IEEE International Conference on Humanoid Robotics, Dec. 2009

Optimized Control Strategies for Wheeled Humanoids and Mobile Manipulators

Mike Stilman   Jiuguang Wang   Kasemsit Teeyapan   Ray Marceau

Abstract— **Optimizing the control of articulated mobile robots leads to emergent behaviors that improve the effectiveness, efficiency and stability of wheeled humanoids and dynamically stable mobile manipulators. Our simulated results show that optimization over the target pose, height and control parameters results in effective strategies for standing, acceleration and deceleration. These strategies improve system performance by orders of magnitude over existing controllers. This paper presents a simple controller for robot motion and an optimization method for choosing its parameters. By using whole-body articulation, we achieve new skills such as standing and unprecedented levels of performance for acceleration and deceleration of the robot base. We describe a new control architecture, present a method for optimization, and illustrate its functionality through two distinct methods of simulation.**

I. Introduction

Wheeled humanoids and mobile manipulators have become a compelling alternative to bipedal robots due to the potential for increased stability and safety in human environments. Many wheeled humanoids use a wide base of support and rely on static stability [1]–[4]. Others, such as the UBot [5], Robonaut [6] and BallBot [7] contact the ground with only two wheels or a ball. The latter systems are similar to bipedal humanoids since they use active control to achieve dynamic balance. Dynamic stability requires greater complexity in control, yet it also yields greater robot capabilities. For instance: (1) Balancing robots require very small support regions; (2) They can use their own mass to generate lever arms and achieve greater manipulation forces; (3) They typically achieve higher velocities which relate to greater momentum when interacting with the world. For quasi-static interactions, Thibodeau [8] showed that a balancing manipulator can achieve greater static forces when interacting with environment objects. We go beyond statics to show the dynamic advantages of articulated balancing robots. This paper presents three simple robot tasks: standing, acceleration and deceleration. In each case our methods use system dynamics to increase robot performance and stability.

Consider the robot Golem Krang in Fig. 1, currently under construction in our lab. The robot uses Schunk rotary modules for arms and a three-DOF torso. The waist joint can be controlled to tilt the entire upper torso with respect to the wheels. Due to upper body articulation, this robot will be able to sit, stand, and perform human-scale manipulation. We now show that the robot will also use joint articulation to increase its performance during navigation. For clarity, we focus on the torso pitch joint and its effects on dynamic balance. We model the system as a planar mechanism consisting of two massive links connected by a revolute joint. The lower link connects to a wheel with sufficient ground friction to maintain no-slip contact. Using this model, we demonstrate that articulated dynamic balance yields improved performance in navigation and naturally leads to human-like emergent behaviors.

The authors are with the Center for Robotics and Intelligent Machines at the Georgia Institute of Technology, Atlanta, GA 30332, USA. Emails: {golem, j.w, kasemsit, razorman}@gatech.edu

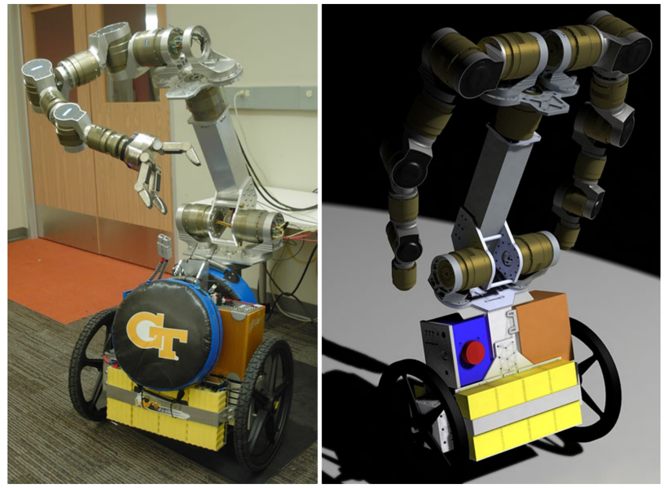


Fig. 1. Dynamically stable, articulated mobile manipulator in construction at the Humanoid Robotics Lab at Georgia Tech. We show how such a system can use upper body articulation to improve performance and stability.

In this paper, we first present a novel workspace controller for maintaining robot balance and placing the robot arms at a desired height. Second, we introduce an optimization strategy for choosing controller gains as well as robot height in order to maximize system performance. Optimization not only leads to increased performance, but also generates exciting results in robot behavior that can only be achieved with an articulated system.

