Multi-Sensor, High Speed Autonomous Stair Climbing

Daniel M. Helmick[1], Stergios I. Roumeliotis[2], Michael C. McHenry[3], Larry Matthies[4]

[1]*NASA - Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, dhelmick@robotics.jpl.nasa.gov*
[2]*California Institute of Technology, Pasadena, CA, stergios@robotics.caltech.edu*
[3]*Rossum Technologies, Inc., Long Beach, CA, mchenry@rossumtech.com*
[4]*NASA - Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, lhm@telerobotics.jpl.nasa.gov*

Abstract

Small, tracked mobile robots designed for general urban mobility have been developed for the purpose of reconnaissance and/or search and rescue missions in buildings and cities. Autonomous stair climbing is a significant capability required for many of these missions. In this paper we present the design and implementation of a new set of estimation and control algorithms that increase the speed and effectiveness of stair climbing. We have developed: (i) a Kalman filter that fuses visual/laser data with inertial measurements and provides attitude estimates of improved accuracy at a high rate, and (ii) a physics based controller that minimizes the heading error and maximizes the effective velocity of the vehicle during stair climbing. Experimental results using a tracked vehicle validate the improved performance of this control and estimation scheme over previous approaches.

1 Introduction

As a significant part of many urban reconnaissance and/or search and rescue mission scenarios, stairs are formidable and critical obstacles. Being able to autonomously climb stairs in a fast, efficient, and robust way could mean the difference between a successful mission and an unsuccessful one.
A small, tracked robot (see Fig. 1) designed for general urban mobility is used in this research to develop the algorithms necessary to autonomously navigate stairs at high speeds.

Previous approaches to autonomous stair climbing for tracked vehicles are presented in [1], [2] and [3]. In [1], the Andros VI mobile robot relies on a set of accelerometers to measure its attitude. The authors assume that the vehicle body accelerations due to the interaction of the tracks with stairs are insignificant compared to the gravitational acceleration. A low pass filter is designed for reducing the effect of the external disturbances before communicating the attitude estimates to the track speed controller. A simple kinematic model is employed for the design of a proportional control law. The gains and the bandwidth of the controller are determined experimentally.

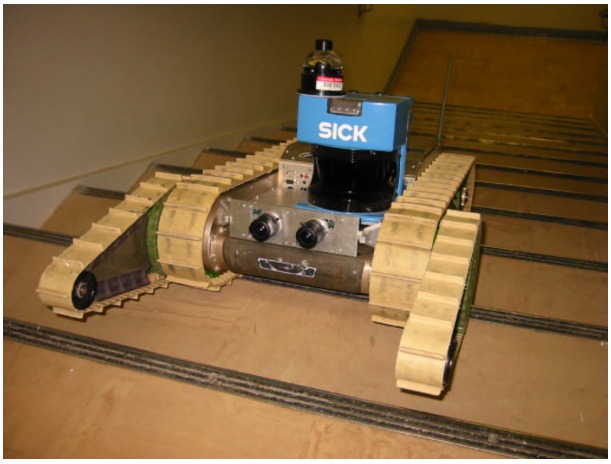

Figure 1: *Vehicle climbing stairs.*

In [2], a similar PD controller is implemented which is also based on the kinematics of the vehicle. The main contribution of this work is the incorporation of a variety of sensors for determining the attitude of the robot: (i) a set of accelerometers that measure the local projection of the gravitational acceleration, (ii) a vision-based edge detection sensor module, and (iii) an array of sonars that measure distances to the sides of the stairwell. A rule-based arbiter was developed for deciding when to process information from each of these sensors. No more than one sensor was used at the same time.

Finally, in [3] a single forward-looking camera was used as the basic navigation sensor. Edge detection algorithms applied to the camera images allowed for estimation of heading angle, θ, and center position

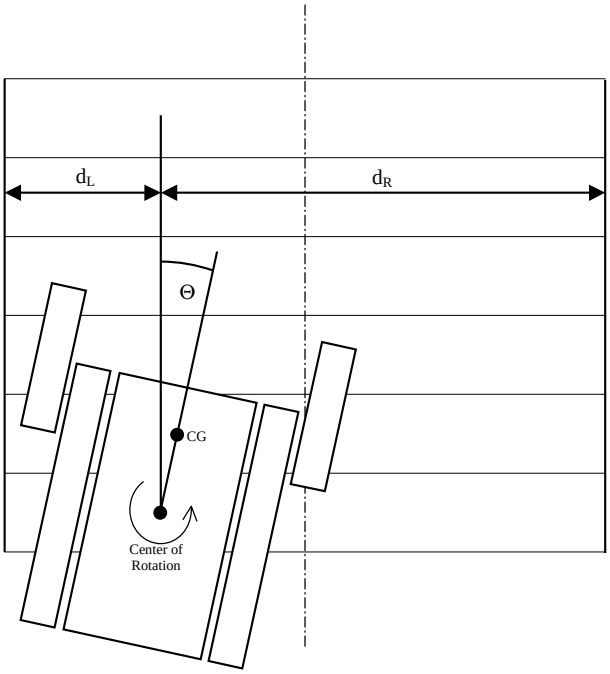

Figure 2: *Diagram of vehicle.*

$\frac{d_L}{d_R}$ (see Fig. 2) at approximately 4 Hz. Heading angle and center position were then heuristically combined to regulate the two track speeds to keep the vehicle heading directly up the stairs while remaining in the center of the staircase. Due to the time between measurements of 0.25 sec, the top speed of the vehicle during stair climbing was limited by this approach. Here, similar to [1] and [2], a simplified kinematics-based proportional controller was employed for keeping the vehicle perpendicular to the stair edges. In addition, a separate control law was implemented for steering the vehicle towards the center of the stairwell when approaching its sides. This work is described in more detail in Section 3.

