Generating Responsive Life-Like Biped Characters

Ben Kenwright
School of Computer Science
Newcastle University
Newcastle, United Kingdom,
b.kenwright@ncl.ac.uk

ABSTRACT
In this paper, we present a real-time method for generating 3D biped character motions that are dynamic and responsive but also believably life-like and natural. Our model uses a physics-based controller to generate intelligent foot placement and upper-body postural information, that we combine with random human-like movements and an inverse kinematic solver to generate realistic character animations. The key idea is modulating procedurally random rhythmic motions seamlessly in with a physics-based model to produce less robot-like static looking characters and more life-like dynamic ones. Moreover, our method is straightforward, computationally fast and produces remarkably expressive motions that are physically accurate while being interactive.

Categories and Subject Descriptors
I.3.7 [**Three-Dimensional Graphics and Realism**]: Animation; Virtual Reality. I.6.8 [**Types of Simulation**]: Animation; Gaming; Visual. I.3.5 [**Computational Geometry and Object Modelling**]: Physically based modelling. K.8.0 [**Games**];

General Terms
Algorithms, Design, Experimentation, Human Factors

Keywords
Natural, responsive, 3D, character, balancing, physics-based, games, non-repetitive, real-time, procedural animation

1. INTRODUCTION
This paper presents a procedural physics-based method for generating biped character animations *without key-frames* that are life-like, responsive, and non-repetitive. Furthermore, we accomplish this in a straightforward and practical way that can be used by time critical systems such as games without using a vast library of motion-capture data.

The task of generating life-like character motions using procedural physics-based techniques is complex and challenging. While some techniques use purely procedural methods to produce exceptionally life-like motions, they can be inflexible or be computationally expensive and hence unable to run at real-time frame-rates. Alternatively, some methods use physics-based models to generate characters that are physically accurate and responsive, but feel robot-like and life-less. On the other hand, hybrid physics-based methods have combined controller techniques with motion-capture data to solve this problem but require custom animation libraries [1–3].

Furthermore, the current majority of animated characters use canned key-framed animation clips (including motion-capture libraries). These clips are blended and looped together to generate motions because they give the maximum amount of control while being straightforward to implement. However, while these motions can embed realistic characteristics and appear life-like, they produce repetitive, inflexible, and non-dynamic movement. In addition, motion-capture libraries can be costly to create and consume large amounts of memory. Additionally, these motion-capture clips are dependent upon a particular scene and character. Nevertheless, this does not mean they are useless. There has been research into methods for modifying these motion-capture libraries to reduce the cost of having to re-create them for each new scene [4–7]. Alternatively, other research has used various motion capture libraries as a method for training existing physics-based systems [8].

Conversely, our method tries to avoid any un-necessary motion-capture libraries. Instead, we generate motions using a physics-based controller and combine them with coherent random actions. This means we get a physically correct model that responds to force disturbances while exhibiting non-repetitive human-like emotions.

For example, the small unpredictable actions that are exhibited by a person such as swaying, twitching, and looking around are crucial for portraying life. A person's motions contain key similarities, but they also contain a certain amount of unpredictable randomness. This randomness is what makes the movement more human-like and less robot-like. We as humans have a keen eye for identifying when this random life-like movement is missing. Hence, we present and discuss how we added coherent random motions to a physically accurate model, to produce more believable synthetic characters.

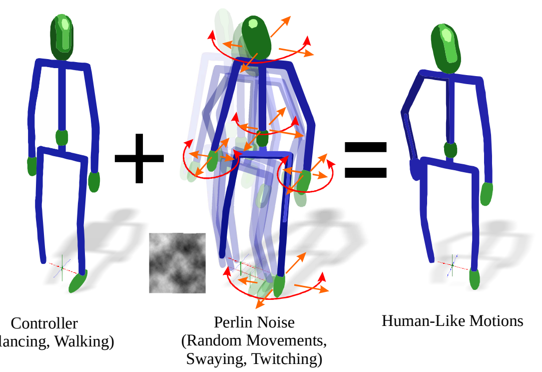


Figure 1. Generating motions using a controller, that may look robot-like, and combining them with human-like random disturbances to produce motions that are more human.

At the heart of our model is an uncomplicated physics-based character controller. The controller gives smart feet and postural information. This information in parallel with random coherent human-like motions is applied to an inverse kinematic solver. Additionally, a small amount of feedback is added between the inverse kinematic model and the physics-based model to couple them.

Figure 1 shows a simplified illustration of what this paper presents. A character controller generates physically accurate motions, which can be responsive but lack human attributes. We add life-like random gestures to the controller to produce more human-like motions. The result is a responsive autonomous character that contains humanistic characteristics.

