INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY

# Evolution Inspired Generalized Controller Based Locomotion

by

Tanmay Randhavane

November 2014

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY

# *Abstract*

BTech Thesis - Part I

by Tanmay Randhavane

We develop a generalized locomotion model for the characters that appear in movies, games, etc. We build a physics based controller to model different gaits and skills of the characters and transformations between them. We first build separate controllers for them, abstract out the common part, parameterize the character specific parts and thus generate a unified controller which is able to model locomotion skill set of characters ranging from reptiles, quadrupeds, bipeds, birds, etc.

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

Every life form has a set of locomotion skills that are developed over thousands of years through adaptation. But at the start of it all, there was only one life form having a limited number of skills. Using this idea, we try to create a generalized locomotion controller which is able to control the motion of various characters and transformations between them. The overall problem can be broken into many small problems, first creating seperate simulators for specific classes of characters. Then abstracting out the common features from the controllers and parameterizing the specific features.

## 1.2 Structure of The Report

The problems of creating a physics based locomotion controller for bipeds and quadrupeds are discussed in section 2 and 3 respectively. Section 4 presents ideas on combining the two. We end in section 5 with future plans.

# Chapter 2

# Biped Controller

## 2.1 Introduction

First step in building a generalized controller is to build a controller for specific classes of characters. There are several biped controllers that have been developed [1–4]. We focus on SIMBICON framework [3].

## 2.2 Controller based Locomotion Model

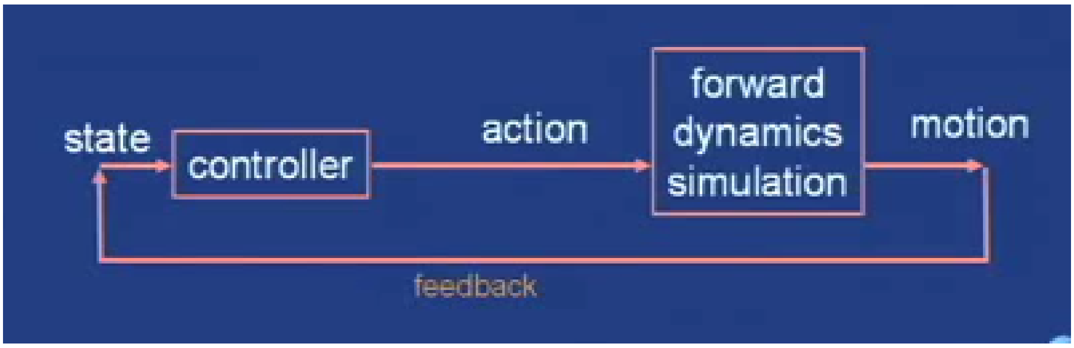


FIGURE 2.1: Controller based Locomotion Model (An abstract view)

In a contoller based locomotion model, the controller creates torques and forces that act on a character given the state of the character. The character state is usually specified in the form of joint angles and location of some of the links. The torques and forces thus generated are passed to the forward dynamics simulator engine which gives the new joint angles and locations. This are passed to the renderer and to the controller as a feedback.

## 2.3 Simple Biped Controller (SIMBICON)

The control strategy of SIMBICON can be described in terms of three elements: pose control graph, torso and swing-hip control, and balance feedback.

### 2.3.1 Pose Control Graph

A pose control graph is a finite state machine in which each state represents a target pose using its target angles. There is a separate state machine for each gait. The transition between states happen after a certain elapsed time or after a foot strike. The joint torques are computed using the current and desired joint angles using proportional-derivative controller.

