Dynamic Control

Jae Eisenmann

CSE 888.x14 Au07

Topics to be Discussed:

- What is Dynamic Control?
- Previous Work in Dynamic Control
- Timing and Tension for Dynamic Characters
- Multiobjective Control with Frictional Contacts

What is Dynamic Control?

- Dynamically controlled characters:
  - Under the influence of forces in their environment
  - Must not only passively react but actively respond to these forces
  - Central Question: How do we combine interactive controllers with physically-based controllers?
    - Intuitive user interface
    - Natural resulting motion

What is Dynamic Control?

- Forces from the Character’s Environment:
  - Gravity
  - Friction
  - Wind/Current
  - Perturbations

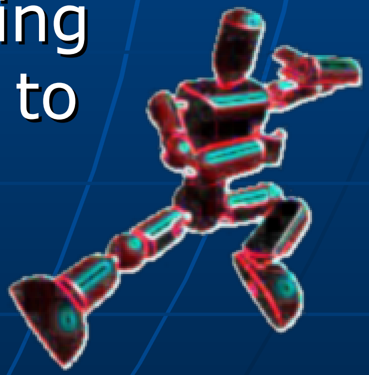
What is Dynamic Control?

- Response to Environmental Forces:
  - Ragdoll Physics (passive)
  - Actuation of Joint Torques (active)

What is Dynamic Control?

- Controllers:
  - Tracking – mimicking mocap data
  - IK - end effector key frames
  - FK - preprogrammed joint angles
  - Behavioral – planning/objectives
  - Physics – rigid body dynamics
  - Various combinations of the above
- Common Problem:

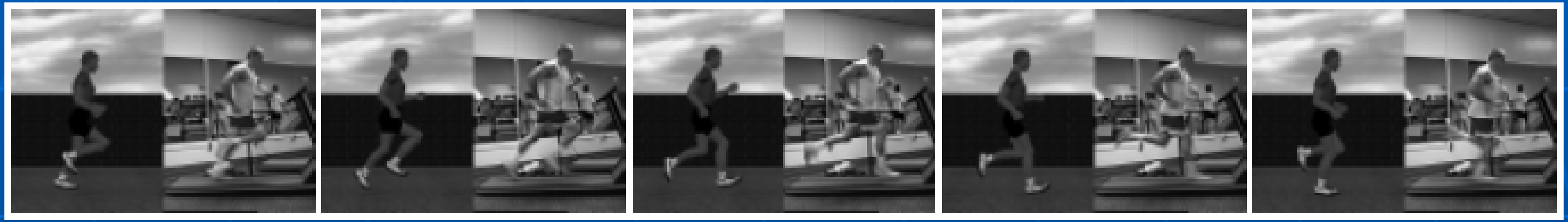
  Controller reuse is difficult when moving from action to action and from model to model



Topics to be Discussed:

- What is Dynamic Control?
- Previous Work in Dynamic Control
- Timing and Tension for Dynamic Characters
- Multiobjective Control with Frictional Contacts

Previous Work in Dynamic Control



- Human Athletic Animation
  (J.K. Hodgins and W.L. Wooten)
  - proportional derivative controllers
    - Running
    - Bicycling
    - Vaulting
    - Diving
  - Sub-real-time performance

Previous Work in Dynamic Control

- Dynamic Legged Locomotion
  (M.H. Raibert and J.K. Hodgins)
  - spring-dampers compute torques
  - requires time-intensive tuning for each new model

Previous Work in Dynamic Control

- Motion Capture-Driven Simulations that Hit and React

  (V.B. Zordan and J.K. Hodgins)
  - Reacts to collisions by changing stiffness/damping terms
  - Trajectory tracking to follow mocap data

Topics to be Discussed:

- What is Dynamic Control?
- Previous Work in Dynamic Control
- Timing and Tension for Dynamic Characters
- Multiobjective Control with Frictional Contacts

