Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS (2012)
J. Bender, A. Kuijper, D. W. Fellner, and É. Guérin (Editors)

Synthesizing Balancing Character Motions

Ben Kenwright

Newcastle University, School of Computing Science, Newcastle-Upon-Tyne, United Kingdom

Abstract

This paper presents a novel method for generating balancing character poses by means of a weighted inverse kinematic constraint algorithm. The weighted constraints enable us to control the order of priority so that more important conditions such as balancing can take priority over less important ones. Maintaining a balancing pose enables us to create a variety of physically accurate motions (e.g., stepping, crouching). Balancing is achieved by controlling the location of the overall centre of mass of an articulated character; while the secondary constraints generate poses from end-effectors and trajectory information to provide continuous character movement. The poses are created by taking into account physical properties of the articulated character, that include joint mass, size, strength and angular limits. We demonstrate the successfulness of our method by generating balancing postures that are used to produce controllable character motions with physically accurate properties; likewise, our method is computationally fast, flexible and straightforward to implement.

Categories and Subject Descriptors(according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Creating flexible life-like character motions that are controllable, physically accurate, and dynamic is challenging and important. Furthermore, there is a significant demand for the creation of indistinguishably realistic computer generated animations and so there has been a tremendous amount of research into the topic over the past few decades [BAPW93] [KEDM11] [YILP07] [RAHO91] [PHBA91] [RGBC96]. Moreover, humanlike characters possess a vast number of degrees of freedom (DOF) that allows them to perform a diverse range of actions. Controlling a character's pose to generate targeted actions that accomplish specific tasks (e.g., reaching, stepping) while balancing and being physically-accurate is of particular interest [BOMT96] [BABO04].

The particular area of interest is the control and generation of balancing motions for the computer graphics and robotics community; specifically how existing or generated motions can be adapted to embrace the physical-properties of a character (i.e., mass, size, strength) and enable them to appear more life-like and realistic. A typical example would be a virtual game character. Whereby, the character is commonly animated using motion capture (MOCAP) data to obtain the desired actions (e.g., walking, standing, sitting), which would then be modified using inverse kinematics (IK) so that the characters limbs appear to interact (i.e., touch) with their environment. Although, this approach produces interactive motions, they ignore the physical properties of the articulated character and focus primarily on obtaining the necessary pose that meets the end-effectors (e.g., hands and feet) constraints; hence, resulting motions can be un-natural and physically implausible.

The crucial factor for maintaining balance during a character's movement is the positional control of the overall centre of mass (COM) of the articulated skeleton. Furthermore, for leisurely motions, we can approximate that the overall COM must reside over the foot support region for it to remain upright and balanced. If balancing poses are interpolated using trajectory information with a priority-based inverse kinematic control technique, we can generate continuously balancing movements. This paper focuses specifically on generating and controlling these leisurely balancing motions (e.g., standing, walking, stretching) using the entire body's mass and joint information.

This paper demonstrates a real-time iterative IK method for controlling the biasing of equally weighted constraints to ensure more important constraints such as balancing are enforced. Furthermore, we demonstrate our method by means of numerous examples (e.g., body types, limb disabilities, diverse terrains) to show the flexibility and potential of our approach for generating physically accurate character motions with balancing properties.

© The Eurographics Association 2012.

Workshop on Virtual Reality Interaction and Physical Simulation 2012
Author: Ben Kenwright
Title: Synthesizing Balancing Character Motions
Pages: 87-96
URL: http://diglib.eg.org/EG/DL/PE/vriphys/vriphys12/087-096.pdf
DOI: 10.2312/PE/vriphys/vriphys12/087-096

1.1. Motivation

The focus of this research is aimed at moving away from traditional inflexible data-driven key-frame methods towards more novel and promising alternatives that can provide both realistic and flexible character motions in addition to being physically accurate. We explore autonomous procedural techniques that exploit physics-based approaches to create interactive, dynamic, life-like characters without the need for motion capture data.

1.2. Contribution

In summary, the main contributions of this paper are:

- Real-time weighted approximation method for biasing the priority of IK constraints
- Interpolation of end-effectors along generated trajectory paths to produce continuously balancing character animations
- The demonstration and explanation of a controllable system for correcting and generating balancing poses

2. Related Work

There has been a tremendous amount of research into the subject of creating more realistic, interactive dynamic characters. The research in this paper follows a similar path to the exciting and attractive work presented by Boulic [BOMA96] and Aydin [AYDI99] who generated realistic body postures by exploiting an IK balance control technique. The process used kinematic constraints and behavioural functions to produce controlled interactive poses for articulated characters in virtual environments.

Alternatively, there has been many novel and interesting techniques that have attempted to solve the problem of producing realistic character motions.

The research by Carvalho [CABT07] attempted to synthesize human movement by solving a constrained optimisation problem on a low-dimensional model to identify crucial movements that are fundamental to realistic motion using a prioritized IK framework.

Yamanel [YAKH04] presented an excellent paper for producing numerous task based animations that possessed human-like qualities by combining balancing constraints, IK methods, and MOCAP data.

Identifying similarities between different motions and extracting the key data that we as humans are able to perceive, and identify human-like characteristics was presented by Kruger et al. [KBAW11].

Furthermore, a straightforward method of incorporating a balancing approximation into a characters posture was presented by Phillips [PHBA91]. This was achieved by attaching an end-effector to the lower torso to offset the models total centre of mass and mimic balancing characteristics in the resulting IK solution.

On the other hand, it can be computationally expensive and challenging to find a life-like, optimal, and natural looking IK solution for a complex human figure due to the large number of degrees of freedom. For example, the model used by Phillips [PHBA91] contained 88 degrees of freedom and was not able to run at real-time frame-rates.

Kawato [KAWA99] was able to generate natural looking motions by determining the minimum amount of torque change for reaching motions.

Numerous practical and inexpensive techniques are becoming available to the general public that enable them to create full-body MOCAP data in real-time without the need for expensive custom recording equipment [KAHK11]. However, adapting and configuring these motions to control a variety of character types with different physical attributes (i.e., sizes, degrees of freedom) can be challenging.