### 2.3.2 Torso and Swing-hip Control

Using only the pose control graph will not give a robust motion because it has no criteria for balance. To provide that, the torso and swing-hip are handled separately. As shown

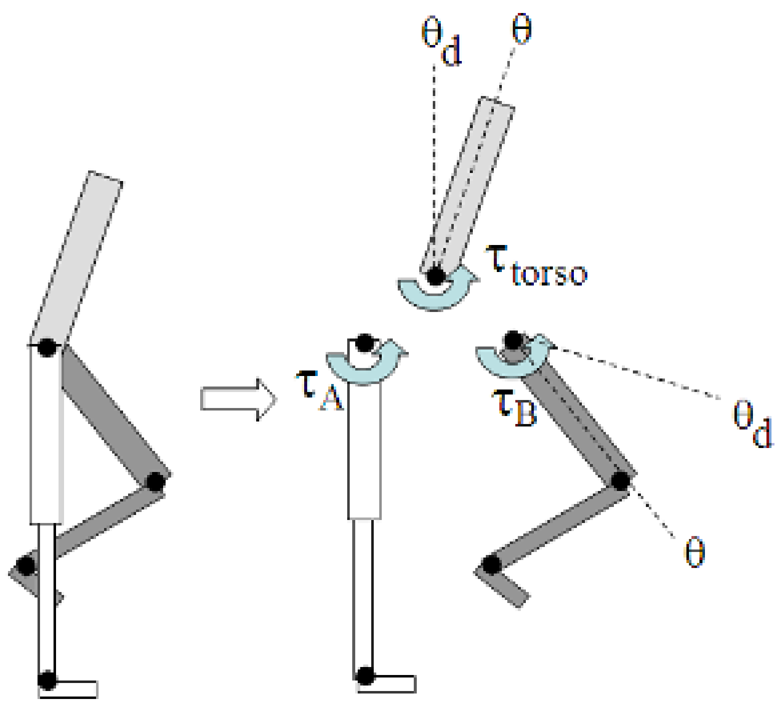


FIGURE 2.2: Controller based Locomotion Model (An abstract view)

in Figure 2.2, the balancing requirement can be stated as, the desired value of $\tau_{torso}$ equals the net torque seen by the torso by the legs, i.e. , $\tau_{torso} = -\tau_A - \tau_B$

Since the foot placement position is independent of the torso angle, the swing-hip is decoupled from the torso. This is achieved by controlling the swing-hip angle in world coordinate frame as opposed to other links which have target angles expressed with respect to their parent links.

### 2.3.3 Balance Feedback

The last component of the control strategy is to apply a balance feedback to the swing foot placement. A feedback from the position and location of the center of mass is used to determine the target swing-hip angle. If the center of mass is moved forward then swing hip angle should be set so that the character moves forward and vice versa.

# Chapter 3

# Quadruped Controller

## 3.1 Introduction

After bipeds another class of characters is quadrupeds. We follow the approach mentioned in [5]. This controller provides torques to the forward dynamics simulator at every time step. Computed torques arise from two resources: vitual internal forces and joint control. Torques from these two sources are summed together.

## 3.2 Gait Controller

The quadruped gait controller consists of following parts.