The controller can perform simple actions such as balancing, responding to disturbances (e.g., taking a corrective step to remain balanced when pushed), walking, and running.

Essential model elements:

- Physics-based model
- Injection of random coherent motions for life-like
- *Key-frameless* automatic motions (balancing, walking, running)
- Intelligent feet placement (no-slipping)

1.1 Motivation

The motivation for this research is aimed at moving away from key-framed repetitive biped animations towards more procedural, scalable, dynamic, and interactive solutions for more life-like characters.

1.2 Contribution

The contribution of this paper is to generate more natural autonomous character motions 'without key-frames' by injecting random life-like movement that emulates real-world twitches, swaying, and other human movement into a physics-based controller model with feedback that can balance and respond to disturbances to produce believable human like characters.

1.3 Roadmap

The roadmap for the rest of the paper is as follows. Section 2 gives a broad overview of the related research; Section 3 describes the individual components of our system, their combination, and their design. Section 3.2 discusses the implementation details. Section 3.3 presents our initial results. Section 3.4 outlines limitations of our system. We conclude with Section 4 and Section 5 that discusses further work and conclusions.

2. RELATED WORK

The field of character animation is vast and diverse. Furthermore, generating more life-like dynamic biped characters is a hot topic of investigation across numerous research fields (i.e., graphics, robotics, and biomechanics.)

Some of the research has focused on generating responsive motions from physics-based models similar to the one used in this paper. Alternative methods have used data driven approaches, whereby, pre-canned animated clips fabricate the final motion for the desired circumstance.

However, we broadly classify the character animation systems into two main types; *data-driven* and *dynamic-simulation-systems*. *Data-driven* systems use generated or pre-canned motions. *Dynamic-simulation* systems use physics-based models to generate motions from forces and torques.

Data-driven methods produce animations based on scripts, procedural data, or key-framed motion capture. Combining a large data base of small animation clips to construct overall animations [6], [9], [10]. Combining motion capture data with dynamic controllers [1–3]

Physics-based methods have shown tremendous possibility. Whereby, *Faloutsos* [11] presented a "virtual stuntman" capable of numerous actions (e.g., walking, running, rolling). Also, *Hogins* [12], [13], simulated human athletic motions (e.g., running, gymnastics, bicycling). However, the controllers for these various motions can require vast amount of user tweaking. On the other hand, *Treuille* [8], developed an offline method of training the controller values automatically from motion-capture data to achieve the desired end result. Furthermore, additional tools have been presented for making the customization of controllers easier [14], [15].

If we focus on human characters we can see that over the past decade there has been a great deal of work done on generating realistic human motion [16]. Kinematic approaches calculate joint angles with no dynamic considerations [17], [18]. Alternatively, numerous physics-based dynamic approaches [15], [19–22] have been presented. Conversely, switching between different methods to produce hybrid-character animation system is not a new idea [23]. In the same way, research has been focused on gestures generation, such as speech synchronizing gestures [24] and broad-spectrum gestures techniques [25].

More specifically, *Neff* gave a method for adjusting the body shape [26] or motion trajectories and timing [27] from acting and choreographic theory. *Rose* [28] generated expressive motions using data-driven methods by interpolating clips. Analysing motion-capture data to extract learning styles of behaviour [29], [30].

The pioneering work by *Perlin* [31], [32] for scriptable procedural system for generating synthetic motion gave exceptionally life-like characters. However, the system was based on scripts, which needed to be accurately tuned for precise timing between body part movements to be physically correct.

The early research presented by *Badler* [33–35] was some of the first work on low-level controllers to perform various actions that were combined in parallel using a state machine and an inverse kinematic body in combination with a high level control interface and AI planning techniques.

Furthermore, the system by *Thiebaux* [36] implemented 'SmartBody' that blends selected control motions and mixes in procedurally generated actions. Alternatively, *Shapiro* [14] sequences of key-frames are combined using a dynamic controller with python scripts. Adaptable, extensible character animation systems that generate life-like synthetic motions are still largely unused in the game industry.

Following this further, various solutions have been presented that focus specifically on how to generate responsive characters, such as *Komura* [37] who simulated reactive motions for running and walking human figures. *Zordan* [2] simulated characters that respond automatically to impacts and smoothly returned to tracking. Then in his later work [3], combined existing human motion-capture data to produce a physics-based responsive motion segments that respond to varying force disturbances (demonstrated using martial art test bed). Furthermore, *Shiratori* [38] developed a controller that generates responsive actions to keep a character balanced. *Tang* [39] interactive character motions for falling with realistic responses to unexpected forces. *McCann* [40] blending between various motion capture segments to produce responsive character motions. *Arikan* [41] presented a physics model to emulate people being pushed around.