Timing and Tension for Dynamic Characters

- Eurographics/ACM SIGGRAPH SCA 2007
- UCLA – Department of Computer Science
- Authors:
  - Brian Allen
  - Derek Chu
  - Ari Shapiro
  - Petros Faloutsos

Timing and Tension for Dynamic Characters

- Overview
  - Method
  - Physical Interpolation of Key-Frames
  - Applications
  - Results
  - Evaluation

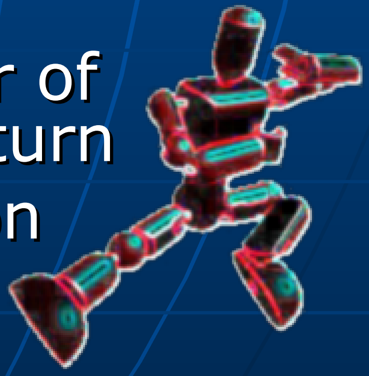
Method

- How can we provide natural-looking motion while honoring time constraints and providing realistic response to perturbations?
- Use traditional proportional-derivative (PD) feedback controllers to interpolate between keyframes:

$$\gamma_k = (\hat{\theta}_k, \hat{\omega}_k, \hat{t}_k)$$

Method

- Compute torque at each joint using knowledge of precomputed net torque at parent joint
- Magnitude of control torque around a joint at each time step: $\tau = k_s(\hat{\theta} - \theta) + k_d(\hat{\omega} - \omega)$
- The PD parameters (k_s & k_d) are continuously altered in order to respond to changes in character state as well as external perturbations
- These parameters are found analytically
- Significant improvement over hand-tuning and heuristic methods
- Tension input is specified by the number of seconds the character should take to return to the target trajectory after perturbation



Physical Interpolation of Key Frames

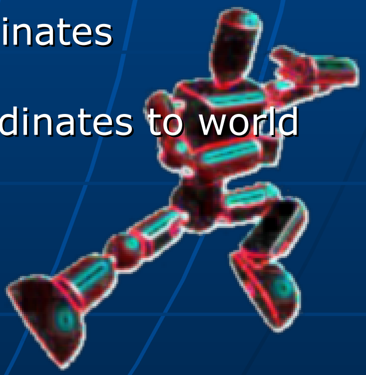
- Analytic equations used to find torques at each time step:

$$m_i = \mathbf{s}_i \cdot \mathbf{D}_i^{i..n} \mathbf{s}_i + M_{i..n} \left\| (\mathbf{c}_i^{i..n} - \mathbf{d}_i) \times \mathbf{s}_i \right\|^2$$

- Where:
  - $\mathbf{c}_i^{i..n}$ combined center of mass (vector)
  - $M_{i..n}$ combined mass of link i and all its children (scalar)
  - $\mathbf{d}_i$ vector from link’s local coordinate frame to joint
  - $\mathbf{s}_i$ joint’s axis (unit vector)
  - $\mathbf{D}_i^{i..n}$ composite inertia tensor of ith joint in link local coordinates

$$\mathbf{a}_i = \sum_{j=0}^{i-1} \frac{\tau_j}{m_j} \mathbf{G}_j^w \mathbf{s}_j$$

- Where:
  - $\mathbf{a}_i$ total angular acceleration at joint i in world coordinates
  - τ_j computed scalar torque around jth joint
  - $\mathbf{G}_j^w$ Transformation Matrix from the jth link local coordinates to world coordinates
  - m_j moment of composite inertia



Physical Interpolation of Key Frames

- Final analytic equation used to find torques at each time step:

$$\tau_i = \frac{m_i}{\lambda^2}(\hat{\theta}_i - \theta_i) + 2\frac{m_i}{\lambda}(\hat{\omega}_i - \omega_i) + a_i \cdot (G_j^w s_i)$$

- Where:

a_i	total angular acceleration at joint i in world coordinates
τ_i	computed scalar torque around jth joint
G_j^w	Transformation Matrix from the jth link local coordinates to world coordinates
m_i	moment of composite inertia
$\hat{t}: \lambda = \hat{t}/n$	the time constant used to ensure that target is reached in time $\hat{t}$
$\hat{\theta}$	desired joint position
$\hat{\omega}$	desired joint velocity
θ	current joint position
ω	current joint velocity