In this paper we present a set of new estimation and control algorithms for improving the speed, accuracy, and effectiveness of autonomous stair climbing. Our main motivation has been the introduction of a new mechanical chassis with enhanced capabilities in terms of torque and maximum velocity. Specifically, the new version of the robot has a top speed of approximately $2.0m/s$ (on flat ground) compared to the previous vehicle's maximum velocity of $0.80m/s$. This increase in speed has resulted in faster dynamics for the vehicle and has amplified the magnitude of the disturbances. These additional challenges accentuated the deficiencies of the previous approach and heightened the necessity for a new set of estimation and control algorithms capable of processing information from a diverse set of sensors and of operating at a significantly higher rate (30 Hz for the controller compared to 4 Hz previously).

The main contributions of this paper are: (i) the development of a Kalman filter estimator for optimally fusing all attitude measurements provided by a variety of sensors, (ii) the implementation and testing, both in simulation and experiment, of a physics based control law for effectively navigating the robot up the stairs while preventing collisions with the sides of the stairwell.

The new stair climbing algorithm can be divided into five discrete components: (i) a vision-based heading and center position estimator, (ii) a laser-based heading and center position estimator, (iii) a Kalman filter to merge the laser/vision data with gyro and tilt sensor data, (iv) a physics-based heading controller, and (v) a centering controller. These algorithms are described in Sections 3 through 6. Section 7 presents the experimental results from testing these algorithms on the actual vehicle. We derive our conclusions in Section 8 and suggest possible directions of future work.

2 Hardware Description

2.1 Vehicle

The vehicle used for this research has a mass of approximately $20kg$ and overall dimensions of $60 \times 50 \times 17$ cm (see Fig. 1). It has three kinematic degrees of freedom (DOF): two independently controlled motors turn the main tracks on the sides of the vehicle as well as the tracks on the arms, and one motor turns both of the arms about a pivot point at the front of the vehicle. The tracks are made of molded rubber and are approximately $7.5cm$ wide, with $1cm$ high cleats oriented perpendicular to the motion of the track and spaced about $4cm$ apart.

2.2 Sensors

The sensor suite used for these algorithms includes the following sensors: 3 gyroscopes, a 2 DOF electrolytic tilt sensor, a pair of cameras, and a LADAR. Only one of the cameras was used by the edge detection algorithm described in Section 3.

The gyroscopes are Systron-Donner QRS11 solid-state gyros with operational ranges of $\pm 200^\circ/s$. The tilt sensor is part of a Precision Navigation TCM2-50 magnetometer and tilt sensor package and has a range of $\pm 50^\circ$ of roll and pitch. The stereo pair consists of two Videology 20VC3405 B/W cameras with frame-grabbers that provide 640 x 480 resolution. The LADAR is a SICK LMS-200, a single axis laser scanner with a 180° field of view and a 75 Hz scan rate. The range of the scanner is calculated using a pulsed time-of-flight measurement with a 905 nm laser. This sensor is capable of 0.25° sample spacing and in its default mode has a maximum range of 8 meters, 1 mm resolution and ± 5 mm accuracy.

3 Vision Algorithm

The vision algorithm incorporated in our system estimates vehicle heading, θ, and center position, $\frac{d_L}{d_R}$, (see Fig. 2). The interested reader is referred to [3] for a detailed description. Here we summarize its main features.

The algorithm is divided into two consecutive steps. The first of these steps is *edge detection and linking*. In order to increase robustness to varying conditions, such as those within shadowy and low contrast environments, a low threshold was selected for the edge detection algorithm. While this choice maximizes the likelihood of detection of an existing edge, it also increases the frequency of false positives. Appropriate filtering is introduced to reduce the number of detected edges. These filters include straight line, parallel and close, and length filters. Additionally, linking of small collinear edges that are close to one another is performed.

The second step is to estimate the *heading and center position* from the resulting edges. With the assumption that the vehicle body plane and the stair edge plane are parallel, there exists a direct mapping from the edge endpoints in the image frame to the vehicle heading:

$$\theta = \arctan\left(\frac{-k}{(y_m - y_0)}\right) \tag{1}$$

where k is the slope of the 2D stair edge, y_m is the y-intercept of the edge, and y_0 is one coordinate of the projection center of the image plane. A figure of merit, G_θ, is associated with the estimation of heading:

$$G_\theta = \frac{\sum_{edges} L_h}{\sum_{edges} \sigma_\theta} \tag{2}$$

where L_h is the horizontal length of each detected edge and σ_θ is the associated uncertainty approximated as:

$$\sigma_\theta = \frac{1}{y_m^2}. \tag{3}$$

Finally, the ratio of the distances from the left and right endpoints of the stair edges is calculated with simple distance equations.

4 Laser Algorithm

Like the vision-based algorithm discussed above, the laser algorithm produces both a heading measurement and an estimate of how well-centered the robot is within the stairway. Because the LADAR performs several functions in addition to its stair climbing role, it must be mounted such that the scanning plane is perpendicular to the robot's vertical axis. In this configuration, the LADAR cannot view individual steps while climbing. Instead the laser algorithm detects straight line segments which are presumed to represent solid walls bounding one or more sides of the stairwell. Some stairways are bounded by balustrades (vertical posts) which support a hand railing rather than a solid wall. The current implementation does not handle this case although such an enhancement would be straightforward.