II. Related Work

Numerous mobile manipulation platforms such as [5], [6], and [9] add upper body articulation to a dynamically stable wheeled base. However, typically these systems do not take advantage of the articulated links to improve navigation and balance. In contrast, work on bipedal humanoids, [10]–[12] has recently focused on the use of whole body articulation to improve balance and performance. For wheeled humanoids, [5] described a system that achieved greater static forces by leaning into objects. To our knowledge, no existing wheeled articulated platform has used its ability to move internal masses in order to increase its dynamic performance. We combine feedback linearization from classical control with numerical optimization to achieve this goal.

In contrast to bipedal platforms, a wheeled mobile manipulator is an underactuated system with only two points of ground contact at the wheels. Classical control of underactuated mechanical systems has focused on low dimensional models such as the cart-pole inverted pendulum, Furuta pendulum, inertia wheel pendulum, convey-crane system, ball and beam system, and the pendubot / acrobot which are well described in [13]. The most similar models to our work are the studies presented in [14], [15], and [16]. However, none of these combine the dynamics of base motion with upper body articulation.

While it is well-known that underactuated systems are not feedback linearizable, [17] showed that linearization can be done around the active or passive joints, referred to as collocated and non-collocated partial feedback linearization (PFL). Extensions proposed energy and passivity-based controls which regulate the energy of the system to some desired equilibrium point [18]–[20]. For these methods, stability guarantees were based on assumptions about system properties that do not directly translate to articulated mobile robots. Less restrictive strategies utilized switching and saturation [21] in the domain of hybrid systems. However, hybrid approaches often operate at the supervisory level, making them challenging to generalize or prove stability.

In contrast to classical methods, research in optimal control and reinforcement learning [22, 23] has applied numerical optimization to generating controllers over the entire system state space. Simple optimization over a discretized high-dimensional state space such as ours requires computation times that are currently infeasible. To mitigate complexity, researchers have applied variable resolutions [24], locally weighted models [25], and state space sampling [26]. Others have directly represented the best actions using quadratic programming [27], LQR [28], and optimized trajectories [29]. Presently, such techniques have not been applied to articulated wheeled robots.

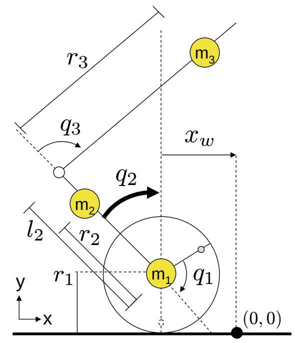
This paper combines classical feedback linearization with optimization. Feedback linearization allows the system to respond appropriately through a wide range of the state space given a simple PD controller. The gains and targets for the controller are chosen to optimally satisfy the desired criteria.

III. Control Design

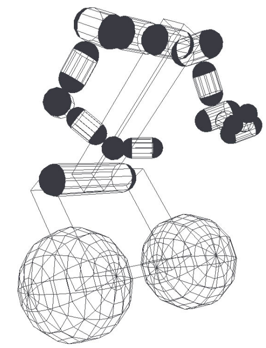
This section presents a simple controller that achieves three objectives: desired robot height, absolute position/velocity and relative position/velocity of the center of mass (CM). We augment the system dynamics equations with control equations that dictate the workspace accelerations of the wheel and the CM of the upper link. We use state feedback to linearize the entire system and compute torques by matrix inversion. This method can be thought of as a simple extension of computed torque for articulated robots. We developed it as a simple and intuitive alternative to partial feedback linearization [17].

A. Dynamics

First, we derive the system dynamics by the Lagrangian approach. The model parameters are shown in Fig. 2 and Table I. In accordance with general robot systems, the equations can be simplified to the following form that is linear in acceleration and torque. $\mathbf{T}$ represents torques while $\mathbf{M}$ and $\mathbf{N}$ are state-dependent matrices, such that $\mathbf{M}$ consists



(a) 2D Parameters



(b) 3D/Collision

Fig. 2. Dynamic models of our system. (a) Planar dynamics used for optimization. (b) 3D dynamics used in srLib simulations.

TABLE I
PHYSICAL PARAMETERS

	m (kg)	r (m)	I_z $(kg\cdot m^2)$	ℓ (m)
Wheel (l1)	12	.23	.166	.23
Link 2 (l2)	67.8	.15	1.4	.51
Link 3 (l3)	60.0	.8	3.0	-

of coefficients on acceleration and $\mathbf{N}$ consists of all other terms including centripetal, coriolis and gravitational forces.

