Dynamic Constraint-based Optimal Shape Trajectory Planner for Shape-Accelerated Underactuated Balancing Systems

Umashankar Nagarajan
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Email: umashankar@cmu.edu

Abstract— This paper presents an optimal shape trajectory planner for *shape-accelerated underactuated balancing systems*, which are destabilized by gravitational forces. These systems have unactuated shape variables and fully actuated external variables. They also have the same number of actuated and unactuated degrees of freedom. Their equations of motion result in nonholonomic acceleration/dynamic constraints, which relate the acceleration of external variables to the position, velocity and acceleration of shape variables. This paper describes a procedure to use the dynamic constraints for planning shape trajectories, which when tracked will result in optimal tracking of desired external configuration trajectories. Examples of planned optimal shape trajectories for the 3D ballbot system, which is a 3D omnidirectional wheeled inverted pendulum, are also presented.

I. Introduction

Underactuated mechanical systems are systems with fewer independent control inputs than the degrees of freedom [1]. Examples of underactuated systems can be found in robotic, aerospace, marine and locomotive applications. These systems are underactuated due to reasons like higher-order dynamics (*e.g.*, spacecraft, helicopter, underwater vehicles, flexible robots), actuator failures (*e.g.*, a failed manipulator joint, aircraft engine failure), and need for dynamic locomotion platforms (*e.g.*, Segway [2], ballbots [3]). A set of intentionally designed underactuated systems used in controls research include Acrobot, Pendubot, Cart-Pole, Beam-Ball and Rotating Pendulum [4].

In robotics, balancing (dynamically stable) mobile robots form a special class of underactuated systems. They include wheeled robots like Segway [2], ballbots [3] and legged robots like BigDog [5], MABEL [6]. Balancing robots will play a vital role in realizing the dream of placing robot workers in human environments. Unlike statically stable mobile robots, balancing (dynamically stable) mobile robots can: (*i*) be tall and skinny with high centers of gravity, (*ii*) have smaller footprints, and (*iii*) accelerate or decelerate quickly [3]. They can also be very effective mobile manipulators [7] with the ability to maintain postural stability, generate forces on external objects and withstand greater impact forces. These characteristics make them ideal for navigation and operation in cluttered human environments. Two-wheeled balancing robots, like Segway, have kinematic constraints restricting their direction of motion, whereas, single spherical-wheeled balancing robots, like ballbots, have omnidirectional motion that make them more suitable for operation in constrained spaces. In this paper, such omnidirectional balancing systems are referred to as *shape-accelerated underactuated balancing systems* (described in Sec. II-B).

An interesting and troubling factor in control and planning of such underactuated systems is the constraint on their dynamics by virtue of underactuation. These constraints are second-order nonholonomic [8] constraints, *i.e.*, non-integrable acceleration/dynamic constraints. These constraints restrict the family of trajectories that the configurations can follow. In control of balancing underactuated systems, which are destabilized by gravitational forces, it is important to maintain balance. Maintaining balance makes it difficult to accurately track any desired configuration trajectories.

In this paper, we propose a planning procedure to ensure good approximate tracking of desired trajectories for such underactuated balancing systems. For example, consider a 3D omnidirectional wheeled inverted pendulum robot like the ballbot [9]. We would like it to track some path on the floor, which could be a desired path from some global motion planning algorithm. The question to be answered here is: *how can the wheel/ball be moved along this path while balancing the pendulum?* The answer to this question is not trivial because of the dynamic coupling between the motion of the ball and the pendulum. Now, consider a balancing controller, as in [9], which ensures accurate tracking of desired lean trajectories for the pendulum, then the question, stated above, can be reformulated as: *how can the lean trajectories of the pendulum be chosen so that the ball moves along the desired path?* This is the question that is addressed in this paper.

In order to answer the above question, we should look into the nonlinear dynamics of the system. Given an accurate model, there have been a variety of controllers designed in nonlinear control literature, which use partial feedback linearization [10], to ensure approximate output tracking [11], [12]. But in real robots, we have to deal with model uncertainties, higher-order dynamics, nonlinear friction effects

and disturbances. Controllers that cancel out nonlinear terms, assuming their accurate knowledge, can fail miserably on real robots. Though the model in-hand may not be accurate, it can often provide some useful information such as the dynamic constraints. In [13], an offline trajectory planning procedure that produces a class of parametric trajectories for the unactuated shape variables to reach static desired configurations using the dynamic constraint was presented. This planning procedure combined with 2-DOF control, an inner balancing control loop and an outer tracking control loop [9], proved to be very successful on the real robot. But, there are a few drawbacks with this approach: (i) it was not easily generalizable to arbitrary motions; (ii) it was computationally expensive; and (iii) it was an offline procedure.

In this paper, we present a generalized approach to generate shape trajectories (*e.g.*, lean angle trajectories in the ballbot), exploiting the structure of dynamic constraints for shape-accelerated underactuated balancing systems, which when tracked will result in optimal tracking of the desired motion. The advantages of this planning procedure are: (i) it can track any desired trajectories satisfying certain conditions (discussed in Sec. III-B); (ii) it is computationally less expensive; and (iii) it is fast and can be performed online on the real robot.

This paper is organized as follows: Sec. II presents the properties of shape-accelerated underactuated balancing systems; Sec. III discusses the special structure of the dynamic constraint for such systems and explains the shape trajectory planning procedure that ensures optimal tracking of desired external configuration trajectories; Sec. IV presents the results of the optimal shape trajectory planner for a 3D ballbot model; and finally, Sec. VI presents the conclusion.

