

Guidance & Navigation System for an Autonomous Lawnmower: CWRU Cutter

B. Hughes, Case Western Reserve University/ENSCO, Inc.; J. Beno, A. Schepelmann, D. Bennett, H. Snow, K. Daltorio, Case Western Reserve University; J. Green, MTD Products, Inc.; R.D. Quinn, Case Western Reserve University

ABSTRACT

"CWRU Cutter" pronounced "crew cutter," is an autonomous lawnmower developed at Case Western Reserve University (CWRU) in collaboration with MTD Products, Inc. for the annual Institute of Navigation (ION) Autonomous Lawnmower Competition. Unlike commercially available autonomous mowers, CWRU Cutter is able to accurately navigate a series of global waypoints via several predefined path primitive types. A series of connected path primitives are automatically generated according to user defined parameters such as desired cutting pattern, deck overlap, and globally registered mowing boundaries. To achieve navigation superior to competitors, CWRU Cutter employs an Extended Kalman Filter (EKF) to estimate the robot's position, velocity, and orientation. A commercial off the shelf differential global positioning system (GPS) receiver is used to obtain a globally registered state observation. L1/L2 dual frequency antennae are used in conjunction with a wide-area differential GPS system to mitigate ionospheric and atmospheric effects on GPS pseudo ranges for both the rover and the base station. An inexpensive COTS inertial measurement unit (IMU) provides acceleration and angular rate measurements. Motor shaft encoders provide wheel displacement observations. To achieve low level obstacle avoidance, outputs from a light detection and ranging (LIDAR) unit and a FireWire camera are combined and passed to a reactive control algorithm. Color and texture processing of monocular vision is used to determine the location of obstacles as well as the location of the globally registered field boundary lines. More recent positioning work includes the evaluation of a Scale Invariant Feature Transform (SIFT) key point based visual odometry estimation algorithm for use in GPS occluded areas. Future work includes the integration of SIFT-based simultaneous localization and mapping (SLAM.) The basic guidance and navigation system was successfully implemented for the 2009 ION Autonomous Lawnmower Dynamic Competition where CWRU Cutter won 1st place.

Index Terms—Intelligent Mobile Robots, Sensor Integration, State Estimation

I. INTRODUCTION

CWRU Cutter is a mobile robot designed to compete in the 2009 ION Autonomous Lawnmower Competition. The original CWRU Cutter was designed to test the

feasibility of autonomously mowing grass with a high quality of cut. The primary objective of CWRU Cutter 2 is to improve its ability to accurately and robustly mow predefined paths and perform reactive edging around stationary obstacles. The second objective of CWRU Cutter is to provide a platform to test less expensive, commercially viable sensors that will allow a safe and robust autonomous lawn mower to be brought to market. The first objective was met by incrementally improving the hardware, low level controller, and sensor implementations. More powerful motors and an improved speed controller provide the low level computer with improved, precise control of mower motion over rough lawn surfaces. CWRU Cutter 1's electric mower deck was used, as it performed well and created an aesthetically pleasing lawn. A string trimmer has been added to the side of the mower deck, allowing the mower to edge along hard obstacles. Several improved sensors have been added to the mower, including differential GPS with Omnistar correction, a servo driven LIDAR (Light Detection and Ranging) mount to provide a more robust solution for finding the edge of low obstacles in grass, and a vision system that uses color and texture information to differentiate between grass and obstacles. These sensors complement the wheel encoders and the inertial measurement unit to provide an accurate physical state estimate of the robot. In order to produce a commercially viable autonomous lawn mower, the limitations of using cheaper sensors must be fully understood. The second objective was met by the vision system, as cameras are widely available and less expensive than LIDAR. Even though the cost of vision systems is generally expensive in terms of computational power, practicality of vision systems has been increasing due to the cost of processors and memory continually decreasing. Decreasing costs make a camera a reasonable sensing option for a commercial mower. SODAR (Sound Detection and Ranging) sensors were tested to find if they can be used for either obstacle detection or edging. A senior design project utilized an ultrasonic transducer to emit pseudo-random noise and find the distance to multiple obstacles. The goal of this project was to use an array of these sensors and obtain robust obstacle data that could replace the LIDAR unit at a fraction of the cost. CWRU Cutter has a modular controller that can be upgraded with input and output modules allowing any

sensor to be added to complement or replace a current sensor. For example, a cheaper GPS unit can quickly be added to the mower and the data can be logged to determine if the sensor could be used to control the robot by comparing it to the high quality data from the Novatel receiver. The mower could then be controlled using the cheaper GPS sensor, while logging the more accurate GPS data for comparison and tuning purposes.