Applications

- Keyframe Animation – input current pose and an array of target keyframes
- Pose Control – (aka keyframe interpolation with keys defined as repeatable poses)
- Tracking Motion Capture – extract keyframe information from recorded motion data

Results

- Performing timed actions in the presence of perturbations
  - Catching
  - YMCA
  - Conducting
- Comparison with hand-tuned PD controller

Evaluation

- Timing constraints are achieved
- Algorithm runs in O(n) time
- Resulting motion does not respond very naturally to perturbations – Instead response is controlled by time input from user
- This is only an incremental improvement from the Zordan and Hodgins paper

Topics to be Discussed:

- What is Dynamic Control?
- Previous Work in Dynamic Control
- Timing and Tension for Dynamic Characters
- Multiobjective Control with Frictional Contacts

Multiobjective Control with Frictional Contacts

- Eurographics/ACM SIGGRAPH SCA 2007
- MIT – Computer Science & Artificial Intelligence Laboratory
- Authors:
  - Yeuhi Abe
  - Marco da Silva
  - Jovan Popovic’

Multiobjective Control with Frictional Contacts

- Overview:
  - Contact Dynamics
  - Multiobjective Control
  - Practical Considerations
  - Results
  - Evaluation

Contact Dynamics

- Contact Mechanics
  - For the case of sustained contact, we can exploit the linear relationship between joint torques, reaction forces, & joint accelerations

Contact Dynamics

- Contact Mechanics

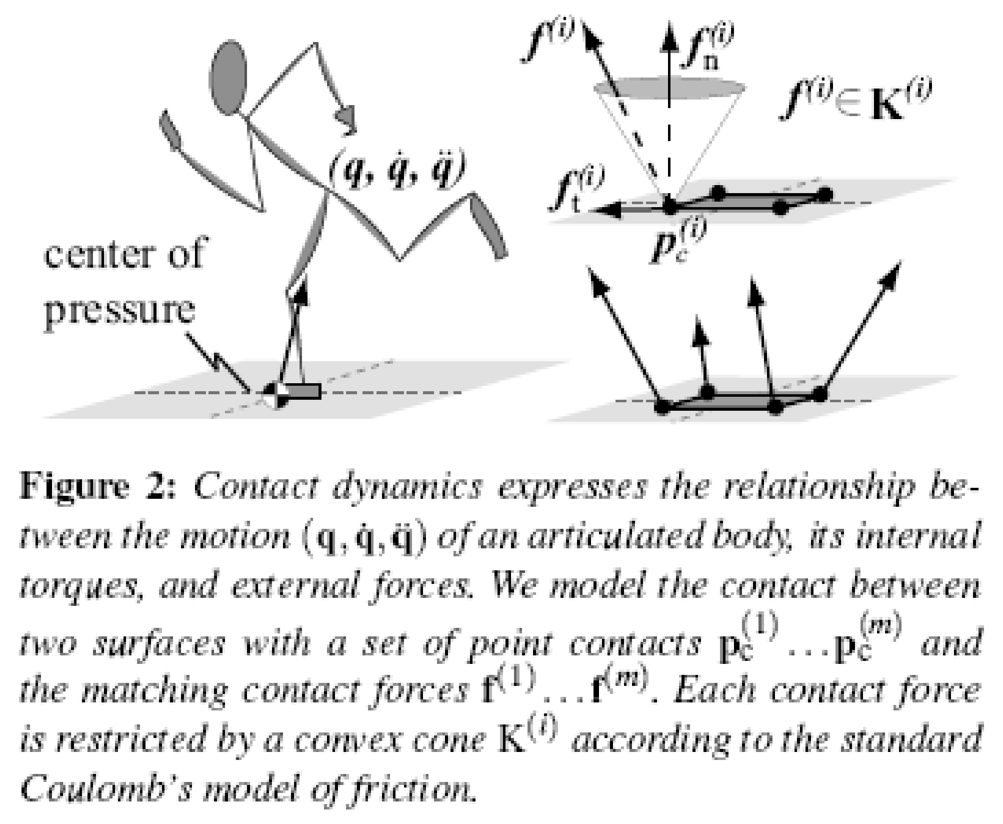
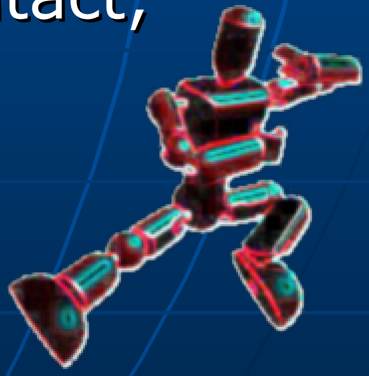


