presumed performance for the theoretical automated versions of each alternative control method. One of the objectives of this concept scoring analysis is to determine if alternative technologies can compete with current vegetation control methods. Another objective is to determine if future development of alternative control methods is warranted. Lastly, it is the intent of this research to develop a method that may be used to compare additional maintenance methods in alternative environments. The following section explains concept scoring and its importance in the design process.

4.1 The Design Process and Concept Scoring

Concept scoring is a small part of the overall design process, the procedure for bringing a product from an idea, a concept, to a marketable product. Most of the following information is taken from the text book Product Design and Development by Ulrich and Eppinger.[23] The design process is a combination of many steps that are as follows: first examine the needs of the customer, next establish target specifications, then generate product concepts, from there select product concepts, next test the product concepts, then set the final specifications, and finally plan downstream development.[23] Concept selection comprises both concept screening and concept scoring.

Concept screening and concept scoring are critical components of the concept selection process. Concept screening and concept scoring are attributed to Stuart Pugh, who developed a method in the 1980s upon which concept screening and concept scoring are based.[23] This method is often called *Pugh concept selection*.[23] Concept screening is a relatively quick way of eliminating concepts that do not satisfy enough target specifications to adequately fulfill the needs of the customer. In addition, those concepts that do not satisfy the customer needs alone may be revised or combined with other concepts to form a concept that will satisfy the customer's demands. Concept screening rates each concept for different selection criteria developed from the customer needs and target specifications. Concept screening uses a yes, no or maybe rating system for simplicity. In addition, all of the selection criteria are weighted evenly. The concepts that score the best are then moved to the concept scoring phase for final selection. Concept screening is not necessary for this report because there are already so few mechanical alternative vegetation control methods with enough information available for an accurate analysis.

Once the field of concepts has been narrowed down with concept screening, the remaining concepts are further scored and ranked in the concept scoring phase. Based on the final ranking, the best concepts are chosen for further development. In concept scoring, the remaining concepts receive a score for each of the selection criteria. This time however, the criteria are weighted so that the most important selection criteria represent a larger portion of the total score. In addition, scores are given from one to five or from one to ten to account for a wider variation in performance between concepts.

Usually a concept is selected to represent an average score for a specific criteria and then the rest of the concepts are scored based on whether they are equal to, better than, or worse than the selected concept. In this concept scoring analysis, herbicide use and mowing served as the benchmark for most of the selection criteria. Once the methods are scored for each selection criteria, the scores are tallied and the concepts are ranked. The highest scoring concepts are therefore selected for further research.

This vegetation control method analysis is comprised of concept scoring matrices, one for each scenario. The scoring matrices have six basic selection criteria categories which comprise the individual scoring criteria. The six basic categories are public safety, worker safety, environment, herbicide use, cost & efficiency, and public perception. These categories are adapted from 1997 Caltrans study, California Roadsides: A New Perspective,[2] which looked into alternative options for roadside vegetation maintenance. This study used a simple concept scoring method with yes, no, and maybe ranks for each selection criteria. These categories are used because they serve the needs of the customer, Caltrans. Some of the same selection criteria are also used but other more applicable criteria are added and irrelevant criteria are removed.

Many of the same criteria are used for all three scenarios with moderate adaptations from scenario to scenario. Some criteria have different importance in each scenario and will receive different weights. In addition, some concepts will receive different scores for the same criteria in different scenarios based upon scenario specific performance. For instance, many control methods have a lower score for the “minimizes foot labor” criteria in the guardrails & posts scenario than in other scenarios. This is because foot labor is often required to perform spot maintenance after initial treatment around posts and guardrails.

Two of the important categories are based on the calculations done in the previous chapter. The daily coverage area and the price per hectare (acre) are important to the overall evaluation of each control method. Figures 4.1 and 4.2 display a comparison for of both the price per hectare (acre) and the daily cover area. For those control methods that require water refills, the information shown assumes a 15 minute refill time. English units are in blue, SI units are in red.

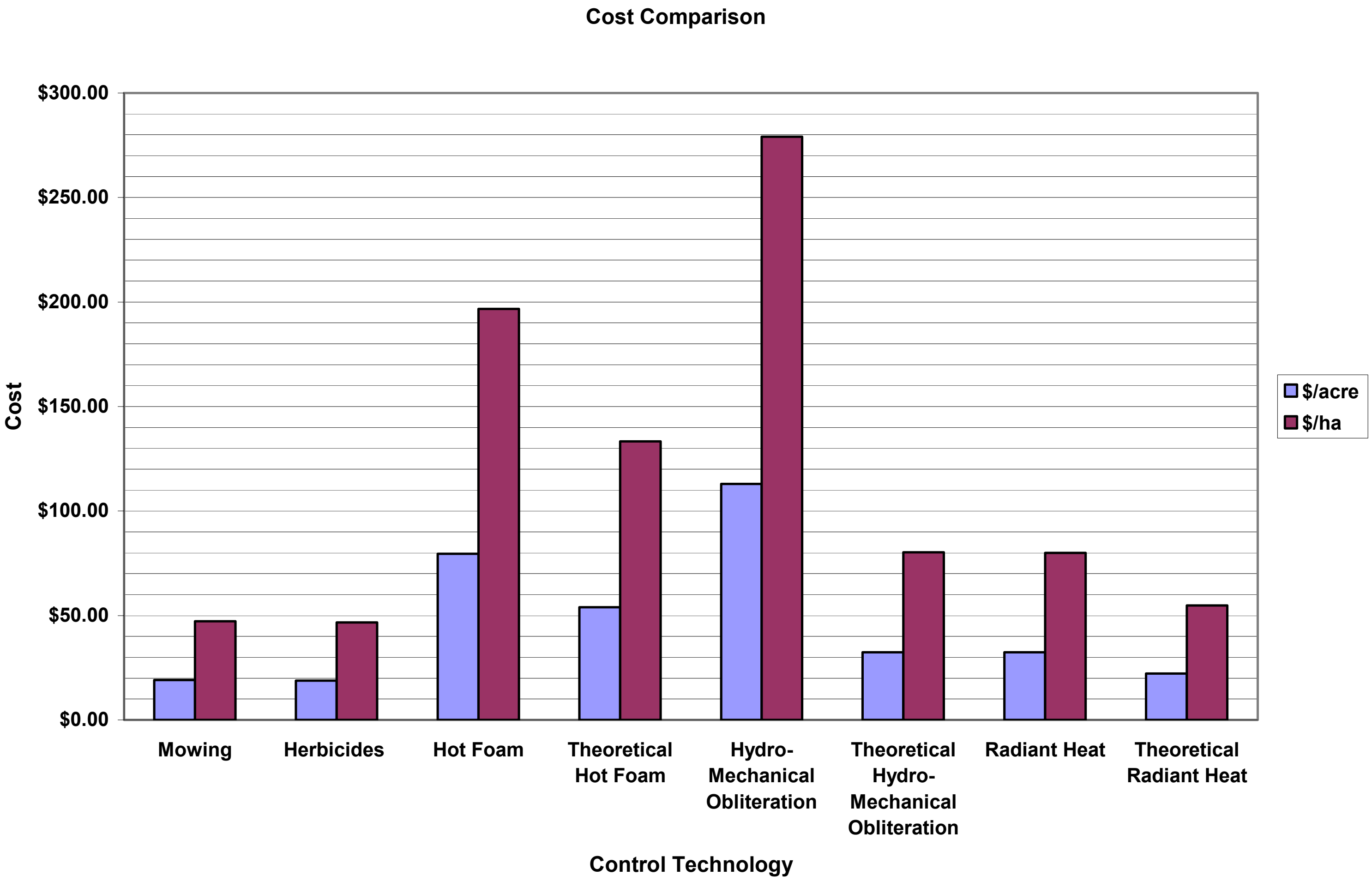


Figure 4.1 Cost Comparisons of Vegetation Control Methods

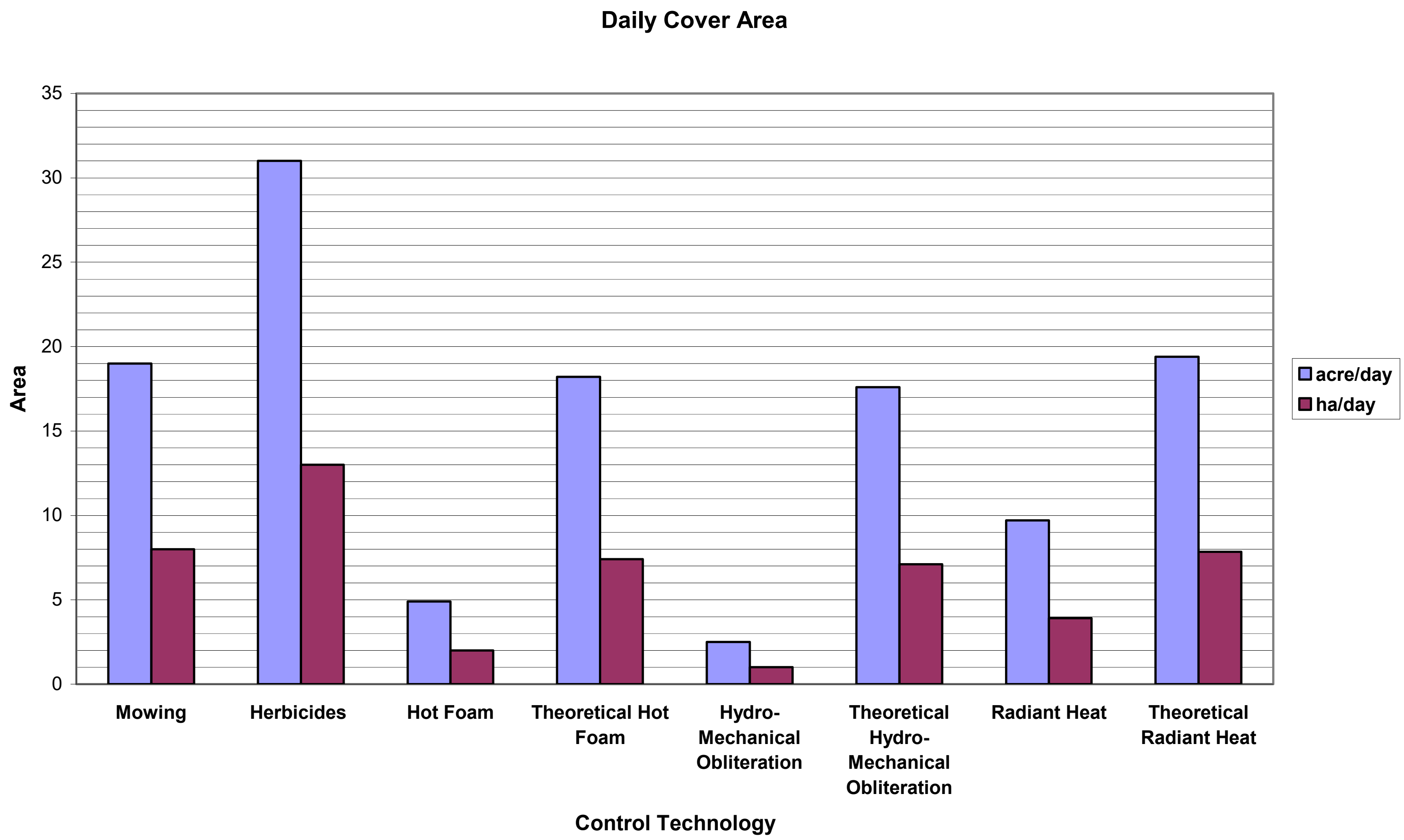


Figure 4.2 Daily Cover Area Comparisons of Vegetation Control Method

The following analyses use information from Figures 4.1 and 4.2 and from information gathered during research to rank each method for each criterion. The important aspects of each scenario and the special criteria used to represent these are discussed in each section. At the end of each section the results of the concept scoring are examined. At the conclusion of this chapter, the most outstanding control methods over all three scenarios are reviewed.

4.2 Guardrails & Posts

As previously discussed, guardrail and post maintenance is difficult because the posts present obstacles to normal maintenance methods. In many instances, additional spot maintenance is required after initial treatment. Furthermore, workers are injured more on guardrails, posts, and metal fasteners that protrude from the posts and guardrails. Successful control methods minimize foot labor, minimize damage to guardrails and posts, minimize the need for additional spot treatments, and minimize worker exposure to guardrail injuries. In addition, those methods that reduce the use of chemicals, and herbicides in particular also score well. The table in Figure 4.3 is the concept scoring matrix used to evaluate the vegetation control technologies for the guardrail & post scenario.

Guardrails & Posts		**Mowing**		**Herbicide**		**Hot Foam**		**Theoretical Hot Foam**		**Hydro-Mech. Obliteration**		**Theoretical Hydro-Mech. Obliteration**		**Radiant Heat**		**Theoretical Radiant Heat**	
SELECTION CRITERIA	**Weight**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**
Public Safety	20																
Minimizes Health Risk	5	5	25	2	10	4	20	4	20	5	25	5	25	5	25	5	25
Minimizes Fire Risk	5	2	10	5	25	5	25	5	25	3	15	3	15	2	10	2	10
Minimizes Traffic Risk	5	3	15	4	20	5	25	5	25	2	10	2	10	4	20	4	20
Minimizes Traffic Congestion	5	4	20	5	25	2	10	4	20	1	5	4	20	3	15	4	20
Worker Safety	20																
Minimizes Manual Labor Intensity	4	4	16	5	20	2	8	5	20	1	4	5	20	3	12	5	20
Minimizes Foot Labor	4	3	12	4	16	1	4	5	20	1	4	4	16	2	8	4	16
Minimizes Safety Equipment Required	4	5	20	1	4	3	12	3	12	2	8	5	20	4	16	5	20
Minimizes Worker Exposure To Guardrail Injuries	4	3	12	4	16	1	4	5	20	1	4	4	16	1	4	4	16
Minimizes Exposure to Chemicals	4	4	16	1	4	3	12	3	12	5	20	5	20	5	20	5	20
Environment	15																
Decreases or Avoids Chemical Concentration	5	5	25	1	5	3	15	3	15	5	25	5	25	5	25	5	25
Decreases or Avoids Runoff	5	5	25	5	25	3	15	3	15	2	10	2	10	5	25	5	25
Decreases or Avoids Erosion	5	5	25	5	25	5	25	5	25	2	10	2	10	5	25	5	25
Herbicide Use	15																
Decreases or Avoids Herbicide Use	15	5	75	1	15	5	75	5	75	5	75	5	75	5	75	5	75
Cost & Efficiency	25																
Minimizes Cost	7	4	28	4	28	2	14	3	21	1	7	4	28	3	21	4	28
Decreases Treatment Time	3	3	9	5	15	2	6	4	12	1	3	3	9	2	6	3	9
Decreases Treatment Frequency	3	4	12	5	15	4	12	4	12	3	9	3	9	4	12	4	12
Resistant to Extreme Weather Conditions	3	3	9	2	6	4	12	4	12	5	15	5	15	3	9	3	9
Easily Applied Around Rail and Posts	3	3	9	4	12	5	15	5	15	3	9	3	9	2	6	2	6
Minimizes Additional Spot Treatments	3	1	3	2	6	5	15	5	15	3	9	2	6	2	6	1	3
Minimizes Damage to Guardrails	3	3	9	5	15	5	15	5	15	1	3	1	3	3	9	3	9
Public Perception	5																
Favorably Affect Public Perception	5	5	25	1	5	5	25	5	25	5	25	5	25	5	25	5	25
Total Score			**400**		**312**		**364**		**431**		**295**		**386**		**374**		**418**
Rank			**3**		**7**		**6**		**1**		**8**		**4**		**5**		**2**

Figure 4.3 Guardrail & Post Scenario Concept Scoring Matrix

4.2.1 Guardrail & Post Scenario Concept Scoring Observations

Those concepts that performed poorly are discussed first, and those that did well are addressed afterwards. In general, the current versions of the alternative vegetation control technologies did not perform well. The radiant heat, hot foam, and hydro-mechanical obliteration ranked fifth, sixth and eighth respectively. These concepts scored poorly for many reasons including but not limited to high cost, low cover area, and low safety levels due to the amount of required foot maintenance. Their automated versions, which improve on all these levels, faired well and are discussed later.

Herbicides also did not score well, ranking seventh in this scenario. This score is not entirely unexpected. Even though herbicides scored high marks for cost and efficiency, the use of chemicals is a huge problem for herbicidal vegetation control. The negative health and environmental effects coupled with the poor public perception creates an environment that is adverse to herbicide use. This adversity is represented in all three of the scenario concept scoring matrices. Herbicide use faired poorly throughout this investigation for the preceding reasons.

The highest score goes to a theoretical automated hot foam control method. This method scored well because it limited foot traffic and worker exposure to injury. In addition, the foam spreads well under the guardrails and additional spot treatments are not required. The method also improves on cost and efficiency over the currently available hot foam technology. The theoretical radiant heat treatment scored second highest for this scenario. This method is beneficial for many of the same reasons as the hot foam. The few downsides are that there is higher fire risk and a greater need for residual spot treatments. Mowing, surprisingly, ranked third for this scenario.

Caltrans current mowing operations are a satisfactory treatment method around the guardrails and posts. Without introducing any new technologies, Caltrans has a feasible alternative to herbicide application.

4.3 Mow Strip

Mow strip maintenance accounts for a large portion of total vegetation maintenance. The most important criteria for mow strip maintenance are efficiency and cost because of the large amount of maintenance required. Some of the other important criteria for successful mow strip maintenance include the minimization of fire risk, effectiveness in extreme weather conditions, minimization of manual labor and foot labor, and the minimization of traffic congestion. The concept scoring matrix shown in Figure 4.4 on following page is used in the evaluation of the technologies for the mow strip scenario.

Mow Strip		**Mowing**		**Herbicide**		**Hot Foam**		**Theoretical Hot Foam**		**Hydro-Mech. Obliteration**		**Theoretical Hydro-Mech. Obliteration**		**Radiant Heat**		**Theoretical Radiant Heat**	
SELECTION CRITERIA	**Weight**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**
Public Safety	20																
Minimizes Health Risk	3	5	15	2	6	4	12	4	12	5	15	5	15	5	15	5	15
Minimizes Fire Risk	7	2	14	5	35	5	35	5	35	3	21	3	21	2	14	2	14
Minimizes Traffic Risk	4	3	12	4	16	5	20	5	20	2	8	3	12	4	16	4	16
Minimizes Traffic Congestion	4	4	16	5	20	2	8	4	16	1	4	4	16	3	12	4	16
Minimizes Damage to Roadway Surface	2	4	8	5	10	5	10	5	10	1	2	2	4	5	10	5	10
Worker Safety	22																
Minimizes Manual Labor Intensity	6	4	24	5	30	2	12	4	24	1	6	5	30	3	18	5	30
Minimizes Foot Labor	5	5	25	5	25	1	5	5	25	1	5	5	25	2	10	5	25
Minimizes Safety Equipment Required	5	5	25	1	5	3	15	3	15	2	10	5	25	4	20	4	20
Minimizes Exposure to Chemicals	6	5	30	1	6	3	18	3	18	5	30	5	30	5	30	5	30
Environment	16																
Decreases or Avoids Chemical Concentration	4	5	20	2	8	3	12	3	12	5	20	5	20	5	20	5	20
Decreases or Avoids Runoff	4	5	20	5	20	3	12	3	12	3	12	3	12	5	20	5	20
Decreases or Avoids Erosion	4	5	20	5	20	5	20	5	20	2	8	2	8	5	20	5	20
Minimizes Fire Risk	4	2	8	5	20	5	20	5	20	3	12	3	12	2	8	2	8
Herbicide Use	15																
Decreases or Avoids Herbicide Use	15	5	75	1	15	5	75	5	75	5	75	5	75	5	75	5	75
Cost & Efficiency	22																
Minimizes Cost	10	5	50	5	50	2	20	3	30	1	10	4	40	4	40	5	50
Decreases Treatment Time	4	3	12	5	20	2	8	3	12	1	4	3	12	3	12	3	12
Decreases Treatment Frequency	4	4	16	5	20	4	16	4	16	3	12	3	12	4	16	4	16
Resistant to Extreme Weather Conditions	4	3	12	2	8	4	16	4	16	5	20	5	20	2	8	2	8
Public Perception	5																
Favorably Affect Public Perception	5	5	25	1	5	5	25	5	25	5	25	5	25	5	25	5	25
Total Score			**427**		**339**		**359**		**413**		**299**		**414**		**389**		**430**
Rank			**2**		**7**		**6**		**4**		**8**		**3**		**5**		**1**

Figure 4.4 Mow Strip Scenario Concept Scoring Matrix

4.3.1 Mow Strip Scenario Concept Scoring Observations

As in the previous section, the poor performers are addressed first and these are followed with the high scorers. The issues with the poor performers are pretty much the same as those in the guardrail scenario. The current alternative methods all performed poorly again, especially the hot foam and high pressure water. Herbicides also scored low again for the same reasons as above. None of the poor performers are really a surprise, but those technologies that scored high are of great interest.

The theoretical radiant heat technology scores the highest and, separated by only three points, is mechanical mowing. Both of these methods have moderate fire risks associated with them, yet the risk of fire is the only major downside. Both methods have low costs, high daily cover areas, minimal foot maintenance, minimal required safety equipment, and no chemical use. Theoretical hot foam and theoretical hydro-mechanical obliteration also scored high and scored within a point of each other. Hydro-mechanical obliteration requires more safety gear and is more of a danger to passing traffic. Hot foam scored lower because it uses some chemicals and is more costly and less efficient than mowing or theoretical radiant heat technologies.

4.4 Urban Vegetation

The control of urban vegetation is uniquely challenging for Caltrans maintenance crews. Vegetation that grows in cracks in the pavement is mainly addressed during crack sealing and resurfacing of the roadway. This vegetation is both unsightly and damaging to the roadway. There are no clear or preferred methods for regular maintenance of urban vegetation. Most of the vegetation is on the roadway surface and not on the roadside, making foot labor extremely dangerous. In addition, the treatment area may require continuous or spot treatment depending on the circumstances. Therefore, the important aspects of a successful urban vegetation control method include ability to treat vegetation in tight spaces, efficient spot treatment capabilities, minimal damage to roadside and roadway barriers, and minimal foot labor. Fire risk is of minimal importance in this scenario because most of the vegetation is surrounded by cement and there is little risk of fire. Therefore, fire risk is not included as a criterion in this scenario. Figure 4.5 contains the concept scoring matrix for control technology performance in the urban vegetation control scenario.

Urban Vegetation		Mowing		Herbicide		Hot Foam		Theoretical Hot Foam		Hydro-Mech. Obliteration		Theoretical Hydro-Mech. Obliteration		Radiant Heat		Theoretical Radiant Heat	
SELECTION CRITERIA	**Weight**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**	**Rating**	**Weighted Score**
Public Safety	20																
Minimizes Health Risk	4	5	20	2	8	4	16	4	16	5	20	5	20	5	20	5	20
Minimizes Traffic Risk	4	3	12	3	12	5	20	5	20	2	8	3	12	4	16	4	16
Minimizes Traffic Conges ion	4	4	16	5	20	2	8	4	16	1	4	4	16	3	12	4	16
Minimizes Damage to Roadway Surface	8	1	8	5	40	5	40	5	40	1	8	1	8	5	40	5	40
Worker Safety	20																
Minimizes Manual Labor Intensity	4	5	20	5	20	2	8	5	20	1	4	5	20	3	12	5	20
Minimizes Foot Labor	8	5	40	5	40	1	8	5	40	1	8	5	40	2	16	5	40
Minimizes Safety Equipment Required	4	5	20	1	4	3	12	3	12	2	8	5	20	4	16	4	16
Minimizes Exposure to Chemicals	4	5	20	1	4	3	12	3	12	5	20	5	20	5	20	5	20
Environment	12																
Decreases or Avoids Chemical Concentration	6	5	30	1	6	3	18	3	18	5	30	5	30	5	30	5	30
Decreases or Avoids Runoff	6	5	30	5	30	3	18	3	18	2	12	2	12	5	30	5	30
Herbicide Use	15																
Decreases or Avoids Herbicide Use	15	5	75	1	15	5	75	5	75	5	75	5	75	5	75	5	75
Cost & Efficiency	28																
Minimizes Cost	4	5	20	5	20	2	8	3	12	1	4	4	16	4	16	5	20
Decreases Treatment Time	4	3	12	5	20	2	8	3	12	1	4	3	12	3	12	3	12
Decreases Treatment Frequency	4	4	16	5	20	4	16	4	16	3	12	3	12	4	16	4	16
Resistant to Extreme Weather Conditions	4	3	12	2	8	4	16	4	16	5	20	5	20	2	8	2	8
Minimizes Damage to Roadway and Barriers	4	2	8	5	20	5	20	5	20	1	4	1	4	5	20	5	20
Effecient For Spot Application	4	2	8	5	20	3	12	5	20	3	12	4	16	3	12	3	12
Easy Treatment In Confined Spaces	4	1	4	5	20	5	20	5	20	4	16	4	16	3	12	3	12
Public Perception	5																
Favorably Affect Public Perception	5	5	25	1	5	5	25	5	25	5	25	5	25	5	25	5	25
Total Score			**396**		**332**		**360**		**428**		**294**		**394**		**408**		**448**
Rank			**4**		**7**		**6**		**2**		**8**		**5**		**3**		**1**

Figure 4.5 Urban Vegetation Scenario Concept Scoring Matrix

4.4.1 Urban Vegetation Scenario Concept Scoring Observations

The poor performers for the urban vegetation maintenance are covered first. Hot foam application, herbicide spray, and hydro mechanical obliteration rank sixth, seventh, and eighth respectively. Hydro-mechanical obliteration does poorly in the most important categories in this scenario. This method is damaging to the road surface and requires extensive application on foot. In addition, hydro-mechanical obliteration treatments are costly and slow. Herbicides score low for the same reasons as in previous scenarios. The short application time and the low cost are not enough to offset the downsides to herbicide use. Hot foam also does poorly because of high cost, required foot application, chemical use, and low daily cover area. The radiant heat treatment method, which does not perform particularly well in the other scenarios, actually performed much better in this scenario. This method, along with the other high scoring methods, is discussed in the proceeding paragraph.

Theoretical radiant heat and the unmodified radiant heat methods are ranked first and third respectively for this scenario. Third is the highest rank for any of the unmodified alternative methods achieved in any scenario. The theoretical radiant heat technology performs better than the current version in cost, daily coverage area, and foot labor criteria. These two versions of the same control method are strong in nearly every criterion except in treatment of confined spaces and in ease of spot application as the burner is always on. Theoretical hot foam treatments also scored high and the method ranks second overall for this scenario. This method scores high in all criteria except for its use of chemical additives, runoff, cost, and daily cover area. Mowing and theoretical hydro-mechanical obliteration scored within a few points of each other and ranked fourth and fifth respectively. Both methods scored fairly well but cause damage to the roadway and may kick debris up into traffic. Overall, theoretical hot foam and both versions of the radiant heat technology performed the best in this scenario.

4.5 Summary

This chapter is an engineering analysis of the potential success of standard and alternative vegetation control methods in specific scenarios. This study serves as a preliminary determinant of the feasibility of adopting alternative roadside vegetation control methods. In addition, this investigation extricates those alternative methods that may have the least potential, and allows research and development to focus on the most promising alternative technologies. This scoring style may be adopted, adjusted and utilized in the future to address the feasibility of new technologies, additional scenarios, and maintenance tasks outside the realm of vegetation maintenance. If any methods dominate the scoring, those will be recommended for future research, adaptation, and possible implementation. A concept scoring matrix is used, for each scenario, to rank each of the methods for specific criteria. As mentioned, some of the criteria are specific to the scenario while other criteria are scenario specific. The control technology scores for each scenario are displayed in Figure 4.6. The maximum score for each scenario is 500 points, while the lowest is 100 points. The methods are listed from highest to lowest score with the high scenario score in bold.

Total Rank	**Mow Strip**	**Guardrails & Posts**	**Urban Vegetation**	**Avg**
Theo. Radiant Heat	**430**	418	**448**	432
Theo. Hot Foam	413	**431**	428	424
Mowing	427	400	396	408
Theo. Hydro-Mech. Obliteration	414	386	394	398
Radiant Heat	389	377	408	391
Hot Foam	359	364	360	361
Herbicides	339	312	332	328
Hydro-Mech. Obliteration	299	295	294	296
Average	384	373	383	380

Figure 4.6 Concept Matrix Scores and Averages

This figure shows a few important results of this analysis. First of all, the theoretical methods all outperform the existing state of the technology. This is not surprising as the theoretical methods are all automated and better adapted for roadside vegetation maintenance. The theoretical roadside versions also performed the best overall, with radiant heat and hot foam applications scoring the highest. The theoretical hot foam method scores the highest in the guardrail & post scenario, while the theoretical radiant heat method is the best in the mow strip and urban vegetation scenarios. Mowing rounds out the top three with solid scores in each scenario. All of the top three methods averaged a score over 400 points.

This analysis also confirms the results of previous analyses. The current state of many alternative methods is not good enough to compete with current Caltrans control methods. This has been an continuing problem for advocates of implementation of alternatives to herbicide use. This study demonstrates, however, that although the current technological state of alternative vegetation control methods are not competitive with existing Caltrans methods, development of alternative methods may lead to technology that is competitive and even superior. The assumptions made for the theoretical control methods are meant to be conservative and logical improvements in alternative method technology. It is possible that research may yield superior technology or find mitigating factors not considered in this analysis. Based on the results of this research, future research is warranted and seems likely to yield improvements in Caltrans vegetation maintenance operations.

The next chapter provides a discussion of the recommendations and conclusion of this paper. This chapter addresses possible combinations of vegetation control technologies. In addition, recommendations are made concerning future research for the alternative technologies that show the most promise. The future of these alternative methods and their possible inclusion in Caltrans' vegetation maintenance program is also considered. Finally, the observations and revelations reached in this study are included in the conclusion.

CHAPTER 5 Recommendations and Conclusions

This chapter covers recommendations and conclusions drawn from the research. Recommendations are made concerning the combination of alternative and current technologies. A combination of two or more methods may lead to the development of more effective and efficient control methods. In addition, research recommendations are included for those technologies that show the most promise. This research should include cost based analysis and technological research that seeks to develop and implement vegetation control methods to replace herbicide application. Finally, the results of the analysis and the revelations from this research are covered in the conclusion.

5.1 Recommendations

The following recommendations should be a guide towards the development of vegetation control techniques that provide safe, environmentally friendly, cost effective, and efficient means of vegetation control.

5.1.1 Combined Technologies

The combination of technologies is an important design and development engineering tool. Sometimes, radically different technologies are incorporated into the same device to improve the breadth of application and thus the desirability of a product. An example is a cell phone that also takes pictures. Picture taking and telephone correspondence are not related technologies, yet the combination of these two technologies into one device makes for a highly desirable electronic gadget. This type of combination is discussed below and includes the Advanced Roadway Debris Vacuum (ARDVAC).

A more applicable version of a combination of technologies is a part of the concept scoring process mentioned above. After the concepts are scored, as in Chapter 4, the strengths and weaknesses of each method are more pronounced. When the scoring is completed, it is time to see if any combinations or changes improve the concepts. Ulrich and Eppinger explain it well:

> "Although the formal concept generation process is typically completed before concept selection begins, some of the most creative refinements and improvements occur during the concept selection process as the team realizes the inherent strengths and weaknesses of certain features of the product concepts."[23]

In this way, it is possible that some of the vegetation control technologies, when combined together, may perform better than each respective technology. The combination of technologies is usually completed during the concept selection phase in product development. Recommendations regarding possible combinations of existing and alternative vegetation control methods are also included below.

Hot Foam & Radiant Heat

Hot foam and radiant heat vegetation control methods are the first candidates for possible combination. The theoretical versions of each of these methods scored well in the concept scoring matrix independently. In addition, both technologies work in generally the same way: applied heat melts the cell structure of the unwanted vegetation, inhibiting growth and killing the plant. These two methods have properties that support the downsides of each other. The main problem with hot foam application is the high water consumption, while the main issue with radiant heat is the fire risk. If these two methods are combined, much less water would be necessary for treatment, and the water that is used would prevent the start of fire. It is also possible that a combined hot foam and radiant heat technology would not require a foam additive for heat containment, as the extra heat could be supplied by the radiant heating elements. A method that adequately combines both of these technologies would be very competitive with existing roadside vegetation control technology. In addition, this technology may be less restricted in different vegetation control scenarios.

Hot Foam & Mowing

A mower and hot foam control method combination is also a prime candidate for future research. Similar to hot foam and radiant heat technologies, mowing and hot foam application both scored well in the concept scoring matrix. In addition, these technologies each possess qualities that may offset the other technology's downsides. As seen in the concept scoring matrices in the previous chapter, the main downside to mowing is fire risk. As mentioned for the preceding combination, the downside to hot foam application is the large demand for water. If these two methods could be combined, the fire risk and the water usage could be lowered dramatically. This method is similar to and could be modeled on the existing mowing and herbicide combination technology, which cuts the vegetation and then applies herbicides to prevent growth. The Brown Brush Monitor is a good example.[24] More research is required to determine if development and implementation of this combined technology is feasible for Caltrans.

Hot Foam System and Other Attachments for the ARDVAC

Figure 5.1 Advanced Roadway Debris Vacuum in Operation

The ARDVAC, pictured above in Figure 5.1, is a roadside maintenance tool developed at AHMCT out of a need to remove litter and debris from medians and roadsides without exposing workers to injury. The ARDVAC uses an articulated nozzle attached to existing vacuum vehicles, which allows for removal of litter and debris from the roadside and around guardrails.[4] Workers are removed from the roadway for some of the most hazardous maintenance tasks with the remote controlled boom and nozzle. Since the development of the articulated nozzle, AHMCT has researched additional nozzle attachments to broaden the scope of the ARDVAC. Currently, AHMCT researchers are looking into tumbleweed cutting attachments and a system for the interchanging of various tool heads. The ARDVAC is a prime platform for the adaptation of alternative technologies for the roadway because of the remote control articulated boom and nozzle.

A hot foam applicator could be attached to the end of the boom with a hot water storage tank setup in the back of the truck usually reserved for vacuumed debris. The articulated nozzle would allow the hot foam to be applied around the guardrails automatically, without any need for workers on the roadway. The hot foam application method scored the highest for the guardrail scenario, and when combined with the ARDVAC, could prove to be a valuable control method. This appears to be the quickest road to implementation of an alternative control technology and more research is required to determine if such an ARDVAC attachment is feasible. Research in this area is highly recommended.

Some other possible ARDVAC attachments include a mowing system, a high pressure water system, and some combinations of technologies. The duff that usually remains after mowing could be sucked up into the ARDVAC, removing a large part of the potential fire hazard associated with mowing operations. In addition, a high pressure water system could be used for removing ice plant and other vegetation that creeps onto roadways from the safety of the ARDVAC cab. Finally, the ARDVAC could serve as a platform for the combination of many technologies and then the most appropriate methods could be used depending on the maintenance landscape. The ARDVAC could serve as an all purpose vegetation control vehicle for Caltrans, and continued research is recommended for the development of alternative vegetation control attachments for the ARDVAC.

5.1.2 Research

Based on the results of the research and analysis completed in this report, appropriate recommendations are made for future research. Research is the bridge between the currently available alternative technologies and the full implementation of these methods into Caltrans' vegetation control plan. Research requires the appropriate investments in time, energy, resources, and money. Many projects have failed due to a lack of one or more of these components, and it is the hope that future vegetation control research is adequately supported. There are a couple of areas where future research is vital to continued development of alternative control technologies and Caltrans' vegetation maintenance plan. These include cost based analysis, research and documentation, level of care determinations, and the prototyping of alternative methods.

One of the main problems in comparing the cost effectiveness of each maintenance method is that expenses are not tracked in a concise standardized manner. For instance, when researching the vegetation maintenance methods discussed in this report, it was difficult to determine how costs were calculated. Basic cost calculations were done so that the methods could be compared fairly in this analysis. As mentioned previously, many costs were not included in the calculations such as research, maintenance, and transportation costs. What is needed is a true accounting of the costs associated with each vegetation maintenance technology so that valid comparisons can be made. It is true that it would be very time consuming to track every single expense, but the numbers now are not nearly accurate enough for a good cost based method comparison. More research is needed in this area, and better accounting by Caltrans would help. Both AHMCT and Caltrans benefit from knowing the true cost of each current maintenance method and those advanced technologies developed here at AHMCT and elsewhere.

Another issue that requires study is the required level of care of vegetation maintenance. Caltrans currently tracks the level of maintenance performed for each stretch of highway in California. There appears to be little known, however, about what is the ideal level of control. What is the level of care that maximizes resources and minimizes cost? How many times a year should maintenance crews mow grass to minimize fire risk and also maintenance costs? It is important to know the answer to these and other related questions. It is possible that it is more efficient for Caltrans to forgo some roadway maintenance and focus on vegetation maintenance or vice versa. This is a very complex issue, though, as the public, Caltrans and politicians have differing views on the best maintenance balance. A look into specific scenarios might be appropriate so that smaller areas are tackled instead of trying to solve the whole problem at once. Research in this area will allow Caltrans to maintain the best levels of care given its operation budget.

Finally, it is important that alternative control technologies are developed for the roadway environment. It is important to compare apples to apples when comparing maintenance tasks. Previously, maintenance technology comparisons looked at alternative vegetation methods at their current technological state instead of in a state that is appropriate for Caltrans roadside maintenance crews. This is not a viable comparison. It is important that research is done so that promising alternative vegetation control methods can be tested fairly against current Caltrans methods. An attempt was made to do this in this report by including theoretical upgrades of alternative control technologies. More research is required so that actual prototypes can be tested and, if successful, adopted for use by Caltrans.

5.2 Conclusion

The big question of this work is whether or not alternative vegetation control methods are competitive with herbicide application. After all the research and analysis, the answer to that question appears to be yes. Despite the fact that herbicides are touted as cheap and effective, the negative public perception, environmental risk, and harmful health effects more than detract from the positives. It is even possible that herbicides are not as cheap as thought. New IMMS information, provided by Caltrans, demonstrates that the low cost of herbicides calculated in this research may be far too low.[8] In this paper herbicides represent about 5% of the cost, while IMMS represents chemical herbicide costs at 30% to 50% of the total cost. This further harms the case for continued widespread herbicide use. In addition, herbicide use scored poorly compared with most alternative methods regardless of the scenario. Chemical herbicides have their place in Caltrans' vegetation control program, but widespread use is mandated for reduction.

PART 1: Alternatives to Mowing and Herbicides

Alternative technologies may prove safer, more efficient, and more cost effective solutions to the herbicide use, if only they are developed for California roadside use. Eventually, alternative technologies may become the standard for vegetation maintenance throughout California.

PART 1 References

1. Californians for Alternatives to Toxics, The Poisoning of Public Thoroughfares: How Herbicides Blight California's Roads, [updated 1999; cited October 12, 2005] http://www.alternatives2toxics.org/catsoldsite/roadside_report.pdf

2. California Department of Transportation (Caltrans), California Roadsides: A New Perspective, [update January 1997; cited October 14, 2005]. Jones & Stokes Associates, Inc. http://www.dot.ca.gov/hq/maint/roadside.htm

3. Legislative Analyst's Office, Analysis of the 2004-2005 Budget Bill: Department of Transportation, [updated February 2004; cited October 14, 2005] http://www.lao.ca.gov/analysis_2004/transportation/trans_04_2660_anl04.htm

4. Advanced Highway Maintenance and Construction Technology Research Center (AHMCT) [cited October 15, 2005] http://ahmct.ucdavis.edu/

5. California Department of Transportation (Caltrans), Maintenance Manual, Chapter C2: Vegetation Control, [cited October 19, 2005] http://dot.ca.gov/hq/maint/manual/chc2(final).pdf

6. James Rathke, Linda Simpson, and Terri Porter, LIM Task Kick-Off Meeting, September 12, 2005

7. California Department of Transportation (Caltrans), Environmental Impact Report on Caltrans' Vegetation Control Program, May 1992, Jones & Stokes Associates

8. California Department of Transportation (Caltrans), Integrated Maintenance Management System (IMMS), Fiscal Year 2004/2005 Vegetation Control Plan Report, September 9, 2005

9. Jason McPhee, Design and Analysis of an Oscillating Cutting System for a Vehicle-Mounted Automated Roadway Debris Vacuum, June 2005

10. Masoud Kayhanian, John Johnston, Howard Yamaguchi, and Steve Borroum; Caltrans Storm Water Management Program [update 2000; cited November 22, 2005], Stormwater Journal http://www.stormh2o.com/sw_0103_caltran.html

11. James Rathke, Caltrans Area Superintendent, Headquarters Roadside Maintenance, Email Correspondence, December 1, 2005

12. California Department of Transportation (Caltrans) Division of Equipment, Caltrans Equipment Catalog: Landscape Maintenance, [update October 25, 2005; cited December 12, 2005]
http://www.dot.ca.gov/hq/eqsc/EquipCatalog/

13. Mandy Tu, Waipuna™ Hot Foam System, The Nature Conservancy [update January 2005; cited December 14, 2005]
http://tncweeds.ucdavis.edu/tools/hotfoam.html

14. William Quarles, Improved Hot Water Weed Control System, IPM Practitioner XXIII(1) [update January 2001; cited December 14, 2005]
http://www.waipuna.com/releases/JAN2001.PDF

15. Sandra Robinson, The Use of Steam as an Alternative Herbicide, [cited December 14, 2005]
http://filebox.vt.edu/users/sarobins/robinson.htm
http://filebox.vt.edu/users/sarobins/robinsn2.htm

16. Cameron Colson, Hydro-Mechanical Obliteration Meeting, October 5, 2005

17. California Department of Transportation (Caltrans), Roadside Management Toolbox, Caltrans Website [update October 2005; cited December 19, 2005]
http://www.dot.ca.gov/hq/LandArch/roadside/

18. Northwest Coalition for Alternatives to Pesticides (NCAP), Radiant Heat Weeders: Managing Weeds Without Herbicides, [update Fall 2003; cited December 21, 2005]
http://www.pesticide.org/radiant.html

19. Forevergreen™ Chemical Free Weed Control [cited December 21, 2005]
http://www.chemfree-weedcontrol.com/index.shtml

20. Weed Control ECOflame® [cited January 2, 2006]
http://www.weedcontrol.nl/Download/brochure_ecoflamecity_en.pdf

21. James Rathke, Caltrans Area Superintendent, Headquarters Roadside Maintenance, Email Correspondence, December 20, 2005

22. Kagan Owens, The Right Way to Vegetation Management, Beyond Pesticides, [updated Spring 1999; cited January 5, 2006]
http://www.beyondpesticides.org/infoservices/pesticidesandyou/Spring%2099/The%20Right%20Way%20to%20Vegetation%20Management.pdf

23. Ulrich, Karl T. and Steven D. Eppinger. Product Design and Development, 3rd edition. San Francisco: McGraw Hill, 2004.

24. Brown Manufacturing Corporation, Brown Brush Monitor [cited February 14, 2006] http://www.brownmfgcorp.com/bmonitor/brown_brushmon_page2.htm

PART 2: DEVELOPMENT OF AN AUTONOMOUS MOWER

CHAPTER 1 Introduction and Project Development

1.1 Project Preface

The California Department of Transportation (Caltrans) is responsible for the sustained maintenance of over 15,000 miles (24,000 km) of highway and more than 230,000 acres (93,000 ha) of right-of-way [1]. Included among the long list of maintenance tasks is vegetation management, which makes up a substantial portion of Caltrans maintenance budget (approximately 40% [2]). Primary vegetation control needs include maintenance of roadbed integrity, visibility, drainage, noxious weed control, aesthetic roadside appearance, and fire protection [1]. Proper vegetation management is crucial to maintaining sustained levels of public safety.

Since the 1950's, Caltrans has relied heavily on chemical control methods (e.g. herbicide use) for vegetation management. These methods have been considered to be the most practical due to their cost effectiveness and low labor requirements. However, in 1992, an Environmental Impact Report (EIR) was released which addressed potential health and ecosystem concerns associated with chemical use. In response, Caltrans adopted an Integrated Vegetation Management (IVM) Program with the goal of reducing herbicide use 50% by the year 2000, and 80% by 2012. Due to budget limitations, however, Caltrans has found it difficult to cut back on herbicide use and has been forced to search for alternative low cost vegetation control technologies.

In order to assist Caltrans in meeting their herbicide reduction objectives, they have commissioned support from the Advanced Highway Maintenance and Construction Technology (AHMCT) Research Center. The AHMCT Research Center has assumed the responsibility of research into new innovative mechanical vegetation control technologies. Some of these alternative technologies includ: hydro-mechanical obliteration, hot foam, and radiant heat [2]. The goal has been to identify those technologies with the greatest potential to meet Caltrans needs, and to select those technologies for further research and development. Caltrans needs include, but are not limited to, increasing public safety, increasing worker safety, improving environmental quality, reducing herbicide use, decreasing life cycle costs, and improving public perception.

In Teeter-Balin's study [2], a quantitative concept scoring matrix was used to compare several alternative vegetation control technologies to existing practices. Based upon Caltrans desires, Balin derived a weighted scoring assessment in six major categories: public safety, worker safety, environment, herbicide use, cost & efficiency, and public perception. He found that new technologies such as hydro-mechanical obliteration, hot foam, and radiant heat did not perform well. He did, however, conclude that development of automated technology which incorporates radiant heat and hot foam application has strong potential for successfully replacing current practices. He also determined that Caltrans current mowing practices scored surprisingly well and would be a suitable alterative to herbicide application.

As a supplementary study, this report explores autonomous mowing technology in an effort to identify alternative vegetation control methods to complement Caltrans current mowing practices. The goal is to identify alternative methods that will enable Caltrans to relieve their dependence on herbicide use, reduce cost, increase public and worker safety, and raise public perception.

1.2 Motivation for Autonomous Mowing Technology

Mowing of medians and rights-of-way is an important vegetation management practice for Caltrans, and it is the only other currently used major vegetation control alternative to herbicide treatment [2]. Mowing is necessary for a variety of reasons, including: the decline of roadside fires, improved aesthetics, enhanced driver visibility, reduced pavement damage, and prevention of drainage obstruction and the spread of noxious weeds [1]. Nevertheless, mechanical mowing tends to be a labor intensive procedure requiring a vast array of expensive and specialized equipment. As a result, the task of mechanical mowing has much need for improvement.

In addition, mechanical mowing has many public and worker safety issues, which require further attention. For example, clear strip mowing (4-8 ft mowing adjacent to the roadside) often requires lane closures, congesting traffic and placing both workers and the public in danger [2]. Other dangerous manual labor tasks such as weed whacking require workers to be on foot, with little protection along busy freeways. Additional safety issues root from fire concerns, for example, the contact of mower blades with rocks may spark and ignite surrounding vegetation. Also, the cut biomass left behind after mowing may act as a fuel source for roadside fires. Roadside fires are of increasing concern throughout California because they place the public in danger and are costly.

In comparison, herbicide treatment has been considered by Caltrans to be the most effective and lowest cost alternative to mowing. However, since the Environmental Impact Report (EIP) in 1992 identified many of the inherent dangers associated with herbicide application, Caltrans has been faced with continual pressure to reduce herbicide use. As a result, AHMCT has been called upon to research the potential of alternative vegetation management techniques. Teeter-Balin concluded that mechanical mowing, even in its current state, is a very practical alternative to herbicide treatment [2].

As noted earlier, mechanical mowing is slow, costly, often places workers in danger, increases duff fires, results in lane closures, and requires frequent yearly treatments. Why then did mechanical mowing score so well in comparison to herbicide application? Caltrans simply cannot rely on herbicide use as they once had, and as a result herbicide application is no longer an acceptable governing practice. In Balin's study [2] *environmental impact* and *reducing herbicide use* were important qualifications for a successful vegetation maintenance alternative. These factors led Balin to determine that mechanical mowing was currently the most practical technology.

1.2.1 Autonomous Solution

Given that mechanical mowing has been regarded as a sensible alternative to herbicide use, why not explore ways to improve the downsides of mowing? The drawbacks associated with mechanical mowing lend well into the attractiveness of an autonomous solution. An autonomous solution could mitigate negative factors such as: safety, labor cost, frequency of treatments, and slow speeds. Such a unit would be able to work continuously, even overnight, with limited human interaction, leading to the enhancement of worker and public safety. With multiple units, labor cost might be significantly reduced, since a single worker could deploy many units (possibly 4-6) along a roadside in a single day. He could then come back at a later time to find the vegetation under control.

Caltrans typically mows medians and shoulders twice a year, however, as a result of political pressure and fire risk, areas may be treated more frequently [2]. A large reduction in vegetation maintenance cost (from an autonomous solution) would permit Caltrans to mow more often. This would lead to less cut biomass, which would biodegrade faster and reduce wild fire problems. It would also improve aesthetics, safety, reduce the spread of noxious weeds, and as a whole improve Caltrans public image as a technologically advanced agency. As a final point, with the support of a new cost effective mowing technology, Caltrans will be better able to cut back on herbicide use, while maintaining a continued high level of vegetation control.

1.2.2 Importance of a Cost Effective Design

In recent years, significant progress has been made in the area of field robotics. This has been due in part to the improvement in sensor technology, along with size reduction and speed improvement of computer technology. These improvements have also led to the lowering of sensor and computer technology prices, making autonomous solutions more practical. With these technological advances, some commercial products, such as the ‘Roomba’ by iRobot (an autonomous vacuum cleaner) and the ‘Robomower’ by Friendly Robotics (an autonomous lawn mower) have entered the market with success. These solutions have exploited low cost sensor technology along with smart manufacturing practices and good marketing strategies to produce cost effective, well appraised products [3].

Many other companies, such as Electrolux, have tried their hand in the robotics market, with little success. Electrolux developed an autonomous vacuum cleaner similar to iRobot’s, making use of sophisticated sensor technology. Instead of using simple bumper obstacle detection as iRobot had, expensive sonar units were used, as well as sophisticated mapping and path planning technology. Even though Electrolux’s unit was much more sophisticated, it sold on the market for $1,600 while iRobot’s sold for only $200. Differences such as these led to Electrolux’s failure to generate a profit. Simply put, their autonomous solution was too expensive [3]. It is obvious that the success of an autonomous vegetation maintenance unit is governed by the ability to engineer a cost effective solution. That is, the long term net worth should outweigh the short term cost investment. Hardware and sensor technology play a key role in designing a cost effective solution. As a result, this report is geared towards the selection of a suitable low cost hardware and sensor array.

1.3 Overview of the Environment

Physical agents perform tasks by manipulating the environment around them. Therefore, it is essential to develop a complete understanding of the environment upon which the agent/robot will interact [4]. Vegetation control along highways deals with harsh and uncertain environments. The state of California stretches from the high north Pacific Forest Ecoprovince to the far south American Desert Ecoprovince. Such diversity within California, makes standard vegetation management practices difficult, as each region is faced with its own hardships. Regardless, certain vegetation maintenance tasks persist throughout much of California, such as mowing of clear strips, medians, shoulders, and interchange quadrants [2]. One such region is that of the Central Valley (District-3), which relies heavily on mowing practices. In an attempt to narrow down the regional scope, District-3 will be studied in more depth throughout this report. The goal will be to examine District-3 in an attempt to generalize the predominant vegetation control strategies throughout the state of California, and to identify areas for improvement.

1.3.1 Environmental Scenarios

As noted earlier, Caltrans predominant means for vegetation maintenance consists of herbicide treatment and mechanical mowing. Each method may be employed for a variety of different environmental scenarios. For example, herbicide application is used for vegetation control of clear strips, urban vegetation, noxious weed growth, and landscaping. Similarly, mechanical mowing is utilized for mowing of clear strips, medians, shoulders, interchange quadrants, guardrails, shrub encroachment, and landscaping tasks. It is also noted that Caltrans may use several different techniques for each environmental scenario. For instance, clear strip application typically makes use of specialized spray trucks and/or tractor mounted mowers, and landscaping classically makes use of manual push mowers and backpack herbicide sprayers.

In an effort to better identify environmental scenarios that require improvement, via an autonomous solution, some of the more predominant (costly, unsafe, and environmentally unfriendly) vegetation control scenarios are discussed in more detail below. As specified by Caltrans, areas for improvement include, improved worker safety, reduction of manual labor, lowering of pesticide use, cost reduction and efficiency improvement.

GUARDRAILS: Maintenance of vegetation growth around guardrails, posts, and signs is a unique and important task. Figure 1.1 displays an overpass support which is protected by guardrails. Proper vegetation maintenance is essential for several reasons, including: reduction of fire risk, improved sign visibility, upheld aesthetics, and access for inspection and maintenance [2]. Obstacles such as guardrails, posts, and signs present a challenging vegetation control scenario, since standard mowing units cannot reach such uncertain places. Current control techniques consist of tractor mounted boom mowers and herbicide treatment. Yet, these methods often leave missed patches which must be touched up by workers on foot, who use either weed eaters or chemical hand spraying units [2]. Workers on foot are more susceptible to harm and such labor intensive tasks are costly.

Figure 1.1 Guardrail around Overpass (Picture from [2])

CLEAR STRIPS: Caltrans vegetation control policy [5] calls for narrow clear strips (sometimes called mow strips or fire strips) next to pavement edges in order to control risk of fire, ensure driver visibility, provide space for emergency use, and to preserve the pavement. These strips act as barriers between the roadway and vegetation growth, and reduce fire risk from sources such as car mufflers and cigarettes, which must travel further before reaching vegetation. These strips are typically 4-8 feet wide but may vary depending on factors such as the likelihood of a fire, the configuration of the pavement edge, the type of vegetation, or the chance of fire spreading. Figure 1.2 displays a clear strip recently treated with herbicide.

Figure 1.2 Clear Strip Recently Treated With Herbicide (Picture from [2])

Before the environmental impact report (EIR) of 1992, Caltrans emphasized the use of pre-emergent chemicals to suppress weed growth along the clear strips. However, with pressure to reduce chemical application volume, Caltrans has adopted an Integrated Vegetation Management (IVM) Program that details alternative techniques. Some of these techniques require hardscaping designs such as pavement or weed mats in clear strip areas, as defined in [1]. Other techniques attempt to reduce herbicide use by combining frequent mowing with chemical treatment. New roadside hardscaping designs are expensive, however, and mowing is typically not as effective as herbicide application. As a result, along with an increase in noxious weeds and fires, herbicide reduction has slowed.

MEDIANS & SHOULDERS: Unlike clear strips, medians and shoulders such as the ones shown in Figure 1.3, are only treated with herbicide to impede noxious weed growth. These sections of vegetation are classically mowed to maintain line of sight, reduce fire risks, and for appearance [2]. Although medians and shoulders vary from highway to highway, they make up the majority of the annual statewide mowing acreage [6]. The annual frequency of shoulder and median mowing is dependent on a number of factors including, fire risk, aesthetics, vegetation height, funding, and driver visibility, but, on average, is performed twice a year. In order to reduce summer time fire risks, mowing is classically done during the spring time, when vegetation growth is at its peak.

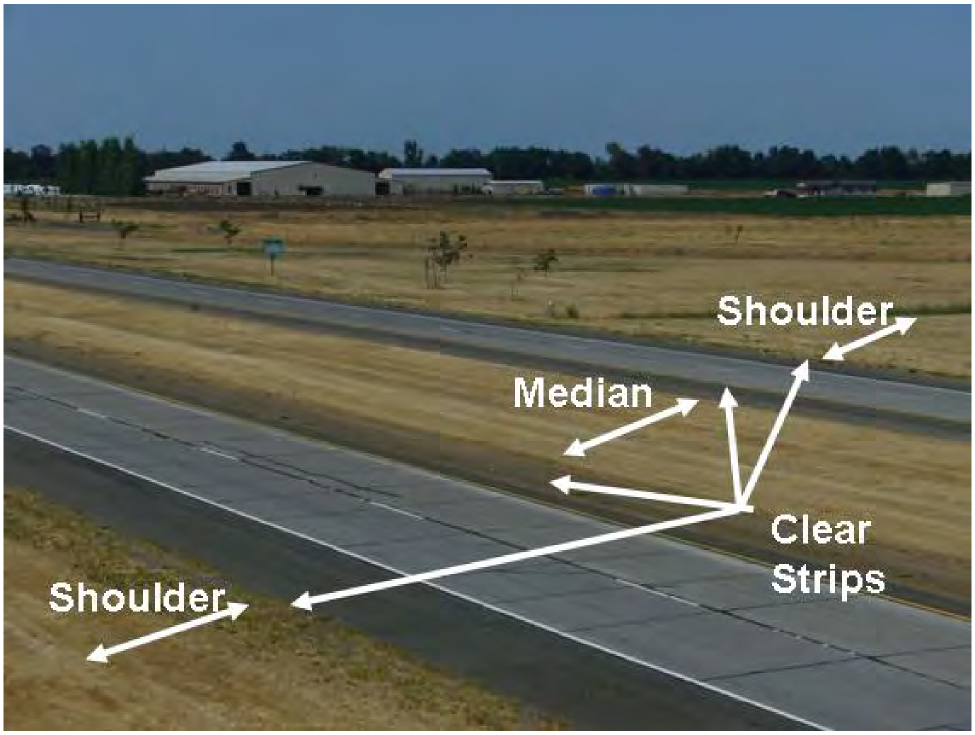


Figure 1.3 Wide Median & Shoulders Which Require Mowing

Caltrans often employs wide tractor mounted mowers to cover large median and shoulder sections. The width of the mowers is dependent on the variability of the terrain. If the terrain is steep and/or rough, smaller units, such as skid-steer loaders or boom mounted mowers may be used to reach these harsh regions. The ability to position mowing equipment within the medians and shoulders is an important safety advantage over clear strip and guardrail maintenance. High capacity mowing units tend be cheaper per unit area, but they are subject to many additional costs such as, travel distance to the job site, equipment expenses, and downtime.

INERCHANGE QUADRANTS: Quadrants, which are sections of vegetation between interchanges and onramps, such as those shown in Figure 1.4, are another scenario of interest. These sections are typically mowed twice a year, and like shoulders and medians, account for a large portion of the annual planned mowing acreage. Depending on the regional location (typically urban areas), quadrants such as those shown in Figure 1.4 may be landscaped, in which case they may be mowed up to 32 times per year for aesthetic appeal. The importance of vegetation control in these areas is much the same as median and shoulder mowing, that is, to ensure driver visibility, reduce fire risks, and for appearance.

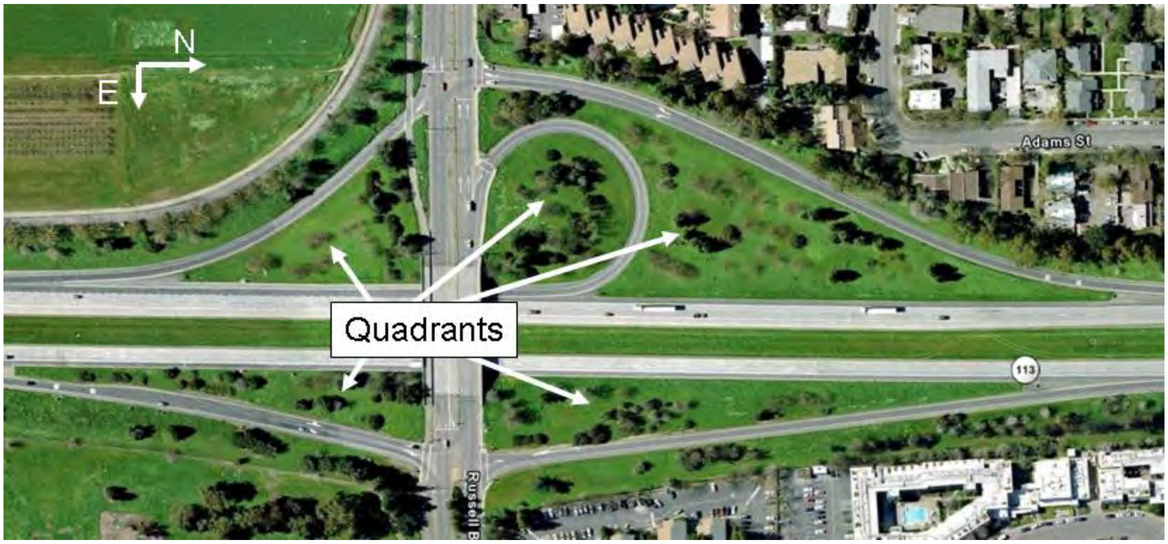


Figure 1.4 Quadrant Areas along a Section Of HW-113 In Davis (Picture From Google Earth)

In medians and shoulders standard mowing equipment is used. However, due to the variability of the terrain and the number of trees and shrubs, high capacity mowers typically cannot be utilized. In non-landscaped areas, small single person units are used and in landscaped areas, varying techniques may be used from push mowers to weed eaters.

LANDSCAPING: Locations which are designated as landscaped areas are maintained more thoroughly, usually for aesthetic appeal in urban areas. Vegetation control strategies in these areas are not limited to mowing and herbicide, and might include, mulching, weeding, irrigating, fertilizing, shrub maintenance, pest management, and pruning. Landscaping is mentioned here because mowing of landscaped areas makes up a notable portion of annual mowing acreage as a result of frequent yearly mowing. Although mowing makes up only a small portion of landscaping jobs, it is costly and labor intensive.

1.3.2 Targeted Environmental Scenario

In Section 1.3.1 a variety of environmental scenarios were introduced, from guardrail vegetation control to high acreage maintenance of quadrants, medians and shoulders. For each scenario, an overview of Caltrans current vegetation maintenance practices, along with various shortcomings, were introduced. In order to identify important overall areas for improvement, data for District-3 will be included in this section. The overall goal is to identify scenarios in which an autonomous solution will improve worker safety, reduce manual labor, lower pesticide use, and reduce maintenance cost. Although the list of scenarios introduced in Section 1.3.1 is only a small portion of Caltrans overall vegetation maintenance scenarios, it represents the majority of the overall statewide vegetation control acreage and budget. Much of the data listed throughout this subsection comes from Caltrans Integrated Maintenance Management Society (IMMS) Annual Report [6].

For the Fiscal 2004/05 year, Caltrans IMMS report planned a total of 29,137.81 annual acres (11,800 ha) to be maintained for District-3. These numbers account for maintenance families C (vegetation control), E (landscaping) and G (Public Facilities). Over 63% of this total acreage was accounted for by mechanical control and about 34% by chemical control. Further analysis indicates that over 85% (~16,000 acres, or 5,500 ha) of this mechanical control acreage was set aside for mowing of shoulders, medians, and quadrants. The other 15% is accounted for by brush removal, shrub encroachment and a variety of other mechanical tasks.

The numbers presented above reveal the vast resources required for vegetation maintenance of shoulders, medians and quadrants. In fact, a study by Clary [7] into Department of Transportation (DOT) practices (including California), identified that although guardrail mowing and other manual mowing tasks had higher unit area cost, the majority of the mowing budget was spent on large-area mowing. This suggests that the greatest potential cost saving comes from studying large-area mowing operations, such as mowing of shoulders, medians, and quadrants. Therefore, this report is largely dedicated to the improvement of large-area mowing practices. In particular, the implementation of an autonomous mowing unit for shoulder and median mowing will be analyzed. However, not to disregard the importance of quadrant mowing and other applications, further insight will be provided into the extension of such a unit to other scenarios (see Chapter 6).

The goal for such an autonomous unit is to support Caltrans needs for alternative vegetation management practices in an effort to reduce herbicide use, lower maintenance costs, improve worker and public safety, and improve aesthetics. How then does an autonomous mowing unit assist in reducing herbicide use? According to the IMMS report for the Fiscal 2004/05 year, herbicide treatment of clear strips accounts for 83% of the total vegetation maintenance (family C) herbicide acreage [6]. In Teeter-Balin's study [2], he determined that mechanical mowing was a suitable alternative to herbicide treatment for clear strips. On the other hand, since chemical control is typically more effective than mowing, regular yearly mowing would be required. This would be well suited for an autonomous mowing unit.

Figure 1.3 demonstrates how the clear strip is really an extension from the roadway to the medians and shoulders. Thus, the task of clear strip mowing overlaps the task of median and shoulder mowing. Therefore, the implementation of an autonomous mowing unit for shoulder and median mowing not only supports the reduction in mowing cost, it also leads to the sustained maintenance of the clear strip and a reduction in the annual required chemical volume. An autonomous solution would also reduce worker related injuries due to traffic and equipment exposure, and since medians and shoulders are adjacent to highway traffic, improved maintenance in these areas leads to a reduction in roadside fires.

