Predictive Feedback for Interactive Control of Physics-based Characters

Joe Laszlo, Michael Neff and Karan Singh[†]

Department of Computer Science, University of Toronto

Abstract

Interactive control of a physically simulated character is a challenging problem, due both to the complexity of controlling multiple degrees of freedom with lower dimensional input and because many interesting motions lie on the fringes of character stability. This paper addresses these problems using a novel technique called predictive feedback, where a glimpse into the near future for a few sample inputs is continuously presented to the animator. We discuss issues related to the spatio-temporal distribution of predictions so that they provide meaningful and timely feedback to an animator interactively controlling a physics-based character with simple input devices, like a mouse or keyboard. We propose a visual presentation of this predictive feedback in which control input samples are chosen in the proximity of the user's current input and the predicted results are co-located with the position of the input necessary to achieve them. We further show how the predictive samples may be automatically interpolated to control aspects of the character's motion, such as balance, thereby freeing the animator to focus on other details. The paper thus contributes a technique for physically simulated characters that simplifies interactive character control and increases the range of motion that can be performed by both novices and experts. Many of the presented concepts extend beyond our specific input device and dynamic character control setting to more general input tasks.

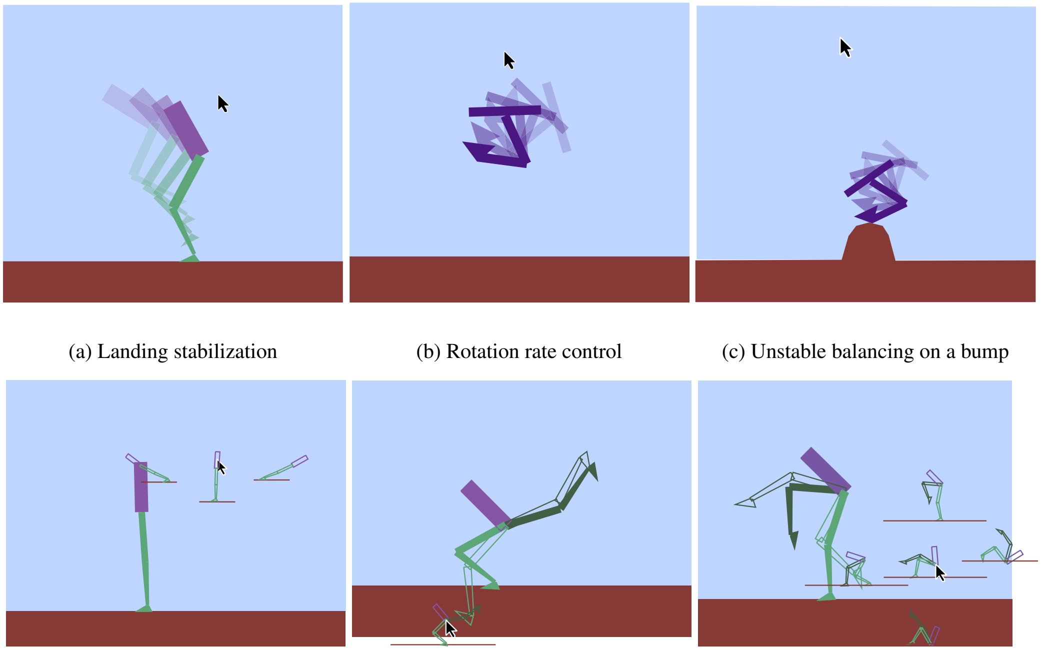
Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Interaction, Physical simulation, Character animation

1. Introduction

Interactive control of character animation has an appealing element of spontaneity that is lacking in scripted animations. Interactive Physics-Based Animation (IPBA), introduced by Laszlo et al. [LvdPF00], provides an engaging form of performance animation, suitable for dynamically interesting motions like tumbling and dancing, as well as interaction with changing environments. Unfortunately, IPBA is difficult to master. For a novice user, it can be hard to complete even simple movement sequences. For an advanced user, more dynamically challenging movements remain impossible.

Inspired by the human ability to employ predictive models while executing motor control tasks, predictive feedback provides users with a model of possible future states that can be used to guide current actions. One of the drawbacks of IPBA is that the visual feedback provided by the moving character provides little help in guiding what control input the user should provide. As a simple example, consider trying to control the distance that a character will jump. Once the character is airborne, the initial trajectory of the centre of mass allows the user to anticipate how far the character will travel. Feedback at this point is too late, however, as the user can no longer control the distance of the jump. The user needs feedback at the point of take off, when useful control can still be provided, but the visual cues at takeoff provide limited information. A real person executing the movement would rely on kinaesthetic feedback, but an animator has no sense of the ground reaction forces experienced by the character. By looking ahead in time and indicating the effect of different control decisions, predictive feedback provides the

[†] joe | neff | karan @dgp.toronto.edu



(a) Landing stabilization

(b) Rotation rate control

(c) Unstable balancing on a bump

(d) Balancing - simple standing

(e) An extreme arabesque

(f) Dynamic posture variation

Figure 1: *Examples of interactively controlled actions that are difficult to perform without error due to unstable character state or poor indication of outcome for control input variations. Many interesting motions involve dynamic balance and fine timing.*

user with the information they need at the time they need it to effectively control the character.

