IEEE Int’l Conf. on Robotics and Automation (ICRA), May 14-18, 2012

Planning in High-dimensional Shape Space for a Single-wheeled Balancing Mobile Robot with Arms

Umashankar Nagarajan, Byungjun Kim and Ralph Hollis

***Abstract*— The ballbot with arms is an underactuated balancing mobile robot that moves on a single ball. Achieving desired motions in position space is a challenging task for such systems due to their unstable zero dynamics. This paper presents a novel approach that uses the dynamic constraint equations to plan shape trajectories, which when tracked will result in optimal tracking of desired position trajectories. The ballbot with arms has shape space of higher dimension than its position space and therefore, the procedure uses a user-defined weight matrix to choose between the infinite number of possible combinations of shape trajectories to achieve a particular desired trajectory in position space. Experimental results are shown on the real robot where different motions in position space are achieved by tracking motions of either the body lean angles, or the arm angles or combinations of both.**

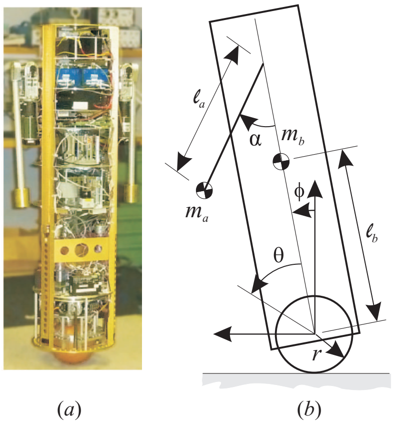
I. Introduction

Balancing, dynamically stable mobile robots can be effective personal robots and are a welcome departure from their statically stable counterparts. They can be tall enough to have eye-level interactions and can be narrow enough to navigate cluttered human environments. They are also capable of safe, gentle physical interaction and have the dynamic capabilities to move with speed and grace comparable to that of humans. The last decade has seen growing interest in two-wheeled balancing robots ([1], [2], [3]); a lot of them inspired by the Segway Robotic Mobility Platform [4]. Our group introduced the ballbot [5], a balancing mobile robot that moves on a single ball. The single, spherical wheel enables the robot to achieve omni-directional motion overcoming the limitations associated with the kinematic constraints of two-wheeled robots. Recently, other groups have also been exploring single-wheeled balancing robot designs [6], [7]. In the present work, we have added a pair of 2-DOF arms to the ballbot making it, as far as we know, the first omnidirectional single-wheeled balancing mobile robot having arms.

Balancing mobile robots, by virtue of underactuation, have constraints on their dynamics that restrict the family of trajectories their configurations can follow. These constraints are second-order nonholonomic constraints, $i.e.$, non-integrable acceleration or dynamic constraints. The configuration space of any dynamic system can be divided into position space and shape space. Position variables describe the position of the system in world coordinates, whereas shape variables are those that affect the inertia matrix of the system. The dynamics of most mechanical systems, especially mobile robots, are independent of the position variables and for the balancing mobile robots, the overall dynamics is dominated by the shape dynamics. The strong coupling between the dynamics of the position and shape variables makes tracking desired position space motions with zero shape change an impossible task. Any attempt to do so will result in jerky motions or drive the system unstable.

Given an accurate model, a variety of nonlinear inversion-based approaches are available in nonlinear control literature to approximately track desired position trajectories [8], [9]. Modeling uncertainties, unmodeled dynamics, nonlinear friction effects and disturbances can make these techniques ineffective on real robots. For balancing mobile robots like the ballbot, the uncertainties arise primarily from the actuator mechanisms and nonlinear friction effects of the soft ball rolling on the floor. However, having a model provides us valuable information like the dynamic constraints that help us exploit the dynamic coupling between the shape and the position configurations.

In [10], we presented a planning procedure for balancing mobile robots like the ballbot that plans shape trajectories, which when tracked will result in approximate tracking of the desired position trajectories. However, this work dealt only with the case where there are an equal number of position and shape variables. The ballbot with arms has a shape space of higher dimension than its position space since the arm configurations are also shape variables. This paper extends the planning procedure presented in [10] to plan in higher dimensional shape space to achieve desired motions in position space and also presents experimental results on the real robot.



(a) (b)

Fig. 1. (a) The ballbot with a pair of 2-DOF arms; and (b) Planar configurations shown in a planar model of the ballbot with a single arm.

This work is supported by NSF Grants IIS-0308067 and IIS-0535183.

U. Nagarajan, B. Kim and R. Hollis are with The Robotics Institute, Carnegie Mellon University, Pittsburgh PA 15213, USA umashankar@cmu.edu, byungjun@cmu.edu and rhollis@cs.cmu.edu

II. The Ballbot with Arms

The ballbot [5] is a balancing mobile robot that moves on a single ball. It has a cylindrical body that is about 1.5 m tall and weighs about 55 kg. The ball is driven using an inverse mouse-ball drive mechanism with four DC servomotors. An inertial measurement unit (IMU) provides the body lean angles. A more detailed system description of the ballbot without the arms is available in [11].

