Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS (2011)
J. Bender, K. Erleben, and E. Galin (Editors)

Methods for Measuring State Error for Control of Animated Human Figures

G. Notman, P. A. Carlisle & S. Manning

The University of Bolton, United Kingdom

Abstract

Proportional plus Derivative (PD) control has been used widely to calculate the required forces to drive physically-based character animation. This approach requires the measurement of the state error or the difference between the measured and desired motion of the animated model. In this paper, three methods for measuring this state error are presented and compared. This includes a new method which focuses on minimising any accumulated linear transform error from rotational joints. All three methods were compared by measuring how precisely they were able to track a series of animation trajectories using a common setup. The new method presented here demonstrates improved tracking precision, particularly in cases where the kinematic dimensions of the animated model vary from those specified in the source animation.

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

Proportional plus derivative (PD) control has been used widely [HWBO95] [FvdPT01] [YLS04] [WJM06] to calculate the required forces to drive physically-based character animation. This approach requires the measurement of the state error or the difference between the measured and desired motion of a physically simulated model. In this paper, three methods for measuring this state error are presented and compared. These include the standard *Local Space Error* method, the *World Space Error* method [WJM06] and a new method called *Linear Compensation Error*. The performance of each of these methods were compared based on how precisely they were able to track a series of animation trajectories from motion capture data.

The contribution of this paper is two-fold. First a novel method for measuring the state error is described, a method which focuses on minimising linear transform error using rotational joints. Secondly, a comparison of the tracking performance of all three methods, using a common configuration, is provided.

The remainder of this paper is organised as follows: in Section 2, background is given in the area of physically-based animation and the application of PD control. Section 3 describes each of the three methods compared in this paper, with Section 4 giving details of the implementation and test configuration. Section 5 provides the resulting tracking performances, with the analysis presented in Section 6. Conclusions and future work can be found in Section 7.

2. Background

Physically-based animation applied to simulated models of human figures, promises dynamic and reactive animation which responds interactively and uniquely to the simulated environment. This approach focuses on applying forces to drive the simulated model as opposed to explicitly setting the model to each desired pose. These internal driving forces are then coupled with other applied forces, from interactions with the simulated environment and possibly other virtual objects, to define the resulting motion. One of the more difficult challenges in physically-based animation is the solving of the control forces to produce a desired motion. One popular approach is to use the PD method to help solve the control problem.

PD control has been coupled with various trajectory planning or specification methods including: State Machines

DOI: 10.2312/PE/vriphys/vriphys11/093-100

[HWBO95] [FvdPT01] [SCAF07], Pose Control Graphs [YLS04] and trajectory tracking methods [ZH02] [WJM06].

It has also been used with dynamics based methods to specify joint accelerations instead of forces [OM01]. In general PD control can be used with any trajectory planning method which defines a joint space trajectory or a trajectory which can be converted to a joint space trajectory. For example [ZH02] coupled existing joint space trajectories tuned with task space control.

2.1. Proportional Plus Derivative Control

PD control is a feedback control method which requires the measurement of the state error e (for the proportional term) and its derivative $\dot{e}$ (for the derivative term).

The standard form of the equation is:

$$\tau = k_p(\theta - q) + k_v(\dot{\theta} - \dot{q}) \tag{1}$$

Where τ is the generalised forces, and k_p & k_v the proportional and derivative gains. The error is defined as the difference between the desired θ and measured q general coordinates and their derivatives. The desired general velocity $\dot{\theta}$ is often omitted, targeting instead a zero velocity (certain motion planning methods like that of pose control graphs [YLS04] which use a sparse set of static poses, may not explicitly specific the transition velocities and so target a zero velocity).

Overly stiff tracking of motions is undesirable for physically-based character animation as the resulting motion can appear unnatural, especially during simulated interactions and contacts [YCP03] [WGF08]. Several approaches have been applied including the scaling down of the gains post impact [ZH02] [WJM06] or by using a response delay method [YCP03]. In this paper the comparison of these error measurement methods focuses on how precisely each method is able to track an animation trajectory using a common setup. More precise control allows for lower tracking gains resulting in animated character models which are more compliant and appear more natural and relaxed. Arguably, improved tracking precision is beneficial to physically-based character animation.

2.2. Regular Numbering

The simulated model of a human figured used in this paper consists of a system of rigid bodies represented by a tree, with the nodes of the tree representing the rigid bodies and the arcs the connecting joints. Using *regular numbering* [Fea07, chp. 4] the following index arrays are defined here as they are referred to throughout this paper:

$\kappa_{(i)}$ = The set of joints supporting body i
$\lambda_{(i)}$ = The parent of body i
$\nu_{(i)}$ = The set of joints supported by body i
$\mu_{(i)} \subseteq \nu_{(i)}$ = The set of supported joints connected to body i

Additionally, the following properties are useful: joint i connects body i to body $\lambda_{(i)}$, root = body b_1, and body $i \in leaf()$ if $\mu_{(i)} = \nu_{(i)} = \emptyset$.