Figure 2: *Contact dynamics expresses the relationship between the motion* $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ *of an articulated body, its internal torques, and external forces. We model the contact between two surfaces with a set of point contacts* $\mathbf{p}_c^{(1)} \ldots \mathbf{p}_c^{(m)}$ *and the matching contact forces* $\mathbf{f}^{(1)} \ldots \mathbf{f}^{(m)}$*. Each contact force is restricted by a convex cone* $\mathrm{K}^{(i)}$ *according to the standard Coulomb's model of friction.*

Contact Dynamics

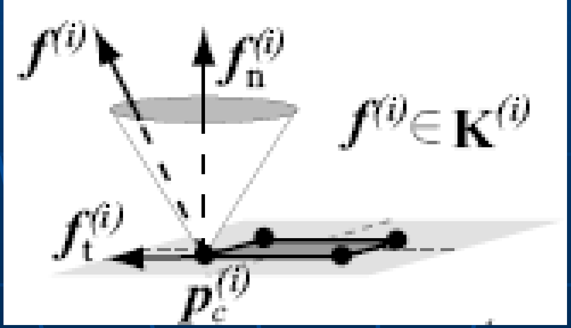
- Contact Mechanics: Constructing the Jacobian $G^{(i)}$
  - Non-slipping contacts with environment restrict the relative velocity of each contact point $p_c^{(i)}$ to zero
  - Expressing this condition in terms of joint velocities and using the Jacobian $G^{(i)}$ to compute body velocity at point of contact, we get:

$$\mathbf{G}^{(i)}\,\dot{q} = \dot{p}_c^{(i)} = \mathbf{0}.$$



Contact Dynamics

- Contact Mechanics: Constructing the Friction Cone $K^{(i)}$
  - Coulomb's friction model limits the tangential component of contact force: $\|f_t^{(i)}\| \le \mu f_n^{(i)}$
  - We gather these limits into the friction cone which limits the direction and magnitude of the contact force: $f^{(i)} \in \mathrm{K}^{(i)} = \{\mathbf{x} \mid \|\mathbf{x}_t\| \le \mu \mathbf{x}_n\}$



Contact Dynamics

- Contact Mechanics
  - A linear map $G^T f$ determines the total joint torque by collecting all the joint forces plus the Jacobian matrices into one vector f and one matrix G

Contact Dynamics

- Active Body Dynamics
  - An active body propels itself using joint torques
  - Joint torques only directly control internal joints, not global position and orientation
  - Thus global position and orientation are unactuated degrees of freedom

Contact Dynamics

- Active Body Dynamics
  - This separation yields two sets of motion equations:

$$M_1(q)\ddot{q} + n_1(q,\dot{q}) + G_1^\top(q)f = u$$

$$M_2(q)\ddot{q} + n_2(q,\dot{q}) + G_2^\top(q)f = 0.$$

  - First two terms: inertial & gravitational
  - Third term: determines total joint torque
  - u represents the torques
  - Manipulation of f is how we accomplish specific objectives
    (but remember: it is restricted by K)

