Physics-Based Human Motion Modeling for People Tracking: A Short Tutorial

Marcus A. Brubaker[1] **Leonid Sigal**[1,2] **David J. Fleet**[1]

[1] Department of Computer Science, University of Toronto, Toronto, ON, M5S 3H5
{mbrubake,ls,fleet}@cs.toronto.edu

[2] Disney Research, Pittsburgh, PA, 15213
lsigal@disneyresearch.com

September 27, 2009

Abstract

Physics-based models have proved to be effective in modeling how people move and interact with the environment. Such dynamical models are prevalent in computer graphics and robotics, where they allow physically plausible animation and/or simulation of humanoid motion. Similar models have also proved useful in biomechanics, allowing clinically meaningful analysis of human motion in terms of muscle and ground reaction forces.

In computer vision the use of such models (*e.g.*, as priors for video-based human pose tracking) has been limited. Most prior models in vision, to date, take the form of kinematic priors that can effectively be learned from motion capture data, but are inherently unable to explicitly account for physical plausibility of recovered motions (*e.g.*, consistency with gravity, ground interactions, inertia, *etc.*). As a result many current methods suffer from visually unpleasant artifacts, (*e.g.*, out of plane rotations, foot skate, *etc.*), especially when one is limited to monocular observations.

Recently, physics-based prior models have been successfully illustrated to address some of these issues. We posit that physics-based prior models are among the next important steps in developing more robust methods to track human motion over time. That said, the models involved are conceptually challenging and carry a high overhead for those unfamiliar with Newtonian mechanics; furthermore good references that address practical issues of importance (particularly as they apply to vision problems) are scarce.

This document will cover the motivation for the use of physics-based models for tracking of articulated objects (*e.g.*, people), as well as the formalism required for someone unfamiliar with these models to easily get started. This document is part of the larger set of materials: slides, notes, and Matlab code, that will allow a capable novice to proceed along this innovative research path.

1 Introduction and Motivation

Tracking human pose from video has been a focal point of vision research for several (10–15) years. Our interest in the problem stems from the abundance of significant applications that such a technology would facilitate. Obvious applications include markerless motion capture for animation and entertainment, smart human-computer interfaces, interactive toys, and of course surveillance. Indeed, humans spend a large fraction of time looking at other people, and arguably future computer vision systems will as well. Beyond the obvious applications, the tracking and analysis of human pose and motion, if sufficiently accurate, should also facilitate several applications in health care, ranging from the analysis of gait pathology and injury rehabilitation to elderly monitoring and assisted living.

Despite hundreds of papers on the subject, the general problem of tracking human motion in unconstrained environments from monocular observations is still largely unsolved. The challenges stem from: (1) the variability in human appearance (due to the variations in body shape and clothing), (2) the complexity of human motion and our interactions with other objects, (3) imaging ambiguities that result from image projection, (4) occlusions that arise from articulations of the body itself (self-occlusion) or other objects in the scene, and (5) the high-dimensionality of the state space that is required to represent the large number of joint degrees of freedom (DOF) in the human body.

To cope with ambiguities and uncertainties in pose inference, one prevailing approach has been to cast the problem in a generative probabilistic framework where prior models can be used to resolve ambiguities. Discriminative methods, based on learned regression models, mapping image features to 3D pose, have also been effective. In both cases, 3D motion capture data (mocap) has been the predominant source of prior information for learning *kinematic* models of human pose and motion. Mocap data represents the 3D pose of the body in terms of joint angles and/or 3D joint positions as a function of time. Several moderately sized mocap datasets have been constructed in recent years, of people performing a wide variety of different activities.

While the use of mocap data has led to impressive results in the last 5 years, it is also clear that such models do not ensure that the recovered motion estimates are physically plausible. The most common problems with kinematic models that are readily observed in 3D pose tracking are out-of-plane rotations (*i.e.*, ignoring gravity), ground plane penetration, and foot-skate, where feet appear to slide along or just above the ground. In an attempt to build models that ensure physically plausible motions and poses, and which generalize well to different scene conditions (*e.g.*, walking up hills or carrying a heavy load), it is natural to consider physics-based models and torque-based parameterizations of human motion.

The remainder of Section 1 develops this motivation and lays the groundwork for the tutorial in greater depth. In subsequent sections we then review classical mechanics, constrained mechanics for articulated bodies, biomechanics and human locomotion, and controller-based models.

1.1 Pose Tracking as Bayesian Filtering

Pose tracking is usually formulated in probabilistic terms. Pose is represented as a state vector, and motion is represented by state sequences. Let $\mathbf{s}_{1:T}$ denote the state sequence from time 1 to time T, where $\mathbf{s}_t$ is the state at time t. Image (observations) sequences are denoted by $\mathbf{z}_{1:T}$, where $\mathbf{z}_t$ is the observation at time t. Bayesian filtering concerns state inference, that is, the computation of the posterior distribution

$$p(\mathbf{s}_{1:T}|\mathbf{z}_{1:T}) = \frac{p(\mathbf{z}_{1:T}|\mathbf{s}_{1:T})p(\mathbf{s}_{1:T})}{p(\mathbf{z}_{1:T})}, \tag{1}$$

where $p(\mathbf{s}_{1:T})$ is our belief about how plausible a given motion is *a priori* and $p(\mathbf{z}_{1:T}|\mathbf{s}_{1:T})$ denotes the likelihood, which measures consistency between the image observations and a given motion. The denominator does not depend on the motion, and is therefore treated as a constant.

For long pose sequences $\mathbf{s}_{1:T}$ is high-dimensional, so inference is challenging. Nevertheless, with the conventional assumptions that the motion is Markov, and that observations are independent when conditioned on the states, the posterior can be expressed recursively:

$$p(\mathbf{s}_{1:t}|\mathbf{z}_{1:t}) \propto p(\mathbf{z}_t|\mathbf{s}_t)p(\mathbf{s}_t|\mathbf{s}_{1:t-1})p(\mathbf{s}_{1:t-1}|\mathbf{z}_{1:t-1}) \ . \tag{2}$$

Here, $p(\mathbf{z}_t|\mathbf{s}_t)$ is the likelihood of the observation at time t, $p(\mathbf{s}_t|\mathbf{s}_{1:t-1})$ captures our prior beliefs about the human motion, and $p(\mathbf{s}_{1:t-1}|\mathbf{z}_{1:t-1})$ is the posterior from previous time.

A good pose tracking algorithm depends on a good likelihood function, which provides an effective use of image evidence, and a good motion model, which assigns higher probabilities to more likely motions. An effective inference method is also critical. Monte Carlo methods are often used to approximate the posterior [9]. Direct gradient-based optimization has also been used effectively [89]. Finally, one also requires some method for initialization and recovery from tracking failures (often with discriminative methods) [80].

While all these components are critical in practice (see [9] for more details on human pose tracking), this document focuses on motion models (*e.g.*, $p(\mathbf{s}_t|\mathbf{s}_{1:t-1})$ in Eq. 2, or $p(\mathbf{s}_{1:T})$ in Eq. 1), and their properties in the context of video-based people tracking and dynamic scene analysis.

1.2 Kinematic Models

Almost all successful approaches to monocular human pose tracking in the last 10 years have relied heavily on motion capture data (mocap). This includes the use of mocap in generative probabilistic models of pose and motion, and the use of mocap in learning discriminative pose estimators. The literature on such models is sufficiently vast that we cannot review all such methods here, but we will outline the major classes of models in use (biased in part by our own research experience in these areas).

Generative Models: Given the high-dimensional nature of human pose (*i.e.*, the large number of joint angles that we wish to estimate, and the relatively small size of mocap datasets, and the nonlinearities of human motion, simple approaches to learning time-series models, such as linear dynamics systems (LDS) are usually unsuitable. Switching LDS models [64, 65] offer one way to handle nonlinearities, especially those that are short-lived (*e.g.*, due to contact of the foot with the ground), but these require large amounts of data and remain difficult to learn. Closely related to SLDS are the class of coordinated mixtures of factor analyzers as formulated by Li and her colleagues [44, 45].

Most recent approaches to learning prior motion models have involved some form of low-dimensional latent variable model. Most human motions, while complex, live on or near some low-dimensional manifold in the space of joint angles. Finding a parameterization for that manifold, through nonlinear dimensionality reduction or latent variable models (*e.g.*, [24, 81]), has therefore been the focus of recent research, often for learning models of specific activities. Gaussian Process Latent Variable (GPLVM) models [42, 89] are an example. GPLVMs have gained popularity due to their ability to learn compact latent spaces (due to their non-linear nature) from relatively few examples. There are a number of GPLVM-based models that fall into this category, including the Gaussian process dynamical model [97, 90], multifactor models for capturing stylistic diversity and different gaits [96], and hierarchical models that model the different parts of the body and their

coordination hierarchically [43, 18]. Other recent models of interest include the use of restricted Boltzmann machines [86], and the use of coordinated Mixtures of factor analyzers [44, 45], and the Kernel Information Embedding [78].

Learning kinematic models to help constrain pose tracking remains an active area of research. While current methods have demonstrated success with a small number of activities or different styles, a large number of open questions remain. Primarily, how these approaches will scale to much larger datasets, how to learn transitions between activities, how compositional models can be learned, and perhaps most importantly, how such models can be designed to generalize well beyond the training data.

Discriminative Models: To learn discriminative models mocap data is typically used to synthesize a database of *images* of people in the same pose but with various types of clothing and/or background [73, 80]. Supervised learning methods are then applied to learn a mapping from images to poses (or more generally the conditional distribution over pose or motion given image features). Some of the tested non-parametric approaches include k-Nearest Neighbors, kernel regression [73], and Gaussian process regression [88]. Parametric methods include linear and relevance vector regression [1], and mixtures of experts [80, 77].

The key challenges with discriminative methods include: (1) the need for mappings between high-dimensional spaces, (2) images are often consistent with multiple poses (*i.e.*, the conditional pose distribution is often multi-modal), (3) training datasets must be enormous, and (4) that efficiently training models from such large datasets (even if they exist) is very challenging. In particular, the set of training images should span the plausible pose space for people, along with the plausible space of clothing, lighting conditions, and camera viewpoint. Since this is impractical, generalization to new poses, different camera angles, or different clothing is a major issue. In practice, generalization beyond the local of training exemplars has been elusive.

1.3 The need for something more – Physics

Generalization is a major issue with current generative and discriminative models. often such models are limited to a range of activities or specific styles of motion. When the input is close to the training data they can perform very well, but when the input deviates from the training data generalization is poor.

Indeed the amount of mocap data required to span human motion may be exceedingly large. This is especially so when one considers interactions with the environment. Properties such as the slope or friction of the ground, the need to step over an object or across uneven ground, or distinctiveness of motion that results from one carrying a heavy object, are typically not modeled. Nor are important physical constraints on ones motion, such as balance with respect to ground slope and gravity, the fact that feet should not penetrate the ground, *etc.*. Such properties are not captured by current kinematic models because interactions with, and dependencies on, the environment are not part of the mocap paradigm. Of course we could aim to gather much more training samples, taking these variations and dependencies into account. However, the amount of data one would need to capture would be extremely, if not impossibly, large. Even if one is able gather all this data, it is unclear if current learning algorithms could cope with such large datasets.

The fact that environmental interactions and other properties of human motion are not well modeled is evident in current trackers that rely on kinematic models. They don't generalize and the don't ensure that the estimated motion is physically plausible. Problems include ground penetration, foot-skate, out of plane rotation, unrealistic accelerations, *etc.*. The fact that current

kinematic-based models do not capture such key physical properties is clearly evident in most modern tracking results. It is not uncommon to see 3D pose tracking results in which the model floats in the air, feet slide along the ground, with impossible to maintain balance, *etc.*.

An alternative to gathering vast amounts of mocap data, taking into account environmental dependencies *etc.*, is to use physics-based models. The basic idea behind the use of physics-based models is that human motion is governed in part by physical principles, which generalize quite naturally (compared to kinematic models). For example, physics-based models should be able to capture: (1) balance, *i.e.*, the way the person leans when he/she walk faster or slower, or the body lean that arises naturally as one walks up or down a hill; (2) sudden (and often visually unobserved) accelerations that are due to the contact of the foot with the ground; (3) changes in the gait style that arise as a function of the mass distribution, *e.g.*, when one carries something heavy or when a person is heavy or light.

It is hoped that physics-based models will therefore be able to generalize to a wide range of human motions without requiring massive amounts of motion capture data. Further physics-based models should ensure that estimated poses and motions are physically plausible, with feet penetrating the floor, or floating over the ground plane, *etc.*. Finally, it is exciting that interactions and contact with other objects in the scene are intrinsic to physics-based models. Accordingly a third potential advantage of physics-based models is that they provide a natural way to begin formulating the inference of people and their dynamic interactions with other objects. Indeed, one would hope to incorporate the inference of interactions into pose trackers.

1.4 Background and Related Work

Physics-based models have been studied and used accessibly in a variety of domains, many of which offer inspiration, ideas, and foundational knowledge for potential applications in vision research. In the context of human or human-like motion the most closely related domains include computer animation, robotics, and biomechanics.

1.4.1 Computer Graphics

Both physics-based models and data-driven kinematic models are common in compute animation. Kinematic models have been very successful in produces realistic human motions (*e.g.*, [37]). Yet the applicability of those approaches are typically limited to direct animation, without much generalization beyond the mocap. Nevertheless, not surprisingly, it is often the case that one would like to adapt the mocap to a new character [66] with different physical properties (*e.g.*, a character that may not exist in reality and thus cannot be captured directly), or to a new environment with different physical properties (*e.g.*, walk on the moon, where gravity is 5/6 of that on Earth).

For this purpose, and for synthesizing complex realistic motions based on a few keyframes specified by an animator, physics-based methods have been very popular. The two main approaches that have been widely studied are the use of space-time animation, and the develop of motion controllers. (*e.g.*, [48]). Physics-based models are particularly interesting in the context of interactive domains (*e.g.*, computer games), where it is desirable for the character to react in a plausible fashion to the applied external forces [16, 51, 105] (*e.g.*, a boxing game in which the motion of the boxer should depend on the magnitude and location of the hit by the opponent).

1.4.2 Robotics

Physical simulation also plays a critical role in robotics. Of particular interest is humanoid robotics where the goal is often to *animate* a bipedal robot to perform (or imitate) a human motion [61]. In doing so it is necessary to model the interactions between the robot and the environment and to produce strategies for actuating the appropriate motors at the joints (that could be thought of as muscles) to achieve a particular motion or task.

While in principle humanoid robotics is interested in modeling and replicating human motion [38, 39, 53], the fact that mechanically the structure of the robot is different from that of the human makes this task challenging. One practical issue is stability. Because humanoid robots tend to be expensive and fragile in robotics one often must ensure that the motion of the robot is stable (*i.e.*, balance is maintained at all times). As a result of this the typical motions of humanoid robots (*e.g.*, Honda's Asimo) tend to look unrealistic and use extraneous amounts of energy (are seriously overpowered). Humans on the other hand, often resort to much riskier but efficient motion paradigms in order to conserve energy and achieve efficiency (by for example using gravity and momentum to achieve, at least in part, the desired motion, without any use of muscles). That said, tools built for simulating articulated motion in robotics are very useful and often serve to motivate and underline the models in computer graphics and vision [103]. In addition, many properties of control and actuation can also be borrowed from robotics.

1.4.3 Biomechanics

Biomechanics studies the principles of human motion and locomotion, kinematics and kinetics. The complexity of biomechanical models depends greatly on the goals of the research. Models that attempt to model complex muscle interactions (down to the level of individual fibers) have also been explored. Among the models pushing the limits of physical simulation and realism are muscula-skeletal models [50, 104] that attempt to model the complex antagonistic muscle interactions within the body (as opposed to simpler angular *motor* models often employed in robotics and vision). Musculo-skeletal models can model complex behaviors such as involuntary response of limbs to neural stimuli (*e.g.*, knee jerk reflex that results from a doctor hitting a patient bellow the knee cap). While realistic, controlling or actuating such models that contain hundreds of parameters is very challenging. In particular, since the muscle activity cannot be observed or measured from kinematics (or even dynamics) such models resort to other modalities of perception (*e.g.*, electromyography[1] (EMG)).

Of particular interest in biomechanics is the growing interest in simplified abstract models of biological locomotion. Many of these models are inspired in part by passive-dynamic nature of human locomotion [55, 56, 57, 15] With such simplified models there are a number of very useful papers that address the fundamental principles that govern human motion (*e.g.*, energy consumption, muscle actuation (*e.g.*, [41, 40, 63]).

Finally, it is also notable that biomechanics plays an important role in sports, where the goal is to model and analyze the performance of the athletes in order to modify their technique to achieve optimal performance (*e.g.*, in gymnastics [75], diving, baseball, *etc.*). Unfortunately, while important within the sport activities in question, the resulting biomechanical models are not readily applicable to more general scenarios that prevail in every day life.

[1] Electromyography is a technique for recording the activation signal of muscles

1.5 Physics-based models in vision

Unlike in computer graphics and robotics where the use of physics-based models is commonplace, in computer vision the use of physics-based models have been rather limited. The use of physics-based models as priors for articulated human motion tracking were first developed by Metaxas and Terzopoulus [59] in 1993 and Wren and Pentland [102] in 1998. In [59] an upper-body physics based model has been used as a prior for tracking within an on-line tracking framework where inference was performed using a Kalman filter. Observations were obtained using a 3D data acquisition system and manually partitioned as belonging to a given body segment. Similarly in [102] an upper-body model was employed with a Kalman filter, but with much weaker image observations coming from a stereo camera. In both [59] and [102], however, the authors did not address the issue of contact or interactions with the environment.

Recently, there has been somewhat of a revival of the physics-based models in tracking through the series of publications based on low-dimensional biomechanically inspired models for lower body walking motions [12, 11] and more general full-body high-dimensional data-driven physics-based prior model [94]. In both cases much weaker observations were used (as compared to [59] and [102]) and the authors cast the tracking problem in a more general Bayesian inference framework. The key purpose of this document is to enable a further understanding of these methods (along with the physics-based formalism involved) and discuss possible future research directions.