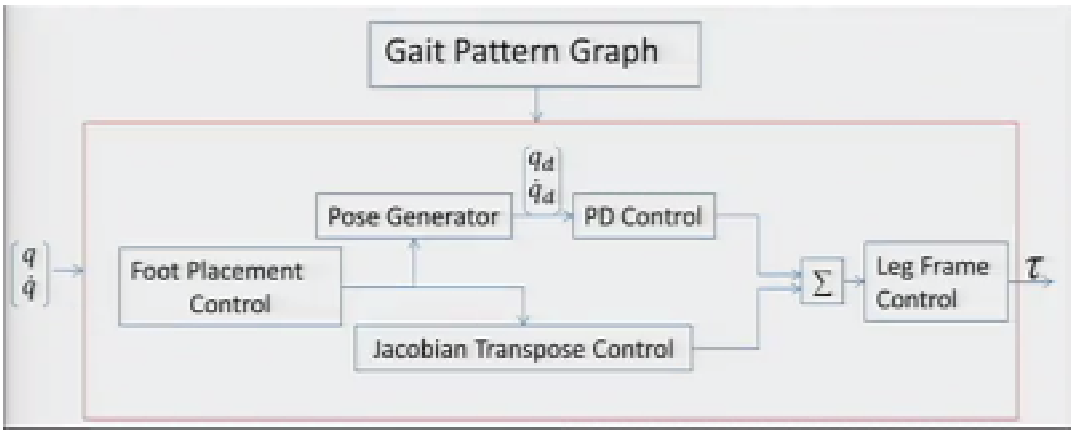


Figure 3.1: Gait Controller

### 3.2.1 Gait Graphs

Each leg of the quadruped can be in a stance or in a swing phase. The timings of these phases are specified by the means of a gait graph.

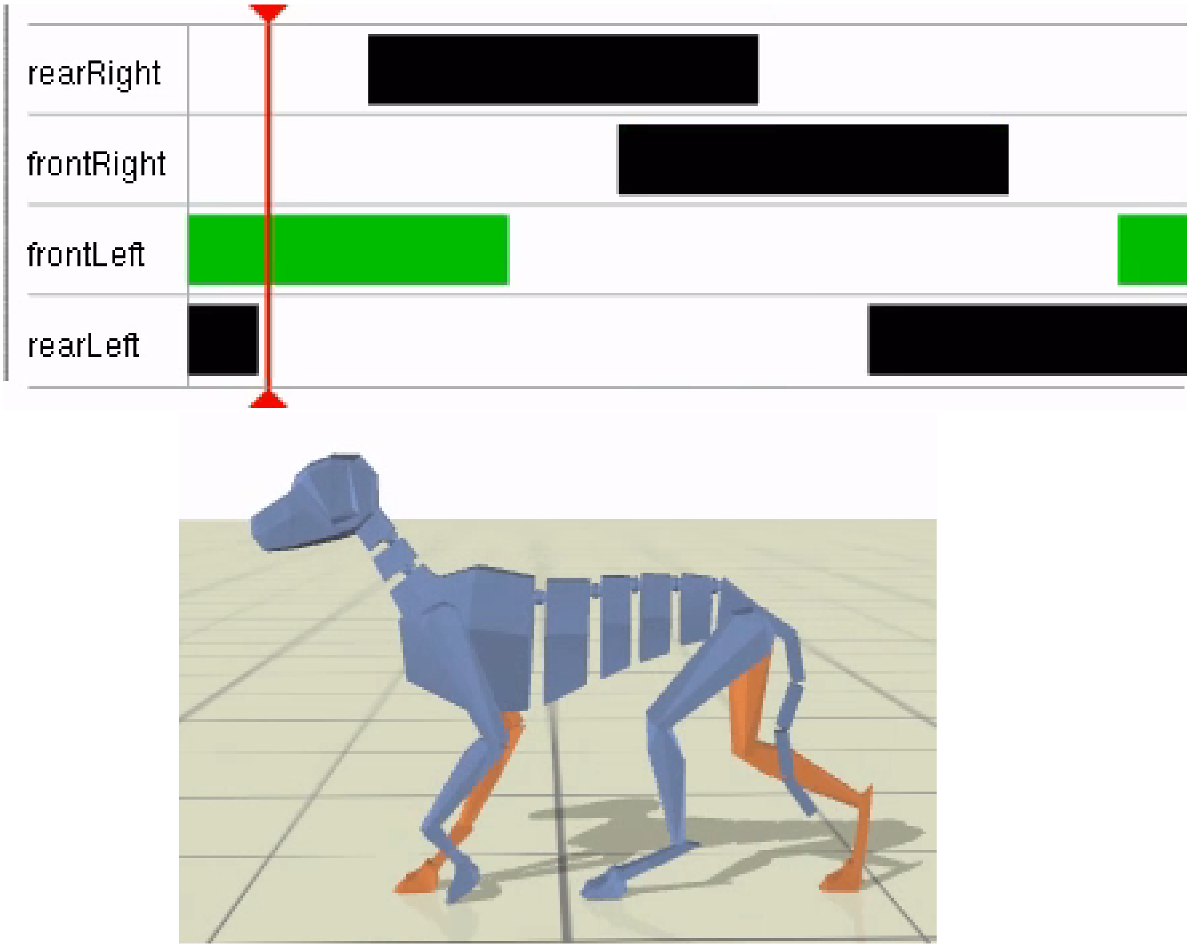


Figure 3.2: Gait Graph (The red line marks the current time)

### 3.2.2 Foot Placement Control

The foot placement control component calculates the target foot placement location for swing legs.