II. HARDWARE

A. Mechanical

CWRU Cutter 2 is a four-wheeled mobile robot that uses a commercially available mower deck and string trimmer supplied by MTD Products, Inc., electric wheelchair components, and a custom manufactured frame. A waterproof Pelican case was used to house the electronics.

Mower Deck

Compared to conventional gas mowers, electric lawnmowers produce less noise, which is favorable for products intended for residential use. Since the unit is battery powered, it is easy to recharge and is more environmentally friendly, since it requires no gasoline and produces no emissions while the unit operates.

String Trimmer

A string trimmer is mounted on the unit's side. This allows CWRU Cutter to edge effectively, which increases the final quality of the cut lawn. The string trimmer is provided by MTD and operates on 12V. The string trimmer also comes from MTD's Troy-Bilt product line. We chose this model for its reputation, reliability, and cost effectiveness.

Drive Motors & Wheels

The drive motors used on CWRU Cutter are sourced from Invacare, a powered wheelchair manufacturer. These powerful motors allow the mower to drive over ruts and bumps without hesitation. CWRU Cutter 1's motors were underpowered and struggled to move and pivot over rough terrain and in heavy grass. This is no longer an issue for CWRU Cutter 2. The drive wheels and hubs are a matched set with the motors and therefore bolt right on. The tire diameter, width, and tread pattern are appropriate for use on grass. Wheel slip and rutting are only a problem on very wet grass, when the grass should not be mowed anyways.

B. Electrical

E-Stop & Safety System

CWRU Cutter has two E-Stop safety chains. If any one component in either chain is disabled, the corresponding subsystem is deactivated. The master E-Stop chain controls the amount of power delivered to the drive wheels. The master E-Stop is also a dependency of the blade/trimmer E-Stop chain.

The master E-Stop chain requires the following acknowledgements for operation:

1. Main E-Stop Switch
2. Software Controlled Master Enable Relay
3. Remote E-Stop Relay

The robot has a basic safety state machine that governs the software-activated E-Stop acknowledgements. The safety state machine allows the robot to drive after five seconds of audible alert via a siren. The controller must also provide a "heart beat" monitored via a watchdog timer for the software relay to be activated.

The blade E-Stop chain requires the following acknowledgements for operation:

1. Blade E-Stop Switch
2. Master E-Stop Relay
3. Software Controlled Blade and Trimmer Relays

Speed Controller Selection

The selection of speed controllers directly affects several factors of CWRU Cutter including controllability, durability, and safety. A few different models of speed controllers have been tested. Victor speed controllers are robust against failure but exhibit a highly nonlinear characteristic response which makes control difficult. Dimension Engineering Sabertooth 2X25 speed controllers are very linear and have a variety of input options. However, the Sabertooth controllers are less robust. Therefore, we chose to use a Roboteq AX2550 speed controller. The Roboteq controller includes a variety of input options, internal circuit protection, a robust form factor, and excellent response characteristics. The Roboteq controller was chosen for the main drive wheels of the robot. The Sabertooth speed controller was chosen for the gimbaled LIDAR mount due to the mount-specific application requirements of reduced power consumption.

Battery Selection

All motor and electronics power on the vehicle comes from 12 8.5Ah 12V lead acid batteries. Four sets of two series batteries are wired in parallel providing 34 Ah at 24 V. One set of four batteries are wired in series providing 8.5 Ah at 48 V. 24 V and 48 V battery chargers are employed to charge the batteries while on the robot. However, batteries are individually charged at 12 V to balance batteries in the set. Primary consideration was given to power retention and maintenance. Lead-acid batteries are capable of retaining their charge for approximately 2 months. This makes them ideal for lawn mowing applications, as a commercial unit would only be in operation two to three times in a thirty day period. Batteries with alternative chemistries can offer more capacity for equal volume and mass. However, increased cost and more complicated charging requirements lead us to choose lead acid batteries.