Beyond tracking, physics-based models have also been used in vision to analyze dynamic interactions between objects in the scene [52, 79]. In [5] the observed motion of a non-rigid object was used to recover the physical parameters of that object, the orientation of gravity and camera parameters. Similarly, a recent method for recovery of surface properties through observed interactions has been proposed [10]. Somewhat further afield, but still related, are emerging methods using physics-based models for crowed dynamics analysis. In such a model the crowed is modeled as a set of particles interacting and exerting *social* forces on one another. The effectiveness of such models has been illustrated in computer graphics for synthesis of crowd behavior; the efficacy of such models in computer vision for analysis of crowd behavior is uncertain.

2 Classical Mechanics

This section provides an overview of classical mechanics for an unconstrained rigid body. Traditional texts on this subject, *e.g.*[27, 87], begin with the motion of point masses and work up to rigid body motion. Instead, this section begins by defining the fundamental properties of rigid bodies, and then immediately provides the rigid-body equations of motion. The hope is to provide a direct introduction to the most relevant subjects with an eye towards their use in modeling human motion.

Readers interested in the derivation of these concepts from the motion of simple point masses are referred to the excellent course notes by Witkin and Baraff [99] or the classic textbook by Thornton and Marion [87]

2.1 Mass properties of a rigid body

To begin let's assume that we have a rigid body with a mass distribution (a mass density function) given by $\rho(x,y,z)$. It specifies the mass per unit volume at each point in 3-space (measured in kgm^{-3}). For points inside the object $\rho(\mathbf{x}) > 0$, and for points outside or in hollow regions, $\rho(\mathbf{x}) = 0$.

The mass properties that affect the motion of the body, that is, its response to external forces, can be obtained directly from the mass density function in terms of the zeroth, first and second-

order moments. In particular, the total mass of the rigid body is given by the zeroth moment, that is

$$m = \int_{\mathbf{x}} \rho(\mathbf{x}) d\mathbf{x} = \int_x \int_y \int_z \rho(x,y,z) dz dy dz. \tag{3}$$

The center of mass is defined as the first moment of the density function

$$\mathbf{c} = m^{-1} \int \mathbf{x} \rho(\mathbf{x}) d\mathbf{x}. \tag{4}$$

The center of mass provides a natural origin for a local coordinate system defined for the part.

In reaction to forces acting on the body, the motion of the body also depends on the distribution of mass about the center of mass. The relevant quantity is often referred to as the inertial description, and is defined in terms of the second moments of mass density function. In particular the rotational motion about a specific axis is determined by the moment of inertia about that axis.

The inertia tensor is a convenient way to summarize all moments of inertia of an object with one matrix. It may be calculated with respect to any point in space, although it is convenient to define it with respect to the center of mass of the body. The inertia tensor is defined as follow,

$$\mathbf{I} = \begin{pmatrix} I_{11} & I_{12} & I_{13} \\ I_{21} & I_{22} & I_{23} \\ I_{31} & I_{32} & I_{33} \end{pmatrix} \tag{5}$$

where

$$I_{ij} = \int_{\mathbf{x}} \rho(\mathbf{x}) (\|\mathbf{r}\|^2 \delta_{ij} - r_i r_j) d\mathbf{x} \tag{6}$$

where $\mathbf{r} \equiv (r_1, r_2, r_3)^T = \mathbf{x} - \mathbf{c}$, $\mathbf{c}$ is the center of mass, and δ_{ij} is Kronecker delta function. The diagonal elements of $\mathbf{I}$ are called *moments of inertia* and off-diagonal elements are commonly called *products of inertia*.

Since the inertia tensor is real and symmetric, it has an complete, orthogonal set of eigenvectors, which provide a natural intrinsic coordinate frame for the body (centered at the origin). Within this coordinate frame it is straightforward to show that the inertia tensor is diagonal:

$$\mathbf{I}' = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \tag{7}$$

The local coordinate axes are referred to as the principal axes of inertia and the moments of inertial along those axes, I_x, I_y and I_z, are the principal moments of inertia. In this local coordinate frame the inertial properties are fixed and can be compactly specified by I_x, I_y and I_z. Analytic expressions for the principal moments of inertia for several simple geometrical primitives are given in Table 1.

Measured in the world coordinate frame the inertia tensor (with the center, for convenience, still defined at the center of mass of the body) about the world coordinate axes is just a a function of the relative orientation between the two coordinate frames (*i.e.*, changes as the body rotates). In particular,

$$\mathbf{I} = \mathbf{R}\mathbf{I}'\mathbf{R}^T \tag{8}$$

where $\mathbf{R}$ is a 3-by-3 rotation matrix specifying the orientation of the local intrinsic coordinate frame with respect to the global reference frame.

Shape	Parameters	Principal Moments of Inertia		
		I_x	I_y	I_z
Rectangular prism	a – depth along x-axis b – height along y-axis c – width along z-axis	$\frac{1}{12}m(b^2+c^2)$	$\frac{1}{12}m(a^2+c^2)$	$\frac{1}{12}m(a^2+b^2)$
Cylinder	l – length along x-axis r – radius in y-z plane	$\frac{1}{2}mr^2$	$\frac{1}{12}m(3r^2+l^2)$	$\frac{1}{12}m(3r^2+l^2)$
Elliptical Cylinder	l – length along x-axis r_y – major axis along y-axis r_z – minor axis along z-axis	$\frac{1}{12}m(4r_z^2+l^2)$	$\frac{1}{12}m(3r_y^2+l^2)$	$\frac{1}{4}m(r_y^2+r_z^2)$
Sphere	r – radius	$\frac{2}{5}mr^2$	$\frac{2}{5}mr^2$	$\frac{2}{5}mr^2$
Ellipsoid	r_x – semi-axes length along x-axis r_y – semi-axes length along y-axis r_z – semi-axes length along z-axis	$\frac{1}{5}m(r_y^2+r_z^2)$	$\frac{1}{5}m(r_x^2+r_z^2)$	$\frac{1}{5}m(r_x^2+r_y^2)$

Table 1: **Principal moments of inertia for standard geometric shapes.** Moments of inertia in the table are defined with respect to the center of mass of the corresponding geometry; all geometrical objects are defined in axis aligned coordinate frames. The values are taken from [71]

.

It can also be useful to compute the inertia tensor with respect to a point other then the center of mass (*e.g.*, a joint about which the part will rotate). To do so one can apply the *parallel axes theorem* that states that the new inertial description about the point $\mathbf{x}_0$ can be computed as

$$\hat{\mathbf{I}} = \mathbf{I} + m\left[\|\mathbf{x}_0 - \mathbf{c}\|^2 \mathbf{E}_{3\times 3} - (\mathbf{x}_0 - \mathbf{c})(\mathbf{x}_0 - \mathbf{c})^T\right], \tag{9}$$

where $\mathbf{E}_{3\times 3}$ is the 3×3 identity matrix.

2.2 Pose of a Rigid Body

Crucial in any discussion of mechanics is the frame of reference. The equations of motion can be specified in any chosen coordinate frame, however their forms vary depending on the particular choice. Here only the two most interesting (and practically useful) reference frames will be considered: the world frame and the body frame. The world frame is a static, motionless frame of reference, considered to be defined relative to a fixed origin and set of axes in the world. The body frame is fixed to the body in question. Its origin is at the center of mass and its axes are aligned with the principal axes of inertia.

The pose of a rigid body can then be defined as the transformation which takes a point in the body frame $\mathbf{x}'$ to a point in the world frame $\mathbf{x}$. This transformation is defined by a linear component, $\mathbf{c}$, which specifies the location of the center of mass in the world frame and an angular component represented by a rotation matrix, $\mathbf{R}$, which aligns axes of the body and world frames. Concretely, for a point on the body $\mathbf{x}'$, the corresponding point in the world is given by the rigid transform $\mathbf{x} = \mathbf{R}\mathbf{x}' + \mathbf{c}$.

While the representation of the linear component as a vector in $\mathbb{R}^3$ is obvious, the representation of the orientation is more subtle. Rotation matrices are an option, however the nine parameters

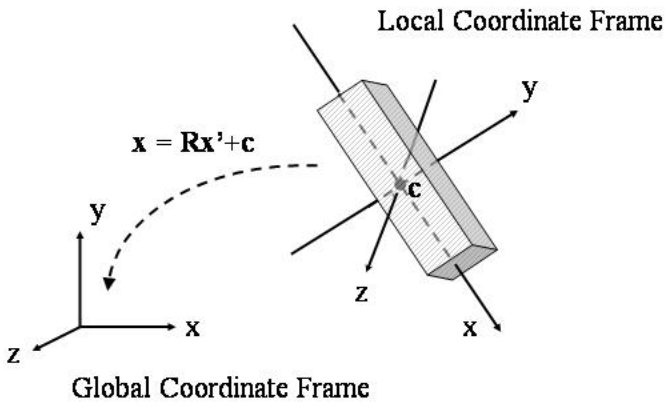

Figure 1: **Pose of a Rigid Body.** Illustration of the reference frames for a rigid body in space.

are many more than the three degrees of freedom of a rotation. Further, during simulation the representation will change over time and ensuring that a matrix remains a rotation during simulation can be difficult. Classical presentations of both kinematics and mechanics typically use Euler angles to represent 3D rotations. With Euler angles, a 3D rotation is specified by a sequence of three rotations about different axes and the entire rotation is defined by the three angles of rotation. Unfortunately, the singularities caused by Gimbal lock and the multiplicity of representations results in Euler angles being a poor choice, particularly in the context of human motion where singularities can be difficult to avoid over long periods of time. Two of the most common and useful alternatives to Euler angles are exponential maps [29] and quaternions [72].

Quaternions are an elegant, singularity free representation of 3D rotations which result in stable and effective simulations. Care must be taken, however, since quaternions represent rotations on a unit sphere in 4D. A review of quaternions is presented in Appendix A.

2.3 Mechanics of a Rigid Body

The motion of a rigid body is traditionally defined in terms of the linear velocity $\mathbf{v}$ of its center of mass and the angular velocity $\boldsymbol{\omega}$ about the center of mass. Linear velocity is simply understood as the instantaneous rate of change over time of the position of the rigid body. In contrast, angular velocity cannot be related as the time derivative of a consistent quantity. Instead, it represents the instantaneous rate of rotation of the body. The magnitude, $\|\boldsymbol{\omega}\|$, is the rate of rotation (*e.g.*, in radians per second), and the direction of the vector $\boldsymbol{\omega}/\|\boldsymbol{\omega}\|$ is the axis of rotation.

Newton's laws of motion relate the time-derivative of momentum to force in a stationary coordinate frame (*e.g.*, the world frame). Linear momentum, $\mathbf{p}$, and angular momentum, $\boldsymbol{\ell}$, are defined as

$$\mathbf{p} = m\mathbf{v} \tag{10}$$
$$\boldsymbol{\ell} = \mathbf{I}\boldsymbol{\omega} \tag{11}$$

for some frame of reference. For motion in the world frame, the Newton-Euler equations of motion specify the linear and angular components of rigid body motion, that is,

$$\dot{\mathbf{p}} = \mathbf{f} \tag{12}$$
$$\dot{\boldsymbol{\ell}} = \tau \tag{13}$$

	World Frame	Body Frame
Momentum	$\dot{\boldsymbol{\ell}} = \tau$	$\dot{\boldsymbol{\ell}}' = \tau' - \boldsymbol{\omega}' \times (\mathbf{I}'\boldsymbol{\omega}')$
Velocity	$\mathbf{I}\dot{\boldsymbol{\omega}} = \tau - \dot{\mathbf{I}}\boldsymbol{\omega}$	$\mathbf{I}'\dot{\boldsymbol{\omega}}' = \tau' - \boldsymbol{\omega}' \times (\mathbf{I}'\boldsymbol{\omega}')$

Table 2: **Various Forms of Eulers Equations of Motion.** The derivatives of angular velocity and momentum in both body and world frames. Any one of these equations can be used to define the angular motion of a rigid body.

where $\mathbf{f}$ represents the linear force acting on the body, τ is the angular force or torque, and the dot indicates the derivative with respect to time. Any frame of reference for which these equations hold is referred to as an inertial frame.

In the body frame, the equations for linear motion are decidedly uninteresting, because the frame is defined to have its origin at the center of mass, and is therefore constant in the local frame through time. In contrast, the equations for angular motion become

$$\mathbf{I}'\dot{\boldsymbol{\omega}}' = \tau' - \boldsymbol{\omega}' \times \left(\mathbf{I}'\boldsymbol{\omega}'\right) \tag{14}$$

where $\mathbf{I}'$ contains the principal moments of inertia, τ' is the torque acting on the system, and $\boldsymbol{\omega}'$ is the angular velocity, with all quantities being measured in the body frame of reference. The equations in Table 2, combined with equation (12), provide the derivatives of angular velocity and momentum.

To simulate the motion of a rigid body we require the notion of state, comprising pose and orientation for a rigid body. The position has a natural representation as the location of the center of mass in the world coordinate frame. Then, velocity and momentum are related to derivatives of state straightforwardly. That is, for position,

$$\dot{\mathbf{c}} = \mathbf{v} = \frac{1}{m}\mathbf{p} \tag{15}$$

is the rate of change of the center of mass $\mathbf{c}$ in the world as a function of linear velocity or linear momentum.

For orientation, the equations of motion in terms of state depend on the choice of representation for orientation. In the case of quaternions, the equations are

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \circ \begin{pmatrix} 0 \\ \boldsymbol{\omega}' \end{pmatrix} \quad \text{or} \quad \dot{\mathbf{q}} = \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{\omega} \end{pmatrix} \circ \mathbf{q} \tag{16}$$

with $\boldsymbol{\omega} = \mathbf{I}^{-1}\boldsymbol{\ell}$ and $\boldsymbol{\omega}' = \mathbf{I}'^{-1}\boldsymbol{\ell}'$.

2.4 Forces and Torques

Newton's laws of motion formally define force as the rate of change of momentum. More concretely, force can be viewed as the result of external actions on an object or system of objects. Forces can come from many sources, such as gravity, magnetism, friction, contact or muscle actuations. From its formal definition then, force is measured in units of mass times length over time squared. The SI unit of force is the Newton (N), where one Newton is the amount of force required to accelerate a one kilogram object at a rate of one meter per seconds squared, that is $\frac{kg\ m}{s^2}$.

Newton's formal definition of force is sufficient when discussing forces acting on a point mass or the center of mass of a system. However, forces can be applied at any point on a system. For

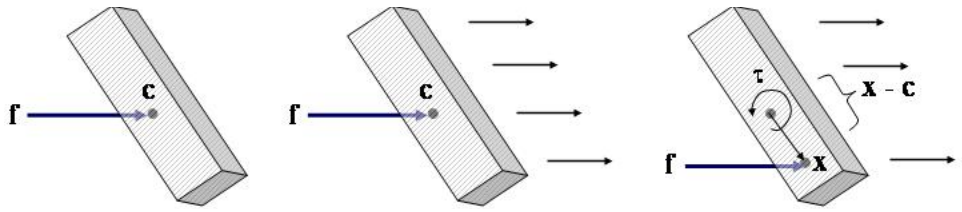

Figure 2: **Rigid motion of the body.** Force applied at a point co-linear with the center of mass will result in only linear motion, where as force applied at a point not co-linear with the center of mass will result in the torque about the center of mass of the body and both linear and angular motion (and momentum).

instance, frictional forces are applied to the surface of a rigid body, not directly on its center of mass. Such forces cause not only a change in linear momentum but also a change in angular momentum. That is, an external force, $\mathbf{f}_e$, acting at a point $\mathbf{x}$ results in a linear force, $\mathbf{f}$, on the center of mass and an angular force or torque, τ about the center of mass. These are related by

$$\mathbf{f} = \mathbf{f}_e \tag{17}$$

$$\tau = (\mathbf{x}-\mathbf{c}) \times \mathbf{f}_e \tag{18}$$

where all quantities are in the world coordinate frame. Torque is measured in units of force times distance which can be seen by rewriting the cross product as

$$\tau = \|\mathbf{x}-\mathbf{c}\| \, \|\mathbf{f}_e\| \, \sin\theta \, \mathbf{n} \tag{19}$$

where θ is the angle between $\mathbf{x}-\mathbf{c}$ and $\mathbf{f}_e$ and $\mathbf{n}$ is a unit vector orthogonal to both $\mathbf{x}-\mathbf{c}$ and $\mathbf{f}_e$. The SI unit for torque is the Newton meter, denoted by $N\ m$. Finally, if there are multiple forces and torques acting on the center of mass of a rigid body, the net result can be summarized by a single force and torque which is the sum of the individual forces and torques.

2.5 Simulating Motion of a Rigid Body

Simulating the motion of a rigid body is done by defining a differential equation and, given an initial condition, integrating these equations over time. The concepts and equations above provide the foundation for doing this. The state vector must describe both the position and orientation of the rigid body as well their instantaneous rate of change. For instance one choice of state vector is

$$\mathbf{y} = \begin{pmatrix} \mathbf{c} \\ \mathbf{q} \\ \mathbf{v} \\ \boldsymbol{\omega}' \end{pmatrix} \tag{20}$$

where, as above, $\mathbf{c}$ is the center of mass, $\mathbf{q}$ is a quaternion, $\mathbf{v}$ is linear velocity, and $\boldsymbol{\omega}'$ is angular velocity. There are several possible alternative permutations of state, by including linear and angular momentum vectors instead of velocity vectors and measuring angular motion in the world instead of the body frame.

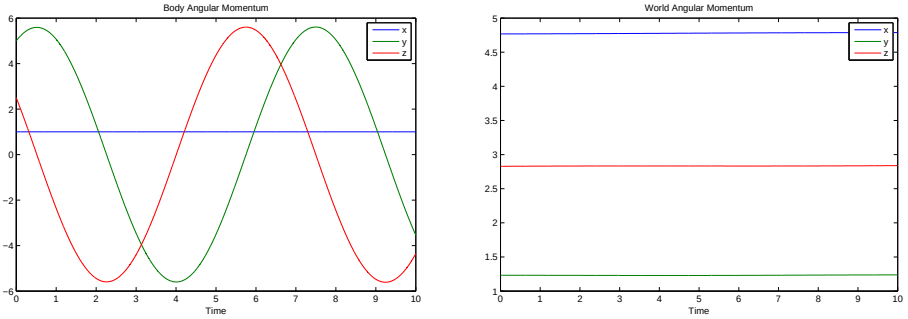

Figure 3: Angular momentum in the body (left) and world (right) coordinate frames for a rigid body.