1.4 Autonomous Environmental Challenges

Each of the different roadside scenarios mentioned in Section 1.3.1 is associated with its own inherent environmental difficulties, such as, obstacles, ditches, holes, slippery gravel, thick vegetation, steep slopes, or rocky terrain. All these uncertainties make standard mowing practices difficult. As a result, Caltrans often employs specialized equipment for certain situations, e.g. guardrail mowers for vegetation in and around guardrails (see [8]), side mounted boom mowers for steep inclines and clear strip mowing, and wide rotary mowers for large area coverage. Similarly, the roadside uncertainties mentioned above make it nearly impossible to develop a general purpose autonomous mowing robot. In fact, most successful commercial solutions have targeted specific applications, while general purpose robots have had little success [3]. Moreover, this report is dedicated to the analysis of an autonomous mowing unit to target shoulders and medians, since this has been identified as the area with the largest potential improvement. This section is focused on distinguishing the characteristics of roadside medians and shoulders in an effort to derive requirements for the implementation of an autonomous mowing unit.

To some extent, mowing of medians and shoulders is regarded as the easiest scenario for Caltrans. In fact, studies have determined that mowing of sections like these requires the lowest cost per unit area [7]. For instance, the terrain is quite often flat and uncluttered from obstacles such as trees and shrubs. Therefore, high capacity rotary mowers may often be utilized. Be that as it may, there are many inherent difficulties which must be overcome in order to develop an effective and safe autonomous solution. Unlike the mowing of turf (e.g. public lawns or even golf courses), roadsides have rocky terrain that is more difficult to traverse. Roadsides are also cluttered with trash, including tires, buckets and other obstacles which interfere with standard mowing units. Even with a vast array of sophisticated sensors it is nearly impossible for an autonomous agent to have a complete depiction of its roadside surroundings.

Observability is the amount of information that an agent/robot gains from its environment that is relevant to its choice of action [4]. If the environment is fully observable, the agent has complete knowledge of the environment's state. Due to sensor noise and incomplete sensor information, fully observable environments are quite rare, the roadside environment is no exception. The extent of practical knowledge gained by an autonomous mowing unit is limited by the selection of sensor technology and the dynamics of the environment. The roadside environment also happens to be stochastic in nature. That is, the future state cannot precisely be predicted given the current state and the agent's actions, because of ambiguity in input and output signals. As an example, wheels will tend to slip yielding drifts in heading and distance, and sensor readings will contain additive noise. The goal of stochastic state estimation is to predict the optimal current state, minimizing the squared error [9]. A large portion of this report is dedicated to optimal state estimation using an Extended Kalman Filter.

Since decisions made in the present affect future decisions, the environment is also considered to be sequential in nature. Conversely, if the actions taken are not dependent on previous actions, the environment is episodic. In a sequential environment, the agent must think ahead, yielding a more difficult situation than an episodic environment. Another characteristic that makes the roadside environment complicated is its dynamic nature. Unlike a static environment, surroundings in a dynamic environment may change between time intervals, e.g. vegetation will alter as it is cut, obstacles may shift overnight and variations in the hardscaping features may be altered by Caltrans workers. However, for most situations, the dynamic nature of the roadside environment is limited to long term intervals, i.e. variations occur over a period of a day or more. Thus, the autonomous mowing of roadsides is not as daunting a task as maneuvering an autonomous vehicle through rush-hour traffic. The more dynamic a task environment is, the harder the solution becomes.

The roadside environment is also classified as continuous, since there are an infinite number of pose states for the agent, namely, the location of the agent can take on an infinite number of positions and orientations. Conversely, a discrete environment contains a finite number of distinct states, as in playing chess [4]. The final environmental classification mentioned here is the difference between a single agent versus a multiagent application. In this report, only a single agent solution is studied, however the more nontrivial case of multiple agents is considered in Chapter 5. The ability to deploy multiple roadside mowing agents at a single time is an important attribute to the cost reduction of autonomous vegetation management, and will be discussed more in Chapter 5.

As might be expected, a partially observable, stochastic, sequential, dynamic, continuous and multiagent environment is the most challenging situation to deal with. The fact that an autonomous roadside mowing agent falls into all of these categories suggests the difficulty in developing a reliable solution. Consequently, autonomous roadside mowing is not practical for all circumstances, nor is it suggested in this report to be a universal solution for Caltrans. Rather, certain roadside median and shoulder sections play well into an autonomous solution. As an example, Figure 1.3 (included in Section 1.3.1) shows a section of HW-113 in Yolo County with 60 ft medians and 90 ft shoulders. Large flat sections such as these account for a large portion of Caltrans annual mowing acreage and represent areas which may benefit by utilizing autonomous mowing technology.

1.5 Project Content

An effective autonomous mowing agent must be able to perceive its environment and to make rational decisions. For mowing applications, it is important for the agent to know where it is, where it was, and where it needs to be in order to efficiently mow. These questions relate to localization, mapping, and path planning, which are all important robotics topics covered in this report. In particular, a low cost sensor array is selected and used for localization using an Extended Kalman Filter. This sensor array takes into account many of the inherent environmental uncertainties mentioned in Section 1.4.

While the topics of localization, mapping and path planning are covered in this report, obstacle avoidance is not. This final subject matter is necessary in developing a fully functional prototype, and although obstacle detection is not covered directly in this report, further insight is provided in Chapter 6. This final subject matter is necessary because obstacles such as shrubs, trash, and ditches may hinder the movement of an autonomous mowing unit. Although roadside medians and shoulders may be free of such obstruction at one point in time, an obstacle may present itself at a future point in time. As an example, a passing truck may lose a piece of furniture which rolls into the center median.

1.6 Project Goals & Contributions

A brief list of project goals and contributions is included in this section. It is noted that although much of the technical research presented in this report is supported by similar studies, this report explores the unique application of an autonomous mowing unit to support roadside vegetation maintenance.

GOALS:

- Determine the practicality of autonomous mowing technology for roadside vegetation maintenance.
- Develop a cost effective solution using moderately priced sensors and data fusion techniques.
- Evaluate the performance in an effort to determine limitations and suggest future development.
- Conceive the potential benefits to Caltrans in the categories of cost reduction, improved safety, herbicide reduction, and improved aesthetics.
- Spur interest in the subject matter to support future development.

CONTRIBUTIONS:

- Full scale testbed used for testing of localization, mapping, and path planning algorithms.
- Real-time software implementation in C# using multithreaded programming concepts.
- A low cost sensor suite for accurate real-time localization in the roadside environment.
- Simulation software to aid in the rapid development of future prototypes.
- Preliminary discussion on how Caltrans can successfully implement autonomous mowing technology to improve their practices.

1.7 Chapter Summary

In summary, since the EIR of 1992 urged Caltrans to significantly cut back on their herbicide usage, Caltrans has been exploring innovative alternative vegetation control methods. AHMCT has been commissioned to research alternatives and past work has indicated that mechanical mowing is the best current mechanical alternative. In an effort to improve mowing strategies, an autonomous vegetation maintenance unit has been proposed. A unit such as this could significantly reduce maintenance costs, cut down on herbicide use, and improve worker and public safety. However, the success of such a unit is limited by the ability to engineer a cost effective unit.

To obtain an understanding of Caltrans vegetation practices a variety of roadside scenarios were studied, including guardrails, clear strips, medians, shoulders, interchange quadrants, and landscaped areas. It was determined that the majority of the annual mechanical mowing budget is dedicated to large-area mowing of medians, shoulders and interchange quadrants. As a result, mowing of shoulders and medians has been selected as an important area for improvement. Since clears strips encompass a portion of shoulders and medians, an autonomous solution would help Caltrans reduce its annual chemical volume. An autonomous solution would also keep Caltrans workers from harm and could lead to reduced fire risks.

To gain an understanding of the environmental challenges associated with roadside mowing, various environmental characteristics were introduced. It was determined that the roadside task environment is partially observable, stochastic, sequential, dynamic, continuous and multiagent, suggesting the challenges of an autonomous solution. To overcome these challenges, an autonomous mowing unit must be able to accurately determine where it has been, where it is, and where it needs to go. These are all important robotics topics which are targeted in this report. In particular, a low cost sensor array is to be selected for accurate real-time localization using an Extended Kalman Filter for data fusion.

CHAPTER 2 Background Research & Literature Review

This chapter is focused on introducing key concepts required for the understanding and development of an autonomous vegetation maintenance solution. Towards the end of this chapter various related research platforms and commercially available products are introduced.

2.1 Robotics Overview

Robots are agents that perform tasks by manipulating the environment around them [4]. They do so by making use of effectors that exert physical forces on their environment. Common effectors include grippers, cutters, joints, wheels and tools. In order for a robot to properly manipulate its environment, it must be able to accurately perceive its surroundings through the use of sensors [4]. As an example, an autonomous mowing unit might use blades as its end effector, and GPS, inertial, and Lidar sensors to perceive its environment.

Robots are quite often grouped into one of three categories: manipulators, mobile robots, and hybrids. Manipulators, or robot arms, are physically anchored to their workspace. They use a series of joints and end effectors to perform high precision tasks such as assisting surgeons, performing outer space tooling tasks, and in manufacturing assembly lines. Mobile robots, on the other hand, move about their environment using wheels, legs, or similar locomotion techniques [4]. Such examples include autonomous ground vehicles (AGV's), unmanned air vehicles (UAV's), and autonomous underwater vehicles (AUV's). The final category, hybrid robots, consist of mobile robots equipped with manipulators. One such example is that of humanoid robots that attempt to mimic the human framework. This report is focused on developing an autonomous mobile robot for roadside vegetation maintenance.

Real robots often deal with environments that are partially observable, stochastic, dynamic, continuous, sequential, and at times multiagent [4]. Chapter 1.4 discussed the challenges associated with each of these environments. In order for an agent to perceive and act rationally in such a complex world, it must make use of a sophisticated array of sensors [4]. Present day robots make use of advanced sensor units, such as inertial navigation units (INU), global positioning systems (GPS), laser ranging units (LIDAR), ultrasound sensors, and camera units. Sensor technology has been a popular area of research over the years and much of this research has been aimed at developing new, more accurate, and lower cost sensor units.

Robotic perception is the key task of mapping sensor measurements to an internal representation of the environment. Perception is a challenging task because sensors are noisy and the environment may be partially observable, stochastic, and dynamic [4]. For the agent to make rational decisions the internal representation must be accurate. This may require one or more of a variety of techniques including error modeling, sensor fusion, and data filtering. The following sections all deal with important topics associated with robotic perception. First, the definition of an agent and what it means for an agent to act rationally is discussed, and then the topic of localization is introduced. Localization, or determining where you are within your environment, is perhaps one of the most important and pervasive perception problems. Finally, additional topics concerning mapping and path planning are discussed.

2.1.1 Agents, Rationality and Autonomy

To this point, the term ‘agent’ has been used somewhat liberally to classify a robot and also to describe the proposed autonomous roadside mowing unit. In reality, the term ‘agent’ is used to describe anything that perceives its environment through sensors and acts upon that environment using actuators [4]. For example, a computer uses keystroke sensors to interpret information from an end user and its screen acts as an actuator to output information. Similarly, a robot might use laser imaging and GPS sensors to perceive its environment and use motors as actuators to move. Throughout this report, the term ‘agent’ will be used interchangeably with the definition of a robot.

Percepts are sensor information collected by an agent from its environment. To make decisions based on sensor percepts, an agent must contain some form of mapping between its percepts and actions. This mapping is referred to as an agent function, or equivalently, an agent program [4]. Classically, the agent program/function is implemented in software on some sort of computing device, e.g. a personal computer or micro-computer. Figure 2.1 demonstrates the interaction of an agent with its environment (derived from [4]). A physical agent is composed of both its agent program and its architecture, where the architecture is composed of an agent’s sensors, actuators and other hardware components.

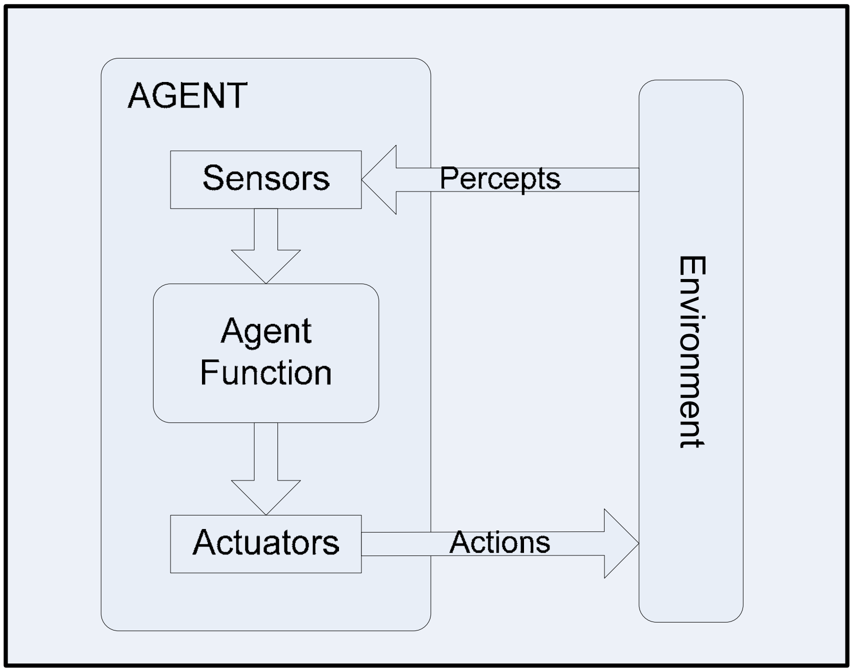


Figure 2.1 Agent Interaction with Its Environment (derived from [4])

The success of an agent is dependent on an agent’s ability to act rationally. Rationality is gauged by a set of performance measures, which define the characteristics of successful behavior. For example, the success of an autonomous mower might be evaluated by the area covered per unit time and by the quality of the cut vegetation. For an agent to act rationally, it must choose actions based on its percepts, history, and prior knowledge, which maximize its performance measures [4].

The grouping of performance measures, the environment, and the agent's sensors and actuators encompass the task environment [4]. A successful design requires a comprehensive specification of the task environment early in the development cycle. It means understanding the environment upon which the agent will interact, developing a distinct set of performance measures based on what one desires from the agent, and selecting an acceptable suite of sensors and actuators, which enable the agent to maximize its performance measures. This process will be carried forth throughout the next few chapters. Some of the more predominant properties of the targeted task environment were introduced in Chapter 1.4. It was determined that the task environment for a roadside autonomous mowing unit is partially observable, stochastic, sequential, dynamic and continuous. The multiagent scenario is explored further in Chapter 5 and 6.

Autonomy is an important attribute that defines the required level of human interaction. An autonomous agent should rely on its own percepts rather than on prior knowledge given by its designer [4]. As an example, an autonomous mowing agent might be ordered to mow a one acre plot of land with no information about the terrain. An autonomous unit should be able to adapt to its environment, dodging obstacles and ditches with no human intervention. Complete autonomy or no human interaction is atypical, since agents must be set up and monitored.

A sensible autonomous roadside mowing should embrace the concepts of autonomy; i.e. it would be desirable to place the unit along the roadside, let it run, and come back later with no required monitoring or human interaction. Clearly, this would never be the case since there are vast uncertainties present in the roadside environment. One cannot tell whether an agent will misjudge a slippery slope and fall into a ditch. Obviously, the goal is to make the unit as autonomous as possible, requiring little to no human interaction.

2.1.2 Localization

As stated above, localization is one of the most fundamental and pervasive perception problems [4]. After all, if an agent has no idea of where it is, how can it rationally interact with its environment? Localization deals primarily with determining ones pose (position and attitude). For mobile robots, this means determining the x, y, z and orientation offset, from some reference location. The task of localization is divided into two distinct categories: relative localization and absolute localization. More detailed descriptions of these topics will be covered below.

The required localization accuracy is dictated by the application. For example, the navigation of naval ships might only require accuracy on the order of 50 meters, while air missile guidance might require sub-meter level accuracy. For autonomous mowing, the required accuracy is dependent on a few factors including efficient area coverage, cost, the particular mowing environment, and the cutting width of the mower (i.e. as the mower cutting width is increased the required accuracy declines because larger overlapping cutting patterns do not affect the efficiency as dramatically). This report is aimed at producing a localization system accuracy that is an acceptable compromise between a low cost design, efficient area coverage, and an acceptable level of vegetation control.

A cost effective design must account for particular sensor technology, since sensors make up a significant portion of the overall platform cost. The cost of sensor technology is largely dominated by the effective accuracy. Thus, although an expensive ($10,000) GPS unit might produce centimeter level accuracy, such technology may not produce a cost effective design, especially if this level of accuracy is unnecessary. The localization implementation (localization algorithm) is just as important as sensor selection. The overall attainable localization accuracy is a result of a complex interaction between individual sensor accuracies, sensor types, system dynamics, path trajectories, and the governing localization algorithms [10]. More information on optimal sensor fusion techniques will be provided in Section 2.1.2 below.

RELATIVE LOCALIZATION: Relative localization, also referred to as dead reckoning, deals with the principal task of determining your pose in comparison to an initial starting point. This is accomplished by integrating one's variation in position over a period of time. Odometry and inertial navigation are the most popular relative localization techniques used for robotic applications [11]. Although relative localization is a popular means for accurate localization over short periods of motion, errors tend to accumulate and significant uncertainties arise over longer periods of time. Such uncertainties can be reduced by incorporating bounded absolute localization information.

Depending on the application, either odometry or inertial navigation techniques may be better suited. Inertial navigation is typically employed when mobile units experience fast speeds and large accelerations, while odometry is better suited for slower moving applications. Slower moving vehicles experience less inertial effects, and it is therefore more difficult to integrate such small acceleration effects to yield accurate positioning data. In other circumstances, a balanced data fusion from both sources results in a much better overall solution.

Odometry makes use of sensors which measure the distance traveled. The most common technique makes use of inexpensive wheel mounted encoder units, which measure the wheel rotation and/or speed. However, other techniques such as ground cameras or laser technology may be used. Odometry provides short term accuracy, but uncertainties accumulate with time. These uncertainties may arise from sources such as wheel slippage, rocky ground, or hilly terrain, and will tend to impact both positioning and orientation information. The effects, however, have a much more significant impact on the orientation. Since orientation is used to derive positioning data, small variations in orientation can have a detrimental effect on the positioning over an extended distance. This is especially true for agents that make use of differential steering. More on this topic will be disused in Chapter 3. As a result, orientation information is often complimented by inertial rate gyroscopes, which are better suited for measuring variations in orientation.

Inertial sensors measure variations in accelerations as a result of linear and rotational effects. By integrating this information, variations in the speed, position, and orientation can be calculated. Like odometry, drifting effects degrade the long term performance of inertial units. On the other hand, advances in MEMS technology has led to the improvement in performance, along with the size and cost reduction of inertial units, resulting in more practical and commonly used units. Popular inertial measurement units, such as the MTi from Xsens, can provide 3D orientation, positioning, and velocity [12].

ABSOLUTE LOCALIZATION: Absolute measurements are measurements which provide pose (position and orientation) information independent of previous location estimates. Since they provide pose information from a single measurement, uncertainties are bounded with time, unlike relative measurements which grow with time [11]. Absolute or global localization is the task of interpreting one's position based on such absolute measurements. Although bounded, absolute measurements are susceptible to a variety of error sources depending on the application and implementation. As an example, global positioning systems (GPS) make use of precisely timed radio frequency data from satellites in orbit to generate positioning data. However, such positioning data is susceptible to a variety of error sources including atmospheric, geographic, multi-path and radio interference.

Absolute measurements make use of landmarks placed within an agent's environment. These landmarks may be active, that is they transmit signals to the agent, or passive in which case the agent must identify them. Active landmarks, also known as beacons, transmit signals picked up by a receiver and used to derive important positioning data. Two common positioning methods include triangulation and trilateration. Triangulation makes use of ranges and angles to three or more beacons in order to derive positioning information, while trilateration makes use of only range data. Of course, to derive such positioning, an agent must have some prior knowledge of the location of each of the beacons [11].

GPS is one example of a positioning system that makes use of active beacons. To derive positioning data, precisely timed signals sent from orbital satellites are used to determine the distance to each of the satellites in range. Then, using trilateration techniques, one's global position in latitude, longitude, and altitude may be derived. Obviously, the location of each satellite must be known precisely and at least four satellites must be in range. As noted above, many factors affect the performance of GPS, however, additional correction data provided by a differential station may be used to compensate for such uncertainties. In optimal situations, sub-meter and even sub-centimeter level accuracy may be attainable. More about GPS units will be discussed in Chapter 4.

As another example, ongoing research has discovered that radio frequency identification (RFID) technology may be used to derive accurate positioning data for robotics applications. RFID systems make use of active or passive tags (with unique identifications) which are tracked by a transmitter/receiver. Stores and hospitals sometimes use RFID technology for asset tracking and to prevent shoplifting. Although many RFID implementations are used only for asset tracking, other systems provide real-time positioning feedback [13, 14]. A variety of research studies have looked into RFID technology as a means for absolute localization, see [15-17]. Other forms of active beacons used for robotics applications include radar, sonar, laser, and even WiFi technology.

Passive landmarks do not transmit information to an agent and must be located and uniquely identified in order to provide positioning data. If these landmarks are intentionally placed in the agent's environment, they are known as artificial landmarks [11]. To obtain absolute positioning feedback, the agent must be given previous information of the exact global location of each of the landmarks. The agent must also be able to uniquely and accurately identify each landmark by means of its sensor percepts. Engineered landmarks may be uniquely identified by their shape, size, color, or some other identification code. Passive RFID tags are one example of passive landmarks that might be placed in an agent's environment, and since they possess unique codes they can correctly be identified. The intrinsic accuracies of such positioning implementations are limited by sensor accuracies, landmark positioning uncertainties, distance to landmarks, geographic setup, and observability. In some cases artificial landmarks are impractical because of the overhead costs associated with surveying, and the cost of the landmarks themselves.

Passive landmarks, which are not intentionally engineered and placed in their environment, are known as natural landmarks [11]. The list of potential natural landmarks is endless and may include features such as rocks, trees, poles, doors, windows, stars, or even roadway lane markers. The capacity to use such features for absolute localization is governed by the ability to uniquely and consistently identify them. Then, if the locations of such features are known by the agent, the agent need only localize itself with respect to the landmarks. If, however, the location of such landmarks is unknown, the task becomes somewhat challenging. The agent must not only be able to uniquely identify distinct landmarks, it must also be able to position them and store a map of them so that such information may be used for self-localization. This approach is known as Simultaneous Localization and Map building (SLAM), see related research articles [15, 18, 19].

The duty of accurate environmental localization is an important but difficult perception problem. While relative localization techniques are inexpensive and accurate over short distances, their uncertainties grow with time and proper compensation is required. One solution is to include absolute measurements, which are typically uncertain over short periods of time but are bounded with time. The task of correctly and efficiently combining relative and absolute data is an example of sensor fusion, which will be discussed next.

2.1.3 Sensor Fusion

The task of combining a variety of sensor sources into a single internal model is the process of sensor fusion. Sensor fusion can be advantageous for a number of reasons. For one, different sensors may provide different pieces of information about an agent's environment, and with suitable sensor fusion, the agent may gain a more comprehensive internal model of the environment. Another reason is that better estimates may be obtained from redundant information. As an example, unbounded position estimates (e.g. Cartesian x and y location) from odometry might be complemented by bounded GPS position estimates. Although both sensors provide the same positioning information, proper sensor fusion yields position predictions that benefit from the short-term accuracy of odometry and the long-term position bounding of GPS. This redundancy in sensor information is also important for situations in which one or the other information sources may be temporarily absent. As a final point, sensor fusion can reduce the negative impacts caused by measurement noise.

Sensor fusion has been a popular area of study throughout the years, and many varying methods have evolved. Popular sensor fusion techniques for localization include Kalman filtering, Markov localization, Monte Carlo localization, and combinations of these algorithms [4, 10]. These techniques all belong to a broader class known as Bayesian estimation. Bayesian estimation algorithms, like the ones mentioned here, are used to predict a state estimate (e.g. position and orientation) given the input, process model, and the sensor output measurements. These sensor measurements are often too noisy and inaccurate to be used directly, and may or may not provide information of direct interest [10]. This process of predicting the state estimate is known as state estimation, or stochastic state estimation, in which inputs and outputs are not known exactly. Bayesian algorithms are all aimed at stochastic state estimation. The overall goal of stochastic state estimation is the determination of an optimal state estimate, which yields the lowest expected squared error [7].

Bayesian algorithms work by using prior knowledge to build an initial belief state. When new measurements become available, their likelihood function is combined with the prior belief state (using Bayes' theorem) to derive an estimated quantity and a posterior probability distribution (a new belief state). In this manner, information that is not directly measurable, but is observable, may be obtained. Such algorithms utilize closed loop predictor-corrector schemes. In this fashion, the process model is used to predict an open loop state estimate. This state estimate is then corrected by merging output measurement data [10].

Kalman filtering is a popular Bayesian localization technique used in many research studies [10, 11, 15-18, 29, 31-35, 38]. In comparison to Markov and Monte Carlo methods, Kalman filtering is less computationally expensive and very precise, making it well suited for real-time mobile robot localization [10]. Therefore, the Kalman Filter has been selected as the preferred sensor fusion technique throughout this report. It is noted however, that the Kalman filter may suffer from bad initial guesses and may even diverge from the correct value if its initial uncertainties are too large. Similarly, because the filter works on noisy and possibly biased sensor measurements, the overall signal-processing algorithm (Kalman filtering and other Bayesian techniques alike) cannot substitute for the overall quality of raw sensor measurements [10]. Rather, these algorithms attempt to formulate the best possible estimates given the information they are supplied. Accordingly, the overall attainable state estimation accuracy is dependent on a complex interaction between individual sensor accuracies, sensor types, system dynamics, path trajectories, and the governing state estimation algorithms [10].

KALMAN FILTERING: More details on the Kalman filter will be introduced here; for more information on alternative Bayesian methods see [4, 10, 15]. Kalman filtering is based on linear dynamic systems that are discretized in time. The state of the system is stored as a vector of real numbers. At each time step, control inputs are propagated through the process model to determine the new state estimate with some noise mixed in. Then, with the aid of visible output measurements containing noise, the state estimates are updated in a predictor-corrector style, as mentioned above [20]. The key advantage is that the measurements are not required to correspond to the internal state directly.

Rather, the measurements are a linear function of the internal state with some additive noise.

In order to make use of the Kalman filter, the system process model must replicate the framework of the Kalman filtering equations. The Kalman filter assumes the system process to be modeled as

$$x_k = A_k x_{k-1} + B_k u_k + w_k \tag{2.1}$$

where, x_k is the new and true state, A_k is the state transition matrix applied to the previous state x_{k-1}, B_k is the control-input model which is applied to the control input u_k, and w_k is assumed to be additive, zero-mean, normal, white noise. Subsequently, the measurement model is:

$$z_k = H_k x_k + v_k \tag{2.2}$$

where, z_k is a measurement made at time *k*, H_k is the observation model which is applied to the true state x_k, and v_k is the observation noise which is assumed to be zero-mean, normal, white noise. Figure 2.2 is a schematic illustration of the Kalman filter model derived from [20]. This model demonstrates how the internal state is hidden from view, and the input (u_k) and measurement (z_k) are the only visible quantities.

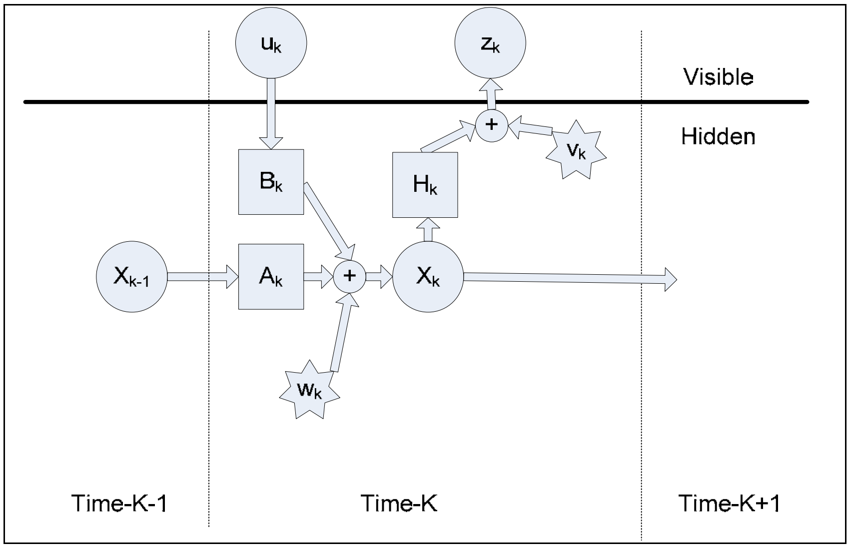


Figure 2.2 Kalman Filter Model (derived from [20])

The Kalman filter uses a recursive state estimation procedure, that requires only two variables, $\hat{x}_{k/k}$ the state estimate at time *k*, and $\hat{P}_{k/k}$ the error covariance matrix at time *k* (a measure of the estimated accuracy of the state estimate) [20]. The procedure is performed as follows. First, the state and error covariance are updated given the previous state, input, and knowledge of the process noise, using the following equations:

$$\hat{x}_{k|k-1} = A_k \hat{x}_{k-1|k-1} + B_k u_k \tag{2.3}$$

$$P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_k \qquad (2.4)$$

where, Q_k is the covariance of the process noise w_k. Then, if no measurement at time *k* is observed, the predicted state and error covariance from equation 2.3 and 2.4 become the actual state and error covariance prediction, $\hat{x}_{k|k}, P_{k|k}$. If, however, a measurement at time *k* is taken, the state estimate and error covariance matrices are updated using the following equations:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \qquad (2.3)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H_k \hat{x}_{k|k-1}) \qquad (2.4)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \qquad (2.5)$$

where, R_k is the covariance of the measurement noise v_k and I is the identity matrix. These equations yield an optimal, lowest mean-square error estimate. The derivation of equations 2.3, 2.4 and 2.5 is not demonstrated here however, and may be found in [20] and [21].

Under certain ideal conditions, the Kalman filter is theoretically proven to be optimal in the sense that it minimizes the estimated error covariance and provides the most likely estimated state [21]. These ideal conditions require the system process matrices, *A, B,* and *H* to be known exactly and also for the process and sensors noises to be independent, white, and to have normal zero mean distributions. In reality, however, rarely do the conditions for optimality actually exist, yet the filter has been shown to work well in spite of various deficiencies [20] and [22]. More information on the principle requirements for which the Kalman filter was designed will be covered in Section 4.3.

The Kalman filter equations introduced above assumed that the system model was linear with respect to time. Most non-trivial systems, however, are nonlinear in nature. The nonlinearity may be associated with the process model, the measurement model or both [20]. For these cases, a more sophisticated version of the Kalman filter, called the Extended Kalman filter (EKF), is required. Use of the EKF requires only that the transition and observation models are differentiable functions. In the case that they are, the EKF makes use of the function derivatives, also known as the Jacobians, to approximate a linear function about the point of interest. For systems that are highly nonlinear, this approximation may not suffice and unbounded divergence may be noted [10]. These situations may require higher order Taylor-series terms, such as the Hessian (2nd order term), to linearize the equations about the point of interest. On the other hand, many situations perform relatively well with only a single order EKF. Other variations of the Kalman filter are included in [21].

Only a brief description of the EKF has been presented here. A more complete introduction to the governing equations is presented in Chapter 4, along with additional insight into the requirements for optimality.

2.1.4 Mapping and Path Planning

Mapping and path planning are other important robotics topics that relate to localization. Mapping is the task of constructing an internal model of one's environment based on sensor percepts. Such a model may contain a variety of information, such as the locations of various environmental features/landmarks, a mapping of the terrain, or information as to the history of an agent's motion. Mapping requires an agent to be able to localize itself with respect to its environment and also to localize other features within its environment.

Mapping techniques are used for a variety of different situations. One popular scenario makes use of internal mapping to assist in localization. In this case, environmental features are used as reference landmarks. If an agent is provided with an internal mapping of these features, the agent need only locate, distinguish, and position itself relative to these features. Another popular area of study attempts to localize features with no knowledge as to their location and to make use of these natural features to assist in individual localization. This is a challenging problem known as simultaneous localization and mapping (SLAM), as was previously discussed.

Aside from assisted localization, mapping techniques may be utilized for other purposes. For example, the terrain might be mapped in an effort to determine the safest path to travel. As another example, a map may be formulated and stored in memory for feedback to an end user, e.g. one may desire the desk layout of an office building floor, or the layout of trees, shrubs, ditches, signs, and guardrails along the roadside. This report makes use of simple mapping techniques for efficient mowing.

Path planning is the task of determining the motion from one configuration to the next. A few examples include, determining the joint movement required to place an end effector at a given location and orientation, obtaining an efficient path for a mobile robot to reach its goal state, or finding the shortest network path for a computer data packet. Popular approaches are broadly categorized as either cell decomposition or skeletonization. Both cell decomposition and skeletonization techniques attempt to reduce the continuous path-planning problem into smaller discrete paths which are modeled as nodes of a graph. In this manner, the problem is transformed into a discrete graph-searching problem. For more information on cell decomposition and skeletonization see [4].

To develop effective path planning solutions, one must have a clear understanding of the desired performance measures. As discussed earlier, performance measures are used to define the characteristics of successful/rational behavior for an agent. The list of desired performance measures is endless, and is largely dependent on the application. For mobile robotics, performance measures often relate to how efficient and effective an agent moves within its environment. For example, some performance measures aim to reduce the agent's overall traveled path length, while others might reward the agent for selecting safer or less strenuous paths. Regardless of what they happen to be, an effective path planning algorithm should take into account prior knowledge (such as a map) and its percept history to derive paths which maximize the agent's performance measures.

The potential field method is one example of a path-planning algorithm that attempts to minimize the path distance to an agent's goal state (it is part of the cell decomposition family). Not to overlook obstacle interference, this method also incorporates a potential field, which grows in magnitude with the distance to the closest obstacle. This potential cost function is an additional term in the overall path optimization. The goal is to derive an optimized path that provides maximum clearance from obstacles while minimizing the overall path length [4]. The potential field method is a popular technique in mobile robotics; however it is only one of many path planning techniques.

Most of today's robots make use of deterministic path planning algorithms that overlook intrinsic localization uncertainties [4]. If localization uncertainties are small, these algorithms may work well, but they may tend to perform poorly if uncertainties become too large. After all, how can a robot plan its path if it has no idea of where it is? Incorporating localization uncertainties into path planning techniques is a challenging task and is saved for later discussion in Chapter 6.

2.2 Platforms and Their Technology

Over the years, a large amount of research and development has been devoted to the nontrivial task of autonomous mowing. Most solutions have been studied in the academic research atmosphere, however some more basic solutions have reached the commercial market. Popular application areas include golf course turf management, lawn maintenance, and agricultural cultivation. This chapter is focused on introducing key related research studies, and the current state of commercially available autonomous mowers.

2.2.1 Research Platforms

The task of autonomous mowing has been a popular area of study in the academic research atmosphere due to its vast array of research topics. These topics include, but are not limited to, machine guidance, navigation, localization, SLAM, image processing, machine learning, path planning, mapping, sensor fusion, computer architecture, mechanical design and circuit design. This subsection is focused on introducing a variety of research studies related to the task of autonomous mowing. As a result of the overall large volume of related literature, only some of the more relevant literature studies are included here, with insight into the location of additional sources.

At the University of Florida, Hakala and Doty developed a low cost autonomous mower known as the LawnNibbler [23]. Their design (Figure 2.3) made use of a radio wire buried along the perimeter of the workspace, which acted as an artificial boundary detected by sensors. They also utilized active beacons for localization and obstacle avoidance. In particular, sonar pulses were sent into the agent's environment. When received by pre-surveyed beacons, these beacons would reciprocate infrared light back to the unit. Based on sonar time-of-flight information, the distance to the beacon could be calculated. Using the distance to three separate beacons the agent was able to use trilateration to calculate its current x and y position. Additionally, infrared sensors mounted to the agent, along with sonar pulses, assisted with obstacle avoidance. Although the unit was capable of mapping where it had cut, it worked on the principal of randomness. Specifically, given enough time, the mower would eventually cut everything in a confined area. The LawnNibbler had a speed of 1 ft/s (0.3 m/s) with a 6 in (15 cm) cutting width.

Figure 2.3 LawnNibbler (left), and LawnShark (right) developed at Univ. of Florida

Continued autonomous mowing development at the University of Florida led to the introduction of the LawnShark [24, 25], see Figure 2.3. The LawnShark borrowed much of the same sensor technology used on the LawnNibbler but was aimed at rapid mechanical development of a mowing platform. To achieve this, the design utilized a Toro rechargeable electric push mower with custom mounted electric motors. This design technique yielded rapid development and resulted in a platform that was actually designed for lawn mowing. Nonetheless, the design, which made use of tank-like (skid) steering, turned out to have a major mobility limitation. Since the sensor technology and algorithms were essentially the same as the LawnNibbler, they are not reiterated here.

Most low cost autonomous area coverage platforms (e.g. vacuum cleaners and mowers) rely on random path patterns, with the hope that eventually, over a long enough period of time, a confined area will be entirely covered. Agents that rely on random area coverage require large amounts of time and are far from efficient. This is due to continual overlapping of previously covered areas. Some studies are aimed at overcoming these efficacy limitations by employing more sophisticated sensors and/or algorithms. A variety of related research will be introduced to follow.

Some techniques that have attempted to counteract efficiency limitations (from random area coverage) have made use of cameras, range finders, or passive sensors to analyze the coverage area [26 - 28]. Analysis of this sort may then be used to draw out conclusions as to whether certain areas have been previously covered or not. This information may then in turn be used to develop more efficient paths. These techniques are made easier when there are significant differences between a covered and uncovered area, e.g. variations in vegetation height, color differences, or texture differences. Of course, factors such as these depend on the application, e.g. mowing, vacuuming or harvesting.

One such study, conducted at the University of Florida [26], made use of an onboard camera unit and texture analysis techniques to differentiate between cut and uncut grass. The goal was to produce a more efficient area coverage algorithm to support past research conducted with the LawnNibbler and LawnShark. Another solution, proposed by Rafaels and developed by Technical Solutions of Frederic [27] Maryland, made use of infrared sensors to examine vegetation height. In this manner, a distinction between cut and uncut grass could be inferred and used for a more efficient coverage algorithm.

Other studies have made use of image processing units (cameras) to assist in automated harvesting of alfalfa hay [28]. Using weighted RGB (red, green, blue) techniques, they have developed an algorithm to detect the boundary between cut and uncut crop. Since the algorithm is adaptive, it compensates for variations in lighting and removes noise due to shadows. Figure 2.4 displays an image of the automated harvester and also a sample output from the crop line tracker. Based on image processing, they are also able to identify obstacles and the end of each crop row. The information acquired by the cameras is used to assist in efficient automated or even autonomous harvesting. Future studies aim to include redundant GPS information to improve the robustness of the solution and also to produce a commercially viable product.

Figure 2.4 Automated Harvester (left), Sample output from crop line tracker (right) (From [28])

Most other techniques which have been aimed at efficient area coverage have focused on accurate localization and mapping. The idea is to use highly accurate localization, along with mapping techniques, to monitor areas that have been previously covered, and to target areas that have not yet been covered. Many sensor combinations and fusion techniques have been proposed. Usually the sensor selection is tailored to a specific application and takes into account cost restrictions. A few pertinent studies will be presented here.

Kiriy [10, 29] proposed a localization sensor array geared towards golf course turf management. He argued, that although differential GPS (DGPS) provided a convenient localization scheme, it was unsatisfactory due to possible GPS signal loss from nearby foliage, buildings, or other interfering sources. Instead, he developed an infrastructure of pre-surveyed artificial landmarks to aid in absolute localization. He considered magnetic markers but later rejected this idea (due to range limitations) for vision-based landmarks. Onboard camera units were used to detect these markers and measure the relative angles using image-processing techniques. An EKF solution was used to fuse visual data with fiber-optic gyroscope (FOG) and odometry measurements. The sensors were mounted to conventional golf course mowing units and manually driven in predefined patterns. The data was collected, post processed, and compared to the ground truth path data, which was obtained with a very accurate 2 cm (0.8 in) DGPS NovAtel unit. Using the proposed sensor array, Kiriy was able to attain an acceptable level of accuracy.

In a similar study, Kurth [15] used RFID technology for localization of an autonomous mower. He placed transmitting RFID tags at known locations within a predefined region.

Then, using a receiver mounted on the mower, time-of-flight and identification information could be collected from the tags in order to calculate positioning information based on trilateration. The mower was preprogrammed to drive in a repeating path among the tags, while GPS, inertial, dead reckoning, and RFID range information was collected. The GPS data was taken with a highly accurate (2 cm) unit and fused with dead reckoning data to provide a precise ground truth comparison path.

To evaluate the performance using only RFID range and dead reckoning data, three alternative probabilistic approaches were used, i.e. extended kalman filtering, sliding batch and particle filtering. All three methods yielded acceptable levels of accuracy (on average 20 cm or better), and the EKF solution was identified as a suitable method for real-time localization, because of its computational efficiency. The EKF performance was, however, significantly affected by the initial uncertainty in position. The proposed solution was also used to test the problem of SLAM, i.e. uncertainty in the tags initial locations.

Aono et al [30] proposed an accurate positioning method using GPS and dead reckoning information. Their study was focused on developing a realistic (cost effective) sensor array for positioning on undulating ground. In particular, sensor information from a GPS unit, a fiber optic gyroscope, roll and pitch sensors, and wheel encoders were fused using Kalman filtering techniques. The GPS unit used was an expensive kinematic-GPS unit which could obtain accuracy on the order of 2 to 3 cm (~0.8 in). To evaluate the effect of GPS accuracy on the overall positioning accuracy determined by the sensor fusion algorithm, varying levels of additive white noise was added to the GPS readings during operation. Testing and evaluation was performed by utilizing a modified riding lawn mower. The authors determined that even with a GPS accuracy of 1.0 m (3.3 ft), the data fusion resulted in an overall Euclidean positioning accuracy of about 0.2 m (0.7 ft). He noted that a less expensive GPS solution providing accuracy on this order of magnitude would yield a realistic sensor suite. Figure 2.5 below presents a schematic of Aono's sensor configuration and the overall control block diagram.

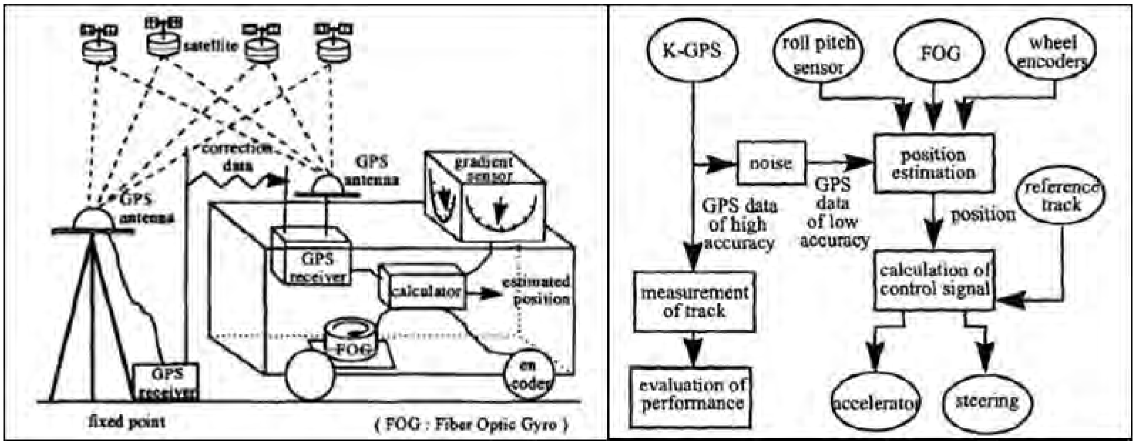


Figure 2.5 Sensor Implementation (Left), System Control Block Diagram (Right) (From [32])

The research presented above demonstrates the vast assortment of techniques used for agent localization and area coverage. What is surprisingly consistent however, is the continual use and success of Kalman filtering for sensor fusion, even for a variety of differing sensor configurations. Kalman filtering is a versatile and robust solution which has found wide use in the field of robotics, and because it is computationally efficient, it is well suited for real-time applications. Many more research studies have tackled the problem of efficient area coverage, autonomous mowing, and low cost sensor fusion. For more insight see [31-37]. Next, an overview of related commercial products will be presented.

2.2.2 Commercial products

Two popular robot lawnmowers that have entered the commercial market are Robomower [38] by Friendly Robotics and Automower [39] by Husqvarna. Similar to the LawnNibbler introduced in Section 2.2.1, both designs rely on random area coverage and an installed perimeter guide wire. These guide wires, which act as artificial boundaries, emit electromagnetic fields that can be detected by Hall effect sensors which are on board. More details on each of the two platforms will be introduced here.

The Robomower comes in a variety of different models, each with different specifications. The standard model is the RL850 which is geared toward yards with a total size of 10,800 ft^2 (~.25 acres, 1,000 m^2) or less. Mower operation works as follows. The mower begins by mowing around the perimeter via the guide wire. Then, the unit begins a random mowing pattern, altering direction when either the guide wire is detected or one of its four pneumatic bump sensors detect an obstacle. While the mowing pattern is somewhat random, the mower makes use of an onboard floating compass to mow in a crisscross V-shaped pattern. This technique is a patented technology aimed at more efficient area coverage. Regardless, the mowing pattern is still random in nature and requires a large amount of time before complete coverage is achieved.

All models of the Robomower make use of a 24-volt battery pack that yields a work time of about 2.5-3.0 hours and requires a charge time of about 20 hours. The average area covered on a single charge is estimated to be 5,400 ft^2 (~0.125 acres, 500 m^2). While the RL850 model must be setup and collected for recharging, the most advanced model the RL1000 makes use of a recharging station for complete automation. That is, the RL1000 model will return to its docking station when its charge level is near completion. Another feature of this model is that it can be preprogrammed with a weekly mowing schedule. The RL1000 model mows at the same pace but is geared towards larger yard sizes, on to order of 21,500 ft^2 (~0.5 acres, 2000 m^2).

The Robomower makes use of three separate blades that operate at high speeds in order to mulch the grass as it is cut and, therefore, the clippings need not be collected. The basic principal is that the mower can cut frequently and yield very fine clippings that decompose quickly acting as a natural fertilizer. For more information on additional features see [38]. Figure 2.6 shows a picture of the Robomower RL850 model along with a schematic of the perimeter wire setup. The Robomower RL850 model sells on the market for about $1400 and the RL1000 model for about $1800.

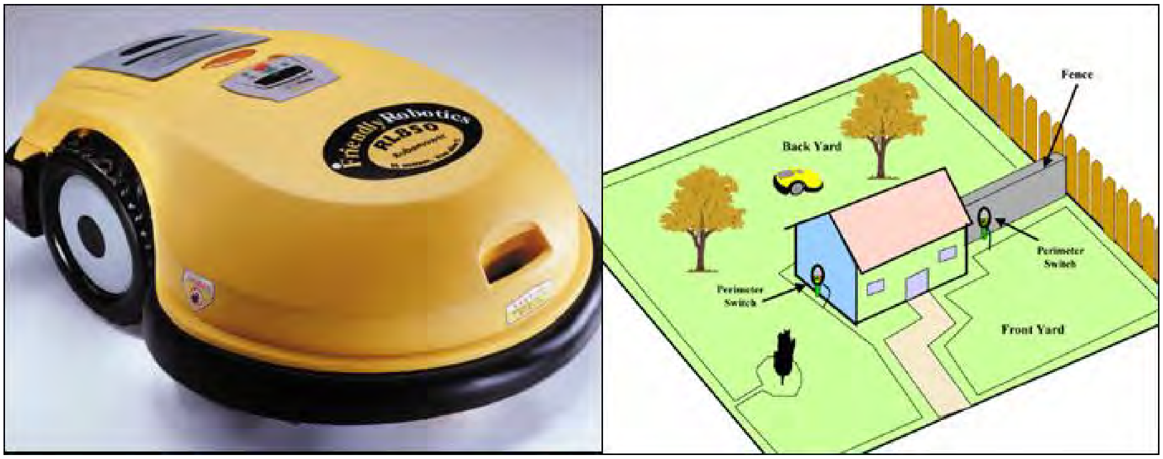


Figure 2.6 Robomower (left), typical setup (right)

The Automower built by Husqvarna operates in much the same way as the Robomower. It uses random cutting patterns aided by a predefined perimeter wire. Similarly, it has a docking station for automatic charging and storage. Unlike the Robomower, however, the Automower is much smaller. It has a cutting width of only 8.7 in (22 cm) and a weight of 19 lbs (85 N), while the Robomower has a cutting width of 21.0 in (53 cm) and a weight of 50 lbs (222 N). Also the cutting pattern is not aided by a compass, thus the cutting pattern is completely random. The Automower is shown in Figure 2.7, along with a simulated cutting pattern. The Automower sells for around \$2150. For more details on the Automower see [39].

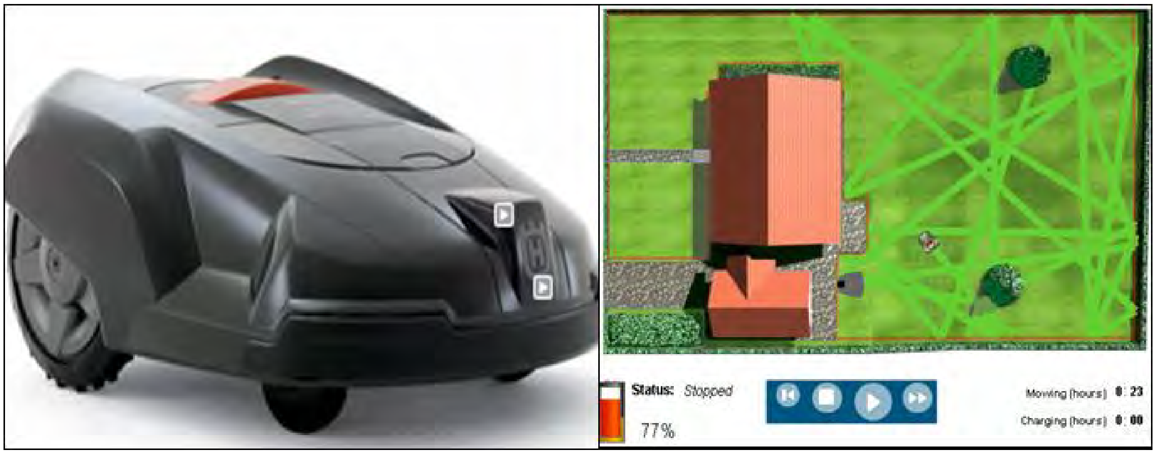


Figure 2.7 Automower (left), Simulated Random Cutting Pattern (right)

A variety of other robotic lawnmowers have entered the commercial market such as the Lawnbott [40] and iMow [41]. They all make use of the same principles, i.e. preinstalled ground wires which act as a defined boarder and some variation of random mowing pattern. Although these units may be practical for smaller household lawns on the order of half an acre (2000 m^2) or less, they are not designed for larger plots of land, because of their slow random area coverage. As an example, the Robomower optimistically covers 5,400 ft^2 (500 m^2) in 2.5 hours. Thus, it would take the Robomower 20 hours of mowing to cover a full acre (4000 m^2) of land and seven separate charge times. Since each charge time takes 20 hours, it would take the Robomower nearly 7 days of continual use to cover a single acre (4000 m^2). Of course, this might be acceptable if someone was very adverse to the manual time consuming task of mowing.

2.3 Chapter Summary

In summary, robots are agents that perform tasks by manipulating their environment. They make use of sensors to obtain information from their surroundings and effectors to exert physical forces on their environment. Performance measures are used to define the characteristics of successful behavior. For an agent to act rationally, it must choose actions (based on sensor information) which attempt to maximize its performance measures. The task environment is a grouping of the agent's environment, performance measures and sensors and actuators. It was noted that the design of a successful agent requires a comprehensive understanding of its task environment early in the design stage.

Next, the topic of localization was covered. Localization is the important task of determining where you are within your environment. Two distinct techniques were discussed: relative and absolute localization. It was noted that relative localization techniques provide accurate pose (position and orientation) information over short periods of time, but suffer from long-term drift. Conversely, absolute localization tends to provide poor short-term accuracy, but offers long term drift free bounding. Robust localization techniques typically attempt to merge (through sensor fusion) both relative and absolute measurement. In this manner, the short-term accuracy of relative localization and the long term bounding of absolute localization can be achieved.

The topic of sensor fusion was covered by introducing a few Bayesian estimation techniques. Of these, the Kalman filter was identified as a popular algorithm that has found wide use in the field of robotics. In particular, the Kalman filter has been a well accepted technique for real-time robotic localization. Explicit details on the framework of the Kalman filter were covered, along with a brief introduction to the Extended Kalman filter (EKF), which is used for nonlinear systems. The Kalman filter is theoretically proven to be optimal under certain presumed conditions, however, even if these presumed conditions are not met, the filter has been found to perform reasonably well. Next, a brief overview of mapping and path planning and how they relate to localization and rationality was covered.

Finally, a literature survey was conducted. A variety of academic research platforms related to the problem of autonomous area coverage was introduced. Some earlier designs relied on the principal of random area coverage, while others attempted to develop more efficient algorithms using unique sensor arrays. A few robotic lawn mowers which have entered the commercial market were discussed. These commercial products also rely on random area coverage techniques. For small regions, random area coverage methods may be acceptable, however random area coverage is impractical for larger regions, e.g. golf course turf management and roadside vegetation control. Many of the research studies covered in this chapter had an influential impact on the design of the autonomous agent developed in this report and will be referenced as appropriate.

CHAPTER 3 Testbed Mechanical Design

To support the core concepts studied in this report and also to serve as a proof-of-concept, a fully functional testbed was designed and built. This chapter begins with the customer requirements as derived from Caltrans needs. The mechanical design of the testbed is then discussed and, when appropriate, additional information on the hardware is included. Finally, the kinematic equations of motion are introduced, with insight into differential drive tracking control.

3.1 Customer Requirements

The design of a practical autonomous ground vehicle is a challenging endeavor that requires a clear understanding of the task environment and the customer's requirements. These customer requirements define exactly what the client wants from a product. Caltrans, the client, has expressed interest in new innovative technologies that are aimed at relieving their dependence on herbicide use, reducing labor costs, improving safety and improving aesthetics. To fulfill these needs, an autonomous roadside mower has been proposed for further development. To assist in this development, a tentative set of customer requirements for such an agent has been developed. These requirements are listed here.

CUSTOMER REQUIREMENTS:

- Above all the agent should be safe to operate.
- The unit should yield a cost effective solution.
- The agent must never stray from its bounds and enter onto the pavement.
- The unit should be easy to setup, operate, and collect.
- Mowing rates should be comparable to existing practices.
- Patches of vegetation should not be left behind.
- The system should require limited monitoring.
- The unit must be relatively small in size for easy transportation.
- The agent must be reliable and easy to repair.
- The agent must be able to cover clear strip sections of vegetation.
- The agent should avoid all ditches, steep slopes, trees, shrubs and other large obstacles.

- If the agent requires structural features, these should be easy to set up and low in cost.
- The cutting system should be effective for all varieties of roadside vegetation.

These design requirements are intended to aid in the mechanical design, sensor selection, and software implementation of the testbed developed for this report, but do not necessarily constitute a fully realized set of requirements. Rather, this report is aimed at evaluating the practicality of such an autonomous mowing agent, and also at suggesting appropriate technology, implementations, and concepts, to aid in the future development of a successful prototype to meet these customer requirements. Specific goals and contributions of this report were included in Section 1.6.

3.2 Testbed Mechanical Design

To support the core concepts studied in this report (namely localization, mapping and path planning), a fully functional testbed/platform was designed and built. This testbed was developed as a research tool, not as an absolute prototype. Regardless, its framework was selected to replicate that which would be practical of such an autonomous roadside mowing unit; that is, with respect to size, configuration, and layout. Also, the testbed was designed so that it could be easily outfitted with an array of sensor and hardware technology. In this manner, it could be used to evaluate this technology and to suggest appropriate technology, implementations, and concepts to aid in the future development of a successful prototype

Original thoughts were to modify a pre-existing platform in such a way that it could be used for this research study. The Robomower from Friendly Robotics (Section 2.2) was considered as a possible candidate. However, this autonomous mowing unit was designed for homeowner lawns, yielding it unsuitable for the rough terrain and harsh vegetation present along the roadside environment. Chandler, et al [24] were able to modify a Toro rechargeable electric push mower to serve as an autonomous mowing platform. In this manner, they were able to overlook the problems encountered in trying to build a mowing platform. In the end, the platform had serious mobility limitations, as a result of its poor skid-like steering. After all, the mower had been designed as a manual push mower and not as a self propelled autonomous unit. Most other studies aimed at developing autonomous mowing technology have made use of outfitted riding lawn mowers with more success (see [10] and [30]). Yet, these platforms tend to be large, heavy, costly, unsafe, and at times have limited maneuverability.

For many of the reasons listed above, a fully customized testbed/platform was proposed. In this way, the inherent mechanical, cost, safety, and size limitations of modifying a pre-designed platform could be avoided. Instead, the testbed could be developed with the customer requirements (listed in Section 3.1) in mind. Namely, it could be designed to succeed in the harsh roadside environment. However, not to waste vast amounts of time developing this testbed, modular design techniques were used. For example, motors, wheels, tubing, fasteners, and other parts were all selected to be interfaced with limited welding and other time consuming machining tasks. In this manner, more time could be spent on defining an appropriate and cost-effective sensor and hardware array, and on evaluating the overall performance of the agent.

3.2.1 Locomotion, Motor, and Battery Selection

An autonomous mobile robot needs some mode of locomotion, such as wheels, tracks, legs, propellers, etc. Although tank-like tracks were considered as a possible means for locomotion, it was later determined that standard wheels would be more practical. Locomotion also requires some mode of steering, hence many steering configurations were considered, including skid, synchronous, Ackerman, and differential [42]. Differential drive steering, one of the most popular modes of locomotion found in mobile robotics, was selected as the preferred method for several reasons. For one, differential drive systems are easy to implement, and yield highly maneuverable platforms. Also, such configurations have been used with much success for existing commercial autonomous mowers (Section 2.2), and for a variety of riding lawnmowers, such as the ones used in [10]. Finally, many successful tracking control algorithms have been developed for differential drive systems. More insight into tracking control will be covered in Section 3.4.

Differential drives are made up of two independent drives, one for each side of the vehicle. These drives typically consist of an actuator/motor, which may or may not be linked to a gearbox, and some type of wheel. The wheels most often lie on one common axis. Proper control of each independent motor yields all steering and traversing motions (see Section 3.3 for more details). Commonly used actuators include various types of gasoline, electric, hydraulic, and pneumatic devices. For the purpose of this report, the power source was chosen to be electric, because of safety, size, and cost limitations. Therefore, the design selection was limited to various types of electric motors. In particular, DC permanent magnet motors were selected as the ideal choice, as a result of their comparatively small, light, and efficient characteristics for a given power rating [43].

Appropriate motor selection is an important consideration, which should take into account various factors, such as: speed and torque requirements, power consumption, current and voltage levels, and efficiency. To assist in a suitable selection, a preliminary set of vehicle specifications was developed, taking into mind the customers requirements from Section 3.1. This list was intended only as an aid upon which suitable motor selection could be based, and does not necessarily represent the final platform specifications:

VEHICLE SPECIFICATIONS:

- The vehicle must be able to reach a maximum speed of 3.0 mph (4.8 km/h), however, on average will operate at 1.5 mph (2.4 km/h)
- The vehicle must be able to accelerate to 1.5 mph (2.4 km/h) in less than 1.0 seconds.

- The motors must be able to provide enough torque to overcome friction, gravity, and acceleration loads in the worst case scenario.
- During regular movement, the motors should operate at near optimum efficiency.
- The overall vehicle weight should be less than 250.0 lb (1100 N); this value representing the overall worst case scenario.
- The vehicle should be able to drive up a 30-degree (.52 radians) incline; once again, this being the worst case scenario.
- Tooling/cutting loads are estimated to be a maximum of 25 lb (110 N)
- The vehicle is assumed to have 14.0 inch (35.6 cm) diameter wheels.

To evaluate the required motor performance characteristics, a tentative dynamic vehicle model was developed. This simple four-wheeled vehicle model is displayed in Figure 3.1. To evaluate this model, D'Alembert's law [44] was used to sum the forces along the x-axis (Figure 3.1) yielding

$$2R_a + 2R_b - C_L - Ma - Mg\sin(\theta) = 0 \qquad (3.1)$$

where, C_L is the mower cutting load, which is assumed to be parallel with the motion of the vehicle, and R_a and R_b are the frictional contact forces of the front and rear wheels and the ground, respectively. It is assumed that the two front and two rear wheels experience even loading effects and thus a factor of 2 appears in equation 3.1 for both R_a and R_b. Since a differential drive was selected as the means for locomotion, it is assumed that the front two wheels are passive casters, in which case $R_a \approx 0$. To determine the required peak (worst case) motor torque, it was necessary to evaluate equation 3.1 above, assuming the highest loading condition, specifically, the steepest slope, and the largest acceleration, platform weight, and cutting resistance. Based on the vehicle specifications listed above, this condition occurs when, $a = 2.2\ ft/s^2$, $\theta = 30$ deg, $Mg = 250\,lb$, and $C_L = 25\,lb$. Using these values R_b was evaluated to be 83.54 lb (371.61 N).

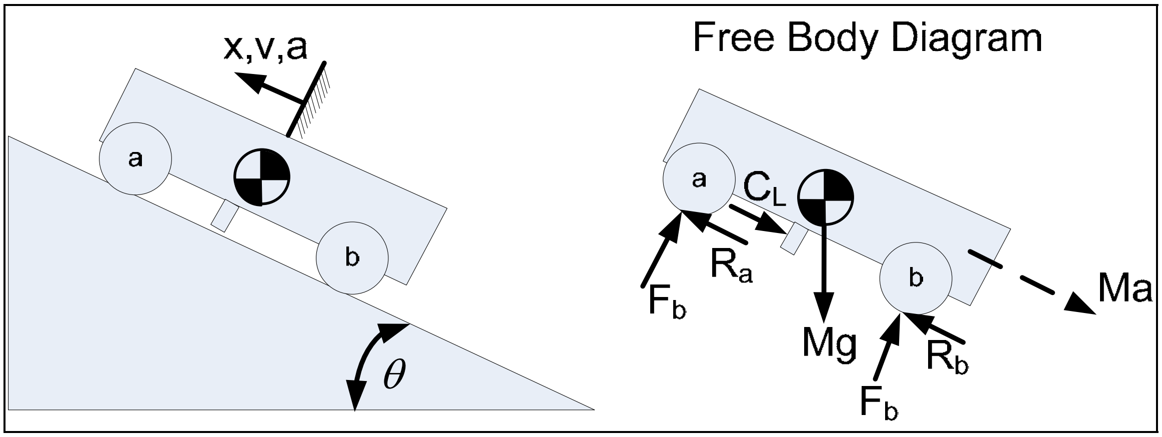


Figure 3.1 Dynamic Model of the testbed

The required torque at each of the rear wheels (wheel-b) is related to R_b by the following equation

$$\tau = R_b * r \qquad (3.2)$$

where, *r* is the effective wheel radius. Evaluating equation 3.2 with the value for R_b calculated above, and an estimated wheel radius of 7 in (18 cm), it was determined that the required peak torque is about 49 ft-lb or 585 in-lb (66 N-m). This value represents a conservative (loads are overestimated), but realistic requirement on the overall peak output torque for each of the two independent motors. On average, however, the torque level is expected to be much lower for a few reasons. For one, a 30 degree incline is atypical, since roadside medians and shoulders are relatively flat overall. Also, inertial loads are only experienced during rapid speed changes, and the platform will have a steady speed during most operation. Therefore, the average case parameters were re-evaluated to be approximately: $a = 0.0\, ft/s^2$, $\theta = 5\deg$, $Mg = 250\, lb$, and $C_L = 15\, lb$. Making use of eqn. 3.1 and 3.2, the average operating torque was evaluated to be approximately 11 ft-lb or 129 in-lb (15 N-m) per motor.

The model introduced above was intended only as a rough, but conservative basis upon which to select appropriate motors. Nevertheless, to meet the derived motor performance characteristics, two identical T64 motors from NPC Robotics were selected for the testbed [45]. The NPC-T64 motor, as shown in Figure 3.2, is a DC permanent magnet motor with a custom mounted direct drive gearbox. This motor has a nominal voltage of 24 volts, yet can operate at up to 36 volts for short periods of time. The peak output power at the nominal voltage is about 1 hp (746 W). The gear box has a gear reduction of 20:1 for high torque output and has mounting holes for easy direct mounting to a tire, as shown in Figure 3.2. This easy direct coupling of motor, gearbox, and tire assisted in the overall rapid modular development of the testbed, since a custom designed drive system was not required.