In our system, the user interactively controls the dynamically simulated motion of a planar character using mouse and keyboard input. Predictive feedback samples the control space of the character in the proximity of the performer's current control input and generates previews of possible states in the near future. A significant design decision was to display the predicted states in the same spatial locations as the input that would generate them, allowing the performer to interactively supply the correct control input by moving towards the desired preview state. This transforms the task required of the user from an unguided, complex and often unintuitive movement in input space, to a much simpler dynamic selection task, in which the user need only move towards the desired preview to generate the intended result.

The three main domains defining the system – input, control and presentation – are shown in Figure 2. Our input domain consists of the mouse and keyboard; the pointer in the figure showing the current *input* location. This generates a desired pose target in the *control* domain, which is shown in wire frame attached to the large figure. As explained later, PD-control is used to generate the torques that move the character towards this target pose. The *presentation* domain includes the larger view of the character being animated along with the smaller previews.

Momentum effects resulting from dynamic simulation mean that different speed input tracing the same path will generate different character motion. The input mapping is thus spatio-temporal. The user must employ a more complex mapping than in a straight kinematic system, but a wider range of movements is accommodated through a simple input device and predictive feedback simplifies the input task.

The vitality of movements near the balance limit is well understood in the performing arts (e.g. [Lab88]), where *precarious balance* is often actively sought, for instance, by a ballerina working on toe, or a Kathak dancer manoeuvring on one foot [Bar91]. This fundamentally dynamic effect is avoided in most physics based techniques which instead avoid singularities by keeping the character close to the centre of the stability region, where movements are more boring, but the character is less likely to fall. The predictive techniques presented in this paper allow an animator to explore the fringes of the stability region by indicating which inputs will cause an unrecoverable fall and which cling to the edge of stability. Challenging motions that would be difficult to generate with any other physics based technique are shown in Figure 1 and in the accompanying video. We make use of the animator's visual system and judgement to solve very difficult control problems.

Another advantage of IPBA is that the user never needs to define the goal of the control problem. By placing both the control input and the feedback in the user domain, an animator can interactively and continuously define the desired movement without an explicit representation of the goal, or even knowing the goal ahead of time. This allows for a very spontaneous and creative workflow.

In this paper we illustrate the use of predictive feedback in two ways. First, as an explicit visual feedback model where predicted states are displayed simultaneously to the animator as they interactively control the character (see Figure 2). Second, as the basis for an automatic control algorithm that generates control input to achieve a desired character state (such as interpolating the ankle rotation samples to keep the torso upright as in Figure 1d).

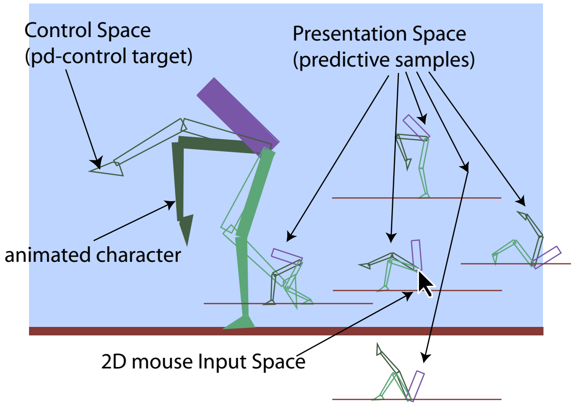


Figure 2: *Predictive feedback is used to control a character. The mouse pointer indicates the current location in the input domain. The wire frame legs attached to the large figure illustrate the desired pose in the control domain, which is determined by the current input. The presentation domain includes the large main character being animated and five smaller predictions for future states which are based on input samples in the local neighbourhood of the current input.*

2. Related Work

This work draws on three areas which are summarized below: physics based animation, interactive character control, and suggestive and predictive interfaces.

2.1. Physics Based Animation

Physics based animation (PBA) computes character motion within a physically simulated environment, generating joint torques to make a character move. Many difficult to animate aspects of real movement, such as how quickly a character tumbles in the air or how a character reacts as he falls on the ground, are directly governed by the physics of the situation and can be well captured with PBA.

One branch of PBA builds controllers to regulate the generation of torques over time. Early work focused on synthesizing controllers for the movement of low DOF characters [vdPF93, NM93, Sim94]. Hand-tuned controllers have been used to animate more complex characters [HWBO95, FvdPT01b]. Recent work has looked at automatically combining control solutions to achieve more complex tasks [FvdPT01a]. A related body of work uses the laws of Newtonian physics as constraints in an optimization solution rather than directly building controllers [WK88, Coh92, LGC94, PW99, FP03].

