Fall 12-21-2016

# A locomotion control algorithm for robotic linkage systems

Jeffrey L. Dohner Dr.
*University of New Mexico*

Jeffrey L. Dohner

*Candidate*

Electrical & Computer Engineering (EC)

*Department*

This dissertation is approved, and it is acceptable in quality and form for publication:

*Approved by the Dissertation Committee:*

Prof. Chaouki Abdallah , Chairperson

Prof. Rafael Fierro

Prof. Meeko Mitsuko Karen Oishi

Prof. Frank Lewis

INTENTIONALLY LEFT BLANK

# A Locomotion Control Algorithm for Robotic Linkage Systems

by

**Jeffrey L. Dohner**

B.Sc., Mechanical Engineering, Purdue University, 1981
M.S., Mechanical Engineering, Massachusetts Institute of Technology, 1983
Ph.D, Mechanical Engineering, Purdue University, 1987
Professional Engineer, New Mexico, 2003, License No.: 16016
M.S., Electrical Engineering, University of New Mexico, 2010

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Engineering

The University of New Mexico
Albuquerque, New Mexico
May, 2017

# Dedication

To my mother Lena and my wife Rebeca

INTENTIONALLY LEFT BLANK

# Acknowledgments

Sixteen years ago, I started taking a few classes to gain a better understanding of electronics. It's something that I always wanted to do. Little did I realize that I would end up following a path of learning that would take me through another Master's degree followed by another Ph.D. It was a wonderful journey, but what was the most enjoyable were the beautiful and fascinating people that I met along the way. They will always be close to me, even though some are no longer in this world.

I would like to start by thanking my committee, Professors Fierro, Oishi, and Lewis. They were there when I needed to see clearer skies through darker days.

Special thank goes to my thesis advisor, Professor Abdallah. He was a proverbial angel in professorial disguise. Over these extended years, I saw his concern for his students; his unstoppable desire to build a stronger university; and an integrity grounded in life's experiences. I am eternally gratefully that when I asked him to be my advisor, he said yes.

Thanks goes to a number of notable professors. With fond memories there is Professor Hayat, Gilmore, Santhanam, and Schamiloglu. Thank you. Your lives have touched so many. A special thanks goes to Professor Peter Dorato. I knew him for such a short time, but his insight and his kindness not only touched me but touched a circle of people that encompasses the planet (not once, but many times over).

I would also like to thank Sandia Laboratories for funding my education through the tuition assistance program and my current manager, Dr. Michael Pasik. My life at Sandia has been somewhat of a roller coaster ride; however, with Michael, the bumps have not been quite as harsh and the dips have not been quite as deep. He has been a great manager.

And, then there is my wife, Rebeca. Thank you for being there. Without you, the shinning sun, the rising moon, and glimmering stars have no meaning

INTENTIONALLY LEFT BLANK

# A Locomotion Control Algorithm for Robotic Linkage Systems

by

**Jeffrey L. Dohner**

B.Sc., Mechanical Engineering, Purdue University, 1981
M.S., Mechanical Engineering, Massachusetts Institute of Technology, 1983
Ph.D, Mechanical Engineering, Purdue University, 1987
Professional Engineer, New Mexico, 2003, License No.: 16016
M.S., Electrical Engineering, University of New Mexico, 2010
Ph.D, Electrical Engineering, University of New Mexico, 2017

## Abstract

This dissertation describes the development of a control algorithm that transitions a robotic linkage system between stabilized states producing responsive locomotion. The developed algorithm is demonstrated using a simple robotic construction consisting of a few links with actuation and sensing at each joint. Numerical and experimental validation is presented.

In this algorithm, transitioning excitations, called rhythms, are formulated using terminal state control solutions. Rhythms are constrained to be low order parameterized functions allowing for the optimal control problem to be replaced by a parametric optimization problem with a limited set of easily solved unknowns.

This algorithm is developed and demonstrated using a simply linkage system consisting of only two links joined at a knee, called a two-link. The two-link can exist in a number of different

configurations of various dynamic orders. The highest order configuration occurs when the two-link is in flight. Transition from a higher order to a lower order configuration occurs when a link impacts or parts from the ground. A “no rebound” condition at the point of impact is assumed. An increase in order occurs as the result of a link parting from the ground such as would occur in the presence of control actuation.

Rhythms can be cascaded together using state transition logic to produce locomotion. Sequential two-link rhythms to stand, crouch, and continuous hop produce responsive locomotion. Uncertainty reduction and controllability is not continuous but intermittent, adding to the complexity of the control problem.

The derived algorithm was validated by hardware implementation. Solid models of two-link parts were generated in SolidWorks and printed using an Ultimaker 2+ 3D printer. Printed solids were assembled with mechanical and electrical substructure to produce instrumentality. Control was hierarchical. High level communications were transmitted via an $I^2C$ interface from a Raspberry Pi 3 model B microcomputer to A/D and D/A converters while low level communication within the two-link occurred between mid-range microchip PIC processors and motor drivers. Rhythms that allowed the two-link to stand, crouch, hop and lie down were programmed in Raspberry Pi software.

# Contents

# Table of Figures

# Table of Tables



# Table of Problems

INTENTIONALLY LEFT BLANK

# Acronyms

| | |
|---|---|
| ACM- | Adaptive Control Method |
| AM- | Additive Manufactured |
| BIBO- | Bounded Input/Bounded Output stability |
| CBM- | Control Based Method |
| COM- | Center Of Mass |
| DOPC- | Discrete Optimization with Constraints |
| EL- | Euler Lagrange (equations) |
| EMF | Electro Magnetic Force |
| FD- | Forward Dynamics |
| GA- | Genetic Algorithm |
| GCIPM- | Gravity Compensated Inverted Pendulum Mode |
| GPIO- | General Purpose Input/Output |
| HCM- | Hopping Control Methods |
| HOT- | Higher Order Terms |
| $I^2C$- | Inter-Integrated Circuit Communication |
| ID- | Inverse Dynamics |
| IPM- | Inverted Pendulum Mode |
| LHP- | Left Hand Plane |
| LQR- | Linear Quadratic Regulator |
| NE- | Newton Euler |
| OBM- | Optimization Base Method |
| OLC- | Open-Loop Control |
| OLOC- | Open-Loop Optimal Control |
| PD- | Passive Dynamics |
| PID- | Proportional, Integral, Derivative |
| PWM- | Pulse Width Modulation |
| Raspi- | Raspberry Pi |
| RBNN- | Radial Basis Neural Network |
| RHP- | Right Hand Plane |
| RO- | Random Optimization |
| STL- | Stereo Lithography |

| | |
|---|---|
| TFEM- | Temporal Finite Element Method |
| TIG- | Tungsten Inert Gas |
| TMIPM- | Two Mass Inverted Pendulum Mode |
| TPBVP- | Two Point Boundary Value Problem |
| VCM- | Virtual Constraints Method |
| ZDM- | Zero-Dynamics Method |
| ZMP- | Zero Moment Point |

# Rhythms

| | |
|---|---|
| s2cr- | Standing To Crouching Right |
| s2cl- | Standing To Crouching Left |
| cr2s- | Crouching Right To Standing |
| cl2s- | Crouching Left To Standing |
| j2cr- | Crouching Right Jumping to Crouching Right |
| j2cl- | Crouching Left Jumping to Crouching Left |
| p2t- | Prone to Transition |
| t2cr- | Transition to Prone |
| p2cr- | Prone to Crouching Right |
| p2cl- | Prone to Crouching Left |
| cr2p- | Crouching Right to Prone |
| cr2t- | Crouching Right to Transition |
| t2cr- | Transition to Crouching Right |
| cl2p- | Crouching Left to Prone |
| bals- | BALance Standing |
| blcr- | BaLance Crouching Right |
| blcl- | BaLance Crouching Left |

INTENTIONALLY LEFT BLANK

# Chapter 1

## 1 Introduction

This dissertation describes the development of a control algorithm that, given a set of connected linkages with actuators and sensors at every joint, uses a set of open loop control inputs, called rhythms, to transition between or through stabilized states resulting in locomotion. These rhythms are simple discontinuous functions of low parametric order.

### 1.1 Motivation

The economic incentive for a replacement for the human in high risk environments and in the menial work environment has driven the search for a viable level of cognition integrated with dexterous instrumentality [1, 2, 3, 4]. Qureshi and Syed [1] discuss the inevitability of robotics in the work force with special focus on the health care industry [5]. They cite the success that robotics has played already in a number of industries [6,7], but warned that robotic integration must be balanced by maintaining employee motivation. Hanson [3] gives a long-term perspective of the benefits of machine intelligence and its relationship to wages, and Laird [4] gives a concise overview of the need for unmanned systems in the armed forces. These articles infer that the past, present, and future impact of bipedal robotics in industry and the service sectors has been, is, and will be significant. Our motivation for this work is to improve the knowledge basis required to bring such a technology development to the market place by developing a control algorithm which can be used to mimic anthropomorphic behavior.

### 1.2 Literature review

A number of review articles on the development of bipedal robotic control exist [8, 9, 10, 11]. Although these reviews are extensive, they are also limited due to the overwhelming amount of information in the area. Likewise, due to the extended nature of the field, an all-encompassing review is not attempted here. The following focuses only on those articles with the greatest relevance to the remaining sections of the dissertation.

With some variation, most review articles divide biped control methods into six or seven general categories. These categories are:

- Inverted Pendulum Methods (IPM));
- Passive Dynamics (PD) walking methods;
- Zero-Moment Point (ZMP) methods;
- Optimization-Based Methods (OBM)
- Hopper Control Methods (HCM)
- Virtual Model Control (VMC) and
- Control Theoretical Methods (CTM).

Each of these categories is discussed below.

### 1.2.1 Inverted Pendulum Mode (IPM) methods

The inverted pendulum mode (IPM) method [12, 13, 14, 15, 16, 17, 18, 19] models biped dynamics as an inverted pendulum with a rigid connection between a planted pivot foot and the center of mass (COM). Kajita, Yamaura, and Kobayashi [12] are often cited as the first to implement the inverted pendulum method in hardware and to prove the utility of the method for walking over a flat surface. They derived a control scheme which maintained the COM at a constant height above the surface for a robot with low mass legs. Later Kajita and Tani [13] showed through simulation that the IPM method can be applied to walking over uneven surfaces as well. Kajita, Matsumoto and Muneharu [14] expanded the concept further by deriving the equations of motion and a control algorithm for a three dimensional system. As shown in Figure 1, they considered that all of the mass of the biped was concentrated at the COM and that the planted foot acted as a pivot point. The projection of the COM on the ground produced a trajectory that was initially moving towards the pivot foot. Without enough energy to overcome gravity this trajectory is pushed away from the planted foot resulting in a repulsive orbital like response. At the proper time, the planted foot is

switched (variable structure control [59]) and the process continues. Kajita et. al. [15] took this work even future by developing an automatic pattern generator for inverted pendulum walking.

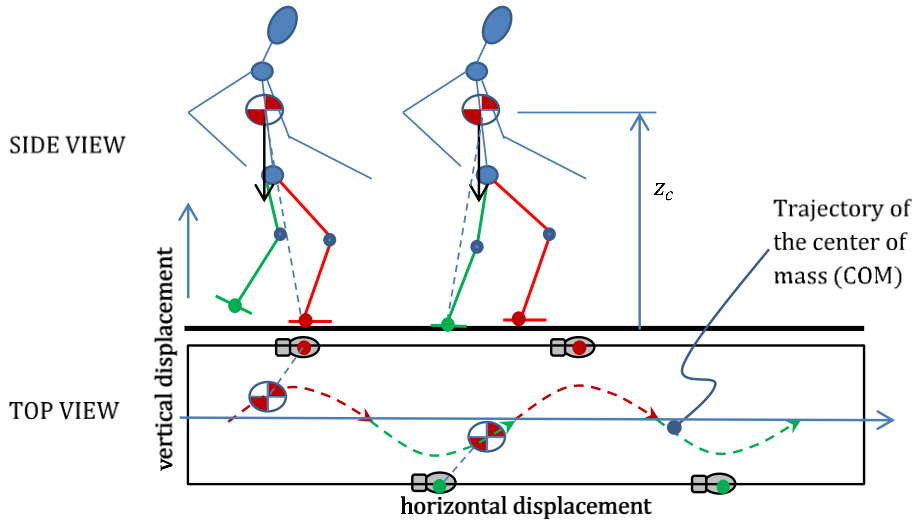



**Figure 1. An illustration of the inverted pendulum mode**

Park and Kim [16] expanded the IPM method by adding swing leg dynamics. As part of their derivation they assumed that inertial forces were small but included gravitational forces. Their approach is referred to as the Gravity-Compensated Inverted Pendulum Mode (GCIPM). Albert and Gerth [17] expanded upon the GCIPM method by including the neglected inertial swing leg forces. Their method is referred to as the Two Masses Inverted Pendulum Mode (TMIPM). Luo, Sheng and Chang [18] included the effect of impacts.

Over the limited time between foot impacts, the response of the IPM system is close to marginally stability. For the situation where there is no error, the limit cycle can be stabilized by switching feet (i.e. variable structure control) at the proper time. Nevertheless, for the situation where there is error, without compensation for this error, uncertainty grows. Figure 2 shows the response of an IPM system for the set of given parameters. This figure shows the response of a system with no uncertainty (i.e. $\Delta = 0.00$); the response of the system with a very small amount of uncertainty (i.e. $\Delta = 0.01$); and the response of a system will the same amount of uncertainty but with a limited

level of feedback torque from the ankle. With only a small amount of additive uncertainty and no feedback, the final variation in the response after one stride is over half the amplitude of the sway. Nevertheless, as shown below, feedback can be used to mitigate most of this deviation.

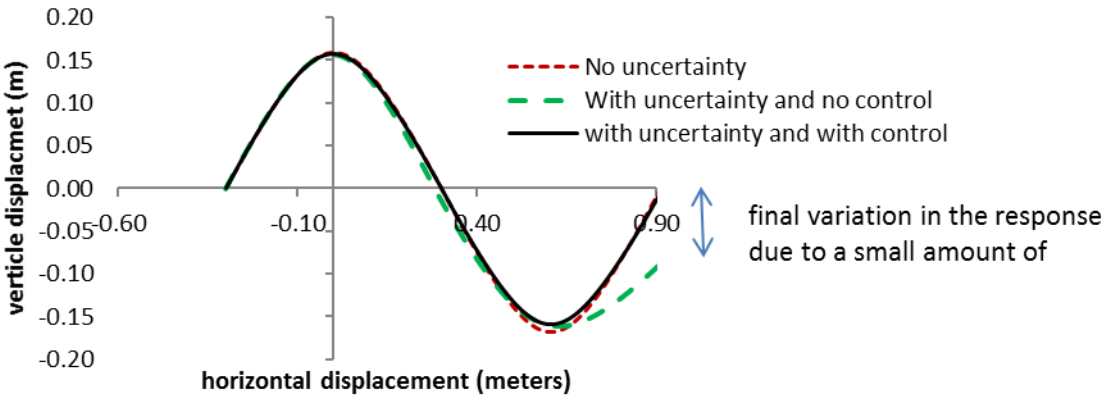



**Figure 2. An illustration of IPM dynamics with and without and feedback**

Feedback can be implemented a number of different ways. One way is by adding a foot. Figure 3 shows an inverted pendulum with a foot of given width. Under the condition of additive uncertainty in (1.1b), the center of pressure (COP) on the foot is moved from the nominal location to an inside location. This produces a restoring force on the inverted pendulum. This restoring force compensates for model uncertainty but only up to the point where the COP is at the foot's edge [19].

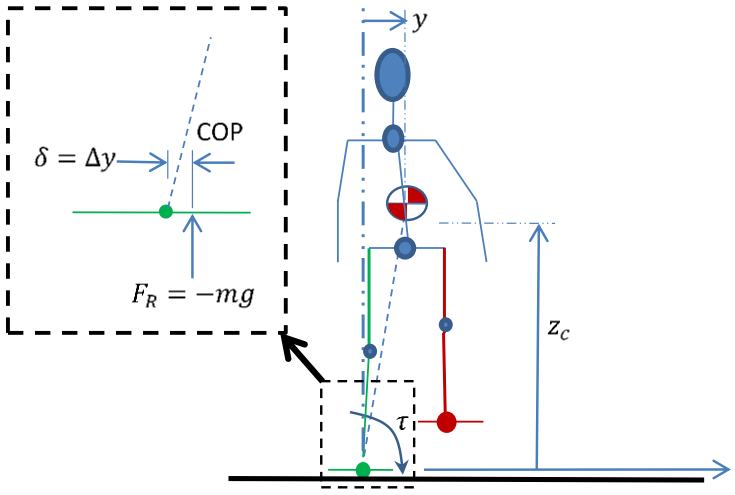



**Figure 3. An illustration of stabilization using a foot**

The major limitation with the IPM is that one foot must always be planted. A biped using this control method of control cannot jump, hop or skip. It is limited to a rather unnatural tottering form of walking.

### 1.2.2 Passive Dynamics (PD) methods

In passive dynamics (PD) [20, 21, 22, 23, 24, 25] there are no internal energy sources which produce locomotion. All energy is derived from a loss of gravitational potential. Mochon and McMahon [20] are often cited as the first to develop a mature mathematical representation of passive walking. The first hardware implementation of this method is by McGeer [21]. Mochon and McMahon termed this type of walking "ballistic walking." McGeer coined the phrase for which is known today, "passive walking".

Passive walking is of interest considering that it represents a form of locomotion that requires very little energy. McGeer claimed over an order of magnitude reduction in energy per step using a PD biped compared to an active system. For this reason, passive gaits have been studied and analyzed in detail. Goswami, Espiau and Keraman [22] analyzed the gait of a passive biped, its limit cycle characteristics and the limited energy required for it to walk. Lui, Tian et. al. [23] expanded upon Goswami, Espiau and Keramans' work by including knee motion and by using a more detailed impact model.

The number of gaits that can be produced by the same PD walker has been shown to be numerous. Borzova and Hurmuzlu [24] studied three separate gaits and showed bifurcations using a Poincare map. They suggested the integration of dampers and springs to produce a more predictable response. In general, PD walkers can produce a rich set of responses, even to the point of chaos.

Although PD walkers are even more limited than IPM bipeds, their gaits are of great interest to researchers since they represent a highly efficient method of locomotion. We take from this review an understanding of the need for a minimal energy solution. We also recognize the need for some representation of impact dynamics [25] – a recurrent theme throughout this literature review.

### 1.2.3 Zero-Moment Point (ZMP) methods

Zero-moment point (ZMP) methods [26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39] have been successfully implemented in hardware [40, 41] and have set the ground work for many future developments. The zero-moment point (ZMP) is the location on the ground where the sum of the inertial and gravitational moments is zero (hence the name, ZMP). It has come to serve as a stability condition for many bipedal algorithms.

Figure 4 illustrates a stable and an unstable system using ZMP when inertial forces are small (i.e. the biped is moving slowly). When the ZMP resides in the area between and including the feet, the system is stable. When the ZMP is outside of this region, the biped is unstable. In this regard, the bipedal structure and the feet behave much like a table which supports its COM. If the COM is too far in one direction or the other the table tips over.

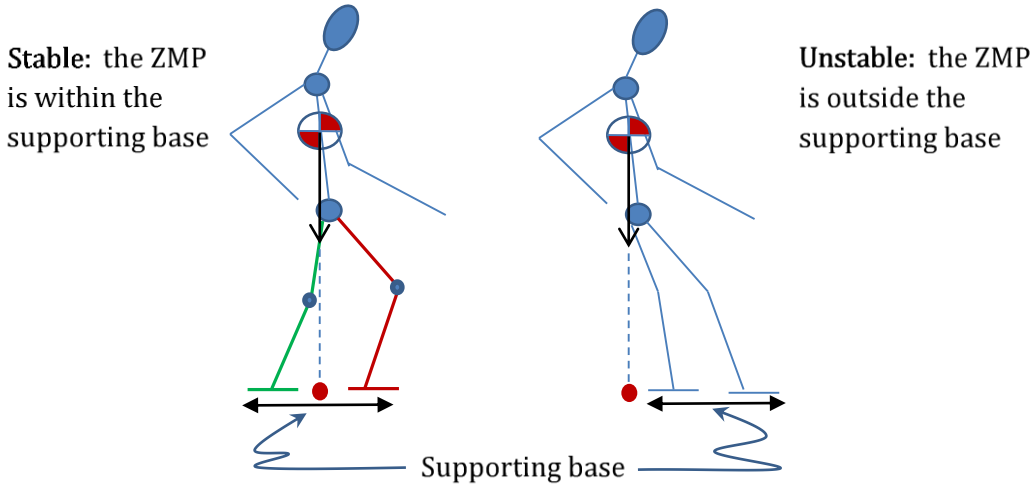



**Figure 4. An illustration of stabilization using the zero moment point method**

To understand how the ZMP is used to produce walking in its simplest form, consider the slow moving biped in Figure 5. As shown in this figure, the biped starts with the ZMP between it two feet. This is a stable configuration. The COM is shifted such that the ZMP is moved to below one foot. At this time, the other foot can be lifted off the ground. Lifting this foot reduces the span of the supporting base. The leg that the lifted foot is attached to is called the swing leg. The swing leg can

be brought forward; however, while doing this, the stance of the rest of the biped must be adjusted such that the COM produces a ZMP that is within the support of the planted foot. As the swing leg is brought forward, the ZMP moves from the heel to the toes of the foot. By the time in which the ZMP has moved to the tip of the toes, the swing leg foot must be planted. Planting the swing leg foot increases the span of the supporting base and the ZMP can be moved to its center. The process is then repeated. At all times, the biped is completely controllable.

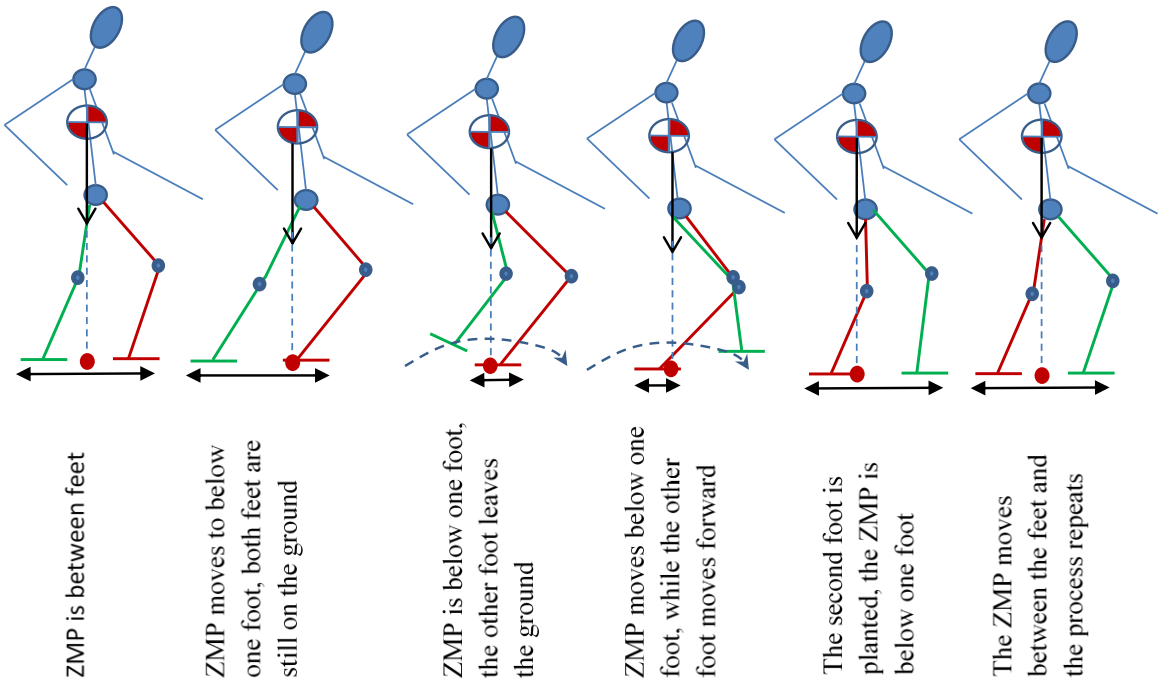



**Figure 5. An illustration of walking using the zero moment point method**

The difficulty with using the ZMP is in finding a gait that will move the COM in such a way that the ZMP moves in a coordinated fashion with the feet such that stability is always maintained. This can be done quasi-statically as shown in the figure; however, this is difficult to do dynamically. To limit complexity, inertial effects can be reduced; however, this results in a *very* slow moving biped.

Vukobratoić [26, 27, 28] is most often cited with the development of the ZMP method. The number of papers referencing his work is extensive making a full review of this rich area of research beyond the scope of this discussion. For the interested reader a notable application of the ZMP method is

given in the 1996 paper by Yamaguchi et. al. [29] who present both theoretical and experimental results for a WABIAN biped. Yamaguchi et. al discuss the complete process used to develop a walking robot including modeling; the derivation of the ZMP equations; and the computation of motion. Inertial forces are included within their analysis – impacts are not.

The process of determining bipedal motion relative to the ZMP constraint is called path planning. In earlier papers path planning was determined from start to finish with little ability to adapt. However, more recent ZMP work has focused on improving the ability for the biped to adapt [30, 31, 32, 33, 34, 35, 36, 37]. Uncertainty in modeled dynamics has also gained some attention [38, 39].

As with the IPM method, the ZMP method requires that one foot always be planted resulting in somewhat unnatural motion. Minimum energy solutions are not a necessary aspect of the method; however, coupled with optimization methods, they give unique solutions out of a plethora of possible trajectories.

### 1.2.4 Optimization Based Methods (OBM)

Most optimization based methods (OBMs) [42, 43, 44, 45, 46] were developed for biomedical applications. This synergistic field of research focuses on mimicking the gait of humans for the purpose of improving the well-being of patients. OBMs fall into two categories:

*Inverse dynamics (ID):* In this situation the gait is measured and is imposed upon a model by determining control inputs that minimize the error between the gait of the model and the measured gait. The optimized model is then used to determine internal forces.

*Forward dynamics (FD):* In this situation optimal control input are applied to the model and the resulting gait is examined. These inputs are determined using a terminal state condition along with the minimization of an integral optimality condition such as minimum energy. The resulting model gait and measured gaits are then compared.

In this paper we are primarily interested in the second category of optimization methods – forward dynamics (FD). This category involves the solution of a terminal constraint, open-loop optimal control problem with integral weighting. In this regard, a FD OBM is also a control theoretic method (CTM).

In recent years FD OBMs have produced impressive results. Eriksson [42] formulated the optimal control problem for linkages using temporal finite element (TFEM) with Hermitian shape functions. His formulation reduces the temporal optimization problem into a nonlinear parameter optimization problem which can solved using a number of commercial global optimization packages [55]. Kaphle and Eriksson [43] went on to use this method to solve for the control inputs for a jumping three-link system and Eriksson and Nordmar [44] expanded upon Eriksson's original work by showing how the original formation could be derived from a predecessor to the Euler Lagrange (EL) equations [47]. Kwon et. al. showed how Eriksson's formulation could be used to control a (backwards) walking biped. Nevertheless, in order for Kwon et. al. to maintain stability they had to use the ZMP method. That is, one foot had to always be on the ground. Similar stability limitations exist for other OMB solutions [46].

A FD OBM can be used to solve for an open-loop optimal control (OLOC) solution [48]. However, using a FD OBM to solve an OLOC problem, although initially attractive, is no panacea for success. FB OBMs require the solution of a large set of nonlinear algebraic equations which may have one, many or no solutions. The FD OBM relies heavily on global search methods to obtain a solution. In many cases this requires a very good initial guess.

### 1.2.5 Hopper Control Methods (HCM)

The ZMP method requires complete controllability at all times. That is, at any time, joint actuators can drive the state of the system to any desired state. In comparison, PD's do not have actuators and therefore, their motion is uncontrolled but stable limit cycle motion. Hoppers [49, 50, 51], like PD's, produce stable limit cycles. Nevertheless, unlike PD's, they can actively change direction and speed.

Raibert [49, 50, 51] is often cited as the initial pioneer of the HCM method. Raibert and Brown [49] produced a two dimensional hopper actuated by pneumatics as early as 1984. They controlled the speed of the hopper by altering the impact location of the foot. Figure 6 shows some of the nomenclature applied to their hopper. The hopper can be in two configurations – a stance configuration where the foot is in contact with the ground and a flight configuration where the foot is off the ground and the hopper is in flight. Figure 6 shows the hopper at the moment of contact with the ground. The hopper enters the stance configuration with the leg at an angle $\theta_1$ and leaves the stance configuration with the leg at an angle $-\theta_1$. The velocity of the hopper in the horizontal direction is $\dot{x}$ and therefore, the distance between the foot and the COM at the time of impact is $\frac{\dot{x}T_s}{2}$ where $T_s$ is the total time in which the hopper is in the stance configuration. The simple proportional control law

$$\Delta x = \frac{\dot{x}T_s}{2} + K(\dot{x} - \dot{x}_d) \tag{1.2}$$

was used to speed the hopper up or to slow it down where $\dot{x}_d$ is the desired speed.

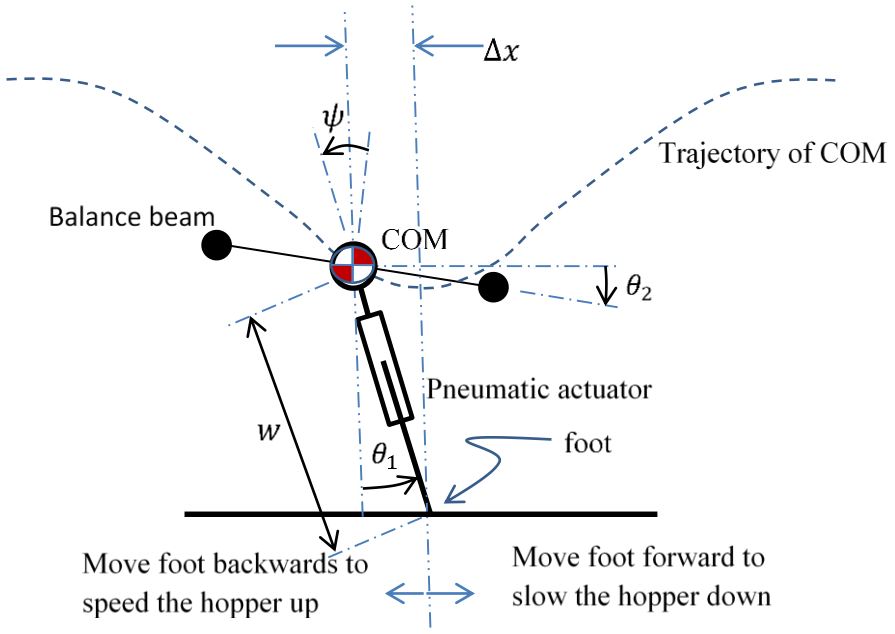



**Figure 6. An illustration of the Raibert and Brown hopper**

If the foot is forward from its expected impact location, the hopper slows down. If the foot is behind this location, the hopper speeds up. To maintain speed, the feedback gain, $K(\dot{x} - \dot{x}_d)$, increases or decreases speed until the desired speed is obtained.

The Raibert and Brown hopper is a system with different levels of controllability. The orientation of the leg is controllable during flight but is uncontrollable during stance. Vertical displacement is uncontrollable during flight but is controllable during stance, and horizontal displacement is uncontrollable during stance and flight; nevertheless, due to coupling between stance and flight dynamics, the limit cycle which drives horizontal displacement is controllable.

Raibert went on to develop a three dimensional hopper [50, 51] based upon his two dimensional efforts. In more recent years, Hyon and Emura have proposed an energy efficient biped hopper [52]. In their formulation, legs containing springs that store and return energy to the system are employed. Long Murphey and Lynch [53] developed a control system for a hopper that climbs by transitioning between stationary walls.

Raibert's hoppers are often cited as successful examples of locomotion producing limit cycle robots. Nevertheless, they are limited in ways that most other walking robots are not in that they cannot simply stand. That is, they must continuously be hopping.

### 1.2.6 Virtual Model Control (VMC)

If a biped were rigidly supported at the hips by a moving table, the resulting table would produce forces and moments that would maintain stability. As the table is moved, these forces and moments change. In the virtual model control (VMC) method, the force and moments applied to the legs are chosen to be from such a virtual support. This is illustrated in Figure 7.

Pratt et. al. [54] showed how these virtual forces can be related back to the forces and moments applied to the legs using the method of virtual work. They went on to show how this method of control could be used to produce locomotion over both flat and rough surfaces. As with the IPM, PD, and ZMP method, a planted foot is always required.

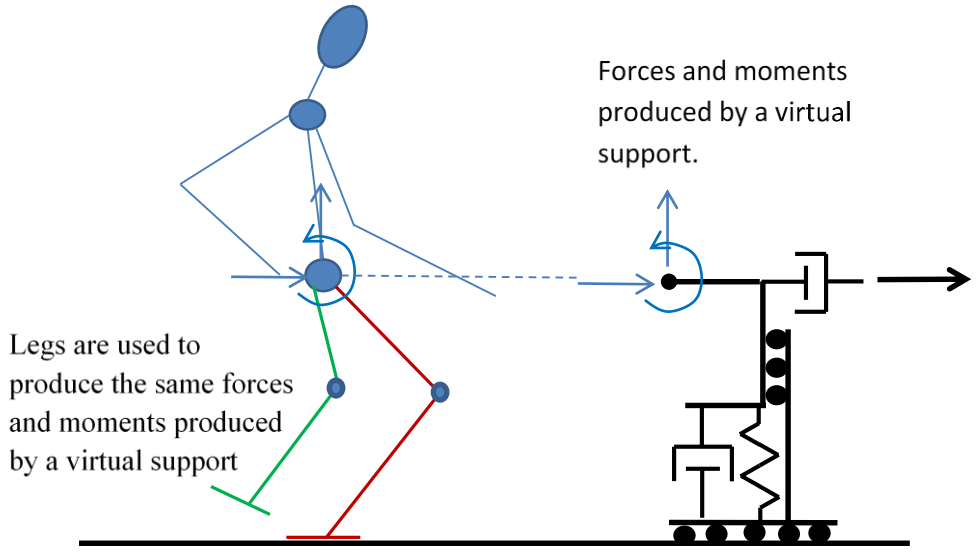



**Figure 7. An illustration of virtual model control**

### 1.2.7 Control Theoretic Methods (CTM)

CTMs are methods that are more closely correlated with the control theoretic literature. They include the use of open-loop optimal control (OLOC) and the zero dynamic method (ZDM). These and adaptive control approaches are discussed below.

#### 1.2.7.1 Open-Loop Optimal Control (OLOC)

Open-loop optimal control (OLOC) is a CTM which may be used to produce open loop control inputs for terminal constraints and the minimization of an integral performance metric. As stated above, Eriksson transformed the biped OLOC into a FD OBM which he solved using a commercial nonlinear programming package [55]. The logic for such a transformation is that the OLOC formulation results in a two-point boundary-value problem (TPBVP) which has historically also been difficult to solve. Some TPBVPs can be solved using a shooting method [48, 56, 57] however for large-order systems and extended simulation times this requires a very good initial estimate of the solution [56]. Bryson [57] offers several solutions for solving the OLOC problem which do not require parameter optimization or a shooting method. He presents several improved gradient solutions

with dynamic constraints enforced through the use of Lagrange multipliers. Although an improvement over past methods, his methods are still limited to relatively simple dynamics.

#### 1.2.7.2 Zero-Dynamics (input-output linearization) Method (ZDM)

Other control-based methods have been presented with promising results. E. Westervelt et. al. [ 58] produced bipedal walking simulations through the use of input-output linearization [59]. Using this method, nonlinear dynamics are subtracted from the dynamical system through feedback reducing the system to a linear one. A second loop is then used to produce command following.

Following Grizzle et. at. [8, 58] and Khalil [59], the dynamical linkage system with no impacts can be represented in first-order affine form by

$$\dot{\vec{X}} = f(\vec{X}) + g(\vec{X})\vec{u} \tag{1.2a}$$

$$\vec{y} = h(\vec{X}) \tag{1.2b}$$

where $\vec{X}$ is a state vector and $\vec{y}$ are a set of outputs. The total time derivative of the output is taken until the control shows up in the output. That is, for the first derivative

$$\dot{\vec{y}} = \frac{\partial h(\vec{X})}{\partial \vec{X}} \dot{\vec{X}} = \frac{\partial h(\vec{X})}{\partial \vec{X}} \{f(\vec{X}) + g(\vec{X})\vec{u}\}. \tag{1.2c}$$

however if the time derivative of the output is not a function of the control (i.e. $\dot{\vec{y}} \neq \dot{\vec{y}}(\vec{u})$) then

$$\ddot{\vec{y}} = \frac{\partial h(\vec{X})}{\partial \vec{X}} \vec{X} = \frac{\partial h(\vec{X})}{\partial \vec{X}} f(\vec{X}). \tag{1.2d}$$

and the process is repeated until $\vec{y}^p = \vec{y}^p(\vec{u})$ where $p$ is called the relative degree of the system. The derivatives in (1.2c) are Lie derivatives [59] with the special notation

$$L_f h(\vec{X}) = \frac{\partial h(\vec{X})}{\partial \vec{X}} f(\vec{X}) \text{ and } L_g h(\vec{X}) = \frac{\partial h(\vec{X})}{\partial \vec{X}} g(\vec{X}).$$

Following this notion, taking Lie derivatives up until the relative degree gives

$$\vec{y}^p = {L_f}^p h(\vec{X}) + {L_g}^p \{{L_f}^{p-1} h(\vec{X})\}\vec{u}. \tag{1.2e}$$

The control input is then represented as the sum of two inputs – one used to negate nonlinear dynamics and the other to stabilize the output. That is

$$\vec{u} = \vec{\gamma} + \vec{v}. \tag{1.2f}$$

If we let

$$\vec{\gamma} = -\left[L_g{}^{p}\{L_f{}^{p-1}h(\vec{X})\}\right]^{-1} L_f{}^{p}\, h(\vec{X}) \tag{1.2g}$$

then (1.2e) becomes

$$\vec{y}^{p} = \vec{v}, \tag{1.2h}$$

a simple linear system with all of its poles at the origin. The control input $\vec{v}$ can then be chosen such that $\vec{y}(t)$ is (asymptotically stable).

Using the above mathematical formulation, a biped can be held to a desired trajectory $\vec{y}_d(t)$ by letting

$$\vec{e} = \vec{y}(t) - \vec{y}_d(t) \tag{1.2i}$$

where $\vec{y}_d(t)$ is the desire output. That is, the above control approach is one of input-output linearization followed by command following. The functions $\vec{y}_d(t)$ are chosen to produce a limit cycle for locomotion through an optimization process. In the work by Grizzle et. al. these functions are parameterized using B*é*zier polynomials [60].

Grizzle et. al. notes one issue of concern – impact dynamics [61]. The above formulation does not take into account the biped striking the ground. In general, the impact condition is stated as

$$\vec{X}^{+} = \Delta(\vec{X}^{-}) \tag{1.2j}$$

where $\vec{X}^{+}$ are the states after the impact and $\vec{X}^{-}$ are the states before the impact. The function $\Delta$ maps the pre-impact state to the post-impact state. A ZDM does not consider these dynamics.

The above method also assumes that $\left[L_g{}^{p}\{L_f{}^{p-1}h(\vec{X})\}\right]$ has an inverse in (1.2g). This is a controllability condition. This condition is valid for a robot that has complete controllability, but, when in flight, this controllability is lost.

Another issue is uncertainty. Control is based upon a model, but considering that no model is perfect, perfect linearization can never occur. The question then becomes how good of a model is really needed? The literature tends to disagree on this issue with IPM researchers using very simple

dynamical models and ZDM researchers using very complex representations even to the point of requiring the use of symbolic manipulation [58].

#### 1.2.7.3 Adaptive Control Methods (ACM)

Adaptive control is often characterized in terms of direct adaptive control and indirect adaptive control [62]. Direct adaptive control performs adaptation of the controller in the feedforward or the feedback paths without the need to produce a model of the system. Indirect adaptive control uses a realization of system produced directly from experimental data which can then be used to develop a controller off-line.

Direct adaptive approaches have been successfully applied to nonlinear robotic systems [63]; however, they are limited to systems (or subsystems) which are fully controllable. There seems to be no limitation as to their use on the controllable portion of a bipedal system; however, there is no evidence of their use in the literature to date.

An indirect adaptive approach does not have any of the above limitations; however, it requires a nonlinear realization of the system. It, in effect, is simply an OLOC solution using an experimentally derived model. The use of these methods has also been limited.

### 1.2.8 Conclusions from literature review

Neglecting model development, the underlying issue that emerges from the literature is the lack of OLOC solutions for linkage systems. If an OLOC solution existed, why use an IPM, VMC or ZMP? An optimal control solution could produce similar results with additional optimality constraints in energy or time – a relevant requirement as highlighted in the PD literature. Even the impact condition reviewed in the above ZDM discussion could be included within an optimal control framework since there are no limitations as to the use of this method across a discontinuity (see appendix C for the general OLOC solution with discontinuities). This leads to the below question:

Question: Why are OLOCs not more frequently applied to linkage system?

From the literature review, we deduce the follow answer.

> Answer: Although OLOC necessary conditions are relatively easy to derive, finding numerical solutions that satisfy these conditions are not.

Due to the lack of ability to solve the OLOC problem using shooting methods, Bryson spent years developing gradient methods [56, 57]; however, even with these years of development, the complexity of the problem that he could solve was limited. In these formulations the control input was discretized allowing for the calculation of permutations that would move the performance metric toward a minimum. The temporal finite element formulation by Eriksson [42, 44] is simply another type of discretization. Like Bryson's solutions, due to the number of unknowns (on the order of 100's to 10,000's), it is difficult to find a solution in a prodigious number of local minimum.

### 1.3 Approach

Our technical approach to the above dilemma is to solve a simpler problem. Depending on the physics, a large number of unknown may not be needed to describe the control input. In this dissertation, it will be shown that the dynamics for which we are concerned is simple enough that the control input can be described in terms of simple functions defined using just a few parameters. In essence this approach is the same as the FD OBM developed by Eriksson; however instead of using a prodigious number of unknowns, we rely on just a few (less than 10). This insight allows for the development of open loop control (OLC) solutions that can be found using a limited search. It will be shown that some of these solutions are even near to the minimum energy optimum.

This simplification is a novel improvement over existing pedal control methods allowing for numerically tractable OLC solutions. We will then use these solutions, called rhythms, with state transition logic to produce responsive robotic locomotion.

## 1.4 Overview of thesis

In chapter 1, the literature is reviewed and the thesis approach is presented. The underlying approach is to solve for control input described using just a few parameters resulting in numerical tractable problems. This assumption is physics dependent.

In chapter 2 linkage dynamics are investigated. The dynamics of a simple system are derived for a number of different configurations. For some configurations, these dynamics will be fully controllable. For others, they will only be stabilizable. It will be shown that the dynamics of the highest order configuration contains the dynamics of all lower order configurations where reduction in model order can be imposed through the use of an equality constraint and impact dynamics. An increase in model order can be imposed through the removal of this constraint and by imposing continuity in the states. Impact dynamics are derived using Newton's third law under the assumption that there is no rebound at the point of impact.

In chapter 3 a set of OLC problems will be solved. The solution to these problems is referred to as rhythms. These solutions are based upon the idea that the control input can be represented by a set of parameterized functions that can be solved for using a global search over a limited parameter space.

In chapter 4 rhythms developed in chapter 3 will be integration into state transition logic to produce robotic locomotion. A number of examples are presented.

In chapter 5 the hardware construction of a robotic system is presented. This robotic this use to validate the control algorithm through the implementation of rhythms. Electrical and mechanical hardware detail will be presented. Mechanical detail includes 3D print of structural parts and the used of welded substructure. Electrical detail includes the description of two electrical subsystems – a high power system to drive the motors and the lower power system to control the motor drives.

Chapter 6 contains are discussion of results. In this section the number of unknown used to define a rhythm will be reduced through the use of virtual constrains. This will allow for the experimental determination of rhythms without the use of a numerical model.

Chapter 7 contains conclusions and proposed future work.

# Chapter 2

## 2.0 Dynamics

In this chapter robot dynamics are investigated. In general, robot dynamics are nonlinear, rigid body dynamics. This means that the vibratory response of linkages can be neglected or is included within other representation of physics such as impact losses.

Of special importance to the bipedal robot problem are changes in state order due to changes in configuration. Transitions between configurations occur due to links impacting or parting from ground. Impacts are represented as impulsive loadings. Impacts can take different forms. They may be elastic in which all energy is conserved or plastic in which all vibratory energy is lost. They can also be defined relative to the resulting boundary constraint. In this dissertation it is assumed that after impact, the point of impact remains in contact with the ground. This implies that there is enough energy absorption with the structure such that the normal velocity of the impact location goes to zero. To do this we part from a more traditional virtual work formation and used Newton's third law integrated in time.

The purpose of this chapter is to investigate dynamics. This does not require the study of a system with a large number of links. Quite the opposite, such a system would only obscure our understanding considering that it would be difficult to determine such dynamics in closed form limiting our ability to mathematically dissect the results. Instead we focus on the simplest possible problem. This will allow us to derive dynamics in closed form to investigate mathematical connections between dynamics of different order. Using this approach, we will show that a lower order dynamic system and the reaction forces that impose its boundary conditions can be derive from higher order dynamics.

The first section of this chapter is dedicated to a discussion of a simple robotic system in various configurations. We will show how the highest order dynamical system can be used to represent lower order systems. The second section discusses dynamic transitions. Impact dynamics as well as

a continuity of state will be discussed in this section. In the third section linearized representations used for state stabilization will be discussed and in the fourth section actuator dynamics will be discussed.

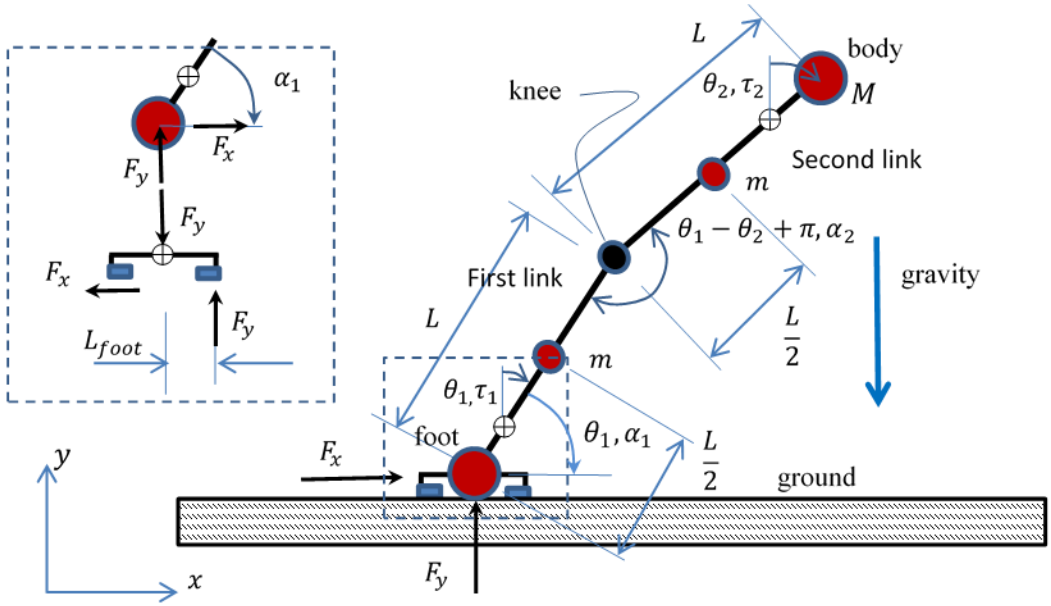



**Figure 8. An illustration of a two-link robot**

## 2.1 Rigid body dynamics

We focus on the dynamics of a simple robot – the two-link [64]. The two-link, shown in Figure 8, consists of two links called the first link and the second link. In this discussion, each link contains two masses, a large mass, $M$, and a small mass, $m$. The first node of the first link is called the foot and the second node of the second link is called the body. Control torques can be represented in terms a foot torque, $\alpha_1$, knee torque, $\alpha_2$, or the torques about the center of mass (COM) of the links, $\tau_1$ and $\tau_2$. These torques are related by the equations

$$\begin{Bmatrix}\alpha_1\\ \alpha_2\end{Bmatrix} = \begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{Bmatrix}\tau_1\\ \tau_2\end{Bmatrix} \text{ and } \begin{Bmatrix}\tau_1\\ \tau_2\end{Bmatrix} = \frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{Bmatrix}\alpha_1\\ \alpha_2\end{Bmatrix}. \quad (2.1)$$

The length of each link is $L$. The mass, $M$, is located at one end of each link and the mass, $m$, is located at the midpoint of each link. Kinematics are defined by angular rotations $\theta_1$and $\theta_2$ about the COM and the $x$ and $y$ location of the foot. The reaction forces on the foot are $F_x$ and $F_y$. Neither the

foot nor the knee are allowed to slide along the ground. The body is allowed to slide with no frictional resistance. The reaction forces on the foot are $F_x$ and $F_y$. As shown in the upper left of Figure 8, the reaction force, $F_y$, is transmitted from the ankle to foot. Given $F_y$, for a foot of length $L_{foot}$ , the maximum torque that can be applied to ankle by the foot $max(\alpha_1)$ is

$$max(\alpha_1) = L_{foot}F_y \geq |\alpha_1| \,. \tag{2.2}$$

### 2.1.1 Dynamic configurations

The two-link may be in a number of different possible configurations where a configuration is determined by which of its three nodes (knee, foot, and/or body) are in contact with the ground. Two-link configurations are shown in Figure 9. A short discussion of the dynamics associated with of these configurations is given below.

*Free dynamics* occurs when the two-link system is free to move through space in the presence of gravity and a knee torque. No nodes are in contact with the ground. Free dynamics continue until one of the three nodes impacts with the ground. This impact produces an impulsive load on the linkage transferring the dynamics to that of another configuration. It is assumed that all impacts are such that no rebound occurs. In free dynamics, four states, $\{x, y, \theta_1, \theta_2\}$, are required to describe the motion of the two-link. The free dynamics configuration is not completely controllable. Controllability implies the ability to move the state from one state to another.

- *Body dynamics* occurs when the body is in contact with the ground. In this situation, the end is allowed to slide along the ground. Three variable, $\{x, \theta_1, \theta_2\}$, are required to describe this motion. When the body reaction force becomes negative, body dynamics become free dynamics. If the foot contacts, the ground body dynamics become bent dynamics. Body dynamics are not completely controllable.