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{T} \tag{1}$$

In our case, this is a system of three equations with q_2, the absolute angle of the lower link, q_1, the relative angle of the wheel and q_3, the relative angle of the upper link. Notice that one of the equations has 0 instead of torque. This is indicative of an underactuated system since only wheel torque, τ_1, and joint torque, τ_3, can be controlled.

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ 0 \\ \tau_3 \end{bmatrix} \tag{2}$$

This is a linear system of three equations with five unknowns. We can therefore add two equations to fully constrain the system. The next section shows that these two equations can be used to control the system.

B. Controller Design

Our controller adds to the system dynamics to stabilize the system around a desired equilibrium. First, we augment the system of equation in Eq. 2 with two more equations that are linear in acceleration. We define the matrices $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{S}$:

$$\mathbf{W} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \\ w_{41} & w_{42} & w_{43} \\ w_{51} & w_{52} & w_{53} \end{bmatrix} \mathbf{V} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ v_4 \\ v_5 \end{bmatrix} \text{ and } \mathbf{S} = \begin{bmatrix} \tau_1 \\ 0 \\ \tau_3 \\ s_4 \\ s_5 \end{bmatrix} \tag{3}$$

such that,

$$\mathbf{W}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}. \tag{4}$$

We now choose two simple PD control laws for the remaining equations. First, we would like to place the CM of the upper link at a desired height y_3^d, achieving a position for the robot arms. The position of y_3 is simply:

$$y_3 = r_1 + l_2 \cos(q_2) + r_3 \cos(q_2 + q_3) \tag{5}$$

To find the acceleration of y_3, we compute the second derivative of Eq. 5 with respect to time. This gives a linear equation in $\ddot{q}_1, \ddot{q}_2$ and $\ddot{q}_3$ with some additional terms, v_4, due to velocity. We now let s_4 be a simple PD controller for the height of the upper link center of mass.

$$\ddot{y}_3 = w_{41}\ddot{q}_1 + w_{42}\ddot{q}_2 + w_{43}\ddot{q}_3 + v_4 = s_4 \tag{6}$$
$$s_4 = k_1(y_3^d - y_3) + k_2(\dot{y}_3^d - \dot{y}_3) \tag{7}$$

Following this design method, we construct the second control equation to achieve robot balance and positioning. Initially we considered direct control over the robot CM, however we found it to have an unintuitive relationship to balance since the wheel must be accelerated in the direction opposite to the motion of the CM. Instead, we chose to control the absolute wheel axle acceleration $\ddot{x}_w$ directly. Both methods proved to be equally effective with very similar gains. For the case of wheel acceleration, we compute wheel position, relative and absolute robot CM position as follows:

$$x_w = r_1(q_1 + q_2) \tag{8}$$
$$x_{cm}^r = \frac{m_2 r_2 \sin(q_2) + m_3(l_2 \sin(q_2) + r_3 \sin(q_2 + q_3))}{m_2 + m_3} \tag{9}$$
$$x_{cm}^a = x_w + x_{cm}^r \tag{10}$$

Differentiating x_w with respect to time and choosing an appropriate control law yields a simple set of linear equations.

$$\ddot{x}_w = w_{51}\ddot{q}_1 + w_{52}\ddot{q}_2 + w_{53}\ddot{q}_3 + v_5 = s_5 \tag{11}$$
$$\begin{aligned} s_5 = {} & k_3(x_{cm}^{rd} - x_{cm}^r) + k_4(\dot{x}_{cm}^{rd} - \dot{x}_{cm}^r) + \\ & k_5(x_{cm}^{ad} - x_{cm}^a) + k_6(\dot{x}_{cm}^{ad} - \dot{x}_{cm}^a) \end{aligned} \tag{12}$$

Given current state information, Eq. 6-7 and Eq. 11-12 complete Eq. 4 such that we simply have five linear equations and five unknowns. Three of the equations are given by system dynamics and two are decided by PD control.

C. Control

Given five equations and five unknowns $(\ddot{q}_1, \ddot{q}_2, \ddot{q}_3, \tau_1, \tau_3)$ we can solve the system to find the accelerations of all the joints and more importantly the torques that achieve them. First, we separate $\mathbf{S}$ into $\mathbf{S} = \mathbf{S_1} + \mathbf{S_2}$.

$$\mathbf{S_1} = \begin{bmatrix}\tau_1 & 0 & \tau_3 & 0 & 0\end{bmatrix}^T$$
$$\mathbf{S_2} = \begin{bmatrix}0 & 0 & 0 & s_4 & s_5\end{bmatrix}^T$$