Figure 3.2 NPC-T64 DC motor (left) T64 motor mounted to flat proof tires (right)

Along with the selection of two T64 motors, two PT5306 tires from NPC Robotics were chosen [45] to mount to the motors. The PT5306 tire, as shown in Figure 3.2, is a 14 in (36 cm) outer diameter flat proof tire. This tire was chosen for a few specific reasons. First, it is flat proof, meaning it is resilient to punctures from nails and other sharp objects that may be present along the roadside. Second, it is not subject to the variations in wheel radius representative of pneumatic tires (note: the wheel radius plays an important role in the overall localization scheme, more will be discussed on this topic in Chapter 4). Finally, it has a relatively large outer diameter, which yields it better suited to travel on the rough roadside terrain. Smaller tires, like the ones used on commercial autonomous mowers (Section 2.2.2) are unsuitably small for the roadside environment, and would most likely get stuck in small holes and ditches. Detailed CAD drawings for the T64 motor and PT5306 flat proof tire can be found at [45].

The rest of this section is devoted to introducing the particular details of the T64 motor and PT5306 combination, and why they were selected to meet the motor performance characteristics derived above. Neglecting small friction and magnetic losses, the motor torque for a permanent magnet DC motor is proportional to the supply current,

$$\tau = K_t i \tag{3.3}$$

where, K_t is torque constant, and the torque speed relationship at steady state is,

$$V = R * \tau / K_t + K_b \omega \tag{3.4}$$

where, K_b is the voltage constant, R is the winding resistance, ω is the motor speed, and V is the applied voltage [43]. Based on eqn. 3.3 and 3.4 and published dynamometer results for the NPC-T64 motors, at the nominal voltage of 24 volts, the following parameter values were calculated $R = .218\Omega$, $K_t = 7.50\, in - lb / amp$ (0.85 N-m/amp) and $K_b = .089 V / RPM$ (0.85 $V / rad / s$). Making use of these values, along with eqn. 3.4, a theoretical plot of the torque vs. speed characteristics could be plotted for a supply voltage of 24 *V*, as displayed in Figure 3.3. This figure also includes the actual results from the dynamometer test for the NPC-T64 motors (also at 24 *V*), as supplied from the manufacturer (lines are included between discrete dynamometer points only to suggest a trend line). It is apparent that the theoretical approximation lines up with the dynamometer test data relatively well; only slight discrepancies occur at higher speeds. This results because the linear approximation presented by eqn. 3.4 neglects friction and magnetic losses that occur at high speeds [46].

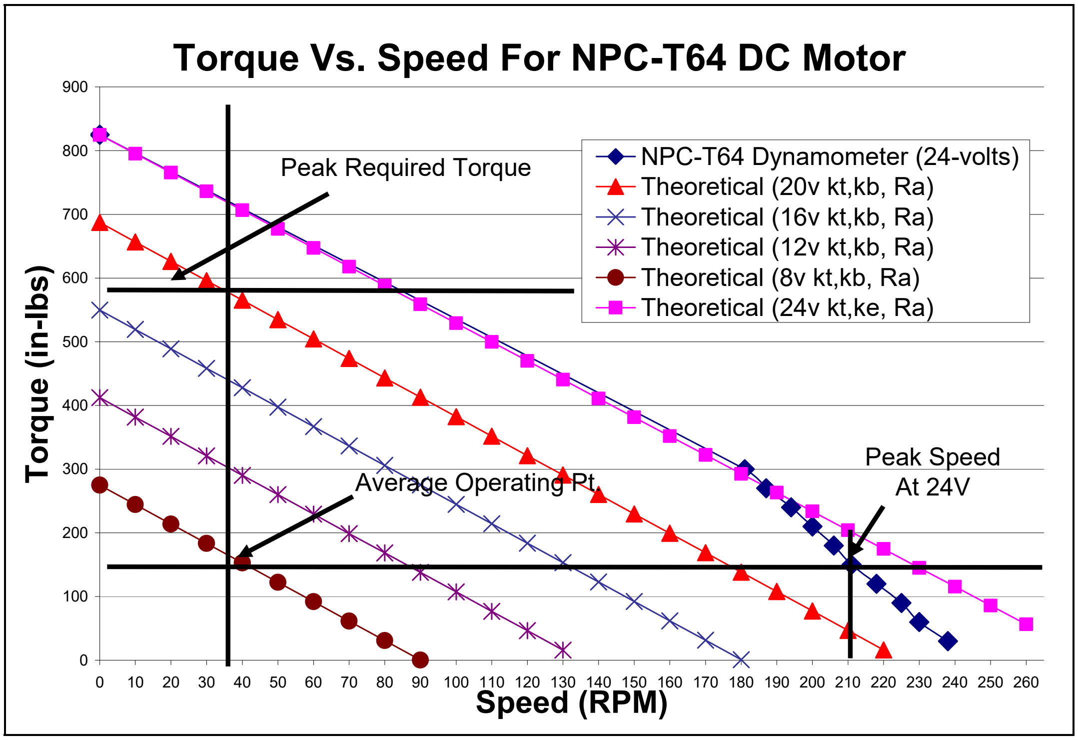


Figure 3.3 Torque vs. speed characteristics for the NPC-T64 DC motor

A number of points are in order regarding Figure 3.3. It is evident from the dynamometer data, that the stall torque of 825 in-lb (93 N-m) for the NPC-T64 motor and gearbox combination is much larger than the required worst case peak torque of 585 in-lb (66 N-m) determined earlier. Thus, provided that a supply voltage of 24 volts is available, the select motors should have plenty of torque to drive the testbed under the worst possible loads. Figure 3.3 also includes theoretical linear torque vs. speed trend lines (as derived from eqn. 3.4) for differing supply voltages. The voltage level required to drive the output shaft at a particular speed and torque is indicated by the intersection of the torque vs. speed trend line and the system operating point. For the proposed testbed, an average torque requirement of 129 in-lb (15 N-m) was calculated. In addition, the vehicle specifications derived earlier, stipulate that the testbed will travel at an average speed of 1.5 mph (2.4 km/h). With a wheel diameter of 14.0 in (35.6 cm), this corresponds to an output motor speed of 36.0 rpm (3.77 rad/s). The average system operating point (the intersection of 36 rpm and 129 in-lb) is displayed in Figure 3.3, and the torque-speed trend line intersects this point when the applied motor voltage is about 8 volts. It turns out that the optimal efficiency for a DC permanent magnet motor occurs at only slightly less than 50% of the stall torque [46], and since the average operating point is nearly half of the stall torque (when the applied voltage is 8), this suggest a good efficiency for the average operating loading. That is to say, a high output power in relationship to the allotted electrical input power is achieved. As a final note, at a supply voltage of 24 volts, the peak vehicle speed (Figure 3.3) is close to 210 rpm (22 rad/s). This is equivalent to nearly 8.8 mph (14.2 km/h), a value that is in excess of the required maximum speed of 3.0 mph (4.8 km/h) defined in the vehicle specifications.

To supply power to the DC motors selected, various battery supplies were considered. The projected supply was intended to provide power to the entire testbed system, not just the motors, i.e. all sensors, hardware and actuators. Accordingly, an adequate supply taking into account various power levels for all electrical components was necessary. It turns out, however, that the power consumption of the motors considerably outclassed the majority of the other proposed electrical components. For example, the motors require an estimated peak power of 2640 W (i.e. 24 V at 110 amps), while the selected GPS unit (Section 4.2) only requires 1.7 W (i.e. 0.52 amps at 3.3 volts). To achieve these lower power levels, simple electrical voltage regulators were anticipated, but in most cases the existing hardware and sensor interface had built in regulators that could handle the proposed 24 volt power supply. The mowing actuation was the only other proposed component to compare with the level of power required for the motors. Research into existing electrical mowing equipment indicated that a voltage level of 24 volts was actually quite common, and current draw was on the same order of magnitude as the T64 motors.

After a thorough evaluation, two Diehard rechargeable automotive batteries, as shown in Figure 3.4, were selected for the testbed power supply [47]. Since these batteries are only 12 volt DC supplies, it was necessary to link them in series to get a 24 volt DC supply. These batteries were picked because of their high discharge rate and long battery life. Their particular dimensions were selected to coincide with the mechanical layout of the testbed, as discussed in the next section.

Figure 3.4 Diehard Gold South Automotive Battery

3.2.2 Frame Design

Caltran's requirements (Section 3.1) call for a moderately sized mowing unit for easy transportation. Equipment transportation is currently a significant drawback for Caltrans. Most of Caltrans existing mowing equipment consists of large tractors with mounted mowing units, which are unsafe and costly to transport [7]. A mowing unit with a baseline of 24 to 40 in (0.6 m – 1.0 m) was conceived as an acceptable size. Undoubtedly, the future goal is to be able to transport multiple units (on the order of 4 to 6) on a single trailer bed, and to have a single employee disperse these units along incremental roadside distances for mowing. A quick look into tow behind trailers indicates that even with an outer profile of 40 in (1.02 m), 6 units could easily be placed on a standard 12 ft (3.66 m) flatbed trailer.

Once again, to assist in the overall rapid development of the testbed, modular extruded aluminum tubing was utilized for the frame. In this manner, the frame could be built quickly, without any welding, and it could be reconfigured easily. This inexpensive tubing, which is developed by Bosch, is high-strength, anodized aluminum designed for rapid and rugged structural framing. The exact tubing profile (1.77x1.77 in, 45x45 mm) used is displayed in Figure 3.5, along with two 90 degree gussets, and T-bolts that were used extensively. Detailed CAD drawings and engineering loading specifications can be found for these components at [48].

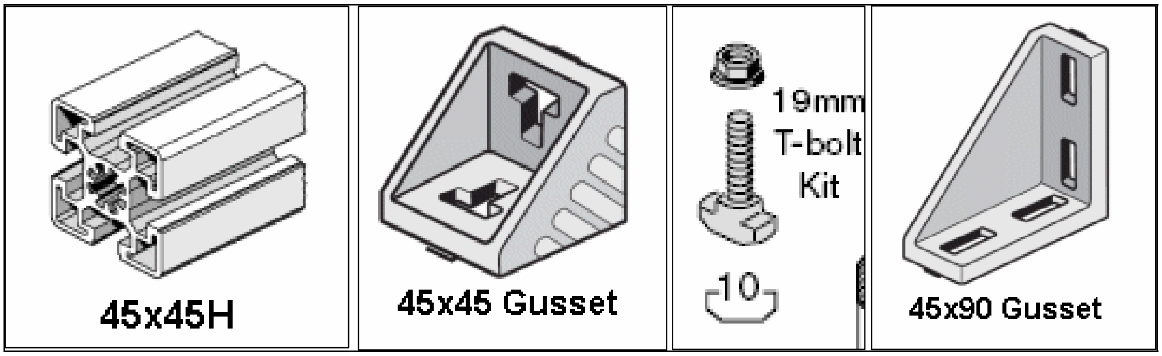


Figure 3.5 Bosch modular components

Making use of these Bosch components, a triangular outer frame was conceived in 3D CAD software (Pro-Engineer), as displayed in Figure 3.6. The dimensions were selected to coincide with the desired baseline. With this configuration, it was anticipated that the two drive motors would be mounted in the back and a third stabilizing passive caster wheel would be used at the front of the vehicle (see Figure 3.7), leaving space between the wheel configuration for the mechanical mower. This common tripod layout is widely used for differential drive vehicles [10, 38-41].

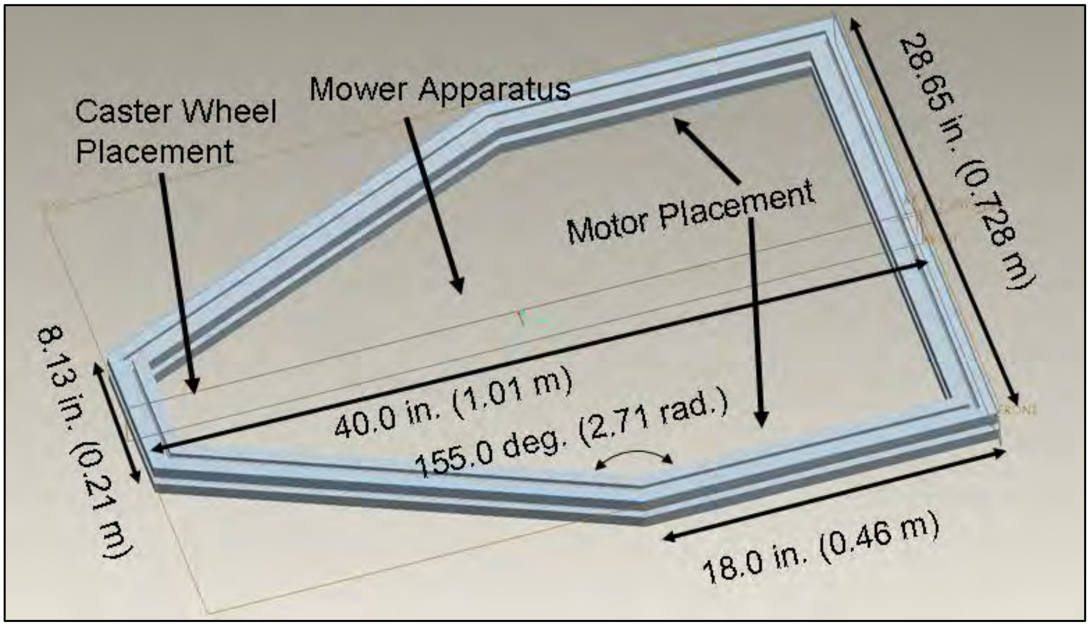


Figure 3.6 Schematic layout of the testbed frame

The tentative layout displayed in Figure 3.6 calls for the proper placement of the power source (two Diehard batteries, Figure 3.4). Ideally, these batteries should be located close to the electrical sink (the two T64 DC motors) to reduce the overall length of the interconnecting electrical wires. These high capacity automotive batteries have a net weight of 56.0 lb (249 N), more than twice the net weight of the two T64 motors. Such a large load has a significant impact on the dynamics of the vehicle and required well thought out placement. Ultimately, they were located directly aft of the motors, towards the back of the testbed for the following reasons. For one, all other significant loads (the caster wheel and cutting system) were to be located in front of the wheel baseline. Thus, the batteries were intended to counter balance the loading about the rear baseline. In this manner, the center of gravity was shifted close to the baseline, reducing the overall rotational inertia during turning maneuvers, and ultimately reducing the wheel slippage from these rotational dynamic effects. It was, however, theoretically verified that the tentative center of gravity was located sufficiently forward of the baseline so that the vehicle would never tip backward during uphill movement with maximum acceleration. Finally, the location of the batteries was selected as to increase the loading over the baseline, in an effort to increase the traction of the wheels and limit wheel slippage.

To mount these batteries, a harness that extruded from the frame was modeled in Pro-Engineer (Figure 3.7). The dimensions of this extruded harness were selected to match the physical dimensions of the batteries. Various dimensions for the selected Diehard batteries (Section 3.2.1) were available. To maintain a compact outer vehicle length and also a short distance between the motors and the batteries, the dimensions were selected to be narrow, but tall and long. The selected dimensions were 5.0 in (12.7 cm) wide, 8.1 in (20.5 cm) high, and 9.1 in (23.0 cm) long. Figure 3.7 also includes the tentative layout of the front caster wheel. This two tire caster was later replaced with a single larger caster that resulted in better maneuverability, see Appendix A.

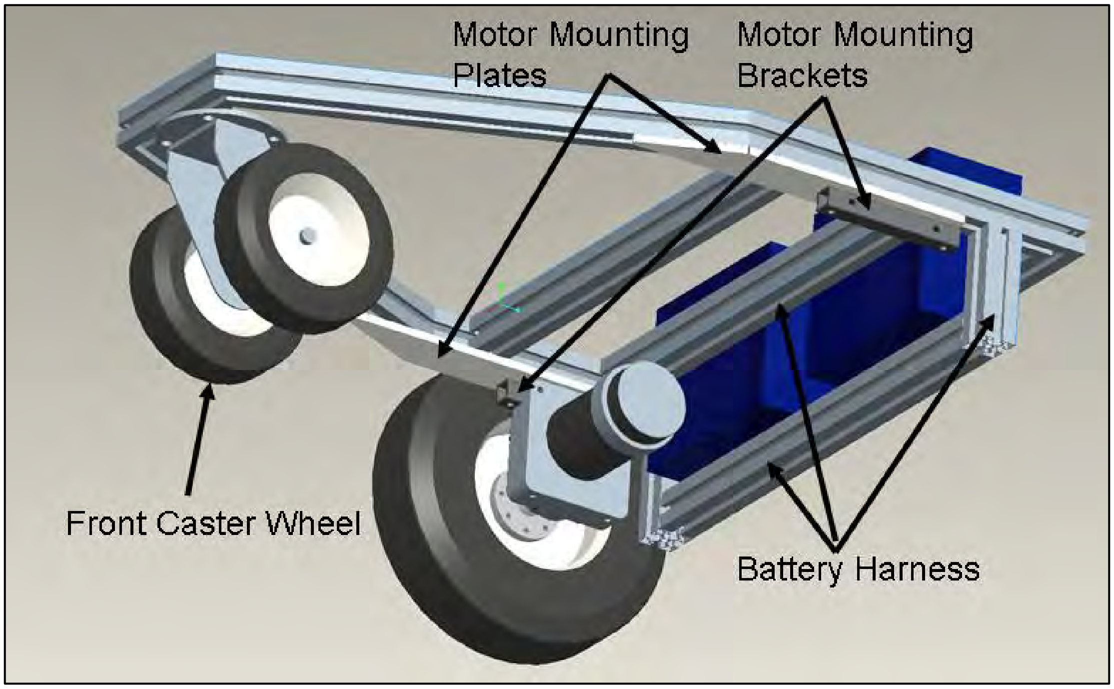


Figure 3.7 Overall 3D testbed solid model

In order to fasten the two T64 motors to the frame, the handy pre-manufactured gearbox mounting holes were utilized. Making use of Pro-Engineer, various configurations were tested. In the end, it was decided to mount these motors under the frame to yield a large undercarriage clearance, and also to leave enough space for the mechanical cutting system. Two square mounting brackets were designed to fasten the motors to the frame. These brackets were modeled in Pro-Engineer as 1 x 1 in (2.54 x 2.54 cm) hollow tubing (Figure 3.7), but were later replaced with more rugged solid 1 x 1 in. extruded Aluminum stock. Actual testing of these mounting brackets indicated that they were sensitive to variations in the aluminum frame and thus an additional mounting plate was conceived to yield rugged and exact perpendicular mounting of the motors. This trapezoidal plate (Figure 3.7) had two distinct purposes, first, it assisted in the perpendicular mounting of the motors, and second, it added structural support to the frame joints. It is noted, that the depth of the battery mounting harness (discussed above) was selected to coincide with the mounting height of the two T64 motors. In this manner, a uniform protective shrouding could be incorporated.

Based on the 3D solid models introduced (Figures 3.6, 3.7), an actual mechanical assembly was put together. This assembly is displayed in Figure 3.8. Additional beams were added to the proposed configuration to provide additional structural support and extra mounting points for hardware and other components. For additional photos and further assembly details see Appendix A.

Figure 3.8 Testbed frame with mounted motors and wheels

In an effort to protect the motors and batteries from damage, custom mounted shrouding was shaped to fit around the rear baseline of the vehicle. Two differing sections were formed. One section (Figure 3.9) was used to protect the undercarriage (motors, encoders, batteries, electrical wires) from rocks, dirt, vegetation and other damaging materials. The second section was used to enclose the sides for the same reasons. This shrouding was formed out of pliable 1/16th in (1.59 mm) aluminum for rugged protection from collisions with rocks and other protrusions, and also to maintain a rustproof exterior. Additional mechanical design details of the end product will be included throughout this report when appropriate.

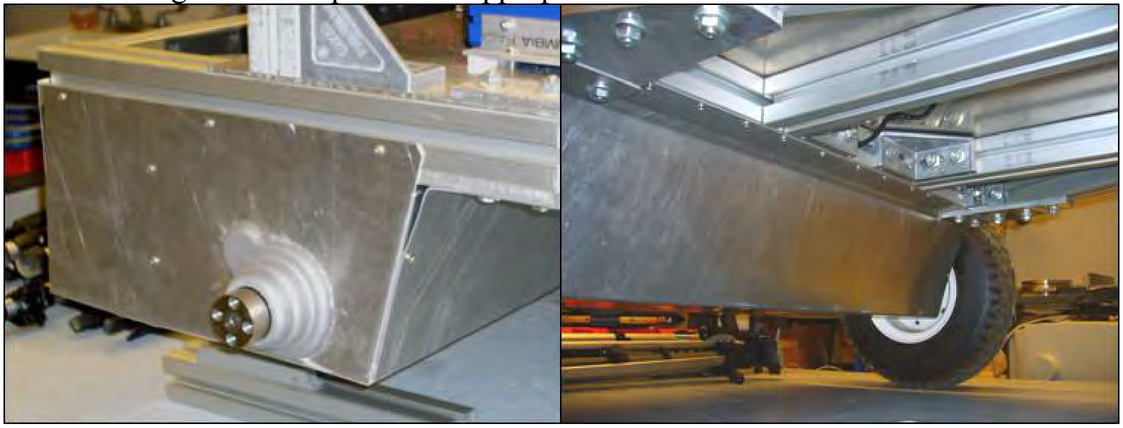


Figure 3.9 Protective shrouding, side shrouding (left) undercarriage shrouding (right)

3.2.3 Motor Control and Power Setup

Motor control is an important overall element in developing a completely autonomous vehicle. For autonomous mowing, this means having precise control over the speed at which each of the drive motors operate. Control of DC rotary motors, such as the ones used for the testbed, is achieved by varying the independent level of power to each of the motors. Under certain situations, i.e. precise control is not necessary or loads are constant, open loop control is adequate. More commonly, however, loads will fluctuate and precise actuation requires feedback control. The roadside environment is no exception; variations in the terrain and vegetation will significantly alter the vehicle dynamics. To achieve such required feedback control an AX2850 from RoboteQ was selected [49].

The RoboteQ AX2850 unit is a dual channel forward/reverse digital robot controller. What makes this controller so well suited for the platform developed in this report is its flexibility, that is it has a vast array of features, each of which can be manually configured for the required application. Some of the more relevant features will be included here; for a more comprehensive list see [49].

AX2850 FEATURES:

- Dual channel configurable PID feedback speed control via optical quadrature encoders.
- RS-232 command interface for computer control and monitoring.
- Alternative Radio-Control command interface.
- Two 32-bit up-down encoder counters that may be used for odometry.
- Configurable safety emergency stop command.
- High current output (up to 120 amps per channel) that is sufficient for the T64 motors.
- Low idle power consumption (100 mA at 24V).
- Battery power regeneration for efficient use.
- Programmable current limits to protect against short circuiting.
- High frequency efficient Pulse Width Modulation (PWM) control.
- Watchdog for automatic motor shutdown in case of communication loss.
- Overheat control via an onboard temperature sensor.

- Programmable acceleration
- An efficient heat sinking design that typically does not require a fan.
- Small lightweight profile.

This controller was convenient because it was relatively easy to interface and because it handled many of the lower level required tasks (such as PID speed control and encoder counting), decreasing the total testbed development time. To mount the AX2850 controller, a 1/8th in (6.24 mm) thick aluminum mounting plate was used to cover the testbed frame (Figure 3.10). This served a few purposes: it secluded the electronics so they could later be enclosed, it acted as a heat sink for the controller, and it provided a convenient surface to which additional sensors and hardware could be mounted. The exact mounting location of the controller on the plate was subject to a few constraints, as specified by the RoboteQ manufacture. Namely, it was necessary to keep both supply power and motor wires as short as possible to reduce the inductance and decrease the electrical noise. As a result, the controller was mounted slightly in front of the batteries just above the motors, Figure 3.10.

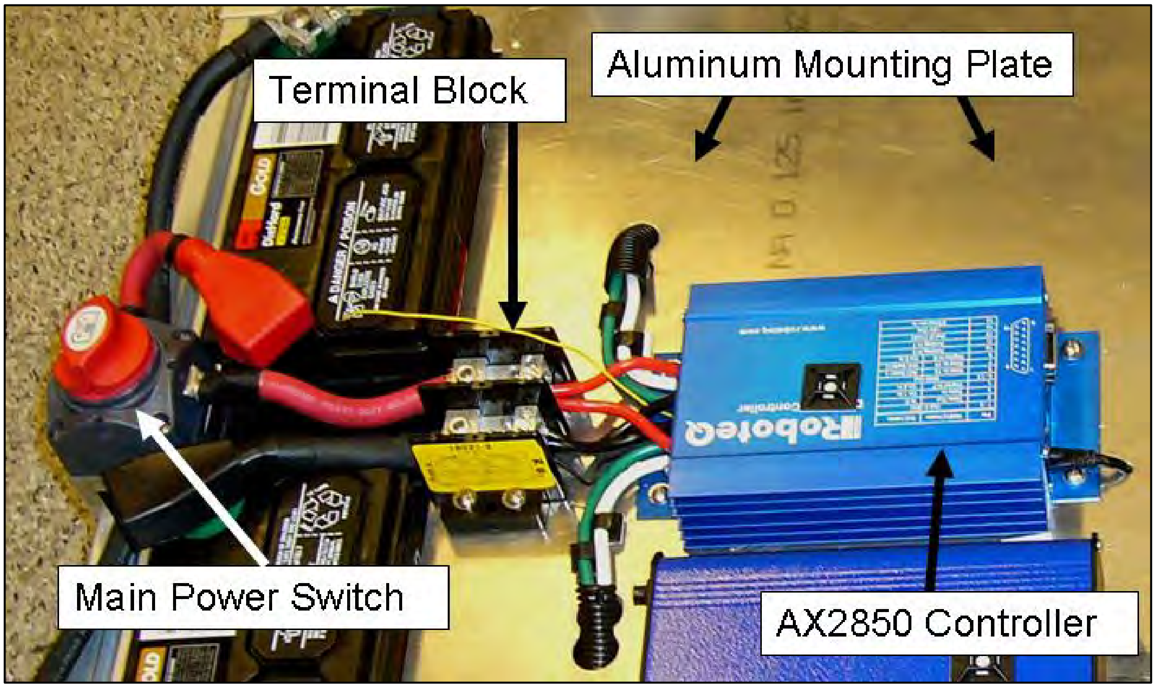


Figure 3.10 Motor controller layout and interface

To distribute power to the controller, a high current (power supply) terminal block was used, Figure 3.10. This terminal block was also used to distribute power to other hardware as was necessary. Also, as suggested by the manufacturer, a high current main battery supply switch was used to cut power from the distribution terminal in the case of emergency. A second toggle switch (not shown in Figure 3.10 but displayed in Appendix A) was used as the central power switch to the controller. More details on the layout and circuit design are included in Appendix A.

The RoboteQ controller makes use of a configurable digital PID control algorithm for precise motor speed control. Experiential parameter tuning was conducted and is included in Appendix A. As a last note, the computer communication interface and encoder setup will be discussed in Chapter 4, then, in Chapter 6, some additional features of the AX2850 controller will be discussed and how they support future development which will meet Caltran's requirements.

3.3 Kinematic Equations of Motion

Although there is no universal convention to modeling the equations of motion for a vehicle, various guidelines have evolved over the years [10, 50]. Research studies suggest that a kinematic model is appropriate for low speed, low acceleration, and lightly loaded applications, while a dynamic model is more appropriate for heavily loaded and highly dynamic situations. Given the low speed, low acceleration, and relatively small loads of the proposed autonomous mowing unit, a kinematic model has been selected as the preferred method. A kinematic model makes use of the relative motion of a vehicle without regards to the masses and forces involved. As such, sensors that measure the relative displacement and/or speeds are necessary. A more detailed discussion on the particular sensor selection is covered in Section 4.2.

To assist in the understanding of the kinematic equations of motion, a schematic layout of the testbed is included in Figure 3.11. The kinematic equations of motion for this differential drive configuration [10] are given by

$$V_A = \frac{V_L + V_R}{2} \tag{3.5}$$

$$\dot{x} = V_A * \cos(\theta) \tag{3.6}$$

$$\dot{y} = V_A * \sin(\theta) \tag{3.7}$$

$$\dot{\theta} = \frac{V_L + V_R}{d} \tag{3.8}$$

where, V_L and V_R are the left and right wheel velocities, A is the tracking point, θ is the heading angle, and d is the wheel base separation. By integrating these equations with respect to some initial known starting condition (x, y, θ) it is possible to computer where the agent/testbed will be at any point in time, based only on the control inputs V_L and V_R [42]. This is known as tracking, dead reckoning, or relative localization, as was introduced in Section 2.1.1.

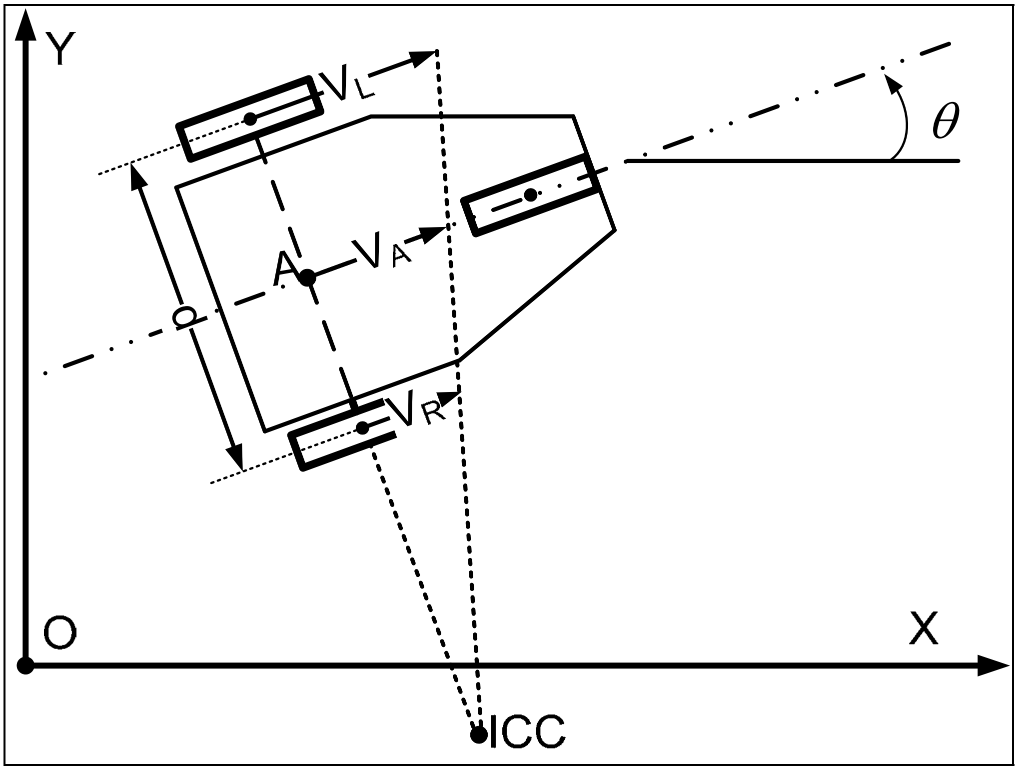


Figure 3.11 Kinematic diagram of the testbed in a horizontal plane

Integrating eqn. 3.6 – 3.8 is not straightforward however, due to a constraint on the velocity. This constraint, which is present for differential drives, is known as a nonholonomic constraint. Under certain conditions integration is possible, however, a more popular solution is to use a discretized approximation. One discretized approximation that has been proposed and widely used in the field of robotics [10, 51] is modeled as

$$\Delta D_t = \frac{\Delta DR_t + \Delta DL_t}{2} \tag{3.9}$$

$$\theta_{t+1} = \theta_t + \frac{\Delta DR_t + \Delta DL_t}{d} \tag{3.10}$$

$$x_{t+1} = x_t + \Delta D_t \cdot \cos\left(\frac{\theta_t + \theta_{t+1}}{2}\right) \tag{3.11}$$

$$y_{t+1} = y_t + \Delta D_t \cdot \sin\left(\frac{\theta_t + \theta_{t+1}}{2}\right) \tag{3.12}$$

where, ΔDR_t and ΔDL_t are the distance traveled by the left and right wheel over the time period from *t* to *t+1*. It is important to remember that these equations are only an approximation of the actual pose (x, y, *θ*). However, under most situations, uncertainties that arise from this approximation are small in comparison to other uncertainties that arise (such as wheel slippage). In any case, these uncertainties are affected by large changes in orientation (θ) between two consecutive time steps. Consequently, if quick changes in orientation are expected, a suitably small time step should be used. Simulation data indicated that these uncertainties were insignificant, for the proposed system.

A purely kinematic approach to tracking, such as the one derived above is susceptible to a number of error sources. These sources vary depending on the sensor configuration, but they all result in drift errors that cumulate over time (see Section 2.1.1). In this report, shaft mounted encoders were used to measure small variations in wheel rotation (details are included in Section 4.2). These values were then used to determine the traveled wheel distance with knowledge of the effective wheel radius. It is noted here that the effective wheel radius was experimentally tuned, but, the value was strongly dependent on the traveling surface. The sensor fusion detailed in Chapter 4 was able to compensate for these uncertainties. Encoders in particular, are vulnerable to uncertainties that arise due to wheel slippage. As noted in Section 2.1.1, these uncertainties have little impact on the overall vehicle displacement, but have detrimental effects on the orientation, which is used to derive positioning data. An example is presented here.

Suppose a differential drive vehicle travels a straight-line distance of 50 ft (15.2 m) along the x-axis of a coordinate system. Furthermore, assume that upon starting, the left wheel slips by a mere inch (2.5 cm). Based on eqn. 3.10 and an estimated wheelbase of 36 in (.91 m), the orientation of the vehicle will be off course by 1.60 deg (.028 rad). Now after traveling 50 ft (15.2 m), the vehicle anticipates that its new location is along the x-axis, however, based on eqn. 3.12 the vehicle is off course by over 16 in (.40 m) in the y-direction. This example is presented to reveal two important points. One, the kinematic equations of motion are extremely sensitive to uncertainties in orientation, and two, as mentioned in Section 2.1.1, odometry is a poor source for obtaining orientation.

As a final note, the equations of motion presented in this section represent the motion of a differential drive vehicle in a simplified 2D planar x, and y coordinate system. Although roadside medians and shoulders on average tend to be relatively flat, some sections are highly non-planar. As a result, localization based only on 2D tracking is inadequate. On the other hand, 3D tracking requires more sophisticated algorithms and sensors, which are complicated and costly. To avoid the limitations present for a 2D only model, additional non-planar compensation is provided by external global measurements. More insight into the details will be presented in Chapter 4.

3.4 Differential Tracking Control

In Section 3.3, the kinematic equations of motion were introduced for tracking purposes. Based on these equations, the robot's pose in relation to an initial starting location could be determined given only the control inputs, i.e. the left and right wheel speed and/or displacement. This process is an example of solving the forward kinematics. A more interesting, but challenging problem, deals with finding the control inputs required for an agent to reach a particular pose state, or to follow a specific trajectory [42]. This task is known as determining the vehicles inverse kinematics.

Tracking control for differentially steered vehicle is somewhat challenging due to natural nonholonomic constraints. For systems that contain such constraints, the number of independent degrees of freedom is less than the total number of independent generalized coordinates [52]. As an example, a differential vehicle on a 2D planar surface has three independent generalized coordinates (x, y, *θ*), while there are only two independent control inputs (actuation of the right and left wheel). Consequently, position and orientation are inherently coupled in the path tracking process and independent control in all three generalized coordinates (x, y, *θ*) is not possible.

Although differentially driven systems do not have independent control in the x, y, and θ coordinates, solutions do exist to the inverse kinematics that will yield a realized goal pose [42]. Perhaps one of the simplest solutions to the inverse kinematics of a differential drive robot is to: turn in place to face the desired x and y location, then drive forward until reaching this location, and then spin in place until reaching the desired orientation (*θ*). Clearly, this does not yield the most ideal solution and other solutions based on smoothly changing trajectories do exist [42]. The rest of this section is focused on introducing the tracking control algorithm used for the autonomous mowing agent developed in this report. This algorithm yields a smooth, exponentially convergent solution that tracks a predefined trajectory with zero steady state error.

Most studies aimed at tracking control of differentially steered vehicles have made use of tracking points that lie on the baseline (common axis between two drive wheels) [53]. In this case, exact tracking of both position and orientation is feasible. Yet, for many practical applications it is not possible to locate the tracking point on the baseline, thus, it is necessary to have tracking control algorithms for points that lie anywhere on the mobile robot. One such solution is presented in [53, 54]. This is the preferred approach taken, since the mechanical layout of the testbed discussed in this chapter calls for a cutting system offset forward from the baseline. A brief introduction to the control equations will be introduced below, but for a more detailed discussion see [53, 54].

Based on a schematic diagram of the testbed, displayed in Figure 3.12, the velocity and angular rotation of tracking point-A is defined by the following equations

$$v = \frac{b}{d}\left(\omega_R r_{\mathrm{Re}} - \omega_L r_{Le}\right) \tag{3.13}$$

$$u = \frac{1}{2}\left(\omega_R r_{\mathrm{Re}} + \omega_L r_{Le}\right) \tag{3.14}$$

$$\omega = \frac{1}{d}\left(\omega_R r_{\mathrm{Re}} - \omega_L r_{Le}\right) \tag{3.15}$$

where, r_{Re} and r_{Le} are the effective right and left wheel radius. Transforming equations 3.13 – 3.15 into matrix form and applying a coordinate frame transformation (from the local frame at point A to the global frame at point O yields the following matrix formulation

$$\begin{bmatrix} \dot{x}_A(t) \\ \dot{y}_A(t) \\ \dot{\theta}_A(t) \end{bmatrix} = \begin{bmatrix} \tfrac{1}{2} r_{\mathrm{Re}} \cos\theta(t) - \tfrac{b}{d} r_{\mathrm{Re}} \sin\theta(t) & \tfrac{1}{2} r_{Le} \cos\theta(t) + \tfrac{b}{d} r_{Le} \sin\theta(t) \\ \tfrac{1}{2} r_{\mathrm{Re}} \sin\theta(t) + \tfrac{b}{d} r_{\mathrm{Re}} \cos\theta(t) & \tfrac{1}{2} r_{Le} \sin\theta(t) - \tfrac{b}{d} r_{Le} \cos\theta(t) \\ \tfrac{1}{d} r_{\mathrm{Re}} & -\tfrac{1}{d} r_{Le} \end{bmatrix} \begin{bmatrix} \omega_R(t) \\ \omega_L(t) \end{bmatrix} \tag{3.16}$$

Since there are only two degrees of freedom for this system, only two independent variables can be tracked. Zhang, et al [53, 54] proposed using $x_A(t)$ and $y_A(t)$ for tracking control variables when the tracking point does not lie on the base line. This is the technique used herein. Their control law is defined as

$$\begin{bmatrix} \omega_R(t) \\ \omega_L(t) \end{bmatrix} = G_p^{-1}(\theta)\left[\begin{bmatrix} \dot{x}_B(t) \\ \dot{y}_B(t) \end{bmatrix} - K_p \begin{bmatrix} x_A(t) - x_B(t) \\ y_A(t) - y_B(t) \end{bmatrix}\right] \tag{3.17}$$

where,

$$K_p = \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix}, (k_x > 0, k_y > 0) \tag{3.18}$$

is the gain matrix which defines the exponential rate of convergence, and $G_p(\theta)$ is the first four terms from the 3x2 matrix in eqn. 3.16, and its inverse is defined as

$$G_p^{-1}(\theta) = \begin{bmatrix} -{}^{d}\!/_{2r_{\mathrm{Re}}b}\sin\theta(t) + {}^{1}\!/_{r_{\mathrm{Re}}}\cos\theta(t) & {}^{d}\!/_{2r_{\mathrm{Re}}b}\cos\theta(t) + {}^{1}\!/_{r_{\mathrm{Re}}}\sin\theta(t) \\ {}^{d}\!/_{2r_{Le}b}\sin\theta(t) + {}^{1}\!/_{r_{Le}}\cos\theta(t) & -{}^{d}\!/_{2r_{Le}b}\cos\theta(t) + {}^{1}\!/_{r_{Le}}\sin\theta(t) \end{bmatrix} \tag{3.19}$$

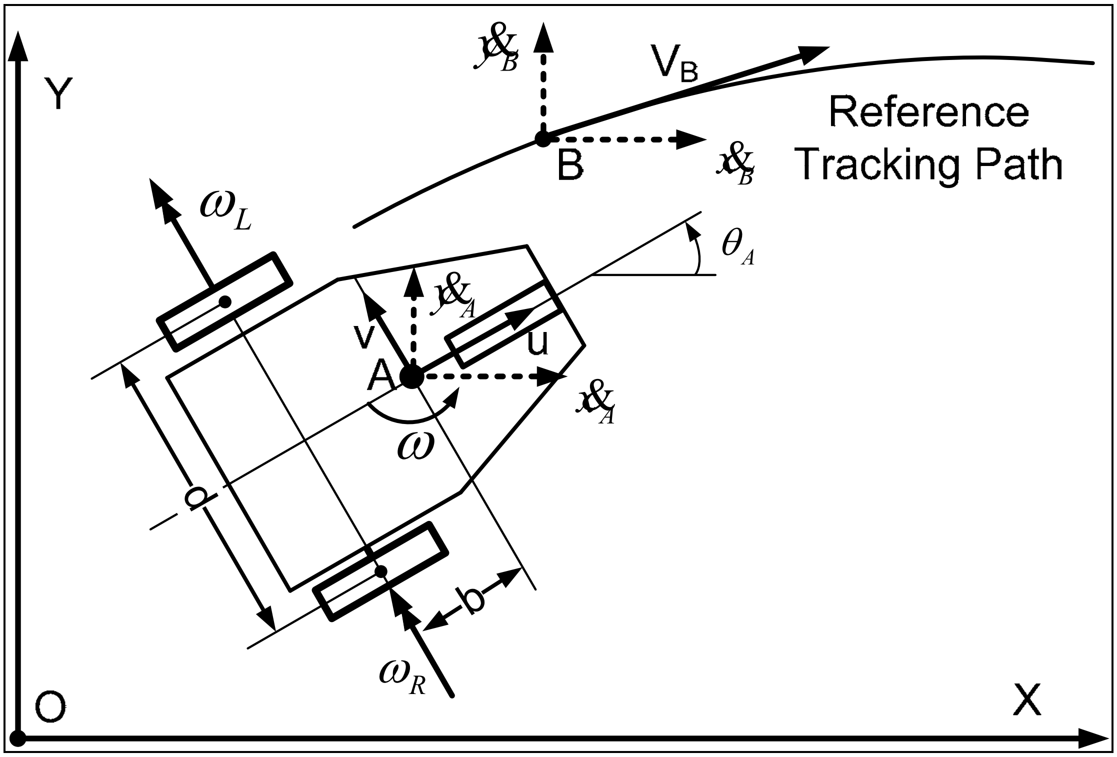


Figure 3.12 Schematic layout of the testbed along with tracking control parameters

Based on the tracking control law defined in eqn. 3.17, the vehicle control inputs ($\omega_R(t), \omega_L(t)$) are determined given the reference tracking path's components of velocity in the x and y global coordinate frame, and given the x and y error offset of the testbed tracking point from the desired reference path point. This control law yields exponentially convergent tracking with the rate equal to the minimum of k_x and k_y. As such, a vehicle can exactly track the x and y position of a predefined reference path. However, this solution does not yield independent control of the heading angle, and exact orientation tracking occurs only when the reference path is a straight line [53, 54]. For autonomous mowing, efficient mowing patterns yield long sections of straight cutting swaths, thus independent orientation tracking is not necessary (see Chapter 4).

As a final note, the tracking control implementation introduced (eqn. 3.17) is not computationally expensive and is thus well suited for real-time autonomous mowing. More discussion of the actual implementation and results of this control scheme will be covered in Chapter 4. Then, in Chapter 6, a similar control algorithm with robust control [53, 54] will be suggested for future prototype development.

3.5 Chapter Summary

Caltrans has expressed the need for innovative alternative vegetation maintenance techniques which are aimed at, reducing their dependence on herbicide, reducing labor costs, improving safety, and improving aesthetics. To target these needs, an autonomous roadside mowing agent has been proposed for further development. To assist in this development, a tentative set of customer requirements has been introduced (early in this chapter). These requirements are based on the theoretical characteristics, which Caltrans would require from such an agent.

Next, a detailed mechanical design analysis of the testbed used in this project was discussed. This testbed was intended as a research tool to evaluate the practicality of an autonomous mowing agent for the roadside environment and not as an absolute prototype. Regardless, its framework was selected to replicate that which would be practical of such a unit, i.e. with respect to size and layout. Making use of modular design techniques, the overall development time of this platform was significantly shortened.

Towards the end of this chapter, a kinematic model for the testbed was developed. Based on this model, the kinematic equations of motion were introduced. To make use of these equations for actual tracking (dead reckoning), a more practical discretized variation of these equations was discussed. Finally, a discussion of differential tracking control was covered. An adequate control law was introduced for exponential, zero steady state error tracking control. This tracking control was based on a tracking point which was offset from the baseline of the vehicle. For such a case, position tracking (x, y) is possible but orientation tracking is not.

CHAPTER 4 Localization, Mapping, and Path Planning Implementation and Simulation

Derived from the customer requirements, a detailed set of performance measures are introduced at the beginning of this chapter. Then, to maximize these measures, a suitable array of sensors and a governing sensor fusion algorithm are detailed, along with additional hardware support. After that, a discussion on mapping and path planning, along with the actual implementation is covered. Lastly, an overview of the real-time simulation, used for software development is discussed.

4.1 Performance Measures

To meet the customer requirements (listed in Section 3.1), some way of evaluating the overall success of an autonomous roadside mowing agent is necessary. As introduced in Section 2.1.1, performance measures provide a means to do just that. Performance measures should be developed based upon what someone desires from an agent and, not how someone envisions the agent should behave. Ideally, an agent that can mow the roadside quickly, efficiently, and effectively is desired.

A quick mowing rate is an obvious characteristic of success, i.e. the quicker a region of vegetation can be mowed, the less labor cost required, and the faster the next section can be targeted. The customer requirements (Section 3.1) stipulate that the overall mowing rate should be comparable to existing practices. For a single autonomous unit with a baseline of only 24 to 40 in (0.6 – 1.0 m, Section 3.2.2) this is not feasible, since a small unit of this size can simply not compare to the high capacity mowers normally used for roadside mowing. The ability to meet this requirement comes from the fact that autonomous technology can work continuously (even overnight) with little human intervention and also multiple units can be deployed. In any case, mowing rate is included as an obvious performance measure, since it is desired to have an agent that can mow at a speed of 3 mph (4.8 km/h) rather than only 0.5 mph (0.8 km/h).

Efficiency is another important qualification for successful autonomous mowing. The mower should never overlap previously covered paths and should always target uncut vegetation. Efficient mowing techniques reduce power consumption and support quicker mowing rates. Therefore efficiency is also included as an important performance measure.

A rational agent can maximize its efficiency by mowing in patterns with large gaps between successive cutting swaths. In this manner, the agent is assured that overlapping does not occur, and thus maximum efficiency is obtained. To counteract this situation, effectiveness is also included in the selection of performance measures. Effective mowing aims to minimize the amount of vegetation that is left behind, i.e. the agent is penalized for uncut vegetation. This is an important criterion for success to Caltrans, since missed patches of vegetation are costly to target by hand. Effective mowing is achieved by partially overlapping the previously cut swath, much the same as someone with a riding lawnmower would do. Overlapping reduces the efficiency and since efficiency and effectiveness act against one another, a suitable compromise is necessary in order to maximize the overall performance.

In summary, three performance measures have been selected to evaluate the rationality/success of the proposed autonomous roadside mowing agent; these are:

- A quick mowing rate
- Efficient area coverage
- Effective vegetation control.

4.2 Sensor Selection and Discussion

Sensor selection is perhaps one of the most important overall design responsibilities. Sensors provide the means by which an agent can perceive and act upon its environment. Proper sensor selection requires a clear understanding of the environment, the customer requirements, and the performance measures. For example, factors such as cost, application, size, and accuracy classically play an important role in this selection process. Accordingly, this section is focused on determining a suitable outfit of sensors that enable the proposed autonomous mowing solution to maximize its performance (in the selected environment), while meeting the customer requirements.

4.2.1 Sensor Selection

In order for an autonomous roadside mowing unit to supplement existing practices, Caltrans requires (Section 3.1) a cost effective solution. This means that Caltrans cost investment should be minimal in comparison to the long term net worth. Of course, the definition of net worth is somewhat subjective, because it is hard to put a price on someone's safety, and the proposed unit is aimed at improving the safety of others. Nevertheless, for the purpose of this report, a cost effective solution is one that will yield a significant reduction in labor, material, and equipment cost in comparison to the total cost investment. Hardware and sensor technology characteristically dominate the overall unit cost. It is the goal of this report to yield an effective but low cost sensor and hardware array.

Cost is not the only requirement that has an impact on the sensor selection. The customer requirements (Section 3.1) also stipulate that the unit should be easy to set up and collect. Therefore, if structural features are necessary, they should be simple to setup and low in cost. Many of the solutions presented in the literature review (Section 2.2) made use of pre-surveyed landmarks for autonomous mowing, i.e. buried ground wire, visual markers, and RFID tags. Depending on the setup requirements, such techniques may or may not be practical for autonomous roadside mowing. For example, installing the buried perimeter ground wire required for commercial platforms is labor intensive. Given that Caltrans maintains thousands of acres of vegetation spread out over thousands of lane miles, the labor and material costs required to install such an extensive infrastructure is not feasible. Another possibility would be to set up and then collect the perimeter guidance wire each time a section is to be mowed, once again hardly a fitting solution. On the other hand, a solution which requires a few RFID tags, or visual markers placed along the roadside might be feasible, that is of course if they are easy and cheap to install. Clearly a solution that does not require roadside structural modifications or extensive setup is better suited to meet Caltrans requirements.

The requirements also state that mowing rates should be comparable to existing mowing practices and that patches of vegetation should not be left behind. These requirements are reflected in the performance measures which judge the success of the agent based upon the speed, efficiency, and effectiveness of its mowing. These are important qualifications because Caltrans is forced to mow many thousands of acres statewide, and must comply with a strict timeline. This timeline takes into account seed germination, peak vegetation growth, rain, aesthetics, etc. and limits the allotted time upon which Caltrans has to mow particular sections of vegetation. Based on these qualifications it is obvious that the slow and inefficient mowing rates characteristic of random area coverage algorithms (Section 2.2.2) are not practical.

Based on many of the factors mentioned above, an approach similar to that taken in [30] has been adopted. Namely, relative sensors are used for odometry and fused with absolute GPS information to produce an accurate localization system upon which efficient, and effective mowing can be achieved. Unlike golf course turf management [10] which is susceptible to GPS signal losses from nearby foliage and buildings, GPS technology is well suited for the roadside environment, especially roadside medians and shoulders which are commonly free from such interference. That is not to say that GPS signal loss will never occur, since bridges, signs and other obstructions may interfere from time to time. However, the goal is to develop a solution which will overcome GPS outages over these short periods. Also, with the onset of sophisticated Global Imaging Software (GIS), GPS information is widely available and provides a convenient way upon which to define areas to be mowed. More on this topic will be covered in Chapter 5. GPS systems can be configured with a single differential station, or even without required structural or surveyed landmarks, yielding a quick setup time. The exact selection of relative and absolute sensors will be discussed next.

RELATIVE SENSORS: As mentioned in the background (Section 2.1.1), odometry and inertial navigation are the two most popular techniques for obtaining relative positioning information. Since inertial navigation is better suited for highly dynamic applications and not for slow moving vehicles, such as an autonomous mower, odometry has been selected as the preferred method. In particular, two relatively inexpensive ($139 each) shaft encoders from Encoder Products Company (Figure 4.1) were selected as the ideal choice. Other techniques such as visual odometry (via cameras) were overlooked due to cost and processing limitations. The two selected 15T encoders [55] are high performance quadrature encoders that could easily be interfaced to the NPC-T64 motors and AX2850 motor controller (Section 3.2.1 and 3.2.3). The assembly is discussed in Section 4.2.2 below and additional information on the features can be found at [55].

Figure 4.1 Encoder 15T from Encoder Products Company

As discussed in Section 3.3, for differential drive systems, wheel mounted encoders provide an excellent means for measuring displacement, but are prone to large errors in orientation. Past research has suggested that inertial rate gyroscopes are an excellent substitute for orientation information and a wide variety of studies [10, 15, 30-33] have taken this approach (i.e. encoders for displacement and gyroscopes for orientation). While the cost of such rate gyroscopes has dropped significantly during recent years, highly accurate units are still relatively expensive, e.g. the MTi by Xsens [12], which is on the order of $2000, and the E-Core RD2060 fiber optical gyroscope used in [10] that is roughly $2500. Instead, a unique approach was taken. Specifically, a GPS unit was used to provide accurate real-time heading information.

Relative measurements are susceptible to a variety of error sources. Thus, while relative measurements provide excellent accuracy over short periods of displacement, drift errors accumulate with time, yielding unbounded uncertainties (Section 2.1.2). To compensate for such unbounded errors, absolute measurement information is necessary.

ABSOLUTE SENSORS: Many absolute measurement systems have been proposed over the years; a few of these were introduced in Section 2.1.1. With the goal of minimizing cost and limiting the use of structural landmarks, a GPS unit has been proposed. As mentioned above, GPS is well suited for the roadside environment. Given the wide variety of GPS products that are available, proper selection should take into account factors such as the desired accuracy, cost, output frequency, and supporting equipment.

Aono et al [30] suggested that with proper fusion techniques, dead reckoning and GPS measurements could be combined to result in an accuracy of about 0.2 m with a GPS accuracy of only 1.0 m. This study evaluated the GPS performance by adding varying amounts of zero mean white noise to an expensive and highly accurate (~2 cm) kinematic GPS unit. In reality, GPS measurements are highly correlated (dependent on time), and a zero mean white noise approximation is subjective. More details on the performance and characteristics of GPS information will be covered in Chapter 5. Regardless, this study provided a rough basis upon which an appropriate GPS unit could be selected.

For efficient median and shoulder mowing, it was conceived that a final system accuracy on the order of about 1 ft (0.3 m) or better would be acceptable. Correspondingly, from Aono's [32] study this would suggest that a GPS unit with an accuracy of roughly 1.5 m (60 in) or better would be appropriate. Of course, the actual attainable accuracy is the result of a complex interaction between the individual sensors accuracies, sensors types, system dynamics, path trajectories, and the governing algorithms. Nevertheless, to achieve the hypothetical desired level of accuracy, a Crescent Vector OEM board from Hemisphere GPS was selected (Figure 4.2). This relatively low cost (~$995) board provides both accurate global positioning data and absolute heading information, using two separated antennas. The Crescent Vector board is able to exploit Wide Area Augmentation Satellite (WAAS) differential correction information to yield a horizontal accuracy of less than 0.6 m (24 in), 95% of the time (as published by the manufacturer). This configuration was convenient for two reasons: WAAS differential correction is widely available throughout the United States, and with WAAS correction, a high level of accuracy is attainable without requiring the setup of a differential station. Furthermore, with an antenna separation of only 1 m (39 in) an absolute root mean square (rms) heading accuracy of less than 0.15 deg (.0026 rad) is achievable (as published).

The Crescent Vector board was a fitting solution because it provided both positioning data and accurate bounded heading information. Even expensive high performance gyroscope units are not able to provide absolute orientation, although some units claim to use earth's magnetic field to compensate for drift. However, magnetic fields are susceptible to interference from a variety of sources. The Crescent board is also aided by an onboard single axis gyro that provides heading accuracy of better than 1 deg (0.02 rad) for periods of up to 3 minutes during GPS signal loss. In Chapter 5, experimental performance data will be provided, but for additional board specifications see [56], and for published performance data see [57].

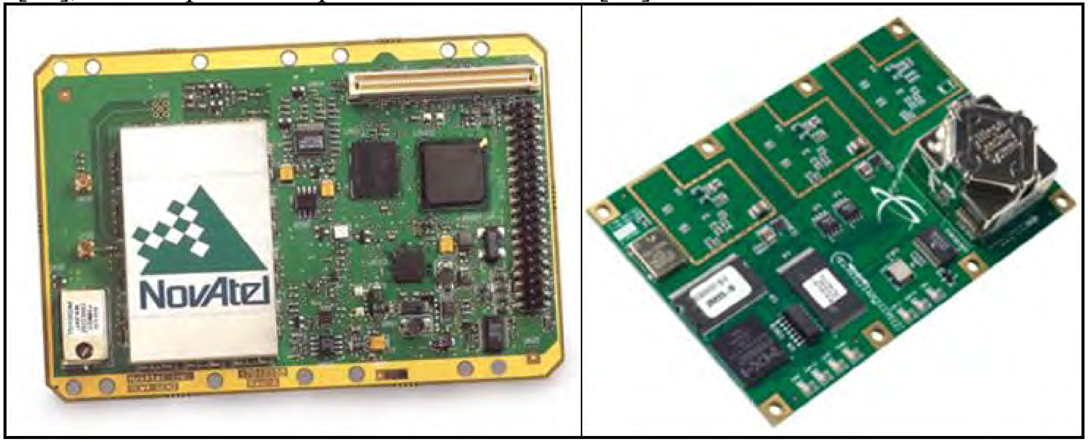


Figure 4.2 NovAtel OEMV-4 (left), Crescent Vector OEM (right)

In order to evaluate the performance of the proposed autonomous agent, it was necessary to have a way of measuring absolute positioning errors. Just as many other studies have done [10, 15, 30], a secondary high performance GPS unit was integrated onto the testbed to provide accurate ground truth (GT) data. The board of choice was a NovAtel OEMV-4 unit (Figure 4.2) running an advanced RT-2 processing algorithm, capable of maintaining absolute global position error on the order of 1 to 3 cm (0.4 to 1.2 in). With this level of accuracy, GT path data from the NovAtel unit could be used to evaluate the overall system performance (Chapter 5). For additional board specifications and published performance data see [58, 59].

4.2.2 Mechanical and Electrical Sensor Configuration

For precise odometry, the selected 15T encoders (Figure 4.1) were coupled directly to the two NPC-T64 motors (Section 3.2.1), rather than the gearbox output shaft (which has a gear reduction of 20:1). The design of the NPC-T64 motor provided for convenient encoder mounting via a pre-tapped standard 10-24 hole in the rear of the motor shaft, and a rear mounting enclosure (Figure 4.3). To pair the encoder to the motor, a standard 1.0 in (2.5 cm) long, 1/4th in (0.6 cm) diameter shoulder bolt with a threaded 10-24 end was used (Figure 4.3). Then, two tapped 6-32 holes were used to bolt the encoder fixture to the T64 back casing. The encoder fixture is a proprietary Flex-Mount design that is tolerant to axial misalignment [55].

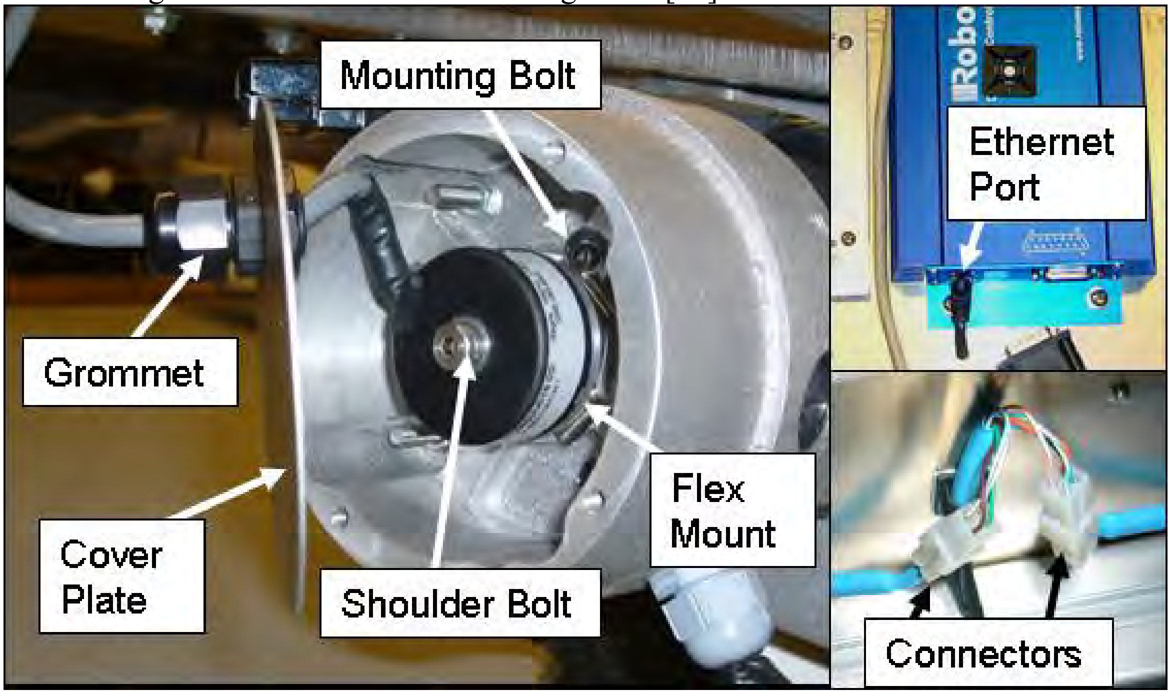


Figure 4.3 Motor mounting enclosure (left), AX2850 Ethernet port (top-right), Encoder to Ethernet connection (bottom-right)

To protect the encoder assembly from particulate matter, a thin aluminum plate (included with the T64 motor) was fastened to the end of the motor casing. Then, using a water tight grommet, the electronically shielded encoder wire was ported through this plate (Figure 4.3). The two encoder lines were routed to meet at the center of the testbed where they were linked to a standard 8-lead twisted pair Ethernet line by means of two wire connectors (Figure 4.3). The Ethernet line was ported through the aluminum frame mounting plate for access to the AX2850 controller Ethernet port (Figure 4.3).

To mount the Crescent Vector GPS board to the testbed, the board was placed in a shielded electronic aluminum enclosure, provided with the OEM board (Figure 4.4). As recommended by the manufacturer, the board should be mounted on a horizontal plane for higher performance. Therefore, the enclosure was mounted next to the AX2850 motor controller, on top of the aluminum mounting plate (Figure 4.4), which was nearly horizontal (off by less than one degree). Mounting was accomplished by using two custom made 1x1 in (2.5x2.5 cm) corner casing brackets.

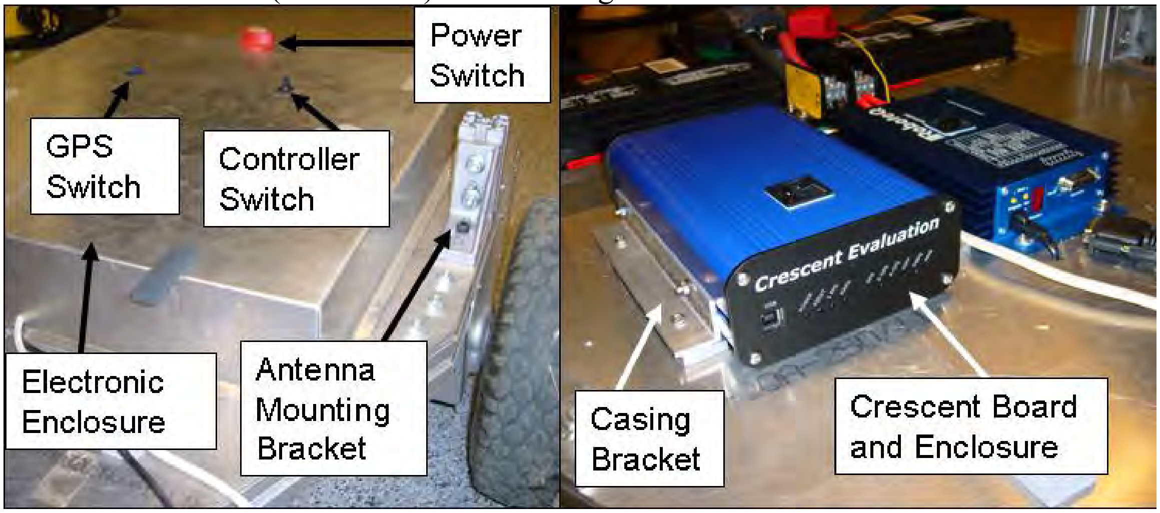


Figure 4.4 Electronic Enclosure (left), Crescent Vector mounting (right)

Power was supplied to the Crescent Vector board by making use of the system power supply (automotive 12 volt batteries) and a configured GPS power switch (Figure 4.4). An internal voltage regulator was used to reduce the supplied voltage from 12 volts (linked across only one battery) to the required 3.3 volts. To reduce GPS signal interference (from electrical lines), an electronic enclosure was formed out of 1/16th inch (1.6 mm) aluminum (Figure 4.4). This enclosure was also intended to protect the electronics, batteries, and hardware from damage. For ease, the GPS, main power, and AX2850 controller switch were ported through the electronic enclosure (Figure 4.4).