C. Computational

CWRU Cutter '08 featured a single National Instruments

CompactRIO RT (real time) programmable automation controller and an Apple Mac Mini NRT (non real time) computer running Windows XP. The NI cRIO (National Instruments Compact Reconfigurable Input/Output Controller) platform includes two computation subsystems: a FPGA (Field Programmable Gate Array) and a 500MHz Real Time processor. The FPGA is a common digital logic device that implements application specific high-speed timing, data acquisition, and control functionality. The Real Time processor, using data acquired by the FPGA, implements higher level computations and outputs the subsequent control commands back to the FPGA. The combination of these two devices allow for a robust software platform. The FPGA interfaces to pluggable modules, defining the RIO functionality. CWRU Cutter 2 features two cRIO systems and a variety of modules. The Mac Mini facilitates an interface with standard Firewire and/or USB cameras for vision processing. A solid state hard drive was chosen to maximize system robustness in the robot's environment.

D. Sensory

Encoders

Encoders attached to the motor shaft of the drive wheels provide wheel speed and wheel position information. The encoders chosen for the application are Grayhill 63R256 256-count quadrature encoders. A 24 to 1 transmission on the output of the motor provides approximately 22,542 ticks/meter. The large number of ticks allow for backwards differentiation of the position with respect to time. This differenced value can be used directly as a motor velocity for control purposes. The large number of counts per meter also allows accurate integration of the robot's position over time. Encoders attached to the motor shaft of the gimbaled LIDAR mount's motors provide both angular feedback and motor velocity feedback. The encoders chosen for this application are installed during manufacture, and as a result, are Maxon 1000-count quadrature encoders. A 74 to 1 transmission on the output of the motor provides additional ticks per revolution of the control hub. An empirical estimation of this value is 296,000 ticks/revolution of the hub.

Inertial Measurement Unit

An IMU (Inertial Measurement Unit) is mounted at the center of the drive axle of the robot. The selected IMU, a Cloud Cap Technology Christa IMU is a six-axis unit that provides angular velocity and acceleration in each Cartesian axis. The IMU provides real time inertial measurement at 100Hz to the controller via RS-232 serial.

Global Positioning System Receivers

Two GPS (Global Positioning System) receivers are used to provide the robot with a global estimate of position, velocity, and heading. Differential GPS has been selected to minimize

the effects of ionospheric lensing and stratospheric bias. The L2 frequency band, used for Omnistar HP corrections, allow for further corrections due to signal lensing. The selected GPS receivers for our robot are Novatel ProPak-V3 units. The robot's GPS antenna is located directly above the IMU to allow for minimal calibration of the lever arm between the two sensor centers. The secondary GPS antenna, used for real time differential GPS corrections, is mounted on a fiberglass surveying tripod and is anchored within fifty meters of the course. A set of Freewave FGR-115RC RS-232 serial packet radios periodically transmit corrections from the differential GPS receiver to the robot.

LIDAR (Light Detection And Ranging)

CWRU Cutter 1 featured a fixed LIDAR mounted with the laser plane just above and parallel to the top of the grass. Theoretically this allows the flowerbed obstacle to be observed while the grass is not seen. In practice tall tufts of grass and robot pitching/rolling prevent this and the LIDAR observes the grass. Subsequently, the robot senses an obstacle and stops immediately, to avoid crashing into the ground. To remedy this, the LIDAR was raised during the competition so the moving obstacle and fence were still detected, and CWRU Cutter 1 planned to stay far enough away from the flowerbed to avoid collisions. This is obviously a less than optimum solution to the problem. Several variations of the gimbaled mount were considered before the final design was chosen and manufactured, but all shared a common feature. The LIDAR mirror center is mounted well above the height of the grass and looks down. This gives the sensor a better perspective and allows for the creation of a local polar height array. The edge of the obstacle can be determined from this data and the mower will never be startled by grass immediately in front of it. CWRU Cutter 2 uses a SICK LMS291-S14. It provides a 90 degree sweep of data directly in front of the sensor in 0.5 degree increments. At each point, the reflectivity of the surface is also returned, which is valuable in determining the difference between grass and manmade objects. The distance at each scan is reported to 1 millimeter resolution up to 8 meters. The sensor outputs data at a rate of 75 scans per second. The gimbaled mount tilts from just above the horizon to 30 degrees below the horizon. It can also be panned side to side 45 degrees each way. The rotation is driven with a crank shaft and connecting rod to convert continuous rotation of the motor into sinusoidal motion of the mechanism. This is desirable because any backlash in the mechanism is taken out because it is always driven in one direction. Controlling the mechanism and maintaining position synchronization is much simpler with this arrangement as well. A figure 8 pattern was chosen for the mode shape the LIDAR should travel through, as it gives a vertical sweep of data to the right, center and left of the robot with each sweep, which occurs about every two seconds.