Emphasising some of the important research in robotics that has contributed to the development of responsive physics-based characters was presented by *Stephens* [43] and *Pratt* [44] who both presented full push recovery controllers; in addition, there was also the work by *Shih* [42] who developed a dynamic biped model for responding to small disturbances.

Nevertheless, to emphasise and recap the research which inspired the direction of our work and is most closely related, is the random coherent motion work from the paper by *Perlin* [32]; who presented initial results that demonstrate enhanced realism by adding random coherent motions to characters. We have extended existing work, for a controller based upon an modified pendulum model to generate postural information [45]. In final consideration, we combined an inverse kinematic solution to unify our scheme and generate a complete skeleton motion system.

3. DEVELOPMENT

3.1 System Overview

Our model is computationally fast, straightforward and effortless to implement. It uses a robust dynamic controller to generate physically realistic poses that can respond to unpredictable force disturbances. The controller also has the added advantage of being easily controllable. We combine our simple controller with an Inverse Kinematic (IK) model to generate a full human character skeletal pose. The resulting life-like style for our character is then achieved by adding minor supplementary random actions to the IK model.

The model is composed of three main parts (as shown in Figure 2.

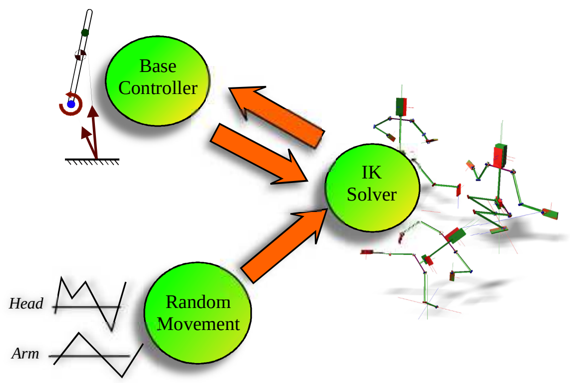


Figure 2. The interconnection of the model components that makeup our character system.

The controller generates key information using a physics-based model that includes intelligent foot positioning and postural orientation.

In addition, the controller generates fundamental upright motions (e.g., standing, walking) without relying on large quantities of key-framed clips. Since the controller is physics-based the generated motions are physically correct.

3.1.1 Base Controller

The controller mechanism is an enhanced inverted pendulum model based on *Kenwright* [45]. It is computationally fast and simple to implement and is ideal for real-time interactive applications such as games.

The low detailed controller was used to generate basic motion information for the inverse kinematic solver. The fundamental motions for the uncomplicated controller are standing, walking and running.

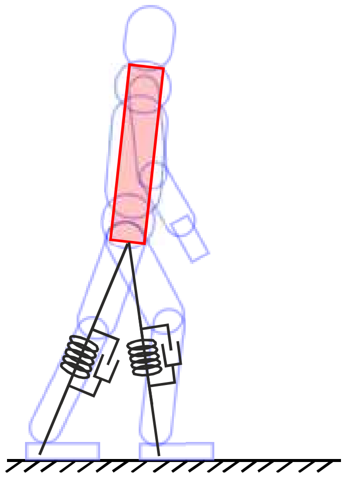


Figure 3. Controller connects to an inverse kinematic solver to give the overall character skeleton solution.

The controller was sufficient for this paper to demonstrate the essential philosophy of our method. Although we do not advance the controllers' model here, we present the dynamic controllers' workings in detail to both provide background for discussion and to make it clear how the physics-system sits and interacts in our system design scheme.

Furthermore, *the logic behind our controller model is based on the similarity that the human muscle is mechanically analogous to a spring-damper system*; subsequently, stiffness and damping factors of the system can be estimated to closely mimic how a person's limbs respond. This hypothesis is the fundamental reasoning, whereby this simple base-model imitates a character's legs and posture.

As shown in Figure 3, the controller model comprises of an elongated rigid body that representing the human body plus two springs for the legs.

A detailed illustration of the key components for the base controller is shown in Figure 4. Where the variables r_P, λ, and r_H represent the dynamic control parameters (e.g., to control step size, upper body stiffness and steering).

The brain of the controller is a state machine. This determines which foot needs to move and where to move it, so the character remains upright and balanced. For example, if the models Centre of Mass (CoM) starts to fall to one side, then the inverted pendulum model determines the location to place the foot to stop the CoM moving in that direction. Furthermore, by controlling how the CoM moves and where the feet are placed, the model is able to achieve walking and running motions. Most importantly, since the model is dynamically updated, it can adjust to handle disturbances, such as pushes and changes in terrain height.