For accurate heading information, the Crescent Vector OEM board requires two different antennas, separated by a predefined distance. Therefore, two CSI Wireless CDA-3RTK [60] antennas were mounted to an aluminum beam at a separation of 0.8 m (31 in) (Figure 4.5). This beam was machined out of 2x2 in (5x5 cm), 1/8th in (3 mm) thick aluminum tubing, for rugged and precise mounting. The performance of the absolute measured heading is directly correlated to the antenna separation. It was determined in previous experimentation that a separation of 0.8 m (31 in) produced accurate heading measurements at an acceptable acquisition rate (i.e. the further apart the antennas are, the better the precision, but the slower the heading is acquired). This separation also remained within the selected outer vehicle profile (~38 in wide). Heading and positioning performance data will be referenced in Chapter 5. To mount the antenna beam (Figure 4.5), a mounting bracket was built out of a 45x90 Gusset, a short section of Bosch aluminum 45x45 tubing (Figure 3.5), and two machined mounting plates.

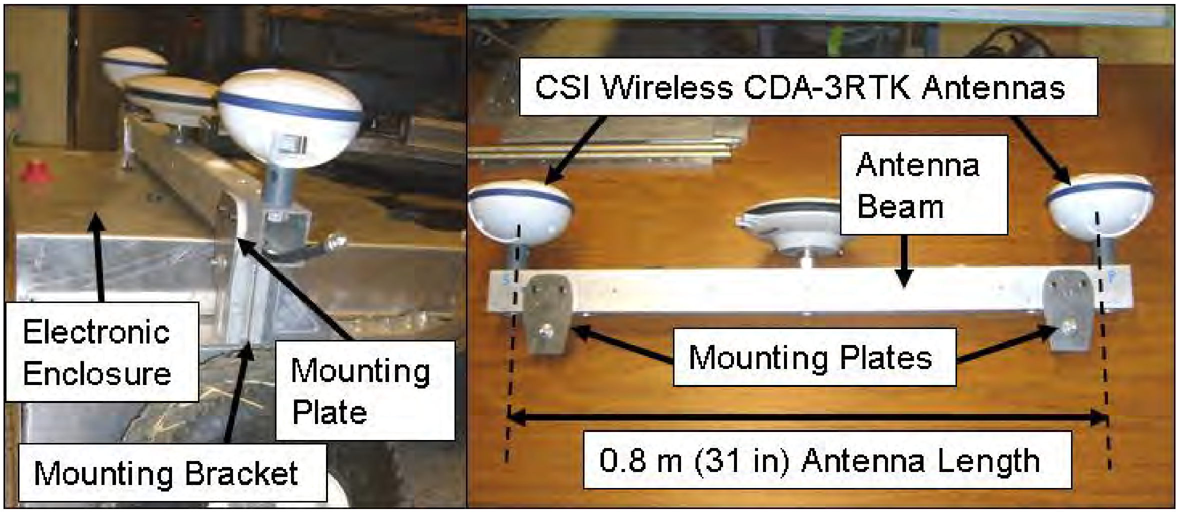


Figure 4.5 Antenna beam mounting (left), antenna layout (right)

Although the NovAtel OEMV-4 board (Figure 4.2) was used for final performance testing (as a source for GT data), it was not a component of final system sensor array and hence did not require permanent attachment. To integrate the unit, the OEMV-4 board was placed within a protective NovAtel box and affixed to the top of the electronics enclosure with adhesive Velcro (Figure 4.6). To supply power to the NovAtel board an external 12 volt battery was used (Figure 4.6), and for signal reception, a NovAtel GPS-702 antenna [61] was attached to the antenna beam (Figure 4.5). Finally, a PDA was used for GPS data longing; the PDA was linked to the NovAtel unit by means of a serial to PDA wire link.

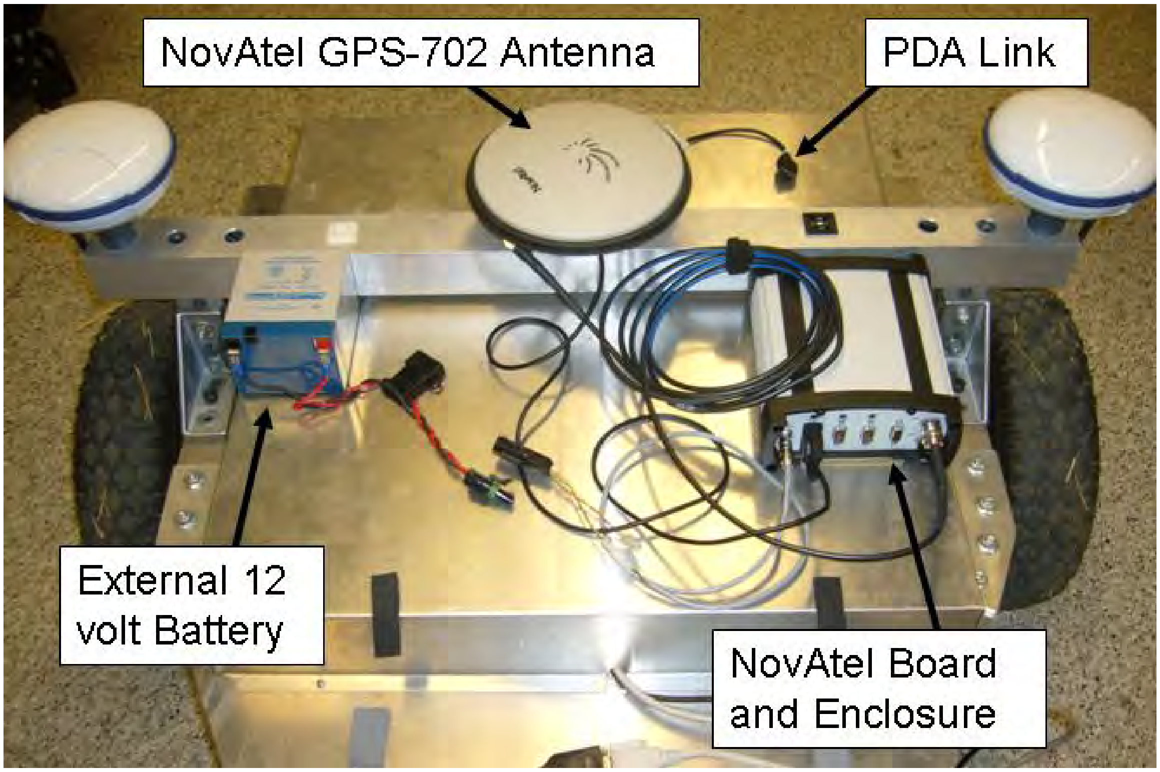


Figure 4.6 NovAtel OEMV-4 configuration

4.2.3 Additional Considerations

In Section 3.3, the tracking equations for a differential drive vehicle in a 2D planar coordinate system were presented. It was noted at that time that tracking based only on a 2D model would be inadequate, since these equations neglect variations in the terrain, such as bumps, small hills, and inclines, which are likely throughout median and shoulder sections. While, a relatively planar surface is expected on average, a robust solution that can account for these uncertainties (when they do occur) is desirable. Many studies [30, 62] have countered such issues by developing more complete 3D tracking models that make use of full attitude sensing (3D orientation). In general, full attitude sensing is accomplished by rather expensive 3-axis gyroscopes.

Alternatively, Kiriy [10] suggested that the errors that arise from a 2D planar approximation can be compensated for by incorporating absolute sensor information. In addition, Fuke et al [62] determined that for most low dynamic situations a 3D dead reckoning model (using accelerometers and gyroscopes) provided no significant positioning improvement over a 2D only model. Therefore, a 2D planar approximation that uses absolute GPS positioning information has been selected as the best approach. Fundamentally, absolute GPS information is used to continually correct for the inaccuracies that result from the 2D tracking model. GPS positioning information, obtained from absolute latitude and longitude measurements, can be viewed as a projection of 3D data onto a 2D map.

It is important to realize that positioning measurements from the GPS unit are subject to geometric antenna positioning variations. Two scenarios are possible: the vehicle is traveling up or down an incline, in which case tangential antenna errors arise, or the vehicle is traveling along an incline (or one wheel is rolling over an obstacle), in which case a transverse antenna error results (Figure 4.7). Of course a combination of these two scenarios is also possible. As it turns out, tangential errors have little impact on the mowing performance (Chapter 5), while transverse errors have a significant impact. However, since the proposed testbed has a low antenna height (~22 in, 56 cm) these errors tend to be quite small (depending on the vehicle pitch or roll angle), and were neglected for initial testing. Further insight into the effect of these errors on the performance is discussed in Chapter 5. It is noted that with an antenna height of 22 in (56 cm) and a roll angle of 5 deg (0.09 rad) the transverse error is less than 2 in (5 cm) and for a roll angle of 10 deg (0.17 rad) the error is less than 4 in (10 cm).

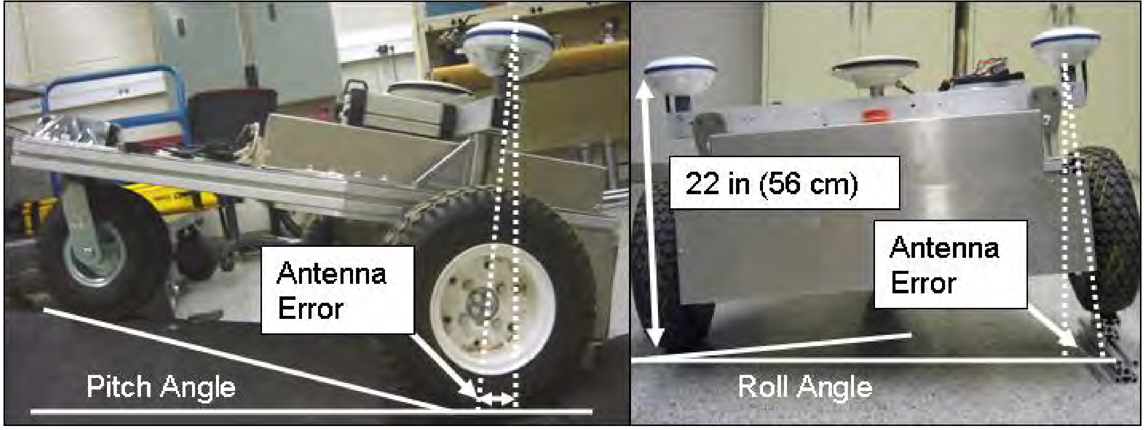


Figure 4.7 Tangential antenna error (left), transverse antenna error (right)

4.3 Proposed Hardware and Communications Implementation

To make a robot truly functional, some way of recording, interpreting, and processing sensor information is needed. This is classically done by interfacing sensors to some form of computing device, such as a personal computer or micro-controller. The exact manner by which sensors and hardware communicate (i.e. the physical wires, protocols, and messaging architecture) classify the overall communications layout. A particular layout may make use of one or more of a variety of networking communications, such as RS-232, RS-485, RS-422, USB, CAN, or Ethernet. One of the most popular modern communication layouts for robotics research [63], and the approach taken in this report, is the layout displayed in Figure 4.8.

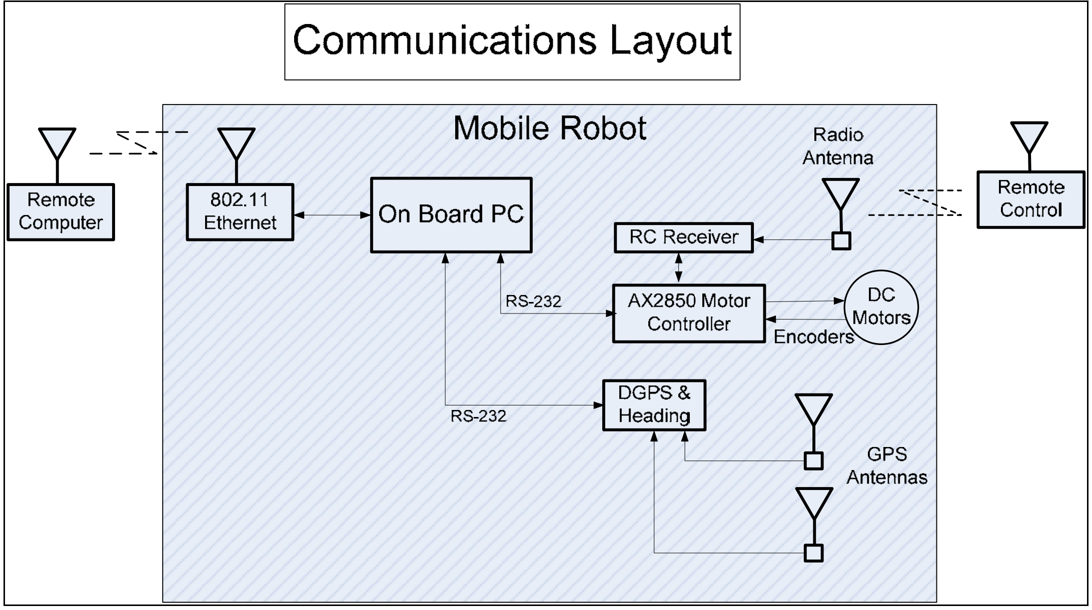


Figure 4.8 Testbed communications layout

The proposed communications layout (Figure 4.8) makes use of an onboard personal computer (PC), which serves as the main controller for the robot. For the detailed testbed, an onboard laptop computer was used to serve this function (Figure 4.9). This laptop computer was an IBM G40 ThinkPad, with a 3.0 GHz Intel Pentium-4 processor, 1.0 GB of RAM, and was running the Windows XP operating system environment. Making use of an onboard laptop is advantageous during research and development, because a laptop has a built in screen and keyboard to facilitate local diagnostics. Of course, future prototype development necessitates replacement with a more compact micro-computer or PC. In recent years, the size and cost reduction of small single board PC's has yielded them suitable for many robotics applications, e.g. the EBC-855-G single board computer with a 1.8 GHz Pentium M CPU [64].

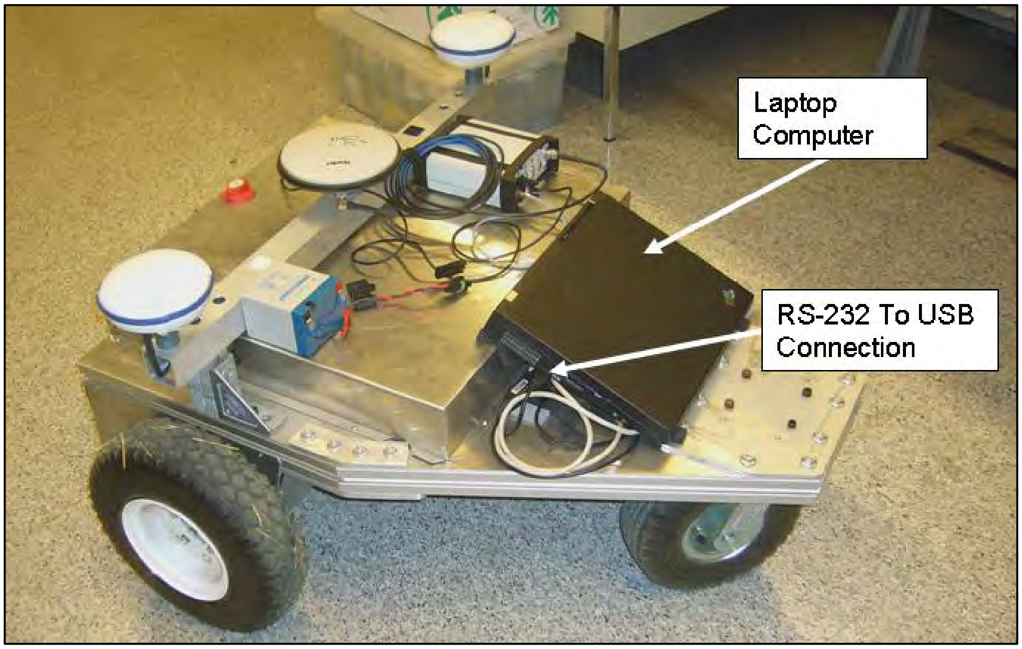


Figure 4.9 Onboard Laptop computer

Communication to both the Crescent Vector GPS board and the AX2850 motor controller was accomplished by using two separate RS-232 serial port lines (Figure 4.8). It is noted, that the AX2850 controller acts as a medium for which motor control and encoder readings can be accomplished. Since no serial ports were readily available on the laptop, a simple 4-port RS-232 to USB cord was used for connection (Figure 4.9). RS-232 is a popular communications medium for most present day sensors, although advancements in Ethernet technology have lead some to experiment with communication layouts which use linked Ethernet sensors [63]. For additional details on the AX2850 and Crescent Vector communication protocols, as well as supporting software development see Appendix B.

In addition to the sensor interfacing, Figure 4.8 displays two additional communication links, a wireless Ethernet connection from a remote computer to the mobile robot, and a wireless radio link from a remote controller to the mobile robot. Although radio communication was never established, the tentative electrical layout is listed in Appendix A, and it was suggested for future prototype development. Namely, the R/C link gives direct vehicle control to an end user (perhaps a Caltrans employee) for easy deployment, collection, and for emergency stops. In contrast, the wireless Ethernet connection was actually implemented, but not widely used. This link was proposed for several reasons. For one, it would allow a Caltrans employee the ability to wirelessly update, record, and configure the onboard PC, e.g. they might upload mowing GPS coordinates, or retrieve mapping information. Second, it provides a means by which multiple mowing units might be able to communicate with one another. Finally, it adds an additional layer of control to the end user for safety reasons. The physical software implementation of this link was an asynchronous TCP Ethernet connection.

4.4 Localization, Mapping, and Path Planning

To make decisions based on sensor percepts an agent must contain some form of internal mapping between its percepts and actions. This mapping is referred to as an agent program (Section 2.1.1), and is typically implemented in software on some form of computing device. For the proposed autonomous mowing agent, the computing device was an IBM ThinkPad (Section 4.3), and the selected software programming language was C#. C# is a powerful object-oriented programming language build on top of the Visual Studio .NET platform. Not to overwhelm the reader, only a broad overview of the developed software will be included as necessary throughout the rest of this chapter. For a more detailed analysis of the developed software application the reader is referred to Appendix B.

For the planned mowing agent to maximize its performance measures (i.e. a fast mowing rate, and efficient and effective coverage) it must be able to localize itself within its environment, map covered areas, and plan its future path. These are all tasks which must be accomplished within the agent program (software). To assist in the overall methodology, a preliminary dataflow and control model for the developed testbed is included in Figure 4.10. This figure demonstrates the overall system processing flow, which is described as follows.

SYSTEM DATA FLOW: As directed by the agent function, the desired left and right wheel speeds are conveyed to the AX2850 motor controller. The motor controller in turn sends PWM signals to each of the motors for correct actuation and records the movement of the wheel rotation by means of shaft mounted encoders. Encoder information is collected by the agent program and combined with previous pose information to develop a new initial state estimate. This state estimate is then fused with GPS pose information by means of an Extended Kalman filter (EKF) algorithm to determine the final state estimate. Based on this final state estimate, prior knowledge, and an internal coverage mapping, a planned path is developed. Finally, depending on the current state estimate and the desired path trajectory (as specified by the planned path), new control inputs for the left and right desired wheel speed are calculated and conveyed to the AX2850 controller. The closed-loop process then begins its next cycle.

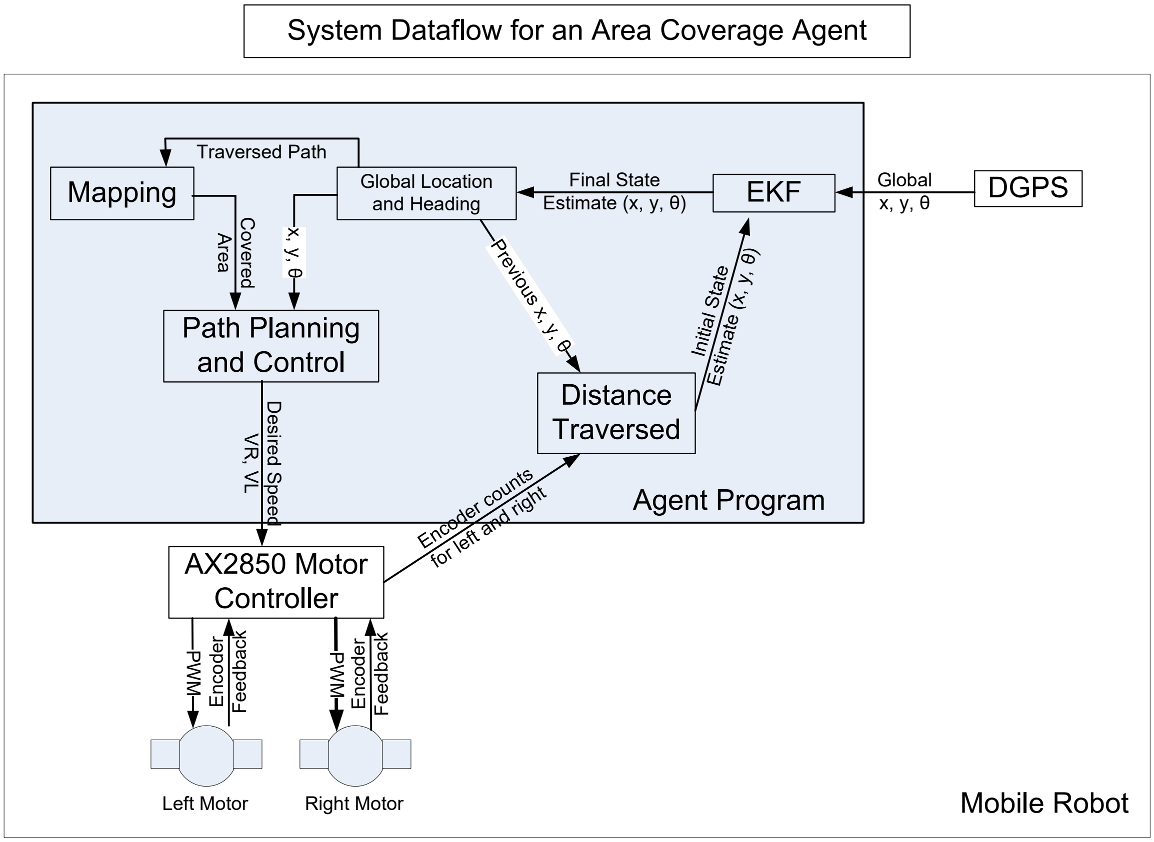


Figure 4.10 Preliminary system dataflow

The building blocks for the model introduced in Figure 4.10 are discussed throughout the rest of this chapter.

4.4.1 Localization

Two distinct sensors units have been selected for localization, i.e. quadrature encoders, and a DGPS unit (Section 4.2). To fuse noisy (stochastic) information from both sources, an Extended Kalman Filter (EKF) has been selected and utilized (the details are developed within this section). Preliminary information on the Kalman Filter and why it is well suited for accurate real-time robotic localization was covered in Section 2.1.2. In fact, it was mentioned in Section 2.1.2, that even if the conditions required for Kalman filtering optimality do not exist, the filter has been show to perform relatively well; rarely do the conditions for optimality actually exist in real-life.

Many studies have successfully made use of the EKF for fusion of GPS and dead reckoning information [30, 65]. Such a solution benefits from the short term accuracy of dead reckoning and the long term drift free bounding of GPS data. Also, GPS information is an excellent way of initializing the EKF state estimate and error covariance to prevent divergence, which is a common setback of the Kalman filter when a bad initial guess exists (Section 2.1.2). Since a 2D planar coordinate system has been selected for localization (Section 4.2.3), three state variables are of importance: the absolute x and y positioning and the heading orientation (θ). As such, a 3-state EKF for stochastic state estimation has been proposed. The approach taken is similar to that used in [10, 29] with appropriate modifications for the system under discussion.

THE EKF MODEL: The discreet EKF model makes use of state transition and observation models which need not be linear functions, but only differentiable functions [20]. These functions are represented as

$$x_{k+1} = f(x_k, u_{k+1}; w_{k+1}, \gamma_{k+1}) \tag{4.1}$$

$$z_{k+1} = h(x_{k+1}; v_{k+1}) \tag{4.2}$$

where, u_{k+1} is the system control input and w_{k+1}, γ_{k+1}, and v_{k+1} are the system process, input, and measurement noise. These equations are similar to that of the linear Kalman filter model (eqn. 2.1 and 2.2) introduced in Section 2.1.2, with two distinct differences: the state transition and observation models represented by eqn. 4.1 and 4.2 may be nonlinear functions, and also, it is assumed that there is an additional uncertainty (γ_{k+1}) associated with the input (u_{k+1}). Note: if the input (u_{k+1}) is known exactly then there is no need to include the input noise (γ_{k+1}) into the EKF model. As with the linear model introduced in Section 2.1.2, it is assumed that the distributions of the uncertainties (w_{k+1}, γ_{k+1} and v_{k+1}) are additive, normal, zero mean, white, and uncorrelated.

Similar to how the linear Kalman filter works, the state and error covariance prediction are updated based on the input, previous state estimate, the system model, and knowledge of the input and process noise, as follows,

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, u_{k+1}, 0, 0) \tag{4.3}$$

$$P_{k+1|k} = A_{k+1} P_{k|k} A_{k+1}^T + B_{k+1} \Gamma_{k+1} B_{k+1}^T + Q_{k+1} \tag{4.4}$$

where, Q_{k+1} is the covariance of the process noise w_{k+1}, Γ_{k+1} is the covariance of the input noise γ_{k+1}, and

$$A_{k+1} = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_{k|k}} \tag{4.5}$$

$$B_{k+1} = \left.\frac{\partial f}{\partial u}\right|_{u_{k+1}} \tag{4.6}$$

Eqn.s 4.5 and 4.6 are commonly known as the system Jacobians, and they represent the linearization of the state transition model (eqn. 4.1) with respect to the previous state estimate, and the input.

For the differential drive system under consideration, an acceptable discretized state transition (process) model was derived in Section 3.3. This localization model is detailed as,

$$x(k+1) = f_x(k) = x(k) + \left({}^{\Delta D_R(k) + \Delta D_R(k)}\!\big/_{2} \right) \cdot \cos\left({}^{\theta(k) + \theta(k+1)}\!\big/_{2} \right) \tag{4.7}$$

$$y(k+1) = f_y(k) = y(k) + \left({}^{\Delta D_R(k) + \Delta D_R(k)}\!\big/_{2} \right) \cdot \sin\left({}^{\theta(k) + \theta(k+1)}\!\big/_{2} \right) \tag{4.8}$$

$$\theta(k+1) = f_\theta(k) = \theta(k) + \left({}^{\Delta D_R(k) - \Delta D_R(k)}\!\big/_{d} \right) \tag{4.9}$$

where, $\Delta D_R(k)$ and $\Delta D_R(k)$ are the distance traveled by the right and left wheel respectively, and d is the wheelbase length (Figure 3.12). The above nonlinear equations correspond to eqn. 4.3, where the predicted system state vector is $\hat{x}_{k+1|k} = [x(k+1) \quad y(k+1) \quad \theta(k+1)]^T$, the system function is $f(x) = [f_x(k) \quad f_y(k) \quad f_\theta(k)]^T$, and the input vector is $u_{k+1} = [\Delta D_R(k) \quad \Delta D_L(k)]^T$. Based on the state transition model (equations 4.7 – 4.9) the Jacobians, given by eqn. 4.5 and 4.6, were evaluated to be:

$$A_{k+1} = \begin{bmatrix} \frac{\partial f_x}{\partial x_k} & \frac{\partial f_x}{\partial y_k} & \frac{\partial f_x}{\partial \theta_k} \\ \frac{\partial f_y}{\partial x_k} & \frac{\partial f_y}{\partial y_k} & \frac{\partial f_y}{\partial \theta_k} \\ \frac{\partial f_\theta}{\partial x_k} & \frac{\partial f_\theta}{\partial y_k} & \frac{\partial f_\theta}{\partial \theta_k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta D_k \cdot \sin(\theta_{Avg}) \\ 0 & 1 & \Delta D_k \cdot \cos(\theta_{Avg}) \\ 0 & 0 & 1 \end{bmatrix} \tag{4.10}$$

where, $\Delta D_k = \left({\Delta D_R(k) + \Delta D_R(k)} \big/ 2 \right)$ and $\theta_{Avg} = \left({\theta(k) + \theta(k+1)} \big/ 2 \right)$ for simplicity and

$$B_{k+1} = \begin{bmatrix} \frac{\partial f_x}{\partial \Delta D_L(k)} & \frac{\partial f_x}{\partial \Delta D_R(k)} \\ \frac{\partial f_y}{\partial \Delta D_L(k)} & \frac{\partial f_y}{\partial \Delta D_R(k)} \\ \frac{\partial f_\theta}{\partial \Delta D_L(k)} & \frac{\partial f_\theta}{\partial \Delta D_R(k)} \end{bmatrix} = \tag{4.11}$$

$$\begin{bmatrix} \frac{1}{2}\cos(\theta_{Avg}) + \frac{\Delta D_k}{2d}\sin(\theta_{Avg}) & \frac{1}{2}\cos(\theta_{Avg}) - \frac{\Delta D_k}{2d}\sin(\theta_{Avg}) \\ \frac{1}{2}\sin(\theta_{Avg}) - \frac{\Delta D_k}{2d}\cos(\theta_{Avg}) & \frac{1}{2}\sin(\theta_{Avg}) + \frac{\Delta D_k}{2d}\cos(\theta_{Avg}) \\ -1/d & 1/d \end{bmatrix}$$

A few notes are in order pertaining to the equations derived above. The prediction phase of the EKF, which makes use of eqn. 4.3 and 4.4, is evaluated by using internal (relative) measurements only. For the proposed system, the encoders, which yield the left and right wheel displacement ($\Delta D_R(k), \Delta D_L(k)$), are the only relative measurements, and they provide sufficient information upon which to predict the entire state (x, y, and θ) and error covariance. Most other related studies [10, 30, 32] use a similar approach (i.e. odometry for wheel displacement) but compliment relative sensor measurements with more accurate orientation information from an inertial rate gyroscope (Section 4.2.1). Instead of taking this approach, large uncertainties in orientation are corrected for by accurate heading measurements from the GPS unit, as discussed below.

While eqn. 4.3 is sufficient for providing simple relative localization (dead reckoning), uncertainties tend to propagate rapidly (depending on the sensor implementation and application), as recorded by the estimated error state covariance (eqn. 4.4). However, by incorporating observations (or measurements) of the true state, the state estimate can be significantly improved. This compensation process is accomplished during the update phase of the EKF, which is quite similar to update phase of the linear Kalman filter (Section 2.1.2). Overall, the EKF works in a predictor-corrector style.

If a given measurement is acquired at time $k+1$, then the state estimate and error covariance matrices are updated using the following equations:

$$K_{k+1} = P_{k+1|k} H_{k+1}^{T} \left(H_{k+1} P_{k+1|k} H_{k+1}^{T} + R_{k+1} \right)^{-1} \tag{4.12}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \left(z_{k+1} - h(\hat{x}_{k+1|k}, 0) \right) \tag{4.13}$$

$$P_{k+1|k+1} = \left(I - K_{k+1} H_{k+1} \right) P_{k+1|k} \tag{4.14}$$

where, R_{k+1} is the covariance of the measurement noise v_{k+1}, I is the identity matrix, and

$$H_{k+1} = \left. \frac{\partial h}{\partial x} \right|_{\hat{x}_{k+1|k}} \tag{4.15}$$

Eqn. 4.15 is the measurement Jacobian, which linerarizes the measurement model (eqn. 4.2) about the predicted state estimate. Since the selected Crescent Vector GPS unit (Section 4.2.1) provides both absolute positioning and heading, the measurement vector is defined as $z_{k+1} = \left[x(k+1) \quad y(k+1) \quad \theta(k+1) \right]^{T}$. It is clear then, that the measurements are directly related to the system state variables $x(k+1)$, $y(k+1)$ and $\theta(k+1)$, implying that the measurement model (eqn. 4.2) is actually a linear function of the state variables, and that the observation matrix H_{k+1} is the 3x3 identity matrix. Thus, the measurement model (eqn. 4.2) can be reevaluated to be

$$z_{k+1} = x_{k+1} + v_{k+1} \tag{4.16}$$

and since the observation matrix is the 3x3 identity matrix, eqn. 4.12 – 4.13 can be simplified to yield

$$K_{k+1} = P_{k+1|k} \left(P_{k+1|k} + R_{k+1} \right)^{-1} \tag{4.17}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \left(z_{k+1} - \hat{x}_{k+1|k} \right) \tag{4.18}$$

$$P_{k+1|k+1} = \left(I - K_{k\_1} \right) P_{k+1|k} \tag{4.19}$$

Based on these equations, a few fundamental observations about the Kalman filter's performance can be made; the same observations can be made about eqn. 4.12 – 4.14. If the measurements are exact (contain no noise), then the measurement covariance (R_{k+1}) is zero and the gain (K_{k+1}) becomes the identity matrix. Plugging this gain into eqn. 4.18 yields $\hat{x}_{k+1|k+1} = z_{k+1}$, i.e. the predicted state estimate is exactly the measurement vector. This is expected because the measurements have been assumed to be perfect (contain no noise). Similarly, the system error covariance (eqn. 4.19) becomes zero, which is also expected, since the measurements are exact. Conversely, as the measurement covariance (uncertainty of the measurement) gets larger, the gain (K_{k+1}) becomes smaller, and thus from eqn. 4.18, the impact that the measurement has on the initial state prediction becomes smaller. Simply put, the less accurate the measurements become, the less they are trusted, and the more the initial prediction is trusted. Of course, the actual fusion of the initial state prediction and the measurements is based on a complex interaction of individual sensor accuracies and the current state error covariance.

Figure 4.11 is presented to provide an overall illustration of the recursive predictor-corrector EKF process (based on the system under discussion). To begin the overall recursive cycle, the starting state and error covariance are used to initialize the algorithm. As noted earlier, the starting GPS position and heading, along with their corresponding uncertainties, are an excellent way to initialize the EKF process. The cycle continues when the left and right wheel displacement are recorded (via the encoders) over a time period of *k* to *k+1*. With these values, the EKF state prediction is updated (using eqn. 4.3 and 4.4) to yield the estimated state and error covariance at time *k+1*. If no measurements at time *k+1* are recorded, then the predicted state estimate and error covariance become the final values at time *k+1*. However, if GPS measurements for the global positioning and heading are recorded at time *k+1,* then the initial state and error predictions are updated using the measurements (using eqn. 4.17-4.19). The recursive cycle continues when new readings of the wheel displacements are recorded.

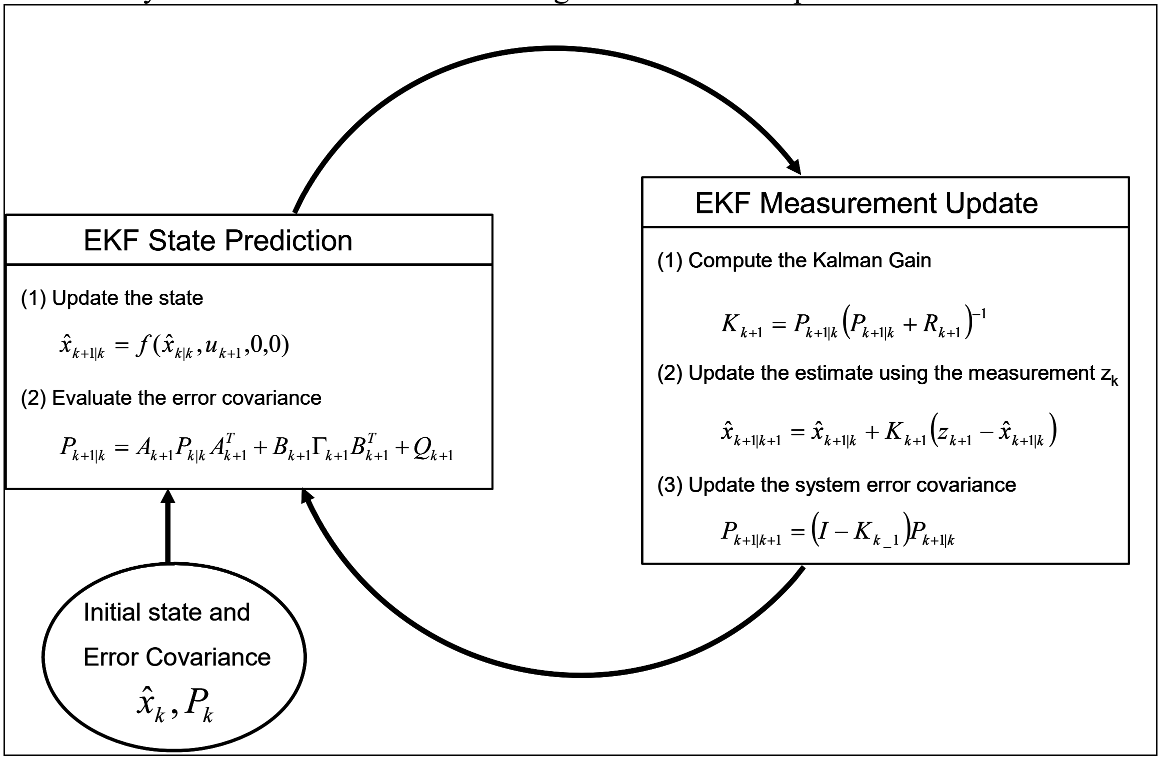


Figure 4.11 Process flow for the system EKF

PROCESS, INPUT, AND MEASUREMENT NOISE: To complete the EKF model introduced above, it is necessary to have a firm understanding of the system process, input, and measurement covariance matrices, given that these matrices dominate the success of the fusion process. To begin, development of the system process and input noise covariance matrices (Q_{k+1}, Γ_{k+1}) will be discussed. Then, the measurement noise covariance matrix (R_{k+1}) will be covered. Many different techniques have been used to model the system process noise, however, the overall technique is largely dependent on the application in question. Some techniques make use of a simple trial and error procedure to yield an acceptable filter performance, but this is a time consuming procedure and suggests reason for a more empirical model. Therefore, to evaluate the system process noise, an approach similar to that proposed by Chenavier and Crowley [66] was taken.

Chenavier and Crowley [66] developed a process noise model based on a similar set of differential drive tracking equations (eqn. 4.7-4.9). Their model is defined as

$$Q_{k+1} = \begin{bmatrix} Q_{11}(k+1) & 0 & 0 \\ 0 & Q_{22}(k+1) & 0 \\ 0 & 0 & Q_{33}(k+1) \end{bmatrix} \tag{4.20}$$

where,

$$Q_{11}(k+1) = K_{dd}\left|\Delta D_k \cdot \cos\theta\right| \tag{4.21}$$

$$Q_{22}(k+1) = K_{dd}\left|\Delta D_k \cdot \sin\theta\right| \tag{4.22}$$

$$Q_{33}(k+1) = K_{d\theta}\left|\Delta D_k\right| + K_{\theta\theta}\left|\Delta\theta_k\right| \tag{4.23}$$

$\Delta\theta_k = \left({\Delta D_R(k) - \Delta D_R(k)} \big/ {d} \right)$, K_{dd} is the odometry drifting coefficient along the distance traveled with respect to an incremental distance ΔD_k, $K_{d\theta}$ is the odometry drifting coefficient along the heading with respect to incremental distance ΔD_k, and $K_{\theta\theta}$ is the odometry drifting coefficient along the heading with respect to incremental heading $\Delta\theta_k$. This model assumes that the process errors are uncorrelated and hence eqn. 4.20 is diagonal. Eqn. 4.20 is intended to approximate the combined system process error, which may encompass a variety of error sources such as variations in wheel radii, wheel slippage, vibrations, and variability in the terrain.

Chenavier and Crowley's technique models the error progression as a random walk process [67]. A random walk process works as follows. If the process in question represents the time derivative of the real variable of interest, then the variance of the integral grows linearly with time and the standard deviation grows with the square root to time. The process equations for the tracking of a differential drive robot work in much the same way. That is, the variables of interest (x, y, θ) are derived by integrating small variations in displacement and orientation. For this model, the variance in the position and orientation grow linearly with respect to variations in displacement and orientation. To understand how this works the following example is presented.

EXAMPLE: If the proposed testbed is driven 50 ft (15.2 m) in a straight line, then some variability in the exact displacement would be expected. Some times it might travel only 49 ft (14.9 m), while other times it might travel 51 ft (15.5 m). This variability is dependent on a few factors such as the wheel slippage, variations in the wheel radii, and variations in the terrain. If these differences are recorded over many trials and the variance and standard deviation are calculated to be 36 in^2 (232 cm^2) and 6 in (13 cm), then the random walk process model provides a means to estimate these values over incremental distances, e.g. the variance after traveling only 3 ft (1 m). By the random walk process assumption, the incremental variance for a given displacement is evaluated to be $K_{dd} = 36\,in^2 / 50\,ft = 0.72\, {in^2}\big/{ft} \left(15.24\, {cm^2}\big/{m}\right)$. Therefore, if the testbed vehicle travels only 1.0 ft (0.3 m), then the variance over this incremental distance is evaluated to be $0.72\, {in^2}\big/{ft} \cdot 1\,ft = 0.72\,in^2$, which yields a standard deviation of 0.85 in (2.2 cm). Therefore, for each foot (0.3m) that the testbed travels it is expected that the variance of the cumulative displacement will increase by $0.72\,in^2$. The coefficients, K_{dd} , $K_{d\theta}$ and $K_{\theta\theta}$ are classically determined in a similar experimental fashion.

The exact values for the coefficients of the selected model (K_{dd} , $K_{d\theta}$ and $K_{\theta\theta}$) were experimentally determined by running a series of tests with the developed platform. These tests were conducted on a section of rough grassy terrain (similar to what would be expected for the roadside). By comparing the variation in displacement and heading (for many tests) over a 50.0 ft (15.2 m) path, the values for K_{dd} and $K_{d\theta}$ were evaluated to be $0.034\,in^2/ft$ and $0.16\deg^2/ft$ respectively. In a similar fashion, the testbed was rotated in place to determine the variance in angular displacement with respect to rotation, and the value for $K_{\theta\theta}$ was determined to be about $0.01\deg^2/\deg$. As recommended by Chenavier and Crowley, these values were later increased to more conservative values so that the system would be tolerant to unusual effects, for example, a large unexpected wheel slip due to a wet grassy spot. The coefficient K_{dd} was also increased to account for absolute positioning uncertainty due to a 2D planar approximation of the terrain, as discussed in Section 4.2.3. In addition to applying more conservative coefficients, the values were manually tuned by experimentation to yield an acceptable filtering performance. The final values for K_{dd} , $K_{d\theta}$ and $K_{\theta\theta}$ were selected to be $0.5\,in^2/ft$, $0.18\deg^2/ft$ and $0.136\deg^2/\deg$ respectively.

The input noise covariance (Γ_{k+1}) is used to express the uncertainty in the system inputs. For the platform under consideration, this relates to uncertainties in the encoder readings, since these values are used to drive the state prediction. Uncertainty in the encoder readings may arise from two distinct sources, the inherent discreet nature of the encoders, and additional backlash present in the motor and gearbox assembly. From experimentation, it was determined that the overall contribution of the input uncertainty to the state error covariance (eqn. 4.4) was insignificant in comparison to the uncertainties which arise from the process noise. Regardless, to be complete, these effects were included in the model and it was assumed that the encoders were uncorrelated with an uncertainty of one encoder tick per reading.

The measurement noise covariance matrix (R_{k+1}) represents the uncertainty in the measurement readings. For the testbed, measurements for all three state variables (x, y, θ) are provided by an onboard Crescent Vector GPS board (Section 4.2.1). To quantify the covariance matrix, the initial approach was to use published uncertainty data for both positioning and heading, however, the actual performance is dependent on a number of factors (Section 2.1.1), and therefore this approach was later replaced. Instead, the internal positioning statistics from the GPS board were recorded for each measurement and applied to the corresponding values of the measurement covariance matrix. Although internal positioning statistics were available for positioning, heading information was not, and instead static testing performance data, which is included in Appendix C, was used. It was determined that over a 50 minute period of time, the standard deviation of the heading was on the order of 0.15 deg (0.0026 rad), which is quite similar to the published data [57]. After experimentation, however, it was determined that such a small uncertainty caused instability in the system due to measurement heading lag during quick changes in orientation. To counteract this instability, the heading uncertainty was increased to 0.75 deg (0.13 rad), which resulted in a good performance (Chapter 5). Future support (Chapter 6) should attempt to quantify this lag and to compensate for it in the digital control.

So far it has been assumed that the process, input, and measurement noise are zero mean, Gaussian, white, uncorrelated noise. In reality, this assumption is never perfectly true, as white noise is an idealization. Regardless, the EKF has been used with much success, even for systems with different noise characteristics. In fact, while a white noise approximation assumes that measurements are completely independent of time, GPS measurements tend to be heavily dependent on time and yet an EKF model has been used with much success for GPS aided localization. GPS measurements actually tend to have slow moving drifts that are dependent on the movement of the GPS orbital satellites. As detailed in Chapter 5, consistent mowing is more important than the overall absolute localization accuracy. Thus, these slow moving GPS positioning biases do not significantly affect the mowing performance. It is noted however, that true optimality exists only when the noise characteristics are met.

MEASUREMENT GATE: During normal operation, measurement uncertainties (caused by sensor noise) are accounted for by the measurement error covariance R_{k+1}. Although uncommon, erroneous measurements may occasionally arise, and since the Kalman filter model does not account for such erroneous measurements, they can seriously degrade the performance of the filter. To counteract such situations, the implementation of a measurement validation gate was necessary. The actual implementation is based on a technique proposed by [68] and used in [10]. Namely, the consistency of the GPS measurements are evaluated by using the normalized innovation squared (NIS)

$$\varepsilon_v(k+1) = v(k+1)^T S(k+1)^{-1} v(k+1) \tag{4.24}$$

where, $v(k+1)$ and $S(k+1)$ are the measurement residual and residual covariance calculated as

$$v(k+1) = z_{k+1} - h(\hat{x}_{k+1|k}, 0) \tag{4.25}$$

$$S(k+1) = \left(H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1}\right) \tag{4.26}$$

Given that the observation matrix is a linear 3x3 identity matrix (as noted earlier), eqn. 4.24 can be simplified to

$$\varepsilon_v(k+1) = \left(z_{k+1} - \hat{x}_{k+1|k}\right)^T \left(P_{k+1|k} + R_{k+1}\right)^{-1} \left(z_{k+1} - \hat{x}_{k+1|k}\right) \tag{4.27}$$

For a consistent filter, the NIS ($\varepsilon_v(k+1)$) has a chi-square distribution with the degrees of freedom equal to the dimension of the measurement (3 for the Vector GPS).

To make use of the NIS, the upper and lower bounding values of the gate are taken from the chi-square table. If the NIS of a particular measurement (evaluated with eqn. 4.27) lies within the selected upper and lower bounding values, then the measurement passes the test and is included in the EKF update cycle. Otherwise, the measurement is discarded as erroneous. A larger bounding range accepts measurements more liberally, while a small range applies stricter acceptance criteria. While the selected chi-square bounding values play a key roll in defining bounds for acceptance, the acceptance is largely dependent on the system error covariance and measurement covariance ($P_{k+1|k}, R_{k+1}$). Thus, when the system error covariance ($P_{k+1|k}$) is large, meaning the filter is quite uncertain of its state estimate, then the gate (eqn. 4.27) will accept measurement errors more freely, that is, allow larger measurement residuals $\left(z_{k+1} - \hat{x}_{k+1|k}\right)$. Alternatively, if the filter is quite certain of its state estimate, then the system error covariance will be small, and large measurement errors are not tolerated. The same two points are true for the measurement covariance (R_{k+1}).

Initially a very small lower bound and large upper bound (yielding a large acceptance range) were selected to remove only very poor measurements. Later on, a 95% confidence interval for the upper and lower chi-square bounds was selected. Although these bounds worked just fine on average, a few instances occurred where the platform ignored correct measurements which caused the system to go way off track (These instances will be covered in Chapter 5). Thus, the final conclusion was to increase these bounds back towards the initial testing values. From early experimentation, it was demonstrated that occasional erroneous GPS heading values degraded the state estimation and cause the tracking to go unstable. Hence, the measurement gate (presented above) worked to remove these erroneous values.

TIME SYNCRONIZATION: The EKF is a sensor fusion technique that works in the time domain (not the frequency domain, like some other techniques). Therefore, inputs (u_{k+1}) and measurements (z_{k+1}) are incorporated into the state estimation based upon their corresponding time-stamp (absolute point in time). Given that the encoder sensor inputs and GPS measurements were not synchronized (encoder inputs were acquired at a rate of about 6 Hz, while Crescent Vector GPS measurements were obtained at a rate of 5 Hz), the EKF required a variable time step. For example, if a GPS measurement was acquired between (with respect to time) the last and current recorded encoder input, the time interval was divided up into two separate time intervals, the time period from the last encoder input till the time of the GPS measurement, and the time from the GPS measurement to the current encoder input. To achieve this process, it was assumed that the encoder counts were evenly distributed across the time interval, and hence could be discretized by utilizing the ratio between the time intervals.

A more detailed overview of the actual software implementation (in C#) for the EKF is covered in Appendix B.

4.4.2 Mapping and Path Planning

Mapping and path planning (Section 2.1.3) are other important topics which must be addressed to formulate a truly autonomous roadside mowing agent. Mapping deals with the primary task of developing an internal model of ones environment based on sensor percepts. While such an internal model may contain a variety of information, such as the locations of obstacles or a mapping of the terrain, the current sensor suite (Section 4.2) provides only localization information, thus the task of mapping is somewhat trivial. Namely, the history of the agent's motion is stored in memory by exploiting the predicted state estimate and error covariance, to aid in future decisions.

It was noted (Section 1.5) that additional sensor information required for true obstacle avoidance is necessary for future development. Such information is important because trees, shrubs, tires, and other obstacles are likely to hinder the movement of the agent. For roadside mowing, the agent must be able to properly identify, locate, and store these obstacles in memory for efficient circumvention. Clearly, such additional sensor information calls for a more complex internal mapping. Further insight will be provided in Chapter 6.

Path planning is the task of determining the motion from one configuration to the next (Section 2.1.3). For autonomous roadside mowing, this means formulating a suitable mowing path which will maximize the agent's performance given its percept history (mapping) and prior knowledge of the environment. At the beginning of this chapter, a set of performance measures were developed based on the customer's requirements. These measures evaluate the success of the proposed agent based upon its mowing rate, efficiency, and coverage effectiveness. To maximize these measures, a suitable mowing path had to be determined. To assist in this process, a brief look into the physical environment was required.

In Section 1.3.2 it was determined that large area mowing of medians and shoulders is an important area for improvement. Large, straight and moderately flat sections such as those displayed in Figure 4.12 are quite common throughout California. To maximize mowing performance in rectangular areas like these, the overall goal is to maintain a consistent mowing rate with modest overlapping of precut swaths. Much like a human must slow down when performing a turn, so must an autonomous mower. Thus, to maximize mowing rate, it is desirable to minimize the required number of turns over a given area. A few scenarios are possible, however, the ideal choice for GPS aided mowing (more details will be covered in Chapter 5) is parallel consecutive mowing swaths such as the zigzag path configuration displayed in Figure 4.13. Because the differential drive platform has a zero turn radius, this is a realistic configuration. This type of area coverage is known as "strip filling" [34] and has been shown to be optimal for a wide variety of applications. For the purpose of this report, only simple rectangular sections of vegetation are studied, but the extension to more irregular sections (such as curved medians, or triangular interchange sections) is somewhat trivial. In fact, Schworer extends this same mowing technique to a variety of irregular mowing sections, including sections cluttered with random obstacles, by using simple cell decomposition (Section 2.1.3) techniques [34].

For the "strip filling" technique displayed in Figure 4.13, an important matter for consideration is the overlapping of consecutive mowing swaths. Obviously it is desirable to minimize the required overlap, thus maximizing efficiency and increasing the mowing rate, but at the same time it is desirable to minimize the amount of uncut vegetation which may occur if the overlap is not large enough. For autonomous mowing, the amount by which an agent must overlap consecutive cutting swaths is dependent on a number of factors, including: the localization accuracy, the desired efficiency, and individual tolerance to uncut vegetation. Clearly, for manual mowing this overlap may only be on the order of a few inches, but for autonomous mowing, values may be substantially larger due to large positioning uncertainties. More details on the actual performance will be covered in Chapter 5.

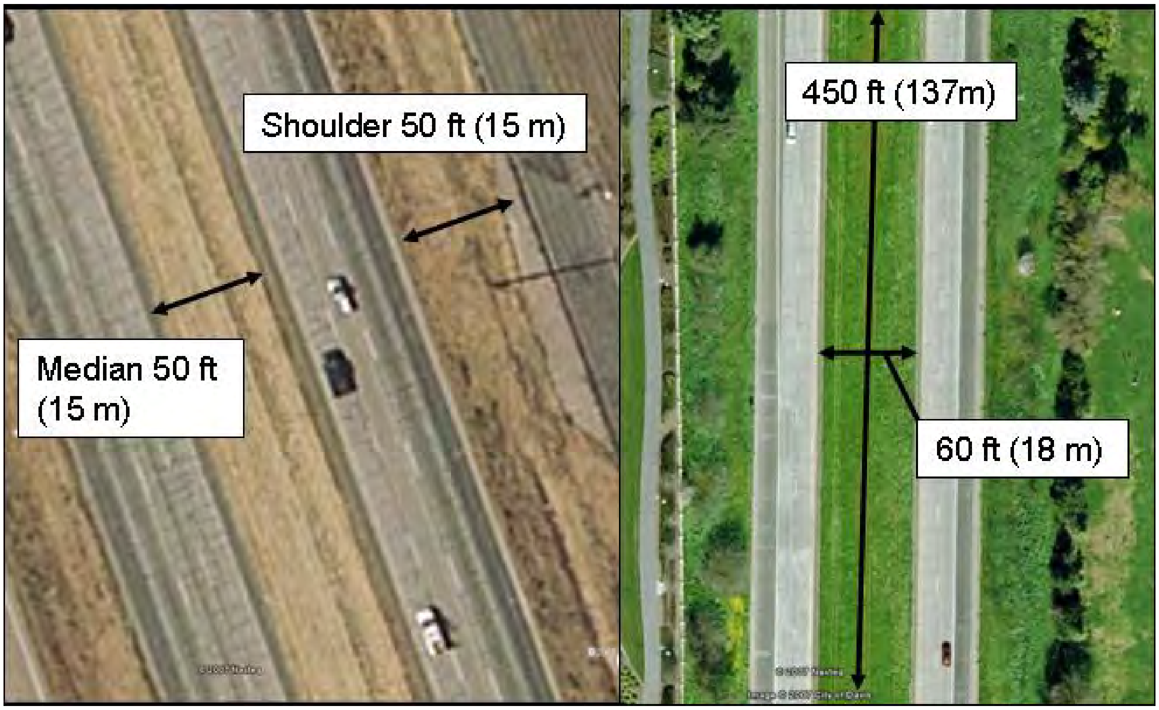


Figure 4.12 Section of Highway Rt-5 near Sacramento (Left), section of Highway Rt-113 in Davis (Pictures from Google Earth)

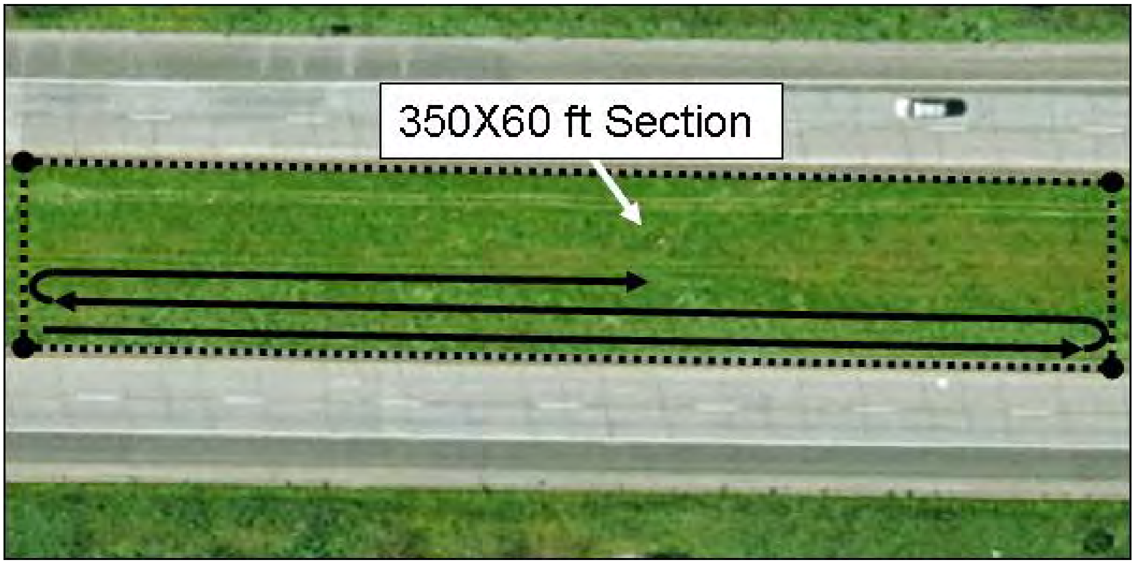


Figure 4.13 Anticipated mowing pattern overlaid on a section of HW-113 near Davis (Picture from Google Earth)

Unlike many other studies [10, 15, 29] that have researched localization techniques only as support for autonomous mowing technology, a fully realized platform is covered in this report. This means, not only developing an accurate localization system, but also developing an adequate tracking control and path planning implementation. Therefore, the actual tracking control plays an important roll in developing a suitable mowing path. The tracking control algorithm outlined in Section 3.4 was used to test a variety of tracking trajectories with varying control parameters. At this point, tracking was based only on odometry (from encoders), however it was anticipated that the final solution would have similar response characteristics (when GPS information was incorporated).

First, testing of the platform's ability to track a straight line path with a horizontal offset of 36 in (91 cm) was performed (Figure 4.14). As expected, the results show exponential tracking (Section 3.4) of the straight line with the convergence rate increasing with the value of the gain. Also, like many feedback control algorithms, the tracking becomes unstable when the gain is set too large. For the platform under discussion, it was initially decided that a value of 0.8 would be adequate for both k_x and k_y. It was later determined, however, that a smaller gain on the order of only about 0.2 for both k_x and k_y resulted in much better tracking stability and was used for the final configuration. The results shown in Figure 4.14 are based on a tracking speed of 1.5 mph (2.4 km/h) or equivalently 26.4 in/sec (61 cm/s) as was initially proposed in Section 3.2.1. In Chapter 6, insight into faster mowing speeds will be detailed.

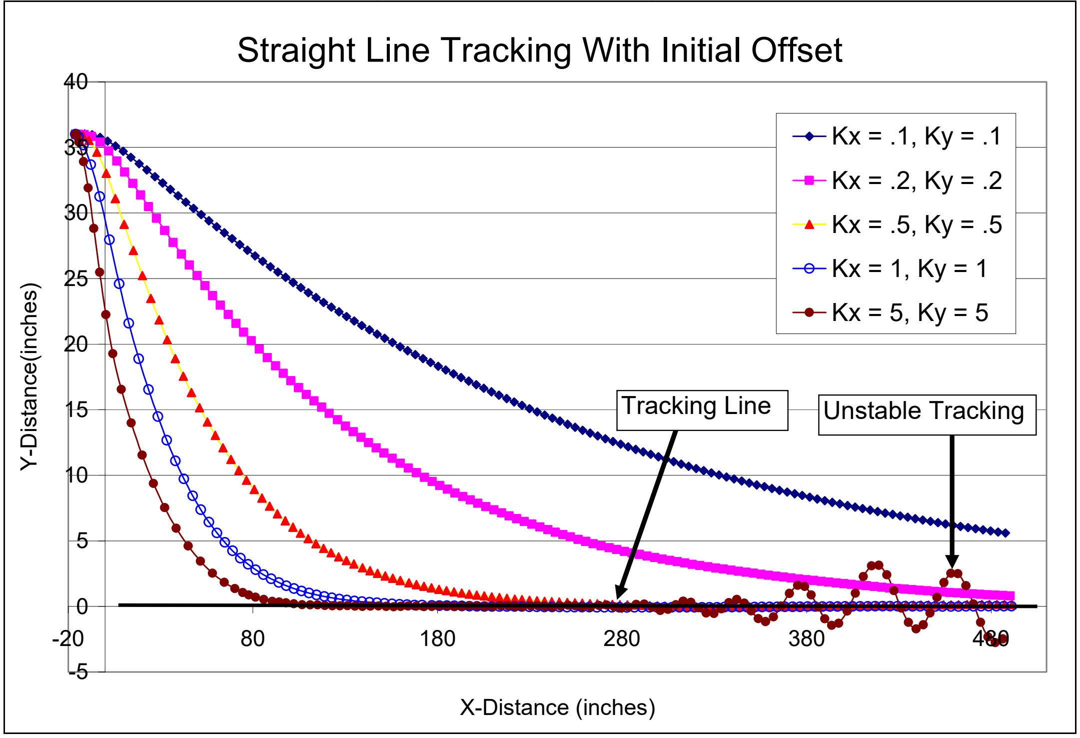


Figure 4.14 Tracking of a straight line trajectory with a horizontal offset of 36 in (91 cm)

Along with satisfactory tracking of a line, maneuvering a turn from one mowing swath to the next is also of particular importance. A few configurations are possible, including some form of arcing turn or straight line turn. Two distinct methods were tested, a straight line 180 degree turn and a 180 degree semicircle turn. To improve the control performance and to reduce dynamic uncertainties (such as wheel slippage) the speed was decreased from 26.4 in/sec (61 cm/s) to only 12.0 in/sec (30.5 cm/s) throughout each of the turns. It is quite obvious from the responses (Figure 4.15) that a semicircle turn yields more smooth and stable turning characteristics and was selected as the preferred choice.