The algorithm first uses a robust estimation technique to find straight line segments within a scan. Those line segments are then used to compute the robot's heading and distances to the walls on each side.

Each individual scan consists of 361 range measurements obtained at 0.5° intervals in a counterclockwise direction. Although the laser is physically rotating at a 75Hz rate, serial throughput limits the effective scan rate to approximately 10 Hz. A sliding window determines a subset of the scan's range measurements which are then fed into the line-fitting algorithm. If a line that adequately approximates that subset is found, the window is extended to include additional measurements and the line-fitting algorithm is applied again. This step is iterated upon until no adequate linear approximation can be identified. At this point the parameters of the line segment are recorded, the window is shifted to a point just past the previous subset, and the same process is repeated.

Fig. 3 shows a single laser scan obtained during a stair-climbing run. The two walls bounding the stairwell are easily identified. The landing is not directly seen in the laser data but begins approximately where the left wall ends. The line fitting

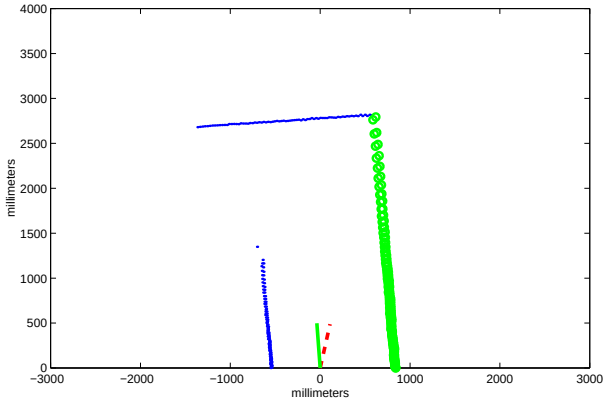

Figure 3: *An example laser scan captured during a stair climbing run. The robot is located at the origin and those range samples which have been determined to be part of the first line segment (in this case the right-hand wall) are drawn as green circles. The short solid green and dashed red lines extending from the origin indicate the laser determined heading and estimated inertial heading respectively.*

itself is based on a random-sampling least median of squares technique [4]. First, a pair of points from the data within the sliding window is randomly selected. The median distance between the line determined by these two samples and the distance to each of the remaining measurements of the data set is then calculated. If the median error (or alternatively the m-th largest error) is within a preset threshold, the data set is determined to be sufficiently approximated by the line containing the two test points. If the threshold is exceeded, a different pair of points is selected at random from the data set and the test is repeated. If, after a predetermined number of attempts, no satisfactory line has been found, the line-fitting portion of the algorithm returns failure.

The extraction of robot heading from the set of lines can be made more robust by making the following assumptions: (i) the robot's heading is always within the interval $[-45°, 45°]$ (where $0°$ denotes the direction directly up the stairs) (ii) the visible bounding walls are either parallel (i.e. the left and right walls) or perpendicular (the wall at the top of the stairs) to the stairway. Using the above assumptions the heading of the robot can be calculated from any of the line segments found within a scan as $mod(\theta_{wall}, \pi/2)$, where θ_{wall} is the direction of the line segment.

Generally, numerous line segments are found within a single scan due to the presence of non-straight line portions of the wall and because of the approximate nature of the line-fitting algorithm. In order to increase estimation accuracy, the measured heading is calculated as the weighted average (by the line segment length) of all the identified line segments. The distances to the left and right walls are computed in a similar fashion.

5 Kalman Filter Based Attitude Estimation

5.1 Dynamic Model Replacement

In order to estimate the attitude of the robot, we have implemented a Kalman filter observer. Sensor modeling was selected instead of dynamic modeling. The main reasons for this are: (i) dynamic modeling would have to be redone every time there is a modification to the robot, (ii) dynamic model-based observers require a large number of states that increase the computational needs without producing superior results [5]. Finally, the convergence of the resulting estimator does not rely on the availability of a precise dynamic model. The observability of the system depends solely on the type of sensor measurements available to the system.[1]

[1]The interested reader is referred to [6] or [7] for a detailed discussion on the subject of sensor vs. dynamic modeling.

5.2 Attitude Kinematics and Error State Propagation Equations

The three-parameter Euler angle representation has been used in most applications of the Kalman filter in robot localization [8, 9]. However the kinematic equations for Euler angles involve non-linear and computationally expensive trigonometric functions. The computational cost using quaternions is less than using Euler angles. It is also more compact because only four parameters, rather than nine, are needed. Furthermore, in the Euler angle representation the angles become undefined for some rotations (the gimbal lock situation) which causes problems in Kalman filtering applications [10, 11]. Among all the representations for finite rotations, only those of four parameters behave well for arbitrary rotations [12].

The physical counterparts of quaternions are the rotational axis, $\hat{n}$, and the rotational angle, θ, that are used in the Euler theorem regarding finite rotations. Taking the vector part of a quaternion and normalizing it, we can find the rotational axis, and from the last parameter we can obtain the angle of rotation [13]. Following the notation in [14] a unit quaternion is defined as:

$$q = \begin{bmatrix} \vec{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} \hat{n} \sin\theta/2 \\ \cos\theta/2 \end{bmatrix} \tag{4}$$

with the constraint $q^T q = 1$, $\hat{n} = [n_x n_y n_z]^T$ is the unit vector of the axis of rotation and θ is the angle of rotation.

