Composeable controllers for physically-based character animation

Petros Faloutsos and Michiel van de Panne
Department of Computer Science, University of Toronto
{pfal,van}@dgp.toronto.edu

April 16, 1999

Introduction

Physically-based characters which are equipped with a large set of basic motion skills are as yet unachievable. It is somewhat surprising that this is the case, given the string of successes in implementing specific controllers for complex dynamic motions such as walking and running, as well as for motions such as platform diving and various gymnastic manoevres. While it is clear that a divide-and-conquer strategy is needed in order to cope with the inordinate number of motions that humans and animals may perform, there has been little thought given to how the resulting control solutions can be integrated together into a single composite controller. Consider, for example, that researcher A designs a controller for the walking of a human character and researcher B designs a controller for the running of the same model. It would be most useful if they could share their controllers with each other (perhaps through an email exchange) and easily build an integrated controller that can both walk and run. Part of this idea is captured by the notion of constructing a library of controllers, but in order to leverage the talents of a group of people, it needs to be a library that everyone can contribute to and use.

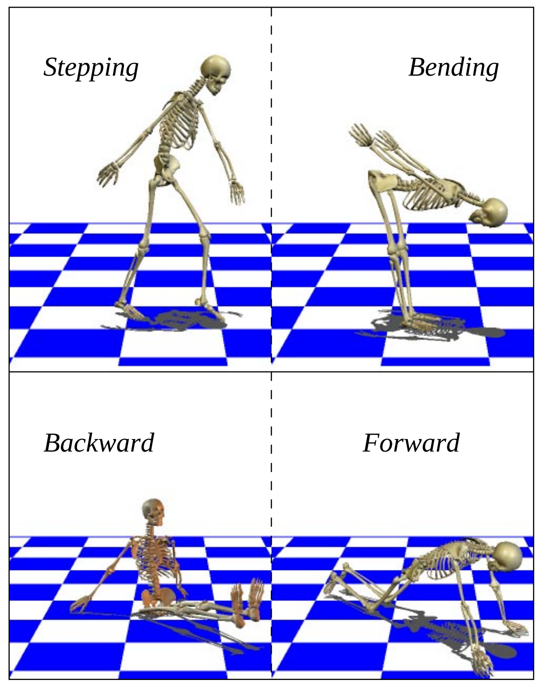

Figure 1: Responses of a human character to various pushes.

Controller abstraction

There are many different forms of control algorithms. A desired composition method should be based on an abstraction that facilitates composeability without limiting the design of the individual controllers. Ideally we want a method that treats controller algorithms as black boxes. The method we propose satisfies these requirements. Our method requires controllers to define *preconditions*, *postconditions* and *expected performance*. *Preconditions* are a set of conditions over the state of the character and the environment. If these conditions are met then the controller can operate and lead the character to satisfying the *postconditions*. The *postconditions* define a range of states for the final state of the character after the execution of the controller, assuming that the preconditions were met. In other words the controller realizes a mapping between a range of input states to a range of output states for the character. Because of unexpected changes in the environment this mapping may not always succeed, which motivates the notion of *expected performance*. The controller should be able evaluate its performance in order to detect failure at any point during its operation. To

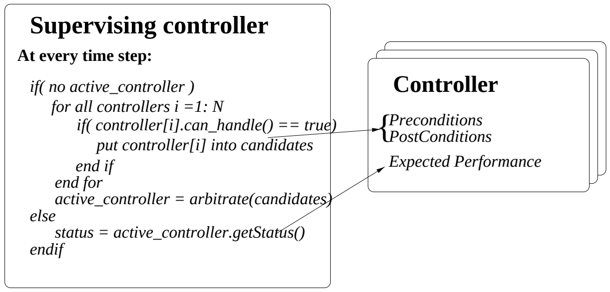

Figure 2: Controller selection and arbitration during simulation.

do this, the controller must have a knowledge of the current and expected state of the character or the environment at all times. Controllers that meet these three requirements can form a pool of available controllers managed by a simple supervising controller. Controllers whose preconditions are satisfied by the current state of the object and whose postconditions are satisfied by the desired state are candidates for becoming operational. In case where more than one controllers are capable of handling a given situation, it is not clear whether an arbitration or a blending method is more suitable. Determining an effective arbitration method is one of our research goals. Similarly, defining the preconditions, postconditions and expected performance for complex characters, motions, and environments is a difficult task and one of our research goals as well. Figure 2 presents an overview of how the supervising controller works and its interaction with the individual controllers at every time step of the simulation.

Another issue in composing controllers is the large number of transitions between different motions. In general, n motions require n^2 transitions which can be very difficult to implement. In our method transitions between two controllers can be provided both by the supervising controller and the individual ones. The supervising controller can realize transitions by choosing in advance a controller whose post conditions overlap with the preconditions of the next one or by using a third controller that can perform the appropriate transition. Individual controllers can facilitate transitions by providing a number of different ending states when appropriate.

Preliminary results

Our prototype system aims to implement a composite controller that is capable of handling a large number of everyday tasks. Such tasks include walking, balancing, bending, falling, and sitting on a chair. While motion capture solutions based on blending and warping techniques may be better in the short-term for such tasks, controller based approaches reveal more about the physics, planning and control of such motions and are the basis of more general solution. We have begun our implementation with the simple tasks of standing, recovering balance when pushed and falling. We designed a number of controllers based in part on experimental studies of how humans detect loss of balance [3] and extensive analysis of protective and falling behaviors [1]. The resulting parameterized controllers have been enhanced with appropriate *preconditions, postconditions*, and *expected performance* and have been integrated using an arbitration-based supervising controller. Figure 1 shows initial results from our experiments, in which a human figure responds to pushes of different magnitude and direction. Depending on which controller's preconditions are satisfied, the character either reacts with balance restoring behaviors (upper row in the figure) or falls in a natural way (lower row).

Our system is built on top of DANCE [2], an open, plug-in architecture that is well suited for exchanging controller code in the form of plug-ins. DANCE software requirements pose no restrictions on the software design of a particular control technique.

Conclusion

In conclusion, we have proposed a general composition method for physically-based controllers. We believe that our method can be the basis of a framework that will allow researchers to exchange and compose controllers. The collective effort of a large number of experts may provide an effective means to build controllers capable of complex behaviors.

References

[1] M. C. Do, Y. Breniere, and P. Brenguier. A biomechanical study of balance recovery during the fall forward. *Journal of Biomechanics*, 15(12):933–939, 1982.

[2] Victor Ng and Petros Faloutsos. Dance: Dynamic animation and control environment. Software system, URL: http://www.dgp.toronto.edu/DGP/DGPSoftware.html.

[3] Yi-Chung Pai and James Patton. Center of mass velocity-position predictions for balance control. *Journal of biomechanics*, 30(4):347–354, 1997.