A Brachiating Robot Controller

Jun Nakanishi[1]
Department of Micro System Engineering
Nagoya University
Nagoya, Aichi 464-8603 Japan

Toshio Fukuda
Center for Cooperative Research in Advanced Science and Technology
Nagoya Universiry
Nagoya, Aichi 464-8603 Japan

Daniel E. Koditschek [2]
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2110 USA

The University of Michigan Technical Report: CSE-TR-405-99

August, 1999

[1] The first author was at the Department of Electrical Engineering and Computer Science, the University of Michigan from September, 1995 to August, 1996 and summer in 1997, 1998 and 1999.

[2] This work was supported in part by the US National Science Foundation under grant IRI-9510673.

Abstract

We report on our empirical studies of a new controller for a two-link brachiating robot. Motivated by the pendulum-like motion of an ape's brachiation, we encode this task as the output of a "target dynamical system." Numerical simulations indicate that the resulting controller solves a number of brachiation problems that we term the "ladder", "swing up" and "rope" problems. Preliminary analysis provides some explanation for this success. The proposed controller is implemented on a physical system in our laboratory. The robot achieves behaviors including "swing locomotion" and "swing up" and is capable of continuous locomotion over several rungs of a ladder. We discuss a number of formal questions whose answers will be required to gain a full understanding of the strengths and weaknesses of this approach.

Contents

1 Introduction

This paper presents our effort to develop a new controller for a two degree of freedom brachiating robot. A brachiating robot dynamically moves from handhold to handhold like a long armed ape swinging its arms as depicted in Figure 1. This paper concerns a simplified two-link robot with one actuator at the elbow connecting two arms, each of which has a gripper (see Figure 2). Since the grippers cannot impose torque on the handhold, this is an underactuated machine, having fewer actuators than its degrees of freedom. Designing a brachiating controller for such a system is challenging since the theory of underactuated mechanisms is not well established.

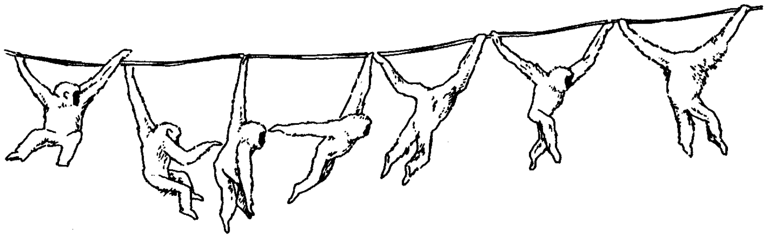

Figure 1: Brachiation of a gibbon: a picture taken from [1]

A growing number of robotics researchers have taken an interest in building machines that are required to interact dynamically with an otherwise unactuated environment in order to achieve a designated task [2]. Brachiating robots take an interesting place within this larger category of robots that juggle, bat, catch, hop and walk. A brachiating and a legged locomotion system share the requirement of an oscillatory exchange of kinetic energy and potential energy in the gravitational field. Brachiation incurs the added problem of dexterous grasps: fumbles not only fail the task but incur a potentially disastrous fall as well.

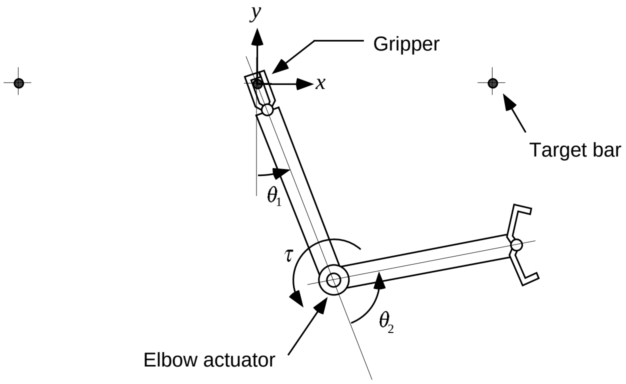

Figure 2: A two-link brachiating robot

1.1 Related work

We now review the relationship of this problem domain to the overlapping areas of dexterous manipulation, legged locomotion and underactuated mechanisms.

Problems of dexterous manipulation have given rise to a growing literature concerned with explicit manipulation of an environment's kinetic as well as potential energy. Arguably, the first great success in this domain must be attributed to Andersson [3] whose ping pong playing robot developed a decade ago was capable of beating many humans. Subsequently, the third author and colleagues [4, 5, 6] developed a family of juggling robots that exhibit increasingly sophisticated strategic as well as mechanical skills in various "games against nature." More recently, Mason and Lynch [7] have studied the problem of dynamic underactuated nonprehensile manipulation from the control theoretic point of view and have successfully implemented a family of formally designed control laws on a one degree of freedom robot which performs dynamic tasks such as snatching, rolling, throwing and catching. Of these antecedents, the present study is most reminiscent of the juggling work since our approach to control entails feedback regulation rather than the open loop pre-planned trajectories developed by Mason and Lynch or the AI system developed by Andersson.

Raibert's landmark success in legged locomotion [8] represents another important influence on the present work. The third author and his colleagues have pursued a number of analytical studies of simple hopping machines that are directly inspired by his work addressing such questions as regulation of hopping height [9], forward velocity [10] and duty factor [11]. Burdick and his colleagues have also investigated numerically the periodic motion of Raibert style hopping robots [12, 13]. The formulation of this brachiation problem in terms of a target dynamics owes much to Raibert's original notion that dynamical dexterity may be encoded in terms of desired energy and achieved with the help of the environment's intrinsic dynamics. Moreover, we have adapted his use of a reverse time symmetry to our problem setting.

Amidst the large and growing controls literature on underactuated mechanisms, this work is closest in method to Spong and his colleagues' studies of the "Acrobot" [14]. They considered the swing up problem of an underactuated system similar to the two-link brachiating robot we treat in this paper. Their control algorithm pumps energy to the system in an instance of Spong's more general notion of partial feedback linearization [15] directed toward achieving a kind of target dynamics whose motions solve the swing up problem. The controller we introduce here bears many similarities to this although the more extended problems of slow brachiation require a rather differently conceived notion of target dynamics, and we are swinging up to an (unstabilizable) handhold (refer to footnote 2 on page 13) — not the vertical equilibrium position as is common in much of the related literature. Thus, equilibrium motions (i.e. hybrid limit cycles) rather than equilibrium points are the regulated goal sets in our problem.

Finally, the second author in collaboration with Saito [16] first introduced to the robotics literature the brachiation control problem. They built the physical two-link brachiating robot we use in this study and experimentally demonstrated the validity of the learned control law, with generalization to a 12 dof mechanism as well [17].

Their view of this problem explored how dynamic behaviors such as brachiation can be generated for robots in a manner analogous to the way humans and animals learn by heuristics. The advantage of their method is that dynamical parameters are not needed (known kinematics and measurements of joint angle/velocity are necessary). However, the method requires a long training period (about 200 experiment with the physical robot) to generate a motion for each configuration of the robot and given distance between the branches.

In contrast, the approach presented in this paper using the notion of target dynamics for a known model is quite different. Although our algorithm requires knowledge of the exact robot's dynamical and kinematic parameters, a single parameter in the "target dynamics" controller suffices to characterize the full range of gaits of brachiation and forward velocities.

2 Problem Setup

2.1 Physical Apparatus

Figure 3 depicts the configuration of our experimental setup. The length of each arm is 0.5m and the total weight of the robot is about 4.8kg.

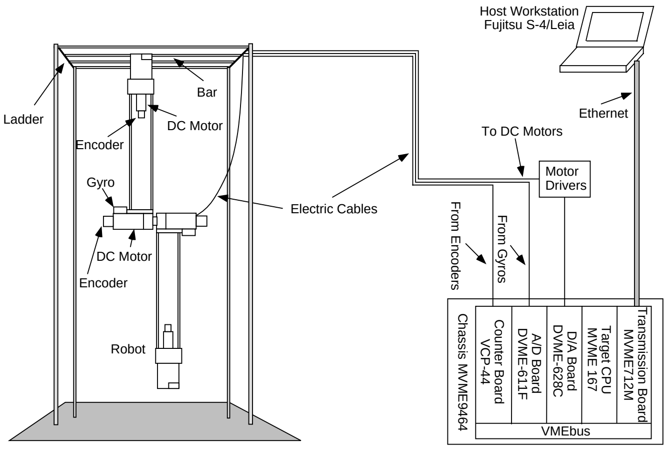

Figure 3: The experimental setup of the two-link brachiating robot.

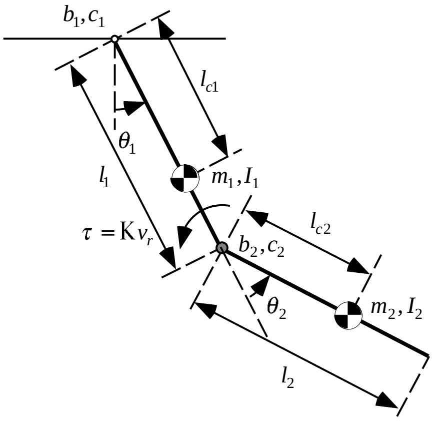

Figure 4: The mechanical model of the two-link brachiating robot used in this paper.

In Saito's original version of this experimental setup, a personal computer equipped with I/O devices was used to control the robot. We have replaced it with a VME bus board computer, MVME 167 (Motorola, CPU MC68040, 33MHz), with a real-time operating system, VxWorks 5.1 and VME bus based I/O devices. The control law is evaluated exactly at a rate of 500Hz.

The elbow joint is actuated by two DC motors with harmonic gears (Harmonic Drive Systems, RH-14-6002). The stator of each motor is fixed to a link, and their rotor shafts are directly connected to each other. As a consequence, we can achieve a total rotational speed at the elbow which is two times faster than the case where there is only one motor. This was necessary since the rated rotational speed of these motors is 360 deg/sec, while we require that the rotational speed of the elbow be grater than 600 deg/sec. An additional benefit of the symmetrical structure of this design is better overall balance in the mechanism. Each gripper is equipped with a DC motor which opens and closes it.

The angle of the first joint is measured by integrating its angular velocity, which is in turn obtained through a gyro (Murata, ENV-05S) attached to the arm. The angle of the second joint and the opening angle of the gripper are measured using optical encoders.

2.2 Model

The dynamical equations used to model the robot take the form of a standard two-link planar manipulator as depicted in Figure 4.

$$\dot{Tq} = \mathcal{L}(Tq, v_r) = \begin{bmatrix} \dot{q} \\ M(q)^{-1}\left(-V(q,\dot{q}) - k(q) - B\dot{q} - C\mathrm{sgn}(\dot{q}) + \begin{bmatrix} 0 \\ Kv_r \end{bmatrix}\right) \end{bmatrix} \tag{1}$$

where $q = [\ \theta_1,\ \theta_2\ ]^T \in \mathcal{Q}$, $Tq = [\ q^T,\ \dot{q}^T\ ]^T \in T\mathcal{Q}$ [1] , M is the inertia matrix, V is the Coriolis/centrifugal vector, k is the gravity vector. B and C denote the viscous and coulomb friction coefficient matrices respectively. We assume that the elbow actuator produces torque proportional to a voltage command, v_r, sent to a driver as $\tau = Kv_r$, where K is a positive constant. The details of the dynamics are presented in Appendix A.

It is generally known that DC motors with harmonic gear mechanisms bear complicated nonlinear characteristics, which are difficult to model [19]. However, for simplicity, we model the dynamics using only viscous and coulomb friction as well as rotor inertia. As the results of parameter identification presented in Appendix B suggest, the model we offer here fits the dynamics of the physical system fairly well. In the paper, we use a lossless version of the model ($B, C \to 0$ in (1)) for the development of the controller and its analysis, but introduce the losses in simulation.

2.3 Problem Statement

Brachiation—arboreal locomotion via arms swinging hand over hand through the trees—is a form of locomotion unique to apes. Most commonly, the animals engage in "slow brachiation," travelling at about the speed of the average human walk. But when excited or frightened, apes can plunge through the forest canopy at astonishing speeds, sometimes covering 30 feet or more in a single jump

[1] Throughout the paper, we shall use the tangent notation of Abraham and Marsden [18]. If $\mathcal{X}$ is a manifold, then $T_x\mathcal{X}$ denotes the tangent vector space at some $x \in \mathcal{X}$ and $T\mathcal{X} = \bigcup_{x\in\mathcal{X}} T_x\mathcal{X}$ denotes the tangent bundle. Moreover Tx will denote some point in $T_x\mathcal{X}$ and $h : \mathcal{X}$ then $\mathcal{Y} \Rightarrow Th : T\mathcal{X} \to T\mathcal{Y}$ is the derived "tangent map," in coordinates, $Th = (h(x), Dh(x)\dot{x})$ where D denotes the Jacobian.

without a break in "stride" (fast brachiation, ricocheting)[20]. In our reading of the biomechanics literature we distinguish three variants of brachiation that we will refer to in this paper as the

- Ladder and swing up problem
- Rope problem
- Leap problem

The first arises when an ape transfers from one branch to another and controlling the arm position at next capture represents the central task requirement. A robotics version of this problem has been previously introduced to the literature by Saito [16, 17]. They presented the robot we consider here with a set of discrete evenly spaced bars and the requirement to swing up from rest, catch the next bar, and then swing from bar to bar by pumping up energy in a suitable fashion. In our view, this problem seems as much akin to that of throwing and catching as to locomotion.

The second problem arises from brachiation along a continuum of handholds—a branch or a rope—that seems most closely analogous to human walking. Since grasps are afforded at will, the resulting freedom of placement can be exploited to achieve a specified forward rate of progress. This is not possible for a two degree of freedom machine on a ladder whose forward velocity is essentially determined by the distance between the bars and its own kinematics. To our knowledge, all previous work on robot brachiation has addressed only the ladder and swing up problems. We have found no studies concerned with the control of forward velocity. We find the results of the rope problem in Section 5 to be somewhat analogous to Schwind's study on the forward velocity control of simplified hopping robots [10].

The third problem arises in the context of fast brachiation where the next branch is far out of reach and the task cannot be accomplished without a large initial velocity and a significant component of free flight. Solving this problem involves not merely a swing phase but a nonholonomic flight as well where the angular momentum of the system is conserved. Roughly analogous to running quickly through a field of boulders, apes can apparently achieve this movement with great regularity and ease. We consider this a fascinating and challenging problem to be addressed when the previous two simpler problems are better understood.

We propose a control algorithm which is effective for the first two "slow brachiation" problems—i.e. the ladder and swing up, and rope problems—inspired by our reading of the biomechanics literature. Specifically, Preuschoft *et al.* [21] studied the mechanics of ape brachiation and identified a close correspondence between slow brachiation and the motion of a simplified pendulum. Accordingly, we have chosen formally to encode the problem of slow brachiation in terms of the output of a "target dynamical system" —the harmonic oscillator —and this task specification lends a slightly new twist to the traditional view of underactuated mechanisms, as we now discuss. Numerical simulations and preliminary analysis suggest that the proposed controller takes its place within a much larger family of target systems that all solve the ladder, swing up problems as well as the rope problem as defined above. The proposed controller is experimentally implemented on the physical two-link brachiating robot described in Section 2.1. Our experimental success encompasses a number of brachiation tasks starting from a variety of different initial hand positions. We have achieved swing locomotion in the ladder problem, where both hands are initially on the ladder; various swing up behaviors from a suspended posture, where only one hand is initially on the ladder; and repeated locomotion over several rungs, where the robot starts with either one or both hands on the ladder.

3 Task Encoding via Target Dynamics

This section presents our control strategy for a two-link brachiating robot. We view the robot's task to be one of solving an "environmental control problem" [4]. For example, in robot juggling a fully actuated robot controls the motion of a ball (which is the unactuated environment) through intermittent interaction. In this case, interaction between the robot and environment only occurs at the ball-robot impact. In contrast, in robot brachiation, the robot and environment (respectively the actuated and unactuated joints) have continuous interaction during the motion. The difficulty in controlling a brachiating robot arises due to this continuous coupling. Hopping robots might be considered as lying in between since they have continuous interaction with the ground only in the stance phase.

Appropriate task encoding plays an important role in achieving robot dynamical dexterity in dynamical environment. Before proceeding, we mention some previous instances of task encoding based on a good understanding of the intrinsic dynamics of a system and an environment.