However, traditional methods of accurately adapting existing MOCAP data to new situations and characters can be time-consuming and costly. For example, incorporating constraint conditions to modifying exiting MOCAP data was demonstrated by Rose et al. [RGBC96] who focused on blending existing motion capture data together while the work by Gleicher [GLEI97] focused on editing existing motion sequences.

Nevertheless, data-driven methods can produce highly realistic, controllable, life-like character motions. Moreover, data-driven methods give artists the greatest control and allow them to customize a characters actions and behaviours in detail. The data for these data-driven methods primarily comes from large painstaking created MOCAP libraries, that possess a vast assortment of realistic animation types (e.g., jump, dance, kick). Hence, there has been a great deal of research into analysing existing motion capture-data to try and generate new movements [ABSP07] [DAAP08] [MLPP09] [MAZS09] [TLCL10] [YAHO09]. Furthermore, to reduce the time and cost of having to recreate large libraries of motions each time a characters properties change (e.g., size, number of joints) a particular focus of work has been on converting existing large libraries of motions to compensate these changes in criteria and produce plausibly corrected motions [ARFO03] [LCRH02] [KOGP02].

Stored data has been used to bias IK solutions towards more natural and believable looking poses; this was initially demonstrated by Rose et al. [ROSC01] and was an extension of their previously related work [ROCO98].

Moreover, controllable motions need to use motion planning techniques that involve the calculation of trajectories to achieve the desired task while avoiding obstacles within the limits of the articulated character [LATO90]. Simeon et al. [SICS02] addressed the naturalness of the motion planning movement for goal driven tasks using IK methods, while focussing specifically on generating motions for a six degree of freedom arm (e.g., reaching and grasping).

Finally, it should be noted, that the software package Euphoria which is a proprietary middleware physics-based character engine by NaturalMotion [NATU12] has successfully demonstrated balancing bipeds in real-time for the videogame industry and was used in the game 'Grand Theft Auto IV' [ROCK12].

3. System Overview

We present a simple framework that focuses on the ability to generate balancing biped character poses. We accomplish this by extending a general IK scheme to include two levels of control with different priority (end-effectors control and overall body centre of mass positional control).

A state-machine logic was used to repeatedly generate viable end-effector path trajectories for simple motions (e.g., walking). The priority IK solver ensured the centre of mass constraint always took priority and remained above the support region when the end-effectors were interpolated along the paths to generate continuously balancing motions.

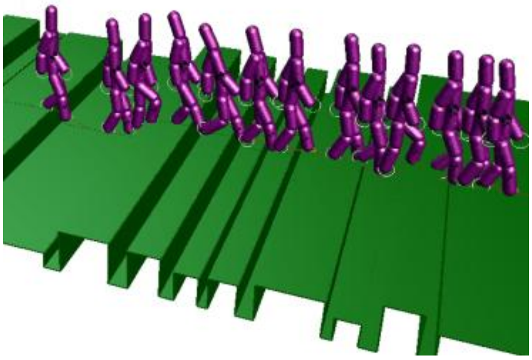


Figure 1: *Walking motion along the sagital plane shown in 3D.*

We demonstrate our approach by controlling a biped character in the sagital plane by means of diverse simulations (e.g., different body types, limb disabilities, diverse terrains). For example, Figure 1 illustrates our model taking corrective balancing steps while walking on a randomly generated terrain.

4. Balancing Characters

We approximate our character as a tree-like structure of rigid links and joints with a single root connected to the ground (see Figure 2). Essentially, the root of the tree represents the foot of the character that is supporting the majority of the weight.

Each limbs position and orientation contributes to the position of the overall centre of mass. For a static balancing posture, the overall centre of mass must reside above the supporting area (i.e., feet) in contact with the ground to remain upright. The IK problem is essentially the calculation necessary joint angles for the articulated skeleton that meets a set of constraints.

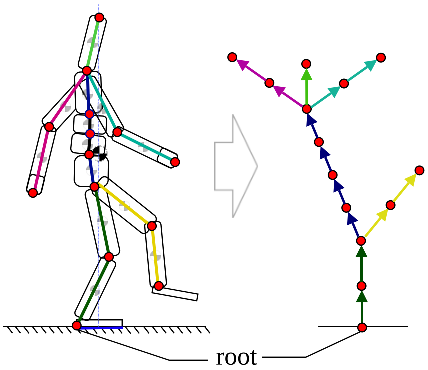


Figure 2: *The structure of an articulated character is similar to that of a tree.*

Furthermore, we can approximate that for slow leisurely movements, a character will remain balanced if their total centre of mass remains above their support region during limb movements. Whereby, we use this approximation as the basis for describing how we can create continuously balancing poses that we extend to generate constantly balancing character animations.

Figure 4 shows how an articulated character remains balanced by ensuring the body's total centre of mass remains above the feet's support region. For this reason, the character can be arranged in a finite number of static balancing poses so long as the total centre of mass remains above the feet's support region.

4.1. Articulated Character Structure

The structure of our character model consists of rigid links that are connected by angular joint (see Figure 3).

We assume that the foot representing the support region is fixed to the ground and immovable; hence, we neglect any friction or slipping.

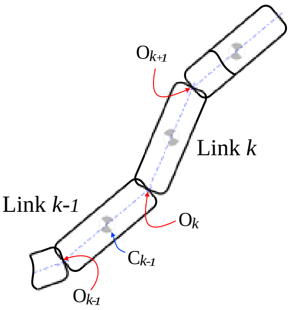


Figure 3: *Interconnected limbs naming convention.*

The IK solver has the support foot as the root. As the character transitions between poses and the support region changes (e.g., left to right foot for walking) the hierarchy is reconstructed during a single transitional step.

The goal of balancing is fed into the inverse kinematic solver. The overall centre of mass of the entire skeleton is calculated each frame. For a character to remain upright and balanced, the overall centre of mass must reside over the support region.

