TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT OF THE DISSERTATION

Robot Traffic School

by

Thomas Denewiler

Master's of Science in Mechanical and Aerospace Engineering

University of California San Diego, 2010

Professor Thomas Bewley, Chair

This dissertation will be abstract.

Chapter 1

Introduction

In this chapter we give a brief overview of autonomous navigation and the small unmanned ground vehicles (UGVs) that will be considered in this thesis. We also describe some of the issues in state estimation and controls that were considered during research in improving the autonomous driving behaviors of the UGVs.

1.1 The Need for Autonomy

Robots have been developed to assist humans in tasks that are generally considered dirty, dangerous or boring. Recently robots have found a useful niche as a tool to help Expolosives Ordinance Disposal (EOD) teams to assess and eliminate threats due to improvised explosive devices (IEDs), commonly referred to as roadside bombs, by allowing humans to maintain a safe stand-off distance while investigating a scene. Clearly this falls under the dangerous category. The current method that EOD teams use involves teleoperation of the robot to get from the base to the object of interest. In the process of teleoperating the robot the operator is exposed and vulnerable to other external threats in the area. As technologies mature to provide humans with better tools shortcomings are discovered, such as the vulnerability due to teleoperation, and that opens up avenues for improvement in the development of these tools. For robots one approach to reducing the amount of work humans are required to perform is to give the robots more intelligence via autonomous behaviors using additional sensors, more specialized actuators and software to automate the largest amount of routine tasks as possible.

When adding autonomy to robots nearly all of the tasks can be summarized by the following questions:

- Where am I?
- What's around me?
- Where do I want to go?
- How do I get there?

Other tasks for small UGVs include sending them into buildings that are dangerous due to structural damage or unknown, possibly hostile elements inside so that the robots can map the interior and additionally to provide human operators with images to assess the danger prior to any humans entering the buildings [Con 06]. After the attacks on the World Trade Center on September 11, 2001 several small UGV systems were used to look for survivors in the rubble and to help assess structural damage to nearby buildings [Everett 02].

The initial attempt at adding autonomy to the EOD robots resulted in somewhat erratic driving behavior, especially near obstacles, as the robot trajectory would not be smooth as it changed speed and attempted to make small corrections to its original path to move around the obstacle. In this thesis we will look at smoothing out the trajectories taken by the robot by looking for improvements in the state estimation (Where am I?) and controls (How do I get there?) algorithms. This work ignores actual obstacle detection (What's around me?) and will be using a simple planning algorithm (Where do I want to go?) to simulate obstacles in the robot path which will force the robot to change direction and speed multiple times.

*** Talk about how speed and efficiency are important characteristics of robots in their typical operating environments and that improved navigation would improve the speed and efficiency. An example is less time waiting to get robot close to IED or to clear a building means less time for humans in hostile environment. In search and rescue situations this would lead to less time for searching and more time for rescuing. ***

1.2 Thesis Outline & Contributions

The problem of autonomous navigation is not isolated to any one technical area but is instead a combination of sensor integration to allow the robot position and obstacles to be observable, noise filtering, state estimation and control algorithms. As much as possible this research has attempted to isolate the effects of each of those areas to enable a quantitative analysis to determine which parts of the system contribute the largest effect on overall robot behavior.

Chapter 2 gives some background on the types of robots used, the sensors on the robots and the operator control unit (OCU). Chapter 3 presents the state estimation algorithm that is used on the robots. Chapter 4 discusses the original control algorithm and a new control algorithm applied as part of this research. Chapter 5 gives the results of experiments run with the robots and looks at the main contributing factors for smooth autonomous navigation. Chapter 6 suggests future avenues of research to pursue. The conclusion is found in Chapter 7.

The main contribution of this research is that the small UGVs investigated here have greater autonomy which will allow for less human oversight of basic functionality. This is especially important for the dirty, dangerous and boring tasks that robots where robots are most useful because it allows the humans to focus their concentration, time and effort on clean, safe and efficient.