The first example of task encoding is in the control of legged locomotion by Raibert[8]. He decomposes the control of legged locomotion into three parts and encodes as:

- Regulation of hopping height: control of the mechanical energy of the system through leg's thrust.
- Control of forward velocity: choice of foot placement at touchdown.
- Control of body posture: servoing the hip during stance.

He implements a simple feedback control law to achieve the desired locomotion according to this task encoding and successfully demonstrated the validity of his control strategy.

The second example is in the robot juggling achieved by the third author *et al.*[4]. Their idea is analogous to that of Raibert's. In order to achieve juggling with the specific apex height of a ball they introduce a "mirror algorithm" by means of which the robot is forced to track the nonlinear reflected mirror trajectory of a ball servoing its mechanical energy around a desired steady state energy level. In these examples, appropriate task encoding achieves such dynamically dexterous behavior as hopping and juggling.

We introduce the notion of "target dynamics" as a particular instance of input/output plant inversion. Specifically, brachiation is encoded as the output of a target dynamical system—a harmonic oscillator, that we must force the robot to mimic. As we have pointed out in Section 1.1, the handhold state we consider, cannot be made to be an equilibrium state under the influence of gravity. Thus, traditional set point stabilizing control schemes are not relevant in the present problem setting.

Under these circumstances, we need to consider some "natural" orbit which achieves the designated task by combining physical insight into the task and the intrinsic dynamics of the system in its environment. We now introduce the notion of task encoding via target dynamics as a means of finding a family of such "natural orbits."

3.1 Input/Output Linearization and Target Dynamics

The notion of target dynamics represents a slight variant on standard techniques of plant inversion. A system is inverted then forced to have the characteristic of a chosen target dynamics. Thus, instead of tracking an exogenously designed reference trajectory, we force the system to generate and track its own reference motion.

Suppose a plant

$$\dot{w} = F(w, v) \tag{2}$$

$$y = H(w) \tag{3}$$

is input/output linearizable. That is, given

$$L_F H(w, v) = DH \cdot F(w, v) \tag{4}$$

if there can be found an implicit function such that for every $u \in \mathcal{U}$ and $w \in \mathcal{W}$, then

$$v = L_F H^{-1}(w, u) \tag{5}$$

implies

$$L_F H(w, v) = u, \tag{6}$$

one calls (5) an input/output linearizing inverse controller in the sense that $\dot{y} = u$.

It is traditional in the underactuated robot control literature to use the linearizing feedback (5) to force y to track some reference trajectory $r_d(t)$. In the present article, we find it more useful to mimic a reference dynamical system,

$$\dot{y} = f(y). \tag{7}$$

This behavior obtains by substituting f for u in (5), yielding the feedback law,

$$v = L_F H^{-1}\left(w, f(y)\right) = L_F H^{-1}\left(w, f \circ H(w)\right). \tag{8}$$

3.2 A Target Dynamics and Its Associated Controller

According to the biomechanics literature [21], slow brachiation of apes resembles the motion of a pendulum. Although the ape's moment of inertia varies during the swing according its change of posture, the motion of a simplified pendulum gives a fairly good approximation. Motivated by this pendulum-like motion , we encode the robot brachiation task in terms of the harmonic oscillator

$$y = Tx = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \; f_\omega(Tx) = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} Tx, \tag{9}$$

where ω, the natural frequency of the virtual pendulum, will play the role of the task level control parameter in the sequel. Supporting this role, the function, f_ω (9), serves as the target dynamical system (7) for all the empirical work reported in this paper.

The choice of output map (3) seems to be much more critical, since it prescribes the combination of states that will be forced to exhibit the selected target dynamics (14). In Figure 5 we illustrate the local change of coordinates from joint space to polar coordinates on $\mathbb{R}^2$,

$$\begin{bmatrix} r \\ \theta \end{bmatrix} = \bar{q} = \bar{g}(q) = \begin{bmatrix} l\sqrt{2(1+\cos\theta_2)} \\ \theta_1 + \frac{1}{2}\theta_2 \end{bmatrix}. \tag{10}$$

Intuitively, pursuing the analogy arising from biomechanical observation [21], the simplest pendulum to be found in the underlying RR kinematic chain obtains from its polar coordinate "angle," θ, motivating the choice,

$$x = h(q) := \theta = [0, 1]\; \bar{g}(q) = \theta_1 + \frac{1}{2}\theta_2. \tag{11}$$

With these choices in place, the controller synthesis is formally complete. In summary, the virtual pendulum angle, θ, is forced to follow the target dynamics, $\ddot{\theta} + \omega^2\theta = 0$. Namely, identify $w = Tq = [\ q^T,\ \dot{q}^T\ ]^T$, $v = \tau$, $F = \mathcal{L}$ in (2) and $H = Th$, and apply the control law formulated in (8) with respect to the target (9):

$$\begin{aligned}\tau = \tau_\omega &:= L_F H^{-1}\left(Tq, f_\omega \circ Th(Tq)\right)\\ &= \left(D_q h \begin{bmatrix} n_{12} \\ n_{22} \end{bmatrix}\right)^{-1} \left[-\omega^2\theta - (\dot{D_q h})\dot{q} + D_q h M^{-1}(V+k)\right]\\ &= \frac{1}{n_{12} + \frac{1}{2}n_{22}} \left[-\omega^2(\theta_1 + \frac{1}{2}\theta_2) + (n_{11} + \frac{1}{2}n_{21})(V_1 + k_1)\right] + V_2 + k_2. \qquad (12)\end{aligned}$$

where

$$M^{-1} = \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{bmatrix}.$$

Notice that

$$D_q h \begin{bmatrix} n_{12} \\ n_{22} \end{bmatrix} = n_{12} + \frac{1}{2}n_{22} = \frac{m_1 l_{c1}^2 + m_2({l_1}^2 - {l_{c2}}^2) + I_1 - I_2}{2\det(M)} \neq 0 \qquad (13)$$

i.e., the invertibility condition of $L_F H$ is satisfied in the particular setting with the parameter values shown in Table 1 in Appendix B.

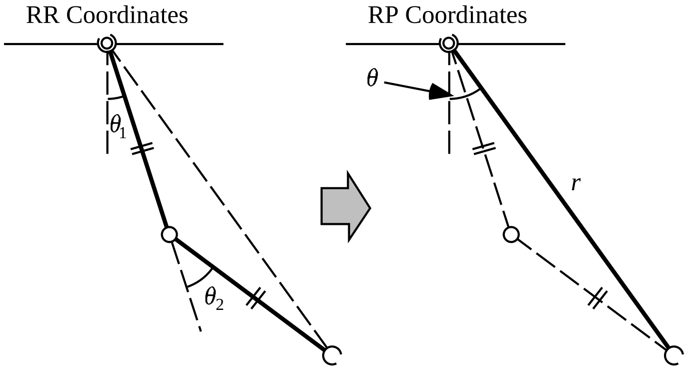

Figure 5: Change of coordinates from RR to RP. We control θ to follow the dynamics $\ddot{\theta} + \omega^2\theta = 0$ using a target dynamics controller.

3.3 A Class of Target Dynamics

In the lab, we use a harmonic oscillator to encode the brachiation task motiveted by the pendulous motion of an ape's brachiation as mentioned above. Here, we explore a class of mechanical oscillators (1 dof lossless Lagrangian systems with a family of potentials) and its feasibility as a target dynamical system. We find it preferable to use a harmonic oscillator whose potential is a Hooke's law spring for the reasons now we discuss.

Consider a class of lossless mechanical oscillator of the form for the target system (7),

$$y = Tx = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, f_\omega(Tx) = \begin{bmatrix} \dot{x} \\ -\omega^2 DU(x) \end{bmatrix} \tag{14}$$

Simulations suggest that any lossless mechanical oscillator (14) can encode brachiation when $U(x)$, an ”artificial potential” function, is even and convex in the region of operation. For future reference, let $\bar{E}$ be the “pseudo” mechanical energy defined by this oscillator,

$$\bar{E} := \frac{1}{2}\dot{x}^2 + \omega^2 U(x). \tag{15}$$

The control law with respect to the target (14) with the output map, $x = h(q)$, is formulated as

$$\begin{aligned} \tau = \tau_\omega &:= L_F H^{-1}\left(Tq, f_\omega \circ Th(Tq)\right) \\ &= \left(D_q h \begin{bmatrix} n_{12} \\ n_{22} \end{bmatrix}\right)^{-1} \left[-\omega^2 D_x U \circ h(q) - (\dot{D_q h})\dot{q} + D_q h M^{-1}(V + k)\right] \\ &= \frac{1}{n_{12} + \frac{1}{2}n_{22}} \left[-\omega^2 D_x U \circ h(q) + (n_{11} + \frac{1}{2}n_{21})(V_1 + k_1)\right] + V_2 + k_2, \end{aligned} \tag{16}$$

In Section 4.1.1 we make the formal observation (Proposition 4.3) that any even potential together with an appropriately “odd” choice of output map (3) will support the Raibert-style reverse time symmetry [8] essential to the efficacy of our task encoding. Indeed, in our numerical investigations in Sections 4.2, 4.4 and 5.4, we have had good experience with many choices for the artificial potential function. In our empirical work, we have found it particularly convenient to adopt the specific Hooke’s Law potential, $U(x) = \frac{1}{2}x^2$, (9), for two reasons.

First, the elliptic integral

$$T_N = 4\int_0^{x_0} \frac{dx}{\sqrt{2(\bar{E}_0 - \omega^2 U(x))}} = \frac{4}{\omega}\int_0^{x_0} \frac{dx}{\sqrt{2(U(x_0) - U(x))}} \tag{17}$$

is solvable in closed form using elementary functions for a Hooke’s law spring. This closed form expression significantly simplifies the computational effort incurred by the root finding procedure of (35) required to tune the “natural frequency,” ω, in the ladder and rope problems.

Second, numerical study addressing the swingup problem in Section 4.4 reveals that the “stiffness” (the second derivative of the potential, U) plays an important role for reasons we do not yet understand well. Specifically, we require not only positive stiffness (i.e., convex potentials, U), but find that some “stiffness margin” profile is key to effective swingup behavior. We plot in Figures 6 and 7 some examples of the potential and its associated stiffness of several spring laws. Generally speaking, “hard” spring laws such as $U(x) = \frac{1}{4}x^4$ or $U(x) = \frac{1}{2}x^2 + \frac{1}{4}x^4$ work very nicely. In contrast, consider the effective torsional spring potential introduced by a gravity loaded simple pendulum, $U(x) = 1 - \cos x$, whose stiffness becomes zero at the boundary of the domain of operation, and a spring law, $U(x) = \frac{1}{2}x^2 - \frac{1}{24}x^4$, whose stiffness becomes negative over the domain of operation. Such “soft” springs (i.e., potentials whose second derivative stiffness functions are not bounded away from zero over the domain of operation) characteristically result in “out of phase” swingups that fail the task. While hardening springs work nicely in simulation, they typically incur unavailably large torques. The Hooke’s Law potential enjoys benefits of positive stiffness and realistic torque requirements.

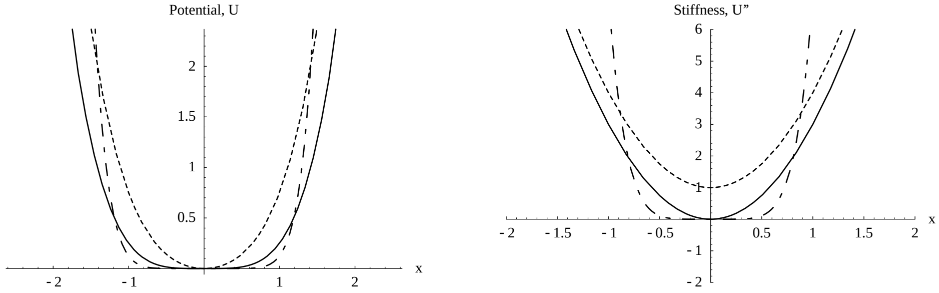

Figure 6: Some examples of the potential and its assotiated stiffness of "hard" spring laws. Solid: $U(x) = \frac{1}{4}x^4$, dashed: $U(x) = \frac{1}{2}x^2 + \frac{1}{4}x^4$, and dot-dashed: $U(x) = \frac{1}{8}x^8$.

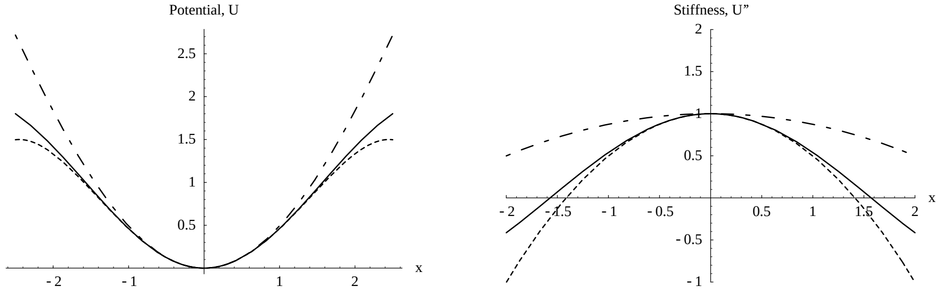

Figure 7: Some examples of the potential and its assotiated stiffness of "soft" spring laws. Solid: $U(x) = 1 - \cos x$, dashed : $U(x) = \frac{1}{2}x^2 - \frac{1}{24}x^4$, and dot-dashed: $U(x) = \frac{1}{2}x^2 - \frac{1}{96}x^4$.

4 Ladder and Swing up Problem

We now move on to the specific problems of robot brachiation. First, we apply the target dynamics method to the ladder problem. Then, we consider the swing up problem. The target dynamics is modified to introduce a limit cycle to achieve the task. Numerical simulations are provided to suggest the effectiveness of the proposed algorithms. In this section, we use the lossless model for the following analysis.

4.1 Ladder problem

As we have pointed out, the ladder problem arises when an ape transfers from one branch to another and the control of arm position at the next capture represents the control task requirement. Here, we restrict our attention to brachiation on a set of evenly spaced bars at the same height. The target dynamics method is applied to the ladder problem. We show how a symmetry property of an appropriately chosen target system— (14) in the present case—can solve this problem.

4.1.1 Neutral Orbits, $\mathcal{N}$

This section follows closely the ideas originally developed in [10, 11]. We discuss a reverse time symmetry inherent in the brachiating robot's dynamics. Here, we are interested in orbits whose forward time motions from the bottom states are a horizontal reflection of their backward time motion from the same initial condition.

First, we show that the natural dynamics of the two-link brachiating robot admit this reverse time symmetry, S. Then, we give a condition under which feedback laws result in closed loops that still admit S. Lastly, following Raibert [8], we introduce the notion of the neutral orbits of the symmetry, and show how they may be used to solve the ladder problem. In the sequel, we will denote the integral curve of a vector field f by the notation f^t.

Definition 4.1 *$f : \mathcal{X} \to T\mathcal{X}$ admits a reverse time symmetry $S : \mathcal{X} \to \mathcal{X}$ if and only if $S \circ f^t = f^{-t} \circ S$.*

Note that when S is linear, after taking time derivatives, this definition might be equivalently stated as $S \circ f = -f \circ S$. In this paper, we are concerned specifically with the symmetry operator

$$S = \begin{bmatrix} -I_{2\times 2} & 0 \\ 0 & I_{2\times 2} \end{bmatrix}. \tag{18}$$

(where $I_{2\times 2}$ denotes the 2×2 identity matrix.) When f admits S in (18), there exist orbits integrated forward in time from some initial conditions which are reflections of orbits backward in time from the same initial conditions odd in angles and even in velocities, i.e., $q(-t) = -q(t)$ and $\dot{q}(-t) = \dot{q}(t)$.

Now, supposing we have chosen a feedback law, $\tau(q, \dot{q})$, denote the closed loop dynamics of the robot as

$$\dot{Tq} = \mathcal{L}_\tau(Tq) = \mathcal{L}\left(Tq, \tau(Tq)\right). \tag{19}$$

Say that τ "admits S" if and only if $\mathcal{L}_\tau$ admits S.