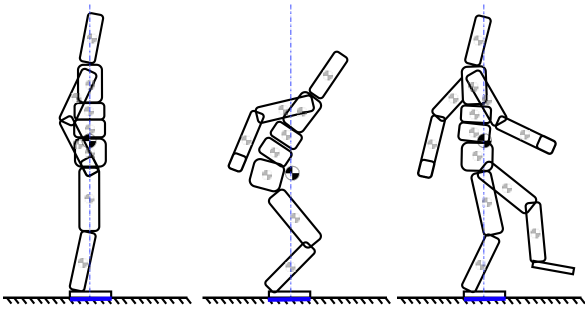


Figure 4: *Illustrating the characters total centre of mass remaining above the support region for balancing.*

We calculate how much each individual joint's change in angle changes the overall COM's position. Since each limb contributes to the overall mass, any individual limb movement will affect the characters ability to balance. Hence,

© The Eurographics Association 2012.

we need to calculate the contributing change in mass for any joint movement for calculating and determining the required balancing pose.

4.1.1. Limitations and Problems

The challenge is being able to generate controllable balancing poses that can be used to emulate a character's movements and how can we incorporate the physical characteristics (e.g., mass, size, joint limits) from the character's body into the final solution.

The trouble begins with a character possessing an infinite number of poses. It can be arduous and difficult to generate and identify analytically the differences between unrealistic and unnatural poses with that of natural human-like ones. While additional constraints can add greater control and reduce the ambiguity of the problem, it can make the solution more computationally expensive and complex. Alternatively, a lack of constraints can on occasions result in solutions oscillating between numerous poses over time due to singularities (e.g., multiple poses that solve the same balancing condition).

Nevertheless, we attempt to reduce these limiting problems by focusing initially on the physical properties of the posture and if necessary incorporate additional constraints to reduce ambiguity and produce more aesthetically pleasing poses. This can, however, require user intervention and customization that can be tedious and undesirable to ensure the generated motions are always natural looking and human-like.

4.1.2. Primary and Secondary IK Problems

4.1.3. Traditional Approach

The traditional method for resolving the IK problem into a primary and secondary priority depends upon splitting the problem into two main parts. A primary problem that has to be met and a secondary less important problem that should be met so long as it does not violate the primary solution.

The exploitation of the redundant *null space* in inverse kinematics to resolve additional tasks was first presented by Liegeois in 1977 [LIEG77]. The method decomposes the task into two parts. The first part is the pseudo-inverse component and the second part is known as the *homogeneous component.* The *homogeneous component* is projected onto the *null space* and takes second place in the solution, as shown in Equation (1).

$$\Delta\boldsymbol{\theta} = \mathbf{J}^{+}\Delta\mathbf{x} + (\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\boldsymbol{\xi} \tag{1}$$

where

- $\Delta\boldsymbol{\theta}$ is the unknown change in joint angles
- $\Delta\mathbf{x}$ is the primary task
- ξ is the secondary task
- $\mathbf{I}$ is an identity matrix
- $\mathbf{J}$ is the Jacobian matrix
- $\mathbf{J}^{+}$ is the pseudo-inverse of the Jacobian matrix J
- $(\mathbf{I} - \mathbf{J}^{+}\mathbf{J})$ is the null space of the Jacobian transform

For example, we can integrate a less important constraint condition by setting the secondary task from Equation (1) to Equation (2). Furthermore, we can cascade multiple tasks and multiple *null space* partitions to achieve a priority ordered hierarchy of goals (as shown by [BOMA96]).

$$\boldsymbol{\xi} = \mathbf{J}_{z}^{+}\Delta\mathbf{z} \tag{2}$$

In summary, the *null space* provides a vector space that has no influence on the constraint, and hence, the *null space* can be attuned to solve secondary goals or influence the solution to find a more comfortable pose.

4.2. Our Approach

Instead of splitting the IK problem into two separate parts to achieve a primary and secondary priority result, we instead formulate a single problem and integrate in a biasing factor to ensure the primary constraint gains greater precedence.

We use the Gauss-Seidel method [WILL88] to solve the single linear constraint problem, while incorporating a *bias* factor into the equation to prioritize one constraint over another. The *bias* factor is proportional to the constraint error. Whereby, we set the control of the centre of mass for balancing as the primary constraint and all other end-effectors constraints are secondary.

4.2.1. Gauss-Seidel Iterative Solver

The Gauss-Seidel method is an iterative approximation technique that we use to solve our IK constraint equations. Additionally, it can use inter-frame spatial coherency to improve speed and offers a straightforward and reliable method of solving linear systems of equations. Additionally, in practice, it has the added ability of converging on acceptable solutions even around singular problems.

The equation for the Gauss-Seidel method solves a linear problem of the form shown in Equation (3).

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3}$$

With the Jacobian constraint matrix for the articulated system we can formulate the IK problem as shown in Equation (4).

$$\mathbf{J}^{T}\mathbf{J}\Delta\boldsymbol{\theta} = \mathbf{J}^{T}\Delta\mathbf{e} \tag{4}$$

We equate the variables as shown in Equation (5).

$$\begin{aligned} \mathbf{A} &= \mathbf{J}^{T}\mathbf{J} \\ \mathbf{b} &= \mathbf{J}^{T}\Delta\mathbf{e} \\ \mathbf{x} &= \Delta\boldsymbol{\theta} = \textit{unknown} \end{aligned} \tag{5}$$

A damping coefficient is added to improve the stability by avoiding singularities, as shown in Equation (6).

$$\mathbf{A} = (\mathbf{J}^{T}\mathbf{J} + \delta\mathbf{I}) \tag{6}$$

where δ is a damping constant, typically 0.001 and $\mathbf{I}$ is an identity matrix.

4.2.2. Multiple Constraints

The Gauss-Seidel method solves a single large linear equation that represents the sum of the different constraint conditions. Furthermore, it possesses a crucial property that we desire, where the result converges on a stable solution for conditions when multiple constraints conflict with one another. In practice, the Gauss-Seidel method will converge on a compromising solution that meets all the conflicting problems halfway.

$$\begin{bmatrix} \mathbf{J}_{COM} \\ \mathbf{J}_{ENDS} \end{bmatrix} \Delta\theta = \begin{bmatrix} \Delta\mathbf{e}_{COM} \\ \Delta\mathbf{e}_{ENDS} \end{bmatrix} \tag{7}$$

where $\mathbf{J}_{COM}$ and $\mathbf{J}_{ENDS}$ are the Jacobian matrices for the centre of mass and the end-effectors, $\Delta\mathbf{e}_{COM}$ and $\Delta\mathbf{e}_{ENDS}$ are the errors between the desired and current centre of mass and end-effectors.