Multiobjective Control

- Optimization
  - Given:
    - Current Pose
    - Current Velocity
  - Compute:
    - Joint Torques
    - Joint Accelerations
    - Contact Forces
  - Maximize: performance of objectives ($g^{(1)} \ldots g^{(L)}$)

$$\begin{aligned} \min_{a,f,u} \quad & \{g^{(1)}, \ldots, g^{(\ell)}\} \\ \text{subject to} \quad & Ma + n + G^{\top} f = \begin{bmatrix} I \\ 0 \end{bmatrix} u \\ & f \in \mathrm{K}, \quad u \in \mathrm{L} \\ & Ga + \dot{G}\dot{q} = 0 \end{aligned}$$

Multiobjective Control

- Optimization
  - Reduces to linear constraint on vector unknowns b/c:

    M, n, & G

    are constant for current pose and velocity
  - Contact forces and control torques are limited by K and L respectively
  - Last equation ensures no-slip condition

$$
\begin{array}{rl}
\min\limits_{a,f,u} & \{g^{(1)},\ldots,g^{(\ell)}\} \\
\text{subject to} & \mathbf{M}a+n+\mathbf{G}^{\top}f=\begin{bmatrix} \mathrm{I} \\ 0 \end{bmatrix} u \\
 & f \in \mathrm{K}, \quad u \in \mathrm{L} \\
 & \mathbf{G}a+\dot{\mathbf{G}}\dot{q}=0
\end{array}
$$

Multiobjective Control

- Quadratic Program
  - Requires either strict priorities for objectives or a combined weighted-sum objective
  - Objectives are of the form:

$$g^{(i)} = \|\ddot{x}^{(i)} - d^{(i)}\| = \|J^{(i)}a + \dot{J}^{(i)}\dot{q} - d^{(i)}\|$$

  - Where J(i) is the Jacobian such that:

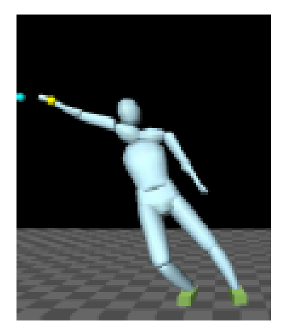
$$\dot{x}^{(i)} = J^{(i)}\dot{q}$$

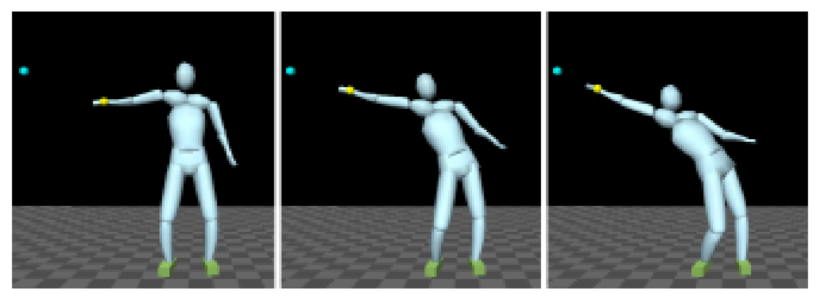
  - Example – when tracking recorded motion trajectories, desired acceleration (d) is:

$$d = k_s\left(m(t) - x\right) + 2\sqrt{k_s}\left(\dot{m}(t) - \dot{x}\right) + \ddot{m}(t)$$

Multiobjective Control

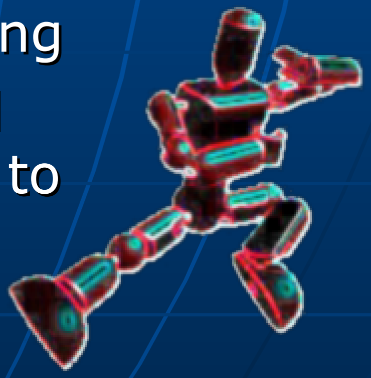
- Control Trade-offs
  - Strict prioritization does not work well in practice because balance tasks usually interfere with other tasks (i.e. tracking recorded motions)
  - The weighted-sum objective approach is a compromise and thus allows more flexibility