The differential equation can then be specified by using the relevant equations from above yielding

$$\dot{\mathbf{y}} = \begin{pmatrix} \mathbf{v} \\ \frac{1}{2}\mathbf{q} \circ \begin{pmatrix} 0 \\ \boldsymbol{\omega}' \end{pmatrix} \\ m^{-1}\mathbf{f} \\ \mathbf{I}'^{-1}\left(\boldsymbol{\tau}' - \boldsymbol{\omega}' \times (\mathbf{I}'\boldsymbol{\omega}')\right) \end{pmatrix} \tag{21}$$

this can then be fed in to any standard initial value problem solver to simulate the resulting motion. For instance, the first-order Euler integration method can be used where

$$\mathbf{y}(t+\delta) = \mathbf{y}(t) + \delta\dot{\mathbf{y}}(t) \tag{22}$$

for some step-size δ. This numerical integration step is simple, fast and easy to implement. Unfortunately it is also inaccurate for anything but very small step sizes or very slow motions. More complex methods can be used and are discussed in Section 4.1. One note of care must be taken with any numerical integration scheme. The quaternion norm will slowly drift away from one over time. To avoid this, at the end of each numerical integration step, the quaternion can be renormalized.

The MATLAB code associated with this report includes implementations of these equations and integrates them with a simple Euler step. The results of this code clearly demonstrates several things. For instance, figure 3 (left) plots $\boldsymbol{\ell}'$, the body frame angular momentum, for a rotating body in the absence of torque. Note how $\boldsymbol{\ell}'$ changes overtime, even with $\boldsymbol{\tau}' = 0$. In contrast, figure 3 (right) plots $\boldsymbol{\ell}$, the world frame angular momentum for the same motion. Here it can be seen that the momentum is conserved.

3 Constrained Dynamics

The equations of motion presented in section 2.3 are for a single, unconstrained rigid body. In practice, for many problems of interest there are multiple interacting bodies and constraints that must be enforced. Examples of such constraints include 1) the constraint that two parts of an articulated body have a fixed point of relative motion at the the joint connecting them, 2) the fact that joints often have a restricted range of movement, 3) ground penetration constraints, and 4) the unit norm constraint that ensures that the quaternion used to represent rotation has norm one.

This section begins with the principle of virtual work that can be used to derive the equations of motion for constrained systems. In Section 3.1 we derive the equations of motion entirely in

terms of quaternions as an example of explicitly enforcing constraints with constraint forces and Lagrange multipliers. In Section 3.2 the generalized coordinates are introduced and used to derive equations of motion for a constrained set of rigid bodies. Finally, Section 3.3 demonstrates a formulaic approach for generating equations of motion for systems of articulated rigid bodies.

3.1 The Principle of Virtual Work

Consider the problem of finding equations of motion for a system constrained by N constraint functions such that $\mathbf{e}(\mathbf{x}) = (e_1(\mathbf{x}), \ldots, e_N(\mathbf{x}))^T = 0$. In the case of a quaterion $\mathbf{q}$, for example, we require that $\mathbf{q}^T\mathbf{q} - 1 = 0$. For a collection of constraints an admissable state, $\mathbf{x}$, is defined to be one for which $\mathbf{e}(\mathbf{x}) = 0$. Differentiating the constraint, we find that an admissable velocity, $\dot{\mathbf{x}}$, satisifies

$$\dot{\mathbf{e}} = \frac{\partial \mathbf{e}}{\partial \mathbf{x}}\dot{\mathbf{x}} = 0\,, \tag{23}$$

and an admissable acceleration is therefore one for which

$$\ddot{\mathbf{e}} = \frac{\partial \dot{\mathbf{e}}}{\partial \mathbf{x}}\dot{\mathbf{x}} + \frac{\partial \mathbf{e}}{\partial \mathbf{x}}\ddot{\mathbf{x}} = 0\,. \tag{24}$$

Now, assume that for an unconstrained version of the system the equations of motion can be written as

$$\mathbf{M}(\mathbf{x})\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{f_e} \tag{25}$$

where $\mathbf{M}$ is a mass matrix, $\mathbf{f}$ are the system forces and $\mathbf{f_e}$ are constraint forces that will be used to enforce the necessary constraints.

To determine the constraint forces the principle of virtual work is applied. The principle of virtual work requires that the work, δW, done by a constraint force must be zero for every admissable velocity $\dot{\mathbf{x}}$. That is,

$$\delta W = \mathbf{f_e}^T\dot{\mathbf{x}} = 0 \tag{26}$$

for all $\dot{\mathbf{x}}$ such that $\frac{\partial \mathbf{e}}{\partial \mathbf{x}}\dot{\mathbf{x}} = 0$. For example, in the case of quaternions the principle of virtual work says, in order to maintain the unit normal constraint on the quaternion representation, the constraint force, in and of itself, should not induce any rotation. In that case, all admissible velocities lie in the tangent plane to the unit sphere in 4D, and that therefore the constraint forces must be normal to the tangent plane.

By combining Equations (23) and (26) we find that the space of such constraint forces can be specified as

$$\mathbf{f_e} = \frac{\partial \mathbf{e}}{\partial \mathbf{x}}^T \lambda \tag{27}$$

where λ is a vector of Lagrange multipliers. Substituting Equation (27) into Equation (25) and combining it with Equation (24) gives

$$\begin{pmatrix} \mathbf{M}(\mathbf{x}) & \frac{\partial \mathbf{e}}{\partial \mathbf{x}}^T \\ \frac{\partial \mathbf{e}}{\partial \mathbf{x}} & 0 \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{x}} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}) \\ -\frac{\partial \dot{\mathbf{e}}}{\partial \mathbf{x}}\dot{\mathbf{x}} \end{pmatrix} \tag{28}$$

which is a fully constrained set of equations.

This approach is broadly applicable. For instance, the equations of motion of a rigid body in terms of quaternion accelerations can be derived by substituting equations (63) and (65) into equation (14), multiplying it by $2\mathbf{Q}(\mathbf{q})$ and adding the constraint $e(\mathbf{q}) = \|\mathbf{q}\|^2 - 1$. This gives

$$\begin{pmatrix} 4\mathbf{Q}\mathbf{J}\mathbf{Q}^T & 2\mathbf{q} \\ 2\mathbf{q}^T & 0 \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{q}} \\ \lambda \end{pmatrix} = \begin{pmatrix} 2\mathbf{Q}\begin{pmatrix} 0 \\ \tau' \end{pmatrix} + 8\dot{\mathbf{Q}}\mathbf{J}\dot{\mathbf{Q}}^T\mathbf{q} \\ -2\|\dot{\mathbf{q}}\|^2 \end{pmatrix} \tag{29}$$

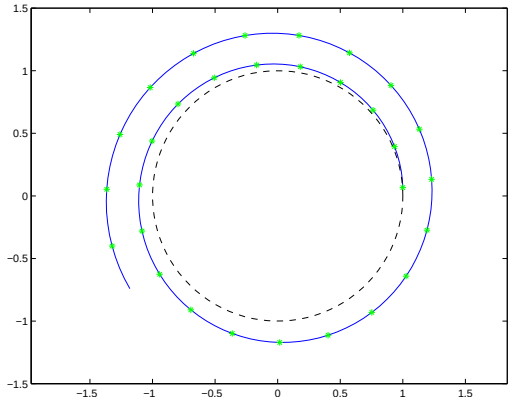

Figure 4: Simulation of a point mass constrained to lie on a circle around the origin. The dashed black line is the circle constrained, the solid blue line is the trajectory of the point mass through time and the green crosses are spaced every second.

where $\mathbf{J} = \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{I}' \end{pmatrix}$ and $\dot{\mathbf{Q}} = \mathbf{Q}(\dot{\mathbf{q}})$. More interesting uses include equations of motion for pendulums or bodies in static contact.

Unfortunately, equations derived with this method will tend to drift during simulation due to the accumulation of error in numerical integration. Figure 4 shows a point mass constrained to lie on a circle around the origin. While it starts close to the circle, it slowly drifts away with time. Several solutions to this problem exist. One approach, which works well with the quaternion constraints in Equation (29), is to reproject the state to satisfy the constraints. This must be done for both the state and its derivatives to be effective. However, it is not always obvious how to do the projection with multiple, complex constraints. Another approach is to change Equation (24) so that $\ddot{\mathbf{e}} = -\alpha\mathbf{e} - \beta\dot{\mathbf{e}}$ for some choice of parameters α and β [101, 99]. This method, sometimes called constraint stabalization, can be thought of as a damped spring which corrects errors in the constraints. Notice that if the constraint is satisified then this modification has no impact on the system. However, neither of these solutions are ideal for large numbers of complex constraints, such as those implied by an articulated body. For that, the concept of generalized coordinates is introduced next.

3.2 Generalized Coordinates

Generalized coordinates are any set of coordinates $\mathbf{u}$ which completely describe the state of a physical system. In the case of a constrained system, these generalized coordinates can implicitly define the constraints. For instance, an articulated set of rigid bodies which represent a person can be described by the relative orientations between connected parts of the body and the position and orientation of a root node. Then, the constraint that certain parts be connected is implied by the choice of $\mathbf{u}$ rather than by explicit constraint functions as was done in the previous section.

Deriving the equations of motion in terms of $\mathbf{u}$ for such a system can be done in a variety of ways. Traditionally the Lagrangian method is used, however it can often be confusing and difficult for novices. Instead, the TMT method [91] is presented as being the most straightforward for modelling human motion. However, it should be noted that the myriad approachs to deriving equations of motion with generalized coordinates are all mathematically equivalent. The derivation of the TMT method is a simple and elegant application of the principle of virtual work and is

presented next.

Beginning as in section 3.1 above, let the state of the unconstrained system be described by the vector $\mathbf{x}$ and let its equations of motion be given by (25) with some constraint forces By definition, there is a function $\mathbf{x}(\mathbf{u})$ which maps the generalized coordinates $\mathbf{u}$ into the state of the unconstrained system. This function is called the kinematic transformation. For example, $\mathbf{u}$ might be a vector of joint angles for an articulated body, and $\mathbf{T}(\mathbf{u})$ might be the mapping from joint angles to the position and orientation of the component parts of the articulated body.

Differentiating the kinematic transformation with respect to time gives the set of admissable velocities

$$\dot{\mathbf{x}} = \mathbf{T}(\mathbf{u})\dot{\mathbf{u}} \tag{30}$$

and the set of admissable accelerations

$$\ddot{\mathbf{x}} = \mathbf{T}(\mathbf{u})\ddot{\mathbf{u}} + \dot{\mathbf{T}}(\mathbf{u},\dot{\mathbf{u}})\dot{\mathbf{u}} \tag{31}$$

where $\mathbf{T} = \frac{\partial \mathbf{x}}{\partial \mathbf{u}}$ is the Jacobian of the kinematic transformation. The principle of virtual work requires, for all $\dot{\mathbf{u}}$, that

$$\delta W = \mathbf{f}_\mathbf{e}^T \mathbf{T}(\mathbf{u})\dot{\mathbf{u}} = 0 \tag{32}$$

which implies $\mathbf{T}(\mathbf{u})^T \mathbf{f}_\mathbf{e} = 0$. Premultiplying equation (25) by $\mathbf{T}(\mathbf{u})^T$ causes the constraint forces $\mathbf{f}_\mathbf{e}$ to vanish. Substituting $\mathbf{x}(\mathbf{u})$ and its derivatives, Equations (30) and (31), then gives

$$\mathbf{T}(\mathbf{u})^T \mathbf{M}(\mathbf{u})\mathbf{T}(\mathbf{u})\ddot{\mathbf{u}} = \mathbf{T}(\mathbf{u})^T \left(\mathbf{f}(\mathbf{u},\dot{\mathbf{u}}) - \mathbf{M}(\mathbf{u})\dot{\mathbf{T}}(\mathbf{u},\dot{\mathbf{u}})\dot{\mathbf{u}}\right) \tag{33}$$

which can be rewritten as

$$\mathscr{M}(\mathbf{u})\ddot{\mathbf{u}} = \mathbf{T}(\mathbf{u})^T \mathbf{f}(\mathbf{u},\dot{\mathbf{u}}) + \mathfrak{g}(\mathbf{u},\dot{\mathbf{u}}) \tag{34}$$

where $\mathscr{M}(\mathbf{u}) = \mathbf{T}(\mathbf{u})^T \mathbf{M}(\mathbf{u})\mathbf{T}(\mathbf{u})$ is called the generalized mass matrix, and $\mathfrak{g}(\mathbf{u},\dot{\mathbf{u}}) = -\mathbf{T}(\mathbf{u})^T \mathbf{M}(\mathbf{u})\dot{\mathbf{T}}(\mathbf{u},\dot{\mathbf{u}})\dot{\mathbf{u}}$.

3.3 Dynamics of Articulated, Rigid Bodies

The TMT method provides a general technique for deriving equations of motion of a constrained system in terms of generalized coordinates. The resulting equation (34) provides a compact and computationally efficient way to define a second-order ordinary differential equation which can be fed directly into standard initial and boundary value problem solvers. However, it may be somewhat unclear how best to apply the TMT method to an articulated body. Below is a step-by-step recipe for doing just this.

1. Define the set of parts which make up the articulated body. For each part i specify its mass, m_i, and inertia tensor, $\mathbf{I}'_i$, in the body frame. Let $\mathbf{c}_i$ and $\mathbf{q}_i$ denote the position of the center of mass and the orientation of the part in the world frame as discussed in 2.2. The forces acting on part i is summarized by the linear force $\mathbf{f}_i$ and torque τ_i.
2. Specify the equations of motion for the unconstrained system by defining the terms in equation (25) beginning with the pose vector

$$\mathbf{x} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{q}_1 \\ \vdots \\ \mathbf{c}_N \\ \mathbf{q}_N \end{pmatrix}. \tag{35}$$

Then the mass matrix is

$$\mathbf{M}(\mathbf{x}) = \begin{pmatrix} m_1\mathbf{E}_{3\times3} & & & & \\ & 4\mathbf{Q}(\mathbf{q}_1)\mathbf{J}_1\mathbf{Q}(\mathbf{q}_1)^T & & & \\ & & \ddots & & \\ & & & m_N\mathbf{E}_{3\times3} & \\ & & & & 4\mathbf{Q}(\mathbf{q}_N)\mathbf{J}_N\mathbf{Q}(\mathbf{q}_N)^T \end{pmatrix} \tag{36}$$

and the force function is

$$\mathbf{f}(\mathbf{x},\dot{\mathbf{x}}) = \mathbf{F}(\mathbf{x})\hat{\mathbf{f}} + \mathbf{a}(\mathbf{x},\dot{\mathbf{x}}) \tag{37}$$

where

$$\hat{\mathbf{f}} = \begin{pmatrix} \mathbf{f}_1 \\ 0 \\ \tau_1 \\ \vdots \\ \mathbf{f}_N \\ 0 \\ \tau_N \end{pmatrix} \tag{38}$$

is the system force vector,

$$\mathbf{F}(\mathbf{x}) = \begin{pmatrix} \mathbf{E}_{3\times3} & & & & \\ & 2\bar{\mathbf{Q}}(\mathbf{q}_1) & & & \\ & & \ddots & & \\ & & & \mathbf{E}_{3\times3} & \\ & & & & 2\bar{\mathbf{Q}}(\mathbf{q}_N) \end{pmatrix} \tag{39}$$

is a matrix which transforms the system force vector, and

$$\mathbf{a}(\mathbf{x},\dot{\mathbf{x}}) = \begin{pmatrix} \mathbf{0} \\ 8\mathbf{Q}(\dot{\mathbf{q}}_1)\mathbf{J}_1\mathbf{Q}(\dot{\mathbf{q}}_1)^T\mathbf{q}_1 \\ \vdots \\ \mathbf{0} \\ 8\mathbf{Q}(\dot{\mathbf{q}}_N)\mathbf{J}_N\mathbf{Q}(\dot{\mathbf{q}}_N)^T\mathbf{q}_N \end{pmatrix} \tag{40}$$

3. Specify a set of generalized coordinates $\mathbf{u}$ which represents the pose of the articulated body and define the kinematic transformation function $\mathbf{x}(\mathbf{u})$. Derive expressions for $\mathbf{T}(\mathbf{u}) = \frac{\partial \mathbf{x}}{\partial \mathbf{u}}$ and

$$\mathfrak{g}(\mathbf{u},\dot{\mathbf{u}}) = -\mathbf{T}(\mathbf{u})^T\mathbf{M}(\mathbf{u})\dot{\mathbf{T}}(\mathbf{u},\dot{\mathbf{u}})\dot{\mathbf{u}} = -\mathbf{T}(\mathbf{u})^T\mathbf{M}(\mathbf{u})\left(\sum_i \frac{\partial \mathbf{T}}{\partial u_i}\dot{u}_i\right)\dot{\mathbf{u}} \tag{41}$$

where u_i and $\dot{u}_i$ refer the the ith component of $\mathbf{u}$ and $\dot{\mathbf{u}}$ respectively.

This can be done efficiently by specifying a kinematic tree for the articulated body. The generalized coordinates are the relative orientations between connected parts plus the position and orientation of the root node of the tree. The orientations in the generalized coordinates can be represented using any method, although quaternions make a convenient choice since composing orientations is simply done by multiplying quaternions.

4. Define any remaining constraints $\mathbf{e}(\mathbf{u})$ on the generalized coordinates $\mathbf{u}$. For instance, if quaternions were used to represent orientations in the generalized coordinates, the unit norm constraints need to be specified. Another example is if a part of the body is occasionally attached to some part of the environment. While this could be enforced with a new set of generalized coordinates, this would require switching equations of motion which is tedious and error prone. Instead, constraint functions can easily be added and removed as needed in order to handle this.