II. UNDERACTUATED MECHANICAL SYSTEMS

The forced Euler-Lagrange equations of motion for a mechanical system are:

$$\frac{d}{dt}\frac{\partial \mathscr{L}}{\partial \dot{q}} - \frac{\partial \mathscr{L}}{\partial q} = F(q)\tau, \quad (1)$$

where, $q \in \mathbb{R}^n$ is the configuration vector, $\mathscr{L}(q, \dot{q}) = K(q, \dot{q}) - V(q)$ is the Lagrangian with kinetic energy K and potential energy V, $\tau \in \mathbb{R}^m$ is the control input and $F(q) \in \mathbb{R}^{n \times m}$ is the force matrix.

A mechanical system satisfying Eq. 1 is said to be an *underactuated system* [1] if $m < n$, *i.e.*, there are fewer independent control inputs than configuration variables.

Eq. 1 for an underactuated system can be written in matrix form as follows:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = F(q)\tau, \quad (2)$$

where, $M(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the matrix of Coriolis and centrifugal terms and $G(q) \in \mathbb{R}^{n \times 1}$ is the vector of gravitational forces.

A. External and Shape 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 variables* (q_x), *i.e.*, $\partial M(q)/\partial q_x = 0$.

Since the inertia matrix is independent of the external variables, the kinetic energy $K(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q}$ is also independent of the external variables, *i.e.*, $\partial K(q, \dot{q})/\partial q_x = 0$. In this case, the Lagrangian system is said to have kinetic symmetry [4].

Eq. 2 can be re-written as:

$$\begin{bmatrix} M_{xx}(q_s) & M_{xs}(q_s) \\ M_{sx}(q_s) & M_{ss}(q_s) \end{bmatrix} \begin{bmatrix} \ddot{q}_x \\ \ddot{q}_s \end{bmatrix} + \begin{bmatrix} h_x(q, \dot{q}) \\ h_s(q, \dot{q}) \end{bmatrix} = \begin{bmatrix} F_x(q) \\ F_s(q) \end{bmatrix}\tau, \quad (3)$$

where, $h(q, \dot{q}) = [h_x(q, \dot{q}), h_s(q, \dot{q})]^T$ is:

$$\begin{bmatrix} h_x(q, \dot{q}) \\ h_s(q, \dot{q}) \end{bmatrix} = \begin{bmatrix} C_{xx}(q, \dot{q}) & C_{xs}(q, \dot{q}) \\ C_{sx}(q, \dot{q}) & C_{ss}(q, \dot{q}) \end{bmatrix} \begin{bmatrix} \dot{q}_x \\ \dot{q}_s \end{bmatrix} + \begin{bmatrix} G_x(q) \\ G_s(q) \end{bmatrix}. \quad (4)$$

The underactuated systems can be classified based on whether the shape variables q_s are fully actuated, partially actuated or unactuated and based on the presence or lack of input couplings in the force matrix $F(q)$ [4].

B. Shape-Accelerated Underactuated Balancing Systems

In this paper, we are interested in shape-accelerated underactuated balancing systems, which form a special class of underactuated systems satisfying the following properties:

(i) The shape variables are unactuated and there is no input coupling, *e.g.*, $F_x(q) = I_m$ and $F_s(q) = 0$. A variety of underactuated systems can be transformed into *no input coupling* form with global change of coordinates [4].

(ii) There are equal number of actuated and unactuated variables, *i.e.*, $\dim(q_x) = \dim(q_s) = m$ and hence $n = 2m$.
In [4], underactuated systems satisfying properties (i)-(ii) are referred to as $Class - IIa$ underactuated systems.

(iii) The potential energy $V(q)$ is independent of q_x, *i.e.*, $\partial V(q)/\partial q_x = 0$. This implies that $G(q) = \partial V(q)/\partial q$ is also independent of q_x. Since both kinetic and potential energies are independent of q_x, the Lagrangian $\mathscr{L}$ is also independent of q_x, *i.e.*, $\mathscr{L}$ is symmetric w.r.t. q_x.

(iv) $M_{xx}(q_s)$ is constant, *i.e.*, $\partial M_{xx}(q_s)/\partial q_s = 0$.

(v) $M_{sx}(q_s)$ has differentially symmetric rows, *i.e.*, $\partial M_{sx}(q_s)/\partial q_s^i = \partial M_{sx}^{i^T}(q_s)/\partial q_s$, where $M_{sx}^i(q_s)$ refers to the ith row of $M_{sx}(q_s)$.
Properties (iii)-(v) make $h(q, \dot{q})$ independent of both q_x and $\dot{q}_x$.

(vi) The system has locally strong inertial coupling [1], *i.e.*, $\text{rank}(M_{sx}(q_s)) = n - m = m$ for all q in the neighborhood of the origin, where origin is the unstable equilibrium. A system satisfying the (local) strong inertial coupling is also known as *Internal/External Convertible System* [12].

(vii) $\partial G_s(q_s)/\partial q_s \neq 0$ at $q_s = 0$ and is invertible.

(viii) $\partial(M_{sx}(q_s)^{-1}G_s(q_s))/\partial q_s \neq 0$ at $q_s = 0$ and is invertible.
The significance of properties (vi)-(viii) will be explained

in Sec. III, where they are used for the optimal shape trajectory planner design.

(ix) $\text{rank}(M_{ss}(q_s)) = m, i.e., M_{ss}(q_s)^{-1}$ exists.

(x) $\partial(M_{ss}(q_s)^{-1}G_s(q_s))/\partial q_s \neq 0$ at $q_s = 0$ and is invertible.

(xi) $M_{ss}(q_s)^{-1}M_{sx}(q_s) \neq 0$ at $q_s = 0$.

Properties (ix)-(xi) ensure that the Jacobian linearization (A, B) of the system (Eq. 3) at origin is controllable and its zero dynamics [10] is unstable at the origin. According to [12], an underactuated system that satisfies properties (ix)-(xi) is called a *balance system* and in this paper, we will refer to it as a *balancing system*.