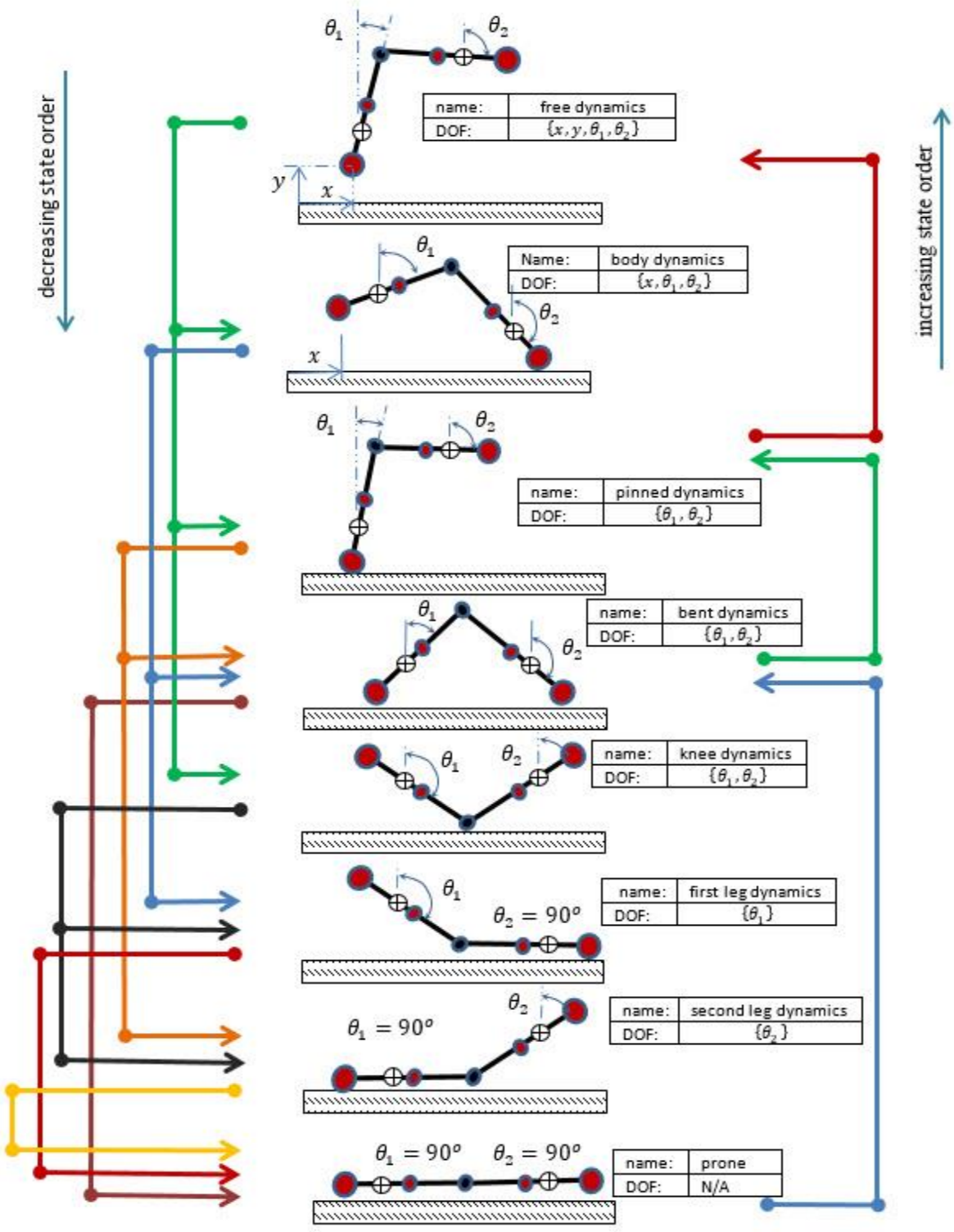



Figure 9. An illustration of two-link configurations

- *Pinned dynamics* occurs when the foot is at ground and the other two nodes are not. The foot is allowed to rotate but not translate (i.e. pinned). The torque produced at the knee producing a resulting reaction force at the foot. This force must be positive. If $F_y$ goes to zero, two-link dynamics becomes free dynamics. If the knee or body impacts the ground, the dynamics transition from pinned dynamics to knee dynamics or bent dynamics. Only the rotation of the links,$\{\theta_1, \theta_2\}$, are needed to describe motion in pinned dynamics. Pinned dynamics are completely controllable.
- *Bent dynamics* occurs when the foot and the body are in contact with the ground. It is assumed that the foot cannot slip but the head can slide allowing the two-link to bend at the knee. Bent dynamics are completely controllable.
- *Knee dynamics* occurs when the knee is in contact with the ground. It is assumed that the knee does not translate. Knee dynamics are not completely controllable.
- *First and second link dynamics* are the dynamics of a single pinned link. These dynamics are completely controllable so long as reaction forces are positive.
- In the prone position, there are no dynamics; however, there are a number of configurations. Only one of these configurations is shown in Figure 9.

In the following, the dynamics of the two-link in the free, pinned, and bent configurations is given. Other dynamics are not presented here.

### 2.1.2 Derivation of free dynamics

A Lagrangian approach (see Appendix A) is used to determine the linkage equations of motion. This requires the production of the Lagrangian, $\mathcal{L}$, which is comprised of the kinetic and potential energy of the system. To describe the kinetic energy, $T$, and the potential energy, $V$, of the two-link during free dynamics, four state variables, $\{x, y, \theta_1, \theta_2\}$, are required. The Lagrangian is given by

$$\mathcal{L} = T - V \,. \tag{2.3a}$$

The Euler-Lagrange (EL) equations for this situation are given by

$$\frac{d}{dt}\frac{\partial}{\partial\dot{\theta}_1}\mathcal{L}-\frac{\partial\mathcal{L}}{\partial\theta_1}=\tau_1, \tag{2.3b}$$

$$\frac{d}{dt}\frac{\partial}{\partial\dot{\theta}_2}\mathcal{L}-\frac{\partial\mathcal{L}}{\partial\theta_2}=\tau_2\,. \tag{2.3c}$$

$$\frac{d}{dt}\frac{\partial}{\partial\dot{x}}\mathcal{L}-\frac{\partial\mathcal{L}}{\partial x}=F_x \tag{2.3d}$$

$$\frac{d}{dt}\frac{\partial}{\partial\dot{y}}\mathcal{L}-\frac{\partial\mathcal{L}}{\partial y}=F_y \tag{2.3e}$$

where $\tau_1$ and $\tau_2$ are torques at the origin of the generalized coordinate for each link and $F_x$ and $F_y$ are the forces on the foot. The kinetic energy at any time is given by

$$T=\frac{1}{2}\{D_1+D_2+D_3+D_4\} \tag{2.4a}$$

where
$$D_1=\ m\left\{\frac{L^2}{4}\dot{\theta}_1^2+\dot{x}\dot{\theta}\ L\cos(\theta_1)-\dot{y}\dot{\theta}\ L\sin(\theta_1)+\ \dot{x}^2+\dot{y}^2\right\}$$

$$D_2=\ m\Big\{L^2\dot{\theta}_1^2+\frac{L^2}{4}\dot{\theta}_2^2+2L\dot{x}\dot{\theta}_1\ L\cos(\theta_1)+L\dot{x}\dot{\theta}_2\ L\cos(\theta_2)$$
$$-2L\dot{y}\dot{\theta}_1\sin(\theta_1)-L\dot{y}\dot{\theta}_2\sin(\theta_2)+L^2\dot{\theta}_1\dot{\theta}_2\ \cos(\theta_1-\theta_2)+\ \dot{x}^2+\dot{y}^2\Big\}$$

$$D_3=\ M\{L^2\dot{\theta}_1^2+L^2\dot{\theta}_2^2+2L\dot{x}\dot{\theta}_1\ L\cos(\theta_1)+2L\dot{x}\dot{\theta}_2\ L\cos(\theta_1)$$
$$-2L\dot{y}\dot{\theta}_1\sin(\theta_1)-2L\dot{y}\dot{\theta}_2\sin(\theta_2)+2L^2\dot{\theta}_1\dot{\theta}_2\ \cos(\theta_1-\theta_2)+\ \dot{x}^2+\dot{y}^2\}$$

$$D_4=\ M\{\dot{x}^2+\dot{y}^2\}$$

and the potential energy is given by

$$V=mg\tfrac{L}{2}\ cos(\theta_1)+mgL\left\{\cos(\theta_1)+\tfrac{1}{2}\cos(\theta_2)\right\}+MgL\{\cos(\theta_1)+\cos(\theta_2)\}+\{2m+2M\}gy \tag{2.4b}.$$

Substituting (2.4) into (2.3) gives the result

$$K\tfrac{d^2\vec{X}}{dt^2}=\begin{bmatrix}A & B\\ C & D\end{bmatrix}\begin{Bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\\ \ddot{x}\\ \ddot{y}\end{Bmatrix}=f\left(\vec{X},\tau_1,\tau_2,F_x,F_y\right)=\begin{Bmatrix}f_1\\ f_2\\ f_3\\ f_4\end{Bmatrix} \tag{2.5a}$$

where

$$\vec{X}=\ [\theta_1,\theta_2,x,y]^T$$

$$A=\begin{bmatrix}\left(\frac{5}{4}m+M\right)L^2 & \left(\frac{1}{2}m+M\right)L^2\cos(\theta_1-\theta_2)\\ \left(\frac{1}{2}m+M\right)L^2\cos(\theta_1-\theta_2) & \left(\frac{1}{4}m+M\right)L^2\end{bmatrix} \tag{2.5b}$$

$$B = C^T = \begin{bmatrix} \left(\frac{3}{2}m + M\right) L\, cos(\theta_1) & -\left(\frac{3}{2}m + M\right) L\, sin(\theta_1) \\ \left(\frac{1}{2}m + M\right) L\, cos(\theta_2) & -\left(\frac{1}{2}m + M\right) L\, sin(\theta_2) \end{bmatrix} \quad (2.5c)$$

$$D = (2m + 2M)\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.5d)$$

$$f_1 = -\left(\frac{1}{2}m + M\right) L^2 sin(\theta_1 - \theta_2)\dot{\theta}_2{}^2 + \left(\frac{3}{2}m + M\right) gLsin(\theta_1) + \tau_1 \quad (2.5e)$$

$$f_2 = \left(\frac{1}{2}m + M\right) L^2 sin(\theta_1 - \theta_2)\dot{\theta}_1{}^2 + \left(\frac{1}{2}m + M\right) gLsin(\theta_2) + \tau_2 \quad (2.5f)$$

$$f_3 = \left(\frac{3}{2}m + M\right) L\dot{\theta}_1^2 \sin(\theta_1) + \left(\frac{1}{2}m + M\right) L\dot{\theta}_2^2 \sin(\theta_2) + F_x \quad (2.5g)$$

$$f_4 = \left(\frac{3}{2}m + M\right) L\dot{\theta}_1^2 \cos(\theta_1) + \left(\frac{1}{2}m + M\right) L\dot{\theta}_2^2 \cos(\theta_2) - (2m + 2M)g + F_y. \quad (2.5f)$$

Equation (2.5) is a complex equation which will not be reduced further. Nevertheless, we will show that other forms of dynamics can be derived from these dynamics.

### 2.1.2 Derivation of pinned dynamics

Following a similar approach to that for free dynamics, the kinetic energy of the pinned dynamic system is given by

$$T = \frac{1}{2}\{D_1 + D_2 + D_3\} \quad (2.6a)$$

where

$$D_1 = \frac{m}{4} L^2 \dot{\theta}_1^2 ,$$

$$D_2 = mL^2 \left\{\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2 \cos(\theta_1 - \theta_2) + \frac{1}{4}\dot{\theta}_2^2\right\},$$

$$D_3 = ML^2\{\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2 \cos(\theta_1 - \theta_2) + \dot{\theta}_2^2\}.$$

The potential energy is given by

$$V = \frac{m}{2} L\, cos(\theta_1) + m\left\{L\cos(\theta_1) + \frac{L}{2}\cos(\theta_2)\right\} + M\{L\cos(\theta_1) + L\cos(\theta_2)\}. \quad (2.6b)$$

Substituting (2.6) into these equations and reducing gives

$$\begin{bmatrix} \left(\frac{5}{4}m+M\right)L^2 & \left(\frac{1}{2}m+M\right)L^2\cos(\theta_1-\theta_2) \\ \left(\frac{1}{2}m+M\right)L^2\cos(\theta_1-\theta_2) & \left(\frac{1}{4}m+M\right)L^2 \end{bmatrix}\begin{Bmatrix}\ddot{\theta}_1 \\ \ddot{\theta}_2\end{Bmatrix} =$$

$$\begin{Bmatrix} -\left(\frac{1}{2}m+M\right)L^2 sin(\theta_1-\theta_2)\dot{\theta}_2{}^2 + \left(\frac{3}{2}m+M\right)gLsin(\theta_1) + \tau_1 \\ \left(\frac{1}{2}m+M\right)L^2 sin(\theta_1-\theta_2)\dot{\theta}_1{}^2 + \left(\frac{1}{2}m+M\right)gLsin(\theta_2) + \tau_2 \end{Bmatrix}. \qquad (2.7a)$$

Equation (2.7a) can further reduced as

$$\ddot{\theta}_1 = f_1 = -\frac{\left\{\frac{m}{4}+M\right\}\left\{\frac{m}{2}+M\right\}\sin(\theta_1-\theta_2)\dot{\theta}_2^2}{\Delta} + \frac{\left\{\frac{m}{4}+M\right\}\left\{\frac{3m}{2}+M\right\}\frac{g}{L}\sin(\theta_1)}{\Delta}$$

$$-\frac{\left\{\frac{m}{2}+M\right\}^2\sin(\theta_1-\theta_2)\cos(\theta_1-\theta_2)\dot{\theta}_1^2}{\Delta} - \frac{\left\{\frac{m}{2}+M\right\}^2\frac{g}{L}\cos(\theta_1-\theta_2)sin(\theta_2)}{\Delta} + \frac{\left\{\frac{m}{4}+M\right\}\{\tau_1/L^2\}}{\Delta} - \frac{\left\{\frac{m}{2}+M\right\}\cos(\theta_1-\theta_2)\tau_2/L^2}{\Delta}$$

(2.7b)

$$\ddot{\theta}_2 = f_2 = \frac{\left\{\frac{m}{2}+M\right\}^2\sin(\theta_1-\theta_2)\cos(\theta_1-\theta_2)\dot{\theta}_2^2}{\Delta} - \frac{\left\{\frac{m}{2}+M\right\}\left\{\frac{3m}{2}+M\right\}\frac{g}{L}\cos(\theta_1-\theta_2)\,sin(\theta_1)}{\Delta}$$

$$+\frac{\left\{\frac{5m}{4}+M\right\}\left\{\frac{m}{2}+M\right\}\sin(\theta_1-\theta_2)\dot{\theta}_1^2}{\Delta} + \frac{\left\{\frac{5m}{4}+M\right\}\left\{\frac{m}{2}+M\right\}\frac{g}{L}sin(\theta_2)}{\Delta} - \frac{\left\{\frac{m}{2}+M\right\}\cos(\theta_1-\theta_2)\,\{\tau_1/L^2\}}{\Delta} + \frac{\left\{\frac{5m}{4}+M\right\}\tau_2/L^2}{\Delta}$$

(2.7c)

where $\Delta = \left\{\frac{5m}{4}+M\right\}\left\{\frac{m}{4}+M\right\} - \left\{\frac{m}{2}+M\right\}^2\cos^2(\theta_1-\theta_2)$.

The reaction forces at the foot of the two-link in pinned dynamics can be calculated by applying Newton's law to the COM of the system. The COM $\{x_{cg}, y_{cg}\}$ is given by

$$\{2m+2M\}x_{cg} = \sum_i m_i x_i = m\left\{\frac{3L}{2}sin(\theta_1) + \frac{L}{2}\sin(\theta_2)\right\} + M\{Lsin(\theta_1) + L\sin(\theta_2)\}$$

$$\{2m+2M\}y_{cg} = \sum_i m_i y_i = m\left\{\frac{3L}{2}cos(\theta_1) + \frac{L}{2}\cos(\theta_2)\right\} + M\{Lcos(\theta_1) + L\cos(\theta_2)\}.$$

Applying Newton's second law gives

$$F_x = \{2m+2M\}\ddot{x}_{cg} = -\left[m\left\{\frac{3L}{2}sin(\theta_1)\left(\dot{\theta}_1\right)^2 + \frac{L}{2}sin(\theta_2)\left(\dot{\theta}_2\right)^2\right\} - m\left\{\frac{3L}{2}cos(\theta_1)\ddot{\theta}_1 + \frac{L}{2}\cos(\theta_2)\ddot{\theta}_2\right\} + \right.$$

$$\left. M\left\{Lsin(\theta_1)\left(\dot{\theta}_1\right)^2 + Lsin(\theta_2)\left(\dot{\theta}_2\right)^2\right\} - M\left\{Lcos(\theta_1)\ddot{\theta}_1 + Lcos(\theta_2)\ddot{\theta}_2\right\}\right]$$

$$\text{if } F_y > 0 \text{ else } F_x = 0$$

(2.8a)

$$F_y = \{2m + 2M\}\ddot{y}_{cg} + 2\{m + M\}g = -\left[m\left\{\frac{3L}{2}cos(\theta_1)\left(\dot{\theta}_1\right)^2 + \frac{L}{2}\cos(\theta_2)\left(\dot{\theta}_2\right)^2\right\} + m\left\{\frac{3L}{2}sin(\theta_1)\ddot{\theta}_1 + \frac{L}{2}\sin(\theta_2)\ddot{\theta}_2\right\} + M\left\{Lcos(\theta_1)\left(\dot{\theta}_1\right)^2 + Lcos(\theta_2)\left(\dot{\theta}_2\right)^2\right\} + M\{Lsin(\theta_1)\ddot{\theta}_1 + Lsin(\theta_2)\ddot{\theta}_2\}\right] + 2\{m + M\}g$$

$$\text{if } F_y > 0 \text{ else} \quad F_y = 0$$

(2.8b)

For $F_y$ greater than zero, the two-link dynamics maintains pinned dynamics; however as soon as $F_y$ goes to zero, the dynamics switch from pinned dynamics to free dynamics.

Equations (2.7a,b) and (2.8a) and (2.8b) can also be obtained directly from (2.5a) by imposing the equality constraint

$$S\left(\vec{X}, \tau_1, \tau_2, F_x, F_y\right) = \begin{Bmatrix}\ddot{x}\\ \ddot{y}\end{Bmatrix} = 0, \text{for } y = 0 \tag{2.8c}$$

in equation (2.5a). Equation (2.7b) comes from the upper left LHS (the A matrix) of (2.5a) and the upper RHS of (2.5a). Equation (2.8a) comes from the resulting third row and (2.8b) comes from the resulting fourth row. In addition to these equations an impulsive load must be applied at the time of impact such $\dot{x}$ and $\dot{y}$ are driven to zero. That is, pinned dynamics are simply free dynamics with the (2.8c) constraint imposed. When $F_y$ goes to zero, the (2.8c) constraint is lifted and pinned dynamics become free dynamics as required by the (2.8b) inequality constraint.

### 2.1.3 Derivation of bent dynamics

In bent dynamics only one state variable is required to define the energies of the system due to the constraint

$$\theta_1 + \theta_2 - \pi = 0. \tag{2.10a}$$

Using this constraint and its derivatives, the kinetic and potential energy of the system are given by

$$T = \frac{1}{2}\{D_1 + D_2\} \tag{2.10b}$$

where

$$D_1 = \frac{m}{2}L^2\dot{\theta}_1^2\,, \tag{2.10c}$$

$$D_2 = 2(m + 2M)L^2\cos(\theta_1)\,\dot{\theta}_1^2, \tag{2.10d}$$

and $$V = mLg\cos(\theta_1). \tag{2.10e}$$

The Euler-Lagrange (EL) equations for this configuration are equations (2.3a) and (2.3b).

Substituting (2.10a-e) into these equations and reducing gives

$$\{m + 4\{m + 2M\}\cos^2(\theta_1)\}\ddot{\theta}_1 = 4\{m + 2M\}\cos(\theta_1)\sin(\theta_1)\,\dot{\theta}_1^2 + 2m\frac{g}{L}\sin(\theta_1) + \alpha/L^2\,. \tag{2.10f}$$

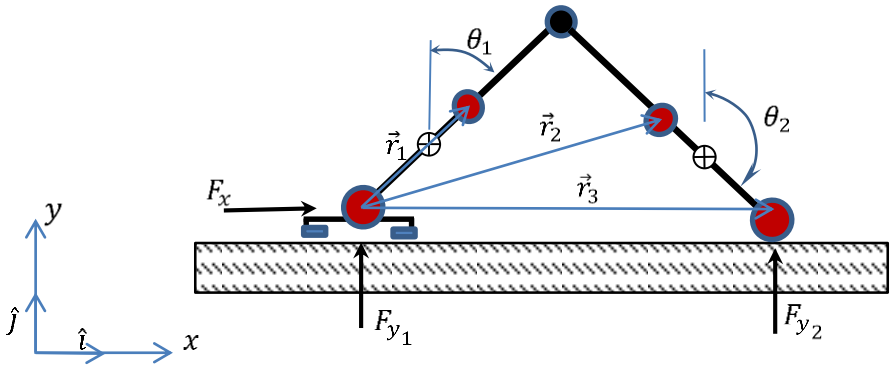



**Figure 10. An illustration of bent dynamics reaction forces**

To determine the reaction force at the body, an angular momentum balance about the foot must be performed. As shown in Figure 10, we introduce a reaction force at that body, $F_{y_2}$. The normal reaction force at the foot is given by $F_{y_1}$. A momentum balance is performed by calculating the displacement to each mass, taking the time derivative to find the velocity of that mass, calculating the angular momentum using a cross product, and applying Newton's second law to determine the reaction force at the body.

The displacement to the first mass is given by

$$\vec{r}_1 = \frac{L}{2}\sin(\theta_1)\,\hat{\imath} + \frac{L}{2}\cos(\theta_1)\,\hat{\jmath} \tag{2.11a}$$

where $\hat{\imath}$ and $\hat{\jmath}$ are unit vectors in the $x$ and $y$ directions. The velocity of the first mass is the derivative of (2.11a)

$$\dot{\vec{r}}_1 = \frac{L}{2}\cos(\theta_1)\,\dot{\theta}_1\hat{\imath} - \frac{L}{2}\sin(\theta_1)\dot{\theta}_1\,\hat{\jmath} \tag{2.11b}$$

and the angular momentum of the first mass is given by

$$M_1 = m\{\dot{\vec{r}}_1 \times \vec{r}_1\} = m \begin{vmatrix} \frac{L}{2}\cos(\theta_1)\,\dot{\theta}_1 & -\frac{L}{2}\sin(\theta_1)\dot{\theta}_1 \\ \frac{L}{2}\sin(\theta_1) & \frac{L}{2}\cos(\theta_1) \end{vmatrix} = m\frac{L^2}{4}\dot{\theta}_1. \tag{2.11c}$$

Performing the same process on the other two moving masses gives

$$M_2 = m\frac{3L^2}{4}\dot{\theta}_1, \tag{2.11d}$$

$$M_3 = 0. \tag{2.11e}$$

Therefore, using Newton's second law,

$$2L\sin(\theta_1)\,F_{y_2} - 2LMg\sin(\theta_1) - 2mLg\sin(\theta_1) = -\frac{d}{dt}\{M_1 + M_2 + M_3\} = -mL^2\ddot{\theta}_1 \text{ and}$$

$$F_{y_2} = \{M + m\}g\ -\frac{mL\ddot{\theta}_1}{2\sin(\theta_1)}. \tag{2.11f}$$

Moreover, from Newton's second law in the $y$ direction,

$$F_{y_1} = 2(M + m)g - mL\left\{cos(\theta_1)\dot{\theta_1}^{\,2} + sin(\theta_1)\ddot{\theta}_1\right\} - F_{y_2} \tag{2.11g}$$

Similar as before, the dynamics described by (2.7b) (the higher order system) contains the dynamics described by (2.10f) (the lower order system). If we subtract the first row of (2.7a) from the second row of (2.7a), apply (2.10a) constraint and reduce, we obtain (2.10f) where $\alpha$ in (2.10f) is $\tau_1 - \tau_2 = \alpha_2$ in the resulting subtraction (as expected).

If we add the first row of (2.7a) to the second row of (2.7a) and apply (2.10a) we obtain

$$\tau_1 + \tau_2 = L^2\ddot{\theta}_1 - 2\{M + m\}Lg\sin(\theta_1). \tag{2.11h}$$

But this is the same as (2.11f) where

$$-2\,F_{y_2}L\sin(\theta_1) = \tau_1 + \tau_2, \tag{2.11i}$$

the added torque due to the reaction force needed to produce the (2.10a) constraint condition.

## 2.2 Dynamic transitions

As shown on the left and right side of Figure 9, the two-link can transition from one set of dynamics to another changing order with transition. During these transitions we assume that only one node is allowed to change its relationship with the ground at a time (for multiple contact problems see

references 65, 66 and/or 67). For example, to reach the prone position from the free position one possibility is for the foot to impact, transitioning to pinned dynamics; from pinned dynamics the body may impact, transitioning to bent dynamics; and from bent dynamics the knee may impact, transitioning to the prone position.

Two-link dynamics are altered by a node impacting or leaving the ground. The resulting post-impact response is calculated using impact (sometimes called impulse) dynamics. An impact reduces order. When a reaction force which constrains a node goes to zero, order is augmented. For example, if the two-link is crouching in a stabilized pinned dynamic configuration and control is applied at the knee, the reaction force at the foot will initially be positive. As the momentum of the two-link increases this force will go to zero. At this instance the two-link transitions from pinned dynamics to a free dynamics and the states are augmented by $\{x, y\}$ where the initial conditions of the augmented states is equal to the location of the foot at the time of departure. Gravity drives the two-link back to earth. At the time of impact, the order is reduced back to that of pinned dynamics.

Transitions involving a node leaves the ground is mathematically straight forward; however, determining state transitions when a node impacts the ground is non-trivial. To find the change in state velocities due to impact, a virtual approach method could be used [58, 68]; however in this paper, a more instructive Newtonian approach is used. In this dissertation we assumed that after impact, the point of impact has zero velocity normal to the ground. The Newtonian approach allows us to solve this problem.

The integral of Newton's second law with an impulsive force $F\delta(t)$ acting on a mass, $m$, is given by

$$m\{v^+ - v^-\} = F \tag{2.15}$$

where $v^+$ is the velocity of the mass instantaneously prior to the impulse, $v^-$ is the velocity of the mass instantaneously after the impulse and $F$ is the magnitude of the impulsive load. In the following, a similar nomenclature will be used to describe the response of the two-link -- a superscript with a minus sign stands for the state just prior to the impulse and a superscript with a

plus sign stands for the state just after the impulse. The process used to determine the resulting state is to apply (2.15) at the instance of impact to each link and to apply motion constraints at the point of impact. This impact does not change the instantaneous location of the linkage system but alters its velocities in such a way that the boundary conditions for the configuration that the two-link is transitioning to are satisfied.

Figure 11 shows the impulse loads and the altered state velocities on the linkages during an arbitrary impact. There are eleven (11) possible unknowns $v_{x,1}^{+}$, $v_{y,1}^{+}$, $\dot{\theta}_1^{+}$, $v_{x,2}^{+}$, $v_{y,2}^{+}$, $\dot{\theta}_2^{+}$, $F_{x_1}$, $F_{y_1}$, $f_x$, $f_y$, and $F_{y_2}$. Velocities are defined about the center of mass (COM) of each link. Some of the knowns are $v_{x,1}^{-}$ , $v_{y,1}^{-}$, $\dot{\theta}_1^{-}$, $v_{x,2}^{-}$, $v_{y,2}^{-}$, and $\dot{\theta}_2^{-}$ along with the orientation of the linkages $\theta_1$ and $\theta_2$. Other knowns are associated with constraint conditions.

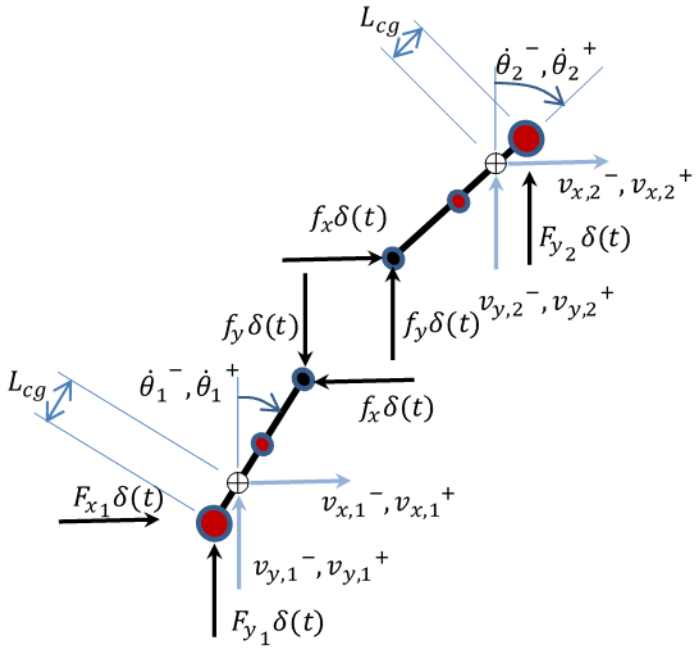



**Figure 11. An illustration of the impulsive forces on a two-link**

Applying (2.15) to the COM of each link gives

$$(m+M)v_{x,1}^{-} = (m+M)v_{x,1}^{+} - F_{x_1} + f_x\,, \tag{2.16a}$$

$$(m+M)v_{y,1}^{-} = (m+M)v_{y,1}^{+} - F_{y_1} + f_y\,, \tag{2.16b}$$

$$(m+M)v_{x,2}^{-} = (m+M)v_{x,2}^{+} - f_x\,, \tag{2.16c}$$

$$(m+M)v_{y,2}^{-} = (m+M)v_{y,2}^{+} - f_y - F_{y_2}\,. \tag{2.16d}$$

Applying angular momentum to the COM of each link gives

$$I_1\dot{\theta}_1^{-} = I_1\dot{\theta}_1^{+} + F_{x_1}\,L_{cg}\cos(\theta_1) - L_{cg}\,F_{y_1}\sin(\theta_1) + f_x\,\big(L-L_{cg}\big)\cos(\theta_1) - f_y(L-L_{cg})\sin(\theta_1) \tag{2.16e}$$

$$I_2\dot{\theta}_2^{-} = I_2\dot{\theta}_2^{+} + f_x\big(L-L_{cg}\big)\,\cos(\theta_2) - f_y\big(L-L_{cg}\big)\sin(\theta_2) + L_{cg}F_{y_2}\sin(\theta_2) \tag{2.16f}$$

where $I_1 = I_2 = m\left\{L_{cg} - \frac{L}{2}\right\}^2 + M\{L_{cg}\}^2$ and $L_{cg} = \frac{L}{2}\left\{\frac{m}{M+m}\right\}$.

Applying kinematic constraints at the knee just after impact gives

$$0 = v_{x,1}^{+} - v_{x,2}^{+} + \dot{\theta}_1^{+}(L-L_{cg})\cos(\theta_1) + \dot{\theta}_2^{+}\big(L-L_{cg}\big)\cos(\theta_2)\,, \tag{2.16g}$$

$$0 = v_{y,1}^{+} - v_{y,2}^{+} - \dot{\theta}_1^{+}\big(L-L_{cg,1}\big)\sin(\theta_1) - \dot{\theta}_2^{+}\big(L-L_{cg}\big)\sin(\theta_2). \tag{2.16h}$$

Applying kinematic constraints for zero motion at the foot gives

$$0 = v_{x,1}^{+} - \dot{\theta}_1^{+}L_{cg}\cos(\theta_1)\,, \tag{2.16i}$$

$$0 = v_{y,1}^{+} + \dot{\theta}_1^{+}L_{cg}\sin(\theta_1)\,. \tag{2.16j}$$

Applying kinematic constraints for zero motion at the end head gives

$$0 = v_{x,2}^{+} + L_{cg}\cos(\theta_2)\,, \tag{2.16k}$$

$$0 = v_{y,2}^{+} - \dot{\theta}_2^{+}L_{cg}\sin(\theta_2)\,. \tag{2.16l}$$

Equations (2.16a) through (2.16h) apply for any impact conditions. However, equations (2.16i) through 2.16l) apply depending on the node undergoing impact and the resulting boundary condition. For the situation where the knee is not on the ground, the form of the matrix relationship representing these relationships is given by

$$\phi = A\psi \tag{2.17a}$$

where

$\phi = [{\phi_u}^T \quad {\phi_l}^T]^T$ is a vector of knowns,

$\psi = [{\psi_u}^T \quad {\psi_l}^T]^T$ is a vector of unknowns,

$\phi_u = \begin{bmatrix} v_{x,1}^- & v_{y,1}^- & \dot{\theta}_1^- & v_{x,2}^- & v_{y,2}^- & \dot{\theta}_2^- & 0 & 0 \end{bmatrix}^T$, $\phi_l$ is a function of the node undergoing impact ,

$\psi_u = \begin{bmatrix} v_{x,1}^+ & v_{y,1}^+ & \dot{\theta}_1^+ & v_{x,2}^+ & v_{y,2}^+ & \dot{\theta}_2^+ & f_x & f_y \end{bmatrix}^T$, $\psi_u$ is a function of the node undergoing impact, and the resulting boundary condition

$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, A_{21} = \mathcal{R}^{Nox8}, A_{12} = \mathcal{R}^{8xNo}, A_{22} = 0^{NoxNo}$

$v_{x,1}^- = \quad (L - L_{cg}) cos(\theta_1) \dot{\theta}_1^-,$

$v_{y,1}^- = -(L - L_{cg}) \sin(\theta_1) \dot{\theta}_1^-,$

$v_{x,2}^- = \quad L cos(\theta_1) \dot{\theta}_1^- + (L - L_{cg}) cos(\theta_2) \dot{\theta}_2^-,$

$v_{y,2}^- = -L sin(\theta_1) \dot{\theta}_1^- - (L - L_{cg}) sin(\theta_2) \dot{\theta}_2^-,$

$$A_{11} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1/(m+M) & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1/(m+M) \\ 0 & 0 & 1 & 0 & 0 & 0 & (L - L_{cg}) \cos(\theta_1)/I_1 & -(L - L_{cg}) \sin(\theta_1)/I_1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1/(m+M) & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1/(m+M) \\ 0 & 0 & 0 & 0 & 0 & 1 & (L - L_{cg}) \cos(\theta_1)/I_2 & -(L - L_{cg}) \sin(\theta_1)/I_2 \\ -1 & 0 & -(L - L_{cg}) \cos(\theta_1) & 1 & 0 & -(L - L_{cg}) \cos(\theta_2) & 0 & 0 \\ 0 & -1 & (L - L_{cg}) \sin(\theta_1) & 0 & 1 & (L - L_{cg}) \sin(\theta_2) & 0 & 0 \end{bmatrix}, \quad (2.17b)$$

In the following sections, quantification of $A_{12}$ and $A_{21}$ is performed for a limited set of impact transitions.

### 2.2.1 Transition: free to pinned dynamics

For this situation the unknowns are $v_{x,1}^+$, $v_{y,1}^+$, $\dot{\theta}_1^+$, $v_{x,2}^+$, $v_{y,2}^+$, $\dot{\theta}_2^+$, $f_x$, $f_y$, $F_{x_1}$, and $F_{y_1}$, and equations (16a) through (16j) are applicable. For this situation, $No = 2$ and

$$\phi_l = [0 \quad 0]^T, \quad (2.18a)$$

$$\psi_l = \begin{bmatrix} F_{x_1} & F_{y_1} \end{bmatrix}^T, \quad (2.18b)$$

$$A_{12} = \begin{bmatrix} -1/(m+M) & 0 \\ 0 & -1/(m+M) \\ L_{cg} \cos(\theta_1)/I_1 & -L_{cg} \sin(\theta_1)/I_1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \quad (2.18c)$$

$$A_{21} = \begin{bmatrix} 1 & 0 & -L_{cg}\cos(\theta_1) & 0 & 0 & 0 \\ 0 & 1 & L_{cg}\sin(\theta_1) & 0 & 0 & 0 \end{bmatrix}, \tag{2.18d}$$

Matrix $A_{21}$ represent the constraint condition at the foot, and matrix $A_{12}$ represents the additional inertial loads and moments resulting from foot reaction loads.

### 2.2.2 Transition: pinned to bent dynamics

For this situation the unknowns are $v_{x,1}^{+}$, $v_{y,1}^{+}, \dot{\theta}_1^{+}, v_{x,2}^{+}$, $v_{y,2}^{+}$, $\dot{\theta}_2^{+}, f_x$, $f_y$, $F_{x_1}$, $F_{y_1}$, and $F_{y_2}$. For this situation, $No = 3$ and

$$\phi_l = [0 \quad 0 \quad 0]^T, \tag{2.19a}$$

$$\psi_l = \left[F_{x_1}, F_{y_1}, F_{y_2}\right]^T, \tag{2.19b}$$

$$A_{12} = \begin{bmatrix} -1/(m+M) & 0 & 0 \\ 0 & -1/(m+M) & 0 \\ L_{cg}\cos(\theta_1)/I_1 & -L_{cg}\sin(\theta_1)/I_1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1/(m+M) \\ 0 & 0 & L_{cg}\sin(\theta_2)/I_2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.19c}$$

$$A_{21} = \begin{bmatrix} 1 & 0 & -L_{cg}\cos(\theta_1) & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & L_{cg}\sin(\theta_1) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -L_{cg}\sin(\theta_2) & 0 & 0 \end{bmatrix}, \tag{2.19d}$$

This transition is only valid for the situation where $F_{y_1}$ is greater than zero.

Other transitions exit; however, their derivations are not presented here.

## 2.3 Actuator dynamics

In sections 2.1 and 2.2 we discussed dynamics and transitions. These are related to rigid body dynamics; however, these are not all of the dynamics in the system. Actuator dynamics must also be considered. Figure 12 shows the elements of a simplified model of motor dynamics. The input voltage $V$ is divided between the voltage across the internal resistance, $R$, and the motor voltage $v_m$. This motor voltage produces a torque at the motor shaft $\tau_m$. This torque is then amplified through a gear box with ratio, $GR$. We assume that inductive motor effects can be neglected.

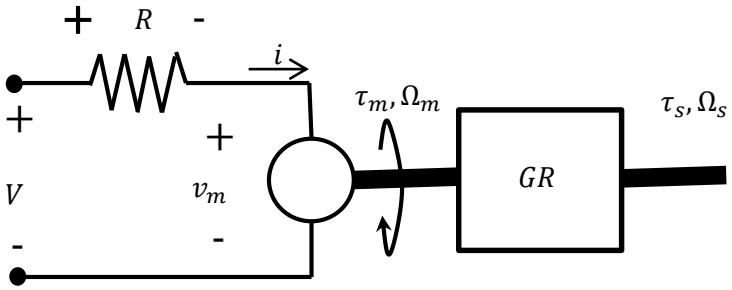



**Figure 12. An illustration of actuator (motor) components**

Transduction between the electrical and mechanical response is given by

$$\tau_m = K_Q\, i \tag{2.22a}$$

$$v_m = \frac{\Omega_m}{K_v} \tag{2.22b}$$

where $K_Q$ and $K_v$ are a function of the motor. Performing a voltage balance gives

$$V = R\, i - v_m. \tag{2.22c}$$

Solving (2.22a) through (2.22c) for $\tau_m$ in terms of the inputs ($v$ , $\Omega_m$) gives

$$\tau_m = \frac{K_Q}{R} v - \frac{K_Q}{R\, K_v} \Omega_m. \tag{2.22d}$$

Two curves are commonly given for motor characteristic – $i$ versus $\tau_m$ at nominal voltage and $\Omega_m$ versus $\tau_m$ at nominal voltage. These plots are shown in Figure 13. The slope of the $i$ versus $\tau_m$ determines $K_Q$ from (2.22a). Other parameters are determined from the $\Omega_m$ versus $\tau_m$, at nominal, voltage curve. As shown in Figure 13 the maximum shaft speed $\Omega_{max}$ occurs when $\tau_m = 0$. The maximum shaft torque $\tau_{max}$ occurs when $\Omega_m = 0$. This is also called the stall torque. Solving (2.22f) for these two conditions gives

$$K_v = \frac{\Omega_{max}}{v_{nom}} \tag{2.22e}$$

$$R = \frac{v_{nom}}{i_{max}}. \tag{2.22f}$$

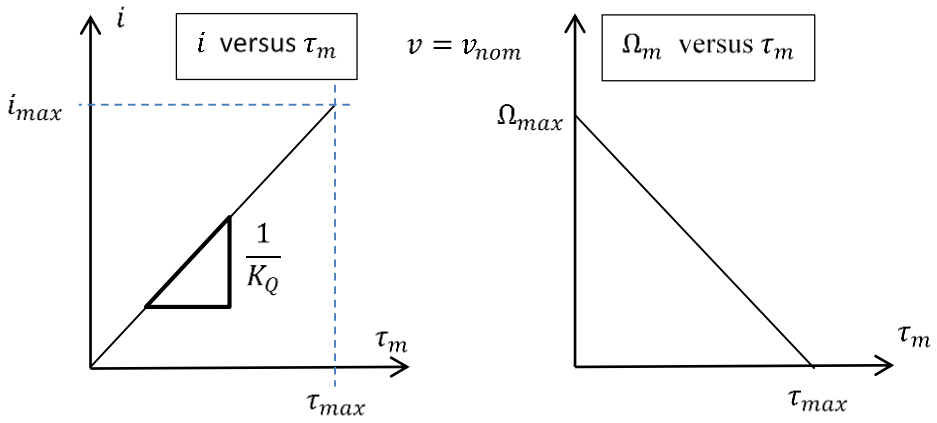



**Figure 13. An illustration of motor characteristics**

The gear box behaves are a transformer between mechanical inputs and responses where

$$\tau_s = GR\,\tau_m, \tag{2.22d}$$

$$\Omega_s = \frac{\Omega_m}{GR}. \tag{2.22e}$$

Applying these equations to (2.22d) gives

$$\tau_s = GR\left[\frac{K_Q}{R}V - \frac{K_Q}{R\,K_v}\Omega_s\right]. \tag{2.22f}$$

Equation (2.22f) allows for the inputs in sections 2.1 through 2.3 to be modified to include the effect of actuator dynamics. Applying the nomenclature used in the above sections to this equation gives

$$\alpha_1 = GR\left[\frac{K_Q}{R}V_1 - \frac{K_Q}{R\,K_v}\dot{\theta}_1\right] \tag{2.23a}$$

$$\alpha_2 = GR\left[\frac{K_Q}{R}V_2 - \frac{K_Q}{R\,K_v}\left(\dot{\theta}_1 - \dot{\theta}_2\right)\right] \tag{2.23b}$$

where $v_1$ is the voltage into the foot motor and $v_2$ is the voltage into the knee motor and $\alpha_1$ and $\alpha_2$ are torques defined in Figure 8.

## 2.4 Dynamics for state stabilization

The control approach used in this paper is to transition the state of the linkage system through state space using a rhythm to a termination state which is stabilized through the use of feedback control.

In this thesis, the two-link is assumed to be stabilized about three states, standing, crouching right, and crouching left. These three stabilized states are associated with pinned dynamics as described in (2.7). To stabilize these states a linearized representation of the dynamics is required. The linearization of (2.7) about $\vec{X}^* = \{\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2\}^T = \{{\theta_1}^*, 0, {\theta_2}^*, 0\}$, under the influence of the equilibrium control, $\vec{u} = \{{v_1}^* \quad {v_2}^*\}^T$ is

$$\frac{d}{dt}\begin{Bmatrix} \theta_1 - {\theta_1}^* \\ \dot{\theta}_1 - {\dot{\theta}_1}^* \\ \theta_2 - {\theta_2}^* \\ \dot{\theta}_2 - {\dot{\theta}_2}^* \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \left.\frac{\partial f_1}{\partial \theta_1}\right|_{\vec{X}^*} & \left.\frac{\partial f_1}{\partial \dot{\theta}_1}\right|_{\vec{X}^*} & \left.\frac{\partial f_1}{\partial \theta_2}\right|_{\vec{X}^*} & \left.\frac{\partial f_1}{\partial \dot{\theta}_2}\right|_{\vec{X}^*} \\ 0 & 0 & 0 & 1 \\ \left.\frac{\partial f_2}{\partial \theta_1}\right|_{\vec{X}^*} & \left.\frac{\partial f_2}{\partial \dot{\theta}_1}\right|_{\vec{X}^*} & \left.\frac{\partial f_2}{\partial \theta_2}\right|_{\vec{X}^*} & \left.\frac{\partial f_2}{\partial \dot{\theta}_2}\right|_{\vec{X}^*} \end{bmatrix} \begin{Bmatrix} \theta_1 - {\theta_1}^* \\ \dot{\theta}_1 - {\dot{\theta}_1}^* \\ \theta_2 - {\theta_2}^* \\ \dot{\theta}_2 - {\dot{\theta}_2}^* \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ \left.\frac{\partial f_1}{\partial v_1}\right|_{\vec{X}^*} & \left.\frac{\partial f_1}{\partial v_2}\right|_{\vec{X}^*} \\ 0 & 0 \\ \left.\frac{\partial f_2}{\partial v_1}\right|_{\vec{X}^*} & \left.\frac{\partial f_2}{\partial v_2}\right|_{\vec{X}^*} \end{bmatrix} \begin{Bmatrix} v_1 - {v_1}^* \\ v_2 - {v_2}^* \end{Bmatrix} \tag{2.20a}$$

where the terms in (2.20a) are given in appendix B. Equation (2.20a) is written in compact form as

$$\frac{d}{dt}\vec{X} = A\vec{X} + B\vec{u}\,. \tag{2.20b}$$

The standing position is a stabilized state with nominal states defined by

$$\vec{X}^* = \begin{Bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \text{ under the steady state excitation } {v_1}^* = {v_2}^* = 0. \tag{2.21a}$$

Crouching positions are stabilized states defined by the crouching angle $\theta^*$ where

$$\vec{X}^* = \begin{Bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{Bmatrix} = \begin{Bmatrix} \theta^* \\ 0 \\ sin^{-1}\left\{-\frac{\frac{3}{2}m+M}{\frac{1}{2}m+M}\sin(\theta^*)\right\} \\ 0 \end{Bmatrix} \text{ under the steady-state excitation}$$

$$v_1{}^* = 0,\ v_2{}^* = \frac{2R}{GR\,K_Q}\{3m + 2M\}\sin(\theta^*). \qquad (2.21b,c)$$

## 2.5 Summary of chapter

This chapter presents a study of the dynamics of a simple robotic system called a two-link. In the first section we examined the rigid body dynamics of this robot in a number of different configurations. It was shown that once the highest order dynamic system (i.e. free dynamics) was known, the dynamics of all lower order systems can be determined along with the reaction forces required to maintain any constraint conditions.

In the second section we examined dynamic transitions. The approach used in this section was different from most of what is found in the literature. In this section we assumed that, post impact, the impact location has no normal velocity. A Newtonian approach could be used to determine the resulting velocity of links.

In the third section actuator dynamics were discussed. This discussion outlined a simple dynamical model for a motor.

In the fourth section, the linearized dynamics of the two-link are presented. These dynamics are necessary for producing stabilized states.

In the following chapter, the nonlinear dynamics presented in this chapter will be used to derive open loop control (OLC) inputs, called rhythms. These rhythms will be used to drive the robotic system from one stabilized state to another producing locomotion. This will be demonstrated in chapter 4. In chapters 5 and 6, hardware is built to validate theoretical development in chapters 2 through 4.

# Chapter 3

## 3.0 Control

Chapter 2 presents closed form representations of two-link dynamics. These representations allowed for an understanding of the relationship between the dynamics of different configurations as well as serving as a numerical tool that can be used to investigate OLC methods.

As stated in chapter 1, finding solutions to satisfy OLOC necessary conditions can be difficult. As an alternative, we will take a more numerically tractable approach. We parameterize the control input using a sparse number of parameters and then optimize these parameters to solve for a suboptimal solution.

Consider the following two simple examples. In problem 1 we solve an OLOC problem using the calculus of variations of solution given in appendix C. In problem 2 we solve the same problem using a OLC method assuming a simplified control input form. We show that both methods produce similar results.

---

### Problem 1: Solution to a OLOC using the shooting method

**Given:**

$\dot{x} = -ax + bu(t), x(t_o) = x_o$

**Find:**

$u(t)$ such that $x(t_f) = x_f$ such that $IP = \int_{t_o}^{t_f} u(t)^2 \ dt$ is mimimized

**Solution:**

The necessary conditions for this problem is a subset of the equations in appendix C

$$\dot{\lambda} = -\frac{\partial H}{\partial x}, \tag{3.1a}$$

$$\dot{x} = -ax + bu, x(t_o) = x_o, \text{ and } x(t_f) = x_f \tag{3.1b}$$

$$\text{with } u(x, \lambda, t) \text{ is chosen such that } \frac{\partial H}{\partial u}(u, x, \lambda) = 0 \tag{3.1c}$$

where $H = u^2 + \lambda(-ax + bu)$.

The above set of equations represents a two-point boundary value problem (TPBV). There are two ordinary differential equations, (3.1a, and b), where the initial and final temporal boundary conditions of (3.b) are completely defined. The boundary conditions of (3.1a) are undefined. Equations (3.1a) and (3.1b) are coupled equations through the control input (3.1c). The TPBV solution determines boundary condition for 3.1a that will make 3.1b true.

A solution can be found using the shooting method. If the initial conditions of both (3.1a) and (3.1b) ($x(t_o)$ and $\lambda(t_o)$) were known, a solution could be found by performing a forward integration of (3.1a,b) where $u(t)$ is determined at any integration time from (3.1c). Using a shooting method numerous forward iterations are performed for various $\lambda(t_o)$ until $x(t_f) = x_f$. For each iteration, a new estimate of $\lambda(t_o)$ is developed based upon the previous results until the proper final state is obtained.

For $a = 1$, $b = 2$, $x_f = 2$, $t_o = 1$, and $t_f = 1$, the solution is shown (in red) in Figure 14. An initial guess of $\lambda(t_o) = 0.5$ produces a terminal value of 2.338 (too high) and an initial guess of $\lambda(t_o) = 1.0$ produces a terminal value of 1.1684 (too low). The next guess was produced by interpolation

$$\lambda(t_o) \sim \frac{2.0 - 1.169}{2.338 - 1.169}\big((-1.0) - (-0.5)\big) + (-0.5) = -0.8554$$

which results in an initial condition which is very close to the desired terminal state value. In general, for nonlinear systems, numerous iterations are needed to produce convergence.

In general, the shooting method is a poor approach to finding a solution to OLOC necessary conditions. For systems with higher order dynamics applied over longer time frames, the shooting methods requires a very good initial estimate for $\lambda(t_o)$ [57]. A gradient method could be used to solve this problem but this solution can result in a prodigious number of minimums.

The OLOC input is shown in Figure 15. Notice that this solution is almost a straight line. If we knew this prior to starting the problem, we might have attempted to use a straight line approximation of the control input defined by an initial value and a slope. We could then perform a global search on

these two parameters to determine a near optimal OLC solution. By doing this we would have recast the OLOC problem into a FD OBM problem.

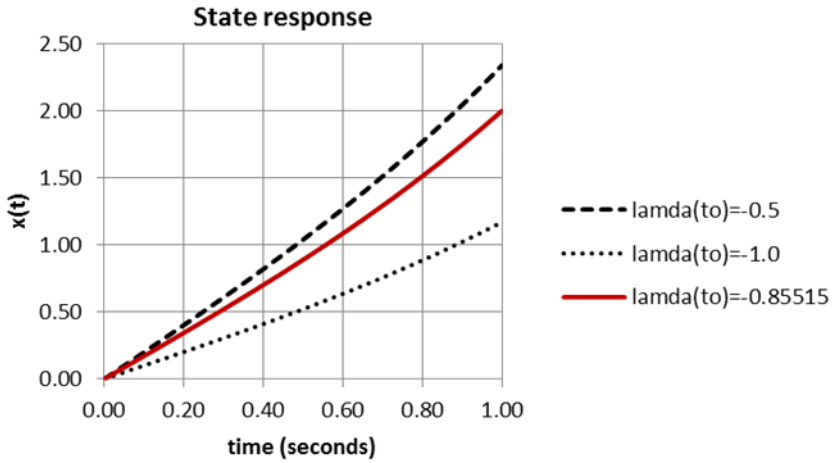



**Figure 14. Problem 1 solution using the shooting method**

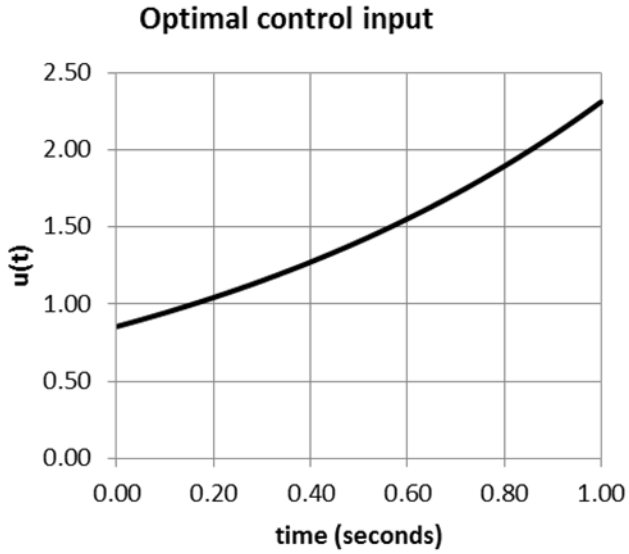



**Figure 15. Problem 1 optimal control input**

---

From insight gained by solving the above optimal control problem, we now assume that the control input is linear and solve the same problem.

## Problem 2: Solution to problem 1 by functional approximation, OLC

**Given:**

$\dot{x} = -ax + bu(t), x(t_o) = x_o$

**Find:**

$u(t)$ such that $x(t_f) = x_f$ such that $IP = \int_{t_o}^{t_f} u(t)^2 \ dt$ is minimized

where

a) $u(t) = \upsilon + \gamma t.$

b) $u(t) = \upsilon$

**Solution:**

**Part a)** This is a constraint optimization problem

$IP = \alpha_0 \upsilon^2 + 2\alpha_1 \upsilon\gamma + \alpha_2 \gamma^2 + \lambda(x(a_o, a_1, t_f) - x_f)$

where $\alpha_0 = \int_{t_o}^{t_f} dt$, $\alpha_1 = \int_{t_o}^{t_f} t \ dt$, $\alpha_2 = \int_{t_o}^{t_f} t^2 \ dt$ and $\lambda$ is a Lagrange multiplier. In this representation we note that the simulated final state $x(\upsilon, \gamma, t_f)$ is a function of the control variables.