While the use of physical simulation has much potential, PBA has seen limited penetration into production animation. Interactive entertainment has used PBA increasingly, but both remain limited primarily to the use of passive (uncontrolled) simulations. This is largely due to the difficulty of creating robust control solutions for a broad array of motions and transitions between them.

Perhaps the biggest potential audience for PBA, however, lies in the everyday user, and eventually even the dedicated "virtual performance artist", who is entertained by the challenge of making a simulated body perform. For example, Motion Playground's Ski Stunt Simulator [vdPL03] has averaged approximately 200,000 plays *per day*, over 3 years. This group is our primary target.

2.2. Interactive Animation

Interactive animation systems attempt to leverage human technical and creative skill by giving animators intuitive interfaces for controlling motion. Contrast this with most control based PBA systems that have focused on autonomous movements. We divide interactive animation systems into three categories: kinematic systems, physically simulated systems and hybrid systems.

Perlin presented a kinematic system in which interactively selected motions for a virtual puppet are smoothly blended together [Per95]. Shin et al. apply the full body motions of an actor to an animated character in real-time [SLGS01]. Lee et al. allow a user to interactively specify which motion capture sequences are used to animate an avatar [LCR*02]. Yin and Pai use a foot pressure sensor to select motion capture clips to play on a virtual character [YP03]. Dontcheva et al. present a system where users compose an animation by interactively specifying layers of motion [DYP03].

Oore et al. [OTH02a] present a computer puppet system based on two six DOF input devices, and demonstrate that local physical models can capture key aspects of physics while largely preserving kinematic control [OTH02b].

For physically simulated characters, one approach to generating control has the animator/performer interactively drive the muscle activations of the character, using an input device, in a manner analogous to the control we exert over the muscles of our own bodies. Troy and Vanderploeg [TV95] provide different degrees of user control over a semi-automatic walking biped. Players of Interplay's "Die By the Sword" [Int98] control their character's sword arm with a mouse to dynamically slash at foes. Laszlo et al. demonstrate successful interactive control of freestanding planar characters, using a mouse and keyboard to perform dynamically interesting movements such as climbing, flipping and swinging [LvdPF00]. Motion Playground's "Ski Stunt Simulator" provides similar control over a physically simulated planar skier performing a range of tricks [vdPL03]. Zhao and van de Panne [ZvdP05] extend interactive control to 3D, controlling a snowboarder, diver and skier, each performing a variety of interesting tricks.

2.3. Suggestive and Predictive Interfaces

Suggestive interfaces [MAB*97, sug01, sug04] present choices using representations of the content in question, for example, the resultant state of a geometric model if an associated editing action is performed. The "Design Galleries" of Marks et al [MAB*97] allow a designer to interactively explore a wide variety of parameter spaces by spatially arranging numerous examples about the designer's current choice. Other work looks at the rapid construction of geometric models by suggesting possible additions to the model [sug01, sug04]. Lee et al. [LCR*02] allow the user to interactively select among character animation sequences in a database based on the current state.

Predictive interfaces present the user with the predicted state(s) of some dynamically varying process, typically with a degree of uncertainty in the prediction. In tele-operated robotics, predictive display is used to overcome delays in visual feedback inherent in many remote control applicatoins [She93, DJ03]. To date, these systems have been primarily applied to anchored manipulators or highly-stable mobile robots. NaturalMotion's simulation-based motion synthesis application, Endorphin [Nat04], displays a sparse time "strobe" sequence of character states as the user varies initial conditions and other simulation parameters.

3. Predictive Feedback Model

Our predictive feedback model evaluates the state of a character in the near future for a selected set of sample inputs. The goal of this evaluation is to use the information to aid a performer in the interactive control of a character. In this section we develop the basic predictive feedback model addressing the choice of control input samples, the predictive window of time and the visual presentation of predicted states to the user in the context of physically simulated characters. Appropriate choices will vary depending on the character being animated, the type and timing of the action being performed and the skill of the performer in manipulating the character. The various choices are parameters under performer control but we provide a good set of defaults in this section.

As explained previously, the model is formally defined with respect to three domains: input, control and presentation. There are many possible implementations for each domain. Our choices have been summarized in Figure 2 and details are provided below.

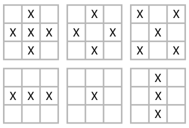


Figure 3: *Relative sample patterns used. The pointer location is taken to be the centre of each grid. The upper left pattern is used by default.*

3.1. Generating Input Samples For Prediction

We first need to decide how to sample the control domain for generating predictive samples. Predictive samples are a limited resource. To provide continuous feedback, both the main animation and predictions must be updated many times a second. During each update loop, the main animation must be simulated and each of the n predictions must be simulated for a significantly longer period of time. Predictions thus dominate the computational cost and it is therefore important to choose samples wisely.