Recently, a pair of 2-DOF arms were added to the robot [Fig. 1(a)]. Each arm is an aluminium tube that is 0.457 m long and 0.89 mm thick with a changeable dummy weight (up to 2 kg) at its end. The arm attaches to its drive unit through a shoulder structure shown in Fig. 2.

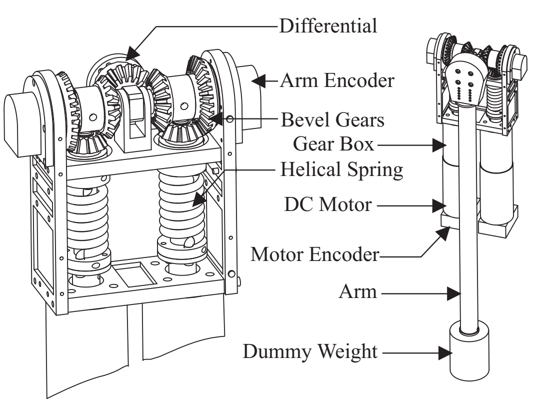


Fig. 2. 2-DOF arm with series-elastic actuators.

Each arm is actuated by a pair of series-elastic actuators, each of which consists of a custom designed helical spring with a torsion coefficient of 16.37 Nm/rad, a brush DC motor with a torque of 0.12 Nm at 3000 RPM, a 91:1 planetary gear train, and a 2000 CPR encoder. Each actuator connects with a pair of bevel gears of gear ratio 1:2 and a 1024 CPR optical encoder is attached to the end of the bevel gear shaft. The differential with three miter gears combines the torque from each actuator. The entire drive unit is fixed to a deck on the ballbot body.

A. Dynamic Model

The ballbot with arms is modeled as a rigid cylinder on top of a rigid sphere with a pair of massless arms having weights at their ends. The model makes the following assumptions: (i) there is no slip between the ball and the floor; and (ii) there is no yaw/spinning motion for either the ball or the body or the arms, $i.e.$, they have two degrees of freedom each. A planar version of the model along with the planar configurations is shown in Fig. 1(b).

There are eight configuration variables for the 3D ballbot model with arms represented by $q = [\theta, \alpha_l, \alpha_r, \phi]$, where, $\theta = [\theta_x, \theta_y]^T$ are configurations of the ball, $\alpha_l = [\alpha_{lx}, \alpha_{ly}]^T$ are configurations of the left arm, $\alpha_r = [\alpha_{rx}, \alpha_{ry}]^T$ are configurations of the right arm, and $\phi = [\phi_x, \phi_y]^T$ are configurations of the body. The forced Euler-Lagrange equations of motion of the ballbot with arms can be written in matrix form as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \begin{bmatrix} \tau \\ \mathbf{0} \end{bmatrix}, \tag{1}$$

where, $M(q) \in \mathbb{R}^{8\times8}$ is the mass/inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{8\times8}$ is the matrix of Coriolis and centrifugal terms, $G(q) \in \mathbb{R}^{8\times1}$ is the vector of gravitational forces and $\tau = [\tau_\theta, \tau_{\alpha_l}, \tau_{\alpha_r}]^T \in \mathbb{R}^{6\times1}$ is the vector of generalized forces. The body configurations ϕ are unactuated, whereas the rest of the configurations are actuated. Equation 1 shows that the ballbot with arms is an *underactuated system* [12] because there are fewer independent control inputs than there are configuration variables.

The configuration variables that appear in the inertia matrix are called *shape variables* (q_s), whereas the configuration variables that do not appear in the inertia matrix are called *external/position variables* (q_x). For the 3D ballbot model with arms, $q_x = \theta$ and $q_s = [\alpha_l, \alpha_r, \phi]^T$. Here, all position configurations are actuated, whereas shape configurations have both actuated and unactuated variables. The matrices in Eq. 1 are of the form:

$$M(q) = \begin{bmatrix} M_{\theta\theta} & M_{\theta\alpha_l}(q_s) & M_{\theta\alpha_r}(q_s) & M_{\theta\phi}(q_s) \\ M_{\alpha_l\theta}(q_s) & M_{\alpha_l\alpha_l}(q_s) & M_{\alpha_l\alpha_r}(q_s) & M_{\alpha_l\phi}(q_s) \\ M_{\alpha_r\theta}(q_s) & M_{\alpha_r\alpha_l}(q_s) & M_{\alpha_r\alpha_r}(q_s) & M_{\alpha_r\phi}(q_s) \\ M_{\phi\theta}(q_s) & M_{\phi\alpha_l}(q_s) & M_{\phi\alpha_r}(q_s) & M_{\phi\phi}(q_s) \end{bmatrix}, \tag{2}$$