The rate of change of the quaternion with respect to time is given by:

$$\frac{d}{dt}q(t) = \frac{1}{2}\Omega(\vec{\omega}(t))q(t) \tag{5}$$

$$\Omega(\vec{\omega}) = \begin{bmatrix} -\lfloor\vec{\omega}\rfloor & \vec{\omega} \\ \vec{\omega}^T & 0 \end{bmatrix}, \lfloor\vec{\omega}\rfloor = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

where $\vec{\omega} = \dot{\vec{\theta}}$ is the rotational velocity vector.

Based on the gyro model in [15] the angular velocity $\vec{\omega}$ is related to the gyro output $\vec{\omega}_m$ according to the equation:

$$\vec{\omega} = \vec{\omega}_m - \vec{b} - \vec{n}_r \tag{6}$$

with

$$E\{\vec{n}_r(t)\} = 0\ ,\ \ E\{\vec{n}_r(t)\vec{n}_r^T(t')\} = N_r\delta(t-t') \tag{7}$$

where $\vec{b}$ is the drift-rate bias and $\vec{n}_r$ is the drift-rate noise assumed to be a Gaussian white-noise process. The drift-rate bias is not a static quantity but is driven by a second Gaussian white-noise process, the gyro drift-rate ramp noise:

$$\dot{\vec{b}} = \vec{n}_w \tag{8}$$

with

$$E[n_w(t)] = 0 \quad , \quad E[\vec{n}_w(t)\vec{n}_w^T(t')] = N_w\delta(t-t'). \quad (9)$$

These two noise processes are assumed to be uncorrelated ($E[\vec{n}_w(t)\vec{n}_r^T(t')] = 0$).
At this point we present an approximate body-referenced representation of the error state vector. The error state includes the bias error and the quaternion error. The bias error is defined as the difference between the true and estimated bias.

$$\Delta\vec{b} = \vec{b} - \vec{b}_i \quad (10)$$

The quaternion error here is not the arithmetic difference between the true and estimated (as it is for the bias error) but it is expressed as the quaternion which must be composed with the estimated quaternion in order to obtain the true quaternion. That is:

$$\delta q = q \otimes q_i^{-1} \Leftrightarrow q = \delta q \otimes q_i \quad (11)$$

The advantage of this representation is that since the incremental quaternion corresponds very closely to a small rotation, the fourth component will be close to unity and thus the attitude information of interest is contained in the three vector component $\delta\vec{q}$ where

$$\delta q \simeq \begin{bmatrix} \delta\vec{q} \\ 1 \end{bmatrix}. \quad (12)$$

Starting from equations:

$$\frac{d}{dt}q = \frac{1}{2}\Omega(\vec{\dot{\theta}})q \quad (13)$$

and

$$\frac{d}{dt}q_i = \frac{1}{2}\Omega(\vec{\dot{\theta}}_i)q_i \quad (14)$$

where $\vec{\dot{\theta}}$ is the true rate of change of the attitude and $\vec{\dot{\theta}}_i$ is the estimated rate from the measurements provided by the gyros, it can be shown [7] that

$$\frac{d}{dt}\delta\vec{q} = -\lfloor\vec{\omega}_m - \vec{b}_i\rfloor\delta\vec{q} - \frac{1}{2}(\Delta\vec{b} + \vec{n}_r) \quad , \quad \frac{d}{dt}\delta q_4 = 0 \quad (15)$$

Using the infinitesimal angle approximation in Eq. (4), $\delta\vec{q}$ can be written as

$$\delta\vec{q} = \frac{1}{2}\delta\vec{\theta} \quad (16)$$

and thus Eq. (15) can be rewritten as

$$\frac{d}{dt}\delta\vec{\theta} = -\lfloor\vec{\omega}_m - \vec{b}_i\rfloor\delta\vec{\theta} - (\Delta\vec{b} + \vec{n}_r) \quad (17)$$

Differentiating Eq. (10) and making the same assumptions for the true and estimated bias as in Eqs. (6) and (8), the bias error dynamic equation can be expressed as

$$\frac{d}{dt}\Delta\vec{b} = \vec{n}_w. \quad (18)$$

Combining Eqs. (17) and (18) we can describe the error state propagation as

$$\frac{d}{dt}\begin{bmatrix} \delta\vec{\theta} \\ \Delta\vec{b} \end{bmatrix} = \begin{bmatrix} -\lfloor\vec{\omega}_m - \vec{b}_i\rfloor & -I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}\begin{bmatrix} \delta\vec{\theta} \\ \Delta\vec{b} \end{bmatrix} + \begin{bmatrix} -I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix}\begin{bmatrix} \vec{n}_r \\ \vec{n}_w \end{bmatrix} \quad (19)$$

or in a more compact form

$$\frac{d}{dt}\Delta x = F\Delta x + Gn \quad (20)$$

This last equation describes the system model employed in the current Kalman filter implementation [7].

5.3 Measurement Model

This estimator combines the gyroscopes angular rates with the absolute orientation measurements from the vision/laser algorithm and the tilt sensor in order to estimate both the attitude of the vehicle and the gyro biases. Each of these sensors measures the projection of a global vector on the sensor's axes expressed in the vehicle's local coordinates. This information can be used to update the current estimates of the filter. For example, if $\vec{v_o}$ is the unit vector along the direction of the gravitational acceleration then the actual and estimated measurements are:

$$\begin{aligned} z &= \Pi C^T(q)\vec{v}_o + \Pi\vec{n} \\ z_i &= \Pi C^T(q_i)\vec{v}_o \end{aligned}$$