Camera

CWRU Cutter utilizes a Unibrain Fire-i Firewire camera to capture images on the Mac Mini. The camera is mounted on the robot to maximize a specific region of interest that will interact and edge obstacles. The camera outputs 320x240 pixel images at a frame rate of up to 30 frames per second. A wide-angle lens was employed to maximize the observable area around the robot.

III. SOFTWARE

National Instruments' LabVIEW graphical programming language has been selected as the team's primary programming tool. Ease of use, interactive real-time debugging, and multiple platform deployment are only a few highlights of LabVIEW's vast feature set. LabVIEW code can be executed on Windows, Mac, and Linux operating systems as well as compiled for several hardware platforms including FPGAs. The selected cRIO PACs (programmable automation controllers) natively run LabVIEW in real time

A. CWRUCutter-Brain cRIO

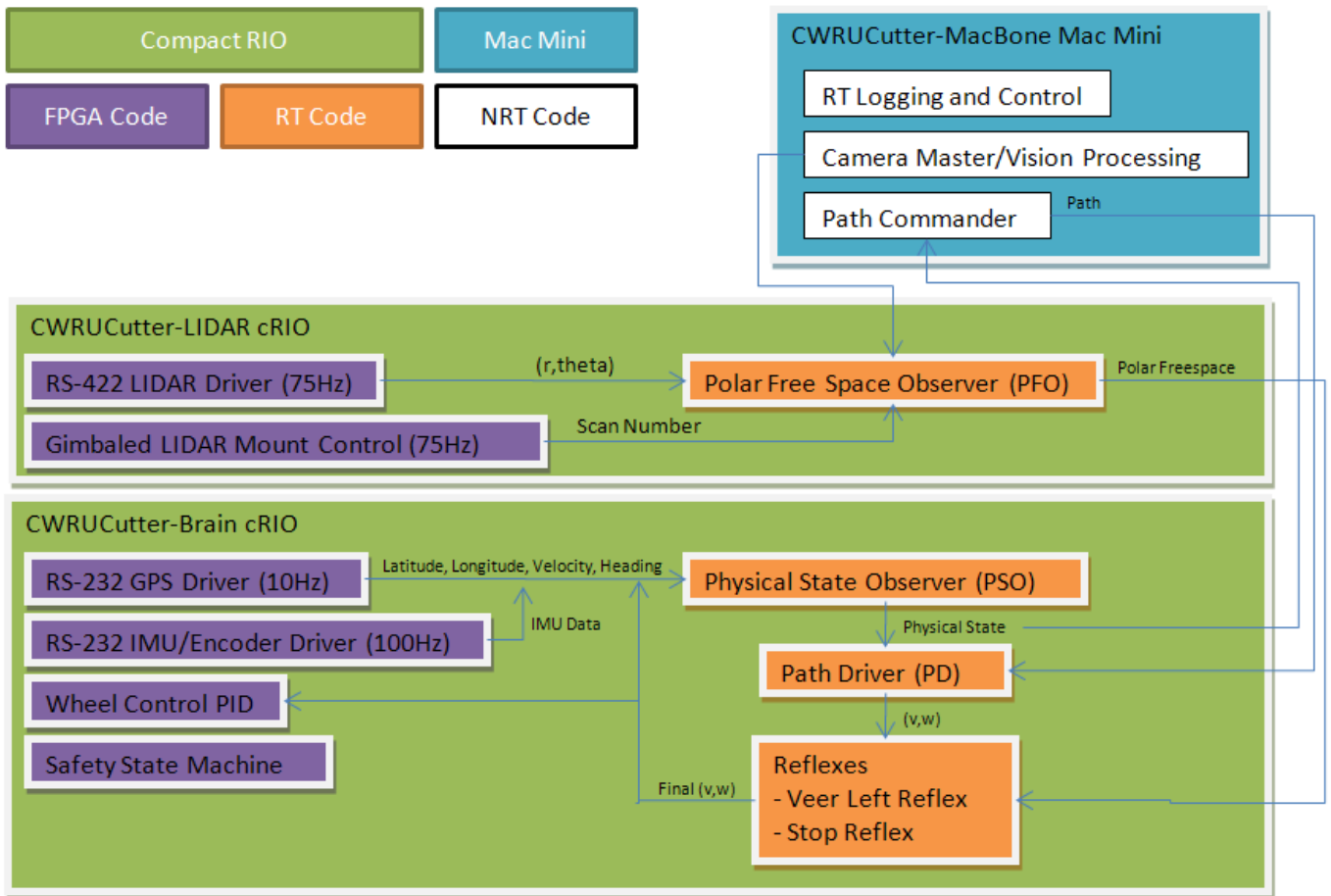
The CWRUCutter-Brain compactRIO is responsible for a series of tasks. Serial drivers, sensor interfacing, and closed loop wheel speed control are the responsibility of the FPGA side of this cRIO. The RT computer side of this cRIO is responsible for physical state estimation, path driving, and reflexive behavioral control.

