Contact Sensing and Mobility in Rough and Cluttered Environments

K.S. Kim, A.S. Kwok, and L. Sentis

***Abstract*— In this study, we explore mobility in a cluttered, uneven and dynamic environment where collisions are inevitable. In particular, we study a mobile robot's ability to precisely track a planned path on an inclined surface, then respond to unknown collisions by using force compliance and estimation of the obstacle's surface normal. In addition, once the contact disturbance disappears, the robot merges automatically back to its original planned path. This study aims at highlighting the need to safely deal with unexpected collisions with external objects and use them to move alongside their contour, opening opportunities perhaps to use contacts as supporting structures. At the same time, it aims to endow capabilities for precise maneuvering in the rough terrains. Our newly designed compliant omnidirectional robot is able to quickly respond to contacts through the use of current control and absolute position / orientation sensing. A whole-body controller developed by the Human Centered Robotics Lab is used to compensate for gravity and create new multicontact constraints in response to the collisions. To pursue the study, we built the mobile base and a multicontact experimental setup. The results show that our model-based controller successfully accomplishes the tasks with good precision while relying on data-driven contact estimation.**

I. Introduction

One of the main purposes of mobile robots is to manipulate objects with high accuracy. This goal requires the robot's mobile base to execute subtle maneuvers, i.e. precise and slow movements that can, for instance, enable the assembly of objects. When operating outdoors, robots must implement these subtle movements in the presence of nearby obstacles that can cause accidental collisions, and at the same time while dealing with rough terrains. With these issues in mind, this paper aims to provide new methods and their experimental validation for the support of future mobile robotic systems in human-centered or outdoor scenarios.

In particular, our everyday world is filled with clutter, obstacles, moving people, and different topographies. There is a need to deal with dynamic obstacles, perhaps by safely handling collisions or using them as supports instead of avoiding them. On the other hand, mobile robots in general avoid contacts. However, our hypothesis is that by occasionally establishing contacts, robots will become more effective, for instance, in moving among crowded environments, recovering from falls, or anchoring their bodies against objects to engage in subtle manipulations.

We present here various methods and experiments to endow the following capabilities: (1) to move skillfully in uneven terrains without falling, (2) to respond reactively to unknown dynamic collisions using force compliance, (3) to estimate at runtime the geometry of the collision surfaces, and (4) to safely maneuver along the contour of obstacles until they have been cleared. To demonstrate the effectiveness of our methods, we have built a compliant holonomic base equipped with omni wheels and passive rollers on the robot's side panels. These rollers are distributed such that the base can smoothly slide along various types of surfaces during contact. We also have built an experimental setup containing an inclined terrain and a movable wall to test the adaptive interactions.

To move on uneven terrains without sliding down, we model gravity effects on the wheels as a function of both the terrain's slope and the base's heading. We obtain robot heading and position using a motion capture sensing system, incorporate the sensor data on the gravity model, and add the resulting gravity disturbances to a floating body dynamic model of the robot. We then compensate for the estimated gravity disturbances using a compliant control strategy.

To respond reactively to collisions, we first limit the forces applied to the obstacles by using compliant control. We achieve this capability by estimating an inverse dynamic model of the robot and using it to control the effective mechanical impedance (i.e. the force generated when constraining the motion).

To estimate the geometry of the collision surfaces, we first detect the contact when the measured distance between the desired path and the current position is larger than the motion uncertainty bounds. Then we estimate the surface normal of the object using a least squares fitting function based on the movement data points. Finally, we project the compliant control model into a mathematical constraint defined as a function of the estimated surface normal. This projection, effectively removes the direction of the wall from the motion controller, leading to a motion that is tangent to the wall while serving as a support.

Overall, although the concept of mobile robots and autonomous ground vehicles bumping into obstacles has been out for some time, this study is unique on its focus on sensing and adapting to dynamic collisions in the rough terrains.

II. Related Work

The concept of moving around by bumping into objects is well known, for instance in bumper cars, but most recently in home robotics such as iRobot's Roomba [3], which changes direction upon colliding with walls and obstacles. Bumping into objects and humans is a major area of concern in the area of free-roaming vehicles and trucks, such as AGV-Forklift systems. It has normally been dealt by using

All authors are with the Department of Mechanical Engineering at the University of Texas at Austin, USA. Contact Information: K.S. Kim: kskim@utexas.edu; A.S. Kwok: alan.kwok@utexas.edu; L. Sentis: lsentis@austin.utexas.edu

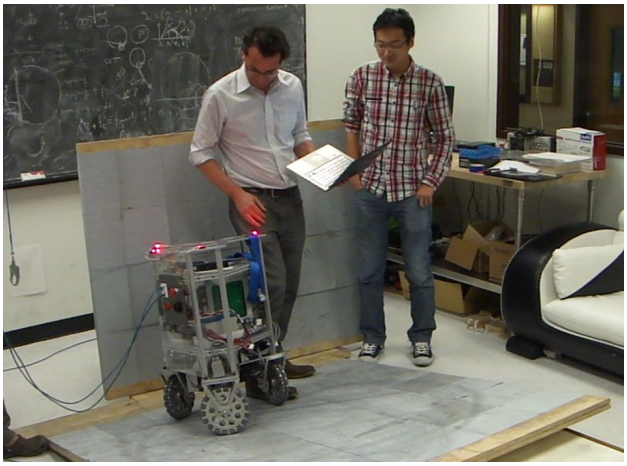


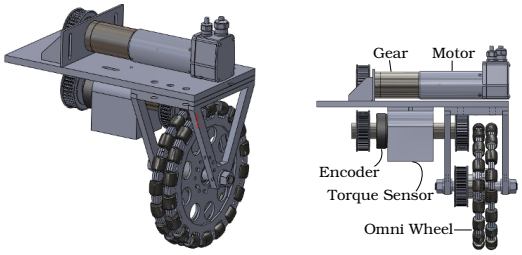
Fig. 1. **Colliding Against Humans:** Trikey safely bumps into people and corrects its heading based on the estimated contact direction.

passive bumpers [15] [8] or intelligent sensing using a variety techniques such as impact management accelerometers [17] or indoor positioning systems [14]. However, encouraging bumping into things has been largely discouraged [4] in large autonomous systems. This is true in most cases except for a few studies, such as manipulating by pushing [6] for instance. Overall, sensing and adapting to obstacles by bumping into them is a fairly unexplored area, which combined with rough terrain mobility seems to be even more unique.