Two matrix subtractions in Eq. 4 yield:

$$\mathbf{W}\ddot{\mathbf{q}} - \mathbf{S_1} = \mathbf{S_2} - \mathbf{V}$$

which is trivially rearranged into the following form:

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & -1 & 0 \\ m_{21} & m_{22} & m_{23} & 0 & 0 \\ m_{31} & m_{32} & m_{33} & 0 & -1 \\ w_{41} & w_{42} & w_{43} & 0 & 0 \\ w_{51} & w_{52} & w_{53} & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \\ \tau_1 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 0 - n_1 \\ 0 - n_2 \\ 0 - n_3 \\ s_4 - v_4 \\ s_5 - v_5 \end{bmatrix}$$

Inverting the augmented $\mathbf{W}$ matrix results in the desired solution for accelerations and torques. We directly apply the torques to our system.

IV. Performance Optimization

A. Overview

The proposed simple workspace controller achieves a stable closed-loop system over a broad space of gains and parameters. However, performance is difficult to characterize and it is unclear how to compute a set of control actions that minimize some predefined cost. Traditionally, this problem can be addressed using optimal control, but doing so would require a complete reformulation of the controller which is difficult due to the constrained and nonlinear nature of our system. Instead, we focus on framing the optimization on top of the existing controller, keeping the same structural design but varying the controller gains through stochastic optimization for performance improvements.

We selected a particular optimization tool called Particle Swarm Optimization (PSO) [30], a stochastic, population-based evolutionary computing technique inspired by the paradigm of social interaction. PSO relies on the trajectories of a group of potential solutions called "particles", which traverses the solution space simultaneously to search for extrema points. Unlike traditional optimization algorithms that rely on gradient information, PSO do not explicitly compute the gradient but rather estimate the search direction through interactions with neighboring particles. At each time instance, a fitness function evaluates the quality of the solutions obtained by each particle and the value is shared across neighboring particles. The particles are attracted to its own best solution as well as the group's best solution, and over time, the group as a whole is drawn stochastically towards the global optimum.

Previously, we have successfully applied PSO in a state feedback design for a helicopter stabilization problem [31]. The merits of PSO lie in its ability to quickly converge to globally optimal solutions, even in large and non-convex solution spaces. The lack of dependence on gradients bypasses a computationally expensive process, especially for a complex and highly nonlinear model such as ours. The use of multiple particles ensures that the algorithm is not easily trapped in local minima, which can also be easily parallelized in future applications that require greater speed.

B. Formulation

We seek a set of controller gains $k_1 \dots k_6$ defined in Eq. 7 and Eq. 12 as well as the CM height y_3^d and desired absolute CM position x_{cm}^{ad} that minimize some cost function $\mathbf{J}$. $\mathbf{J}$ is defined for three scenarios:

1) Stand up: from a sit down position, stand up while minimizing translation and control effort
2) Acceleration: in a standing pose, accelerate to achieve a positive velocity as quickly as possible
3) Deceleration: traveling at some positive velocity, decelerate to stop as quickly as possible

In the fitness function, the dynamics of the system are simulated for T seconds and the resulting outputs are evaluated based on $\mathbf{J}$. The costs are:

$$\mathbf{J}_{stand} = \alpha \max(|q_1|) + \beta \sum_0^T \mathbf{u}\Delta t \quad (13)$$

$$\mathbf{J}_{acceleration} = \alpha st(\dot{q}_1) + \beta \sum_0^T \mathbf{u}\Delta t \quad (14)$$

$$\mathbf{J}_{deceleration} = \alpha st(\dot{q}_1) + \beta \sum_0^T \mathbf{u}\Delta t \quad (15)$$

where $st(\cdot)$ is the settling time for the given state and α, β are constant weights.

Using PSO, the controller parameters under consideration are encoded within particles, with appropriate restrictions placed as boundaries for the search space. Five particles are used in the search, initialized randomly within the search space. The algorithm iterates for a fixed number of iterations and returns the best solution found.

V. Experiments and Results

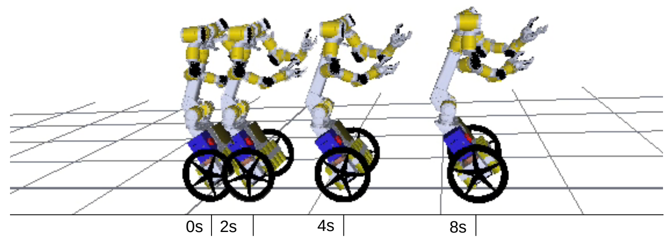
We evaluated the performance of our optimized controller relative to a set of hand tuned gains. Experiments were conducted on both a planar model in Matlab and a 3D model using srLib dynamic simulation. For standing, acceleration and deceleration, we observed significant improvement in translation, settling time and energy respectively. These results are summarized in Table II. Furthermore, since the optimizer could vary both robot height and target center of mass, we observed new behaviors that were generated automatically from optimized motion.