Necessary conditions are

$\frac{\partial IP}{\partial \upsilon} = 2\alpha_0 \upsilon + \alpha_1 \gamma + \lambda \frac{\partial x(\upsilon,\gamma,t_f)}{\partial \upsilon} = 0$

$\frac{\partial IP}{\partial \gamma} = \alpha_1 \upsilon + 2\alpha_2 \gamma + \lambda \frac{\partial x(\upsilon,\gamma,t_f)}{\partial \gamma} = 0$

$\frac{\partial IP}{\partial \lambda} = x(a_o, a_1, t_f) - x_f = 0$

To find a solution to these conditions we use a gradient solution. If not at the optimal solution these conditions become

$\frac{\partial IP}{\partial \upsilon} = 2\alpha_0 \upsilon + \alpha_1 \gamma + \lambda \frac{\partial x(\upsilon,\gamma,t_f)}{\partial \upsilon} = \epsilon_1$

$\frac{\partial IP}{\partial \gamma} = \alpha_1 \upsilon + 2\alpha_2 \gamma + \lambda \frac{\partial x(\upsilon,\gamma,t_f)}{\partial \gamma} = \epsilon_2$

$\frac{\partial IP}{\partial \lambda} = x(\upsilon, \gamma, t_f) - x_f = \epsilon_3$

We would like to drive $\vec{\epsilon}^T\vec{\epsilon}$ to zero. To do this define

$$R = \begin{bmatrix} \frac{\partial \epsilon_1}{\partial \upsilon} & \frac{\partial \epsilon_1}{\partial \gamma} & \frac{\partial \epsilon_1}{\partial \lambda} \\ \frac{\partial \epsilon_2}{\partial \upsilon} & \frac{\partial \epsilon_2}{\partial \gamma} & \frac{\partial \epsilon_2}{\partial \lambda} \\ \frac{\partial \epsilon_3}{\partial \upsilon} & \frac{\partial \epsilon_3}{\partial \gamma} & \frac{\partial \epsilon_3}{\partial \lambda} \end{bmatrix}$$

where

$$\frac{\partial \epsilon_1}{\partial \upsilon} = 2\alpha_o + \lambda \frac{\partial^2 x(\upsilon, \gamma, t_f)}{\partial \upsilon^2}$$

$$\frac{\partial \epsilon_1}{\partial \upsilon} = 2\alpha_1 + \lambda \frac{\partial^2 x(\upsilon, \gamma, t_f)}{\partial \upsilon \partial \gamma}$$

$$\frac{\partial \epsilon_1}{\partial \lambda} = \frac{\partial x(\upsilon, \gamma, t_f)}{\partial \upsilon}$$

$$\frac{\partial \epsilon_2}{\partial \upsilon} = 2\alpha_1 + \lambda \frac{\partial^2 x(\upsilon, \gamma, t_f)}{\partial \upsilon \partial \gamma}$$

$$\frac{\partial \epsilon_2}{\partial \upsilon} = 2\alpha_2 + \lambda \frac{\partial^2 x(\upsilon, \gamma, t_f)}{\partial \gamma^2}$$

$$\frac{\partial \epsilon_2}{\partial \lambda} = \frac{\partial x(\upsilon, \gamma, t_f)}{\partial \gamma}$$

$$\frac{\partial \epsilon_3}{\partial \upsilon} = \frac{\partial x(\upsilon, \gamma, t_f)}{\partial \upsilon}$$

$$\frac{\partial \epsilon_3}{\partial \upsilon} = \frac{\partial x(\upsilon, \gamma, t_f)}{\partial \gamma}$$

$$\frac{\partial \epsilon_3}{\partial \lambda} = 0$$

Approximations to the above partial derivatives are given by

$$\frac{\partial x(\upsilon,\gamma,t_f)}{\partial \upsilon} \sim \frac{x(\upsilon+h,\gamma,t_f) - x(\upsilon-h,\gamma,t_f)}{2h} + HOT$$

$$\frac{\partial x(\upsilon,\gamma,t_f)}{\partial \gamma} \sim \frac{x(\upsilon,\gamma+h,t_f) - x(\upsilon,\gamma-h,t_f)}{2h} + HOT$$

$$\frac{\partial^2 x(\upsilon,\gamma,t_f)}{\partial \upsilon^2} \sim \frac{x(\upsilon+h,\gamma,t_f) - x(\upsilon,\gamma,t_f) + x(\upsilon-h,\gamma,t_f)}{h^2} + HOT$$

$$\frac{\partial^2 x(v,\gamma,t_f)}{\partial \gamma^2} \sim \frac{x(v,\gamma+h,t_f)-x(v,\gamma,t_f)+x(v,\gamma-h,t_f)}{h^2} + HOT$$

$$\frac{\partial^2 x(v,\gamma,t_f)}{\partial v \partial \gamma} \sim \frac{x(v+h,\gamma+h,t_f)-x(v+h,\gamma-h,t_f)-x(v-h,\gamma+h,t_f)+x(v-h,\gamma-h,t_f)}{4h^2} + HOT\ .$$

Now define a vector of estimates of unknowns $\phi_n = [v \quad \gamma \quad \lambda]^T$ where $\phi_n$ represents the $n^{th}$ estimate. Then to first order

$$\phi_{n+1} = \phi_n - \mu R^{-1}\vec{\epsilon}$$

where $\mu$ is a small number and $R$ is re-evaluated every iteration.

Performing 1000 iterations with $\mu = 0.01$ for $a = 1, b = 2, x_f = 2, t_o = 1$, and $t_f = 1$, $\vec{\epsilon}^T\vec{\epsilon} = 7.78e-04$, a number close to zero. The solution and control input are shown in Figure 16 and Figure 17. Also shown in these figures is the problem 1 solution. These solutions are very similar. They have almost the same performance, $IP \sim 2.35$ (within numerical error).

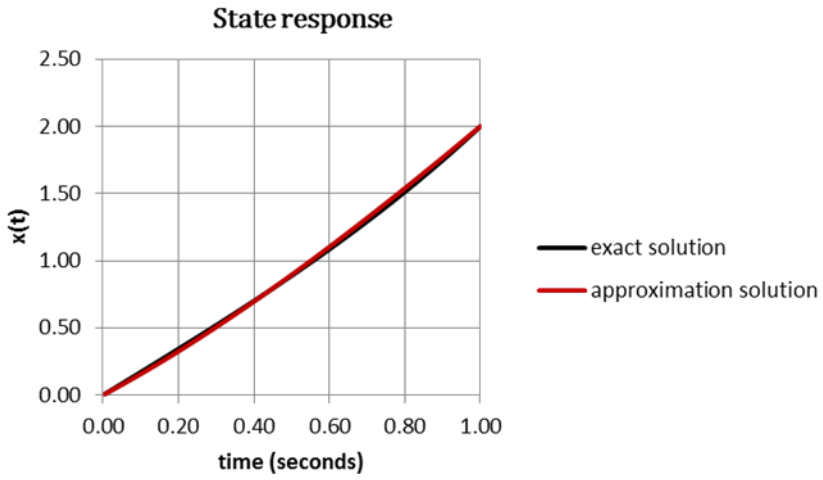



**Figure 16. OLOC and OLC state response comparison**

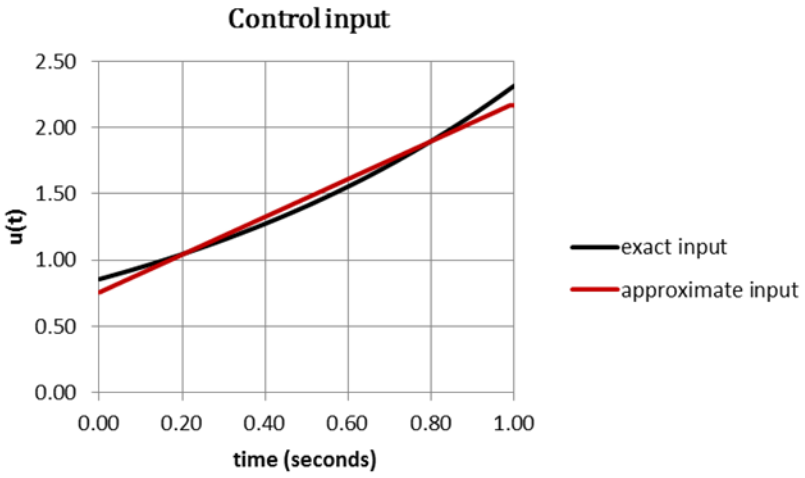



**Figure 17. OLOC and OLC control input comparison**

**Part b)** In this part, there is only one unknown, $\upsilon$. Given one terminal constraint condition, $x(\upsilon, t_f) = x_f$, the unknown is uniquely determined. Therefore, it is not possible to also minimize an integral performance metric. In this dissertation, this form of solution is called a degenerate solution.

The necessary condition for this problem is simply

$x(\upsilon, t_f) - x_f = \epsilon$

This can be solved using the same gradient approach as in part a). Using this method gives $\upsilon = 1.587$. The state response is given in Figure 18. Notice that the path is different however the terminal constraint is maintained.

A comparison of control input is given in Figure 19. Even through a minimization of control energy was not part of the formulation, it can still be calculated. For this problem the integral control energy is $IP = 2.52$ which is within 6% error from the optimal solution given in problem 1.

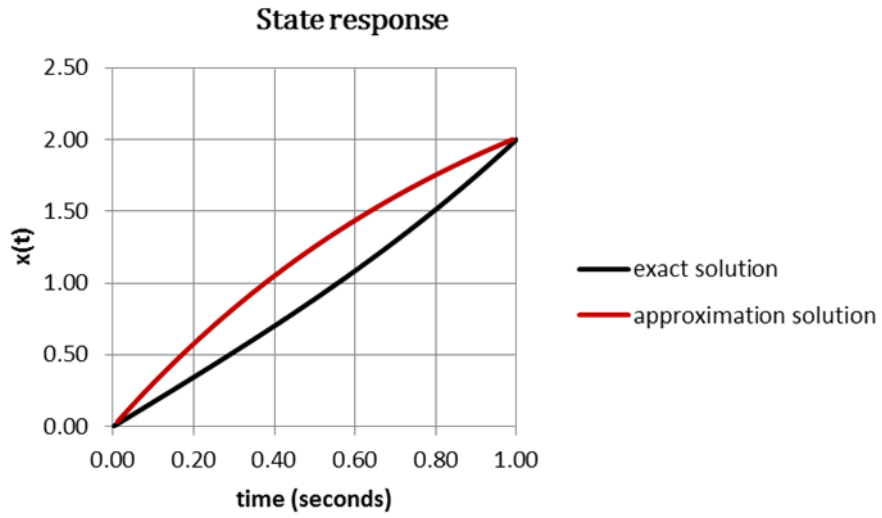



Figure 18. OLOC and OLC state response comparison, degenerate solution

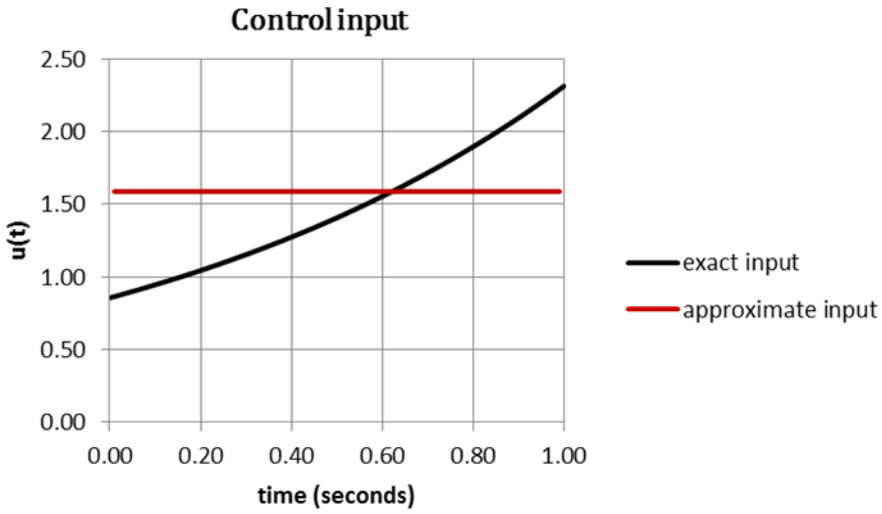



Figure 19. OLOC and OLC control input comparison

---

The above problems illustrate how OLC solutions can be used (for some dynamic systems) to produce near optimal minimal energy solutions.

A possible complaint of this method is that this requires knowledge as to the form of the control input. This is not a strong limitation considering that if we are uncertain as to the form of the control input, several forms can be explored to find the one that gives the best performance.

### 3.1 Functional approximations

Following the above assumption, we assume that control inputs can be represented using simple piecewise continuous functions as illustrated in Figure 20.

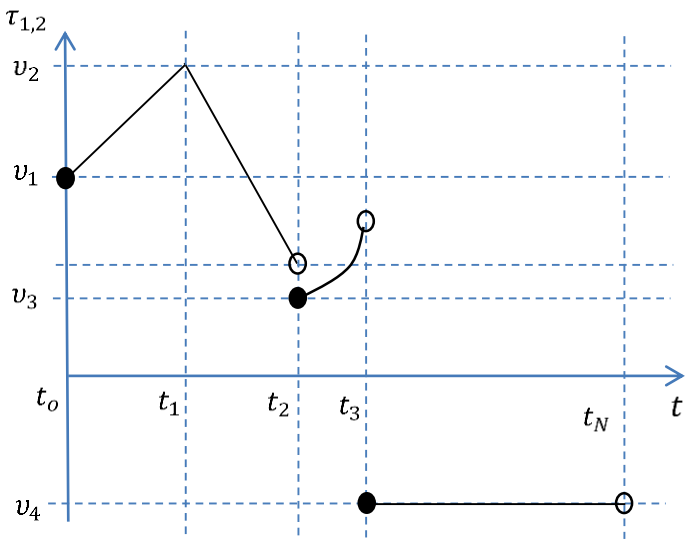



**Figure 20. An illustration of piecewise linear control inputs**

These inputs $\tau_j(t), j \in [1,2]$ that can be generalized as

$$\tau_j(t) = \sum_{i=0}^{Nm-1}\{\upsilon_i + \gamma_i t + \beta_i t^2 \dots\}\{H(t_i) - H(t_{i+1})\} \tag{3.1a}$$

where $t_{Nm} = t_f$, $H(t_i)$ is the Heaviside step function and $Nm$ is the number piecewise sections that comprise the input.

To find an OLC solution, necessary conditions are formulated. To do this we minimize the control energy,

$$\min_{\upsilon_1,\gamma_1,\beta_1,\upsilon_2,\gamma_2\dots} IP = \int_{t_o}^{t_f} \tau_1(t)^2 + \tau_2(t)^2 \dots dt \Leftrightarrow \min_{\upsilon_1,\gamma_1,\beta_1,\upsilon_2,\gamma_2\dots} IP = \alpha_o {\upsilon_o}^2 + \alpha_1 \upsilon_o \gamma_o + \cdots$$

with respect to the terminal constraints.

$$\vec{X}(t_f) = \left[\theta_1(t_f), \theta_2(t_f), \dot{\theta}_1(t_f), \dot{\theta}_2(t_f)\right]^T = \vec{X}_f = \left[\theta_{1,f}, \theta_{2,f}, \dot{\theta}_{1,f}, \dot{\theta}_{2,f}\right]^T$$

Substituting (3.1) into the above integral results in a parametric optimization problem

$$\min_{\upsilon_1,\gamma_1,\beta_1,\upsilon_2,\gamma_2\dots} IP = \alpha_o {\upsilon_o}^2 + \alpha_1 \upsilon_o \gamma_o + \cdots + \vec{\lambda}^T(\vec{X}(t_f) - \vec{X}_f)$$

where $\vec{\lambda} = [\lambda_1, \lambda_2, \lambda_3, \lambda_4\ ]$ (for the two-link).

Necessary conditions then become

$$\frac{\partial IP}{\partial \upsilon} = 2\alpha_0\upsilon + \alpha_1\gamma + \cdots\ \vec{\lambda}^T \frac{\partial \vec{X}(t_f)}{\partial \upsilon} = 0 \tag{3.1b}$$

$$\frac{\partial IP}{\partial \gamma} = \alpha_1\upsilon + 2\alpha_2\gamma + \vec{\lambda}^T \frac{\partial \vec{X}(t_f)}{\partial \gamma} = 0$$

$$\vdots$$

$$\frac{\partial IP}{\partial \vec{\lambda}} = X(t_f) - X_f = 0.$$

Equation 3.1b represents a set of nonlinear algebraic equation when must be solved numerically. This is similar to the work performed by Eriksson using TFEM [44]; however, the number of unknowns is far less allowing for a tractable solution.

A number of approaches can be used to find a solution to these equations. Here we use a gradient approach. Using this approach, we look at the situation where (3.1b) are evaluated off the optimal. In this case (3.1b) becomes

$$\frac{\partial IP}{\partial \upsilon} = 2\alpha_0\upsilon + \alpha_1\gamma + \cdots\ \vec{\lambda}^T \frac{\partial \vec{X}(t_f)}{\partial \upsilon} = \epsilon_1 \tag{3.1c}$$

$$\frac{\partial IP}{\partial \gamma} = \alpha_1\upsilon + 2\alpha_2\gamma + \vec{\lambda}^T \frac{\partial \vec{X}(t_f)}{\partial \gamma} = \epsilon_2$$

$$\vdots$$

$$\frac{\partial IP}{\partial \lambda_1} = \theta_1(t_f) - \theta_{1,f} = \epsilon_{M+1}.$$

$$\vdots$$

$$\frac{\partial IP}{\partial \lambda_4} = \dot{\theta}_2(t_f) - \dot{\theta}_{2,f} = \epsilon_{M+4}.$$

where $M$ is the number of parameters used to describe all control inputs.

We define $\vec{\epsilon} = [\epsilon_1, \epsilon_2, \dots]^T$, and neglecting $\beta_i's$ (assuming linear piecewise functions), let

$$\vec{\phi} = [\upsilon_o \quad \gamma_o \quad \upsilon_1 \quad \gamma_1 \quad \upsilon_2 \quad \dots\ \vec{\lambda}^T]^T, \tag{3.1d}$$

a vector of $M$ unknowns. Expanding $\vec{\epsilon}$ in a Taylor series and keeping the first term gives

$$\vec{\epsilon} = R\{\vec{\phi} - \vec{\phi}^*\}$$

where $\vec{\phi}^*$ is the solution which zeros (3.1c) and

$$R = \begin{bmatrix} \frac{\partial \epsilon_1}{\partial \upsilon_o} & \frac{\partial \epsilon_1}{\partial \gamma_o} & \frac{\partial \epsilon_1}{\partial \upsilon_1} & \cdots & \frac{\partial \epsilon_1}{\partial \lambda_1} & \frac{\partial \epsilon_1}{\partial \lambda_2} & \frac{\partial \epsilon_1}{\partial \lambda_3} & \frac{\partial \epsilon_1}{\partial \lambda_3} \\ \frac{\partial \epsilon_2}{\partial \upsilon_o} & \frac{\partial \epsilon_2}{\partial \gamma_o} & \frac{\partial \epsilon_2}{\partial \upsilon_1} & \cdots & \frac{\partial \epsilon_2}{\partial \lambda_1} & \frac{\partial \epsilon_2}{\partial \lambda_2} & \frac{\partial \epsilon_2}{\partial \lambda_3} & \frac{\partial \epsilon_2}{\partial \lambda_4} \\ \frac{\partial \epsilon_3}{\partial \upsilon_o} & \frac{\partial \epsilon_3}{\partial \gamma_o} & \frac{\partial \epsilon_3}{\partial \upsilon_1} & \cdots & \frac{\partial \epsilon_3}{\partial \lambda_1} & \frac{\partial \epsilon_3}{\partial \lambda_2} & \frac{\partial \epsilon_3}{\partial \lambda_3} & \frac{\partial \epsilon_3}{\partial \lambda_4} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \in \mathcal{R}^{M+4,M+4}$$

Solving for $\vec{\phi}^*$ gives

$$\vec{\phi}^* = \vec{\phi} - \mu R^{-1}\vec{\varepsilon} \tag{3.1e}$$

where $\mu = 1$. In general, a first order approximation is not sufficient and (3.1e) must be applied repeatedly with $\mu \ll 1$.

For the situation where $M = 4$, the solution becomes degenerate. In this case, an integral optimization cannot be performed since there are only enough unknowns to maintain terminal constraints. In this situation the necessary conditions are

$\theta_1(t_f) - \theta_{1,f} = 0$

$\vdots$

$\dot{\theta}_4(t_f) - \dot{\theta}_{4,f} = 0.$

This set of equations is solved using the same gradient approach discussed given.

For the situation where $M < 4$, a unique solution does not exist.

---

In the following, we will develop OLC solutions for the two-link dynamics presented in chapter 2. Initially we start with the simplest of two-link dynamics where we are able compare the OLC solution to the OLOC minimum energy solution. As we progress to move complicated dynamics, obtaining a OLOC solution will not be possible; however, a OLC solution under the (3.1a) assumption is tractable.

## Problem 3. Two-link minimum energy solution, a comparison

**Given:**

- Two-link parameters given in appendix D
- Pinned dynamics as given by (2.7a)

$$\begin{bmatrix} \left(\frac{5}{4}m+M\right)L^2 & \left(\frac{1}{2}m+M\right)L^2\cos(\theta_1-\theta_2) \\ \left(\frac{1}{2}m+M\right)L^2\cos(\theta_1-\theta_2) & \left(\frac{1}{4}m+M\right)L^2 \end{bmatrix} \begin{Bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{Bmatrix} =$$

$$\begin{Bmatrix} -\left(\frac{1}{2}m+M\right)L^2 sin(\theta_1-\theta_2)\dot{\theta}_2{}^2 + \left(\frac{3}{2}m+M\right)gLsin(\theta_1) + \tau_1 \\ \left(\frac{1}{2}m+M\right)L^2 sin(\theta_1-\theta_2)\dot{\theta}_1{}^2 + \left(\frac{1}{2}m+M\right)gLsin(\theta_2) + \tau_2 \end{Bmatrix}. \tag{2.7a}$$

- The final time $t_f = 0.1$ (sec).
- The initial condition

$$\vec{X}_o = \begin{Bmatrix} \theta_1 = -40 \cdot \frac{\pi}{180} \\ \dot{\theta}_1 = 0 \\ \theta_2 = sin^{-1}\left\{-\frac{\frac{3}{2}m+M}{\frac{1}{2}m+M}\sin(\theta_1)\right\} \\ \dot{\theta}_2 = 0 \end{Bmatrix},$$

and the desired final condition

$$\vec{X}_f = \begin{Bmatrix} \theta_1 = 0 \cdot \frac{\pi}{180} \\ \dot{\theta}_1 = 0 \\ \theta_2 = sin^{-1}\left\{-\frac{\frac{3}{2}m+M}{\frac{1}{2}m+M}\sin(\theta_1)\right\} \\ \dot{\theta}_2 = 0 \end{Bmatrix}.$$

**Find:**

a) The optimal control solution which minimizes

$IP = \int_0^{t_f}\{\tau_1^2 + \tau_2^2\}\,dt$ with respect to (2.7a) dynamics

b) A terminal state solution which assumes that the control voltage inputs are simple linear function.

That is,

$$\tau_1 = \upsilon_1 + \nu_1 t/t_f \text{ and } \tau_2 = \upsilon_2 + \nu_2 t/t_f$$

**Solution:**

Optimal control necessary conditions are a subset of the necessary conditions given in appendix C. These conditions were solved for using a shooting method. The solution for part b was solved for using the above gradient approach. The solution for part b was a degenerate solution. Both solutions are shown below.

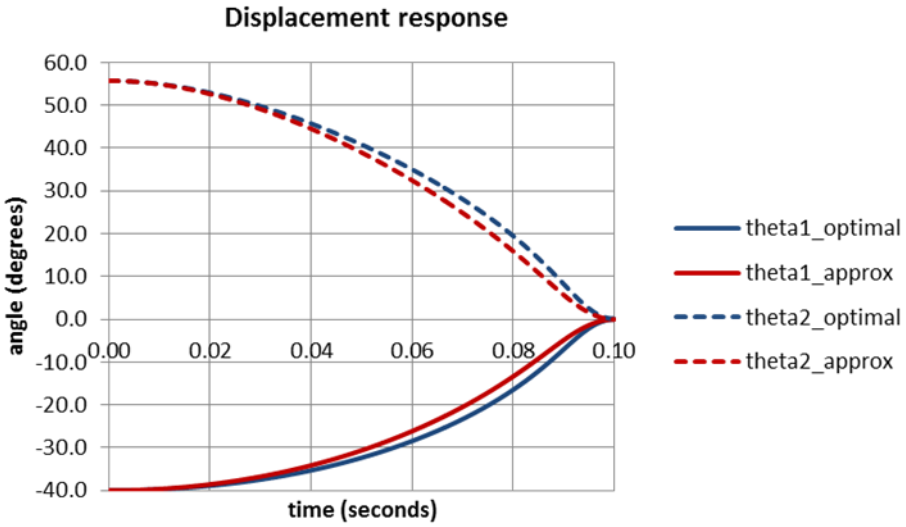



**Figure 21. Problem 3: comparison of minimum energy solutions, displacement**

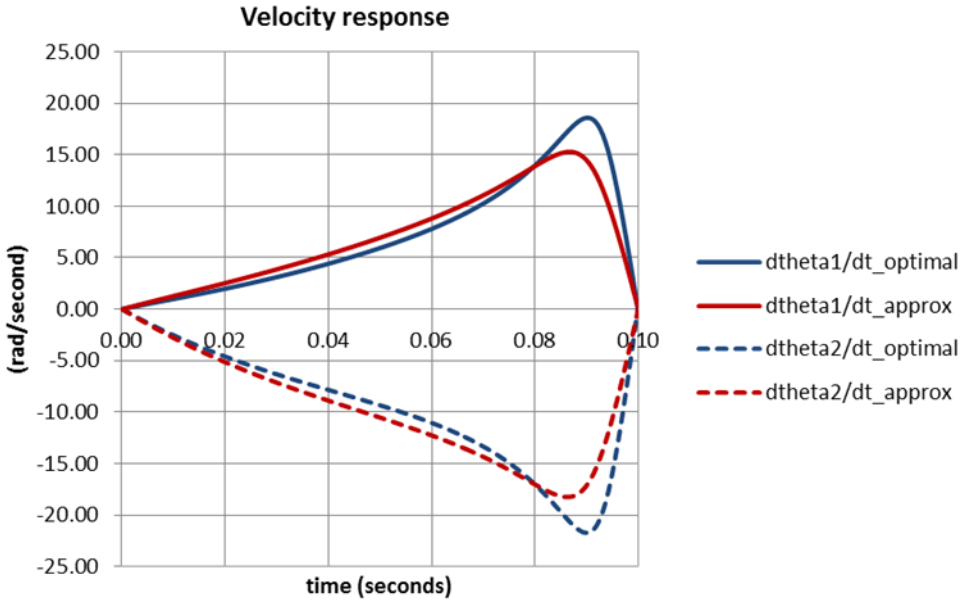



Figure 22. Problem 3: comparison of minimum energy solutions, rate

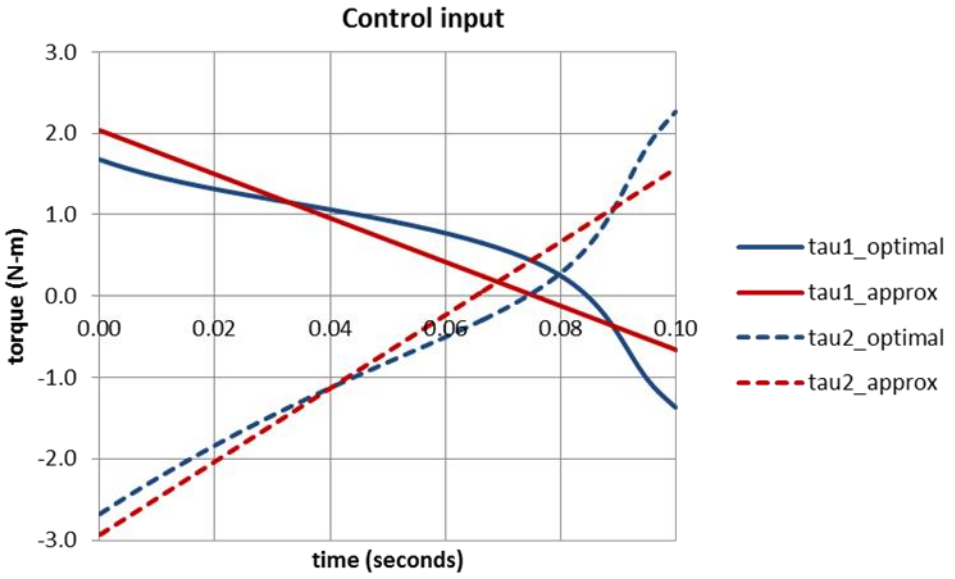



Figure 23. Problem 3: comparison of the minimum energy solutions, control

The optimal control minimum energy solution produces an $IP = 0.32$. Calculating the same performance metric for part b) gives $IP = 0.33$ - a variation of less than 3% from part a) results.

Problem 3 shows that for a class of systems a linear control input is a very good assumption producing an error from the minimum energy solution within 3%.

Through-out the rest of this dissertation we assume that control inputs can be represented as given in (3.1a). In the next problems, we develop rhythms using this assumption. As part of this development, we increase complexity by including actuator dynamics and variations in state order due to changes in configuration. Finding an OLOC solution for this problem will be difficult. Finding a OLC under the (3.1a) assumption is tractable.

---

**Problem 4. Rhythm: Crouching right to standing (cr2s)**

**Given:**

- Two-link parameters given in Appendix D.
- Pinned dynamics as given by (2.7a)

$$\begin{bmatrix} \left(\frac{5}{4}m+M\right)L^2 & \left(\frac{1}{2}m+M\right)L^2\cos(\theta_1-\theta_2) \\ \left(\frac{1}{2}m+M\right)L^2\cos(\theta_1-\theta_2) & \left(\frac{1}{4}m+M\right)L^2 \end{bmatrix}\begin{Bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{Bmatrix} =$$

$$\begin{Bmatrix} -\left(\frac{1}{2}m+M\right)L^2 sin(\theta_1-\theta_2)\dot{\theta}_2{}^2 + \left(\frac{3}{2}m+M\right)gLsin(\theta_1) + \tau_1 \\ \left(\frac{1}{2}m+M\right)L^2 sin(\theta_1-\theta_2)\dot{\theta}_1{}^2 + \left(\frac{1}{2}m+M\right)gLsin(\theta_2) + \tau_2 \end{Bmatrix}. \quad (2.7a)$$

- The reaction force at the foot given as by (2.8b)

$$F_y = -\left[m\left\{\frac{3L}{2}cos(\theta_1)\left(\dot{\theta}_1\right)^2 + \frac{L}{2}\cos(\theta_2)\left(\dot{\theta}_2\right)^2\right\} + m\left\{\frac{3L}{2}sin(\theta_1)\ddot{\theta}_1 + \frac{L}{2}\sin(\theta_2)\ddot{\theta}_2\right\} + M\left\{Lcos(\theta_1)\left(\dot{\theta}_1\right)^2 + Lcos(\theta_2)\left(\dot{\theta}_2\right)^2\right\} + M\left\{Lsin(\theta_1)\ddot{\theta}_1 + Lsin(\theta_2)\ddot{\theta}_2\right\}\right] + 2\{m+M\}g\,. \quad (2.8b)$$

- Actuator dynamics are given by (2.23a) and (2.23b)

$$\alpha_1 = GR\left[\frac{K_Q}{R}V_1 - \frac{K_Q}{R\,K_v}\dot{\theta}_1\right] \quad (2.23a)$$

$$\alpha_2 = GR\left[\frac{K_Q}{R}V_2 - \frac{K_Q}{R\,K_v}\left(\dot{\theta}_1 - \dot{\theta}_2\right)\right] \quad (2.23b)$$

where from (2.1)

$$\begin{Bmatrix}\tau_1\\ \tau_2\end{Bmatrix} = \frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{Bmatrix}\alpha_1\\ \alpha_2\end{Bmatrix}. \tag{2.1}$$

- The initial condition

$$\vec{X}_o = \begin{Bmatrix} \theta_1 = -40 \cdot \frac{\pi}{180} \\ \dot{\theta}_1 = 0 \\ \theta_2 = sin^{-1}\left\{-\frac{\frac{3}{2}m+M}{\frac{1}{2}m+M}\sin(\theta_1)\right\} \\ \dot{\theta}_2 = 0 \end{Bmatrix},$$

and the desired final condition

$$\vec{X}_f = \begin{Bmatrix} \theta_1 = 0 \cdot \frac{\pi}{180} \\ \dot{\theta}_1 = 0 \\ \theta_2 = sin^{-1}\left\{-\frac{\frac{3}{2}m+M}{\frac{1}{2}m+M}\sin(\theta_1)\right\} \\ \dot{\theta}_2 = 0 \end{Bmatrix}.$$

- The final time $t_f = 0.6$ (sec).
- A total foot size which is 0.2 times the length of a link (i.e. $L_{foot} = 0.08\ meters$).

**Find:**

a) A solution that assumes that the control voltage inputs are simple linear function,

That is,

$$V_1 = v_1 + v_1 t/t_f \text{ and } V_2 = v_2 + v_2 t/t_f$$

b) If the size of foot is sufficient to maintain full actuation

c) A suitable gear ration for the motor gearbox.

**Solution:**

a) The solution for the situation where the control input is linear is shown in the below figures.

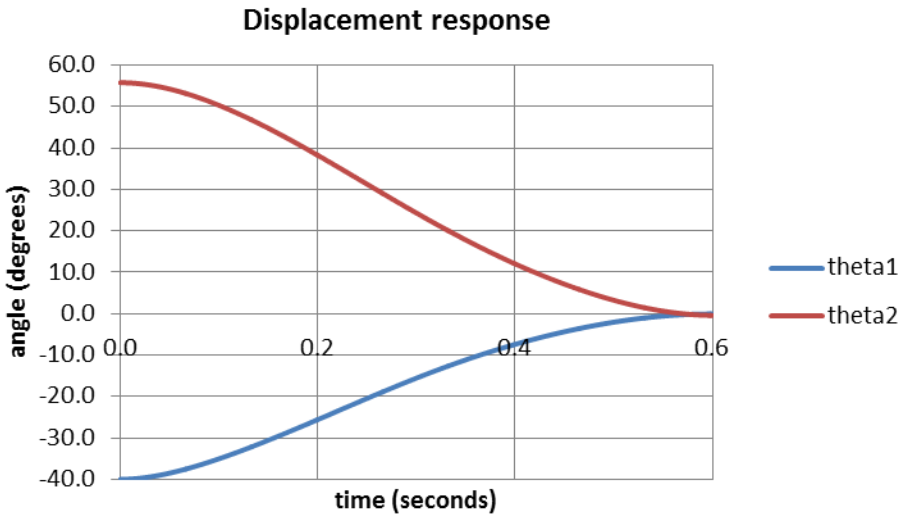



Figure 24. Problem 4: Displacement response, (cr2s)

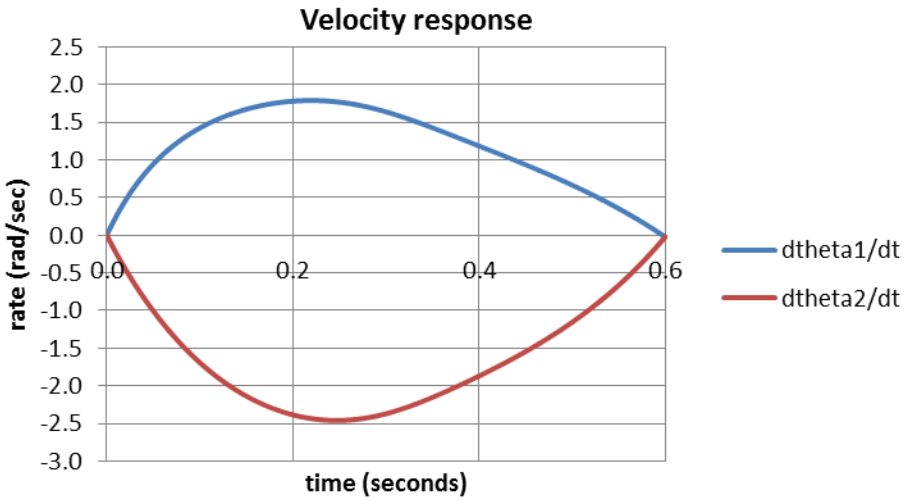



Figure 25. Problem 4: Velocity response, (cr2s)

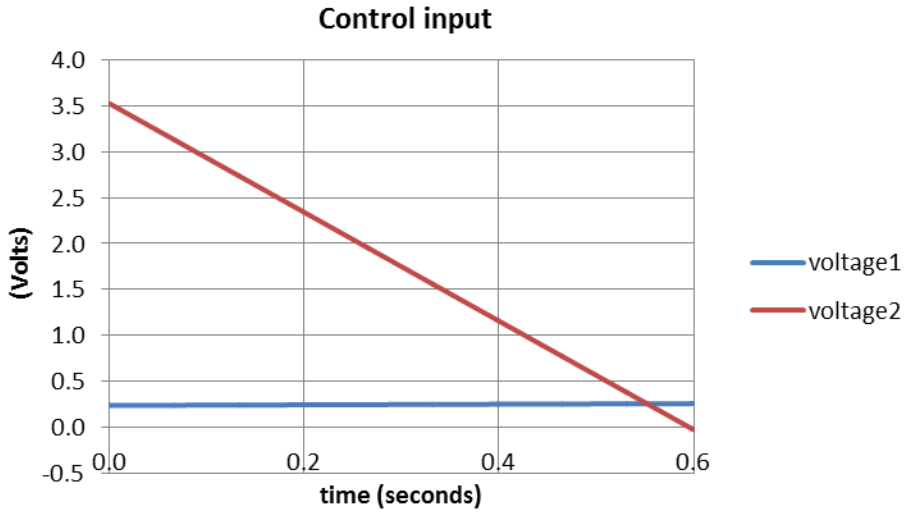



**Figure 26. Problem 4: Control input, (cr2s)**

b) The foot torque, $\alpha_1$, is limited by the size of the foot $L_{foot}$ as given in (2.2). To determine if the foot is of suitable foot size we plot $F_y L_{foot}$ (the maximum allowable torque)and $|\alpha_1|$as a function of time. For sufficiency $|\alpha_1|$ must be below $F_y L_{foot}$ for all time. As shown in Figure 27, $L_{foot} = 0.08\, meters$ satisfies this condition.

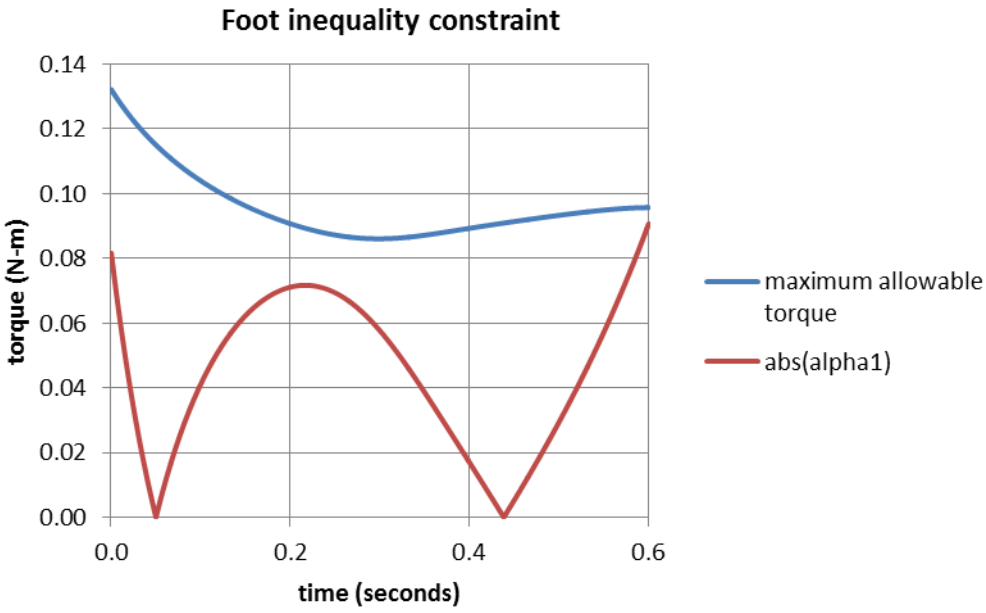



**Figure 27. Problem 4: Foot inequality constraint, (cr2s)**

d) The gear ratio of the motor is determined by modifying the ratio until the torque and rpm are within the motor specifications. Motor torques and rotational speeds are given below for a gear ratio of 40. They are below the maximum levels define for these motors.

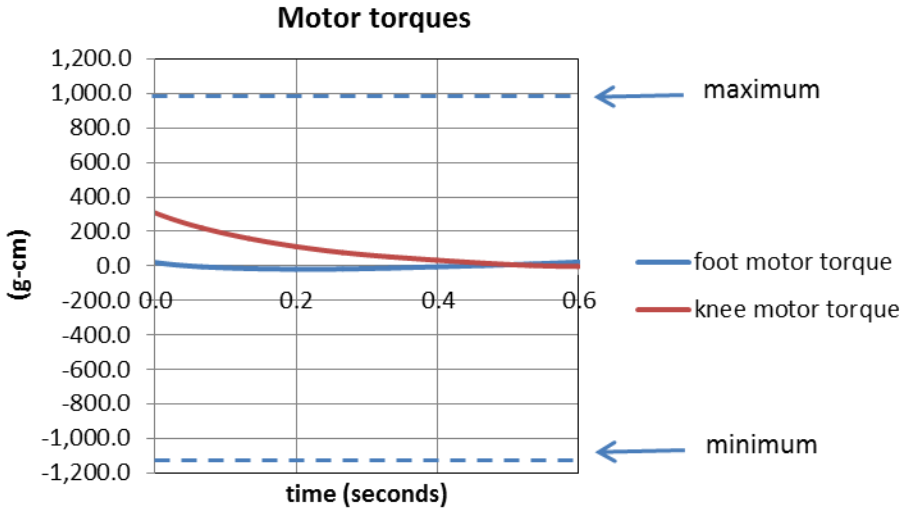



Figure 28. Problem 4: Motor torques, (cr2s)

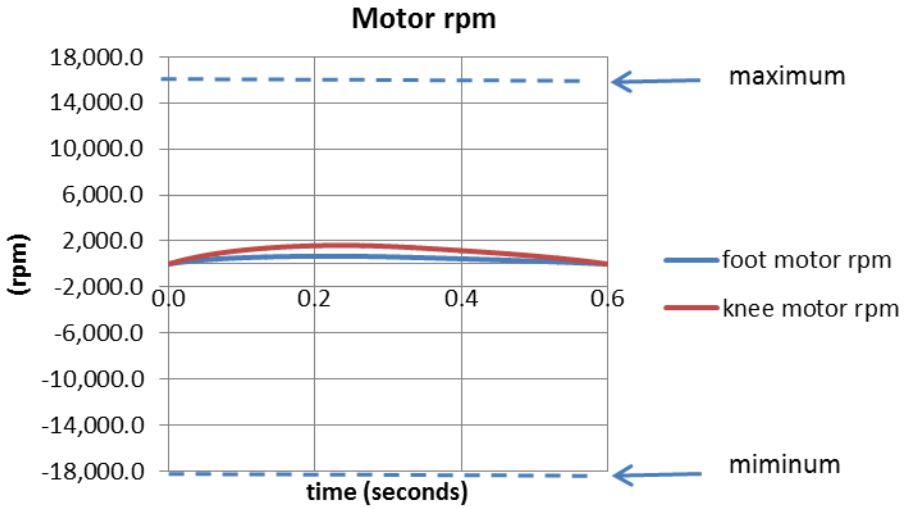



Figure 29. Problem 4: Motor rpm's, (cr2s)

## Problem 5. Rhythm: Standing to crouching right (s2cr)

**Given:**

The same conditions and parameters as given in problem 4 accept:

The initial condition is given by

$$\vec{X}_o = \begin{Bmatrix} \theta_1 = 0 \cdot \frac{\pi}{180} \\ \dot{\theta}_1 = 0 \\ \theta_2 = sin^{-1}\left\{-\frac{\frac{3}{2}m+M}{\frac{1}{2}m+M}\sin(\theta_1)\right\} \\ \dot{\theta}_2 = 0 \end{Bmatrix},$$

and the desired final condition is given by

$$\vec{X}_f = \begin{Bmatrix} \theta_1 = -40 \cdot \frac{\pi}{180} \\ \dot{\theta}_1 = 0 \\ \theta_2 = sin^{-1}\left\{-\frac{\frac{3}{2}m+M}{\frac{1}{2}m+M}\sin(\theta_1)\right\} \\ \dot{\theta}_2 = 0 \end{Bmatrix}.$$

- The final time $t_f = 1.0$ (sec).

**Find:**

- A solution which assumes that the control voltage inputs are simple linear function.

  That is,

$$V_1 = v_1 + \nu_1 t/t_f \text{ and } V_2 = v_2 + \nu_2 t/t_f$$

**Solution:**

- Results are given below.

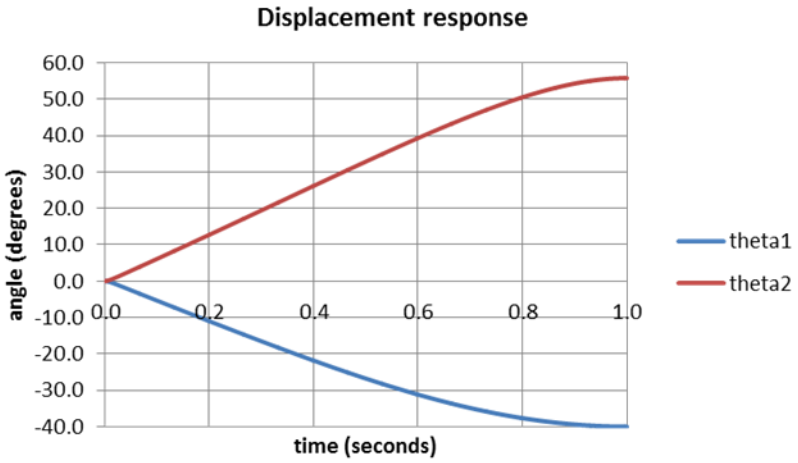



Figure 30. Problem 5: Displacement response, (s2cr)

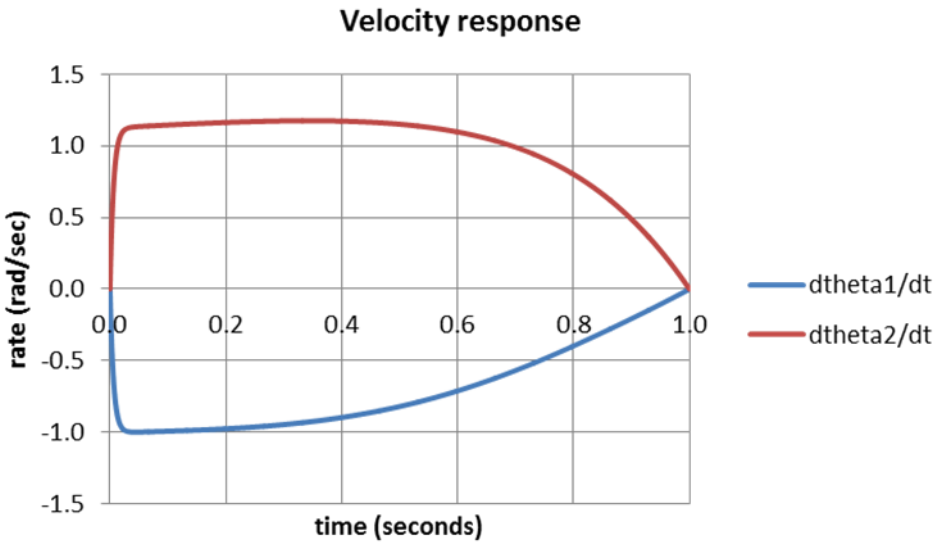



Figure 31. Problem 5: Velocity response, (s2cr)

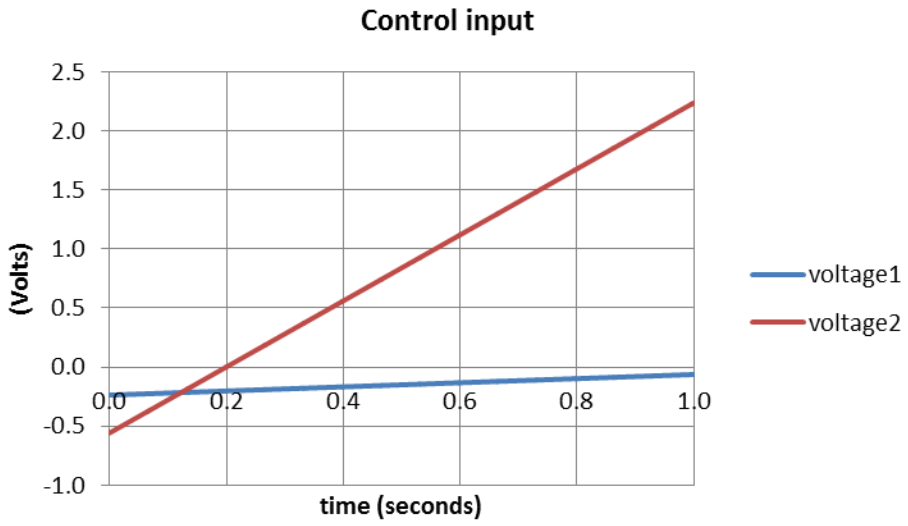



**Figure 32. Problem 5: Control input, (s2cr)**

---

### 3.2 Control of a system with changing dynamic order

Consider the situation of moving from a prone position to a crouching position. The two-link, in the prone positions, has only one input. It attempts to move to the intermediate state $(\theta_1, \theta_2) = (0, \pi)$ however, somewhere before reaching this position, its head must leave the ground and the number of control inputs increases to two. It then uses both control inputs to drive to the stabilized crouching state.

---

**Problem 6. Rhythm: Prone right to crouching right (p2cr)**

**Given:**

- Two-link parameters given in Appendix D,
- Nonlinear bent dynamics given by (2.10f)

$$m + 4\{m + 2M\}\cos^2(\theta_1)\,\ddot{\theta}_1 = 4\{m + 2M\}\cos(\theta_1)\sin(\theta_1)\,\dot{\theta}_1^2 + 2m\frac{g}{L}\sin(\theta_1) + \alpha/L^2 \qquad (2.10f)$$

and the reaction forces at the foot and head given by (2.11f) and (2.11g)

$$F_{y_2} = \{M + m\}g - \frac{mL\ddot{\theta}_1}{2\sin(\theta_1)} \qquad (2.11f)$$

$$F_{y_1} = 2(M+m)g - mL\left\{cos(\theta_1)\dot{\theta_1}^2 + sin(\theta_1)\ddot{\theta}_1\right\} - F_{y_2}. \tag{2.11g}$$

- Nonlinear pinned dynamics given by (2.7)

$$\begin{bmatrix} \left(\frac{5}{4}m+M\right)L^2 & \left(\frac{1}{2}m+M\right)L^2\cos(\theta_1-\theta_2) \\ \left(\frac{1}{2}m+M\right)L^2\cos(\theta_1-\theta_2) & \left(\frac{1}{4}m+M\right)L^2 \end{bmatrix}\begin{Bmatrix}\ddot{\theta}_1 \\ \ddot{\theta}_2\end{Bmatrix} =$$

$$\begin{Bmatrix} -\left(\frac{1}{2}m+M\right)L^2 sin(\theta_1-\theta_2)\dot{\theta}_2^2 + \left(\frac{3}{2}m+M\right)gLsin(\theta_1) + \tau_1 \\ \left(\frac{1}{2}m+M\right)L^2 sin(\theta_1-\theta_2)\dot{\theta}_1^2 + \left(\frac{1}{2}m+M\right)gLsin(\theta_2) + \tau_2 \end{Bmatrix}. \tag{2.7a}$$

and the reaction force as given by (2.8b)

$$F_y = -\left[m\left\{\frac{3L}{2}cos(\theta_1)\left(\dot{\theta}_1\right)^2 + \frac{L}{2}\cos(\theta_2)\left(\dot{\theta}_2\right)^2\right\} + m\left\{\frac{3L}{2}sin(\theta_1)\ddot{\theta}_1 + \frac{L}{2}\sin(\theta_2)\ddot{\theta}_2\right\} + M\left\{Lcos(\theta_1)\left(\dot{\theta}_1\right)^2 + Lcos(\theta_2)\left(\dot{\theta}_2\right)^2\right\} + M\left\{Lsin(\theta_1)\ddot{\theta}_1 + Lsin(\theta_2)\ddot{\theta}_2\right\}\right] + 2\{m+M\}g\,. \tag{2.8b}$$

Actuator dynamics as given by (2.23a) and (2.23b)

$$\alpha_1 = GR\left[\frac{K_Q}{R}V_1 - \frac{K_Q}{R\,K_v}\dot{\theta}_1\right] \tag{2.23a}$$

$$\alpha_2 = GR\left[\frac{K_Q}{R}V_2 - \frac{K_Q}{R\,K_v}\left(\dot{\theta}_1 - \dot{\theta}_2\right)\right] \tag{2.23b}$$

- The initial condition

$$\vec{X}_o = \begin{Bmatrix} \theta_1 = 90\cdot\frac{\pi}{180} \\ \dot{\theta}_1 = 0 \\ \theta_2 = 90\cdot\frac{\pi}{180} \\ \dot{\theta}_2 = 0 \end{Bmatrix},$$

and the desired final condition

$$\vec{X}_f^{\,d} = \begin{Bmatrix} \theta_1 = -40\cdot\frac{\pi}{180} \\ \dot{\theta}_1 = 0 \\ \theta_2 = sin^{-1}\left\{-\frac{\frac{3}{2}m+M}{\frac{1}{2}m+M}\sin(\theta_1)\right\} \\ \dot{\theta}_2 = 0 \end{Bmatrix}.$$

**Find:** An $V_1(t)$ and $V_2(t)$ that will move the two-link from $\vec{X}_o$ to $\vec{X}_f^{\,d}$ with limited foot torque.