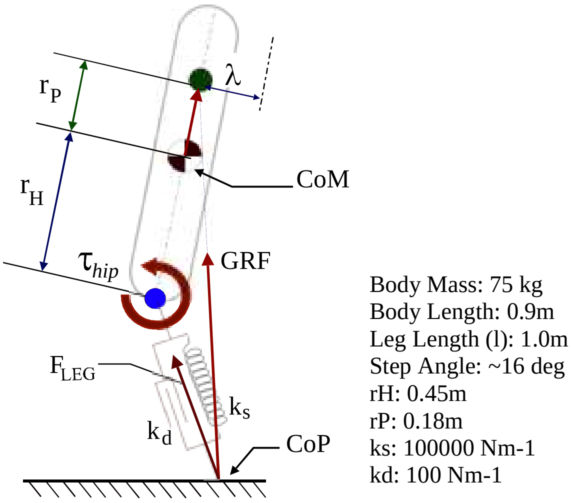


Figure 4. Base controller model.

The dynamic equations corresponding to the controller model (shown in Figure 4) are given below in Equations (3.1) to (3.5).

$$\hat{V}_C = \frac{(VPP - CoP)}{||VPP - CoP||} \quad \textbf{(3.1)}$$

$$\hat{V}_{TRUNK} = \frac{(VPP - CoM)}{||VPP - CoM||} \quad \textbf{(3.2)}$$

$$||GRF|| = F_{LEG} \cdot \hat{V}_C \quad \textbf{(3.3)}$$

$$GRF = ||GRF|| \,.\, \hat{V}_C \quad \textbf{(3.4)}$$

$$\tau_{hip} = GRF \ \times \hat{V}_{TRUNK} \quad \textbf{(3.5)}$$

where GRG is the ground reaction force, CoP is the centre of pressure and CoM is the centre of mass. In essence, the essential result is from Equations (3.5) that calculates the necessary body torque to keep the characters body upright. For a more detailed explanation of how the model functions, see *Kenwright* [45].

Additionally, the controller has a number of advantages. Firstly, the complete dynamic state of the character is contained together with postural position, orientation and feet information. Secondly, the controller model significantly simplifies the dynamic calculations while providing essential motion data. Finally, the controller model does not have to worry about individual limb joint angles or complex constraints.

3.1.2 Character Inverse Kinematic Model

The low detail base controller generates information for the desired feet positions and postural orientation that is used to generate the pose for a high detailed character model.

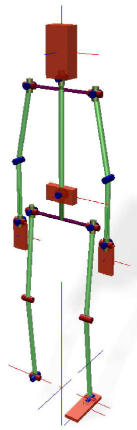
We combined the physics-based controller with an inverse kinematic solution to avoid the necessary problem of having to hand tune gains to achieve accurate simulations [12], or similarly needing to use example motion capture clips to train the gains [8].

The high detailed character model has five end-end effectors. As shown in Figure 5, the body is represented by rigid body segments connected using 14 links. The character gives us 30 degrees of freedom (DOF).

The feet's positions and body orientation are taken from the controller and passed to the inverse kinematic solver to generate the characters pose (as shown in Figure 3).

On some occasions, the target end-effectors positions and orientations cannot be achieved by the inverse kinematic solution and would achieve a best guess approximation. As shown in Figure 6, the target end-effectors are red while the current end-effectors are green.

The support foot that is holding the characters weight is used as the base for the inverse kinematic solver. As shown in Figure 5 and Figure 6, the support foot is identified as the foot that is not drawn.



Joint	DOF
Head	3
Shoulder	3
Elbow	1
Hand	2
Pelvis	3
Hip	3
Knee	1
Foot	2
Total	**30**

Figure 5. Character joint configuration.

The inverse kinematic system also enforces joint limits.

3.1.3 Random Movement

Random coherent motions are generated, such as head turning and arms swaying which are applied to the moveable end-effectors. The foot supporting the weight of the character cannot be moved.

Random motions are added using coherent noise functions. The original work based on *Perlin* [32] would directly affect the joint angles, alternatively, we only apply the random motions to the end-effectors. Similar using end-effectors to generate gesture motions [46][47][48].

The random coherent motions are added to the moveable end-effectors (e.g., hands, head, pelvis, non-support foot). This results in positional and orientation discrepancies between the inverse kinematic body and the physics-based body. These discrepancies are fed back to the physics-based model to add postural corrections so that the character remains balanced.

It is necessary for the motions to be coherent so that the character produces smooth natural looking animations. The basic random motions that added the most life-like look to our character where: arms swaying, head looking around, and random arm poses.

The rhythmic random movement parameters were generated by hand using a trial-and-error approach. Whereby, the fundamental movements were determined by observing miscellaneous people's actions and identifying repetitive similarities that expressed what that person's mood might be (e.g., bored or tired).