Proposition 4.2 *The closed loop dynamics $\mathcal{L}_\tau$ admits S as in (18), i.e., $S \circ \mathcal{L}_\tau(Tq) = -\mathcal{L}_\tau \circ S(Tq)$ if and only if $\tau(q, \dot{q})$ has the property $\tau(-q, \dot{q}) = -\tau(q, \dot{q})$.*

Proof:

$$\begin{aligned}
\mathcal{L}_\tau \circ S(Tq) &= \begin{bmatrix} \dot{q} \\ M(-q)^{-1}\left(-V(-q,\dot{q}) - k(-q) + \begin{bmatrix} 0 \\ \tau(-q,\dot{q}) \end{bmatrix}\right) \end{bmatrix} \\
&= \begin{bmatrix} \dot{q} \\ M(q)^{-1}\left(V(q,\dot{q}) + k(q) + \begin{bmatrix} 0 \\ \tau(-q,\dot{q}) \end{bmatrix}\right) \end{bmatrix}
\end{aligned} \tag{20}$$

since $M(-q) = M(q), V(-q, \dot{q}) = -V(q, \dot{q}), k(-q) = -k(q)$. On the other hand,

$$S \circ \mathcal{L}_\tau(Tq) = \begin{bmatrix} -\dot{q} \\ M(q)^{-1}\left(-V(q,\dot{q}) - k(q) + \begin{bmatrix} 0 \\ \tau(q,\dot{q}) \end{bmatrix}\right) \end{bmatrix} \tag{21}$$

From (20) and (21) we see that if $\tau(-q, \dot{q}) = -\tau(q, \dot{q})$, then $S \circ \mathcal{L}_\tau(Tq) = -\mathcal{L}_\tau \circ S(Tq)$. On the other hand, if $S \circ \mathcal{L}_\tau(Tq) = -\mathcal{L}_\tau \circ S(Tq)$, $\tau(q, \dot{q})$ has to satisfy the property $\tau(-q, \dot{q}) = -\tau(q, \dot{q})$. ■

Proposition 4.3 *If the artificial potential function $U(x)$ in (14) is even, $U(-x) = U(x)$, and if $x = h(q)$, i.e a smooth scalar valued function, has the property $h(-q) = -h(q)$, then the feedback law (16), τ_ω, admits S.*

Proof:

$$\begin{aligned} \tau(-q,\dot{q}) = & \left(D_q h(-q)\begin{bmatrix} n_{12} \\ n_{22} \end{bmatrix}\right)^{-1}\left\{-\omega^2 D_x U \circ h(-q) - \frac{d}{dt}[D_q h(-q)]\dot{q}\right. \\ & \left. + D_q h(-q) M(-q)^{-1}\left[V(-q,\dot{q}) + k(-q)\right]\right\} \\ = & \left(D_q h(q)\begin{bmatrix} n_{12} \\ n_{22} \end{bmatrix}\right)^{-1}\left\{\omega^2 D_x U \circ h(q) + \frac{d}{dt}[D_q h(q)]\dot{q}\right. \\ & \left. + D_q h(q) M(q)^{-1}\left[-V(q,\dot{q}) - k(q)\right]\right\} \\ = & -\tau(q,\dot{q}), \end{aligned} \tag{22}$$

since

$$M(-q)^{-1} = M(q)^{-1}, V(-q,\dot{q}) = -V(q,\dot{q}), k(-q) = -k(q),$$

and

$$D_x U \circ h(-q) = D_x U \circ (-h(q)) = -D_x U \circ h(q)$$

because $h(-q) = -h(q)$ and $D_x U(-x) = -D_x U(x)$ (the derivative of an even function is an odd function). Furthermore, recalling h is a scalar valued function,

$$\frac{d}{dt}(D_q h) = D_q(D_q h)\frac{d}{dt}q = {D_q}^2 h\dot{q},$$

$$\frac{d}{dt}[D_q h(-q)]\dot{q} = [{D_q}^2 h(-q)]\dot{q}^2 = -[{D_q}^2 h(q)]\dot{q}^2 = -\frac{d}{dt}[D_q h(q)]\dot{q}$$

because $D_q h(-q) = D_q h(q)$ and ${D_q}^2 h(-q) = -{D_q}^2 h(q)$. Since $\tau(-q,\dot{q}) = -\tau(q,\dot{q})$, Proposition 4.2 applies. ■

Define the fixed points of the symmetry S to be

$$\text{Fix}S := \{Tq \in TQ \mid S(Tq) = Tq\} \tag{23}$$

In the present case, i.e., for S in (18) note that

$$\text{Fix}S = \{(q,\dot{q}) \in TQ | q = 0\}$$

Define the set of "neutral orbits" to be the integral curves which go through the fixed point set,

$$\mathcal{N} := \bigcup_{t \in \mathbb{R}} \mathcal{L}^t(\text{Fix}S) \tag{24}$$

Note that a neutral orbit has a symmetry property about its fixed point—namely, if $Tq_0 \in \text{Fix}S$, then

$$S \circ \mathcal{L}^t(Tq_0) = \mathcal{L}^{-t} \circ S(Tq_0) = \mathcal{L}^{-t}(Tq_0)$$

4.1.2 The Ceiling, $\mathcal{C}$, and its Neutral Orbits

Define the "ceiling"

$$\mathcal{C} = \{q \in Q | \cos\theta_1 + \cos(\theta_1 + \theta_2) = 0\}. \tag{25}$$

to be those configurations where the hand of the robot reaches the height $y = 0$ as depicted in Figure 8 [2].

In general, arms which drop from the ceiling, whether under active torque control or not, will not pass through a bottom state — they will not trace out a neutral orbit. Our problem now is to find a virtual frequency, ω, matched to the desired distance, d, that renders this ceiling state neutral.

The ceiling, $\mathcal{C}$, can be parameterized by two branches,

$$\mathcal{C} = \text{Im } c_- \cup \text{Im } c_+, \tag{26}$$

of the maps, $c_\pm : [0, 2l] \to \mathcal{C}$,

$$c_\pm(d) = \begin{bmatrix} \pm \arcsin\left(\frac{d}{2l}\right) \\ \pm \left[\pi - 2\arcsin\left(\frac{d}{2l}\right)\right] \end{bmatrix}. \tag{27}$$

In the sequel, we will be particularly interested in initial conditions of (19) originating in the zero velocity sections of the ceiling that we denote $T\mathcal{C}_0$. Now note that $S(T\mathcal{C}_0) \subseteq T\mathcal{C}_0$ since

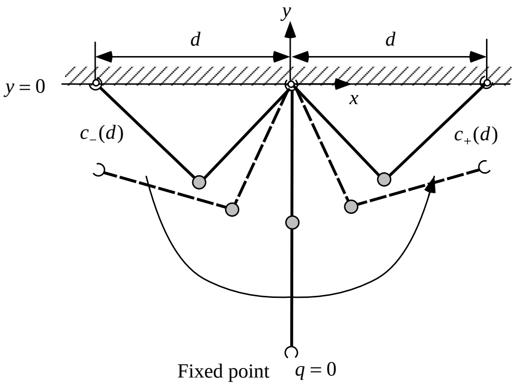

Figure 8: A ceiling configuration. The ceiling is parametrized by the distance between the grippers d. A left branch $c_-(d)$ and right branch $c_+(d)$ are defined in this manner.

$$S \begin{bmatrix} c_-(d) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} c_+(d) \\ 0 \\ 0 \end{bmatrix}. \tag{28}$$

[2] Notice that this handhold state, "ceiling," cannot be made to be an equilibrium state under the influence of the gravity since we cannot find τ such that $\mathcal{L}(Tq, \tau) \equiv 0$ in (1) when $k(q) \neq 0$.

Proposition 4.4 *If a feedback law, τ, admits S and if* $\begin{bmatrix} c_-(d) \\ 0 \\ 0 \end{bmatrix} \in \mathcal{N} \cap T\mathcal{C}_0$, *then there can be found a time* $t_N \in I\!R$ *such that for* $\nu = \frac{t_N}{4}$ *we have*

$$\mathcal{L}_\tau^{2\nu}\left(\begin{bmatrix} c_-(d) \\ 0 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} c_+(d) \\ 0 \\ 0 \end{bmatrix} \tag{29}$$

i.e., a time at which the left branch at zero velocity in the ceiling reaches the right branch in the ceiling also at zero velocity.

Proof: By the definition of $\mathcal{N}$, there can be found a time $\nu \in I\!R$ at which

$$\mathcal{L}_\tau^{\nu}\left(\begin{bmatrix} c_-(d) \\ 0 \\ 0 \end{bmatrix}\right) := Tq^* \in \text{Fix}S \tag{30}$$

Therefore,

$$\mathcal{L}_\tau^{-\nu}(Tq^*) = \begin{bmatrix} c_-(d) \\ 0 \\ 0 \end{bmatrix}. \tag{31}$$

Applying the symmetry S, we have

$$\begin{bmatrix} c_+(d) \\ 0 \\ 0 \end{bmatrix} = S \begin{bmatrix} c_-(d) \\ 0 \\ 0 \end{bmatrix} \qquad \text{(from (28))}. \tag{32}$$

But

$$S \begin{bmatrix} c_-(d) \\ 0 \\ 0 \end{bmatrix} = S \circ \mathcal{L}_\tau^{-\nu}(Tq^*) \qquad \text{(from (31))}, \tag{33}$$

hence,

$$\begin{aligned} \begin{bmatrix} c_+(d) \\ 0 \\ 0 \end{bmatrix} &= S \circ \mathcal{L}_\tau^{-\nu}(Tq^*) = \mathcal{L}_\tau^{\nu} \circ S(Tq^*) \\ &= \mathcal{L}_\tau^{\nu}(Tq^*) = \mathcal{L}_\tau^{\nu} \circ \mathcal{L}_\tau^{\nu}\left(\begin{bmatrix} c_-(d) \\ 0 \\ 0 \end{bmatrix}\right) \\ &= \mathcal{L}_\tau^{2\nu}\left(\begin{bmatrix} c_-(d) \\ 0 \\ 0 \end{bmatrix}\right) \end{aligned} \tag{34}$$

■

Thus, we conclude that any feedback law, τ, which admits S, solves the ladder problem, assuming we can find a d such that $[\ c_-(d)^T,\ 0,\ 0\ ]^T \in \mathcal{N}$. Note that finding such a ceiling point requires solving the equation

$$\Phi(d, t_N) = \left[I_{2\times 2} \ , 0_{2\times 2} \right] \mathcal{L}_\tau^\nu \left(\begin{bmatrix} c_-(d) \\ 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{35}$$

where $\nu = \frac{t_N}{4}$, for d and t_N simultaneously. Of course solving this equation is quite difficult: it requires a two dimensional "root finding" procedure for a function whose evlauation entails integrating the dynamics (1).

4.1.3 Application of Target Dynamics

The feedback law τ_ω (16) arising from the target (9) and ouput map (11) admits S since Proposition 4.3 applies. The special target, (9), enjoys a very nice property relative to the difficult root finding problem (35). Namely, using this control algorithm, t_N is given by

$$t_N(\tau_\omega) = 4 \int_0^{\frac{\pi}{2}} \frac{d\theta}{\sqrt{2(\bar{E}_0 - \frac{1}{2}\omega^2\theta^2)}} = \frac{2\pi}{\omega}. \tag{36}$$

By this assignment, then we have reduced the dimension of the tuning parameter space by half: we need merely solve (35) for d. More formally, we seek an implicit function $d^* = \lambda^{-1}(\omega)$ such that $\Phi\left(\lambda^{-1}(\omega), \frac{2\pi}{\omega}\right) = 0$. Of course, we are more likely in practice to take an interest in tuning ω as a function of a desired d^*. Thus, we are most interested in determining

$$\omega = \lambda(d^*). \tag{37}$$

In general, we can expect no closed form expression for λ or λ^{-1}, and we compute an estimate, $\hat{\lambda}$, using a standard numerical scalar root finding method (i.e. the "false position" method or "secant" method) whose convergence properties are well known [22].

We plot in Figure 9 a particular instance of $\hat{\lambda}$ for the case where the robot parameters are shown in Table 1 in Appendix B. ω is tuned according to this mapping .We will use these parameter values throughout the sequel for the sake of comparison between this and subsequent figures.

4.2 Simulation

4.2.1 Simulation with a Hooke's Law Potential for Target Dynamics

Consider the case $d^* = 0.6$ for this parameter set above. The initial condition of the robot is $Tq_0 = [\, c_-(d^*)^T,\ 0,\ 0\,]^T$. From the numerical solution depicted in Figure 9, $\omega = \hat{\lambda}(0.6) = 3.36$ with the choice of the Hooke's spring law, $U(x) = \frac{1}{2}x^2$. In this simulation, the lossy model (1) is used and friction terms are added in the inverse dynamics controller (16) as follows:

$$\begin{aligned} \tau =: \tau_\omega &= L_F H^{-1}\left(Tq, f_\omega \circ Th(Tq)\right) \\ &= \left(D_q h \begin{bmatrix} n_{12} \\ n_{22} \end{bmatrix} \right)^{-1} \left[-\omega^2\theta - (\dot{D_q h})\dot{q} + D_q h M^{-1}\left(V + k + B\dot{q} + C\mathrm{sgn}(\dot{q})\right) \right] \end{aligned} \tag{38}$$

where $v_r = \frac{1}{K}\tau$. Figure 10 shows the resulting movement of the robot. The joint trajectories and the voltage command to the motor driver are shown in Figure 11.

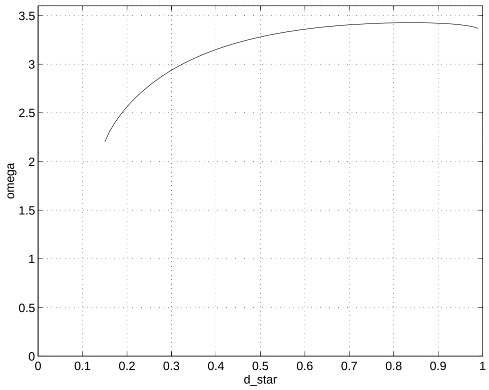

Figure 9: Numerical approximation $\omega = \hat{\lambda}(d^*)$ with $U(x) = \frac{1}{2}x^2$ for target dyamics. Target dynamics controller, τ_ω, is tuned according to this mapping, $\hat{\lambda}$, that is designed to locate neutral orbits originating in the ceiling.

Note that the closed loop dynamics of the system does not strictly admit a reverse time symmetry discussed above, since the uncancelled friction terms of the first joint enter the dynamics of the unactuated degree of freedom. However, under these circumstances, numerical simulation shown in Figures 10 and 11 suggests that the desired brachiation can be achieved. In practice, we have found that model mismatch seems to affect behavior of the physical robot rather considerably as discussed in Appendix C.

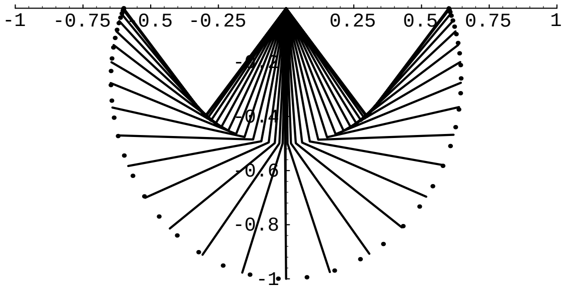

Figure 10: Movement of the robot (simulation) with $U(x) = \frac{1}{2}x^2$ in the target dynamics. The symmetry properties of the neutral orbit from the ceiling solves the ladder problem.

4.2.2 Simulation using Various Artifitial Potential Functions for Target Dynamics

In this section, we present numerical studies on the ladder problem using various choices of the artifitial potential, $U(x)$. Consider even artifitial potential functions such as, $U(x) = \frac{1}{4}x^4$, $U(x) = \frac{1}{2}x^2 + \frac{1}{4}x^4$ and $U(x) = \frac{1}{8}x^4$, which are in the family of "hard" springs, and $U(x) = 1 - \cos x$ and $U(x) = \frac{1}{2}x^2 - \frac{1}{24}x^4$, which are in the falimy of "soft" springs. In these simulations, the interval between the bars is $d = 0.6$.