**Solution:**

The solution consists of two parts. Initially, in part 1, the two-link is in the prone configuration $(\theta_1, \theta_2) = (\pi/2\,, \pi/2)$ and its dynamics are bent dynamics. It begins by attempting to reach the state $(\theta_1, \theta_2) = (0, \pi)$ using a linear control excitation as given in (3.1a) for N=1. The state response and the time history of the reaction forces are given in the below figures. The head reaction force goes to zero at about 0.24 seconds.

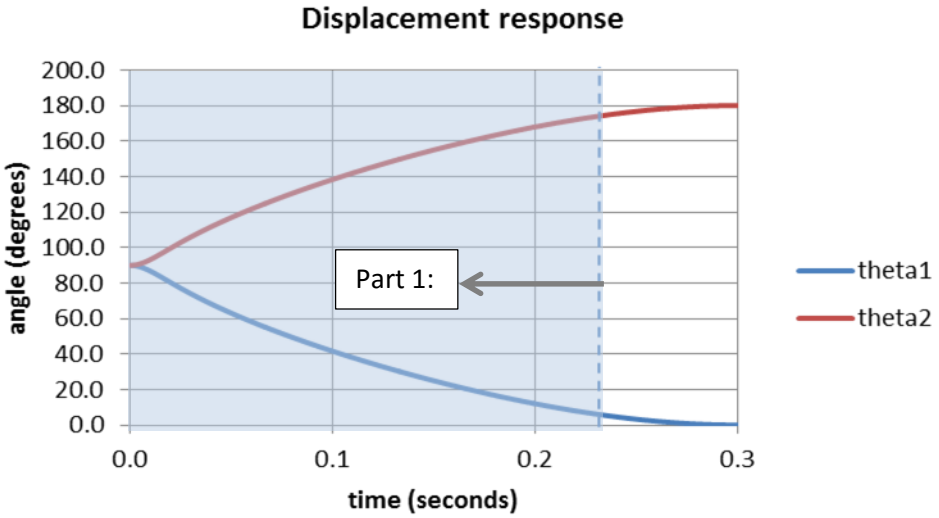



**Figure 33. Problem 6: Displacement response before transition, (p2cr)**

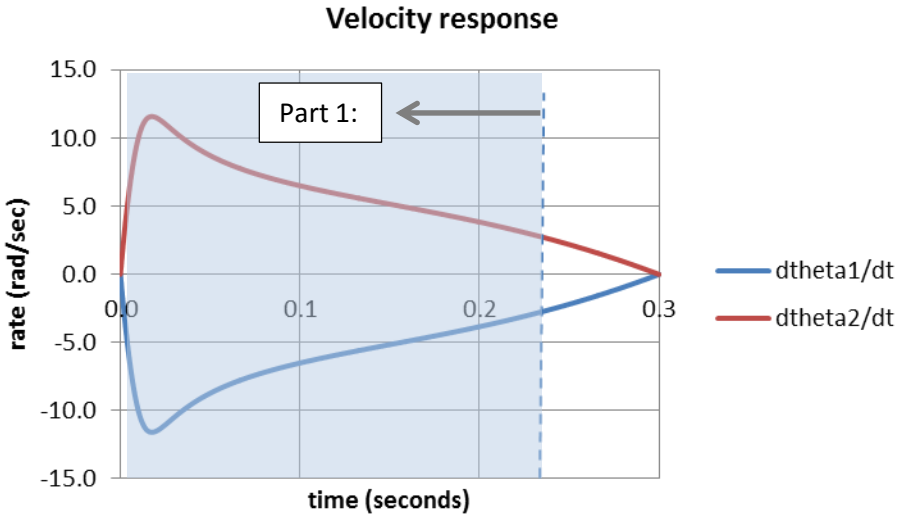



**Figure 34. Problem 6: Velocity response before transition, (p2cr)**

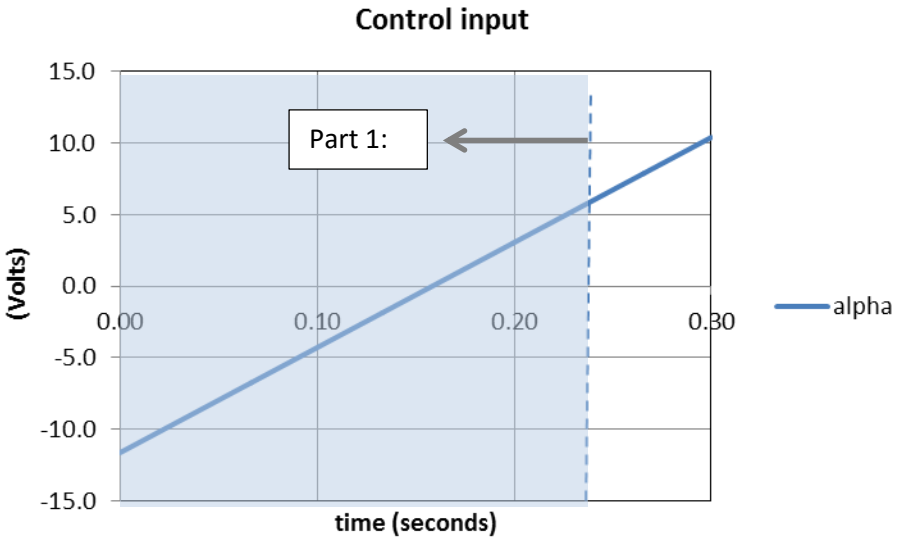



Figure 35. Problem 6: Control input before transition, (p2cr)

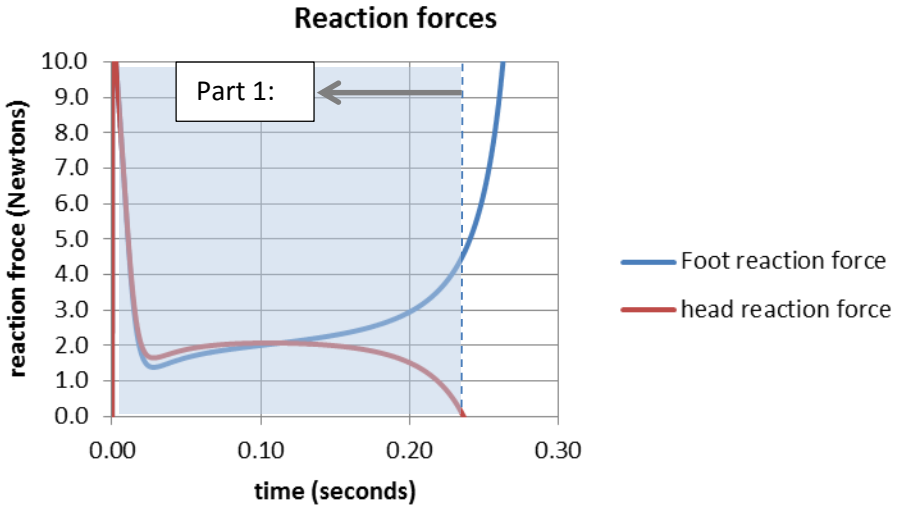



Figure 36. Problem 6: Reaction force before transition, (p2cr)

As shown in Figure 36, the reaction force at the head goes negative at 0.24 seconds. At this time, the head *must* leave the ground and the dynamics must change from bent dynamics to pinned dynamics. Slightly before a negation of the reaction force is the end of part 1 and the beginning of part 2 of the solution. In part 2, control drives the two-link into a crouching right position.

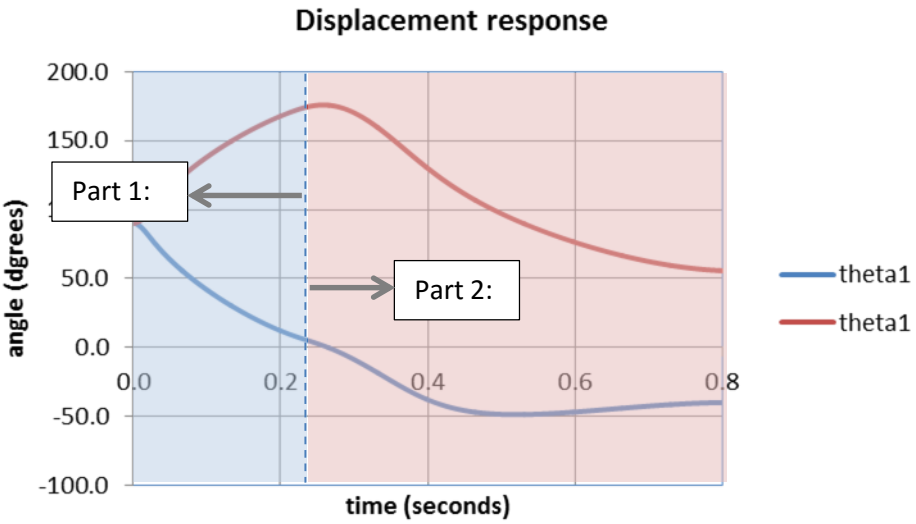



Figure 37. Problem 4: Displacement response of two-link, (p2cr)

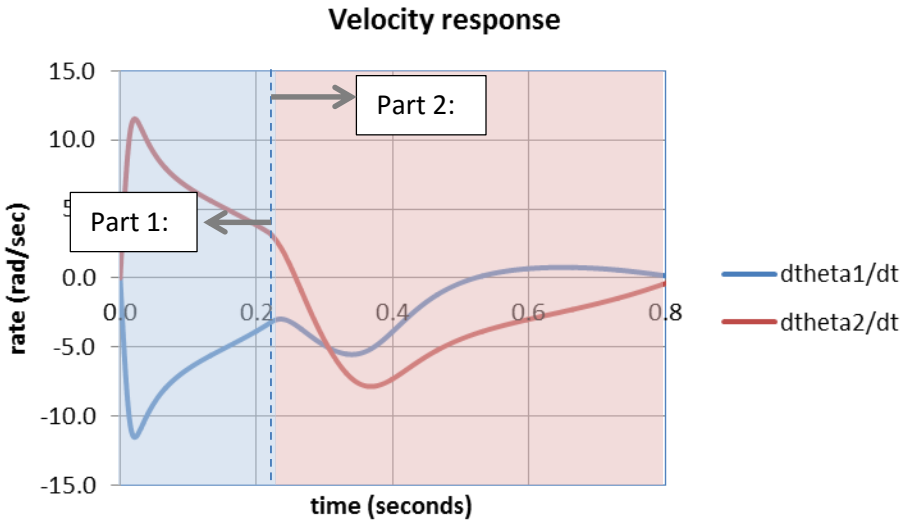



Figure 38. Problem 6: Velocity response of two-link, (p2cr)

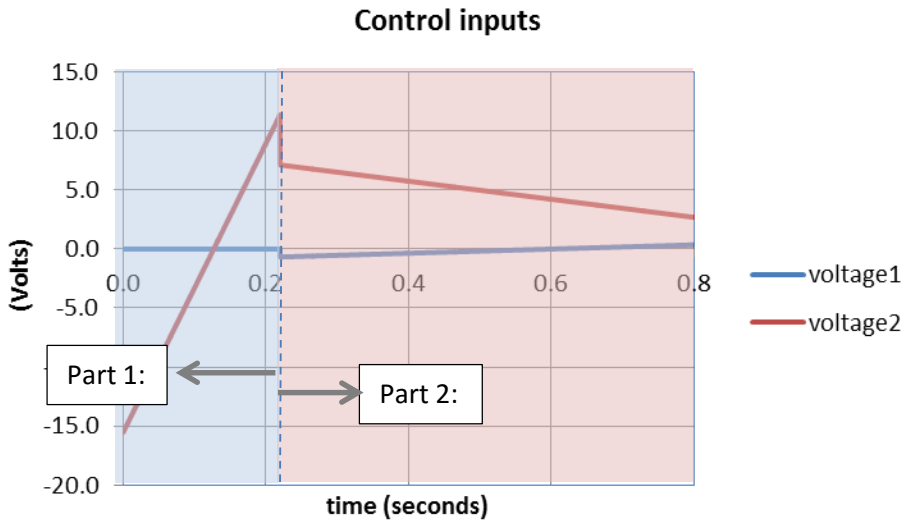



Figure 39. Problem 6: Control input, (p2cr)

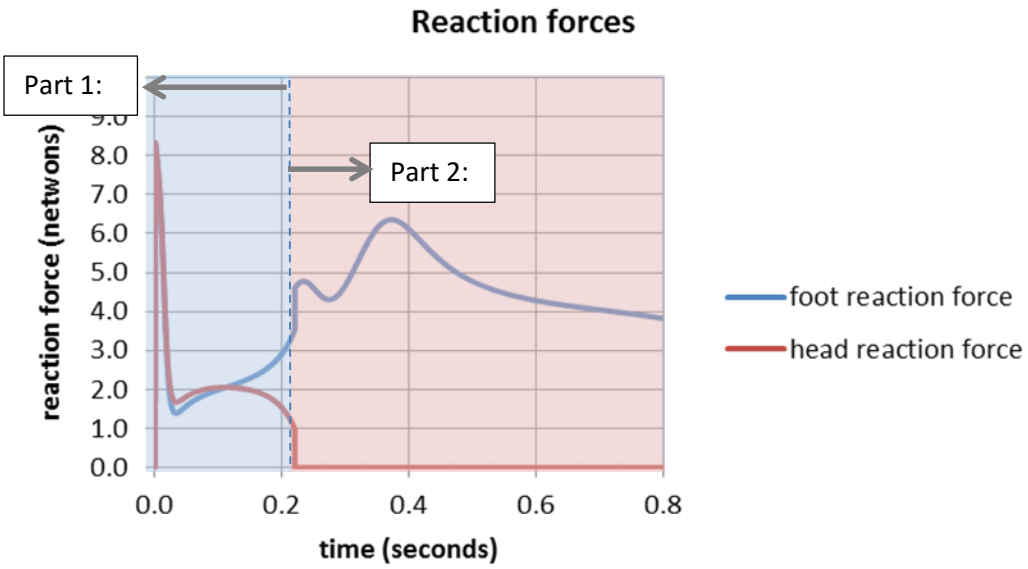



Figure 40. Problem 6: Reaction force at the foot and the head, (p2cr)

As before, the inequality constraint must be satisfied. During the first part of the solution, this condition is naturally satisfied considering that there is no foot torque; however, in part 2, a foot torque is used to upright the two-link.

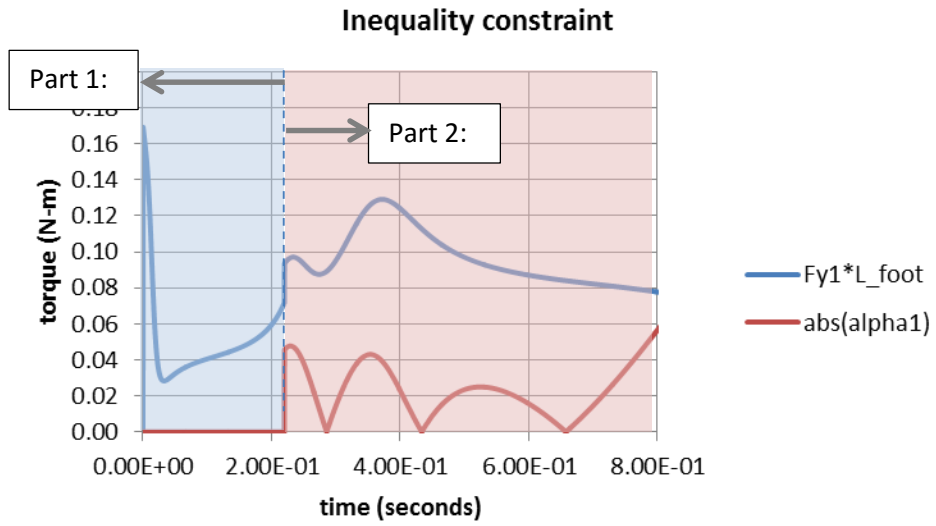



**Figure 41. Problem 6; Inequality constraint, eq. (2.1)**

---

The below problem illustrates the derivation of a controlled fall from a crouching right position to a prone position.

---

**Problem 7. Rhythm: Crouching right to prone right (cr2p)**

**Given:**

- Two-link parameters given in Appendix D,
- Nonlinear pinned dynamics given by (2.7)

$$\begin{bmatrix} \left(\frac{5}{4}m+M\right)L^2 & \left(\frac{1}{2}m+M\right)L^2\cos(\theta_1-\theta_2) \\ \left(\frac{1}{2}m+M\right)L^2\cos(\theta_1-\theta_2) & \left(\frac{1}{4}m+M\right)L^2 \end{bmatrix} \begin{Bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{Bmatrix} =$$

$$\begin{Bmatrix} -\left(\frac{1}{2}m+M\right)L^2 sin(\theta_1-\theta_2)\dot{\theta}_2{}^2 + \left(\frac{3}{2}m+M\right)gLsin(\theta_1) + \tau_1 \\ \left(\frac{1}{2}m+M\right)L^2 sin(\theta_1-\theta_2)\dot{\theta}_1{}^2 + \left(\frac{1}{2}m+M\right)gLsin(\theta_2) + \tau_2 \end{Bmatrix}, \quad (2.7a)$$

and the reaction force as given by (2.8b)

$$F_y = -\left[m\left\{\frac{3L}{2}cos(\theta_1)\left(\dot{\theta}_1\right)^2 + \frac{L}{2}\cos(\theta_2)\left(\dot{\theta}_2\right)^2\right\} + m\left\{\frac{3L}{2}sin(\theta_1)\ddot{\theta}_1 + \frac{L}{2}\sin(\theta_2)\ddot{\theta}_2\right\} + \right.$$

$$\left. M\left\{Lcos(\theta_1)\left(\dot{\theta}_1\right)^2 + Lcos(\theta_2)\left(\dot{\theta}_2\right)^2\right\} + M\left\{Lsin(\theta_1)\ddot{\theta}_1 + Lsin(\theta_2)\ddot{\theta}_2\right\}\right] + 2\{m+M\}g\,, \tag{2.8b}$$

- Actuator dynamics are given by (2.23a) and (2.23b)

$$\alpha_1 = GR\left[\frac{K_Q}{R}V_1 - \frac{K_Q}{R\,K_v}\dot{\theta}_1\right] \tag{2.23a}$$

$$\alpha_2 = GR\left[\frac{K_Q}{R}V_2 - \frac{K_Q}{R\,K_v}\left(\dot{\theta}_1 - \dot{\theta}_2\right)\right] \tag{2.23b}$$

where from (2.1)

$$\begin{Bmatrix}\tau_1\\ \tau_2\end{Bmatrix} = \frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{Bmatrix}\alpha_1\\ \alpha_2\end{Bmatrix}. \tag{2.1}$$

- Nonlinear bent dynamics given by (2.10f)

$$m + 4\{m+2M\}\cos^2(\theta_1)\,\ddot{\theta}_1 = 4\{m+2M\}\cos(\theta_1)\sin(\theta_1)\,\dot{\theta}_1^2 + 2m\frac{g}{L}\sin(\theta_1) + \alpha/L^2\,, \tag{2.10f}$$

and the reaction forces at the foot and head given by (2.11f) and (2.11g)

$$F_{y_2} = \{M+m\}g - \frac{mL\ddot{\theta}_1}{2\sin(\theta_1)}, \tag{2.11f}$$

$$F_{y_1} = 2(M+m)g - mL\left\{cos(\theta_1)\dot{\theta}_1{}^2 + sin(\theta_1)\ddot{\theta}_1\right\} - F_{y_2}, \tag{2.11g}$$

- Impact dynamics given by (2.17), and (2.19)

$$\phi = A\psi \tag{2.17a}$$

where

$$\phi = [\phi_u{}^T \quad \phi_l{}^T]^T\,,$$

$$\phi_u = \left[v_{x,1}^- \quad v_{y,1}^- \quad \dot{\theta}_1^- \quad v_{x,2}^- \quad v_{y,2}^- \quad \dot{\theta}_2^- \quad 0 \quad 0\right]^T, \phi_l = [0 \quad 0 \quad 0]^T,$$

$$\psi = [\psi_u{}^T \quad \psi_l{}^T]^T\,,$$

$$\psi_u = \left[v_{x,1}^+ \quad v_{y,1}^+ \quad \dot{\theta}_1^+ \quad v_{x,2}^+ \quad v_{y,2}^+ \quad \dot{\theta}_2^+ \quad f_x \quad f_y\right]^T, \psi_l = \left[F_{x_1}, F_{y_1}, F_{y_2}\right]^T,$$

$$v_{x,1}^- = \left(L - L_{cg}\right)cos(\theta_1)\dot{\theta}_1^-,$$

$$v_{y,1}^- = -\left(L - L_{cg}\right)\sin(\theta_1)\dot{\theta}_1^-,$$

$$v_{x,2}^{-} = \quad Lcos(\theta_1)\dot{\theta}_1^{-} + \left(L - L_{cg}\right)cos(\theta_2)\dot{\theta}_2^{-},$$

$$v_{y,2}^{-} = -Lsin(\theta_1)\dot{\theta}1 - \left(L - L_{cg}\right)sin(\theta_2)\dot{\theta}_2^{-},$$

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

$A_{11} =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1/(m+M) & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1/(m+M) \\ 0 & 0 & 1 & 0 & 0 & 0 & (L-L_{cg})\cos(\theta_1)/I_1 & -(L-L_{cg})\sin(\theta_1)/I_1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1/(m+M) & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1/(m+M) \\ 0 & 0 & 0 & 0 & 0 & 1 & (L-L_{cg})\cos(\theta_1)/I_2 & -(L-L_{cg})\sin(\theta_1)/I_2 \\ -1 & 0 & -(L-L_{cg})\cos(\theta_1) & 1 & 0 & -(L-L_{cg})\cos(\theta_2) & 0 & 0 \\ 0 & -1 & (L-L_{cg})\sin(\theta_1) & 0 & 1 & (L-L_{cg})\sin(\theta_2) & 0 & 0 \end{bmatrix},$$

(2.17b)

$$A_{12} = \begin{bmatrix} -1/(m+M) & 0 & 0 \\ 0 & -1/(m+M) & 0 \\ L_{cg}\cos(\theta_1)/I_1 & -L_{cg}\sin(\theta_1)/I_1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1/(m+M) \\ 0 & 0 & L_{cg}\sin(\theta_2)/I_2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \qquad (2.19c)$$

$$A_{21} = \begin{bmatrix} 1 & 0 & -L_{cg}\cos(\theta_1) & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & L_{cg}\sin(\theta_1) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -L_{cg}\sin(\theta_2) & 0 & 0 \end{bmatrix}, \qquad (2.19d)$$

$A_{22} = 0^{NxN}$.

- The initial condition

$$\vec{X}_o = \begin{Bmatrix} \theta_1 = -40 \cdot \frac{\pi}{180} \\ \dot{\theta}_1 = 0 \\ \theta_2 = sin^{-1}\left\{-\frac{\frac{3}{2}m+M}{\frac{1}{2}m+M}\sin(\theta_1)\right\} \\ \dot{\theta}_2 = 0 \end{Bmatrix},$$

and the desired final condition

$$\vec{X}_f^{\,d} = \begin{Bmatrix} \theta_1 = 90 \cdot \frac{\pi}{180} \\ \dot{\theta}_1 = 0 \\ \theta_2 = 90 \cdot \frac{\pi}{180} \\ \dot{\theta}_2 = 0 \end{Bmatrix}.$$

**Find:** An $\alpha_1(t)$ and $\alpha_2(t)$ that will allow the two-link to fall from a crouching right position into a prone position.

**Solution:**

As in problem 6, the solution is comprised of multiple parts. Responses and control inputs are given below

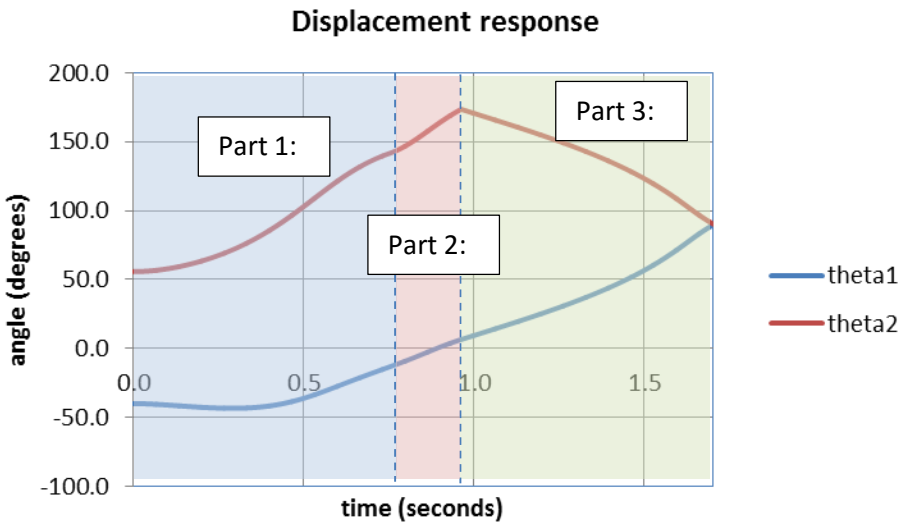



**Figure 42. Problem 7: Displacement response, (cr2p)**

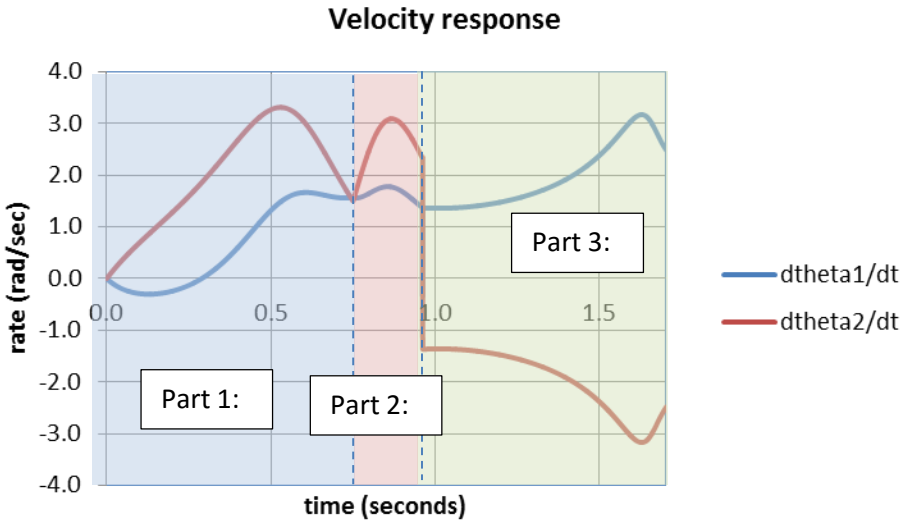



Figure 43. Problem 7: Velocity response, (cr2p)

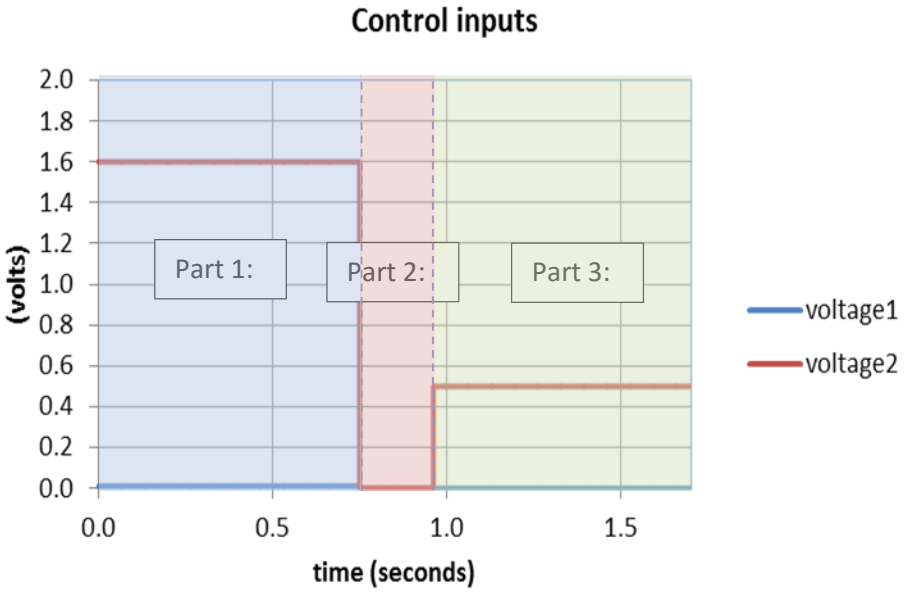



Figure 44. Problem 7: Control inputs, (cr2p)

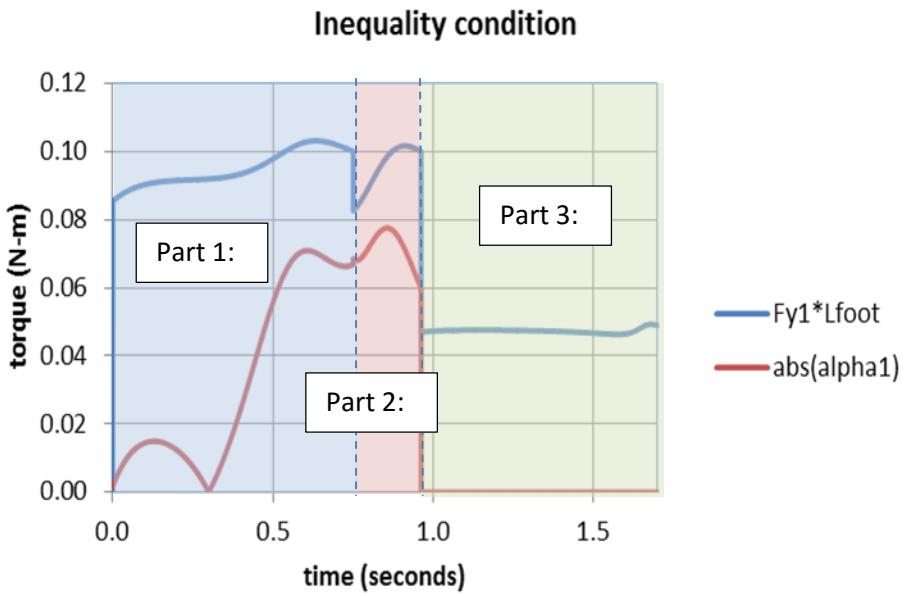



**Figure 45. Problem 7: Inequality constraint, (cr2p)**

---

Problem 7 involved a limited level of control. In essence it was a controlled fall.

---

**Problem 8. Rhythm: Jumping to crouching right (j2cr)**

**Given:**

- Two-link parameters given in Appendix D,
- Nonlinear pinned dynamics given by (2.7)
- Nonlinear free dynamics given by (2.5)
- Free to pinned impact dynamics given by (2.17) and (2.18)
- The initial condition

$$\vec{X}_o = \begin{Bmatrix} \theta_1 = -40 \cdot \frac{\pi}{180} \\ \dot{\theta}_1 = 0 \\ \theta_2 = sin^{-1}\left\{-\frac{\frac{3}{2}m+M}{\frac{1}{2}m+M}\sin(\theta_1)\right\} \\ \dot{\theta}_2 = 0 \\ x = x_o \\ \dot{x} = 0 \\ y = 0 \\ \dot{y} = 0 \end{Bmatrix},$$

and the desired final condition

$$\vec{X}_f = \left\{ \begin{array}{c} \theta_1 = -40 \cdot \frac{\pi}{180} \\ \dot{\theta}_1 = 0 \\ \theta_2 = sin^{-1}\left\{ -\frac{\frac{3}{2}m+M}{\frac{1}{2}m+M} \sin(\theta_1) \right\} \\ \dot{\theta}_2 = 0 \\ x = x_f \\ \dot{x} = 0 \\ y = 0 \\ \dot{y} = 0 \end{array} \right\} \text{where } x_f > x_o.$$

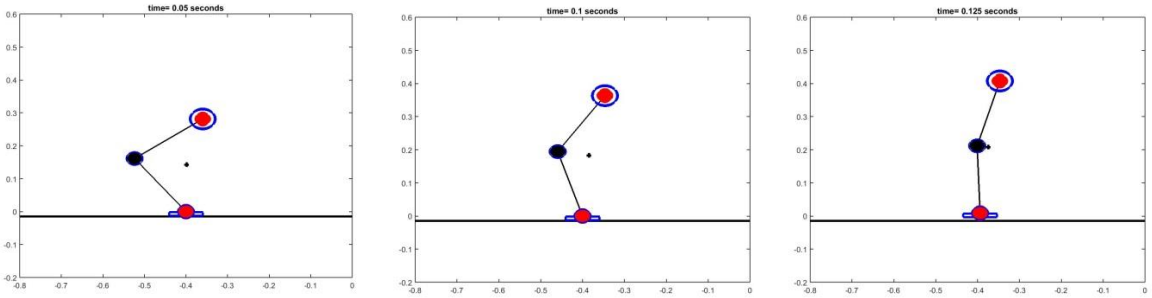
**Find:** An $\alpha_1(t)$ and $\alpha_2(t)$ that will allow the two-link to hope to $x_f > x_o$.

**Solution:**

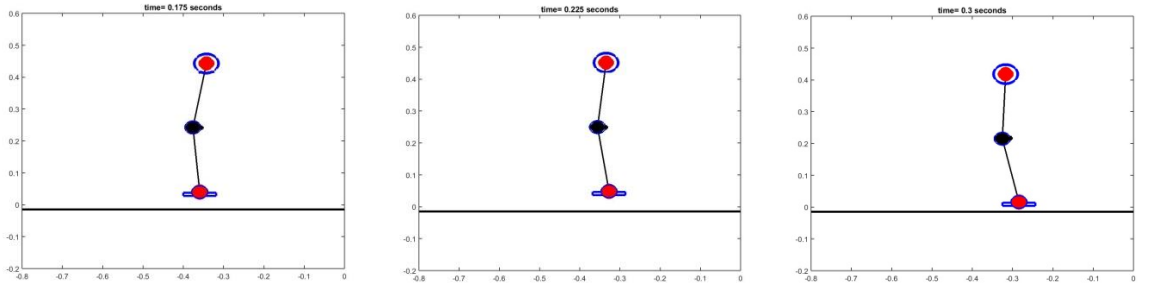
The jump solution consists of four parts.

1) The jump: The two-link extends effort at the knee joint while tilting forward with the foot. The reaction force at the foot will be positive for only a limited time during this extension. When this force goes to zero, the two-link transitions to flight dynamics.
2) The flight: The two-link undergoes under-actuated control through the knee torque. This control changes the orientation of the two-link on impact.
3) The impact: When the foot impacts the ground, the dynamics of the two-link shifts back from flight dynamics to pinned dynamics. The effect of this impact is to alter the velocity states such that the motion of the impacting foot goes to zero. If the kinetic energy of the two-link just after impact is too high, recovering back to a crouching position is difficult. To minimized this energy, the orientation of the two-link during fight is adjusted
4) The recovery: After impact, the two-link will be off-balance. It must exert a set of torques that moves it back to the couching state.

**The jump:** The two-link aims for a direction in space and extends its full effort in this direction.




**The flight:** The foot is off the ground. The two-link undergoes under-actuated control



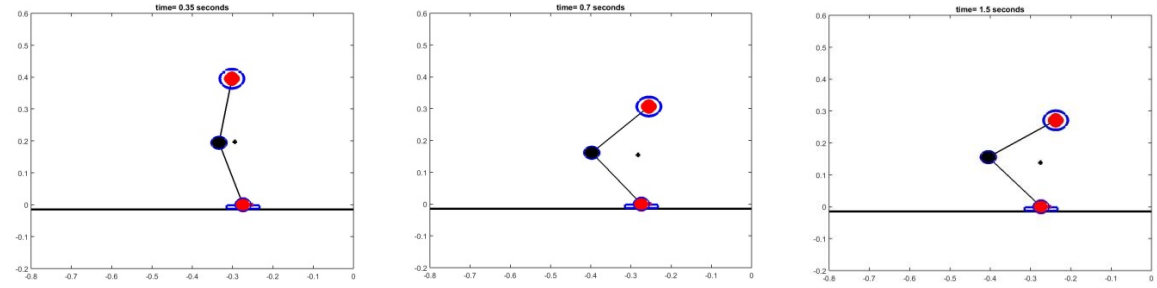




**Figure 46. Problem 8: The jump, flight, impact, and recovery**

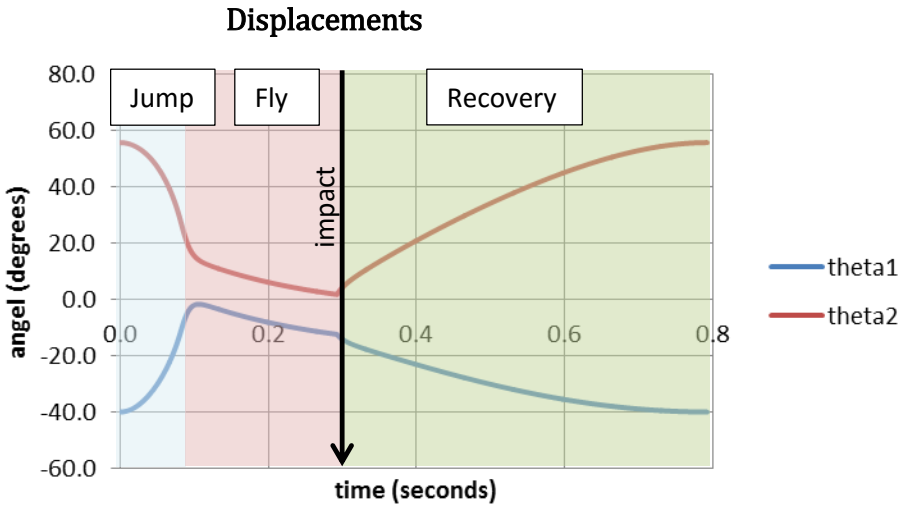



Figure 47. Problem 8: Angular displacements of two-link, (j2cr)

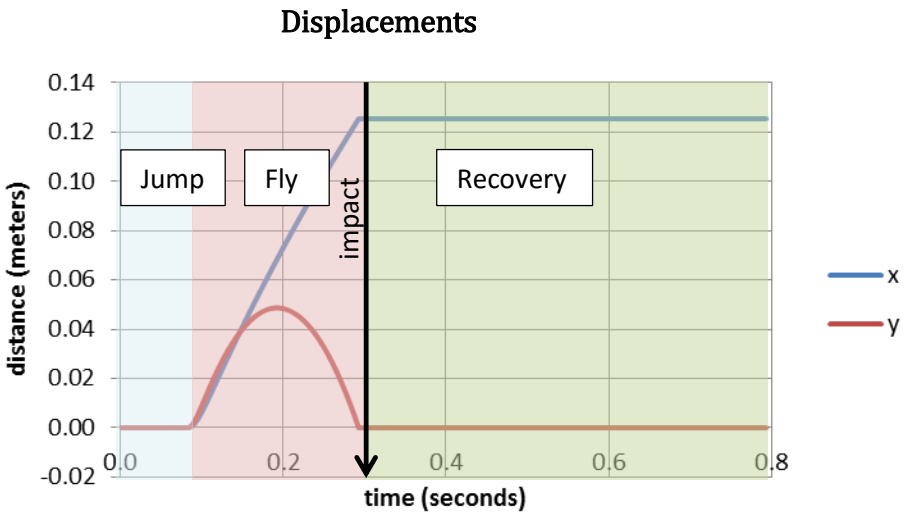



Figure 48. Problem 8: Displacement of two-link foot, (j2cr)

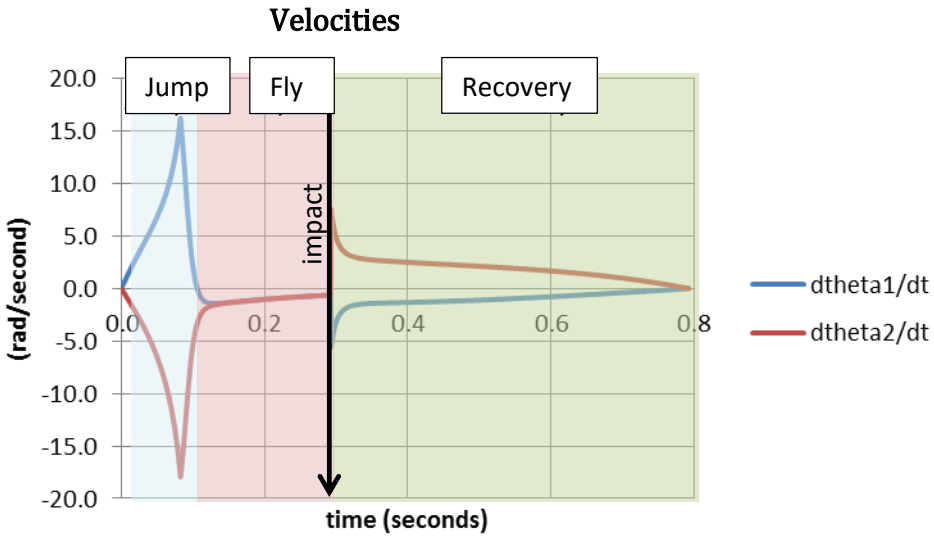



Figure 49. Problem 8: Angular velocity response of two-link, (j2cr)

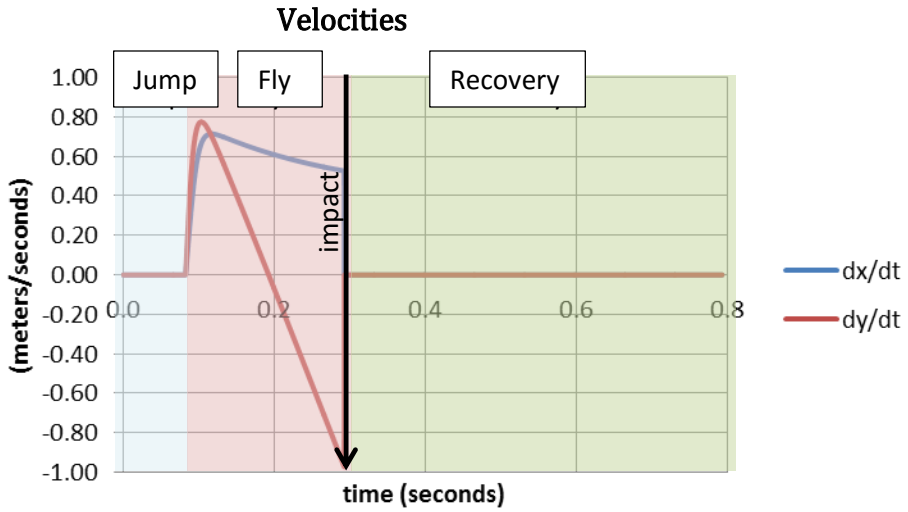



Figure 50. Problem 8: Velocity response of two-link foot, (j2cr)

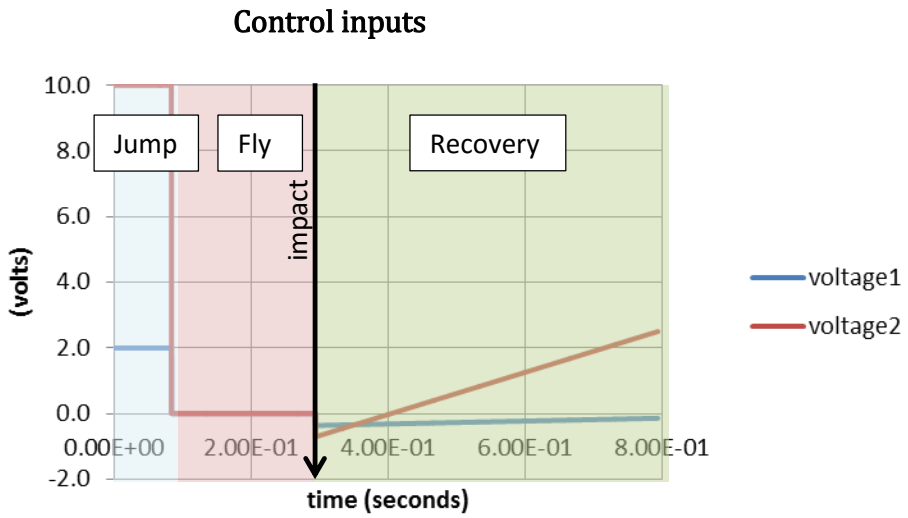



Figure 51. Problem 8: Input voltages, (j2cr)

---

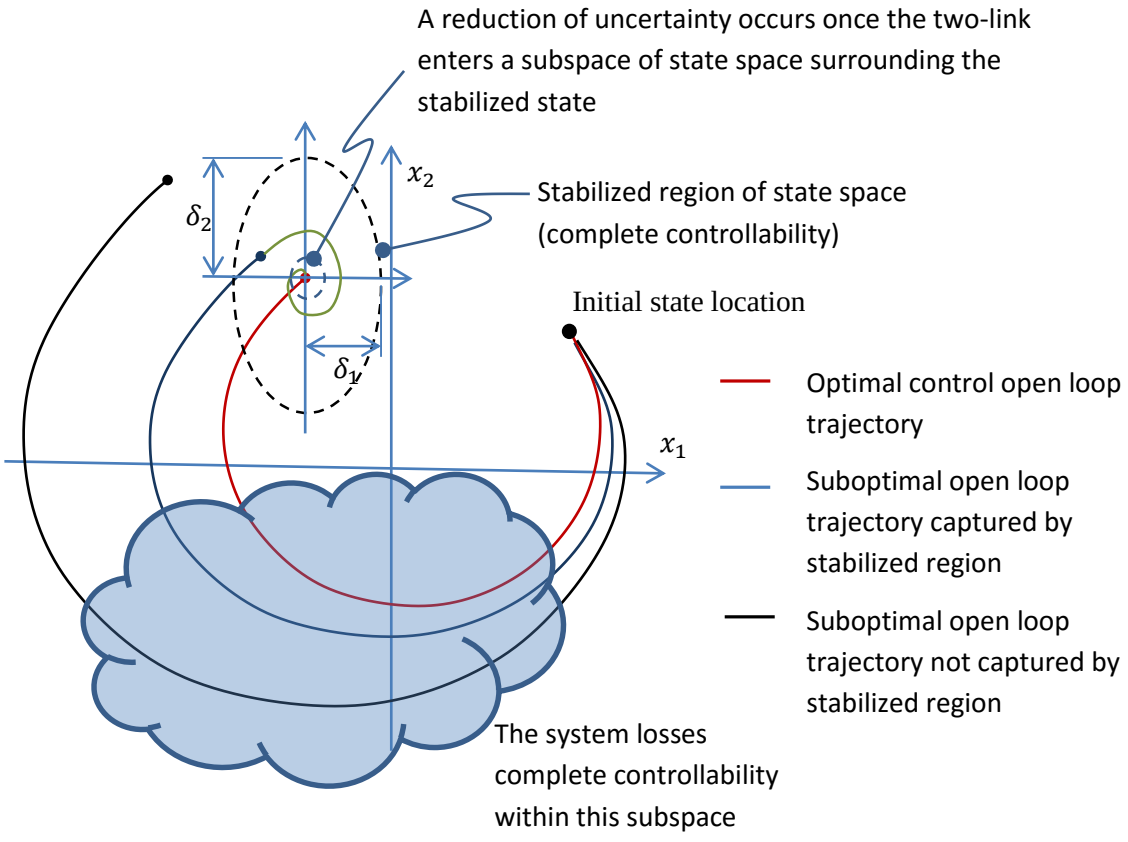



Figure 52. An illustration of the capture subdomain and uncertainty reduction

### 3.3 Stabilizing controllers

As shown in Figure 52 the OLC input throws the dynamics of the system through the state space where it is captured in a stabilized subspace. If this pitch is too much in error, the trajectory misses the subspace and a capture may not occur. For a capture to occur, the pitch does not have to be perfect. The pitch need only be as accurate as is necessary to land within the capture subspace. Once in this subspace, closed loop control is used to pull the trajectory to a smaller subspace about the stabilized state. Pratt et. al [69] define a similar concept called a capture point. This is "the point on the ground where the robot can step in order to bring itself to a complete stop."

In this section a linear quadratic regulator (LQR) is used to capture dynamics. In later chapters discussing two-link hardware implementation, we show that a proportional controller works just as well. The time response of the LQR is dependent upon the minimum depth of the closed loop poles (i.e. $eig\{A - BK\}$) into the left half plane (LHP). If $Q = \gamma I^{8x8}$ as $\gamma$ become progressively larger the weighting on states is increased and the relative weighting on the control is reduced. The poles of the closed-loop system migrate deeper into the LHP however this migration is bounded. Neglecting foot size, this implies that the speed at which the system can respond after capture is bounded.

The size of the foot limits response times since a higher gain controller will require a greater torque from the foot but the size of the foot is limited and therefore, so is this torque. Bounded torque implies bounded response time. In the following problem, a capture region is quantified by calculating the distance along the axis of a hyper-ellipsoid in state space that will capture system dynamics.

