Tip-over Prevention: Adaptive Control Development

Leah Kelley

Massachusetts Institute of Technology
Cambridge, MA 02139
Email: lckelley@mit.edu

Kurt Talke, Patrick Longhini and Garret Catron

SSC Pacific, Unmanned Systems Branch
San Diego, CA 92110
Email: {kurt.talke, patrick.longhini, garret.catron}@navy.mil

***Abstract*—Skid-steered, tracked, teleoperated robots are used to perform high-risk critical missions such as bomb disposal under conditions deemed too risky to send a human. Tracked robots work well on structured surfaces such as asphalt roads, but not as well in unstructured environments, including collapsed buildings or on rough terrain covered with sand, brush, rocks and debris. Often the robots carry heavy payloads that raise their centers of mass, increasing their risk of tipping over. Since it is often not feasible to send a human to right a toppled robot, tip-over is equivalent to mission failure. Hence, an autonomous behavior to prevent robot tip-over is desired. In this research, a tipover prevention behavior is developed.**

I. Introduction

Military and civilian crews use skidsteered teleoperated robots to perform dangerous missions, such as bomb disposal. Missions are often conducted in harsh environments, such as inside collapsed buildings or on irregular terrain covered with a variety of media, including sand, brush, mud, rocks and debris. Should a robot stop functioning during its mission, it may be too risky to send a human operator to repair it. In such cases, robot malfunction is equivalent to mission failure.

One common malfunction leading to mission failure is robot tip-over. Tracked robots used for bomb disposal have been shown to work well on flat structured terrain such as asphalt roads, but are at high risk of tip-over when operated on rough or inclined surfaces [1]. The risk of tip-over increases when they carry heavy payloads that raise their centers of mass. Although some robots may have manipulator arms and/or flippers that provide some self-righting capabilities, many others are not so equipped. Further complicating the scenario, a remote operator driving a robot may believe it is on flat ground when looking at its on-board camera feed, when in reality the robot is on an inclined surface near tip-over. Hence, an autonomous tip-over prevention behavior, capable of operating during both autonomous and tele-operated missions is needed to mitigate these concerns.

A. Tip-Over Detection

Several methods have been developed to quantify the stability of mobile robots, including the Zero-Moment Point (ZMP), Force-Angle (FA) and Moment-Height Stability (MHS) stability measures [2]–[7]. Experimental studies have been performed to determine whether the ZMP, FA and MHS metrics can be used to detect real-world tip-over conditions [2]. The studies showed that all metrics can be used to detect tip-over, with the FA and MHS metrics providing more effective stability measurements than the ZMP metric, in some cases indicating tip-over 3 microseconds prior. In follow-up work, the FA metric was selected for use in a tip-over detection algorithm [8].

The FA stability measure approach determines stability based on the angle of the vector sum of all non-supporting forces applied at the robot's center of mass [6]. The force angle is the minimum angle formed between the force-vector sum and a set of vectors pointing from the robot's center of mass to the edges of the convex support polygon formed by the robot's ground contact points, as shown in Figure 1. If the line the force vector sum follows intersects the ground close to the edges of the support polygon but within its area, the angle between the force and edge vectors is small, and the robot is in danger of tip-over. If the force vector intersects an edge of the support polygon, the angle is zero and the robot is about to tip-over. If the force vector intersects the ground outside the support polygon, the minimum angle is less than zero and the robot has already tipped over. In Figure 1, the force vector points outside the support polygon, indicating the robot has tipped over.

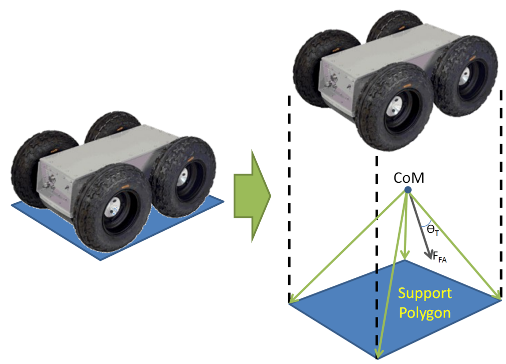

Fig. 1. Tip-over angle definition in Force-Angle stability measure

B. Robot and Vehicle Stabilization

Mobile robot stability control has been well-studied [4], [5], [9]–[20]. Roll stabilization has been achieved by controlling the steering angle, adding controllable anti-roll torsion bars, controlled braking and velocity limiting. Path planning algorithms exclude tip-over states from their feasible trajectories,

IEEE International Conference on Robotics and Automation (ICRA), Seatle, WA, 26-30 May, 2015

though they require predefined waypoints and require a priori knowledge of the terrain. Mobile manipulators, rocker-bogie vehicles and robots with flippers can be stabilized by adjusting their centers of mass. Such methods are not applicable to tele-operated, non-reconfigurable robots [21]. Research on stabilization control of such robots assumes they ascend inclines at the angle of steepest ascent, thus reducing the problem to two dimensions. They also ignore terrain roughness and variation.