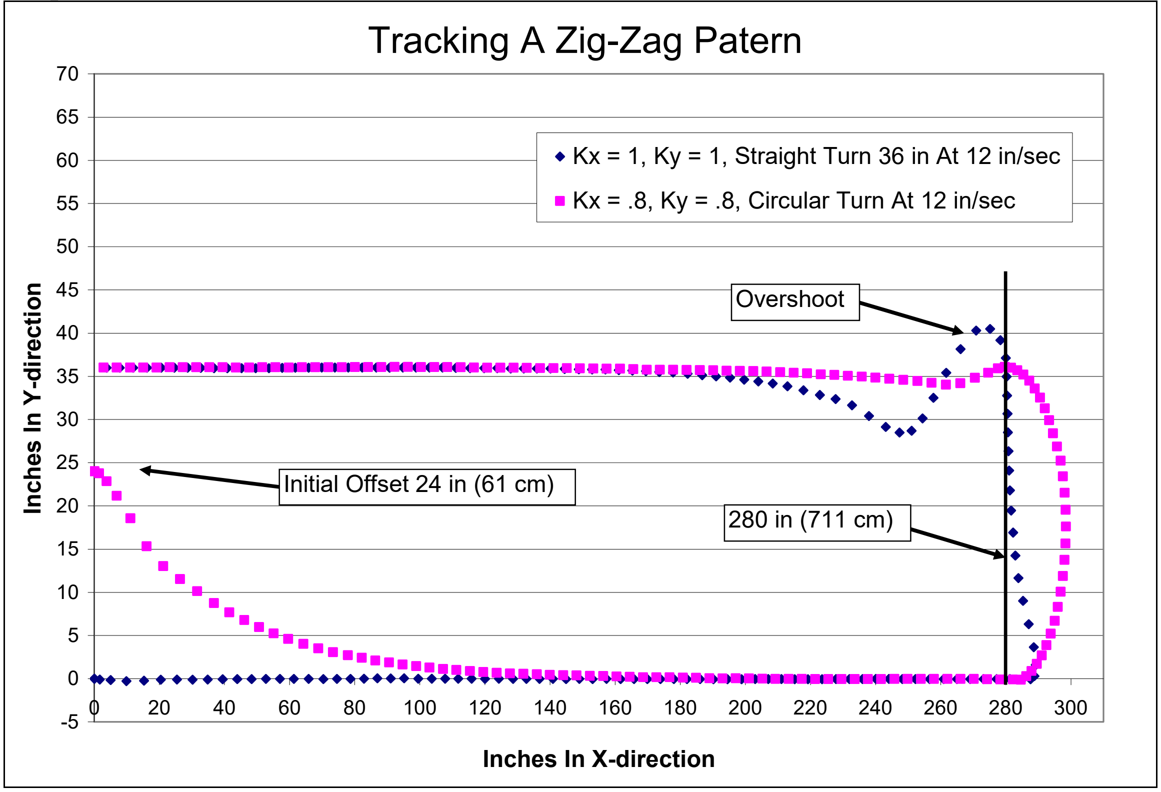
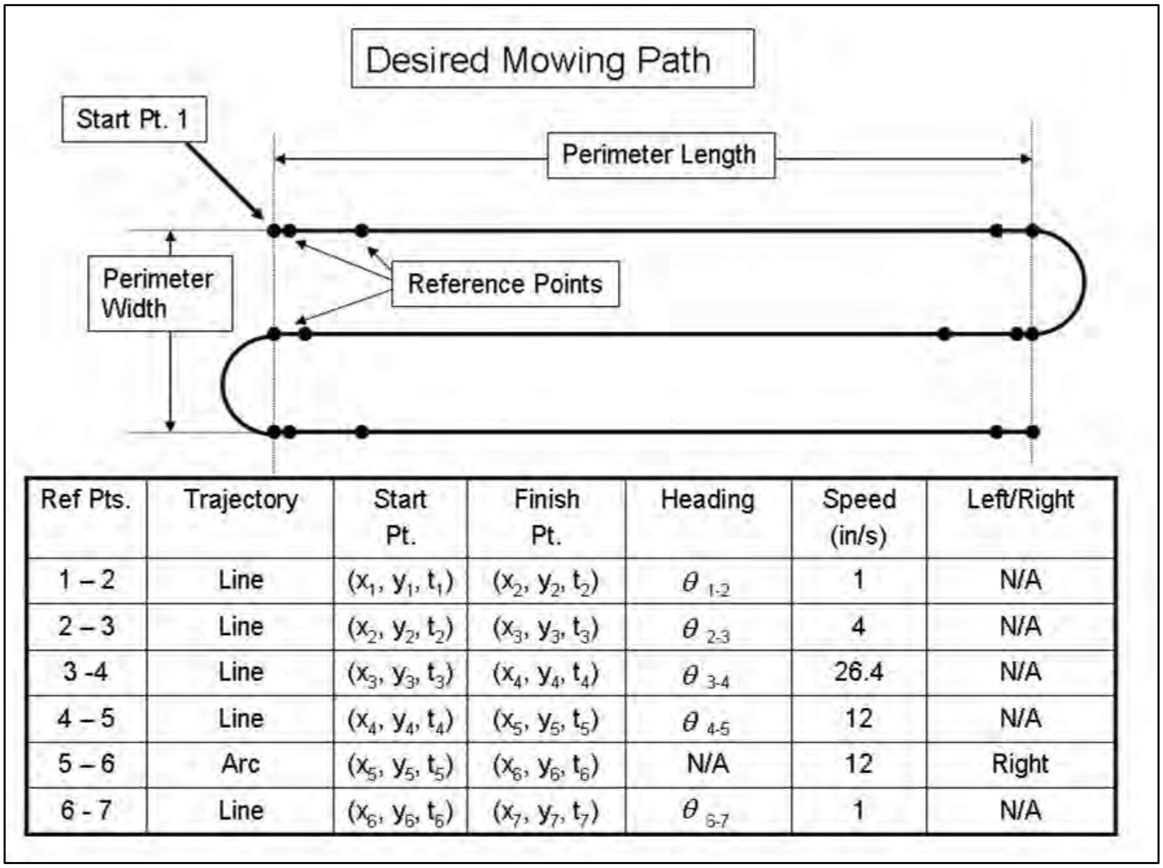


Figure 4.15 Tracking a zigzag pattern

To derive a complete (strip filling) reference path, a simplified online path planning algorithm is used to build the entire reference path, which is stored in memory as a series of line and arc segments, as displayed in Figure 4.16. Each distinct trajectory segment is correlated to an array of distinguishing parameters. For example, each segment is stored with a global starting and ending position, time, speed, heading, and a few other parameters. These parameters proved the control inputs necessary for real-time tracking control during mowing operation. The length between two consecutive turning maneuvers is actually defined as a series of three segments. These segments are defined as follows. Before each turning maneuver, the platform speed drops from 26.4 ft/s (61 cm/s) to 12.0 ft/s (30.5 cm/s) to reduce wheel slippage and other dynamic effects going into and during the turn. After each turning maneuver, the platform slows down to a speed of only 1.0 in/s (2.5 cm/s) this allows the heading to catch up as there is about a 1 to 2 second GPS heading lag during quick rotational changes. The speed then increases to 4.0 in/s (12.7 cm/s) this increases the stability of the tracking control algorithm by enabling the platform to get back on track before increasing to the straight line operational mowing speed of 26.4 ft/s. The three segments discussed here make up only a small portion of the entire length between consecutive turning maneuvers.



Ref Pts.	Trajectory	Start Pt.	Finish Pt.	Heading	Speed (in/s)	Left/Right
1 – 2	Line	(x_1, y_1, t_1)	(x_2, y_2, t_2)	$\theta_{1\text{-}2}$	1	N/A
2 – 3	Line	(x_2, y_2, t_2)	(x_3, y_3, t_3)	$\theta_{2\text{-}3}$	4	N/A
3 -4	Line	(x_3, y_3, t_3)	(x_4, y_4, t_4)	$\theta_{3\text{-}4}$	26.4	N/A
4 – 5	Line	(x_4, y_4, t_4)	(x_5, y_5, t_5)	$\theta_{4\text{-}5}$	12	N/A
5 – 6	Arc	(x_5, y_5, t_5)	(x_6, y_6, t_6)	N/A	12	Right
6 - 7	Line	(x_6, y_6, t_6)	(x_7, y_7, t_7)	$\theta_{6\text{-}7}$	1	N/A

Figure 4.16 Planned reference path stored in memory

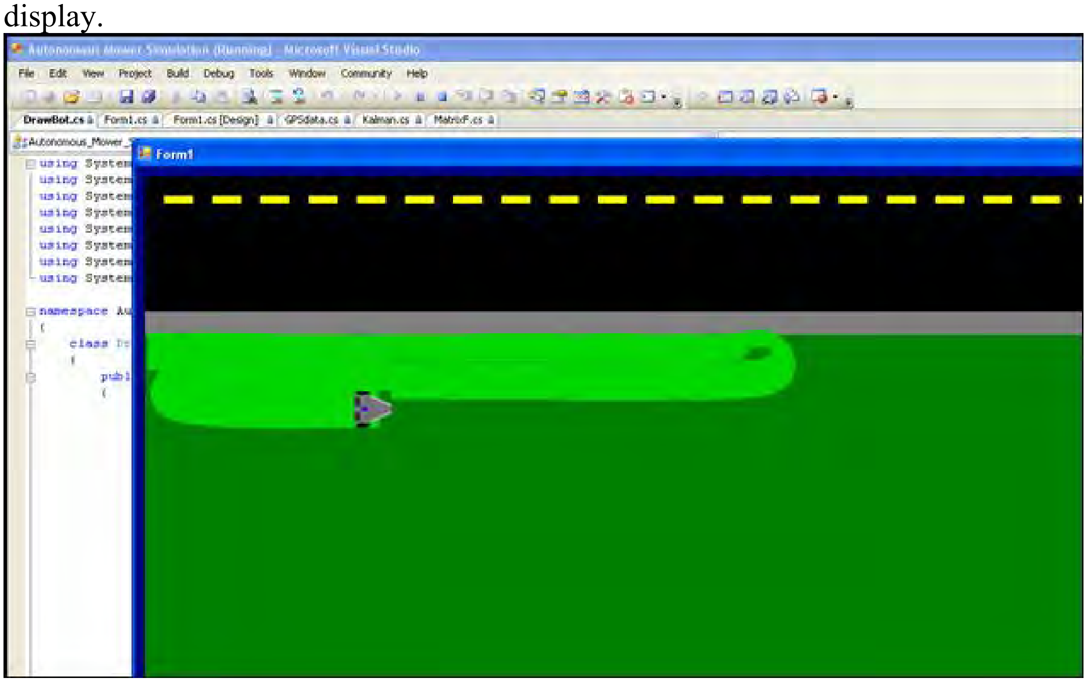
The current path planning algorithm makes use of a predefined rectangular mowing section (based on the length, width, and direction) to derive the corresponding mowing path. This technique was developed for ease of testing, however, a more appropriate and easily upgraded implementation is suggested for future development. Specifically, an algorithm that makes use of absolute GPS coordinates (waypoints). The actual path planning is performed only once, in software online during startup. A more suitable solution, is an adaptive algorithm that makes use of the stored percept history and the predefined knowledge of the environment to continually update the path. Although not actually implemented, insight into a more sophisticated adaptive path planning algorithm, to yield better mowing performance, is covered in Chapter 6. Also, a more complete solution must account for additional future sensors, for example, sensors required to provide obstacle avoidance information. Insight into future development will be provided in Chapter 6.

For additional information on software implementation of all path planning and mapping algorithms see Appendix B.

4.5 Simulation Implementation

Platform testing, parameter tuning, and software development are all time intensive procedures. To reduce such demanding tasks, a real-time replicating simulation was developed. This simulation made use of the same coding environment (C#) and almost identical code as the testbed software. In this fashion, coding algorithms were quickly developed, tested, tuned, and debugged. Then, when the desired performance was attained, the code could simply be ported over to the actual platform software with few modifications. Besides rapid software development, the simulation played a key role in the understanding and development of the EKF sensor fusion algorithm. For example, the localization performance with different sensor configurations, parameters, and accuracies could be tested.

Although the coding structure between the platform and simulation are almost identical, there are a few important differences worth mentioning. For one, the simulation is run on a computer system, not within a physical environment. Thus all sensors, actuators, and hardware had to be detailed in software in order to simulate the actual characteristics of the testbed. That being the case, driving classes were developed to simulate the motion of the vehicle, actuation of the motors, and measurements from the sensors. However, certain intrinsic factors were overlooked. For instance, motor actuation was assumed to be instantaneous (speed is achieved with no time delay), and dynamic vehicle effects were not accounted for (only kinematic motion). In addition, the system process, input, and measurement uncertainties, were modeled as normal, zero mean (unbiased), uncorrelated, white noise. As noted in Section 2.1.2, these are the conditions necessary for EKF optimality, however, rarely do they exist in real life. Therefore, the simulation was somewhat ideal overall. Regardless, the simulation permitted rapid and successful software development for the testbed. Figure 4.17 displays a screenshot of the simulation during execution and its graphical real-time display.



```
using System
using System
using System
using System
using System
using System
using System
using System

namespace Au
{
    class D
    {
        publ
        {
```

Figure 4.17 Simulation screenshot of the graphical display

4.6 Chapter Summary

At the beginning of this chapter a set of performance measures were developed in order to evaluate the success of the final mowing solution. These measures were derived based upon the customer requirements listed in Section 3.1. Specifically, it was determined that a successful autonomous roadside mowing agent should have a quick mowing rate, mow efficiently, and mow effectively. Next, a sensor configuration to meet these performance measures was selected.

Many sensor configurations for accurate localization have been researched over the years. Some have made use of advanced and extensive GPS units, and others have proposed using an extensive network of visual or active landmarks. Neither of these solutions is principally useful to Caltrans for autonomous roadside mowing. Instead, a relatively low cost GPS unit which provides both accurate heading and positioning has been selected, along with inexpensive odometry encoders. It was noted that a GPS solution is particularly well suited for the roadside environment, due to low signal interference, which is a characteristic of roadside median and shoulder sections. The details of these units, along with proper mounding and electrical configuration were covered in Section 4.2.1 and 4.2.2.

To fuse information from the encoders and GPS unit, an Extended Kalman filter (EKF) was used. For the testbed under consideration, the EKF works in a predictor-corrector style as follows. When encoder readings are obtained, the EKF updates the state estimate and error covariance based upon the kinematic equations of motion and knowledge of the input and process noise. Then, if a GPS measurement is attained at this time, the EKF fuses these measurements, in an efficient manner, to update the original state estimate and error covariance. The exact details were covered in Section 4.4.1, along with a supplementary measurement gate to preserve the consistency of the filter.

To enable the proposed agent to maximize its performance measures, the topics of mapping and path planning were covered, along with their actual implementation. Since the selected sensor configuration provides only localization information, the current mapping algorithm is trivial. That is, state and error covariance information are stored in memory (via the EKF output) for future decisions. The selected path planning technique is known as a “strip filling” algorithm, which is optimal for area coverage algorithms. For the agent to perform such a mowing pattern, more details into the actual implementation and performance of the tracking control algorithm (Section 3.4) were discussed. Based on these results a final path implementation that works online in software was detailed.

To conclude this chapter, the details of a supplementary software simulation (written in C#) were covered. This simulation was developed as a supporting tool, upon which code development, testing, and debugging could be done. Code from the simulation could then be ported over to the actual testbed with little modification. Along with software development, the simulation aided in the analysis and development of the EFK solution. In this manner, EKF tuning and the performance testing of different sensor configurations was carried forth.

CHAPTER 5 Platform Testing and Discussion

To evaluate the performance of the proposed agent, a series of outdoor tests were conducted. This chapter begins by discussing the testing setup, procedure, and additional considerations. Then, the results are presented followed by a detailed analysis of the performance and additional insight into ways in which the future performance may be enhanced. After this, a look into GIS software and its practical role in supporting autonomous mowing will be discussed, followed by a hypothetical look into the implementation of multiple agents into Caltrans practices. Finally, the benefits associated with autonomous mowing technology are explained.

5.1 Testing Setup

To evaluate the performance of the proposed agent, a series of outdoor tests were performed. The overall testing procedure was an iterative process aimed at debugging, tuning, and optimizing the performance of the agent. Since this process was a time consuming endeavor, only a limited number of tests were performed. With these tests, a sufficient amount of information was gathered upon which to tune the agent, evaluate the agent's final performance, and determine techniques and suggestions for improved performance and future development.

5.1.1 Final Testing Configuration

This chapter is focused on the analysis of the four final outdoor tests conducted, however, reference to earlier testing will be included when necessary. The four final tests were performed on a large open grassy field on campus, at the University of California, Davis (Figure 5.1). This field was quite uneven, which was attributed to numerous vehicle tracks and animal holes (Figure 5.1 and 5.2), making it an excellent site to test the robustness and performance of the agent. Such a harsh environment is analogous to what would typically be encountered along roadside medians and shoulders.

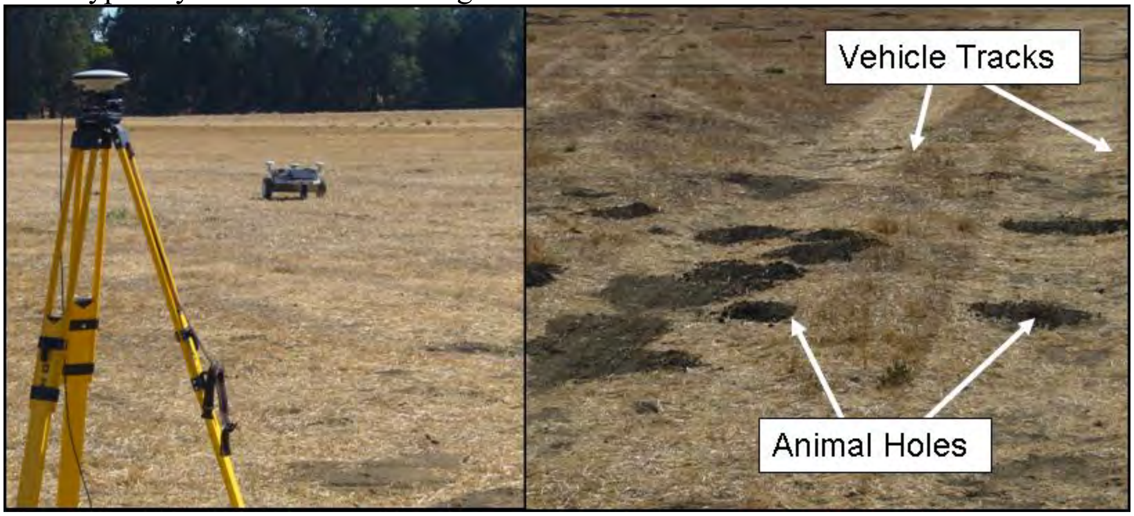


Figure 5.1 Final testing (left), vehicle tracks and animal holes (right)

Figure 5.2 Animal hole (left), platform traversing over an animal hole (right)

As discussed in Section 4.2.1, a second (in addition to the onboard Crescent Vector GPS unit) onboard NovAtel GPS board was used (Section 4.2.1) to provide ground truth (GT) information to assess the mowing performance. To improve the performance of the onboard NovAtel GT positioning information, a second stationary NovAtel GPS differential station (Figure 5.1) was set up to collect GPS information. After testing was completed, information from the onboard NovAtel and stationary NovAtel unit were post-processed using daily web based corrections to provide absolute positioning accuracies (1 sigma) on the order of 2 cm (0.8 in) or less, in both the North-South and East-West directions. This technique was used for all four of the final testing cases.

The four final tests operated over a predefined rectangular region that was 150 ft (46 m) long and 42 ft (13 m) wide, corresponding to a total area of 6300 ft^2 (585 m^2) or roughly 1/7th of an acre. To cover this area, a pre-planned path (to track) was formulated in software online at the start up of each test, and stored in memory as a series of straight line and arc segments. The details of this algorithm and the reference path composition were discussed in Section 4.4.2. The widths between consecutive mowing paths were arbitrarily selected to be 36 in (0.9 m) on center. Based on the tracking speeds outlined in Section 4.4.2, the total running time for each of the four tests was approximately 20 minutes. The average mowing speed was 1.5 mph (2.4 km/h) for all four tests.

All four tests were performed at different locations on the field in order to generalize the agent's performance. Also, to simplify performance analysis and to asses the directional variation in GPS performance, two of the four tests were conducted with paths parallel to the North-South direction, and two of the four tests were conducted with paths parallel to the East-West direction. In addition, two tests (one North-South and one East-West) were performed one day and the other two were conducted seven days later. The details of these tests and their results will be covered in Section 5.2. It is noted here, that mowing actuation was not actually accomplished during each of these tests, only path tracking.

5.1.2 Additional Considerations

Early testing brought forth various problems that had to be addressed before the final tests were conducted. For one, it was determined that heading measurements obtained from the Crescent Vector GPS unit contained a significant amount of lag during quick rotation (Section 4.4.1). This effect is detailed in Figure 5.3. In addition, the Crescent Vector GPS positioning measurements tended to diverge for a short period of time during turning maneuvers, however, this divergence only occurred when the primary GPS antenna (the one used to measure the GPS position) was located on the outside of the turn, since there is little change in position when the antenna is on the inside of the turn. An example of this effect is displayed in Figure 5.4. Details on how these problems were addressed is discussed next.

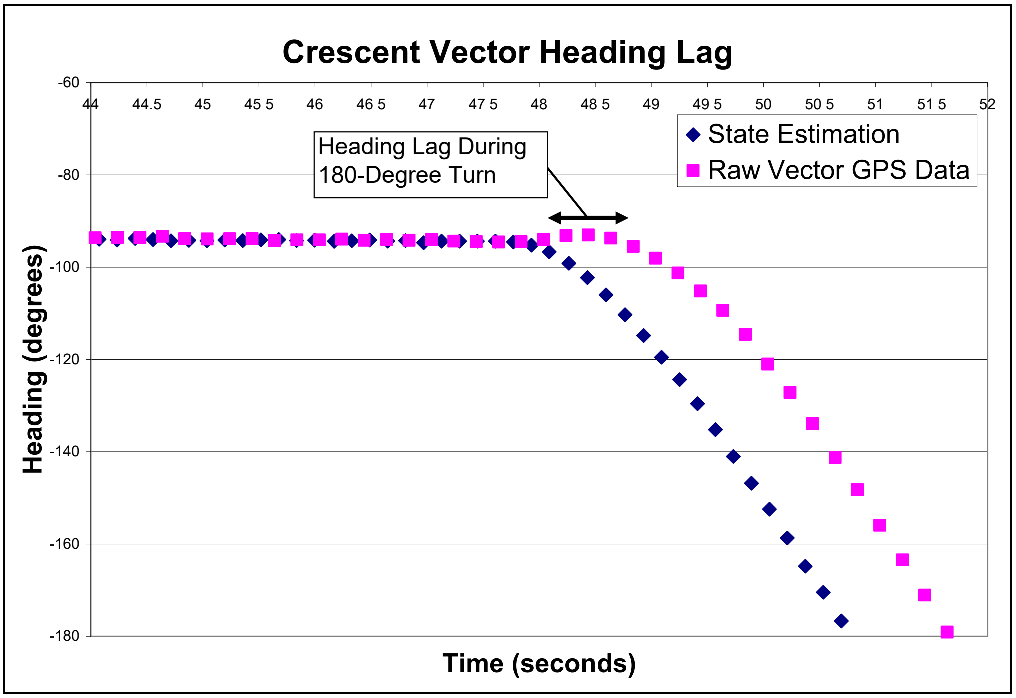


Figure 5.3 Crescent Vector GPS heading lag during a turn

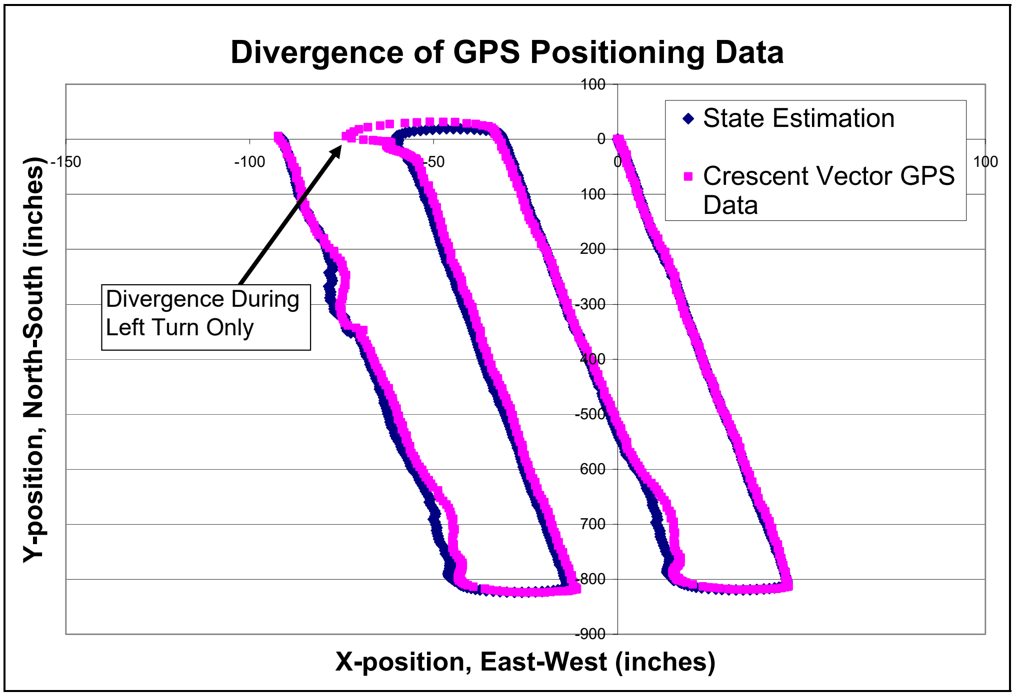


Figure 5.4 Crescent Vector GPS positioning divergence

To understand how these problems affected the performance of the agent and how they were handled it is important to comprehend the EKF fusion process. The EKF fusion process (detailed in Chapter 4) incorporates GPS measurements into the state estimation based upon the agent's current uncertainty and the accuracy of the measurements. Hence, the more uncertain the agent is of its current state, and/or the more accurate the measurements are, the more the measurements are trusted. Since odometry is a poor source for evaluating the orientation but a good way of measuring displacement, the fusion process relies heavily on the GPS heading measurements, but not so heavily on the positioning readings. Furthermore, since the heading values contained lag, the tracking control would tend to overcompensate for the lag and cause instability in the system; this is a common side effect for such time delays [69]. As an example (as noted in Section 4.4.1), when the uncertainty of the heading measurement was selected to be 0.15 degrees (as taken from a static heading test Appendix C) and the tracking control gains k_x and k_y were selected to be 0.8, the system would often spiral out of control during path tracking. On the other hand, the GPS positioning divergence had only a small impact on the performance of the system.

To address these problems a number of preventative steps were taken. First, the top speed of the vehicle was reduced during turning maneuvers (Section 4.4.2). This modification was enough to almost completely remove the positioning divergence associated with quick turns. Next, the measurement noise covariance matrix (R_k) was altered to be time variant based upon the following two rules. One, if the vehicle was making a turn, where upon the primary antenna was on the outside of the turn, the GPS positioning uncertainty was increased by a ratio of 1.5 (in the x and y direction) to account for possible positioning divergence. Second, if the vehicle was making a turn (right or left), the heading measurements were completely removed from the EKF fusion process because they contained a great deal of lag, even when the turning speed was reduced.

The next step was to reduce the tracking control gains, which smoothed the tracking control and added stability to the system (Section 4.4.2). Subsequently, the heading measurement uncertainty was increased to 0.75 degrees, causing the heading lag to have less impact on the state estimation. This also helped to reduce the instability caused by small dynamic heading variations from the uneven terrain. In fact, the calculated static heading uncertainty value of 0.15 degrees (Appendix C) is quite optimistic, and a value of 0.75 degrees is more reasonable for the highly dynamic environment. Finally, the vehicle speed was reduced for a short period after a turn was made to allow the heading to catch up and the vehicle to get back on track (Section 4.4.2).

The steps taken above were enough to yield a good performance even in the harsh terrain upon which the tests were conducted. Even so, an alternative solution and possibly a more effective solution would be to quantify the heading lag and compensate for this time delay in the outer digital control loop; e.g. put a time delay in the EKF fusion process.

Testing was an iterative procedure aimed at tuning and optimizing the performance, therefore, not all software parameters were the same for the final four tests. The first two tests made use of predefined GPS measurement error statistics based on published performance data. Namely, the x and y GPS positioning uncertainty was conservatively selected to be 19.7 in (~0.5 m). Conversely, the last two tests made use of positioning error statistics published by the Crescent Vector unit. These values varied based on a number of factors, including satellite availability and interference, but were typically on the order of 11.8 in (0.3 m). In addition, the first two tests made use of large (liberal) measurement gate bounding values, while the last two tests utilized a more strict 95% confidence interval (Section 4.4.1). Since the rules derived earlier stipulate that heading measurements should be discarded during turning maneuvers, there were a few instances where the 95% bounds were too strict and heading and positioning measurements were improperly discarded after making a turn. These few instances will be addressed later.

5.2 Final Testing Results and Analysis

This section is focused on a presentation of the final outdoor testing results and a thorough performance analysis of these results. Then in Section 5.2.3, additional discussion and suggestions for improved performance are covered.

5.2.1 Final Testing Results

To simplify the discussion of the agent's final outdoor tests, each of the final four tests will be referenced in numerical order (starting from the oldest and ending with the most recent). Figure 5.5 displays the path tracking results for the first of the four final outdoor tests. Plots for the last three tests (Test-2 to 4) are included in Appendix-C (Figure C.2 to C.4). Each of these figures display two distinct path plots, State Estimation and Ground Truth. The Ground Truth path is simply a plot of the recorded GT data, i.e. the high accuracy (recorded to be less than 2 cm) positioning information obtained from post processing the NovAtel DGPS data (Section 5.1). On the other hand, the State Estimation path is a plot of the agent's internal sensor mapping, formulated by fusing encoder measurements with the inexpensive Crescent Vector GPS (Section 4.2.1) positioning and heading information, by means of the EKF solution outlined in Chapter 4.

For visual comparison, the GT starting position was shifted to overlap with the starting State Estimation position (for Test-1 to 4). To be exact, the EKF solution outlined in Section 4.4.1 is initialized by making use of the global starting position (x and y) obtained from the low cost Crescent Vector GPS unit (Section 4.2). However, because the Crescent Vector GPS unit has a global horizontal positioning accuracy on the order of 0.6 m (24 in at 2 sigma), the state estimation was often initialized with an absolute positioning bias. Consequently, the starting state estimation did not coincide with the high accuracy (less than 2 cm at 1 sigma) absolute positioning GT obtained from the NovAtel data. More on this topic will be covered in Section 5.2.3.

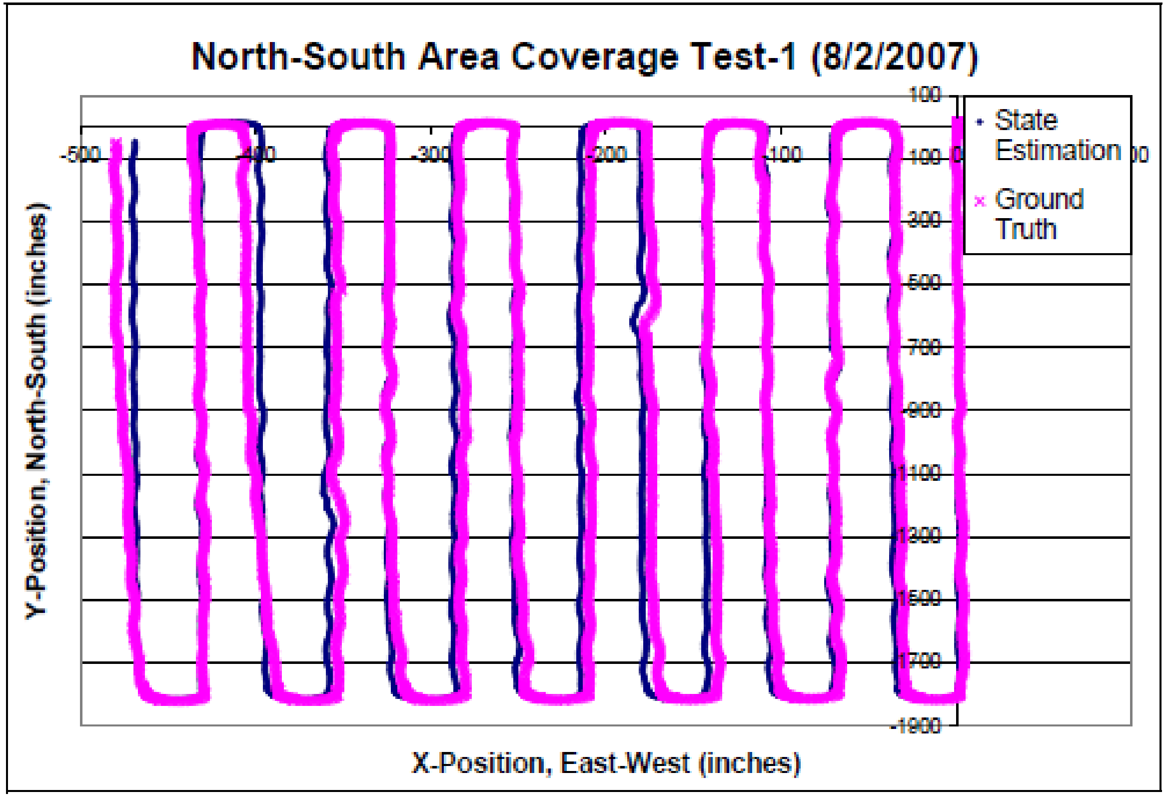


Figure 5.5 Path tracking results from Test-1

5.2.2 Performance Analysis

To evaluate the success of the proposed agent, three distinct performance measures were derived (Section 4.1). These include, a quick mowing rate, efficient area coverage, and effective area coverage. Although the mowing speed has an impact on the mowing rate (area coverage per unit time), it was restricted to 1.5 mph (2.4 km/h) for the purpose of this study. Therefore, the overall attainable performance of the vehicle is impacted solely by the required overlap of consecutive parallel mowing paths. As an example, if the agent is very uncertain of its continual position, then a large mowing overlap is required to ensure that all vegetation is mowed. In this manner, the effectiveness is maximized but the efficiency and mowing rate decrease, yielding a poor overall performance. Clearly, as the required path overlap declines, less energy is wasted targeting previously cut areas, yielding a higher level of efficiency, a quicker mowing rate, and improved overall performance. It is noted here, that since mowing speed does have a notable impact on the mowing performance, this topic will be covered in Chapter 6.

A few techniques for evaluating the required path overlap are discussed here. One method is to measure the absolute/global localization accuracy, i.e. the better the global positioning accuracy is, the more certain the agent is of its position and the smaller the required overlap becomes. This is the most common technique taken, however, most studies that have taken this approach [10, 15, 29, 31, 32] have either been directly interested in evaluating the global localization accuracy, or have focused only on developing a partial autonomous mowing solution. For example, in [10, 29] the authors were concerned with developing a novel sensor suite to support autonomous golf course turf management, but final testing was accomplished by manually (not autonomously) driving the testbed in predefined paths to measure the localization accuracy. While good absolute localization supports a higher level of performance, it does not guarantee optimal performance. The difference is that a truly autonomous mowing unit must not only be able to localize itself accurately, it must also be able to track defined paths precisely and to deal with uncertainties in the environmental, e.g. slipping on a steep slope, or loss of GPS signal (note GPS signal loss is mentioned in Chapter 6 for future testing).

Instead, the favored approach taken in this work was to evaluate the consistency of consecutive mowing swaths (parallel path lines) for the final tests (similar to the approach taken in [71]). By measuring the variability of the widths between consecutive mowing swaths, one is able to statistically predict the mowing effectiveness, efficiency, and rate, for a given overlap. This is a more fitting technique because it gives a direct evaluation of the actual mowing performance (including tracking control and other uncertainties), not just the localization accuracy. Furthermore, this evaluation provided clear insight into the expected roadside mowing performance, primarily because the harsh site selected is similar to that which would be anticipated along the roadside.

To evaluate the final performance, the high accuracy GT positioning data was used to measure the widths between consecutive mowing swaths for each of the final four tests (Test-1 to 4). However, only GT data along the straight-line mowing sections of the path was used, since the turns between consecutive swaths were not part of the required mowing area (Section 4.4.2). The results for Test-1 are displayed in Figure 5.3, while plots for the last three tests are included in Appendix-C (Figure C.5 to C.7). As noted earlier, because there were two distinct instances whereupon the agent deviated from the tracking path, (explicitly, one time during Test-3 and one time during Test-4) mowing widths during these sections (Figure C.3 and C.4) were neglected in the consistency width plots. These instances will be addressed further in Section 5.2.3.

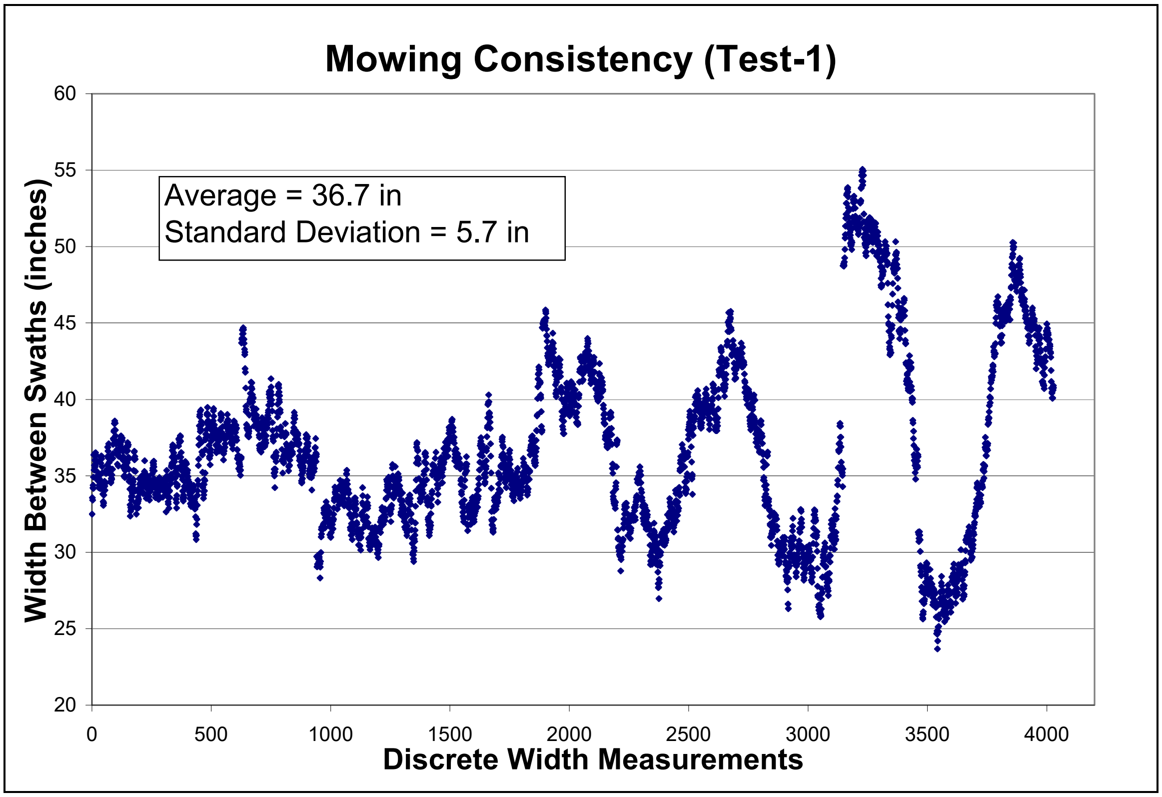


Figure 5.6 Plot of the width between consecutive mowing swaths for Test-1

For Test-1 (Figure 5.6), the average width between consecutive mowing swaths was evaluated to be 36.7 in (0.9 m), which is nearly equal to the predefined path width of 36.0 in (0.9 m, Section 5.1). However, Figure 5.6 demonstrates that there is a significant amount of variability about the desired path width. As mentioned above, this is caused by factors such as: localization uncertainty, tracking control precision, and environmental uncertainties. To analyze the mowing effectiveness, efficiency, and rate, a histogram of all the discreet measured widths for Test-1 was formulated, as displayed in Figure 5.7. Also displayed in Figure 5.7, is a normal/Gaussian probability distribution with a mean and standard deviation equal to the values calculated for Test-1 (Figure 5.6). The histogram data in Figure 5.7 lines up well with the normal probability distribution, validating the use of the normal probability distribution for statistically estimating the mowing performance (for Test-1). Histogram plots for the remaining three tests yield similar results.

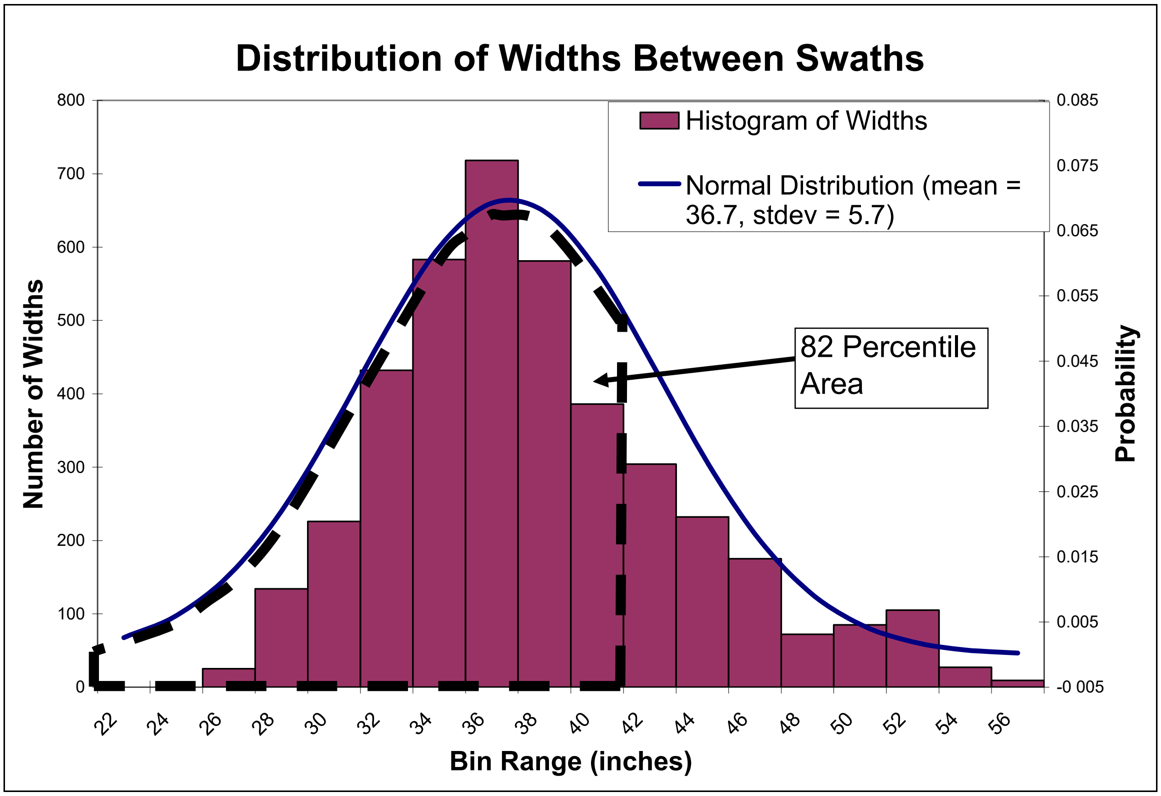


Figure 5.7 Histogram of the width between mowing swaths for Test-1

A hypothetical performance analysis (based on the data for Test-1) is covered here. The purpose is to detail the effect that the mowing width consistency has on the agent's performance. Assume that the proposed agent has a 42 in (1.1 m) cutting width. Furthermore, suppose that the width between consecutive mowing swaths is selected to be 36 in (.9 m), resulting in a 6 in (15 cm) mowing overlap. Based on the statistics for the normal distribution curve detailed in Figure 5.7, there is about an 82% probability that the width between consecutive mowing swaths will remain less than 42 in (1.1 m). Correspondingly, this suggests that during 18% of the traveled mowing distance, the width will be greater than 42 in (1.1 m), in which case the agent will be missing some of the vegetation.

To determine the mowing effectiveness, the percentage of the entire mowing area that is left uncut needs to be determined. Based on the values plotted in Figure 5.6, the average width for all the width values greater than 42 in (1.1 m) was 46.3 in (1.2 m). Therefore, on average, during 18% of the agent's traveled distance, a 4.3 (11.9 cm) strip of vegetation will be left uncut. This is equivalent to a mowing effectiveness of $(36\,in - 4.30\,in \cdot 0.18)/36\,in = 97.85\%$. In other words, for every acre (43,560 ft^2, 4000 m^2) of vegetation that is mowed, about 940 ft^2 (87 m^2) of uncut vegetation will be left behind.

Additionally, the mowing efficiency is calculated by simply determining the ratio between the effective cutting width (accounting for the overlap) and the mowing width capacity, i.e. 36.0 in/42.0 in = 86.0%. Finally, the mowing rate is equivalent to the effective mowing width times the speed, or 36 in * 26.4 in/s = 6.6 ft^2/s. This rate is slightly optimistic because the platform must slow down during turning maneuvers, and also, future obstacle avoidance is likely to reduce the efficiency to some extent. In summary, the results from Test-1 suggest that if the proposed agent has a 42 in (1.1 m) mowing width capacity and the mowing width overlap is selected to be 6 in (15 cm), the attainable effectiveness, efficiency, and mowing rate, are 97.9%, 86.0%, and 6.6 ft^2/s respectively.

From this analysis, a few observations are clear. For one, to improve the effectiveness, one must increase the mowing overlap. However, by increasing the overlap, the efficiency and likewise the mowing rate decline. Therefore, the only way to improve the effectiveness, while maintaining the same level of efficiency and rate, is to improve the consistency of the widths between consecutive mowing swaths. Another important observation is that since effectiveness and efficiency tend to work against each other, the optimal balance between the two is largely dependent on the customer's personal tolerance to uncut vegetation. Hence, if Caltrans simply will not tolerate uncut vegetation, then the optimal performance will weigh the mowing effectiveness with more impact than the efficiency. With these observations in mind, it is not the intent of this report to determine what the exact balance should be, but rather to determine what the attainable mowing width consistency is, given the proposed design (sensors, hardware, and software). It is also the intent of this report to evaluate the practicality and the potential value of the proposed agent to Caltrans.

A number of questions are still yet to be answered. For example: why is the variability of the widths between consecutive mowing swaths different for each of the final four tests and why does the width consistency tend to diverge towards the end of Test-1? These are topics which are addressed in the next section.

5.2.3 Additional Discussion

A deeper analysis into the results of the final outdoor tests brings forth a number of interesting observations. These observations are intended to support future development. In particular, a number of key observations detail how the agent's performance might be significantly improved.

Early platform testing (before the final four tests) was performed outdoors, on smooth flat pavement. The results for one such test are presented in Figure 5.8 (the results from another such test are displayed in Figure 5.3). One can notice from this figure, that unlike the final four test results (Figure 5.5 and Figure C.2 to C.4), the mowing paths are almost perfectly straight. This is attributed to the successful integration of the high accuracy heading measurements, provided by the Crescent Vector GPS unit, into the state estimation (as determined though simulation testing). Furthermore, an analysis of the consistency of the width between consecutive mowing swaths (Figure C.8) indicated an average width of 30.0 in (.8 m), which is exactly the predefined width selected for this trial. It also showed a standard deviation of only 3.6 in (9 cm), which is much better than the results for the final four tests, and quite good considering the positioning accuracy of the Crescent Vector GPS units is on the order of only 0.3 m (12 in, 1-sigma).

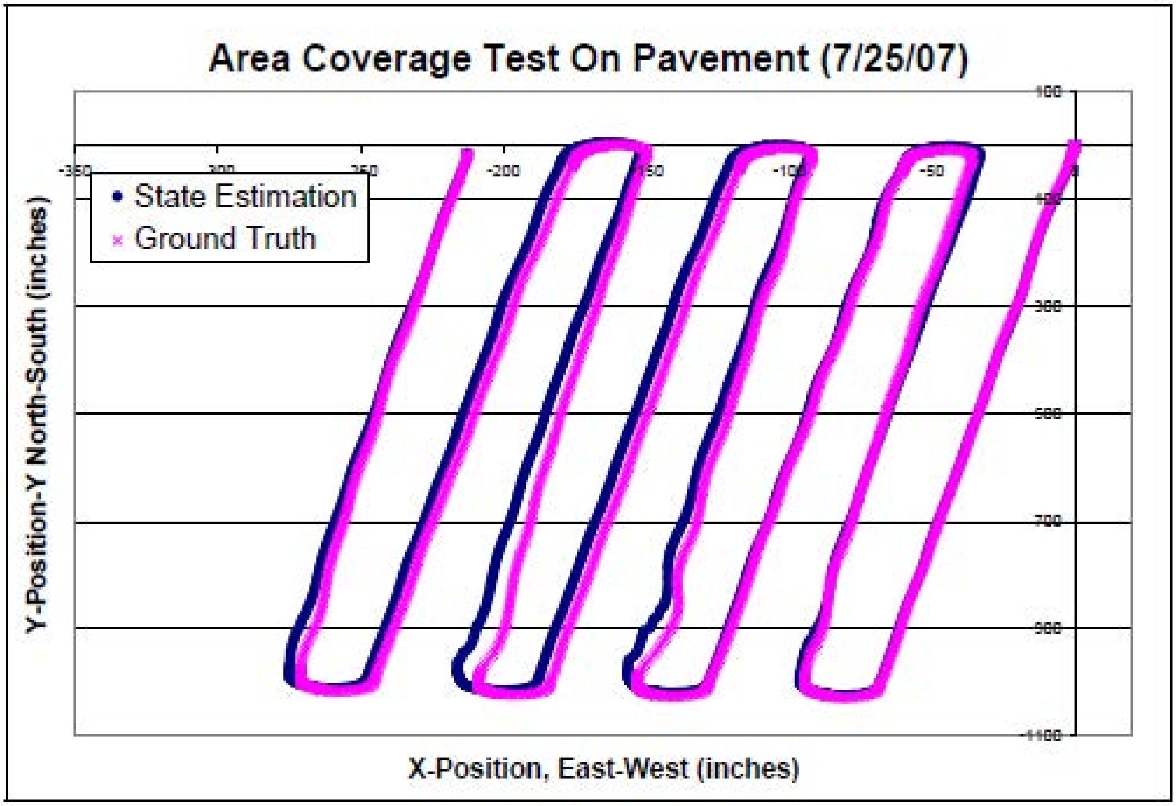


Figure 5.8 Path tracking result for a test performed on pavement

These results indicate a clear discrepancy between trials run on the pavement and ones run on the grassy terrain (Section 5.1). The harsh grassy terrain (upon which the final tests were run) had several negative impacts on the system which were not present for the pavement trials. First, since the terrain was uneven and rocky, the platform experienced significant loading differences between the left and right motors. Furthermore, since the underlying PID motor control (Appendix-A) was not able to instantaneously account for these variations, they were likely to cause unequal motor actuation and hence affect the straight-line tracking ability. Another impacting factor (which was discussed in Section 4.2.3) is the antenna offset errors present during uneven travel. Although these values were determined to be relatively small (on the order of a few inches), they were not accounted for in either the Crescent Vector GPS measurements or the high accuracy GT data. Therefore, the mowing consistency calculations covered in this chapter are slightly conservative (the performance is slightly better than indicated) because the antenna error is not compensated for in the GT data. Finally, small abrupt dynamic changes in the vehicle's attitude, e.g. when one wheel is driving over a dirt mound, were likely to decrease the performance of the absolute heading provided by the GPS board, especially since the heading measurements were determined to have a limited response rate (contain lag). Any combination of these factors was likely to cause the waviness apparent in the path tracking results for Test-1-4.

In light of many of the factors mentioned above, the consistency plots for the final tests (Figure 5.6 and Figure C.4 – C.6) indicate large periods of low variability and short periods of high variability. For example, the results for Test-1 (Figure 5.6) indicate that up until about 2/3rd into the test run, the variation in the width is quite small. In fact, the standard deviation of the widths for the first ten mowing swaths is only 3.5 in (8.9 cm), which is much better than the value calculated for the entire test (5.7 in, 14.5 cm). To understand why this is the case, the path tracking results for Test-1 are plotted in Figure 5.9 along with the raw Crescent Vector GPS positioning data. This plot indicates that the state estimation and GPS positioning data line up very well during the first ten mowing swaths of the test, while the two plots tend to diverge a few times during the last four mowing swaths. This divergence was possible because GPS positioning measurements for this test were only fused into the state estimation data (using the EKF) with an accuracy of 0.5 m (19.7 in), meaning they were not trusted heavily. This observation suggests that while it is partially a coincidence that the GPS data lines up well during only the first ten mowing swaths, the performance is much better when the Crescent GPS data is trusted much more.

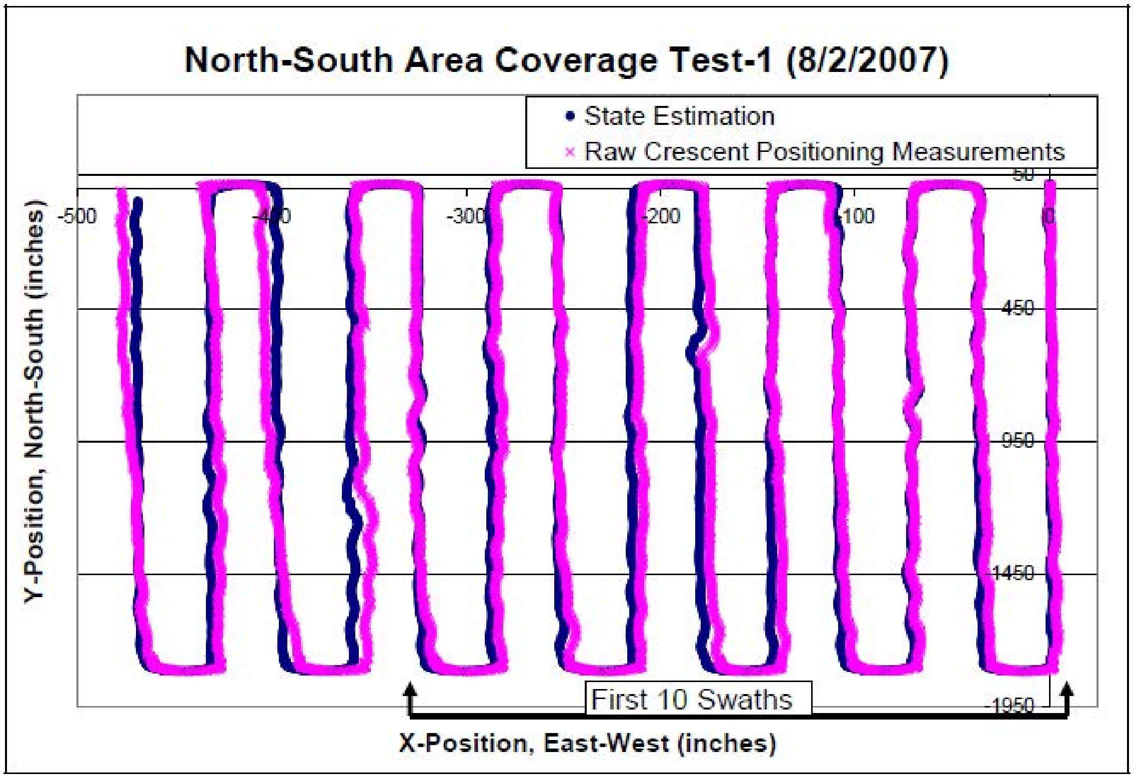


Figure 5.9 Path tracking results for Test-1 with raw Crescent Vector GPS data included

The results from Test-4 support a similar position. One can notice that the width consistency during the first half of Test-4 (Figure C.7) is very poor, but excellent during the second half. The standard deviation of the widths for the last six mowing swaths is 3.7 in (9.6 cm), which is much better than the value for the entire test (7.6 in, 19.3 cm). In a similar fashion, the path tracking results for Test-4 is displayed in Figure 5.10, along with the raw Crescent GPS positioning data. Much like Test-1, the state estimation lines up very well with the GPS positioning data during the consistent portion of the test (the last six mowing swaths), and not so well during the highly inconsistent portion of the test. Unlike Test-1, the reason that the state estimation and GPS data line up well during the last half of the Test-4 can be explained by a few derived observations, discussed next.

As mentioned earlier in this chapter, Test-4 represents one instance whereupon the measurement gating bounds were selected to be too strict. Therefore, after making the 6th turn, the GPS measurements were incorrectly neglected. What is interesting, however, is that after relying only on the encoder measurements for a short period of time, the state estimation uncertainty became so large that the GPS measurements were once again accepted through the measurement gate and the solution converged. As a result of this divergence, the state estimation uncertainty was quite large and the GPS measurements were fused into the EKF solution with much more importance during the second half of the test. In addition, it was found that during the sixth mowing swath, the internal GPS positioning statistics changed from an estimated North-South positioning uncertainty of 0.3 m (11.8 in) to a value of 0.2 m (7.9 in). Furthermore, since Test-3 and 4 made use of these internal GPS statistics for the EKF measurement covariance matrix, the GPS values were trusted more during the latter half of Test-4 (after the sixth mowing swath).

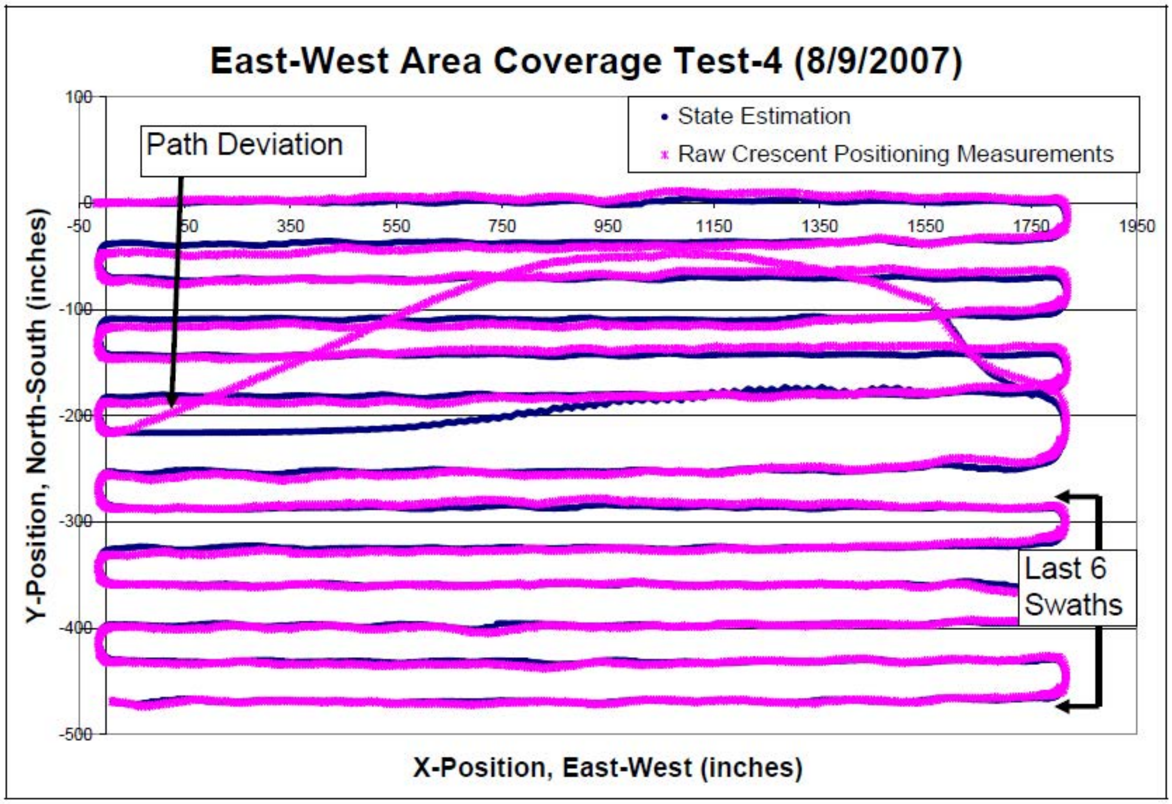


Figure 5.10 Path tracking results for Test-4 with raw Crescent Vector GPS data included

Regardless of the exact means by which the Crescent GPS positioning data and state estimation line up so well, only during the consistent portions of the final tests, the results clearly indicate that an improvement in performance is achievable by trusting the Crescent GPS measurements more. To support this analysis, a deeper look into the characteristics of the lower accuracy Crescent Vector GPS unit was conducted. Figure 5.11 displays a plot of the GT positioning information (obtained from the NovAtel GPS unit) in comparison to the lower accuracy Crescent Vector GPS positioning data, for the last six mowing swaths of Test-4. For visual comparison, the positioning bias between the two plots was removed at the start of the eighth mowing swath, as indicated in the figure. Figure 5.11 displays a few important results. First, besides a small bias, the two dynamic positioning plots are almost identical, which suggests why the performance was much better when the Crescent GPS data was trusted heavily. Second, the difference between the two plots is a slow moving bias/drift.

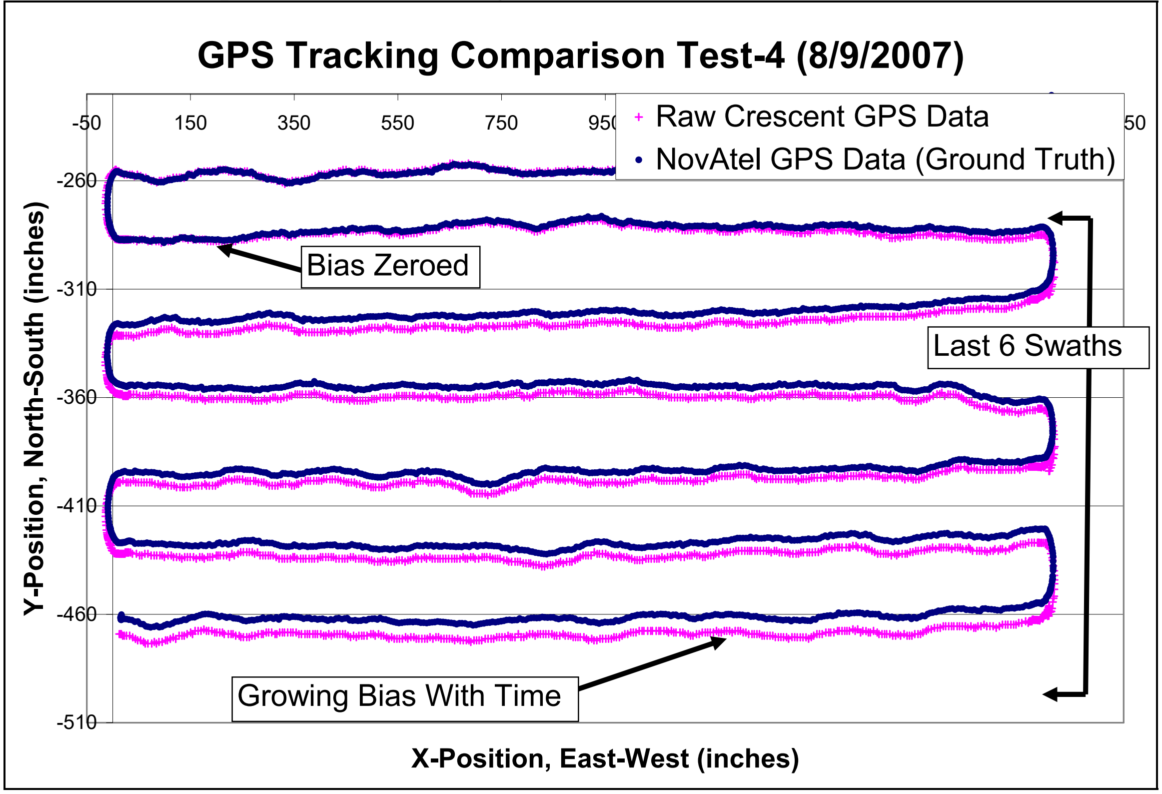


Figure 5.11 GPS positioning comparison for part of Test-4

The slow moving drift observed in Figure 5.11 is a common characteristic of GPS measurements, which is caused by daily satellite movement. To further interpret this drifting effect, the absolute positioning error of the Crescent Vector GPS data for Test-4 was plotted (Figure 5.12). This was accomplished by subtracting the absolute position (provided by the high accuracy GT data) from the raw Crescent GPS positioning data. The results in Figure 5.12 expose the positioning drift first noticed in Figure 5.11, i.e. the absolute error during initialization (the start of the run) is relatively 14 in (36 cm) and 2 in (5 cm) in the North and West directions respectively, but gradually drifts over the 20 minute test to a final value of about 5 in (13 cm) and 4 in (10 cm) in the North and West directions. One can notice that while this drift can be significant over a long period of time, the drift is quite small over a period of a few minutes or less. Furthermore, the variability in position over a short period of time is limited to the inherent noise characteristics of the GPS measurements. The same results were found for each of the final four tests, a similar plot for Test-2 is included in Appendix-C (Figure C.9).