Navigation using visual landmarks [1], in rough terrains using optimal control [10], and also in rough terrains while avoiding obstacles [21], [20] have been recently explored in the context of accurate mobile navigation. Although we are related to those studies, none of them addresses the issue on bumping and adapting to dynamic obstacles.

Robots with spring loaded casters [12] or trunks [13] have been developed for contact interactions in rough terrains but used in a very limited, even conceptual level. A step climbing robot based on omnidirectional wheels was presented in [22], however the focus was on stepping over obstacles rather than colliding on them and moving along their contour. An area where robots in contact have made a large impact is in physical human robot interaction using robot manipulators. Some of these pioneering studies include [23], [11], [2]. In contrast, our study focuses on mobile bases and on performing the adaptations on rough terrains.

In the area of contact sensing, there exists pioneering work focused on localization using force [5] or tactile sensing [7], [16]. Our methods for detecting contact are much simpler, based on comparing wheel or visual odometry with respect to the desired trajectories. However, in contrast with those studies we focus on mobile base contact mobility instead of manipulation. Our methods are related to those discussed in [9], which explores using articulated suspension to enhance tipover stability. One major difference is that we focus on colliding with objects which the previous work does not address.



(a) Model View (b) Front View

Fig. 2. **Drive Train of Trikey's Wheels:** The drive train consists of 3 axes (motor, sensor, and wheel), which are connected by timing belts to minimize mechanical backlash.

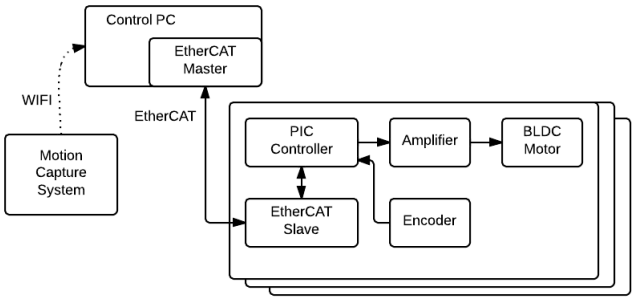


Fig. 3. **Embedded System:** Each actuation module consists of a motor, an amplifier, an encoder, a PIC processor, and a EtherCAT module. They are connected to a small desktop PC through a daisy chained EtherCAT bus.

III. Omni-directional Holonomic Mobile Robot Suited for Contact

Our base, Trikey (Fig 1), is an omni-directional mobile base designed for compliant interactions. The 40kg robot includes three 250W Maxon brushless DC motors with omni wheels in a triangular configuration (axes separated by 120 °). At 1:43 gear reduction, each wheel can achieve a maximum speed of 100 rpm and stall torque of 100N-m. The drive train (Fig 2) is specially designed to minimize friction, and it includes a rotary strain gauge torque sensor to enable torque feedback control of the wheels. It comprises motors, encoders and torque sensors, and omni wheels connected through a timing belt transmission.

As shown in the electrical diagram of Fig. 3, an on-board desktop PC communicates with embedded PIC motor controllers via EtherCAT serial ports. This communication architecture allows the robot to daisy chain its signals into one channel, resulting in more compact and efficient electrical wiring. The PIC controllers generate an analog signal every 500 μs to control the Elmo motor amplifiers for current control - the torque sensors where not enabled yet for this study. Four 12V lead acid batteries are embedded inside the base, allowing the robot to operate untethered for half an hour, while supplying average power of 150W.

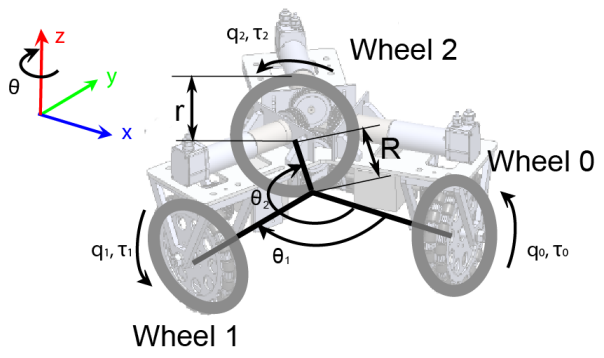


Fig. 4. Trikey Structure shows that the wheel axes are configured at 120 °.

IV. Control Approach

A. Constrained Dynamics

Our robot consists of a 3-dof wheeled system (shown in Fig. 4). To avoid directly deriving the kinematics between the robot's Cartesian frame and the wheel angles, we develop a planar floating body dynamic model of the base consisting of 6-dof generalized coordinates and kinematic constraints which will be later described in detail. This model includes the position and orientation of the center of the base on a plane, and the joint positions of the wheels:

$$\mathbf{q} = \begin{bmatrix} x & y & \theta & q_0 & q_1 & q_2 \end{bmatrix}^T \in \mathbf{R}^6. \tag{1}$$

To derive the floating base dynamic equation, we used Lagrange formalism as follows

$$\begin{aligned} L = T - V = &\frac{1}{2} M \left(\dot{x}^2 + \dot{y}^2\right) \\ &+ \frac{1}{2} I_M \dot{\theta}^2 + \frac{1}{2} I_m \left({\dot{q_0}}^2 + {\dot{q_1}}^2 + {\dot{q_2}}^2\right), \end{aligned} \tag{2}$$

where L is the Lagrangian, T is the kinetic energy, and V is the potential energy. Moreover, M, I_M and I_m are the total mass of the mobile robot, the total inertia around the vertical axis, and the inertia of a wheel around its rotating axis, respectfully. Using the Lagrangian,

$$\frac{\partial}{\partial t}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau, \tag{3}$$

we obtain the unconstrained system dynamics

$$\begin{bmatrix} M\ddot{x} \\ M\ddot{y} \\ I_M\ddot{\theta} \\ I_m\ddot{q_0} \\ I_m\ddot{q_1} \\ I_m\ddot{q_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \tau_0 \\ \tau_1 \\ \tau_2 \end{bmatrix}, \tag{4}$$

which can be expressed in matrix form as

$$\mathbf{A}\ \ddot{\mathbf{q}} = \mathbf{U}^T\ \mathbf{T}, \tag{5}$$

where

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} M\ \mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 1} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{1\times 2} & I_M & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{3\times 2} & \mathbf{0}_{3\times 1} & I_m\ \mathbf{I}_{3\times 3} \end{bmatrix}, \\ \mathbf{U} &= \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{I}_{3\times 3} \end{bmatrix}, \\ \mathbf{T} &= \begin{bmatrix} \tau_0 & \tau_1 & \tau_2 \end{bmatrix}^T. \end{aligned} \tag{6}$$