A. Acceleration

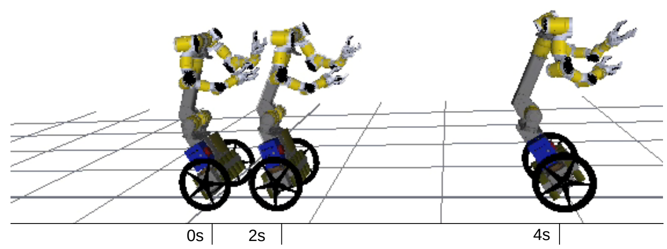
In order to make the robot accelerate, we give the controller a non-zero reference velocity for the absolute center of mass, $\dot{x}_{cm}^{ad} = .5m/s$. Both the hand tuned gains and the parameters selected by optimization achieve this velocity starting from zero and track it. However, as seen in Fig. 3, the optimized controller reaches the target velocity in less than half the time. As shown in the accompanying video, the tuned controller continued to oscillate while the optimized gains result in a gradual, smooth change to the velocity. This final traces of optimized control are given in Fig. 3(c).

B. Standing Up

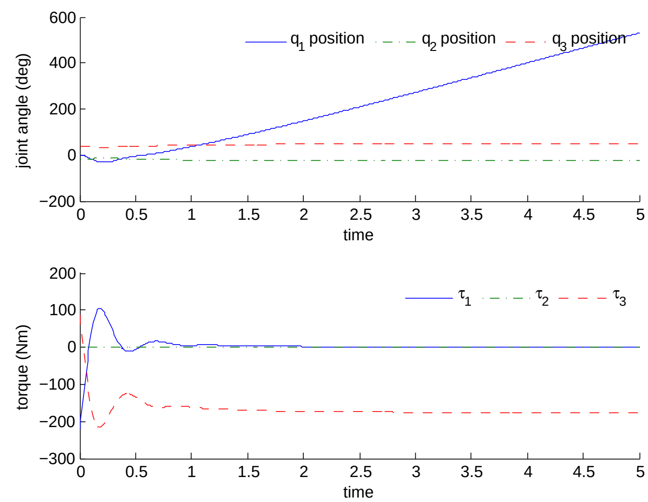
We designed our robot with the capability to sit and stand, allowing it to achieve a statically stable sitting configuration



(a) Control prior to optimization.



(b) Control after optimization.



(c) Simulated execution of optimized controller.

Fig. 3. Acceleration: optimized control operates faster with less oscillation.

yet also use dynamic balance for complex dynamic tasks. Setting the initial angles in the range of ± 100 makes the robot take a sitting position. From this pose, we set a target height for the second link of $y_3 = 1.2m$. As shown in Fig. 4, the optimized controller significantly outperformed our choice of parameters. The controller showed almost no oscillation and the simulated robot nearly stood in place.

Standing also showed the first emergent behavior of our two-link system. Not only does the robot unfold its torso, but it first slides the wheels back approximately $50cm$ and then lifts the torso to stabilize at the desired setpoint.

