Physics based Simulation for Locomotion of Biped Characters

MTP Stage II Report

Submitted in partial fulfillment of requirements for the degree of

Master of Technology

Computer Science and Engineering

By

Neha Shikha Kachhap
Roll Num: 173050080

Guided By

Prof. Parag Chaudhuri

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
Acknowledgement

I would like to express my gratitude towards Prof. Parag Chaudhari for his guidance and constant support all through the session. All the meetings held during the session were helpful and discussions made during the meetings helped me to keep progressing in my work.
Abstract

Realism in virtual environment is of utmost importance and animations portraying the natural world should be akin to reality. Simulation of these virtual environment with naturally occurring physical constraints pose a tough challenge to animate the virtual characters and objects which are a part these environments. There are many methods to do character animation like hand-drawn animation, key-framing, motion capture, physics-based simulations etc. Physics-based simulation offers a solution to create motion which abide the laws of nature, which are the product of muscles, gravity and other external forces acting on the skeleton of the body of the virtual characters. Character animation using Physics-based simulation have become one of most cited methods of computer animation over the past few years both for rigid and soft body simulations.

This approach simulates animated objects and computes their target motion/gait using physical simulation. It makes sure that object motions are obeying laws of physics, which leads to physically realistic motions. This method frees the animator from handling the low-level motion details, since this is taken care by physical simulation. This is very helpful where the animation involves motion of inanimate objects, the animator only needs to input the initial state of the object along with some control parameters and a physical simulator computes the required motion. In comparison to traditional key-frame animation, Physics-based simulation provides the capability to adapt to changes in the environment at runtime. Key frame animation is static and non-responsive and it requires careful measures otherwise it can produce unexpected results.

Physics-based simulation offers a subtle way to create motion as it occurs naturally, which is the product of muscles, gravity and other external forces being applied on the character or object. Motion capture also gives realistic motion but it has its own limitations. In motion capture we are limited to available data and it becomes difficult to animate fictional characters. Additionally using it we can not realize scenarios of interaction with environment. Using physics-based character animation we can simulate interaction of character with its environment. Additionally it proves to very helpful in animating fictional characters.

The important elements such as foot impacts and effective balance adjustments, naturally occur in the motion while working with physical simulations. But, however simulating motions in physical simulations is also difficult as we do not have control on the character as the motion are generated by applying forces and torques on the physics-based character. Also the character need to be balanced so that it would not fall over in the next step. So developing good balance strategy for locomotion is also a tough task and it has been a subject of research for a long time. Another challenge is to make the character’s locomotion robust against external pushes and forces.
# Contents

1 Introduction 6
   1.1 Physics-Based Simulation ................................................. 6
   1.2 Motivation ................................................................. 7
   1.3 Challenges ............................................................... 7

2 Literature Survey 8
   2.1 SIMBICON: Simple Biped Locomotion Control ......................... 8
      2.1.1 Balance Control Strategy ........................................ 8
   2.2 Generalized Biped Walking Control .................................... 10
      2.2.1 Control Framework ................................................ 10
   2.3 Optimizing Locomotion Controllers Using Biologically-Based Actuators and Objectives ........................................... 12
      2.3.1 Control Parameterization .......................................... 12
      2.3.2 Optimization .......................................................... 13
   2.4 Flexible Muscle-Based Locomotion for Bipedal Creatures ............ 13
   2.5 Comparison ............................................................... 14
      2.5.1 Control Strategy ..................................................... 14
      2.5.2 Character models .................................................... 15
      2.5.3 Optimization techniques ............................................ 15

3 Implementation 16
   3.1 2D Biped Locomotion .................................................... 16
      3.1.1 Controller transitions ............................................. 17
      3.1.2 Robustness ............................................................ 17
   3.2 3D Biped Locomotion .................................................... 18
      3.2.1 Manual Controller ................................................... 18
      3.2.2 Motion Capture Controller ......................................... 19
      3.2.3 A 3D Biped Controller using Inverted Pendulum Model ........ 20
      3.2.4 Parameterization ..................................................... 21

4 Conclusion 22
Chapter 1

Introduction

1.1 Physics-Based Simulation

Virtual character animation involves animating virtual characters as if they were moving in real. Character animation is used in various fields. One most used area is Gaming, where characters are required to interact in the simulated world of the game arena. It can also be used in medical science fields to depict various kinds of motions in various situations, to depict movements of human muscles and tissues etc which can help students to virtually see the scenario and understand better. Character animation can also be used for various other educational purposes for making the demonstration of acts, plays, sports, special arts etc.