$$C(q,\dot{q}) = \begin{bmatrix} \mathbf{0} & C_{\theta\alpha_l}(q_s,\dot{q}_s) & C_{\theta\alpha_r}(q_s,\dot{q}_s) & C_{\theta\phi}(q_s,\dot{q}_s) \\ \mathbf{0} & C_{\alpha_l\alpha_l}(q_s,\dot{q}_s) & \mathbf{0} & C_{\alpha_l\phi}(q_s,\dot{q}_s) \\ \mathbf{0} & \mathbf{0} & C_{\alpha_r\alpha_r}(q_s,\dot{q}_s) & C_{\alpha_r\phi}(q_s,\dot{q}_s) \\ \mathbf{0} & C_{\phi\alpha_l}(q_s,\dot{q}_s) & C_{\phi\alpha_r}(q_s,\dot{q}_s) & C_{\phi\phi}(q_s,\dot{q}_s) \end{bmatrix}, \tag{3}$$

$$G(q) = \begin{bmatrix} \mathbf{0} \\ G_{\alpha_l}(q_s) \\ G_{\alpha_r}(q_s) \\ G_\phi(q_s) \end{bmatrix}, \tag{4}$$

where, each $M_{ij} \in \mathbb{R}^{2\times2}$, each $C_{ij} \in \mathbb{R}^{2\times2}$ and each $G_i \in \mathbb{R}^{2\times1}$. The elements of the above matrices are not presented here due to space constraints. Equations 1$-$ 4 show that the dynamics of the system is independent of both position and velocity of the position variables, $i.e.$, $(\theta, \dot{\theta})$.

The last two equations of motion corresponding to the unactuated shape variables ϕ form the *dynamic constraint equations*. These are *second-order nonholonomic constraints* as they are not even partially integrable [13]. The dynamic constraint equations in Eq. 1 are given by

$$\begin{aligned} M_{\phi\theta}(q_s)\ddot{\theta} + M_{\phi\alpha_l}(q_s)\ddot{\alpha}_l + M_{\phi\alpha_r}(q_s)\ddot{\alpha}_r + M_{\phi\phi}(q_s)\ddot{\phi} \\ + C_{\phi\alpha_l}(q_s,\dot{q}_s)\dot{\alpha}_l + C_{\phi\alpha_r}(q_s,\dot{q}_s)\dot{\alpha}_r + C_{\phi\phi}(q_s,\dot{q}_s)\dot{\phi} \\ + G_\phi(q_s) = \mathbf{0}. \end{aligned} \tag{5}$$

III. Planning in Shape Space

Our objective here is to use the dynamic constraint equations to plan shape trajectories, which when tracked will result in approximate tracking of desired position trajectories. Eq. 5 shows that non-zero shape changes can result in acceleration in position space. Such systems are called *shape-accelerated underactuated balancing systems* [10].

The dynamic constraint equations help us understand the relationship between shape configurations and acceleration in position space. Let's first discuss a simple case where the robot sticks to a constant, non-zero shape configuration.

A constant, non-zero shape configuration q_s with $\dot{q}_s = \mathbf{0}$ and $\ddot{q}_s = \mathbf{0}$ reduces the dynamic constraints equations in Eq. 5 to:

$$M_{\phi\theta}(q_s)\ddot{\theta} + G_\phi(q_s) = \mathbf{0}. \tag{6}$$

Since $M_{\phi\theta}$ is invertible in the neighborhood of the origin [10], one gets

$$\begin{aligned} \ddot{\theta} &= -M_{\phi\theta}(q_s)^{-1}G_\phi(q_s) \\ &= \Gamma'(q_s). \end{aligned} \tag{7}$$

If the system tracks a constant shape configuration, the nonlinear map $\Gamma'(q_s) \in \mathbb{R}^{2\times 6}$ provides the constant acceleration the system will achieve in position space. Equation 7 is also valid for cases where the non-zero shape configurations cancel each other's effect to produce zero acceleration in position space. However, the nonlinear map $\Gamma'(q_s)$ is not invertible as the shape space is of higher dimension than the position space. Therefore, given a desired constant acceleration in position space, there is no unique constant shape configuration that will achieve it. In fact, there are an infinite number of constant shape configurations that will achieve the desired motion. Here, we present ways to find such configurations. Jacobian linearization of Eq. 7 w.r.t. q_s at $q_s = \mathbf{0}$ gives

$$\begin{aligned} \frac{\partial\ddot{\theta}}{\partial q_s}\bigg|_{q_s=\mathbf{0}} &= \frac{\partial\Gamma'}{\partial q_s}\bigg|_{q_s=\mathbf{0}} \\ &= \left[\ \frac{\partial\Gamma'}{\partial\alpha_l}\bigg|_{q_s=\mathbf{0}},\quad \frac{\partial\Gamma'}{\partial\alpha_r}\bigg|_{q_s=\mathbf{0}},\quad \frac{\partial\Gamma'}{\partial\phi}\bigg|_{q_s=\mathbf{0}}\ \right] \\ &= \left[\ K^0_{\alpha_l},\quad K^0_{\alpha_r},\quad K^0_\phi\ \right] \\ &= K^0_{q_s}. \end{aligned} \tag{8}$$