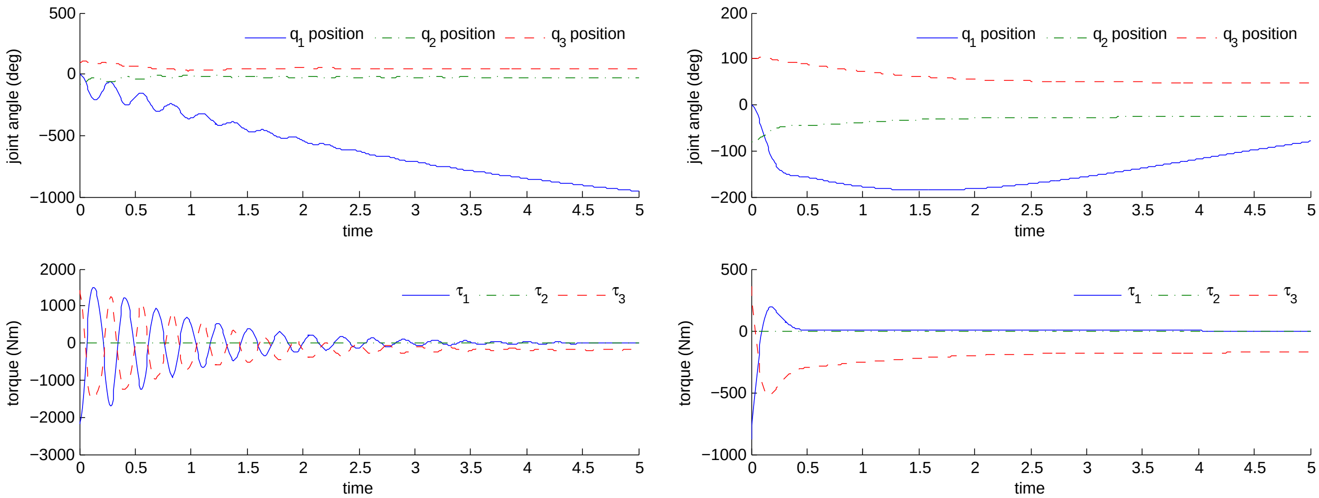
TABLE II

Performance Optimization

	Max Translation	Energy
Original Stand up	16.64	92407
Optimized Stand up	3.24	107450
	Settling Time	Energy
Original Acceleration	4.68	80076
Optimized Acceleration	2.26	82739
Original Deceleration (Sit)	4.82	180580
Optimized Deceleration (Sit)	2.37	178250
Optimized Deceleration (Stand)	2.16	86305

C. Deceleration

The most interesting robot behaviors were observed in deceleration. Just as with acceleration, we simply set the target velocity to $\dot{x}_{cm}^{ad} = 0m/s$. As shown in Fig. 7, while our original gains did stop the robot, it took a significant amount of time to settle. Such a slow deceleration would not be sufficient to prevent the robot from experiencing or causing



(a) Control prior to optimization.

(b) Control after optimization.

Fig. 4. Stand Up Controller: optimization leads to a smoother controller which nearly makes the robot stand in place.

hazardous situations. The optimizer was asked to select the remaining parameters, including the gains, the height of the upper link and the target mass position. Not only did it find significantly safer solutions for deceleration, but also generated two distinct behaviors.

On separate runs of optimization, the particles settled on distinct local minima. In one case, the robot found that lowering the upper link, y_3 required lower gains and gave a much faster settling time. We attribute this to the increased sensitivity of our balancing system when it has a lower center of mass. On a different run, optimization found a desirable absolute robot position, x_{cm}^{ad}, and placed a high gain, k_3, on achieving it. The robot moves the wheels forward and leans back in order to stop. These two new behaviors are demonstrated in Fig. 5 and Fig. 6.

VI. Discussion

In this paper, we presented a controller that generates whole body motion for wheeled humanoids and mobile manipulators. We also introduced an optimization strategy for choosing control parameters. The combination of these simple elements led to stable motion in a number of simulated tasks: standing, acceleration and deceleration. The most interesting result was the set of emergent behaviors such as leaning and sitting that emerged from optimizing task performance.

Upon completion of our wheeled humanoid robot, we will validate this controller on a real robot platform. We have individually tested the balancing capabilities of the base, validated the model parameters and ensured that our torso actuators are capable of producing the required torques. Future work on this topic will extend the controller to include the robot arms. We will also consider optimization over trajectories to generate more complex body motions that handle both obstacle avoidance and physical interactions.

References

[1] D. Katz, E. Horrell, Y. Yang, B. Burns, T. Buckley, A. Grishkan, V. Zhylkovskyy, O. Brock, and E. Learned-Miller, "The UMass mobile manipulator uMan: An experimental platform for autonomous mobile manipulation," in *Workshop on Manipulation in Human Environments at Robotics: Science and Systems*, 2006.

[2] F. Zacharias, C. Borst, M. Beetz, and G. Hirzinger, "Positioning mobile manipulators to perform constrained linear trajectories," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept. 2008, pp. 2578–2584.

[3] C. Borst, T. Wimbock, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. R. Giordano, R. Konietschke, W. Sepp, S. Fuchs, C. Rink, A. Albu-Schaffer, and G. Hirzinger, "Rollin' Justin - mobile platform with variable base," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 1597–1598.

[4] T. Inamura, K. Okada, S. Tokutsu, N. Hatao, M. Inaba, and H. Inoue, "HRP-2W: A humanoid platform for research on support behavior in daily life environments," *Robotics and Autonomous Systems*, 2008.