In the above model, because there is no direct dependency between the wheel and the base motion, we cannot control the position and the orientation of the robot using the wheel torques. As such, we introduce a new dependency based on rolling constraints on the wheels with respect to the terrain. We consider the velocity of the wheel center with respect to the terrain in direction of the wheel rotation, $\mathbf{v}_w$, and the velocity of the wheel center in the perpendicular plane $\mathbf{v}_r$. Here r stands for roller, as in the side rollers of the omni wheels. Then we can write the velocity of the i-th wheel, $\mathbf{v}_i$ as

$$\begin{aligned} \mathbf{v}_i &= \mathbf{v}_{w,i} + \mathbf{v}_{r,i} \\ &= \left[-v_{w,i}\sin\left(\theta+\theta_i\right) + v_{r,i}\cos\left(\theta+\theta_i\right)\right]\mathbf{i}_x \\ &\quad + \left[v_{w,i}\cos\left(\theta+\theta_i\right) + v_{r,i}\sin\left(\theta+\theta_i\right)\right]\mathbf{i}_y, \end{aligned} \tag{7}$$

where θ_i is the heading of the i-th wheel shown in Fig. 4. Additionally,

$$v_{w,i} = |\mathbf{v}_{w,i}| = r\ \dot{q}_i. \tag{8}$$

The velocity of each wheel can be described as the function of the position and the orientation of the center of the base from the rigid body kinematics.

$$\begin{aligned} \mathbf{v}_i &= \mathbf{v}_{CoM} + \omega \times \mathbf{p}_i \\ &= \dot{x}\ \mathbf{i}_x + \dot{y}\ \mathbf{i}_y + \dot{\theta}\ \mathbf{i}_z \times \mathbf{p}_i, \end{aligned} \tag{9}$$

where $\mathbf{p_i}$ is the vector from the center of the base to the i-th wheel:

$$\mathbf{p}_i = R\cos\left(\theta+\theta_i\right)\mathbf{i}_x + R\sin\left(\theta+\theta_i\right)\mathbf{i}_y. \tag{10}$$

When we combine Eqs. (7) and (9), we can derive the following kinematic equality.

$$\begin{aligned} &\begin{bmatrix} -v_{w,i}\sin\left(\theta+\theta_i\right) + v_{r,i}\cos\left(\theta+\theta_i\right) \\ v_{w,i}\cos\left(\theta+\theta_i\right) + v_{r,i}\sin\left(\theta+\theta_i\right) \end{bmatrix} \\ &\quad = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + R\ \dot{\theta}\begin{bmatrix} -\sin\left(\theta+\theta_i\right) \\ \cos\left(\theta+\theta_i\right) \end{bmatrix}. \end{aligned} \tag{11}$$

From Eqs. (8) and (11), we eliminate $v_{r,i}$ and derive the relationship between joint positions of the wheels and the position and orientation of the center of the base,

$$r\ \dot{q}_i = -\dot{x}\sin\left(\theta+\theta_i\right) + \dot{y}\cos\left(\theta+\theta_i\right) + R\ \dot{\theta}. \tag{12}$$

The above equation expressed for each wheel can be combined into matrix form as

$$r\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \mathbf{J}_{c,x}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}, \tag{13}$$

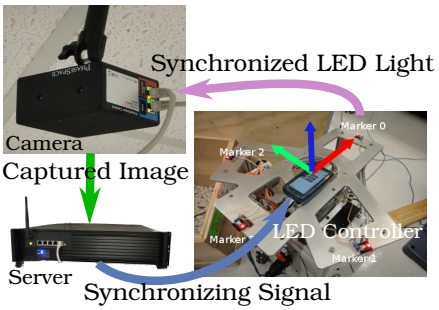


Fig. 5. **Motion Capture System:** four markers on Trikey allow for heading and position detection.

with,

$$\mathbf{J}_{c,x} \triangleq \begin{bmatrix} -\sin(\theta) & \cos(\theta) & R \\ -\sin\left(\theta - \frac{2}{3}\pi\right) & \cos\left(\theta - \frac{2}{3}\pi\right) & R \\ -\sin\left(\theta + \frac{2}{3}\pi\right) & \cos\left(\theta + \frac{2}{3}\pi\right) & R \end{bmatrix}. \quad (14)$$

By re-arranging Eq. (13) to relate to generalized velocities $\mathbf{q}$ we get

$$\mathbf{J}_c\ \dot{\mathbf{q}} = \mathbf{0}, \quad (15)$$

which uses a constrained Jacobian defined as

$$\mathbf{J}_c = \mathbf{J}_{c,wheel} \triangleq \begin{bmatrix} \mathbf{J}_{c,x} & -r\mathbf{I}_{3\times 3} \end{bmatrix} \in \mathbf{R}^{3\times 6}. \quad (16)$$

The above rolling constraints, can be incorporated into the robot's dynamics by extending Eq. (5) to

$$\mathbf{A}\ \ddot{\mathbf{q}} + {\mathbf{J}_c}^T \lambda_c = \mathbf{U}^T\ \mathbf{T}. \quad (17)$$

where λ_c are Lagrangian multipliers that model the constrained forces generated due to rolling friction between the wheels and the terrain.