5. The final equations of motion are then

$$\begin{pmatrix} \mathscr{M}(\mathbf{u}) & \frac{\partial \mathbf{e}}{\partial \mathbf{u}}^T \\ \frac{\partial \mathbf{e}}{\partial \mathbf{u}} & 0 \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{u}} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{T}(\mathbf{u})^T \mathbf{f}(\mathbf{u}, \dot{\mathbf{u}}) + \mathbf{g}(\mathbf{u}, \dot{\mathbf{u}}) \\ -\frac{\partial \dot{\mathbf{e}}}{\partial \mathbf{u}} \dot{\mathbf{u}} \end{pmatrix} \tag{42}$$

4 Numerical Simulation of Motion

The previous sections have dicussed the fundamentals of physics required to describe the motion of rigid and articulated bodies. This section discusses some of the practical considerations in implementing them. Section 4.1 discusses methods appropriate for integrating the derived differential equations. Section 4.2 discusses using off-the-shelf physics simulators.

4.1 Integrating Equations of Motion

As previously seen, equations of motion form a first-order or second-order ordinary differential equation. In both cases this can be expressed as a first-order ODE[2]

$$\dot{\mathbf{y}} = \mathbf{g}(t, \mathbf{y}) \tag{43}$$

where t is time, $\mathbf{y}$ is the state-vector and $\mathbf{g}$ is a function specifying the time derivative. Simulation of motion corresponds to an *initial value problem* where given some initial state $\mathbf{y}_0$ at time t_0 the state $\mathbf{y}_1$ at time t_1 is desired. This is a classical problem in numerical analysis and readers in a formal review of the subject should consult a text book such as [35]. However, simulation of the ODEs which result from physical motion have a variety of requirements not often found in standard texts and software packages. Event detection, constraint enforcement and energy conservation are the three of the most common and most difficult to handle.

Event Detection When modeling contact and other complex forms of constraints, it is often important to know when a particular event (such as a collision) occurs. This is commonly defined as detecting zeros of a function $\mathbf{e}(t, \mathbf{y})$. This function could be the distance of a foot off the ground in order to detect collisions or the angle of the knee in order to detect hyperextension. Simple approaches to this can be bolted on to standard integration methods by checking for a change in the sign of the event function over a single step [74]. If a zero is detected, the event can be localized by searching the interval to find the precise time of the event. Unfortunately, such approaches are prone to problems. If the event function oscillates, the step size is large or the sign of the function doesn't change around the zero, events can be missed. Alternately, the same event can be detected multiple times due to numerical issues. Once found, event localization can be extremely computationally expensive, particularly for multistep integration methods.

[2] A Nth order ODE can be transformed into a first order ODE by augmenting the state to include the first $N-1$ derivatives.

All of these issues can be handled by designing the numerical integration method with events in mind. In particular, Grabner and Kecskeméthy [28] designs a variant of the adaptive stepsize Runge-Kutta 4/5 method of Dormand [22] which gracefully handles the above mentioned issues. They do this by defining a new ODE which includes the value of the event functions in the state vector. This is then integrated using a standard adaptive step-size method. This ensures that the step-sizes are small enough to accurately capture variations in the event function. Further, the Runge-Kutta pair used produces a fifth order polynomial approximation of the state throughout a step which can be used to check for zeros and efficiently localize them if present. Finally, multiple detections of the same event are avoided by appropriately adjusting the initial conditions after an event has been detected.

Constraint Enforcement As noted in Section 3, if constraints are enforced explicitly with constraint forces, numerical errors in integration can result in drift. While using generalized coordinates avoids this in many cases, sometimes, as with the unit norm constraint for quaternions, they cannot be used. Constraint stabilization provides one solution to this problem [92]. However, it can result in stiffer equations of motion if constraints need to be strongly enforced or are expected to be significantly violated.

Alternately, the projection method can be used where, at the end of each integration step, the state is changed so that the constraints are exactly. This method is known to work well for quaternions, where the quaternion $\mathbf{q}$ is scaled to have unit norm and it's velocity is project such that $\mathbf{q}^T\dot{\mathbf{q}} = 0$. In this case, the projection is fast and easy to compute and ensures that the quaternions are always interpretable as rotations.

It is important to note that the projection method should only be used when, analytically, the constraints would always be satisfied. It is not valid to simply integrate unconstrained equations of motion and project the state after each step in order to enforce constraints. Such an approach will result in strange behaviour and unrealistic motion as momentum and energy will not be properly conserved.

Energy Conservation Energy conservation in some frame of reference is a fundamental property of physical motions. Unfortunately, as demonstrated with constraints, quantities which should, theoretically, be conserved, rarely are due to approximate numerical integration. Further, other properties of physical systems, such as time reversability and stable orbits of closed systems, are often violated with traditional numerical methods. While some of this may be unavoidable due to finite numerical precision, most of it is due to the nature of approximations in traditional numerical integration schemes. Though well beyond the scope of this discussion, geometric numerical integration methods, poineered by Hairer et al. [30] and others, seek to design methods which preserve properities of the system.

4.2 Off-The-Shelf Simulators

In Section 3 we discussed the formulation of equations of motion for constrained systems either with the use of explicit constraints (Section 3.1) or through the use of low dimensional parameterizations in the form of generalized coordinates (Section 3.2). While it is educational to derive the equations of motion from first principles, it is also tedious and there are many (public and consumer) physics-based simulators in existence that facilitate this process. For example: Bullet [47], Crisis [93], Havok[3], Newton [62], Novodex (Ageia PhysX) [2], and Open Dynamics Engine

[3] A consumer simulation engine used in many computer games (see `http://www.havok.com/`).

(ODE) [82] to name a few.

Since these engines are generic (*i.e.*, are not specific to any given kinematic structure) they typically formulate the equations of motion using constraints [93, 82] (rather then generalized coordinates, which would require symbolic derivation of equations of motion for every kinematic structure of interest). There are a few essential factors that determine the overall performance of the physics engine: (1) allowed geometry, (2) supported constraint types, (3) precision of collision detection for determining contact, (4) collision handling paradigm (*e.g.*, penalty-based or constraint-based), (5) numerical integrator accuracy, and (6) supported material properties (*e.g.*, for example ability to model static friction, dynamic friction, or restitution). Depending on the application demands different engines may be optimal, however, one of the most widely choices (particularly in research) is ODE [82]. A thorough review of performance of many of above mentioned physics engines is given by Boeing and Braunl in [7].

It is worth mentioning that many of the above engines (including ODE and Crisis) support a variety of joint constraints (often referred to as *anchors*). For example, 3 degree-of-freedom ball-and-socket joints, 2 degree-of-freedom universal joints, 1 degree-of-freedom hinge joints and prismatic joints are common. These engines also provide means of actuating these joints with *motors* that are typically attached to the anchors and are actuated by submitting desired velocities or positions (for trajectory control of this form will be discussed in Section 6.1.1). The key benefit of these engines, however, are the efficient collision detection libraries that test whether a collision between the environment and an object (or between objects) has occured during a simulation step. To ensure efficency in the collision detection, collision detection must be implemented for every type of geometry supported by the engine (similarly to ray-intersection in a ray tracing). Consequently, engines such as PhysX and Bullet support general non-convex triangular mesh objects making them readily applicable to any geometry (ODE has limited support for static triangular meshes).

Computationally, all of the above mentioned engines are designed for real-time performance and hence can easily simulate in real time at high frame rates (sometimes at 1000 frames per second or higher for a typical scene). The high frame rate simulation is often necessary to overcome the instabilities of collision detection and numerical simulation.

Lastly, it is worth mentioning that there are also tools for deriving equations of motions symbolically, using generalized coordinates. The prime example is SD/FAST library [84], which is able to derive equations of motion symbolically even for large systems containing up to 300 rigid bodies and 1000 degrees of freedom. While symbolic derivation of equations of motion allows faster and more stable simulation, SD/FAST does not provide support for collision detection, putting the onus on the user for implementing this typically non-trivial task.

5 Biomechanics of Human Motion

Biomechanics is the study of the biological organisms as a physical system. This section presents the most important results and measurements for building physical models of humans. It also reviews some results in the characterization of human locomotion including models which have been successfully used to build trackers.

This section cannot possibly be a complete introduction of the field, but is instead a collection of the most interesting or useful results in the context of this paper. For a more thorough treatment, readers are referred to the excellent text books [109, 110, 71] from which much of this material is drawn.

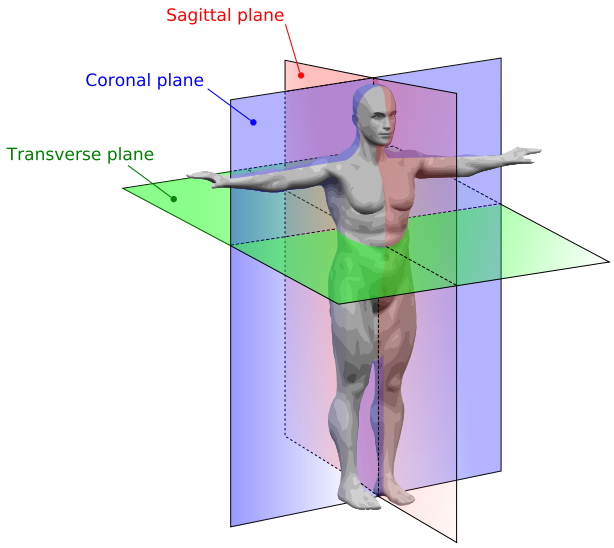

Figure 5: The three traditional anatomical planes of the body. ©Yassine Mrabet, CC-BY-SA 1.0.

5.1 Kinematics

The human body is a complex collection of bones, muscles and other soft tissues. How segments of the body, and the bones which constitute them, are connected to each other is the subject of kinematics. Of importance for computer vision in general and physics in particular, is how to define the pose of a person which can be used as generalized coordinates.

Below the major joints of the human body are discussed along with their range of motion. Appropriate or typical simplifications are discussed in each case. It should be noted that these simplifications, though often crude, are generally necessary for computer vision and graphics. Some estimates place the number of degrees of freedom in the human body at well over 200, far more than is reasonable or necessary is most applications. One simple way to understand this complexity is to realize that the joints of the body are not rigid. Cartilage can compress and expand and ligaments can stretch making seemingly simple ball-and-socket joints like the hip suddenly have a full six degrees of freedom. Further, joints rarely rotate about a fixed set of orthogonal axes, instead often rotating about axes which change as a function of pose.

For the purposes of this description it is convenient to define the traditional anatomic planes of the body. The sagittal plane runs vertically through the body and includes the direction of forward motion. The coronal plane also runs vertically but is perpendicular to the sagittal plane. Finally, the transverse plane is parallel to the ground. These are illustrated in Figure 5.

Hip The hip joint is where the proximal end of the femur attaches to the pelvic girdle. The ball-like head of the femur fits into a concave portion of the pelvis known as the acetabulum. Both the head of the femur and the acetabulum are covered in cartilage which allows a smooth movement between the surfaces. Because of the geometry, the joint is well modeled by a three-degree of freedom, ball and socket joint.

Knee The knee joint is actually considered to consist of two separate joints: the tibiofemoral joint and the patellofemoral joints. The patellofemoral joint, the joint between the patella (*i.e.*,

the knee-cap) and the femur, is primarily of clinical interest but may also be of interest in more detailed muscle models. During knee flexion the patella moves significantly along the femur which can change the effective strength of the quadriceps.

The tibiofemoral joint, which is what is commonly meant by the "knee joint", is the joint between the distal end of the femur and the proximal end of the tibia. The tibiofemoral joint rotates in all three planes of motion, however the range of motion in many of these planes is small and depends strongly on the amount of flexion, *i.e.*, rotation in the sagittal plane. Rotation in the coronal plane is, at most, only a few degrees. Rotation in the transverse plane ranges from practically nothing when the knee is fully extended, to a range of up to 75 degrees with 90 degrees of flexion. The motion of the tibiofemoral joint is further complicated as the center of rotation is not well defined and is not fixed.

In spite of these complications, the knee joint is often modeled as a simple, one degree of freedom hinge joint. The axis of rotation is usually assumed to be normal to the sagittal plane and the center of rotation is fixed. This model is generally sufficient for most applications in computer vision and computer graphics. As new applications arise in biomechanics this gross simplification may no longer be tenable.

Ankle Like the knee, the ankle joint actually consists of two joints: the talocrural joint and the subtalar joint. Unlike the knee, both joints are significant in the motion of the distal segment, the foot. Both joints are effectively hinge joints but with axes which are oblique to the anatomic planes.

The talocrural joint joins the distal ends of the tibia and the fibula to the talus. The axis of rotation is roughly defined by the line through tips of the malleoli, the bony protrusions on either side of the ankle. The center of rotation is approximately located at the midpoint of a line between the lateral (outer) malleolus and a point 5mm below the tibial (inner) malleolus.

The subtalar joint joins the talus with the calcaneus and rotates about an axis which is about 42 degrees out of the transverse plane, pointing up, and 23 degrees out of the sagittal plane, pointing towards the opposing foot. Thus, rotation of the tibia about this joint has the effect of raising or lowering the inside of the foot.

Measurement of the motion of these joints independently is difficult. For most purposes, the joint is combined into a single two degree of freedom joint between the shank and the foot. A biologically and kinematically accurate choice of these degrees of freedom would be the angles of rotation about the two aforementioned axes. The space of rotation spanned by these two angles is also reasonably approximated by a rotation with no spin in the coronal plane, *i.e.*, in the direction of the tibia. Quaternions (see appendix A) and exponential maps [29] both can be easily constrained to lie in this 2D space.

Trunk and Neck The spine has 33 vertibrae including the sacrum and coccyx. Because the intervertebral discs can compress, both linear and angular motions are possible between them giving more 100 articulations in the spinal column. However, the motion of the vertibrae are not independent and the spinal column is can be divided into five segments.

Starting from the skull, the first 7 vertebrae are called cervical vertebrae, which constitutes the neck. The 7th cervical vertebra, C-7, can be identified as the bony protrusion at the base of the neck. C-7 is often used as an anatomical landmark and is sometimes called the cervicale. The next 12 vertebrae are the thoracic vertebrae, followed by the 5 lumbar vertebrae. The next 5 vertebrae are fused together and form the sacrum which are attached to the hip bones and form part of the pelvic girdle. The final 4 vertebrae are also fused together to form the coccyx, more commonly known as the tail bone.

Detailed kinematic models of the spine and torso are rarely necessary. In most applications in computer vision, fairly simple models suffice. Typically, the most complex models divide the body (and thus the spine) into the neck (cervical vertebrae), thorax (thoracic vertebrae), abdomen (lumbar vertebrae) and the pelvis (sacral vertebrae). The joints between the head, neck, thorax, abdomen and pelvis are all then assumed to be three degree of freedom ball-and-socket joints. Further simplifications of this model are commonly made by combining the thorax and abdomen or the thorax, abdomen and pelvis into a single part. Additionally, in many vision applications the the head and neck are also combined.

Shoulder The shoulder complex contains three joints. The proximal end of the humerus fits into the glenoid cavity of the scapula, or shoulder blade, to form the glenohumeral joint. The glenohumeral joint is a ball-and-socket joint, similar to the hip, but the glenoid cavity is shallower than the acetabulum, making the shoulder more prone to dislocation. The acromion of the scapula connects to the clavicle, or collar bone, by way of the acromioclavicular joint. This joint serves primarily to orient the glenoid cavity on the scapula to provide a wider range of motion for the humerus. The acromion can be identified as the bony protrusion located above the glenohumeral joint. The clavicle is connected to the sternum through the sternoclavicular joint which is located to the side of the suprasternal notch. This is also a three degree of freedom ball-and-socket joint.

The above suggests a redundant nine degree of freedom kinematic relation between the humerus and the sternum. This number can be reduced somewhat because motion of the clavicle and the scapula are not independent. Taken together, these two bones form the shoulder girdle which has roughly four degrees of freedom relative to the sternum: two translational degrees of freedom in the sagittal plane and two rotations, one each in the transverse and coronal planes.

Kinematic models in computer graphics and computer vision typically use even simpler models. Many regard the shoulder complex as rigidly attached to the sternum, leaving only the three degree of freedom glenohumeral joint. Such models are sufficient for many tracking applications which focus primarily on locomotion. In contrast, if more complex motions are considered (*e.g.*, gymnastics) such coarse approximations are clearly inadequate.

Elbow The elbow joint actually consists two joints, the humeroulnar and humeroradial joints, which connect the distal end of the humerus to the proximal ends of the ulna and radius. The humeroradial joint is a ball-and-socket joint and the humeroulnar is a hinge joint. Together, the two joints form a hinge joint between the upper and lower arm. In the pose show in Figure 5, the axis of rotation for this joint is approximately normal to the transverse plane.

The forearm has an additional rotational degree of freedom cause by the ability of the radius to rotate relative to the ulna in the humeroradial joint. This results in a deformation of lower arm which can be viewed as a spin of the distal end of the radius about an axis defined by the length of the ulna.

Together, these two rotations can be readily modeled as a sequence of one degree of freedom rotations. If the hand is not being considered, then the spin of the radius about the ulna can generally be ignored as it is difficult to measure in most computer vision domains.

5.2 Anthropometrics

Anthropometrics is the study of human size and weight. Of particular interest for this documents are measurements of the stature, limb length and segment mass properties. Several studies have

	Female			Male		
	5th%	50th%	95th%	5th%	50th%	95th%
Total body mass (kg)	49.44	59.85	72.43	66.22	80.42	96.41
Total body height (m)	1.52	1.62	1.72	1.65	1.76	1.90

Table 3: **Total body mass and height.** Total body mass and height for males and females at 5th, 50th and 95th percentile of the respective population (the 50th percentile can be thought of as the mean of value for the population). Reported values are based on those in [33].

been made of such parameters and a number of standard tables are available [71] either with averages for the entire population, or separated by the group (*e.g.*, sex and age). Studies also differ as to the exact definition of segments and segment endpoints. For the purposes of this document we utilize the values in [19] that are based on the measurements originally made by Zatsiorsky et al. [108].

It is important to note that, due to the difficulty of performing these studies, the available data is not generally representative of a broad population. For instance, the classic and often used study by Dempster [21] was performed on the cadavers of middle-aged and older men, some of whom were chronically ill prior to death. As a result, the mean age (69 years) and weight (60 kilograms) are not representative. The numbers of reported here from [108] are based on a live population of 100 men and 15 women. The men were typical young adults, however the women were national athletes resulting in an biased estimates for females. A more recent study on living subjects [23] showed that significant variations existed between four subpopulations and that these variations were not well accounted for by existing models. For many applications in computer vision and computer graphics these issues are negligible, however recent work (*e.g.*, [70]) suggests that errors in body segment parameters could have a significant impact on the forces estimated and, hence, the validity of their interpretation.