2.3. Definition of State Error

Let θ be the vector of generalised coordinates for each joint i, and $\mathbf{q}$ the current measured coordinates. The state error $\mathbf{e}$ and its derivative $\dot{\mathbf{e}}$ are measured as:

$$\begin{aligned} \mathbf{e} &= \theta - \mathbf{q} \\ \dot{\mathbf{e}} &= \dot{\theta} - \dot{\mathbf{q}} \end{aligned} \tag{2}$$

Where the $-$ operator denotes the measurement of difference between each element i of θ and q, and is dependent on the specific representation used.

To compare variables for linear degrees of freedom the elements can be subtracted directly $\theta_i - q_i$, for any number of linear degrees of freedom of a joint. However, not all angular representations can be compared directly. It is possible to compare Euler angles, but they may suffer from aliasing when multiple rotational degrees of freedom are involved. Using quaternion representation for the rotations of joints, provides a robust method for comparing rotations for a full three rotational degrees of freedom.

There exists many different joint models and methods for defining the joint parameters. These vary from a simple single degree of freedom (DOF) minimal representation (e.g. D-H parameters [DH55]) to joints as complex as the full 6-DOF joint model defined by Featherstone [Fea07, p.79].

In this paper the complete transforms (homogeneous or equivalent) of each body b_i as a result of joint j_i are used as joint parameters. These transforms may contain both data which is constant and variable for the joint depending on the number of degrees of freedom of that joint. Although this representation is not as compact or a simple as a minimal one, it is flexible to between one and six degrees of freedom.

3. Error Methods

In this section, three methods for measuring the state error are presented and discussed. They include *Local Space Error*, *World Space Error*, and a new method: *Linear Compensation Error*.

3.1. Local Space Error

Local space error is the measurement of the difference between the vector of desired transforms θ and that of the current or measured state $\mathbf{q}$ when both are defined in a local reference frame. That is the transforms θ_i and q_i are relative transforms from the frame of the parent body $b_{\lambda(i)}$ to the frame of body b_i.

Local Space Error:

$$\begin{aligned} e_i &= {}_{\lambda(i)}\,\theta_i - {}_{\lambda(i)}\,q_i \\ \dot{e}_i &= {}_{\lambda}\,(i)\dot{\theta}_i - {}_{\lambda(i)}\,\dot{q}_i \end{aligned} \tag{3}$$

Where the subscript prefix denotes the frame of reference for which the values are specified in. With this approach the derivatives ${}_{\lambda(i)}\dot{\theta}_i$ and ${}_{\lambda(i)}\dot{q}_i$ are the measurements of the relative velocities of body $b_{\lambda(i)}$ and body b_i, and can be referred to as the velocity of joint j_i.

The drawback to measuring the state error in this relative way, is that any error from supporting joints $k_{(i)}$ cannot be accounted for. The error is measured locally and even if this is reduced to zero at the joint j_i, the transform of the body b_i may still have error due to error existing at any of the joints in $k_{(i)}$.

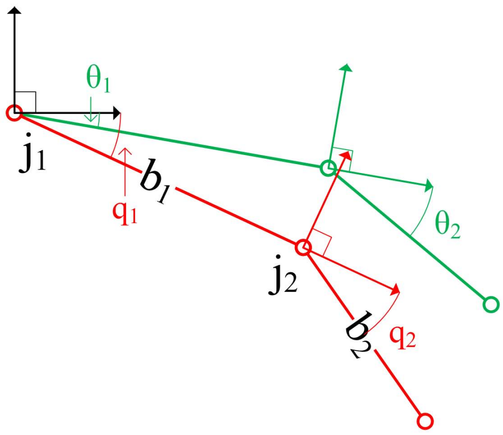

Figure 1: *The* Local Space Error *method cannot correct for error from supporting joints $k_{(i)}$. In this planar example, with* 1*-DOF rotational joints, the desired state* θ *is indicated by the green line, with the red line indicating the current state* $\mathbf{q}$*. Where j_i indicates the joint of body b_i. Observe that even though the rotational error at joint j_2 is* 0 *($q_2 = \theta_2$), due to the error from the supporting joint j_1 the final rotational transform of b_2 is different from that specified by* θ.