**Problem 9. Stabilizing controller: Standing upright (bals)**

**Given:**

- Two-link parameters given in Appendix D.
- Linearized dynamics as given by (2.20.a) ( Appendix D ), and (21.a)

$$\frac{d}{dt}\begin{Bmatrix}\theta_1-\theta_1^{\,*}\\ \dot\theta_1-\dot\theta_1^{\,*}\\ \theta_2-\theta_2^{\,*}\\ \dot\theta_2-\dot\theta_2^{\,*}\end{Bmatrix}=\begin{bmatrix}0 & 1 & 0 & 0\\ \left.\frac{\partial f_1}{\partial\theta_1}\right|_{\vec X^*} & \left.\frac{\partial f_1}{\partial\dot\theta_1}\right|_{\vec X^*} & \left.\frac{\partial f_1}{\partial\theta_2}\right|_{\vec X^*} & \left.\frac{\partial f_1}{\partial\dot\theta_2}\right|_{\vec X^*}\\ 0 & 0 & 0 & 1\\ \left.\frac{\partial f_2}{\partial\theta_1}\right|_{\vec X^*} & \left.\frac{\partial f_2}{\partial\dot\theta_1}\right|_{\vec X^*} & \left.\frac{\partial f_2}{\partial\theta_2}\right|_{\vec X^*} & \left.\frac{\partial f_2}{\partial\dot\theta_2}\right|_{\vec X^*}\end{bmatrix}\begin{Bmatrix}\theta_1-\theta_1^{\,*}\\ \dot\theta_1-\dot\theta_1^{\,*}\\ \theta_2-\theta_2^{\,*}\\ \dot\theta_2-\dot\theta_2^{\,*}\end{Bmatrix}+\begin{bmatrix}0 & 0\\ \left.\frac{\partial f_1}{\partial v_1}\right|_{\vec X^*} & \left.\frac{\partial f_1}{\partial v_2}\right|_{\vec X^*}\\ 0 & 0\\ \left.\frac{\partial f_2}{\partial v_1}\right|_{\vec X^*} & \left.\frac{\partial f_2}{\partial v_2}\right|_{\vec X^*}\end{bmatrix}\begin{Bmatrix}v_1-v_1^{\,*}\\ v_2-v_2^{\,*}\end{Bmatrix}$$

(2.20a)

$$\vec X^*=\begin{Bmatrix}\theta_1\\ \dot\theta_1\\ \theta_2\\ \dot\theta_2\end{Bmatrix}=\begin{Bmatrix}0\\ 0\\ 0\\ 0\end{Bmatrix}\text{ under the steady state excitation } v_1^{\,*}=v_2^{\,*}=0. \tag{2.21a}$$

- Nonlinear dynamics as given by (2.7)

$$\begin{bmatrix}\left(\frac{5}{4}m+M\right)L^2 & \left(\frac{1}{2}m+M\right)L^2\cos(\theta_1-\theta_2)\\ \left(\frac{1}{2}m+M\right)L^2\cos(\theta_1-\theta_2) & \left(\frac{1}{4}m+M\right)L^2\end{bmatrix}\begin{Bmatrix}\ddot\theta_1\\ \ddot\theta_2\end{Bmatrix}=$$

$$\begin{Bmatrix}-\left(\frac{1}{2}m+M\right)L^2sin(\theta_1-\theta_2)\dot\theta_2^{\,2}+\left(\frac{3}{2}m+M\right)gLsin(\theta_1)+\tau_1\\ \left(\frac{1}{2}m+M\right)L^2sin(\theta_1-\theta_2)\dot\theta_1^{\,2}+\left(\frac{1}{2}m+M\right)gLsin(\theta_2)+\tau_2\end{Bmatrix}.$$

(2.7a)

- Actuator dynamics are given by (2.23a) and (2.23b)

$$\alpha_1=GR\left[\frac{K_Q}{R}V_1-\frac{K_Q}{R\,K_v}\dot\theta_1\right] \tag{2.23a}$$

$$\alpha_2=GR\left[\frac{K_Q}{R}V_2-\frac{K_Q}{R\,K_v}(\dot\theta_1-\dot\theta_2)\right] \tag{2.23b}$$

where from (2.1)

$$\begin{Bmatrix}\tau_1\\ \tau_2\end{Bmatrix} = \frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{Bmatrix}\alpha_1\\ \alpha_2\end{Bmatrix}. \tag{2.1}$$

**Find:**

a) A LQG feedback controller with $1/\Re e(\lambda_{min}) < 0.5\ seconds$.

b) For the controller in a), the extent of the capture subspace.

**Solution:**

a) The linearized equations of motion give

$$A = \begin{bmatrix}0 & 1 & 0 & 0\\ 451 & -225 & -378 & 73.2\\ 0 & 0 & 0 & 1\\ -486 & 253 & 459 & -78\end{bmatrix}, B = \begin{bmatrix}0 & 0\\ -7.64 & 206\\ 0 & 0\\ 22.9 & -237\end{bmatrix}$$

The open loop poles of the system given by (2.20a) and (2.21a) are

$$\lambda_i = \ [-310 \ \ -4.33 \ \ 2.36 \ \ 7.39].$$

As expected, the system is unstable. Setting $\gamma = 20$ and $Q = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 10 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 10\end{bmatrix}$ gives a LQR feedback matrix of

$$F = \begin{bmatrix}5.08 & 1.05 & 2.99 & 0.960\\ 3.85 & -0.236 & -3.96 & -0.552\end{bmatrix}.$$

This results in a set of closed loop eigenvalues

$$\lambda_i = \ [-384 \ \ -11.2 \ \ -2.07 \ \ -2.66]$$

which are all real and negative. The eigenvalue closest to the imaginary axis has a time constant of

$$1/\Re e(\lambda_{min}) < 0.483\ seconds$$

which meets the requirements.

b) An estimate of the capture subspace is found by implementing the LQR solution on (2.7a) and perturbing the initial conditions. For the situation where $\vec{X}(t=0) = \vec{X}^*$, the solution remains at $\vec{X}$ and the foot torque, , $\alpha_1$ is zero. For the situation where one of the states is perturbed $\alpha_1$ is time varying; however since the foot toque is bounded by (2.2),

perturbations are limited. The limits of these perturbations define an estimate of the capture subdomain.

For example, for $\vec{X}(t=0) = \begin{Bmatrix} \delta_1 = 0.08 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$, the reaction force times the foot length as a function of time and $|\alpha_1|$ as a function of time are as given in Figure 53. Between zero and 0.025 seconds, the (2.2) constraint is violated. This perturbation of $\vec{X}$ is outside of the capture subdomain as defined here.

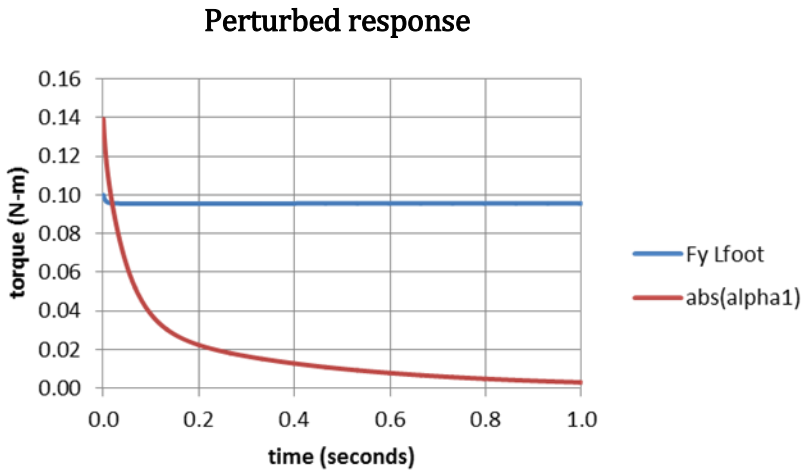



**Figure 53. Problem 9: Foot torque, failure of (2.2) inequality constraint, (bals)**

If we reduce the perturbation by letting $\vec{X}(t=0) = \begin{Bmatrix} \delta_1 = 0.055 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$, the reaction force times the foot length as a function of time and $|\alpha_1|$ as a function of time are as given in Figure 54 . This perturbation is within the capture subdomain.

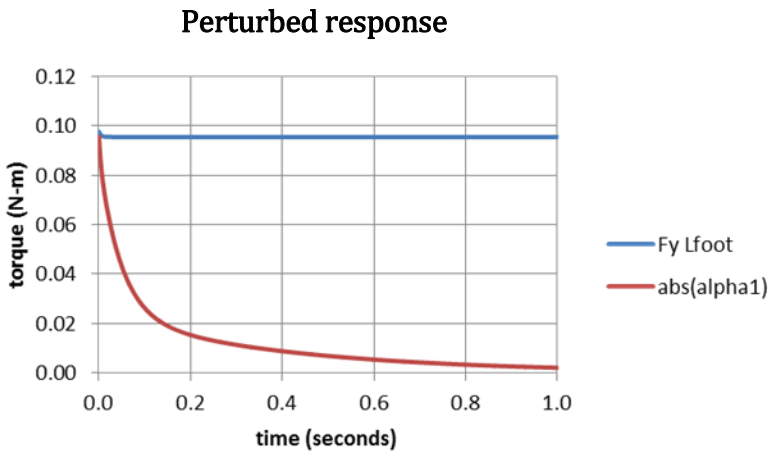



**Figure 54. Problem 9: Foot torque: (2.2) inequality constraint is satisfied, (bals)**

By probing the state space around a stabilized state, a capture subdomain can be estimated. More complete estimate of this subdomain are summarized in Table 1.

**Table 1. Problem 3: Extent of capture subdomain**

| | $\delta_1$, ($\theta_1$ axis) | $\delta_2$, ($\dot{\theta}_1$ axis) | $\delta_3$, ($\theta_2$ axis) | $\delta_4$, ($\dot{\theta}_2$ axis) |
|---|---|---|---|---|
| maximum | 0.055 (radians) | 0.21 (rad/sec) | 0.087 (radians) | 0.28 (rad/sec) |
| minimum | -0.055 (radians) | -0.21 (rad/sec) | -0.087 (radians) | -0.28 (rad/sec) |

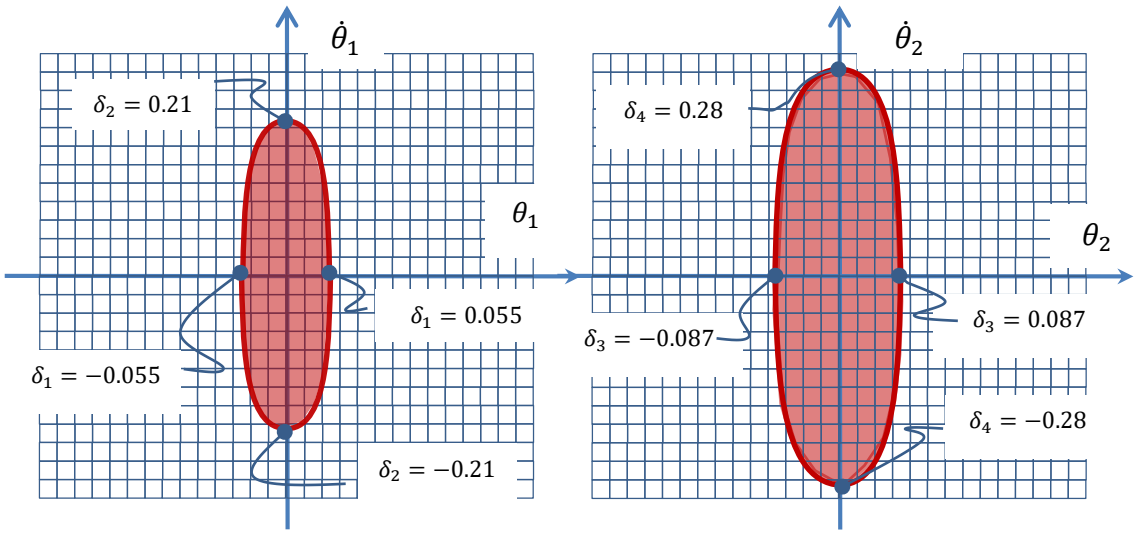



**Figure 55. Problem 9: Extent of capture subdomain**

Future simulations show that as the time constant of the system is lowered, the control response at the foot increases and, for a given foot length, the capture subdomain is reduced in extent.

---

### 3.4 Summary of chapter

This chapter presents the development of OLC solutions that transition two-link dynamics between stabilized states. The underlying assumption is that control inputs have a simple form being defined using only a few parameters. With only a few parameters, the process of finding a solution is vastly simplified from a OLOC method. In the later part of the chapter, state stabilization and uncertainty reduction using LQR was discussed. In this chapter several rhythms were derived.

In the next chapter, the solutions from this chapter will be will be integrated into state transition logic to produce robotic locomotion.

# Chapter 4

## 4.0 Locomotion

In this chapter a set of rhythms required developed for a two-link with mass and geometric properties given in Appendix D will then be integrated into state transition logic to produce locomotion.

### 4.1 Crouching to standing (cr2s), (cl2s)

The rhythm cr2s was determined in problem 2. The cl2s rhythm is simply the negation of the cr2s rhythm. The cr2s rhythm is summarized in Appendix E.

### 4.2 Standing to crouching (s2cr), (s2cl)

The rhythm (s2cr) was determined in problem 5. The s2cl rhythm is simply the negation of the s2cr rhythm. The s2cr rhythm is summarized in Appendix E.

### 4.3 Prone to crouching (p2cr), (p2cl)

The p2cr rhythm was solved for in problem 6. The p2cl rhythm is its negation. A summary of the p2cr rhythm is given in Appendix E.

### 4.4 Crouching to prone (cr2p), (cl2p)

The cr2p rhythm is the controlled fall rhythm that was solved for in Problem 7. The cl2p rhythm is its negation. A summary of the cr2p rhythm is given in Appendix E.

### 4.5 Jumping from a couched position to a couched position (j2cr), (j2cl)

Jumping from a crouched position back into a couching right position was covered in

---

Problem 8. The (j2cl) is it negation. A summary of the (j2cr) and (j2cl) rhythms are given in Appendix E.

### 4.6 Balancing in a stabilized position (bals), (blcr), (blcl)

As shown in Problem 9 balancing in a stabilized position may be achieved using the linear quadratic regulator (LQR) formulation given by equations (3.6c), (3.6d), and (3.6e). The LQR solution is applied locally. That is, a rhythm is used to transition the state of the two-link without closed-loop control. This results in error in the final state. Once in the capture zone, LQR control is applied to pull the two-link closer to the stabilized state.

### 4.7 State Transitions using Rhythms

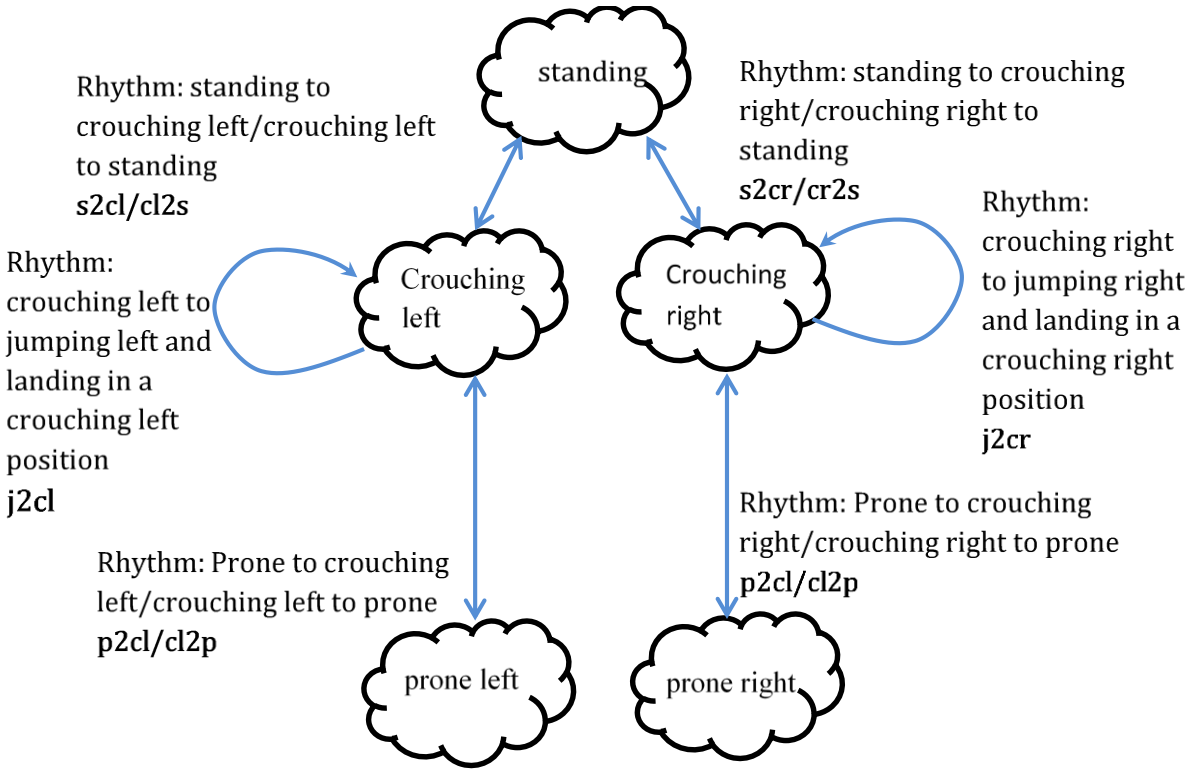



**Figure 56. An illustration of stabilized states and transitions**

The Figure 56 state-transition diagram contains a number of stabilized stated. These are the standing, crouching left, and crouching right states. Transitions between stabilized states occur through the use of rhythms. For example, the sequence of rhythms p2cr, j2cr, cr2s, s2cl, j2cl,cl2p will move the two-link from the prone position to the crouching position; make it jump once to the

right; make it stand and crouch to the left; make it jump back to the left; and then make it lie down to the right.

---

**Problem 10. Locomotion through Rhythms**

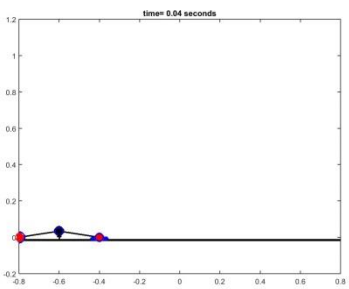
**Given:**

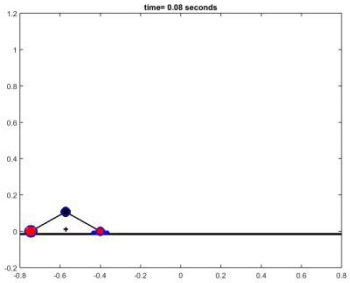
- The rhythms defined in appendix E
- The dynamics of a two-link

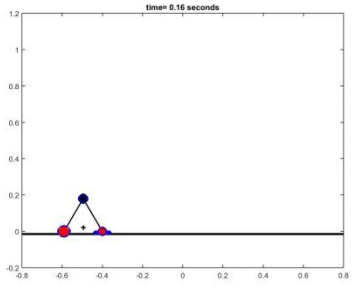
**Find:** A control input that will move the two-link from a prone left position to a standing position; allow it to jump across the page; turn around; jump back and then lie down in a prone position.

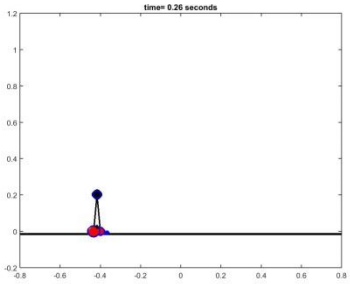
**Solution:** Simulation results of the two-link beginning in a prone left position; standing and then jumping across the page to the right; stopping; standing; and jumping back across the page before lying down in a prone position is given on the following pages. The state transition diagram for this range of motions is given in Figure 58. Rhythms are color coded. Captures are used after each rhythm to minimize uncertainty.

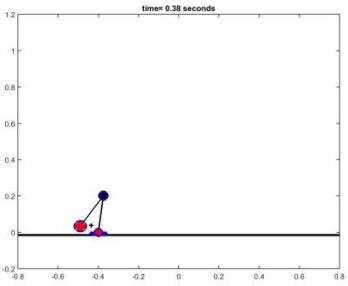
## Rhythm(1): p2cl

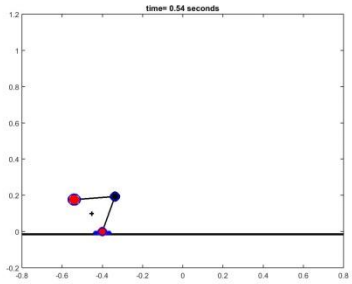


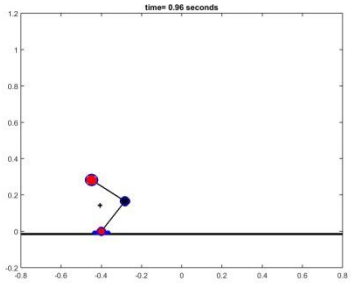



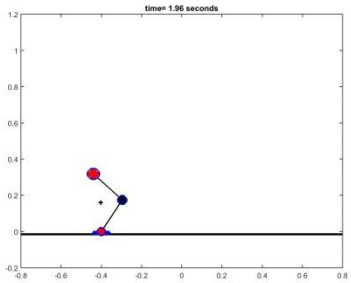



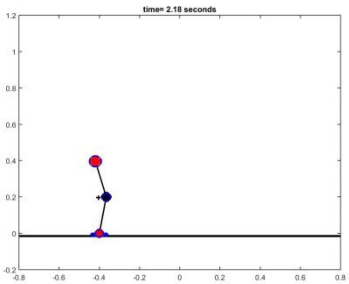



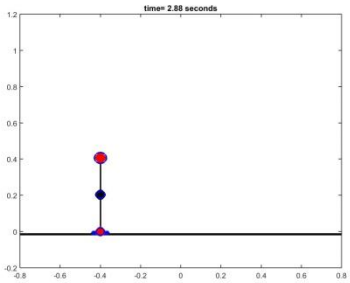



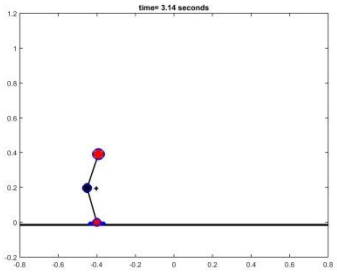

## Rhythm(2): cl2s

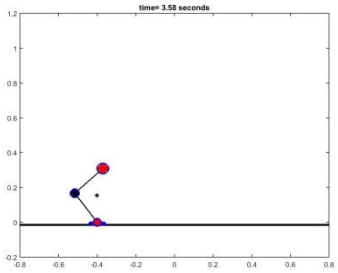


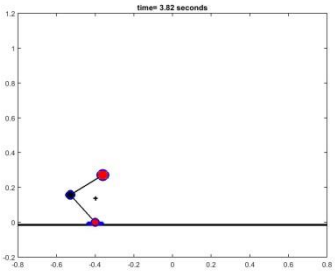



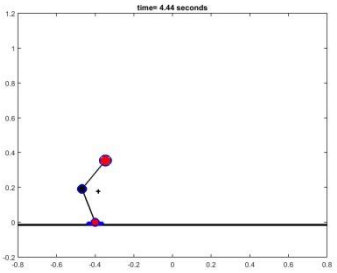
**Rhythm(3): s2cr**

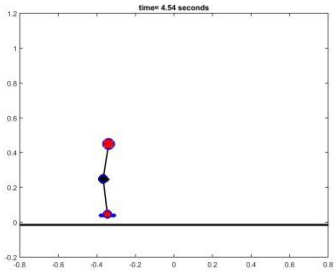


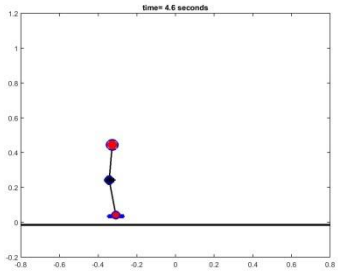



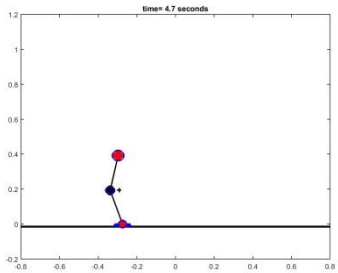



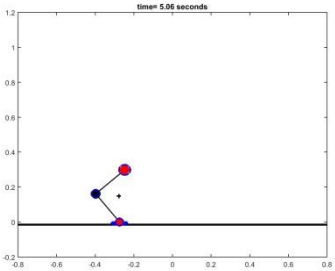

**hythm(4): j2cr**

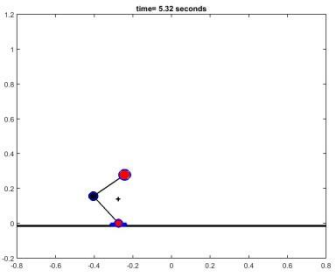



**Rhythm(5): j2cr**

**Rhythm(6): j2cr**

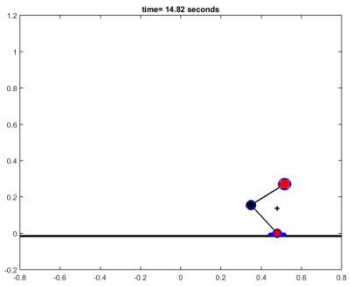
**Rhythm(7): j2cr**

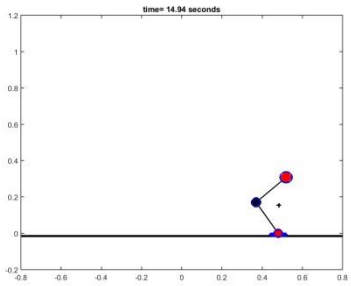
**Rhythm(8): j2cr**

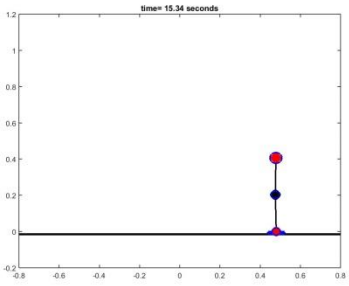
**Rhythm(9): j2cr**

**Rhythm(10): j2cr**

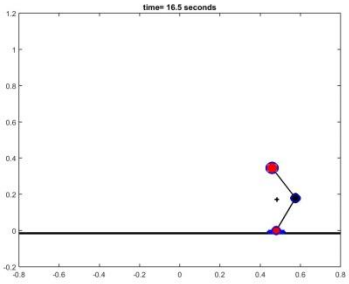
**Rhythm(11): cr2s**

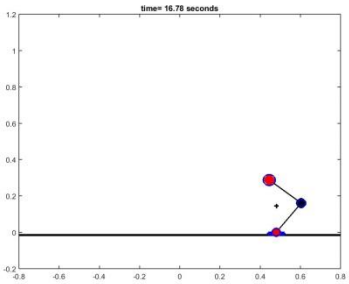


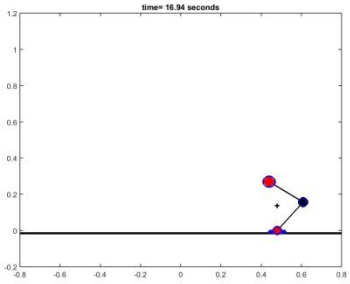




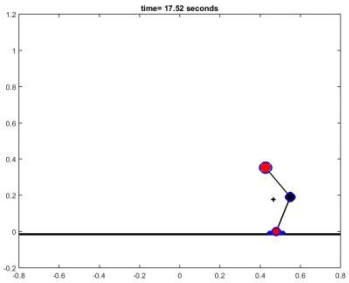

**Rhythm(12): s2cl**

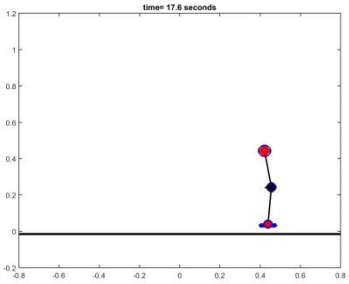


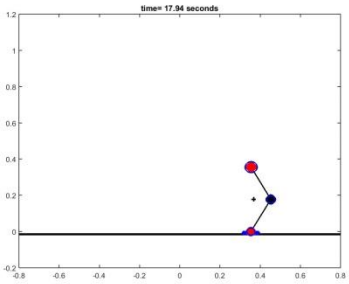


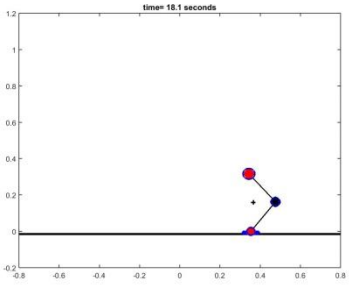


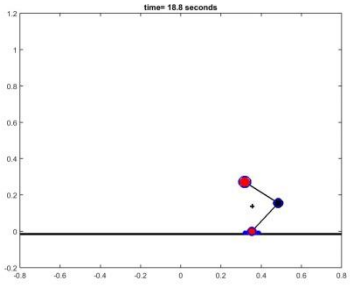

**Rhythm(13): j2cl**












**Rhythm(14): j2cl**

**Rhythm(15): j2cl**

**Rhythm(16): j2cl**

**Rhythm(17): cl2p**

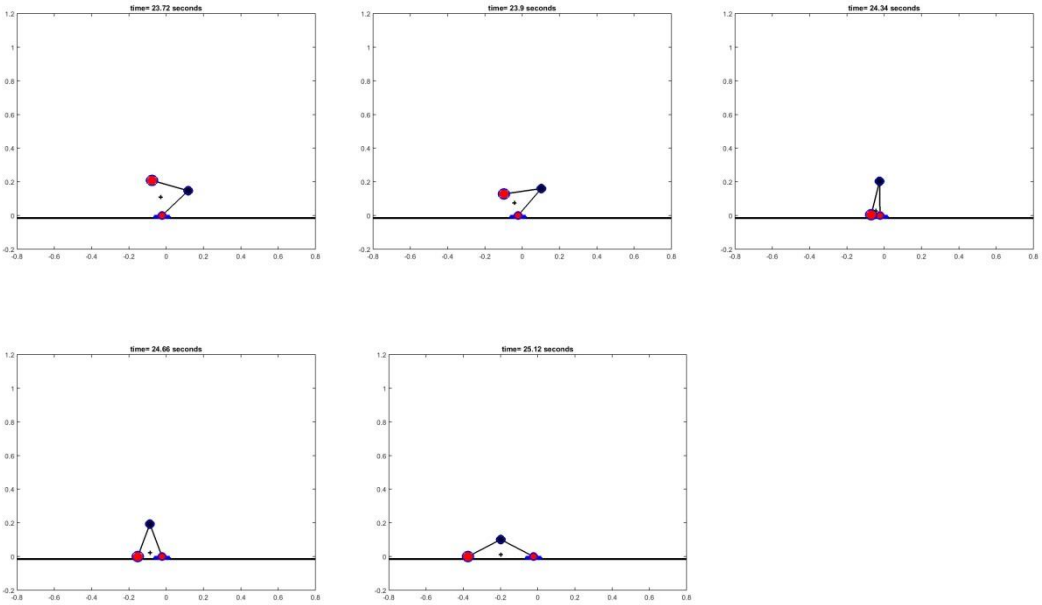



Figure 57. Two-link locomotion

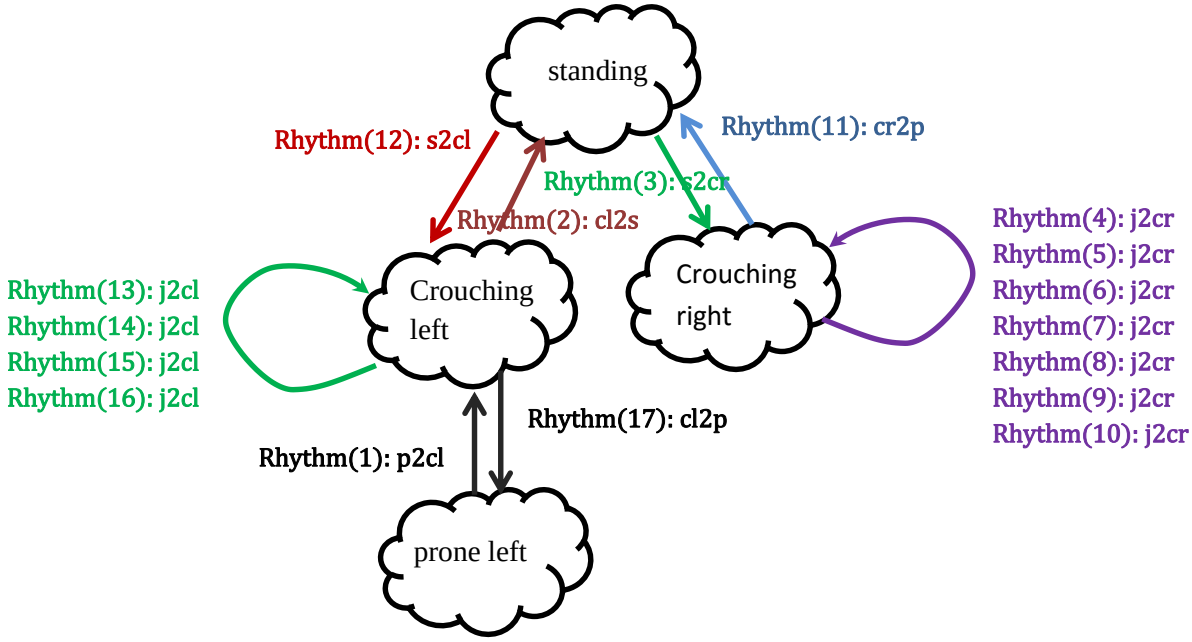



Figure 58. Problem 10. State transitions

## 4.8 Summary of chapter

In this chapter OLC solutions derived in chapter 3 where used to produce rhythms. These rhythms were used to move through state transition logic to produce responsive two-link locomotion. Periodically captures were used to minimize uncertainty.

In the next chapter, experimental analysis will be used to validate of the above control algorithm.

# Chapter 5

## 5.0 Hardware implementation

In the previous chapters the theoretical development of a control algorithm for robotic linkage systems was presented. Although useful for development, important and often unexpected dynamics can be missed through the use of theory alone. To mitigate this risk, experimental validation is necessary. In pursuing validation, a two-link robot was manufactured, code was developed and theory was implemented in hardware. This validation process is discussed in this and the next chapter. A photo of the manufactured two-link discussed in this chapter is shown in Figure 59.

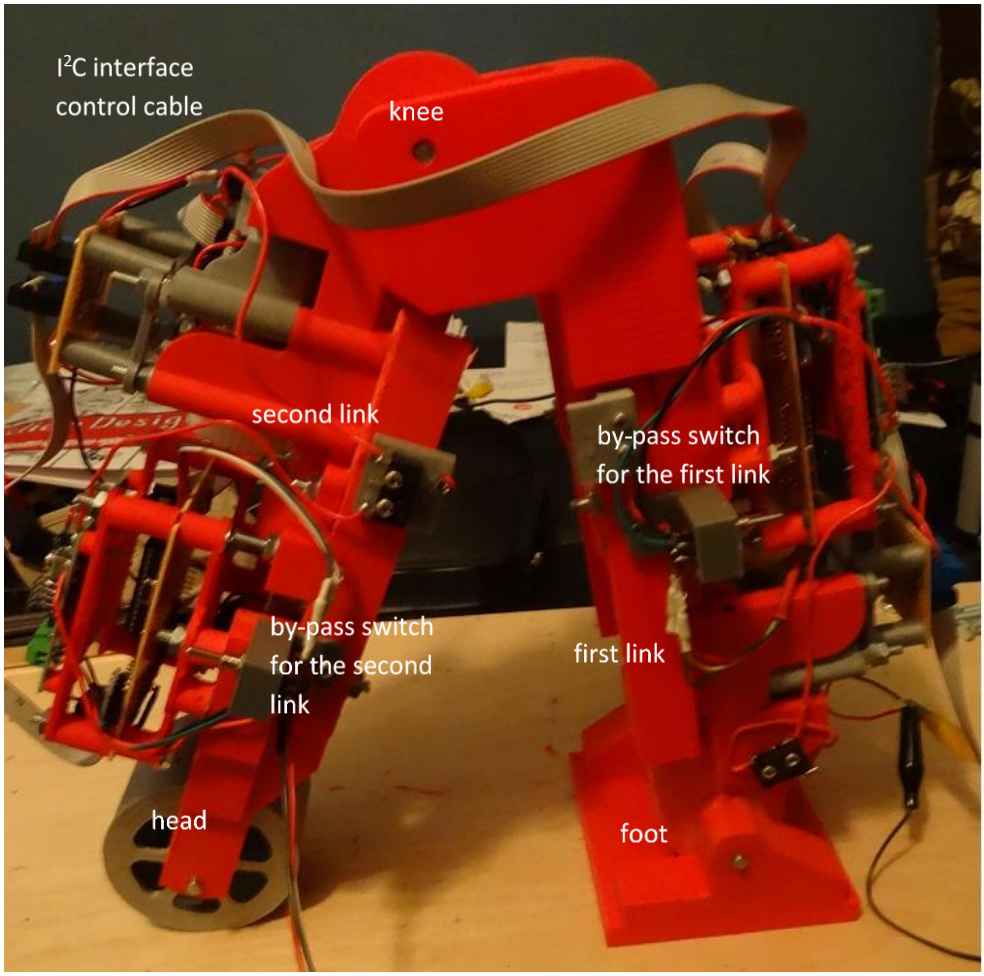



**Figure 59. Realization of the Two-Link**

The two-link consists of a set of additive manufactured (AM) solid parts bolted and pressed together with integrated motors, gears, drive shafts, bearings, circuit boards and weld substructure. These solid and integrated parts can be categorized into two general subsystems – the mechanical subsystem and the electrical subsystem.

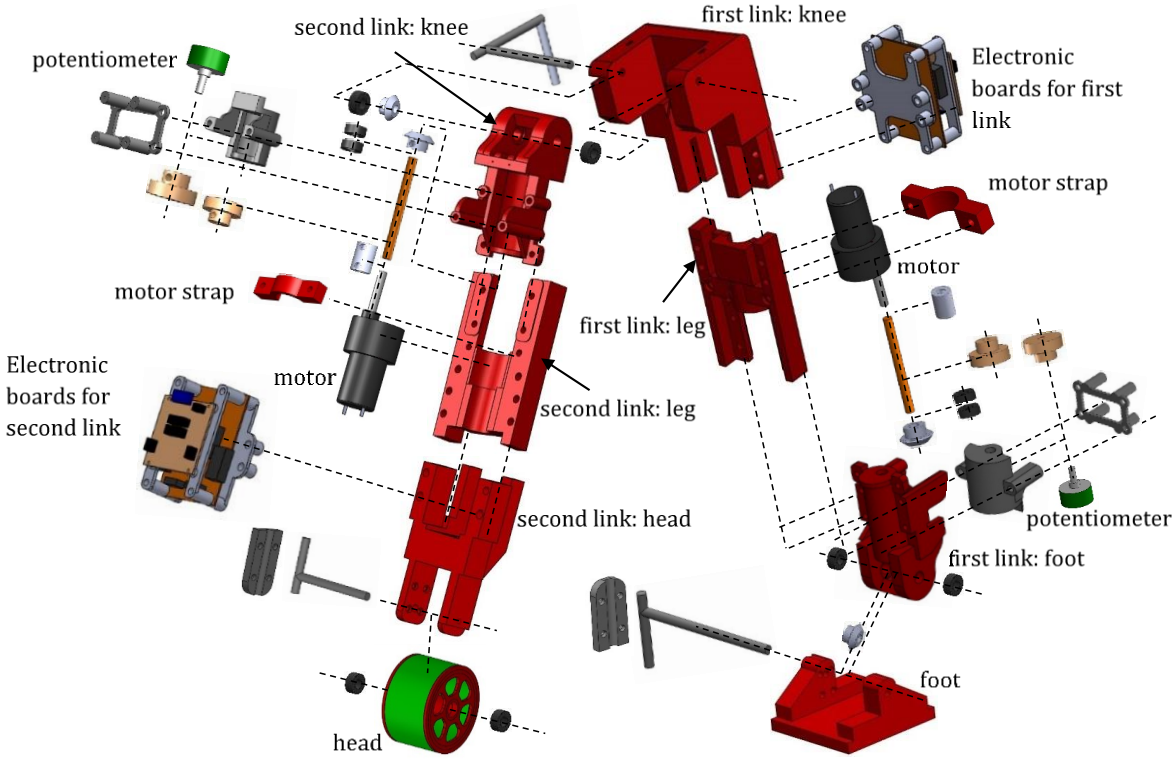



**Figure 60. Mechanical structure comprising the two-link**

## 5.1 Mechanical subsystem

Mechanical subsystem fabrication was performed using AM and tungsten inert gas (TIG) welding. AM substructure allowed for timely successive designs iterations converging to a functional solution. The structure was built, modified and remanufactured over a dozen times. AM substructure was embedded with metal substructure consisting of 6 mm bar stock, gears, motors and bearings. An illustration of this substructure is presented in Figure 60. A listing of parts comprising AM substructure is given in Appendix F.

### 5.1.1 Additive manufactured (AM) substructure

AM parts were produced on a Ultimaker 2+ printer [70] using either ABS or PLA plastic. The process used to produce AM parts was

- to construct solid models in SolidWorks [71];
- to build a virtual system using solid models connected by constraints;
- to perform design checks by exercising assemblies in software;
    - tolerances were check to insure that part mated together property;
    - the kinematics of assemblies were checked to identify unexpected collisions;
- to save these solid modes in stereo lithograph (STL) files;
- to convert STL files to g-code files using the Cura software (supplied by Ultimaker [70]);
- to down load this g-code file into the Ultimaker 2+ for printing and;
- to print and then clean the part.

For PLA material, no preparation of the print table was needed prior to part production. For ABS, preparation of the print table was performed by spreading a solution of ABS in acetone onto the table. This solution was used to ensure that the ABS print adheres to the table. A photo of an ABS part undergoing construction is shown in Figure 61. Present modifications in ABS material have eliminated the need to undergo this preparation for the Ultimaker 2+. Printing at a high honey comb density with supporting structure is recommended.

Upon completion of a print, the part was cleaned. Parts are manufactured with supporting structure which can be removed by fracturing this structure near to a surface and then sanding the remains using a low speed Dremel tool or melting this structure with a hot knife. Parts are then integrated using sliding interfaces, bolts and contact glue.

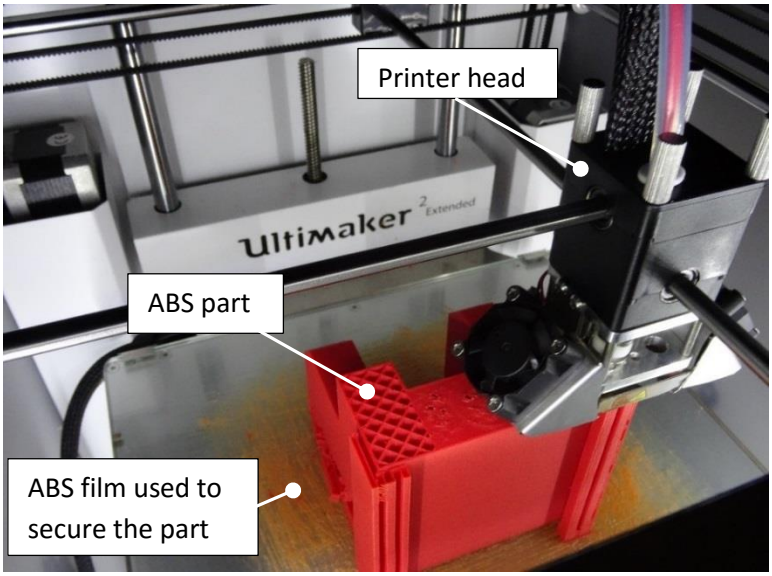



**Figure 61. Three dimensional printing of two-link robotic part**

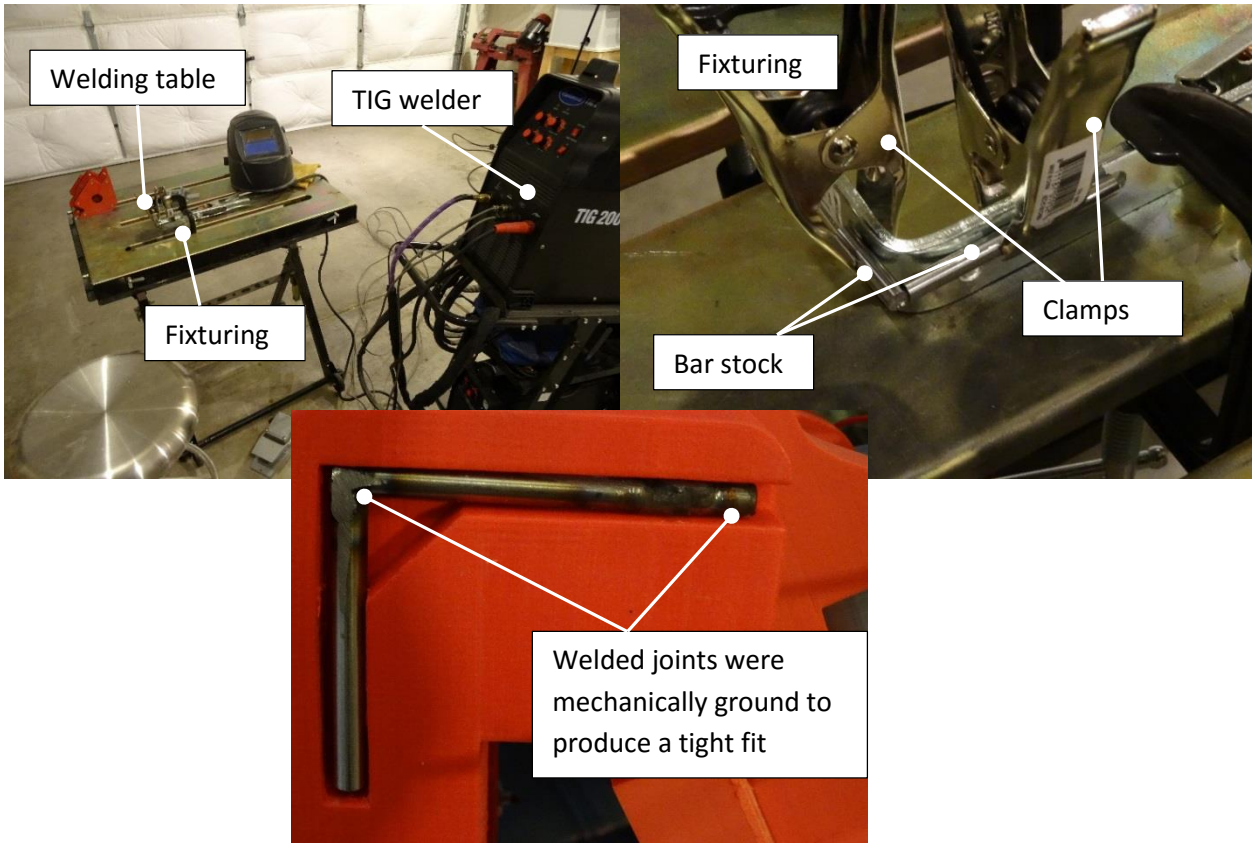



**Figure 62. Welded joints**

### 5.1.2 Metal substructure

Metal substructure consists of hardened steel rods, bolts, nuts, gears, washers, and bearings. Steel rods were used to reduce stress concentrations within AM substructure. These rods exist at each joint and at the head of the two-link. Rods were manufactured using TIG welding to assemble 6mm diameter stock. A photo of this process is given in Figure 62.

Developmental testing of the two-link showed that accuracy was of importance to performance. Gear slop (a dead band phenomena) produced a non-repeatable response making control difficult. To minimize this effect, washers were used to minimize gaps between mating teeth. These washers, shown in Figure 63, were iteratively machined on a Craftsman micro mill until a tight fit was achieved.

Threaded connections were undergoing high torque were problematic. Gears and couplers were held in place using threaded set screws which would loosen during impacts. To minimize this problem, set screws were replaced by those which produced very tight fits and shafts were machined flat under screw heads to avoid slip. Set screws were so tight that after assembly it was no longer possible to remove them. Bearings were used throughout the two-link to minimize wear and to maximize repeatability.

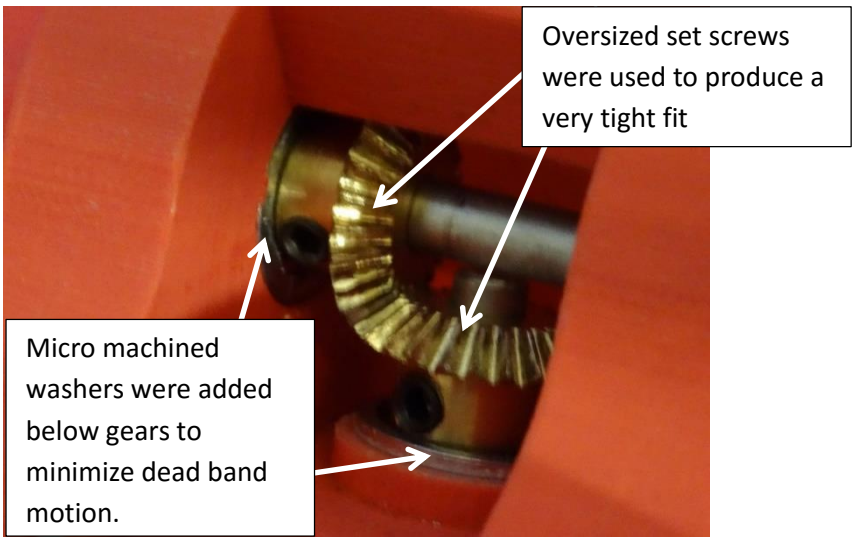



**Figure 63. Structure used to minimize the dead band in gears**

A listing of the parts used to produce integration metal substructure is given in Appendix G and H.

### 5.2 Electrical subsystem

The electrical subsystem is illustrated in Figure 64. There are two subsystems within this subsystem – a low power signal subsystem and a high power driver subsystem. The low power signal subsystem controls the high power driver subsystem which provides power to the motors.

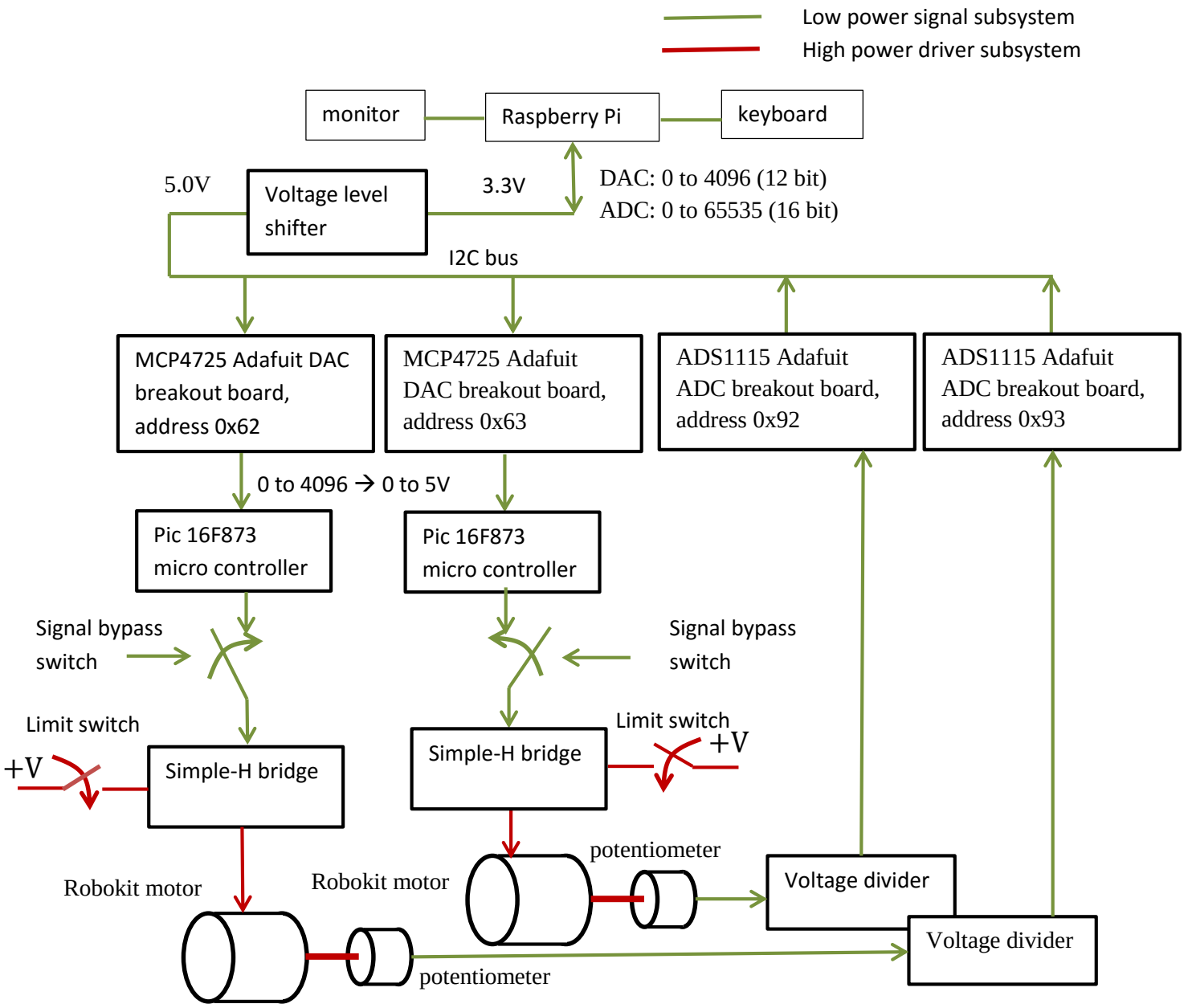



**Figure 64. Electrical subsystem**

### 5.2.1 Low power electrical subsystem

The low power signal subsystem consists of a Raspberry Pi (Raspi) B+ card size microcomputer, a voltage shifter, two Adafruit MCP4725 breakout boards, two Adafruit ADS1115 breakout boards, two Microchip PIC16F873 midrange microprocessors and a set of potentiometers integrated into voltage dividers.