Conversely, while a good set of rhythmic random movements can produce highly realistic life-like characters the opposite is also possible. Whereby, a bad choice of randomness can result in uncommon and bizarre looking gestures.

However, the inverse kinematic constraints prevent the random motions from performing absurd actions, for example the head spinning all the way around.

3.1.4 Feedback

We added feedback from the inverse kinematic solution to the controller to make the results more visually pleasing. The feedback is proportional to the difference between the current inverse kinematics body's location and the current controllers' body's location. The feedback adds a correcting force to the rigid body of the controller. This feedback would alter the physical accuracy of the model. However, with feedback the resulting motions were more natural and life-like. We approximate these corrective posture feedback forces down to ankle torques. For example, as a person sways and moves, their ankles and feet generate corrective torque forces to keep that person balancing and upright (see Figure 7).

3.2 Implementation

The character model pose was constructed from the inverse kinematic solution. The IK model as shown in Figure 5 had five end-end effectors (non-stance foot, pelvis, left-hand, right-hand and head). The end-effectors positions and orientations for the non-support foot and pelvis were taken from the controller. In addition, random motion gestures were applied to the five end-effectors.

We constructed our model in Microsoft's XNA platform [49] with C# and effortlessly ran at real-time frame-rates. System information: Windows7 64-bit 16Gb Memory, Intel i7-2600 3.4Ghz CPU. Compiled and tested with Visual Studio 2010.

3.3 Results

We would have the character stand then apply random motions to look around and have the arms sway. We then extended the test by pushing the model around so that it swayed due to push forces and eventually took a corrective step.

Furthermore, we had the character walk around, simultaneously applying random actions such as looking up, minor swaying from leaning left and right, and fluctuations from the hands and arms.

The key observation the results presented for our autonomous character either standing or walking was the feeling of 'life'; as if the character was aware of itself. Alternatively, as we expected, without the simple random swaying, arm movements and looking around the character appeared static and robotic-like.

Clearly, we believe, our results show that combining a generated controller's movement with simple coherent random life-like actions produces characters that feel '*alive*'.

Figure 6, shows some screenshots of our character looking around before being pushed and having to take a corrective step.

3.4 Limitations

Our single controller implementation was only able to generate a limited number of motions (e.g., upright standing, walking and running). However, it demonstrates the innermost principle for generating more life-like characters using a procedural physics-based model with random movement. Therefore, actions, such as get-up, climbing, sitting would require additional controllers or mixing key-framed animations to extend the repertoire of behaviors.

In addition, our model focused on human biped animations; however, it is logical to assume that alternative controllers in conjunction with coherent noise can make other creatures (e.g., animals, aliens) more life-like.

4. FURTHER WORK

We only worked on a single upright balancing biped controller. However, by combining additional controllers, such as get-up, climb, and fight would give a larger repertoire of actions and would enable our model to be a viable solution for generating believable virtual characters.

Furthermore, since we only generated a fundamental set of random coherent motions to mix-in with the base-controller, an extended list of actions would enable us to emphasis more moods and behaviours. For example, identifying a dissimilar range of key random motions (e.g., nervous, angry or a guilty person might exhibit) to build up a much more diverse collection of expressions for the scriptable library.

Moreover, the parameters for generating the random motions were fabricated by hand using a trial-and-error approach. Whereby, over a period of time, we would observe the subtle rhythmic motions of real people to produce a key set of random parameters that mimicked a person's subconscious behaviour and added a life-like quality to our biped. Alternatively, a more systematic method of analysing, correlating, and extracting recurring random rhythmic movements from motion capture libraries or video data could be investigated and employed.

We believe our model presents a good starting point for developing crowds of autonomous characters that exhibit unique life-like motions with the ability to respond to unforeseen circumstances.

5. CONCLUSIONS

In this paper, we have described a method for generating real-time life-like responsive motions for game characters. To accomplish this goal, we combined a physics-based controller with random human-like gestures. Our approach could be extended to generate numerous autonomous characters that produce active non-repetitive animations. For our implementation, we experimented with a single controller that generated a small set of actions (e.g., standing, walking, and running). The model responded well to disturbances such as pushes and pulls. Furthermore, motions for travelling around on various terrains were more life-like and engaging when combined with random gestures.

The controllers' adaptive dynamic nature means that our character walks realistically on various terrains (e.g., un-even ground) while generating intelligent foot placements.

The model produces pleasing motions that contain both physically accurate results and human-like features. The controller enables the feet to be placed smartly (e.g., the feet do not slide but are moved realistically as a real-person would). The algorithm is relatively straightforward and computationally efficient that makes it practical for time-critical systems such as games.