Finally, the numbers presented here are a convenient starting point, but are far from the final word in models of body segment parameters. More complex models, including geometric models and linear and non-linear regression models, are reviewed in [110].

Total Body Mass and Statue The measurements of the basic parameters that include total body height (statue) and total body mass over the population of males and females is presented in Table 3. Since most other quantities in this section are normalized by these quantities, the values in Table 3 can be used to generate segment lengths and mass properties for an individual within a typical population. The values here are based on those reported in [33].

Segment Lengths The measurements of segment lengths are presented in Table 4. They have been reported as a percentage of total body height, *i.e.*, the height of the subject while standing. Segment end points are defined by joint centers or other anatomic landmarks which are defined in the glossary or in section 5.1.

Mass and Moments of Inertia Three properties of a segment are necessary to specify its mass and inertial properties: mass, location of the center of mass and the principal moments of inertia. These three measurements are reported in Table 5 for men and Table 6 for women.

The mass of a segment is reported as a percentage of the total mass of the subject. The position of the center of mass is reported as its distance from the proximal end (as defined in Table 4),

Segment	Segment Endpoints		Length (% of height)	
	Proximal	Distal	Female	Male
Head + Neck	Vertex	Cervicale	14.05	13.95
Head	Vertex	Gonion	11.54	11.68
Trunk	Cervicale	Hip Joint	35.44	34.65
Upper Trunk	Cervicale	Xiphoid process	13.14	13.91
Mid Trunk	Xiphoid process	Navel	11.83	12.38
Lower Trunk	Navel	Hip Joint	10.46	8.37
Upper Arm	Shoulder joint	Elbow joint	15.86	16.18
Forearm	Elbow joint	Wrist joint	15.23	15.45
Hand	Wrist joint	3rd Metacarpale	4.50	4.95
Thigh	Hip joint	Knee joint	21.24	24.25
Shank	Knee joint	Lateral malleolus	24.92	24.93
Foot	Heel	Toe tip	13.16	14.82

Table 4: **Segment lengths.** Segment lengths for males and females as a percentage of the total height of the person. The parameters for the trunk are given as a single segment or as a combination of 3 segments. All values are adopted from [19] (which is based on measurements of [108]). Here we express all the segment lengths as a percentage of the total height of the person rather than absolute lengths with respect to the mean state (see [19]).

measured as a percentage of the length of the segment. The center of mass is assumed to lie on the line connecting the proximal and distal ends of the segment.

The principal moments of inertia around each the segments center of mass are reported assuming that the principal axes of inertia are aligned with the natural planes of the segment. The longitudinal axis is defined as the axis connecting the proximal and distal ends of the segment. The sagittal axis is orthogonal to the longitudinal axis, and parallel to the sagittal plane defined in Figure 5 for a subject with arms at their sides, palms facing in. The transverse axis is then defined as being orthogonal to both the sagittal and longitudinal axes.

The moments of inertia about these axes are reported in terms of the radius of gyration, as is customary in biomechanics. Formally, the radius of gyration about a given axis is the distance from the axis where a point mass has the same moment of inertia about the axis. The radius of gyration, r, of an object with mass m is related to the moment of inertia I as $I = mr^2$. In Tables 5 and 6 the radius of gyration is expressed as a percentage of the segment length. For convenience, we also computed the *relative* principal moments of inertia in Tables 5 and 6, where we define the relative moments of inertia as by normalizing the true moments of inertia with respect to mass and height (squared of the height to be exact). Therefore to obtain the principal moments of inertial the unit less entries in the last 3 columns of the respective tables must be multiplied by the total mass times the square of the total height of the body (mh^2).

5.3 Dynamics

While kinematics and anthropometrics describe how the structure and geometry of the body are attached, dynamics uses both to describe how the body responds to applied forces. Motion is the result of momentum and, more importantly, the forces acting on the system. There are three main

Segment	Mass (%)	CM Position (%)	Radii of gyration Sagittal (%)	Trans. (%)	Long. (%)	Rel. Principal Moments Sagittal (%)	Trans. (%)	Long. (%)
Head + Neck	6.94	50.02	30.3	31.5	26.1	1.24	1.34	0.92
Head	6.94	59.76	36.2	37.6	31.2	1.24	1.34	0.92
Trunk	43.46	51.38	32.8	30.6	16.9	56.14	48.86	14.90
Upper Trunk	15.96	50.66	50.5	32.0	46.5	7.88	3.16	6.68
Mid Trunk	16.33	45.02	48.2	38.3	46.8	5.81	3.67	5.48
Lower Trunk	11.17	61.15	61.5	55.1	58.7	2.96	2.38	2.70
Upper Arm	2.71	57.72	28.5	26.9	15.8	0.58	0.51	0.18
Forearm	1.62	45.74	27.6	26.5	12.1	0.29	0.27	0.06
Hand	0.61	79.00	62.8	51.3	40.1	0.06	0.04	0.02
Thigh	14.16	40.95	32.9	32.9	14.9	9.01	9.01	1.85
Shank	4.33	44.59	25.5	24.9	10.3	1.75	1.67	0.29
Foot	1.37	44.15	25.7	24.5	12.4	0.20	0.18	0.05

Table 5: **Mass properties and distribution for MALEs.** Definitions of segments are given in Table 4. Segment masses are relative to a total body mass; segment center of mass (CM) positions are given in percent of the segment length from the proximal end of the segment (again see Table 4). All values (except for principal moments) are taken from [19] (which is based on measurements of [108]). The relative principal moments (normalized by the the total mass and square of the height) are computed directly from the radii of gyration.

sources of force that will be discussed below: gravity, muscles and contact. There are, of course, other sources of force which can be relevant to the modeling of human motion. For instance, wind resistance, the spring like forces of ligaments and models of the friction and damping at joints can be significant in some applications. However, these are often small relative to the forces discussed below.

Gravity Gravity, the force of the Earth on other bodies, is not a force in that it's effect on an object does not depend on the mass of an object. This was famously demonstrated when Galileo simultaneously dropped a cannon ball and a bowling ball from the Leaning Tower of Pisa and observed that they struck the ground at approximately the same time. Instead, Earth's gravity is better understood as an acceleration directly on the center of mass of an object.

However, it is more convenient to express gravity as an equivalent force which can be easily included in, *e.g.*, equation (38). If the direction of gravity is the unit vector $\hat{\mathbf{v}}$ then the equivalent force acting on an object of mass m is $gm\hat{\mathbf{v}}$ where g is the rate of gravitational acceleration, which is approximately 9.81 meters per seconds squared.

That the effects of gravity are not dependent on the mass of an object is significant. It means that models where the weight of a person is unknown can still accurately include the effects of gravity so long as segment lengths are known. Conversely, this means that, without additional information about force (*e.g.*, the magnitude of a ground reaction force), the motion of a person cannot provide information about the total mass of the person.

Segment	Mass (%)	CM Position (%)	Radii of gyration Sagittal (%)	Trans. (%)	Long. (%)	Rel. Principal Moments Sagittal (%)	Trans. (%)	Long. (%)
Head + Neck	6.68	48.41	27.1	29.5	26.1	0.97	1.15	0.90
Head	6.68	58.94	33.0	35.9	31.8	0.97	1.15	0.90
Trunk	42.57	49.64	30.7	29.2	14.7	50.39	45.59	11.55
Upper Trunk	15.45	50.50	46.6	31.4	44.9	5.79	2.63	5.38
Mid Trunk	14.65	45.12	43.3	35.4	41.5	3.84	2.57	3.53
Lower Trunk	12.47	49.20	43.3	40.2	44.4	2.56	2.20	2.69
Upper Arm	2.55	57.54	27.8	26.0	14.8	0.50	0.43	0.14
Forearm	1.38	45.59	26.1	25.7	9.4	0.22	0.21	0.03
Hand	0.56	74.74	53.1	45.4	33.5	0.03	0.02	0.01
Thigh	14.78	36.12	36.9	36.4	16.2	9.08	8.83	1.75
Shank	4.81	44.16	27.1	26.7	9.3	2.19	2.13	0.26
Foot	1.29	40.14	29.9	27.9	13.9	0.20	0.17	0.04

Table 6: **Mass properties and distribution for FEMALEs.** Definitions of segments are given in Table 4. Segment masses are relative to a total body mass; segment center of mass (CM) positions are given in percent of the segment length from the proximal end of the segment (again see Table 4). All values (except for principal moments) are taken from [19] (which is based on measurements of [108]). The relative principal moments (normalized by the the total mass and square of the height) are computed directly from the radii of gyration.

Muscles Skeletal muscles[4] are tissues which connect bones and are able to voluntarily contract and relax, inducing forces between parts of the body and, therefore, producing motion. The human body contains hundreds of muscles which allow it to produce a wide range of motions and forces. Because muscles produce force by contracting, they can only result in one direction of motion. As a result, most muscles are paired with an antagonistic muscle which operate in opposition. For instance, the contraction of the quadriceps muscle can only cause an extension of the shank (*i.e.*, straightening of the knee), in order to cause flexion (*i.e.*, the bending of the knee) the hamstring must be contracted. As a result kinematic joints are typically spanned by multiple muscles.

Some muscles, known as biarticular muscles, span more than one kinematic joint. One example is the rectus femorus which is part of the larger quadriceps muscle group. It attaches to the front of the hip bone and, spanning both the knee and hip joints, connects to tibia by way of the patella. From a purely mathematical perspective, these muscles are redundant. However, considering them in models of, *e.g.*, walking can result in more efficient locomotion with simple control strategies [20]. They are also believed to play a significant role in energy conservation in running [63].

Other interesting properties of muscles may ultimately be relevant to effectively modeling human motion. For instance, noise in neural control, limited response times, pose dependent strength and energy storage may all be important in modeling.

When considering the body to be an articulated system as described in section 5.1, muscular activity results in torques about joints. Most applications in computer vision and some in computer graphics opt to abstract away actual muscles and deal solely with joint torques which have

[4] As opposed to smooth and cardiac muscles which help make up organs.

a dimension equal to the number of joint degrees of freedom. This model is akin to placing motors at each joint, similar to some robots. Such a model is attractive and compact but is unable to exploit much of the knowledge available about muscles. In [49], joint torques are used but the passive spring-like nature of tendons and ligaments is modeled using antagonistic springs which are smoothly switched between using a sigmoid. The right level of muscle modeling for human pose tracking is not yet clear. However, more detailed models than joint torques may prove valuable.

Contact Ground contact is critical in the motion of any articulated object. With only muscles or joint torques the global position and orientation of the body is underactuated. Contact with the ground effectively fixes one segment of the body to the surface, reducing the number of degrees of freedom. More generally, contact with surfaces and objects other than the ground provides the fundamental mechanism for describing interactions with the world.

Unfortunately, contact is difficult to model for a number of reasons and the discontinuous nature of contact causes difficulties in simulation. Detecting when contacts occur results in complex algorithms for simulation and event detection (see section 4.1). The large forces exerted during contact result in stiff equations of motion which become difficult to efficiently integrate. Once in contact with the ground, the transitions between static and dynamic contact are complex and difficult to model.

Instead, approximate contact models are often used. These models are generally some form of non-linear spring. The result is a contact force which is always active but becomes negligible when the contact points are far from other surfaces. For instance, Anderson and Pandy [3] used an exponential spring with sigmoid modulated linear damping. The parameters of the ground model were fixed to those which produced stable and efficient numerical simulations. With this model they were able to producing jumping motions and, in separate work, walking motions [4], by finding muscle excitations which optimized some objective function.

Brubaker et al. [9] noted two possible contact models: a non-linear approximation similar to Anderson and Pandy [3] and root forces. Root forces are an additional three linear forces and three torques which act on the root node of the body. They fully actuate the body and, thereby, allow a full range of motion, including the effects of ground contact. Fully actuating the body also decouples the temporal dependence of forces on states by allowing any acceleration to be explained through applied forces. Unfortunately, root forces are entirely implausible and can result in unrealistic motions. In [9], a combination of the two models were used. The use of root forces allowed the parameters of the non-linear approximation, including the position and orientation of the ground, to be efficiently estimated for a motion sequence by minimizing the magnitude of root forces needed.

5.4 Models of Locomotion

Locomotion is a crucial part of human motion and has received considerable study in biomechanics and elsewhere. Despite hundreds of years of research, the means by which humans walk and run are still not fully understood. This is directly evident in the difficulty faced by roboticists and animators interested in producing efficient, physically realistic walking and running motions. However, there has been much learned which serves to guide and inspire work in modeling for human pose tracking. This section discusses several interesting characteristics of human motion and simplified models of locomotion, some of which have been successfully used in tracking.

Energetic Efficiency One common, nearly axiomatic property of biological motion generally is it's efficiency. That the human body has evolved to be able to efficiently move can be easily argued. Further when walking at a given speed, people choose a step-length which minimizes metabolic energy [107] and deviations from that step-length result in less efficient motions [17]. This has led to energy minimization as one of the standard measures in a range of applications. In computer animation, the magnitude of joint torques is often used to measure the desirability of motions when designing controllers [98] or entire motions [49, 100].

However, the magnitude of joint torques is widely understood as a poor substitute for the energy expended. Because, as explained above, joint torques are the net result of a variety of phenomena it is impossible to distinguish the passive elements (*e.g.*, tendons and ligaments) from the energy consuming muscle activations. Indeed, the exact measure of energy that may have been minimized through evolution is unclear and is clearly moderated by requirements for robust and reliable locomotion.

Using an extremely simple model of bipedal locomotion, Kuo [40] explored a range of energy measures in order to find one which predicted the relationship between speed and step-length observed in walking. Also focused on walking, Anderson and Pandy [4] used a model which explicitly included muscles, tendons and ligaments, and thereby allowed a plausible measure of metabolic energy consumption. This appears to contradict the assertion by Liu et al. [49] that energy minimization is not sufficient to constrain locomotion and some style-like parameters (muscle preferences and joint stiffnesses) are necessary. However, both can be reconciled by noting that the style parameters used by Liu et al. [49] may be comparable to the more detailed model of muscles, tendons and ligaments used by Anderson and Pandy [4].

Angular Momentum Herr and Popovic [31] argue that during walking, the angular momentum of the entire body is a regulated quantity which is kept small. This is done, at least in part, by moving the arms in opposite phase with the lower body [46]. This property was successfully used by Wang et al. [98] to optimize a walking control strategy. They found that the minimization of angular momentum was critical to both realistic arm swing and a robust controller. As noted by Herr and Popovic [31], regulation of angular momentum is clearly not a property of all human motions.

Monopode Model The regulation of angular momentum in walking suggests that the gross motion of the body can be primarily understood through by the linear motion of the center of mass. In fact, the motion of the center of mass during legged locomotion of animals as diverse as cockroaches and horses can be explained by a point mass attached to a massless, retractable leg [6]. This model, called the monopode, is able to represent both walking and running and, when optimizing cost of transport, is able to discover both [83]. Seminal work in robotics showed that efficiently running (or hopping) robots, similar to monopodes, could be built using relatively simple control laws [68, 69].

Passive Dynamics Energetic efficiency remains one of the primary differences between human locomotion and the motion of humanoid robots [15]. It has long been known that much of human locomotion can be explained through *passive dynamics*, that is motion entirely driven through momentum and gravity. This was thoroughly demonstrated by a range of models which were able to walk down hill without active control strategies [55, 56, 58]. That is, these models exhibit walking motions as an inherent property of their mechanical design.

The ideas of passive dynamics have spawned a range of robots which attempt to exploit these properties [15]. However, passive dynamic motions are only able to walk on downhill surfaces and are often extremely slow. Thus the challenge is to effectively combine some form of active control with the passive dynamics.

One passive dynamic model, called the Anthropomorphic Walker, consists of two straight legs, with rounded feet and a point mass torso fixed at the hip and exists only in the sagittal plane. Kuo [41] augmented the model include a torsional spring between the legs and an impulsive push-off from the ground when the supporting leg switches, emulating toe-off. By varying the parameters of the spring and impulse, a range of walking speeds and step lengths can be found [12, 13]. Kuo [40] was also able to show that, for an appropriate definition of energy, that the preferred speed and step-length relationship observed in people could be found. Brubaker et al. [12] used this model, combined with a more complex kinematic model, to perform monocular people tracking.

6 Visual Tracking and Control

While simulation of a rigid or articulated body motion is a relatively well understood problem, and many solutions exist in the form of physical simulation engines (discussed in Section 4.2, *e.g.*, [93, 82]), the ability to control the simulation to produce desired or realistically looking human motion is still very challenging. Controllers need to be designed that allow realistic simulation of the desired motion (by properly actuating the kinematic structure). They should also be robust to changes in environment and noise. For example, in designing a control strategy that generates joint forces that cause a model to *walk*, one should also account for some tolerance to the incline on which the motion is performed (or the small variations/imperfections of the ground surface).

In general, control strategies can be classified into two paradigms: explicit and implicit. In explicit control one typically assumes that a desired motion exists (from motion capture data for example) and the controller is faced with the task of mimicking the motion (in presence of possible environmental perturbations and interactions). In the implicit control one often relies on more abstract biomechanical properties of the motion in an attempt to build a controller that preserves these abstract properties (rather then exact pose). Some potential biomechanical properties[5] of interest were discussed in Section 5.4 (*e.g.*, the maintenance of balance maintenance and the energetics of the human motion). Hybrid controllers that rely on both of these sources of information are also common.

In vision, we are typically interested in inferring the pose of the person over time. Assuming the motion is generated by some control strategy, the problem of tracking the motion over time can be formulated as one of inference over controller parameters. Below we will talk about common control strategies applicable for simulation of complex human-like motions.

6.1 Explicit control strategies (trajectory control)

6.1.1 Constraint-based controllers

Constraint-based controllers [36, 94] are among the simplest explicit control strategies, in which one assumes that desired values for all degrees of freedom[6] are given, $\mathbf{u}_d$. Given the values for

[5] Unfortunately, while biomechanical properties of simple cyclic motions (*e.g.*, walking, jogging) have been well studied, the properties of more complex motions have not been getting the same attention.