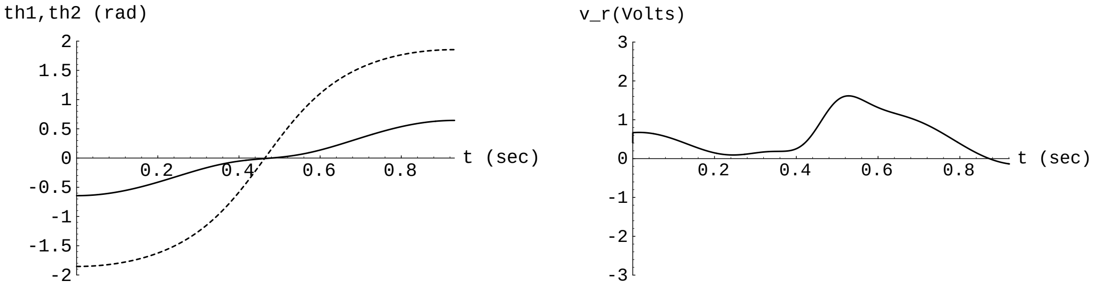

Figure 11: Simulation results of the ladder problem with $U(x) = \frac{1}{2}x^2$ for the target dynamics. Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver.

As the following simulation results depicted in Figures 12 – 16 suggest, all of these spring potentials work nicely in the ladder problem. Notice that a higher order "hard" spring such as $U(x) = \frac{1}{8}x^8$ calls for a large torque as shown in Figure 14.

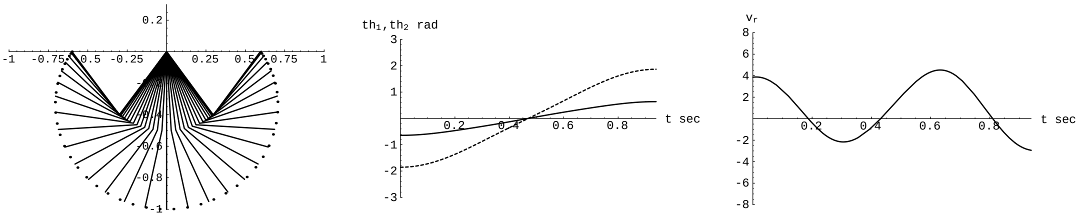

Figure 12: Simulation results of the ladder problem with a "hard" spring, $U(x) = \frac{1}{4}x^4$, in the target dynamics ($\omega = 2.513$). Left: Movement of the robot. Center: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver.

4.3 Swing up Problem

The swing up problem entails pumping up from the suspended posture at rest and catching the next bar. In order to achieve this task it is necessary not only to add energy in a suitable fashion but also to control the arm position at the capture of the next target bar. This suggests that we need to introduce a stable limit cycle to the system with suitable magnitude and relative phase in state. The idea we present here is a simple modification of the foregoing target dynamics. We define the "pseudo energy" with respect to the target variable and add a compensation term to the target dynamics in order to introduce the desired limit cycle.

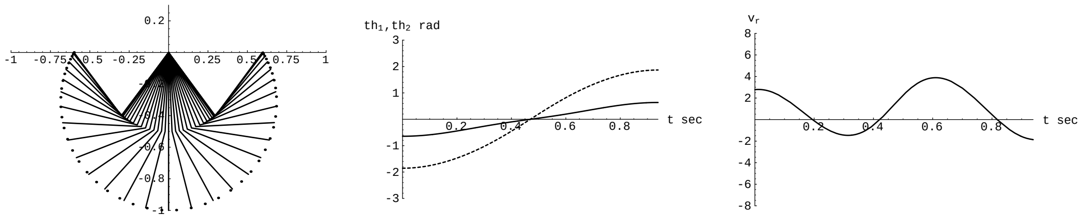

Figure 13: Simulation results of the ladder problem with a "hard" spring, $U(x) = \frac{1}{2}x^2 + \frac{1}{4}x^4$, in the target dynamics ($\omega = 2.00$). Left: Movement of the robot. Center: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver.

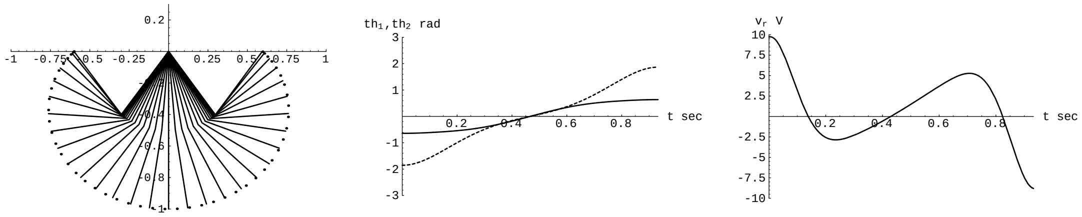

Figure 14: Simulation results of the ladder problem with a "hard" spring, $U(x) = \frac{1}{8}x^8$, in the target dynamics ($\omega = 1.288$). Left: Movement of the robot. Center: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. This potential achieves the task, but calls for a large torque.

4.3.1 Swing up Controller

As we have mentioned, swing up requires energy pumping in a suitable fashion. To achieve this we modify the target dynamics (14) as

$$\dot{Tx} = \begin{bmatrix} \dot{x} \\ -K_e(\bar{E} - \bar{E}^*)\dot{x} - \omega^2 DU(x) \end{bmatrix} := f_{\bar{E}^*}(Tx) \tag{39}$$

where, $x = h(q) = \theta = \theta_1 + \frac{1}{2}\theta_2$ as defined in (11)
K_e: a positive constant
$\bar{E}$: "pseudo energy" (15)
$\bar{E}^*$: the desired pseudo energy level

To achieve this target dynamics, the control law is formulated as

$$\begin{aligned} \tau_{\bar{E}^*} &= L_F H^{-1}\left(Tq, f_{\bar{E}^*} \circ Th(Tq)\right) \\ &= \left(D_q h \begin{bmatrix} n_{12} \\ n_{22} \end{bmatrix}\right)^{-1} \left[-\omega^2 DU(x) \circ h(q) - K_e(\bar{E} - \bar{E}^*)\dot{x} - (\dot{D_q}h)\dot{q} + D_q h M^{-1}(V + k)\right] \\ &= \frac{1}{n_{12} + \frac{1}{2}n_{22}} \left[-\omega^2 DU(x) \circ h(q) - K_e(\bar{E} - \bar{E}^*)\dot{x} + (n_{11} + \frac{1}{2}n_{21})(V_1 + k_1)\right] + V_2 + k_2 \end{aligned} \tag{40}$$

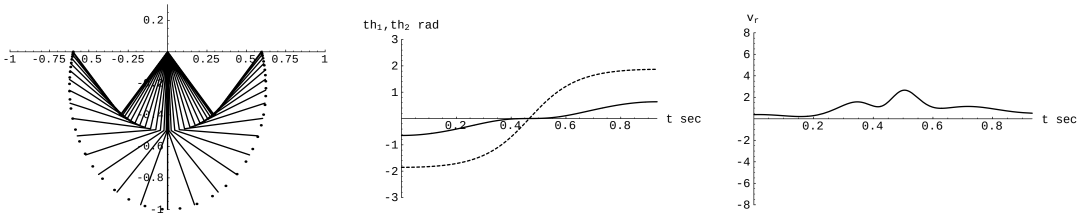

Figure 15: Simulation results of the ladder problem with a "soft" spring, $U(x) = 1 - \cos x$, in the target dynamics. Left: Movement of the robot ($\omega = 3.977$). Center: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver.

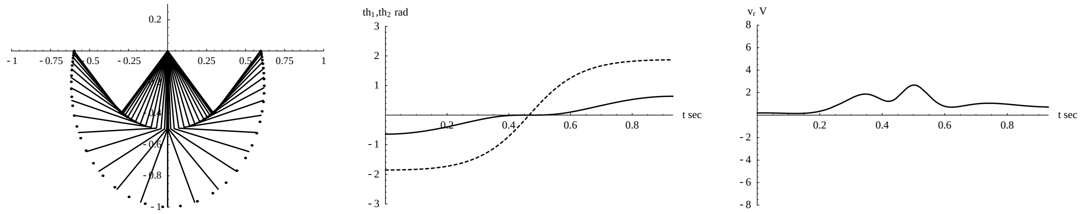

Figure 16: Simulation results of the ladder problem with a "soft" spring, $U(x) = \frac{1}{2}x^2 - \frac{1}{24}x^4$, in the target dynamics ($\omega = 3.499$). Left: Movement of the robot. Center: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver.

Now consider the time derivative of $\bar{E}$ along the motion

$$\dot{\bar{E}} = -K_e(\bar{E} - \bar{E}^*)\dot{x}^2 \tag{41}$$

If $\bar{E} > \bar{E}^*$ then the pseudo energy $\bar{E}$ decreases, and if $\bar{E} < \bar{E}^*$ then $\bar{E}$ increases. Therefore, $\bar{E}$ will converge to the desired level $\bar{E}^*$ eventually. This implies that the target dynamics has a stable limit cycle whose trajectory is characterized by $\frac{1}{2}\dot{x}^2 + \omega^2 U(x) = \bar{E}^*$ on the phase plane of $(x, \dot{x})$.

Although the system's motion projected onto the target subspace must exhibit the desired limit cycle, the swingup task still requires a coordination of the full four dimensional robot trajectory in order to guarantee the arm extension is correct at the moment the "virtual pendulum" angle reaches the ceiling. But for this task, in contrast to the ladder problem, we can make no assumption regarding the robot's initial conditions – the arm might start out in any configuration (typically, at small velocity) near the bottom state following a small "kick" of torque administered to break out of that passively stable equilibrium state. In particular, there is no comparable means of appeal to a tuned symmetry as before. Unfortunately, no general method is presently known to stabilize a highly nonlinear underactuated mechanical system around a specific (necessarily non-equilibrium) orbit. Hence, we are reduced to empirical tuning of the pumping gain, K_e in order to find task worthy values.

The effect of K_e on the target system is quite straightforward — equation (37) shows that it sets the time constant for convergence to the specified lower dimensional target limit cycle, hence, higher gains must result in quicker approach to the "virtual" steady state behavior. In contrast, the four dimensional closed loop system can be expected to exhibit extremely complex (revolute-revolute kinematic chains are "chaotic") orbits. Certainly, there is no reason to expect limit cycles from the true four dimensional system as its orbits accumulate toward the three dimensional limit set. Empirically, however, we find there are favorable regimes for small K_e wherein the system's motion tends toward "near-neutral" orbits resulting very slow swingup — that is, a relative phasing between the virtual angle and extension that brings the gripper to the next handhold at an acceptably small velocity. Fortunately, a numerical one parameter search is quite simple to implement. We have found it relatively straightforward to achieve effective swingup controllers both in simulation as well as in the lab by simply incrementing the value of this pseudo-energy pumping gain (starting from very small values), recording the favorable values as they recur, and then running with a favorable value whose associated pseudo-energy convergence rate is fast enough to yield a viable handhold over three or four swings.

4.4 Simulation

4.4.1 Simulation with a Hooke's Law Potential for Target Dynamics

What follows here is a presentation of different swing up behaviors resulting from changes in the rate of energy pumping as characterized by K_e using a Hooke's law potential, $U(x) = \frac{1}{2}x^2$, for target dynamics. The next bar is located at the distance $d^* = 0.6$, and we choose $\omega = \hat{\lambda}(0.6) = 3.36$ according to the mapping depicted in Figure 9. Since the bottom condition with zero velocity is an equilibrium state of the closed loop dynamics, we give small initial velocity to the second joint to initiate the swing motion in the desired direction [3]. In the following simulations, we assume that the robot can catch the bar when it comes very close to the desired handhold.

Very Slow Swing up ($K_e = 0.05$) Figure 17 shows the joint trajectories and the voltage command to the motor driver. The robot catches the bar at $t = 19.2$ seconds. A near neutral orbit is achieved with small choice of K_e.

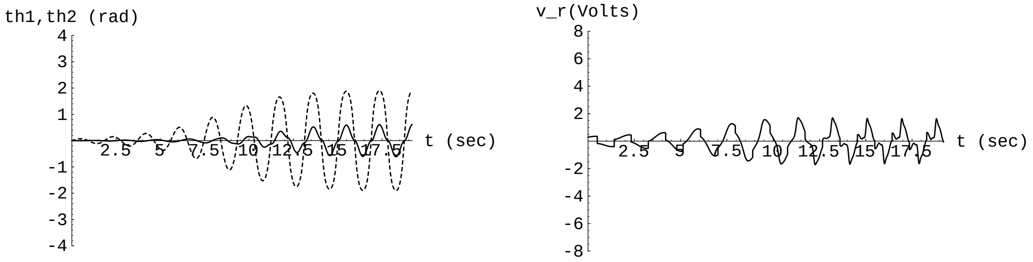

Figure 17: Simulation results of very slow swing up behavior ($K_e = 0.05$) using $U(x) = \frac{1}{2}x^2$. Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. The robot captures the bar at $t = 19.2$ seconds. Small choice of K_e achieves a near neutral orbit in the long time swing behavior.

[3] Here, we give a small "kick" velocity, $\dot{\theta}_{20} = \pm 0.2$, to the second joint in these simulations.

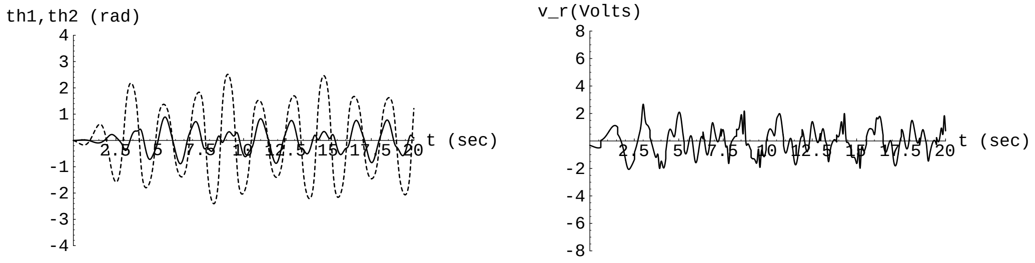

Figure 18: Simulation results of "chaotic" swing behavior ($K_e = 0.18$) using $U(x) = \frac{1}{2}x^2$. Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. "Chaotic" swing motion is observed when we let the robot keep swinging with large choice of K_e.

Slow Swing up ($K_e = 0.20$) Figure 19 shows the joint trajectories and the voltage command to the motor driver. The robot catches the bar at $t = 7.23$ seconds.

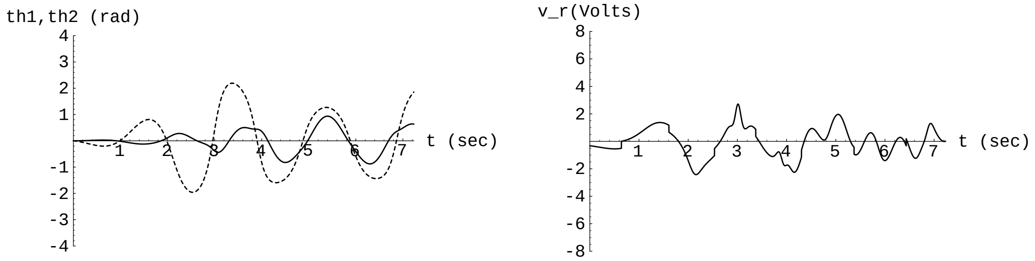

Figure 19: Simulation results of slow swing up behavior ($K_e = 0.20$) using $U(x) = \frac{1}{2}x^2$. Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. The robot captures the bar at $t = 7.23$ seconds.

Fast Swing up ($K_e = 0.228$) Figure 20 shows the joint trajectories and the voltage command to the motor driver. The robot catches the bar at $t = 3.55$ seconds.

Faster Swing up ($K_e = 0.328$) Consider the case where $K_e = 0.328$. Figure 21 shows the joint trajectories and the voltage command to the motor driver. The robot catches the bar at $t = 2.78$ seconds.

4.4.2 Simulation using Various Artifitial Potential Functions for Target Dynamics

In this section, we present numerical studies addressing the swing up problem using various choices of the artifitial potential, $U(x)$. We consider artifitial potential functions such as, $U(x) = \frac{1}{4}x^4$,

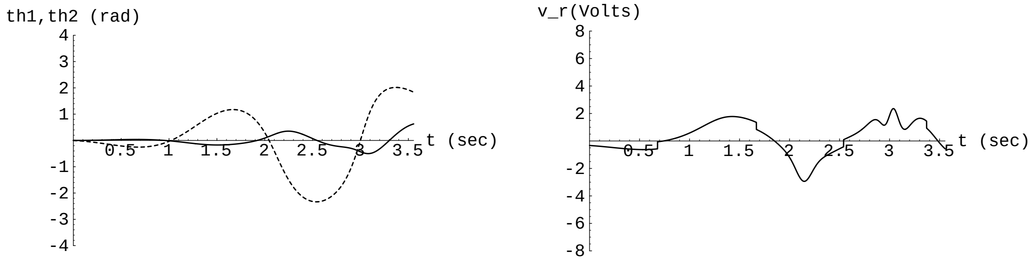

Figure 20: Simulation results of fast swing up behavior ($K_e = 0.228$) using $U(x) = \frac{1}{2}x^2$. Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. The robot captures the bar at $t = 3.55$ seconds.