The target trajectories for virtual characters were drawn through traditional hand drawing which involves drawing a sequence of characters movements in a regular interval of time. Key-frames are also similar to hand-drawn animation where character’s particular poses are drawn known as key-frames and animation between the key-frames is then interpolated. But these two methods require good animating skills and is also time-consuming. Another method is to capture motion of a live actor doing some actions and then reconstructing the motion through the captured data. This method is efficient but is costly and we are dependent on available motion capture data database.

Additionally, when there is interaction between characters in the virtual environment animation becomes tough as the characters can interact in many different ways and modeling every possibilities becomes difficult. Animations being driven by motion capture data and drawing also needs a substantial amount of tuning and adjustments.

Physics-based simulation offers a solution to create motion which abide the laws of nature, which are the product of muscles, gravity and other external forces acting on the skeleton of the body of the virtual characters. Character animation using Physics-based simulation is have become one of most cited methods of computer animation over the past few years for animations of cloth, water and rag-doll character simulations. The thesis work of [Geijtenbeek, 2013] illustrates various aspect of research on physics-based character animation.
1.2 Motivation

I got interested in this area because it provides realism in rag-doll character animation. Reconstruction of motion from captured data also results in smooth animation but it restricts us by the number of motions available. Additionally in physics based simulation character interactions are handled efficiently.

1.3 Challenges

To understand how physics-based characters are modelled and simulated we first need to understand how they are controlled and what difficulties are faced by animators while simulating them. Under-actuation in simulating physics-based characters is a big problem. As any living being, the physics-based characters are also not guided by externals forces, they move by forces and torques produced by actuators present in their joints. This makes very difficult to track the position and orientation and balance of the character.

Authoring motions of characters is also difficult, given that motions are an indirect end-product of what can be controlled, namely the forces and torques being applied to the joints of the character. Equally important, characters need to be balance itself if they are not to fall over. The development of good balance strategies has proven to be a daunting process and has been the subject of several decades of ongoing research in computer animation and robotics. Some approaches to simulate balanced and robust locomotion of physics-based character to unexpected perturbation and external pushes have used feedback strategies to provide balance to the character in [Yin et al., 2007].

The thesis work of [Geijtenbeek, 2013] illustrates various aspect of research on physics-based character animation. The work also draws attention to various challenges in animating physics-based characters, focuses on their modelling and control strategies. Another main problem of physics-based simulation makes the character’s behaviour virtually real and stable where in real world it would not be. It includes mainly character’s motion, orientation and position. To cope up with this problem [Yin et al., 2007] used motion-captured data of a live actor but that also limits our animation to available data of motions. The physics based characters also require additional parameter tuning which makes locomotion controllers somewhat specific to a particular character. So to eliminate this problem [Coros et al., 2010] developed control solutions for locomotion that provide robust control over balance, that allow naive artists to easily design new motions, that are fast to compute, and that generalize well in a variety of ways without requiring further tuning.
Chapter 2

Literature Survey

Throughout the session we went through research papers to understand various methods to employ physics-based character simulation. Following are the four different methods that employ different strategies.

2.1 SIMBICON: Simple Biped Locomotion Control

[Yin et al., 2007] provides a control system which can produce a variety of biped gaits like walking, running, jumping, skipping and hopping. It provides simple framework for biped control. The starting point is to use a simple finite state machine or pose control graph. Each state consists of a target pose representing target angles with respect to their parent links for all joints. All individual joints attempt to gain the target angles using proportional-derivative (PD) controllers. Transitions between states occur after fixed duration of time, or, for some states, after a new foot contact has been established. Walking gaits can be modeled using as few as four states, while running gaits can be modeled using as few as two states. The controllers can be applied on both 2D and 3D physically-based characters. [Yin et al., 2007] also illustrates robustness by illustrating withstanding external pushes and adaptation of the character to the unexpected perturbation on terrain. The paper also provide mechanisms to change from controller to another. Feedback-error learning is applied to reduce the high controller gains.