[6] For all practical purposes the degrees of freedom referred to here can be thought of as generalized coordinates discussed in Section 3.2.

all degrees of freedom of an articulated structure, a constraint-based controller formulates a set of equality constraints (one for every degree of freedom). A constraint solver is then used to solve the set of linear equality and inequality constraints that come from (1) desired kinematics, (2) the structure of the model (*e.g.*, joint constraints can also be expressed as equalities), and (3) the environment (*e.g.*, non penetration constraints take the form of inequalities) to obtain the forces that should be applied at every time. The equality motion constraints in a vector form can be written as follows,

$$\mathbf{e}(\mathbf{u}) = \mathbf{u} - \mathbf{u}_d = 0. \tag{44}$$

It is worth noting that these constraints are conceptually different from those discussed in Section 3. Unlike, for example, the constraint on the norm of the quaternion, the *motion constraints* of the controller are never truly satisfied to start with. Hence, care must be taken to ensure that if the constraint is violated, the extent to which it is violated is minimized (penalized). This can be done through a mechanism often referred to as *constraint stabilization*, where constraint takes the form,

$$\ddot{\mathbf{e}} = -\alpha\mathbf{e} - \beta\dot{\mathbf{e}}, \tag{45}$$

where α and β control the speed with which the constraint is approached. This was already discussed briefly in Section 3. A more thorough discussion of the formulation of such constraints and their implementation is given in [92].

This approach, while relatively simple, has a few key drawbacks: (1) constraint solvers are often unable to find a set of forces that are able to reproduce the motion exactly, (2) the forces that are required to reproduce the motion may by physically unrealistic, (3) the noise that is often present in the desired trajectories translates into noise in the forces and results in jerky motions. To address (2) and (3) restrictions on the allowable forces (again in terms of inequality constraints) are formulated along with the *gain* parameters that control the speed at which the desired trajectories should be followed (*e.g.*, α and β in Eq 45. The inability to find a set of forces that are able to reproduce the motion is a much harder problem in practice that requires non-trivial solutions.

Constraint-based controllers can also be formulated in terms of trajectories of points on the surface of the body, rather then trajectories of the degrees of freedom. In such cases the control takes the form of inverse dynamics [14, 94]. In such cases, it is also sometimes possible to specify a sparse set of goal trajectories (*e.g.*, for the extremities) and let the constraint solver deal with the remaining forces based on prior knowledge. Constraint-based controllers are simple in the sense that the basic functionality to implement these exists in any constraint-based simulation engines (*e.g.*, [93, 82]); implementing such control from scratch, however, is not trivial.

6.1.2 Proportional derivative controllers

An even simpler, and very common alternative to constraint-based control (particularly in the computer animation domain), is a proportional derivative (PD) controller [32, 111]. Like constraint-based control strategies, PD controllers assume the existence of the *desired trajectory* for all degrees of freedom, $\mathbf{u}_d$ (also known as the *set point* for the controller), but rely on a much simpler control rule for generating the forces (joint torques) that approximate this trajectory:

$$\tau = k_p(\mathbf{u}_d - \mathbf{u}) - k_d\dot{\mathbf{u}}, \tag{46}$$

where $\mathbf{u}$ is the current pose, $\dot{\mathbf{u}}$ is the current velocity, $\mathbf{u}_d$ is the desired pose, and k_p and k_d are constants designating the *gains* that specify how fast the desired pose should be reached and haw

fast the damping is acting on the system. This could also be thought of as a linear spring (hence the set point terminology) applied to a particular degree of freedom with an ability to decimate energy (for $k_d > 0$). Typically one needs to be careful when setting the gains, since choosing the appropriate relative gains of k_p to k_d is critical for stable simulation (an empirical rule of thumb approach is to take $k_d = 0.1k_p$). It is worth noting that the gains can, and often are, set to be different for different joints (and even degrees of freedom) to account for different response properties of joints within the body. In the very least one needs to set the these values proportional to the mass parameters of the body that the controller is acting on to ensure that the smaller bodies will not rotate at high velocities with respect to larger bodies in the system.

6.1.3 Proportional integral derivative controllers

Another popular alternative in robotics (less so in animation) is the proportional integral derivative (PID) controller [85],

$$\tau = k_p(\mathbf{u}_d - \mathbf{u}) + \frac{k_i}{\Delta_t}\int_{\Delta_t}(\mathbf{u}_d - \mathbf{u}) - k_d\dot{\mathbf{u}}, \tag{47}$$

that has a similar form to the PD controller, with the exception of the middle term that accounts for the fact that the controller should become more aggressive if the error over time between the desired and actual states increases. Like the other terms, k_i designates the gain for the new term. In general with both PD and PID controllers one must be careful to set the gains so that: (1) the desired pose is followed and (2) that the controller does not overshoot, resulting in oscillatory behavior. Since (2) is particularly problematic if the desired trajectory is noisy, it's often customary to set the gains in such a way as to slightly undershoot the desired trajectory. A good Matlab tutorial of PD/PID control is available from `http://www.engin.umich.edu/class/ctms/pid/pid.htm`.

6.1.4 Discussion and applications to human tracking in vision

Constraint-based controllers and PD controllers share some similarities in the mathematical formulation (notice the similarity between Eq. 45 and Eq 46). The key difference is that constraint-based controllers use Linear Complementarity Problem (LCP) solvers to find the actuation forces to be applied, where as PD controllers specify these forces explicitly (as a function of deviation from the desired pose). In other words, in absence of environmental interactions both can be tuned to produce similar results. In the presence of imminent environmental interactions constraint-based controllers, however, are likely to produce more robust and repeatable behavior, since the LCP solver takes those constraints into account. The PD controller, on the other hand, will produce the same actuation forces irrespective of environment or resulting interactions with it.

Since both constraint-based controllers, PD, and PID controllers rely on dense kinematic trajectories to simulate the motion, these models can easily be applied in vision as priors for human motion tracking (as in [94] for example), by simply using existing conditional kinematic models (that are abundant in vision) as goal states (set points). The controller parameters in this case (apart from gains in PD) are simply the kinematic states themselves allowing the inference to be formulated in the familiar form.

6.2 State controllers

The problem with the above controllers is that they are all only able to *track* the pre-specified trajectory and it is hard to use them to generate continuous cyclic motions. In addition, there is

no compact representation for the controllers. To address both issues state controllers have been introduced [105]. State controllers represent a given (typically cyclic motion) in terms of a few set points for PD or PID controllers. One can loosely think of this process as one of locally linearizing the typically non-linear motion [60]. In a given local region a controller with a single set point is used until the system determines that it is no longer applicable in which case it switches to the next controller. In most cases the switching is a function of the current state and triggers on a pre-specified event (*e.g.*, foot hitting the ground [105]). The switching is often tied to the contact state of the system due to the non-linearities and discontinuities that often result from such events. State space controllers have been used to generate a wide range of activities, starting from simple cyclic activities like walking [105, 106], jogging [76], and jumping [76] to complex sport motions [32] (*e.g.*, gymnastic, cycling).

The key difference from the simpler per-frame PD or PID strategy is that the set points in this sparser representation often need to be exaggerated. The goal here is not to reach the set point, but have the set point apply forces necessary to produce the desired trajectory. An analogy would be a puppeteer pulling on the strings rather than specifying where each part of the puppet should be. In doing so it becomes less intuitive how the set points should be specified, and often optimization is involved in picking the set-points for PD controllers that are effective. The optimizations involved are typically over the set points of the controllers, the gains, and sometimes even the initial conditions, resulting in complex high-dimensional and non-convex optimization problems [106]. The switching between states, however, is hard to optimize (since it typically involves a discrete event) and is most often specified by hand [105, 106, 76].

The complexity of optimizations limits the number of states a controller can have (to typically 3 or 4 for high dimensional human motions). As such, only simple atomic motions can be synthesized using a single state-space controller. To model more complex motions a common solution is to develop a family of atomic motion controllers and transition between them (in principal this is equivalent to a much larger state-space controller). Since controllers are typically quite sensitive to initial conditions doing so is not a trivial matter. One simple solution is to add a mandatory flight phase during which the transition happens (as in [76] where the user can interactively change between walking, jogging and jumping). A more formal and general solution is to train a set of *composable* controllers [25]. In [25] the notion of the controller is abstracted to include pre- and post-conditions, that specify valid input and output ranges for the atomic controllers.

6.3 Implicit Controllers

The controllers discussed thus far typically only concern themselves with generation of realistic motion, but do not account for biological and/or biomechanical mechanisms that underline these motions. In other words, while the controllers above can be designed to produce physically-valid and realistically looking motions, the actuation forces that they apply carry little resemblance to those that a human would apply. In addition, the above controllers are only capable of producing the desired motion. They do not reproduce the basic involuntary behaviors that our body is equipped with to cope with perturbations in the environment. As such, the controllers above tend to work only within a narrow operational range (though they can be optimized for stability to particular class of perturbations).

6.3.1 Balance Controllers

One of the basic biomechanical properties that one often needs to model is *balance*. When a person is walking on a moving platform he or she does not necessarily change the way her muscles

are actuating at the joints, but rather applies a general balancing strategy that alters the motion (whatever it may be) to maintain balance and prevent a fall. In other words, a good balance control strategy should be applicable to a variety of underlying motions.

Many balance controllers have been proposed over the years. Perhaps the most standard is the ZMP feedback controller often used in bi-pedal robotic design (*e.g.*, used extensively in design of Honda's Asimo robot). *Zero moment point* is defined as the point in space where dynamic reaction force at the contact of the foot with the ground (which for the purposes of ZMP analysis must be assumed to be planar and have sufficient friction to disallow slipping) does not produce any moment (inertia force equals to 0). An alternative, but closely related definition is that ZMP is defined as a point on the ground at which the net moment of inertia forces and the gravity forces have no component along the horizontal (with respect to the ground) exes [95]. The position of this point has been shown to be closely related to the stability of the bipedal motion. In other words, if the ZMP point is within the support polygon (support polygon is defined as the convex polygon encompassing all the contact points), then the motion is stable. If the ZMP is outside the support polygon then the bi-pedal mechanism is in fact falls by rotating about the edge of the support polygon.

With this notion in mind, ZMP balance controllers are tasked with keeping the ZMP point within the desired stability region (support polygon) [67]. It is unclear to what extent real human motions adhere to the ZMP control strategies. In particular, since the flight phases of human run, for example, do not contain any contact points (and thus support polygon) the balance control during those phases cannot be done using ZMP principals. Furthermore conditions of planar non-slipping ground are also too restrictive to explain balance in more general human locomotion. That said it is plausible that ZMP is one of the principals underlying sub-set of balance strategies. A detailed and formal discussion of ZMP and its properties/uses can be found in [95].

Alternatively, simple ad-hoc PD balance controllers have also been shown to produce very stable balance feedback control [105]. In [105] the target angle (fixed point of PD servo) for the torso is given by the following equation:

$$\theta_d = \theta_d 0 + c_d d + c_v v, \tag{48}$$

where $\theta_d 0$ is the original desired angle of the torso that is augmented with two terms that account for balance feedback based on the horizontal distance between the center of mass of the body and the contact point, d, and the current velocity of the center of mass, v; c_d and c_v are constants that are optimized for a given motion (walking in case of [105]). In other words, if the velocity of the center of mass of the body is large ($|v| > 0$), torso must compensate by preemptively balancing; similarly if the distance between the contact point and the center of mass is large, ($|d| > 0$), the torso must compensate as well. If the center of mass of the body is above the point of contact ($d = 0$), then the body is in static balance and no feedback control is needed.

Generic balance controllers have been shown to be very effective means of adding stability to action specific controllers discussed previously. In [105] the balance controller added remarkable stability to external perturbations and changes environment; in [60], similarly, balance controller on top of linear locally optimal controller allowed realistic adaptation to variations in the environment.

6.3.2 Biomechanically Inspired Control

Section 5.4 discussed simple biomechanical principles that have been found to underlie human locomotion. It seems plausible that these principles could be used as high level goals for optimizing controller performance to produce natural looking motions. Indeed controller parameters can be

optimized to reproduce these biomechanical properties without any reliance on motion capture data as the goal of the resulting simulation. This is a very powerful paradigm, since in essence the controller and the motion it will produce are derived from the first principles.

For example, parameters of low-dimensional biomechanically-inspired models, such as Anthropomorphic Walker, can be optimized to produce stable walking gait for a given incline and/or speed/step length [11]. These parameters can be thought of as a particularly simple control strategy.

A somewhat similar approach was taken in [98], where controllers for a considerably more complex high-dimensions model ware build. A 4-state walking controller objective was formulated that encompassed many of the biomechanically meaningful properties discussed in Section 5.4: (1) gait constraints, such as the average speed and step length and symmetry in the gait, (2) angular momentum conservation, (3) efficiency, and (4) power limits for different the joints. The resulting motions that optimized controller produces are natural and robust.

6.4 Optimization as an alternative to motion control

One popular alternative to designing the motion controllers that produce the forces necessary to generate realistically looking motions (open loop) is batch optimization. Optimization can be used to optimize kinematic motions subject to physical constraints of the world. Such methods are particularly appealing in computer graphics, where often a set of keyframes exist (or are chosen by an animator) that specify the full or partial kinematic pose of a character at specific times and the goal is to generate dense kinematic motion that matches the keyframes and is physically realistic. Alternatively, for motion adaptation and re-targeting problems, often a dense kinematic motion exists as an *exemplar*, and the goal is to modify the motion to account for the changed environment [66], temporal execution [54], or morphology (segment length and mass distribution) of the character [26], to produce physically plausible motion but under novel circumstances.

Space-time methods [100] formulate this class of problems in the form of non-linear high-dimensional constrained optimization, where the objective is to minimize the discrepancy between the specified keyframes and desired trajectory subject to constraints encoding physical plausibility (intuitively taking the form of $0 = F - ma$) and the efficiency of motion. Since the method treats physics as a (typically) soft constraint, the motion does not necessarily need to be physically realistic, but is heavily biased toward such solutions. We omit the details of the formulation here and refer the reader to [100, 66] for details. Since the optimization is over the kinematic pose trajectories, for high dimensional characters this results in high-dimensional optimization problems. One effective approach to dealing with this is by simplifying the kinematic structure of the character and solving a simpler space-time optimization first [66] (in the lower dimensional representation that preserves the essential physical properties of the motion, *e.g.*, inertial properties of the body and contact points). One also must be careful in choosing initial conditions for the optimization, since the optimization can easily converge to undesirable local optima. It is worth noting that expressive motions that can also account for stylistic variations can also be generated using this class of methods [49]. An on-line variant of such an approach was proposed in [34].

While appealing for applications in computer graphics and even perhaps for biomechanics [8], it is not clear at the moment if this class of methods is suitable for problems in computer vision tracking applications. First, space-time optimization is inherently a batch optimization method and it is unclear how the performance would degrade (or even if it can work) in an on-line tracking scenario. Second, all space-time methods so far relied on perfect but temporally (and spatially) sparse constraints in 3D; in tracking, on the other hand, one typically has to deal with dense (both temporally and spatially) but very weak 2-dimensional image constraints.

6.5 Adding uncertainty and noise modeling

Controllers generate joint forces to produce a desired motion. In practice the motion observed in the video, however, is likely to exhibit stylistic variations and noise that would differ from the predicted motion. To account for this, as is typical with filtering methods, one must assume a noise model on top of the deterministic dynamical process underlying the motion (here modeled using a physical simulation).

The noise can be accounted for in a number of ways within the framework, resulting in a very different behavior. For example, the noise can be added to the controller parameters [12, 11] under an assumption that small variations in the parameters of the controller will result in the desired motion variations observed in an image. The benefit of such an approach is that the resulting sequence of states recovered using the inference infrastructure can be simulated directly and the final inferred motion preserves physical realism. Alternatively, one can add the noise to the kinematic states that result after physical simulation is performed [94]; this is a more traditional filtering model. As a result, however, since noise is not guaranteed to follow any physical laws of motion the final inferred motion may similarly not preserve the physical aspects of the motion completely. In doing so, this approach bares some high-level similarity to space-time optimization, where physics is used to bias solutions towards more realistic interpretations, but is not guaranteed to be maintained. The benefit of the later approach is that it is typically easier to track the motion and the method is less susceptible to local optima (at the expense of some physical realism of course). A combination of two noise models is also possible and may be a viable practical alternative.

Acknowledgements. This work was financially support in part by the Canadian Institute for Advance Research (CIfAR), NSERC, and the University of Toronto. We thank Martin Delasa, Aaron Hertzmann, and Jack Wang for useful discussions on physics-based models.

A Quaternions

Quaterions are an extension of complex numbers with a long and interesting history. A full treatment of quaternions is beyond the scope of this report. Instead, they are introduced here from a practical perspective in the context of representing 3D rotations in dynamics. A quaternion $\mathbf{q}$ can be thought of as a combination of a scalar part $w \in \mathbb{R}$ and a vector part $\mathbf{u} \in \mathbb{R}^3$ and is written as $\mathbf{q} = (w, \mathbf{u}^T)^T$.

A.1 Quaternion Algebra

Quaternion addition, subtraction and multiplication by a scalar are defined in the obvious ways

$$\mathbf{q}_0 + \mathbf{q}_1 = \begin{pmatrix} w_0 + w_1 \\ \mathbf{u}_0 + \mathbf{u}_1 \end{pmatrix} \tag{49}$$

$$\mathbf{q}_0 - \mathbf{q}_1 = \begin{pmatrix} w_0 - w_1 \\ \mathbf{u}_0 - \mathbf{u}_1 \end{pmatrix} \tag{50}$$

$$a\mathbf{q}_0 = \begin{pmatrix} aw_0 \\ a\mathbf{u}_0 \end{pmatrix} \tag{51}$$

for quaternions $\mathbf{q}_0 = (w_0, \mathbf{u}_0^T)^T$, $\mathbf{q}_1 = (w_1, \mathbf{u}_1^T)^T$ and scalar a. More interestingly, multiplication of quaternions is defined as

$$\mathbf{q}_0 \circ \mathbf{q}_1 = \begin{pmatrix} w_0 w_1 - \mathbf{u}_0 \cdot \mathbf{u}_1 \\ w_0 \mathbf{u}_1 + w_1 \mathbf{u}_0 + \mathbf{u}_0 \times \mathbf{u}_1 \end{pmatrix} \tag{52}$$

where $\cdot$ and $\times$ are the usual dot and cross products in $\mathbb{R}^3$. Quaternion multiplication is non-commutative as $\mathbf{q}_0 \circ \mathbf{q}_1 \neq \mathbf{q}_1 \circ \mathbf{q}_0$ in general. However, it is associative such that $\mathbf{q}_0 \circ (\mathbf{q}_1 \circ \mathbf{q}_2) = (\mathbf{q}_0 \circ \mathbf{q}_1) \circ \mathbf{q}_2$. The conjugate of a quaternion is defined as $\mathbf{q}^* = (w, -\mathbf{u}^T)^T$ which can be used to define the multiplicative inverse

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|^2} \tag{53}$$

where $\|\mathbf{q}\| = \sqrt{w^2 + \|\mathbf{u}\|^2}$ is the usual Euclidean norm. The quaternion inverse satisifies the relation

$$\mathbf{q}^{-1} \circ \mathbf{q} = \mathbf{q} \circ \mathbf{q}^{-1} = \mathbf{1} \tag{54}$$

where $\mathbf{1} = (1, \mathbf{0}^T)^T$ is the multiplicative identity for quaternion multiplication. Finally, if $\|\mathbf{q}\| = 1$, then $\mathbf{q}^{-1} = \mathbf{q}^*$.