If instead of a general transform, a minimal representation is used, it may be difficult to measure the state error of the joints with any other method other than the *Local Space Error* method. For example, given a single degree of freedom rotational joint specified using minimal representation, there exists only a single joint variable relating to the angle of the joint. This angle defines the angle of rotation about the axis of the joint. In Fig. 1 the joint variables would relate to the angular components of θ with all the joint axes aligned as vectors perpendicular to the plane. If however, the joint axes were arbitrary and varied with each joint, it would not be possible to measure the angular error of the joints directly without referring to the axis of each joint.

3.2. World Space Error

In the *Local Space Error* method, both the desired θ and measured $\mathbf{q}$ transforms are defined relative to the parent body $b_{\lambda(i)}$. If instead of a relative transform, the elements of those two vectors define the full transform from the world frame (frame 0) to the reference frame of body b_i, then it is possible to measure the absolute or world space error.

World Space Error:

$$\begin{aligned} e_i &= {}_0\,\theta_i - {}_0\,q_i \\ \dot{e}_i &= {}_0\,\dot{\theta}_i - {}_0\,\dot{q}_i \end{aligned} \tag{4}$$

In this approach, the derivatives ${}_0\dot{\theta}_i$ and ${}_0\dot{q}_i$ are no longer the measurement of relative velocity across joint j_i but are the world space velocities of body b_i.

[WJM06] first applied this approach for physically-based character animation, and stated that the measurement of local transforms was a holdover from older mechanical sensor limitations (e.g. angular potentiometers and optical encoders). They stated that the world space method enabled improved tracking and balance performance, allowed motion transitioning without explicit blending, and simplified the tuning of the joint specific gain values. The benefits of this approach are visualised in fig. 2. It is important to note that the difference between world and local error is not simply the reference frame of which the transform is defined in. By comparing the desired and measured transformations of

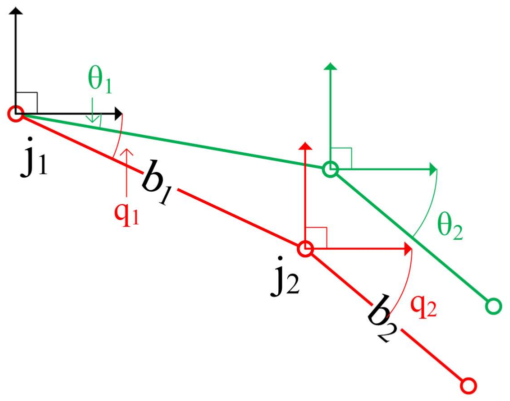

Figure 2: World Space Error *can eliminate accumulated error. The rotational transform error at joint j_2 is* 0 *($q_2 = \theta_2$), and even though there is error in the supporting joint j_1, the final rotational transform of body b_2 matches that specified by θ_2. The error in the supporting joint j_1 does result in a linear transform error in joint j_2.*

each body, specified in the reference frame 0, the resulting error includes any accumulated error from preceding joints. The *World Space Error* for joint j_i can be described as the summation of the error of the supporting joints ($k_{(i)}$) for body (b_i).

It is possible to convert from local to world space values as follows:

$$\begin{aligned} {}_0\theta_i &= {}^{0}_{\lambda(i)}A_{\lambda(i)}\theta_i \\ {}_0q_i &= {}^{0}_{\lambda(i)}B_{\lambda(i)}q_i \end{aligned} \tag{5}$$

Where ${}^{0}_{\lambda(i)}A$ is the transform from the reference frame ${}_{\lambda(i)}\theta_i$ to ${}_0\theta_i$ and ${}^{0}_{\lambda(i)}B$ the transform from the reference frame ${}_{\lambda(i)}q_i$ to ${}_0q_i$. Note that $A \neq B$ unless all error in $k_{(i)}$ is zero.

It is not possible to directly measure the error in this way if a minimal representation is used as the variable joint parameters are measured about the local joint axis defined by the constant parameters. Conversion is required to a more general form which defines the transformation between the reference frames of the two connected bodies.

This approach can be generalised as *accumulated error*, as it is not required that the compared transforms be defined in the reference frame 0. If instead a reference frame l_i is used for joint i, then the resulting error is the accumulation of transform error from the reference frame l to i. For world space error $l = 0$ and local space $l = \lambda(i)$. Providing $l_i < j_{\lambda(i)}$ this method has benefits over the *Local Space Error* method as it contains the error accumulated across more than one joint. [WJM06] used an egocentric reference frame ($l = 1$), with the transforms defined in the reference frame of the root body (b_1). This removed certain undesirable effects which can be caused by a large amount of error e_1 in the transform of the root body (b_1).