Control signals are sent from the Raspi to the Adafruit MCP4725 using the inter-integrated circuit ($I^2C$) interface. The MCP4725 produces a 0 to 5 V analog output which drives the analog input of the PIC16F873. The PIC16F873 produces two pulse width modulated (PWM) signals – one driving a motor in the forward direction and the other driving the same motor in the opposite direction. Motor driver H-bridges are crossover components from the low power signal subsystem to the high power driver subsystem. PWM signals drive H-bridges producing mirror images of their inputs but at higher voltage and amperage. Motor voltages are 12 V with a maximum current into of 20 A. A Radio shack, 12V, 20 A power supply powers the H-bridges. India Robotics brushed motors and gear boxes drive power trains which drive potentiometers. Potentiometers are integrated into voltage bridges producing voltages that are measured by Adafruit ADS1115 breakout boards. These voltages are proportional to the angular rotation of the foot and knee. The Adafruit ADS1115 breakout boards communicate these voltages back to the Raspi using the $I^2C$ interface. Raspi software (in C) closes the loop by relating measured voltages from the ADS1115 boards to applied voltages sent to the MCP4725 boards.

Control can be overridden by switching PWM inputs from the microcontrollers to an outside PWM source. This allows for the two-link to be moved to a given configuration without using the Raspi as a controller – a convenience needed for testing.

$I^2C$ communications occurs by the Raspi sending the seven-bit address of the board that it wishes to talk to followed by sending or receiving two eight bit words of data. Between each transmission the receiving board sends an acknowledgement – a single bit long. The Raspi must be configured to use

the I²C interface. There are several web based tutorials that explain how to set this configuration [72]. With the Raspi configured, a number of internal subroutines can be used in a C program to send and receive data from the Adafruit break out boards.

For example, to produce a 3.0V output on the MCP4725 with address b1100010 (0X62) the following set of C program lines are required

```
:
int D2A_address0 = 0x62;   //Define the address of the MCP4725
uint8_t D2A_writeBuf0[3];  //declare a 3 word, 8 bit binary array
:
// open the D2A for writing address 0
D2A_I2CFile0 = open("/dev/i2c-1",O_RDWR);
ioctl(D2A_I2CFile0, I2C_SLAVE, D2A_address0);
:
V_out_0 = 3.0;             //Set the output voltage to the desired
value
:
// convert V_out_0 to a Hexadecimal number
a     = (int)2000*v_out_0+2048;
b     = a/16;
h1    = a%16;
a     = b;
b     = a/16;
h2    = a%16;
h3    = b;
:
//compute the high and low decimal number and output
D2A_writeBuf0[0] = h3;
D2A_writeBuf0[1] = 16*h2+h1;
write(D2A_I2CFile0,D2A_writeBuf0,2);
:
// close out the i2c interface
close(D2A_I2CFile0);
```

The Raspi B+ general purpose input/output (GPIO) port is shown in Figure 65. Pin 3 is the data line and pin 5 is the clock line used to implement the I²C interface. Figure 66 shows the response of these pins on an oscilloscope as the Raspi attempts to communication with address b1100010. The clock is in green and the data is in yellow. When the clock line drops from high to low, the first bit is read. Sequential bits are read each time the clock goes high. As shown in this figure, the seven-bit address of the device that the Raspi is trying to communicate with is b1100010 (0x62). The next bit is a zero followed by a zero acknowledgement from the MCP4725; followed by an 8-bit word

followed by an acknowledgement; followed by another 8-bit word; followed by a final acknowledgement. The MCP4725 uses the last four bits of the first word and the full eight bits of the second word to produce a 12-bit word which it converts to an analog signal between 0 to 5 volts. The first four bits of the first word are used to set the configuration of the device. For more detail as to the functioning of these configuration bits see reference 73. The bit rate is 100kHz. Since 27 bits are required to send one 12-bit word, the maximum data rate is 3.7kHz – a limited bandwidth but sufficient for robotic control.

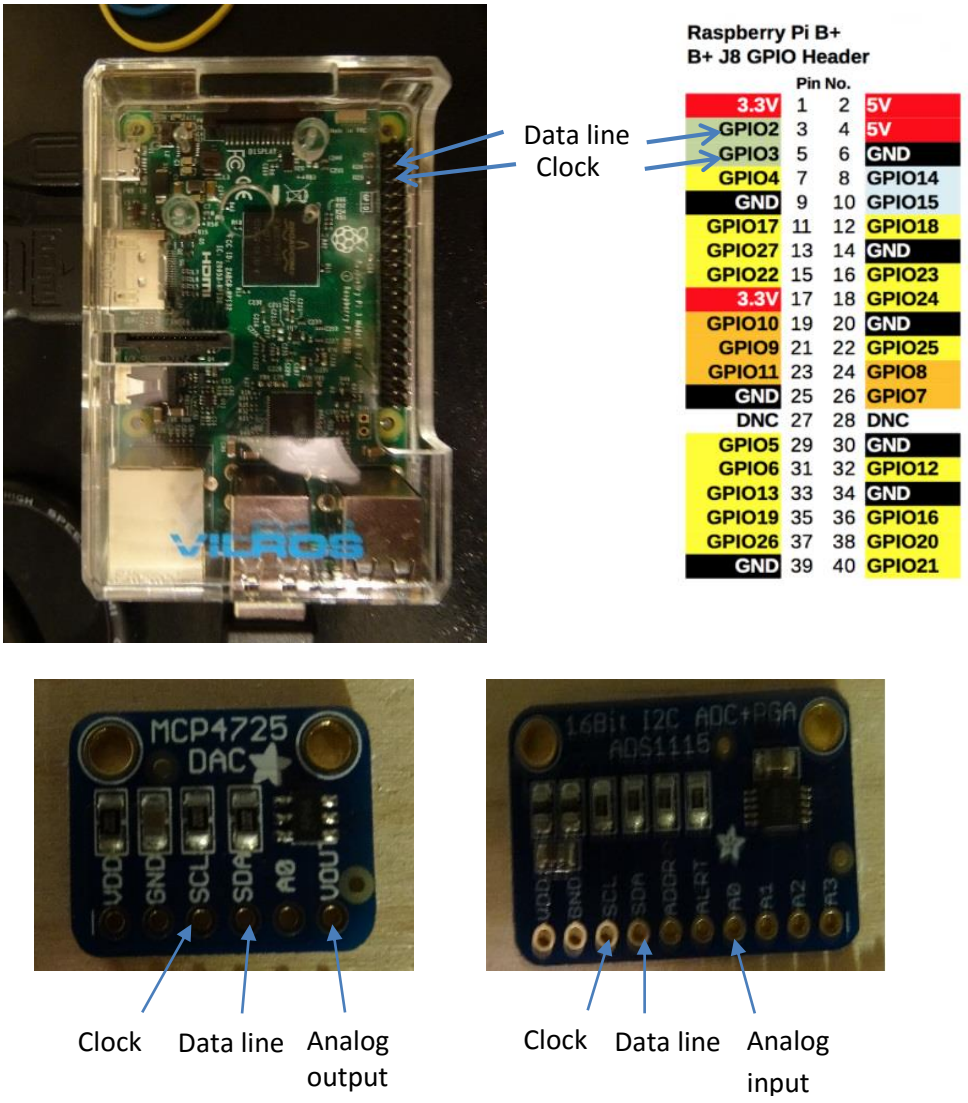



**Figure 65. Raspi general pin input/output (GPIO) configuration and MCP4725 and ADS1115 breakout boards**

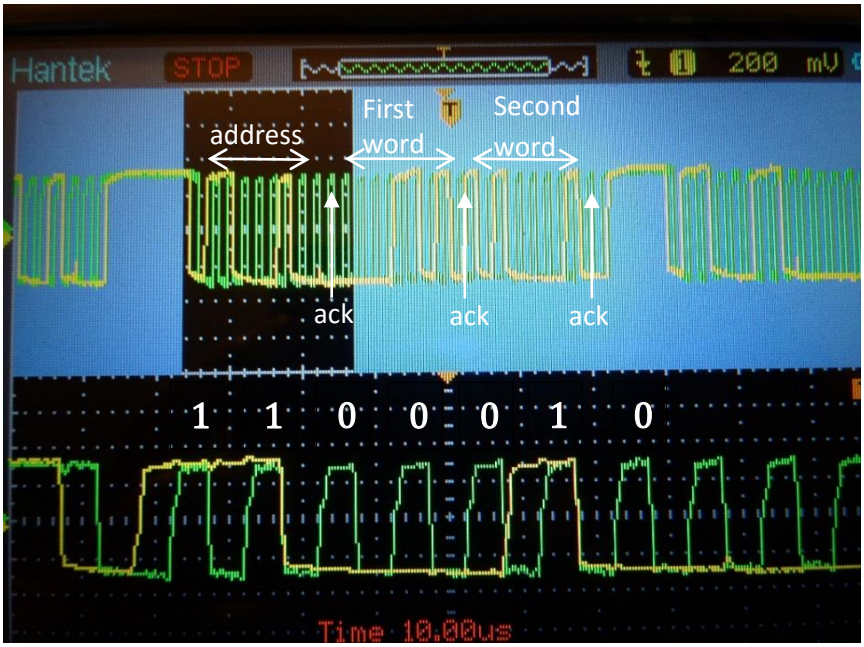



**Figure 66. I2C communication, address b1100010**

Writing through the I²C interface to the ADS1115 can be performed by using the below code:

```
:
int A2D_address0 = 0x48;
:
// declare buffers
uint8_t A2D_writeBuf0[3];
uint8_t A2D_readBuf0[2];
int16_t A2D_val0;
:
// open the A2D for reading address 0
A2D_I2CFile0 = open("/dev/i2c-1",O_RDWR);
ioctl(A2D_I2CFile0, I2C_SLAVE, A2D_address0);
:
// set parameters for address 0 A2D
A2D_writeBuf0[0] = 1;
A2D_writeBuf0[1] = 0b11000001;
A2D_writeBuf0[2] = 0b11100011;
write(A2D_I2CFile0, A2D_writeBuf0, 3);

// is the address 0 A2D ready to read?
A2D_readBuf0[0] = 0;
A2D_readBuf0[1] = 0;
while ((A2D_readBuf0[0] & 0x80) == 0)
{
read(A2D_I2CFile0, A2D_readBuf0, 2);
}
```

```
:
// A2D is ready to read, read data, address 0
A2D_writeBuf0[0] = 0;
write(A2D_I2CFile0, A2D_writeBuf0, 1);
read(A2D_I2CFile0, A2D_readBuf0, 2);

// compute input angles
A2D_val0  = A2D_readBuf0[0]<<8|A2D_readBuf0[1];
v_in_0 = (float)A2D_val0*6.144/32767;
v_in_0 =-(v_in_0-1.63);
v_in_0 = 90.0/2.053*v_in_0;
:
// close out the i2c interface
close(A2D_I2CFile0);
```

For more detail on the ADS1115 see reference 74.

The MCP4725 outputs a 0 to 5V analog signal. This signal is feed into the analog to digital converter of the PIC16F873. The PIC16F873 produces two PWM signals that are used to drive a motor forward and in reverse. Only one set of PWM signals are non-zero at a time. We will refer to the PWM signal that drives a motor forward as PWM_fwd and the PWM signal that drives a motor in reverse as PWM_rev.

An illustration of PWM_fwd and PWM_rev for various voltage inputs is given in Figure 67. PWM signals are “on” (at 5 volts) or “off” at (0 volts). The PWM “on/off” voltage excitation is used to drive the motor hard during a limit period of time overcoming dead band effects due to friction. The duration of the pulse is then used to determine the speed of the motor. As shown in Figure 67, zero occurs for a PIC input voltage of 2.5 volts. For a voltage slightly greater, the pulse width of PWM_fwd jumps to a finite but small value while PWM_rev is zero. For a voltage slightly lower than 2.5 volts, PWM_fwd goes to zero and PWM_rev jumps to the same small pulse width value. This small jump it is used to eliminate the effect of the dead band. Pin outputs are shown in Figure 68. In the next section we will see below how PWM_fwd and PWM_rev are used to drive a motor through an H-bridge.

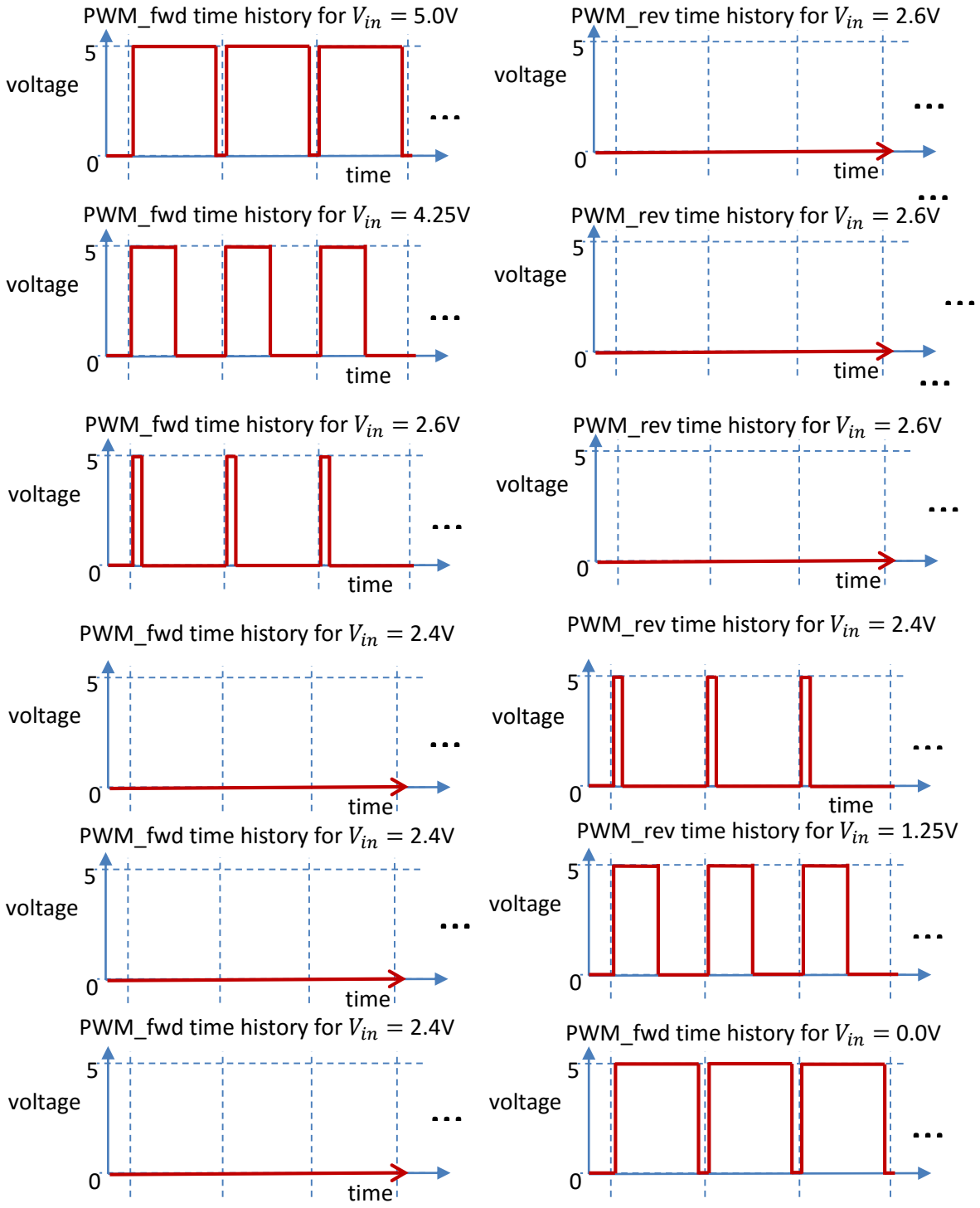



**Figure 67. Pulse width modulated signals as a function of PIC input voltage, $V_{in}$**

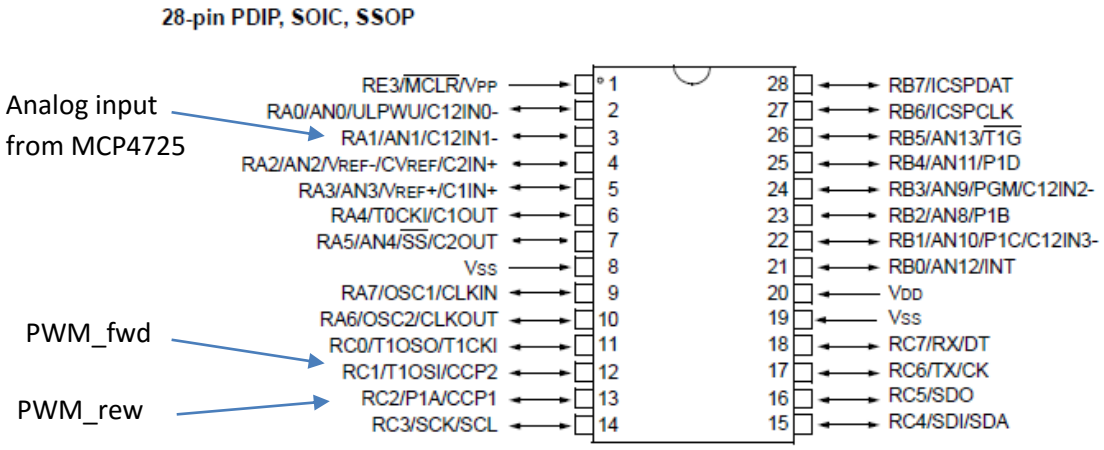



**Figure 68. Pin output/input of the PIC16F873**

### 5.2.2 High power electrical subsystem

Transition from low power 0 to 5V signals to high power 0 to 12V drive voltages occurs through the use of an H-bridge. For the two-link in Figure 59, a "Simple-H" H-bridge was used [75]. The "Simple-H" is a 20A, 12V H-bridge driven by 0 to 5 volt PWM inputs. An H-bridge like the "Simple-H" takes the circuit form shown in Figure 69. The transistors in this figure are high frequency, high amperage MOSFET transistors with internal diodes. The PWM_fwd and PWM_rev signals drive the H-Bridge through a bipolor transistor or an optically coupled transistor. When PWM_fwd is high (i.e. 5V) and PWM_rev is low (ie. 0V), MOSFET power transistors Q1 and Q4 are off and transistors Q2 and Q4 are on. This allows for high amperage current to flow through the motor in the forward direction. When PWM_fwd is low and PWM_rev is high, Q1 and Q4 are on and Q3 and Q3 are off. This allows current to flow in the opposite direction. When both PWM_fwd and PWM_rev are low, the motor is not driven. Internal diodes within each power MOSFET allow any inductive motor current to flow back toward the 12V source. In general, the repetition rate of the PWM_fwd and PWM_rev signals are on the order of 20kHz. This produces acoustic responses from the motors that are ultrasonic - a frequency undetectable by the human ear, but detectable by most animals.

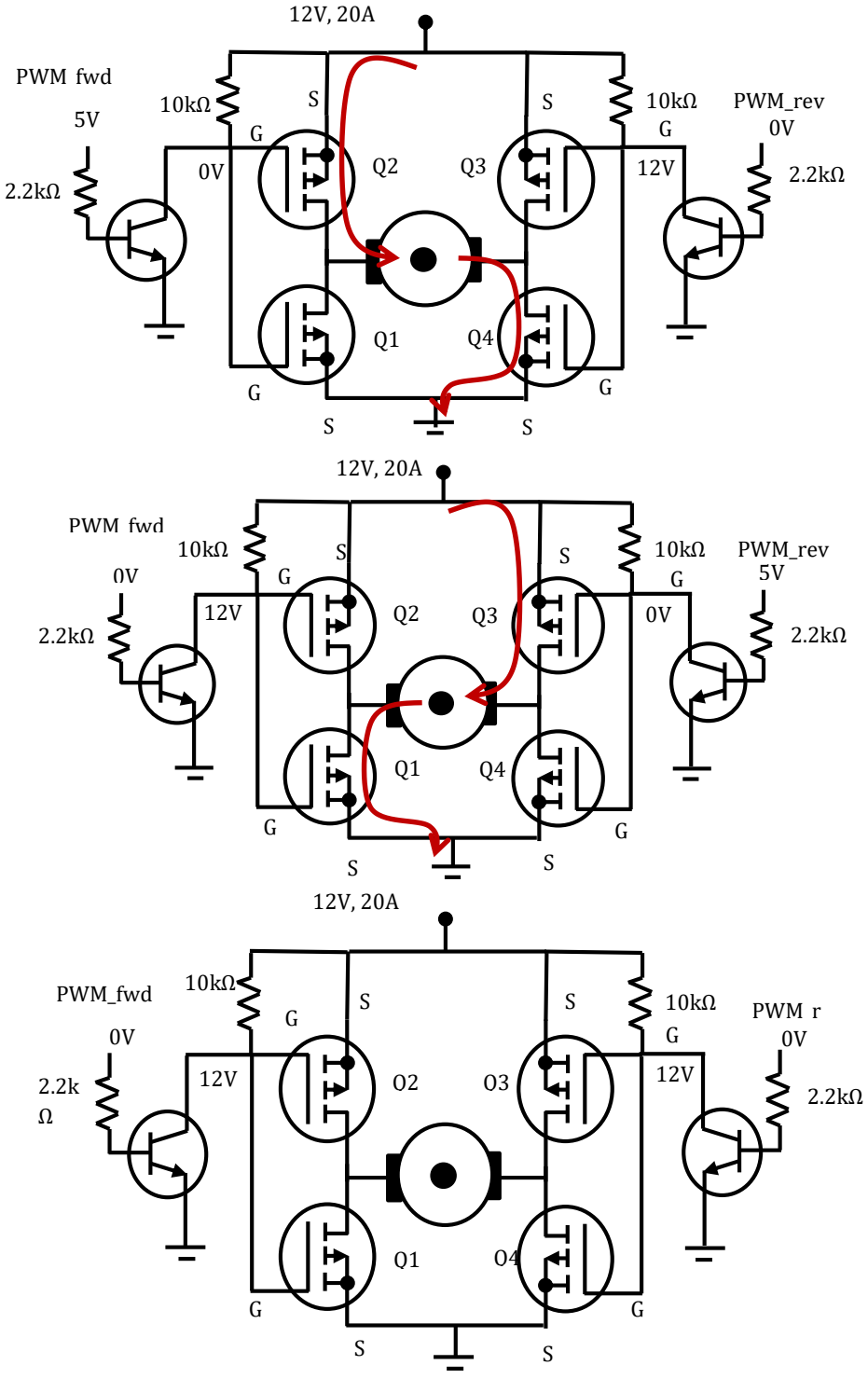



Figure 69. H-bridge

## 5.3 Summary of chapter

In this chapter we discussed the hardware construction of the two-link robot. The two-link has two major subsystems – a mechanical subsystem and an electrical subsystem. The mechanical subsystem is comprised of additive manufactured parts integrated with components. The electrical subsystem is further divided into subsystems - a low power signal subsystem and a high power driver subsystem. Two-link control is coded on a card size Rasberry Pi microcomputer. In this code rhythms are implemented.

In the next chapter, we will show how these rhythms will be used to produce robotic locomotion. Through part of this locomotion the robot is completely controllable. At other times it will be only partially controllable. As the robot moves, its state order changes.

INTENTIONALLY LEFT BLANK

# Chapter 6

## 6.0 Experimental results

In Chapter 3 we assumed that open loop control inputs, called rhythms, can be represented as simple piecewise functions. In this Chapter we discuss the experimental derivation of these rhythms and show how they can be used to produce robotic locomotion.

### 6.1 Experimentally derived virtual constraints

As the linkage system moves, it must do so in a coordinated fashion. For example, to move from a prone configuration to a crouching configuration the foot must stay planted while the legs move. In the theoretical development of this work, these constraints were imposed using control input. In this chapter, they will be imposed using virtual constraints.

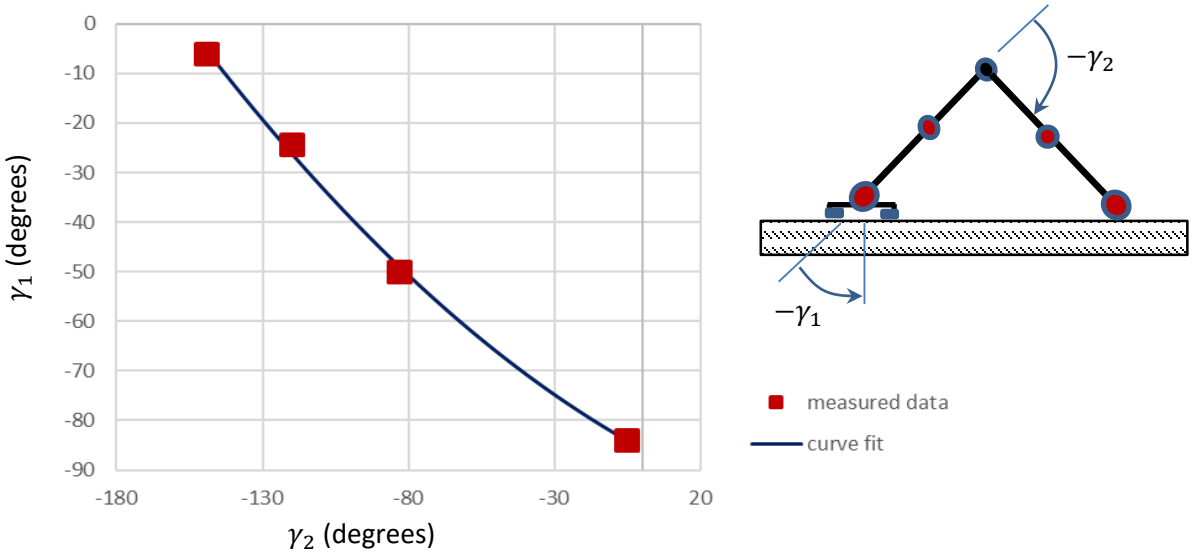



**Figure 70. Virtual constraint for bent dynamics**

Virtual constraints can be experimentally derived by measuring joint locations while the two-link moves. Figure 70 show the experimentally derived virtual constrain for bent dynamics. In this figure the measurement coordinate system is defined in terms of $\gamma_1 = -\theta_1$ and $\gamma_2 = \theta_1 - \theta_2$. This relationship was found by moving the links of the two-link (using the bypass switch in Figure 64)

and measuring the output at a finite set of points. These point were then curve fit with a low order polynomial function. In equation form, the virtual constraint is given by

$$\gamma_1 = 0.0015\,{\gamma_2}^2 - 0.3218\,\gamma_2 - 85.7955\,. \tag{6.1}$$

This constraint is implement in code by measuring $\gamma_2$, calculating $\gamma_1$ and then using a proportional control loop to enforce $\gamma_1$.

A different constraint is required for pinned and free dynamics. Figure 71 contains an illustration of this constraint.

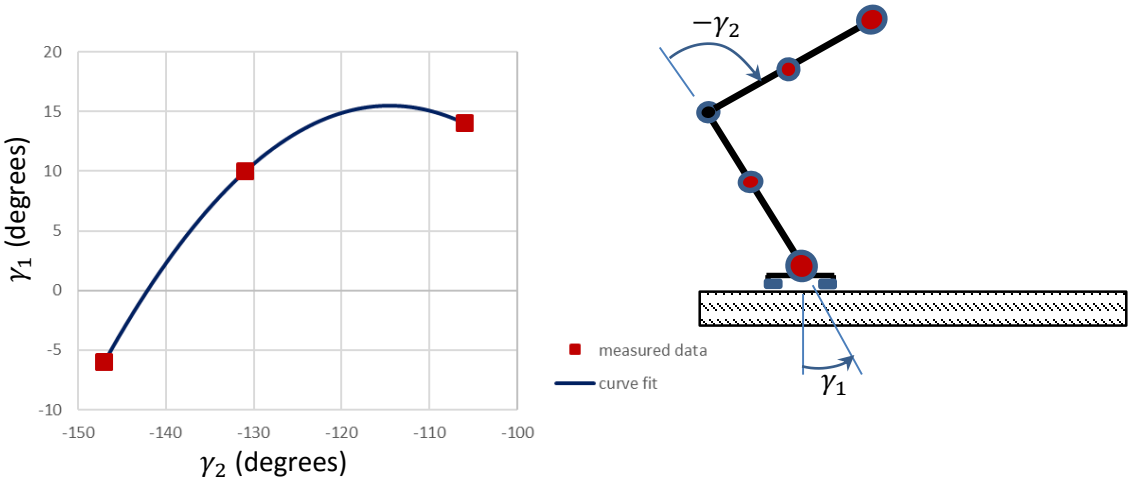



**Figure 71. Virtual constraint for pinned and free dynamics**

Using the same approach as above, this virtual constraint can be stated mathematically as

$$\gamma_1 = -0.0205\,{\gamma_2}^2 - 4.6956\,\gamma_2 - 253.5337\,. \tag{6.2}$$

### 6.2 Experimentally derived rhythms

As stated in section 3.1, we assumed that the form of the control input is a simple set of low order functions. For example, to move the two-link from the prone position to the start of the pinned position, a transitional state, the form of the control rhythm is assumed to be

$$\tau_2 = At + B \text{ for } 0 < t < t_1 \tag{6.3}$$

where $A$ and $B$ are constants. This is used with the virtual constraint, equation 6.1, and the proportional feedback law

$$\tau_1 = K(\widehat{\gamma_1} - \gamma_1) \tag{6.4}$$

where $\widehat{\gamma_1}$ is the measure angle, $\gamma_1$ is the angle given from 6.1 (the virtual constraint) and $K$ is a proportional control gain. The duration of time over which equation 6.3 is imposed, $t_1$, is found by the condition $\widehat{\gamma_2} > \gamma_{2_f}$ where $\gamma_{2_f}$, like $A$, $B$, and $K$ are control parameters. There are so few unknowns in the above algorithm that it is possible to solve for these unknowns by adjusting the parameters experimentally, using the two-link as a simulation tool. A model of the system was not needed. After only a limited number of refinements, a suitable control scheme was found. We call this rhythm "p2t".

After reaching the transitional state between the prone to the pinned configuration, a capture can occur. A proportional controller is sufficient to perform this capture. The capture is maintained up until time $t_3$.

After capture, the control rhythm that moves the two-link into a crouching right state is

$$\tau_2 = C \text{ for } t_3 < t < t_4 \tag{6.5}$$

where $C$ is another control variable. This is used with equation 6.2 and the proportional feedback law.

$$\tau_1 = K(\widehat{\gamma_1} - \gamma_1). \tag{6.6}$$

Time, $t_4$, is found from the condition $\hat{\gamma}_2 > \gamma_{2_f}$ where again $\gamma_{2_f}$ is a set defined value. We call this rhythm "t2cr".

Jumping occurs using a simple on/off rhythm.

$$\tau_2 = \begin{cases} -D \text{ for } t_4 < t < t_5 \\ E \text{ for } t_5 < t < t_6 \end{cases} \tag{6.7}$$

where $D$, $E$, $t_5 - t_4$, and $t_6 - t_5$ are variables. Again, the number of unknown variables is limited allowing for a solution through the experimental adaptation of the parameters. We call this rhythm "j2cr".

Rhythms are implemented in code in Appendix I. The parameters that are used in this code are a function of processor speed. For implementation on a Raspberry pi B+ processor, these parameters are given in Table 2

**Table 2. Definition of parameters used to define rhythms**

| rhythm | p2t | | t2cr | j2cr | |
|---|---|---|---|---|---|
| parameter | A | B | C | D | E |
| value | 0.35 | -0.35 | 0.25 | -0.12 | 0.90 |

## 6.3 Robotic locomotion

The above rhythms were used to move the two-link from a prone position to a couching position, to make it hop across a table and then to force it back into the prone position. The response of the two-link is given in Figure 72.

In this figure the p2t rhythm is used to move the robot to the transition between the prone and pinned configuration. This rhythm is implemented by equations 6.1, 6.3 and 6.4. The robot is then captured using a proportional controller. The robot then moves from this transitional state to a crouching position using the t2cr rhythm. This rhythm is implemented using equations 6.2, 6.5 and 6.6. The robot is captured again. The robot then hops three times with very short captures in between. The rhythm that implements this motion is given in equation 6.7. Final, the robot returns to the transition between the pinned and bent configurations and lies down.

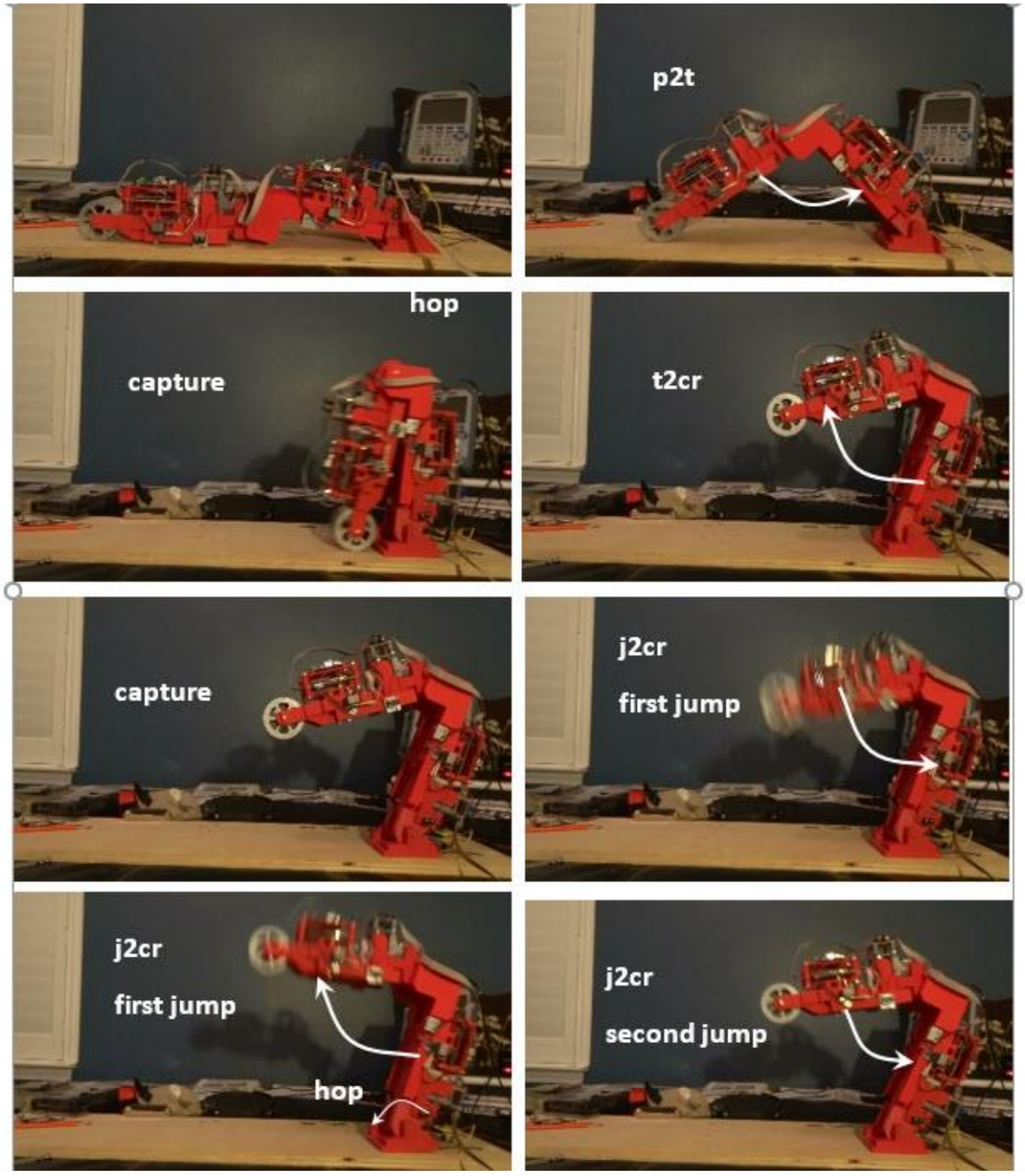



Figure 72. Robotic locomotion through rhythms

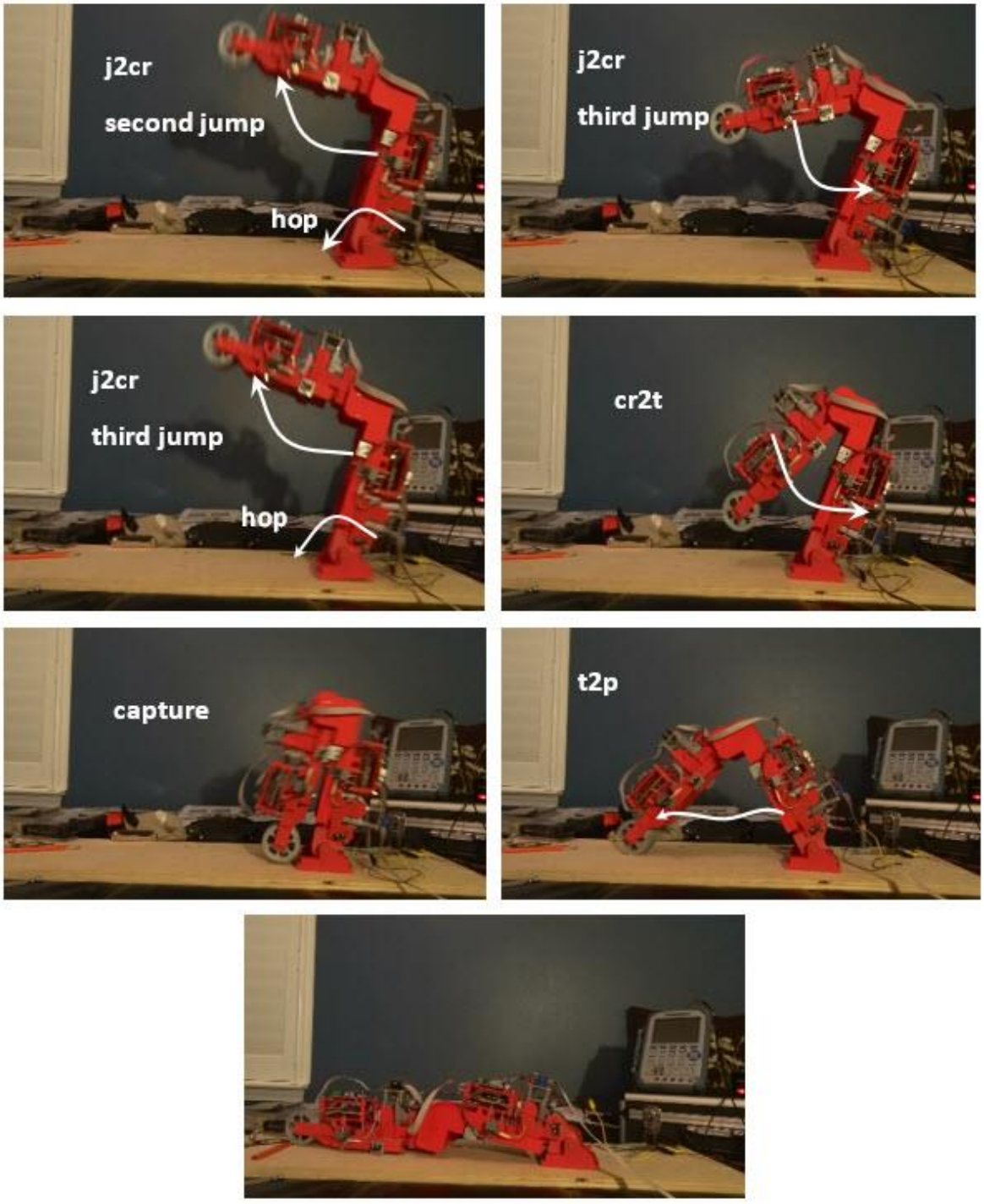



Figure 72. Robotic locomotion through rhythms

## 6.4 Summary of chapter

In this Chapter we have shown how theory developed in Chapters 2 through 4 can be used to control a physical two-link robot. Virtual constraints were used to impose displacement conditions between DOFs such that the two-link maintains its configuration. Each rhythm could be solved for one at a time by using the hardware as a simulation tool.

INTENTIONALLY LEFT BLANK

# Chapter 7

## 7.0 Discussion

The Chapters in this dissertation take us from a review of the literature into dynamic modeling, theoretical developments, and hardware implementation.

In Chapter 1 a literature review was performed. The question that emerged from this review was: Why aren't optimal control solutions used more often for robotic systems? The answer given in this Chapter is that although necessary conditions are easy to derive, solving for these conditions is very difficult. A possible option for avoiding this issue would be to constrain the complexity of the control input and to solve the limited number of variables that define it. This is the underlying thesis that this dissertation is based upon.

In Chapter 2 the dynamics of a simple robotic system was studied. It was found that higher order dynamics could be used to represent lower order dynamics. Transitions between configurations of different order could be obtained using state continuity and impact dynamics. It was assumed that when a linkage system impacts the ground the normal velocity at the point of impact goes to zero.

In the Chapter 3 a control algorithm was present using the Chapter 2 model. This algorithm is based upon the assertion that the input could be represented as a simple function called a rhythm. For a simple problem, it was shown that this assumption produced control inputs that were close to the optimal minimum energy solution. It was then used to solve more complex problems which could not be solved otherwise. In Chapter 4, rhythms for the two-link were derived and used to produce simulated locomotion.

In Chapter 5, the construction of a two-link robot was discussed and in chapter 6 it was implemented with rhythms to produce locomotion.

### 7.1 Conclusions

In this dissertation an algorithm for the control of a linkage system was presented. This algorithm is based on the principle that the control input required to move a robotic linkage system from one

configuration to another can be represented by a simple set of open loop piecewise functions called rhythms. Proving this (for a limited set of dynamics) was the major contribution of this dissertation. Complex calculus of variations or high order parameter optimization solutions were not needed. Simple functions whose parametric values can be easily solved for performed well.

## 7.2 Future work

Proposed future work consists of the following:

1) The low power signal subsystem used Adafruit breakout broads to communicate with the Raspi. These were not needed considering that $I^2C$ communications can occur directly with mid-range Microchip processors. Nevertheless, to implement this requires an enhanced level of machine coding which was beyond the scope of the present work. It is suggested that the microprocessors be reprogrammed to handle both PWM generation as well as $I^2C$ communication. This will allow the microprocessor to take any address between 0 and $2^7$=128 while reducing the foot print of the electronics. This foot print can be further reduced by the integration of the H-bridge onto the circuit board producing an integrated motor controller.
2) The power to weight ratio of a linkage system determines its ability to jump. The Chapter 5 two-link weighted about 2 Kg. The maximum power that the supply could deliver was 12 V x 20 A = 240 W. This gives a maximum power to weight ratio of 120 W/Kg. As shown in Table 2, the j2cr rhythm used a voltage input of 0.90. The maximum that this number can be is 1.0 and therefore, the robot was close to its maximum power to weight ratio. It barely made it off the ground.

   If on board Li-ion batteries were used to power the two-link an increase in weight of about ½ Kg would occur. This would result in a reduction in the maximum power to weight ratio by 20% (i.e. 96 W/Kg) leaving it earth bound. It is suggested that both

the batteries and Raspi be brought on board to form an autonomous unit; however, to do so requires that structural weight be reduced.

3) Using a model or physical hardware, it is suggested that higher levels of learning be developed. With so few parameters, the rhythms in chapter 6 could derive by modifying values on line. With this ease of learning, it is reasonable to believe that learning can be performed completely autonomously.
4) A mathematical model of two-link hardware was not necessary nor was it determined. Nevertheless, such a model would be helpful to aid in further research. The mass properties of the two-link could be found experimentally and other properties (such as friction and voltage to torque transduction) could be found by fitting dynamic data. An experimentally derived solution for two-link rigid body dynamics is included in Appendix K. Further analysis is required to determine other relevant parameters.

The long term goal of this work is to aid in the production of a robotic system that can take the place of humans in high consequence environments. With the present evolution in manufacturing and electronics, it is reasonable to believe that this will occur someday. The real question then is: When? There is a need, but the solution does not exit.

INTENTIONALLY LEFT BLANK

## Appendix A: The Euler-Lagrange (EL) equations

The below discussion follows the work of Crandell, et. al. [47]. A connection of links can be considered to be a set of masses. These internal masses are acted upon by two types of forces – a set of forces applied to the links which transmit load to the mass and gravity. For the $i^{th}$ mass

$$\mathrm{f_i} - \frac{\partial}{\partial \mathrm{t}} \mathrm{p_i} = 0 \tag{A1}$$

where $\mathrm{f_i}$ is the total force on the mass (including both applied loads and gravity) and $\mathrm{p_i}$ is its momentum. Multiplying equation A1 by the variation in the response of the mass, $\delta R$ and summing over all masses, we obtain

$$\sum_{i=1}^{N} \left( \mathrm{f_i} - \frac{\partial}{\partial \mathrm{t}} \mathrm{p_i} \right) \delta R_i = 0. \tag{A2}$$

Equation A2 consists of two terms where the first term on the left is the total virtual work done on the system. This is convenient since it can be equated to the virtual work done on the system written in term of generalized coordinates as

$$\sum_{i=1}^{N} \mathrm{f_i}\, \delta R_i = -\delta V + \sum_{j=1}^{n} \Xi_j \delta \varepsilon_j \, 0 \tag{A3}$$

where $\delta V$ is the variation of the potential energy of the system due to gravity and $\sum_{j=1}^{n} \Xi_j \delta \varepsilon_j$ is the virtual work done on the system by applied loads, $\Xi_j$ is the generalized load of the $j^{th}$ generalized coordinate, and $\delta \varepsilon_j$ is the variation of the $j^{th}$ generalized coordinate. There are $n$ generalized coordinates.

The second term in equation A3 is the virtual momentum of the system which can be expanded using the change rule as

$$\sum_{i=1}^{N} -\frac{d\vec{p}_i}{dt} \cdot \delta \vec{R}_i = \sum_{i=1}^{N} -\frac{d}{dt}\left( \vec{p}_i \cdot \delta \vec{R}_i \right) + \sum_{i=1}^{N} \vec{p}_i \cdot \frac{d}{dt}\left( \delta \vec{R}_i \right) \tag{A4}$$

The goal now is to represent this momentum in terms of generalized coordinates. With this goal in mind, the second term on the right hand side is identified to be the total virtual kinetic co-energy of the system. That is

$$\delta T^* = \sum_{i=1}^{N} \vec{p}_i \cdot \frac{d}{dt}\left( \delta \vec{R}_i \right) = \sum_{i=1}^{N} \vec{p}_i \cdot \left( \delta \vec{V} \right) \tag{A5}$$

where $\delta \vec{V} = \frac{d}{dt}\left( \delta \vec{R}_i \right)$. Although Crandell, et. al. [47] makes the point that (A5) is the virtual kinetic co-energy; however, for Newtonian dynamics, the kinetic co-energy and kinetic energy are the same. That is $T = T^*$ where $T$ is the kinetic energy of the system – a term which can be described in term of the generalized coordinates of the system.

Substituting eq. A3, A4, and A5 into A2 gives

$$\delta T - \delta V + \sum_{j=1}^{n} \Xi_j \delta \varepsilon_j - \sum_{i=1}^{N} -\frac{d}{dt}\left( \vec{p}_i \cdot \delta \vec{R}_i \right) = 0. \tag{A6}$$

The last term on the left hand side of equation A6 is the only term with reference to the internal masses of the system; however limiting ourselves to solutions with well-defined initial and final condition, we can let $\delta \vec{\mathrm{R}}_i = 0$ at the end times to eliminate its effect after integrating over all time to obtain

$$\int_{t_1}^{t_2} \left( \delta L + \sum_{j=1}^{n} \Xi_j \delta \varepsilon_j \right) dt = 0 \tag{A7}$$

where $L = T - V$ (called the Lagrangian). Notice that all of the terms in equation A7 can be defined solely in terms of the generalized coordinates of the system.

The goal now is to mathematically massage the Lagrangian such that it can be represented as a set of terms times the variation of the generalized coordinate, $\delta\varepsilon_j$ allowing for $\delta\varepsilon_j$ to be taken outside of the integral.

Following this approach, the Lagrangian can be expanded as

$$\delta L = \sum_{j=1}^{n}\left(\frac{\partial L}{\partial \dot{\varepsilon}_j}\delta\dot{\varepsilon}_j + \frac{\partial L}{\partial \varepsilon_j}\delta\varepsilon_j\right) \tag{A8}$$

where $\dot{\varepsilon}_j = \frac{d\varepsilon}{dt}$. Substituting equation A8 into equation A7, performing integration by parts on the first term and using the fact that the variation is zero at the time limits gives

$$-\delta\varepsilon_j \int_{t_1}^{t_2} \sum_{j=1}^{n}\left[\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\varepsilon}_j}\right) - \frac{\partial L}{\partial \varepsilon_j} - \Xi_j\right] dt = 0. \tag{A9}$$

In order for this equation to be true and $\delta\varepsilon_j \neq 0$, each member of the sum within the in integral must be zero. This results in the Euler-Lagrange equations

$$.\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\varepsilon}_j}\right) - \frac{\partial L}{\partial \varepsilon_j} = \Xi_j. \tag{A10}$$

## Appendix B: MATLAB coding of dynamics and transitions

The below code has been included so that the interested reader can copy this code directly allowing for the minimization of coding errors.