2.1.1 Balance Control Strategy

The control strategy of [Yin et al., 2007] can be described in terms of three sub-parts: a finite state machine, torso and swing-hip control, and balance feedback.

- **Finite state machine**
  
  At first finite state machine is designed for each controller. Here each state has a particular target pose. The number of states depicts the sequence of target poses, so complicated gaits will require more states. The figure 2.1a shows four states in a walking gait controller. At each state the torques are computed by proportional-derivative control,

\[
\tau = k_p(\theta_d - \theta) - k_d\dot{\theta} \tag{2.1}
\]

8
where $k_p$ and $k_d$ are controller gains.

(a) A pose control graph for walking gait

(b) Elements of the balance control strategy: (i) Relationship between torso, stance-hip, and swing-hip torques; (ii) Center-of-mass position and velocity

Figure 2.1: Balance control strategy of [Yin et al., 2007]

- **Torso and Swing-hip Control**
  The torso and the swing hip have target angles which are not calculated with respect to their parents but to the *world frame* as shown in 2.1b. There is no target angle for stance hip so the torques applied on swing hip $\tau_B$, torso $\tau_{torso}$ and the stance hip $\tau_A$ are calculated as

$$\tau_{torso} = -\tau_A - \tau_B$$  \hspace{1cm} (2.2)

- **Balance Feedback**
  The last element of the control strategy is to apply balance feedback for swing-foot placement. So [Yin et al., 2007] employs a feedback form

$$\theta_d = \theta_{d0} + c_d d + c_v v$$ \hspace{1cm} (2.3)

to swing hip, where $\theta_d$ is the target angle fed to PD controller, $\theta_{d0}$ is the default target angle decided by the pose control graph, $d$ is the horizontal distance between centre of mass (COM) of the biped character and stance foot and $v$ is the COM velocity. This feedback policy is implemented in both 2D and 3D bipeds.
2.2 Generalized Biped Walking Control

[Coros et al., 2010] gives a control strategy for physics simulated walking motions that can generalize across various gait parameters, motion styles, character proportions and a variety of other skills like pulling or pushing objects and crossing over objects.

2.2.1 Control Framework

The proposed control strategy consists of four key components as shown in 2.2 are:

1. A motion generator provides target trajectories that are tracked by proportional-derivative (PD) controllers.
2. A gravity compensation model which adds computed torques to the control, which allows for low-gain tracking.
3. To balance the character, foot placement is done with the help of an inverted pendulum model.
4. Continuous balance adjustments are produced by using Jacobian transpose control to mimic the effect of a virtual desired force.

- **Motion Generator**
  The motion generator provides desired target trajectories for the joints of the biped to produce various motion styles. The target trajectories are directly modelled either according to their parent joints or according to the character coordinate frame. The character coordinate frame is comprised of the up vector of the world and the facing direction of the character. The target torques are then produced by proportional derivative controller.

![Figure 2.2: Control framework of [Coros et al., 2010]](image-url)
• Inverted Pendulum Foot Placement

To compute the desired stepping point \( x_d, z_d \) [Coros et al., 2010] uses an inverted pendulum model (IPM) as shown in Figure 2.3. The sum of the potential and kinetic energy of IPM at the current state to its future point of support being in balanced rest state, using its current velocity \( v \) and its current height \( h \) and distance, \( d \) is

\[
\frac{1}{2}mv^2 + mgh = \frac{1}{2}mv'^2 + mgh' \tag{2.4}
\]

where \( v' = 0 \) and \( h' = L = \sqrt{h^2 + d^2} \). So using equation 2.4 \( d \) is calculated as

\[
d = v \sqrt{\frac{h}{g} + \frac{v^2}{4g^2}}. \]

So IPM is applied in sagittal plane to evaluate \( x_d = d \) and is applied in coronal plane to calculate the value of \( z_d \).

\[\text{(a) Inverted pendulum model}\]

\[\text{(b) Jacobians used for velocity tuning (left) and gravity compensation (right)}\]

Figure 2.3: IPM, velocity and gravity compensation models of [Coros et al., 2010]

• Velocity Tuning

[Coros et al., 2010] applies velocity tuning by applying a virtual force, \( F_V \) on the center of mass in order to accelerate or decelerate it towards the desired velocity. The virtual force acting on both sagittal and coronal planes are computed by using jacobian transpose control \( \tau_V = J_V^T F_V \). Here \( J_V \) is a jacobian of centre of mass for chain of body links between the stance foot and the head of the biped.