3.3. Linear Compensation Error

This new method focuses on minimising the linear transform of supported joints $\mu_{(i)}$ using a rotational joint. Observe that in Fig. 2, the rotational transform error at joint j_2 is 0 ($q_2 = \theta_2$), and that this results in the rotational transform of body b_2 which matches that specified in the transform θ_2. However, due to the rotational error present at joint j_1, the linear transform of the origin of body b_2 does not match that specified by θ_2.

This new method has been developed for use with systems that primarily contain rotational joints. This method solves for a new desired transform θ' which uses a rotation which minimises the linear transform error for joints $\mu_{(i)}$ supported by body b_i. As a rotational joint, j_i cannot itself directly minimise its own linear transform error it must rely on the rotation of the joint of the parent body $j_{\lambda(i)}$ to minimise its linear transform error.

First the transforms of the elements of θ & $\mathbf{q}$ are decomposed into their rotational R and linear p components. Given a distance d_i along the z-axis of the frame of body b_i, and the world space desired, and measured linear transforms of body b_i, ${}_0\theta_{(p)i}$ & ${}_0q_{(p)i}$, the additional rotation ${}_0R_i'$ is calculated using the following:

First calculate the world space position ${}_0z_i$:

$${}_0z_i = {}^0_i\theta_i \begin{pmatrix} 0 & 0 & d_i \end{pmatrix} \tag{6}$$

Next, the two unit vectors of the directions from the desired and measured origins to the point ${}_0z_i$ are calculated:

$$r_{\theta(i)} = \frac{{}_0z_i - {}_0\theta_{(p)i}}{\left\| {}_0z_i - {}_0\theta_{(p)i} \right\|} \tag{7}$$

$$r_{q(i)} = \frac{{}_0z_i - {}_0q_{(p)i}}{\left\| {}_0z_i - {}_0q_{(p)i} \right\|} \tag{8}$$

The additional transform ${}_0R_i'$ required to rotate from ${}_0\theta_i$ to ${}_0\theta_i'$, can be calculated from the above two vectors using the axis-angle representation. Where the axis $r = r_{q(i)} \times r_{\theta(i)}$ and the angle $\alpha = r_{q(i)} \cdot r_{\theta(i)}$. This will not work If the two vectors are parallel ($r_{q(i)} \parallel r_{\theta(i)}$), but in that case no additional rotation is required.

$${}_0\theta_i' = \beta_0 R_i' + {}_0\theta_i \tag{9}$$

Where the $+$ operator denotes the concatenation of the rotations and is representation dependent. An optional weighting value β ($0 \leq \beta \leq 1$) may also be applied if one only wishes to apply a portion of the rotation of ${}_0R_i'$ to ${}_0\theta_i$.

The state error can then be calculated using the following:

$$\begin{aligned} e_i &= {}_0\theta' - {}_0q_i \\ \dot{e}_{(p)i} &= {}_0\dot{\theta}_{(p)i} - {}_0\dot{q}_{(p)i} \\ \dot{e}_{(R)i} &= {}_0\dot{\theta}_{(R)i} - {}_0\dot{q}_{(R)i} \\ &+ \dot{e}_{(p)i} \times ({}_0z_i - {}_0\theta_{(p)i})d_i^{-2} \end{aligned} \tag{10}$$

The linear and rotational error derivatives $\dot{e}_{(p)i}$ and $\dot{e}_{(R)i}$ are calculated separately. Both components are the same as the *World Space Error* derivatives with the rotational velocity having an additional term. This additional term calculates for the rotational velocity caused by error in the linear velocity of body b_i. The rotational and linear transforms of the leaf bodies are considered here to be of the highest priority due to their importance in *task* related motions and as they normally suffer from the most accumulated error. As a result, this method is applied to all the joints except the ones belonging to the leaf bodies $i \notin leaf()$. For the leaf joints the *World Space Error* method is used instead. The use of this *Linear Compensation Error* method on all the joints except leaf joints minimises the linear transform error of the leaf bodies. Finally, the use of the *World Space Error* on the leaf joints will minimise the rotational transform error of the leaf bodies.