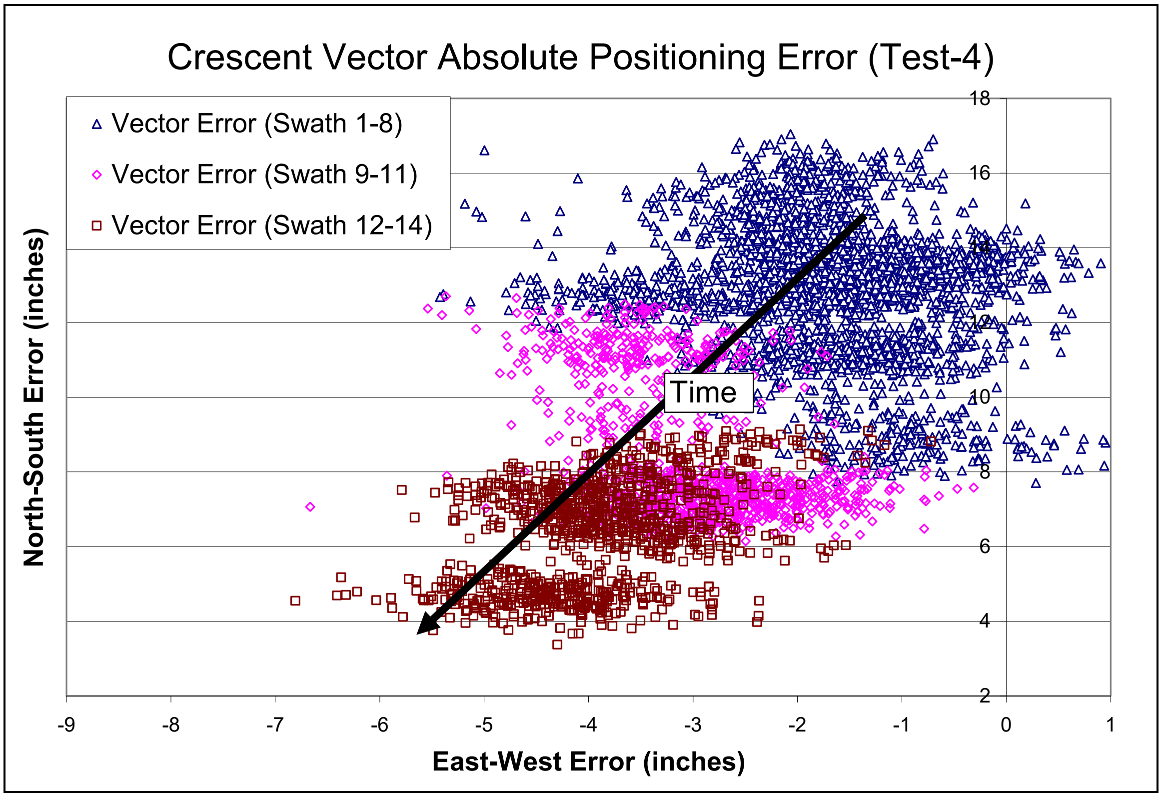


Figure 5.12 Crescent Vector absolute dynamic positioning error

The EKF model discussed in Section 4.4.1 alluded to the fact that GPS errors are heavily dependent on time. Therefore, while the EKF solution has the ability to effectively remove the short term (high frequency) inherent GPS noise, it is not capable of accounting for the long-term drift [31]. Some studies have recognized this deficiency [31, 65] and attempted to compensate for this drift. Three distinct approaches have been pursued: accept and ignore these biases and the impact they have on the absolute localization, attempt to quantify the drift by making use of more advanced models that take into account such things as the number of satellites in view, or finally, incorporate additional absolute measurements which attempt to compensate for these biases. For the purpose of this study a variation of the first solution has been proposed, as described next.

A deeper analysis of Figure 5.11 (and similarly Figure 5.12) indicates that the GPS positioning drift after two consecutive parallel mowing swaths is on the order of at most a few inches (~0-3 in or 0-8 cm). Therefore, the drift has only a small impact on the consistency of the width between consecutive mowing swaths, and other dominating factors inherently require a path overlap larger than this value. This suggests that the slow moving GPS drift can be ignored, assuming that the time required to perform two consecutive mowing swaths is limited to several minutes or less. Earlier it was found that, excluding drift, the lower accuracy Crescent GPS data lined up very well with the high accuracy GT data (Figure 5.11), similarly, the agent's performance was improved when the Vector GPS positioning data lined up well with the state estimation (Figure 5.9, Figure 5.10). All this suggests that instead of basing the GPS measurement covariance matrix (R_k) on the internal statistics, which approximates the positioning variance over the time period of many hours, a much tighter bound on the GPS uncertainty should be used. That is, an uncertainty which takes into account the short term (high frequency) GPS noise and not the long term drift. In this manner, the agent will fuse the Crescent GPS positioning measurements into the state estimation with more importance. Hence, the performance will most assuredly improve, and quite possibly surpass the performance found during the last six mowing swaths of Test-4 and the first ten mowing swaths of Test-1.

A few additional remarks pertaining to the Crescent Vector GPS drift are in order. Although the GPS positioning drift increases with time, it is bounded with a certain accuracy. In fact, the published positioning statistics for the Crescent Vector GPS unit (likewise the internal predicted statistics) represent this bound, which includes not only the short term noise characteristics but also the long term drift. For the Crescent Vector GPS unit, these bounds are defined over a 24 hour period, and under ideal conditions, (using WASS) a horizontal positioning accuracy of 0.6 m (24 in) is attainable 95% of the time, or 0.3 m (12 in), 68% of the time [60]. With this in mind, one must realize that if the area to be mowed is defined by a series of perimeter GPS waypoints (as suggested in Section 5.3), the mower may begin mowing one day with an absolute horizontal bias/error of 0.2 m (8 in) in the East direction and begin mowing in the exact same spot a week later with a bias of 0.3 m (12 in) in the West direction. In a similar manner, after mowing a 50 ft (15 m) wide median, the absolute bias/error may have drifted by a value of 0.6 m (24 in) in the transverse (perpendicular to the mowing swaths) direction. To account for this absolute bias/drift, one can simply add a small amount of width to the outer perimeter of the required mowing area, to be assured that the entire area is covered.

Based on the suggested modifications, specifically to alter the GPS uncertainty covariance matrix, it has been determined that the standard deviation of the width between consecutive mowing swaths can be reduced to 4 in (10 cm) or less. This attainable level of accuracy is excellent, considering that the published horizontal positioning accuracy (1-sigma) of the Crescent Vector GPS unit is only 0.3 m (12 in). Section 5.3 is focused on evaluating the mowing performance and the potential value of the proposed solution to Caltrans, based on this level of accuracy.

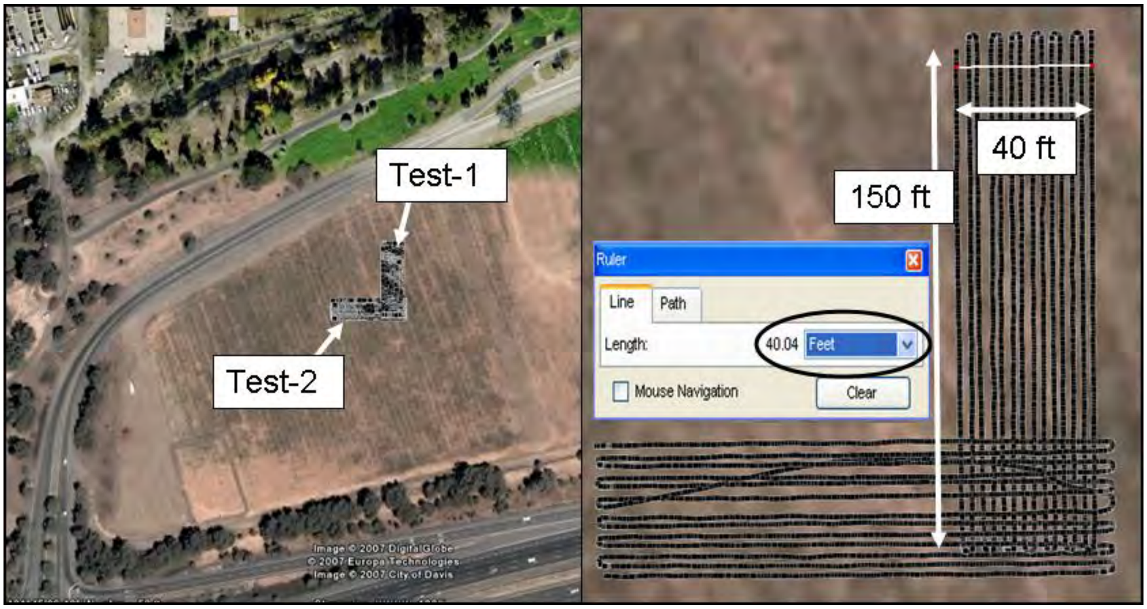
5.3 Theoretical Implementation and Benefit Analysis

The successful integration of autonomous mowing into Caltrans current mowing practices is an intricate task which requires a comprehensive plan that takes into account factors such as: statewide mowing acreage, mowing schedules, knowledge of areas that are well suited for autonomous mowing, and development of a supporting infrastructure (e.g. recharging stations). While the complete development of such a plan is beyond the scope of this report, this section is focused on introducing various tools and ideas to support such a plan. First in Section 5.3.1, the discussion of GIS software and its practical role in supporting autonomous mowing is covered. Then, a hypothetical implementation of multiple mowing agents and their work plan is discussed in Section 5.3.2. Finally, a preliminary benefit analysis is looked at in Section 5.3.3. In Chapter 6, an additional analysis of future development and ideas is discussed.

5.3.1 Integration of GIS Software

In recent years, the development of interactive Geographic Information System (GIS) software (e.g. Google Earth) has allowed one to visualize, process, and analyze geographic information [70]. GIS software technology is used for a variety of applications, for instance: asset management, scientific investigations, urban planning, history, sales and cartography. In fact, current research at AHMCT is focused on extending GIS software capabilities to support Caltrans needs. For example, by making use of GIS software to map, visualize, and store important information about roadside culverts, signs, landscaped areas and other features, Caltrans is better equipped to successfully maintain California's Highway infrastructure.

Many of the inherent challenges associated with autonomous roadside mowing are simplified by utilizing GIS software. For example, Caltrans would be able to map out regions that require vegetation control, map regions that have previously been mowed, and store other relevant information (e.g. the location of trees, shrubs, and guardrails, the perimeter of the mowing area, or the required mowing schedule). To demonstrate how Caltrans might use interactive GIS software to map and visualize previously targeted regions of vegetation, the GT GPS positioning information taken from Test-1 and 4 (Figure 5.5 and Figure C.4) were imported into Google Earth, as displayed in Figure 5.13. This figure displays the area coverage from the two tests as preformed in the large grassy field discussed in Section 5.1.1. Making use of a software measuring tool within Google Earth the width and length of Test-1 were found to coincide identically to the results plotted in Figure 5.5.



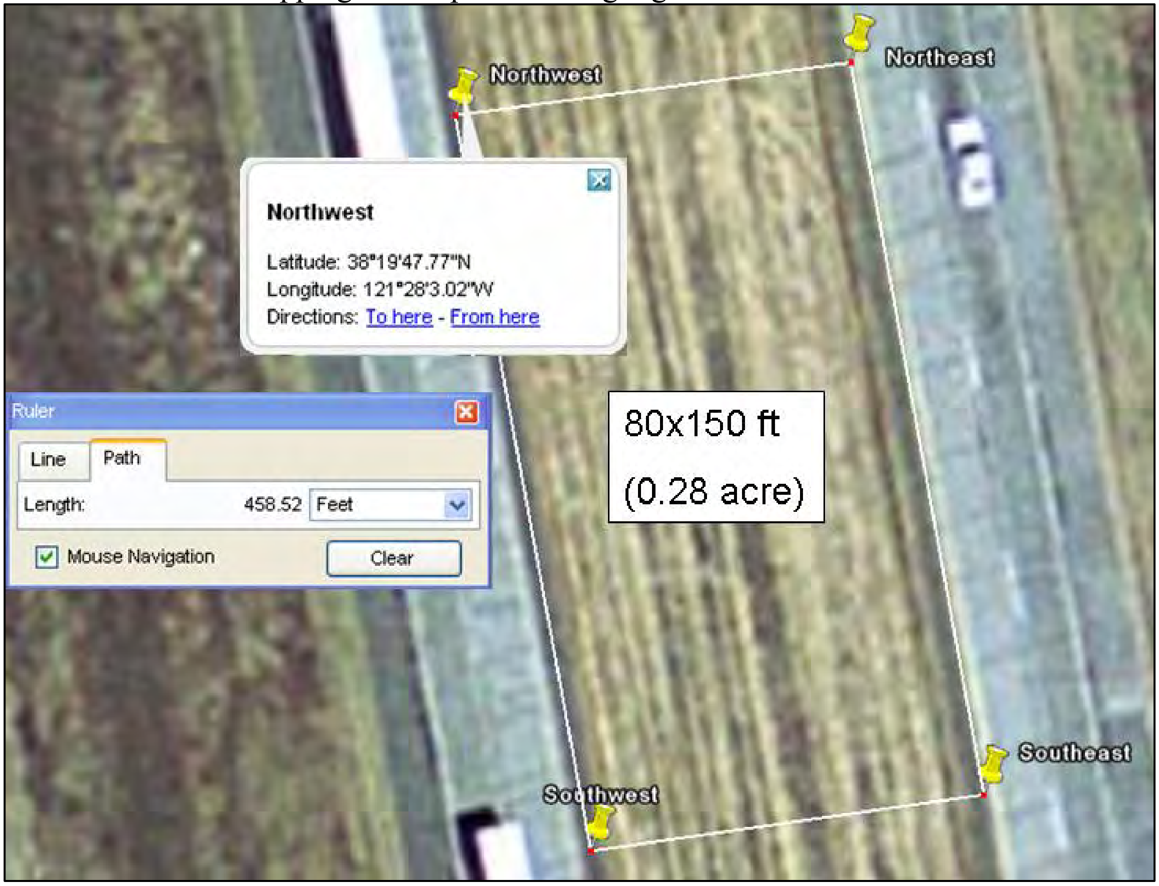
Length: 40.04 Feet

☐ Mouse Navigation

Figure 5.13 Google Earth mapping of the GT GPS data for Test-1 and 4, view at 1700 ft (left), view at 400 ft (Right)

Additionally, to demonstrate how Caltrans might use GIS software to plan out regions of vegetation that require mowing, a possible mowing area is mapped in Google Earth using a series of GPS way points (latitude and longitude values), see Figure 5.14.

Upon mapping such a mowing area, the defined perimeter, represented by a series of GPS waypoints, could be uploaded directly to the mowing agent. Furthermore, the agent would be able to use this defined perimeter to develop an optimal mowing path online during startup, just as accomplished for each of the final tests. While the absolute positioning accuracy of Google Earth GIS data is dependent on a given region (it can vary from one foot to tens of feet), other low cost GIS software packages provide high accuracy positioning data that can be utilized in the same manner. In addition, many other GIS software tools such as area calculations, polygon building, and GPS waypoint tools facilitate the mapping of complex mowing regions.



Northwest

Latitude: 38°19'47.77"N

Longitude: 121°28'3.02"W

Directions: To here - From here

Length: 458.52 Feet

☑ Mouse Navigation

Figure 5.14 Google Earth mapping of a Section of Interstate-5 near Sacramento to be mowed

The synthesis of GIS software and autonomous mowing provides many possibilities, which extend beyond just defining regions to be mowed. A few ideas are mentioned here. In addition to mapping the perimeter to be mowed, as in Figure 5.14, one might also use GIS data to map out non-traversable areas, e.g. trees, shrubs, guardrails, or ditches. After uploading this information to the agent, the agent would be able to develop a mowing path which will avoid these areas. Also, depending on the positioning accuracy of this knowledge, the agent might be able to utilize these landmarks to further improve its mowing performance. Additionally, a Caltrans employee might make use of the real-time positioning feedback from each autonomous mowing unit (by means of a broadband Ethernet link) to visualize its real-time global position in GIS software. In this manner, the worker would be able to track the location and status of each unit remotely. As a final idea, by making use of a GIS database, Caltrans would be able to develop a consistent yearly mowing schedule.

5.3.2 Hypothetical Implementation of Multiple Autonomous Units

Although a mechanical cutting system was not actually attached to the testbed at the time of this writing, current efforts are focused on adding a rugged 40 in (1 m) sickle bar cutting system, with an effective cutting width of 38 in (~1 m). Therefore, to support the analysis throughout the rest of this chapter, it will be assumed that the proposed agent has a cutting width capacity of 38 in. Furthermore, based on the results of Section 5.2.3, it will be assumed that the standard deviation of the width between consecutive mowing swaths is 4.0 in (10 cm). Finally, it is assumed that Caltrans is quite adverse to uncut/missed vegetation, and henceforth requires a path overlap of two standard deviations, or 8 in (20 cm).

Based on the hypothetical assumptions listed above, it was determined that the proposed agent has the following performance characteristics: an area coverage effectiveness of 99.92%, an efficiency of 78.9%, and a mowing rate of 5.5 ft^2 /s (0.5 m^2/s). These performance values were determined by replicating the procedure covered in Section 5.2.2. Making use of these values, the hypothetical implementation of multiple autonomous mowing units will be discussed next. This implementation suggests ways upon which Caltrans might successfully integrate autonomous mowing into their current mowing practices.

HYPOTHETICAL IMPLEMENTATION: After arriving to work at 8:00 AM Monday morning, a Caltrans employee browses through the maintenance database and determines that a large section of Interstate-5 near Sacramento requires immediate mowing. To handle the task, he opens his interactive GIS software package and looks over the section that requires mowing. The GIS database indicates that the section is an 8.7 mile (14.0 km) long by 80 ft (24 m) wide center median, between Elk Grove Blvd. and County Highway E13 (part of this region is displayed in Figure 5.14). Furthermore, the GIS database (which stores information about the particular section) indicates that the area is well suited for Caltrans fleet of autonomous vegetation control units. After browsing through the database for an appropriate strategy, he realizes that this is the first time the autonomous units will be utilized for this stretch of the highway.

In preparation for the mowing, some simple calculations are made to determine that he should deploy six units, collected each day, for a period of four days. The calculations are based upon the following reasoning. Each unit has an effective mowing rate of 5.5 ft^2 /s (0.5 m^2/s), hence the agent will require 2.2 hours to mow one acre (4000 m^2) of vegetation. It is assumed that the battery life of each unit is nearly 3 hours (as estimated by analyzing the power consumption of the platform), thus each unit will cover roughly one acre for each recharge cycle, taking into account the time required to travel to and from the recharging station and the slower time required to perform turns. In addition, the recharging time for the automotive batteries used on the agent is conservatively estimated to be 3 hours (in reality charging cycles of as little as 1 hour are possible). Consequently, each mowing unit is able to cover 4 acres (16,000 m^2) over a 24 hour time period, and since 6 units will be deployed, the total mowing area is 24 acres (~100,000 m^2).

Given that the median width is 80 ft (24 m), each targeted mowing acre (4,000 m^2) corresponds to a section of the center median that is 545 ft (166 m) long. Furthermore, the entire 24 acres (~100,000 m^2), which is mowed over the period of 24 hours, corresponds to a total lane mile distance of nearly 2.5 miles (4 km). As a result, it will take the fleet of six autonomous mowing agents about 4 days to complete the job (i.e. cover the 8.7 lane miles). A schematic layout of this mowing strategy is included in Figure 5.15.

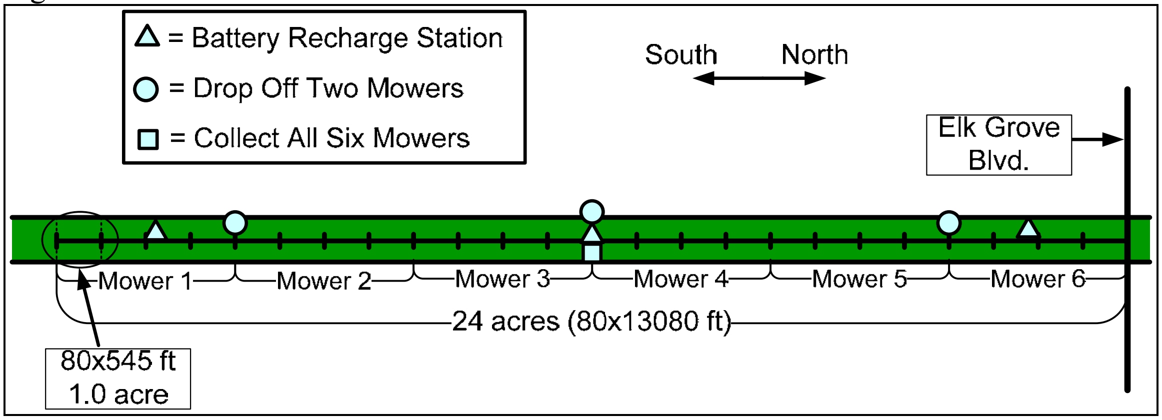


Figure 5.15 Proposed mowing layout for a fleet of six autonomous mowers

To support this mowing strategy (displayed in Figure 5.15), the employee makes use of his GIS software package to measure out 24 equal length acre sections of the center median, starting from Elk Grove Blvd. Furthermore, after utilizing the software tools to measure out these mowing sections, the employee records the GPS way points for the perimeter of each section (as performed in Section 5.3.2), and saves them to his laptop. He also makes a note of the global positioning of any other features (e.g. trees, holes, culverts, guardrails, recharging stations, etc.) within these sections that may previously have been stored in the GIS database. With the information now saved on their laptop, the employee travels to the nearest Caltrans equipment facility to pick up the six mowing units which will be used.

Upon arriving at the equipment facility, the Caltrans employee attaches a flatbed tow-behind trailer to his pickup truck and starts loading the agents onto the trailer. Each unit has an easy to use remote control interface that is used to drive the units up onto the trailer by means a handheld controller. After loading all six units onto the trailer, it is approximately 10:00 AM and he leaves the facility and travels to the first site. Upon arriving to the first site, which is 2180 ft (664 m) south of the intersection of Interstate-5 and Elk Grove Blvd., he uses the R/C interface to unload mowing units 5 and 6 onto the center median (Figure 5.15). Then he uses his laptop to transfer the GPS way points to each of the units and starts the execution of both mowers. In a similar fashion, the employee travels to the next site at which point he deploys mowing units 3 and 4, and then travels to the last site whereupon they deploy the last two units 1 and 2 (Figure 5.15). At this point it is roughly 11:30 and the employee returns to his office.

Each mowing unit begins its mowing cycle by traveling to the first section which requires mowing. Each mowing unit, will cover four successive acres, starting from the northern most acre (of its 4 acre section) and ending with the southern most acre (Figure

5.15). In this manner, the six mowing units need not worry about collision avoidance between one another, since there will always be at least a few one-acre sections between them while mowing. Upon arriving at the first section, the mower will preplan an optimal mowing path based upon the GPS waypoint perimeter defined by the Caltrans employee, and any other information, such as the location of obstacles to avoid. Each unit acts as its own individual entity, mowing without concern for the other mowing units. After the autonomous agent finishes each acre (4,000 m^2), or determines that its battery life is too low, it will make use of previous knowledge as to the location of the recharging stations (which are spaced at one mile increments along the roadside) to locate the nearest station, travel to it, and begin its charging cycle. After charging is complete, it will continue with its previous work, or if the previous acre is complete, the agent will travel to its next acre site, plan a new mowing path and begin.

Each autonomous mowing unit has an onboard broadband Ethernet connection which provides real-time feedback to the Caltrans employee from their office computer. This information is linked to the GIS software package to provide real-time visual feed back as to the location, status, and mowed area for each unit. The Caltrans employee has the rest of his day free to work on other jobs; occasionally looking over the status of the mowing units to ensure that they are operating properly. Before leaving his office for the night, he maps out 24 more uncut acre plots of vegetation, which will be loaded into the agents the following day. The mowing units continue work throughout the night until all of the required acres are complete, at which point, each individual mowing unit will travel to the center of the 24 acre mowing section for collection the next morning (Figure 5.15).

After arriving to work the next day, the Caltrans worker makes use of his GIS software to evaluate the progress of the mowers, which at this time should have completed mowing. If the mowers are complete, he will travel to the roadside site to collect all the units. He will then travel to the next consecutive 24 acre median section, reload the GPS information and start the units once again, just as he had the previous day. If by any chance one of the units was unsuccessful in finishing its mowing area, he will leave a single unit behind to complete any unfinished mowing. The overall procedure will continue for the next few days until the entire 8.7 mile (14 km) stretch of Interstate-5 is complete. Upon completion, he will return all the mowing units to the equipment facility for periodic maintenance. In addition, after returning to his office, he will store any valuable information, e.g. perimeter GPS way points, complications, the location of obstacles, etc, into the GIS database to assist in the future mowing of that stretch of highway.

Of course, the implementation detailed here is not the only implementation, or even the optimal implementation, but rather just one possible solution. Another implementation might involve four units in the center median and one mower on either side of the shoulder. Many of the ideas developed and used in this hypothetical analysis will be expanded on further in Chapter 6, e.g. recharging stations, broadband Ethernet links, and battery level monitoring.

5.3.3 Benefit Analysis

Up until now, the support for autonomous mowing has been largely theoretical. A more quantitative evaluation is taken within this section in an attempt to justify the integration of autonomous mowing into Caltrans current mowing practices. This analysis will focus on the following categories: cost reduction, reduced herbicide use, enhanced safety (worker and public), and improved aesthetics. This is a preliminary analysis, as only a partial (proof-of-concept) solution has been developed at this time.

COST ANALYSIS: A deeper analysis into the cost benefits associated with autonomous mowing was supported by examining Caltrans IMMS Vegetation Control Costs Report for the Fiscal 2004/2005 year [71]. In particular, by comparing the average costs per acre for Caltrans current mowing practices to the anticipated costs per acre (4,000 m^2) for autonomous mowing, one is able to predict the potential benefits associated with an autonomous solution. For the purpose of this analysis, Mechanical Control data for Family C was looked at. This data corresponds to all roadside vegetation mowing that requires tractor mounted mowers of any kind, i.e. mowing of medians, shoulders, mow strips, and quadrants, brush removal, and other mechanical tasks.

The IMMS report [71] indicates that the average mowing cost per acre varies significantly from district to district, which is most likely attributed to the relative complexity and amount of mowing for each district However, it is of no surprise, that District 3 happens to have the lowest cost per acre, at a cost of \$73.40, since earlier in Section 1.3.2 it was noted that nearly 85% of the mechanical mowing associated with District 3 was large area mowing, which is typically lower in cost per unit area (District 3 also happens to have the second largest total mowing acreage). Therefore, as a conservative approach, this average value will be used for comparison, even though values as large as \$370.00 per acre were noted for other districts. This value is also more fitting because the proposed unit is aimed mainly at improving large area mowing practices. The IMMS report also indicates that nearly 90% of the annual mowing cost is attributed to labor costs; this is true for all twelve districts. The remaining 10% of the cost is attributed to other factors such as material costs and equipment costs.

In comparison, the implementation discussed in Section 5.3.2 suggested that a single employee could maintain six autonomous units which are capable of mowing 24 acres per day. Furthermore, as a conservative approximation, it will be assumed that this employee spends his entire eight hour day setting up, collecting, and monitoring the autonomous units. Therefore, as a rough approximation, the labor costs associated with autonomous mowing is only \$10.80 per acre (i.e. 8 hours at the average employee wage of \$32.50 equals \$260 divided by 24 acres of cut vegetation). In addition, it is assumed that extra costs associated with material, transportation, and maintenance, are quite small in comparison to the labor costs, however, it is difficult to make such an assumption at this stage of the development. Therefore, as a conservative but rough estimation, it is assumed that the average total cost per acre for autonomous mowing technology is \$15.00 per acre.

For District 3 it was noted that 85%, or about 16,000 acres (6,500 ha) of vegetation, is accounted for by mowing of shoulders, medians, and quadrants (Section 1.3.2). To quantify the potential cost benefit associated with autonomous mowing it is assumed that only 25% of this acreage can utilize autonomous mowing technology, i.e. 4,000 acres (1600 ha). The cost of mowing 4,000 acres utilizing old mowing practices is nearly \$294,000, and only \$60,000 based on the estimated cost for autonomous mowing. Moreover, even if autonomous technology can only be employed for 25% of the total acreage, the potential yearly cost savings is about \$230,000, for District 3 alone. This is a significant amount of money that could be applied to the upfront cost associated with integrating such autonomous technology. That is, the cost associated with purchasing or renting autonomous mowing units and the cost required to integrate a supporting infrastructure (e.g. recharging stations, etc.). As a final note, it is anticipated that the production cost for an autonomous mower (such as the on developed in this report) would be less than \$5,000. This is a rough estimate, but was supported by evaluating the cost of the testbed developed in this project.

Based on the results from above, it is expected that the cost benefits associated with the integration of autonomous mowing technology would more than outweigh the upfront cost investment for Caltrans within a single year. This analysis was focused only on the benefits associated with the mowing of large median and shoulder sections, however, the potential cost savings is also impacted by the reduction in herbicide use. It is also noted that the proposed autonomous agent has the potential to significantly impact other mowing practices. For instance, one could utilize the technology developed in this project to target landscaped areas or other manual labor tasks.

REDUCTION IN HERBICIDE USE: The reduction in herbicide use is more difficult to quantify, but follows from a replacement of herbicide treatment with autonomous mowing. In Section 1.3.2 it was noted that 83% of the herbicide used for roadside vegetation maintenance (family C) is used for clear strips. It was also determined in Teeter-Balin's study [2] that mechanical mowing of clear strips was a suitable alternative to herbicide treatment. However, since herbicide treatment is typically more effective than mowing, frequent yearly mowing is required in order to maintain the same level of vegetation control.

The IMMS Vegetation Control Costs Report for the Fiscal 2004/2005 year [71] indicates that the average statewide cost for herbicide treatment is nearly \$185 per acre (4,000 m^2). Relatively 50% of this cost is made up of labor cost, 45% of material cost, and the remaining percentage for vehicle costs. Therefore, by replacing herbicide treatment of clear strips with autonomous mowing, Caltrans might be able to not only reduce their dependence on herbicide, but also benefit from substantial cost savings, even if sections must be mowed 4 or 5 times a year to achieve the same level of vegetation control (as it was estimated that autonomous mowing costs only \$15 per acre but herbicide treatment cost \$185 per acre). Another possibility might be, to make use of autonomous mowing and herbicide treatment in alternate years. As a final note, autonomous mowing of medians and shoulders also supports sustained clear strip maintenance since the clear strip is part of both.

IMPROVED AESTHETICS: Improved aesthetics follows primarily from a significant reduction in mowing costs. Moreover, instead of mowing only once or twice a year due to budget limitations, the cost savings associated with the integration of autonomous mowing technology would provide Caltrans with the freedom to mow three or four times a year, resulting in heightened levels of vegetation control and similarly improved aesthetics.

ENHANCED SAFETY: Safety is perhaps one of the most important considerations for Caltrans. Therefore, an alternative mowing solution would be of little value to Caltrans if it did not in some way improve the level of safety. Fortunately, autonomous mowing technology supports heightened levels of safety in several ways. One of the most obvious ways is that it reduces worker exposure to traffic. That is, beside the small amount of time required to deploy and collect the mowing agents (hypothetically on the order of 30 to 60 minutes); the autonomous units work with limited human interaction. Thus, unlike current methods that require Caltrans employees to work along the roadside, an autonomous unit allows one to monitor the mowing progress from a remote station.

Another means by which autonomous mowing could improve safety is by reducing lane closures, which place both workers and the public in harms way. Instead of mowing from the roadside with a tractor mounted boom mower, as is typically done for clear strip mowing, an autonomous unit can work directly within the clear strip.

A less obvious safety improvement results from reducing the roadside wildfire concern. To be more exact, current mowing budgets limit the frequency of yearly mowing, therefore, when mowing is permitted, large amounts of vegetation are left behind to dry out and increase the risk of wildfires. On the other hand, an autonomous solution would permit more frequent mowing; resulting in less cut biomass (or duff) which will biodegrade faster and reduce the fire concern. Finally, heightened levels of vegetation control yield better driver visibility and therefore public safety.

In brief, a number of clear indicators, as identified above, suggest that autonomous mowing technology is not only practical but feasible, and worthy of future development.

5.4 Chapter Summary

To evaluate the performance of the proposed agent, four final outdoor tests were conducted. These tests were performed within a large uneven grassy field on campus at the University of California, Davis. The details of the final testing configuration were discussed at the beginning of this chapter. Next, additional consideration for the Crescent Vector GPS heading lag and positioning divergence was mentioned, followed by an explanation of how these uncertainties were handled to yield smooth precise tracking control.

After introducing the testing configuration, the path tracking results for each of the final four tests were presented. The resulting plots displayed the tracking paths for the internal state estimation model in comparison to the high accuracy ground truth path. It was determined that the width consistency between consecutive mowing swaths would be used to evaluate the agent's performance. After presenting the width consistency plots for each of the final tests, a statistical performance analysis was presented based upon a hypothetical mowing unit with an effective cutting width of 42 in (1.1 m) and a path overlap of 6 in (15 cm). This analysis detailed how the required path overlap affected the mowing effectiveness, efficiency, and rate.

A deeper analysis into the final width consistency plots indicated long periods of large consistency and short periods of high variability. To explain this observation, a deeper look into the characteristics of the Crescent Vector GPS unit was conducted. A number of important conclusions were made. For one, it was determined that the GPS positioning measurements tend to be very precise over short periods of time, but contain a slow moving bias. Furthermore, while the EKF filtering process is good at removing high frequency positioning noise, it cannot compensate for the positioning drift. However, since mowing consistency is more important than absolute localization, this slow moving bias can be ignored, given that the time between two consecutive mowing swaths is kept below several minutes. Moreover, it was concluded that the EKF solution should make use of a measurement covariance matrix that accounts for the short term GPS noise characteristics, but neglects the long term drift. If this is done, the width consistency between consecutive mowing paths was estimated to be 4.0 in (10 cm) or less, at one standard deviation. This attainable level of performance is excellent considering that the published positioning accuracy for the Crescent Vector GPS unit is on the order of 0.3 m (12 in) at one standard deviation.

Following this analysis, a look into the features of GIS software and its practical role in supporting the integration of autonomous mowing technology into Caltrans mowing practices was explained. Next, the hypothetical implementation of multiple autonomous mowing agents into Caltrans mowing practices was considered. This study expanded how an employee might utilize GIS software to map out areas that require mowing, and how they might utilize multiple units to cover large sections each day. In conclusion, a more quantitative analysis was presented to explore the benefits associated with autonomous mowing. It was determined that the integration of autonomous mowing technology will enable Caltrans to improve safety, improve aesthetics, reduce their herbicide use, and substantially decrease their yearly mowing costs.

CHAPTER 6 Conclusions and Future Development

A brief summary of the findings throughout this report are discussed at the beginning of this chapter. Then, in an attempt to support the development of a future prototype to meet Caltrans needs, recommendations for future testing and development are explained.

6.1 Report Conclusions

Since the Environmental Impact Report of 1992, which urged Caltrans to cut back on their herbicide usage, Caltrans has been exploring alternative vegetation control techniques. To support Caltrans pursuits, AHMCT has conducted a detailed research study into the practicality of autonomous mowing technology. Early analysis concluded that that the majority of Caltrans mechanical mowing budget is dedicated to vegetation control of large medians, shoulders, and interchange quadrants. In addition, it was determined that the majority (over 80%) of the herbicide used for roadside vegetation maintenance is accounted for by clear strip application. In response, the development of an autonomous mowing agent to target roadside medians, shoulders, and clear strip sections has been proposed within this report.

To serve as a research tool for this study, a fully functional testbed was designed and built. Taking into mind Caltrans needs for a cost-effective design, the platform utilized a low cost sensor array consisting of wheel mounted shaft encoders and a low cost DGPS unit. The measurements from both sensor sources were merged by means of an Extended Kalman filter. After thorough testing and analysis of the proposed solution it was determined that a mowing consistency on the order of 4.0 in (10 cm) was attainable. Furthermore, it was concluded that the proposed solution has the potential to be of great value to Caltrans with respect to the categories of cost reduction, reduced herbicide use, improved safety, and enhanced roadside aesthetics, justifying the future development of a prototype.

6.2 Future Development and Testing

This report has detailed a variety of topics related to the development of autonomous mowing technology and the potential value of such technology to Caltrans. While many important topics such as localization, mapping, and path planning have been discussed, only a partial solution has actually been designed and built at the time of this writing. To support the successful development of a future prototype, this section is focused on introducing recommended future development, testing, and various supporting ideas.

OBSTACLE AVOIDANCE: A practical autonomous roadside mowing agent requires obstacle avoidance, since trees, shrubs, guardrails, ditches, tires, etc. may hinder its movement at any time. There are two main tasks associated with obstacle avoidance. First, an agent requires some way of identifying obstacles and other non-traversable areas, and second, based on this knowledge, the agent must plan a course of action to circumvent these obstacles. If the agent is given prior knowledge as to the locations of obstacles within its environment, the task is limited to planning a path to avoid these obstacles. More likely, however, the agent will not have complete knowledge as to the location of all obstacles. In fact, since the environment is dynamic the agent must account for changes, such as the location of obstacles.

To accomplish obstacle identification, the proposed agent should utilize additional sensors to perceive the surrounding roadside environment. Sensor technology such as Laser Detection and Ranging (LIDAR), Image Processing Units (IPU), Sonar, and bump sensors are all potentially viable. Future development should attempt to identify, possibly through experimentation, a sufficient sensor array to accomplish obstacle detection. The selected sensors should reflect Caltrans requirements for a cost effective, safe, and reliable agent.

Obstacle circumvention is typically accomplished on-line in software, and primarily deals with the task of re-planning the agent's current tracking path to account for detected obstacles. Many techniques have been proposed over the years, such as the potential field method [4], which was discussed in Section 2.1.3. Also of noteworthy attention is the cell decomposition [4] technique used by Schworer [34], to pre-plan an optimal "strip filling" mowing path for irregular sections of vegetation. His implementation also accounted for random and unknown obstacles during mowing. Future development should attempt to identify an efficient and effective method for achieving obstacle circumvention.

In addition to simple obstacle avoidance, a more robust overall solution should attempt to map the location and dimensions of various obstacles. Such a solution has several distinct advantages. For one, by creating an internal mapping of these obstacles, the agent is better prepared to circumvent these obstacles later in its mowing cycle. Second, the knowledge of these obstacles can be stored in a database for more efficient future mowing. Finally, if these obstacles are accurately identified and stored in a database, an agent could potentially utilize this knowledge to better localize itself during future mowing.

MOWING PERFORMANCE: The mowing performance was studied in detail in Chapter 5. The results indicated that by altering the measurement covariance matrix (R_k) to trust the GPS positioning measurements more, an enhanced level of performance would be attainable. Therefore, future development should strive to evaluate the proper measurement covariance matrix and to re-evaluate the attainable mowing consistency, which was estimated to be better than 4.0 in (10 cm) at one standard deviation. To calculate the measurement covariance, one should quantify the short term high frequency positioning variance (i.e. approximately the GPS positioning noise) but neglect the long term drift (Section 5.2.3).

As noted in Chapter 5, the Crescent Vector GPS heading lag (during turning maneuvers) had a negative impact on the tracking stability and similarly the mowing performance. Several steps were taken to temporarily solve this problem (Section 5.1.2), e.g. the heading measurements were neglected during turning maneuvers and the confidence in the heading measurements was decreased. While these steps yielded good performance, an alternative solution and possible a more effective solution would be to quantify the lag, and to compensate for this value in the EKF process. By quantifying the heading lag, one is able to fuse the heading measurements into the state estimation at the correct absolute point in time; otherwise the lag tends to corrupt the state estimation. This is a more fitting solution, because the current technique relies only on dead reckoning during turning maneuvers, and, as mentioned in Section 2.1.1, dead reckoning is a poor means for obtaining heading.

Another potential solution would be to utilize the heading measurements from the onboard Vector GPS gyroscope during turning maneuvers. This technique has its own advantages. For one, the gyroscope measurements are quite accurate over short periods of time and can be easily read from the Vector GPS unit. Also, the gyroscope is an excellent substitute for obtaining heading measurements during brief periods of GPS outage. Future work should experiment with both techniques, possibly combining the two methods, to improve the state estimation heading accuracy and hence the mowing performance. Additionally, future experimentation ought to verify that an acceptable level of performance may be obtained even during short GPS outages, which are possible within the roadside environment.

While the mowing performance was evaluated on a harsh (uneven and bumpy) grassy field (Section 5.1), this field was quite flat in the global sense. Therefore, future testing must evaluate the robustness of the proposed localization scheme under other operating conditions, e.g. on an inclined slope or with additional resistance equivalent to the expected mechanical cutting resistance. Although a 2D localization model was used, the GPS positioning measurements are used in the EKF process to correct for the inaccuracies that result from the 2D model. With that in mind, one would expect that hilly terrain would have little impact on the performance, however, experimentation should verify this hypothesis.

Another important operating condition for future testing is an alteration in the top mowing speed, since the mowing speed has a significant impact on the mowing rate and hence, the overall performance. For the purpose of this study the top mowing speed was confined to 1.5 mph (2.4 km/h), however, future testing should experiment with faster speeds. While increasing the top mowing speed will increase the mowing rate, it naturally has an adverse effect on the mowing consistency. The reasons for this are numerous, for one, dead reckoning occurs at a frequency of about 6 Hz (due to limitations on the RoboteQ controller, Section 3.2.3) thus, by increasing the speed, larger positioning changes occur between discreet positioning updates, yielding larger discretized errors and bigger positioning uncertainties. In addition, there are larger positioning changes for each discreet motor control update, resulting in tracking control that is not as smooth. Finally, larger speeds yield adverse dynamic effects, in particular, larger wheel slippage. Besides performance setbacks, one must realize that there may be other mechanical limitations. For instance, effective mowing, as defined by the particular mechanical mower, may limit the top mowing speed. Regardless, the potential improvement in mowing performance justifies additional experimentation with faster mowing speeds. In fact, preliminary simulation experimentation indicated relatively small performance setbacks associated with an increase in mowing speed.

Lastly, it was mentioned in Section 4.4.2 that an adaptive path planning algorithm is better suited to yield enhanced mowing performance. The current path planning algorithm develops a tracking path only once, on-line at the start up of each test. There are a few downsides to this approach. For one, such a technique does not allow provisions to altering circumstances. For example, a practical agent must account for random obstacles throughout its environment and circumvent them by altering the tracking path (as discussed earlier). A less obvious downside is the potential loss in mowing performance. Instead of making use of a predefined width between consecutive mowing swaths (as was done in the final outdoor tests), a better solution is to utilize adaptive widths based upon the present localization confidence. For example, when the agent's estimated positioning variance (eqn. 4.4) is low, a smaller mowing overlap is required to maintain the same level of mowing effectiveness; on the other hand, the overlap should be increased when the agent is quite uncertain of its positioning. In this manner the agent is able to account for alterations in the localization certainty that may arise from changes in the satellite visibility, terrain, speed, etc, yielding improved mowing efficiency. In brief, future work should attempt to design an adaptive path planning algorithm that takes into account the continual positioning uncertainty and obstacle avoidance. To assist in this design, the simulation (discussed in Section 4.5) is recommended for use.

SUPPORTING INFRASTRUCTURE: While the proposed system (i.e. hardware, sensors, and software) was selected to require limited structural roadside modifications (Section 4.2.1), prolonged periods of autonomous mowing, possibly for days on end, requires some method for recharging the mowing units. The technique proposed in Section 5.3.2 was to integrate designated recharging stations at one mile increments along the roadside; anywhere autonomous mowing would take place. These stations could possibly make use of standard electrical lines (maybe even coinciding with roadside call boxes), or may even utilize advanced solar technology. As an alternative method, a Caltrans employee may simply unload additional fully charged batteries that can be interchanged by the mowing units during operation. Such an implementation would require no roadside modifications, but would require additional overhead labor.

Expanding on the concept of a designated recharging station, one might design a weatherproof station, allowing the agent to be integrated to the roadside. Like so, the agent would hide away in its sheltered station until mowing was required. This technique is used by some commercial platforms (Section 2.2.2) and is potentially a fitting solution for autonomous mowing of highway landscaped sections (Section 1.3.1), which are mowed nearly every week. Future work should attempt to develop a suitable charging solution. As a final note, the RoboteQ motor controller (Section 3.2.3) offers the functionality to monitor the battery voltage level, enabling the agent to determine when a charging cycle is necessary.

MULTIAGENT COOPERATION: The ability to deploy several autonomous mowing units at a single time (with modest labor requirements) offers a distinct advantage over existing mowing practices. Such a situation was hypothesized in Section 5.3.2, whereupon six individual agents were deployed, each operating without regards for the other. While such a situation is realistic, the TCP Ethernet communication link that was established (Section 4.3) provides an assortment of additional possibilities. Such a link allows multiple units to communicate with one another, relaying assorted information, such as: status, individual location, mapped areas, the location of recharging stations, or even a call for assistance. In this manner the agents would be able to work as a team, assisting and monitoring one another. One may imagine a team of autonomous agents working together, e.g. high capacity mowers which target large sections of vegetation, small agile units that target vegetation in and around guardrails, fast units that accurately map the terrain and relay it to others for proper path planning, and agents which seek out and apply herbicide to noxious weeds (e.g. yellow star thistle). The possibilities are endless, and future work should examine varying alternatives to determine techniques of potential value to Caltrans.

FURTHER DESIGN: The testbed detailed in this report did not integrate a mechanical cutting system, therefore, the design of a cutting system is an obvious necessity for future development. This design should take into account the characteristics of the proposed testbed and also the roadside vegetation, e.g. dimensions of the testbed, speed limitations, weight, pay load, vegetation thickness, etc. In addition, to support an easy collection and deployment of future prototypes, the R/C communications link discussed in Section 4.3 must be implemented.

ADDITIONAL SUPPORTING IDEAS: Several other potential enhancements to the autonomous mower are listed as follows:

- Utilize GIS software to map out roadside features (e.g. trees, signs, guardrails) and use them for SLAM [15, 18, 19].
- Add a low cost IPU or other color detection sensor to detect the roadside edge, as an additional step in preventing the unit from entering the roadway.
- Remote monitoring, possibly using broadband Ethernet communication, for theft prevention.
- Incorporate a low cost IPU or other passive sensors to detect the cutting line, in an attempt to improve the mowing efficiency.
- Use a more robust tracking control algorithm (e.g. the one detailed in [54]) to account for uncertainties in the effective wheel radius and wheel base width. This would prevent against unlikely, but possible unstable tracking control.

EXPANSION TO OTHER SCENARIOS: While the autonomous mowing technology developed within this report has been largely geared towards mowing of median and shoulder sections, further development could yield solutions to alternative labor intensive tasks. For instance, the mowing of quadrants and landscaped areas, efficient herbicide application of roadside noxious weeds, or even an autonomous agent to pick up trash and debris along the roadside. Future research should attempt to determine if the technology detailed in this project can be applied to other situations.

PART 2 References

1. California Department of Transportation (Caltrans), *California Roadsides: A new Perspective.* 1997: Jones & Stokes Associates, Inc.

2. Teeter-Balin, Jonah, *Feasibility of Alternative Vegetation Maintenance Control Technologies,* (Thesis, UC Davis, 2004)

3. Jain, L.C. and J. Fulcher, *Applied Intelligent Systems: New Directions*. 2004: Springer.

4. Russell, S.J. and P. Norvig, *Artificial intelligence: a modern approach*. 1995: Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

5. California Department of Transportation (Caltrans), *Maintenance Manual, Chapter C2: Vegetation Control.* 2005

6. California Department of Transportation (Caltrans), Integrated Maintenance Management System(IMMS), *Fiscal Year 2004/2005 Vegetation Control Plan Report, 2005*

7. Clary, A.G., *The Role of Equipment in Maintenance Operations.* 1969.

8. US Ditcher, Inc. GRM60 Guardrail mower [cited Oct. 22, 2007] <http://www.guardrailmower.com/>

9. Fairman, F.W., *Linear Control Theory: The State Space Approach*. 1998: John Wiley and Sons.

10. Kiriy, E., *A Localization System for Autonomous Golf Course Mowers*.

11. Negenborn, R., *Robot Localization and Kalman Filters*. 2003, UTRECHT UNIVERSITY.

12. Xsens Motion Technologies, MTi, [cited Oct. 23, 2007] <http://www.xsens.com/index.php?mainmenu=products&submenu=machine_motion&subsubmenu=MTi>

13. PinPoint® Asset Management Solutions, [cited Oct. 23, 2007] <http://www.pinpointrtls.com/>

14. Trolley Scan (Pty) Ltd [cited Oct. 23, 2007] <http://trolleyscan.com/>

15. Kurth, D., *Range-Only Robot Localization and SLAM with Radio*. 2004.

16. Hahnel, D., et al., *Mapping and Localization with RFID Technology.* 2004: p. 1015–1020.

17. Chon, H.D., et al., *Using RFID for Accurate Positioning.* 2004: p. 6–8.

18. Montemerlo, M., et al., *FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges.* 2003. **1156**.

19. Choset, H. and K. Nagatani, *Topological simultaneous localization and mapping (SLAM): towardexact localization without explicit localization.* 2001. **17**(2): p. 125-137.

20. WIKIPEDIA: The Free Encyclopedia, *Kalman filter.* [cited Oct. 23, 2007] < http://en.wikipedia.org/wiki/Kalman_filtering>

21. Zarchan, P. and H. Musoff, *Fundamentals Of Kalman Filtering: A Practical Approach*. 2005: Aiaa.

22. Welch, G. and G. Bishop, *An Introduction to the Kalman Filter.* 2001.

23. ROBOTBOOKS.COM, *Robot Lawn Mower Designed to Change Suburban Landscape*, University of Florida, Gainesville, [cited Oct. 23, 2007] <http://www.robotbooks.com/lawn-mower-robot.htm>

24. Chandler, R.C., K.A. Meiszer, and A.A. Arroyo, *LawnShark: A New Platform for Autonomous Mowing and Navigation.* **5**: p. 29-30.

25. aboutAI.net, *Home Robots,* August 21,2000 (cited Oct. 23, 2007) <http://aboutai.net/DesktopDefault.aspx?tabindex=1&tabid=2&article=aa082100a.htm>

26. Chandler, R.C. and A.A. Arroyo, *Texture Analysis: A Technique for Autonomous Lawnmower Navigation.*

27. Hicks, R.W. and E.L. Hall, *A Survey of Robot Lawn Mowers.* 2000. **4197**: p. 262-269.

28. Ollis, M. and A. Stentz, *Vision-based perception for an automated harvester.* 1997. **3**.

29. Kiriy, E. and M. Buehler, *Three-state Extended Kalman Filter for Mobile Robot Localization.* 2002.

30. Aono, T., et al., *Positioning of vehicle on undulating ground using GPS and deadreckoning.* 1998. **4**.

31. Bruch, M.H., et al., *Accurate Waypoint Navigation Using Non-differential GPS.* 2002: p. 9-11.

32. Goel, P., S.I. Roumeliotis, and G.S. Sukhatme, *Robust localization using relative and absolute position estimates.* 1999. **2**.

33. Bonnifait, P. and G. Garcia, *A multisensor localization algorithm for mobile robots and itsreal-time experimental validation.* 1996. **2**.

34. Schworer, I.J., *Navigation and Control of an Autonomous Vehicle*. 2005.

35. Agrawal, M. and K. Konolige, *Real-time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS.* 2006: p. 1063-1068.

36. Cooper, S. and H. Durrant-Whyte, *A Kalman filter model for GPS navigation of land vehicles.* 1994. **1**.

37. Willgoss, R., V. Rosenfeld, and J. Billingsley, *High Precision GPS Guidance of Mobile Robots.* **1**(501): p. 1001.

38. Friendly Robotics: Robomow, [cited Oct. 23, 2007] <http://www.friendlyrobotics.com/>

39. HusqvarnaAuto: Mowers, [cited Oct 23, 2007] <http://www.usa.husqvarna.com/products_robotic_mowers.aspx?cid=c61>

40. Lawnbott Evolution Review: Lawnbott [cited Oct. 23, 2007] <http://www.bamabots.com/lawnbott-evolution-review.htm>

41. Epinions.com: Toro iMow Robotic Mower 30050 [cited Oct. 23, 2007] <http://www.epinions.com/hmgd-Lawn_and_Garden-Power_Tools-Mowers-Toro_iMow_Robotic_Mower_30050>

42. Dudek, G., *Computational Principles of Mobile Robotics*. 2000: Cambridge University Press.

43. Rizzoni, G., *Principles and Applications of Electrical Engineering*. 2005: McGraw-Hill Science/Engineering/Math.

44. Close, C.M. and D.K. Frederick, *Modeling and analysis of dynamic systems*. 2001: Wiley New York.

45. NPC Robotics, [cited Oct. 23, 2007] <http://www.npcrobotics.com >

46. Rinderle, James, *Permanent Magnet DC Motor Characteristics*. April 13, 2001.

47. Sears: DieHard Gold South, [cited Oct. 23, 2007] <http://www.sears.com/shc/s/p_10153_12605_02833151000P?vName=Automotive&cName=Batteries+%26+Chargers&keyword=diehard>

48. Bosch Rexroth: Aluminum Framing Shop, [cited Oct. 23, 2007] <http://www13.boschrexroth-us.com/framing_shop/>

49. RoboteQ: AX2550/AX2850, [cited Oct. 23, 2007] <http://www.roboteq.com/ax2550-folder.html>

50. Hong, D., S.A. Velinsky, and X.I.N. Feng, *Verification of a wheeled mobile robot dynamic model and control ramifications.* 1999. **121**(1): p. 58-63.

51. Lucas, G.W., *A tutorial and elementary trajectory model for the differential steering system of robot wheel actuators.* 2000.

52. Crandall, S.H., et al., *Dynamics of Mechanical and Electromechanical Systems. Robert E.* 1968, Krieger Publishing Co., Malabar, Florida.

53. Zhang, Y.L., S.A. Velinsky, and X. Feng, *On the tracking control of differentially steered wheeled mobile robots.* 1997. **119**(3): p. 455-461.

54. Zhang, Y.L., S.A. Velinsky, and D. Hong, *Robust Control of a Differentially Steered Wheeled Mobile Robot.* ASME DSC **119**, 1997.

55. Encoder Products Company: Model 15T Thru-Bore [cited Oct. 23, 2007] <http://www.encoder.com/model15th.html>

56. Crescent® Vector OEM: *High-Performance GPS Heading and Positioning Module,* [cited Oct. 23, 2007] <http://www.navtechgps.com/pdf/CrescentVectorOEM.pdf>

57. Crescent® Vector OEM: *Technical Brief,* [cited Oct. 23, 2007]

58. NovAtel® OEM4-G2: Product Datasheet <http://www.novatel.com/Documents/Papers/oem4g2.pdf>

59. NovAtel® , *Introducing NovAtel's New AdVance RTK^{TM}.* <http://www.insidegnss.com/pdf/Advance_RTK_White_Paper_Rev_1.pdf>

60. Hemisphere GPS, Formerly CSI-Wireless Inc. *CDA-3RTK: High Performance GPS Antenna* <http://www.saderet.co.uk/Admin/Datasheet/CDA3RTKantenna.pdf>

61. NovAtel® , *GPS-702L: Product Datasheet* <http://www.novatel.com/Documents/Papers/GPS-702L.pdf>

62. Fuke, Y. and E. Krotkov, *Dead reckoning for a lunar rover on uneven terrain.* 1996. **1**.

63. Holland, J.M., *Designing mobile autonomous robots: inside the mind of an intelligent machine*. 2004: Elsevier, Inc. Burlington, MA, USA

64. WinSystems® An Employee-Owned Company, EBC-855 Intel Pentium M SBC with Video [cited Oct. 24, 2007] <http://sbc.winsystems.com/products/sbcs/ebc855.html>

65. Rezaei, S. and R. Sengupta, *Kalman Filter Based Integration of DGPS and Vehicle Sensors for Localization.*

66. Chenavier, F. and J.L. Crowley, *Position estimation for a mobile robot using vision and odometry.* 1992: p. 2588-2593.

67. Kelly, A., *A 3D State Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles.* 1994.

68. Bar-Shalom, Y., X.R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. 2001: Wiley New York.

69. Phillips, C.L. and H.T. Nagel, *Digital control system analysis and design*. 1989: Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

70. WIKIPEDIA: The Free Encyclopedia, *Geographic information system.* [cited Oct. 24, 2007] < http://en.wikipedia.org/wiki/GIS>

71. California Department of Transportation (Caltrans), Integrated Maintenance Management System(IMMS), *Fiscal Year 2004/2005 C2 and E1 IMMS Cost Data*

72. McMaster-Carr® Casters, Part Number: 22925T72, [cited Oct. 24, 2007] <http://www.mcmaster.com/>

73. WIKIPEDIA: The Free Encyclopedia, *PID controller.* [cited Oct. 24, 2007] < http://en.wikipedia.org/wiki/PID_controller>

PART 2 Appendix-A

Additional Assembly Photos:

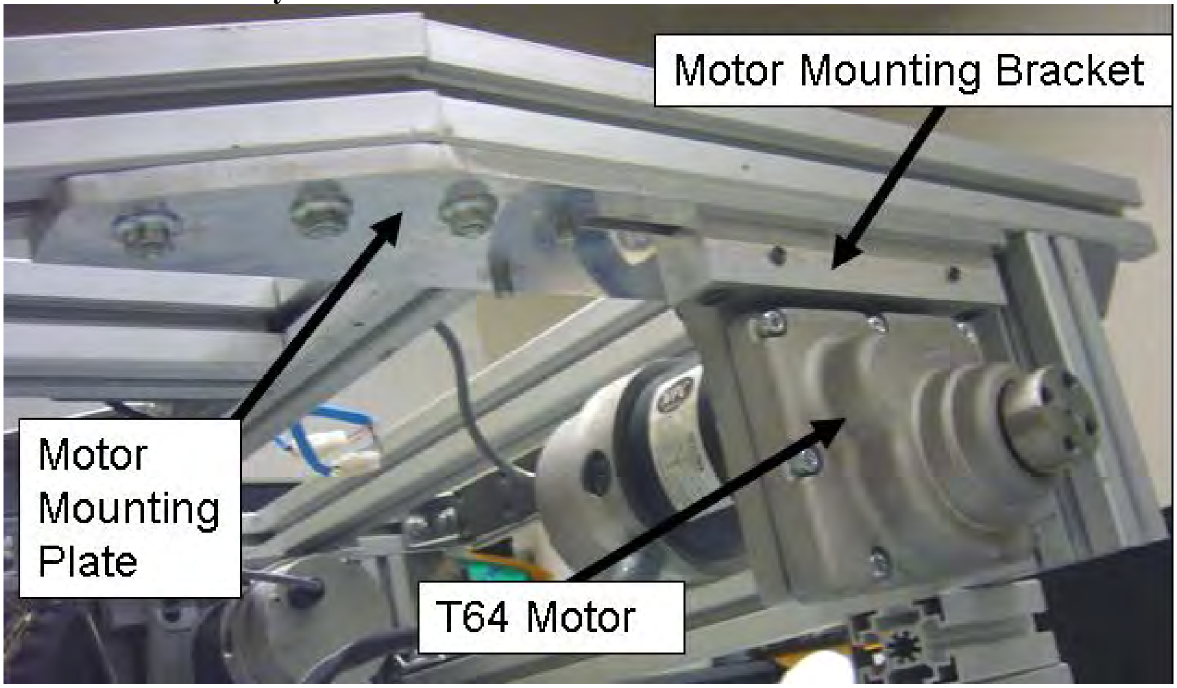


Figure A.1 Motor mounting assembly

Figure A.1 displays the final motor mounting assembly. The assembly is made up of the motor/gearbox, motor mounting plate, and bracket. The mounting plate, which was machined out of 3/8th in (0.95 cm) aluminum, was used to tie the frame together and to square up the motor mounting bracket for perpendicular mounting of the two T64 motors. To allow for adjustable placement of the motors, a channel was machined into the plate so that the mounting brackets could be slid forward/backward as was necessary to alter the vehicle's center of gravity. The mounting bracket was machined out of 1x1 in (2.54x2.54 cm) aluminum stock. It has two 5/16th in (0.79 cm) tapped holes which were used to secure the motor to the bracket (from the backside), and two vertical countersunk holes to fasten the assembly to the frame.

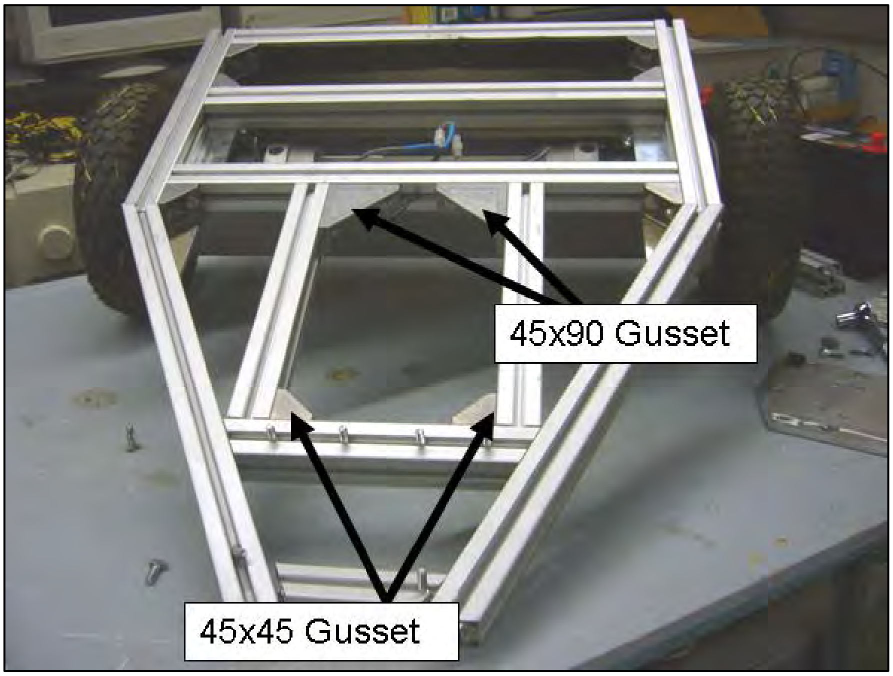


Figure A.2 Outer testbed frame

Figure A.2 is a photo of the testbed frame, which details the Bosch corner gussets (Figure 3.5) used to secure the frame together. The 45x45 gussets are fastened with two t-bolts while the 45x90 gussets have four separate t-bolt mounting holes

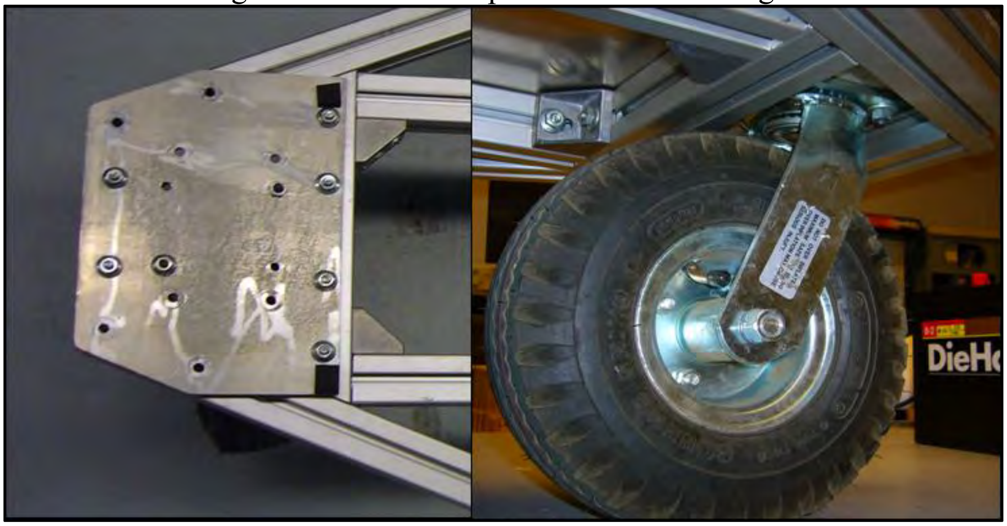


Figure A.3 Caster wheel mounting plate (Left), Pneumatic caster wheel (right)

In Figure A.3, the pneumatic caster wheel and mounting plate are displayed. The mounting plate was machined out of 1/4th in (6 mm) aluminum and was used to secure the caster wheel to the frame and also to add structural support in the front of the testbed. The pneumatic caster wheel, from McMaster-Carr [72], has a 10.0 in (25.4 cm) outer diameter with a loading capacity of 350 lbs (1557 N). Although this tire is pneumatic, variations in the radius of the wheel do not effect the localization (as do the drive wheels) and was judged as an acceptable choice. It was expected that the 10 in. outer diameter would be acceptable for traversing the rocky roadside terrain.