The matrix $K^0_{q_s} \in \mathbb{R}^{2\times 6}$ is not invertible, whereas its constituent submatrices $K^0_{\alpha_l} \in \mathbb{R}^{2\times 2}$, $K^0_{\alpha_r} \in \mathbb{R}^{2\times 2}$, $K^0_\phi \in \mathbb{R}^{2\times 2}$ are all invertible. The matrix $K^0_{q_s}$ and in turn, its constituent submatrices are functions of only the system parameters, and hence are constant matrices. These constant submatrices quantitatively represent the contribution each group of shape variables have on the acceleration of the system in position space, $i.e.$, $K^0_{\alpha_l}$, $K^0_{\alpha_r}$ and K^0_ϕ represent the contribution the left arm angles, right arm angles and body angles have on the acceleration of the ball respectively. Therefore, their inverses let us derive shape configurations, which when tracked will result in desired acceleration in position space. However, the individual inverses assume that the other shape configurations are zero.

Therefore, given a desired constant acceleration in position space $\ddot{\theta}^d$, the constant shape configurations that must be tracked to achieve $\ddot{\theta}^d$ can be written as:

$$q^p_s = WK^0_\theta\ddot{\theta}^d, \tag{9}$$

where,

$$W = \begin{bmatrix} W_{\alpha_l} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & W_{\alpha_r} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & W_\phi \end{bmatrix} \in \mathbb{R}^{6\times 6}, \tag{10}$$

$$K^0_\theta = \begin{bmatrix} (K^0_{\alpha_l})^{-1} \\ (K^0_{\alpha_r})^{-1} \\ (K^0_\phi)^{-1} \end{bmatrix} \in \mathbb{R}^{6\times 2}. \tag{11}$$

Here, the matrices $(K^0_{\alpha_l})^{-1}$, $(K^0_{\alpha_r})^{-1}$ and $(K^0_\phi)^{-1}$ are all diagonal matrices and hence, the weight matrix W is chosen such that $W_{\alpha_l} + W_{\alpha_r} + W_\phi = I_2$, a 2×2 identity matrix. The weight matrix W allows the user to relatively weigh the contribution each group of shape variables make in achieving the desired acceleration in position space $\ddot{\theta}^d$. For example, the user can choose from either a pure body lean motion or a pure arm motion or any combination of the two.

In this paper, we do not use the conventional pseudo-inverse approach as the pseudo-inverse of $K^0_{q_s}$ will return only a single set of shape configurations q_s, whereas, the decoupled inverse approach using the weight matrix W offers more flexibility and allows us to explore the space of infinite possible shape configurations. This is particularly useful when certain physically meaningful behaviors are desired, for example, constrained arm motions for carrying objects, no arm motion while moving between narrow walls, $etc.$

A. Optimal Shape Trajectory Planner

When the desired acceleration in position space is a non-constant, time-varying trajectory $\ddot{\theta}^d(t)$, the corresponding shape trajectories $q^p_s(t)$ will also be time-varying. The dynamic constraint equations in Eq. 5 can be re-written as:

$$\begin{aligned} \ddot{\theta} &= -M_{\phi\theta}(q_s)^{-1}\Big(M_{\phi\alpha_l}(q_s)\ddot{\alpha}_l + M_{\phi\alpha_r}(q_s)\ddot{\alpha}_r + \\ &\quad M_{\phi\phi}(q_s)\ddot{\phi} + C_{\phi\alpha_l}(q_s,\dot{q}_s)\dot{\alpha}_l + C_{\phi\alpha_r}(q_s,\dot{q}_s)\dot{\alpha}_r + \\ &\quad C_{\phi\phi}(q_s,\dot{q}_s)\dot{\phi} + G_\phi(q_s)\Big) \\ &= \Gamma(q_s,\dot{q}_s,\ddot{q}_s). \end{aligned} \tag{12}$$

In this paper, we extend the shape trajectory planner presented in [10] to plan in high dimensional shape space to achieve optimal tracking of desired position trajectories $\ddot{\theta}^d(t)$. Inspired by Eq. 9, given a desired acceleration trajectory in position space $\ddot{\theta}^d(t)$ and a weight matrix W, we propose to find a constant linear map K_θ such that the planned shape trajectory

$$q^p_s(t) = WK_\theta\ddot{\theta}^d(t), \tag{13}$$

when tracked will result in an acceleration trajectory $\ddot{\theta}^p(t)$ that approximates the desired acceleration trajectory $\ddot{\theta}^d(t)$. Here, W and K_θ have the same structure as in Eq. 10 and Eq. 11 respectively.