This method also has applications in situations where the kinematic dimensions (i.e. the linear transforms between the rotational joints) of the animation trajectory vary from those

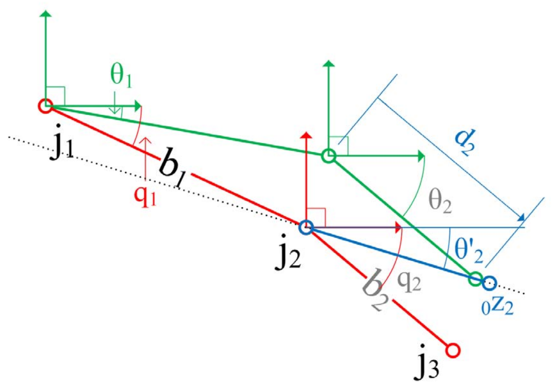

Figure 3: Linear Compensation Error *calculates a new rotation transform which minimises the linear error of the supported joints $\mu_{(i)}$. The new rotational transform for joint j_2 is calculated to minimise the linear transform error of joint j_3. The new desired transform of body b_2 is shown in blue. As the linear distance between joint j_2 and joint j_3 is fixed (j_2 is a rotational joint), the new desired transform θ'_2, if minimised, does not guarantee zero linear transform error for joint j_3. However, the targeted position $({}_0z_2)$ for joint j_3, will fall on the line which passes through the points ${}_0q_{(p)2}$ and ${}_0q_{(p)3}$ (shown as a dotted line) at a distance of d_2 from j_2.*

found in the animated model. This can be the case when animation trajectories are re-used on models which have varied dimensions.

4. Evaluation Methods

The performance of each of these three methods was tested by tracking a series of animations using a simulated model. All the properties of the simulation were identical with the only variable that of the method for calculating the state error. The performance was calculated based on the overall tracking precision of each method.

The trajectories used for these evaluations consisted of a series of punching and kicking motions (approximately 80 separate punching and 40 separate kicking actions). These animations were recorded from a single actor using a NaturalPoint Optitrack system based on six FLEX:V100r2 cameras. In total 22560 frames of animation were used which equated to 225.6 seconds of motion sampled at $100Hz$. Minimal processing was applied to the raw animation data to reduce noise. This processing consisted of applying an automatic gap filling algorithm and a $6Hz$ low pass filter.

4.1. Implementation Details

A real-time game orientated physics engine was used as the simulation environment. Its purpose was to calculate the forward dynamics of the simulated bodies, enforce constraints and to report the transforms of the simulated bodies. The simulation was run at a step frequency of $100Hz$.

The dynamic model, of a human figure, consisted of 18 linked rigid bodies. Each body was connected by a 3-DOF rotational joint. The masses and inertial properties of each body were calculated using a mathematical model based on a modified form of the Hanavan Model (Hanavan-Kwon) [Han64] [Kwo93]. In addition to the modifications by Kwon, a further modification was applied which involved splitting the upper torso body into three separate bodies to allow for the articulation of the clavicle joints. The mathematical model required a total of 41 anthropometric inputs. These measurements were taken from the same actor who performed the motion capture.

The dynamic model was controlled as a fixed-based fully actuated system using inverse dynamics coupled with PD control of the joint accelerations. This approach is similar to that used by [OM01]. The equation for which is based on the standard equation for a system of rigid bodies:

$$\tau = \mathbf{H}(\mathbf{q})\left(\ddot{\theta} + k_p\mathbf{e} + k_v\dot{\mathbf{e}}\right) + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) \qquad (11)$$

Where τ is the vector of generalised forces, $\mathbf{H}$ the mass matrix, $\mathbf{C}$ the Coriolis & centripetal and $\mathbf{G}$ the gravity term. The terms k_p & k_v refer to the proportional and derivative gains respectively. A single set of scalar gain values for k_p & k_v were used across all joints. The k_v gain value was set in relation to k_p using the critically damped ratio of $k_v = 2\sqrt{k_p}$.

The constant gain values for all joints were set at:

$$\begin{aligned} k_p &= 200 \\ k_v &= 28.284 \end{aligned} \qquad (12)$$

A vector form of The Recursive Newton Euler Algorithm [LWP80] was used to solve for the generalised forces τ. Note that the standard form of this algorithm requires state variables (q, $\dot{q}$) which are measured in the local frame of reference.

Balance was maintained using a fully-actuated 6-DOF joint connecting the root body (the pelvis) to a point in the virtual world. This fully actuated joint is similar to the root controller used in [WJM06] [KKP06]. The effect on the model's dynamics due to contact with external objects, including the ground, is not directly accounted for in the control equation (11). Therefore, to limit the possible influence of contacts, including variably timed foot contacts, the dynamic model was simulated in isolation. Support from contact with the ground was not required due to the use of the full actuated joint connected to the root.

The trajectory data consisted of a series of discrete data points or animation frames each of which specified a full set of body transforms and were evenly spaced with a frequency of 100 frames per second. These frames were directly interpolated to resolve θ and differentiated once for the trajectory velocity $\dot{\theta}$ and then once more for the acceleration $\ddot{\theta}$. A first order forward finite difference method was used with a sample spacing $0.01s$, matching that of the frequency of the animation frames. Sample spacings lower than $0.01s$ would have yielded the same values due to the interpolation method and frequency of the animation frames.