Where Π is the projection matrix, $C(q)$ is the rotational matrix, and $\vec{n}$ is the sensor measurement noise assumed to be a white-zero mean Gaussian process. The linearized measurement equation is derived from the previous relations and it can be shown [7] that is:

$$\Delta z = z - z_i = \Pi C^T(q_i)\lfloor\vec{v}_o\rfloor\delta\vec{\theta} + \Pi\vec{n} \quad (21)$$

Finally, the measurement matrix and the covariance of the measurement noise are given by:

$$H = \Pi C^T(q_i)\lfloor\vec{v}_o\rfloor \ , \ R = \Pi E\{\vec{n}\vec{n}^T\}\Pi^T \quad (22)$$

As shown in [6], this estimator acts as a high pass filter on the gyro signals by filtering out the low frequency noise component (bias) while weighing more their contribution during high frequency motion when the vision/laser algorithm is susceptible to disturbances. If absolute orientation measurements

are available continuously, the filter is capable of continuously tracking the gyro biases. In our case attitude updates are available at a lower rate than the gyro measurements. Therefore the filter updates the bias estimates only intermittently (Fig. 4) based on its effect on the attitude estimates during the previous interval of motion. The resulting attitude estimates are then fed to the control algorithm in order to determine the appropriate steering commands.

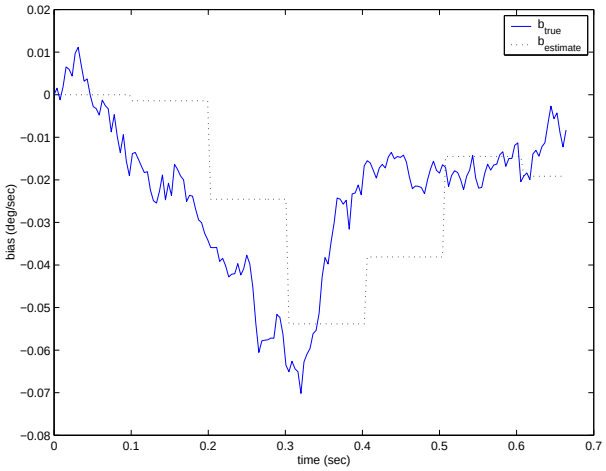

Figure 4: *Bias Estimation (simulation results): The flat parts of the estimate depict the constant bias assumption in the integrator. The sharp step changes occur when absolute attitude measurements become available (at 10 Hz).*

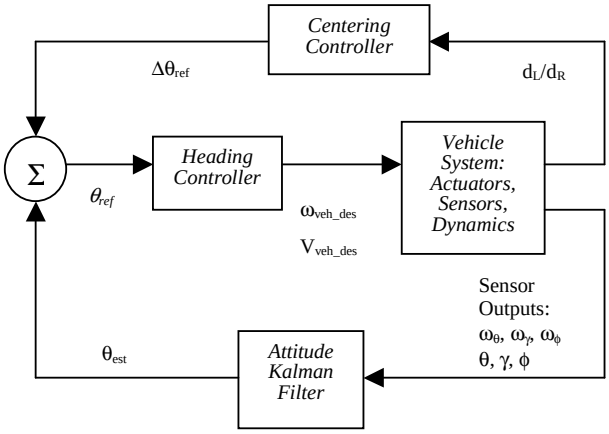

Figure 5: *Control system block diagram.*

6 Control System

6.1 Overview

There are three main goals of this stair climbing control algorithm: (i) maximize the time that the vehicle is heading directly up the stairs, (ii) keep the vehicle at a safe distance away from the edges of the staircase, (iii) accomplish the first two goals while moving as quickly as possible.

The first goal is necessitated partly by the observation that the actual stair climbing speed is a nonlinear function of the vehicle heading. If the heading is perturbed slightly off from zero, track slip increases dramatically.

Under ideal conditions the first two goals are not mutually exclusive and a single tiered control system could be designed that would minimize the heading error while traveling straight up the stairs. As long as the vehicle started at the center of the stairs then it could be expected that it would finish close to the center of the stairs. Unfortunately, there are significant disturbances in this system due to the complex interactions of the tracks and the stairs. The magnitude of these disturbances increases with the stair climbing speed therefore control becomes more difficult when attempting to climb stairs at a higher rate. The fact that the system state is significantly affected by these disturbances creates the need for a two tiered control system. This approach minimizes the heading error while the vehicle is near the centerline of the stairs. Whenever the vehicle approaches the side of the staircase, the centering controller steers the robot towards the centerline and resumes nominal operation as soon as the vehicle is at a safe distance away from the side of the staircase.

Two state variables are used for control purposes during stair climbing (see Fig. 2). The first variable is the heading, θ, of the robot expressed in local coordinates. This variable is computed based on the attitude estimation from the Kalman filter (Section 5). The second variable is a normalized distance from the center of the stairs, $\frac{d_L}{d_R}$. This variable is obtained directly from the vision or the laser algorithm described above.

Two different control schemes are implemented for each of these state variables. For the heading, a model-based control loop is designed. For the centering, a step function with hysteresis is used. Both of these schemes are described below; see Fig. 5 for a system block diagram. In order to reduce the control variable to a single input, a constant linear velocity input is assumed.