Furthermore, for multiple end-effectors, we extend Equation (7) to Equation (8).

$$\begin{bmatrix} \mathbf{J}_{COM} \\ \mathbf{J}_{ENDS_k} \\ \ldots \\ \mathbf{J}_{ENDS_{n-1}} \end{bmatrix} \Delta\theta = \begin{bmatrix} \Delta\mathbf{e}_{COM} \\ \Delta\mathbf{e}_{ENDS_k} \\ \ldots \\ \Delta\mathbf{e}_{ENDS_{n-1}} \end{bmatrix} \tag{8}$$

4.2.3. Primary and Secondary

The limiting factor of the Gauss-Seidel method is it has no way of prioritising one constraint over another. However, ***we propose a novel method of adding a biasing factor based on the constraint error magnitude to enable us to prioritise one constraint over another***.

The Gauss-Seidel method by default when there are two or more constraints with conflicting goals will converge on an equally weighted solution that is half way in-between (see Figure 5a).

The biasing of the primary balancing constraint is achieved using a feedback value based upon the vertical error between the whole-body characters centre of mass and the support region centre.

$$\begin{bmatrix} \mathbf{J}_{COM} \\ \mathbf{J}_{ENDS} \end{bmatrix} \Delta\theta = \begin{bmatrix} \mu\Delta\mathbf{e}_{COM} \\ \Delta\mathbf{e}_{ENDS} \end{bmatrix} \tag{9}$$

where μ is a scaling factor we introduced to ensure that the centre of mass constraint takes priority over other constraints when the constraints conflict. For example, if the reach of the hand end-effectors takes the total centre of mass out of the support region of the feet (see Figure 5b).

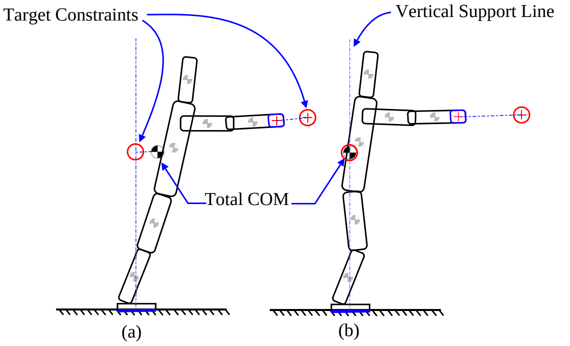


***Figure 5:** (**a**) Two constraints with equal priority (i.e., hand end-effector and COM constraint. (**b**) Biasing COM constraint error so that it takes priority.*

4.3. Jacobian

The Jacobian matrix is a powerful tool for representing how the joint angles of a character change with respect to a constraint functions coordinates. We construct a single Jacobian matrix for both the end-effectors and the total centre of mass. However, it should be noted that the Jacobian for the centre of mass constraint is highly coupled to every joint angle in the body (i.e., movement of every joint will affect the position of the overall centre of mass).

4.3.1. End-Effectors (Arms and Legs)

The Jacobian matrix for the arms and legs is calculated each frame. The Jacobian matrix elements are calculated by taking the cross product between the directional vector, that is the direction from the base of the joint to the end-effector, with the axis of rotation.

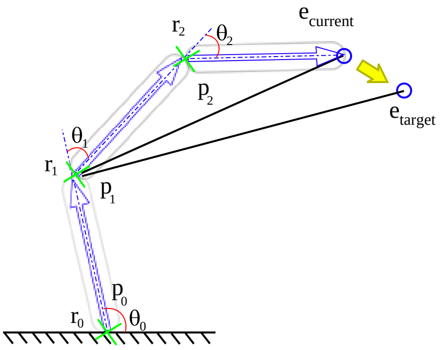


***Figure 6:** Calculating the end-effector Jacobian.*

For example, calculating the end-effector Jacobian for Figure 6 is given in Equation (10) and Equation (11).

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial\mathbf{e}}{\partial\theta_0} \\ \dfrac{\partial\mathbf{e}}{\partial\theta_1} \\ \dfrac{\partial\mathbf{e}}{\partial\theta_2} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_0 \times (\mathbf{e}_{current} - \mathbf{p}_0) \\ \mathbf{r}_1 \times (\mathbf{e}_{current} - \mathbf{p}_1) \\ \mathbf{r}_2 \times (\mathbf{e}_{current} - \mathbf{p}_2) \end{bmatrix} \tag{10}$$

and

$$\Delta\mathbf{e} = \mathbf{e}_{current} - \mathbf{e}_{target} \tag{11}$$

© The Eurographics Association 2012.

where $\mathbf{e}_{current}$, $\mathbf{e}_{target}$ represent the end-effectors current and target location, $\mathbf{r}_k$ is the axis of rotation for the joint *k*, and $\mathbf{p}_k$ represents the end position of joint *k*.

For further reading, the thesis by Baerlocher [BAER01] presents a comprehensive introduction to the technique of using inverse kinematic techniques to manipulate articulated character postures.

4.3.2. Centre of Mass (COM)

The Jacobian matrix for representing changes in joint angles with change in overall mass of a tree-like structure is more involved for a complex articulated structure. This is mostly due to each links angular displacement being dependent upon the mass and position of its child links.

The overall total centre of mass of the combined group of links is calculated using Equation (12).

$$COM_{total} = \frac{\sum_k m_k COM_k}{\sum_k m_k} \tag{12}$$

where COM_{total} represents the position of the weighted sum of all the links centre of masses, m_k is the mass of link *k*, and COM_k is the position of the centre of mass of link *k*.