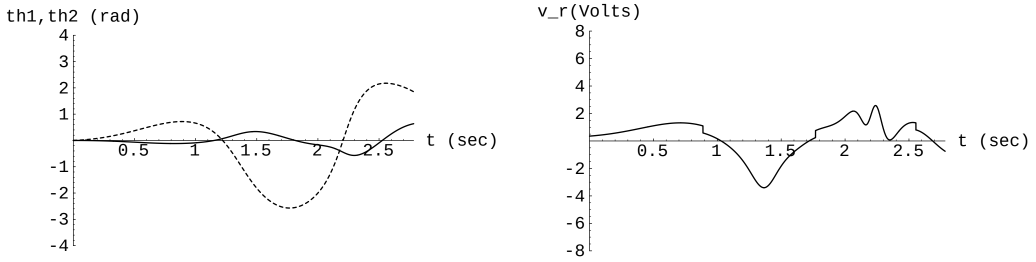

Figure 21: Simulation results of faster swing up behavior ($K_e = 0.328$) using $U(x) = \frac{1}{2}x^2$. Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. The robot captures the bar at $t = 2.78$ seconds.

$U(x) = \frac{1}{2}x^2 + \frac{1}{4}x^4$, and $U(x) = \frac{1}{8}x^8$, which are in the family of "hard" springs, and $U(x) = 1 - \cos x$, $U(x) = \frac{1}{2}x^2 - \frac{1}{24}x^4$, and $U(x) = \frac{1}{2}x^2 - \frac{1}{96}x^4$, which are in the falimy of "soft" springs.

Figures 22 – 27 shows the simulation results of the swing up problem using these spring potential functinos. "Hard" spring laws such as $U(x) = \frac{1}{4}x^4$, $U(x) = \frac{1}{2}x^2 + \frac{1}{4}x^4$ and $U(x) = \frac{1}{8}x^8$ work nicely and achieve a near neutral orbit as shown in Figures 22 – 24. However, as observed in Figures 22 and 24, notice that potential functions with zero stiffness around the origin results in oscillation with a long period when the amplitude of the swing is small.

In contrast, "soft" spring laws without "stiffness margin" profile such as $U(x) = 1 - \cos x$, $U(x) = \frac{1}{2}x^2 - \frac{1}{24}x^4$ fail the swing up task as shown in Figures 25 and 26. However, a "soft" spring law with some "stiffness margin" profile bounded away from zero (see Figure 7) achieve a near neutral orbit as shown in Figure 27.

As discussed in Section 3.3, these simulation results suggest that the stiffness profile plays an important role in achieving effective swing up behavior, while all of these work nicely in the ladder problem. Further investigation is necessary to gain full understanding of this matter.

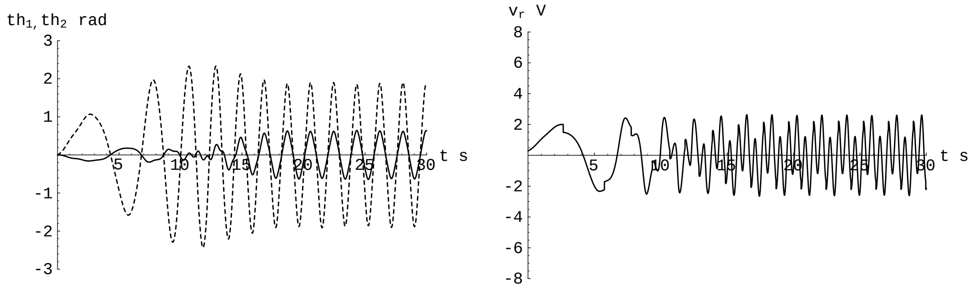

Figure 22: Simulation results of the swing up problem with $U(x) = \frac{1}{4}x^4$ for the target dynamics ($K_e = 0.05$). Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. This spring law achieves a near neutral orbit. Notice that the period of swing is long when the amplitude of oscillation is small.

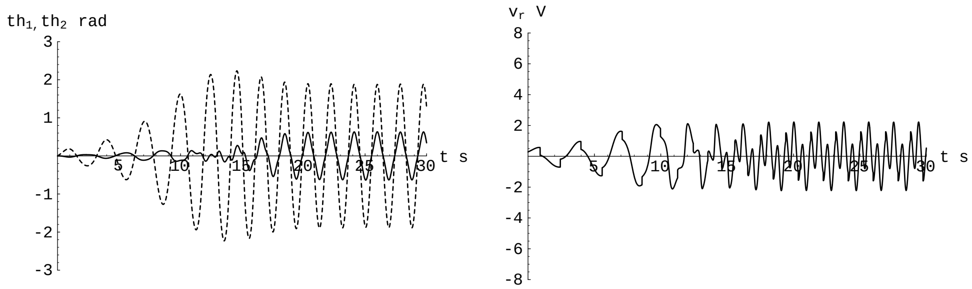

Figure 23: Simulation results of the swing up problem with $U(x) = \frac{1}{2}x^2 + \frac{1}{4}x^4$ for the target dynamics ($K_e = 0.05$). Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. This spring law achieves a near neutral orbit.

5 Rope problem

In this section, we consider the rope problem: brachiation along a continuum of handholds such as afforded by a branch or a rope. First, the average horizontal velocity is characterized as a result of the application of the target dynamics controller, τ_ω, introduced above. Then, we consider the regulation of horizontal velocity using this controller. An associated numerical "swing map" suggests that we indeed can achieve good local regulation of the foward velocity through the target dynamics method.

5.1 The Iterated Ladder Trajectory Induces a Horizontal Velocity

Supposing that the robot starts in the ceiling with zero velocity, then it must end in the ceiling under the target dynamics controller since θ follows the dynamics $\ddot{\theta} + \omega^2 = 0\theta$. However, if d and ω are not "matched" as $\omega = \lambda(d)$, then the trajectory ends in the tangent of the right branch of the ceiling, $Tq \in T\mathcal{C}_+$, with $\dot{\theta} = 0$ but $r \neq d$ and $\dot{r} \neq 0$. Shortly, we will present numerical evidence

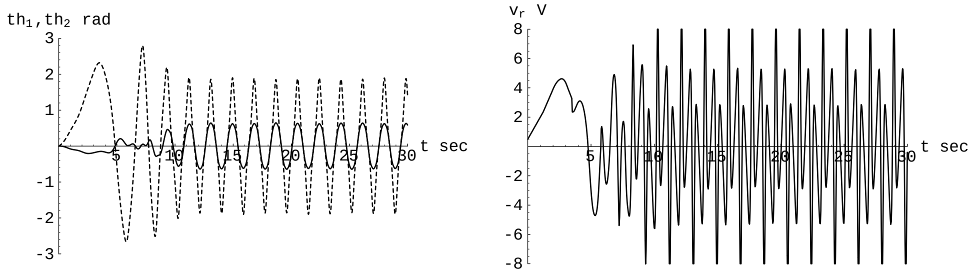

Figure 24: Simulation results of the swing up problem with $U(x) = \frac{1}{8}x^8$ for the target dynamics ($K_e = 0.05$). Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. This spring law achieves a near neutral orbit. Notice that the period of swing is long when the amplitude of oscillation is small.

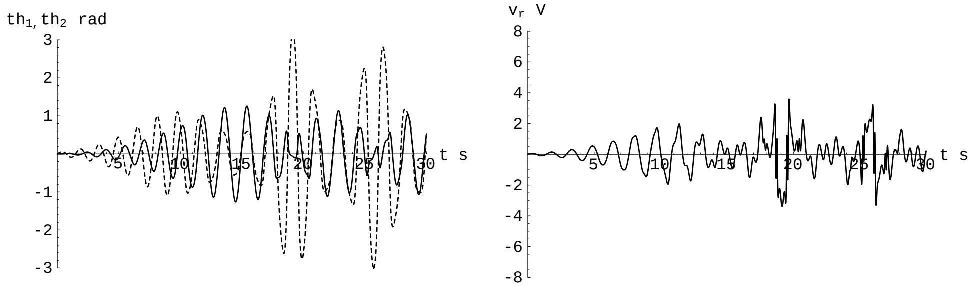

Figure 25: Simulation results of the swing up problem with $U(x) = 1 - \cos x$ for the target dynamics ($K_e = 0.05$). Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. This spring law fails the effective swingup task.

suggesting that when $d = d^* + \delta$ for small δ, then $\dot{r}$ at $Tq \in T\mathcal{C}_+$ is also small. Assuming that any such small nonzero velocity is "killed" in the ceiling, brachiation may be iterated by opening and closing the grippers at left and right ends in the appropriately coordinated manner. Namely, imagine that the robot concludes the swing by grasping firmly with its gripper the next handhold in the ceiling and thereby damps out the remaining kinetic energy. Imagine at the same instant that it releases the gripper clutching the previous handhold and thereby begins the next swing. We will call such a maneuver the Iterated Ladder Trajectory ("ILT").

It is natural to inquire as to how quickly horizontal progress can be made along the ladder in so doing. Notice in Figure 28 that when a gripper moves a distance $2d^*$ in the course of the ladder trajectory, and if the trajectory is immediately repeated, as described above, then the body, m_1, will also move a distance of d^* each swing, hence, its average horizontal velocity will be

$$\bar{\dot{h}} = \frac{d^*\omega}{\pi} = \frac{d^*\lambda(d^*)}{\pi} := \tilde{V}(d^*) \tag{42}$$

according to the discussion in Section 4.1. In Figure 29, we now plot the ceiling-to-velocity map

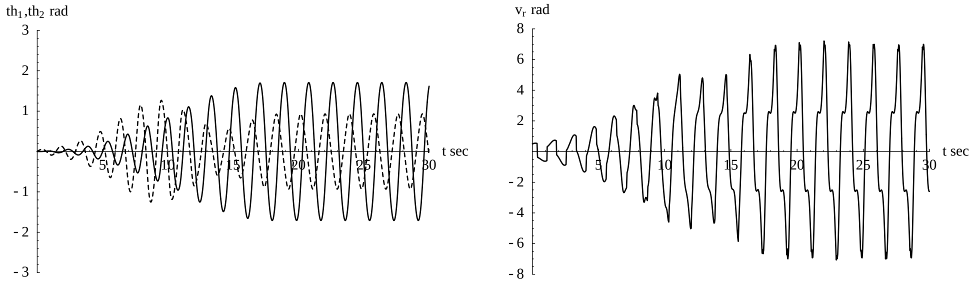

Figure 26: Simulation results of the swing up problem with $U(x) = \frac{1}{2} - \frac{1}{24}x^4$ for the target dynamics ($K_e = 0.05$). Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. This spring law results in a "out of phase" swing up which fails the task.

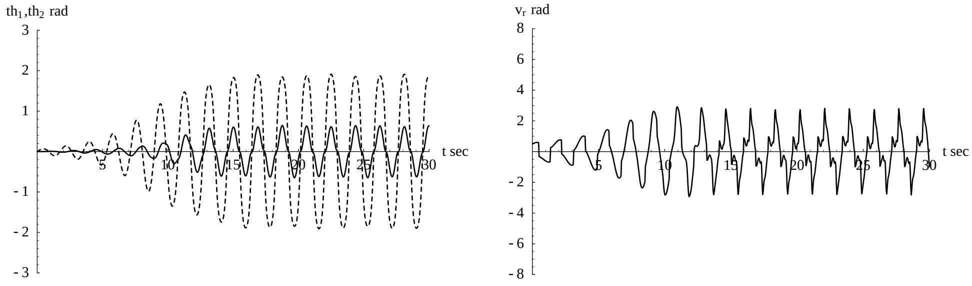

Figure 27: Simulation results of the swing up problem with $U(x) = \frac{1}{2} - \frac{1}{96}x^4$ for the target dynamics ($K_e = 0.05$). Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. With some "stiffness margin," this spring law achieves a near neutral orbit.

$\bar{\dot{h}} = \tilde{V}(d^*)$ for the choice of $U(x) = \frac{1}{2}x^2$ in target dynamics and the robot parameters in Table 1, where $\tilde{V}$ is computed using the numerical approximation, $\hat{\lambda}$ discussed in Section 4.1.3.

5.2 Inverting the Ceiling-to-Velocity Map

Consider now the task of obtaining the desired forward velocity $\bar{\dot{h}}^*$ of brachiation. If $\tilde{V}$ is invertible, then $d^* = \tilde{V}^{-1}(\bar{\dot{h}}^*)$ and we can tune ω in the target dynamics as

$$\omega = \lambda \circ \tilde{V}^{-1}(\bar{\dot{h}}^*) \tag{43}$$

to achieve a desired $\bar{\dot{h}}^*$ where λ is again the mapping (37).

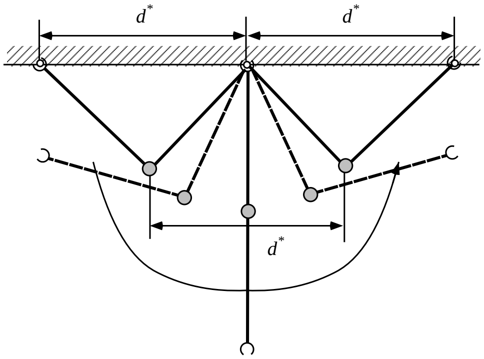

Figure 28: Progress of the robot per swing. The robot's body proceeds d^* per swing while a gripper moves $2d^*$.

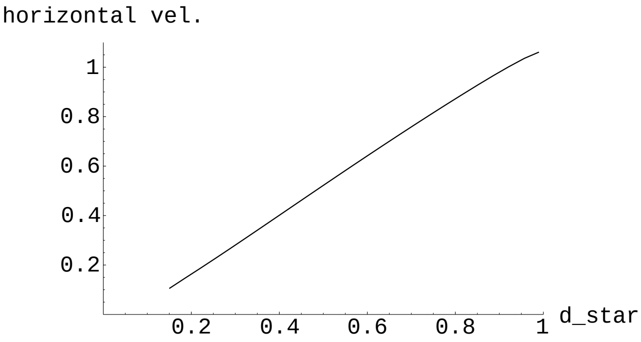

Figure 29: The ceiling-to-velocity map, $\tilde{V}$ using $U(x) = \frac{1}{2}x^2$. This mapping is inverted to obtain the desired forward velocity $\bar{\dot{h}}^*$.

5.3 Horizontal Velocity Regulation

Consider the ceiling condition with zero velocity

$$T\mathcal{C}_{0\pm} = \left\{ \begin{bmatrix} c_\pm(d) \\ 0 \\ 0 \end{bmatrix} \in T\mathcal{C} \mid d \in [0, 2l] \right\} \tag{44}$$

Define the maps, $C_\pm$, and their inverses, $C_\pm^{-1}$, as

$$C_\pm : [0, 2l] \to T\mathcal{C}_{0\pm} : d \mapsto \begin{bmatrix} c_\pm(d) \\ 0 \\ 0 \end{bmatrix} \tag{45}$$

$$C_\pm^{-1} : T\mathcal{C}_{0\pm} \to [0, 2l] : \begin{bmatrix} c_\pm(d) \\ 0 \\ 0 \end{bmatrix} \mapsto d \tag{46}$$

A target dynamics controller (9) gives

$$\mathcal{L}^{2\nu}_{\tau_\omega} \circ C_-(d) \in T\mathcal{C}_+ \text{ , where } \nu = \frac{\pi}{2\omega} \tag{47}$$

since θ follows the dynamics $\ddot{\theta} = -\omega^2\theta$. Now, if $\omega = \lambda(d)$, then

$$\mathcal{L}^{2\nu}_{\tau_\omega} \circ C_-(d) = C_+(d) = \begin{bmatrix} c_+(d) \\ 0 \\ 0 \end{bmatrix} \in T\mathcal{C}_{0+} \text{ , where } \nu = \frac{\pi}{2\omega} \tag{48}$$

because of the symmetry properties of the neutral orbits, demonstrated in Proposition 4.4.