C. Tip-over Prevention Approaches for Tracked Vehicles

In this research, two approaches to tip-over prevention for skid-steered vehicles with no ability to change their centers of mass were pursued. A heuristically developed tip-over detection behavior, using the FA measure has been implemented and tested on two tele-operated mobile platforms [8]. This paper presents a model-based feedback controller, where a mathematical model of the dynamics of skid-steered robots is developed. The dynamic model includes robot-terrain specific interactions, such as ground-vehicle contact friction, that change as the terrain changes and cannot be predicted. Hence, adaptive control techniques are used to design the controllers. Because the FA measure indicates which edge of the support polygon a tip-over is likely to occur, pitch and roll can be controlled separately. Here, Model-Reference Adaptive Control (MRAC) is used for pitch and adaptive back-stepping control is used for roll. The two controllers were implemented and tested on an iRobot *Packbot* and a Segway *RMP 440*. Experimental results show that the controllers are able to stabilize the robot under a variety of conditions.

II. Physical Modeling

Understanding a system's dynamic behavior helps determine an appropriate control strategy. The mathematical model of a skid-steered mobile robot is presented below.

A. Skid-Steered Robot Dynamics

First principles are applied to describe the motion of a ground robot in body-centric coordinates. Consider a free-body diagram skid-steered mobile robot on an arbitrary incline, as shown schematically in Fig. 2. Newton's laws expressed in vector form in the body frame are [22]:

$$m\left[\dot{\bar{v}} + \bar{\omega} \times \bar{v} + \dot{\bar{\omega}} \times \bar{r}_g + \bar{\omega} \times (\bar{\omega} \times \bar{r}_g)\right] = \bar{F} \tag{1}$$

$$J\dot{\bar{\omega}} + \bar{\omega} \times (J\bar{\omega}) + m\bar{r}_g \times (\dot{\bar{v}} + \bar{\omega} \times \bar{v}) = \bar{\tau} \tag{2}$$

where m is the mass of the body, $\dot{\bar{v}}$ is a vector of linear accelerations in the body frame, $\dot{\bar{\omega}}$ is a vector of angular accelerations of the body frame, $\bar{v}$ is a vector of linear velocities in the body frame, $\bar{\omega}$ is a vector of angular velocities in the body frame, $\bar{r}_g$ is the location of the center of mass with respect to the origin of the body frame and can be expanded as $[r_x r_y r_z]^T$, J is a 3x3 matrix of inertias, $\bar{F}$ are the external forces acting on the body, and $\bar{\tau}$ are the external moments acting on the body.

1) Friction Modeling: Experimental data was used to estimate coefficients for the Coulomb friction model which consistently resulted in positive friction coefficients. A signum function is used to ensure that friction always acts in opposition to the robot motion. The frictional forces act at the robot/ground surface, and are:

$$F_f = \begin{bmatrix} -\operatorname{sgn}(v_x) m g \mu_1 \cos(\theta)\cos(\phi) \\ -\operatorname{sgn}(v_y) m g \mu_2 \cos(\theta)\cos(\phi) \\ 0 \end{bmatrix} \tag{3}$$

where μ_1 and μ_2 are the Coulomb friction coefficients.

Though the frictional forces act at the robot/ground interface, they cannot be assumed to be equally divided along the track lengths. For simplicity, the frictional force vector is assumed to act at a point on the robot/ground surface directly below the robot center of mass.

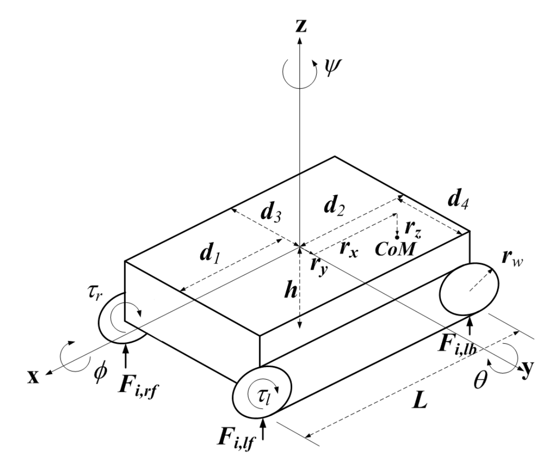

Fig. 2. Free body diagram of a skid-steered robot

2) Impact Forces: When the robot traverses rough terrain, it experiences impacts that cause vertical displacement and moments about the x and y axes as shown in Figure 2.