Using [19], we derive a constraint null space $\mathbf{N}_c$ matrix and a constrained mass matrix Λ_c

$$\Lambda_c = \left(\mathbf{J}_c\ \mathbf{A}^{-1}\ \mathbf{J}_c^T\right)^+, \quad (18)$$

$$\mathbf{N}_c = \mathbf{I} - \overline{\mathbf{J}_c}\mathbf{J}_c, \quad (19)$$

where $\overline{\mathbf{J}_c} \triangleq \mathbf{A}^{-1}\mathbf{J}_c^T\Lambda_c$ is the dynamically consistent generalized inverse of $\mathbf{J}_c$ and $(\cdot)^+$ is the pseudo-inverse operator. By left-multiplying Eq. (17) by $\mathbf{J}_c\Lambda_c$ and merging it with the time derivative of Eq. (15), we solve for the Lagrangian multipliers λ_c as

$$\lambda_c = \overline{\mathbf{J}_c}^T\mathbf{U}^T\mathbf{T} - \Lambda_c\dot{\mathbf{J}}_c\dot{\mathbf{q}}. \quad (20)$$

Finally, by the above Equation into Eq. (17), we derive the robot constrained dynamics as

$$\ddot{\mathbf{q}} = \mathbf{A}^{-1}\ \mathbf{N}_c^T\ \mathbf{U}^T\ \mathbf{T} - \mathbf{J}_c^T\Lambda_c\dot{\mathbf{J}}_c\dot{\mathbf{q}}. \quad (21)$$

Once more, we highlight that we did not need to explicitly derive complex wheel kinematics.

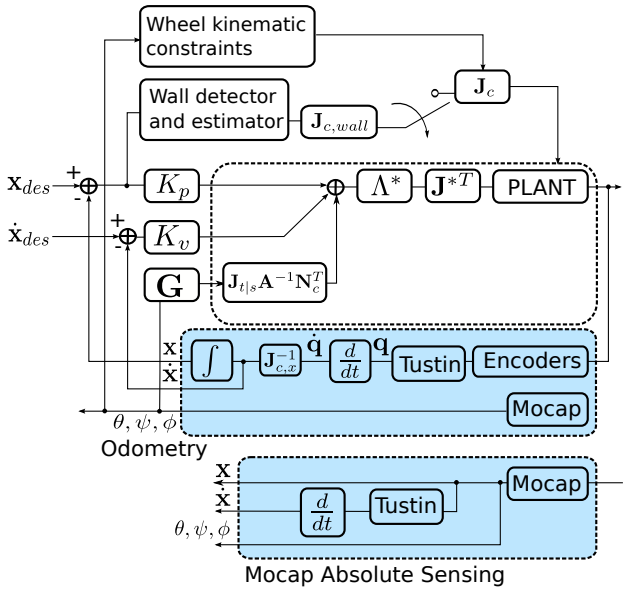


Fig. 6. **Control Diagram:** includes PD gains, an operational space controller, a constraint generator, and a gravity compensator. The output of the constraint generator, $\mathbf{J}_c$ is used to compute Λ^*, $\mathbf{J}_{t|s}$, $\mathbf{N}_c$, and $\mathbf{J}^*$.

B. Operational Space Controller

We define an operational space controller based on the robot's Cartesian coordinates x, y, and θ. In turn, the task kinematics can be defined as

$$\begin{aligned} \mathbf{x} &\triangleq \begin{bmatrix} x & y & \theta \end{bmatrix}^T, \\ \mathbf{J} &\triangleq \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix}, \\ \dot{\mathbf{x}} &= \mathbf{J}\ \dot{\mathbf{q}}. \end{aligned} \quad (22)$$

As discussed in [19], constrained kinematics can be expressed as a function of wheel velocities alone,

$$\dot{\mathbf{x}} = \mathbf{J}^* \begin{bmatrix} \dot{q}_0 & \dot{q}_1 & \dot{q}_2 \end{bmatrix}^T, \quad (23)$$

where

$$\mathbf{J}^* = \mathbf{J}_{t|s}\ \overline{\mathbf{U}\mathbf{N}_c}, \quad (24)$$

$$\mathbf{J}_{t|s} \triangleq \mathbf{J}\ \mathbf{N}_c, \quad (25)$$

are projections of the Jacobian into the space consistent with the rolling constraints and $\overline{(\cdot)}$ is a dynamically consistent inverse operator.

A dual expression of the above kinematics exists, providing the torque control strategy in operational space (see [18]) as

$$\mathbf{T} = \mathbf{J}^{*T}\ \mathbf{F}, \quad (26)$$

$$\ddot{\mathbf{x}} = \mathbf{J}_{t|s}\ \mathbf{A}^{-1}\ \mathbf{N}_c^T\ \mathbf{U}^T\mathbf{T} - \mathbf{J}_{t|s}\mathbf{J}_c^T\Lambda_c\dot{\mathbf{J}}_c\dot{\mathbf{q}}, \quad (27)$$

$$\mathbf{F} = \Lambda^*\ \mathbf{a}^{ref} + \Lambda^*\mathbf{J}_{t|s}\mathbf{J}_c^T\Lambda_c\dot{\mathbf{J}}_c\dot{\mathbf{q}}, \quad (28)$$

where $\mathbf{a}^{ref}$ is the acceleration command applied to the system in the operational space, as described in Eq. (22), and

$$\mathbf{\Lambda}^* \triangleq \left(\mathbf{J}_{t|s}\ \mathbf{A}^{-1}\ \mathbf{J}_{t|s}^T\right)^{-1}, \quad (29)$$

is the effective mass of the robot in the constrained space.