Define a projection Π, from the ceiling's tangents into the zero velocity section,

$$\Pi : T\mathcal{C}_\pm \mapsto T\mathcal{C}_{0\pm}. \tag{49}$$

In other words, Π is a map that "kills" any velocity in the ceiling. We introduce this projection to model the ILT maneuver in cases when $\dot{r} \neq 0$ for $Tq \in T\mathcal{C}$. We plot, $\dot{r}$, the approaching velocity in the right branch of the ceiling for $d \in [0, 2l]$ where $d^* = 0.6496, \omega = 3.385$ in Figure 30.

To gain an intuitive feeling for the magnitude of "leftover energy" that must be "killed" before the next swing begins, we will compare it to the energy of the steady state swing. In the worst case, the kinetic energy in the ceiling $T\mathcal{C}_+$ resulting from the initial condition $Tq_0 = C_-(0.02)$ is $K(T\mathcal{C}_+) = 1.432$ J. The maximum kinetic energy during a swing when $d = d^*$ is $K_{d^*} \text{ max} = 10.209$ J. The ratio $\frac{K(T\mathcal{C}_+)}{K_{d^*} \text{ max}} = 0.1402$ seems to be acceptably small. Consider instead, more favorable range, where $d = d^* + \delta$ and $\delta = -0.2$. Now the kinetic energy killed in the ceiling is $K(T\mathcal{C}_+) = 0.2574$ J, and the ratio $\frac{K(T\mathcal{C}_+)}{K_{d^*} \text{ max}} = 9.593 \times 10^{-3}$ in this case is very small despite fairly large error (31 %) in the initial condition. This suggests that the idea of killing any approaching horizontal velocity in the ceiling may be physically reasonable.

We now have from (47)

$$\Pi \circ \mathcal{L}^{2\nu}_{\tau_\omega} \circ C_-(d) \in T\mathcal{C}_{0+} \quad \text{, where } \nu = \frac{\pi}{2\omega} \tag{50}$$

hence we may define a "swing map", σ_ω, as a transformation of $[0, 2l]$ into itself,

$$\sigma_\omega(d) := C_+^{-1} \circ \Pi \circ \mathcal{L}^{2\nu}_{\tau_\omega} \circ C_-(d) : [0, 2l] \to [0, 2l] \tag{51}$$

Note that if $\omega = \omega^* = \lambda(d^*)$, then

$$\sigma_\omega(d^*) = d^* \tag{52}$$

that is, d^* is a fixed point of the appropriately tuned swing map.

It is now clear that the dynamics of the ILT maneuver can be modeled by the iterates of this swing map, σ_ω. Physically, suppose we iterate by setting the next initial condition in the ceiling to be

$$Tq_0[k+1] = C_- \circ \sigma_\omega(d[k]). \tag{53}$$

This yields a discrete dynamical system governed by the iterates of σ_ω,

$$d[k+1] = \sigma_\omega(d[k]).$$

Numerical evidence suggests that the iterated dynamics converges, $\lim_{k\to\infty} \sigma^k_{\omega^*}(d) = d^*$, when d is in the neighborhood of d^* as depicted in Figure 31 (local asymptotic stability of the fixed point d^*). We plot the swing map calculated numerically for the case where $U(x) = \frac{1}{2}x^2$ for target dynamics, $\bar{h} = 0.7, d^* = 0.6496, \omega = 3.385$ and the robot parameters in Table 1 are used (see Figure 31).

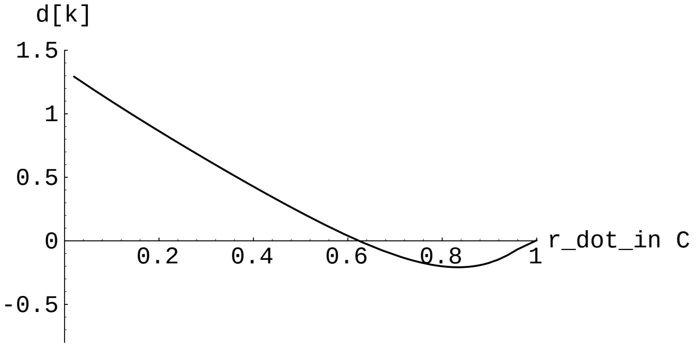

Figure 30: Approachig horizontal velocity of the robot gripper for the case $d^* = 0.6496, \omega^* = 3.385$ where $\omega^* = \hat{\lambda}(d^*)$ using $U(x) = \frac{1}{2}x^2$, and the robot parameters are in Table 1. When the error in the initial condition from d^* is small, the resulting approaching velocity in the ceiling is also small.

5.4 Simulation

5.4.1 Simulation with a Hooke's Law Potential for Target Dynamics

Suppose we want to achieve the desired horizontal velocity, $\bar{\dot{h}}^* = 0.7$. The parameters shown in in Table 1 and $U(x) = \frac{1}{2}x^2$ for target dyamics are used. For this case, ω is obtained as $\omega = \hat{\lambda}(0.6496) = 3.385$.

First, consider ILT with the proper initial condition

$$Tq_0^* = \begin{bmatrix} c_-(d^*) \\ 0 \\ 0 \end{bmatrix} \tag{54}$$

which is proper in the sense $\bar{\dot{h}}^* = \tilde{V}(d^*)$. The simulation result in this case is shown in Figure 32—a faithfully executed ILT at d^*.

Suppose, instead, that we select $\omega = \lambda(d^*)$ but the initial d_0 is wrong. We present simulation results with the initial condition

$$Tq_0 = \begin{bmatrix} c_-(d^* + \delta) \\ 0 \\ 0 \end{bmatrix}, \text{where } \delta = -0.2 \tag{55}$$

in Figure 33. As the numerical swing map of (31) suggets, we nevertheless achieve asymptotically the desired locomotion, i.e., $d \to d^*$.

With the assumption that any velocity in the ceiling is killed, the size of the domain of attraction to d^* under σ_{ω^*} is fairly large according to the numerical evidence shown in Figure 31.

5.4.2 Simulation with Various Spring Potential Functions for Target Dynamics

In this section, we consider several spring potential laws for target dyamics addressing the rope problem. Consider the potential functions such as $U(x) = \frac{1}{4}x^4$, $U(x) = \frac{1}{2}x^2 + \frac{1}{4}x^4$ and $U(x) =$

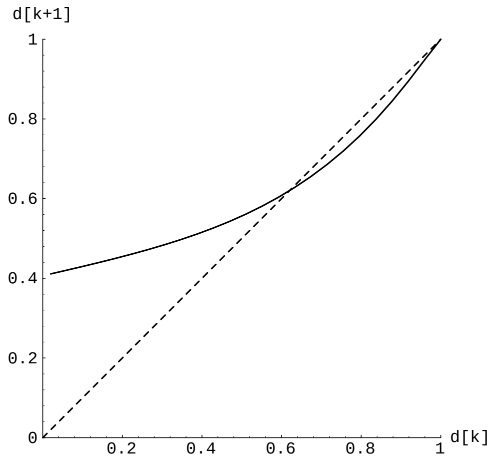

Figure 31: Swing map, σ_ω, (solid) and identity (dashed) for the case $\bar{\dot{h}} = 0.7, d^* = 0.6496, \omega^* = 3.385$ using $U(x) = \frac{1}{2}x^2$ for the target dynamics where $\omega^* = \hat{\lambda}(d^*)$, and the robot parameters are in Table 1. This swing map has an attracting fixed point at d^*.

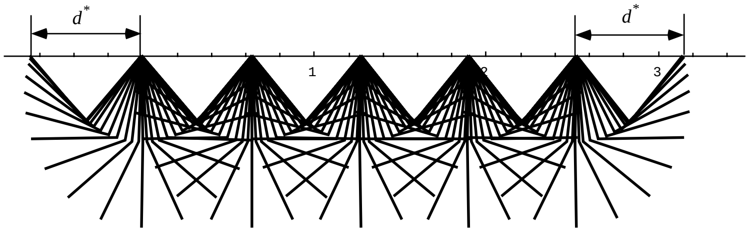

Figure 32: Brachiation along the bar with the initial condition (54) with $U(x) = \frac{1}{2}x^2$ for the target dynacmis. ILT locomotion at the fixed point d^* yields the desired average horizontal velocity, $\bar{\dot{h}}^*$.

$1 - \cos x$. We plot in Figure 34 the swing map for the case $\bar{\dot{h}} = 0.7$. Figure 35 – 37 show the simulation results with the initial condition (55).

As the numerical swing maps in Figure 34 sugget, we nevertheless achieve asymptotically the desired locomotion, i.e., $d \to d^*$. These swing map and simulation results suggest that both "hard" and "soft" spring potential laws work nicely in the rope problem as well as in the ladder problem.

6 Experiments

We present results of the experimental implementation of the proposed controller in order to validate our control strategy. The proposed algorithm is applied to the ladder and swing up problem, however the rope problem cannot be experimentally carried out with the robot considered in this paper because of the structure of the gripper.

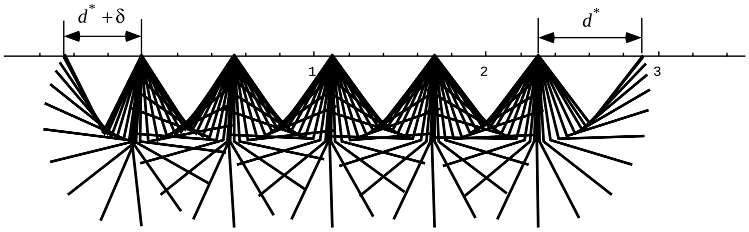

Figure 33: Brachiation along the bar with the initial condition (55) with $U(x) = \frac{1}{2}x^2$ for the target dynamics. Convergence of $d \to d^*$ is illustrated as the numerical swing map (Figure 31) indicates, and this yields convergence to the desired average velocity, $\bar{\dot{h}}^*$.

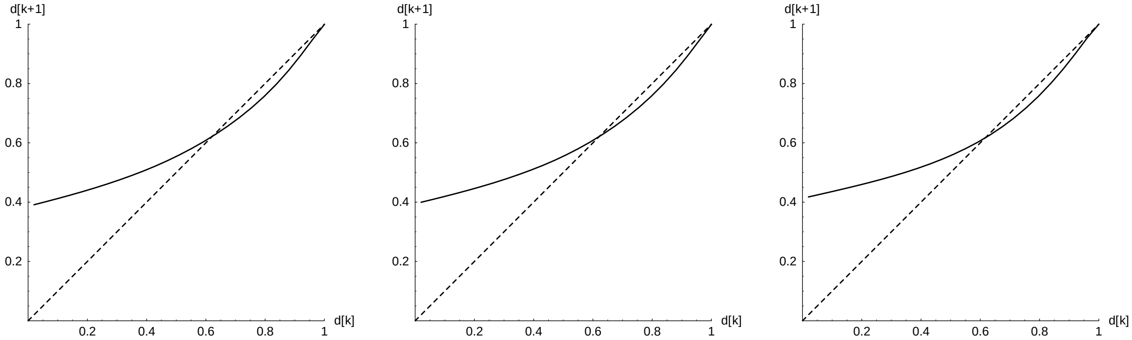

Figure 34: Swing map, σ_ω, (solid) and identity (dashed). Left: $U(x) = \frac{1}{4}x^4$, $\bar{\dot{h}} = 0.7$, $d^* = 0.6522$, $\omega^* = 2.534$. Center: $U(x) = \frac{1}{2}x^2 + \frac{1}{4}x^4$, $\bar{\dot{h}} = 0.7$, $d^* = 0.6520$, $\omega^* = 2.016$ Right: $U(x) = 1 - \cos x$, $\bar{\dot{h}} = 0.7$, $d^* = 0.6479$, $\omega^* = 4.0066$. These swing maps has an attracting fixed point at d^*.

6.1 Ladder Problem

This section considers the ladder problem—brachiation on a set of evenly spaced bars at the same height. In the experimental setting, the next bar is located at a distance of 0.6m.

As discussed in section 4.1, the symmetry property of neutral orbits solves that ladder problem. We need to choose ω in the target dynamics (9) for a given ladder distance, d^*. For our experimental setting, ω is tuned to be $\omega = 3.36$ according to the mapping depicted in Figure 9

Early attempts to implement the controller (38) failed. Swing motion close to the desired behavior was achieved, but the gripper did not come close enough to the target bar to catch it [4]. A central component contributing to these failures was the model mismatch. Therefore, we tuned the parameters of the model manually. Some experience is helpful in the refinement of these parameters:

[4] In practice, we need to consider the time lag in opening the gripper when the robot initiates locomotion, something not taken into account in the analytical work. It takes approximately 0.08 to 0.1 seconds to release the bar after the command to open the gripper is sent. Empirically, we have observed that this time affects the swing behavior of the robot. As a result, we choose to send the open command of the gripper 0.08 seconds before the target dynamics controller is turned on.

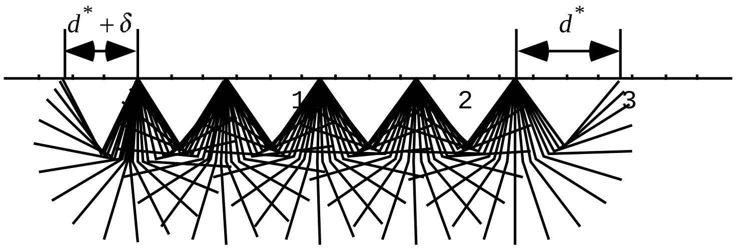

Figure 35: Brachiation along the bar with the initial condition (55) with $U(x) = \frac{1}{4}x^4$ for the target dynamics where $d^* = 0.6522$, $\omega = 2.534$. Convergence of $d \to d^*$ is illustrated as the numerical swing map (Figure 34) indicates, and this yields convergence to the desired average velocity, $\bar{\dot{h}}^*$.

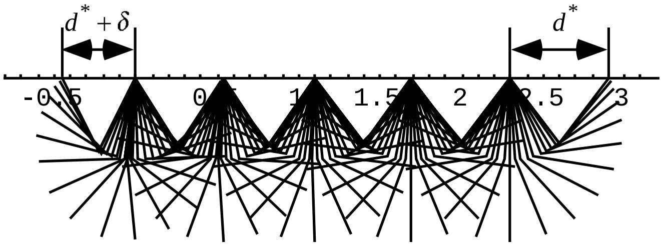

Figure 36: Brachiation along the bar with the initial condition (55) with $U(x) = \frac{1}{2}x^2 + \frac{1}{4}x^4$ for the target dynamics where $d^* = 0.6520, \omega = 2.016$. Convergence of $d \to d^*$ is illustrated as the numerical swing map (Figure 34) indicates, and this yields convergence to the desired average velocity, $\bar{\dot{h}}^*$.

we choose to use $m_1 = 3.39, m_2 = 1.30, c_2 = 0.73$ and $b_2 = 0.33$ instead of the values in Table 1 for the ladder problem. This assignment yielded success.

A typical motion of the physical robot is plotted in Figure 38, while the joint trajectories and the voltage commands sent to the driver are shown in Figure 39. The mean locomotion time of ten runs is 0.973 seconds with ±0.015 second error [5], which is very close to its analytically calculated value, $t = \frac{\pi}{\omega} = 0.935$ seconds.

Notice that the symmetry of the neutral orbit is not perfectly achieved in the motion of the robot. We discuss the discrepancy between the simulation and experimental results in Appendix C.

6.2 Swing up Problem

As we have mentioned, the swing up problem represents the task of swinging up from the suspended posture at rest and catching the next bar. The results of the experimental implementation of the proposed controller are presented.

[5] In the sequel, the error refers to the maxinum deviation from the mean.

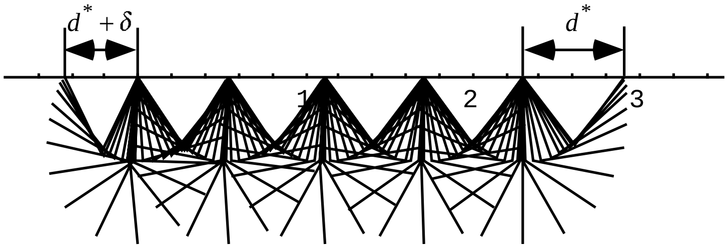

Figure 37: Brachiation along the bar with the initial condition (55) with $U(x) = 1 - \cos x$ for the target dynamics where $d^* = 0.6479$, $\omega = 4.0066$. Convergence of $d \to d^*$ is illustrated as the numerical swing map (Figure 34) indicates, and this yields convergence to the desired average velocity, $\bar{\dot{h}}^*$.