The shape trajectory planning procedure can now be formulated as an optimization problem, where the elements of K_θ are determined with the objective of minimizing

$$J = \int_t \left\|\ddot{\theta}^p(t) - \ddot{\theta}^d(t)\right\|^2_2, \tag{14}$$

where, $\ddot{\theta}^p(t) = \Gamma(WK_\theta\ddot{\theta}^d(t), WK_\theta\dddot{\theta}^d(t), WK_\theta\ddddot{\theta}^d(t))$. Any optimization algorithm that solves nonlinear least-squares problems can be used. $K_\theta = K_\theta^0$ ensures optimality for a constant desired acceleration trajectory, whereas it may not necessarily ensure optimality for a general $\ddot{\theta}^d(t)$ but will act as a good initial guess for the optimization process.

The shape trajectory planning procedure presented above talks only about tracking desired acceleration trajectories $\ddot{\theta}^d(t)$ but not desired position trajectories $\theta^d(t)$. Desired position trajectories $\theta^d(t)$ can be tracked by tracking the corresponding acceleration trajectories $\ddot{\theta}^d(t)$ only if the system starts at the correct initial conditions, $i.e.$, $\theta^p(0) = \theta^d(0)$ and $\dot{\theta}^p(0) = \dot{\theta}^d(0)$. When the initial conditions match, approximate tracking of the desired acceleration trajectories result in approximate tracking of the desired position trajectories.

B. Control Architecture

The shape planner assumes that there exists controllers that can accurately track the planned shape trajectories. For the 3D ballbot model with arms, the shape configurations include actuated arm configurations and unactuated body configurations. We use the balancing controller described in [11] to track the planned body angle trajectories. In order to track the planned arm angle trajectories, we use the computed torque method [14] that provides open-loop control values and a PID position controller for closed-loop control.

We can observe that the tracking of desired position trajectories $\theta^d(t)$ by tracking planned shape trajectories $q_s^p(t)$ is open-loop with no feedback on the position configurations. This procedure cannot ensure approximate tracking when the system starts at wrong initial conditions. Moreover, while testing on real robots, there are more issues such as modeling uncertainties, unmodeled dynamics, nonlinear friction effects and noise that will prevent good tracking of desired position trajectories. To overcome these issues, we use a feedback position trajectory tracking controller, similar to the one in [15], that adds compensation shape trajectories $q_s^c(t)$ to the planned shape trajectories $q_s^p(t)$, thereby producing the desired shape trajectories $q_s^d(t)$ that are tracked by the balancing and arm controllers as shown in Fig. 3.

C. Choosing Desired Position Trajectories

Since the planned shape trajectories $q_s^p(t)$ depend on the desired acceleration trajectories $\ddot{\theta}^d(t)$, the shape trajectory planner requires that the desired position trajectories $\theta^d(t)$ must be at least of differentiability class C^2, $i.e.$, the first two derivatives exist and are continuous. However, it is preferred to have $\theta^d(t)$ be of differentiability class C^4 so that the planned shape trajectories $q_s^p(t)$ and their first two derivatives ($\dot{q}_s^p(t)$, $\ddot{q}_s^p(t)$) that depend on them exist and are continuous.

The desired position trajectories $\theta^d(t)$ must also satisfy acceleration bounds that depend on the shape variables used to achieve these motions. For the results presented in Sec. IV, the acceleration bounds are set to 1 m/s^2 and 0.082 m/s^2 for using the body angles and arm angles respectively. These values correspond to a 5° body lean and a 55° arm angle (for a 1 kg end mass) respectively. These bounds represent the neighborhood around the origin where the linear proportionality between shape change and ball acceleration is valid. Since the ballbot's body has a larger moment of inertia than its arm, there is a larger acceleration bound for the body angles than the arm angles.

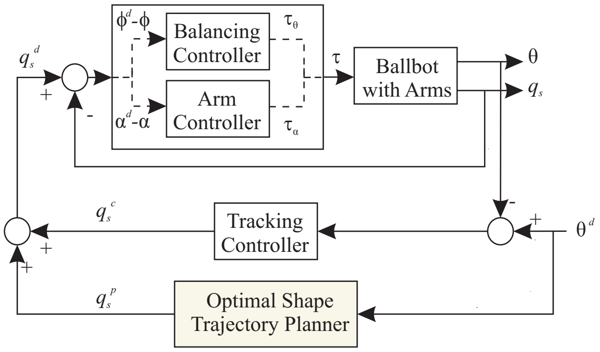


Fig. 3. Control Architecture

D. Choosing Weight Matrices

The shape trajectory planner assumes that a valid weight matrix W is chosen. The conditions that determine its validity are as follows. Each element w_{ij} of the weight matrix W must be non-negative, $i.e.$, $w_{ij} \geq 0$. The weight matrix W must be of the form shown in Eq. 10 and its constituent submatrices must sum to a 2×2 identity matrix, $i.e.$, $W_{\alpha_l} + W_{\alpha_r} + W_\phi = I_2$.