An alternative representation of quaternion multiplication in terms of matrix-vector products can be useful in algebraic manipulations. Treating a quaternion $\mathbf{q}$ as a vector in $\mathbb{R}^4$ with $\mathbf{q} = (w, x, y, z)^T$, then quaternion multiplication can be written as

$$\begin{aligned} \mathbf{q}_0 \circ \mathbf{q}_1 &= \mathbf{Q}(\mathbf{q}_0)\mathbf{q}_1 & (55) \\ &= \bar{\mathbf{Q}}(\mathbf{q}_1)\mathbf{q}_0 & (56) \end{aligned}$$

where

$$\begin{aligned} \mathbf{Q}(\mathbf{q}) &= \begin{pmatrix} w & -(x,y,z) \\ (x,y,z)^T & w\mathbf{E}_3 + \mathbf{X}(x,y,z) \end{pmatrix} & (57) \\ \bar{\mathbf{Q}}(\mathbf{q}) &= \begin{pmatrix} w & -(x,y,z) \\ (x,y,z)^T & w\mathbf{E}_3 - \mathbf{X}(x,y,z) \end{pmatrix} & (58) \end{aligned}$$

are referred to as the quaternion matrices and

$$\mathbf{X}(x,y,z) = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix} \tag{59}$$

is the skew-symmetric matrix representing the cross-product $\mathbf{u} \times \mathbf{x} = \mathbf{X}(\mathbf{u})\mathbf{x}$. By the associativity of quaternion multiplication, quaternion matrices satisfy the relation $\mathbf{Q}(\mathbf{q}_0)\bar{\mathbf{Q}}(\mathbf{q}_1) = \bar{\mathbf{Q}}(\mathbf{q}_1)\mathbf{Q}(\mathbf{q}_0)$ for any pair of quaternions $\mathbf{q}_0$ and $\mathbf{q}_1$. The quaternion matrices of the conjugate quaternion are $\mathbf{Q}(\mathbf{q}^*) = \mathbf{Q}(\mathbf{q})^T$ and $\bar{\mathbf{Q}}(\mathbf{q}^*) = \bar{\mathbf{Q}}(\mathbf{q})^T$.

A.2 Unit Quaternions and Spatial Rotations

A 3D rotation of θ radians about an axis represented by the unit vector $\mathbf{v} \in \mathbb{R}^3$ can be expressed as the unit quaternion $\mathbf{q} = (\cos(\theta/2), \sin(\theta/2)\mathbf{v}^T)^T$. In fact, any unit quaternion $\mathbf{q} = (w, \mathbf{u}^T)^T$ can be thought of as a rotation of $\theta = 2\tan^{-1}\frac{\|\mathbf{u}\|}{w}$ radians about the axis $\mathbf{v} = \frac{\mathbf{u}}{\|\mathbf{u}\|}$.

The rotation of a point $\mathbf{x}'$ by a unit quaternion $\mathbf{q}$ can be computed using quaternion multiplication as:

$$\begin{pmatrix} 0 \\ \mathbf{x} \end{pmatrix} = \mathbf{q} \circ \begin{pmatrix} 0 \\ \mathbf{x}' \end{pmatrix} \circ \mathbf{q}^{-1} \tag{60}$$

where $\mathbf{x}$ is the rotated point. It follows from this that if $\mathbf{q}_0$ and $\mathbf{q}_1$ are unit quaternions representing rotations then their product, $\mathbf{q}_1 \circ \mathbf{q}_0$, is the unit quaternion that represents rotating a point by $\mathbf{q}_0$ and then by $\mathbf{q}_1$. That is, composition of rotations is equivalent to multiplication of quaternions.

Rather than using quaternion multiplication directly, it can be more efficient to compute a rotation matrix $\mathbf{R}(\mathbf{q})$. This can be done in two ways. Using the quaternion matrices

$$\begin{pmatrix} 1 & 0 \\ 0 & \mathbf{R}(\mathbf{q}) \end{pmatrix} = \mathbf{Q}(\mathbf{q})\bar{\mathbf{Q}}(\mathbf{q}^{-1}) = \mathbf{Q}(\mathbf{q})\bar{\mathbf{Q}}(\mathbf{q})^T \tag{61}$$

where the last equality is only true if $\|\mathbf{q}\| = 1$. Alternatively, using the elements of the quaternion $\mathbf{q} = (w,x,y,z)^T$ directly

$$\mathbf{R}(\mathbf{q}) = \begin{pmatrix} w^2+x^2-y^2-z^2 & 2(xy-wz) & 2(xz-wy) \\ 2(yx+wz) & w^2-x^2+y^2-z^2 & 2(yz-wx) \\ 2(zx-wy) & 2(zy+wx) & w^2-x^2-y^2+z^2 \end{pmatrix} \tag{62}$$

is the explicit form for the rotation matrix of a unit quaternion. One feature of this form is that if $\mathbf{R}(\mathbf{q})$ is used with a non-unit quaternion, then it corresponds to a rotation followed by a scaling.

Quaternions can also be used to represent rotations with less than three rotational degrees of freedom. For instance, suppose a two degree of freedom joint is required where the rotation of the joint must not spin about an axis $\mathbf{v}$. A unit quaternion $\mathbf{q} = (w,\mathbf{u}^T)^T$ represents such a rotation if and only if $\mathbf{u} \cdot \mathbf{v} = 0$. This requirement can be easily ensured by linearly reparameterizing $\mathbf{u}$ in a 2D basis orthogonal to $\mathbf{v}$. If $\mathbf{v}$ is aligned with a coordinate vector, then this is equivalent to fixing that coordinate of $\mathbf{u}$ to be zero. Altering a quaternion to represent a single degree of freedom joint is similarly straight forward.

By virtue of embedding rotations in $\mathbb{R}^4$, quaternions are able to avoid the singularites of other rotational representations. They also provide a compact and computationally efficient formula for performing rotations. However, they do suffer from some drawbacks.

First, quaternions are ambiguous as a rotation of θ about $\mathbf{v}$ or a rotation of $-\theta$ about $-\mathbf{v}$ are represented by the same unit quaternion. However, since these two rotations are effectually equivalent this ambiguity is rarely a concern. Second, the quaternions $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation. This duality is generally not problematic except when attempting to build, *e.g.*, PD controllers which treat state variables as existing in a linear space. In such cases, care must be taken to ensure that all quaternions lie in the same halfspace (flipping signs when necessary) and even then, the PD controller is not assured to take an efficient path to the target quaternion. Alternatively, quaternions can be converted into Euler angles or exponential maps where such controllers are better studied and can work better. Ideally though, alternative forms of PD control can be derived which operate explicitly on quaternions and do not suffer from this problem. Third, quaternions cannot represent rotations of magnitude larger than 2π. This complicates tasks such as measuring how many full rotations an object has undergone over a period of time. In the context of human motion this is rarely an issue as typical angular velocities are much slower than the data sampling intervals.

Finally, but most importantly, a quaternion only represents a rotation if it is of unit norm. Ensuring that a quaternion continues to have a norm of one throughout a simulation typically requires changes, both in the equations of motion and in the simulation method. These issues are discussed

further in sections 3 and 4.1 respectively. Dually, care must be taken when computing derivatives from a sequence of quaternions, *e.g.*, from motion capture data. Simple finite differences neglect the consequences of the unit norm constraints on the derivatives of quaternions.

A.3 Quaternion Dynamics

In order to use quaternions in dynamics the first step is to relate angular velocity to derivatives of quaternions. This is derived in [72] and the equations are reproduced here for convenience. If a quaternion $\mathbf{q}$ represents the rotation from the body frame to the world frame (see section 2.2) and $\dot{\mathbf{q}}$ is its derivative with respect to time, then the angular velocity in the body and world frames are

$$\begin{pmatrix} 0 \\ \boldsymbol{\omega}' \end{pmatrix} = 2\mathbf{q}^* \circ \dot{\mathbf{q}} \quad \text{or} \quad \dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \circ \begin{pmatrix} 0 \\ \boldsymbol{\omega}' \end{pmatrix} \tag{63}$$

$$\begin{pmatrix} 0 \\ \boldsymbol{\omega} \end{pmatrix} = 2\dot{\mathbf{q}} \circ \mathbf{q}^* \quad \text{or} \quad \dot{\mathbf{q}} = \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{\omega} \end{pmatrix} \circ \mathbf{q} \tag{64}$$

respectively. Differentiating these expressions with respect to time

$$\begin{pmatrix} 0 \\ \dot{\boldsymbol{\omega}}' \end{pmatrix} = 2\left(\mathbf{q}^* \circ \ddot{\mathbf{q}} + \dot{\mathbf{q}}^* \circ \dot{\mathbf{q}}\right) \quad \text{or} \quad \ddot{\mathbf{q}} = \frac{1}{2}\left(\dot{\mathbf{q}} \circ \begin{pmatrix} 0 \\ \boldsymbol{\omega}' \end{pmatrix} + \mathbf{q} \circ \begin{pmatrix} 0 \\ \dot{\boldsymbol{\omega}}' \end{pmatrix}\right) \tag{65}$$

$$\begin{pmatrix} 0 \\ \dot{\boldsymbol{\omega}} \end{pmatrix} = 2\left(\ddot{\mathbf{q}} \circ \mathbf{q}^* + \dot{\mathbf{q}} \circ \dot{\mathbf{q}}^*\right) \quad \text{or} \quad \ddot{\mathbf{q}} = \frac{1}{2}\left(\begin{pmatrix} 0 \\ \dot{\boldsymbol{\omega}} \end{pmatrix} \circ \mathbf{q} + \begin{pmatrix} 0 \\ \boldsymbol{\omega} \end{pmatrix} \circ \dot{\mathbf{q}}\right) \tag{66}$$

gives expressions which relate $\ddot{\mathbf{q}}$ with $\dot{\boldsymbol{\omega}}$ and $\dot{\boldsymbol{\omega}}'$.

Glossary

acetabulum The socket like part of the pelvic girdle which connects to the femur. Several different bones make up the surface of the acetabulum with the three main contributors being the ilium, the ischium and the pubis. 21, 22, 39

acromion The part of the scapula which connects to the end of the clavicle. It can be identified as the bony protrusion located above the glenohumeral joint. 22, 23

angular momentum ($\boldsymbol{\ell}$, $\boldsymbol{\ell}'$) The angular equivalent of linear momentum. It is related to angular velocity through the inertia tensor. Angular momentum depends on the frame of reference in which it is measured. Notationally, $\boldsymbol{\ell} = \mathbf{I}\boldsymbol{\omega}$ is measured in the world frame and $\boldsymbol{\ell}' = \mathbf{I}'\boldsymbol{\omega}'$ is measured in the body frame. 10, 12, 41

angular velocity ($\boldsymbol{\omega}$, $\boldsymbol{\omega}'$) The rate of change of orientation in space. Generally, the magnitude of an angular velocity vector specifies the instantaneous rate of rotation (*e.g.*, in radians per second) about the axis specified by the direction of the vector. Angular velocity depends on the frame of reference in which it is measured. Notationally, $\boldsymbol{\omega}$ is measured in the world frame and $\boldsymbol{\omega}'$ is measured in the body frame. 10, 39

body frame A frame of reference fixed to a moving body with the origin located at the center of mass and the axes aligned with the principal axes of inertia. 9, 16, 39–41

center of mass ($\mathbf{c}$) The center of an object as determined by its density function. It is be computed as the first moment of the mass density function $\rho(\mathbf{x})$, $\mathbf{c} = m^{-1}\int \mathbf{x}\rho(\mathbf{x})d\mathbf{x}$ where m is the total mass of the object. 8, 16, 24, 39–41

cervicale The 7th cervical vertibra of the spine. It can be identified as the bony protrusion at the base of the neck. 22, 25
clavicle The bone which connects the sternum to the scapula. More commonly known as the collar bone, it runs from the sternal notch (the indentation below the throat) to connect to the scapula above the glenohumeral joint, above the shoulder joint. 22, 23, 39, 41

distal An anatomical direction indicating the end of a segment which is furthest from the torso. *e.g.*, the knee joint is located at the distal end of the thigh. 21–24, 39–41, *See in contrast* proximal

femur The thigh bone. Its proximal end connects to the pelvis in the acetabulum to form the hip joint. The distal end joins with the patella and the tibia to form the knee joint. 21, 39, 41
force (**f**) When regarding the motion of the center of mass of a point mass or an unconstrained rigid body, force is the time derivative of linear momentum. More generally, a force is the result of an external action on a system. 10, 11, 40, 41, *see* Newton
frame of reference A coordinate frame from which motion is measured. 9, 39–41, *See for example* world frame, body frame & inertial frame

generalized coordinates (**u**) Generalized coordinates are a set of coordinates **u** such that the position and orientation of every part of a constrained system can be described as a function of **u**. For instance, the position and orientation of the pelvis plus a set of joint angles are one choice of generalized coordinates for an articulated body. 15, 16, 20
glenoid cavity The part of the scapula where the proximal end of the humerus attaches. The glenoid cavity serves as the socket for a ball-and-socket joint between the humerus and the scapula. 22, 23

humerus The primary bone of the upper arm. The proximal end connects to the glenoid cavity of the scapula through the glenohumeral joint. 22, 23, 39–41

inertia tensor (**I**, **I**′) A rank two tensor (*i.e.*, a matrix) which is the second moment of the mass density function and plays the angular counterpart to total mass. The most compact and general formula is $\mathbf{I} = \int_{\mathbf{x}} \rho(\mathbf{x})(\|\mathbf{r}(\mathbf{x})\|^2 \mathbf{E}_{3\times 3} - \mathbf{r}(\mathbf{x})\mathbf{r}(\mathbf{x})^T) d\mathbf{x}$ where $\mathbf{r}(\mathbf{x}) = \mathbf{x} - \mathbf{c}$ and **c** is the center of mass. It is important to note that the inertia tensor is dependent on the frame of reference in which it is measured, *i.e.*, the coordinate frame in which the integral is taken. Notationally, **I** is computed in the world frame and **I**′ is computed in the body frame. 8, 16, 39, 40
inertial frame Any frame of reference in which Newton's equations of motion are valid. A frame of reference in which momentum is conserved in the absence of external forces. 10, 40, 41

linear momentum (**p**) The total mass times the linear velocity of a point mass or a rigid body. In an inertial frame the time derivative of linear momentum is force. 10, 12, 39
linear velocity (**v**) The rate of change of position in space. 10, 40

mass density function (ρ) A function specifying the mass density of an object at a specific point in space. 8, 39–41

Newton (*N*) The SI unit of measure for force. The amount of force required to accelerate a one kilogram object at a rate of one meter per seconds squared, that is $1N = 1\frac{kg\ m}{s^2}$. 12, 40
Newton meter (*N m*) The SI unit of measure for torque. The result of applying one Newton of force at a point which is one meter from the center of mass and in a direction perpendicular to the direction of the center of mass. 12

principal axes of inertia A set of body fixed orthogonal axes for which the the inertia tensor is diagonalized. Together with the center of mass, the principal axes define the body frame. 8, 9, 24, 39

principal moments of inertia The diagonal entries of the inertia tensor in the body frame. Alternately, the eigenvalues of the inertia tensor, independent of reference frame. 8, 11, 24

principle of virtual work The principle that, in a constrained system, the work done by constraint forces must be zero for every system velocity which satisfies the constraints. Sometimes called d'Alembert's Principle. 14, 15

proximal An anatomical direction indicating the end of a segment which is closer to torso. *e.g.*, the hip joint is located at the proximal end of the thigh. 21–24, 39–41, *See in contrast* distal

radius One of two bones which make up the lower arm, with the other being the ulna. The proximal end of the radius connects to the humerus through a ball-and-socket joint called the humeroradial joint. Both ends of the radius connect to the ulna through the proximal and distal radioulnar joints. These pivot joints, combined with humeroradial joint, allow the distal end of the radius to rotate around the ulna. 23, 41

radius of gyration (r) A measure of the moment of inertia about an axis which is independent of total mass. The radius of gyration about a given axis can be understood as the distance from the axis where a point mass, with the same total mass, has the same moment of inertia about the axis. The radius of gyration, r, of an object with mass m is related to the moment of inertia I as $I = mr^2$. The units of the radius of gyration is length and, as such, is often reported as a percentage of the length of the segment in question. 24

scapula The bone which includes the shoulder blade and the glenoid cavity. 22, 23, 39, 40

sternum The breast bone. The sternum joins the front of the rib cage with cartilage and consists of three parts. From top to bottom is: the manubrium, the body and the xiphoid process. The manubrium attaches to the right and left clavicles to form the sternoclavicular joints. 23, 39, 41

tibia The shin bone. The tibia is the larger of the two bones making up the lower leg. The other is the fibula. The proximal end joints with the femur at the knee joint. The distal of the tibia and fibula attach to the talus to form the talocrural joint at the ankle. 21, 22, 26, 39

torque (τ, τ') The time derivative of angular momentum in an inertial frame. The spin resulting from a force applied at a point other than the center of mass, $\tau = (\mathbf{x} - \mathbf{c}) \times \mathbf{f}_e$, where $\mathbf{f}_e$ is a force applied at point $\mathbf{x}$ of a body with center of mass located at $\mathbf{c}$. Notationally, τ denotes torque measured in the world frame, and τ' denotes torque measured in the body frame. 10, 12, 40, *see* Newton meter

total mass (m) The mass of an entire object. The integral of the mass density function: $m = \int_{\mathbf{x}} \rho(\mathbf{x}) d\mathbf{x}$. 7, 16, 24, 39, 40

ulna One of two bones which make up the lower arm, with the other being the radius. The proximal end of the ulna connects to the humerus through a hinge joint called the humeroulnar joint. Both ends of the ulna connect to the radius through the proximal and distal radioulnar joints. These pivot joints, combined with humeroradial joint, allow the distal end of the radius to rotate around the ulna. 23, 40

world frame An inertial frame of reference which is statically fixed to the environment. 9, 16, 39–41

xiphoid process The bottom most part of the sternum. The xiphoid process can be identified as the downward pointing, bony protrusion from the front of the ribcage. It is often used as an anatomical landmark. 25, 41

References

[1] A. Agarwal and B. Triggs, "3d human pose from silhouettes by relavance vector regression," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2004, pp. 882–888.