[5] P. Deegan, B. Thibodeau, and R. Grupen, "Designing a self-stabilizing robot For dynamic mobile manipulation," in *Robotics: Science and Systems-Workshop on Manipulation for Human Environments*, 2006.

[6] R. Ambrose, H. Aldridge, R. Askew, R. Burridge, W. Bluethmann, M. Diftler, C. Lovchik, D. Magruder, and F. Rehnmark, "Robonaut: Nasa's space humanoid," *Intelligent Systems and their Applications, IEEE*, vol. 15, no. 4, pp. 57–63, Jul/Aug 2000.

[7] T. Lauwers, G. Kantor, and R. Hollis, "A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, May 2006, pp. 2884–2889.

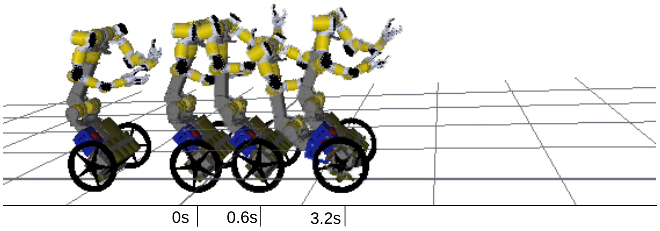
[8] B. Thibodeau, P. Deegan, and R. Grupen, "Static analysis of contact forces with a mobile manipulator," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, May 2006, pp. 4007–4012.

[9] P. Abeygunawardhana and M. Toshiyuki, "Stability improvement of two wheel mobile manipulator by real time gain control technique," in *Industrial and Information Systems, 2007. ICIIS 2007. International Conference on*, Aug. 2007, pp. 79–84.

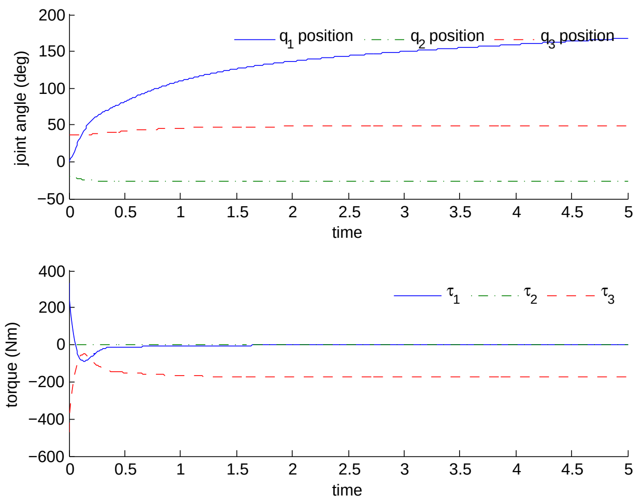
[10] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 2, Sept. 2003, pp. 1620–1626 vol.2.

[11] K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired zmp," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3, 2002, pp. 2684–2689 vol.3.

[12] K. Harada, S. Kajita, F. Kanehiro, K. Fujiwara, K. Kaneko, K. Yokoi, and H. Hirukawa, "Real-time planning of humanoid robot's gait for force-controlled manipulation," *Mechatronics, IEEE/ASME Transactions on*, vol. 12, no. 1, pp. 53–62, Feb. 2007.

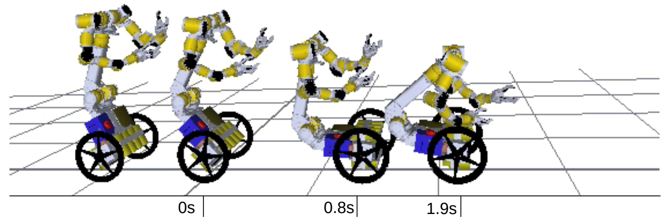


(a) Optimizer chooses to lean back for deceleration.

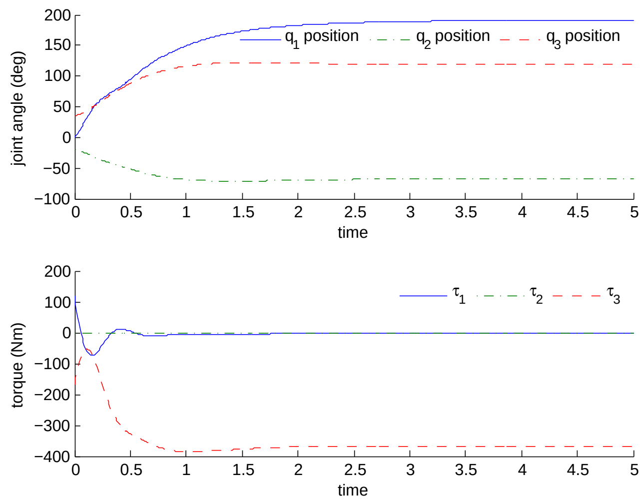


(b) Traces of simulated execution.

