Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2006)
M.-P. Cani, J. O'Brien (Editors)

The Dynamic Controller Toolkit

Ari Shapiro[†] Derek Chu[‡] Petros Faloutsos[§]

University of California, Los Angeles

Figure 1: *Animators can keyframe poses and synthesize dynamic movement through sequential pose transitions.*

1. Introduction

We introduce a toolkit for creating dynamic controllers for articulated characters under physical simulation. Users can create dynamic controllers for interactive use through a combination of both visual and scripting tools. The goal of the toolkit is to integrate dynamic control methods into a usable interactive system for non-computer scientists and non-roboticists, and provide the means to quickly generate physically-based motion.

Although physical simulation is well-understood, controlling a physically simulated character in order to make her perform desired actions is not. The difficulty in developing dynamic controllers is partly due to the complexity of control algorithms and partly due to a lack of availability of tools for animators to easily create dynamically-based motion.

The toolkit is built with the DANCE animation system [SFNTH05] and allows animators to develop dynamics-based controllers through a combination of:
- key-framed based control,
- reduced dimensionality physics,
- scripting controllers via a controller language,
- interactive control of dynamic characters.

2. Pose-Based Control

Keyframing is a popular kinematic technique that can automatically generate sequences of animation from a small set of user-described poses. Physically simulated characters can also utilize keyframed poses by treating the keyframe as the desired pose and using proportional derivative (PD) control to drive the character to the desired position.

2.1. Reduced Physical Dimensions and Forces

We allow the animator to change the virtual environment in order to make the design of controllers easier. For example, we can immediately constrain the forces affecting the character to a particular axis, as well as reduce or eliminate accelerations (and thus velocities) in other directions.

The effect of reducing physical forces on a simulated character can change dramatically the ability of an animator to create effective dynamic controllers. For example, walking in 2D is much easier to do than in 3D and can even be accomplished for limited steps by interactively selecting the proper pose as shown by [LvF96] and in our accompanying video. By constraining physical forces, we can greatly simplify the development of a robust walking controller.

2.2. Controller Scripting

We use scripting functions that allow the designer to access various aspects of the sensors. Sensor-based information includes the location of the character's limbs, presence of other object in the environment, whether or not the character has been impacted by another character or object and so forth.

The presence of a scripting language allows much of the complicated balancing and movement strategies to be abstracted for the end user. Experts can write parameterized functions that accomplish simple tasks to be used by animators.

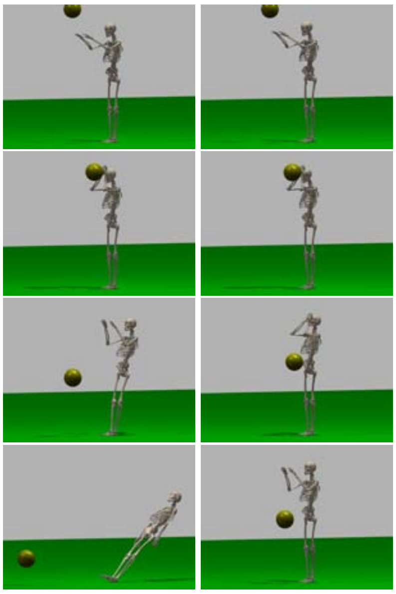

Figure 2: *Physical forces can be modulated on a character based on certain events. The left character uses a pose controller blocks the ball, causing the character to fall. For the character on the right, forces on the hip are disabled for a short period of time after contact, allowing the character to remain upright. However, forces are still applied to the remainder of the character and the impact can be seen on the upper body.*

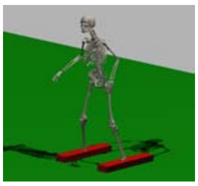

Figure 3: *The toolkit allows the user to change the physical characteristics in order to make complex tasks more easy. Here, a character uses ski-like feet in order to promote balance.*

3. Interactive and Reactive Control

Interactively controlling physically-based characters is also difficult. The difficulty stems from both the high number of parameters needed to control an interactive character as well as the difficulty in specifying proper parameters to accomplish meaningful movements.

Users can toggle between interactive and autonomous control. We achieve this flexibility by allowing the user to combine controllers that perform simple, posture-oriented movements and those that perform complex control strategies such as balance. For example, we have developed a set

Figure 4: *The user controls the 3D character interactively. The controllers are chosen at a timing indicated by the user, and can be simple pose based controls or complex, scripted controllers.*

of falling, balancing and protective strategies for 2D and 3D characters which can be activated interactively at any time. This incorporation of autonomous control gives the character reactive skills that allow it to recover gracefully from various undesirable positions and situations, such as being prone or supine on the ground, or under attack from another character.

4. Discussion

Our application seeks to fill a void in the capabilities of animators and researchers to create character animation. This gap in capability can be seen in the amount and quality of tools available for kinematically-driven character animation versus that for dynamically-driven character animation. The barrier to entry for the development of dynamic controllers is high and left in the hands of the experts in robotics, graphics, artificial intelligence and the like. This is one of the largest obstacles in the way of adopting and developing dynamic control for character animation.

References

[LvF96] LASZLO J., VAN DE PANNE M., FIUME E.: Limit cycle control and its application to the animation of balancing and walking. In *Computer Graphics (SIGGRAPH '96 Proceedings)* (1996), pp. 155–162.

[SFNTH05] SHAPIRO A., FALOUTSOS P., NG-THOW-HING V.: Dynamic animation and control environment. In *GI '05: Proceedings of the 2005 conference on Graphics interface* (School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005), Canadian Human-Computer Communications Society, pp. 61–70.