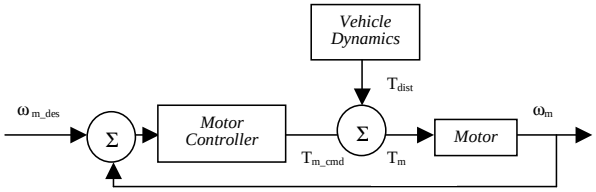

Figure 6: *Motor subsystem block diagram.*

6.2 Heading Control

Model. An approximate dynamic vehicle system model based on first principles is developed in order to design a heading controller for use during stair climbing. This model-based approach is selected because most of the developed systematic tools for the design and analysis of feedback controllers rely on the existence of a dynamic model. This is also important since it allows for the characterization of the inherent instability of the vehicle during open-loop stair climbing. This instability results from the fact that the center of rotation is below the center of gravity of the vehicle (see Fig. 2), creating a classic inverted pendulum control problem. A detailed description of modeling techniques for tracked vehicles is presented in [16] and [17]. Here we have approximated the dynamics of the tracked vehicle climbing stairs as a first order linear system. This approximation does not invalidate the model, it simply limits its range of application. The main advantage of this linearized model is that it increases the number of formal control system design techniques available when designing the heading controller.
The model consists of two motor subsystems (see Fig. 6) and a vehicle dynamics model.
Each motor controller is modeled as a PD controller,

$$k_p + k_d s \tag{23}$$

and each motor is modeled as a first order system,

$$\frac{k_m}{\tau s + 1}. \tag{24}$$

As can be determined from the open-loop step response of the motor, its rise time is approximately $\tau = 20ms$. This is extremely fast compared to the dynamics of the vehicle, thus a kinematic relationship between the rotation velocity error of the motor, $\omega_{m\_err} = \omega_{m\_des} - \omega_m$, and motor torque, T_m, closely approximates the motor controller dynamics. This relationship can be expressed as

$$T_m = k_{mc}\omega_{m\_err} \tag{25}$$

where k_{mc} is defined as the motor controller gain. By applying the *Final Value Theorem* to Eq. (23), it can be shown that $k_{mc} = k_p$. The rotational velocity of the vehicle, ω_{veh}, the track velocity, V_{track}, the torque about the point O, T_O, and the force exerted by each track, F_{track}, are given by:

$$\omega_{veh} = (V_L - V_R)/b \tag{26}$$
$$V_{track} = \omega_m r_s / n_g \tag{27}$$
$$T_O = (b/2)(F_L - F_R) \tag{28}$$
$$F_{track} = T_m(n_g/r_s) \tag{29}$$

where the subscript *track* can mean either L or R depending on the side being referred to, V_L and V_R are left and right track velocities, respectively, b is the distance between the tracks, ω_m is the angular velocity of the corresponding motor, r_s is radius of the sprocket that drives the track, and n_g is the gear ratio between the motor and the sprocket. Substituting in Eqs. (28) and (29) the error equations corresponding to Eqs. (26) and (27) the relationship between ω_{veh} and T_O, is:

$$T_O = k_{veh}\omega_{veh\_err} = k_{veh}(\omega_{veh\_des} - \omega_{veh}) \tag{30}$$

where

$$k_{veh} = (k_{mc}/2)(b\ n_g/r_s)^2. \tag{31}$$

The vehicle dynamics are modeled as

$$\dot{\omega}_{veh} I_z = T_O + mgd_{CG}\sin\alpha\sin\theta - M_r \tag{32}$$

where $\dot{\omega}_{veh}$ is the rotational acceleration, I_z is the moment of inertia, m is the mass, g is gravitational acceleration, d_{CG} is the longitudinal distance from the center of gravity to point O, θ is the heading, M_r is the turning resistance, and α is the inclination of the staircase. Replacing k_{grav} for $mgd_{CG}\sin\alpha$, and invoking the small angle approximation for $\sin\theta$, this model can be written in standard state-space form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{k_{grav}}{I_z} & \frac{-k_{veh}}{I_z} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{M_r}{I_z} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{k_{veh}}{I_z} \end{bmatrix} \omega_{veh\_des} \tag{33}$$

where x_1 is θ and x_2 is ω_{veh}.
Most of these parameters, such as d_{CG}, r_s, and n_g, are easily measured or are known parameters of the vehicle. M_r is calculated as:

$$M_r = \mu W/2L \tag{34}$$

where W is the weight of the vehicle, L is the contact length of the tracks, and μ is the coefficient of lateral resistance estimated from experimental data in [18]. I_z is computed by weighing individual subcomponents of the vehicle and measuring their location relative to the CG.

Controller Design. Once the state-space model shown in Eq. (33) is developed, many techniques can be employed to design the controller. The design technique selected for this application is a pole placement method. This approach has the advantage of being able to explicitly specify the resulting dynamics of the controlled system within the constraints of the actuators [19].
Eq. (33) is compactly written as:

$$\dot{\vec{x}} = A\vec{x} + B + Cu. \tag{35}$$

The result of this design is a control law expressed as:

$$u = -K\vec{x} \tag{36}$$

where, $u = \omega_{veh\_des}$ is the control input, $\vec{x}$ is the state vector, A, B, and C are system matrices, and K is a vector of controller gains.

A few modifications to Eq. (35) are required before applying the pole placement design method. The first of these is to discretize it. The discretization rate is chosen to be equal to the control rate, which is specified to be 30 Hz. Based on simulations of the model this rate is determined to be fast enough to react to the dynamics of the vehicle while slow enough to place reasonable computational demands on the system. The discretized form of Eq. (35) is:

$$\vec{x}(k+1) = \Phi\vec{x}(k) + \beta + \Gamma u(k) \tag{37}$$

where Φ, β, and Γ are the discrete system matrices. The second modification is the augmentation of the state with an integral term. This is to eliminate any steady state error that may occur in the system. The third modification is to add a reference signal to the equation to allow for the centering control to affect the system when rendered necessary. The result of these last two modifications is

$$\begin{aligned}\begin{bmatrix} x_I(k+1) \\ \vec{x}(k+1) \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & \Phi \end{bmatrix} \begin{bmatrix} x_I(k) \\ \vec{x}(k) \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 \\ \beta \end{bmatrix} + \begin{bmatrix} k_I \\ \Gamma \end{bmatrix} u(k) \\ &\quad - \begin{bmatrix} 1 \\ 0 \end{bmatrix} r(k) \end{aligned} \tag{38}$$

where x_I is the integral state variable, k_I is its gain, and $r(k)$ is the reference signal computed by the centering controller.