Practical Considerations

- Stabilizing Contacts
  - Problem: contacts will break, either by numerical errors or by external disturbances
  - Controller must adapt to these cases
    - Add a minimum threshold to the friction cone (K)
    - Conservative estimation of contact points within contact region to prevent tangential slipping
    - In case of contact breakage from external disturbances, either collapse friction cone to encourage immediate recovery or remove contact point and add a new motion objective for recovery



Practical Considerations

- Maintaining Balance
  - Center of Mass (COM) must be in a generally upright, centered position in order for a character to maintain balance
  - If COM moves out of reasonable position, character may never recover due to underactuation
  - Return trajectory feasibility is given by:

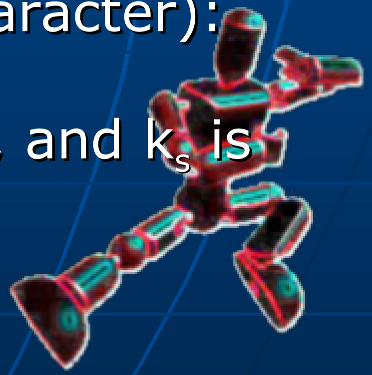
$$M_2(q)\ddot{q} + n_2(q,\dot{q}) + G_2^\top(q)f = 0$$

  - Since this cannot be solved in linear time, a heuristic is used to create another objective that will move the COM towards some point above the mid-point of the two footprints (in the case of a humanoid character):

$$d = k_s\left(x_d - x\right) - k_d\dot{x}$$

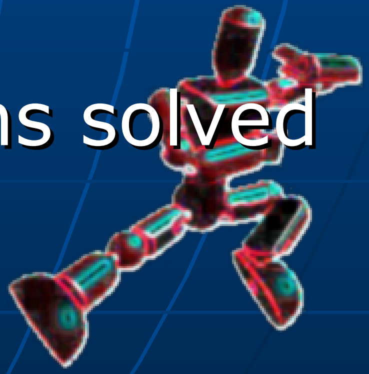
  - Where x_d is desired position, k_d is a constant, and k_s is proportional to:

$$1/\sqrt{\|x_d - x\|}$$



Results

- Implementation Details
  - Models created manually
  - Inertial properties computed from volume of the limbs and standard mass distributions
  - Forward dynamics with frictional contacts were computed with Open Dynamics Engine
  - Quadratic Programming problems solved with MOSEK.

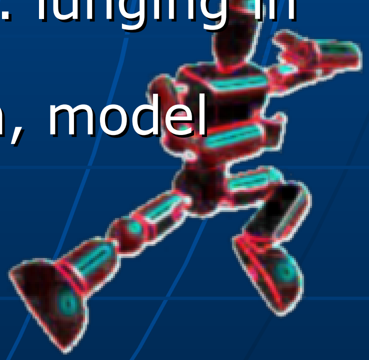


Results

- Sobriety (end effector)
- Pelted (motion tracking)
- Platform (single pose)
- Alien (different model)
- Wall (hand contact)
- Mishap (end effector)

Evaluation

- Intuitive use and artistic direction
  - Motion tracking
  - End effector objectives
- Works well even in complex frictional contact configurations
  - Hand on wall
  - Uneven footing
- Friction cones prevent foot-slip so target character need not match proportions of recorded postures
- Weighted-sum multiobjective formulation allows for natural corrective motion
- Counter-intuitive recoveries occur naturally (i.e. lunging in the direction of the fall to maintain balance)
- Objectives are independent of mass distribution, model geometry, and contact dynamics
- Real-time control
- Controller only works for "standing" poses



In Conclusion…

- Timing and Tension for Dynamic Characters
  - Only incremental advancement
  - Resulting motion not very convincing
- Multiobjective Control with Frictional Contacts
  - Very convincing resulting motion
  - Controller easily adapts to many different models and environments