The shape-accelerated underactuated balancing systems will have equations of motion of the form:

$$\begin{bmatrix} M_{xx} & M_{xs}(q_s) \\ M_{sx}(q_s) & M_{ss}(q_s) \end{bmatrix} \begin{bmatrix} \ddot{q}_x \\ \ddot{q}_s \end{bmatrix} + \begin{bmatrix} h_x(q_s, \dot{q}_s) \\ h_s(q_s, \dot{q}_s) \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix}, \quad (5)$$

where,

$$\begin{bmatrix} h_x(q_s, \dot{q}_s) \\ h_s(q_s, \dot{q}_s) \end{bmatrix} = \begin{bmatrix} 0 & C_{xs}(q_s, \dot{q}_s) \\ 0 & C_{ss}(q_s, \dot{q}_s) \end{bmatrix} \begin{bmatrix} \dot{q}_x \\ \dot{q}_s \end{bmatrix} + \begin{bmatrix} 0 \\ G_s(q_s) \end{bmatrix}. \quad (6)$$

We can see from Eq. 5 that the equations of motion of these systems are such that any non-zero shape configuration will result in acceleration of the external variables and in turn acceleration of the entire system, hence the name *shape-accelerated underactuated balancing systems*.

Some examples of shape-accelerated underactuated balancing systems are planar and 3D cart-pole system with unactuated lean angles, planar wheeled inverted pendulum (e.g. Segway [2] in a plane) and 3D omnidirectional wheeled inverted pendulum (e.g. the ballbot [13, 9]).

III. DYNAMIC CONSTRAINT-BASED OPTIMAL SHAPE TRAJECTORY PLANNER

In this paper, our objective is to plan shape trajectories, which when tracked will result in optimal tracking of desired external configuration trajectories. The planning procedure presented in this section exploits the special structure of dynamic constraints of shape-accelerated underactuated balancing systems described in Sec. II-B. So, let's first look at the dynamic constraint, its structure and the information that it provides.

A. Dynamic Constraint

The second set of m equations of motion associated with the unactuated shape variables in Eq. 5 given by

$$M_{sx}(q_s)\ddot{q}_x + M_{ss}(q_s)\ddot{q}_s + h_s(q_s, \dot{q}_s) = 0 \quad (7)$$

can be written as:

$$\Phi(q_s, \dot{q}_s, \ddot{q}_s, \ddot{q}_x) = 0. \quad (8)$$

Eq. 7 and Eq. 8 are called *second-order nonholonomic constraints*, *nonholonomic acceleration constraints*, or *dynamic constraints* because there exists no function Ψ such that $\ddot{\Psi} = \Phi(q_s, \dot{q}_s, \ddot{q}_s, \ddot{q}_x)$. The dynamic constraint equations are not even partially integrable, *i.e.*, they cannot be converted

into first-order nonholonomic constraints. This is ensured by the fact that the gravitational vector $G(q_s)$ is not a constant and the inertia matrix $M(q_s)$ is dependent on the unactuated shape variables q_s. For a detailed discussion of these conditions, refer to [14].

Due to the properties listed in Sec. II-B, the dynamic constraint equations for shape-accelerated underactuated balancing systems are independent of the position and velocity of external variables. This special structure relates the acceleration of external variables to the position, velocity and acceleration of shape variables. In this section, we will attempt at understanding this relationship.

Let's consider the function

$$\begin{aligned} \Phi'(q_s, \ddot{q}_x) &= \Phi(q_s, 0, 0, \ddot{q}_x) \\ &= M_{sx}(q_s)\ddot{q}_x + h_s(q_s, 0) \\ &= M_{sx}(q_s)\ddot{q}_x + G_s(q_s). \end{aligned} \quad (9)$$

It follows from Eq. 8 that

$$\Phi'(q_s, \ddot{q}_x) = 0. \quad (10)$$

By implicit function theorem, if both $\partial\Phi'/\partial q_s$ and $\partial\Phi'/\partial\ddot{q}_x$ at $(q_s, \ddot{q}_x) = (0, 0)$ exist and are invertible, then there exists an invertible map $\Gamma : q_s \to \ddot{q}_x$ in the neighborhood of the origin such that $\Phi'(q_s, \Gamma(q_s)) = 0$ and $\Phi'(\Gamma^{-1}(\ddot{q}_x), \ddot{q}_x) = 0$. From Eq. 9, we get

$$\left.\frac{\partial\Phi'}{\partial q_s}\right|_{(q_s, \ddot{q}_x)=(0,0)} = \left.\frac{\partial G_s(q_s)}{\partial q_s}\right|_{q_s=0}, \quad (11)$$

$$\left.\frac{\partial\Phi'}{\partial\ddot{q}_x}\right|_{(q_s, \ddot{q}_x)=(0,0)} = M_{sx}(q_s)\Big|_{q_s=0}. \quad (12)$$

From properties (vi) and (vii) in Sec. II-B, we can see that the Jacobians in both Eq. 11 and Eq. 12 exist and are invertible and hence, there exists an invertible map $\Gamma : q_s \to \ddot{q}_x$ in the neighborhood of the origin such that $\Phi'(q_s, \Gamma(q_s)) = 0$ and $\Phi'(\Gamma^{-1}(\ddot{q}_x), \ddot{q}_x) = 0$. Γ can be derived directly from Eq. 9 and Eq. 10 as follows:

$$\begin{aligned} \ddot{q}_x &= -M_{sx}(q_s)^{-1}G_s(q_s) \\ &= \Gamma(q_s) \end{aligned} \quad (13)$$

in the neighborhood of the origin.

Jacobian linearization of Eq. 13 w.r.t. q_s at $q_s = 0$ gives

$$\begin{aligned} \left.\frac{\partial\ddot{q}_x}{\partial q_s}\right|_{q_s=0} &= -\left.\frac{\partial(M_{sx}(q_s)^{-1}G_s(q_s))}{\partial q_s}\right|_{q_s=0} \\ &= K_{q_s}^0. \end{aligned} \quad (14)$$

We know that $K_{q_s}^0$ is invertible by property (viii) in Sec. II-B. This implies, we have a linear map $K_{q_x}^0 = (K_{q_s}^0)^{-1}$ such that

$$q_s = K_{q_x}^0 \ddot{q}_x, \quad (15)$$

and

$$\Phi'(K_{q_x}^0 \ddot{q}_x, \ddot{q}_x) = 0 \quad (16)$$

in the neighborhood of the origin.