Chapter 2

Background

*** Put in a description of the PackBot, Talon and Urbot along with a description of the algorithms originally used and the results obtained using those algorithms. Talk about JAUS and MOCU a little bit. Discuss more about how the robots are used by EOD. Include that for robots to drive around on their own all they require is an estimate of where they are, a path to follow, a controller to determine the actuator outputs and motor controllers to perform the controller outputs. ***

2.1 PackBot

2.2 Talon

2.3 Urbot

2.4 MOCU & JAUS

2.5 The Duals: Estimation & Controls

It is very difficult to simply work on either state estimation or controls individually as there is a large amount of coupling between the two areas. Although the main goal is to make the robots drive more smoothly and that the actuator and motor outputs are ultimately generated by the control system it is still the case that the role of state estimation is equally important. If there exists large meaurement errors, drift or bias in the sensor readings then the robot will not have a very good idea of where it is locted and there will not be a controller that can stabilize the system. *** Talk about observability and controllability. Mention theory that shows link between estimation and control. ***

An example would be when the only sensor available for measurements is an IMU which suffers from drift and bias, where both effects are exagerrated by temperature. There have been situations in which an IMU was in a robot with the motors turned off so that the robot is not moving. However, due to excessive heat in the electronics bos the IMU measurements report that the heading of the robot keeps moving in circles at a rate of $\frac{\pi}{5} rad/s$. With a controller that was known to keep the robot stable when the IMU was working properly started forcing the robot to turn in circles when the motors were turned on even thought the command was to stay in one place. This shows the importance of state estimation on overall robot performance – it is not enough to only have a good controller.

Chapter 3

State Estimation

*** Talk about quantifying the performance of the ACS Kalman filter [Sights 06]. Discuss training of the covariance matrices. Show the position estimation using the original covariance matrices and the ones found from training. If I get to identifying bias and/or drift in the IMU put that here as well. ***

The Space and Naval Warfare Systems Center, San Diego (SSC-SD) robotics group has developed the Autonomous Capabilities Suite (ACS) which incorporates many different technologies into a single software package that can be run on a wide variety of different robots and is able to easily accomodate different payload and sensor suites [Sights 06]. One of the ACS libraries is the adaptive extended Kalman filter which is used on the EOD robots for state estimation and is the main method used for answering the question "Where am I?". The idea behind the Kalman filter is relatively straightforward. The robot has some basic idea of where it is in the world but there is some uncertainty involved in that estimate due to different measurement accuracies from multiple sensors that measure the same state, noise in the individual sensor measurements and an imperfect model of how the robot moves through the world. Some of the uncertainty of the model can be explained by the fact that not all of the necessary measurements are being carried out and the states can be unobservable. *** Say more here about noise/uncertainty. ***

An example is a robot driving in a straight line where the left track may be moving on a flat surface while the right track is moving on an uneven surface as in Figure 3.1. The wheel encoders that measure how far each track is moving will report that the right track is traveling a greater distance than the left track which could mean that the robot is turning counter-clockwise or that the robot tracks are moving over different surface types. At the same time the robot will be getting measurements about its heading from both the IMU and GPS sensors that will have some noise as well. In this example both the IMU and GPS sensors would likely say that the robot is traveling in a straight line on average (as long as the controller is performing adequately). The job of the Kalman filter is to determine how much each sensor should be trusted when trying to determine where the robot really is in the world and how fast it is moving. This is accomplished by looking at each of the noise parameters for both the system model and the measurements as being zero mean, white noise, uncorrelated, Gaussian variables ... *** Clean up this language. Consider putting it in a different section. ***

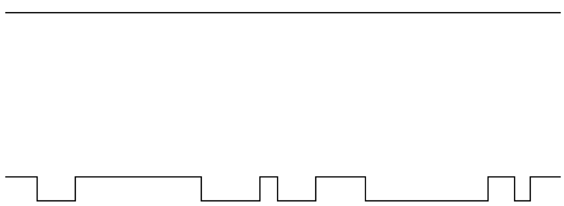

Figure 3.1: Different topographies for the left track and the right track when the ground is smooth on the left side and bumpy on the right side. The top line is for the left track and the bottom line is for the right track.

3.1 State Space Models

Kalman filters and control systems (see Chapter 4) use the idea of a multi-dimensional state space to encapsulate all of the relevant information that is known about a system. In the case of robots the dynamics are typically captured by position, orientation, linear and angular velocities, acceleration and sometimes jerk. The standard equations to describe the state space of a system are

$$\begin{aligned} \dot{x} &= f(x,u,t) \\ \dot{y} &= h(x,t) \end{aligned} \tag{3.1}$$

The state variables are given in vector form by x and the sensor measurements are contained in the vector y. The state space equations are a means of representing how the state and measurements of a system change through time based on the initial state of the system and the inputs, u, to the system which allows the trajectory (or motion through time) and the effect of the trajectory on the measurements to be calculated using compact notation. The inputs are assumed to include any external forces applied to the system as well as actuation provided by the system itself.