RS232 Drivers

LIDAR, GPS, IMU, and encoder data are interfaced through the FPGA. All critical data for the control of the robot are fed in through 32-bit DMA channels to avoid communication problems and system lag with passing around data. The data is available immediately after an entire message is received from each sensor.

Wheel Control PID

Wheel speed is controlled using a PID controller that operates on the FPGA at 100 Hz. Velocity commands are read from the RT controller at a rate of 10 Hz from the Path Driver in the real time control loop. Wheel velocity is



on the vxWorks operating system.

A software architecture has been designed to allow for the independent development of modular software components.

measured directly on the FPGA and passed to the RT controller for state estimation purposes.

Safety State Machine

Several requirements are constantly checked at a very low level in the control for the robot to be allowed to enter or maintain an active state. A heartbeat is triggered from the real time controller at 10 Hz to indicate an error has not occurred and the program is running. If this is not received in .25 seconds, the mower is immediately stopped. The state of the remote E-stop and push button E-stops are checked, and if either is triggered the mower is stopped immediately.

Physical State Observer

Since a variety of sensors are available for use in state estimation and each sensor has its own downfall, the Kalman Filter can be used to probabilistically determine the most accurate state estimate. Moreover, the Kalman filter will estimate the state recursively and iteratively in real time – constantly driving the uncertainty of the solution downward. A dynamic system is defined that follows the standard form

$$\underline{x}_k = \phi_k \underline{x}_{k-1} + \Gamma_k \underline{u}_{k-1} + \underline{w}_{k-1}$$

where F_k relates the previous state to the current state, \underline{x}_k is the state at time k , \underline{x}_{k-1} is the time at $k-1$, B_k is the control input model, \underline{u}_{k-1} is the control vector, and w_{k-1} is the process noise at time $k-1$ that is assumed to be zero mean Gaussian white noise defined by

$$\underline{w}_k \sim N(0, \mathbf{Q}_k)$$

where \mathbf{Q}_k is the covariance matrix that defines the process noise.

Given an observation of data, a mapping to the state vector can be assumed by knowing the observation follows

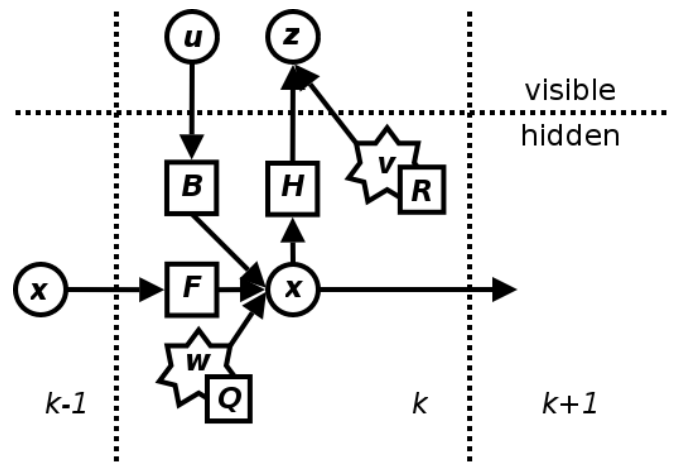
$$\underline{z}_k = \mathbf{H}_k \underline{x}_k + \underline{v}_k$$