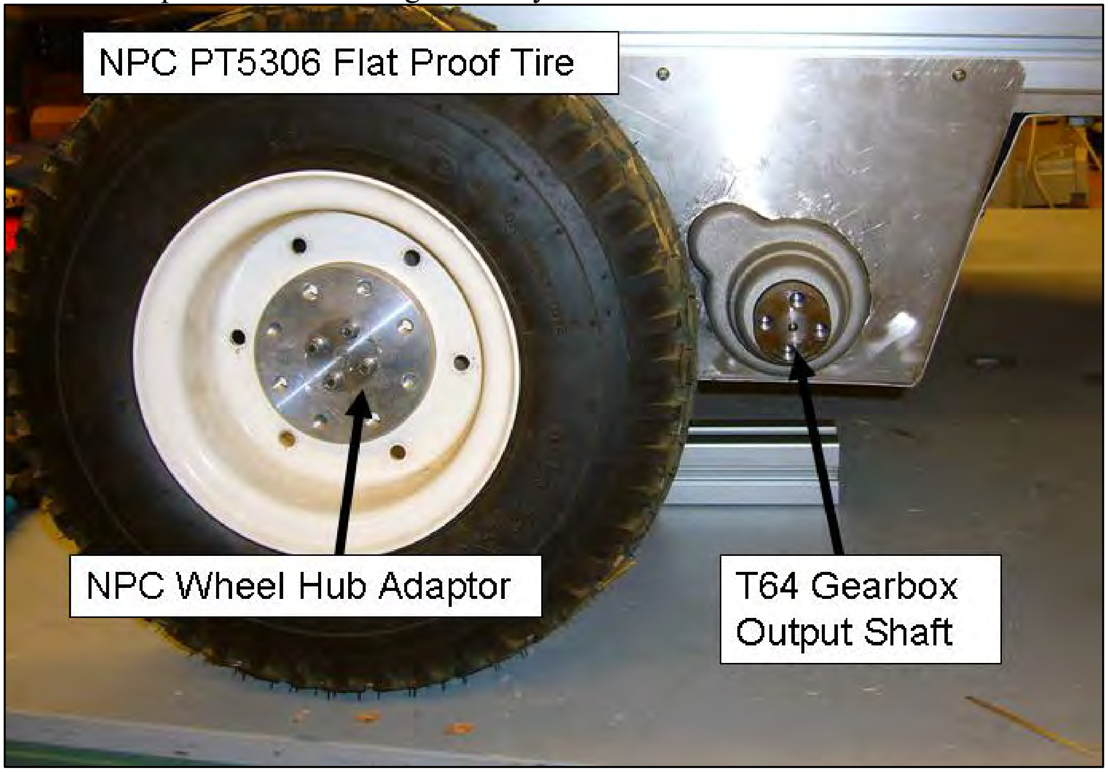


Figure A.4 Drive wheel assembly

In Figure A.4, the drive wheel subassembly is presented. The NPC-T64 motor gearbox unit has an output shaft with four threaded 5/16-24UNF holes. To mount the NPC PT5306 flat proof tires to this shaft, a 3.5 in (8.89 cm) wheel hub from NPC Robotics was used. For detailed drawings of the NPC PT5306 tire, wheel hub, and T64 motor, see [45].

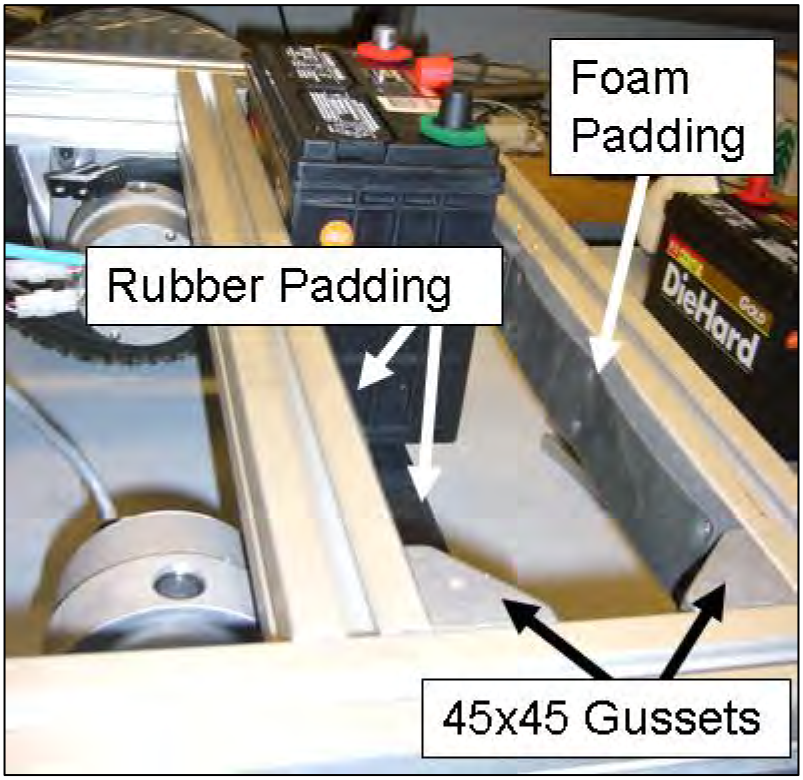
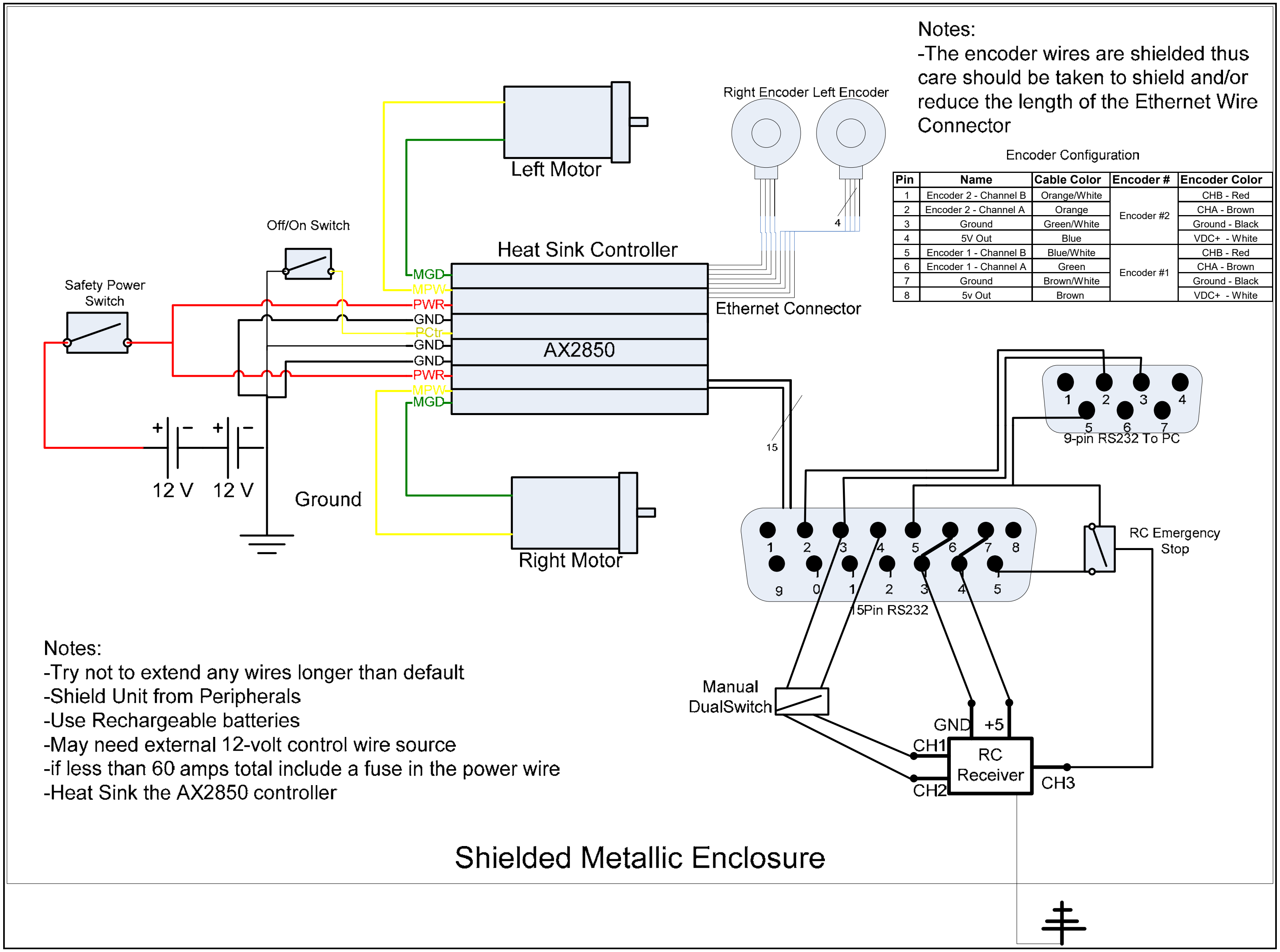


Figure A.5 Battery harness

The battery harness displayed in Figure A.5 was designed to secure the two DieHard automotive batteries, which were used for the main power supply. To secure the batteries in place, rubber padding was attached to the bottom and front beams and a ½ in. (1.27 cm) foam padding was attached to the back beam.

Motor Controller Setup and Discussion:

The circuit diagram for the RoboteQ AX2850 controller [49] is displayed in Figure A.6.



Notes:
-The encoder wires are shielded thus care should be taken to shield and/or reduce the length of the Ethernet Wire Connector

Encoder Configuration

Pin	Name	Cable Color	Encoder #	Encoder Color
1	Encoder 2 - Channel B	Orange/White	Encoder #2	CHB - Red
2	Encoder 2 - Channel A	Orange		CHA - Brown
3	Ground	Green/White		Ground - Black
4	5V Out	Blue		VDC+ - White
5	Encoder 1 - Channel B	Blue/White	Encoder #1	CHB - Red
6	Encoder 1 - Channel A	Green		CHA - Brown
7	Ground	Brown/White		Ground - Black
8	5v Out	Brown		VDC+ - White

Notes:
-Try not to extend any wires longer than default
-Shield Unit from Peripherals
-Use Rechargeable batteries
-May need external 12-volt control wire source
-if less than 60 amps total include a fuse in the power wire
-Heat Sink the AX2850 controller

Figure A.6 Motor controller circuit design

As noted in Section 3.2.3, this configuration makes use of two separate switches: a safety power switch, used to disconnect the power supply from the batteries, as was suggested by the manufacture, and a controller on/off power switch. The controller on/off switch works as follows; when the power control wire (PCtr, Figure A.6) is left floating (unconnected) the controller makes use of an internal DC to DC voltage regulator to supply the necessary power to the controller from the main power supply. However, when this second switch is flipped to connect the power control lead to ground, the controller shuts off.

Figure A.6 also demonstrates the proper configuration of the two quadrature encoders, which are interfaced to the controller via a standard eight lead twisted pair Ethernet wire. To communicate with the controller, two configurations are possible, R/C PWM or a RS232 link. A combined solution was initially proposed, and although only the RS232 interface was implemented, it is suggested to use the dual configuration demonstrated in Figure A.6 in future development. In this manner, a user can utilize the R/C interface to easily setup and collect the robot with a handheld RC controller. Then, when operation begins, the interface can easily be switched to the necessary RS232 autonomous control. Also included in this RC interface is a configurable remote emergency stop as yet another level of safety (Figure A.6).

Figure A.7 displays the actual implementation of the safety power switch and controller on/off switch; the RS232 link is also annotated. This figure also annotates one of the two angle plates that were used to secure the aluminum mounting plate to the frame. These angle plates, which were machined out of 1/4th in (0.64 cm) aluminum, added structural support to the frame, since they split but end joints of the frame.

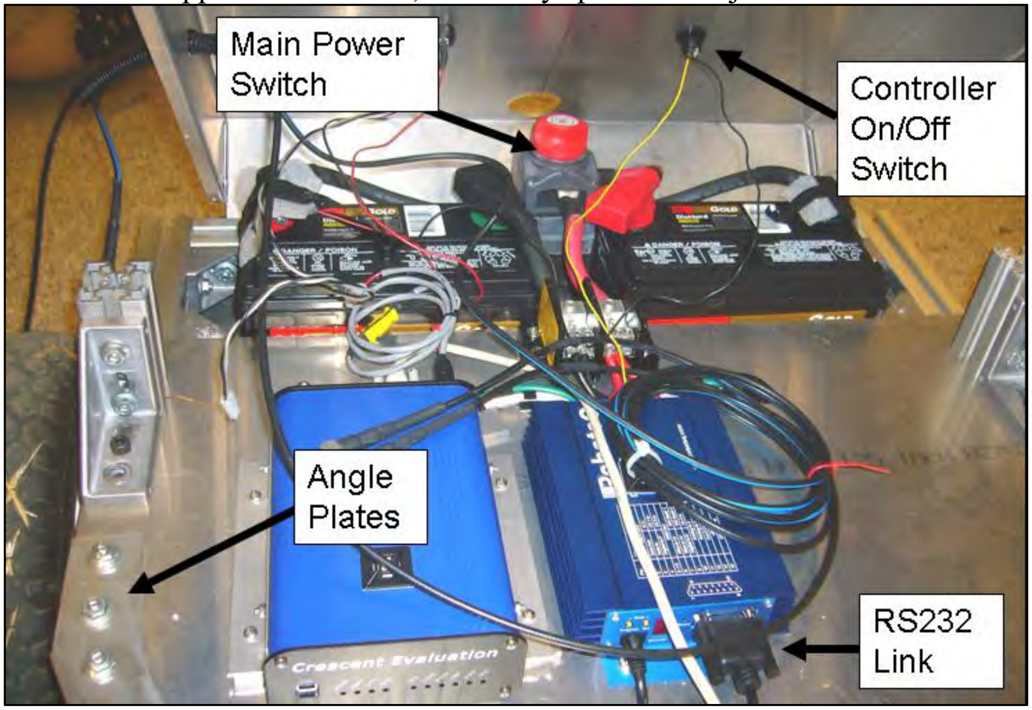


Figure A.7 Controller Layout

The interconnection of the two automotive DieHard batteries was accomplished with standard terminal posts and high current 00-gage stranded copper wire, Figure A.8.

The current rating for this wire was sufficient to meet the highest conceived total current draw for the entire system. The batteries are connected in series to yield a 24 volt supply which is passed through the main power switch as discussed above. To mount this power switch, an aluminum mounting block was machined. This block was bolted to the rear frame, and the power switch was then bolted to the top of the block for a secure fitting. A channel was machined from the bottom of the mounting block to secure the interconnecting battery wire.

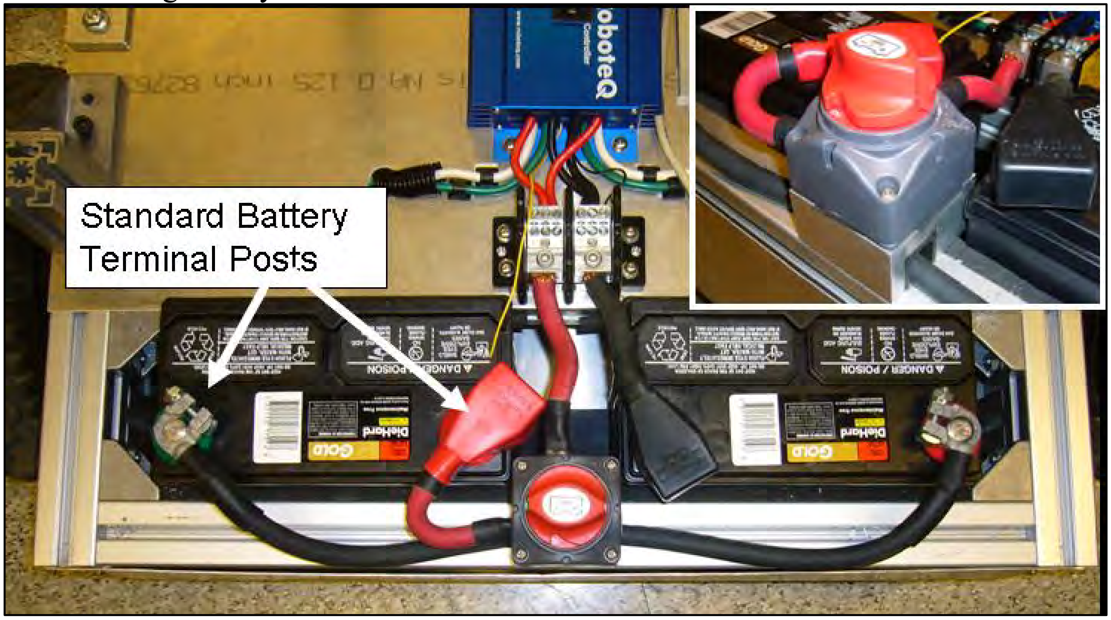


Figure A.8 Battery connection, power switch mounting block (upper-right)

To connect the 8 gauge controller wires to the motors, two sets of high current connectors were used (Figure A.9). Since the controller was mounted to the top of the aluminum mounting plate, two port holes were drilled through this medium for access to the motor connector.

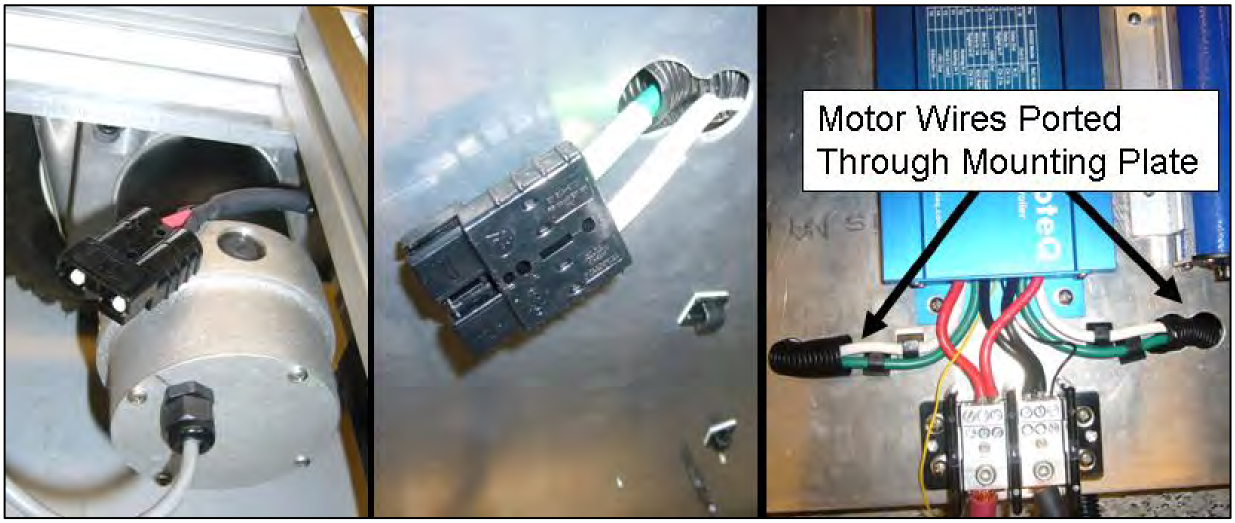


Figure A.9 Motor connector (left), controller connector (center), controller with ported wires (right)

Motor Controller Tuning:

The AX2850 motor controller makes use of a digital pulse width modulation (PWM) PID control algorithm for precise speed control. To improve the response of this algorithm it is suggested to manually tune the PID parameters for the application in questions. Many theoretical PID tuning techniques are available, such as those proposed by Ziegler and Nichols [73]. A more practical solution and the one suggested by the manufacturer [49] was manual online tuning by feel. This technique involves multiple trial and error testing with a range of varying parameters. The ultimate goal is to yield a solution that has a fast response time, low overshoot, and quick settling time. As listed in the vehicle specifications (Section 3.2.1), the platform should be able to reach the average speed of 1.5 mph (2.4 km/h) in less than one second. It is also desirable to have low oscillation, resulting in better tracking.

Since the platform will operate under varying loading situations, it was recommended [49] to tune the parameters with the minimum and then the maximum expected loading, and then to find a compromise between these two extreme cases. To achieve this, the testbed was first tuned on a flat surface with no additional loading. Then, to represent a rough approximation of the heavily loaded case, 60 lb (267 N) of weight was dragged on a rubber matt behind the platform, Figure A.10. The reference speed was set to 36.0 rpm since this was identified as the average operating speed.

Figure A.11 represents a typical response output plot (lines were added between points to suggest the overall trend). Since these plots quickly become cluttered only a few experimental trials are displayed in Figure A.11. After testing many parameter combinations, for both loading situations, a number of observations were made. At these relatively low speeds (only 36 rpm) the differential parameter had a strong effect on the stability of the control, and it was necessary to keep this value small. Increasing the differential gain should smooth out the response dynamics, however, this was not the case, as is typical for digital control at low speeds (due to discretized differentiation). Also, the rise time and settling time were essentially unchanged over a wide range of parameter values. The rise time was on the order 0.4 seconds for the unloaded case and only about 0.5 seconds for the fully loaded case, both of which are less than the required response time of 1 second. To further decrease the rise time, the controller has a built in programmable acceleration parameter. However, because it was desired to maintain minimal slippage during abrupt speed changes this value was left at a low setting. The overshoot was also relatively unchanged over a wide range of parameter values. After many trials, it was determined that values of 1.5, 1.5, and 2.0 were acceptable values for the proportional, integral, and derivative terms respectively, for both loading situations.

Figure A.10 PID tuning configuration with 60 lb (267 N) of weight pulled

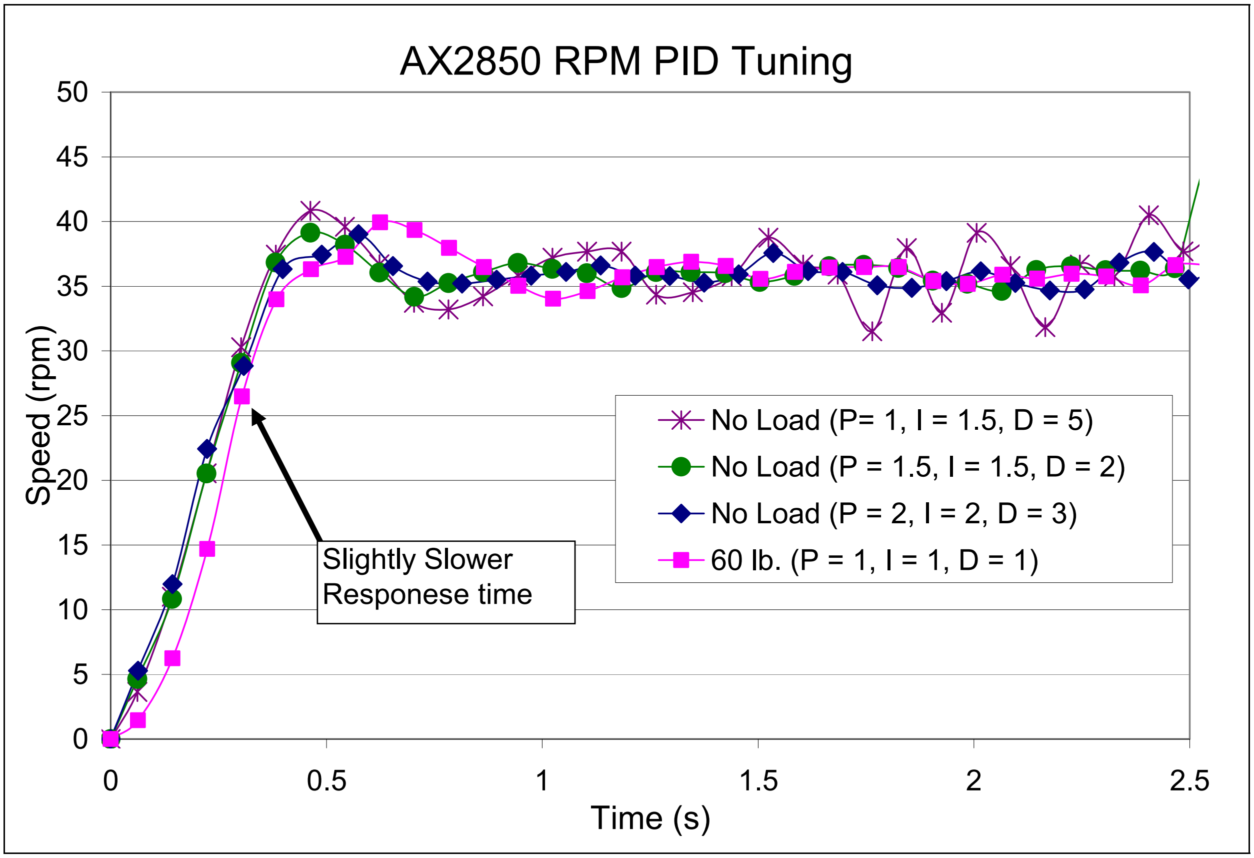


Figure A.11 Typical PID output response plot

PART 2 Appendix-B

Agent Software Overview:

To support the core concepts detailed in Chapter 4, an overview of the agent software/program architecture is discussed below. Then, later on in Appendix B.6 – B.45, the actual software coding is listed.

To assist in understanding the software architecture, a detailed class block diagram is displayed in Figure B.1. This schematic displays the relationship between the classes which make up the agent program. For simplicity, a few additional classes are not included, for instance, *MatrixF.cs*, which supports all matrix operations and *SocketListen.cs*, which supports wireless Ethernet communication. The classes outlined in Figure B.1 are grouped according to their overall function. Functions that include: main thread execution, motor control interfacing, GPS interfacing, state estimation, and path planning and tracking control. Discussion of the software architecture will be broken up based upon each individual class grouping (function).

Main Thread: Core software execution is handled by a single main thread, which is generated in the *Program.cs* class and utilizes a single subroutine, *pathFollow()*. This subroutine handles the highest level of control, namely, reading sensor measurements, interpreting their meaning, determining the appropriate actuation, and actually accomplishing actuation. Of course, all this can only be accomplished by utilizing a sophisticated array of supporting classes. To make use of these supporting classes, the *pathFollow()* subroutine generates a number of class object instances; including an instance of the *RoboteQdataProcessing.cs*, *PathControl.cs,* and *StateEstimation.cs* (also *GPSdataProcessing.cs* however this instance is never directly referenced).

The instance of the *RoboteQdataProcessing.cs* is used to link the main thread to the classes which support motor control. In this manner the main thread/subroutine has access to the encoder counts (stored by the motor controller) and also independent speed control over the right and left motor. Similarly, the instance of *PathControl.cs* links the main subroutine to the path planning and tracking control classes. Thus, when necessary, the main thread may request and retrieve the required tracking control inputs for the right and left wheel speed for the tracking of a planned path. Finally, the instance of *StateEstimation.cs* links the main subroutine to the stochastic state estimation classes. This allows encoder measurements, retrieved from the motor controller, to be passed to the state estimation classes.

Motor Controller Interface: Four distinct classes are utilized for the AX2850 motor controller interface: *RoboteQdata.cs, RoboteQdataProcessing.cs, SerialCommunication.cs*, and *GlobalTime.cs*. Core serial (RS232) communication to the AX2850 motor controller is handled by the *SerialCommunication.cs* class, which is built on top of Sax.NET supporting libraries. Sax.NET is an add-on software library package built for serial communication in the C# environment. To make use of the *SerialCommunication.cs* class, an object instance is generated in the *RoboteQdataProcessing.cs* class at the same time that the main thread (*pathFollow()* subroutine) generates an instance of the *RoboteQdataProcessing.cs* class. In this manner, byte commands can be sent directly from the *RoboteQdataProcessing.cs* object instance to the instance of *SerialCommunication.cs* and passed directly to the AX2850 motor controller.

The *SerialCommunication.cs* creates an event driven RS232 connection. Hence, when ASCII byte data is received from the motor controller, program execution is diverted to a corresponding subroutine (within *SerialCommunication.cs*) to handle the buffered data. Within this routine, a distinct time stamp is retrieved from the *GlobalTime.cs* class by means of a static subroutine. Hence, each data packet retrieved from the AX2850 motor controller is given its own individual time stamp. To process incoming data (from the AX2850 controller), buffered data, along with its corresponding time stamp is encapsulated within the *RoboteQdata.cs* class as instance objects, and passed to the *RoboteQdataProcessing.cs* object instance, for storage in a linked list. Routines within the *RoboteQdataProcessing.cs* are used to extract, interpret, and provide error checking for all incoming AX2850 controller data. For example, encoder counts can be extracted and passed back to the main thread.

GPS Interface: The GPS interface is set up in much the same manner as the Motor Controller Interface. A second instance of the *SerialCommunication.cs* class, which is generated within the *GPSdataProcessing.cs* constructor, is used to set up an event driven RS232 connection to the Crescent Vector OEM board. While the AX2850 controller is a polled device, the Vector Board was set up to send a continuous byte stream of measurements. Thus, while an instance of the *GPSdataProcessing.cs* class is generated in the main thread (*pathFollow()* subroutine), it is only used to set up the serial connection and is never referenced directly in the main thread.

Data received (from the GPS board) by the *SerialCommunication.cs* instance is encapsulated within *GPSdata.cs* instance objects, along with a time stamp retrieved from the *GlobalTime.cs* class, and passed forward to the *GPSdataProcessing.cs* object instance. Like the Motor Controller Interface, raw *GPSdata.cs* data passed to the *GPSdataProcessing.cs* class is processed and stored in a linked list as distinct positioning and heading measurements with a unique timestamp. As will be discussed next, the *Kalman.cs* class has static access to these measurements when they are needed.

State Estimation: State estimation is accomplished using two classes, *StateEstimation.cs* and *Kalman.cs*. An instance of the *Kalman.cs* class is generated within the constructor of *StateEstimation.cs*, hence, it is generated at the same point that the main thread generates an instance of *StateEstimation.cs*. For continual state estimation (pose estimation), encoder counts received from the AX2850 are passed to the *StateEstimation.cs* object instance and then on to the *Kalman.cs* object instance. The GPS measurements are also retrieved statically from the *GPSdataProcessing.cs* class and passed to the *Kalman.cs* object instance for data fusion. After a new state estimate has been generated, the value is passed back to the StateEstimation.cs class for storage in a linked list.

Path Planning and Tracking Control: Path planning and tracking control is accomplished by means of the following classes: *PathControl.cs, BuildPath.cs*, and *PathPlanning.cs*. Before any tracking control can be preformed, a distinct preplanned path is defined. This path is generated by making use of the static subroutine *planPath()* contained within the *PathPlanning.cs* class, which takes as input the user specified perimeter bounds and the desired cutting width. The generated path is stored in an instance of the *BuildPath.cs* class, as a series of consecutive straight line and turn trajectories (Chapter 4.4.2). To retrieve motor control data (i.e. V_R, V_L), the main thread utilizes an instance of the *PathControl.cs* class. This instance contains a subroutine *getSpeeds()* that uses the *BuildPath.cs* instance along with the *StateEstimation().cs* path history to retrieve appropriate motor control inputs, which are passed back to the main thread for handling.

For further insight into the overall outer control loop processing , see Chapter 4.4.

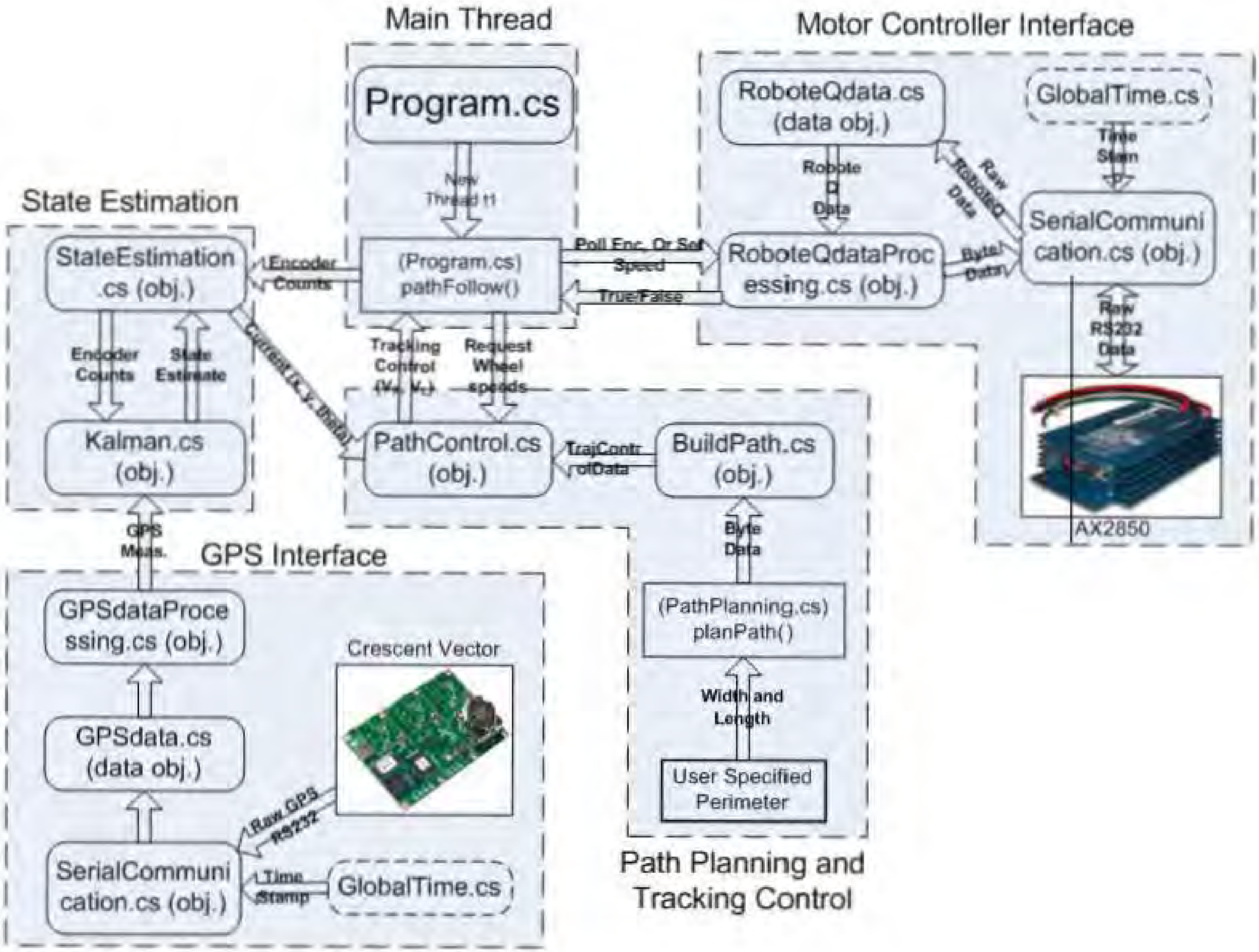


Figure B.1 Class Block Diagram for the agent software

Code Listing: A nearly complete agent software code listing is included here. The code listing is divided up by each distinct class. The actual class interaction is displayed in Figure B.1 and was discussed earlier.

Program.cs

```
enum SerialType { roboteQ, GPS, IMU, LIDAR }

namespace MowerControl
{
	class Program
	{
		public static sbyte speedL, speedR;
		public static bool startProgram = false;
		public static bool stopEverything = false;
		private static SocketListen socL;
		private static RoboteQdataProcessing rDP;
		private static GPSdataProcessing gpsDP;
		private static StateEstimation sE;

		static void Main(string[] args)
		{
			speedL = 0;                           //initialize the speeds to zero, these values hold the global speed
			speedR = 0;                           //note speed set to 73 yields 2.2 ft/sec (1.5 mph) or 36 rpm
			GlobalTime.startGlobalTime();         //This will start the global Referenced Time
			gpsDP = new GPSdataProcessing();      //Generate an instance of GPSdataProcessing Class for serial data interface
			rDP = new RoboteQdataProcessing(); //Generate an instance of RoboteQdataProcessing for serial motor control
			socL = new SocketListen();           //Create an instance of SocketListen
			socL.StartListening();               //Start listening for an incoming socket connection (event driven)

			//Start of by calibrating GPS measurements by averaging over multiple measurements
			do
			{
				Thread.Sleep(50);
			} while (GPSdataProcessing.inCal == true);

			Thread t1 = new Thread(pathFollow);      //Create a thread for the overall control execution
			t1.Start();                              //Start the thread for path following
		}

		//Main Control Loop to Follow a Defined Path
		pulic static void pathFollow()
		{
			Location curLoc;                      //Holds The Temporary Present Location
			int rEnc = 0;
			int lEnc = 0;
			Time timeStamp = new Time();
			bool wasRead, wasSet;

			//call the readEncoders routine to zero out the encoders
```

```
wasRead = rDP.readEncoders(ref lEnc, ref rEnc, EncType.rel, ref timeStamp);
Console.WriteLine("Poll Encoders: " + wasRead + "\nEncoder Left is: " + lEnc + " Encoder Right is: " + rEnc);
Thread.Sleep(100);                  //Small Wait after Reading Encoders

//Note: State Estimation Class Holds Estimate Of the Wheel Base Center Point,
//While Path Control Workes Off of Tracking Point 16-inches forwards of wheel baseline
double x0, y0, theta;                             //Starting x,y,theta location
theta = GPSdataProcessing.getCaliHeading();       //Theta will be starting heading, East = 0-deg, North = 90-deg
x0 = -16 * Math.Cos(theta*Math.PI/180.0);         //16" offet of tracking point forward of rear baseline
y0 = -16 * Math.Sin(theta*Math.PI/180.0);

//Generate an Instance of StateEstimation for Continual EKF pose estimation
sE = new StateEstimation(new Location(x0,y0,0.0,theta,GlobalTime.getTime().timeIs(),0,0,0,0,0));

//Build the Path Given the Rectangular Bounds (currently 1800x470 in rectangular perimieter)
BuildPath pathToFollow = PathPlanning.planPath(GlobalTime.getTime().timeIs(), theta, 1800,470,0,0,36.0);

//Define a new pathControl with Kx and Ky gains, along with the tracking point offset
PathControl pC = new PathControl(16.0, .2, .2);

//Main path following control loop
do
{
       //read the encoder counts
       wasRead = rDP.readEncoders(ref lEnc, ref rEnc, EncType.rel, ref timeStamp);
       if (wasRead == false)
              Console.WriteLine("Problem reading Encoders");

       sE.updateSE(lEnc, rEnc, timeStamp);     //Update the state estimation using the EKF
       curLoc = sE.getLastLoc();               //retreive the current location estimation
       Thread.Sleep(40);                       //small wait after reading the encoders to free up the serial line

       //based on location determine the control speeds,
       pC.getSpeeds(ref speedL, ref speedR, curLoc, pathToFollow);

       //Use these determined control speeds to set the wheel speeds
       wasSet = rDP.setSpeed(speedL, speedR);
       if (wasSet == false)
              Console.WriteLine("problem Setting speed");

       Thread.Sleep(50);                       //Small wait after setting speeds to free up the serial line
} while (GlobalTime.getTime().timeIs() <= (BuildPath.endTimeIs()) && stopEverything == false);


speedL = 0; speedR = 0;                                                //Zero The speeds to stop the platform
       Console.WriteLine("Set Speed To: " + speedL + " and " + speedR + "  Was success: "
                           + rDP.setSpeed(speedL, speedR));

//Print Out the State Estimation and GPS Values
Program.printReport();
```

```
             }

             //Routine to Write State Estimation and GPS data to file
             private static void printReport()
             {
                    FileInfo dataHis = new FileInfo(@"C:\Documents and Settings\Aaron\Desktop\DataHis.xls");
                    StreamWriter outData = dataHis.CreateText();
                    ArrayList dataHistory = sE.getHistory();                    //Retrieve the State estimation History
                    outData.WriteLine("RefHead: \t" + GPSdataProcessing.refHead + "\tRefLong: \t" + GPSdataProcessing.refLong +
                                      "\tRefLat: \t" + GPSdataProcessing.refLat + "\n");
                    outData.WriteLine("Time\tXt\tYt\tTheta\tEncL\tEncR\tXc\tYc");
                    for (int j = 0; j < dataHistory.Count; j++)
                    {
                           //Note: xTemp and yTemp are the Tracking point position, and xTemp2 and yTemp2
                           //are the wheel baseline position
                           double thetaTemp = ((Location)dataHistory[j]).theta;
                           double xTemp = ((Location)dataHistory[j]).x + 16.0 * Math.Cos(thetaTemp * Math.PI / 180.0);
                           double yTemp = ((Location)dataHistory[j]).y + 16.0 * Math.Sin(thetaTemp * Math.PI / 180.0);
                           double xTemp2 = ((Location)dataHistory[j]).x;
                           double yTemp2 = ((Location)dataHistory[j]).y;
                           outData.WriteLine(((Location)dataHistory[j]).time + "\t" + xTemp + "\t" + yTemp + "\t" + thetaTemp + "\t"
                                               + ((Location)dataHistory[j]).encoderL + "\t" + ((Location)dataHistory[j]).encoderR +
                                                "\t" + xTemp2 + "\t" + yTemp2);
                    }

                    //Now print out the GPS data history in the same file
                    ArrayList dataHistory2 = GPSdataProcessing.measHisSaved;    //Retrieve the Raw Crescent Vector Data
                    outData.WriteLine("Time:\tXt:\t" + "Yt:\tXc:\tYc\t" + "Head:\t" + "Lat:\tLong:\tLatError:\tLongError:");
                    for (int t = 0; t < dataHistory2.Count; t++)
                    {
                           //Note: xGPS and yGPS are the tracking point while xGPS2 and yGPS2 are the wheel baseline point
                           double xGPS = ((GPSmeasurement)dataHistory2[t]).x + 16 *
                                               Math.Cos(((GPSmeasurement)dataHistory2[t]).headRad);
                           double yGPS = ((GPSmeasurement)dataHistory2[t]).y + 16 *
                                               Math.Sin(((GPSmeasurement)dataHistory2[t]).headRad);
                           double xGPS2 = ((GPSmeasurement)dataHistory2[t]).x;
                           double yGPS2 = ((GPSmeasurement)dataHistory2[t]).y;
                           outData.WriteLine(((GPSmeasurement)dataHistory2[t]).time + "\t" + xGPS + "\t" + yGPS + "\t" + xGPS2 +
                                               "\t" + yGPS2 + "\t" + ((GPSmeasurement)dataHistory2[t]).heading + "\t" +
                                               ((GPSmeasurement)dataHistory2[t]).latitude + "\t" +
                                               ((GPSmeasurement)dataHistory2[t]).longitude + "\t" +
                                               ((GPSmeasurement)dataHistory2[t]).latError + "\t" +
                                               ((GPSmeasurement)dataHistory2[t]).longError);
                    }
                    outData.Close();
             }
      }
}
```

RoboteQdataProcessing.cs

```
enum Command { setSpeed, readEncoders, unknown}
enum EncType { rel, abs, relSum }

namespace MowerControl
{
    class RoboteQdataProcessing
    {
        //2 ArrayLists for sent and received data
        public static ArrayList sentData;
        public static ArrayList receivedData;
        //Store an Instance of serialCommunication
        private SerialCommunication dataLine;

        //Constructor for this class
        public RoboteQdataProcessing()
        {
            sentData = new ArrayList();
            receivedData = new ArrayList();
            dataLine = new SerialCommunication(SerialType.roboteQ);  //0=communication to roboteq controller
        }

        //Routine to Poll and retrieve the Encoder Counts
        public bool readEncoders(ref int encL, ref int encR, EncType eType, ref Time time)
        {
            //clear array lists so that no bad previous data is accumulated
            clearData();
            byte [] byteData;

            if (eType == EncType.rel)
               byteData = stringToByte("?Q4\r?Q5\r");
            else if (eType == EncType.abs)
               byteData = stringToByte("?Q0\r?Q1\r");
            else
               byteData = stringToByte("?Q2\r");

            //send the above commands to retrieve the encoder counts
            dataLine.send_Bytes(byteData);

            //Mark time when data is sent
            Time tempTime = GlobalTime.getTime();
            sentData.Add(new RoboteQdata(byteData,tempTime, Command.readEncoders));

            do //wait for data coming in
            {
                Thread.Sleep(10);
            } while (!dataLine.isThereNewData());

            Time tempTime2 = GlobalTime.getTime();  //store the time when all encoder data is in by
```

```
            time = tempTime.AverageTime(tempTime2);  //average the time stamps from when data sent to when all received
                                                             (~10 ms accuracy)

             //read the encoder counts
            return encoderVals(eType, ref encL, ref encR);
        }

        //routine used to compare the input and output data streams from the RoboteQ controller
        private bool encoderVals(EncType encT, ref int encL, ref int encR)
        {
            RoboteQdata dataIn = ((RoboteQdata)sentData[0]);
            RoboteQdata dataOut = ((RoboteQdata)receivedData[0]);
            bool testCase = false;
            byte i = 0;
            byte toCont = 0;
            byte fromCont = 0;
            ArrayList leftVal = new ArrayList();
            ArrayList rightVal = new ArrayList();

            //now loop through all the bytes
            do
            {
                //if read all data then remove node and get new one
                if (dataOut.getByteSize() == 0)
                {
                    receivedData.RemoveAt(0);
                    if (receivedData.Count == 0)  //if no new data wait some time for buffer to fill up
                    {
                        int t = 0;
                        do
                        {
                            Thread.Sleep(10);
                            if (receivedData.Count != 0)
                                break;
                            t++;
                        } while (t < 10);

                        //now verify wheater or not the buffer has new data
                        if (receivedData.Count == 0)
                        {
                            encL = 0;
                            encR = 0;
                            clearData();
                            return false;  //then no luck getting data and exit
                       }
                       else
                           dataOut = ((RoboteQdata)receivedData[0]);
                    }
                    else
                        dataOut = ((RoboteQdata)receivedData[0]);  //set stream to new data object
```

```
             }

             if (i < 4)  //note at i = 3 then the value should be \r for toCont ex. ?Q4\r?Q5\r is sent 4th character
             {
                 toCont = dataIn.getData();
                 fromCont = dataOut.getData();
             }
             else
                 fromCont = dataOut.getData();

            if (toCont == fromCont)
            {
                if (i < 4)
                {
                    dataIn.removeByte();  //note toCont will still hold its last value of \r
                    dataOut.removeByte();
                }
                else
                {
                    dataOut.removeByte();  //only when both hit \r
                    testCase = true;
                }
            }
            else
            {
                if (i <= 3) //in this case the data wasn't sent properly
                {
                    encL = 0;
                    encR = 0;
                    clearData();
                    return false;
                }
                else
                {
                    leftVal.Add(fromCont);
                    dataOut.removeByte();
                }
            }
            i++;
        } while (testCase == false);

        if (encT == EncType.relSum)
        {
            int sum = hexToInt(leftVal);
            encL = sum / 2;
            encR = sum / 2;
            return true;
        }
        else  //is not sum and need to get right encoder count also
        {
```

```
//reinitialize
testCase = false;
i = 0;
//now loop through all the remaining bytes
do
{
    if (dataOut.getByteSize() == 0)
    {
        receivedData.RemoveAt(0);
        if (receivedData.Count == 0)
        {
            int t = 0;
            do
            {
                Thread.Sleep(10);
                if (receivedData.Count != 0)
                    break;
                t++;
            } while (t < 10);
            if (receivedData.Count == 0)
            {
                encL = 0;
                encR = 0;
                clearData();
                return false;
            }
            else
                dataOut = ((RoboteQdata)receivedData[0]);
        }
        else
            dataOut = ((RoboteQdata)receivedData[0]);  //set stream to new data object
    }

    if (i < 4)  //note at i = 3 then the value should be \r for toCont
    {
        toCont = dataIn.getData();
        fromCont = dataOut.getData();
    }
    else
        fromCont = dataOut.getData();

    if (toCont == fromCont)
    {
        if (i < 4)
        {
            dataIn.removeByte();
            dataOut.removeByte();
        }
        else //i is >= 4
        {
```

```
                            dataOut.removeByte();
                            testCase = true;  //now holds the second encoder value
                        }
                    }
                    else
                    {
                        if (i <= 3) //in this case the data wasn't sent properly
                        {
                            encL = 0;
                            encR = 0;
                            clearData();
                            return false;
                        }
                        else
                        {
                            rightVal.Add(fromCont);
                            dataOut.removeByte();
                        }
                    }
                    i++;
                } while (testCase == false);
            }
            encL = hexToInt(leftVal);
            encR = -1*hexToInt(rightVal);  //note negated since motor is opposite side
            clearData();
            return true;
        }

        //routine to cear the data history
        private void clearData()
        {
            sentData.Clear();
            receivedData.Clear();
        }

        //Routine to Set the Motor Speed, returns true if speed is set
        public bool setSpeed(sbyte leftS, sbyte rightS)
        {
            //clear array lists so that no bad previous data is accumulated
            clearData();
            //sbyte varries from -128 to 127
            String data = "";
            if (leftS < 0)  //note channel-1 is left so correct
                data = "!a" + toHex(Math.Abs(leftS));
            else
                data = "!A" + toHex(leftS);

            //note B and b reversed since motor is on oposite side
            if (rightS < 0)
                data = data + "\r!B" + toHex(Math.Abs(rightS));
```

```
        else
            data = data + "\r!b" + toHex(rightS);

        data = data + "\r";
        //now determine the byte value
        byte[] byteData = stringToByte(data);
        dataLine.send_Bytes(byteData);

        sentData.Add(new RoboteQdata(byteData, GlobalTime.getTime(), Command.setSpeed));  //note 0 corresponds to indicator
                                                                                           of setting speed
        do
        {
            Thread.Sleep(10);
        } while (!dataLine.isThereNewData());
        //determine if the speeds were set correctly
        return analyseSetData(Command.setSpeed);
    }

    //routine used to compare the input and ouput data streams from the RoboteQ controller
    private bool analyseSetData(Command com)
    {
        bool value = false;
        if(sentData.Count >0)
        {
            //if it is a set motor command then execute here
            if (com == Command.setSpeed)
                value = matchForPlus();
        }
        clearData();  //clear the data history
        return value;
    }

    //routine to analyse the byte stored byte streams and look for the "+" value
    private bool matchForPlus()
    {
        RoboteQdata dataIn = ((RoboteQdata)sentData[0]);
        RoboteQdata dataOut = ((RoboteQdata)receivedData[0]);
        //now loop through all the bytes
        if (dataIn.getByteSize() == (dataOut.getByteSize() - 4))  //if all data is in this case should be true
        {
            do
            {
                byte toCont = dataIn.getData();
                byte fromCont = dataOut.getData();

                if (toCont == fromCont)
                {
                    dataIn.removeByte();
                    dataOut.removeByte();
                }
```

```
                else if (fromCont == 43)  //in this case find a '+'
                {
                    dataOut.removeByte();
                    dataOut.removeByte();
                }
                else
                    return false;
            } while (dataIn.getByteSize() > 0);
            return true;
        }
        else //in this case wait some more time for more data to come in if necessary
        {
            do
            {
                if (dataOut.getByteSize() == 0)
                {
                    receivedData.RemoveAt(0);
                    if (receivedData.Count == 0)
                    {
                        int t = 0;
                        do
                        {
                            Thread.Sleep(10);
                            if (receivedData.Count != 0)
                                break;
                            t++;
                        } while (t < 10);

                        //now verify wheater or not the buffer has new data
                        if (receivedData.Count == 0)
                            return false;
                        else
                            dataOut = ((RoboteQdata)receivedData[0]);
                    }
                    else
                        dataOut = ((RoboteQdata)receivedData[0]);
                }
                byte toCont = dataIn.getData();
                byte fromCont = dataOut.getData();

                if (toCont == fromCont)
                {
                    dataIn.removeByte();
                    dataOut.removeByte();
                }
                else if (fromCont == 43)  //in this case find a '+'
                {
                    dataOut.removeByte();
                    dataOut.removeByte();
                }
```

```
                    else
                        return false;
                } while (dataIn.getByteSize() > 0);
                return true;
            }
        }

    }
}
```

RoboteQdata.cs

```
namespace MowerControl
{
    class RoboteQdata
    {
        //array list "data" stores all the byte values specified
        private ArrayList data = new ArrayList();
        private Time timeStamp;
        private Command type;

        public RoboteQdata(byte[] data, Time timeStamp, Command type)
        {
            byte [] dataTemp = data;
            for (int i = 0; i < data.Length; i++)
                this.data.Add((byte)dataTemp[i]);
            this.type = type;
            this.timeStamp = timeStamp;
        }
        public Time getTimeStamp()
        {
            return timeStamp;
        }
        public byte getData()
        {
            if (data == null)
                return 0;
            else
                return (byte)data[0];
        }
        public void removeByte()
        {
            data.RemoveAt(0);
        }
        public Command getType()
        {
            return type;
        }
        public int getByteSize()
        {
```

```
            return data.Count;
        }
    }
}
```

GPSdataProcessing.cs

```
enum GPSmessage { latLon, heading, unknown }

namespace MowerControl
{
    class GPSdataProcessing
    {
        public static ArrayList measHistory;      //GPS measurement Data history
        public static ArrayList measHisSaved;     //Saved GPS measurement history for printout
        private static ArrayList receivedData;    //Store Incoming Byte Data
        private SerialCommunication dataLine;     //Store an Instance of serialCommunication
        private const double a = 6378137.00;       //Earths semi-major axis in meters
        private const double b = 6356752.31;       //Earths semi-minor axis in meters
        private const double f_1 = 298.257223563;
        private const double e_2 = .00669437999014;
        public static double refLat;              //reference latitude, longitude and heading
        public static double refLong;
        public static double refHead;
        private static double Rn, Rm;            //Store the radii of curvature in prime vertical and meridian
        public static bool inCal;
        private static double latTotal, longTotal, headTotal;
        private static int calIterator;
        private static int numbCalCycles;

        //Construtor for this class
        public GPSdataProcessing()
        {
            measHistory = new ArrayList();
            measHisSaved = new ArrayList();
            receivedData = new ArrayList();
            dataLine = new SerialCommunication(SerialType.GPS);
            inCal = true;  calIterator = 0; latTotal = 0; longTotal = 0;
            refLat = 0; refLong = 0; refHead = 0;
            numbCalCycles = 20;                   //Number of calibration readings to average
        }

        //Routine to Poll the GPS unit, get data back and convert to X, Y, and theta coordinates
        public static void calculateGPSmeas(GPSdata data)
        {
            //First Store the New data, then analyze it
            receivedData.Add(data);
            double heading, latitude, longitude, latError, longError;  //store the GPS values

            if (data.getData() != 36)  //if first byte is not $ then exit with false
```

```
            return;
        else
        {
            //First Pull out Heading From the HPR Data Message
            int i = 0;
            do { data.removeByte(); i++; } while (i < 10);  //remove bytes to get to time

            if (data.getData() == 44)
                return;
            else
            { do { data.removeByte(); } while (data.getData() != 44);}

            //Now Should be at the heading value
            data.removeByte();  //remove the comma before the heading value
            //now get the heading value
            heading = 0;
            while (data.getData() != 46)  //note 46 is . in ASCII
            {
                int temp = (int)(data.getData() - 48);
                data.removeByte();
                heading = heading * 10 + temp;
            }
            data.removeByte();  //remove the decimal point and now add the heading fraction
            double div = 10;
            while (data.getData() != 44)
            {
                double temp = ((int)(data.getData() - 48)) / div;
                data.removeByte();
                heading += temp;
                div *= 10;
            }

            //Convert Heading into ENC so that zero is East and North is 90-degrees
            heading = heading + 180.0;
            if (heading > 0.0 && heading < 270.0)
                heading = 90 - heading;
            else
                heading = 450 - heading;

            //Now remove bytes until start of the GLL message
            while (data.getData() != 36) { data.removeByte(); }
            i = 0;
            //Now iterate until at the latitude, remove 7-bytes to get to latitude
            while (i < 7) { data.removeByte(); i++; }

            if (data.getData() == 44)
                return;
            latitude = 0;  //initialize latitude to zero
            i = 0;      //Note: no separation between the degrees and the minutes so have to use 2
            while (i < 2)  //46=.
```

```
{
    int temp = (int)(data.getData() - 48);
    data.removeByte();
    latitude = latitude * 10 + temp;
    i++;
}
double min = 0.0;              //will hold the minutes and decimal-minutes
while (data.getData() != 46)
{
    int temp = (int)(data.getData() - 48);
    data.removeByte();
    min = min * 10 + temp;
}
data.removeByte();      //remove the period before the decimal-minutes
div = 10;
while (data.getData() != 44)  //go till the end of the latitude value
{
    double temp = ((int)(data.getData() - 48)) / div;
    data.removeByte();
    min += temp;
    div *= 10;
}

//Now calculate the final value for latitude
latitude = latitude + min / 60.0;

//now remove bytes until get to the longitude value
i = 0;
while (i < 3) { data.removeByte(); i++; }  //take 3-bytes out to get to the longitude

//Now pull out the longitude value
if (data.getData() == 44)  //if not value then
    return;
longitude = 0;  //initialize latitude to zero
i = 0;
while (i < 3)  //46=.
{
    int temp = (int)(data.getData() - 48);
    data.removeByte();
    longitude = longitude * 10 + temp;
    i++;
}
min = 0.0; //will hold the minutes and decimal-minutes
while (data.getData() != 46)
{
    int temp = (int)(data.getData() - 48);
    data.removeByte();
    min = min * 10 + temp;
}
data.removeByte();  //remove the period before the decimal-minutes
```

```
            div = 10;
            while (data.getData() != 44)  //go till the end of the latitude value
            {
                double temp = ((int)(data.getData() - 48)) / div;
                data.removeByte();
                min += temp;
                div *= 10;
            }
            longitude = -1.0*(longitude + min / 60.0);  //note minus because west on the globe

            //Now Get the error statistics for the lat and long error in meters,
             //Remove bytes till get to the GST Message
            latError = 0; longError = 0;
            while (data.getData() != 36) { data.removeByte(); }
            i = 0;
            //Now iterate until at the latitude lat, long error statistics
            while (i < 6)
            {
                if (data.getData() == 44)
                    i++;
                data.removeByte();
            }

            latError += (data.getData() - 48);
            data.removeByte(); data.removeByte();
            latError += (data.getData() - 48) / 10.0;
            data.removeByte(); data.removeByte();
            longError += (data.getData() - 48);
            data.removeByte(); data.removeByte();
            longError += (data.getData() - 48) / 10.0;
        }
        //Store new MEasurement
        GPSmeasurement newGPSmeas =  new GPSmeasurement(latitude, longitude, heading,
                                                               data.getTimeStamp().timeIs(),latError,longError);

        //If in calibration mode then find average, else store in GPS meas. History
        if (inCal == true)
            calibrate(newGPSmeas);
        else
            wgsToENU(newGPSmeas);
    }

    //convert from WGS to an East nort Up coordinate system
    private static void wgsToENU(GPSmeasurement wgsIn)
    {
        double x = Rn * Math.Cos(refLat) * (wgsIn.longRad-refLong);
        double y = Rm * (wgsIn.latRad-refLat);

        //Convert To inches and reflect to the center of the robot
        x = x * 39.3700787402;
```

```
            y = y * 39.3700787402;
            double antOffset = .4 * 39.3700787402;  //note - if primary is left, and + if primary is right
                                                        .8-meters is the total antena length so half
            //Account for antena offset and also shift coordinates so center is zero
            x = x + antOffset * Math.Sin(wgsIn.headRad)- antOffset * Math.Sin(refHead * Math.PI / 180.0);
            y = y - antOffset * Math.Cos(wgsIn.headRad)+ antOffset * Math.Cos(refHead * Math.PI / 180.0);
            //Because the tracking point is forward of state estimation shift baseline point back
            x = x - 16 * Math.Cos(refHead* Math.PI/180.0);         //16" because offet of tracking point
            y = y - 16 * Math.Sin(refHead * Math.PI/180.0);
            wgsIn.x = x;
            wgsIn.y = y;
            measHistory.Add(wgsIn);
            measHisSaved.Add(wgsIn);
            System.Console.WriteLine("Head: " + wgsIn.heading + "\nx: " + wgsIn.x + "\ny: " + wgsIn.y +
                                 "\nLongError: " + wgsIn.longError + "\nLatError: " + wgsIn.latError);
        }

        //Routine used for Calibration
        private static void calibrate(GPSmeasurement wgsIn)
        {
            calIterator++;
            if (calIterator == (numbCalCycles+1))
            {
                //Calculate the Radii
                refLong = longTotal / numbCalCycles;
                refLat = latTotal / numbCalCycles;
                refHead = headTotal / numbCalCycles;
                double temp = (1 - e_2 * Math.Sin(refLat) * Math.Sin(refLat));
                Rn = a / (Math.Sqrt(temp));
                Rm = (a * (1 - e_2)) / (Math.Pow(temp, 1.5));
                inCal = false;
            }
            else
            {
                longTotal += wgsIn.longRad;
                latTotal += wgsIn.latRad;
                headTotal += wgsIn.heading;
            }
        }
        //Routine to get the First measurement History value
        public static GPSmeasurement getFirstMeas()
        {
            return (GPSmeasurement)measHistory[0];
        }
        //Routine to remove the first Measurement History Data
        public static void removeFirstMeas()
        {
            measHistory.RemoveAt(0);
        }
        //Routine to get the Measurement History Count
```

```
        public static int getMeasCount()
        {
            return measHistory.Count;
        }
        //To get the calibrated heading use this subroutine
        public static double getCaliHeading()
        {
            return refHead;
        }
        //routine to convert a string to an appropriate byte array
        private byte[] stringToByte(String data)
        {
            StringReader reader = new StringReader(data);
            MemoryStream writer = new MemoryStream(data.Length);
            bool finished = false;

            while (!finished)
            {
                int c = reader.Read();
                if (c == -1)
                    finished = true;
                else
                    writer.WriteByte((byte)c);
            }
            return writer.ToArray();
        }
    }

    //Just a structure to hold LAT, LON, Heading data
    public struct GPSmeasurement
    {
        public double latitude, longitude, heading, time;
        public double latRad, longRad, headRad, longError, latError;
        public double x, y, z;
        public GPSmeasurement(double lat, double lon, double head, double time, double latError, double longError)
        {
            this.latitude = lat;
            this.latRad = lat * Math.PI / 180.0;
            this.longitude = lon;
            this.longRad = lon * Math.PI / 180.0;
            this.heading = head;
            this.headRad = head * Math.PI / 180.0;
            this.time = time;
            x = 0;y = 0;z = 0;
            this.latError = latError; this.longError = longError;
        }
    }
}
```

GPSdata.cs

```
namespace MowerControl
{
    class GPSdata  //this class is very similar to the RoboteQdata.cs class
    {
        //array list "data" stores all the byte values specified
        private ArrayList data = new ArrayList();
        private Time timeStamp;
        private GPSmessage type;

        public GPSdata(byte[] data, Time timeStamp, GPSmessage type)
        {
            byte [] dataTemp = data;
            for (int i = 0; i < data.Length; i++)
                this.data.Add((byte)dataTemp[i]);
            this.type = type;
            this.timeStamp = timeStamp;
        }
        public Time getTimeStamp()
        {
            return timeStamp;
        }
        public byte getData()
        {
            if (data == null)
                return 0;
            else
                return (byte)data[0];
        }
        public void removeByte()
        {
            data.RemoveAt(0);
        }
        public GPSmessage getType()
        {
            return type;
        }
        public int getByteSize()
        {
            return data.Count;
        }
    }
}
```

SerialCommunication.cs

```
namespace MowerControl
{
    class SerialCommunication
    {
```

```
        //set up a Sax communication variable
        private Sax.Communications.SerialConnection serialConnection;
        private SerialType deviceType;
        private bool newData = false;

        //Constructor to setup and open a connection with roboteQ, GPS, IMU etc...
        public SerialCommunication(SerialType type)
        {
            deviceType = type;
            if (deviceType == SerialType.roboteQ)
                roboteQserial();
            else if (deviceType == SerialType.GPS)
                gpsSerial();
        }
        //sets up a specific connection for the roboteQ controller
        private void roboteQserial()
        {
            serialConnection = new SerialConnection();
            serialConnection.DataAvailable += new
                                                Sax.Communications.SerialConnection.DataAvailableEventHandler(roboteQdataAvailable);

                    //Close any ongoing connection if open
            if (serialConnection.IsOpen)
                serialConnection.Close();
            //Now set the parameters to communicate with the RoboteQ controller and open a connection
            try
            {
                serialConnection.Options = new SerialOptions("COM10", 9600, Parity.Even, 7, CommStopBits.One, false,
                                                             false, false, false, false, false);
                //now open the connection
                serialConnection.Open();
            }
            catch (Exception exception)
            {
                Console.WriteLine("{0} Exception caught.", exception);
            }
        }

        //sets up a specific connection for the GPS board
        private void gpsSerial()
        {
            serialConnection = new SerialConnection();
            serialConnection.DataAvailable += new Sax.Communications.SerialConnection.DataAvailableEventHandler(gpsDataAvailable);

             //Close any ongoing connection if open
            if (serialConnection.IsOpen)
                serialConnection.Close();
            //Now set the parameters to communicate with the RoboteQ controller and open a connection
            try
            {
```

```
                serialConnection.Options = new SerialOptions("COM9", 57600, Parity.None, 8,
                                                      CommStopBits.One,false,false,false,false,false,false);
                //now open the connection
                serialConnection.Open();
            }
            catch (Exception exception)
            {
                Console.WriteLine("{0} Exception caught.", exception);
            }
        }

        //Subroutine to send bytes of data on the configured serial port
        public void send_Bytes(byte[] input)
        {
            serialConnection.Write(input, 0, input.Length);
            newData = false;
        }

        //Handles the data available in the buffer event for the roboteq motor controller
        public void roboteQdataAvailable(object sender, EventArgs e)
        {
            Time timeStamp = GlobalTime.getTime();    //store the time when data comes in
            Thread.Sleep(30);                         //wait 20-ms, this should provide the required time to get all data
            int nBytes = serialConnection.Available;
            byte[] buffer = new byte[nBytes];
            serialConnection.Read(buffer, 0, buffer.Length);

             //store data into a linked list for later analysis
            RoboteQdataProcessing.receivedData.Add(new RoboteQdata(buffer, timeStamp, Command.unknown));
            newData = true;                           //to assist in verifying data this variable indicates when data is received
        }

        //Handles the data available in the buffer event for GPS unit
        public void gpsDataAvailable(object sender, EventArgs e)
        {
            Time timeStamp = GlobalTime.getTime();    //store the time when data first arrives
            Thread.Sleep(100);                        //wait 100-ms, this will provide the required time to get all measurement data
            int nBytes = serialConnection.Available;
            byte[] buffer = new byte[nBytes];
            serialConnection.Read(buffer, 0, buffer.Length);
            GPSdataProcessing.calculateGPSmeas(new GPSdata(buffer, timeStamp, GPSmessage.unknown));
        }

        //Indicates if there is new data processed
        public bool isThereNewData()
        {
            return newData;
        }
    }
}
```