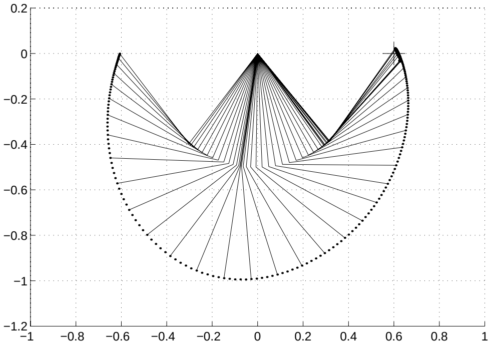

Figure 38: Movement of the robot (experiment). The target bar is located at a distance of 0.6m marked by the "+".

6.2.1 Experimental Results

In order to achieve the task, we need to bring the effective actuated portion of the state, θ, to the right pseudo energy level, while simultaneously ensuring that the unactuated degree of freedom, r, coincide with the regulated length between the bars, d^*.

What follows is a presentation of the different swing up behaviors resulting from changes in the rate of energy pumping, as characterized by K_e. The distance between the bars is 0.6m. We consider three cases where $K_e = 0.03, 0.47$ and 0.9. These parameters are chosen manually based on our experience in numerical simulation and experiments. In order to successfully swing up, we have found it necessary to slightly modify the desired pseudo energy level and some of the model parameters. The nominal pseudo energy is chosen to be $\bar{E}^*_{nom} = \frac{1}{2}\omega^2 \left(\frac{\pi}{2}\right)^2$ so that the gripper reaches the height of the bar, which corresponds to the condition, $\theta = \frac{\pi}{2}$ when $\dot{\theta} = 0$. In the initial attemps using the nominal pseudo energy level, we found that the gripper of the robot came close to the bar, but did not reach the enough height up to the ceiling to catch it. Thus, we introduce a slight modification to this value as $\bar{E}^* = 1.1\bar{E}^*_{nom}$ to increase the amplitude of the oscillation so that the gripper reaches the height of the bar, and we choose to use $m_1 = 3.39, m_2 = 1.30$ instead

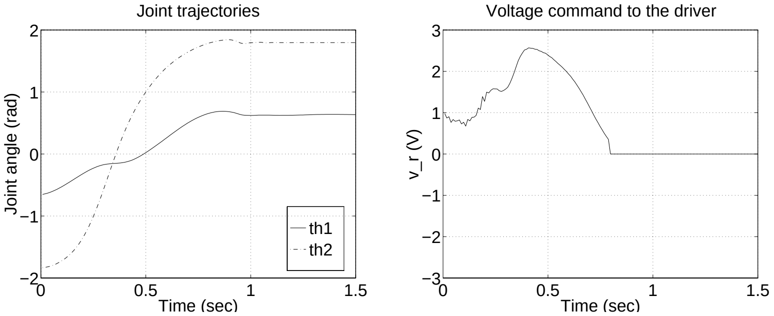

Figure 39: The experimental results of the ladder problem. Left: Joint trajectories, Right: Voltage command to the motor driver

of the values in Table 1. The initial direction of the swing motion depends solely upon the initial states of the system since the motion of the robot is governed by the closed loop dynamics. Only small deviation from the origin on the phase plane determines this direction. Thus, we introduce an impulse-like initial torque before the controller is turned on so that the robot starts its swing motion in the desired direction at every run. The experimental results of swing up problem do not exactly match those of numerical simulations presented in section 4.3. We investigate this matter in Appendix C.

Slow Swing up ($K_e = 0.03$) Consider the case where $K_e = 0.03$. Figure 40 shows the joint trajectory and the voltage command to the motor driver. The mean time of ten runs for this slow swing up behavior is 7.474 seconds with ±0.080 second error.

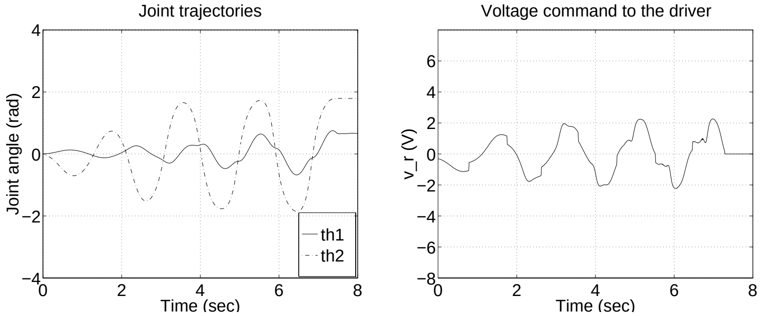

Figure 40: Experimental results of slow swing up behavior ($K_e = 0.03$). Left: Joint trajectories, right: Voltage command to the motor driver. The robot captures the bar when $t \sim 7.5$ seconds.

Fast Swing up ($K_e = 0.47$) Consider the case where $K_e = 0.47$. Figure 41 shows the joint trajectory and the voltage command to the motor driver. This choice K_e yields relatively fast swing up. The mean swing up time of ten runs for this swing up is 3.843 seconds with ±0.146 second error.

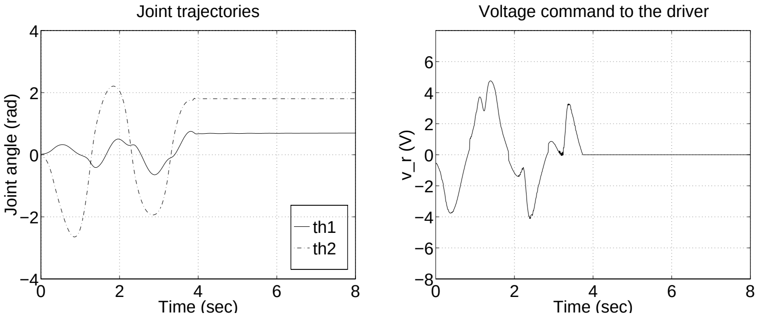

Figure 41: Experimental results of fast swing up behavior ($K_e = 0.47$). Left: Joint trajectories, right: Voltage command to the motor driver. The robot captures the bar when $t \sim 3.8$ seconds.

Faster Swing up ($K_e = 0.9$) Consider the case where $K_e = 0.9$. Figure 42 shows the joint trajectory and the voltage command to the motor driver. This choice of K_e yields a "faster" swing up maneuver. The mean swing up time of ten runs for this movement is 2.913 seconds with ±0.025 second error. In this case, the initial impulse-like torque is applied in the opposite direction to the previous two cases in order to start swinging in the CCW direction.

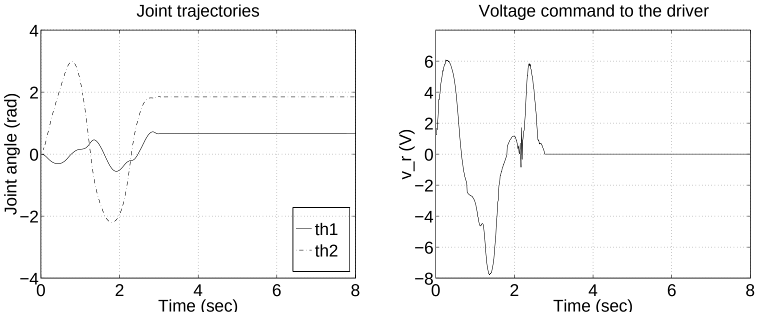

Figure 42: Experimental results of faster swing up behavior ($K_e = 0.9$). Left: Joint trajectories, right: Voltage command to the motor driver. The robot captures the bar when $t \sim 2.9$ seconds.

6.3 Continuous Locomotion

Here we exhibit the demonstration of continuous locomotion over several rungs of the ladder. Figure 43 depicts repeated locomotion of the robot initiated at the ceiling and moving from left to right. This motion can be considered as the iteration of the ladder trajectory. After each swing, the initial condition is reset, and the function of each arm is switched. This switching is done manually by looking at the motion of the robot to make sure that the it does not fall off from the ladder by mistakenly releasing the grasping bar before catching the next bar with some automated manner, which may result in serious damage to the robot. Due to the symmetrical structure of the robot, the same model is used in each swing where the configuration of the robot is "flipped over." In Figure

44, we show a picture of continuous locomotion initiated from the suspended posture. This is a combination of the "faster" swing up maneuver and the iterated ladder trajectory. First, the robot swings three times—going forth (1) and back (2) to gain momentum, and again swinging forward (3) to catch the bar—with the swing up controller ($K_e = 0.9$) described above. Then the control law is switched into the locomotion controller.

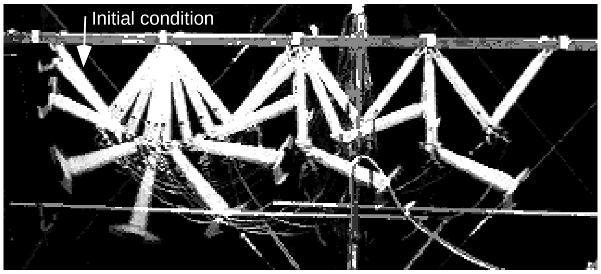

Figure 43: A Picture of continuous locomotion started in the ceiling. The robot iterates brachiation three times moving from left to right.

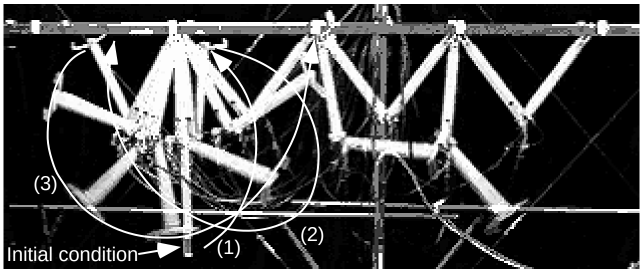

Figure 44: A Picture of continuous locomotion initiated from the suspended posture. First, the robot swings three times—going forth (1) and back (2) to gain momentum, and again swinging forward (3) to catch the bar—with the swing up controller ($K_e = 0.9$) described above. Then the control law is switched into the locomotion controller.

In these experiments, we have observed that disturbances caused by the cable, which hangs down from above, can occasionally have a detrimental effect on the robot's motion. In particular, sometimes, the robot has difficulty reaching the bar because of the dragging effect of the cable. Thus, some care has to be taken so that the influence of the cable can be reduced. Nonetheless, we feel that these experimental results demonstrate the relevance of our strategy despite the many practical issues which have not been formally treated, such as model mismatch, inaccuracy of sensors and actuators, and the presence of various disturbances.

7 Conclusion

We have presented preliminary studies of a new brachiating controller for a simplified two-link robot. The algorithm uses a target dynamics method to solve the ladder, swing up and rope problems. These tasks are encoded as the output of a target dynamical system inspired by the pendulum-like motion of an ape's (slow) brachiation. We provide numerical simulations suggesting the effectiveness of the proposed algorithm. We also present our empirical success in the implementation of the target dynamics method to a physical two-link brachiating robot. The proposed algorithm is applied to achieve the ladder and swing up behaviors. We achieve swing locomotion in the ladder problem and various swing up behaviors with different rates of energy pumping, as characterized by K_e. We demonstrate repeated locomotion over several rungs of the ladder as well. The experimental success bears out the validity of our control strategy in spite of the presence of model mismatches and physical effects previously unconsidered, although some manual tuning is required to implement these ideas.

In section 7.1 we review some of the open questions this raises and in section 7.2 we address future work.

7.1 Open Problems

These numerical simulations and experimental results suggest that the proposed algorithm is effective for solving robot brachiation problems. However, a formal mathematical analysis will be required in order to truly understand how these ideas work. Most importantly, we need to consider the internal boundedness of the states of the closed loop system. The unactuated dynamics of our closed loop take the form of a one degree of freedom mechanical system forced by a periodic input. Such problems of parametric resonance are known to be complex. A second open problem concerns the swing map. Numerical studies suggest the local stability of the fixed point d^* but this must be verified analytically, and the extent of the domain of attraction must be characterized.

7.2 Future Work

The controller developed in this paper requires exact model knowledge of the robot. "Passive" and, hence, less model dependent strategies will be addressed in our future work pursuing the analogy between the brachiation problem and the control of hopping robots. This analogy becomes particularly useful as we begin to contemplate studies of robot brachiation using more complicated models with higher degrees of freedom, where modelling of such systems is much more difficult. Specifically in Schwind's study on the control of simplified spring loaded inverted pendulum (SLIP) hopping robots [10, 11], a particular choice of a spring law allows us to integratate the system's dynamical equation of the stance phase analytically, and gives us the stance map in closed form. We suspect a similar approach may make the slow brachiation problem more analytically tractable.

Finally, the study of the fast brachiation—the leap problem—seems compelling. For reasons discussed in the introduction, this problem lies in the more distant future. We believe there are generalizable principles of brachiation which may be established through the study of this simplified two degree of freedom model.

In the longer run, we believe that the ideas presented in this paper may have wider application to such areas of robotics as dexterous manipulation, legged locomotion and underactuated mechanisms.

Appendix

A Model of a Two-link Brachiating Robot

This section describes the detailed dynamics of the two-link brachiating robot depicted in Figure 4. The equation of the motion of the system is given by

$$\dot{Tq} = \mathcal{L}(Tq, v_r) \tag{56}$$

where

$$q = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \in \mathcal{Q}, \; Tq = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \in T\mathcal{Q}$$

$$\mathcal{L}(Tq, v_r) = \begin{bmatrix} \dot{q} \\ M(q)^{-1} \left(-V(q,\dot{q}) - k(q) - B\dot{q} - C\mathrm{sgn}(\dot{q}) + \begin{bmatrix} 0 \\ Kv_r \end{bmatrix} \right) \end{bmatrix}$$

$$\begin{aligned} m_{11} &= m_1 {l_{c1}}^2 + I_1 + m_2({l_1}^2 + 2l_1 l_{c2} \cos\theta_2 + {l_{c2}}^2) + I_2 \\ m_{12} &= m_{21} = m_2({l_{c2}}^2 + l_1 l_{c2} \cos\theta_2) + I_2 \\ m_{22} &= m_2 {l_{c2}}^2 + I_2 \end{aligned}$$

$$V(q,\dot{q}) = -m_2 l_1 l_{c2} \sin\theta_2 \begin{bmatrix} 2\dot{\theta}_1\dot{\theta}_2 + {\dot{\theta}_2}^2 \\ -{\dot{\theta}_1}^2 \end{bmatrix}$$

$$k(q) = \begin{bmatrix} m_1 g l_{c1} \sin\theta_1 + m_2 g(l_1 \sin\theta_1 + l_{c2}\sin(\theta_1 + \theta_2)) \\ m_2 g l_{c2} \sin(\theta_1 + \theta_2) \end{bmatrix}$$

$$B = \mathrm{diag}\{b_i\}, \quad C = \mathrm{diag}\{c_i\},$$

m_i and I_i are the mass and the moment of inertia of each link respectively, and l_i is the link length. The center of mass of each link is located on the center line which passes through adjacent joints at a distance l_{ci}. c_i and b_i denote the coulomb and viscous friction coefficients respectively. We assume that the elbow actuator produces torque proportional to a voltage command, v_r, sent to a driver as $\tau = Kv_r$, where K is a positive constant. In this paper, we assume that the length of each link is the same, $l_1 = l_2 = l$.

B Parameter Identification

We need to identify the dynamical parameters corresponding to the robot's Lagrangian dynamics. We initially considered an off-line least squares estimation method with torque filtering [23], but were unable to obtain a good estimate of the parameter set with this scheme. This may be because of somewhat inaccurate and noisy sensory data particularly obtained with the rate gyro [6]. In consequence, we resorted to a rather simple identification procedure, where the inertia parameters are obtained either or direct measurement or from the manufacturer's data, and the preliminary estimate of the friction coefficients are obtained from the natural dissipation of the system. These parameters

[6] Standard least squares method is susceptible to noise and inaccuracy in the measurement. Numerical studies suggests that we indeed obtain good estimation of the set of dynamical parameters with the absence of error and noise in the measurement.

were refined iteratively by comparing step and sinusoid responses obtained experimentally to those generated by simulations using the "best" parameters. In this comparison, we considered step response with various amplitude as well as sinusoid response with various amplitude and frequencies. The results of the parameter identification are listed in Table 1. Here, the mass of the two motors at the elbow joint is included in the first link, however, we could also derive an equivalent model having symmetry in the link parameters since there is redundancy in the inertia parameters. The efficacy of this parameter identification approach is illustrated in Figure 45 which shows examples of the comparison between experimental runs and simulations using the parameters of Table 1.