The weight matrix can also be used to account for self-collision constraints. Its elements can be chosen such that the arm motions do not collide with the body. Let's consider the case of the ballbot achieving a lateral ball motion using just the arms. Here, a single arm cannot produce the whole motion as it will result in collision with the body. So, one arm must be used for the "acceleration-phase" and the other arm must be used for the "deceleration-phase". This can be achieved by using a different weight matrix for each phase. Such a case is experimentally demonstrated in Sec. IV-B.

E. Real-Time Planning

For the results presented in Sec. IV, the optimization tolerance values for both the residual norm and the parameter values were set to $< 10^{-3}$. On a standard Intel Core-2 Duo processor, the optimization implementation in MATLAB converges in < 9 seconds. A well optimized C/C++ implementation can provide the results an order of magnitude faster, which allows real-time planning on the ballbot.

IV. Experimental Results

This section presents the experimental results of the ballbot with arms achieving desired position space motions using the optimal shape trajectory planner and the control architecture described in Sec. III. User-defined weight matrices are used to choose between the body and the arm motions. For all the results presented in this paper, the ballbot arms have 1 kg dummy weights at their ends. The companion video, "Shape Space Planning for Ballbot with Arms", shows the ballbot with arms perform the motions presented here.

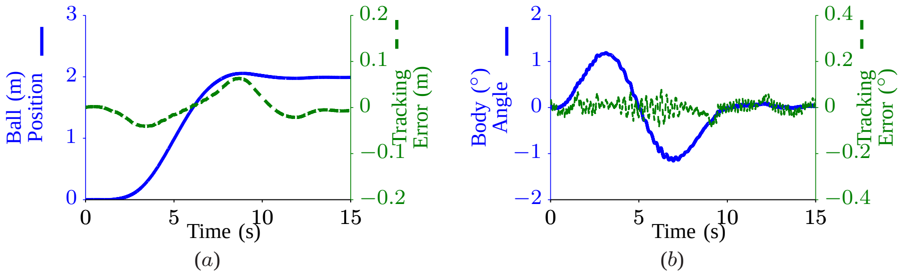


Fig. 4. Tracking forward straight line ball motion using only body motion: (a) ball position trajectory; (b) body angle trajectory.

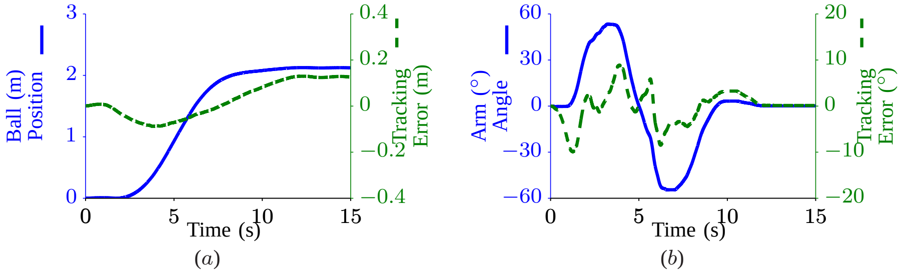


Fig. 5. Tracking forward straight line ball motion using only arm motion: (a) ball position trajectory; (b) left arm angle trajectory.

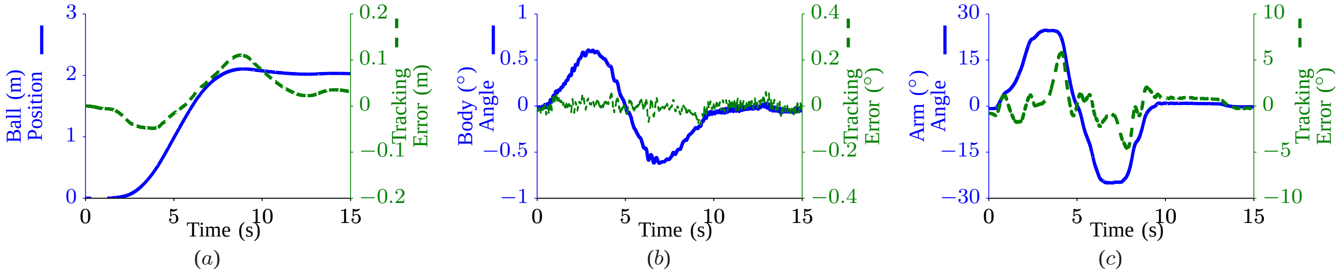


Fig. 6. Tracking forward ball motion using both arm and body motions: (a) ball position trajectory; (b) body angle trajectory; (c) right arm angle trajectory.