We can see that $\ddot{q}_x$ is a constant if and only if q_s is a constant. In order for the external configuration of the system in Eq. 5 to have a constant desired acceleration $\ddot{q}_x^d$, the system should stick to a constant shape configuration given by $q_s^d = K_{q_x}^0 \ddot{q}_x^d$.

When $\ddot{q}_x$ is not constant, $\dot{q}_s$ and $\ddot{q}_s$ are non-zero. Let's take $a = (q_s, \dot{q}_s, \ddot{q}_s)$ and $b = \ddot{q}_x$ and then Eq. 8 can be written as $\Phi(a, b) = 0$. Taking the Jacobian w.r.t. b at $(a, b) = (0, 0)$, we get

$$\left. \frac{\partial \Phi(a, b)}{\partial b} \right|_{(a,b)=(0,0)} = \left. M_{sx}(q_s) \right|_{q_s=0} \qquad (17)$$

From property (vi) in Sec. II-B, we can see that Eq. 17 exists and is invertible. Hence, by implicit function theorem, there exists a map $\Gamma' : a \to b$ such that $\Phi(a, \Gamma'(a)) = 0$. The map Γ' is not invertible since $\partial \Phi(a, b)/\partial a$ at $(a, b) = (0, 0)$ exists but is not invertible.

We can see that in order to track a non-constant, time-varying $\ddot{q}_x^d(t)$, there is no function that maps $\ddot{q}_x^d(t)$ to $(q_s^d(t), \dot{q}_s^d(t), \ddot{q}_s^d(t))$ such that the dynamic constraints in Eq. 8 are satisfied. In the following subsection, we propose an optimal planner that ensures approximate tracking of $\ddot{q}_x^d(t)$.

B. Optimal Shape Trajectory Planner

In underactuated balancing systems, we are often interested in tracking desired trajectories for the external configuration variables without losing balance. Shape-accelerated underactuated balancing systems in Sec. II-B have constraints on the acceleration of these external variables w.r.t. the shape variables' position, velocity and acceleration as given below:

$$\begin{aligned} \ddot{q}_x &= -M_{sx}(q_s)^{-1}(M_{ss}(q_s)\ddot{q}_s + h_s(q_s, \dot{q}_s)) \\ &= \Gamma'(q_s, \dot{q}_s, \ddot{q}_s) \end{aligned} \qquad (18)$$

So, for a desired acceleration trajectory for the external configuration $\ddot{q}_x^d(t)$, we would like to plan shape configuration trajectories $(q_s^p(t), \dot{q}_s^p(t), \ddot{q}_s^p(t))$, which when tracked will result in $\ddot{q}_x^p(t)$ (from Eq. 18) such that $\ddot{q}_x^p(t) = \ddot{q}_x^d(t)$. But, we have seen that $\Gamma'^{-1} : \ddot{q}_x(t) \to (q_s(t), \dot{q}_s(t), \ddot{q}_s(t))$ that satisfies $\Phi(\Gamma'^{-1}(\ddot{q}_x(t)), \ddot{q}_x(t)) = 0$ does not exist. Hence, we propose to find a map $\Omega : \ddot{q}_x^d(t) \to (q_s^p(t), \dot{q}_s^p(t), \ddot{q}_s^p(t))$ such that $\|\ddot{q}_x^p(t) - \ddot{q}_x^d(t)\|_2^2$ is minimized. Here, $\ddot{q}_x^p(t) = \Gamma'(q_s^p(t), \dot{q}_s^p(t), \ddot{q}_s^p(t))$.

Inspired from Eq. 15, we propose to use a linear map $K_{q_x} : \ddot{q}_x^d \to q_s^p$ such that

$$q_s^p(t) = K_{q_x} \ddot{q}_x^d(t), \qquad (19)$$

$$\dot{q}_s^p(t) = K_{q_x} \ddot{\dot{q}}_x^d(t), \qquad (20)$$

$$\ddot{q}_s^p(t) = K_{q_x} \ddot{\ddot{q}}_x^d(t). \qquad (21)$$

The shape trajectory planning procedure can now be formulated as an optimization problem, where the elements of K_{q_x} are determined with the objective of minimizing the function

$$J = \|\ddot{q}_x^p(t) - \ddot{q}_x^d(t)\|_2^2, \qquad (22)$$

where, $\ddot{q}_x^p(t) = \Gamma'(K_{q_x}\ddot{q}_x^d(t), K_{q_x}\ddot{\dot{q}}_x^d(t), K_{q_x}\ddot{\ddot{q}}_x^d(t))$. It is to be noted that the parameter space is m^2-dimensional and any optimization algorithm that solves a nonlinear least-squares problem can be used.

For a desired constant acceleration trajectory, $K_{q_x} = K_{q_x}^0$ ensures optimality. For any general $\ddot{q}_x^d(t)$, $K_{q_x} = K_{q_x}^0$ may not necessarily ensure optimality but will act as a good initial guess for the optimization process.

As one can see, the optimal shape trajectory planner described above ensures that $\ddot{q}_x^p(t)$ approximately tracks $\ddot{q}_x^d(t)$ but does not ensure that $q_x^p(t)$ approximately tracks $q_x^d(t)$ or $\dot{q}_x^p(t)$ approximately tracks $\dot{q}_x^d(t)$. This can be ensured only if the initial conditions for the external variables are met, i.e., $q_x^p(0) = q_x^d(0)$ and $\dot{q}_x^p(0) = \dot{q}_x^d(0)$.