Control domain samples should be generated as mappings from reasonably placed input domain samples because the system's primary goal is to provide a user with feedback on their interactive input. Indeed, arbitrary control samples are not even guaranteed to be mapped by the input domain.

There are two basic sampling strategies that were tested: samples taken from fixed locations in input space and samples taken from locations relative to the user's input. Fixed sampling had several disadvantages:

- A large number of samples are often required, leading to a high performance cost.
- The user has to divide his attention between the current input location and the sample locations.
- Sometimes there are gaps in the sampling and the most effective samples are missed.
- There is no user control over the samples that are selected.

The relative sampling strategy addresses these concerns. Fewer samples were generally sufficient with this strategy because the provided samples were in the current input region and hence more relevant. By locating samples around the current input location, attention is less divided. Finally, the user has control over the location of the predictions by moving the input device. When needed, the user can quickly "scrub" over the input domain to find suitable control locations for a desired movement. The main simulation can be paused while predictions continue to be updated, allowing exploration of the possibilities from the current state.

A number of sampling strategies were evaluated and the most useful patterns were based on a three by three grid with the current input location in the middle cell as shown in Figure 3. The guiding principle is to provide a good spacing over the local area. The five point cross is the default sampling strategy, provided there is enough computational power to generate that number of samples. More samples than this provide diminishing returns, and do not justify the performance cost. For some movement problems such as balancing, the control input is essentially linear and three collinear samples are sufficient. A single sample can be used with long look aheads for simple motions or by an experienced animator. The number of samples is largely dictated by the ability to generate predictions for the samples at an interactive rate. The effective number of samples is also limited by issues of visual clutter and cognitive load.

The sample distance in each direction needs to provide a meaningful change in the predicted character state from the current state. In this respect sampling distance is dependent on the predictive time interval and the stability of the character's current state. The distance must also be chosen to provide an appropriate visual display. The samples should be easily visible within the users field of view. We have not yet codified this relationship, but found for desktop interaction a spacing of 75 pixels or about 2cm is a good default.

Since the ideal sampling strategy and sampling distance are ultimately dependent on the context of the movements the animator is trying to complete, we allow the user to customize these settings.

3.2. Predictive Time Interval

The choice of an appropriate time interval for predictive sampling is important. A short predictive interval may not lead to a significant change in character state and a large interval may miss intermediate character states that are potentially useful. For computationally intensive simulations, the interval length is limited by the need to maintain interactivity while also generating the predictive samples; longer intervals being more computationally expensive. Performance can be optimized by stopping the predictive simulations short of their interval if a state condition fails, for instance, if the character falls over.

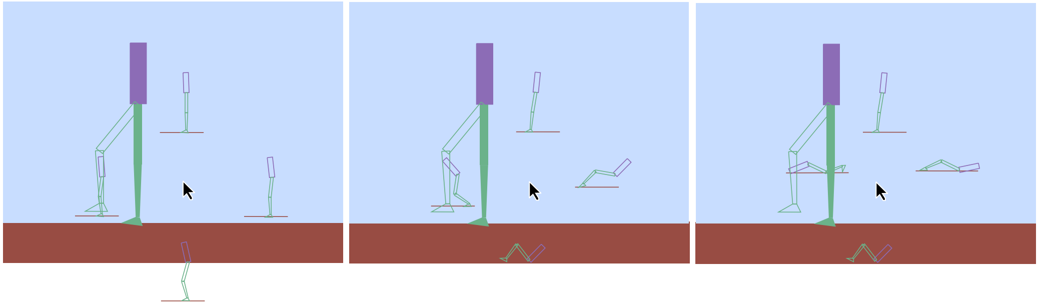
Equally important in the choice of the predictive interval is the issue of user interactivity. Large look aheads generally result in more stable user input at the cost of performance spontaneity. Smaller time steps on the other hand make the interactive control more sensitive but impose a greater cognitive and motor load on the user. The ideal prediction interval may vary significantly with character state and animator intent. To accommodate this, we choose to provide direct control over the current interval via the mouse wheel. Figure 4 shows the difference in predicted character state as the interval is increased, illustrating the limited useful window for interval length. Look-aheads, or predictive simulations, of 0.3s or 0.4s are a good default with intervals in the range of 0.2s to 0.5s being the norm and predictions of about 1s being the maximum. Some tasks, such as bipedal balance, can be achieved with a predictive interval as short as 0.1 s.

3.3. Presenting Feedback

A number of predicted character states must be visualized for the user so that he can easily understand the input necessary to achieve a desired state, while also not being distracted from the overall performance by visual clutter. A user understandable mapping from the presentation domain to the input domain is required. In this paper we use a *spatially coincident mapping* between the cursor input and its displayed predictive sample. Quite simply, a predictive sample is placed at the location a user would need to move his mouse to in order to achieve that state. This arrangement provides the performer with a clear interaction mechanism. A complex input space movement task is effectively transformed into a simpler dynamic selection task. This new task is significantly easier for novice users, and makes it possible for advanced users to complete more difficult movements. For example, an experienced user of the system could not perform the Cossack dance or Luxo head balance shown in the accompanying video until predictive feedback was provided.