• Gravity Compensation

This component of the control strategy allows for low gain PD tracking for limbs and the body of the biped. Gravity compensation is applied using Jacobian transpose method. A virtual force \( F_i = -m_i g \) is applied to the centre of mass (in upwards direction) of every link between link \( i \) and the root of the biped. Hence gravity compensation torques are computed for all joints that lie between link \( i \) and the root link.
2.3 Optimizing Locomotion Controllers Using Biologically-Based Actuators and Objectives

[Wang et al., 2012] presents a motion controller where the skeleton of the body is driven to its target poses using 3D muscle models. The sagittal hip, knee and ankle joints are actuated using a set of eight Hill-type musculotendon models in each leg (Figure ??). In Hill-type model, each musculotendon unit consists of three elements, a contractile element \((CE)\) which helps in generating force \((F_{CE})\), a parallel elastic element and a serial elastic element which represents tendons that connect the muscle to the bones.

The joint torques generated by a given MTU is a function of the current body configuration shown in Figure 2.4a. The MTU models are applied for the lower body joints and the torques for rest of the degree of freedoms is calculated via pose control graphs as in [Yin et al., 2007].

![Diagram](image.png)

(a) Relation between musculotendon model, controller and simulator. (b) a) Humanoid model, 16 hill-type MTUs shown in red, b) 5 Uni-articular MTUs and c) 3 Bi-articular MTUs.

Figure 2.4: Image courtesy [Wang et al., 2012]

2.3.1 Control Parameterization

There are two different control laws which are applied on the muscles depending on the whether the leg is in stance phase or swing phase. The controls states referred by the system are transition when the signed distance between stance ankle and COM of the body satisfies certain threshold conditions as shown in Figure 2.5. Depending on the features of the body there are different mapping of the controls laws. These mapping serve as building blocks of the control law.
2.3.2 Optimization

This paper optimizes control parameters and initial state parameters using Covariance Matrix Adaptation. The optimization aims to move the character forward without falling down. The paper employs an objective based on minimization of metabolic energy expenditure, which helps in choosing the most effortless gait that achieves the task. The task includes moving COM of the character forward at a target velocity while maintaining its balance, maintaining its head stability and upper body orientation.

2.4 Flexible Muscle-Based Locomotion for Bipedal Creatures

In [Geijtenbeek et al., 2013] all actuation torques are generated by 3D simulated muscles. These 3D muscles are of Hill-type model which has three consequent elements. Using Hill-type muscle model, actuation forces are produced by contraction of muscle fibres. The muscle fibres contract according to the actuation levels of the current state of the body. These 3D muscle models helped them to model biped characters of different shapes and features (Figure 2.6).

The skeleton of a character and its muscle has a two-way interaction, muscles apply forces that change the pose of the skeleton, while skeleton pose fully determine the length of the muscle fibre which then influences the contraction of the muscles fibres. The muscle routing on the skeleton is defined by a set of a vector, comprising of points of attachment of the muscle on to the body. The paper also optimizes the attachment points of the muscles in further generation.
The length of the muscle fibre and the actuation state of the character are used to generate scalar muscle force for a particular muscle. These muscle forces are transmitted to skeleton at the attachment points to the body the muscle is attached to and generates a torque over the joints it spans.

![Figure 2.6: Physics-based simulation of locomotion for creatures of different shapes driven by 3D muscle-based control [Geijtenbeek et al., 2013]](image)

[Geijtenbeek et al., 2013] uses Covariance Matrix Adaptation to optimize the total set of parameters used muscle physiology, muscle geometry, state transition, target features, feedback control and initial character state. This paper focuses more on optimization of muscle routing that enables and simplifies the design of the approach.