A. *Forward Ball Motion*

Here, we present the results of the ballbot with arms tracking a desired straight line ball motion of 2 m in the forward direction using three different shape motions: (i) pure body motion; (ii) pure arm motion; and (iii) a combination of both body and arm motions.

1) Pure Body Motion: Figure 4(a) shows the real ballbot with arms tracking the desired ball motion by tracking just the desired body angle trajectory, which is a sum of the planned body angle trajectory given by the shape trajectory planning procedure and the compensation trajectory obtained through feedback. The arms were maintained at zero angles for this experiment. The experimental body angle trajectory along with its tracking error is shown in Fig. 4(b). The planner's effectiveness is demonstrated by the small compensation body angles, which remained within $\pm 0.08°$.

2) Pure Arm Motion: The ball position tracking performance while using just the arms to achieve the 2 m motion is shown in Fig. 5(a). The left arm angle trajectory is shown in Fig. 5(b) and a similar result was obtained for the right arm. The compensation arm angles remained within $\pm 5°$, while the body angles were maintained within $\pm 0.05°$. Compared to the results in Fig. 4(a), Fig. 5(a) shows that there is larger ball position tracking error while using just the arms. This is due to the relatively poor trajectory tracking performance of the arm controller as shown in Fig. 5(b), which in turn is due to some excessive backlash in the arm gears. We plan to improve this design in the future.

3) Body and Arm Motion: Here, the body and the arm motions equally share (50-50) the effort of tracking the desired ball motion as shown in Fig. 6(a). Resulting trajectories for the body and the right arm are shown in Fig. 6(b) and Fig. 6(c) respectively. A similar result was obtained for the left arm. The compensation body and arm angles remained within $\pm 0.06°$ and $\pm 5°$ respectively.

B. *Lateral Ball Motion*

Here, we present the results of the ballbot with arms tracking a desired straight line ball motion of 1 m in the lateral direction using just the arm motions. The resulting ball motion and the tracking error are shown in Fig. 7(a). The arms are moved sideways and the right arm is used during the "acceleration-phase" to initiate the motion as shown in Fig. 7(b), whereas the left arm is used during the "deceleration-phase" to bring the system to rest as shown in Fig. 7(c). As discussed in Sec. III-D, two different weight matrices were used for the two phases in order to avoid

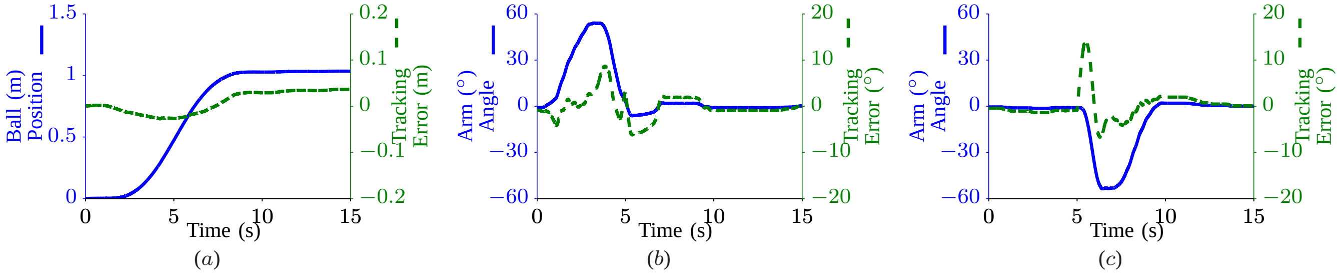


Fig. 7. Tracking lateral ball motion using only arm motion: (*a*) ball position trajectory; (*b*) right arm angle trajectory; (*c*) left arm angle trajectory.

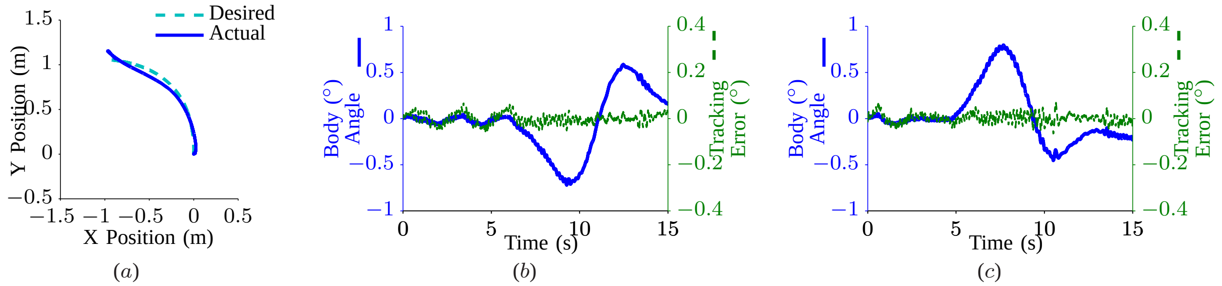


Fig. 8. Tracking curvilinear ball motion using only body motion: (*a*) XY ball position; (*b*) X body angle trajectory; (*c*) Y body angle trajectory.