where \mathbf{H}_k transfers the observation into state-space, \underline{x}_k is the true state at time k , and \underline{v}_k is the measurement or observation noise described by

$$\underline{v}_k \sim N(0, \mathbf{R}_k)$$

where \mathbf{R}_k is the covariance matrix that defines the observation noise.

From these equations, it only makes sense that inherently the observation noise is part of the observed state \underline{z}_k .



The Kalman filter will find the best solution to fit the previous equations by not only estimating the solution, but the solution's covariance as well. The Kalman filter has two main sections. First, the predictor part of the filter runs to attempt to form a ballistic estimate of the solution. This solution estimate is then updated with sensor measurements as they are received. The Kalman filter does not require sensors measurements to come regularly or in order. The solution will be valid and the same regardless the order they are processed. This aspect supports a robust platform for robotics so that if a sensor stops communication, the solution will remain valid but have an increased uncertainty.

The prediction update equations are

$$\underline{x}_k^- = \phi_k \underline{x}_{k-1}^- + \Gamma_k \underline{u}_{k-1}$$

$$\mathbf{P}_k^- = \phi_k \mathbf{P}_{k-1}^- \phi_k^T + \mathbf{Q}_{k-1}$$

where \underline{x}_k^- is the priori estimated state at time k and \mathbf{P}_k^- is the priori estimated covariance at time k .

The minus superscript symbol on the state vectors and covariance matrix \mathbf{P}_k^- indicates a ballistic priori update to the state vector. In other words, this indicates that the solution is somewhat open-loop and requires some feedback or it will diverge from the proper solution.

The measurement update equations are

$$\underline{y}_k = \underline{z}_k - \mathbf{H}_k \underline{x}_k^-$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}_k^{-1}$$

$$\underline{x}_k^+ = \underline{x}_k^- + \mathbf{K}_k \underline{y}_k$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$$

where \underline{y}_k is the measurement innovation or measurement residual, \mathbf{S}_k is the covariance residual, \mathbf{K}_k is the optimal Kalman gains derived from the assumptions of our dynamic model, \underline{x}_k^+ is the posteriori estimated state, and \mathbf{P}_k^+ is the posteriori estimated state covariance.

The plus superscript symbol on the state vectors and covariance matrix \mathbf{P}_k indicate that feedback was incorporated into the solution. On intervals that a measurement update is processed, the prediction state is still processed first. In our implementation, we have several sensor measurements. The corresponding \underline{z}_k , \mathbf{H} and \mathbf{R} matrices are presented for each measurement. First, we define our state vector:

$$\underline{x} = [x \quad y \quad z \quad v_x \quad v_y \quad v_z \quad \theta \quad \varphi \quad \psi \quad \omega_x \quad \omega_y \quad \omega_z]^T$$

where x , y , and z are the position of the robot, the next three variables are its component velocities, the next three variables represent the roll, pitch, and yaw, and the final three variables represent the angular rate in each axis of rotation (roll, pitch, and yaw.)

The measurements are:

- GPS Position Measurement

$$\underline{z}_{k, gpspos} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\mathbf{R}_{gpspos} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix}$$

$$\mathbf{H}_{gpspos} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where x , y , and z are the east, north, and up positions on the earth.

- GPS Velocity Measurement

$$\underline{z}_{k, gpsvel} = \begin{bmatrix} v_{horiz} \\ v_{vert} \\ \Psi \end{bmatrix}$$

$$\mathbf{R}_{gpsvel} = \begin{bmatrix} \sigma_{v,horiz}^2 & 0 & 0 \\ 0 & \sigma_{v,vert}^2 & 0 \\ 0 & 0 & \sigma_{\Psi}^2 \end{bmatrix}$$

$$\mathbf{H}_{gpsvel} = \begin{bmatrix} 0 & 0 & 0 & \cos \Psi_k & \sin \Psi_k & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

where Ψ_k is the last best Kalman estimate of heading (yaw), Ψ is the measured heading (yaw), v_{horiz} is the measured horizontal velocity of the robot, and v_{vert} is the measured vertical velocity of the robot.