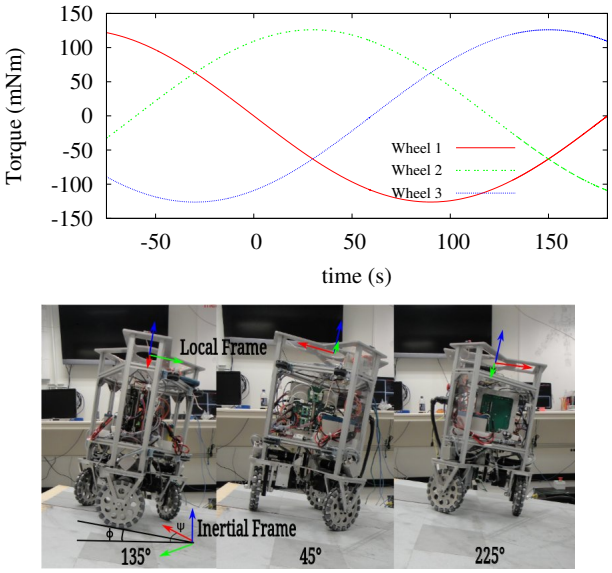


Fig. 7. **Gravity compensation:** Trikey stays up on the surface while a user turns around the base. Notice that no controllers where used to maintained the robot on the fixed position. Instead, gravity compensation prevented the robot from falling down. The graph shows the torques applied to the wheels.

C. Gravity Compensation

In the case of a sloped surface, every point on the X-Y plane has a certain potential energy. If the slope's degree is ϕ and the heading is ψ (see Fig. 7), the potential energy can be defined as

$$V = mg\cos\phi\left(x\cos\psi + y\sin\psi\right). \tag{30}$$

We can now extend Eq. (17) to include gravity as

$$\mathbf{A}\ \ddot{\mathbf{q}} + \mathbf{J}_c{}^T\lambda_c + \mathbf{G} = \mathbf{U}^T\ \mathbf{T}, \tag{31}$$

where

$$\mathbf{G} = mg\cos\phi\left[\begin{array}{cccccc}\cos\psi & \sin\psi & 0 & 0 & 0 & 0\end{array}\right]^T. \tag{32}$$

Again, using [19], λ_c is eliminated and the above dynamic equation can be further evolved to yield

$$\ddot{\mathbf{q}} = \mathbf{A}^{-1}\ \mathbf{N}_c^T\ \mathbf{U}^T\ \mathbf{T} - \mathbf{A}^{-1}\mathbf{N}_c^T\mathbf{G} - \mathbf{J}_c^T\Lambda_c\dot{\mathbf{J}}_c\dot{\mathbf{q}}. \tag{33}$$

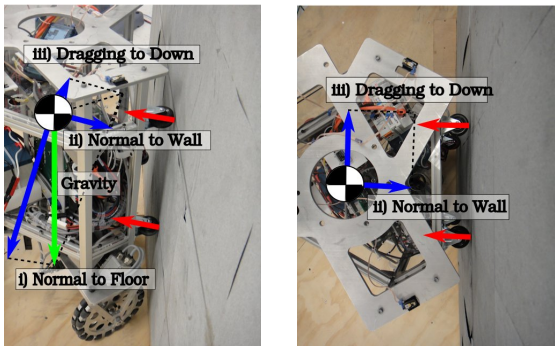
Finally, Eqs. (27) and (28) can be extended to include gravity effects,

$$\ddot{\mathbf{x}} = \mathbf{J}_{t|s}\ \mathbf{A}^{-1}\ \mathbf{N}_c^T\ \left(\mathbf{U}^T\mathbf{T} - \mathbf{G}\right) - \mathbf{J}_{t|s}\mathbf{J}_c^T\Lambda_c\dot{\mathbf{J}}_c\dot{\mathbf{q}}, \tag{34}$$

with

$$\mathbf{F} = \Lambda^*\ \mathbf{a}^{ref} + \Lambda^*\mathbf{J}_{t|s}\mathbf{A}^{-1}\mathbf{N}_c{}^T\mathbf{G} + \Lambda^*\mathbf{J}_{t|s}\mathbf{J}_c^T\Lambda_c\dot{\mathbf{J}}_c\dot{\mathbf{q}}. \tag{35}$$

From Eqs. (26) and (35), we can obtain motor torque commands while compensating for gravity disturbances (i.e without falling along the slope).



(a) Side View (b) Top View

Fig. 8. **Details on contact with the wall** The rollers attached on the side of the robot reduce the friction while in contact.

D. Contact Sensing and Adaptation

We focus for now on collisions and in particular on collisions against walls. A wall can be easily modeled as a first order inequality constraint. However, for simplicity we use an equality constraint defined as

$$y = ax + b. \tag{36}$$

A Jacobian describing such constraint can be formulated as

$$\begin{array}{l}\mathbf{J}_{c,wall} \triangleq \left[\begin{array}{cccccc} a & -1 & 0 & 0 & 0 & 0\end{array}\right], \\ \mathbf{J}_{c,wall}\ \dot{\mathbf{q}} = \mathbf{0}.\end{array} \tag{37}$$

When a new collision is encountered, such constraint can be added to the previous rolling constraints described in Eq. (16) extending the constrained Jacobian to

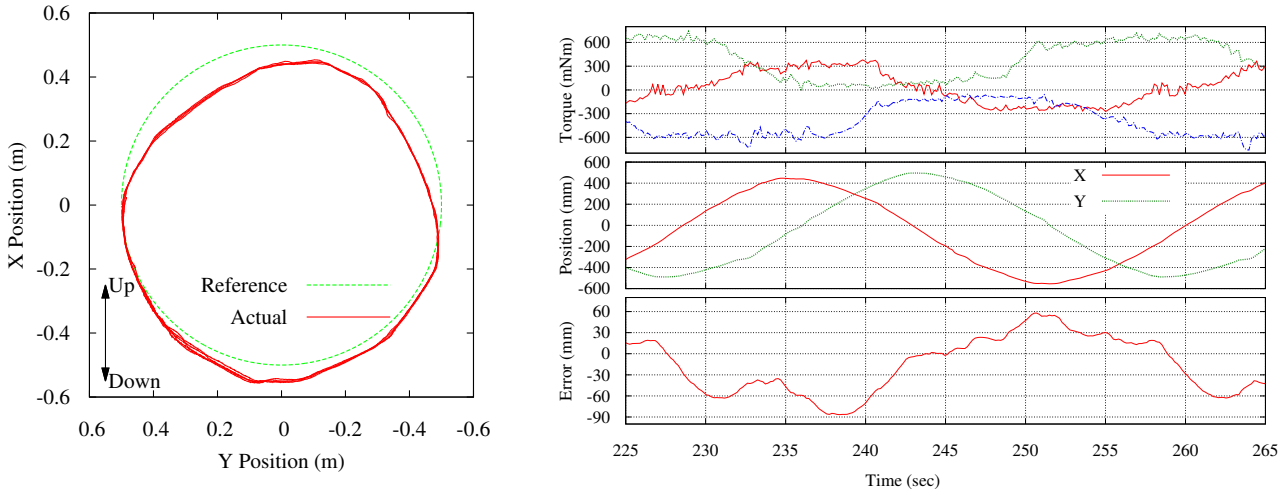
$$\mathbf{J}_c \triangleq \left[\begin{array}{c}\mathbf{J}_{c,wheel} \\ \mathbf{J}_{c,wall}\end{array}\right] \in \mathbf{R}^{4\times 6}. \tag{38}$$