Suppose, $v$ and $v_d$ are the current and desired velocities respectively, then foot placement location is computed as,

$$P_2 = P_{LF} + (v - v_d)s \tag{3.1}$$

Here, $P_{LF}$ is the default stepping location with respect to the leg frame location and $(v - v_d)s$ is the feedback part with $s$ as the scaling factor.

$P_{LF}$ is computed as

$$P_{LF} = v_d * \sqrt{\frac{h}{g} + \frac{||v_d||^2}{4g^2}} \tag{3.2}$$

### 3.2.3 Pose Generator

After getting the target position of the foot, the orientation of joints is determined by the pose generator. Target angle joints are computed using inverse kinematics (for leg joints) or predefined trajectories (for head and neck) or interpolated leg-frame orientations (for spine) to get the target joint angles. These joint angles are used by PD controllers to generate first source of torques.

### 3.2.4 Virtual Forces

Virtual forces provide the second source of torques. Virtual forces are applied using Jacobian transpose to compute required internal torques according to $\tau = J^T F$. Virtual forces include gravity compensation at spine links, neck and head and tail, velocity tracking force, height tracking force and phase tracking force on leg frames.

### 3.2.5 Leg Frame Control

As was the case in the biped controller, the net torque on the leg frame should be zero. Therefore, torque at stance leg $\tau_{stance}$ is left free and it is computed as,

$$\tau_{stance} = \tau_{LF} - \tau_{swing} - \tau_{spine} \tag{3.3}$$

This helps in maintaining the balance of the quadruped.

The torques thus obtained are then added up and passed to the forward dynamics simulator which acts on the character and generates the next step of the motion.

# Chapter 4

# Implementation

## 4.1 Quadruped Controller

Skeleton model for a German Shepherd dog and a controller for the same model was implemented.

### 4.1.1 Character Modelling

A skeleton model (figure 4.1) for a female German Shepherd dog was created. It has a shoulder height of 47cm, a hip height of 45 cm and a total mass of 34 kg. The articulated figure consists of 30 links: 4 links for each leg, 6 for the back, 4 for the tail and 4 for the neck-and-head. The link lengths of the legs were computed from [6].

All links in the skeletons are modelled as cylinders of diameter 0.4 cm. The joints between the links are different according to the movement they should allow.

Open Dynamics Engine(ODE) is used for the forward dynamics simulation. Each link of the character is represented as an ODE body and a geometry associated to that body. This body has a specific mass. Density is assumed to be constant for all the links.

### 4.1.2 Controller

The controller is implemented as per the reference [3]. Major parts of the controller include swing leg treatment, stance leg treatment and inverse kinematics.