3.2 The Kalman Filter

The ACS Kalman filter is typical of all Kalman filters in that it consists of a prediction update step and a measurement update step where the prediction update is run as fast as possible and the measurement update is run whenever new sensor data becomes available as in Figure 3.2. The prediction update step uses the model of the dynamics of the system and a measurement of elapsed time to determine where the system is in the world. The measurement update step is basically a feedback step to help correct for errors in the system model [Kelly 94].

From [Kelly 94], [Simon 06] the prediction update step marches the system dynamics forward in time using the equations

$$\begin{aligned} \hat{x}_{k+1}^{-} &= \Phi_k \hat{x}_k \\ P_{k+1}^{-} &= \Phi_k P_k \Phi_k^T + \Gamma_k Q_k \Gamma_k^T \end{aligned} \tag{3.2}$$

and the measurement update step provides feedback from sensor data using the equations

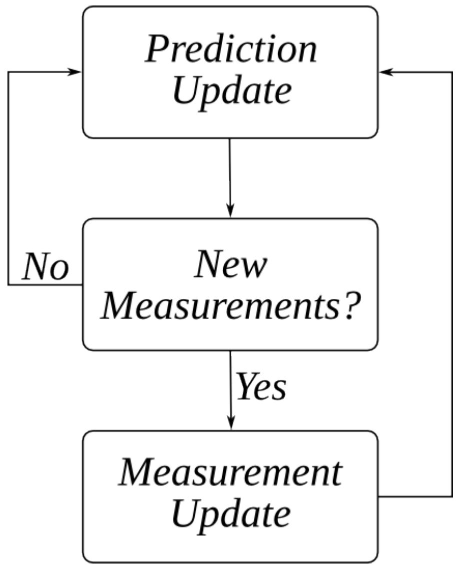

Figure 3.2: The Kalman filter algorithm.

$$
\begin{aligned}
K_k &= P_k^- H_k^T \left[H_k P_k^- H_k^T + R_k \right]^{-1} \\
\hat{x}_k &= \hat{x}_k^- + K_k \left[z_k - H_k \hat{x}_k^- \right] \\
P_k &= \left[I - K_k H_k \right] P_k^-
\end{aligned}
\tag{3.3}
$$

The state space equations for the robot are

$$
\begin{aligned}
x_{k+1} &= \Phi_k x_k + \Gamma_k w_k \\
z_k &= H_k x_k + v_k
\end{aligned}
\tag{3.4}
$$

3.3 Adaptive Extended Kalman Filter

*** I really need to clean up the language here. [Busse 03] looks like a great source. Even the notation as far as *a priori* and *a posteriori* needs fixing. *** Attempting to determine the proper values for the covariance matrices Q in (3.2) and R in (3.3) can be a laborious process and is often considered more of an art than a science with engineer experience being a critical factor. *** Discuss why Q and R are important and what function they serve in the Kalman filter.

*** The ACS Kalman filter has been implemented with an adaptive scheme to update the covariance matrices in real time as the robot moves around and sensor measurements are taken into account [Sights 06], [Mehra 72], [Busse 03]. Q and R are updated at alternating time steps in the EKF.

The first step is to calculate Q^* using

$$Q^* = \left(x - x_{k+1}^-\right)\left(x - x_{k+1}^-\right)^T + P_{k+1}^- - P - Q \tag{3.5}$$

Then Q can be updated using

$$Q = Q + \frac{1}{L_Q}\left(Q^* - Q\right) \tag{3.6}$$

Next R^* is calculated using

$$R^* = (y - Hx)(y - Hx)^T - HP_{k+1}^- H^T \tag{3.7}$$

and R can be updated using

$$R = R + \frac{1}{L_R}\left(R^* - R\right) \tag{3.8}$$

*** Discuss the implications of the adaptive EKF. ***

3.4 Discriminative Training of Kalman Filter Parameters

*** Investigate the difference between adaptive filtering and training. It seems like they accomplish the same thing, namely, convergence to some values for the covariance matrices. Do they use the same metrics? Do they converge to the same covariance matrices? Is it just online vs. offline training? *** [Abbeel 05] describes a method to automatically learn what the covariance matrices should be. When used in conjunction with the adaptive EKF scheme this could allow for faster convergence times when the robots are started and for smaller ranges for the adaptation coefficients L_Q and L_R in (3.6) and (3.8).