Miniature versions of predicted character state are displayed at the location of their respective sample input, as seen in Figure 4. The size of the miniatures is set to reduce visual overlap and clutter for the given number and spatial distribution of predictive samples. Having the predictive samples clustered around the current input is effective because it places the limited number of samples in the context where they can be used: near the users's current input, and under their direct control. Once a user has conceived a motion given the character's current state, it can be performed almost solely by moving towards the predictive samples. For some movements, as motor skills improve, the user relies less on the visual predictive feedback, while for more complicated movements, even advanced users require predictive feedback. While the visual layout may appear distracting to observers, in practice users appear able to transition focus smoothly between predictive previews and the primary animation as required by the performance.

Adaptive presentations are used to highlight particularly useful predictions. Two display adaptations are used: culling, where predictions that fail a condition, such as leaning over too far for a balance task, are not displayed; or colour highlighting, where predictions that do particularly well are shown using a brighter colour. Sometimes a user must quickly scrub through the input space to search for input that will lead to a desired result. The display adaptations are particularly useful in this case as they make useful input locations visually jump out as they are discovered.



(a) Predictive interval = 40ms (b) Predictive interval = 520ms (c) Predictive interval = 1780ms

Figure 4: *The effect of different choices of predictive time interval on visual feedback. i) Too small a predictive interval. ii) Useful interval size. iii) Too long a predictive interval.*

3.4. Physical Simulation Specifics

Thus far none of concepts presented are simulation specific. We now describe specific issues to the formulation of predictive feedback for physically simulated characters.

While it is not necessary that the predictive oracle be the same as the simulation mechanism used for the interactive character control, we would like the predicted results to be as accurate as possible. In this paper, the same physical simulator is used for the oracle and character control, providing a "perfect" prediction mechanism.

A 2D physical simulation system has been implemented following Laszlo et al. [LvdPF00]. The control input specifies desired states or poses for the character. A Proportional Derivative (PD) actuator at each DOF is used to move the character to these desired states. PD control can be thought of as a spring and damper system, expressed as:

$$\tau = k_s(\theta_{desired} - \theta) - k_d\dot{\theta}, \quad (1)$$

where τ is a torque, θ is the current angle of the DOF, $\theta_{desired}$ is the desired angle of the DOF, $\dot{\theta}$ is the angular velocity of the DOF and k_s and k_d are the spring and damper gains respectively. The controller generates a torque to reduce the error between the actual and desired angle for the DOF.

Simulation based predictions are perfectly context sensitive. Since the actual simulation and the predicted poses use the same simulation code and initial state, the predictions are always an accurate reflection of what the future state of the character will be for given control parameters. This allows the animator to react to and to incorporate unanticipated events such as bumps and mis-steps into their performance. The primary disadvantage of simulation based prediction is maintaining the interactivity of overall simulation due to computational costs.

The dynamic simulation code must be designed carefully in order to allow the primary simulation to be properly interrupted and resumed, and to perform multiple predictive simulations. In particular, in saving, initializing and restoring the state of the simulation, all internal simulator state information should be included along with the state of the simulated character itself. Examples of internal simulation state that might not normally be considered include: contact point state (if using a penalty method with *friction cone* Coulomb friction model approximation), internal integrator state, variable environment parameters and random number generator seeds for any stochastic elements that might affect the predictive results. In general, any state information that has an effect on the final animated motion must be included.

From the user's point of view, the predictive simulation occurs in parallel with the primary simulation. Indeed, the problem is well suited to a multi-processor implementation. In our particular single processor implementation, however, we choose to time-multiplex the primary and predictive simulations, explicitly interleaving them, so that the proportion of time spent in each is explicitly controlled. Simulation proceeds in chunks by first computing the primary simulation for some fixed time interval then pausing it, saving all associated state, performing one or more predictive simulations, storing the results then restoring and resuming the primary simulation.

It is possible to provide a relation that indicates the number and length of predictions that can be accommodated, if the average cost of simulation is A seconds per simulation second (for simplicity, the cost of swapping state will be ignored). Where appropriate, we will use capital letters to refer to real time and small letters to refer to simulation time. Let K be the period of one update cycle ($1/K$ is the frame rate, typically 15 or 30 fps). The input space is sampled at the beginning of the interval, the predictions and primary simulation are updated based on the input and the display is updated at the end of the interval. Hence, the latency is simply K. $K = T_p + n * L$ where T_p is the time spent on primary simulation, L is the time spent on each prediction and n is the number of predictions. The simulation rate R is how quickly the simulation is run. One third real time is a typical value. $R = \frac{t_p}{T_p + n*L}$. Given that $T_p = A * t_p$ and $L = A * l$,