For simplicity, it is assumed that the net impact force, $\bar{F}_I$, acts below the center of mass on the robot/ground interface, and the moments $\bar{M}_I$ caused by impact are defined as:

$$\bar{F}_I = \begin{bmatrix} 0 \\ 0 \\ f_{I,z} \end{bmatrix}, \quad \bar{M}_I = \bar{r}_g \times \bar{F}_I = \begin{bmatrix} r_y f_{I,z} \\ -r_x f_{I,z} \\ 0 \end{bmatrix} \tag{4}$$

3) Actuator Forces: The tracked, skid-steered robot under consideration has two driving motors, one at each front sprocket, as shown in Figure 2. The following system inputs are defined from the actuator forces:

$$u_1 = \tau_r + \tau_l, \quad u_2 = -d_3\tau_r - d_4\tau_l \tag{5}$$

where τ_r is the torque applied by the right driving motor on the right sprocket, τ_l is the torque applied by the left driving motor on the left sprocket, and d_3 is the lateral distance from the center of mass to the right sprocket, and d_4 is the lateral distance from the center of mass to the left sprocket. Note that d_3 is negative since it is to the right of the center of mass. Also shown is r_w, the radius of the drive sprocket.

4) Constraints and Simplified Equations of Motion: The robot is assumed constrained to the ground surface, so that

$$v_z = 0, \qquad \dot{v}_z = 0 \tag{6}$$

Therefore, the sum of the inertial, centripetal, Coriolis, and gravitational terms can be used to estimate the impact forces:

$$\begin{aligned} f_{I,z} = \dot{\omega}_x r_y - \dot{\omega}_y r_x - \omega_y v_x + \omega_x v_y - \omega_x^2 r_z - \omega_y^2 r_z \\ + \omega_x \omega_z r_x + \omega_y \omega_z r_y + g\cos(\theta)\cos(\phi) \end{aligned}$$

The equations of motion under the assumed constraints are thus

$$\begin{aligned}
\dot{v}_x &= -\dot{\omega}_y r_z + \dot{\omega}_z r_y + \omega_z v_y - \omega_y \omega_x r_y - \omega_z \omega_x r_z \\
&\quad + \omega_y^2 r_x + \omega_z^2 r_x + g\sin(\theta) \\
&\quad - \operatorname{sgn}(v_x) g \mu_1 \cos(\theta)\cos(\phi) + \tfrac{1}{m r_w} u_1 \\[1ex]
\dot{v}_y &= \dot{\omega}_x r_z - \dot{\omega}_z r_x - \omega_z v_x + \omega_x^2 r_y - \omega_x \omega_y r_x \\
&\quad - \omega_z \omega_y r_z + \omega_z^2 r_y - g\cos(\theta)\sin(\phi) \\
&\quad - \operatorname{sgn}(v_y) g \mu_2 \cos(\theta)\cos(\phi) \\[1ex]
\dot{\omega}_x &= \tfrac{1}{J_x}[m\dot{v}_y r_z + m\omega_y v_x r_y + m\omega_z v_x r_z - m\omega_x v_y r_y \\
&\quad - J_z \omega_z \omega_y + J_y \omega_y \omega_z - mgr_y \cos(\theta)\cos(\phi) \\
&\quad + mgr_z \cos(\theta)\sin(\phi) - \operatorname{sgn}(v_y) mgh\mu_2 \cos(\theta)\cos(\phi) \\
&\quad + r_y f_{I,z}] \\[1ex]
\dot{\omega}_y &= \tfrac{1}{J_y}[-m\dot{v}_x r_z - m\omega_y v_x r_x + m\omega_x v_y r_x + m\omega_z v_y r_z r \\
&\quad + J_z \omega_z \omega_x - J_x \omega_x \omega_z + mgr_z \sin(\theta) \\
&\quad + mgr_x \cos(\theta)\cos(\phi) + sgn(v_x) mgh\mu_1 \cos(\theta)\cos(\phi) \\
&\quad - r_x f_{I,z} + \tfrac{r_w - (h + r_z)}{r_w} u_1] \\[1ex]
\dot{\omega}_z &= \tfrac{1}{J_z}[m\dot{v}_x r_y - m\dot{v}_y r_x - m\omega_z v_x r_x - m\omega_z v_y r_y \\
&\quad - J_y \omega_y \omega_x + J_x \omega_x \omega_y - mgr_x \cos(\theta)\sin(\phi) \\
&\quad - mgr_y \sin(\theta) - \operatorname{sgn}(v_y) r_x mg\mu_2 \cos(\theta)\cos(\phi) \\
&\quad + \operatorname{sgn}(v_x) r_y mg\mu_1 \cos(\theta)\cos(\phi) + \tfrac{1}{2r_w} u_2]
\end{aligned} \tag{7}$$

III. Adaptive Control Design

As the robot travels over different terrains, the friction interaction between the tracks and the ground changes, so the robot stabilization controller needs to adapt to these changes. The dynamic model is parameterized such that the unknown, time-varying friction coefficients are linear with respect to measurable states, hence, adaptive control can be used.

The tip-over detection algorithm provides the likelihood and direction the robot is most likely to tip-over as shown in Section I-A. Thus, the angle to stabilize can be chosen based on that direction, allowing pitch and roll to be controlled separately rather than attempting to stabilize both at once.