The design of the heading controller affects several aspects of the system. The first obvious effect is on the dynamics of the resulting system in terms of stability, response speed, and damping. A secondary consideration, contradictory to the first, is the minimization of the energy expended during stair climbing. A balance of these two is achieved by selecting a damped system on the order of $\zeta = 0.7$ without affecting the natural frequency of the system significantly. The effect of the controller design on the response of the system can be iterated upon both in simulation and experimentation to refine the design.

6.3 Centering Control

The input to the centering controller, as mentioned above, is the ratio $\frac{d_L}{d_R}$. The output of this controller is used as the reference signal for the heading controller (see Fig. 2).

Compared to the heading controller, a much simpler approach to centering control is employed. Stair climbing experiments with this vehicle indicate that the optimal heading is $\theta = 0$ and that the effectiveness of tracks on stairs quickly decreases as a nonlinear function of heading. The physics of this can be explained by the fact that nearly all of the force transmitted between the tracks and the stairs during stair climbing occurs when a track cleat slides against a stair edge. With each of the tracks generally spanning three stair edges, this situation takes place almost continuously when the vehicle is facing directly up the stairs, and never when the vehicle is facing slightly angled to the stairs.

This fact dictated our approach to centering control. Since the only significantly negative impact of not going up the center of the stairs is the threat of interactions with the side of the staircase, there is generally a region in the center of the stairs that can be considered homogeneously safe. In this region the centering controller sends a reference signal of zero to the heading controller. When $\frac{d_L}{d_R}$ reaches a minimum or maximum threshold, the centering controller modifies the reference signal of the heading controller to steer the vehicle towards the center of the staircase. To avoid oscillations of the reference signal around these threshold values a hysteresis function is selected, so that the thresholds have different values depending on the direction the center position, $\frac{d_L}{d_R}$, approaches these from.

7 Experimental Results

Experiments were conducted on the bottom floor staircase in building 198 at the Jet Propulsion Laboratory. Each step has a height of 15 cm and a depth of 30 cm providing a slope of 30 deg. The stairwell has a width of 150 cm, which leaves approximately 50 cm of clearance on each side of the robot as it climbs the stairs. All four sides of the stairwell are made of drywall.

The vehicle was initially placed in front of the stairs and was commanded to begin climbing. Then it would deploy its arms to a forward position that allowed it to climb the first step. At this point, the robot would start the control algorithm described in the paper to autonomously navigate the stairs. After traveling a distance on the stairs the arms would deploy to a flat configuration to maximize the traction of the treads on the stairs. The robot would continue climbing until it reached the top of the stairs, which it would detect by the leveling of the tilt sensors, and stop. There are three parameters that quantify the improvement of stair climbing performance: effective velocity, V_{eff}, root mean square of heading, θ_{RMS}, and root mean square of the normalized center po-

sition, $(\ln(\frac{d_L}{d_R}))_{RMS}$. Note that V_{eff} is the length of the staircase divided by the time of ascent, not the commanded speed of the vehicle.
Table 1 shows the comparison of these three parameters averaged over multiple experiments using the previous algorithm [3] and the new algorithm presented in this paper. These results indicate that the new algorithm has doubled the effective stair climbing velocity while improving the root mean square of the heading by 8% and the root mean square of the normalized center position by 31%.
Figs. 8 7 and present the results from example stair

Table 1: *Comparison of previous and new algorithms.*

	Prev. Algorithm	New Algorithm
$V_{eff}(cm/sec)$	24	48
$\theta_{RMS}(^\circ)$	8.5	7.2
$(\ln(\frac{d_L}{d_R}))_{RMS}$	0.93	0.64

climbing trials using the previous algorithm and the new algorithm. These plots are not representative of the entire data set (primarily due to significant variability between different trials), but they do reflect the differences in performance of the algorithms.
In contrast, as shown in Fig. 8, the heading in the previous algorithm oscillates widely. This is due to the constant modification of the reference heading based on the center position. In addition, the large variations in both heading and center position are the result of not considering the dynamics of the vehicle in the controller design.

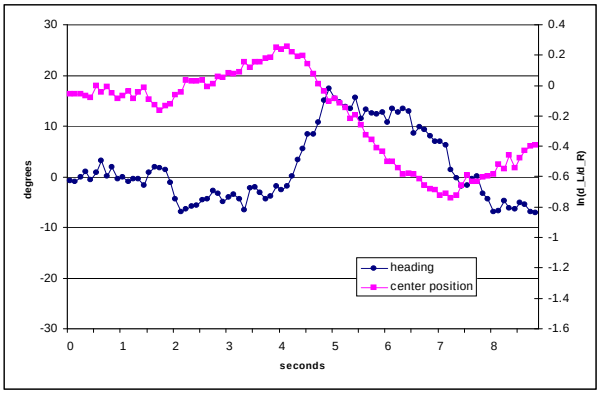

Figure 7: *Heading and center position of stair climbing run using the new algorithm.*

8 Conclusions

This research was mainly motivated by the improvement of the mechanical ability of a tracked urban vehicle to climb stairs. In order to fully realize the stair