We can now re-use the control equations defined in Eqs. (26) and (35) but using the above augmented Jacobian. As described in [19] the robot will minimize the distance to the desired trajectory if there exists a contact constraint in between.

To effectively modify the desired trajectory with respect to the contact constraint we need to detect the contact and estimate its direction. Although there are many sensing options, such as using onboard visual or LIDAR information, we choose to implement a simple sensing strategy based on a motion capture system. We measure the distance to the desired trajectory based on the motion captured data, and if the error is greater than a threshold we determine that a contact exists, i.e.

$$(\textbf{Wall Detection}) = \left\{\begin{array}{ll}\textbf{On}, & |\mathbf{x} - \mathbf{x}_{des}| \geq (\textbf{ErrTh}) \\ \textbf{Off}, & |\mathbf{x} - \mathbf{x}_{des}| < (\textbf{ErrTh})\end{array}\right. \tag{39}$$

where $\mathbf{x}$, $\mathbf{x}_{des}$, and $(\textbf{ErrTh})$ are the current and desired operational space coordinates, and the error threshold, respectively. We further estimate the wall heading by using a



(a) Circular Trajectory

(b) Torque, Position, and Accuracy

Fig. 9. **Circular Motion on Inclined Surface:** This figure shows the accuracy of tracking the desire circular path on a 10° terrain and aided by a MoCap system.

least square fitting formula defined as

$$a_n = \frac{n\left(\sum x'_i\, y'_i\right) - \left(\sum x'_i\right)\left(\sum y'_i\right)}{n\left(\sum {x'}_i^2\right) - \left(\sum x'_i\right)^2}, \tag{40}$$

$$x'_i \triangleq x_i + x_d, \tag{41}$$

$$y'_i \triangleq y_i + y_d. \tag{42}$$

where x_i and y_i are the X-Y data points of the center of the base while the base, x_d and y_d are the distances from the center of the base to its edge contacting the wall, and a_n is the estimated direction of the wall for iteration n. If x_d and y_d are constant, a_n depends only on x_i and y_i,

$$a_n = \frac{n\left(\sum x_i\, y_i\right) - \left(\sum x_i\right)\left(\sum y_i\right)}{n\left(\sum x_i^2\right) - \left(\sum x_i\right)^2}. \tag{43}$$

V. Experiments

We have built an adjustable inclined platform with a movable wall (Fig. 1) and attached sand papers to the terrain to increase surface traction. The trajectory of the robot is measured by a motion capture system from Phase Space that provides about 1mm precision. By attaching four active markers on the mobile robot (Fig. 5), the motion capture server processes and detects a single rigid body in real time. We use the captured data in the feedback control loop and the wall detection algorithm. Fig. 6 depicts the controller used in the experiments. The feedback input can be based on odometry or motion captured data. Velocity is derived by differentiation of the data using the Tustin Z transform. The state variables are delivered to the PD control loop, then fed to the wall detector and estimator module, to the wheel constraints, and to the gravity compensator. To verify the controller, we command the robot follow a circular trajectory in the inclined terrain. Trikey turns circles of various radii on both a flat surface and inclined terrains. The desired angular velocities are 1 rad/s for a 1m radius, and 0.5 rad/s for a 0.5m radius. The gains of the PD controller are adjusted to limit the forces upon collision through the compliant controllers. Additionally, we limit the velocity gain to avoid feedback chattering due to gear backlash.

A. Tracking Experiment on an Inclined Surface

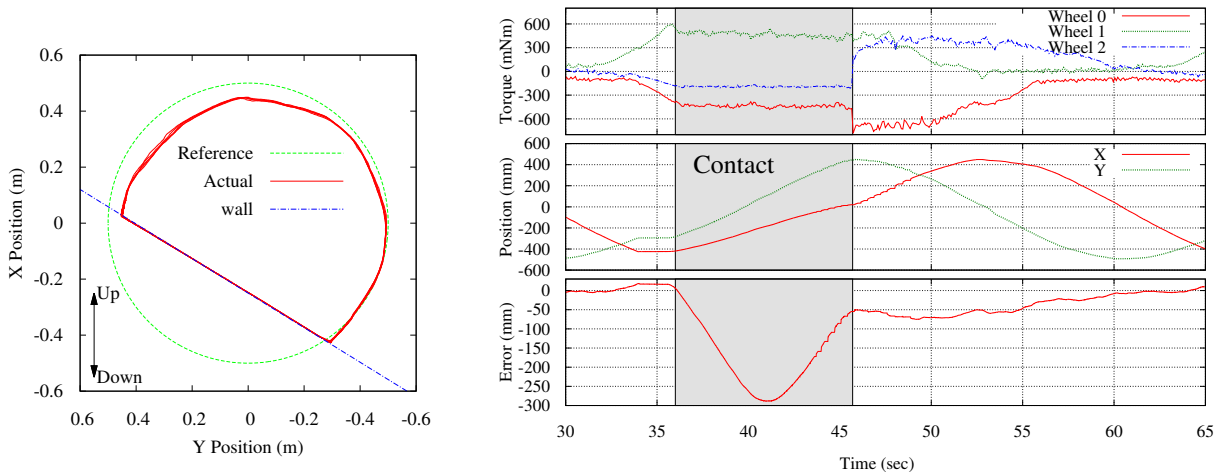
We placed the robot on a 10° slope. Obviously, when the robot is not powered it falls quickly down the surface because of the effect of gravity. We then turn on our Cartesian controller of Eq. 35, with zero desired accelerations, i.e. $a^{ref} = 0$, and the base stays at its initial position without falling down because of gravity compensation (see Fig. 7). Notice, that we rotate the base by hand to various orientations and it remains on the same location. For further validation, we now command the base to follow a circular trajectory on the inclined terrain as shown in in Fig. 9. As we can see, the tracking accuracy is about 10cm for the relatively moderate gains that we have chosen for our implementation.