A. Pitch Controller Design

A state-accessible Model-Referenced Adaptive Controller [23] is developed to control the pitch velocity, shown in Fig. 3. The pitch angle ϕ is directly affected by the control input u_1, or the sum of the right and left torques, as is seen in Eqn. (7). The forward accelerations and angular velocities can be measured from the IMU data, and the forward velocity can be measured from the wheel encoder data. Assuming a no-slip condition, the lateral velocity can be estimated from the yaw velocity using:

$$v_y = r_x \dot{\psi}_{imu} \tag{8}$$

where $\dot{\psi}_{imu}$ is the measured yaw velocity from the IMU sensors.

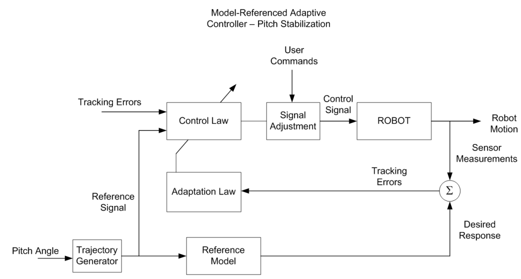

Fig. 3. Pitch controller block diagram

The angular acceleration for pitch can be written as:

$$\dot{\omega}_y = f_{1,p} + \mu_1 f_{2,p} + b u_1 \tag{9}$$

where

$$\begin{aligned} f_{1,p} = \tfrac{1}{J_y}[-mr_z \dot{v}_x - m\omega_y r_x v_x + m\omega_x r_x v_y + m\omega_z r_z v_y \\ + (J_z - J_x)\omega_z \omega_x + mgr_z \sin(\theta) \\ + mgr_x \cos(\theta)\cos(\phi) - f_{I,z} r_x] \end{aligned} \tag{10}$$

$$f_{2,p} = \frac{\operatorname{sgn}(v_x) mgh \cos(\theta)\cos(\phi)}{J_y} \tag{11}$$

$$b = \frac{r_w - (h + r_z)}{J_y r_w} \tag{12}$$

The Coulomb friction coefficient is μ_1, assumed to be quasi-constant but unknown. The controller is designed so that the pitch velocity follows a stable first-order reference model, as in [23]:

$$\dot{\omega}_{y,m} = -a_m \omega_{y,m} + br \tag{13}$$

where a_m is positive and r is the reference signal. The parameter estimate $\hat{a_f}$ and parameter estimate error $\tilde{a}_f$ are defined:

$$\hat{a}_f = \frac{\hat{\mu}_1}{b}, \qquad \tilde{a}_f = \hat{a}_f - \frac{\mu_1}{b} \tag{14}$$

where $\hat{\mu_1}$ is the estimate of the Coulomb friction coefficient.

The control and adaptive laws are defined as:

$$u_1 = -\frac{a_m}{b}\omega_y - \frac{1}{b} f_{1,p} - \hat{a}_f f_{2,p} + r \tag{15}$$

$$\dot{\hat{a}}_f = \dot{\tilde{a}}_f = b e f_{2,p} \tag{16}$$

and the error e as:

$$e = \omega_y - \omega_{y,m} \tag{17}$$

Since μ_1 is assumed constant $\dot{\tilde{a}}_f = \dot{\hat{a}}_f$. Under the above control and adaptation laws, the closed loop and error dynamics become:

$$\dot{\omega}_{y,CL} = -a_m \omega_{y,CL} - (b\hat{a}_f - \mu_1) f_{2,p} + br \tag{18}$$

$$\dot{e} = -a_m e - b\tilde{a}_f f_{2,p} \tag{19}$$

Consider a Lyapunov candidate function and its derivative:

$$V = \frac{e^2}{2} + \frac{\tilde{a}_f^2}{2} \tag{20}$$

$$\dot{V} = e\dot{e} + \tilde{a}_f \dot{\tilde{a}}_f \tag{21}$$

Substituting Eqns. (16) and (19) into Eqn. (21) yields:

$$\dot{V} = e\left[-a_m e - b\tilde{a}_f f_{2,p}\right] + \tilde{a}_f\left[be f_{2,p}\right] = -a_m e^2 \leq 0 \tag{22}$$

which is negative semi-definite, since a_m is positive. The candidate Lyapunov function Eqn. (20) is positive definite. For stability, all signals must be bounded, and the error between the actual and reference states must tend to zero. Since Eqn. (22) is negative semi-definite and Eqn. (20) is positive definite, then

$$V\left(e\left(t\right), \tilde{a}_f\left(t\right)\right) < V\left(e\left(0\right), \tilde{a}_f\left(0\right)\right) < \infty \tag{23}$$

Thus V is bounded and e and $\tilde{a}_f$ are bounded. The reference model used for the pitch controller is stable, since the reference signal r is bounded by design, as discussed later in Section III-C, e is bounded, and $-a_m$ is negative. Therefore, the reference velocity $\omega_{y,m}$ is bounded, and, in turn, the actual pitch velocity ω_y is bounded. While the controller is active, it is assumed that the parameter μ_1 is constant, so $\hat{a}_f$ is bounded since $\tilde{a}_f$ is bounded. Finally, $\dot{e}$ is bounded since in Eqn. (19), e and $\tilde{a}_f$ are bounded, b is a constant and the limits on $f_{2,p}$ are $|f| \leq \frac{mgh}{J_y}$. Using Barbalet's Lemma as presented by Corollary 2.9 in [23] it can be shown that the error e tends to zero.

