Linear Bellman Combination for Simulation of Human Motion

Jovan Popović
Adobe Systems
University of Washington
Everyone loves to see stories, but only a few can draw them...

Idea
Everyone

Picture
Artists

Motion
 Animators

Chris Georgenes. Copyright Mudbubble LLC.
... because current methods require moving hundreds of parts...
...and because many tools ignore object interaction

Watch

Designers

Play

Designers + Developers
Physical laws provide systematic rules for automatic animation
Simulated motions are detailed, consistent, ...

Guendelman et al. “Coupling water and smoke to thin deformable and rigid shells,” 2005
...and respond to unpredictable user interactions

Real-Time, PS3

Parker and O'Brien, "Real-Time Deformation and Fracture in a Game Environment," 2009
But, we are missing a systematic approach for character animation

Random Everyday Actions

Heck et al., "Parametric Motion Graphs," 2007
Characters are often animated separately from their surroundings.

Pelted without dynamics.

Original Motion
So rigid bodies respond but animated characters do not
Physically based character animation requires muscle forces

Pelted with dynamics
The challenge is to provide muscle forces that yield desirable motions.

The challenge is to provide muscle forces that yield desired motions

- high-dimensionality
- underactuation
- contacts
- uncertainty
- efficiency
Trajectory Tracking and Bellman Combination

Marco da Silva
MIT

Uldarico Muico
University of Washington
Trajectory-tracking Approach

- reference trajectory optimization
- optimal feedback policy
Reference Trajectory
Dynamical System

$u$  \rightarrow \text{character dynamics} \rightarrow x$
Dynamical System

\[ \tau_1 \tau_2 \tau_3 \tau_4 \tau_5 \tau_6 \tau_7 \tau_8 \tau_9 \tau_{10} \tau_{11} \]

joint torques

\[ u \quad x \]

character dynamics
Dynamical System

\[ x \rightarrow u \rightarrow \text{look-ahead controller} \rightarrow \text{character dynamics} \rightarrow x \]
Look-ahead Control

\[ \sum_{t=1}^{N} \| x_t - \bar{x}_t \|^2_Q + \| u_t \|^2_R \]

minimize tracking error + control effort

subject to character dynamics

\[
\begin{cases}
F(x_{t+1}, u_{t+1}) = x_{t+1} + A_t (x_t - \bar{x}_t) + B_t (u_t - \bar{u}_t) & \text{if } t > 1 \\
\end{cases}
\]

optimal control

linear quadratic regulator
feedback control

character dynamics

\( u \)

\( x \)
Trajectory-tracking control reproduces lifelike motions...
...responds to changes in the environment...
...and produces complex motions
We need to combine several controllers to improve trajectory tracking.
One way to reuse different controls is to apply them in sequence

[Burridge et al 1999] [Faloutsos et al 2001]

get up
get out of bed
comb your hair
go downstairs

state space

goal
goal
goal
Another way is to combine them in parallel...

\[ \pi = w_1 \pi_1 + w_2 \pi_2 \]

short step \hspace{1cm} long step

\[ \pi = w_1 \pi_1 + w_2 \pi_2 \]

state space
\[ \pi = w_1 \pi_1 + w_2 \pi_2 \]

...so that we can modulate accomplished tasks.
\[ \pi = w_1 \pi_1 + w_2 \pi_2 \]

short step  \hspace{1cm}  long step

in-between step

\[ \pi = w_1 \pi_1 \]

\[ \pi = w_2 \pi_2 \]
Combination can also coordinate different strategies.
If coordination is used on strategies with the same goal...

left step
start A

left step
start B

state space

right step
start
...it can enhance regions of competence
Doing this naively generates unexpected results

\[
\pi(x, t) = \pi_1(x, t) + \pi_2(x, t)
\]

short step  long step
Linear Bellman combination provides a closed-form expression

\[ \pi(x, t) = \sum_{i=1}^{n} \alpha_i(x, t) \pi_i(x, t) \]

\[ \alpha_i(x, t) = \frac{w_i z_i(x, t)}{\sum_{j=1}^{n} w_j z_j(x, t)} \]

Combined control can be optimal

Can interpolate previous goals to accomplish new tasks

Robust controls can be created from less robust components
For example, it can be used to interpolate stride length

\[ \pi(x, t) = \alpha_1(x, t)\pi_1(x, t) + \cdots + \alpha_n(x, t)\pi_n(x, t) \]

Linear Bellman Combination can also be used to interpolate the step length of walking controllers
Control goal is described by terminal cost function in state space.

\[ g(x_T) \]

terminal cost
Value function gives the cost to reach the goal from any state

\[ v(x, t) = E[g(x_T) + \int_t^T q(x(s)) + \|\pi(x, s)\|^2 \, ds] \]

terminal cost

integrated cost
Value function equals terminal cost at terminal time

\[ v(x, t) = E[g(x_T) + \int_T^T q(x(s)) + \|\pi(x, s)\|^2 \, ds] \]

- high cost
- low cost

state space

time = T
As we go back in time, the value function varies

\[ v(x, t) = E[g(x_T) + \int_{4t/5}^{T} q(x(s)) + ||\pi(x, s)||^2 ds] \]