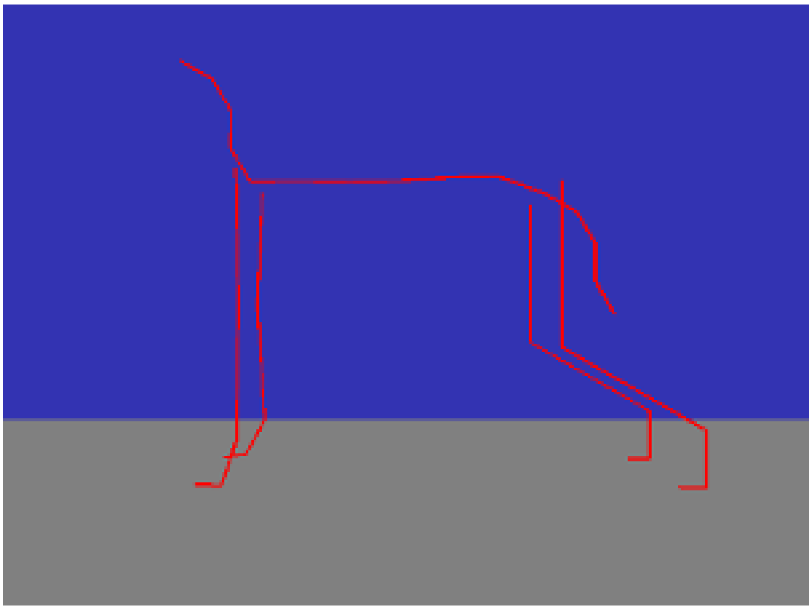

Figure 4.1: Skeleton Model

#### 4.1.2.1 Swing Leg Treatment

For each leg, in each time step, we check whether it is in swing phase or stance phase using the gait graph. Gait graph is stored as a set of 8 values: the start and end times of swing phase in a time cycle for each leg. When a leg is entering the swing phase the target foot placement position is calculated as specified in section 3.2.2. At an intermediate instance, the target foot position is calculated by interpolating between current foot placement position and target foot placement position.

The target position thus obtained is passed to the inverse kinematics model which gives the change angles at the links. These angles are fed to the PD controllers to get the torques to be applied on swing leg links.

#### 4.1.2.2 Inverse Kinematics

A general inverse kinematics class is implemented. It takes link lengths, their transformation matrices and the target end effector position and returns the joint angle changes with the help of damped least square inverse of the jacobian of the links.

Inverse kinematics is not applied to all four links in the legs. The trajectories of the swing foot and ankle links are modelled as a function of the swing phase and thus are not included in the inverse kinematics. The end-effector position is translated to the frame of the knee and then inverse kinematics is applied to get the angles at the leg frame and the knee.

#### 4.1.2.3 Stance Leg Treatment

For legs which are in stance phase, velocity tracking forces are applied. The leg frame control torque is computed (section 3.2.5) and applied to the stance leg frames.

# Chapter 5

# Future Work

The work can progress in following steps -

- The source code for SIMBICON biped controller is available. After perfecting the locomotion skills of quadrupeds, the next step would be to combine the biped and quadruped controllers into a common controller.
- Character morphing between bipeds and quadrupeds can be modelled and controller can be modified to provide physics for this transformation.
- Once we have a common biped and quadruped controller, we can further build and integrate controllers for other classes of characters, for example, birds [7].

# Bibliography


[1] M. de Lasa I. Mordatch and A. Hertzmann. Robust physics-based locomotion using low-dimensional planning. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH*, 29(71), July 2010. URL http://dl.acm.org/citation.cfm?id=1778808.

[2] K.B. Cheng J. Lee Y. Tsai, W. Lin and T. Lee. Real-time physics-based 3d biped character animation using an inverted pendulum model. *IEEE Transactions on Visualization and Computer Graphics*, 16:325 – 337, March-April 2010. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5161260&tag=1.

[3] K. Loken K. Yin and M. van de Panne. Simbicon: simple biped locomotion control. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH*, 26(105), July 2007. URL http://dl.acm.org/citation.cfm?id=1276509.

[4] S. Kim Y. Lee and J. Lee. Data-driven biped control. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH*, 29(129), July 2010. URL http://dl.acm.org/citation.cfm?id=1781155.

[5] B. Jones L. Reveret S. Coros, A. Karpathy and M. van de Panne. Locomotion skills for simulated quadrupeds. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH*, 30(59), July 2011. URL http://dl.acm.org/citation.cfm?id=1964954.

[6] German shepherd standards. . URL http://www.casabeggiato.eu/index.php?page=standard&lang=eng.

[7] .