B. Roll Controller Design

Unlike pitch, roll is not directly affected by actuating the drive motors, as seen in Eqn. (7). However, roll is affected by both the forward and yaw velocities, which are directly affected by the actuators u_1 and u_2, respectively. Therefore, an adaptive back-stepping approach in which the yaw velocity $\dot{\psi}$ is used as a control input to the roll controller is used, as shown in Fig. 4. The yaw velocity is controlled by the difference between the two torques, u_2. The adaptive backstepping approach uses stabilization and tuning functions to avoid over-parameterization, as described in [24].

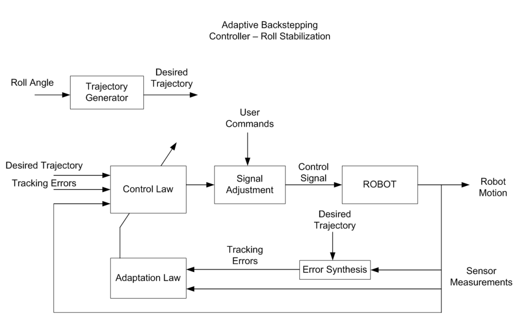

Fig. 4. Roll controller adaptive back-stepping block diagram

The roll dynamics from Eqn. (7) can be written in the form

$$\dot{\omega}_x = f_{1,r} + f_{2,r}\omega_x + B_r\omega_z + \mu_2 g_1 \tag{24}$$

where

$$\begin{aligned} f_{1,r} = \tfrac{1}{J_x}[&mr_z\dot{v}_y + mr_y\omega_y v_x + mgr_z\cos(\theta)\sin(\phi) \\ -\ &mgr_y\cos(\theta)\cos(\phi) + f_{I,z}r_y] \end{aligned} \tag{25}$$

$$f_{2,r} = \frac{-mr_r v_y}{J_x}, \quad g_1 = \frac{-mgh\,\mathrm{sgn}(v_y)\cos(\theta)\cos(\phi)}{J_x} \tag{26}$$

$$B_r = \frac{mr_z v_x - J_z\omega_y + J_y\omega_y}{J_x} \tag{27}$$

From Eqns. (24) and (27) the parameter B_r is known explicitly, but can at times take on a zero value. When this occurs, the yaw velocity will have no effect on the roll velocity. Thus, the roll controller will only be activated under the condition

$$|B_r| \geq \delta > 0. \tag{28}$$

In the instances where roll stability is needed but the above condition is not satisfied, the robot can be commanded to use one of the heuristic behaviors, such as turn-to, or can be commanded to turn so pitch must be stabilized instead [8].

From Eqn. (7), the yaw dynamics can be written in the form

$$\dot{\omega}_z = k_r u_2 + f_{3,r} + f_{4,r}\omega_z + \mu_2 g_2 + \mu_1 g_3 \tag{29}$$

where

$$\begin{aligned} f_{3,r} = \tfrac{1}{J_z}[&m\dot{v}_x r_y - m\dot{v}_y r_x - J_y\omega_y\omega_x + J_x\omega_x\omega_y \\ &- mgr_x\cos(\theta)\sin(\phi) - mgr_y\sin(\theta)] \end{aligned} \tag{30}$$

$$f_{4,r} = \frac{-mv_x r_x - mv_y r_y}{J_z}, \quad k_r = \frac{1}{2r_w J_z} \tag{31}$$

$$g_2 = \frac{-r_x mg\,\mathrm{sgn}(v_y)\cos(\theta)\cos(\phi)}{J_z} \tag{32}$$

$$g_3 = \frac{r_y mg\,\mathrm{sgn}(v_x)\cos(\theta)\cos(\phi)}{J_z} \tag{33}$$

To design the adaptive backstepping controller, the following set of error coordinates are defined:

$$z_1 = \omega_x - \omega_{x,d}, \qquad z_2 = \omega_z - \alpha - \frac{\dot{\omega}_{x,d}}{B_r} \tag{34}$$

where α is a stabilizing function. Assuming that B_r is constant during the updates, the derivatives of the error coordinates are

$$\dot{z}_1 = \dot{\omega}_x - \dot{\omega}_{x,d}, \qquad \dot{z}_2 = \dot{\omega}_z - \dot{\alpha} - \frac{\ddot{\omega}_{x,d}}{B_r} \tag{35}$$