Given below are the conditions to be met in order to use the shape trajectory planner described above to track $q_x^d(t)$:

(i) $q_x^d(t)$ must at least be of *class* C^2, i.e., $\dot{q}_x^d(t)$ and $\ddot{q}_x^d(t)$ exist and are continuous. If $\ddot{q}_x^d(t)$ does not exist then, the planned shaped trajectory $q_s^p(t)$ that is proportional to $\ddot{q}_x^d(t)$ will not exist as well.

(ii) $q_x^d(t)$ is preferred to be of *class* C^4, i.e., first four derivatives exist and are continuous so as to ensure the existence of $q_s^p(t), \dot{q}_s^p(t), \ddot{q}_s^p(t)$ that depend on them. This condition also avoids discontinuities in the planned shape trajectories and its first two derivatives.

(iii) Initial conditions for the external variables are met, i.e., $q_x^p(0) = q_x^d(0)$ and $\dot{q}_x^p(0) = \dot{q}_x^d(0)$. The desired position trajectory $q_x^d(t)$ can be tracked by approximate tracking of $\ddot{q}_x^d(t)$ only if the system starts at the correct initial position and velocity for the external variables.

IV. EXAMPLE: 3D BALLBOT MODEL

As an example, we present here the results of the optimal shape trajectory planner described in Sec. III-B for a 3D ballbot model, which is a 3D omnidirectional wheeled inverted pendulum. The ballbot (Fig. 1(a)) is modeled as a rigid cylindrical body on top of a rigid spherical wheel/ball with the following assumptions: (i) there is no slip between the ball and the floor, and (ii) there is no yaw/spinning motion for both the body and the ball, i.e., they have two degrees-of-freedom each. It is to be noted that the results presented here are that of simulation and are not experimental results.

The forced Euler-Lagrange equations of motion are given by Eq. 1 with q_x corresponding to the configuration of the ball/wheel, and q_s corresponding to the configuration of the body (Euler angles). Here, $q_x = [\theta_x, \theta_y]^T$ and $q_s = [\phi_x, \phi_y]^T$. Planar versions of the ball and body configurations are shown in Fig. 1(b).

The ball configurations, θ_x and θ_y, are chosen such that the linear position of the ball/wheel (with radius r) on the XY ground plane is given by $x_w = r\theta_x$ and $y_w = r\theta_y$. This produces input coupling in $F(q)$:

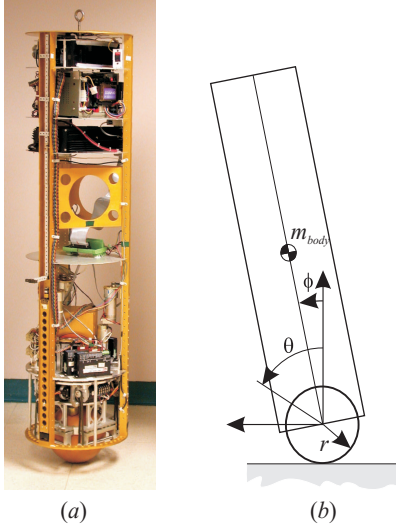$$F(q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix}. \qquad (23)$$

Fig. 1. (a) The ballbot balancing, (b) Planar ballbot model with ball and body configurations shown.

By global change of coordinates $\theta'_x = \theta_x - \phi_y$ and $\theta'_y = \theta_y + \phi_x$, the equations of motion with the new configuration vector $q' = [\theta'_x, \theta'_y, \phi_x, \phi_y]^T$ has no input coupling in $F'(q')$:

$$F'(q') = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \qquad (24)$$

The new forced Euler-Lagrange equations are:

$$M'(q')\ddot{q}' + C'(q', \dot{q}')\dot{q}' + G'(q') = F'(q')\tau. \qquad (25)$$

It is to be noted that the underactuated system in Eq. 25 satisfies all the properties of shape-accelerated underactuated systems and balancing systems listed in Sec. II-B. The expressions for M', C', G' are omitted here due to lack of space.

The last two equations of motion in Eq. 25 form the dynamic constraint equations of the system and these equations are used for the optimal shape trajectory planner in Sec. III-B. The optimization algorithm used here is Levenberg-Marquardt algorithm (LMA), which is a widely used tool for minimization problems in least-squares curve fitting and non-linear programming. The dynamic equations were simulated in MATLAB and the optimization was implemented using MATLAB's $lsqnonlin$ function. Some of the planning results are presented below.

A. Results of Optimal Shape Trajectory Planning

This section presents a variety of desired $x_w(t)$ and $y_w(t)$ that satisfy the conditions in Sec. III-B and the corresponding planned optimal shape trajectories, $\phi^p_x(t)$ and $\phi^p_y(t)$, which should be tracked to achieve them. It is important to note that the desired trajectories, $x_w(t)$ and $y_w(t)$, are trajectories of the center of the ball and not trajectories of the system's center of gravity. On a flat floor, these trajectories match the trajectories of the ball's contact point with the floor.