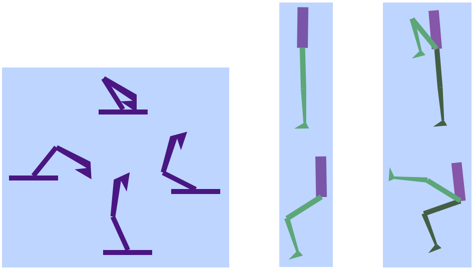


Figure 5: *Control poses for Luxo are shown on the left. The vertical biped poses for the squat are shown in the middle and for the Cossak dance on the right.*

$T_p = RAK$ and the relationship between the number of predictions and the prediction interval is $n * l = K(1/A - R)$. For a given latency, a tradeoff must be made between the number of samples and the maximum look ahead interval that can be accommodated.

The main simulation rate effectively defines a tempo for the animator's control which we allow them to vary during simulation. The tempo they can attain varies with skill, and is tightly coupled with the character and the actions being performed. Small, light characters, as we use, move more quickly than more massive characters. For this reason, animators often work at less than "real time" rates with these models. The ultimate criterion is simply what motion looks good given the animator's aesthetic desires.

3.5. Mapping Input Domain to Control Domain

At any point in time, the user's 2D input is used to generate a single desired pose in the control domain and PD-control moves the character to this pose. It is possible to define multiple input-to-control mappings for a character and use keyboard shortcuts to transfer between them.

A single mapping is used with the Luxo lamp character which linearly interpolates the four poses shown in Figure 5. For the biped character, left and right movements adjust the ankle angle to cause the character to lean in the direction of mouse movement. The vertical dimension is varied depending on the intended movement. The vertical hip and knee poses for the squat and Cossack dance are shown in Figure 5. The tab key is used to switch the legs used in the poses while performing the Cossack dance.

New mappings can be interactively defined to expand the character's action repertoire. Due to dynamic ground interactions, the set of movements a mapping supports is much larger than the kinematic poses in the control space. For example, Luxo's backflip is not explicitly represented, but emerges from the combination of correctly timed pose changes and ground interaction forces. Predictive feedback allows an animator to quickly discover the possible motions that can be generated with a given mapping, thus evaluating a mapping's utility. The spaced samples essentially provide a wide brush, allowing the user to quickly explore swaths of input space.

Predictive feedback is not restricted to the simple linear control mappings used here. Arbitrary mappings can be accommodated provided that the control information generated by the user's movement through the input domain towards the prediction is comparable to the control information used to generate that prediction. The input-to-control mapping can be non-linearly warped and autonomous controllers for activities like walking can also be embedded in the input domain so that user input can be employed to regulate them.

4. Predictive Feedback Based Automatic Control

Access to multiple predictive samples opens up the possibility of using them for automatic control by using the predictions to construct local linear models of the character's control space. More specifically, the generated predictions can potentially automate control in cases where the attributes of the character's desired behaviour can be defined. An example behaviour might have the character maintain his torso over his feet. The current predictive samples can be linearly interpolated to yield control input that will achieve this objective. For example, look ahead poses of the character leaning to the left and to the right can be interpolated to compute the ankle joint control required to maintain the character's balance, as in the upright torso pose in Figure 1d. We call this *predictive feedback based automatic control*. Note for this technique, any desired attribute of a character's state is only locally optimized based on the evaluated predictions. Stated differently, the region of the input domain from whence the predictive samples are selected is still interactively specified by the animator and the solution is limited to this local domain. Automatic control thus refines coarse input from the animator to achieve a desired state.

Specifying a desired behaviour allows automatic control of some DOFs while the animator can still control others. This simplifies the animator's task by allowing control to be layered. For instance, the ankles can be automatically adjusted to maintain an upright torso, affecting balance control, while the animator interactively moves a character from a crouch to a stand in order to generate a jump.

Figure 6 illustrates automatic balance control using predictive feedback. In manual control the horizontal mouse cursor position is mapped to the desired ankle rotation. In automatic control the ankle rotation is set based on the value of the red dot. This is determined by estimating a location between the three predicted poses that will have an upright torso and then interpolating the input locations that yielded these predictions to determine the input that corresponds to the estimated pose. The mouse cursor defines the input region used to select the predictive samples and they must be

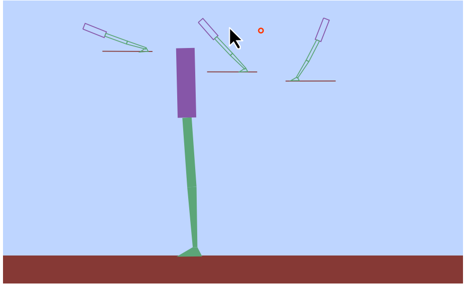


Figure 6: *Interpolated automatic control. The red dot shows an automatically interpreted input location used to control the ankles based on the left and right predictive evaluations.*

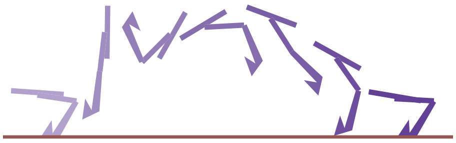


Figure 7: *Frames from a complicated animation sequence in which Luxo lands on his head.*

within a linear region in order for automatic control to operate correctly. For example, if in one prediction for the balance task, the character has already fallen over, interpolating the states may not yield an effective control pose.