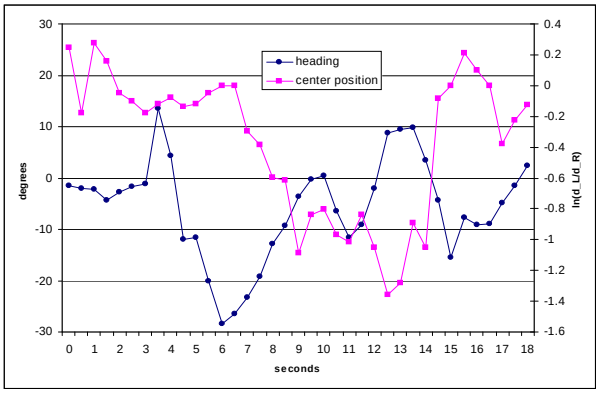

Figure 8: *Heading and center position of stair climbing run using the old algorithm.*

climbing capabilities of the new robot, improvements to the previously implemented stair climbing algorithm [3] were necessary. We have described the design and implementation issues pertinent to a set of new estimation and control algorithms that enhance the stair climbing capabilities of the robot. Experimental results have validated the improvement in speed and effectiveness of the new stair climbing algorithms compared to [3].
As part of our future work we intend to extend these algorithms to control the behavior of the vehicle during two separate phases: right before and after landing. Currently the velocity of the robot at the top of the stairwell is not reduced, resulting in an abrupt landing. Finally, after the vehicle has landed, it has to search for the beginning of the new flight of stairs and align itself towards this. We believe that our new estimation and control algorithms will significantly impact the speed and accuracy of these tasks.

Acknowledgments

The authors would like to thank the rest of the DARPA TMR team at JPL for their contributions to this research: Robert Hogg, Chuck Bergh, Steve Goldberg, and Carlos Villalpando.
This work was supported by the Tactical Mobile Robotics Program of the Defense Advanced Research Projects Agency (DARPA) Advanced Technology Office under contract NAS 7-1407, task order 15089.

References

[1] J.D. Martens and W.S. Newman, "Stabilization of a mobile robot climbing stairs," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, May 8-13, pp. 2501–2507.

[2] S. Steplight, G. Egnal, S.H. Jung, D.W. Walker, C.J. Taylor, and J.P. Ostrowksi, "A mode-based sensor fusion approach to robotic stair-climbing," in *Pro-*

ceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, Takamatsu, Japan, Oct.11-Nov.5, vol. 2, pp. 1113–1118.

[3] Y. Xiong and L. Matthies, "Vision-guided autonomous stair climbing," in *International Conference on Robotics and Automation*, San Francisco, CA, April 2000, pp. 1842–7.

[4] David M. Mount, Nathan S. Netanyahu, Kathleen Romanik, Ruth Silverman, and Angela Y. Wu, "A practical approximation algorithm for the LMS line estimator," in *Symposium on Discrete Algorithms*, 1997, pp. 473–482.

[5] E. J. Lefferts and F. L. Markley, "Dynamics modeling for attitude determination," *AIAA Paper 76-1910*, Aug. 1976.

[6] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey, "Circumventing dynamic modeling: Evaluation of the error-state Kalman filter applied to mobile robot localization," in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Detroit, MI, May 10-15 1999, vol. 2, pp. 1656–1663.

[7] Stergios I. Roumeliotis, *Robust Mobile Robot Localization: From single-robot uncertainties to multi-robot interdependencies*, Ph.D. thesis, Electrical Engineering Department, University of Southern California, Los Angeles, CA, May 2000.

[8] Y. Fuke and E. Krotkov, "Dead reckoning for a lunar rover on uneven terrain," in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, MN, April 22-28 1996, vol. 1, pp. 411–416.

[9] Ph. Bonnifait and G. Garcia, "A multisensor localization algorithm for mobile robots and it's real-time experimental validation," in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, MN, April 22-28 1996, vol. 2, pp. 1395–1400.

[10] M. D. Shuster, "A survey of attitude representations," *Journal of the Astronautical Sciences*, vol. 41, no. 4, pp. 439–517, Oct.-Dec. 1993.

[11] W. F. Phillips and C. E. Hailey, "Review of attitude representations used for aircraft kinematics," *Journal of Aircraft*, vol. 38, no. 4, pp. 718–737, Jul.-Aug. 2001.

[12] J. Stuelpnagel, "On the parameterization of the three-dimensional rotation group," *SIAM Rev.*, vol. 6, no. 4, pp. 422–430, Oct. 1964.

[13] J. J. Craig, *Introduction to Robotics*, chapter 2, pp. 55–56, Addison-Wesley, 2nd edition, 1989.

[14] E. J. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 5, no. 5, pp. 417–429, Sept.-Oct. 1982.

[15] R. L. Farrenkopf, "Analytic steady-state accuracy solutions for two common spacecraft estimators," *Journal of Guidance and Control*, vol. 1, pp. 282–284, July-Aug. 1978.

[16] J.Y. Wong, *Theory of Ground Vehicles*, John Wiley and Sons, 1993.

[17] M.G. Bekker, *Theory of Land Locomotion*, University of Michigan Press, 1956.

[18] I. Hayashi, "Practical analysis of tracked vehicle steering depending on longitudinal track slippage," in *Proc. 5th Int. Conf. of the International Society for Terrain-Vehicle Systems*, Detroit-Houghton, MI, 1975.

[19] G.F. Franklin, J.D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, Addison Wesley Longman, 1998.