More and more simulated game characters are going beyond 'rag-doll' like physics-based models that are combined with repetitive animations to more smart thinking solutions. These smarter solutions ask the question – 'how would a person respond to it in the real world?' – 'how can we emulate that movement in an algorithm?' Moving towards these, smarter more novel solutions, like the one we present in this paper, will result in a new generation of immersive games with more life-like characters.

6. REFERENCES

[1] P. Wrotek and O. Jenkins, "Dynamo: dynamic, data-driven character control with adjustable balance," *Proceedings of the 2006 ACM*, 2006.

[2] V. B. Zordan and J. K. Hodgins, "Motion capture-driven simulations that hit and react," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA '02*, 2002, p. 89.

[3] V. B. Zordan, A. Majkowska, B. Chiu, and M. Fast, "Dynamic response for motion capture animation," *ACM Transactions on Graphics*, vol. 24, no. 3, p. 697, Jul. 2005.

[4] A. Witkin and Z. Popovic, "Motion warping," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 105–108.

[5] H. J. Shin, J. Lee, S. Y. Shin, and M. Gleicher, "Computer puppetry: An importance-based approach," *ACM Transactions on Graphics (TOG)*, vol. 20, no. 2, pp. 67–94, Apr. 2001.

[6] A. Bruderlin and L. Williams, "Motion signal processing," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 97–104.

[7] C. F. Rose, B. Bodenheimer, and M. F. Cohen, *Verbs and adverbs: multidimensional motion interpolation using radial basis functions*. Citeseer, 1999, pp. 1-17.

[8] A. Treuille, Y. Lee, and Z. Popović, "Near-optimal character animation with continuous control," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 7, 2007.

[9] R. Heck, "Parametric motion graphs," *Proceedings of the 2007 symposium on Interactive*, pp. 1-8, 2007.

[10] L. Kovar and M. Gleicher, "Motion graphs," *ACM Transactions on Graphics (TOG)*, vol. 1, pp. 473-482, 2002.

[11] P. Faloutsos, M. van de Panne, and D. Terzopoulos, "The virtual stuntman: dynamic characters with a repertoire of autonomous motor skills," *Computers & Graphics*, vol. 25, no. 6, pp. 933–953, 2001.

[12] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien, "Animating human athletics," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 71–78.

[13] W. L. Wooten and J. K. Hodgins, "Simulating leaping, tumbling, landing and balancing humans," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 2000, vol. 1, no. March, pp. 656–662.

[14] A. Shapiro, D. Chu, B. Allen, and P. Faloutsos, "A dynamic controller toolkit," *Proceedings of the 2007 ACM SIGGRAPH symposium on Video games - Sandbox '07*, p. 15, 2007.

[15] P. Faloutsos, M. van de Panne, and D. Terzopoulos, "Composable controllers for physics-based character animation," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, no. 1, pp. 251–260.

[16] F. Multon and L. France, "Computer animation of human walking: a survey," *computer animation*, vol. 10, pp. 1-20, 1999.

[17] D. Zeltzer, "Motor control techniques for figure animation," *IEEE Computer Graphics and Applications*, vol. 75, pp. 53-59, 1982.

[18] R. Boulic, "Hierarchical kinematic behaviors for complex articulated figures," *Interactive computer animation*, no. 1, pp. 1-27, 1996.

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

[20] K. Hase, K. Miyashita, S. Ok, and Y. Arakawa, "Human gait simulation with a neuromusculoskeletal model and evolutionary computation," *The Journal of Visualization and Computer Animation*, vol. 14, no. 2, pp. 73-92, May 2003.

[21] M. McKenna and D. Zeltzer, "Dynamic simulation of autonomous legged locomotion," in *ACM SIGGRAPH Computer Graphics*, 1990, vol. 24, no. 4, pp. 29–38.

[22] M. H. Raibert and J. K. Hodgins, "Animation of dynamic legged locomotion," *ACM SIGGRAPH Computer Graphics*, vol. 25, no. 4, pp. 349-358, Jul. 1991.

[23] a. Bruderlin and T. W. Calvert, "Goal-directed, dynamic animation of human walking," *ACM SIGGRAPH Computer Graphics*, vol. 23, no. 3, pp. 233-242, Jul. 1989.

[24] J. Cassell, H. H. Vilhjálmsson, and T. Bickmore, "BEAT: the behavior expression animation toolkit," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, no. August, pp. 477–486.

[25] S. Kopp and I. Wachsmuth, "Planning and motion control in lifelike gesture: a refined approach," in *Computer Animation 2000. Proceedings*, 2000, no. May, pp. 92–97.

[26] M. Neff and E. Fiume, "Methods for exploring expressive stance," *Graphical models*, vol. 68, no. 2, pp. 133–157, 2006.

