Modular Dynamic Response from Motion Databases

Jason Mallios, Neil Mehta, Chipalo Street, Odest Chadwicke Jenkins*
Brown University

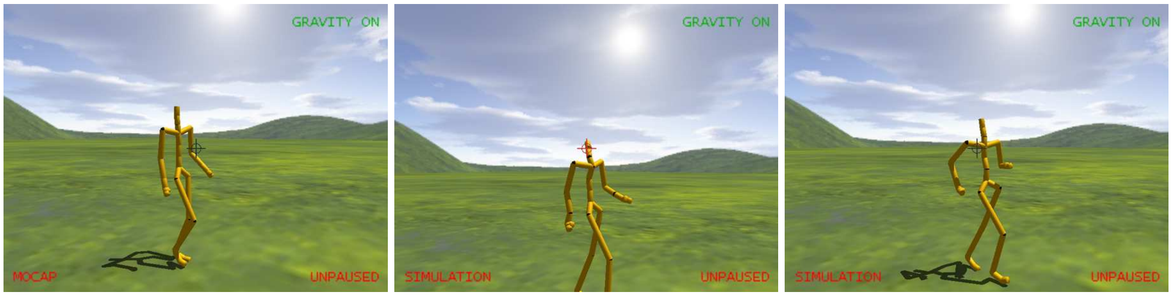

Figure 1: An example of dynamic response. During mocap playback (left), force is applied to the humanoid (center) and controlled via simulation to transition back into mocap playback (right). Example of motion capture playback in our system.

1 Introduction

Animation of humanoid figures is a significant component in many current applications (e.g., video games, movie-making). However, the process of creating viable animations can be tedious, time-consuming, and expensive due to the complexity of controlling a character through a large number of degrees-of-freedom. Such difficulties can be further compounded when the character is subject to external force, as often the case in video games. Recent *dynamic response* methods by Zordan et al. [Zordan et al.] and Mandel [Mandel] go beyond limp passive "ragdoll" animation by transitioning between motion capture playback and controlled physical simulation. When an unanticipated external force is applied to the character, a search is performed in the motion capture database to find the closest matching pose to the character's current configuration. The found pose serves as a desired configuration for the character to servo towards and transition back into motion capture playback. Current dynamic response methods use a monolithic motion database that: 1) requires a significant computation burden for search (70% of computation time for Zordan et al.) and 2) does not readily incorporate user input. We address both of these limitations by using a modular collection of motion databases each representing some action (e.g., run, punch, kick). Through modularity, we invoke smaller independent search procedures on each database, where the choice of desired pose is informed by the action represented by each database. Eventually, we envision dense motion databases constructed from learned parameterized models [Jenkins and Matarić ; Kovar and Gleicher].

2 Implementation and Results

A real-time humanoid simulation (shown in Figure 1) was implemented using the Open Dynamics Engine for dynamics and G3D for rendering. The humanoid kinematics and motion capture databases were automatically parsed and constructed from ASF/AMC motion capture files obtained from the CMU Motion Capture Data Base[1]. Motion capture data was manually prepartitioned into actions (for walk, run, jump, and punch) and stored. Each database maintained an associated KD-tree for approximate searching on windows of motion centered on each pose. Transitions between motion capture databases were found *a priori* by searching for inter-database motion windows within a chosen tolerance.

The implemented simulation is a proof-of-concept illustrating the use of modular dynamic response. Motion capture playback is performed by traversing the inter-database motion graph. The user can apply external force to the humanoid using a crosshair from the camera viewpoint (as shown in Figure 1). This external force places the humanoid into simulation mode. A search is performed to find the desired configuration $\theta_{desired}$ used to generate actuation forces according to a PD-servo at each degree-of-freedom i: $\tau_i = k_s(\theta_{desired,i} - \theta_{actual,i}) - k_d \dot{\theta}_{actual,i}$. The search for $\theta_{desired}$ is executed on a specified set of actions provided by the user. For instance, the user can decide to only search in the scope matching punching and walking motions. We believe that search scope can be properly corresponded to user input (e.g., buttons on a game pad) to yield user desired responses.

The accompanying video demonstrates the real-time dynamic response system. We encountered several issues in the servo and transition system. First, the AMotor provided by ODE did not quite suit our needs and required the implementation of an *ad hoc* custom controller. While usable, this custom controller tends to overshoot and could benefit from inertial scaling used by Zordan. Transitioning into motion capture from simulation led to noticeable pops in the character motion. This artifact is due to the coarseness in the parameter tuning for estimating appropriate transitions.

JENKINS, O. C., AND MATARIĆ, M. J. Performance-derived behavior vocabularies: Data-driven acqusition of skills from motion.

KOVAR, L., AND GLEICHER, M. Automated extraction and parameterization of motions in large data sets.

MANDEL, M. Versatile and interactive virtual humans: Hybrid use of data-driven and dynamics-based motion synthesis.

ZORDAN, V. B., MAJKOWSKA, A., CHIU, B., AND FAST, M. *Dynamic Response for Motion Capture Animation.*

*e-mail: jmallios—nmehta1—cstreet—cjenkins@cs.brown.edu

[1] The database at mocap.cs.cmu.edu was created with funding from NSF EIA-0196217.