B. Collision Experiment Against a Wall

Our second experiment consists on colliding with an unknown wall. As shown in Fig. 10, the robot tries to follow the desired circular trajectory but the contact constraint kicks in prompting the robot to move alongside the wall. Notice that to enable a smooth rolling of the base against the wall we attach 4 roller wheels to the side of the robot's base.

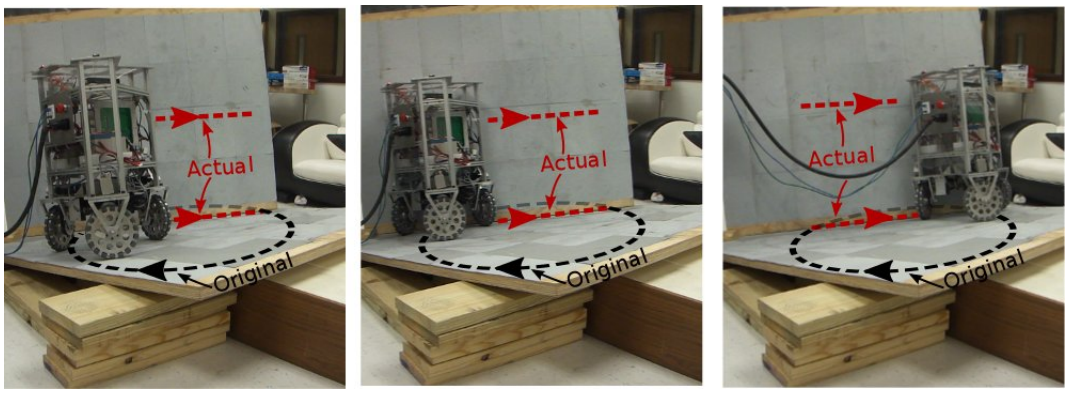
Fig. 10 also shows the torque commands applied to the wheels. We can see that the torque values remain constant during the contact phase, and increase right after the contact. That might be the result of applying less torques to compensate for gravity because the wall supports the base.

In Fig. 11, we change the orientation of the wall to various angles and observe that it is correctly detected and the estimated orientation is used to activate at runtime the contact consistent controller described in Fig. 6.



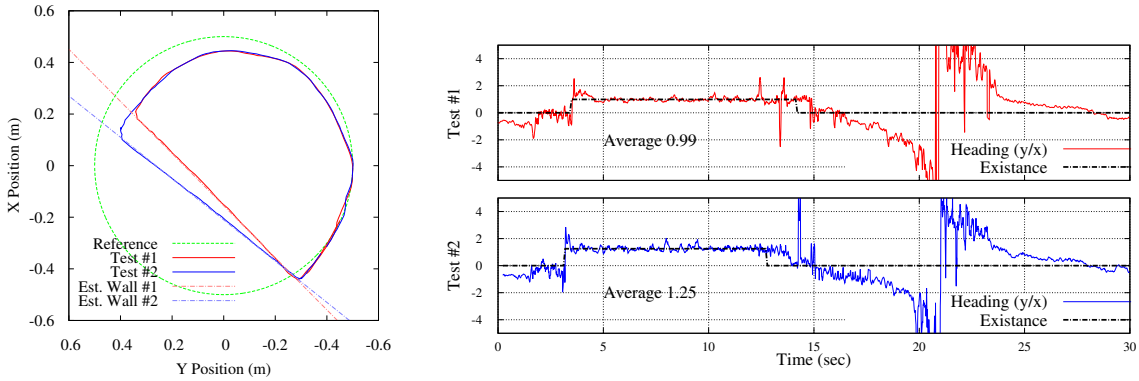
(a) Trajectory

(b) Torque, Position, and Error



(c) Trajectory on Terrain

Fig. 10. **Circular Motion During Contact:** Trikey tries to move on a circle until it encounters a wall. After safely colliding against it using the proposed compliant controller, it estimates the wall's direction based on the algorithm of Eq. (43) Aided by the switching controller of Fig. 6, it proceeds to move along the wall without applying excessive forces on the normal direction. Such behavior is enabled by projecting the acceleration on the null space of the wall constraint - refer to discussion around Eq. (38). The robot finally merges with the planned trajectory when the obstacle is cleared.



(a) Trajectory and Estimated Wall

(b) Estimated Wall Existence and Wall Direction

Fig. 11. **Detection of Wall at Various Orientations:** In these graphs obtained from experimentation, the wall is randomly oriented to various angles by an operator. Using the procedure discussed in Eq. (43), the robot estimates the direction of the wall while moving and by means of our motion capture system. As we can see in the right graphs, for the slope at 45° the robot estimates the slope to be of value 0.99 which is very close to the expected value of 1. For the slope at 50○ the robot estimates a slope of 1.25 which is also close to the expected value. The time period where the robot is in contact is between the times of 4 to 12.5 seconds.

VI. Conclusions

We have developed unique methods to maneuver in rough terrains with unknown contacts. We have also conducted various experiments to show the accuracy of our methods and the ability to reactively respond and adapt to the contact

disturbances. We believe that these capabilities will be very important to operate in human-centered and outdoor environments. Our sensing strategy currently relies on position error detection and linear fitting to the data points. Although it is a simplistic approach it has shown to be effective for wall detection.