self-collision. The compensation arm angles remained within $\pm5^\circ$, while the body angles remained within $\pm0.05^\circ$.

C. Curvilinear Ball Motion

The ballbot with arms tracking a curvilinear ball motion using just the body motion is shown in Fig. 8(*a*). The body angle trajectories and the tracking errors are shown in Fig. 8(*b*) and Fig. 8(*c*). The compensation body angles remained within $\pm0.15^\circ$ for this experiment.

V. Conclusions

This paper presents an optimal shape trajectory planner that uses dynamic constraint equations to plan trajectories in high-dimensional shape space, which when tracked will result in optimal tracking of desired position trajectories. The planner uses a user-defined weight matrix that determines the contribution of each shape variable in achieving the desired position space motion. A feedback position trajectory tracking controller was used to achieve better tracking. This paper also successfully demonstrated the effectiveness of the shape planner and the control architecture in tracking desired motions on the real ballbot with arms by choosing between pure body motions, pure arm motions and their combinations.

VI. Future Work

We will test the shape planning procedure with heavier weights at the end of the arms in order to achieve faster and more dynamic motions. The work presented must be extended to include cases where some shape configurations are constrained and non-zero, *e.g.*, carrying an object with the arm; and still achieve desired motions in position space. The problem of handling disturbances will also be addressed. In this paper, the weight matrices were chosen by the user and we will explore ways to automatically choose these weight matrices depending on the desired navigation task.

References

[1] Y. Takahashi, S. Ogawa, and S. Machida, "Step climbing using power assist wheel chair robot with inverse pendulum control," in *Proc. IEEE Intl. Conf. on Robotics and Automation*, 2000, pp. 1360–65.

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

[3] M. Stilman, J. Olson, and W. Gloss, "Golem Krang: Dynamically stable humanoid robot for mobile manipulation," in *IEEE Int'l Conf. on Robotics and Automation*, 2010, pp. 3304–3309.

[4] H. G. Nguyen, J. Morrell, K. Mullens, A. Burmeister, S. Miles, N. Farrington, K. Thomas, and D. Gage, "Segway robotic mobility platform," in *SPIE Proc. 5609: Mobile Robots XVII*, October 2004.

[5] R. Hollis, "Ballbots," *Scientific American*, pp. 72–78, October 2006.

[6] L. Havasi, "ERROSphere: an equilibrator robot," *Intl. Conf. on Control and Automation*, pp. 971–976, June 27-29 2005.

[7] M. Kumagai and T. Ochiai, "Development of a robot balancing on a ball," *Intl. Conf. on Control, Automation and Systems*, 2008.

[8] N. Getz and J. K. Hedrick, "An internal equilibrium manifold method of tracking for nonlinear nonminimum phase systems," in *in 1995 American Control Conference, (Seattle), American Automatic Control Council*, 1995, pp. 2241–2245.

[9] S. Devasia, D. Chen, and B. Paden, "Nonlinear inversioon-based output tracking," in *IEEE Transactions on Automatic Control*, vol. 41, no. 7, 1996, pp. 930–942.

[10] U. Nagarajan, "Dynamic constraint-based optimal shape trajectory planner for shape-accelerated underactuated balancing systems," in *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, 2010.

[11] U. Nagarajan, A. Mampetta, G. Kantor, and R. Hollis, "State transition, balancing, station keeping, and yaw control for a dynamically stable single spherical wheel mobile robot," *Proc. IEEE Int'l. Conf. on Robotics and Automation*, pp. 998–1003, 2009.

[12] M. W. Spong, "The control of underactuated mechanical systems," in *First International Conference on Mechatronics*, Mexico City, 1994.

[13] G. Oriolo and Y. Nakamura, "Control of mechanical systems with second-order nonholonomic constraints: Underactuated manipulators," in *Proc. IEEE Conf. on Decision and Control*, 1991, pp. 2398–2403.

[14] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Berkeley: CRC Press, 1994.

[15] U. Nagarajan, G. Kantor, and R. Hollis, "Trajectory planning and control of an underactuated dynamically stable single spherical wheeled mobile robot," *Proc. IEEE Int'l. Conf. on Robotics and Automation*, pp. 3743–3748, 2009.