GlobalTime.cs

```
namespace MowerControl
{
    class GlobalTime
    {
        [DllImport("kernel32.dll")]
        extern static short QueryPerformanceCounter(ref long x);
        [DllImport("kernel32.dll")]
        extern static short QueryPerformanceFrequency(ref long x);

        private static long start = 0, end = 0;
        private static long freq = 0;

        //initialize timer when start the program, this will begin time and set the Freq.
        public static void startGlobalTime()
        {
            QueryPerformanceCounter(ref start);
            QueryPerformanceFrequency(ref freq);
        }
             //Subroutine to return the current time
        public static Time getTime()
        {
            QueryPerformanceCounter(ref end);
            int sec = (int)((end - start) * 1.0 / freq);
            long dif = (end-start)-freq*sec;
            return (new Time(sec,dif*1.0/freq));
        }
    }

       //Struture to hold a time stamp in seconds and its remainder
    public struct Time
    {
        public int seconds;
        public double remainder;
        public Time(int s, double r)
        {
            seconds = s;
            remainder = r;
        }
        public override String ToString()
        {
            return ((double)(seconds + remainder)).ToString();
        }
        public Time AverageTime(Time t1)
        {
            double avg = (this.seconds + this.remainder + t1.seconds + t1.remainder)/2.0;
            int sec = (int)avg;
            return (new Time(sec, (avg- 1.0 * sec)));
```

```
        }
        public double timeIs()
        {
            return this.seconds + this.remainder;
        }
    }
}
```

PathPlanning.cs

```
namespace MowerControl
{
    class PathPlanning
    {
        private const double fullSpeed = 26.4;  //inches per second = 1.5MPH
        private const double turnSpeed = 12.0;  //Turn Speed, arbitrary
        private const double slowSpeed = 12.0;  //inches per second
        private const double stopSlow = 5.0;
        private const double stopSpeed = 1.0;   //Almost stop

        //static routine that takes the coordinates of the rectangular area to cover and returns the full
       //path as a series of lines and arcs
        public static BuildPath planPath(double t0, double thetaDeg, double pathLength, double pathWidth, double xo, double
                                         yo,double cuttingWidth)
        {
            double x1, y1, x2, y2;
            BuildPath path;
            Trajectory traj1 = new Trajectory();
            Trajectory traj2 = new Trajectory();
            double theta = thetaDeg * Math.PI/180.0;
            double tempAng = theta - Math.PI / 2.0;
            double cutWidth = 0.0;
            x1 = xo;
            y1 = yo;
            x2 = x1 + pathLength * Math.Cos(theta);
            y2 = y1 + pathLength * Math.Sin(theta);

            //Build First Line and First Turn and add the the BuildPath Arraylist
            traj1.lineTraj(x1, y1, x2, y2, t0, fullSpeed, TrajType.line);
            traj2.turnTraj(traj1.xn, traj1.yn, traj1.tn, turnSpeed, TrajType.turn, cuttingWidth / 2, traj1.theta,false);
            cutWidth += cuttingWidth;

            path = new BuildPath(traj1);
            path.addNewTraj(traj2);

            double xnTemp = 0;
            double ynTemp = 0;

                    //Iterate on the cutting width forming lines and arcs until the perimeter width is met
            while((cutWidth + cuttingWidth) < pathWidth)
```

```
        {
            if (traj2.turnLeft == false)
            {
                xnTemp = traj1.x0 + cuttingWidth * Math.Cos(tempAng);
                ynTemp = traj1.y0 + cuttingWidth * Math.Sin(tempAng);
                traj1.lineTraj(traj2.xn, traj2.yn, xnTemp, ynTemp, traj2.tn, fullSpeed, TrajType.line);
                traj2.turnTraj(traj1.xn, traj1.yn, traj1.tn, turnSpeed, TrajType.turn, cuttingWidth / 2, traj1.theta, true);
            }
            else
            {
                xnTemp = traj1.x0 + cuttingWidth * Math.Cos(tempAng);
                ynTemp = traj1.y0 + cuttingWidth * Math.Sin(tempAng);
                traj1.lineTraj(traj2.xn, traj2.yn, xnTemp, ynTemp, traj2.tn, fullSpeed, TrajType.line);
                traj2.turnTraj(traj1.xn, traj1.yn, traj1.tn, turnSpeed, TrajType.turn, cuttingWidth / 2, traj1.theta, false);
            }
            path.addNewTraj(traj1);
            path.addNewTraj(traj2);
            cutWidth += cuttingWidth;
        }

                //Finish the last path line
        xnTemp = traj1.x0 + cuttingWidth * Math.Cos(tempAng);
        ynTemp = traj1.y0 + cuttingWidth * Math.Sin(tempAng);
        traj1.lineTraj(traj2.xn, traj2.yn, xnTemp, ynTemp, traj2.tn, fullSpeed, TrajType.line);
        path.addNewTraj(traj1);
        //To reduce inertial uncertainties and to compensate for Slow Heading Updates, Add Extra Paths
        //For slow start up, slow down before a turn and a pause after the turn for the heading to catch up due to a lag
        BuildPath path2 = new BuildPath();
        ArrayList pathTraj = path.getPathHis();
        Trajectory temp = (Trajectory)pathTraj[0];
        Trajectory curTraj = temp;
        curTraj.tn = 0.0;


        while(pathTraj.Count > 0)
        {
            temp = (Trajectory)pathTraj[0];
            //If Traj is turn then just add to Path2 else split up into start middle and finish
            if (temp.type == TrajType.turn)
            {
                Trajectory turn = new Trajectory();
                turn.turnTraj(temp.x0, temp.y0, curTraj.tn, turnSpeed, TrajType.turn, temp.turnRadius,
                              temp.headingAngle, temp.turnLeft);
                path2.addNewTraj(turn);
                curTraj = turn;
                pathTraj.RemoveAt(0);
            }
            else
            {
                double length = Math.Sqrt(Math.Pow((temp.x0 - temp.xn),2)+Math.Pow((temp.y0 - temp.yn),2));
```

```
}
```

```
                    Trajectory stop = new Trajectory();
                    Trajectory startUp = new Trajectory();
                    Trajectory middle = new Trajectory();
                    Trajectory end = new Trajectory();

                    double xn = temp.x0 + 3 * Math.Cos(temp.theta);
                    double yn = temp.y0 + 3 * Math.Sin(temp.theta);
                    stop.lineTraj(temp.x0, temp.y0, xn, yn, curTraj.tn, stopSpeed, TrajType.line);

                    xn = temp.x0 + 44 * Math.Cos(temp.theta);
                    yn = temp.y0 + 44 * Math.Sin(temp.theta);
                    startUp.lineTraj(stop.x0, stop.y0, xn, yn, stop.tn, stopSlow, TrajType.line);

                    xn = temp.x0 + (length-12) * Math.Cos(temp.theta);
                    yn = temp.y0 + (length-12) * Math.Sin(temp.theta);
                    middle.lineTraj(startUp.xn, startUp.yn, xn, yn, startUp.tn, fullSpeed, TrajType.line);

                    xn = temp.x0 + length * Math.Cos(temp.theta);
                    yn = temp.y0 + length * Math.Sin(temp.theta);
                    end.lineTraj(middle.xn, middle.yn, xn, yn, middle.tn, slowSpeed, TrajType.line);

                    pathTraj.RemoveAt(0);
                    path2.addNewTraj(stop);
                    path2.addNewTraj(startUp);
                    path2.addNewTraj(middle);
                    path2.addNewTraj(end);
                    curTraj = end;
                }
            }
            return path2;
        }
    }
}
```

BuildPath.cs

```
namespace MowerControl
{
    public enum TrajType { line, arc, porabola, turn, dontMove }  //the types of paths

    class BuildPath
    {
        private ArrayList path = new ArrayList();
        public static Trajectory curPath;
        private static double endTime = 0.0;

        //constructor which clears old paths and starts new list with first path as input
        public BuildPath(Trajectory firstTraj)
        {
            path.Clear();
```

```
            curPath = firstTraj;
            path.Add(firstTraj);
            endTime = firstTraj.tn;
        }
        //constructor no arguments
        public BuildPath()
        {
            path.Clear();
        }
        //Routine to Get the history of paths
        public ArrayList getPathHis()
        {
            return path;
        }
        //add to the list of path trajectories
        public void addNewTraj(Trajectory newTraj)
        {
            if (path.Count == 0)
                curPath = newTraj;
            path.Add(newTraj);
            endTime = newTraj.tn;
        }
        //get the ending path time
        public static double endTimeIs()
        {
            return endTime;
        }
        //returns feedback control data for the tracking algorithm
        public TrajControlData getControlData(double curTime)
        {
            //if time is past the current path then remove from list and get next path, which
            //should folow in order by time, if there are no paths then set up a generic wait in place
            if (curTime > curPath.tn)
            {
                if (path.Count == 1)
                {
                    Trajectory dontMove = new Trajectory();
                    dontMove.stopTraj(curPath.xn, curPath.yn, curPath.xn, curPath.yn, curPath.tn, TrajType.dontMove);
                    path.Add(dontMove);
                }
                path.RemoveAt(0);
                curPath = (Trajectory)(path[0]);
            }

            if (curPath.type == TrajType.line)
            {
                double deltaD = (curTime - curPath.t0)*curPath.speed;  //note speed should be in in./s
                double xdotgf, ydotgf, xgf,ygf;
                xdotgf = curPath.speed * Math.Cos(curPath.theta);
                ydotgf = curPath.speed * Math.Sin(curPath.theta);
```

```
                xgf = curPath.x0 + deltaD*Math.Cos(curPath.theta);
                ygf = curPath.y0 + deltaD*Math.Sin(curPath.theta);
                return new TrajControlData(xdotgf, ydotgf, xgf, ygf);
            }
            else if (curPath.type == TrajType.turn)
            {
                double deltaD = (curTime - curPath.t0) * curPath.speed;  //note speed should be in in./s
                double xdotgf, ydotgf, xdotgfPrime, ydotgfPrime, xgfPrime, ygfPrime, xgf, ygf, tempAngle;
                tempAngle = deltaD / curPath.turnRadius;                  //this used the relationship theta*r = deltaD

                //in local coordinates
                xdotgfPrime = curPath.speed * Math.Cos(tempAngle);
                ydotgfPrime = curPath.speed * Math.Sin(tempAngle);

                           //if turn right then y is negative
                if (curPath.turnLeft == false)
                    ydotgfPrime = ydotgfPrime * -1.0;

               //now reflect into the global coordinates
                xdotgf = xdotgfPrime * Math.Cos(curPath.headingAngle) - ydotgfPrime * Math.Sin(curPath.headingAngle);
                ydotgf = ydotgfPrime * Math.Cos(curPath.headingAngle) + xdotgfPrime * Math.Sin(curPath.headingAngle);

                //Now get the position in local coordinates
                xgfPrime = curPath.turnRadius * Math.Sin(tempAngle);
                ygfPrime = curPath.turnRadius - curPath.turnRadius * Math.Cos(tempAngle);
                if (curPath.turnLeft == false)
                    ygfPrime = ygfPrime * -1.0;

                //now reflect into global coordinates
                xgf = curPath.x0 + xgfPrime * Math.Cos(curPath.headingAngle) - ygfPrime * Math.Sin(curPath.headingAngle);
                ygf = curPath.y0 + ygfPrime * Math.Cos(curPath.headingAngle) + xgfPrime * Math.Sin(curPath.headingAngle);
                return new TrajControlData(xdotgf, ydotgf, xgf, ygf);
            }
            else if (curPath.type == TrajType.dontMove)
            {
                return new TrajControlData(0, 0, curPath.xn, curPath.yn);
            }
            else
                return new TrajControlData(0, 0, 0, 0);
        }

    }

    //Holds the section of trajectory data, for example the data for a straight-line
    //note the Constructor is heavily loaded with inputs and depends on the type of maneuver
    public struct Trajectory
    {
        public double x0, y0, xn, yn, t0, tn, theta, speed, turnRadius, headingAngle;  //Heading In Radians
        public TrajType type;
        public bool turnLeft;
```

```
        //Routine to Build a Line
        public void lineTraj(double x0, double y0, double xn, double yn, double t0, double speed, TrajType type)
        {
            this.x0 = x0; this.y0 = y0; this.xn = xn; this.yn = yn; this.t0 = t0; this.speed = speed;
            this.type = type;
            //need to handle cases where either numerator or denominator are zero of Atan
            if (xn < x0)
                theta = Math.Atan((yn - y0) / (xn - x0)) + Math.PI;
            else
                theta = Math.Atan((yn - y0) / (xn - x0));
            tn = t0 + (Math.Sqrt((Math.Pow((yn - y0), 2) + Math.Pow((xn - x0), 2))))/speed;
        }
        //Routine to build a circular 180-degree left or right turn
        public void turnTraj(double x0, double y0, double t0, double speed, TrajType type, double turnRadius,
                             double headingAngle, bool turnLeft)
        {
            this.x0 = x0; this.y0 = y0; this.t0 = t0; this.speed = speed; this.turnLeft = turnLeft;
            this.type = type;this.turnRadius = turnRadius;this.headingAngle = headingAngle;  //heading angle is in radians

            double tempAngle;
            if (turnLeft == true)
                tempAngle = headingAngle + Math.PI / 2.0;  //heading angle is last direction facing, in radians
            else
                tempAngle = headingAngle - Math.PI / 2.0;

            xn = x0 + 2 * turnRadius * Math.Cos(tempAngle);
            yn = y0 + 2 * turnRadius * Math.Sin(tempAngle);
            tn = t0 + (Math.PI*turnRadius) / speed;
            theta = 0;  //not important for turning
        }
        //Stay in place Trajectory
        public void stopTraj(double x0, double y0, double xn, double yn, double t0, TrajType type)
        {
            this.x0 = x0; this.y0 = y0; this.xn = xn; this.yn = yn; this.t0 = t0; this.type = type;
            this.type = type;
            tn = t0+10000;  //just a large non ending time value
            theta = 0;
        }
    }

    //holds the feed back data required for the state space path following control
    public struct TrajControlData
    {
        public double xdotgf, ydotgf, xgf, ygf;

        public TrajControlData(double xdotgf, double ydotgf, double xgf, double ygf)
        {
            this.xdotgf = xdotgf;
            this.ydotgf = ydotgf;
```

```
            this.xgf = xgf;
            this.ygf = ygf;
        }
    }
}
```

PathControl.cs

```
namespace MowerControl
{
    class PathControl
    {
        private readonly double d;                      //d-is the distance from baseline to tracking point
        private const double rLeff = 6.58959420;        //Note: effective radius is dependent on terrain
        private const double rReff = 6.58959420;
        private const double bEff = 34.3;               //Effective wheel baseline width
        private const double topRPM = 62.6380997;
        private double kx, ky;

        //Constructor that sets the tracking contrl gains and tracking point offset
        public PathControl(double d, double kx, double ky)
        {
            this.d = d;
            this.kx = kx;
            this.ky = ky;
        }
        //Subroutine to determine the wheel speeds based on the current location and path to follow
        public void getSpeeds(ref sbyte leftS, ref sbyte rightS, Location curLoc, BuildPath path)
        {
            double thetaRad = curLoc.theta * Math.PI / 180.0;
            double projX = curLoc.x + d * Math.Cos(thetaRad);
            double projY = curLoc.y + d * Math.Sin(thetaRad);

            //now calculate the control values
            TrajControlData tCd = path.getControlData(curLoc.time);
            double controlX = tCd.xdotgf - kx * (projX - tCd.xgf);
            double controlY = tCd.ydotgf - ky * (projY - tCd.ygf);

            //now use matrix multiplication to calculate
            //to save processing time equate sin theta and other like terms separate
            double d2brR = bEff / (2 * d * rReff);
            double d2brL = bEff / (2 * d * rLeff);
            double r_1R = 1 / rReff;
            double r_1L = 1 / rLeff;

            double term_1 = ((r_1R * Math.Cos(thetaRad)) - (d2brR * Math.Sin(thetaRad)));
            double term_2 = ((r_1L * Math.Cos(thetaRad)) + (d2brL * Math.Sin(thetaRad)));
            double term_3 = ((r_1R * Math.Sin(thetaRad)) + (d2brR * Math.Cos(thetaRad)));
            double term_4 = ((r_1L * Math.Sin(thetaRad)) - (d2brL * Math.Cos(thetaRad)));
```

```
            //Now Calculate the Wheel Speeds
            double omeghaR = term_1 * controlX + term_3 * controlY;
            double omeghaL = term_2 * controlX + term_4 * controlY;

            //now calculate the desired RPM and equate to speed to set, note 127-set speed is equal to 62.6380997
            int speedR = roundInt(((omeghaR / (2 * Math.PI)) * 60)*127/topRPM);
            int speedL = roundInt(((omeghaL / (2 * Math.PI)) * 60)*127/topRPM);


            //to help the control keep ratio the same if either value is over 127
            if (speedL > 127 || speedR > 127)
            {
                double ratio = speedR / (speedL * 1.0);
                if (speedR > 127)
                {
                    rightS = 127;
                    leftS = (sbyte)roundInt(127 / ratio);
                }
                else if (speedL > 127)
                {
                    leftS = 127;
                    rightS = (sbyte)roundInt(127 * ratio);
                }
            }
            else
            {
                if (speedR < -127)
                    rightS = -127;
                else
                    rightS = (sbyte)speedR;

                if (speedL < -127)
                    leftS = -127;
                else
                    leftS = (sbyte)speedL;
            }
        }

        //rounding routine
        private int roundInt(double value)
        {
            int intVal = (int)value;
            if (value - intVal >= .500)
                intVal++;
            return intVal;
        }

    }
}
```

StateEstimation.cs

```
namespace MowerControl
{
    class StateEstimation  //this is the state estimation for the wheelbase centerline point
    {
        private ArrayList locationHistory = new ArrayList();       //List to hole location History
        private Location lastLocation;                             //Last location
        private const double rLeff = 6.58959420;
        private const double rReff = 6.58959420;
        private const double bEff = 34.3;  //This value along with rLeff and rReff are very difficult to determine
                                           //and the GPS heading will go a long way in compensating for uncertainties in these values

        private Kalman kstateEst;              //Hold instance of the Kalman Class
        private int maxLocations = 1000000; //Max number in location history

        //The three variables below represent the stdev of the drift per unit distance or unit rotation
        private const double kdd = .0416666667;      //variance in distance per distance traveled, (in^2/in)
        private const double kdth = .015;            //variance in heading per distance traveled, (deg^2/in)
        private const double kthth = .136;           //variance in heading per change in heading (deg^2/deg)

        //StateEstimation constructor which generates an instance of the Kalman class
        public StateEstimation(Location firstLocation)
        {
            kstateEst = new Kalman(kdd, kdth, kthth, firstLocation);
            locationHistory.Clear();                 //remove any old history and start a new one
            this.lastLocation = firstLocation;
        }
        //Update the SE, if GPS Data Available between time steps, then fuse it in
        public void updateSE(int encL, int encR, Time timestamp)
        {
            //Loop through all the GPS measurements to find first time that is larger than the last location time
            GPSmeasurement tempGPSmeas = new GPSmeasurement(0, 0, 0, 0.0, 0, 0);
            while(GPSdataProcessing.getMeasCount() > 0)
            {
                tempGPSmeas = GPSdataProcessing.getFirstMeas();
                if(tempGPSmeas.time < lastLocation.time)
                    GPSdataProcessing.removeFirstMeas();
                else
                    break;
            }
            //If between the last encoder measurement and the present one, then incorporate the GPS by using the Kalman Filter
            if ((tempGPSmeas.time > lastLocation.time) && (tempGPSmeas.time < timestamp.timeIs()))
            {
                //Should now have a GPS measurement between the last location update and the new location timestamp
                double stateTimeDiff = timestamp.timeIs() - lastLocation.time;
                double sToKtimeDiff = tempGPSmeas.time - lastLocation.time;
                double ratio = sToKtimeDiff / stateTimeDiff;
                int encDiffL = (int)(ratio * encL);
```

```
                int encDiffR = (int)(ratio * encR);
                //Now update from last state estimate to the GPS measurement
                Location tempLoc = kstateEst.updateKalman(encDiffL, encDiffR, tempGPSmeas.time, lastLocation, tempGPSmeas);
                //Now update from GPS measurement to the present
                          lastLocation = kstateEst.upKalmanNoMeas(encL - encDiffL, encR - encDiffR, timestamp.timeIs(), tempLoc);
                locationHistory.Add(lastLocation);
            }
            else  //If no GPS measurement available, then update with EKF but no measurement correction
            {
                lastLocation = kstateEst.upKalmanNoMeas(encL, encR, timestamp.timeIs(), lastLocation);
                locationHistory.Add(lastLocation);
                limitLocationSize();
            }
        }

        //Limit the Size of Location History
        private void limitLocationSize()
        {
            if (locationHistory.Count > maxLocations)
                locationHistory.RemoveAt(0);
        }
        public ArrayList getHistory()
        {
            return locationHistory;
        }
        public Location getLastLoc()
        {
            return lastLocation;
        }
        public int getCount()
        {
            return locationHistory.Count;
        }
    }

    //Structure to hold pose and other parameter data
    public struct Location
    {
        public double x, y, z, theta;  //theta is the heading in degrees, and x, y are the inch location
        public double time, angularRotationR, angularRotationL;
        public int encoderL, encoderR, encDifference;

        public Location(double x, double y, double z, double theta, double time, double angularRotationR,
                        double angularRotationL, int encoderL, int encoderR, int encDifference)
        {
            this.x = x;
            this.y = y;
            this.z = z;
            this.theta = theta;
            this.time = time;
```

```
            this.angularRotationR = angularRotationR;
            this.angularRotationL = angularRotationL;
            this.encoderL = encoderL;
            this.encoderR = encoderR;
            this.encDifference = encDifference;
        }
    }
}
```

Kalman.cs

```
namespace MowerControl
{
    class Kalman
    {
        private ArrayList kalmanStates = new ArrayList();   //List to store Kalman history
        private KalmanParameters lastState;
        private double Ks, Ksth, Kthth;                     //Parameters from the StateEstimation class
        private double Rx, Ry, Rth;                         //variables to hold the GPS standard deviations
        private const double rLeff = 6.58959420;
        private const double rReff = 6.58959420;
             private const double bEff = 34.3;    //This value along with rLeff and rReff are very difficult to determine and the
                                                  //GPS heading will go a long way in compensating for uncertainties in these values
       //Kalman constructor
        public Kalman(double kdd,double kdth,double kthth, Location firstLocation)
        {
            //Form the Diagonal Process Variance drift Coefficients, be careful to account for degrees to radians conversions
            this.Ks = kdd;                                         //variance inches^2/inch
            this.Ksth = ((kdth*Math.PI*Math.PI)/(180.0*180.0)); / /was in deg^2/inch now in rad^2/inch
            this.Kthth = (kthth*Math.PI)/180.0;                    //was in deg^2/deg, now in rad^2/rad

            //Derive Qo
            MatrixF Q = updateQ(0.0,0.0,firstLocation.theta);      //should be zero because only over distance
                                                                   //does the process error become non-zero
            MatrixF P = new MatrixF(3, 3, 0.0m);   //Use starting uncertainty in Location and heading of about +-3" and +-.1-degrees
            P.array[0, 0] = 9m;
            P.array[1, 1] = 9m;
            P.array[2, 2] = .0000030462m;

             //The Input uncertainty covariance matrix is derived here, Derived assuming 1-encoder tick stdev for each
             //measurement, also uncorrelated so triangular matrix
            double U11 = (1.0 / (720 * 20)) * Math.PI * (rLeff + rReff);  //equals the distance over one tick

             //Encoders are the same and uncorrelated so U is formed as, which is time invariant
            MatrixF U = new MatrixF(2, 2);
            U.array[0, 0] = (decimal)(U11*U11);
            U.array[1, 1] = (decimal)(U11*U11);

                    MatrixF A = new MatrixF(3, 3);
            MatrixF B = new MatrixF(3, 3);
```

```
        MatrixF H = new MatrixF(3, 3);

         //assume that measurements for x, y, and theta are uncorrelated, The uncertainties are set here
        GPSmeasurement temp = new GPSmeasurement(0, 0, 0, 0, 0, 0);
        MatrixF R = updateR(temp);

        MatrixF K = new MatrixF(3, 3);
        decimal[] startState = {(decimal)firstLocation.x, (decimal)firstLocation.y, (decimal)firstLocation.theta };
        MatrixF X = new MatrixF(startState,1);
        //Initial starting KalmanParameters
        lastState = new KalmanParameters(Q, K, R, P, U, A, B, H, X, firstLocation.time);

    }

    //subroutine to update the state estimate using an Extended Kalman filter approach
    public Location updateKalman(int encL, int encR, double timestamp, Location lastLocation, GPSmeasurement gpsMeas)
    {
        //First Update the non-linear Discretized equations of motion
        double deltaDisR = (encR * 2 * Math.PI * rReff) / (14400);   //Note: 14400 is #ticks per revolution
        double deltaDisL = (encL * 2 * Math.PI * rLeff) / (14400);
        double deltaD = (deltaDisL + deltaDisR) / 2.0;
        double deltaTheta = ((deltaDisR - deltaDisL) / bEff);        //angle in radians
        double deltaTdegrees = deltaTheta * 180.0 / Math.PI;         //angle in degrees
        double newTheta = lastLocation.theta + deltaTdegrees;
        double avgAngle = ((lastLocation.theta + newTheta) / 2.0) * Math.PI / 180.0;  //in radians
        double newX = lastLocation.x + deltaD * Math.Cos(avgAngle);
        double newY = lastLocation.y + deltaD * Math.Sin(avgAngle);

         //To be consistent verify that up to 180+ and up to -180 Neg
        if (newTheta > 180.0)
            newTheta = newTheta - 360;
        else if (newTheta < -180.0)
            newTheta = newTheta + 360;

        //Now the first estimate of the state can be made
        decimal [] temp = {(decimal)newX,(decimal)newY,(decimal)(newTheta*Math.PI/180.0)};  //note important to convert the theta
                                                                                            //to radians
        MatrixF Xk1_k0 = new MatrixF(temp,3);

                //Now update Q, B, U, & A
        MatrixF Qk1 = updateQ(deltaD, deltaTheta, avgAngle);
        MatrixF Ak1 = updateA(deltaD, avgAngle);
        MatrixF Bk1 = updateB(deltaD, avgAngle);

        //Now calculate the updated covariance Matrix, Note U is time invariant, while Q is not
        MatrixF Pk1_k0 = Ak1 * lastState.P * MatrixF.Transpose(Ak1) + Bk1 * lastState.U * MatrixF.Transpose(Bk1) + Qk1;

        //Note H is a time invariant liner function, it turns out to be the Identity matrix for the GPS measurements,
        MatrixF Hk1 = new MatrixF(3,3,1);
```

```
//Form the Measurment Vector based on the input GPSmeasurement
MatrixF Zk1 = new MatrixF(3, 1);
Zk1.array[0, 0] = (decimal)gpsMeas.x;
Zk1.array[1, 0] = (decimal)gpsMeas.y;
Zk1.array[2, 0] = (decimal)gpsMeas.headRad;

//Because of the bound between -180 and +180-degrees include if statement below
if (((double)Math.Abs(Zk1.array[2, 0]) > Math.PI / 2.0) && ((double)Math.Abs(Xk1_k0.array[2, 0]) > Math.PI / 2.0)
      && (Math.Sign(Zk1.array[2, 0]) != Math.Sign(Xk1_k0.array[2, 0])))
{
    if ((double)Zk1.array[2, 0] < 0.0)
        Zk1.array[2, 0] = 2.0m * (decimal)Math.PI + Zk1.array[2, 0];
    else
        Xk1_k0.array[2,0] = 2.0m * (decimal)Math.PI + Xk1_k0.array[2,0];
}

//Get the new R-matrix, depends on what the current path is and the GPSmeasurement Stats
MatrixF Rk1 = updateR(gpsMeas);

        //Calculate the normalized innovation squared
MatrixF nInovSq = MatrixF.Transpose(Zk1 - Xk1_k0) * MatrixF.inv(Pk1_k0 + Rk1) * (Zk1 - Xk1_k0);

        //If nInovSq is out of 95% bounds then disregard measurement, maybe try even larger upper threshold
if (nInovSq.array[0, 0] < .0001m || nInovSq.array[0, 0] > 9.348m)
{
    Rk1.array[0, 0] = 100000m;  //very large
    Rk1.array[1, 1] = 100000m;
    Rk1.array[2, 2] = 40m;      //~360 stdev
}

        MatrixF Xk1_k1;

//Now calculate the Gain matrix, this equation is simplified because Hk is the identity matrix,
MatrixF Kk1 = Pk1_k0 * MatrixF.inv(Pk1_k0 + Rk1);

//Now update the state estimate and the covariance matrix, this matrix is also simplified because
//H is a linear identity matrix
Xk1_k1 = Xk1_k0 + Kk1 * (Zk1 - Xk1_k0);
MatrixF identity = new MatrixF(3, 3, 1m);
MatrixF Pk1_k1 = (identity - Kk1) * Pk1_k0;

//Also update the angular RPM of the right and left wheel
double timeDiff = timestamp - lastLocation.time;
double angularRateR = ((encR * 60.0) / (20.0 * 720)) / timeDiff;  //the wheel RPM
double angularRateL = ((encL * 60.0) / (20.0 * 720)) / timeDiff;

//Note U is time invariant uncertainties and is pulled from the last Kalman state
KalmanParameters newEstState = new KalmanParameters(Qk1, Kk1, Rk1, Pk1_k1, lastState.U.Clone(),
                                                    Ak1, Bk1, Hk1, Xk1_k1, timestamp);
```

```
                //Note: need to convert the angle back to degrees
            Location newLoc = new Location((double)Xk1_k1.array[0, 0], (double)Xk1_k1.array[1, 0], 0.0, ((double)Xk1_k1.array[2, 0])
                                    * 180.0 / Math.PI, timestamp, angularRateR, angularRateL, encL, encR, (encR - encL));

            lastState = newEstState;
            kalmanStates.Add(newEstState);
            return newLoc;
        }

        //subroutine to update the state estimate using a kalman filter approach, without a measurement update
        public Location upKalmanNoMeas(int encL, int encR, double timestamp, Location lastLocation)
        {
            //First Update the non-linear Discretized equations of motion
            double deltaDisR = (encR * 2 * Math.PI * rReff) / (14400);    //14400 is #ticks/revs
            double deltaDisL = (encL * 2 * Math.PI * rLeff) / (14400);
            double deltaD = (deltaDisL + deltaDisR) / 2.0;
            double deltaTheta = ((deltaDisR - deltaDisL) / bEff);         //angle in radians
            double deltaTdegrees = deltaTheta * 180.0 / Math.PI;          //angle in degrees
            double newTheta = lastLocation.theta + deltaTdegrees;
            double avgAngle = ((lastLocation.theta + newTheta) / 2.0) * Math.PI / 180.0;  //in radians
            double newX = lastLocation.x + deltaD * Math.Cos(avgAngle);
            double newY = lastLocation.y + deltaD * Math.Sin(avgAngle);

            //To be consistent verify that up to 180+ and up to -180 Neg
            if (newTheta > 180.0)
                newTheta = newTheta - 360;
            else if (newTheta < -180.0)
                newTheta = newTheta + 360;

            //Now the first estimate of the state can be made
            decimal[] temp = { (decimal)newX, (decimal)newY, (decimal)(newTheta * Math.PI / 180.0) };  //note important to
                                                                                                     //convert the theta to radians
            MatrixF Xk1_k0 = new MatrixF(temp, 3);

                    //Now update Q, B, & A
            MatrixF Qk1 = updateQ(deltaD, deltaTheta, avgAngle);
            MatrixF Ak1 = updateA(deltaD, avgAngle);
            MatrixF Bk1 = updateB(deltaD, avgAngle);

            //Now calculate the updated covariance Matrix, Note U is time invariant, while Q is not
            MatrixF Pk1_k0 = Ak1 * lastState.P * MatrixF.Transpose(Ak1) + Bk1 * lastState.U * MatrixF.Transpose(Bk1) + Qk1;

            //Note H is a time invariant linar function, it turns out to be the Identity matrix for the GPS measurements,
            MatrixF Hk1 = new MatrixF(3, 3, 1);

            //Also update the angular RPM of the right and left wheel
            double timeDiff = timestamp - lastLocation.time;
            double angularRateR = ((encR * 60.0) / (20.0 * 720)) / timeDiff;  //the wheel RPM
            double angularRateL = ((encL * 60.0) / (20.0 * 720)) / timeDiff;
```

```
            //Update the Kalman State and add to history but don't need to return any location only when measurement alters the state
            KalmanParameters newEstState = new KalmanParameters(Qk1, null, lastState.R.Clone(), Pk1_k0, lastState.U.Clone(), Ak1,
                                                                Bk1, Hk1, Xk1_k0, timestamp);
            Location newLoc = new Location((double)Xk1_k0.array[0, 0], (double)Xk1_k0.array[1, 0], 0.0, ((double)Xk1_k0.array[2, 0])
                                            * 180.0 / Math.PI, timestamp, angularRateR, angularRateL, encL, encR, (encR - encL));
            lastState = newEstState;
            kalmanStates.Add(newEstState);
            return newLoc;
        }

        //Routine To Update R,
        private MatrixF updateR(GPSmeasurement gpsMeas)
        {
            MatrixF R = new MatrixF(3, 3);

            //Because of a big (~1-2 sec) heading lag during turning, phase out this meassurement till the heading stabalizes
            if (BuildPath.curPath.type == TrajType.turn || BuildPath.curPath.speed == 1.0)
                Rth = 25;   //Since Heading is slow to update don't trust during turnign manuvers, will have
                            //stop to catch up after the turn
            else
                Rth = .75; //not .25 because of lag is to sensitive, and system can become unstable

            //Based on the error statistics from the GPS board, build the uncertainty, because turning left has a tendency
            //to diverge higher uncertainty
            if (BuildPath.curPath.turnLeft == false)
            {
                Rx = 1.5 * gpsMeas.longError * 39.37;
                Ry = 1.5 * gpsMeas.latError * 39.37;
            }
            else
            {
                Rx = gpsMeas.longError * 39.37;
                Ry = gpsMeas.latError * 39.37;
            }

            R.array[0, 0] = (decimal)(Rx * Rx);  //this is variance in X-measurment
            R.array[1, 1] = (decimal)(Ry * Ry);  //this is variance in Y-measurement
                    R.array[2, 2] = (decimal)Math.Pow(((Rth * Math.PI) / 180.0), 2);
            return R;
        }

        //Routine To Update Process Error which is not time invariant
        private MatrixF updateQ(double deltaD, double deltaTheta, double avgAngle)
        {
            //Note the model taken here is the one proposed by Crowley
            decimal Q11 = (decimal)(Ks * Math.Abs(deltaD * Math.Cos(avgAngle)));
            decimal Q22 = (decimal)(Ks * Math.Abs(deltaD * Math.Sin(avgAngle)));
            decimal Q33 = (decimal)(Ksth * Math.Abs(deltaD) + Kthth * Math.Abs(deltaTheta));
            MatrixF X = new MatrixF(3, 3);
            X.array[0, 0] = Q11;
```

```
            X.array[1, 1] = Q22;
            X.array[2, 2] = Q33;
            return X;
        }

        //Routine To Update Jacobian Of system with respect to state variables
        private MatrixF updateA(double deltaD, double avgAngle)
        {
            //Note the model taken here is the one proposed by Crowley
            decimal A13 = (decimal)(-1.0*deltaD * Math.Sin(avgAngle));
            decimal A23 = (decimal)((deltaD * Math.Cos(avgAngle)));
            MatrixF X = new MatrixF(3, 3, (decimal)1.0);
            X.array[0, 2] = A13;
            X.array[1, 2] = A23;
            return X;
        }

        //Routine To Update Jacobian of System with respect to inputs
        private MatrixF updateB(double deltaD, double avgAngle)
        {
            //Note the model taken here is the one proposed by Crowley
            //the covariance will represent degrees in radians
            MatrixF X = new MatrixF(3, 2);
            double d2B = deltaD/(2*bEff);
            X.array[0, 0] = (decimal)(.50 * Math.Cos(avgAngle) + d2B * Math.Sin(avgAngle));
            X.array[0, 1] = (decimal)(.50 * Math.Cos(avgAngle) - d2B * Math.Sin(avgAngle));
            X.array[1, 0] = (decimal)(.50 * Math.Sin(avgAngle) - d2B * Math.Cos(avgAngle));
            X.array[1, 1] = (decimal)(.50 * Math.Sin(avgAngle) - d2B * Math.Cos(avgAngle));
            X.array[2, 0] = (decimal)(-1.00 / bEff);
            X.array[2, 1] = (decimal)(1.00 / bEff);
            return X;
        }
    }

    //Just a structure to hold all of the Kalman Parameter matrices
    public struct KalmanParameters
    {
        public MatrixF Q, K, R, P, U, A, B, H, X;     //Q is system covariance, K is optimal kalman gain for time step,
                                                      //R is measurement noise covariance, P is the State Error covariance matrix,
                                                      // U is the input noise covariance matrix
        public double timeStamp;

        public KalmanParameters(MatrixF Q, MatrixF K, MatrixF R, MatrixF P, MatrixF U, MatrixF A, MatrixF B, MatrixF H,
                                MatrixF X, double time)
        {
            this.Q = Q;
            this.K = K;
            this.R = R;
            this.P = P;
            this.U = U;
```

```
            this.A = A;
            this.B = B;
            this.H = H;
            this.X = X;
            timeStamp = time;
        }
    }
}
```

PART 2 Appendix-C

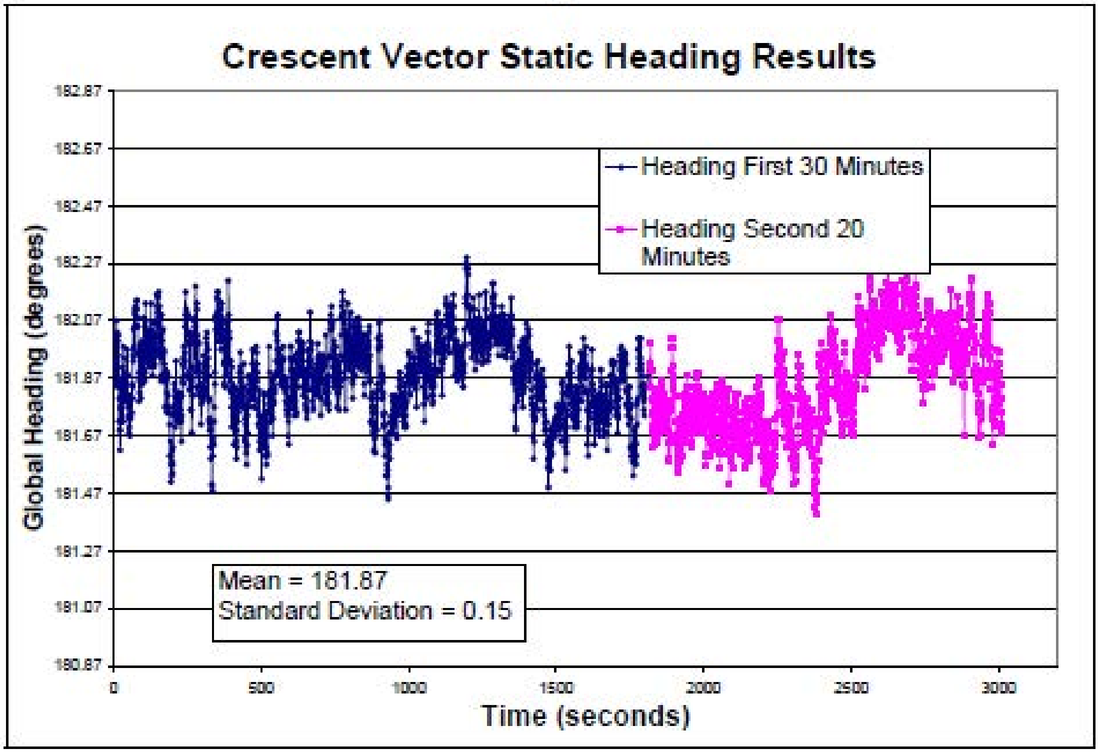


Figure C.1 Crescent Vector GPS static heading test results

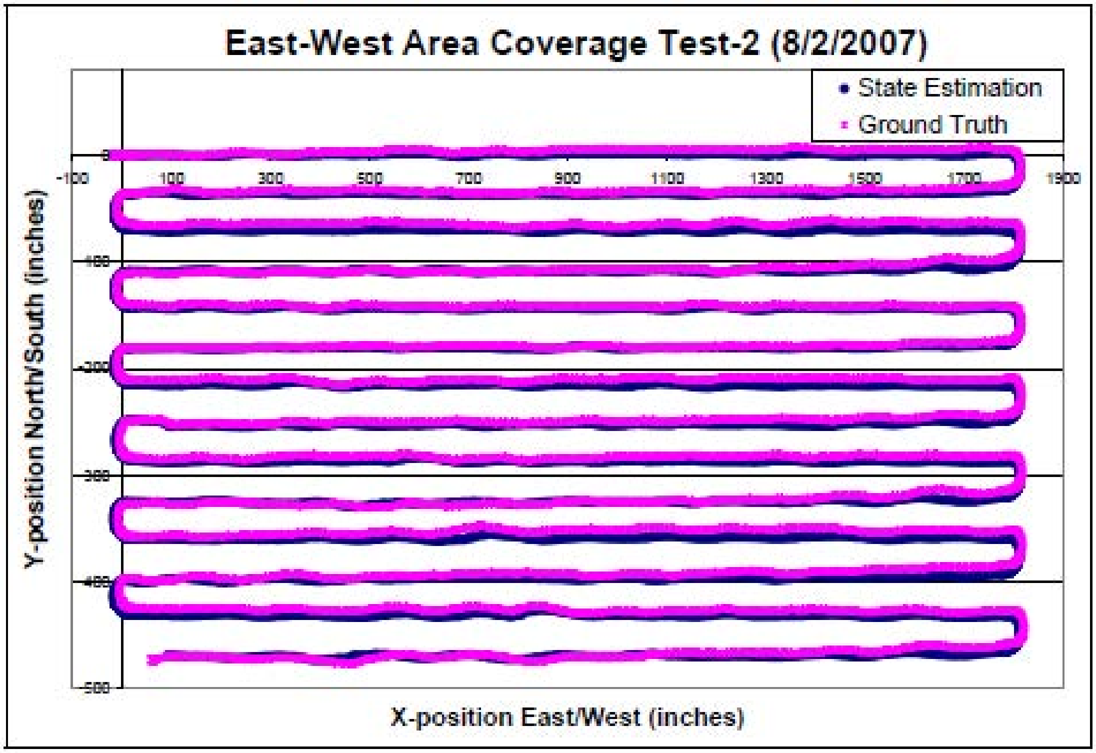


Figure C.2 Path tracking results from Test-2

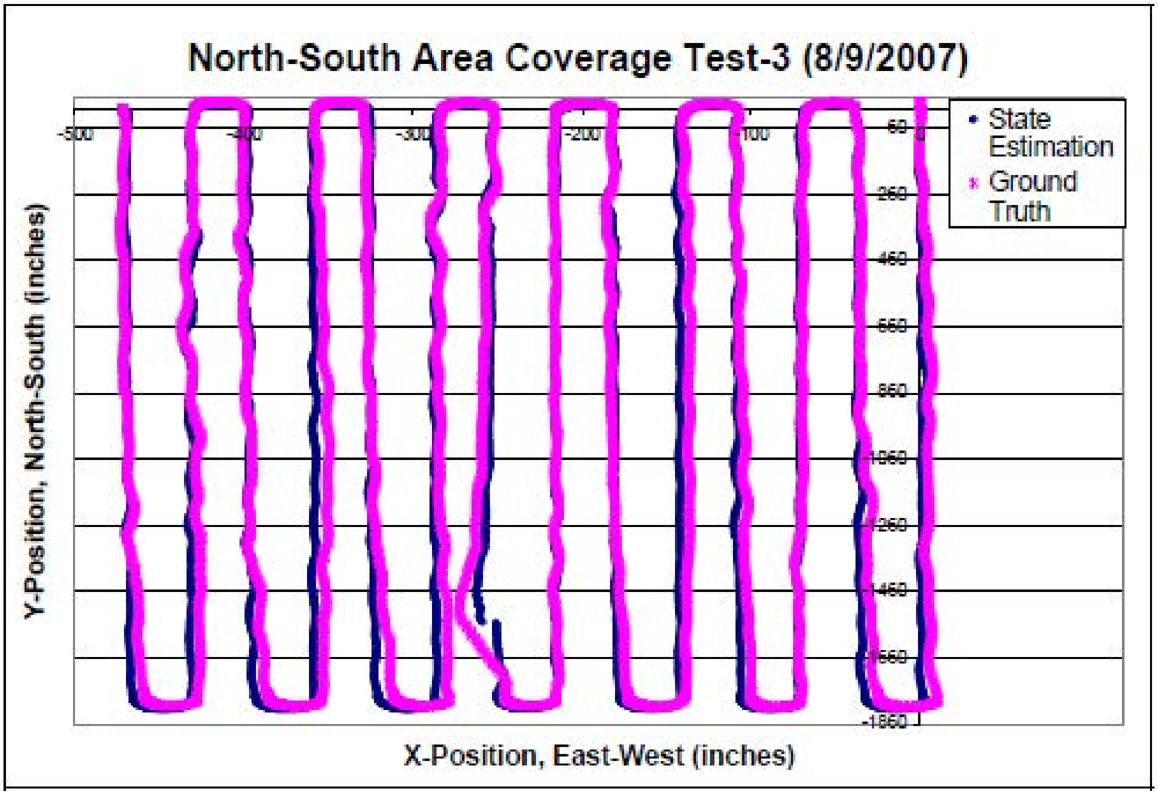


Figure C.3 Path tracking results from Test-3

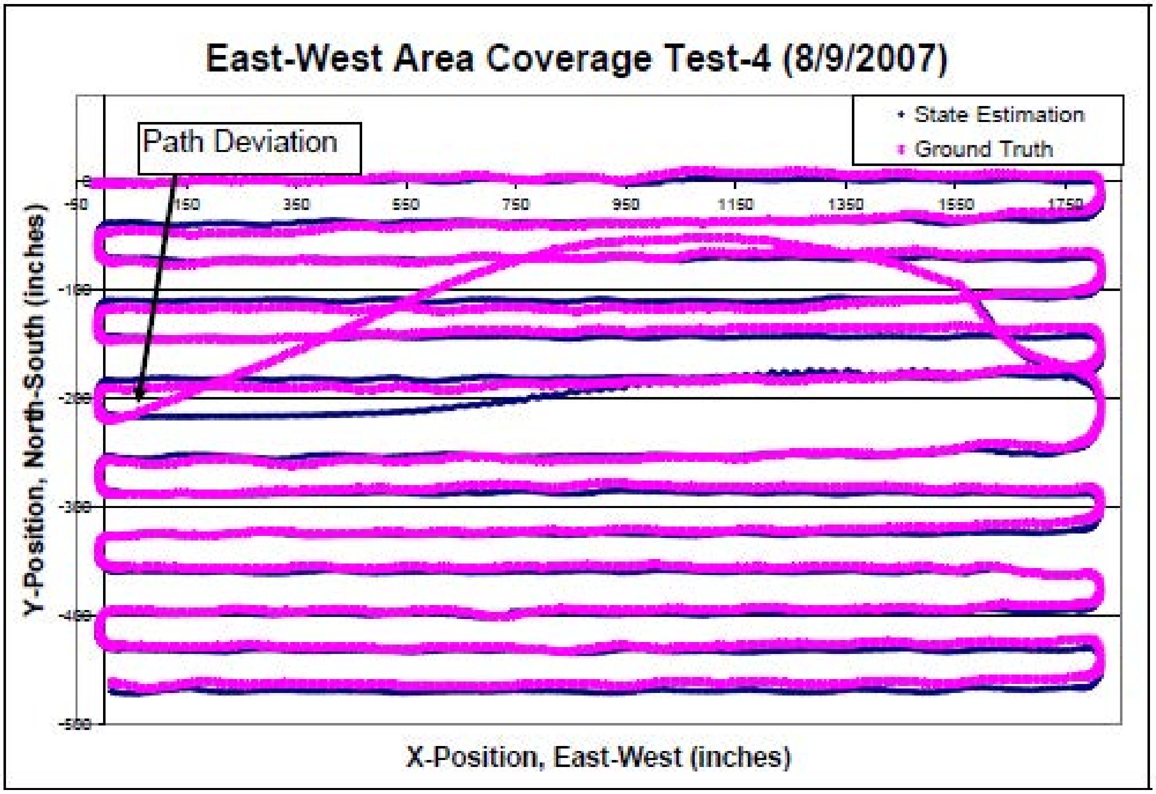


Figure C.4 Path tracking results from Test-4

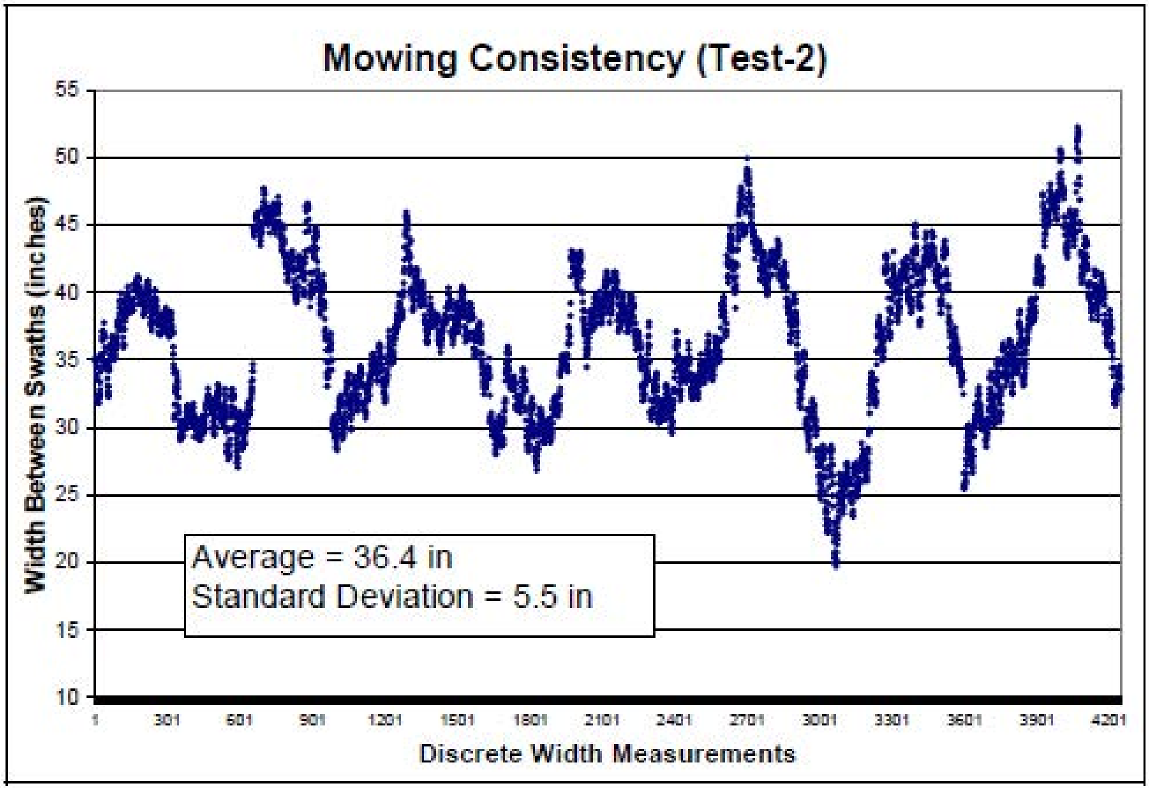


Figure C.5 Plot of the width between consecutive mowing swaths for Test-2

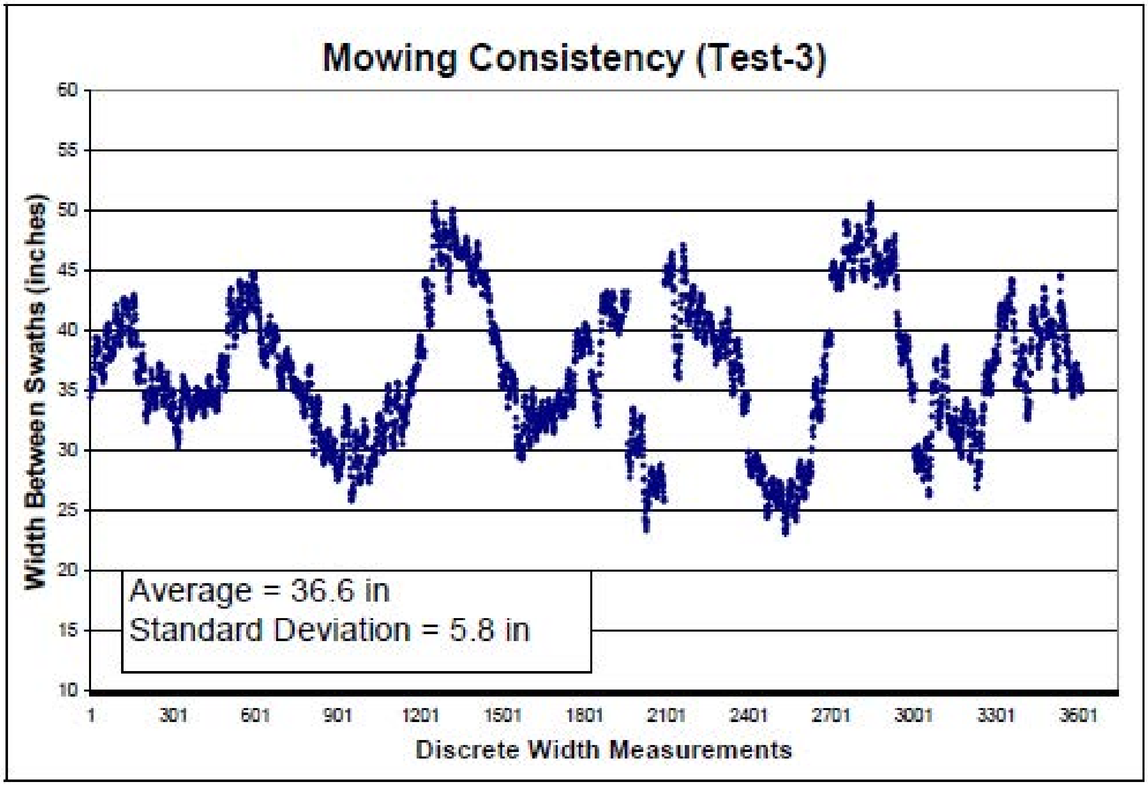


Figure C.6 Plot of the width between consecutive mowing swaths for Test-3

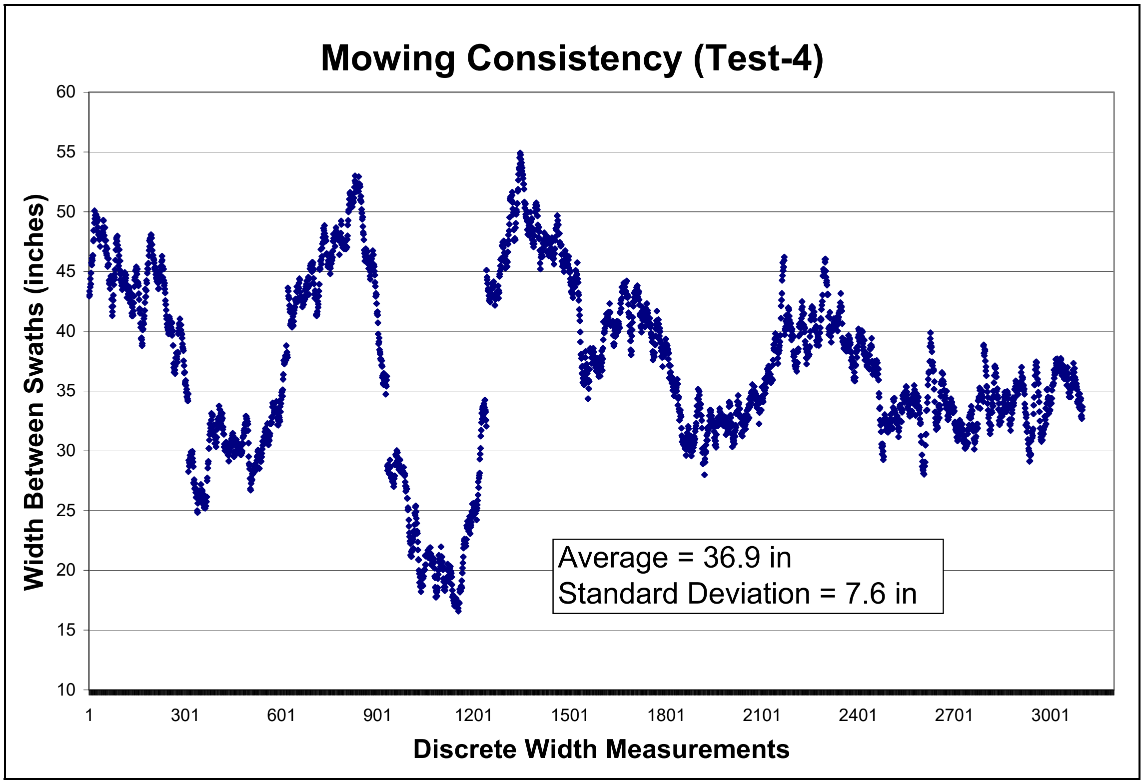


Figure C.7 Plot of the width between consecutive mowing swaths for Test-4

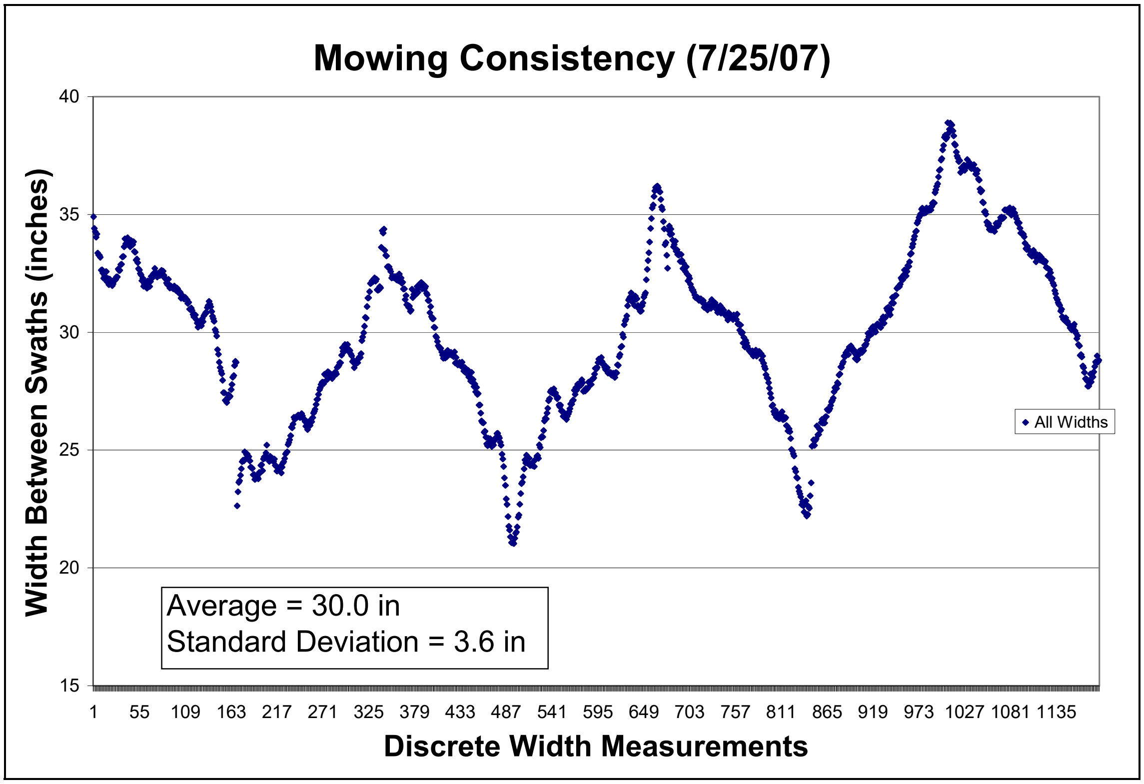


Figure C.8 Plot of the width between consecutive mowing swaths for pavement test

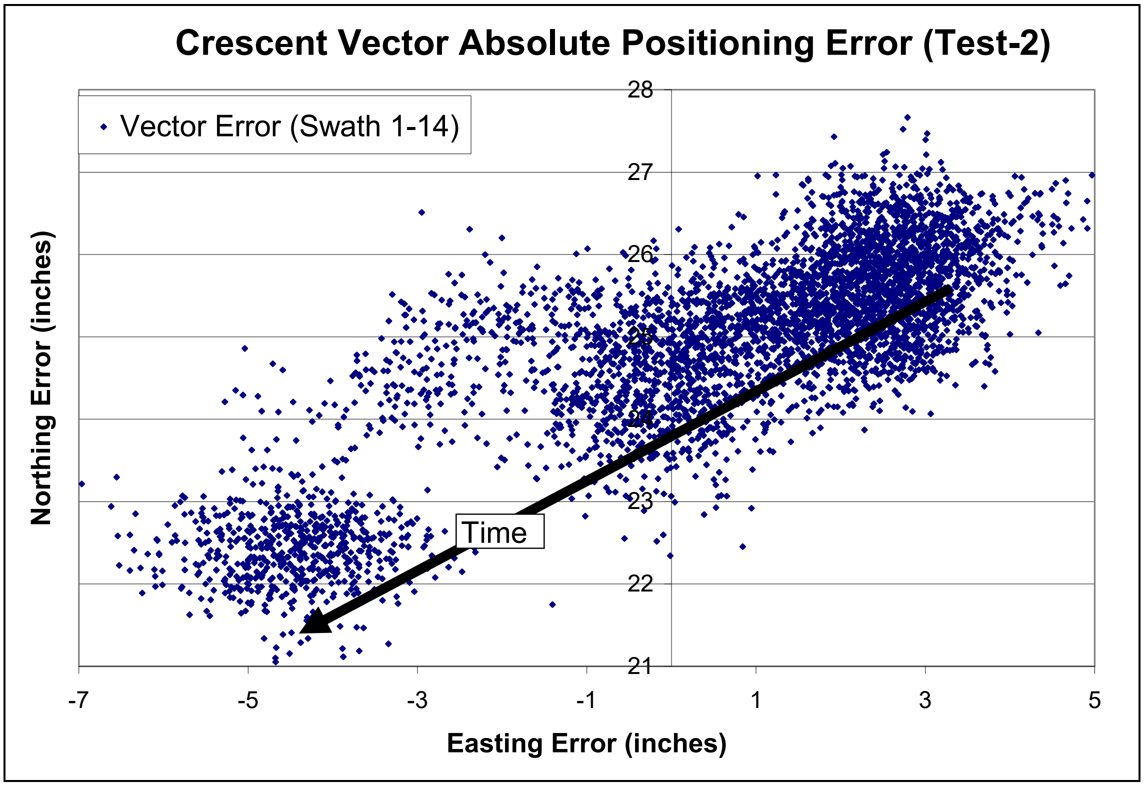


Figure C.9 Crescent Vector absolute dynamic positioning error for Test-2