1) Straight Line Motion: Let's start with the simplest of trajectories that involve moving along a straight line between static configurations, $i.e.$, starting from rest at one point on the floor and coming to rest at another point on the floor. The desired $x_w(t)$ and $y_w(t)$ are chosen to be nonic (9th degree) polynomials in t so that their first four derivatives satisfy the boundary conditions.
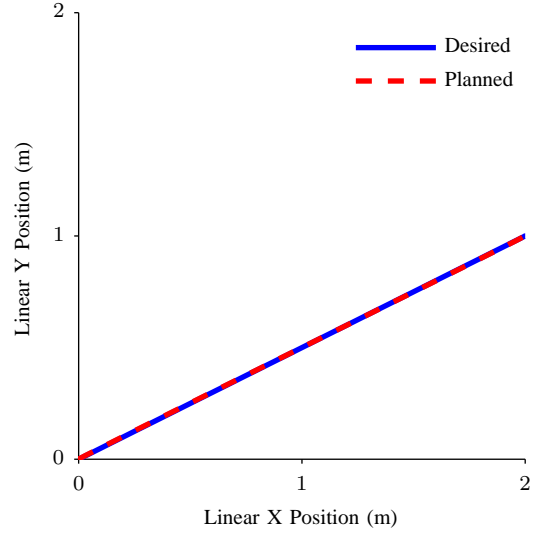

Fig. 2. Straight Line Motion - Linear XY (This figure is best viewed in color.)

$$x_w(t) = \sum_{i=0}^{9} a_i t^i,$$
$$y_w(t) = \sum_{i=0}^{9} b_i t^i, \qquad (26)$$

where the coefficients a_is and b_is are determined based on the initial and final desired configurations.
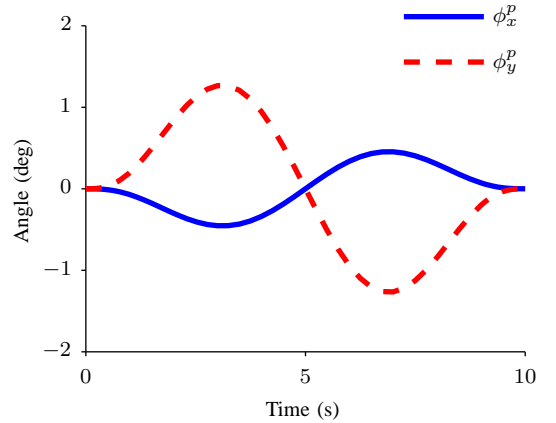

Fig. 3. Straight Line Motion - Planned Shape Trajectories

In Fig. 2, we can see the planned XY motion approximately tracking the desired XY motion. The planned shape trajectories that produce such a motion are shown in Fig. 3. The trajectories for the planned XY motion are obtained from Eq. 18. The tracking error statistics are: RMSE = 0.0147 m and Maximum Error = 0.0238 m.
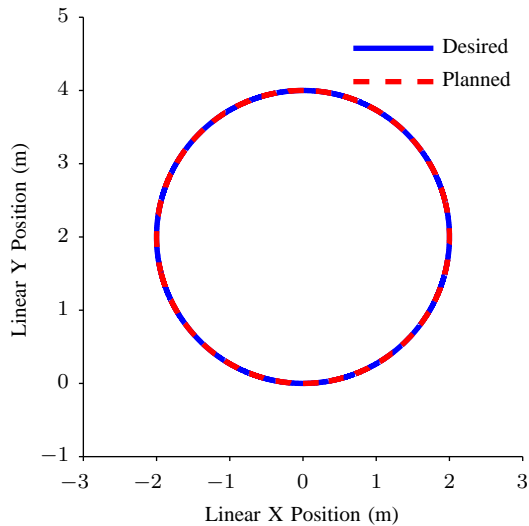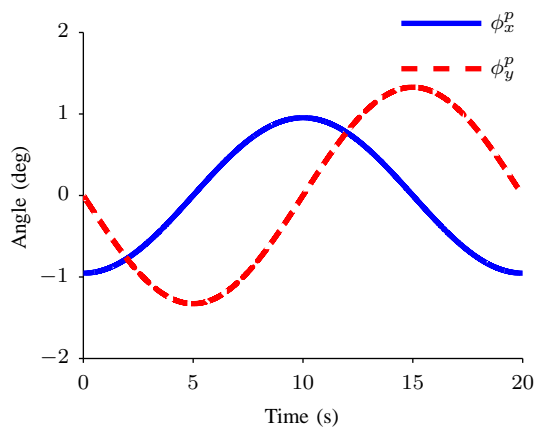

Fig. 4. Circular Motion - Linear XY (This figure is best viewed in color.)

2) Circular Motion: Here, we would like the robot to move along a circular path on the floor. The desired $x_w(t)$ and $y_w(t)$ are:

$$x_w(t) = R\sin(\omega t),$$
$$y_w(t) = R(1 - \cos(\omega t)), \quad (27)$$

where, $R = 2$ m and $\omega = \pi/10$ rad/s.

Fig. 5. Circular Motion - Planned Shape Trajectories

In Fig. 4, we can see the planned XY motion approximately tracking the desired XY motion. The planned shape trajectories that produce such a motion are shown in Fig. 5. The trajectories for the planned XY motion are obtained from Eq. 18. It is

to be noted that the results presented here are observed only when the appropriate initial conditions are met. The tracking error statistics are: RMSE = 4.3x10^{-4} m and Maximum Error = 5.6x10^{-4} m.
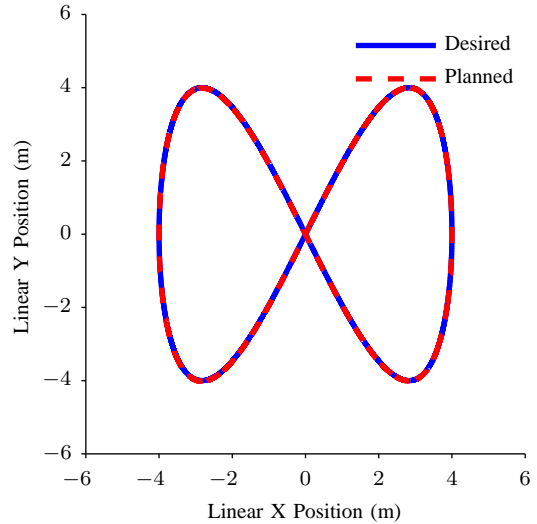

Fig. 6. Figure-8 Motion - Linear XY (This figure is best viewed in color.)