5. Discussion

5.1. Results and Usability

The accompanying video shows numerous examples of an expert user interacting with the system and the resulting animations, including dynamically challenging tumbling motions as in Figure 7 and a Cossack kick dance. Automatic control is shown for the difficult task of biped balancing, allowing the animator to adjust the character's posture from standing to crouching. Varying the timing affords a range of movements including squatting, jumping and back flips.

While a formal user study is still to be completed, the system has been exercised by numerous novice users along with an expert user that has extensive experience with and without predictive feedback. Novice users often found that certain movements that they could not complete without predictive feedback became trivial with it, given simple initial instructions. Biped balancing is an example of this. Experience also suggests that novices find control of crouching and jumping, rotation rate to prepare for landing, and the correct execution of a landing fairly straightforward with predictive feedback, but challenging or impossible without it. It is unclear if predictive feedback aids a novice user to learn the motor movements required in input space to complete a movement without feedback. Worth noting, predictive feedback is valuable for experts as well as novices. Without predicitve feedback, many balistic character actions, such as Luxo's flipping, rely heavily on external damping forces to stabilize the performance of energetic actions, making them much more robust to input errors. Predictive feedback enables successful execution of undamped dynamically unstable movements that would otherwise be impossible to perform reliably, such as landing Luxo after multiple aerial rotations, or on his head, or executing the Cossack kick dance sequence, and allows rapid exploration of the much broader range of actions removing this restriction affords.

5.2. Future Work

An important issue is how well the predictive feedback technique can scale to higher DOF characters. The most straightforward way to accomplish this is to employ dimensional reduction where a small set of input DOFs are used to control a much larger set of character DOFs. Dimensional reduction is common in both artificial and natural control. In previous work, Laszlo et al. [LvdPF00] use just 2 DOF to interactively control an 18 DOF, 4-limbed, trotting planar cat. Many previous control solutions targeted at PBA adopt dimension reduction as well, to simplify control parameter specification, making them potential candidates for coupling with predictive feedback in some form. As an example, imagine stabilizing a running controller by using predictive feedback and allowing a user to interactively manipulate the controller's parameters that maintain balance. Predictive feedback provides a potential method for effectively employing these reduced degree of freedom mappings. Preliminary work on the control of 3D characters indicates that the chief obstacle here appears to be providing useful visualizations of 3D movement on a single perspective 2D display. This problem is inherent in any single view based analysis of 3D motion, as is apparent when a coach adopts multiple views to correct athletic performance. Standard approaches are likely to be the most effective, such as allowing the user to vary the viewpoint or compositing multiple viewpoints on a single display.

Another interesting area for future research involves providing feedback on the effect of mode changes. Currently, the predictions are related to the current input-to-control mapping, but a user may change this during interaction, as is done for walking motions, where the user switches which leg is controlled. While learning the correct mode switching in this case is relatively straightforward, it would be very useful to provide feedback indicating the movement possibilities of alternate input mappings while in the current mode.

5.3. Conclusion

We have found predictive feedback to be an effective tool for the interactive control of simulated characters, providing benefit to both novice and expert interactive "performers" alike. While our experiments have been limited to relatively simple input-control mappings and characters, we expect that the technique can be applied successfully to more sophisticated mappings, and characters of greater complexity, through the use of appropriately abstracted interactive controllers. Our hope is that it may help to make the largely untapped possibilities offered by PBA more accessible, particularly in the area of performance-driven animation.

Acknowledgements

Thanks to the anonymous reviewers for their thoughtful feedback, including suggesting the 3D manipulation example. Thanks also to members of the DGP lab, past, present, and extended, for valuable discussions on the work.

References

[Bar91] BARBA E.: Theatre anthropology: First hypothesis. In *A Dictionary of Theatre Anthropology: The Secret Art of The Performer*, Barba E., Savarese N., (Eds.). Routledge, London, 1991.

[Coh92] COHEN M. F.: Interactive spacetime control for animation. *Computer Graphics (Proceedings of SIGGRAPH 92) 26*, 2 (July 1992), 293–302.

[DJ03] DENG Z., JÄGERSAND M.: Predictive display system for tele-manipulation using image-based modeling and rendering. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS 2003)* (Las Vegas, Nevada, US, October 27 - 31, 2003 2003).

[DYP03] DONTCHEVA M., YNGVE G., POPOVIĆ Z.: Layered acting for character animation. *ACM Transactions on Graphics 22*, 3 (July 2003), 409–416.