The general equation for the forward finite difference method [Wei11] is:

$$\Delta_n^k \equiv \Delta^k a_n \equiv \sum_{i=0}^{k} (-1)^i \binom{k}{i} a_{n+k-i} \tag{13}$$

The state variables **q** were measured using the reported position and orientation of the bodies in the simulation. The derivative of which $\dot{\mathbf{q}}$, was calculated using a first order backwards finite difference method. A sample spacing of $0.01s$ was also used in order to match the simulation time step. The reported velocities (linear & angular) of the bodies were not directly usable for $\dot{\mathbf{q}}$ as they were the instantaneous unconstrained velocities of the individual bodies. Due to the joint constraints acting on each body, the finite difference method provided a better estimate of the actual displacements over time.

The state error **e** was measured at every simulation time step and logged. The linear error was measured as the distance (in centimetres) between the desired and actual origins of each body. The angular error was measured as a single angle of rotation (in degrees using the axis-angle notation) between the desired and actual rotations. All measurements indicate the total transform error in frame 0. The simulations were repeated and output was precisely consistent to a minimum precision as shown in the included tabled data.

5. Results

The combined sequence of animation trajectories was tracked by each of the three methods and the *mean* and *max* error was logged, and separated into linear and angular components. The error of the joints connecting the leaf bodies are shown separately. The weighting β for the *Linear Compensation Error* method was set to 1.0 in these evaluations.

The results shown have been normalised using the *Local Space Error* method as a baseline. All the local error values were equal to 1.0 and so have been omitted. Using this normalisation allowed for a direct comparison of each of the methods. The specific numerical values may be of little value as they reflect, in part, the configuration and the tracking gains. However, the relative performance of each method is important. The numerical values of these measurements have also been included in Appendix A. For Table 1 & Table 2, $n_{all} = 406080$, $n_{leaf} = 112800$.

Table 1: *Normalised Results: Original Dimensions*

	mean		*max*	
All	Angular	Linear	Angular	Linear
World Space	0.872	0.880	0.984	0.902
Linear Comp	0.999*	0.856	0.932	1.049
Leaf				
World Space	0.846	0.818	0.984	0.902
Linear Comp	0.888	0.866	0.932	1.049

Both the linear and angular error values are bounded on a positive range with the angular component $0 \leq e_R \leq 180$ and linear component $0 \leq e_P$. The measurement of *mean* is therefore valid.

The kinematic link dimensions (body lengths and distance between joints) of the simulated model were scaled by an arbitrary factor of 1.1. This was done for the purpose of testing the *Linear Compensation Error* method when tracking trajectories where the kinematic dimensions did not match those specified in the trajectory.

Table 2: *Normalised Results: Scaled Dimensions*

	mean		*max*	
All	Angular	Linear	Angular	Linear
World Space	0.877	1.002	0.979	0.933
Linear Comp	2.516*	0.786	0.948	0.937
Leaf				
World Space	0.858	1.006	0.979	0.933
Linear Comp	0.909	0.798	0.948	0.937

*The absolute angular error measured here is expected to be larger due to the tracking of θ' instead of θ.

6. Analysis

6.1. Normal Results

Using the *Local Space Error* method as a baseline for all comparisons, the *World Space Error* method shows an improvement of between $12-13\%$ for both the angular and linear *mean* error values across all joints. Comparatively, the *Linear Compensation Error* method shows an improvement of about 14% for the *mean* linear transform error. The *mean* angular transform error of the *Linear Compensation Error* method is of little importance as the tracking of θ' across all but the leaf joints is expected to result in increased angular error when measured against θ. As for the leaf joints, both methods showed similar improvements of 15% & 11% in *mean* angular error. This is expected as the same method is used across both, to minimise angular error of the leaf joints. The *mean* linear error of leaf nodes is minimised by the supporting joints and the values reflect the unique method applied. The results show an improvement of about 18% for the *World Space Error* method, compared to an improvement of about 13% when using the *Linear Compensation Error* method.

The results show that the *max* error is a measurement from one of the leaf joints. The *World Space Error* method shows an improvement of less than 2% for the *max* angular error and about 10% for the *max* linear error. The *Linear Compensation Error* method shows an improvement of about 7% for the *max* angular error with a 5% larger *max* linear error.