- Wheel Speed Measurement

$$\underline{z}_{k, wheel} = \begin{bmatrix} V_L \\ V_R \end{bmatrix}$$

$$\mathbf{R}_{gpsvel} = \begin{bmatrix} \sigma_{wh.vel.}^2 & 0 \\ 0 & \sigma_{wh.vel.}^2 \end{bmatrix}$$

$$\mathbf{H}_{gpsvel} = \begin{bmatrix} 0 & 0 & 0 & .5 \cdot \cos \Psi_k & .5 \cdot \sin \Psi_k & 0 & 0 & 0 & 0 & 0 & 0 & -T^{-1} \\ 0 & 0 & 0 & .5 \cdot \cos \Psi_k & .5 \cdot \sin \Psi_k & 0 & 0 & 0 & 0 & 0 & 0 & T^{-1} \end{bmatrix}$$

where V_L and V_R are the measured left and right wheel speeds of the robot, Ψ_k is the best estimate of heading at time k , and T is the track of the robot. (The track of the robot is the distance between the two differential drive wheels.)

- IMU Measurement

$$\underline{z}_{k,imu} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

$$\mathbf{R}_{imu} = \begin{bmatrix} \sigma_{imu,rate}^2 & 0 & 0 \\ 0 & \sigma_{imu,rate}^2 & 0 \\ 0 & 0 & \sigma_{imu,rate}^2 \end{bmatrix}$$

$$\mathbf{H}_{imu} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Path Driver

A priority for CWRU Cutter 2 is to mow straight and parallel paths rather than random paths driven by other commercial autonomous mowers. In addition to line segments, the mower also needs to be able to either pivot or arc to get between lines. Pivoting (attempting to maintain a zero net velocity while turning) is especially important in tight spaces. In addition to making complex cutting shapes possible, arcs are necessary for turning in thick grass where pivots were found to cause stalls. Our path driver follows sets of three different types of path segments: lines, arcs, and pivots. For example, a serpentine lawn-mowing pattern might be a line north, followed by a tight 180 degree arc, followed by an offset line south, etc. For our mower, we generate these paths offline after measuring the GPS coordinates of the field corners.

Reflexes

A look-up table of swept areas was generated for representative (v,w) couples at given maximum decelerations. For intermediate (v,w) couples, the values were determined by two-dimensional linear interpolation. It is assumed that CWRU Cutter 2 continues for two timesteps at the given (v,w). This takes into account that by the time the mower has decided what command to use, the information is already one timestep old. The positions of the mower at these times are not included in the determination of the swept volume, because by the time actions take affect this area has already been covered. Allowing the swept area to be shifted in the direction of the proposed motion allows the robot to move away from obstacles that impinge slightly upon its current footprint. CWRU Cutter 2's position as it slows with the maximum allowable deceleration along a constant curvature path is determined at every tenth of a timestep until it comes to a stop. For each determined body position, the polar ranges of the body with respect to some origin fixed on the original body are determined. This simulates what the LIDAR and camera, kept at their original

locations, would see at each of the footprint boundary edges. The maximum polar value at each angle of every range is kept. This is a polar representation of the area the footprint would cover if it were stopping as fast as possible.

B. CWRUCutter-MacBone Mac Mini

Simulation and Logging

We tested real-time reflexes and planning in a LabVIEW simulation environment. With simulation, we can test the controller code while simulating the physical state observer, LIDAR, and kinematics of the system. In real time operation, logs of all sensory input and physical state variables are logged together so they can be played back later for analysis or used as an input to the simulation environment to test a new piece of code before investing time running the code on the robot.

Camera Master/Vision processing

CWRU Cutter 2 utilizes a Unibrain Fire-i firewire camera to detect grass and obstacles based on visual texture and color. Frames are captured and processed at 10Hz by the Mac Mini. Lighting conditions and image exposure vary drastically in an outdoor setting. To create a robust system that functions in varying lighting conditions, a template region that remains grass throughout the mowing cycle is defined in the frames. This creates a local, dynamic template that is automatically updated in each frame and negates changing brightness and color values in the rest of the image.