[2] Ageia, "Physx." [Online]. Available: http://www.ageia.com/

[3] F. C. Anderson and M. G. Pandy, "A dynamic optimization solution for vertical jumping in three dimensions," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 2, pp. 201–231, 1999.

[4] ——, "Dynamic optimization of human walking," *Journal of Biomechanical Engineering*, vol. 123, pp. 381–390, 2001.

[5] K. Bhat, S. Seitz, J. Popovic, and P. Khosla, "Computing the physical parameters of rigid-body motion from video," in *European Conference on Computer Vision (ECCV)*, 2002.

[6] R. Blickhan and R. J. Full, "Similarity in multilegged locomotion: Bouncing like a monopode," *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, vol. 173, no. 5, pp. 509–517, Nov. 1993.

[7] A. Boeing and T. Braunl, "Evaluation of real-time physics simulation systems," in *International Conference on Computer Graphics and Interactive Techniques (Graphite)*, 2007, pp. 281–288.

[8] D. Brogan, K. Granata, and P. Sheth, "Spacetime constraints for biomechanical movements," in *Applied Modeling and Simulation*, 2002.

[9] M. Brubaker, L. Sigal, and D. Fleet, "Video-based people tracking," in *Handbook on Ambient Intelligence and Smart Environments*, H. Nakashima, H. Aghajan, and J. Augusto, Eds. Springer Verlag, November 2009.

[10] ——, "Estimating contact dynamics," in *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[11] M. A. Brubaker and D. J. Fleet, "The Kneed Walker for human pose tracking," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2008.

[12] M. A. Brubaker, D. J. Fleet, and A. Hertzmann, "Physics-based person tracking using simplified lower body dynamics," *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2007.

[13] ——, "Physics-based person tracking using the anthropomorphic walker," *International Journal of Computer Vision*, 2010.

[14] M. B. Cline, "Rigid body simulation with contact and constraints," Master's thesis, University of Texas at Austin,, 1999.

[15] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient Bipedal Robots Based on Passive-Dynamic Walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[16] S. Coros, P. Beaudoin, and M. van de Panne, "Robust task-based conrol policies for physics-based characters," in *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2009.

[17] J. E. Cotes and F. Meade, "The energy expenditure and mechanical energy demand in walking," *Ergonomics*, vol. 3, no. 2, pp. 97–119, 1960.

[18] J. Darby, B. Li, N. Costen, D. J. Fleet, and N. D. Lawrence, "Backing off: Hierarchical decomposition of activity for 3d novel pose recovery," 2009.

[19] P. de Leva, "Adjustments to zatsiorsky-seluyanov's segment inertia parameters," in *Journal of Biomechnaics*, vol. 29, no. 9, 1996, pp. 1223–1230.

[20] J. C. Dean and A. D. Kuo, "Elastic coupling of limb joints enables faster bipedal walking," *Journal of the Royal Society Interface*, vol. 6, no. 35, pp. 561–573, June 2009.

[21] W. T. Dempster, "Space requirements of the seated operator: Geometrical, kinematic, and mechanical aspects of the body with special reference to the limbs," Wright-Patterson Air Force Base 55-159, Tech. Rep., 1955.

[22] J. R. Dormand, *Numerical Methods for Differential Equations*. CRC Press, 1996.

[23] J. L. Durkin and J. J. Dowling, "Analysis of body segment parameter differences between four human populations and the estimation errors of four popular mathematical models," *Journal of Biomechanical Engineering*, vol. 125, pp. 515–522, August 2003.

[24] A. Elgammal and C.-S. Lee, "Inferring 3d body pose from silhouettes using activity manifold learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[25] P. Faloutsos, M. van de Panne, and D. Terzopoulos, "Composable controllers for physics-based character animation," in *ACM Transactions on Graphics (ACM SIGGRAPH)*, 2001.

[26] M. Gleicher, "Retargeting motion to new characters," in *ACM Transactions on Graphics (ACM SIGGRAPH)*, 1998.

[27] H. Goldstein, C. P. Poole, and J. L. Safko, *Classical Mechanics*, 3rd ed. Addison Wesley, 2001.

[28] G. Grabner and A. Kecskeméthy, "An integrated Runge-Kutta root finding method for reliable collision detection in multibody systems," *Multibody System Dynamics*, vol. 14, pp. 301–316, 2005.

[29] F. S. Grassia, "Practical parameterization of rotations using the exponential map," *Journal of Graphics Tools*, vol. 3, no. 3, pp. 29–48, 1998.

[30] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, 2nd ed. Springer, 2006.

[31] H. Herr and M. Popovic, "Angular momentum in human walking," *Journal of Experimental Biology*, vol. 211, pp. 467–481, 2008.

[32] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien, "Animating human athletics," in *ACM Transactions on Graphics (ACM SIGGRAPH)*, 1995, pp. 71–78. [Online]. Available: http://www.cc.gatech.edu/gvu/animation/papers/sig95.pdf

[33] R. Huston, *Principles of Biomechanics*. CRC Press, 2009.

[34] S. Jain, Y. Ye, and C. K. Liu, "Optimization-based interactive motion synthesis," in *ACM Transactions on Graphics (ACM SIGGRAPH)*, 2009.

[35] D. Kincaid and W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, 3rd ed. Brooks/Cole, 2001.

[36] E. Kokkevis, "Practical physics for articulated characters," in *Game Developers Conference*, 2004.

[37] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," in *ACM Transactions on Graphics (ACM SIGGRAPH)*, 2002.

[38] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue., "Motion planning for humanoid robots under obstacle and dynamic balance constraints," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2001.

[39] D. Kulic, D. Lee, C. Ott, and Y. Nakamura, "Incremental learning of full body motion primitives for humanoid robots," in *International Conference on Humanoid Robots*, 2008.

[40] A. D. Kuo, "A simple model of bipedal walking predicts the preferred speed–step length relationship," *Journal of Biomechanical Engineering*, vol. 123, no. 3, pp. 264–269, Jun. 2001.

[41] ——, "Energetics of actively powered locomotion using the simplest walking model," *Journal of Biomechanical Engineering*, vol. 124, pp. 113–120, February 2002.

[42] N. D. Lawrence, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, 2005.

[43] N. D. Lawrence and A. J. Moore, "Hierarchical gaussian process latent variable models," in *Proceedings of the International Conference in Machine Learning (ICML)*, 2007, pp. 481–488.

[44] R. Li, M.-H. Yang, S. Sclaroff, and T.-P. Tian, "Monocular tracking of 3d human motion with a coordinated mixture of factor analyzers," in *European Conference on Computer Vision (ECCV)*, 2006.

[45] R. Li, T.-P. Tian, S. Sclaroff, and M.-H. Yang, "3d human motion tracking with a coordinated mixture of factor analyzers," *International Journal of Computer Vision (IJCV)*, 2010.

[46] Y. Li, W. Wang, R. H. Crompton, and M. M. Gunther, "Free vertical moments and transverse forces in human walking and their role in relation to arm-swing," *Journal of Experimental Biology*, vol. 204, pp. 47–58, 2001.

[47] B. P. Library, " ." [Online]. Available: http://bullet.sourceforge.net/

[48] C. K. Liu and Z. Popovic, "Synthesis of complex dynamic character motion from simple animations," in *ACM Transactions on Graphics (SIGGRAPH)*, 2002.

[49] C. K. Liu, A. Hertzmann, and Z. Popovic, "Learning physics-based motion style with nonlinear inverse optimization," in *ACM Transactions on Graphics (ACM SIGGRAPH)*, vol. 24, no. 3, 2005, pp. 1071–1081.

[50] D. G. Lloyd and T. F. Bessier, "An emg-driven musculoskeletal model to estimate muscle forces and knee joint moment in vivo," in *Journal of Biomechanics*, vol. 36, 2003, pp. 765–776.

[51] A. Macchietto, V. Zordan, and C. Shelton, "Momentum control for balance," in *ACM Transactions on Graphics (SIGGRAPH)*, 2009.

[52] R. Mann and A. Jepson, "Toward the computational perception of action," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998, pp. 794–799.

[53] M. J. Mataric, O. C. Jenkins, A. Fod, and V. B. Zordan, "Control and imitation in humanoids," in *American Association for Artificial Intelligence (AAAI)*, 2000.

[54] J. McCann, N. S. Pollard, and S. S. Srinivasa, "Physics-based motion retiming," in *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2006.

[55] T. McGeer, "Passive dynamic walking," *International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990.

[56] ——, "Passive walking with knees," in *Proceedings of the International Conference on Robotics and Automation*, vol. 3, 1990, pp. 1640–1645.

[57] ——, "Passive bipedal running," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 240, no. 1297, pp. 107–134, 1990.

[58] ——, "Dynamics and control of bipedal locomotion," *Journal of Theoretical Biology*, vol. 163, pp. 277–314, 1993.

[59] D. Metaxas and D. Terzopoulos, "Shape and nonrigid motion estimation through physics-based synthesis," in *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 15, no. 6, June 1993, pp. 580–591.

[60] U. Muico, Y. Lee, J. Popovic, and Z. Popovic, "Contact-aware nonlinear control of dynamic characters," in *ACM Transactions on Graphics (ACM SIGGRAPH)*, 2009.

[61] S. Nakaoka, A. Nakazawa, F. Kanehiro, K. Kaneko, M. Morisawa, H. Hirukawa, and K. Ikeuchi, "Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances," in *International Journal of Robotics Research*, vol. 26, no. 8, 2007, pp. 829–844.

[62] Newton, "Game dynamics." [Online]. Available: http://www.newtondynamics.com/

[63] T. F. Novacheck, "The biomechanics of running," *Gait and Posture*, vol. 7, pp. 77–95, 1998.

[64] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert, "Learning and inference in parametric switching linear dynamic systems," in *International Conference on Computer Vision (ICCV)*, vol. 2, 2005, pp. 1161–1168.

[65] V. Pavlovic, J. M. Rehg, and J. MacCormick, "Learning switching linear models of human motion," in *Neural Information Processing Systems*, 2000.

[66] Z. Popovic and A. Witkin, "Physically based motion transformation," in *ACM Transactions on Graphics (ACM SIGGRAPH)*, 1999.

[67] V. Prahlad, G. Dip, and C. Meng-Hwee, "Disturbance rejection by online zmp compensation," in *Robotica*, vol. 26, no. 1, 2007, pp. 9–17.

[68] G. A. Pratt, "Legged robots at MIT: what's new since Raibert?" *IEEE Robotics & Automation*, vol. 7, no. 3, pp. 15–19, 2000.

[69] M. Raibert, *Legged Robots that Balance*. MIT Press, 1986.

[70] G. Rao, D. Amarantini, E. Berton, and D. Favier, "Influence of body segment' parameters estimation models on inverse dynamics solutions during gait," *Journal of Biomechanics*, vol. 39, pp. 1531–1536, 2006.

[71] D. G. E. Robertson, G. E. Caldwell, J. Hamill, G. Kamen, and S. N. Whittlesey, *Research Methods in Biomechanics*. Human Kinetics, 2004.

[72] A. L. Schwab and J. P. Meijaard, "How to draw Euler angles and utilize Euler parameters," in *Proc. of IDETC/CIE*, 2006.

[73] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter sensitive hashing," in *Proceedings of the International Conference on Computer Vision*, 2003.

[74] L. F. Shampine and S. Thompson, "Event location for ordinary differential equations," *Computers and Mathematics with Applications*, vol. 39, pp. 43–54, 2000.

[75] A. L. Sheets and M. Hubbard, "Development and understanding of gymnastics skills using multi-body dynamics and computer simulation," in *Science and Gymnastics Symposium*, 2003.

[76] T. Shiratori and J. K. Hodgins, "Accelerometer-based user interfaces for the control of a physically simulated character," in *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2008.

[77] L. Sigal and M. J. Black, "Predicting 3d people from 2d pictures," in *Conference on Articulated Motion and Deformable Objects Conference (AMDO)*, 2006, pp. 185–195.

[78] L. Sigal, R. Memisevic, and D. J. Fleet, "Shared kernel information embedding for discriminative inferene," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[79] J. Siskind, "Grounding and lexical semantics of verbs in visual perception using force dynamics and event logic," in *Journal of Artificial Intelligence Research*, vol. 15, 2001, pp. 31–90.

[80] C. Sminchisescu, A. Kanaujia, and D. Metaxas, "BM^3E: Discriminative Density Propagation for Visual Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[81] C. Sminchisescu and A. D. Jepson, "Generative modeling for continuous non-linearly embedded visual inference," in *International Conference on Machine Learning (ICML)*, 2004.

[82] R. Smith, "Open dynamics engine." [Online]. Available: http://www.ode.org/

[83] M. Srinivasan and A. Ruina, "Computer optimization of a minimal biped model discovers walking and running," *Nature*, vol. 439, no. 7072, pp. 72–75, Jan. 2006.

[84] I. Symbolic Dynamics, "Sd/fast." [Online]. Available: http://www.sdfast.com/

[85] K. K. Tan, W. Qing-Guo, and H. C. Chieh, *Advances in PID Control*. Springer-Verlag, 1999.

[86] G. Taylor, G. Hinton, and S. Roweis, "Modeling human motion using binary latent variables," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 19, 2007.

[87] S. T. Thornton and J. B. Marion, *Classical Dynamics of Particles and Systems*, 5th ed. Brooks/Cole, 2004.

[88] R. Urtasun and T. Darrell, "Local probabilistic regression for activity-independent human pose inference," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[89] R. Urtasun, D. Fleet, and P. Fua, "Monocular 3d tracking of the golf swing," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[90] R. Urtasun, D. J.Fleet, and P. Fua, "Gaussian process dynamical models for 3d people tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[91] R. Q. van der Linde and A. L. Schwab, "Lecture Notes Multibody Dynamics B, wb1413, course 1997/1998," Lab. for Engineering Mechanics, Delft Univ. of Technology, 2002. [Online]. Available: http://tam.cornell.edu/~als93/LinSch02.pdf

[92] M. Vondrak, L. Sigal, and O. C. Jenkins, *Motion Control*. IN-TECH, Vienna, 2009, ch. Dynamics and Control of Multibody Systems.

[93] M. Vondrak, "Crisis physics engine." [Online]. Available: http://crisis.sourceforge.net/

[94] M. Vondrak, L. Sigal, and C. O. Jenkins, "Physical simulation for probabilistic motion tracking," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[95] M. Vukobratovic and B. Borovac, "Zero-moment point - thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.

[96] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Multifactor gaussian process models for style-content separation," in *International Conference on Machine Learning (ICML)*, 2007, pp. 975–982.

[97] ——, "Gaussian process dynamical models for human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 283–298, Feb 2008.

[98] ——, "Optimizing walking controllers," *ACM Transactions on Graphics (ACM SIGGRAPH Asia)*, 2009.

[99] A. Witkin and D. Baraff. (2001) Physically based modelling. SIGGRAPH Course. [Online]. Available: http://www.pixar.com/companyinfo/research/pbm2001/

[100] A. Witkin and M. Kass, "Spacetime constraints," in *ACM Transactions on Graphics (ACM SIGGRAPH)*, vol. 22, August 1988, pp. 159–168.

[101] A. Witkin, M. Gleicher, and W. Welch, "Interactive dynamics," *ACM SIGGRAPH Computer Graphics*, vol. 24, no. 2, pp. 11–21, March 1990.

[102] C. R. Wren and A. Pentland, "Dynamic models of human motion," in *Automatic Face and Gesture Recognition*, 1998.

[103] K. Yamane, J. J. Kuffner, and J. K. Hodgins, "Synthesizing animations of human manipulation tasks," in *ACM Transactions on Graphics (SIGGRAPH)*, 2004.

[104] K. Yamane, Y. Fujita, and Y. Nakamura, "Estimation of physically and physiologically valid somatosensory information," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[105] K. Yin, K. Loken, and M. van de Panne, "SIMBICON: Simple biped locomotion control," in *ACM Transactions on Graphics (ACM SIGGRAPH)*, 2007.

[106] K. Yin, S. Coros, P. Beaudoin, and M. van de Panne, "Continuation methods for adapting simulated skills," in *ACM Transactions on Graphics (ACM SIGGRAPH)*, 2008.

[107] M. Y. Zarrugh, F. N. Todd, and H. J. Ralston, "Optimization of energy expenditure during level walking," *European Journal of Applied Physiology and Occupational Physiology*, vol. 33, pp. 293–306, 1974.

[108] V. M. Zatsiorsky, V. N. Seluyanov, and L. G. Chugunova, "Methods of determining mass-inertial characteristics of human body segments," in *Contemporary Problems of Biomechanics*, 1990, pp. 272–291.

[109] V. M. Zatsiorsky, *Kinematics of Human Motion*. Human Kinetics, 1998.

[110] ——, *Kinetics of Human Motion*. Human Kinetics, 2002.

[111] V. B. Zordan and J. K. Hodgins, "Tracking and modifying upper-body human motion data with dynamic simulation," in *Computer Animation and Simulation*, 1999.