3) Figure-8 Path: Here, we would like the robot to move along a figure-8 path on the floor. The desired $x_w(t)$ and $y_w(t)$ are:

$$x_w(t) = A_x \sin(\omega_x t),$$
$$y_w(t) = A_y \sin(\omega_y t), \quad (28)$$

where, $A_x = 4$ m, $\omega_x = \pi/20$ rad/s, $A_y = 4$ m, and $\omega_y = \pi/10$ rad/s.
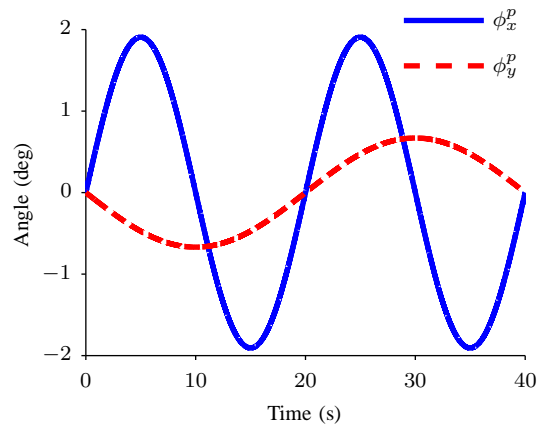

Fig. 7. Figure-8 Motion - Planned Shape Trajectories

In Fig. 6, we can see the planned XY motion approximately tracking the desired XY motion. The planned shape trajectories that produce such a motion are shown in Fig. 7. It is to be noted that the results presented here are observed only when the appropriate initial conditions are met. The trajectories for

the planned XY motion are obtained from Eq. 18. The tracking error statistics are: RMSE = 0.0105 m and Maximum Error = 0.0177 m.
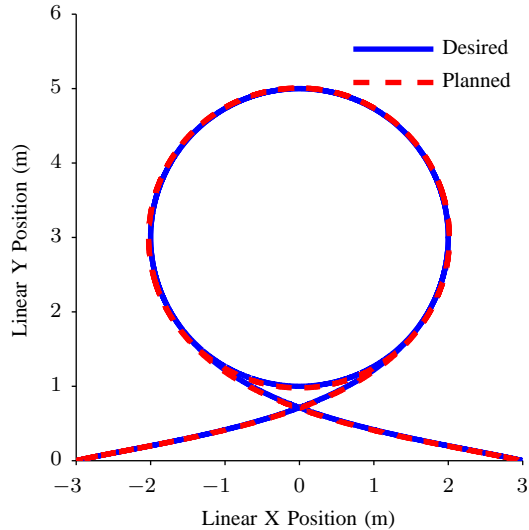

Fig. 8.　Join-Circle-Leave Motion - Linear XY (This figure is best viewed in color.)

4) Join-Circle-Leave Motion: The tracking of a circle, sine wave and figure-8 paths require that the appropriate initial conditions are met. But in reality, we would like to start at any random initial configuration and track a particular path on the floor. Here, we present the result of starting from rest, joining a circular path at a desired configuration, making 1.5 revolutions, leaving the circular path at a diametrically opposite point and coming to rest. This motion consists of 3 different trajectories fused together. The *joining* and *leaving* trajectories are nonic (9th degree) polynomials that satisfy the initial and final configurations required to join and leave the circular path respectively. The nonic polynomials ensure that the first four derivatives of the fused trajectories are continuous and not just piecewise continuous.
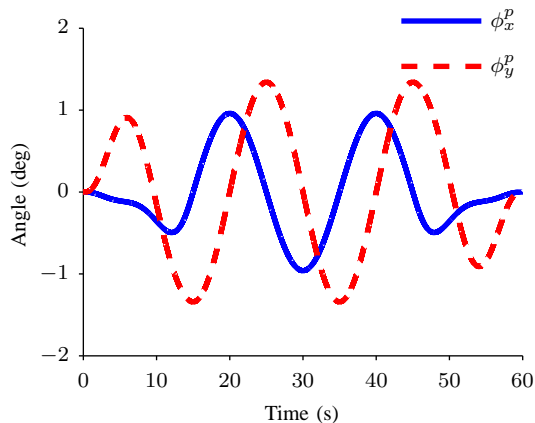

Fig. 9.　Join-Circle-Leave Motion - Planned Shape Trajectories

In Fig. 8, we can see the planned XY motion approximately tracking the desired XY motion. The planned shape trajectories that produce such a motion are shown in Fig. 9. The trajectories for the planned XY motion are obtained from Eq. 18. The tracking error statistics are: RMSE = 0.0167 m and Maximum Error = 0.0286 m.

B. Real-Time Planning

For all the results presented in Sec. IV-A, the optimization tolerance values for both the residual norm and parameter values were set to $< 10^{-4}$. On a standard Intel Core-2 Duo processor, the optimization implementation in MATLAB converges in < 1 second. This ensures that the optimal shape trajectory planner presented in this paper can be used for real-time planning on the robot.

V. Balancing and Tracking Control

The entire planning procedure presented in this paper assumes that there exists a balancing controller, similar to the one in [9], which has good shape trajectory tracking performance. Given the balancing controller and the optimal shape trajectory planner, we can achieve good approximate tracking of the desired external configuration trajectories. But it is to be noted that this tracking is open-loop and with wrong initial conditions, we have no way to ensure approximate tracking of the desired external configuration trajectories. Moreover, while testing on the real robot, modeling uncertainties, unmodeled dynamics, nonlinear friction effects and noise may prevent good approximate tracking. Hence, we should have an external trajectory tracking controller, similar to the one in [13], to ensure better tracking.
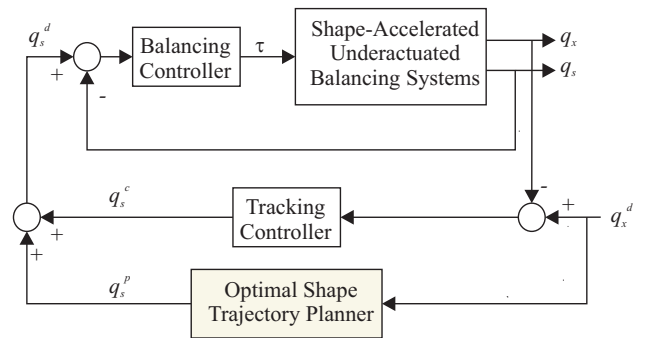

Fig. 10.　Control Architecture

We propose to follow [13] in using the control architecture shown in Fig. 10 for good approximate tracking of desired external configuration trajectories (q_x^d) on the real robot. The balancing controller tracks the desired shape trajectories (q_s^d), which are a sum of planned (q_s^p) and compensation (q_s^c) shape trajectories. The planned shape trajectories are given by the optimal shape trajectory planner, whereas, the compensation shape trajectories are provided by the tracking controller, which tries to compensate for the deviation of the external configuration trajectories from the desired ones.