The control and adaptive laws are derived using Lyapunov-stability analysis. Presentation of the complete derivation is beyond the scope of this paper. The control laws and stability are presented here. The roll control law is defined as

$$\begin{aligned} u_2 = \tfrac{1}{k_r}[&-\hat{\mu}_2 g_2 - \hat{\mu}_1 g_3 - c_2 z_2 - B_r z_1 - f_{3,r} \\ &- f_{4,r}z_2 - f_{4,r}\alpha - \tfrac{f_{4,r}\dot{\omega}_{x,d}}{B_r} + \tfrac{c_1^2 z_1}{B_r} - c_1 z_2 \\ &- \tfrac{\dot{\omega}_x f_{2,r}}{B_r} - \tfrac{\dot{\hat{\mu}}_2 g_1}{B_r} + \tfrac{\ddot{\omega}_{x,d}}{B_r}] \end{aligned}$$

where c_1 and c_2 are positive constants. The stabilizing function α is defined as

$$\alpha = \frac{-f_{1,r}}{B_r} + \frac{-f_{2,r}}{B_r}\omega_x + \frac{-\hat{\mu_2}}{B_r}g_1 + \frac{-c_1\dot{z}_1}{B_r} \tag{36}$$

Its derivative is

$$\dot{\alpha} = \frac{-c_1\dot{\omega}_x}{B_r} + \frac{c_1\dot{\omega}_{x,d}}{B_r} + \frac{-\dot{\omega}_x f_{2,r}}{B_r} + \frac{-\dot{\hat{\mu}}_2 g_1}{B_r} \tag{37}$$

The adaption laws for $\dot{\hat{\mu}}_1$ and $\dot{\hat{\mu}}_2$ are

$$\dot{\hat{\mu}}_1 = g_3 z_2, \qquad \dot{\hat{\mu}}_2 = \tau_1 + z_2\left[g_2 + \frac{g_1 c_1}{B_r}\right] \tag{38}$$

where the tuning function $\tau_1 = z_1 g_1$.

It is also assumed that the friction coefficients μ_1 and μ_2 are constant while the controller is active. Therefore,

$$\dot{\tilde{\mu}}_1 = \dot{\hat{\mu}}_1, \quad \text{and} \quad \dot{\tilde{\mu}}_2 = \dot{\hat{\mu}}_2. \tag{39}$$

Consider a Lyapunov candidate function

$$V = \frac{z_1^2}{2} + \frac{z_2^2}{2} + \frac{\tilde{\mu}_2^2}{2} + \frac{\tilde{\mu}_1^2}{2} \tag{40}$$

and its derivative

$$\dot{V} = z_1\dot{z}_1 + z_2\dot{z}_2 + \tilde{\mu}_2\dot{\tilde{\mu}}_2 + \tilde{\mu}_1\dot{\tilde{\mu}}_1 \tag{41}$$

The closed loop error dynamics for z_1 and z_2 are

$$\dot{z}_1 = -c_1 z_1 + B_r z_2 - \tilde{\mu}_2 g_1 \tag{42}$$

$$\dot{z}_2 = -c_2 z_2 - B_r z_1 - \left[g_2 + \frac{c_1 g_1}{B_r}\right]\tilde{\mu}_2 - g_3\tilde{\mu}_1 \tag{43}$$

Substituting the closed loop error dynamics and the adaption laws into Eqn. (41) yields

$$\dot{V} = -c_1 z_1^2 - c_2 z_2^2 \tag{44}$$

Since c_1 and c_2 are positive constants, Eqn. (44) is negative semi-defininte and Eqn. (40) is positive definite. Thus, V, z_1, z_2, $\tilde{\mu}_1$ and $\tilde{\mu}_2$ are bounded. It can be shown, that the controller is stable under the constraints assumed in this paper.

C. Reference Trajectory Generation

For stabilization, the robot should be commanded to rotate opposite its tipping direction. Accurate parameter estimation is not needed for pitch control, so a step in pitch angle opposite the tipping direction is commanded when the pitch controller is activated by $r = -sgn(\theta_{imu,0})k_2$, where $\theta_{imu,0}$ is the pitch angle recorded at the time the pitch controller is activated and k_2 is a user-defined constant. This signal need not be differentiable.

The adaptive backstepping roll controller requires a twice differentiable signal, since $\omega_{x,d}$, $\dot{\omega}_{x,d}$ and $\ddot{\omega}_{x,d}$ are all required. A sinusoid is a good reference signal candidate since it is infinitely differentiable.

IV. Experimental Validation

The adaptive roll and pitch controllers, or advanced control, were implemented on both an iRobot *Packbot* and a Segway *RMP 440*, each equipped with a custom payload. The payload contains a processor that runs the tip-over detection and prevention algorithms, GPS, Ethernet for wireless communication, and a MicroStrain 3Dm-GX3 inertial measurement unit (IMU). The data from the IMU sensor include pitch, roll, yaw, pitch speed, roll speed, yaw speed and accelerations in (x, y, z) in the robot body frame. The controllers were tested by driving the robots on rough terrain such as sand, coarse rock, dry brush and dirt. Both robots were driven uphill forwards and backwards, downhill, and along inclined surfaces. Experiments were also conducted using the *Packbot* with an additional payload consisting of a raised platform with a 20-pound weight to elevate the center of mass and determine how well the advanced control works with a less-stable platform.