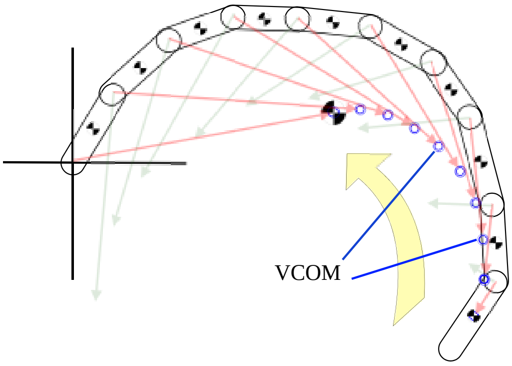


Figure 7: *Linked chain illustrating the incremental virtual COM from the end-effector inwards.*

The Jacobian matrix represents a linear relationship between how the total centre of mass will change with each joint angle. We use the principle that each joints change in angle will need to account for all the children links masses attached to itself. This method was also used by Boulic [BOMA96]. The method works by creating an incremental set of virtual centre of mass points. For example, if we start at the base of the articulated system and calculate the total centre of mass for the first link from the root, then two links from the root, then three links from the root and so on. We would create a forward set of incremental virtual centre of mass (VCOM) points that are calculated using Equation (13).

$$VCOM_k = \frac{\sum_k m_k COM_k}{\sum_k m_k} \tag{13}$$

However, we are primarily interested in the reverse set of virtual centre of mass points, which we implement by performing the incremental steps from the end-effector (i.e., leaf) nodes inwards towards the root (see Figure 7).

We can determine the change of the overall COM with change in angle by taking the cross product of the axis of rotation and the vector of the joint pivot point to the inward virtual centre of mass. Figure 13 shows the vector for the change in overall centre of mass position with change in joint angle.

$$\frac{dCOM_{total}}{d\theta_k} = r_k \times (O_k - VCOM_k) \tag{14}$$

From Equation (14) we can calculate the Jacobian matrix for the centre of gravity end-effector. Hence, this enables us to specify a desired position for the centre of gravity. For example, in 3D, Equation (15), we can specify that the centre of gravity remains above the foot support region.

$$\begin{aligned} \Delta e_{COM} &= [x_{error}, y_{error}, z_{error}] \\ &= [s_x - COM_{total\,x},\ 0,\ s_z - COM_{total\,z}] \end{aligned} \tag{15}$$

where s_x and s_z are the centre of the support regions x and z position in world coordinates. Furthermore, if we desire we can use the vertical *y* error to control if we want our character crouching or standing straight.

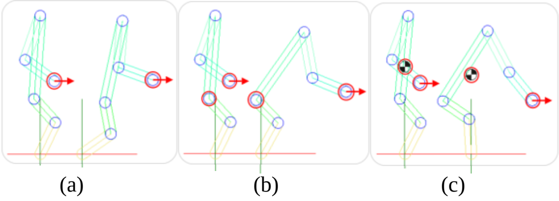

(a) (b) (c)

Figure 8: *Simulation of a low-dimensional model illustrating reaching (**a**) single end-effector for the hand, (**b**) second end-effector attached to the pelvis, (**c**) centre of mass constraint.*

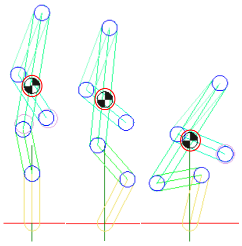


Figure 9: *Biasing the global COM downwards or upwards to make the character crouch or stand-upright.*

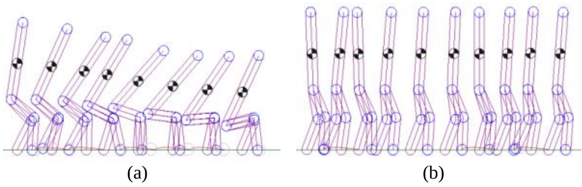

(a) (b)

Figure 10: *Walking cycle without vertical biasing (**a**) and with vertical upward biasing (**b**).*

4.4. Limitations of a Mass Only Constraint

The constraint for enforcing the characters total centre of mass remains above the support region is insufficient alone

© The Eurographics Association 2012.

to produce realistic character poses. For example, if we implement a simple state machine walking algorithm, that follows the following steps:

1. The foot that is the root of the IK is the support foot. The total COM for the body is positioned above it.
2. We then interpolate the non-support foot along a Bezier path from its current location to a new location just in front of the support foot.
3. Once the non-support foot has reached its target, we make this foot the root of the IK.
4. We then interpolate the total COM towards above the new support foot. At each point during the interpolation, we run the IK solver to generate the intermediate characters poses between frames.
5. Once the total COM is above the new root support foot, we go to step 1 and repeat.

With zero error correction to the vertical body centre of mass, the simulated walk is reasonably life-like and stable. However, in practice, the walk will eventually converge on a non-realistic crouched stepping walk due to a lack of upper body orientation constraint (see Figure 10). Adding a small upward bias error, however, remedies the crouching problem to produce a more natural upright posture.

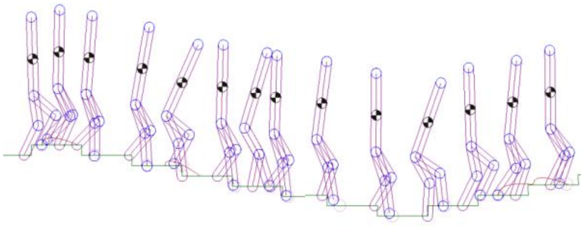


Figure 11: *Walking up and down stairs.*

4.5. Controlling Variations in Motion

We address the challenge of modelling and synthesizing various motions to mimic humans.

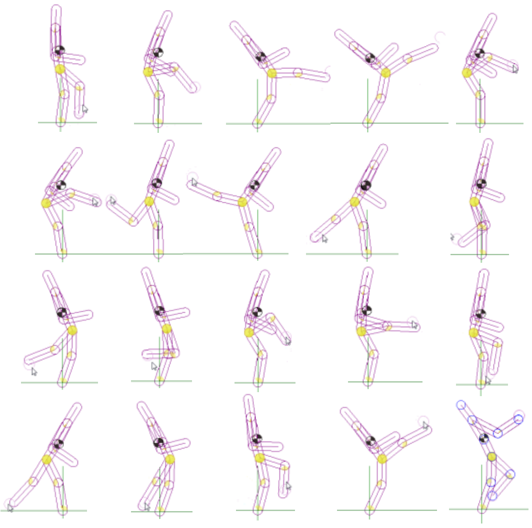


Figure 12: *Numerous balancing poses are illustrated, whereby, the total COM remains above the support foot while the non-support foot is moved around.*

Since the body centre of mass constraint ensures the final pose will always be balanced, we are free to elevate and position the non-support foot and arms to avoid obstacles without worrying about keeping upright and stable.