[FP03] FANG A. C., POLLARD N. S.: Efficient synthesis of physically valid human motion. *ACM Trans. Graph. 22*, 3 (2003), 417–426.

[FvdPT01a] FALOUTSOS P., VAN DE PANNE M., TERZOPOULOS D.: Composable controllers for physics-based character animation. *Proceedings of SIGGRAPH 2001* (August 2001), 251–260.

[FvdPT01b] FALOUTSOS P., VAN DE PANNE M., TERZOPOULOS D.: The virtual stuntman: dynamic characters with a repertoire of autonomous motor skills. *Computers & Graphics 25*, 6 (Dec. 2001), 933–953.

[HWBO95] HODGINS J. K., WOOTEN W. L., BROGAN D. C., O'BRIEN J. F.: Animating human athletics. *Proceedings of SIGGRAPH 95* (August 1995), 71–78.

[Int98] INTERPLAY: Die by the sword, 1998. Game.

[Lab88] LABAN R.: *The Mastery of Movement*, fourth ed. Northcote House, London, 1988. Revised by Lisa Ullman.

[LCR*02] LEE J., CHA J., REITSMA P. S. A., HODGINS J. K., POLLARD N. S.: Interactive control of avatars animated with human motion data. In *SIGGRAPH 2002 Conference Proceedings* (2002), pp. 491–500.

[LGC94] LIU Z., GORTLER S. J., COHEN M. F.: Hierarchical spacetime control. *Proceedings of SIGGRAPH 94* (July 1994), 35–42.

[LvdPF00] LASZLO J., VAN DE PANNE M., FIUME E.: Interactive control for physically-based animation. In *Siggraph 2000, Computer Graphics Proceedings* (2000), pp. 201–208.

[MAB*97] MARKS J., ANDALMAN B., BEARDSLEY P. A., FREEMAN W., GIBSON S., HODGINS J. K., KANG T., MIRTICH B., PFISTER H., RUML W., RYALL K., SEIMS J., SHIEBER S.: Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proceedings of SIGGRAPH 97* (1997), pp. 389–400.

[Nat04] NATURALMOTION: Endorphin, 2004. Animation s/w.

[NM93] NGO J. T., MARKS J.: Spacetime constraints revisited. *Proceedings of SIGGRAPH 93* (August 1993), 343–350.

[OTH02a] OORE S., TERZOPOULOS D., HINTON G.: A desktop input device and interface for interactive 3d character animation. *Graphics Interface '02* (2002).

[OTH02b] OORE S., TERZOPOULOS D., HINTON G.: Local physical models for interactive character animation. *Computer Graphics Forum 21*, 3 (Sept. 2002), 337–346.

[Per95] PERLIN K.: Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics 1*, 1 (March 1995), 5–15.

[PW99] POPOVIC Z., WITKIN A.: Physically based motion transformation. *Proceedings of SIGGRAPH 99* (August 1999), 11–20.

[She93] SHERIDAN T. B.: Space teleoperation through time delay: Review and prognosis. vol. 9.

[Sim94] SIMS K.: Evolving virtual creatures. *Proceedings of SIGGRAPH 94* (July 1994), 15–22.

[SLGS01] SHIN H. J., LEE J., GLEICHER M., SHIN S. Y.: Computer puppetry: An importance-based approach. *ACM Transactions on Graphics 20*, 2 (Apr. 2001), 67–94.

[sug01] A suggestive interface for 3d drawing. In *Proceedings of the 14th annual ACM symposium on User interface software and technology* (2001), Joe Marks E. D. M., (Ed.), ACM Press, pp. 173–181.

[sug04] A suggestive interface for image guided 3d sketching. In *Proceedings of ACM CHI 2004 Conference on Human Factors in Computing Systems* (2004), Elizabeth Dykstra-Erickson M. T., (Ed.), ACM Press, pp. 591–598.

[TV95] TROY J., VANDERPLOEG M.: Interactive simulation and control of planar biped walking devices. In *Workshop on Simulation and Interaction in Virtual Environments* (1995), pp. 220–224.

[vdPF93] VAN DE PANNE M., FIUME E.: Sensor-actuator networks. *Proceedings of SIGGRAPH 93* (August 1993), 335–342.

[vdPL03] VAN DE PANNE M., LEE C.: Ski stunt simulator: Experiments with interactive dynamics. In *Proceedings of the 14th Western Computer Graphics Symposium* (2003).

[WK88] WITKIN A., KASS M.: Spacetime constraints. *Computer Graphics (Proceedings of SIGGRAPH 88) 22*, 4 (August 1988), 159–168.

[YP03] YIN K., PAI D. K.: Footsee: an interactive animation system. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 329–338.

[ZvdP05] ZHAO P., VAN DE PANNE M.: User interfaces for interactive control of physics-based 3d characters. In *I3D: ACM SIGGRAPH 2005 Symposium on Interactive 3D Graphics and Games* (Apr. 2005).