For our controller, we have relied on sensor-based constraints given the obstacle estimations. An optional strategy would be to implement hybrid position / force control strategies that do not rely on constraints. However, in doing so we require to estimate the contact forces in order to compensate for them in operational space. In contrast, by accounting for reaction forces as contact constraints, direct estimation is not needed. We have observed that the force on the wall can help the robot to support its body effectively, reducing the amount of effort to compensate for gravity. This kind of behavior might be beneficial to recover from certain types of adverse scenarios or to deal with wheel slippage.

Compliant control not only allows the robot to create practical impedance behaviors for accidental collisions but also enables the base to compensate for gravity disturbances. Compensating for gravity allows the controller to employ smaller PD gains, as there are less disturbances unaccounted for. Using the same methods, we have also conducted successful experiments on safely colliding against humans (see Fig. 1).

In the near future we will seek to limit the amount of force applied to the wall to be safer and improve the interaction with the colliding objects. Also, we plan to use wheel odometry and the torque sensors on the wheels to help detecting the wall and providing better compliance control. As a natural extension of this research, we would like to handle collisions with more sophisticated object shapes and also with objects in motion, such as walking humans.

Further along, we plan to incorporate an upper body humanoid robot and perform mobile manipulation tasks while dealing with unexpected contacts. We also plan to incorporate visual sensors such as LIDAR and cameras to gain navigation autonomy.

Overall, we find that the area of human-centered robotics and outdoor navigation will benefit by incorporating compliant and online data-driven controllers to effectively deal with accidental or intentional contacts.

REFERENCES

[1] Maximilian Beinhofer, Jörg Müller, and Wolfram Burgard. Landmark placement for accurate mobile robot navigation. *European Conference on Mobile Robotics*, 2009.

[2] Alessandro De Luca, Alin Albu-Schaffer, Sami Haddadin, and Gerd Hirzinger. Collision detection and safe reaction with the dlr-iii lightweight manipulator arm. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 1623–1630. IEEE, 2006.

[3] Jodi Forlizzi and Carl DiSalvo. Service robots in the domestic environment: a study of the roomba vacuum in the home. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 258–265. ACM, 2006.

[4] Thierry Fraichard. A short paper about motion safety. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1140–1145. IEEE, 2007.

[5] Klaas Gadeyne and Herman Bruyninckx. Markov techniques for object localization with force-controlled robots. In *10th Intl Conf. on Advanced Robotics*. Citeseer, 2001.

[6] Brian P Gerkey and Maja J Mataric. Pusher-watcher: An approach to fault-tolerant tightly-coupled robot coordination. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 464–469. IEEE, 2002.

[7] W Eric L Grimson and Tomas Lozano-Perez. Model-based recognition and localization from sparse range or tactile data. *The International Journal of Robotics Research*, 3(3):3–35, 1984.

[8] Fujun He, Zhijiang Du, and Lining Sun. Intelligent detection of bumps in a mobile robot. In *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*, pages 136–141. IEEE, 2006.

[9] Karl Iagnemma, Adam Rzepniewski, Steven Dubowsky, and Paul Schenker. Control of robotic vehicles with actively articulated suspensions in rough terrain. *Autonomous Robots*, 14(1):5–16, 2003.

[10] Z. Kausar, K. Stol, and N. Patel. Performance enhancement of a statically unstable two wheeled mobile robot traversing on an uneven surface. In *Robotics Automation and Mechatronics (RAM), 2010 IEEE Conference on*, pages 156–162, 2010.

[11] Charles C Kemp, Aaron Edsinger, and Eduardo Torres-Jara. Challenges for robot manipulation in human environments [grand challenges of robotics]. *Robotics & Automation Magazine, IEEE*, 14(1):20–29, 2007.

[12] Oh-Hun Kwon, Hyunsoo Song, and Dong-Soo Kwon. A mobile robot platform based on spring loaded casters for physical interaction. In *RO-MAN, 2011 IEEE*, pages 156–161. IEEE, 2011.

[13] Hun-Ok Lim and Kazuo Tanie. Human safety mechanisms of human-friendly robots: passive viscoelastic trunk and passively movable base. *The International Journal of Robotics Research*, 19(4):307–335, 2000.

[14] JC Fuentes Michel, Hendrik Millner, and Martin Vossiek. A novell wireless forklift positioning system for indoor and outdoor use. In *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, pages 219–227. IEEE, 2008.

[15] Roberto Muselli, Armando Neri, and Gianluigi Orsi. Anti-collision safety device for forklift trucks and the like, May 25 1993. US Patent 5,213,383.

[16] Anna Petrovskaya and Oussama Khatib. Global localization of objects via touch. *Robotics, IEEE Transactions on*, 27(3):569–585, 2011.

[17] Ulyss R Rubey. Shockswitch, August 26 1980. US Patent 4,219,708.

[18] L. Sentis. *Synthesis and Control of Whole-Body Behaviors in Humanoid Systems*. PhD thesis, Stanford University, Stanford, USA, 2007.

[19] L. Sentis, J. Petersen, and R. Philippsen. Experiments with balancing on irregular terrains using the dreamer mobile humanoid robot. In *Robotics Science and Systems*, Sidney, July 2012.

[20] Pierre Sermanet, Raia Hadsell, Marco Scoffier, Matt Grimes, Jan Ben, Ayse Erkan, Chris Crudele, Urs Miller, and Yann LeCun. A multirange architecture for collision-free off-road robot navigation. *Journal of Field Robotics*, 26(1):52–87, 2009.

[21] Matthew Spenko, Yoji Kuroda, Steven Dubowsky, and Karl Iagnemma. Hazard avoidance for high-speed mobile robots in rough terrain. *Journal of Field Robotics*, 23(5):311–331, 2006.

[22] Atsushi Yamashita, Hajime Asama, Hayato Kaetsu, Isao Endo, and Tamio Arai. Development of a step-climbing omni-directional mobile robot. In *Proc. Int. Conf. on Field and Service Robotics*, pages 327–332, 2001.

[23] M. Zinn, O. Khatib, B. Roth, and J.K Salisbury. A new actuation approach for human-friendly robot design. *International Journal of Robotics Research*, 23(4/5):379–398, April-May 2004.