Location of the template region is based on CWRU Cutter mowing and edging behavior. Grass texture is extracted from the image via Sobel edge detection. Visual texture is defined by the collection of edges in an image region. Grass has a distinct, "messy" texture which is easily differentiable from other surfaces found in yards, like mulch, fences, and flower bed linings. The Sobel algorithm calculates the gradient between a pixel and its 8 neighbors, providing information about the rate of color change around a pixel. If the color changes rapidly between a pixel and its neighbor, it is likely that an edge is present. The magnitude of the gradient is plotted at the center pixel, creating an intensity plot of edges. Regions in the image are compared to the template region. If the mean intensity value of the current region is within the standard deviation of the template mean, all pixels in the current region are identified as grass and placed in a binary array. Color is used to identify obstacles. Since obstacles in the course have a distinct color they can easily be identified in an image. Red, green, and blue color planes are extracted from the original frame and corresponding pixel values are compared. Since the obstacles in the course are either black or white, they have the same ratio of red, green, and blue color values. Therefore, the ratios of corresponding pixels in the extracted

color planes are compared. If the ratios of the corresponding pixels are approximately equal to 1, they are identified as obstacles and placed in a second binary array.

The obstacle array is then subtracted from the grass array, providing a clear picture of what constitutes traversable terrain the image. The resulting data is then abstracted to resemble LIDAR data by calculating ranges between the optical origin in the frame and the nearest obstacle. This data is then combined with the LIDAR to provide a clear indication of how far away a mower is from an obstacle.

Path Commander

The path commander manages a longer queue of the path commands and passes down the next command when the Path Driver is on the second command.

The planner generates edging commands around the entire playing field and sweeps for endpoints along the desired diagonal cut lines. The planner integrates directly with the path commander, which pops off the current and next planner commands.

CWRU Cutter 2 modifies its behavior based on its location on the field. The string trimmer is only enabled when the mower is close to an obstacle. If CWRU Cutter travels into the safety buffer, it stops. If it encounters a non-moving obstacle in a zone it expects to be clear, it travels in a wider-than-normal path around the obstacle because the observation could represent a person or other dynamic obstacle.

[1] "7 Secrets to a Perfect Lawn and Yard," Consumer Reports, May 2008, pp. 23.

[2] Ritzmann, R.E., "The Neural Organization of Cockroach Escape and Its Role in Context Dependent Orientation," Chapter VI in Biological Neural Networks in Invertebrate Neuroethology and Robotics, R. Beer, R. E. Ritzmann, and T. Mckenna, Eds.: Academic Press, 1993.

[3] Chen C., Quinn R., and Ritzmann R., "A Crash Avoidance System Based Upon the Cockroach Escape Response Circuit," in IEEE International Conference on Robotics and Automation (ICRA) Albuquerque, NM, USA, 1997.

[4] <http://svnbook.red-bean.com/en/1.2/svn.advanced.locking.html>

[5] http://en.wikipedia.org/wiki/Image:Kalman_filter_model.png

[6] Kalman, R.E. "A New Approach to Linear Filtering and Prediction Problems." Transactions of the ASME--Journal of Basic Engineering D ser. 82 (1960): 35-45.

[7] Grewal, M., et. all. Global Positioning Systems, Inertial Navigation, and Integration. 2. Hoboken, NJ: Wiley, 2008, pp. 394.

[8] Kanayama, Y. and Fahroo, F., "A New Line Tracking Method for Nonholonomic Vehicles," Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, April 1997, pp. 2910-2911.

[9] Kyodo America. Kyodo America - LawnBott - How LawnBott works. Kyodo America - LawnBott. [Online] Kyodo America, 2009. [Cited: May 1, 2009.] <http://www.lawnbott.com/products/howitworks/>.

[10] RoboMow Canada. RoboMow Canada FAQ. RoboMow Canada - It Mows. You Don't. [Online] Friendly Robotics, 2009. [Cited: May 1, 2009.] <http://www.robomowercanada.com/faq.html>.

[11] Friendly Robotics. Safety Warnings & Precautions. Safety Warnings & Precautions - Friendly Robotics. [Online] Friendly Robotics, 2009. [Cited: May 1, 2009.] <http://www.friendlyrobotics.com/cservice/manuals/?pg=1>.