In Fig. 11, we show the result of the control architecture in Fig. 10 tracking the external configuration trajectories corresponding to a circular motion starting with zero initial conditions, *i.e.*, there is no initial linear velocity in the x direction, as in Fig. 4. We can see that the motion asymptotically converges to the circular motion, which shows the effectiveness of the tracking controller in compensating for the errors due to wrong initial conditions.
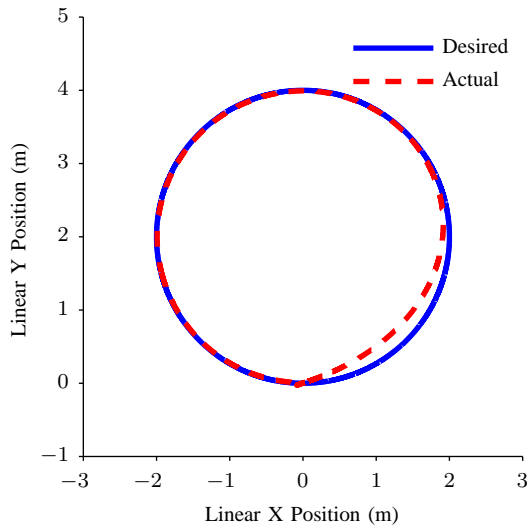

Fig. 11. Tracking Circular Motion with Zero Initial Conditions (This figure is best viewed in color.)

The contribution of this paper is the optimal shape trajectory planner, whereas, the balancing and tracking controller designs are borrowed from our previous work [13]. In [13], we have shown that the control architecture in Fig. 10 is capable of successfully tracking straight line motions on the real robot. Based on these previous successful experimental results, we strongly believe that this control architecture will enable the real robot to track the motions presented in Sec. IV-A. It is to be noted that the planner in [13] was designed based on our intuition of how the robot should lean in order to achieve a straight line motion, whereas, in this paper, we have presented a generalized shape trajectory planner that will enable the ballbot to approximately track any desired external configuration trajectory.

VI. CONCLUSION

An algorithm to plan optimal shape trajectories for shape-accelerated underactuated balancing systems was presented. These shape trajectories, which when tracked will result in approximate tracking of the desired external configuration trajectories subject to conditions on differentiability of the desired trajectory and correct initial conditions. The planning procedure presented assumes that there exists a balancing controller that can accurately track desired shape trajectories. A feedback trajectory tracking controller used in combination

with the optimal shape planner can ensure better tracking when the initial conditions are not met.

In the 3D ballbot example, we have seen that given $(x_w^d(t), y_w^d(t)) \in C^4$, we can use the optimal shape trajectory planner to obtain the desired shape trajectories, which when tracked will result in the desired motion. In a general motion planning setting, any standard path planning algorithm like A^* can be used to obtain a desired $(x_w(s), y_w(s))$ path. In order to use the optimal shape trajectory planner discussed in this paper as a tool for overall motion planning, we have to parametrize this path in time. We propose that using *nonic splines*, which ensure that the first four derivatives for the overall trajectory will be continuous throughout its domain, will help us achieve this goal. Future work will include experimental testing of the planned shape trajectories presented here and design of a global motion planner that provides desired external configuration trajectories for the optimal shape trajectory planner.

REFERENCES

[1] M. W. Spong, "The control of underactuated mechanical systems," in *First International Conference on Mechatronics*, Mexico City, 1994.
[2] 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*, Philadelphia, PA, October 2004.
[3] R. Hollis, "Ballbots," *Scientific American*, pp. 72–78, October 2006.
[4] R. Olfati-Saber, "Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles," Ph.D. dissertation, Massachusetts Institute of Technology, February 2001.
[5] M. Raibert, K. Blankespoor, G. Nelson, R. Playter, and the Big-Dog Team, "Bigdog, the rough-terrain quadraped robot," in *In Proc. 17th World Congress of the International Federation of Automatic Control*, 2008, pp. 10 822–10 825.
[6] J. Grizzle, J. Hurst, B. Morris, H.-W. Park, and K. Sreenath, "Mabel, a new robotic bipedal walker and runner," in *American Control Conference*, St. Louis, MO, June 2009.
[7] 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*, August 2006.
[8] J. R. Ray, "Nonholonomic constraints," *American Journal of Physics*, vol. 34, pp. 406–408, 1966.
[9] 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*, May 2009.
[10] A. Isidori, *Nonlinear Control Systems*. Berlin: Springer-Verlag, 1989.
[11] A. Isidori and C. I. Byrnes, "Output regulation of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 35, no. 2, pp. 131–140, 1990.
[12] N. H. Getz, "Dynamic inversion of nonlinear maps with applications to nonlinear control and robotics," Ph.D. dissertation, University of California at Berkeley, December 1995.
[13] 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*, May 2009.
[14] G. Oriolo and Y. Nakamura, "Control of mechanical systems with second-order nonholonomic constraints: Underactuated manipulators," in *In Proc. 30th IEEE Conference on Decision and Control*, vol. 3, 1991, pp. 2398–2403.