[27] M. Neff and E. Fiume, "Aesthetic edits for character animation," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2003, pp. 239–244.

[28] C. Rose, M. F. Cohen, and B. Bodenheimer, "Verbs and adverbs: Multidimensional motion interpolation," *Computer Graphics and Applications, IEEE*, vol. 18, no. 5, pp. 32–40, 1998.

[29] M. Brand and A. Hertzmann, "Style machines," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 183–192.

[30] C. K. Liu, A. Hertzmann, and Z. Popović, "Learning physics-based motion style with nonlinear inverse

optimization," *ACM SIGGRAPH 2005 Papers on - SIGGRAPH '05*, p. 1071, 2005.

[31] K. Perlin and A. Goldberg, "Improv: A system for scripting interactive actors in virtual worlds," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 205–216.

[32] K. Perlin, "Real time responsive animation with personality," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 1, no. 1, pp. 5–15, Mar. 1995.

[33] C. B. Phillips, J. Zhao, and N. I. Badler, *Interactive real-time articulated figure manipulation using multiple kinematic constraints*, vol. 24, no. 2. ACM, 1990.

[34] N. I. Badler, M. S. Palmer, and R. Bindiganavale, "Animation control for real-time virtual humans.," *Communications of the ACM*, vol. 42, no. 8, pp. 64-73, Aug. 1999.

[35] N. I. Badler, C. B. Phillips, and B. L. Webber, *Simulating humans: computer graphics animation and control*. Oxford University Press, USA, 1993.

[36] M. Thiebaux, S. Marsella, A. N. Marshall, and M. Kallmann, "Smartbody: Behavior realization for embodied conversational agents," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, 2008, no. Aamas, pp. 151–158.

[37] T. Komura, H. Leung, and J. Kuffner, "Animating reactive motions for biped locomotion," *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '04*, p. 32, 2004.

[38] T. Shiratori, B. Coley, R. Cham, and J. K. Hodgins, "Simulating balance recovery responses to trips based on biomechanical principles," *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '09*, p. 37, 2009.

[39] B. Tang, Z. Pan, L. Zheng, and M. Zhang, "Interactive generation of falling motions," *Computer Animation and Virtual Worlds*, vol. 17, no. 3-4, pp. 271-279, Jul. 2006.

[40] J. McCann and N. Pollard, "Responsive characters from motion fragments," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 6, Jul. 2007.

[41] O. Arikan, D. a. Forsyth, and J. F. O'Brien, "Pushing people around," *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA '05*, no. July, p. 59, 2005.

[42] C. L. Shih, W. A. Gruver, and T. T. Lee, "Inverse Kinematics and Inverse Dynamics for Control of a Biped Walking Machine," *Journal of Robotic Systems*, vol. 10, no. 4, pp. 531-555, 1993.

[43] B. Stephens, "Humanoid push recovery," *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pp. 589-595, Nov. 2007.

[44] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, 2006, pp. 200–207.

[45] B. Kenwright, R. Davison, and G. Morgan, "Dynamic Balancing and Walking for Real-Time 3D Characters," *Motion in Games*, pp. 63–73, 2011.

[46] M. Neff, M. Kipp, I. Albrecht, and H.-P. Seidel, "Gesture modeling and animation based on a probabilistic re-creation of speaker style," *ACM Transactions on Graphics*, vol. 27, no. 1, pp. 1-24, Mar. 2008.

[47] B. Hartmann and M. Mancini, "Implementing Expressive Gesture Synthesis for Embodied Conversational Agents," *gesture in human-Computer*, 2006.

[48] S. Kopp and I. Wachsmuth, "Synthesizing multimodal utterances for conversational agents," *Computer Animation and Virtual Worlds*, vol. 15, no. 1, pp. 39-52, Mar. 2004.

[49] Microsoft, "Microsoft Corporation, XNA Game Studio 4.0," 2012. .