3.5 Identifty IMU Parameters

Chapter 4

Controls

*** Talk about Lyapunov and PID controllers. ***

4.1 PID

*** Talk about how PID controllers work. Discuss the difficlties of tuning the PID controllers. ***

4.2 Lyapunov

*** Talk about how Lyapunov controllers work [Khalil 02]. Show which control Lyapunov function I chose [Rusu 05]. For a given path show the linear and angular velocities that are output by each controller. ***

Chapter 5

Results

*** I want to show plots with the position estimate using GPS only, KF with learned Q/R but no adapting, KF with no adapting or training, KF with adapting, KF with learned and adaptive, KF with different encoder equations. Would be cool to plot these on an overhead image of the test area. ***

*** I want to show plots of the variance of the KF position estimate and the derivative of the control outputs of linear and angular velocity. The real goal is to have smooth velocities which will show up as constant accelerations and I want to see if there is any correlation between the variance of the position estimate and the accelerations, especially when the variance of the position esimate has a large amplitude. This would indicate that the controller is not necessarily doing a poor job and I could relate this to the example of the robot controller causing the robot to spin in circles when the IMU is giving faulty outputs. Note that this would not be a sufficient condition to show that the controller is performing properly but would only be an indication that the KF output needs improvement. There are likely ways of assessing controller performance if the KF output variance is large though. ***

Chapter 6

Future Work

*** Suggest avenues of study for future work. ***

Chapter 7

Conclusion

*** Summarize the results here. ***

Appendix A

Source Code

*** Put source code here if applicable. Consider putting a list of Acronyms in an appendix. ***

Bibliography

[Abbeel 05] Pieter Abbeel, Adam Coates, Michael Montemerlo, Andrew Y. Ng & Sebastian Thrun. *Discriminative Training of Kalman Filters*. In Proceedings of Robotics: Science and Systems, Cambridge, USA, June 2005.

[Busse 03] Franz D. Busse, Jonathan P. How, James Simpson & Nasa Goddard. *Demonstration of Adaptive Extended Kalman Filter for Low Earth Orbit Formation Estimation Using CDGPS*. Journal of the Institute of Navigation, vol. 50, pages 79–94, 2003.

[Con 06] *Report to Congress: Development and Utilization of Robotics and Unmanned Ground Vehicles*. Technical report, Office of the Under Secretary of Defense, Acquisition, Technology and Logistics, Portfolio Systems Acquisition, Land Warfare and Munitions, Joint Ground Robotics Enterprise, October 2006.

[Everett 02] H.R. Everett, R.T. Laird & M.R. Blackburn. *After Action Report to the Joint Program Office: Center for the Robotic Assisted Search and Rescue (CRASAR) Related Efforts at the World Trade Center*. Technical report, Space and Naval Warfare Systems Center, San Diego, August 2002.

[Kelly 94] Alonzo Kelly. *A 3D State Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles*. Technical Report CMU-RI-TR-94-19, Robotics Institute, Pittsburgh, PA, May 1994.

[Khalil 02] Hassan K. Khalil. Nonlinear Systems. Prentice Hall, Inc., Third edition, 2002.

[Mehra 72] Raman K. Mehra. *Approaches to Adaptive Filtering*. October 1972.

[Rusu 05] Radu Bogdan Rusu & Marius Borodi. On Computing Robust Controllers for Mobile Robot Trajectory Calculus: Lyapunov. Unpublished technical report, 2005.

[Sights 06] B. Sights, E.B. Pacis, G. Ahuja, G. Kogut & H.R. Everett. *An Adaptive Localization System for Outdoor/Indoor Navigation for Autonomous Robots*. In SPIE Proceedings 6230: Unmanned Systems Technology VIII, Defense & Security Symposium, April 2006.

[Simon 06] Dan Simon. Optimal State Estimation. John Wiley & Sons, Inc., First edition, 2006.