4.6. Practical Considerations

4.6.1. Vertical Balancing Constraint

We can inject various control factors as biasing values into the initial problem to produce a more human-like pose. For example, when we calculate the error quantity for the balancing constraint, that is the centre of mass above the support region, it a value only in the horizontal plane and does not include any vertical component. We can add a small bias quantity to the vertical error to make the solver favour a more upright or downright pose. However, the vertical upright bias must remain second in priority to the end-effectors vertical error; otherwise, it can limit or prevent the end-effectors from reaching their viable target goals.

4.6.2. Stability Enhancements

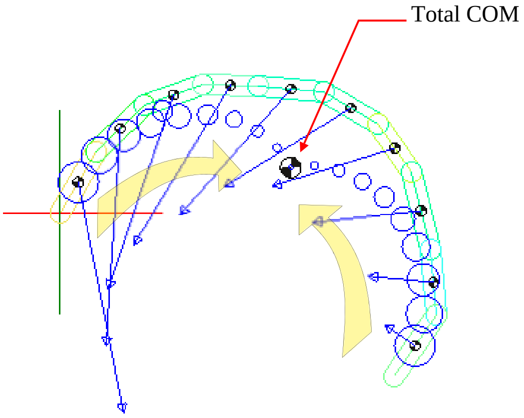
We propose incorporating additional measures to improve stability without sacrificing visual quality. Firstly, to avoid singularity problems and to produce a viable reliable solution, we incorporate a *damping* value into the Jacobian transform matrix, see Equation (6). Secondly, the incremental angular changes between iterations are clamped to acceptable limits to ensure radical sporadic jumps are prevented.

4.6.3. Centre of Mass Deviation

We can check that our theory and implementation details are correct for the deviation in overall centre of mass with deviation in joint angle. This is accomplished by iteratively rotating each individual joint angle by a miniscule quantity then recalculating the overall centre of mass of the system and recording how much the overall centre of mass changed. We illustrate the incremental virtual mass points and the direction of the change in mass with change in angle with an arrow in Figure 13.

4.7. Simulations

The simulations neglected any environment interaction (e.g., collision avoidance and moveable terrain) and focused purely on the priority controlled mass constraint for balanced motions.



© The Eurographics Association 2012.

Figure 13: *Single linked chain and its centre of mass being calculated from the base outwards and from the end inwards.*

4.7.1. Standing on One Leg

To illustrate the model in action we have the character stand on a single leg with his arms out and perform simple actions (e.g., lean forwards, lean backwards, put arms forward, move non-support leg around) (see Figure 14).

4.7.2. Walking and Stepping

We use a state-machine logic to generate the transitional walking motions for our model. At each logic state, the trajectory paths for the legs are calculated and the contact points for the ground are calculated (see Figure 15).

Hence, we could generate balancing motions that could walk on a variety of terrain (e.g., flat, sloping and stairs) without intervention.

4.7.3. Stairs and Inclined Surfaces

Since our model consistently keeps the total centre of mass above the support region, the character is able to automatically adjust its pose to compensate for non-flat terrain. Furthermore, if the terrain slowly moves as the character steps on it the posture will instinctively adjust itself to remain balanced. For example, if the character walks on a gradually declining slope the body posture will continually compensate to stay balanced (see Figure 11).

4.7.4. Un-Even Body Sizes

We explored non-uniform character mass-distributions by offsetting certain limb centre of mass locations (e.g., stomach). We also generated a range of character dimensions, (e.g., different size legs, long and thin, and short and fat body types to illustrate the flexibility of our approach (see Figure 16 and Figure 17).

4.8. Articulated Rigid Body Character

The generated poses were used to supply a straightforward physics-based model. At each frame, we extract the joint angles from the character pose to produce the necessary joint torques that would move the articulated rigid body to match the IK pose (i.e., physics-based model that ghosts the IK pose).

4.8.1. Proportional Derivative (PD) Controller

We use a similar method presented in previous work that employed a state machine logic in combination with a actuator mechanism [YILP07] [RAHO91] [SCCH09] for the rigid body joint torques. The necessary joint torques are calculated using a proportional derivative (PD) controller as shown in Equation (16).

$$\tau = k_p(\theta_d - \theta) - k_d\theta' \qquad (16)$$

where θ_d is the desired joint angle, θ and θ' represent the current joint angle and current angular velocity, and k_p and k_d represent the spring and damping constants.

The tuning of the PDs was done manually, since it is difficult to select spring damping coefficients for the controller to achieve the desired posture within a specific time reliably and safely.

4.8.2. Ghosting

The posture that was produced using the prioritized IK method is used to generate the joint torques that control the articulated character's rigid body movement. This produces a ghosting effect whereby the rigid skeleton follows the movements of the motions generated by the IK system. Furthermore, the support foot of the rigid body skeleton is fixed to the same location as the IK root.

The rigid body skeleton can respond to disturbances and prevents the skeleton from passing through other rigid bodies. However, our implementation only enabled the rigid body skeleton to ghost the postures and did not contribute to the generation of the character's motion.

5. Experiments and Results

In this paper, we demonstrated a method of generating continuously balancing character poses that could be extended to produce animated actions. Our model focused on static and slow-motion movement where the dynamic balancing properties could be ignored.

We performed a range of simulations to illustrate the advantages of our method and the potential it contains for generating motions without pre-recorded animation data.

Our simple character simulations have demonstrated the fundamental principle of our approach. Examples of such poses depicted in Figure 14, Figure 12, Figure 11, and Figure 10 illustrate the diverse balancing poses that can be interpolated to generate persistent balancing motions; alternatively, we also demonstrated how a simple state-machine logic (see Figure 17) can produce rhythmic motions such as walking without difficulty. Furthermore, the generated poses are able to provide motions that can dynamically compensate for non-level terrain such as stairs.

Additionally, we extended the 2D spatial walking motions into 3D to visualize the potential of our method (see Figure 1). The simulations were performed on an Intel Core i7-2600 CPU with 16GB of memory running 64bit Windows 7.