6.2. Scaled Kinematics Results

In Table 2 the results are shown when tracking a trajectory with a simulated model which has been scaled by a factor of 1.1. Both methods show very similar reductions to the leaf *mean* angular error values, as those found in the normal results. However, the *mean* linear error of the *World Space Error* method, shows no improvement compared to the *Local Space Error* method. The *Linear Compensation Error* method does show an improvement of more than 21% for the *mean* linear error across all joints and more than 20% for the leaf joints. The *max* angular error for both methods is very similar to those found in the normal results. However, the improvement in the *max* linear error for both the *World Space Error* and *Linear Compensation Error* methods is reduced to less than 7%.

In the normal setup, both the *World Space Error* and *Linear Compensation Error* methods show comparable improvements to the tracking performance when compared to the *Local Space Error* method. Across all joints, the *Linear Compensation Error* showed a greater improvement to the *mean* linear error over the *World Space Error* method. However, in the case of the leaf joints, the *World Space Error* method performed slightly better. The *max* angular error was improved by both methods with the *Linear Compensation Error* method showing a larger improvement. The *Linear Compensation Error* method however, resulted in a larger *max* linear error when used with the normal setup.

In the setup where the kinematics dimensions were scaled, the *World Space Error* method was not able to improve the *mean* linear error at all. However, the *Linear Compensation Error* method showed a significant improvement in the linear *mean* error values.

7. Conclusions

A new method for measuring state error for a system containing rotational joints has been presented in this paper. The tracking performance of this new *Linear Compensation Error* method was compared against two existing methods with some improvement demonstrated. The results of the the comparisons show that both the *World Space Error* and *Linear Compensation Error* methods improved the tracking precision for both the linear and angular error components, when compared to the *Local Space Error* method. Additionally the new *Linear Compensation Error* demonstrated reduced *mean* tracking error when compared to the *World Space Error* method, except in the case of the *mean* linear error of the leaf joints.

In the test case where the kinematic dimensions were scaled by a factor of 1.1, the *World Space Error* method showed no linear tracking improvement over the *Local Space Error* method. However, the new *Linear Compensation Error* method improved the precision of the *mean* linear tracking by more than 20%.

7.1. Future Work

The test setup for tracking a trajectory with varied kinematic dimensions, used only a single modification of an arbitrary scaling factor of 1.1 applied to the simulated model. Further testing of the *Linear Compensation Error* method across a wider range of configurations may yield further insight to potential benefits of this new method. Additionally, it may be interesting to test the performance of this new method without dynamics based control, using instead local PD controllers tuned with joint specific gains.

Appendix A: Numerical Data

For both Table 3 & Table 4: $n_{all} = 406080$, $n_{leaf} = 112800$

Table 3: *Tracking Results: Normal Numerical*

	mean		*max*	
All	Angular	Linear	Angular	Linear
Local Error	1.21°	0.53*cm*	52.91°	13.49*cm*
World Error	1.05°	0.47*cm*	52.09°	12.16*cm*
Linear Comp	1.21°	0.46*cm*	49.30°	14.14*cm*
Leaf				
Local Error	1.57°	0.74*cm*	52.91°	13.49*cm*
World Error	1.33°	0.60*cm*	52.09°	12.16*cm*
Linear Comp	1.40°	0.64*cm*	49.30°	14.14*cm*

Table 4: *Tracking Results: Scaled Dimensions Numerical*

	mean		*max*	
All	Angular	Linear	Angular	Linear
Local Error	1.23°	4.26*cm*	55.31°	18.80*cm*
World Error	1.08°	4.27*cm*	54.16°	17.55*cm*
Linear Comp	3.09°	3.35*cm*	52.43°	17.62*cm*
Leaf				
Local Error	1.63°	6.89*cm*	55.31°	18.80*cm*
World Error	1.40°	6.93*cm*	54.16°	17.55*cm*
Linear Comp	1.48°	5.50*cm*	52.43°	17.62*cm*