Description		i=1	i=2
Mass	m_i(kg)	3.499	1.232
Moment of inertia	I_i(kgm^2)	0.090	0.033
Link length	l_i(m)	0.50	0.50
Location of CG	l_{ci}(m)	0.414	0.333
Viscous friction	b_i(Nm/s)	0.02	0.14
Coulomb friction	c_i(Nm)	0.02	0.45
Torque constant	K(Nm/V)	1.752	

Table 1: The dynamical parameters of the robot obtained by the procedure described in Section B.

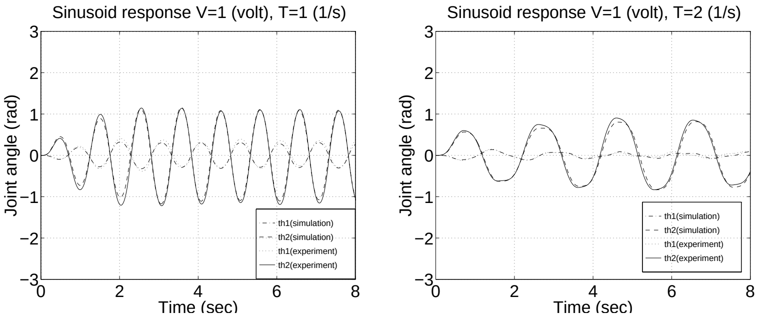

Figure 45: Examples of the comparison between experimental runs and simulations. Left: voltage command $v_r = \sin(2\pi t)$, right: voltage command $v_r = \sin(\pi t)$. These plots show close matching between the numerical simulations using the obtained model and experiments.

C Discrepancy between Simulations and Experiments

C.1 Unmodelled Nonlinear Characteristics of Harmonic Drive DC Motors

We see some discrepancy in the motion of the robot and the choices of K_e to achieve similar swing up behavior of the robot between the numerical simulations and experiments presented above. This section presents our efforts to understand how various physical effects and model mismatch affect the behavior of the robot.

The proposed controller using input/output linearization technique aggressively cancels nonlinearities in the plant dynamics to achieve the target dynamics, which requires exact model knowledge of the system. As we have pointed out, harmonic drives bear complicated nonlinear dynamics [19]. We suspect that unmodelled nonlinear characteristics of the harmonic drive DC motors at the elbow joint, such as nonlinear viscous friction, stiction and torque saturation, may be one of the main reasons of such discrepancy. Consider a slightly modified friction model, which includes coulomb friction, linear and cubic viscous friction and stiction, denoted by

$$\tau_{fric} = c_2 \mathrm{sgn}(\dot{\theta}_2) + b_2 \dot{\theta}_2 + \bar{b}_2 \dot{\theta}_2^{\,3} + s_2 \mathrm{sgn}(\dot{\theta}_2) e^{-\kappa|\dot{\theta}_2|} \tag{57}$$

where c_2 is the coulomb friction coefficient, b_2 and $\bar{b}_2$ are the viscous friction coefficients, s_2 represents stiction torque, and κ denotes the lubrication coefficient [24]. The introduction of the cubic term in (57) seems to be reasonable as described in the study on the modelling of harmonic drive gear transmission mechanisms [19]. We assume that torque produced by the actuator saturates when torque commands exceeds the regular operating range of the motor.

C.2 Simulation

In the following numerical studies, we present our effort to reproduce the circumstances in the experiments in order to understand the reasons for the discrepancy. For the plant model, we use the dynamics denoted by (1), but the friction terms of the second joint are substituted by (57). The inertia parameters shown in Table 1, and the friction parameters, $c_1 = 0.02, b_1 = 0.02, c_2 = 0.22, b_2 = 0.14, \bar{b}_2 = 6.02 \times 10^{-3}, s_2 = 0.5$ and $\kappa = 20$ are used for the robot. However, we choose to use $\bar{b}_2 = 0$ for the slow swing up case ($K_e = 0.03$) since we have found in numerical simulation that setting $\bar{b}_2 = 0$ gives better match. In fact, as discussed in [19], harmonic drives have other complicated characteristics such as variation of friction depending on the position of harmonic-drive output and dramatic increase of dissipation at some operating ranges that excite system resonance, which are difficult to model. Furthermore, [19] points out that friction in some drives can actually decrease over some velocity ranges as reported in [25]. In the following simulations, the torque saturation is introduced at ±5.2Nm[7]. For the controller, we use the same control law (38) and the same dynamical parameters that are used for the experimental implementation. Although we have yet to gain full understanding of the circumstances, the following simulations do, indeed, match closely with observed experimental results suggesting that our assumptions of unmodelled dynamics and torque saturation of the actuator may be reasonable for the explanation of some of the causes of the discrepancy we have seen.

C.2.1 Ladder Problem

Consider the same case as the experiments presented in section 6.1. The next bar is located at the distance of $d^* = 0.6$ and we choose $\omega = \hat{\lambda}(0.6) = 3.36$. The same parameters are used for the controller as we have chosen in the experiments. The movement of the robot is depicted in Figure 46, while the joint trajectories and the voltage commands sent to the driver are shown in Figure 47. The gripper reaches $d = 0.623$ at $t = 0.810$ seconds. The numerical simulation closely match the experimental results.

[7] According to the manufacturer's data, the rated torque of this motor is 3.2Nm, the instantaneous maximum torque is 14Nm, and the rated current is 1.8A which corresponds to the torque about 5.26Nm.

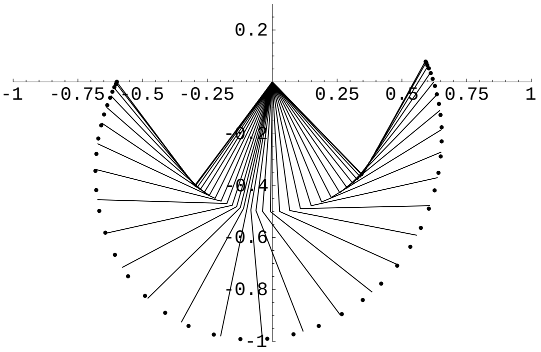

Figure 46: Movement of the robot considering unmodelled characteristics of the actuator. The numerical simulations closely match the corresponding experimental results.

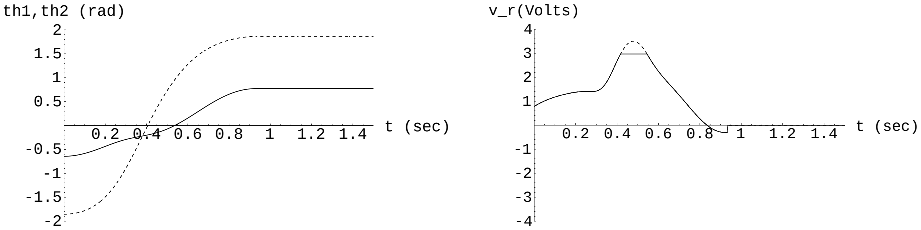

Figure 47: The simulation results of the ladder problem considering unmodelled characteristics of the actuator. Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. Solid line corresponds to the produced torque by the actuator considering saturation. Dashed line denotes the voltage command. The numerical simulation closely match the corresponding experimental results in Figure 39.

C.2.2 Slow Swing up ($K_e = 0.03$)

The same parameters are used for the controller as we have chosen in the experiments (K_e and dynamical parameters of the robot). Figure 48 shows the joint trajectory and the voltage command to the motor driver. The gripper reaches at $d = 0.532$ at $t = 7.36$ seconds. The numerical simulations closely match the experimental results.

C.2.3 Fast Swing up ($K_e = 0.47$)

Consider the same case of the fast swing up ($K_e = 0.46$) as the experiments presented in section 6.2. The same parameters are used for the controller as we have chosen in the experiments (K_e and dynamical parameters of the robot). Figure 49 shows the joint trajectory and the voltage command to the motor driver. The gripper reaches at $d = 0.599$ at $t = 3.83$ seconds. The numerical simulations closely match the experimental results.

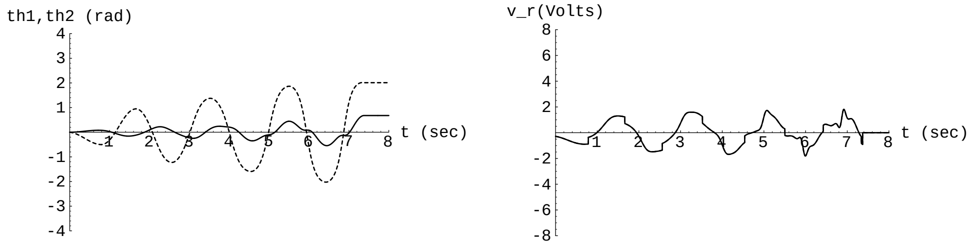

Figure 48: Simulation results of slow swing up behavior ($K_e = 0.03$) considering unmodelled characteristics of the actuator. Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. Solid line corresponds to the produced torque by the actuator considering saturation. Dashed line denotes the voltage command.The numerical simulations closely match the corresponding experimental results.

C.2.4 Faster Swing up ($K_e = 0.9$)

Consider the same case of the faster swing up ($K_e = 0.9$) as the experiments presented in section 6.2. The same parameters are used for the controller as we have chosen in the experiments (K_e and dynamical parameters of the robot). Figure 50 shows the joint trajectory and the voltage command to the motor driver. The gripper reaches at $d = 0.645$ at $t = 2.79$ seconds. The simulation closely matches the experimental results.

References

[1] Elwyn L. Simons, *Primate Evolution: An Introduction to Man's Place in Nature*, Macmillan, 1972.

[2] Daniel E. Koditschek, "Dynamically dexterous robots," in *Robot Control: Dynamics, Motion Planning and Analysis*, Mark W. Spong, F. L. Lewis, and C. T. Abdallah, Eds., pp. 487–490. IEEE Press, 1993.

[3] R. L. Andersson, *A Robot Ping-Pong Player: Experiment in Real-Time Intellgent Control*, MIT Press, 1988.

[4] Martin Bühler, Daniel E. Koditschek, and Peter J. Kindlmann, "A family of robot control strategies for intermittent dynamical environments," *IEEE Control Systems Magazine*, vol. 10, no. 2, pp. 16–22, Feb. 1990.

[5] Alfred A. Rizzi, Louis L. Whitcomb, and Daniel E. Koditschek, "Distributed real-time control of a spatial robot juggler," *IEEE Computer*, vol. 25, no. 5, pp. 12–24, May 1992.

[6] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *International Journal of Robotics Research*, vol. 18, no. 6, pp. 534–555, June 1999.

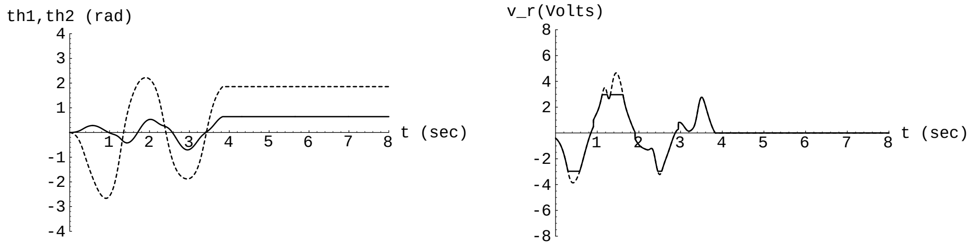

Figure 49: Simulation results of fast swing up behavior ($K_e = 0.47$) considering unmodelled characteristics of the actuator. Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. Solid line corresponds to the produced torque by the actuator considering saturation. Dashed line denotes the volgate command. The numerical simulations closely match the corresponding experimental results.

[7] Kevin M. Lynch and Matthew T. Mason, "Dynamic nonprehensile manipulation: Controllability, planning and experiments," *International Journal of Robotics Research*, vol. 18, no. 1, pp. 64–92, Jan. 1999.

[8] Marc H. Raibert, *Legged Robots that Balance*, MIT Press, 1986.

[9] Daniel E. Koditschek and Martin Bühler, "Analysis of a simplified hopping robot," *International Journal of Robotics Research*, vol. 10, no. 6, pp. 587–605, Dec. 1991.

[10] William J. Schwind and Daniel E. Koditschek, "Control the forward velocity of the simplified planner hopping robot," in *IEEE International Conference on Robotics and Automation*, 1995, pp. 691–696.

[11] William J. Schwind and Daniel E. Koditschek, "Characterization of monoped equilibrium gaits," in *IEEE International Conference on Robotics and Automation*, 1997, pp. 1986–1992.

[12] A. F. Vakakis, J. W. Burdick, and T. K. Caughey, "An "interesting" strange attractor in the dynamics of a hopping robot," *International Journal of Robotics Research*, vol. 10, no. 6, pp. 606–618, Dec. 1991.

[13] Robert T. M'Closkeys and Joel W. Burdick, "Periodic motions of a hopping robot with vertical and forward motion," *International Journal of Robotics Research*, vol. 12, no. 3, pp. 197–218, June 1993.

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

[15] Mark Spong, "Partial feedback linearization of underactuated mechanical systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept. 1994, pp. 314–321.

[16] Fuminori Saito, Toshio Fukuda, and Fumihito Arai, "Swing and locomotion control for a two-link brachiation robot," *IEEE Control Systems Magazine*, vol. 14, no. 1, pp. 5–12, Feb. 1994.

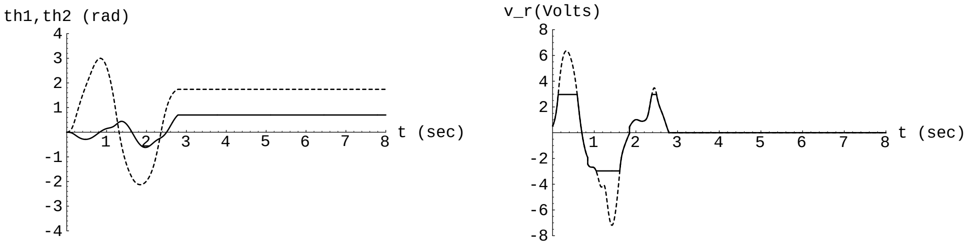

Figure 50: Simulation results of faster swing up behavior ($K_e = 0.9$) considering unmodelled characteristics of the actuator. Left: Joint trajectories (θ_1: solid, θ_2: dashed). Right: Voltage command to the motor driver. Solid line corresponds to the produced torque by the actuator considering saturation. Dashed line denotes the voltage command.The numerical simulations closely match the corresponding experimental results.

[17] Fuminori Saito, *Motion Control of the Brachiation Type of Mobile Robot*, Ph.D. thesis, Nagoya University, Mar. 1995, (in Japanese).

[18] Ralph Abraham and Jerrold E. Marsden, *Foundations of Mechanics*, The Benjamin/Cummings Publishing Company, Inc, 1978.

[19] Timothy D. Tuttle and Warren P. Seering, “A nonliear model of a harmonic drive gear transmission,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 368–374, June 1996.

[20] S. Eimerl and I. DeVore, *The Primates*, TIME-LIFE BOOKS, 1966.

[21] H. Preuschoft and B. Demes, “Biomechanics of brachiation,” in *The Lesser Apes*, H. Preuschoft, D. J. Chivers, W. Y. Brockelman, and N. Creel, Eds., pp. 96–118. Edinburgh University Press, 1984.

[22] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, 1980.

[23] Jean-Jacques E. Slotine and Weiping Li, *Applied Nonlinear Control*, Prentice-Hall, 1991.

[24] Suguru Arimoto, *Control Theory of Non-linear Mechanical Systems —A Passivity-based and Circuit-theoretic Approach*, Oxford University Press, 1996.

[25] T. Marilier and J. A. Richard, “Non-linear mechanic and electric behavior of a robot axis with a ’harmonic-drive’ gear,” *Robot. Computer Integrated Manufact.*, vol. 5, no. 2–3, pp. 129–136, 1989.

List of Figures

List of Tables