Free dynamics: equations (2.5a) through (2.5f)

```
function [theta1,dtheta1,theta2,dtheta2,x,dx,y,dy,Nf,flag] = ...
        integration_free(m,M,L,grav,dt,N, ...
        theta1o,dtheta1o,theta2o,dtheta2o,xo,dxo,yo,dyo,tau1,tau2)
%-------------------------------------------------------------------------------------------
% EQUATIONS (2.5a) through (2.5f)
%-------------------------------------------------------------------------------------------
%
% INPUTS:
% [m,M,L,grav]            - geometric and mass parameters
% dt                      - time step
% N                       - desired number of time steps
% [theta1o,dtheta1o,theta2o,dtheta2o,xo,dxo,yo,dyo] – initial state
% [tau1,tau2]             - control inputs
%
% OUTPUTS:
% [theta1,dtheta1,theta2,dtheta2,x,dx,y,dy] – state response
% Nf                      - number of time steps
% flag                    - configuration of final state
%                         1, free dynamics
%                         2, pinned dynamics
%                         3, other
%
%-------------------------------------------------------------------------

%set initial conditions
theta1(1)  = theta1o;
dtheta1(1) = dtheta1o;
theta2(1)  = theta2o;
dtheta2(1) = dtheta2o;
x(1)       = xo;
dx(1)      = dxo;
y(1)       = yo;
dy(1)      = dyo;
t(1)       = 0;
X  = [theta1(1);dtheta1(1);theta2(1);dtheta2(1);x(1);dx(1);y(1);dy(1)];

%integrate
for it=1:N

 A     = [ (5/4*m+M)*L^2   (1/2*m+M)*L^2*cos(theta1(it)-theta2(it)); ...
         (1/2*m+M)*L^2*cos(theta1(it)-theta2(it)) (1/4*m+M)*L^2 ];
 B     = [ (3/2*m+M)*L*cos(theta1(it)) –(3/2*m+M)*L*sin(theta1(it)); ...
         (1/2*m+M)*L*cos(theta2(it)) –(1/2*m+M)*L*sin(theta2(it))];
```

```
C       = B';
D       = (2*m+2*M)*[1 0;0 1];
Mx      = [A,B;C,D];
f1      = -(1/2*m+M)*L^2*sin(theta1(it)-theta2(it))*dtheta2(it)^2 ...
          +(3/2*m+M)*grav*L*sin(theta1(it)) + tau1(it);
f2      =  (1/2*m+M)*L^2*sin(theta1(it)-theta2(it))*dtheta1(it)^2 ...
          +(1/2*m+M)*grav*L*sin(theta2(it)) + tau2(it);
f3      =  (3/2*m+M)*L*dtheta1(it)^2*sin(theta1(it)) ...
          +(1/2*m+M)*L*dtheta2(it)^2*sin(theta2(it));
f4      =  (3/2*m+M)*L*dtheta1(it)^2*cos(theta1(it)) ...
          +(1/2*m+M)*L*dtheta2(it)^2*cos(theta2(it)) ...
          -(2*m+2*M)*grav;
RHS     =  [f1;f2;f3;f4];
LHS     = Mx\RHS;

ddtheta1(it)  =  LHS(1);
ddtheta2(it)  =  LHS(2);
ddx(it)       =  LHS(3);
ddy(it)       =  LHS(4);

G       =  [dtheta1(it);ddtheta1(it);dtheta2(it);ddtheta2(it); ...
           dx(it);ddx(it);dy(it);ddy(it)];
X       =  G*dt +X;

theta1(it+1)  =  X(1);
dtheta1(it+1) =  X(2);
theta2(it+1)  =  X(3);
dtheta2(it+1) =  X(4);
x(it+1)       =  X(5);
dx(it+1)      =  X(6);
y(it+1)       =  X(7);
dy(it+1)      =  X(8);

%break if the foot is at ground, switch to pinned dynamics (user place holder)
flag = 1;

end;

Nf = it;

return;
```

Pinned dynamics: equation (2.7a), (2.8a), (2.8b)

```
function [theta1,dtheta1,theta2,dtheta2,x,dx,y,dy,Fx1,Fy1,Nf,flag] = ...
         integration_pinned(m,M,L,grav,dt,N, ...
         theta1o,dtheta1o,theta2o,dtheta2o,xo,dxo,yo,dyo,tau1,tau2)
%-------------------------------------------------------------------------------------------
% EQUATION (2.7a), (2.8a), (2.8b)
%-------------------------------------------------------------------------------------------
%
% INPUTS:
% [m,M,L,grav]              - geometic and mass parmeters
% dt                        - time step
% N                         - desired number of time steps
% [theta1o,dtheta1o,theta2o,dtheta2o,xo,dxo,yo,dyo] – initial state
% [tau1,tau2]               - control inputs
%
% OUTPUTS:
% [theta1,dtheta1,theta2,dtheta2,x,dx,y,dy]    - state response
% [Fx,Fy]                   - foot reaction forces
% Nf                        - number of time steps
% flag                      - configuration of final state
%                           1, free dynamics
%                           2, pinned dynamics
%                           3, bent dynamics
%-------------------------------------------------------------------------

%set initial conditions
theta1(1)  = theta1o;
dtheta1(1) = dtheta1o;
theta2(1)  = theta2o;
dtheta2(1) = dtheta2o;
x(1)      = xo;
dx(1)     = 0;
y(1)      = 0;
dy(1)     = 0;
t(1)      = 0;
X         = [theta1(1);dtheta1(1);theta2(1);dtheta2(1)];

%integrate
for it=1:N

 %compute the states

 Mx        = [ (5/4*m+M)*L^2   (1/2*m+M)*L^2*cos(theta1(it)-theta2(it)); ...
               (1/2*m+M)*L^2*cos(theta1(it)-theta2(it)) (1/4*m+M)*L^2 ];

 RHS(1,1) =  -(1/2*m+M)*L^2 * sin(theta1(it)-theta2(it)) * dtheta2(it)^2 ...
             +(3/2*m+M)*grav*L*sin(theta1(it)) + tau1(it);
 RHS(2,1) =    (1/2*m+M)*L^2*sin(theta1(it)-theta2(it))*dtheta1(it)^2 ...
             +(1/2*m+M)*grav*L*sin(theta2(it)) + tau2(it);
```

```
 LHS    = Mx\RHS;

 ddtheta1(it)  =  LHS(1);
 ddtheta2(it)  =  LHS(2);

 G         =  [dtheta1(it);ddtheta1(it);dtheta2(it);ddtheta2(it)];
 X         =  G*dt + X;

 theta1(it+1)  =  X(1);
 dtheta1(it+1) =  X(2);
 theta2(it+1)  =  X(3);
 dtheta2(it+1) =  X(4);
 x(it+1)       =  x(it);
 dx(it+1)      =  0;
 y(it+1)       =  0;
 dy(it+1)      =  0;
 t(it+1)       =  t(it) + dt;

 % compute the reaction forces, equations (2.8a), (2.8b)
 fx  =  (3*m/2+M)*L*( -sin(theta1(it))*dtheta1(it)^2+cos(theta1(it))*ddtheta1(it) ) ...
     +( m/2+M)*L*( -sin(theta2(it))*dtheta2(it)^2+cos(theta2(it))*ddtheta2(it) );
 fy  = -(3*m/2+M)*L*( cos(theta1(it))*dtheta1(it)^2+sin(theta1(it))*ddtheta1(it) ) ...
     -( m/2+M)*L*( cos(theta2(it))*dtheta2(it)^2+sin(theta2(it))*ddtheta2(it) ) ;
 fy  =  fy +(2*m+2*M)*grav;
 Fx1(it+1)   =  fx;
 Fy1(it+1)   =  fy;

 %if the reaction force goes negative go to pinned dynamics (user place holder)
 flag = 2;

end;

Nf=it+1;

return;
```

Bent dynamics: equation (2.10f)

```
function [theta1,dtheta1,theta2,dtheta2,x,dx,y,dy,Fy1,Fy2,Nf,flag] = ...
         integration_bent(m,M,L,grav,dt,N, ...
         theta1o,dtheta1o,theta2o,dtheta2o,xo,dxo,yo,dyo,alpha)
%-------------------------------------------------------------------------------------------
% EQUATION (2.10f)
%-------------------------------------------------------------------------------------------
%
% INPUTS:
% [m,M,L,grav]          - geometric and mass parameters
% dt                    - time step
% N                     - desired number of time steps
% [theta1o,dtheta1o,theta2o,...
%   dtheta2o,xo,dxo,yo,dyo]    - initial state
% alpha                 - control inputs
%
% OUTPUTS:
% [theta1,dtheta1,theta2, ...
%   dtheta2,x,dx,y,dy]     - state response
% [Fy1,Fy2]               - foot reaction forces
% Nf                      - number of time steps
% flag                    - configuration of final state
%                           1, free dynamics
%                           2, pinned dynamics
%                           3, bent dynamics
%-------------------------------------------------------------------------

%initial conditions
theta1(1)  = theta1o;
dtheta1(1) = dtheta1o;
theta2(1)  = theta2o;
dtheta2(1) = dtheta2o;
x(1)       = xo;
dx(1)      = dxo;
y(1)       = yo;
dy(1)      = dyo;
t(1)       = 0;
X          = [theta1(1);dtheta1(1);theta2(1);dtheta2(1)];

%integrate
for it=1:N

 %compute the state response
 DEN        =   m+4*(m+2*M)*cos(theta1(it))^2;

 ddtheta1(it) = ( 4*(m+2*M)*cos(theta1(it))*sin(theta1(it))*dtheta1(it)^2 ...
             + 2*m*grav/L*sin(theta1(it)) + alpha(it)/L^2 )/DEN;
 G            =  [dtheta1(it);ddtheta1(it)];
 X            =  G*dt +X;
```

```
 theta1(it+1)  =  X(1);
 dtheta1(it+1) =  X(2);
 theta2(it+1)  =  pi-theta1((it+1);
 dtheta2(it+1) = -dtheta1(it+1);
 x(it+1)       =  x(it);
 dx(it+1)      =  0;
 y(it+1)       =  0;
 dy(it+1)      =  0;

 % compute the reaction forces
 fy2 =  -m*L*ddtheta1(it)/(2*sin(theta1(it))) + (M+m)*grav;
 fy1 =  -m*L*(cos(theta1(it))*dtheta1(it)^2 ...
      +sin(theta1(it))*ddtheta1(it))-fy2 + 2*(M+m)*grav;
 Fy1(it+1)    =      fy1;
 Fy2(it+1)    =      fy2;
 t(it+1)      =  t(it)+dt;

 % determine the configuration (user place holder)
 flag=3

end;

return;
```

Free to pinned transition: equations (2.18a) through (2.18d)

```
function [dtheta1p,dtheta2p,fx1,fy1] = ...
transition_free_2_pinned(m,M,L,theta1,dtheta1,theta2,dtheta2,dx,dy)
%-------------------------------------------------------------------------------
% EQUATIONS (2.18a) through (2.18d)
%-------------------------------------------------------------------------------
%
% INPUTS:
% [m,M,L] – geometric parameters
% [theta1,dtheta1,theta2,dtheta2]-state prior to impact
%
% OUTPUTS:
% [dtheta1p,dtheta2p]-velocities after impact
% [fx1,fy1] – impulsive forces
%
%-------------------------------------------------------------------------------

%impact dynamics
Lcg     = L/2*(m/(m+M));
I1      = m*(Lcg-L/2)^2 + M*(Lcg^2);
I2      = I1;
vx1m   =  (L-Lcg)*cos(theta1)*dtheta1 + dx;
vy1m   = -(L-Lcg)*sin(theta1)*dtheta1 + dy;
vx2m   =      L *cos(theta1)*dtheta1 + (L-Lcg) * cos(theta2)*dtheta2 + dx;
vy2m   =     -L *sin(theta1)*dtheta1 – (L-Lcg) * sin(theta2)*dtheta2 + dy;
z       = [vx1m,vy1m,theta1,vx2m,vy2m,theta2,0,0,0,0]';
const1 = (L-Lcg)*cos(theta1)/I1;
const2 = (L-Lcg)*sin(theta1)/I1;
const3 = (L-Lcg)*cos(theta2)/I2;
const4 = (L-Lcg)*sin(theta2)/I2;
A11 = [ ...
 1  0              0 0 0              0 1/(m+M)       0; ...
 0  1              0 0 0              0      0  1/(m+M); ...
 0  0              1 0 0              0  const1  -const2; ...
 0  0              0 1 0              0 -1/(m+M)      0; ...
 0  0              0 0 1              0      0 -1/(m+M); ...
 0  0              0 0 0              1  const3  -const4; ...
-1  0 –(L-Lcg)*cos(theta1) 1 0  -(L-Lcg)*cos(theta2)      0       0; ...
 0 -1  (L-Lcg)*sin(theta1) 0 1   (L-Lcg)*sin(theta2)      0       0];
A12 = [          -1/(m+M)                  0    ; ...
                      0           -1/(m+M)    ; ...
        Lcg*cos(theta1)/I1    -Lcg*sin(theta1)/I1    ; ...
                      0                 0    ; ...
                      0                 0    ; ...
                      0                 0    ; ...
                      0                 0    ; ...
                      0                 0    ];

A21 = [ 1 0   -Lcg*cos(theta1) 0 0 0 0 0; ...
```

```
      0 1    Lcg*sin(theta1) 0 0 0 0 0];
A22 = zeros(2,2);
A         = [A11,A12;A21,A22];
zf        = A\z;
vx1p      = zf(1);
vy1p      = zf(2);
dtheta1p  = zf(3);
vx2p      = zf(4);
vy2p      = zf(5);
dtheta2p  = zf(6);
fx1       = zf(9);
fy1       = zf(10);

return;
```

Pinned to bent transition: equation (19.a) through (19.d)

```
function [dtheta1p,dtheta2p,fx1,fy1,fy2] = ...
      transition_pinned_2_bent(m,M,L,...
      theta1,dtheta1m,theta2,dtheta2m)
%-------------------------------------------------------------------------------------
% EQUATIONS (19.a) through (19.d)
%-------------------------------------------------------------------------------------
%
% INPUTS:
% [m,M,L] – geometric parameters
% [theta1,dtheta1,theta2,dtheta2]-state prior to impact
%
% OUTPUTS:
% [dtheta1p,dtheta2p]-velocities after impact
% [fx1,fy1,fy2] – impulsive forces
%
%-------------------------------------------------------------------------------------

%impact dynamics
Lcg   = L/2*(m/(m+M));
I1    = m*(Lcg-L/2)^2 + M*(Lcg^2);
I2    = I1;
vx1m  =  (L-Lcg)*cos(theta1)*dtheta1m;
vy1m  = -(L-Lcg)*sin(theta1)*dtheta1m;
vx2m  =      L *cos(theta1)*dtheta1m + (L-Lcg) * cos(theta2)*dtheta2m;
vy2m  =     -L *sin(theta1)*dtheta1m – (L-Lcg) * sin(theta2)*dtheta2m;
z     = [vx1m,vy1m,theta1,vx2m,vy2m,theta2,0,0,0,0,0]';
const1 = (L-Lcg)*cos(theta1)/I1;
const2 = (L-Lcg)*sin(theta1)/I1;
const3 = (L-Lcg)*cos(theta2)/I2;
const4 = (L-Lcg)*sin(theta2)/I2;
A11 = [ ...
 1  0              0 0 0              0 1/(m+M)       0; ...
 0  1              0 0 0              0      0  1/(m+M); ...
 0  0              1 0 0              0  const1  -const2; ...
 0  0              0 1 0              0 -1/(m+M)      0; ...
 0  0              0 0 1              0      0 -1/(m+M); ...
 0  0              0 0 0              1  const3  -const4; ...
-1  0 –(L-Lcg)*cos(theta1) 1 0  -(L-Lcg)*cos(theta2)      0       0; ...
 0 -1  (L-Lcg)*sin(theta1) 0 1   (L-Lcg)*sin(theta2)      0       0];
A12 = ...
 [       -1/(m+M)                0                0  ; ...
              0           -1/(m+M)                0  ; ...
  Lcg*cos(theta1)/I1    -Lcg*sin(theta1)/I1              0  ; ...
              0                0                0  ; ...
              0                0           -1/(m+M)  ; ...
              0                0     Lcg*sin(theta2)/I2 ; ...
              0                0                0  ; ...
              0                0                0 ];
```

```
A21 = [ 1 0 –Lcg*cos(theta1) 0 0            0 0 0; ...
     0 1  Lcg*sin(theta1) 0 0            0 0 0; ...
     0 0            0 0 1 -Lcg*sin(theta2) 0 0];
A22 = zeros(3,3);
A          = [A11,A12;A21,A22];
zf         = inv(A)*z;
vx1p       = zf(1);
vy1p       = zf(2);
dtheta1p   = zf(3);
vx2p       = zf(4);
vy2p       = zf(5);
dtheta2p   = zf(6);
fx1        = zf(9);
fy1        = zf(10);
fy2        = zf(11);

return;
```

State derivatives of the Hamiltonian: equations (2.7b) and (2.7c) and their state derivatives

```
%-------------------------------------------------------------------------------------------------------------------
% EQUATIONS (19.a) through (19.d)
%-------------------------------------------------------------------------------------------------------------------
%
% INPUTS:
% [m,M,L,grav] – geometric parameters and gravity
% [theta1,dtheta1,theta2,dtheta2]-state prior to impact
%
% OUTPUTS:
% [f1,f2] – equations (2.7b) and (2.7c)
% [d1dth1,d1ddth1,d1dth2,d1ddth2] = df1/d(th1), df1/d(dth1/dt), df1/d(th2), df1/d(dth2/dt)
% [d2dth1,d2ddth1,d2dth2,d2ddth2] = df2/d(th1), df2/d(dth1/dt), df2/d(th2), df2/d(dth2/dt)
%-------------------------------------------------------------------------------------------------------------------


function [f1,f2,df1dth1,df1ddth1,df1dth2,df1ddth2,...
      df2dth1,df2ddth1,df2dth2,df2ddth2]= ...
   funcval2(m,M,L,grav,theta1,dtheta1,theta2,dtheta2,tau1,tau2)

 % compute the derivatives of the Hamiltonian with respect to the states
 DEN          =  (m/4+M)*(5/4*m+M)-(m/2+M)^2*cos(theta1-theta2)^2;

 f1           =  (- (m/4+M)*(m/2+M)*sin(theta1-theta2)*dtheta2^2 ...
                  + (m/4+M)*(3/2*m+M)*grav/L*sin(theta1) ...
                  - (m/2+M)^2*sin(theta1-theta2)*cos(theta1-theta2)*dtheta1^2 ...
                  - (m/2+M)^2*grav/L*cos(theta1-theta2)*sin(theta2) ...
                  + tau1/L^2*(m/4+M) ...
                  -tau2/L^2*(m/2+M)*cos(theta1-theta2))/DEN;

 f2           =  (  (m/2+M)^2*sin(theta1-theta2)*cos(theta1-theta2)*dtheta2^2 ...
                  - (m/2+M)*cos(theta1-theta2)*(3/2*m+M)*grav/L*sin(theta1) ...
                  + (5/4*m+M)*(m/2+M)*sin(theta1-theta2)*dtheta1^2 ...
                  + (5/4*m+M)*(m/2+M)*grav/L*sin(theta2) ...
                  - (m/2+M)*cos(theta1-theta2)*tau1/L^2 ...
                  +(5*m/4+M)*tau2/L^2 )/DEN;

 df1dth1        =  (- (m/4+M)*(m/2+M)*cos(theta1-theta2)*dtheta2^2 ...
                  + (m/4+M)*(3/2*m+M)*grav/L*cos(theta1) ...
                  - (m/2+M)^2*cos(theta1-theta2)^2*dtheta1^2 ...
                  + (m/2+M)^2*sin(theta1-theta2)^2*dtheta1^2 ...
                  + (m/2+M)^2*grav/L*sin(theta1-theta2)*sin(theta2) ...
                  +  tau2/L^2*(m/2+M)*sin(theta1-theta2))/DEN ...
                   ...
                 +  f1/DEN ...
                 *(-2*(m/2+M)^2*cos(theta1-theta2)*sin(theta1-theta2));

 df1ddth1       =  - (m/2+M)^2*sin(theta1-theta2)*cos(theta1-theta2)*dtheta1/DEN;
 df1ddth1       =  2*df1ddth1;
```

```
df1dth2      = ( (m/4+M)*(m/2+M)*cos(theta1-theta2)*dtheta2^2 ...
           + (m/2+M)^2*cos(theta1-theta2)^2*dtheta1^2 ...
           - (m/2+M)^2*sin(theta1-theta2)^2*dtheta1^2 ...
           - (m/2+M)^2*grav/L*sin(theta1-theta2)*sin(theta2) ...
           - (m/2+M)^2*grav/L*cos(theta1-theta2)*cos(theta2) ...
           - tau2/L^2*(m/2+M)*sin(theta1-theta2))/DEN ...
             ...
           + f1/DEN ...
           *( 2*(m/2+M)^2*cos(theta1-theta2)*sin(theta1-theta2));

df1ddth2     = - (m/4+M)*(m/2+M)*sin(theta1-theta2)*dtheta2/DEN;
df1ddth2     = 2*df1ddth2;



df2dth1      = ( (m/2+M)^2*cos(theta1-theta2)^2*dtheta2^2 ...
           - (m/2+M)^2*sin(theta1-theta2)^2*dtheta2^2 ...
           + (m/2+M)*sin(theta1-theta2)*(3*m/2+M)*grav/L*sin(theta1) ...
           - (m/2+M)*cos(theta1-theta2)*(3*m/2+M)*grav/L*cos(theta1) ...
           + (5/4*m+M)*(m/2+M)*cos(theta1-theta2)*dtheta1^2 ...
           + (m/2+M)*sin(theta1-theta2)*tau1/L^2)/DEN ...
            ...
           + f2/DEN ...
           *(-2*(m/2+M)^2*cos(theta1-theta2)*sin(theta1-theta2));

df2ddth1     = (5/4*m+M)*(m/2+M)*sin(theta1-theta2)*dtheta1/DEN;
df2ddth1     = 2*df2ddth1;

df2dth2      = (- (m/2+M)^2*cos(theta1-theta2)^2*dtheta2^2 ...
           + (m/2+M)^2*sin(theta1-theta2)^2*dtheta2^2 ...
           - (m/2+M)*sin(theta1-theta2)*(3*m/2+M)*grav/L*sin(theta1) ...
           - (5/4*m+M)*(m/2+M)*cos(theta1-theta2)*dtheta1^2 ...
           + (5/4*m+M)*(m/2+M)*grav/L*cos(theta2) ...
           - (m/2+M)*sin(theta1-theta2)*tau1/L^2)/DEN ...
                                ...
           + f2/DEN ...
           *( 2*(m/2+M)^2*cos(theta1-theta2)*sin(theta1-theta2));

df2ddth2     = (m/2+M)^2*sin(theta1-theta2)*cos(theta1-theta2)*dtheta2/DEN ;
df2ddth2     = 2*df2ddth2;



return;
```

## Appendix C: Open Loop Optimal Control (OLOC) necessary conditions

Necessary conditions for terminal constraint control with changing in state order and impact dynamics are analytically derived.

Consider the situation of minimizing

$$IP = \phi\left(\vec{X}(t_f)\right) + \int_{t_o}^{t_f} L\{\vec{X}(t), \vec{u}(t)\}\, dt \tag{C.1}$$

subject to the dynamics

$$\frac{d}{dt}\vec{X} = f(\vec{X}, \vec{u})\ , t \in [t_o, {t_1}^-), t \in ({t_1}^+, t_2], \tag{C.2}$$

$$\vec{X}({t_1}^+) = \Delta(\vec{X}({t_1}^-)), t \in [{t_1}^-, {t_1}^+] \tag{C.3}$$

with initial condition

$$\vec{X}(t = 0) = \vec{X}_o, \tag{C.4}$$

terminal constraint condition

$$\vec{\psi}\left(\vec{X}(t_f)\right) = 0, \tag{C.5}$$

and inequality constraints on the control

$$|u_i| \le u_{max}. \tag{C.6}$$

$L\{\vec{X}(t), \vec{u}(t)\}$ is an integral *weighting* such as would be needed to minimize total state energy (i.e. the instantaneous state energy integrated over time from 0 to $t_f$),

$\phi\left(\vec{X}(t_f)\right)$ is a *weighting* on the terminal state. This is often called a *soft constraint*. Constraints which are rigidly enforced are called hard constraints.

$\vec{\psi}\left(\vec{X}(t_f)\right)$ is a vector of *hard constraints* between the final states.

Multiplying (C.1e) by the Lagrange multipliers $\vec{v}^T$ and adding to (C.1); solving (C.2) for zero (i.e. $-\dot{\vec{X}} + f(\vec{X}, \vec{u}) = 0$), multiplying by Lagrange multipliers, $\vec{\lambda}^T$, integrating over time and adding the result to (C.1a); and multiplying gives,

$$IP = \varphi\left(\vec{X}_1(t_f)\right) + \vec{v}^T\overrightarrow{\psi}\left(\vec{X}(t_f)\right) + \left\{\int_{t_o}^{t_1^-} L\{\vec{X}(t), \vec{u}(t)\} + \vec{\lambda}^T\left\{-\dot{\vec{X}} + f(\vec{X}, \vec{u})\right\}\right\} dt$$
$$+\left\{\int_{t_1^+}^{t_f} L\{\vec{X}(t), \vec{u}(t)\} + \vec{\lambda}^T\left\{-\dot{\vec{X}} + f(\vec{X}, \vec{u})\right\}\right\} dt\ . \tag{C.8}$$

Equation (C.8) is simply equation (C.1) with the additional Lagrange multipliers $\vec{v}^T$, $\vec{\lambda}^T$, and $c^T$ added for the purpose of insuring that (C.2), (C.5) and (C.6) are satisfied. Taking the variation of $IP$

$$\delta IP = \left[\left\{\frac{\partial\varphi\{\vec{X}(t)\}}{\partial\vec{x}} + \vec{v}^T\frac{\partial\psi\{\vec{X}(t)\}}{\partial\vec{x}}\right\}\delta\vec{X}(t)\right]_{t=t_f} + \int_{t_o}^{t_1^-}\left\{\frac{\partial\mathcal{H}(\vec{X},\ \vec{\lambda}^T,\ \vec{u}\ )}{\partial\vec{x}}\delta\vec{X} - \vec{\lambda}^T\delta\dot{\vec{X}}\right\}dt$$
$$+\int_{t_1^+}^{t_f}\left\{\frac{\partial\mathcal{H}\left(\vec{X},\vec{\lambda}^T,\vec{u}\right)}{\partial\vec{x}}\delta\vec{X} - \vec{\lambda}^T\delta\dot{\vec{X}}\right\}dt + \int_{t_o}^{t_1^-}\left\{\frac{\partial\mathcal{H}\left(\vec{X},\vec{\lambda}^T,\vec{u}\right)}{\partial\vec{\lambda}^T} - \dot{\vec{X}}\right\}\delta\vec{\lambda}^T dt + \int_{t_1^+}^{t_f}\left\{\frac{\partial\mathcal{H}\left(\vec{X},\vec{\lambda}^T,\vec{u}\right)}{\partial\vec{\lambda}^T} - \dot{\vec{X}}\right\}\delta\vec{\lambda}^T dt$$
$$+\int_{t_o}^{t_1^-}\left\{\frac{\partial\mathcal{H}\left(\vec{X},\vec{\lambda}^T,\vec{u}\right)}{\partial\vec{u}}\right\}\delta\vec{u}\, dt + \int_{t_1^+}^{t_f}\left\{\frac{\partial\mathcal{H}\left(\vec{X},\vec{\lambda}^T,\vec{u}\right)}{\partial\vec{u}}\right\}\delta\vec{u}\, dt. \tag{C.9}$$

where $\mathcal{H}\left(\vec{X},\ \vec{\lambda}^T,\ \vec{c}^T,\vec{u}\ \right) = L\{\vec{X}(t),\vec{u}(t)\} + \vec{\lambda}^T f\left(\vec{X},\vec{u}\right)$.

Using integration by parts on $\int_{t_a}^{t_b} \lambda^T \dot{\vec{X}} dt = \vec{\lambda}^T \vec{X}\Big|_{t_a}^{t_b} - \int_{t_a}^{t_b} \dot{\vec{\lambda}}^T \vec{X} dt$ and substituting the result into (C.9) gives

$$\delta IP = \left[\left\{\frac{\partial \phi}{\partial \vec{x}}(t) + \vec{\nu}^T \frac{\partial \vec{\psi}}{\partial \vec{x}}(t) - \vec{\lambda}^T(t)\right\}\delta\vec{X}(t)\right]_{t=t_f} + \left[\vec{\lambda}^T(t)\delta\vec{X}(t)\right]_{t=0}$$

$$+ \int_{t_o}^{t_1{}^-} \left\{\mathcal{H}_{\vec{x}}\left(\vec{X},\vec{\lambda}^T,\vec{u}\right) + \dot{\vec{\lambda}}^T\right\}\delta\vec{X}\, dt + \int_{t_1{}^+}^{t_f} \left\{\mathcal{H}_{\vec{x}}\left(\vec{X},\vec{\lambda}^T,\vec{u}\right) + \dot{\vec{\lambda}}^T\right\}\delta\vec{X}\, dt$$

$$- \left\{\vec{\lambda}^T(t_1^+) - \vec{\lambda}^T(t_1^-)\ \left[\vec{X}(t_1^-)_{\vec{X}(t_1{}^+)}\right]\right\}\delta\vec{X}(t_1{}^+)$$

$$+ \int_{t_o}^{t_1{}^-} \left\{\mathcal{H}_{\vec{\lambda}}\left(\vec{X},\ \vec{\lambda}^T,\vec{u}\ \right) - \dot{\vec{X}}\right\}\delta\vec{\lambda}^T dt + \int_{t_1{}^+}^{t_f} \left\{\mathcal{H}_{\vec{\lambda}}\left(\vec{X},\ \vec{\lambda}^T,\ \vec{u}\ \right) - \dot{\vec{X}}\right\}\delta\vec{\lambda}^T dt$$

$$+ \int_{t_o}^{t_1{}^-} \{\mathcal{H}_{\vec{u}}\left(\vec{X},\ \vec{\lambda}^T,\vec{u}\ \right)\}\delta\vec{u}\, dt + \int_{t_1{}^+}^{t_f} \{\mathcal{H}_{\vec{u}}\left(\vec{X},\ \vec{\lambda}^T,\ \vec{u}\ \right)\}\delta\vec{u}\, dt \qquad \text{(C.10)}$$

where $\mathcal{H}_{\vec{a}}\left(\vec{X},\ \vec{\lambda}^T,\vec{u}\ \right) = \frac{\partial \mathcal{H}\left(\vec{X},\vec{\lambda}^T,\vec{c}^T,\vec{u}\right)}{\partial \vec{a}}$.

Minimizing (C.10) results in the below equations:

From the sixth and seventh term of (C.10)

$$\dot{\vec{X}} = f(\vec{X},\vec{u}) \text{ for } t = [t_o, t_1{}^-) \text{ and } t = (t_1{}^+, t_f]. \qquad \text{(C.11)}$$

From the third and fourth term of (C.10)

$$\dot{\vec{\lambda}}^T = -\mathcal{H}_{\vec{x}}\left(\vec{X},\vec{\lambda}^T,\vec{u}\right) \text{ for } t = [t_o, t_1{}^-) \text{ and } t = (t_1{}^+, t_f]. \qquad \text{(C.12)}$$

From the second term of (C.10) and $\delta\vec{X}(t_o) = 0$, for $\vec{X}(t_o)$ specified

$$\vec{X}(t_0) = \vec{X}_o. \qquad \text{(C.13)}$$

From the first term of (C.10), for $\delta\vec{X}\left(t_f\right) \neq 0,\ \vec{X}\left(t_f\right)$ unspecified

$$\vec{\lambda}^T\left(t_f\right) = \frac{\partial \phi}{\partial \vec{x}}\left(t_f\right) + \vec{\nu}^T \frac{\partial \vec{\psi}}{\partial \vec{x}}\left(t_f\right). \qquad \text{(C.14)}$$

From the fifth term of (C.10),

$$\vec{\lambda}^T(t_1^+) = \vec{\lambda}^T(t_1^-)\ \left[\vec{X}(t_1^-)_{\vec{X}(t_1{}^+)}\right]\ t \in [t_1{}^-, t_1{}^+] \qquad \text{(C.15)}$$

and from the (C.3) impact condition

$$\vec{X}(t_1^+) = \Delta(\vec{X}(t_1{}^+)), t \in [t_1{}^-, t_1{}^+]$$

For the situation where the control is not hard bounded, $\vec{u}$ is chosen such that $\frac{\partial \mathcal{H}\left(\vec{X},\vec{\lambda}^T,\vec{c}^T,\vec{u}\right)}{\partial \vec{u}} = 0$ ; however, in general, the optimal control $\vec{u}^*$ can be hard bounded (as given in (C.7)) and

$$\vec{u}^* = \min_{\vec{u}} \mathcal{H}\left(\vec{X},\vec{\lambda}^T,\vec{u}\right). \qquad \text{(C.16)}$$

Equations (C.11) through (C.16) are the necessary conditions for the minimization/maximization of (C.1).

## Appendix D: Two-link properties used in the example problems

The example problems used throughout this thesis have properties that are given in Table 3.

**Table 3. Two-link properties**

| property | Value | Description |
|---|---|---|
| $m$ | $0.06\ (kilogram)$ | Leg mass |
| $M$ | $0.18\ (kilogram)$ | Head and foot mass |
| $L$ | $8 * 0.0254\ (meter)$ | Length of leg |
| $\theta^*$ | $40^o, -40^o, 0^o$ | Crouching angle |

From Figure 8 summing the moments about the foot gives

$$mg\frac{L}{2}sin(\theta_1) + mg\left\{L\sin(\theta_1) + \frac{L}{2}sin(\theta_2)\right\} + Mg\{L\sin(\theta_1) + Lsin(\theta_2)\} = 0$$

Solving for $\theta^* = -\theta_1$gives the (2.21b) state condition

$$\begin{Bmatrix}\theta_1\\ \dot{\theta}_1\\ \theta_2\\ \dot{\theta}_2\end{Bmatrix} = \begin{Bmatrix}\theta^*\\ 0\\ sin^{-1}\left\{\frac{\frac{3}{2}m+M}{\frac{1}{2}m+M}\sin(\theta^*)\right\}\\ 0\end{Bmatrix}.$$

The steady state excitation which will produce this state is determined from (2.7a). Setting all time terms to zero and using (2.1) gives $\alpha_2{}^* = \{3m + 2M\}\sin(\theta^*)$ for $\alpha_1 = 0$.

Motor properties were taken from the RKI-1188 data sheet produced by Robokits India [76] (see spec sheet below).

**Table 4. Motor properties**

| property | Value from Spec. Sheet | Value in MKS units | Description |
|---|---|---|---|
| $\Omega_{max}$ | 18000 (rpm) | 1885 (rad/sec) | No-load speed |
| $v_{nom}$ | 12 (V) | 12 (V) | Nominal voltage |
| $i_{max}$ | 7.8 (A) | 7.8 (A) | No-load current |
| $\tau_{max}$ | 1010 (g-cm) | 0.103 (N-m) | Stall torque |
| G | | | |

From Table 4, motor parameters can be calculated.

$$K_v = \frac{\Omega_{max}}{v_{nom}} = 157.1\ \left(\frac{rad/sec}{V}\right)$$

$$R = \frac{v_{nom}}{i_{max}} = 1.59\ (\Omega)$$

$$K_Q = \frac{\tau_{max}}{i_{max}} = 0.0132\ \left(\frac{N\cdot m}{A}\right)$$

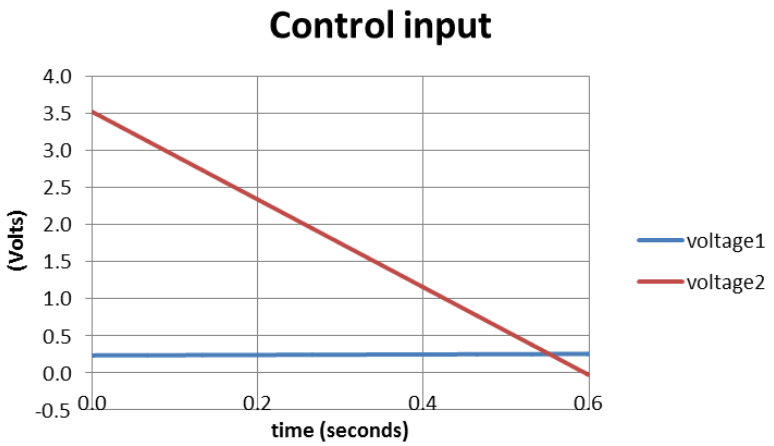
$$\frac{K_Q}{R} = 0.0083\ \left(\frac{N\cdot m}{V}\right)$$

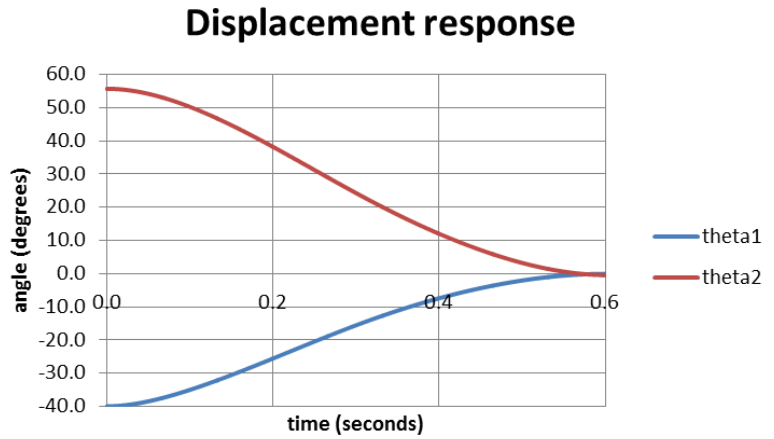
$$\frac{K_Q}{RK_v} = 5.28e-5\ \left(\frac{N\cdot m}{rad/sec}\right).$$

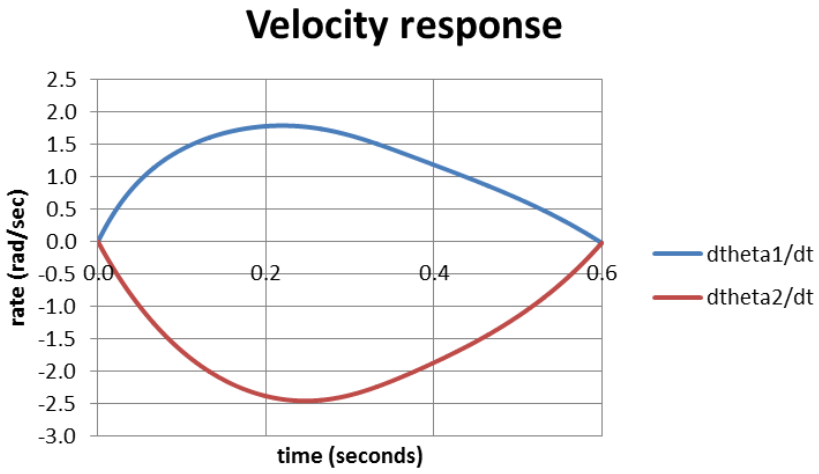
INTENTIONALLY LEFT BLANK

# Appendix E: Two-link rhythms: Theoretical development

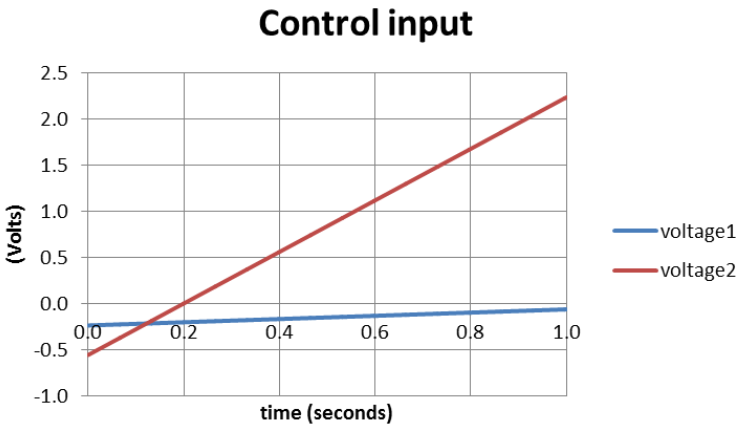
**Rhythm: crouching right to standing (cr2s)**

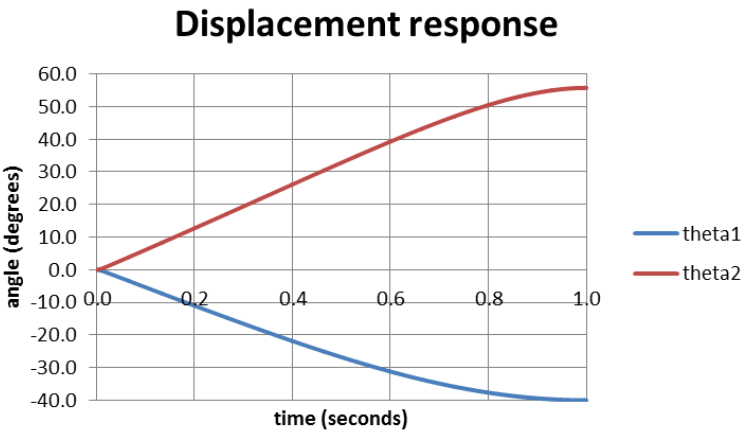


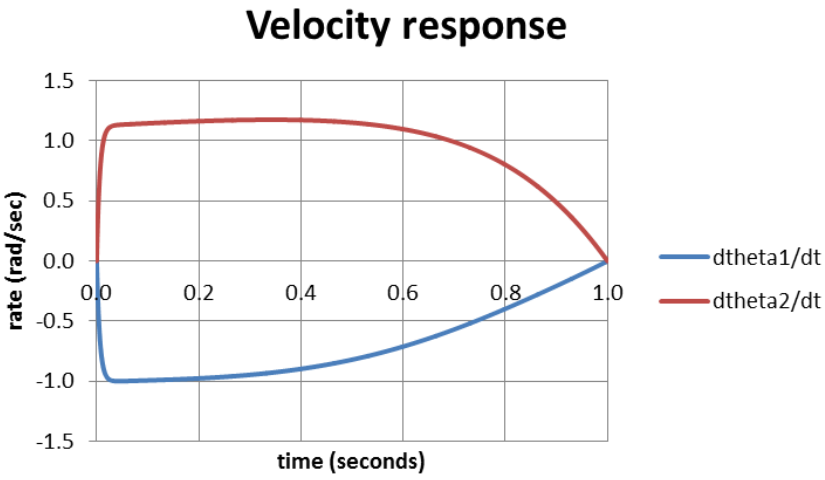



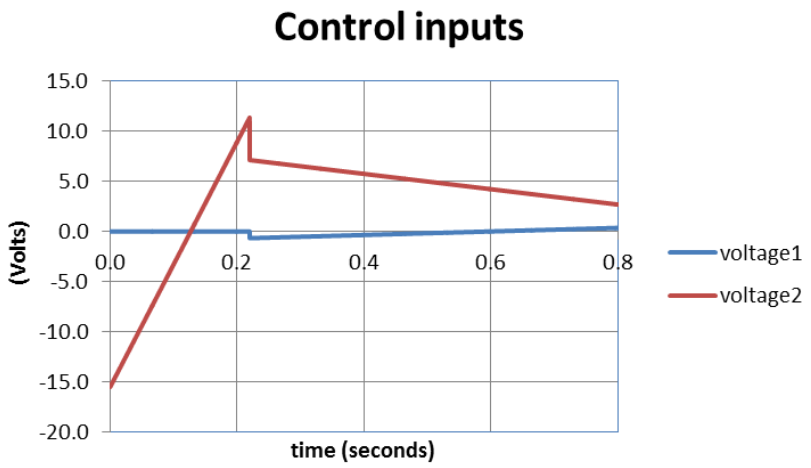
**Rhythm: crouching standing to crouching right (s2cr)**

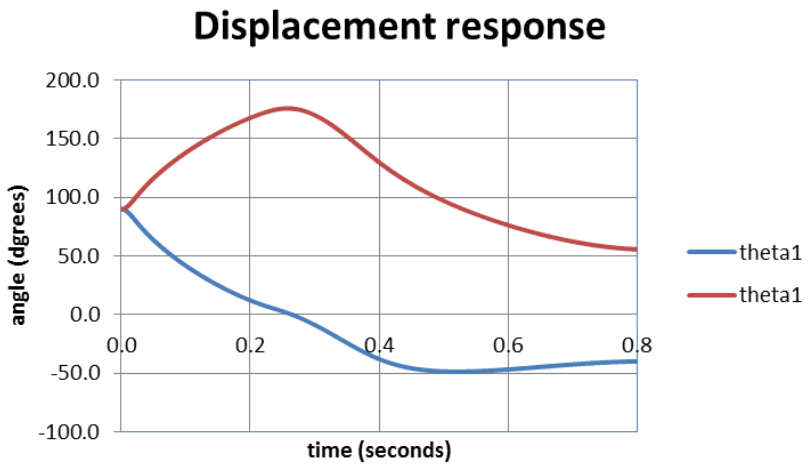


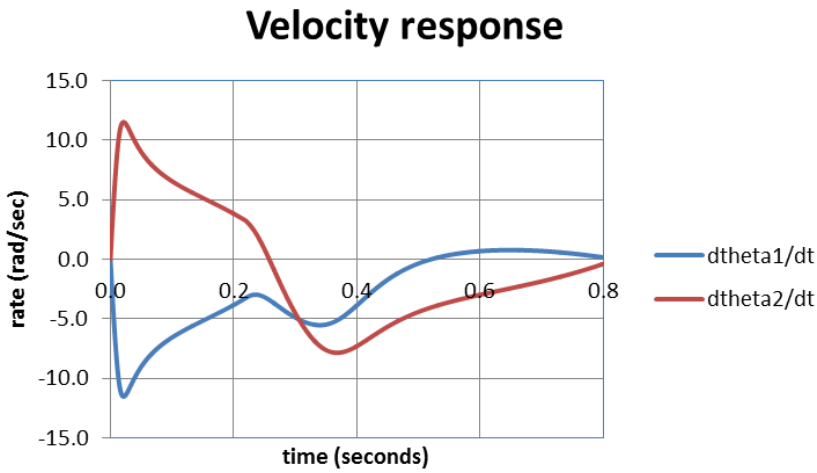



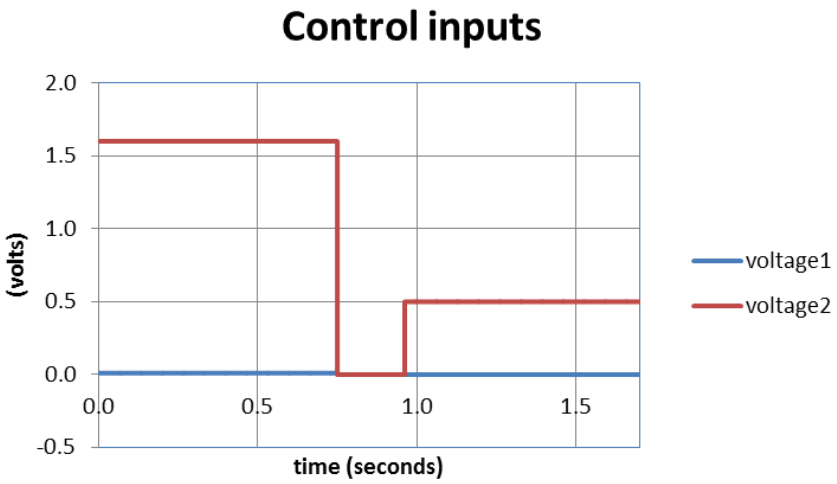
**Rhythm: prone to crouching right (p2cr)**

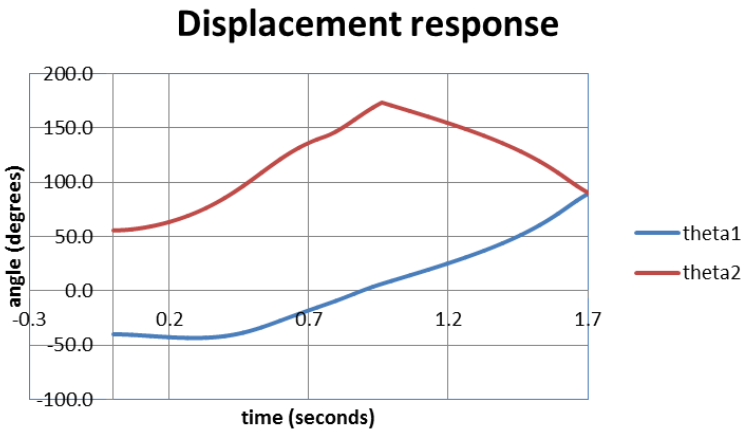


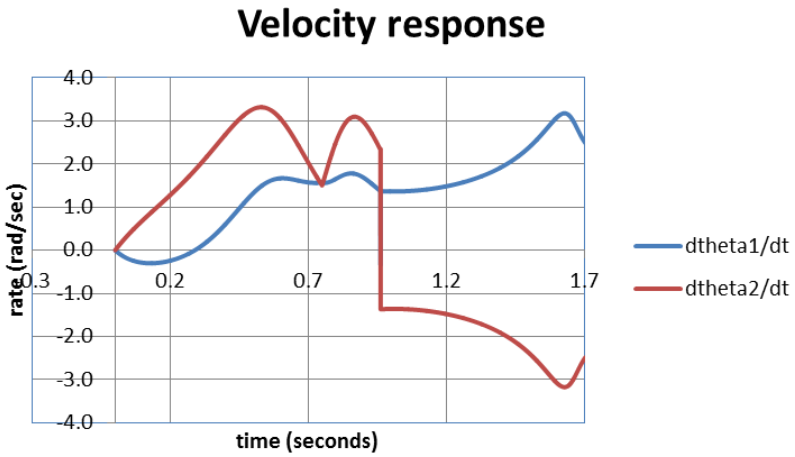



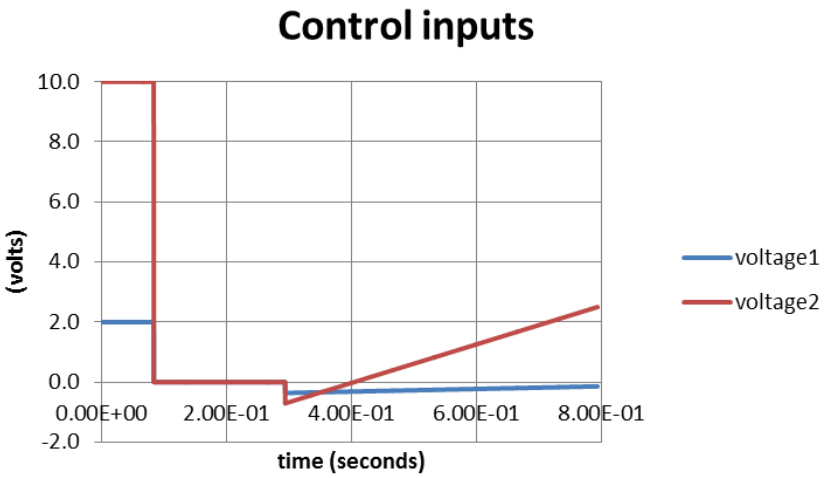
**Rhythm: crouching right to prone (cr2p)**

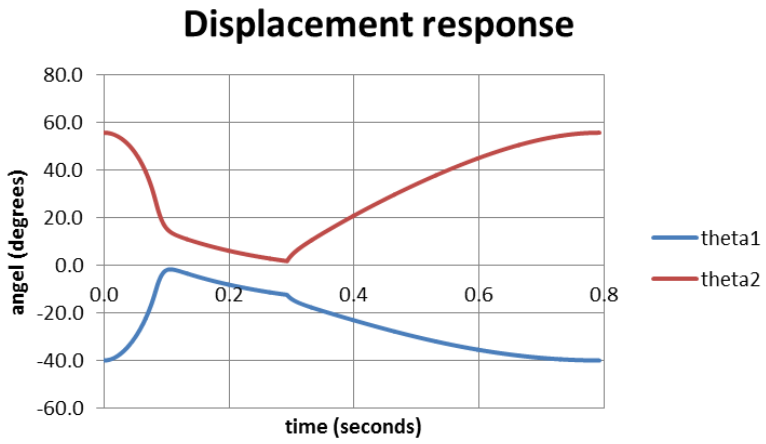


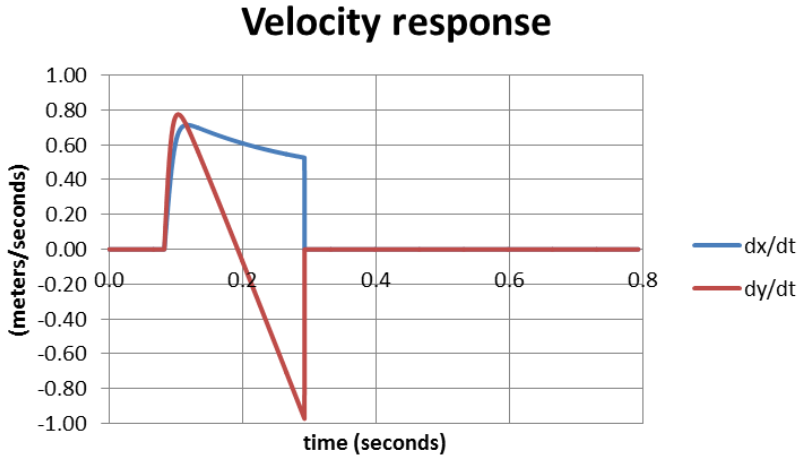



**Rhythm: jump to crouching right (j2cr)**



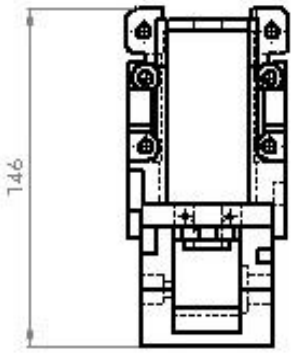



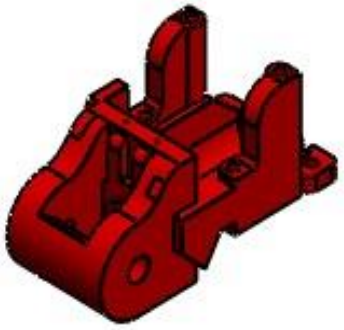
INTENTIONALLY LEFT BLANK

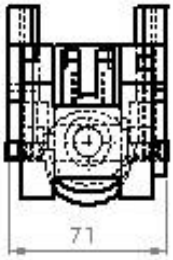
## Appendix F: Additive manufactured parts

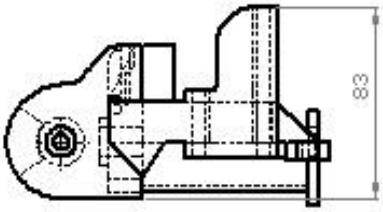
The following pages show the solid models used to produce the additive manufactured substructure in the two-link robot. Rough dimensions are given.