The advanced control exhibited positive results, providing the user with a fast, automatic reactive behavior on the order of tenths of a second, much faster than human reaction time for correcting an unstable robot. Often, the controller activates before the user can realize that the robot was in danger of tip-over. When the controller is active for longer periods of time, the user does not lose control of the robot to the controller behavior. Rather, the controller shares control with the operator and redirects the robot to a safer path, based on its current state. This is an improvement over the heuristically developed behaviors presented in [8], since here the user still maintains some control.

A. Pitch Control Experiments

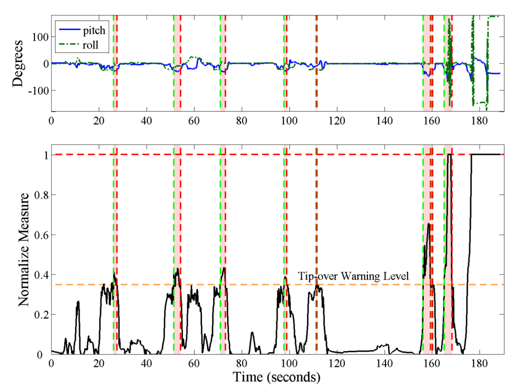

Fig. 5. *RMP440* Advanced Control: (top) displays vehicle pitch and roll; (bottom) normalized tip-over measure. Threshold for tip-over warning set to 0.38.

1) Segway RMP 440: Figure 5 demonstrates the advanced control active on the Segway *RMP440*. The normalized measures displays the tip-over parameter normalized between 0 and 1, where 0 is stable and 1 is unstable or in the tipped over state. From 0 to 155 seconds the advanced control made corrections to avoid tip-over. At 155 seconds, the first attempt up the hill, the *RMP440* did not tip-over, with the pitch controller activating. However, the next two attempts ended in a tip-over, likely a result of the activation threshold being set too high.

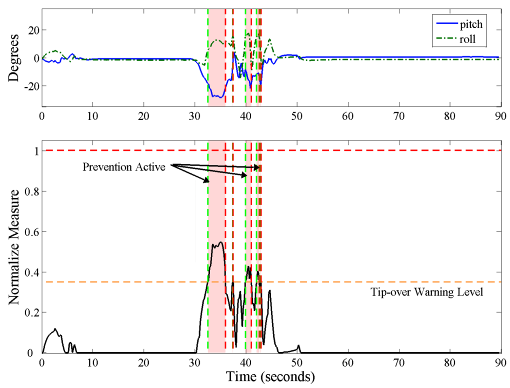

Fig. 6. *Packbot* Advance Control: (top) displays vehicle pitch and roll; (bottom) normalized tip-over measure. Threshold for tip-over warning set to 0.38.

2) iRobot Packbot: Raised Center of Mass: In order to better evaluate the advanced control, a weighted mass, or "dummy" payload was attached to the *Packbot* to raise the center of mass. As shown when the advanced control is active in Figure 6, if the normalized measure crosses the threshold tip-over threshold, the controller will make the appropriate correction to mitigate the instability as seen at 33, 41, and 43 seconds.

B. Roll Control Experiments

Similar to the pitch controller results in Section IV-A. The roll control experiments demonstrated that Segway *RMP440* and iRobot *Packbot* made the appropriate corrections to avoid tip-over. Figure 7 shows sequences of clips taken from a typical situation displaying first the roll control activating, correcting the roll instability, followed by the pitch controller activating, bringing the vehicle to a safe, stable position.

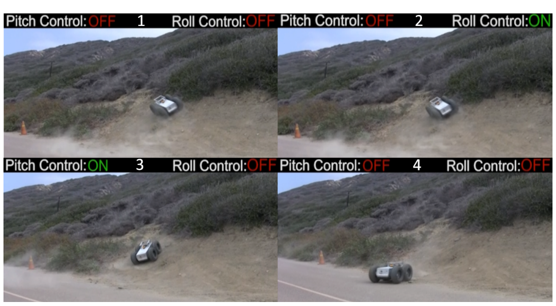

Fig. 7. Advance Controller: Pitch and Roll activatation sequence

C. Advance Control Off

For comparison purposes, results without any controller are shown in Figure 8. The shaded regions correspond to when the normalized tip-over measure crosses the threshold and the behavior for preventing tip-over would have activated. As seen between 65 and 86 seconds, the *Packbot* does not tip-over during the first couple times above the threshold. However, at 90 seconds, a tip-over occurs. Comparing to when the advanced controller was active as was shown in Figure 6, the advanced controller prevents tip-over when the tip-over measure rises above the threshold value.