7. APPENDIX

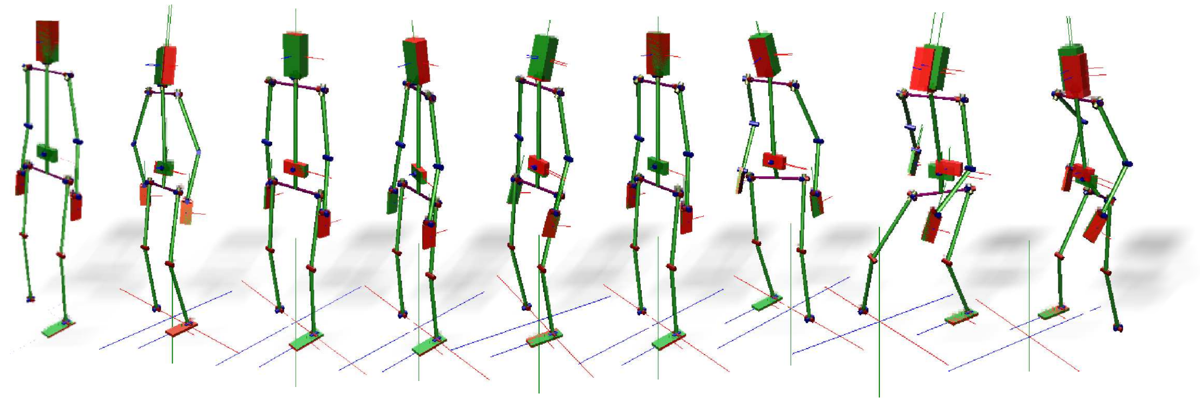


Figure 6. This figure shows an illustration of our real-time character standing and looking before being pushed.

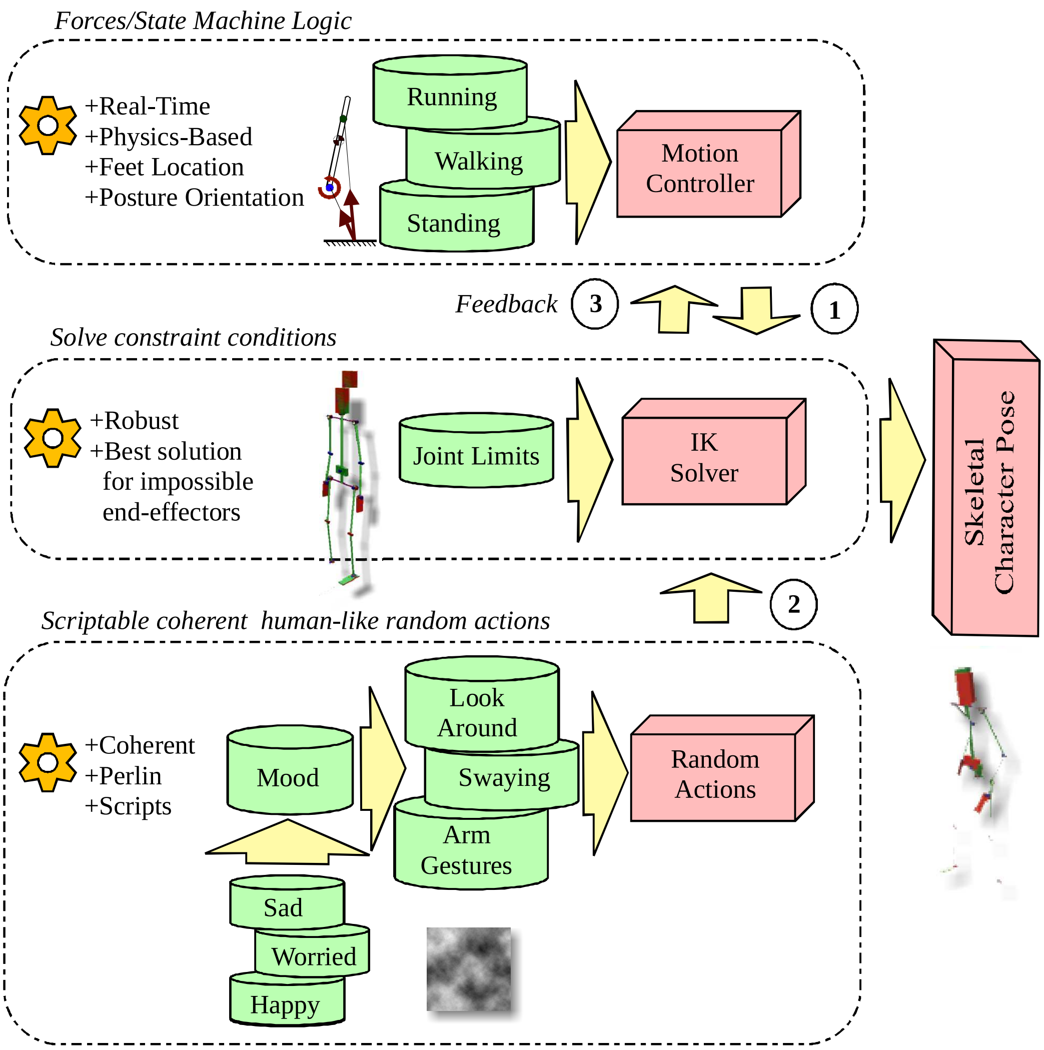


Figure 7. Our system and the various interconnecting parts.