### 2.5 Comparison

#### 2.5.1 Control Strategy

[Coros et al., 2010] is similar to [Yin et al., 2007] in terms of generating target trajectories and torques using PD controllers. It differs from [Yin et al., 2007] in foot placement strategy where, [Coros et al., 2010] uses predictive inverted pendulum model to calculate the desired stepping point on the other hand [Yin et al., 2007] uses a additional feedback term in the target angle of the swing hip using the horizontal distance of the centre of mass of the character from the position of the stance foot. Additionally [Coros et al., 2010] also does velocity tuning and apply gravity compensation torques using Jacobian Transpose method.

Though [Geijtenbeek et al., 2013] and [Wang et al., 2012] seem to be similar approaches because they simulate the motions of the character using 3D simulated muscles models are different from each other in many aspects. The very first difference is [Wang et al., 2012] only uses 3D muscle models on sagittal hip, knee and ankle joints to actuate them and the rest of the degree of freedom target angles are tracked by pose control graphs as in [Yin et al., 2007]. Where as [Geijtenbeek et al., 2013] used 3D muscle models to drive the motion of the entire body.
2.5.2 Character models

Both [Coros et al., 2010] and [Yin et al., 2007] model character having rigid bodies and attached via joints. These both approaches uses target trajectories to drive towards the target pose and the torques are directly applied to the joints. The target trajectories are referenced either by a set of data or motion captured data.

In [Geijtenbeek et al., 2013] and [Wang et al., 2012] we see that they employ 3D muscle model using Hill-type musculotendon units. The target forces generated are in accordance to the contraction of the muscle at a particular activation state.

2.5.3 Optimization techniques

[Yin et al., 2007] requires a little parameter tuning to generated desired motion style and do not generalize to variety of characters. Where as [Coros et al., 2010] aims to simulate walking motions that can generalize across gaits parameters, motion styles, character proportions and a variety of skills.

[Wang et al., 2012] optimizes the parameters of the control laws so that the simulation produces a motion of biological model with minimum metabolic energy. [Geijtenbeek et al., 2013] also perform optimization to get efficient muscle routing over the skeleton of the body.
Chapter 3

Implementation

I have implemented motion controllers for planar and 3D biped characters based on the approach of [Yin et al., 2007] and [Coros et al., 2010] using Open Dynamics Engine ([Smith et al., 2006]). In my implementation I referred the open source code provided on the web page of [Yin et al., 2007] and [Coros et al., 2010] and Dart toolkit ([Lee et al., 2018]).

3.1 2D Biped Locomotion

The character model is same as used in [Yin et al., 2007]. The 2D character model has 7 links consisting of torso, thighs, shin and toes. The model has 6 internal degree of freedom and 9 degree of freedom in total. Proportional Derivative gain values of $k_p = 300 \text{ Nm/rad}, k_d = 30 \text{ Nms/rad}$ are used for all respective joints. Joint limits are enforced on the hips and knees. A time step of 0.004 s is used.

I made two controllers each for walking and running of the 2D biped character. The controllers refer the pose-control graphs for target poses as in [Yin et al., 2007]. The control laws are same for all states of the pose-control graph. The number of states in control-pose graph depends on the number of leg changes. The target angles of joints are tracked via proportional-derivative controllers. The controllers also require additional parameter tuning to find appropriate proportional-derivative controller gains and the duration of states. Balance feedback as mentioned [Yin et al., 2007] are added on the target angles for the swing hip for foot placement.
3.1.1 Controller transitions

We can easily switch between the controllers at any instance of time in the simulation. This is accomplished during state transitions, by jumping from state $n$ of the current controller to state $n + 1$ of the desired controller. Before the start of the simulation, the character is by default in the walking state and it starts to walk as soon as the simulation is started. There are key controls to start the simulation and to switch between the controllers.

3.1.2 Robustness

The motion of the character is stable even left for hours. It is also robust to external pushes applied to its root which is its torso.
3.2 3D Biped Locomotion

The 3D character model is the same as used in [Laszlo et al., 1996]. The model has 28 internal degree of freedom and 34 degree of freedom in total. The simulation time step used is 0.005 sec.