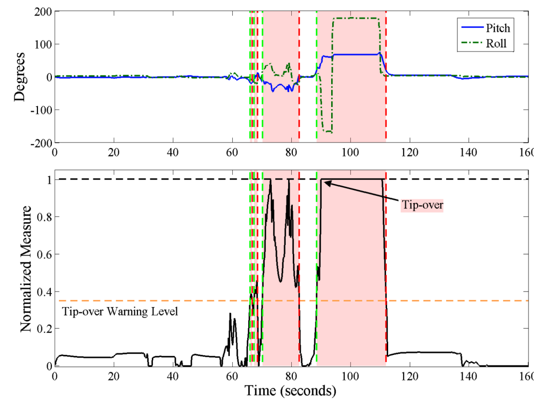

Fig. 8. Adaptive control turned off for the *Packbot*. The shaded regions show where the behavior would have been active to prevent tip-over.

V. Conclusion

Tip-over presents a significant risk for autonomous and tele-operated unmanned ground vehicles. Using a first-principles-based physics model, an advanced controller enabling both a state accessible Model Referenced Adaptive Control approach for pitch control and an adaptive back-stepping approach for roll control was designed, mathematically proven, and experimentally validated. Testing results on the iRobot *Packbot* and the Segway *RMP440* showed promise for the controller to successfully deter tip-over scenarios. The adaptive model-based controller can mitigate many of the issues seen with system-ID-based controllers, and provide a viable means to successfully avert tip-over and mission failure.

Acknowledgment

The authors would like to acknowledge the Joint Ground Robotics Enterprise (JGRE) for providing multi-year funding for this effort, Susie Alderson for program management, Aaron Burmeister, Saam Ostovari for their technical guidance, and Bart Everett for his technical direction and advice.

References

[1] J. Walker, "Unmanned ground combat vehicle contractors selected." Defense Advanced Research Projects Agency News Release, Feb 2001.
[2] P. Roan, et al., *Proc. IEEE ICRA*, 2010, pp. 4431–4436.
[3] B. Borovac and M. Vukobratović, *Int. J. Hum. Robot*, vol.1, no.1, 2004.
[4] Q. Huang, et al., *Proc. IEEE IROS.*, vol. 2, 1994, pp. 839–846.
[5] J. Kim et al., *Proc. IEEE ICRA.*, vol. 2, 2002, pp. 1967–1972.
[6] E. Papadopoulos and D. Rey, *Proc. IEEE ICRA*, vol.4, 1996, pp. 3111.
[7] S. Ali, A. Moosavian, and K. Alipour, *Robotics, Automation and Mechatronics, 2006 IEEE Conf. on*, 2006, pp. 1–6.
[8] K. Talke, L. Kelley, P. Longhini, and G. Catron, *Proc. SPIE 9084, Unmanned Systems Technology XVI*, June 2014, pp. 90 840L–11.
[9] S. C. Peters and K. Iagnemma, *Proc. IEEE ICRA.*, 2006, pp. 3711–3716.
[10] J. Ackermann and D. Odenthal, *Proc. Int. Conf. Adv. in Vehicle Control and Safety*, 1998.

[11] M. Krid and F. Benamar, *Proc. IEEE/RSJ IROS*, 2011, pp. 274–279.
[12] B. Johansson and M. Gafvert, *IEEE CDC.*, vol. 5, 2004, pp. 5461–5466.
[13] M. Richier, R. Lenain, B. Thuilot, and C. Debain, *2nd Int Conf. Comms, Computing and Control Apps (CCCA)*, Dec 2012, pp. 1–6.
[14] K. Iagnemma and S. Dubowsky, *Mobile Robots in Rough Terrain: Estimation, Motion Planning, and Control with Application to Planetary Rovers*, Springer Berlin Heidelberg, 2004, vol. 12.
[15] T. Howard and A. Kelly, *Int.l J. of Robot Research*,v.26(2),2007, pp.141.
[16] M. Spenko, S. Dubowsky, and K. Iagnemma, *Proc. 8th Int. IFAC SYROCO*, vol. 8 (1), 2006, pp. 604–609.
[17] N. Hootsmans and S. Dubowsky, *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 3, 1991, pp. 2336–2341.
[18] D. Rey and E. Papadopoulos, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Sys.*, vol. 3, 1997, pp. 1273–1278.
[19] Y. Li and Y. Liu, *Int. J. Vehicle Autonomous Sys*, v.4(1), 2006, pp. 24.
[20] Y. Liu and G. Liu, *IEEE/ASME Trans on Mech.*, v.15(4), 2010, pp. 623.
[21] C. Terupally, J. Zhu, and R. Williams, *American Control Conference (ACC)*, 2007, pp. 2861–2866.
[22] D. Boskovic and M. Krstić, *Proc. IEEE Int. Conf. Control Apps.*, vol. 2. 1999, pp. 1768–1773.
[23] K. Narendra and A. Annaswamy, *Stable Adaptive Systems*, 2nd ed. Mineola, NY: Courier Dover Publications, 2005.
[24] M. Krstić, I. Kanellakopoulos, and P. Kokotović, *Systems & Control Letters*, vol. 19, no. 3, pp. 177–185, 1992.