6. Conclusion and Discussion

This paper has shown a novel method for generating character poses with balancing properties. The generated poses were constructed using priority weighted inverse kinematic constraints. Experimentation has demonstrated the practicality of our approach and its potential for creating a diverse range of actions without the need for large motion capture libraries. Our new method was developed through constraint-based principles with a guided intuition towards creating life-like movement without key-frame data; instead, we targeted procedural logic driven trajectories and IK techniques with physical properties.

However, there are some limitations that remain unresolved with our method; e.g., uncomfortable poses can on occasion be produced, trajectory calculations are based on a specific task, the motions control a physics-based model but this physics-based model does not feedback dynamic interactive forces into the original motion.

Alternatively, an artist can tweak the constraint trajectories to ensure the final motions are more life-like; whereby, they can identify and avoid unnatural poses.

We contemplate further work for development would be the implementation of the method in 3D, and the exploration of the potential of creating balancing motions for non-human character skeletons (for example, multiple legs and arms). In addition, while the physics-based constraint solver and inverse kinematic system worked separately in our implementation they both, however, use an iterative Gauss-Seidel style solver which could be combined to obtain a two-way coupling to provide a more realistic and computationally faster solution.

Reference

[ABSP07] ABE, Y., SILVA, M.D., AND POPOVIĆ, J., "Multiobjective control with frictional contacts," *In Symposium on Computer Animation (SCA)*, pp. 249–258, 2007.

[ARFO03] ARIKAN, O., FORSYTH, D. A., AND O'BRIEN, J.F., "Motion synthesis from annotations," *ACM SIGGRAPH 2003 Papers on - SIGGRAPH '03*, vol. 22, no. 3, pp. 402–408, 2003.

[AYDI99] AYDIN, Y., "Balance control and mass centre adjustment of articulated figures in interactive environments," *The Visual Computer*, vol. 15, no. 3, pp. 113–123, May 1999.

[BAPW93] BADLER, N.I., PHILLIPS, C.B., AND WEBBER, B.L., *Simulating Humans: Computer Graphic , Animation, and Control*. Oxford University Press, 1993.

[BAER01] BAERLOCHER, P., "Inverse kinematics techniques for the interactive posture control of articulated figures," EcolePolytechnique Federale de Lausanne, 2001.

[BABO04] BAERLOCHER, P. AND BOULIC, R., "An inverse kinematics architecture enforcing an arbitrary number of strict priority levels," *The Visual Computer*, vol. 20, no. 6, pp. 402–417, Jun. 2004.

[BOMA96] BOULIC, R. AND MAS, R., "Hierarchical kinematic behaviors for complex articulated figures," *Interactive computer animation*, no. 1, pp. 1–27, 1996.

[BOMT96] BOULIC, R., MAS, R., AND THALMANN, D., "A robust approach for the control of the center of mass with inverse kinetics," *Computers graphics*, vol. 20, no. 5, pp. 693–701, 1996.

[CABT07] CARVALHO, S., BOULIC, R., AND THALMANN, D., "Interactive low-dimensional human motion synthesis by combining motion models and PIK," *Computer Animation and Virtual Worlds*, vol. 18, no. 4–5 (2007), pp. 493–503, 2007.

[DAAP08] DASILVA, M., ABE, Y., AND POPOVIĆ, J., "Interactive simulation of stylized human locomotion," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 82:1–82:10, Aug. 2008.

[GLEI97] GLEICHER, M., "Motion editing with spacetime constraints," *Proceedings of the 1997 symposium on Interactive 3D*, pp. 139–148, 1997.

[KAHK11] KAHN, S., HARTMANN, P., AND KUIJPER, A., "Towards symmetry axis based markerless motion capture," *Workshop on Virtual Reality Interaction and Physical Simulation*, no. 8, 2011.

[KAWA99] KAWATO, M., "Internal models for motor control and trajectory planning Mitsuo Kawato," *Current Opinion in Neurobiology*, vol. 9, no. 6, pp. 718–727, 1999.

[KEDM11] KENWRIGHT, B., DAVISON, R., AND MORGAN, G., "Dynamic Balancing and Walking for Real-Time 3D Characters," *Motion in Games*, pp. 63–73, 2011.

[KOGP02] KOVAR, L., GLEICHER, M., AND PIGHIN, F., "Motion graphs," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 1–10, Jul. 2002.

[KBAW11] KRÜGER, B., BAUMANN, J., ABDALLAH, M., AND WEBER, A., "A Study On Perceptual Similarity of Human Motions," *Workshop on Virtual Reality Interaction and Physical Simulation*, no. 8, 2011.

[LATO90] LATOMBE, J.C., *Robot Motion Planning*. Boston, MA: Kluwer Academic Publishers, 1990.

[LCRH02] LEE, J., CHAI, J., REITSMA, P., AND HODGINS, J., "Interactive control of avatars animated with human motion data," *SIGGRAPH '02 Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, vol. 21, no. 3 (July), pp. 491–500, 2002.

[LIEG77] LIEGEOIS, A., "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans Sys Man and Cybernetics*, no. 12, pp. 868–871, 1977.

[MAZS09] MACCHIETTO, A., ZORDAN, V., AND SHELTON, C.R., "Momentum control for balance," *ACM Transactions on Graphics*, vol. 28, no. 3 (July), p. 80, 2009.

[MLPP09] MUICO, U., LEE, Y., POPOVIĆ, J., AND POPOVIĆ, Z., "Contact-aware nonlinear control of dynamic characters," *ACM Transactions on Graphics*, vol. 28, no. 3 (July), p. 81, Jul. 2009.

[NATU12] NATURALMOTION, "Euphoria," 2012. [Online]. Available: www.naturalmotion.com.

[PHBA91] PHILLIPS, C.B. AND BADLER, N.I., "Interactive behaviors for bipedal articulated figures," *ACM SIGGRAPH Computer Graphics*, vol. 25, no. 4, pp. 359–362, Jul. 1991.

[RAHO91] RAIBERT, M.H. AND HODGINS, J.K., "Animation of dynamic legged locomotion," *ACM SIGGRAPH Computer Graphics*, vol. 25, no. 4, pp. 349–358, Jul. 1991.