Fig. 5. Deceleration 1: Leaning behavior emerges from optimization.



(a) Optimizer chooses to lower height for deceleration.



(b) Traces of simulated execution.

Fig. 6. Deceleration 2: Sitting behavior emerges from optimization.

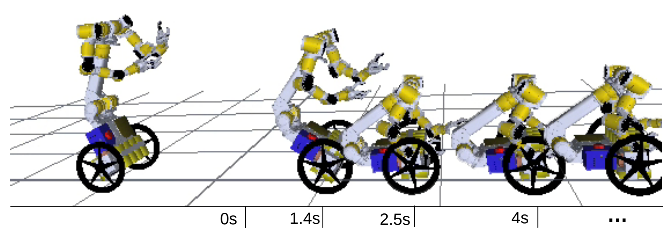


Fig. 7. Deceleration: Experimentally tuned gains.

[13] I. Fantoni and R. Lozano, *Non-linear control for underactuated mechanical systems*. Springer, 2001.

[14] M. Bergerman, C. Lee, and Y. Xu, "Experimental study of an underactuated manipulator," in *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 2, Aug 1995, pp. 317–322 vol.2.

[15] H. Arai, K. Tanie, and N. Shiroma, "Feedback control of a 3-dof planar underactuated manipulator," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 1, Apr 1997, pp. 703–709 vol.1.

[16] A. De Luca, R. Mattone, and G. Oriolo, "Stabilization of underactuated robots: theory and experiments for a planar 2r manipulator," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, Apr 1997, pp. 3274–3280 vol.4.

[17] M. Spong, "Partial feedback linearization of underactuated mechanical systems," in *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on*, vol. 1, Sep 1994, pp. 314–321 vol.1.

[18] ——, "The swing up control problem for the acrobot," *Control Systems Magazine, IEEE*, vol. 15, no. 1, pp. 49–55, Feb 1995.

[19] I. Fantoni, R. Lozano, and M. Spong, "Energy based control of the pendubot," *Automatic Control, IEEE Transactions on*, vol. 45, no. 4, pp. 725–729, Apr 2000.

[20] Y. Dong, Z. Wang, Z. Feng, D. Wang, and H. Fang, "Energy-based control for a class of under-actuated mechanical systems," in *Image and Signal Processing, 2008. CISP '08. Congress on*, vol. 3, May 2008, pp. 139–143.

[21] M. Spong and L. Praly, "Control of underactuated mechanical systems using switching and saturation," *Lecture Notes in Control and Information Sciences*, pp. 162–172, 1996.

[22] A. Barto and R. Sutton, *Reinforcement Learning: an introduction*. MIT Press Cambridge, MA, 1998.

[23] G. Boone, "Minimum-time control of the acrobot," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, Apr 1997, pp. 3281–3287 vol.4.

[24] "Variable resolution discretization for high-accuracy solutions of optimal control problems," in *In International Joint Conferences on Artificial Intelligence*, 1999, pp. 1348–1355.

[25] C. Atkeson, A. Moore, and S. Schaal, "Locally weighted learning," *Artificial intelligence review*, vol. 11, no. 1, pp. 11–73, 1997.

[26] C. Atkeson and B. Stephens, "Random sampling of states in dynamic programming," *IEEE Transactions on Systems Man and Cybernetics*, vol. 38, no. 4, pp. 924–929, 2008.

[27] S. Kudoh, T. Komura, and K. Ikeuchi, "The dynamic postural adjustment with the quadratic programming method," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3, 2002, pp. 2563–2568 vol.3.

[28] A. Kuo, "An optimal control model for analyzing human postural balance," *Biomedical Engineering, IEEE Transactions on*, vol. 42, no. 1, pp. 87–101, Jan. 1995.

[29] C. Atkeson and B. Stephens, "Multiple balance strategies from one optimization criterion," in *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, 29 2007-Dec. 1 2007, pp. 57–64.

[30] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov/Dec 1995, pp. 1942–1948 vol.4.

[31] J. Wang, B. T. Brackett, and R. G. Harley, "Particle Swarm-Assisted State Feedback Control: From Pole Selection to State Estimation," in *2009 American Control Conference*, St. Louis, Missouri, USA, June 2009.