**Part Name: First link, foot**
**Material: ABS**
**Drawing units: mm**

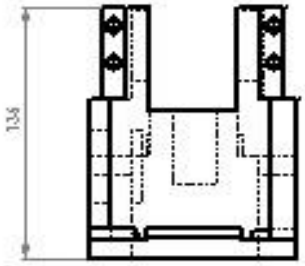


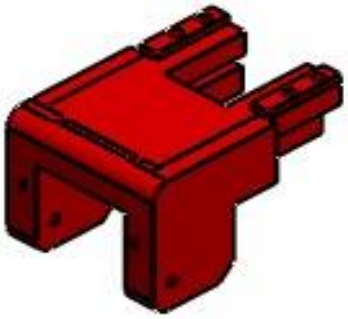



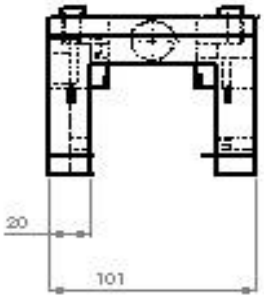


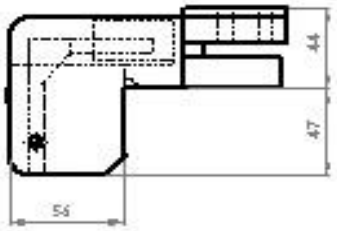




**Part Name: First link, knee**
**Material: ABS**
**Drawing units: mm**

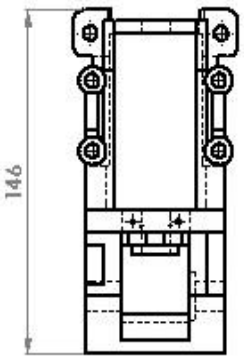


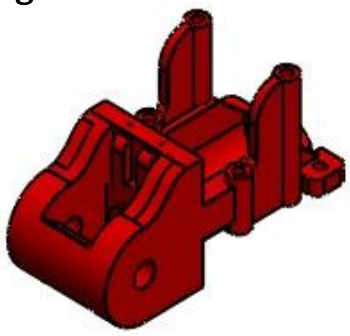



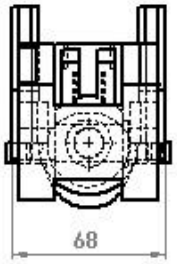


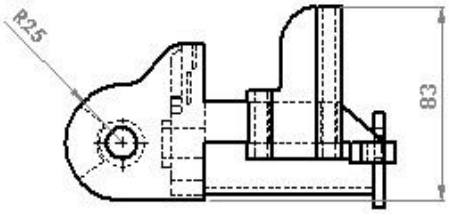



Part Name: Second link, knee
Material: ABS
Drawing units: mm

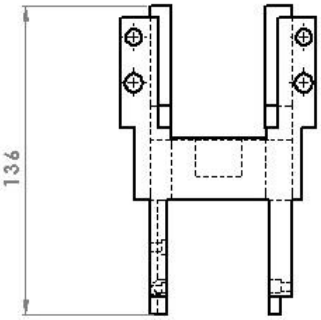


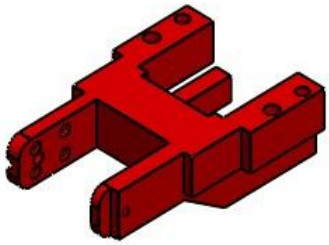



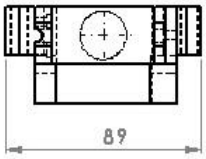


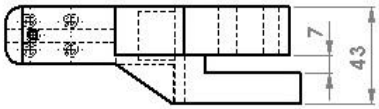




Part Name: Second link, head
Material: ABS
Drawing units: mm

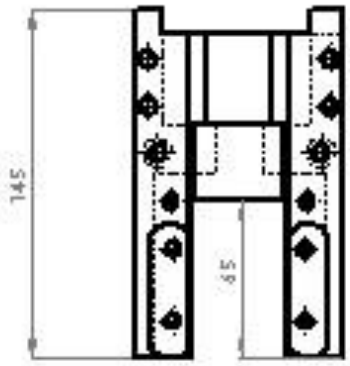


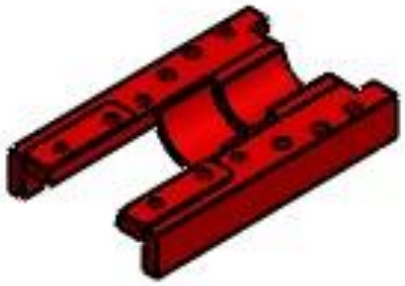



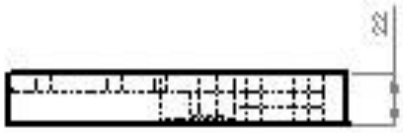


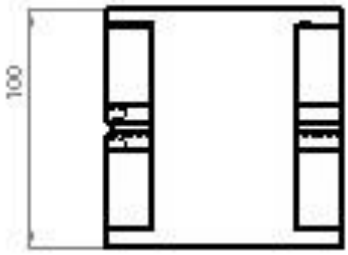
Part Name: Leg
Material: ABS
Drawing units: mm

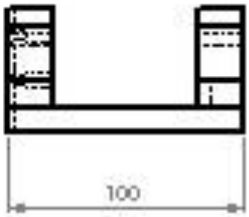


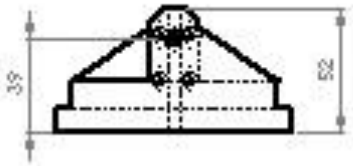



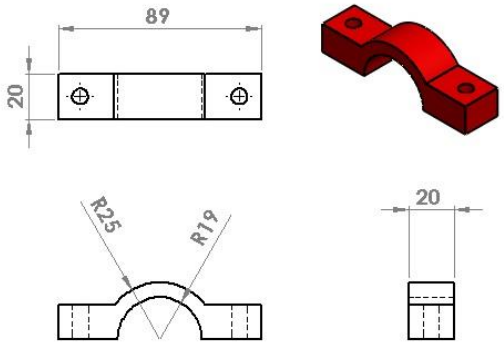




Part Name: Foot
Material: ABS
Drawing units: mm

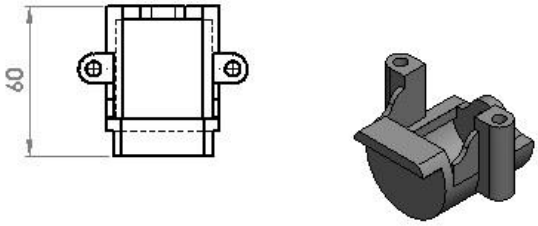




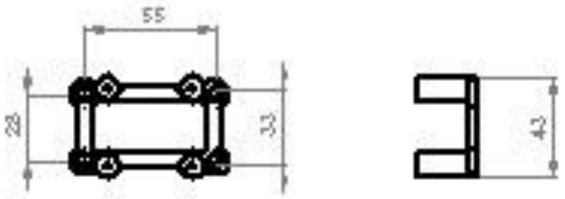

Part Name: motor strap
Material: ABS
Drawing units: mm

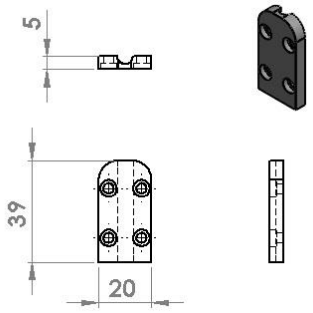



Part Name: senor support
Material: PLA
Drawing units: mm

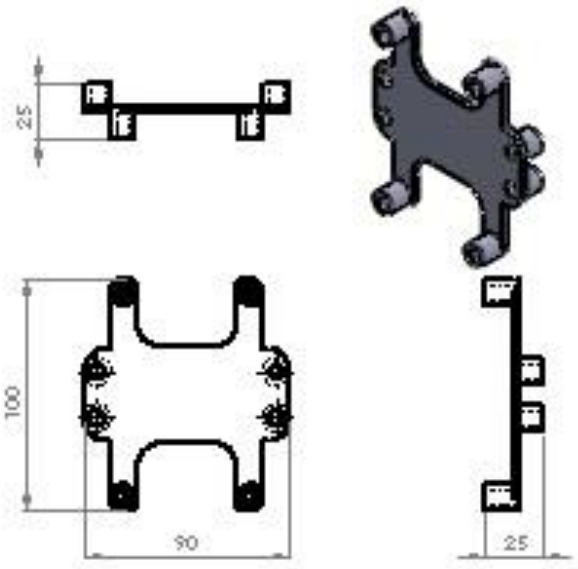


Part Name: wiring support
Material: PLA
Drawing units: mm

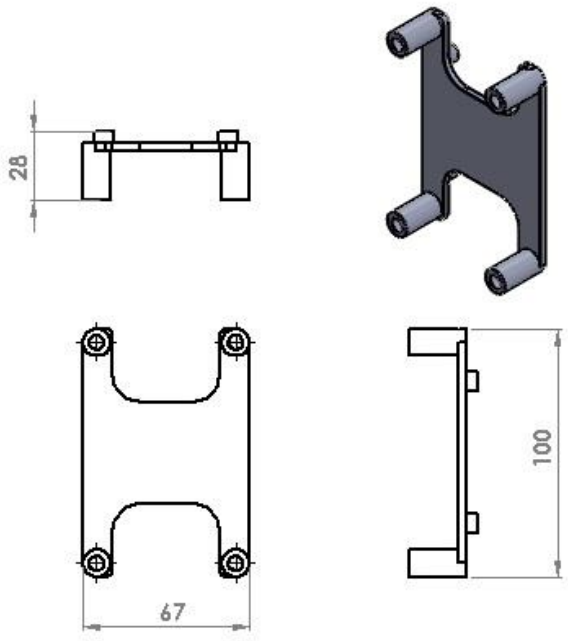




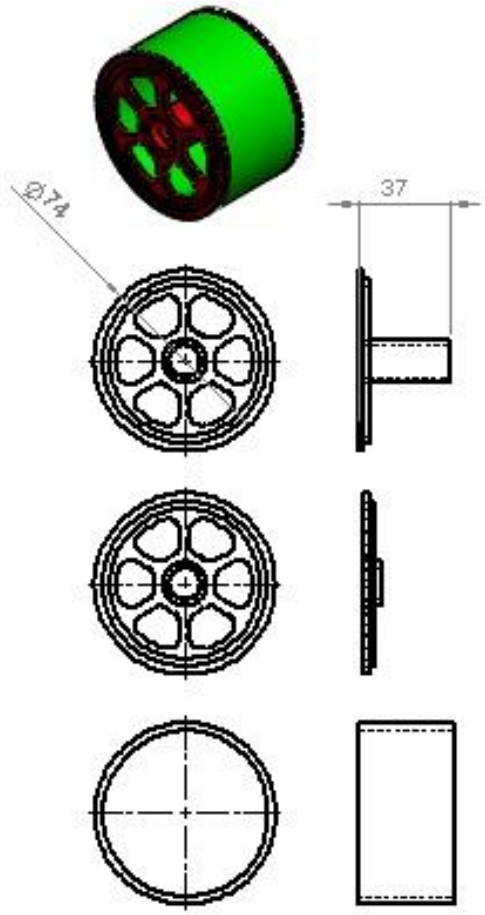

Part Name: pin support
Material: PLA
Drawing units: mm




Part Name: control board support
Material: PLA
Drawing units: mm




Part Name: H bridge board support
Material: PLA
Drawing units: mm



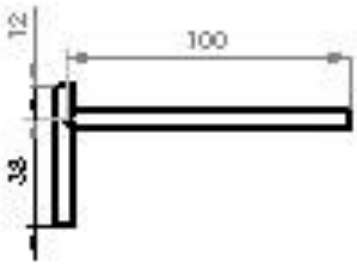

Part Name: wheel left spokes
Material: PLA
Drawing units: mm

Part Name: wheel right spokes
Material: PLA

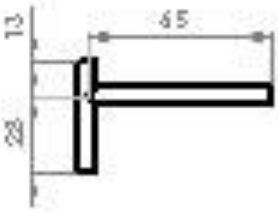
Part Name: wheel rim
Material: PLA

## Appendix G; Welded parts

The pins in the two-link were constructed by TIG welding together 6mm diameter bar stock. The blow drawings show the dimension of these parts.

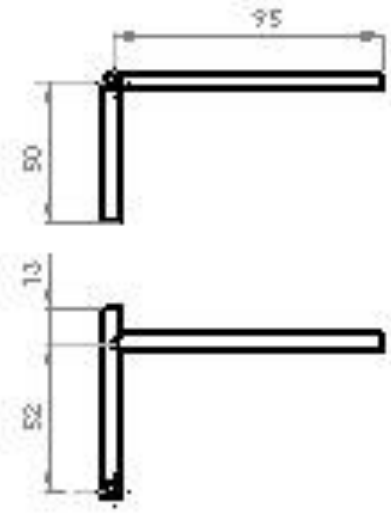



Part Name: foot pin
Material: 6mm diameter bar stock
Drawing units: mm




Part Name: head pin
Material: 6mm diameter bar stock
Drawing units: mm

Part Name: knee pin
Material: 6mm diameter bar stock
Drawing units: mm

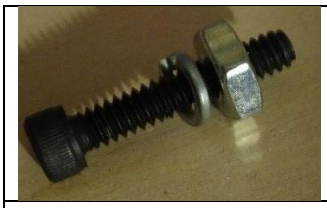
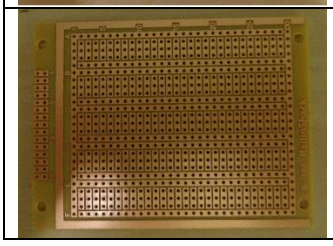


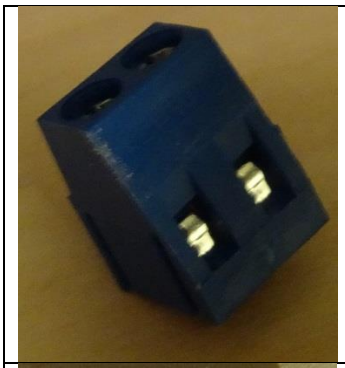
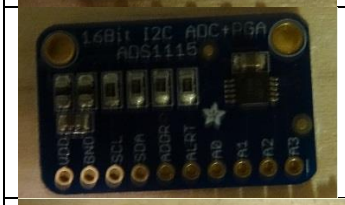
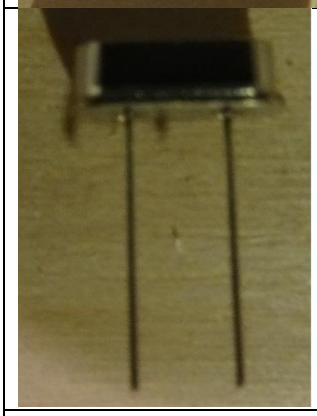

INTENTIONALLY LEFT BLANK

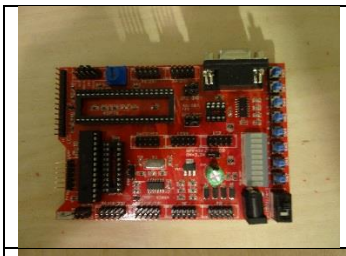
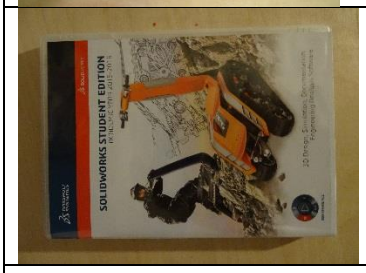
# Appendix H: Purchased parts

Table 5. Purchased parts list

| Photo of Part | Part description | number | supplier | Comments |
|---|---|---|---|---|
| | | | | |
|  | High torque DC geared motor | 2 | Robokit India | 60 rpm |
|  | 24T, 6mm bore, 32 bevel gear | 4 | ServoCity | Part number 615406 |
|  | 6mm to 6mm set screw shaft coupler | 2 | ServoCity | Part number 625230 |
|  | Ball bearings, 6mm bore, 13mm outer | 10 | Ebay or VXB bearings | item# 360189200823 6mm bore, 13mm outer diameter |
|  | 32 tooth, 32 pitch, 6mm bore | 2 | MotionCo. | PC: GR1M060B-6 60 tooth spear gear, 6mm bore |
|  | TemCo microlimit switch | 10 | ebay | Item# 301029432479 This part can be found in many electronic stores. |

| | | | | |
|---|---|---|---|---|
|  | 6-32X3/4’” bolts and nuts | 8 | | Hobby stores which sell RC parts will carry these bolts. |
|  | 6-32 steel bolts and hex nuts, cut or purchased to length. | 28, a variety of lengths | Home Depot | Bolts can be obtained for a variety of lengths. Odd lengths can be can be cut from threaded stock. |
|  | 6mm shaft | Multiple lengths | ServoCity | Cut to length and machined using a micro mill. |
|  | $10k\Omega$ linear rotational potentiometer, ¼” shaft (6mm preferred if available) | 2 | NTE or Radio Shack | Try to find a potentiometer with low friction. The shaft of the potentiometer must be trimmed down to 6mm diameter. |
|  | 20 amp H-bridge, Simple-H | 2 | Jamco Electronics | If you use a substitute, make sure that it can produce 20 amps for PWM excitations up to 20kHz. |
|  | Radio Shack IC solder board | 2 | Radio Shack | Part: 276-1688 This is very good board for IC solder boarding. |

| | | | | |
|---|---|---|---|---|
|  | Wire connectors | 5 | Radio Shack or similar | This is common part that is carry by most electronic shops. |
|  | MCP4725 DAC, breakout board | 2 | Adafruit | Part #: W2032 |
|  | 16-Bit ADC with Ampplifier, 1115 | 2 | Adafruit | Part #: W1678 |
|  | PIC16F873A-I/SP-ND mid-range micro controller | 2 | Digi-Key | Part: PIC16F873A-I/SP-ND |
|  | Crystal 10.0 MHz | 2 | Digi-Key | Part: 631-1101-ND |
|  | PicKit 3 programmer | 1 | Micro Chip | You can also use a Pickit 2. The software to program in assembly can be download from the Micro Chip web site. |

| | | | | |
|---|---|---|---|---|
|  | KanaKit evaluation board | 1 | KanaKit | Some type of evaluation board is useful. |
|   | Solid modeling software | 1 | SolidWork or other | Some type of modeling capability is needed to modify your design. |
| | Miscellaneous parts | | | Smaller parts such as resisters, wires, ect… are not included within this list. These parts can be found in most electronic stores. |

## Appendix I: Raspi C-code for robotic rhythms and virtual constraints

```
#include <stdio.h>
#include <fcntl.h>
#include <inttypes.h>
#include <linux/i2c-dev.h>
#include <math.h>
#include <time.h>

int main() {

// define A2D addresses (ADS1115's)
int A2D_address0 = 0x48;
int A2D_address1 = 0x49;
int A2D_I2CFile0;
int A2D_I2CFile1;

// define D2A addresses (MPC4725's)
int D2A_address0 = 0x62;
int D2A_address1 = 0x63;
int D2A_I2CFile0;
int D2A_I2CFile1;

// declare working variables
int i;
int a,b,c,h1,h2,h3;
double v_in_0,v_out_0;
double v_in_1,v_out_1;
double theta0_set,theta1_set;
double kp_0=0.10;
double kp_1=0.10;
int Nt=10000;
int stage=1;
int I1=0,I2=0,I3=0,I4=0,I5=0,I6=0,I7=0;
double A=0.40;
int D=150;


// declare buffers
uint8_t A2D_writeBuf0[3];
uint8_t A2D_readBuf0[2];
int16_t A2D_val0;
uint8_t A2D_writeBuf1[3];
uint8_t A2D_readBuf1[2];
int16_t A2D_val1;
uint8_t D2A_writeBuf0[3];
uint8_t D2A_readBuf0[2];
uint8_t D2A_writeBuf1[3];
uint8_t D2A_readBuf1[2];

// open the A2D for reading address 0
```

```
A2D_I2CFile0 = open("/dev/i2c-1",O_RDWR);
ioctl(A2D_I2CFile0, I2C_SLAVE, A2D_address0);

// open the A2D for reading address 1
A2D_I2CFile1 = open("/dev/i2c-1",O_RDWR);
ioctl(A2D_I2CFile1, I2C_SLAVE, A2D_address1);

// open the D2A for writing address 0
D2A_I2CFile0 = open("/dev/i2c-1",O_RDWR);
ioctl(D2A_I2CFile0, I2C_SLAVE, D2A_address0);

// open the D2A for writing address 1
D2A_I2CFile1 = open("/dev/i2c-1",O_RDWR);
ioctl(D2A_I2CFile1, I2C_SLAVE, D2A_address1);

// loop through time
//for (i=1;i<5000000;i++) {
for (i=1;i<Nt;i++) {


// set parameters on the address 0 A2D
A2D_writeBuf0[0] = 1;
A2D_writeBuf0[1] = 0b11000001;
A2D_writeBuf0[2] = 0b11100011;
write(A2D_I2CFile0, A2D_writeBuf0, 3);

// address 0 A2D ready to read?
A2D_readBuf0[0] = 0;
A2D_readBuf0[1] = 0;
while ((A2D_readBuf0[0] & 0x80) == 0)
{
read(A2D_I2CFile0, A2D_readBuf0, 2);
}

// A2D is ready to read, read data, address 0
A2D_writeBuf0[0] = 0;
write(A2D_I2CFile0, A2D_writeBuf0, 1);
read(A2D_I2CFile0, A2D_readBuf0, 2);

// compute input angles, address 0 (foot)
A2D_val0  = A2D_readBuf0[0]<<8|A2D_readBuf0[1];
v_in_0 = (float)A2D_val0*6.144/32767;
v_in_0 =-(v_in_0-1.63);
v_in_0 = 90.0/2.053*v_in_0;

// set parameter on the A2D, address 1
A2D_writeBuf1[0] = 1;
A2D_writeBuf1[1] = 0b11000001;
A2D_writeBuf1[2] = 0b11100011;
write(A2D_I2CFile1, A2D_writeBuf1, 3);

// address 1 A2D ready to read?
```

```
A2D_readBuf1[0] = 0;
A2D_readBuf1[1] = 0;
while ((A2D_readBuf1[0] & 0x80) == 0)
{
read(A2D_I2CFile1, A2D_readBuf1, 2);
}

// A2D is ready to read, read data, address 1
A2D_writeBuf1[0] = 0;
write(A2D_I2CFile1, A2D_writeBuf1, 1);
read(A2D_I2CFile1, A2D_readBuf1, 2);

// compute input angles, address 1 (knee)
A2D_val1  = A2D_readBuf1[0]<<8|A2D_readBuf1[1];
v_in_1 = (float)A2D_val1*6.144/32767;
v_in_1 =-(v_in_1-0.289);
v_in_1 = 90.0/1.89*v_in_1;

// echo the input
printf("stage= %d, i= %d, foot = %f (degs), knee = %f (degs)
\n",stage,i,v_in_0,v_in_1);



//rhythm from prone to transition
if(stage==1){
v_out_1    =  -0.35*(1-(float)i/2000);
theta0_set =   0.0015*v_in_1*v_in_1 -0.3218*v_in_1 -85.7955;
v_out_0    =   0.3*kp_0*(theta0_set-v_in_0);
if (v_in_1<-145) {stage=2;theta1_set=-145;theta0_set=6;}
}

//capture at the transition
if(stage==2){
I1=I1+1;
v_out_0 = 0.4* kp_0*(theta0_set-v_in_0);
v_out_1 = 0.6* kp_1*(theta1_set-v_in_1);
if(I1>200){stage=3;}
}

//rhythm from transition to crouching
if(stage==3){
I2=I2+1;
theta0_set = -0.0041*v_in_1*v_in_1 - 0.9027*v_in_1 - 35.0961;
v_out_0= 0.080*(theta0_set-v_in_0);
v_out_1=0.25;
if (v_in_1>-106){stage=4;theta0_set=13;theta1_set=-106;}
}

//capture at crouching
if(stage==4){
I3=I3+1;
```

```
v_out_0 =  0.5*kp_0*(theta0_set-v_in_0);
v_out_1 =  0.5*kp_1*(theta1_set-v_in_1);
if(I3>200){stage=6;}
}

//first part of jump rhythm
if(stage==6){
I5=I5+1;
theta0_set = -0.0205*v_in_1*v_in_1 - 4.6956*v_in_1 - 253.5337;
v_out_0= 0.080*(theta0_set-v_in_0);
v_out_1=-0.10;
if (v_in_1<-147){stage=7;}
}

//second part of jump rhythm
if(stage==7){
I6=I6+1;
v_out_1    =   0.15;
theta0_set =   -0.5387*v_in_1 -89.8062;
v_out_0    =   0.3*kp_0*(theta0_set-v_in_0);
if (v_in_1>-5) {stage=8;theta0_set=-84;theta1_set=-5;}
}

//capture at the end of a jump
if(stage==8){
I7=I7+1;
v_out_0 =  0.5*kp_0*(theta0_set-v_in_0);
v_out_1 =  0.5*kp_1*(theta1_set-v_in_1);
//if(I3>200){stage=9;}
}

// echo the outputs
//printf("i= %d, theta1 = %f (degs), theta2 = %f (degs) \n", i,
theta1_set, theta2_set);



// convert v_out_0 to a Hexadecimal number
a     = (int)2000*v_out_0+2048;
b     = a/16;
h1    = a%16;
a     = b;
b     = a/16;
h2    = a%16;
h3    = b;

//convert the high and low decimal and output
D2A_writeBuf0[0] = h3;
D2A_writeBuf0[1] = 16*h2+h1;
write(D2A_I2CFile0,D2A_writeBuf0,2);

// convert v_out_1 to a Hexadecimal number
```

```
a     = (int)2000*v_out_1+2048;
b     = a/16;
h1    = a%16;
a     = b;
b     = a/16;
h2    = a%16;
h3    = b;

//convert the high and low decimal and output
D2A_writeBuf1[0] = h3;
D2A_writeBuf1[1] = 16*h2+h1;
write(D2A_I2CFile1,D2A_writeBuf1,2);

}  // end of time loop

// close out the i2c interface
close(A2D_I2CFile0);
close(A2D_I2CFile1);
close(D2A_I2CFile0);
close(D2A_I2CFile1);

return 0;

} // end of main
```

INTENTIONALLY LEFT BLANK

## Appendix J: PIC machine code for pulse width modulation

```
; 16F873a PWM example code.
LIST P=16F873, R=DEC    ; Use the PIC16F873 and decimal system
#include "P16F873A.INC"     ; Include header file
__config  _CP_OFF & _HS_OSC & _WDT_OFF & _PWRTE_ON & _LVP_OFF & _BODEN_ON

            offset  EQU 0x0
            ;offset  EQU 0x2F
            ;offset  EQU   0x00

   cblock       0x20
                save_status
                save_low
                rpm_left
                rpm_right
                output_leds
   endc

   ORG          0x0000
   GOTO         START
   ORG          0x0004

START:          ;Setup AtoD on RA0
   movlw        B'01000001'
   movwf        ADCON0

                ; set up output and 2nd part of ADC
   BANKSEL      TRISC
   CLRW
   MOVWF        TRISC
   clrf         ADCON1
   BANKSEL      PORTC

   MOVLW        b'00001100'
   MOVWF        CCP1CON
   MOVWF        CCP2CON

   MOVLW        0xFE
   BANKSEL      PR2
   MOVWF        PR2
   BANKSEL      TMR2

   MOVF         T2CON,W
   ANDLW        0xF8
   IORLW        0x00
   MOVWF        T2CON

   MOVF         T2CON,W
   ANDLW        0x07
```

```
   IORLW       0x00
   MOVWF       T2CON

   CLRF        CCPR1L
   CLRF        CCPR2L

   BSF         T2CON, TMR2ON

MainLoop:    CALL    read_pot
   CALL        SpeedSet
   goto        MainLoop


SpeedSet:       MOVF  rpm_left,0
   MOVWF       CCPR1L
   MOVF        rpm_right,0
   MOVWF       CCPR2L
   RETURN

read_pot:       bsf     ADCON0,GO
Wait:           btfsc   ADCON0,GO
   goto        Wait
   rlf         ADRESH,1
   movf        STATUS,0
   movwf       save_status
   btfsc       save_status,0
   call        set_upper
   btfss       save_status,0
   call        set_lower
   return

set_upper:      movf    ADRESH,W
   movwf       rpm_left
   movlw       offset
   addwf       rpm_left,1
   btfsc       STATUS,C
   call        max_left
   movlw       0x00
   movwf       rpm_right
   return

max_left:       movlw 0xFF
   movwf       rpm_left
   return

set_lower:      movf    ADRESH,W
   sublw       0xFF
   movwf       rpm_right
   movlw       offset
   addwf       rpm_right,1
```

```
    btfsc       STATUS,C
    call        max_right
    movlw       0x00
    movwf       rpm_left
    return

max_right:      movlw  0xFF
    movwf       rpm_right
    return

    END
```

INTENTIONALLY LEFT BLANK

## Appendix K: Experimentally derived equations of motion

In this dissertation, a control algorithm for a robot linkage system was presented. The algorithm was demonstrated using a specific linkage system containing four point masses. In this appendix we two-link dynamics by using the mass properties of the links.

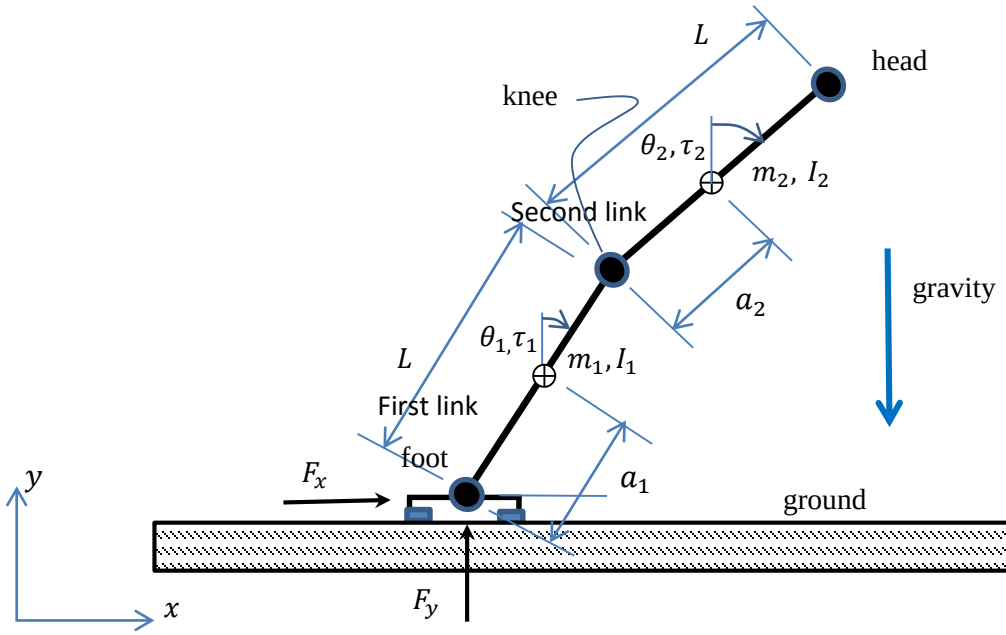



**Figure 73. Generalized two-link dynamics**

The definition of these properties are given in Table 6.

**Table 6. Definition of link mass properties and the experimental method to determine them**

| Mass property | Description | Experimental method of determination |
|---|---|---|
| $a_1$ | Location of the COM of first link | Using a balance to find the COM |
| $m_1$ | Total mass of the first link | Using a scale to measure weight |
| $I_1$ | Moment of inertia at the COM of the first link | Resonance testing about the COM |
| $a_2$ | Location of the COM of the second link | Using a balance to find the COM |
| $m_2$ | Total mass of the second link | Using a scale to measure weight |
| $I_2$ | Moment of inertia at the COM of the second link | Resonance testing about the COM |

Using the Euler-Lagrange (EL) equations free dynamics are given by

$$\boldsymbol{M}\ddot{\vec{X}} = \boldsymbol{G} + \boldsymbol{F}$$

where

$\vec{X} = [\theta_1, \theta_2, x, y]^T$

$$\boldsymbol{M} = \begin{bmatrix} m_1a_1^2 + m_2L^2 + I_1 & m_2a_2Lcos(\theta_1 - \theta_2) & m_1a_1 + m_2L & -(m_1a_1 + m_2L)sin(\theta_1) \\ m_2a_2Lcos(\theta_1 - \theta_2) & m_2a_2^2 + I_2 & m_2a_2cos(\theta_2) & -m_2a_2sin(\theta_2) \\ m_1a_1 + m_2L & m_2a_2cos(\theta_2) & m_1 + m_2 & 0 \\ -(m_1a_1 + m_2L)sin(\theta_1) & -m_2a_2sin(\theta_2) & 0 & m_1 + m_2 \end{bmatrix}$$

$$\boldsymbol{G} = \begin{bmatrix} -m_2a_2Lsin(\theta_1 - \theta_2)\dot{\theta}_2^2 + (m_1a_1 + m_2L)gsin(\theta_1) \\ m_2a_2Lsin(\theta_1 - \theta_2)\dot{\theta}_1^2 + m_1a_1gsin(\theta_2) \\ (m_1a_1 + m_2L)sin(\theta_1)\dot{\theta}_1^2 + m_2a_2sin(\theta_2)\dot{\theta}_2^2 \\ (m_1a_1 + m_2L)cos(\theta_1)\dot{\theta}_1^2 + m_2a_2cos(\theta_2)\dot{\theta}_2^2 \end{bmatrix}$$

$$\boldsymbol{F} = [\tau_1 \quad \tau_2 \quad F_x \quad F_y]^T$$

As described in chapter 2, from these equations, all lower order dynamics can be derived. Setting the time derivatives of $x, y$ to zero and solving for the reaction forces gives the generalized pinned dynamics response

$$\begin{bmatrix} m_1a_1^2 + m_2L^2 + I_1 & m_2a_2Lcos(\theta_1 - \theta_2) \\ m_2a_2Lcos(\theta_1 - \theta_2) & m_2a_2^2 + I_2 \end{bmatrix} \begin{Bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{Bmatrix}$$

$$= \begin{bmatrix} -m_2a_2Lsin(\theta_1 - \theta_2)\dot{\theta}_2^2 + (m_1a_1 + m_2L)gsin(\theta_1) \\ m_2a_2Lsin(\theta_1 - \theta_2)\dot{\theta}_1^2 + m_1a_1gsin(\theta_2) \end{bmatrix} + \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

$$F_x = (m_1a_1 + m_2L)\ddot{\theta}_1 + m_2a_2cos(\theta_2)\ddot{\theta}_2 - (m_1a_1 + m_2L)sin(\theta_1)\dot{\theta}_1^2 - m_2a_2sin(\theta_2)\dot{\theta}_2^2$$

$F_y = -(m_1a_1 + m_2L)sin(\theta_1)\ddot{\theta}_1 - m_2a_2sin(\theta_2)\ddot{\theta}_2$-$(m_1a_1 + m_2L)cos(\theta_1)\dot{\theta}_1^2 - m_2a_2cos(\theta_2)\dot{\theta}_2^2$ .

Again, following chapter 2, taking the difference of the above equations, generalized bent dynamics can be found

$$[m_1a_1^2 + m_2(a_2^2 + 2a_2Lcos(2\theta_1) + L^2) + I_1 + I_2]\ddot{\theta}_1$$

$$= 2m_2a_2Lsin(2\theta_1)\dot{\theta}_1^2 + (m_1a_1 + m_2L - m_2a_2)gsin(\theta_1) + (\tau_1 - \tau_2)$$

These equations are solely a function of the mass properties of the links which were found experimentally. Figure 74 and Figure 75 contain photos of the experimental apparatus. It consist of a carriage attached to a spring.

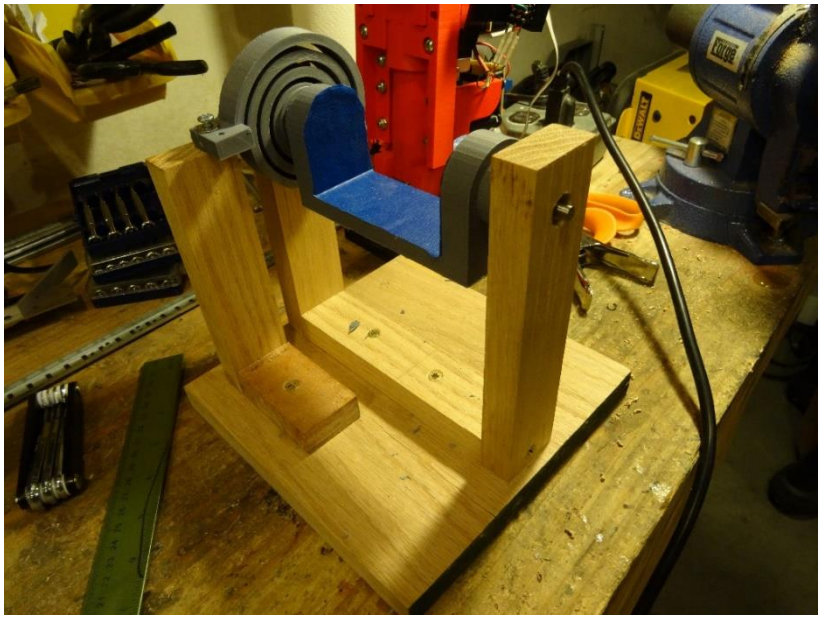


Figure 74. Experimental apparatus for determining link mass properties

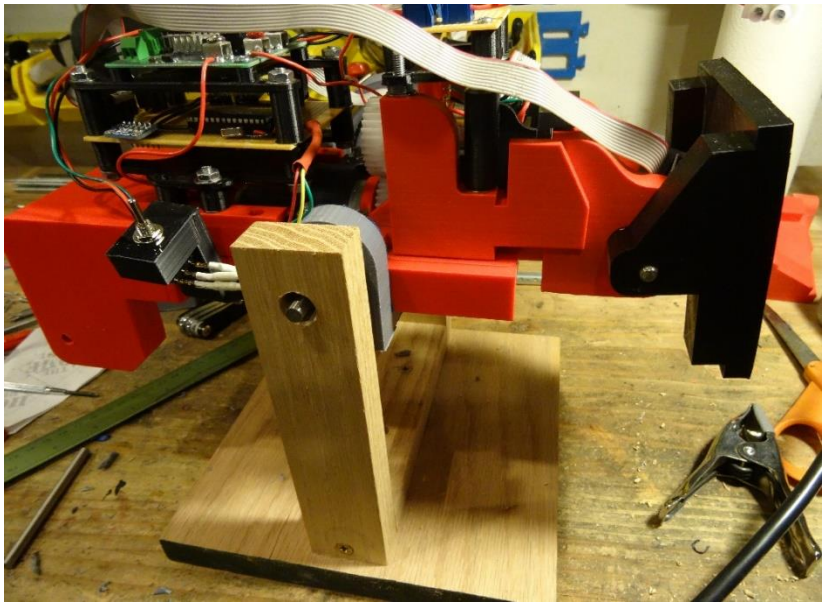


Figure 75. Link within the experimental apparatus

The link is placed within the carriage and it translated until it is balanced in a horizontal position. The balance point is the center of gravity of the link. The link can then be depressed at one end and allowed to vibrate at its natural frequency. By measuring both the natural frequency and the

stiffness of the spring, the moment of inertial of the link can be determined. The mass of the link can be determined simply by measuring its weight. Table 7 give the mass properties of the links.

**Table 7. Mass properties of two-link legs**

| Parameter | Value | Unit |
|---|---|---|
| $m_1$ | 1.37 | $kg$ |
| $m_2$ | 1.10 | $kg$ |
| $I_1$ | 0.0164 | $kg \cdot m^2$ |
| $I_2$ | 0.0062 | $kg \cdot m^2$ |
| $a_1$ | 0.12 | $m$ |
| $a_2$ | 0.13 | $m$ |
| $l_1$ | 0.272 | $m$ |
| $l_2$ | 0.280 | $m$ |

## References


[1] Qureshi, M., O., and Syed, R., S., (2014). The Impact of robotics on employment and motivation of employees in the service sector, with special reference to health care. *Safety and health at work*, **5**: 198-202.

[2] Computing Community Consortium, (2009). A roadmap for US robotics from internet to robotics. White paper, http://www.cra.org/ccc/ .

[3] Hanson, R., (2009). Economic growth given machine intelligence. White paper, hanson@econ.berkeley.edu .

[4] Laird, R., (2009). Evolving U.S. Department of Defense (DoD). Unmanned Systems Research, Development, Test, Acquisition & Evaluation (RDTA&E). *SPIE Proc. 7332: Unmanned Systems Technology XI, Defense Security Symposium, Orlando, FL,* 13-17.

[5] Pinkuhiremath, (2013). Robotics in healthcare: Personalizing care and boosting the quality, access and efficiency of healthcare. http://www.studymode.com/essays/Robotics-In-Healthcare-Personalizing-Care-And-46056144.html .

[6] Greenblatt, D., (2013). The ways robots create jobs. http://www.inc.com/drew-greenblatt/6-ways-robots-createjobs.html .

[7] Sims, D., (2013). Are robots killing jobs or creating them? http://news.thomasnet.com/IMT/2013/02/05/are-robots-killing-jobs-or-creating-them .

[8] Grizzle, J., W., Chevallereau, C., Sinnet, R., W., Ames, A., D., (2014), Models, feedback control, and open problems of 3D bipedal robotic walking. *Automatica.* **50**: 1955-1988.

[9] Hurmuzlu, Y., Génot, F., and Brogliato, B., (2004). Modeling, stability and control of biped robots – a general framework. Automatica. **40**: 1647-1664.

[10] Xiang, Y., Arora, J., S., and Abdel-Malek, K., (2010). Physics-based modeling and simulation of human walking: A review of optimization-based and other approaches. *Structural and multidisciplinary optimization.* **42**(1): 1-23.

[11] Wahde, M., and Pettersson, J., (2001). A brief review of bipedal robotics research, White paper. Chalmers University of Technology.

[12] Kajita, S., Yamaura, T., and Kobayashi, A., (1992). Dynamic walking control of a biped robot along a potential-energy conserving orbit. *IEEE Transactions on robotics and automation.* **8**(4): 431-438.

[13] Kajita, S., and Tani, K. (1991). Study of dynamic biped locomotion on rugged terrain. *Proceedings of the 1991 IEEE International conference on robotics and automation.* 1405-1411.

[14] Kajita, S., Matsumoto, O., and Muneharu, S., (2001). Real-time 3D walking pattern generation for a biped robot with telescopic legs. *Proceedings of the 2001 IEEE International conference on robotics & automation, Seoul, Korea.* 2299-2306.

[15] Kajita, S., Kanehiro, F., Kaneko, K., Fuiwara, K., Yokio K., and Hirukawa, H., (2002). A realtime pattern generator for biped walking. *Proceedings of the 2002 IEE International conference on robotics & automation.* 31-37.

[16] Park, J., H., and Kim, K., D., (1998). Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control. *Proceedings of the 1998 IEEE International conference on robotics & automation.* 3528-3533.

[17] Albert, A., and Gerth, W., (2003). Analytic path planning algorithms for bipedal robots without a trunk. *Journal of intelligent and robotic systems.* **36**: 109-127.

[18] Luo,R., C., Sheng J., Chang P., (2013). Biped robot limit cycle walking generation considering energy dissipation caused by impact*. International conference on advanced robotic and intelligent systems.* 17-22.

[19] Ha, T., and Chio, C., (2007). An effective trajectory generation method for bipedal walking. *Robtics and autonomous systems.* **55**: 795-810.

[20] Mochon, S., and McMahon, T., (1980). Ballistic walking: an improved model. *Mathematical biosciences.* **52**:241-260.

[21] McGeer, T., (1990). Passive dynamics walking. International journal of robotic research. **9**(2):62-82.

[22] Goswami, A., Espiau, B., and Keramane A., (1997). Limit cycles in a passive compass gait biped and passivity-mimicking control laws. *Autonomous robots*. **4**:273-386.

[23] Liu, Z., Zhang, A., Tian, Y., Zhang P., and Gao, D., (2010). Further analysis of the kneed passive-dynamic biped robot, *Proceedings of the 2010 Chinese control and decision conference*. 1777-1782.

[24] Borzova, E., and Hurmuzlu, Y., Passively walking five-link robot. *Automatica.* **40**:621-629.

[25] Asano, F., (2013). High-speed dynamic gait generation for limit cycle walkers based on forward-tilting impact posture. *Multibody system dynamics*. **30**:287-310.

[26] Vukobratovic, M., and Borovac, B., (2004). Zero-moment point – Thirty five years of its life. *International journal of humanoid robotics.* **1**(1): 157-173.

[27] Vukobratović, M., Borovac, B., Potkonjak, V., and Jovanović M., (2009). Dynamic balance of humanoid systems in regular and irregular gaits: An expanded interpretation. *International journal of humanoid robotics.* **6**(1): 117-145.

[28] Vukobratović, M., and Juricic, D. (1969). Contribution to synthesis of biped gait. *IEEE Transactions on bio-medical engineering.* **16**(1):1-16.

[29] Yamajuchi, J., Soga, E., Inoue, S., and Takanishi, A., (1996). Development of a bipedal humanoid robot – control method of whole body cooperative dynamic biped walking. Proceedings of the 1999 *IEEE International conference on robotics & automation*, **1**:368-374.

[30] Lim, H., Kaneshima, Y., and Takanishi, A**.,** (2002). A., Online walking pattern generation for biped humanoid robot with trunk. IEEE computer society press.**2**:3111-3116.

[31] Lim, H., and Takanishi A., (2007). Biped walking robots created at Waseda University: WL and WABAIN Family. *Philosophical transactions of the royal society A: Mathematical physical and engineering sciences*. **365**:49-64.

[32] Vermeulen, J., Verrelst, B., Vanderborght, B., Lefeber, D., and Guillaume, P., (2006). Trajectory planning for the walking biped "Lucy." *The international journal of robotics research.* **25**(9):867-887.

[33] Zheng, Y., F., and Shen, J., (1990). Gait synthesis for the SD-2 biped robot to climb sloping surface. *IEEE Transactions on robotics and automation*. **6**(1):86-96.

[34] Huang, Q., Yokoi, K., Kajita, S., Kaneko K., Arai H., Koyachi, N., Tanie, K., (2001). Planning walking patterns for a biped robot, IEEE Transactions on robotics and automation. 17(3): 280-289.

[35] Jiang, Y., Yuan, H., Zhu, C., and Cui Y., (2013). Gait design of biped robot based on USBSSC32. *IEEE 8th Conference on Industrial Electronics and Applications (ICIEA).* 978-981.

[36] Massah B., A., Zamani, A., Salehinia, Y., Aliyari Sh, M., Teshnehlab, M., (2013). A hybrid controller based on CPG and ZMP for biped locomotion. Journal of mechanical science and technology. **27**(11):3473-3486.

[37] Shimmyo, S., Sato, T., and Ohnishi, K., (2013). Biped walking pattern generation by using preview control based on three-mass model. *IEEE Transactions on industrial electronics*. **60**(11):5137-5147.

[38] Shin, H., and Kim, B., K., (2014). Energy-efficient gait planning and control for biped robots utilizing the allowable ZMP region. *IEEE Transactions on robotics*. **30**(4):986-993.

[39] Zhan, J., Xiong, R., and Sun Y., (2013). A ZMP and CoM online replanning method for stable walking of bipedal robots. *2013 25th Chinese Control and Decision Conference (CCDC).* 2605-2611.

[40] Hirai, K., Masato, H., Yuji, H., and Takenaka, T., (1998). The development of Honda humanoid robot. *Proceedings of the 1998 IEEE International conference on robotics & automation*. 1321-1326.

[41] Ishida, T., Yoshihiro, K., Yamaguchi J., Fujita, M., and Dio T., T., (2001). Motion entertainment by a small humanoid robot based on OPEN-R. *Proceedings of the 2001 IEEE/RSJ International conference on intelligent robots and systems.* 1079-1086.

[42] Eriksson, A., (2007). Temporal finite elements for target control dynamics of mechanisms. *Computer and structures*. 85:1399-1408.

[43] Kaphle, M., and Eriksson, A., (2008). Optimality in forward dynamics simulations. Journal of biomechanics. **41**:1213-1221.

[44] Eriksson, A., and Nordmark, M., (2011). Temporal finite element formulation of optimal control in mechanics. *Computer methods in applied mechanics and engineering.* **199**:1783-1792.

[45] Kwon, H., Xiang, Y., Bhatt R., Rahmatalla, S., Arora, J., and Abdel-Malek, K., (2014). Backward walking simulation of humans using optimization. *Structural and multidisciplinary optimization*. **50**(1):169-179.

[46] Kim, H., H., Xiang, Y., Yang J., Arora J., S., Abdel-Malek, K., (2010). Dynamic motion planning of overarm throw for a biped human multibody system. *Multibody system dynamics*. **24**:1-24.

[47] Crandall, S., Karnopp, D., Kurtz, E., F., Pridmore-Brown, D., C., (1968). *Dynamics of mechanical and electromechanical systems.* McGraw-Hell Book Company, New York, NY.

[48] Hull, D., (2003). *Optimal control theory for applications.* Spring-Verlag, New York, NY.

[49] Raibert, M., H., Brown Jr., H., B., (1984). Experiments in balance with a 2d one-legged hopping machine. *Journal of dynamics systems, measurement and control*. **106**: 75-81.

[50] Raibert, M., H., Brown, H., B., Jr., Murthy, S., S., (1984). 3D balance using 2D algorithms? In *First International Symposium of Robotics Research*, M. Brady, R. P. Paul (eds.), (MIT Press, Cambridge). 279--301.

[51] Raibert, M., H., Brown, H., B., Jr., Chepponis, M., (1984). Experiments in balance with a 3D one-legged hopping machine. *International J. Robotics Research.* **3**:75-92.

[52] Hyon, S., and Emura, T., (2004). Running control of a planar biped robot based on energy-preserving strategy. *Proceedings of the 2004 International conference on robotics & automation.* 3791-3796.

[53] Long, A., W., Murphey, T., D., and Lynch K., M., (2011). Optimal motion planning for a class of hybrid dynamical systems with impacts. *IEEE International Conference on Robotics and Automation (ICRA).* 4220 – 4226.

[54] Pratt, J., Chew, C., Torres, A., Dilworth, P., and Pratt, G., (2001). Virtual model control: An intuitive approach for bipedal locomotion. *The international journal of robotics research*. **20**(2):129-143.

[55] MATLAB and Optimization Toolbox, The MathWorks, Inc., Natick, Massachusetts, United States.

[56] Bryson Jr., A. E., and Ho, Y., (1975). *Applied optimal control, optimization, estimation and control.* Hemisphere Publishing Corporation.

[57] Bryson Jr., A.E. (1999). *Dynamic optimization*. Addison-Wesley. Menlo Park, CA.

[58] Weservelt, E., R., Grizzle, J., Chevallereau, C., Choi, J., and Morris, B., (2007). *Feedback control of dynamic bipedal robot locomotion.* CRC Press, Taylor and Francis. New York, NY.

[59] Khalil, H., K., Nonlinear systems, third edition, Prentice Hall, Upper Saddle River, New Jersey.

[60] Bezier, P., (1972). *Numerical control: Mathematics and applications*. John Wiley & Sons, New York.

[61] Grizzle, J., W., Gabriel, A., and Plestan, F., (2001). Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. *IEEE transaction on automatic control*. 46(1):51-64.

[62] Narendra, K., Annaswamy, A., M., (1989). *Stable adaptive systems.* Prentice Hall, Englewood Cliffs, NJ.

[63] Lewis, F. L., Jagannatha, S., and Yesildirek, A., (1998). *Neural network control of robot manipulators and non-linear systems.* Taylor and Francis. New York, NY.

[64] Kojima, H., Kato, G., (2005), Dynamic analysis of two-link articulated hopping robot with stopper mechanism*, Theoretical and Applied Mechanics, Japan*. **54**:143-150.

[65] Seghete, V., Murphey, T., D., (2013). A propagative model of simultaneous impact: existence, uniqueness, and design consequences. *Automation science and engineering, IEEE Transactions*, **11**(1):154-168.

[66] Seghete, V., and Murphey, T., D., (2009) Multiple instantaneous collisions in a variational framework. *Proceedings of the 48th IEEE conference on decision and control held jointly with the 2009 28th Chinese control conference, CDC/CCC 2009.*

[67] Pekarek, D., and Murphey, T.,D., (2012). Global projections for variational nonsmooth mechanics. P*roceedings of the 51th IEEE conference on decision and control.*

[68] H*ü*rmuzl*ü*, Y., G*é*not, F., et. al., (1994). Rigid body collisions of planar kinematic chains with multiple contact points. *International journal of robotics research*. **13**(1): 82-92.

[69] Pratt, J., Carff J., Drakunov, S., and Goswami, A., (2006) "Capture point: A step toward humanoid push recovery," in *IEEE/RAS Int. Conf. on Humanoid Robots*, 2006, pp. 200–207.

[70] Ultimaker®. https://ultimaker.com/ .

[71] *Solidworks®2015*, Dassault Systemes.

[72] https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c .

[73] Micochip Technology Inc. (2009) 12-bit Digital-to-analog converter with EEPROM memory in SOT-23-6.

[74] Texas Instruments, Ulta-Small, Low-Power, 16-Bit Analog-to-Digital Converter with Internal Reference, ADS115.

[75] Simple-H, Robot Power, www.robotpower.com.

[76] Robokits India, SF-39 Rudra Square,, Judges Bungalow, Bodakdev, Ahmedabad, Gujarat 380015, India.