3.2.1 Manual Controller

I made two controllers each for walking of 3D biped character. The controllers refer the pose-control graphs for the next target poses. The target angles of joints are tracked via proportional-derivative controllers. The controllers also require additional parameter tuning to find appropriate proportional-derivative controller gains and the duration of states. In 3D model the balance feedback term is also applied to the coronal plane as the 3D biped translates in both planes i.e. sagittal and coronal planes. Without this coronal balance strategy the character rolls over on the other side whenever the biped raises its swing hip to move ahead. Torque magnitudes are limited to $k_p$. The largest $k_p$ value is 1000, and is used for the waist joints. All other joints use $k_p = 300$ or less and $k_d = 0.1 \times k_p$. I can also take turns to its right and left.

![Figure 3.3: Walk](image)
3.2.2 Motion Capture Controller

In this controller, we use motion capture data as a base for developing the controller as described in [Yin et al., 2007]. In this we track the reference target trajectories and calculate equivalent torques to mimic the data. Balance strategy is the same as used in the manual controller. Torque magnitudes are limited to $k_p$. The largest $k_p$ value is 1000, and is used for the waist joints. All other joints use $k_p = 300$ or less and $k_d = 0.1 k_p$. I have made a walking controller which has a pose-control graph having only two states. The transitions between states occur only on foot contact. I have maintained a phase $\phi \in [0, 1]$, which helps in transition between the states. I developed the controller for walking only but it can be used to make controllers for various other gaits.
3.2.3 A 3D Biped Controller using Inverted Pendulum Model

In this approach I implemented a control strategy stated by [Coros et al., 2010] for physically-simulated walking motions. Now I have implemented walking controller only but this controller can be used for running also. The control strategy mainly consists of four key components as shown in 2.2. So the very first component is the motion generator which generates the desired joint angle trajectories. The target joint angle trajectories can be modeled relative to their parent links or to the character coordinate frame which is the root of the character.

Use of foot placement is used as a balance strategy. Inverted Pendulum model is used to compute the desired location of the swing foot. This does not require any additional parameter tuning. Along with proportional derivative control we also used gravity and velocity error compensation torques. The walking controller also uses a simple virtual velocity tuning force similar to that stated in [Pratt et al., 2001] to provide some control over the centre of mass velocity. The controller also uses gravity compensation torques, which enables lower gain in proportional derivative control for the limbs and the body of the character model.

All balance control parameters are kept fixed and the controller requires no parameter tuning. The Proportional derivative gains values used are $k_p$ and $k_d$ where $k_d = 2 \sqrt{k_p}$.

Figure 3.6: Walking pattern including collision of the legs
3.2.4 Parameterization

The robust nature of the controllers means that either manually or from motion capture data additional control strategies can be modelled over it. More sophisticated controllers can be created to control walking styles or walking directions. Turning is generated by changing the heading direction of the root of the character in both the manual and mocap controller. Since the balance controller only uses the lower limbs of the character model, the upper body is left free for different styles or additional tasks. For example, we can choose to keep the arms straight pointing downwards or let them swing naturally.

In the balance strategy as stated in subsection 3.2.3 the parameter settings are fixed across all motions. A default trajectory needs to be provided as a reference for the height of the swing ankle as a function of phase and the legs should be prohibited to collide among themselves.
Chapter 4

Conclusion

The controllers mentioned in subsections 3.2.1 and 3.2.2 are not fully automated and still require manual hand tuning for proportional derivative gains. The controllers also do not model the reaction time delays of human motion. As a result, some of the motion are unrealistically stable. Additionally, in mocap controller motions that do not contain periodic stepping motions will be a problem. In the controller mentioned in subsection 3.2.3 we can see collision of swing leg with the stance leg and the character gets unstable after taking a few steps.

The development of controllers which can control the locomotion of biped is a challenging problem. The needs in animations such as modeling multiple gait styles, stylized motions, motion in different terrains and being able to react to external forces and pushes in the virtual environment also adds difficulties to this challenge.
Chapter 5

References