[ROCO98] ROSE, C. AND COHEN, M.F., "Verbs and Adverbs: Multidimensional Motion," *IEEE Computer Graphics & Applications*, vol. 18, no. 5, pp. 32–40, 1998.

© The Eurographics Association 2012.

[RGBC96] Rose, C., Guenter, B., Bodenheimer, B., and Cohen, M.F., “Efficient generation of motion transitions using spacetime constraints,” *In Proceedings of ACM SIGGRAPH 1996*, pp. 147–154, 1996.

[ROSC01] Rose, C.F., Sloan, P.-P.J., and Cohen, M.F., “Artist-Directed Inverse-Kinematics Using Radial Basis Function Interpolation,” *Computer Graphics Forum*, vol. 20, no. 3, pp. 239–250, Sep. 2001.

[ROCK12] Rockstar Games, www.rockstargames.com/IV/, Jan 2012.

[SCCH09] Shiratori, T., Coley, B., Cham, R., and Hodgins, J.K., “Simulating balance recovery responses to trips based on biomechanical principles,” *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA ’09*, pp. 37–46, 2009.

[SICS02] Simeon, T., Cortes, J., and Sahbani, A., “A manipulation planner for pick and place operations under continuous grasps and placements,” *Conf. on Robotics and Automation*, no. May, pp. 2022–2027, 2002.

[TLCL10] Tsai, Y.-Y., Lin, W.-C., Cheng, K.B., Lee, J., and Lee, T.-Y., “Real-time physics-based 3D biped character animation using an inverted pendulum model.,” *IEEE transactions on visualization and computer graphics*, vol. 16, no. 2, pp. 325–37, 2010.

[WILL88] William, H., *Applied Numerical Linear Algebra*. Prentice Hall, 1988, p. 528.

[YAHO09] Yamane, K. and Hodgins, J., “Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data,” *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2510–2517, Oct. 2009.

[YAKH04] Yamane, K., Kuffner, J.J., and Hodgins, J.K., “Synthesizing animations of human manipulation tasks,” *ACM Transactions on Graphics*, vol. 23, no. 3, p. 532, Aug. 2004.

[YILP07] Yin, K.K., Loken, K., and van de Panne, M., “Simbicon: Simple biped locomotion control,” *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 105, 2007.

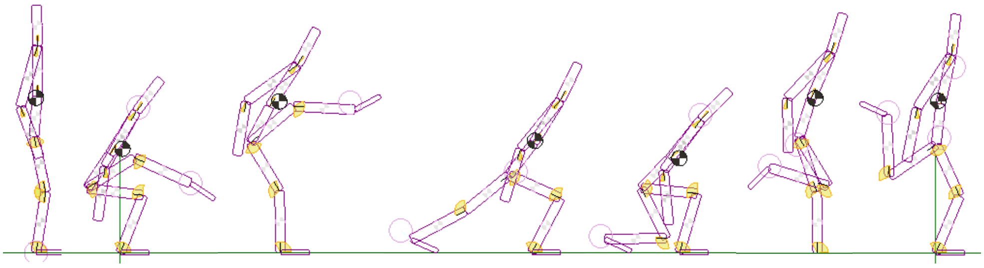


Figure 14: *Maintaining balance while holding diverse poses.*

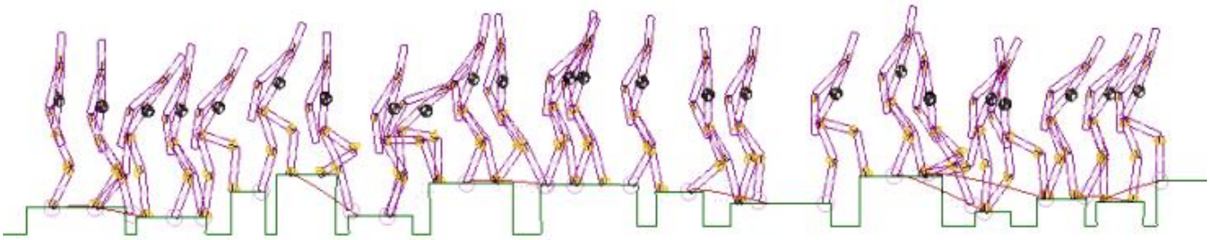


Figure 15: *Walking on diverse height terrain with selective foot placement locations.*

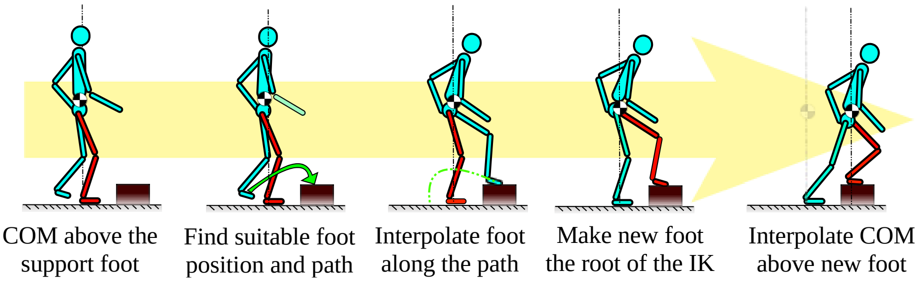


Figure 18: *State machine logic for balanced stepping and walking (from left to right).*

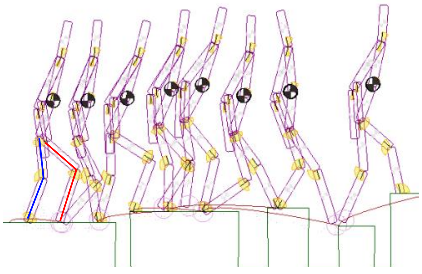


Figure 16: *Left leg is 10% longer than the right leg (hobbling).*

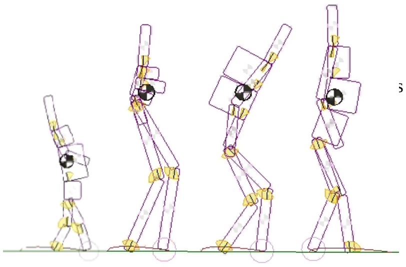


Figure 17: Modifying body proportions and mass offsets.