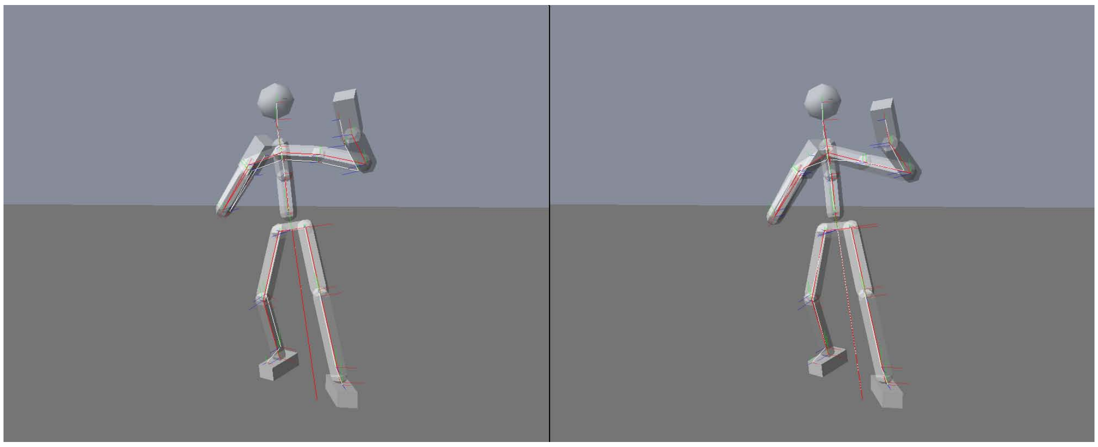

Figure 4: *In this static comparison the kinematic dimensions of the simulated model have been scaled by 1.1. On the left is the* World Space Error *method and on the right is the* Linear Compensation Error *method. The desired pose is shown in white and the actual pose drawn in red. Notice that the* World Space Error *method maintains a pose which is parallel to the desired pose. The* Linear Compensation Error *method attempts to minimise the linear transform error. The effect is is particularly noticeable at the shoulder joints. However, some linear transforms are unreachable as demonstrated by the ankle joints.*

References

[DH55] DENAVIT J., HARTENBERG R. S.: A kinematic notation for lower-pair mechanisms based on matrices. *Trans ASME J. Appl. Mech 23* (1955), 215–221. 2

[Fea07] FEATHERSTONE R.: *Rigid Body Dynamics Algorithms*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. 2

[FvdPT01] FALOUTSOS P., VAN DE PANNE M., TERZOPOULOS D.: The virtual stuntman: dynamic characters with a repertoire of autonomous motor skills. *Computers & Graphics 25*, 6 (2001), 933 – 953. Artificial Life. 1, 2

[Han64] HANAVAN E. P.: *A mathematical model of the human body*. Tech. Rep. AMRL-TR-64-102, AD-608-463, Aerospace Medical Research Laboratories, Wright-Patterson Air Force Base, Ohio, 1964. 5

[HWBO95] HODGINS J. K., WOOTEN W. L., BROGAN D. C., O'BRIEN J. F.: Animating human athletics. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1995), SIGGRAPH '95, ACM, pp. 71–78. 1, 2

[KKP06] KIM S., KIM M., PARK M.: Combining motion capture data with pd controllers for human-like animations. In *SICE-ICASE, 2006. International Joint Conference* (oct. 2006), pp. 904 –907. 5

[Kwo93] KWON Y. H.: *The effects of body segment parameter estimation on the experimental simulation of a complex airborne movement.* PhD thesis, Pennsylvania State University, 1993. 5

[LWP80] LUH J. Y. S., WALKER M. W., PAUL R. P. C.: On-line computational scheme for mechanical manipulators. *Journal of Dynamic Systems, Measurement, and Control 102*, 2 (1980), 69–76. 5

[OM01] OSHITA M., MAKINOUCHI A.: A dynamic motion control technique for human-like articulated figures. *Computer Graphics Forum 20*, 3 (2001), 192–203. 2, 5

[SCAF07] SHAPIRO A., CHU D., ALLEN B., FALOUTSOS P.: A dynamic controller toolkit. In *Proceedings of the 2007 ACM SIGGRAPH symposium on Video games* (New York, NY, USA, 2007), Sandbox '07, ACM, pp. 15–20. 2

[Wei11] WEISSTEIN E. W.: Forward difference." from mathworld–a wolfram web resource. Online:, June 2011. http://mathworld.wolfram.com/ForwardDifference.html/. 6

[WGF08] WEINSTEIN R., GUENDELMAN E., FEDKIW R.: Impulse-based control of joints and muscles. *IEEE Transactions on Visualization and Computer Graphics 14* (January 2008), 37–46. 2

[WJM06] WROTEK P., JENKINS O. C., MCGUIRE M.: Dynamo: dynamic, data-driven character control with adjustable balance. In *Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames* (New York, NY, USA, 2006), Sandbox '06, ACM, pp. 61–70. 1, 2, 3, 4, 5

[YCP03] YIN K., CLINE M. B., PAI D. K.: Motion perturbation based on simple neuromotor control models. In *In Pacific Conference on Computer Graphics and Applications (PG) (2003* (2003), pp. 445–449. 2

[YLS04] YANG P.-F., LASZLO J., SINGH K.: Layered dynamic control for interactive character swimming. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2004), SCA '04, Eurographics Association, pp. 39–47. 1, 2

[ZH02] ZORDAN V. B., HODGINS J. K.: Motion capture-driven simulations that hit and react. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2002), SCA '02, ACM, pp. 89–96. 2