- high cost
- low cost

state space

time = 4T/5
\[ v(x, t) = E[g(x_T) + \int_{3t/5}^{T} q(x(s)) + \|\pi(x, s)\|^2 \, ds] \]

- **high cost**
- **low cost**

**state space**

**terminal cost**

**integrated cost**

\[ \text{time} = 3T/5 \]
\[ v(x, t) = E[g(x_T) + \int_{2t/5}^{T} q(x(s)) + \|\pi(x, s)\|^2 \, ds] \]

Terminal cost

Integrated cost

time = 2T/5

state space

high cost

low cost
\[ v(x, t) = E[g(x_T) + \int_{t/5}^{T} q(x(s)) + \|\pi(x, s)\|^2 \, ds] \]
$v(x, t) = E[g(x_T) + \int_0^T q(x(s)) + ||\pi(x, s)||^2 \, ds]$
Optimal policies descend gradient of value function

\[ \pi^*(x, t) = -\nabla v(x, t) \]
Optimal policies descend gradient of value function

\[ \pi^*(x, t) = -\nabla v(x, t) \]
Optimal policies descend gradient of value function

\[ \pi^*(x, t) = -\nabla v(x, t) \]
Optimal policies descend gradient of value function

$$\pi^*(x, t) = -\nabla v(x, t)$$
Value function solves a non-linear partial differential equation with a boundary condition defined by terminal cost function

$$v(x, T) = g(x_T)$$

boundary condition
(goal)

$$- \frac{\partial}{\partial t} v = \min_{\pi} \left[ q(x) + ||\pi||^2 + \mathcal{L}^\pi v \right]$$

Hamilton-Jacobi-Bellman equation

non-linear PDE
Transform a nonlinear form of HJB into a *linear* PDE

\[
- \frac{\partial}{\partial t} z = q(x) z - \mathcal{L}z
\]

Linear Bellman equation

\[
z(x, T) = h(x_T) = \exp(-g(x_T))
\]

boundary condition (goal)

[Todorov 2008]

[Kappen 2005]

[Fleming 1978]

[Holland 1977]
Map cost space into reward space

Cost Space

exp(−v)

Reward Space

value function

non-linear HJB PDE

v(x, T) = g(x)

boundary condition (goal)

reward function

linear Bellman equation

z(x, T) = h(x_T) = \exp(-g(x_T))

boundary condition (goal)
Linearity allows us to add reward functions if $z_1$ solution to the linear Bellman with boundary condition (goal) $h_1$ and $z_2$ solution to the linear Bellman with boundary condition (goal) $h_2$ then

$$w_1 z_1 + w_2 z_2$$

solution to

$$w_1 h_1 + w_2 h_2$$
We now need to find combined controller forces

\[ z = w_1 z_1 + w_2 z_2 \]

\[ \pi = -\nabla v \]
To determine solution in cost space use inverse mapping

\[
\begin{align*}
\text{Reward Space} & \quad \text{Cost Space} \\
\text{combined} & \quad \text{combined} \\
\text{reward function} & \quad \text{value function} \\
z = w_1 z_1 + w_2 z_2 & \quad v = -\log(w_1 z_1 + w_2 z_2)
\end{align*}
\]
To derive our weight equation, apply optimal policy in cost space

Cost Space

combined value function

\[ v = - \log(w_1 z_1 + w_2 z_2) \]

\[ -\nabla v = \frac{w_1 z_1 \nabla v_1 + w_2 z_2 \nabla v_2}{w_1 z_1 + w_2 z_2} \]
The weight of component policy is proportional to reward and goal

\[ \alpha_i(x, t) = \frac{w_i z_i(x, t)}{\sum_{j=1}^{n} w_j z_j(x, t)} \]

\[ \pi(x, t) = \sum_{i=1}^{n} \alpha_i(x, t) \pi_i(x, t) \]

**Combined controller**

**weight equation**

- \( \alpha_i \): time/state varying blend weights
- \( w_i \): fixed goal weight
- \( z_i(x, t) \): reward function
Derivation assumes control-affine dynamics and quadratic control cost

Control-Affine Dynamics

1) \[ dx = \left[ a(x) + B(x)u \right] dt + B(x)dw \]

   passive dynamics \hspace{1cm} \text{control} \hspace{1cm} \text{Brownian noise}

Quadratic Control Penalty

2) \[ q(x) + ||u||^2 \]

   position \hspace{1cm} \text{control cost}
Linear Bellman combination interpolates final dive orientation

\[ \alpha \]
Combination coordinates between different recoveries

Linear Bellman Combination for Controller Coordination
Combination coordinates controllers for different terrains

walk on flat ground $\pi_1(x, t)$ \quad + \quad \pi_2(x, t)$ walk on an incline
While a single component fails to advance

walk on flat ground $\pi_1(x, t)$
Summary

- Trajectory tracking yields controls that reproduce lifelike motions
  - Reference trajectories need to be dynamically consistent
  - Tracking fails for large changes and disturbances
- Linear Bellman combination repurposes individual controllers
  - Interpolation modulates accomplished tasks
  - Coordination enhances overall competence
- Common practical application invalidates theoretical requirements
  - Simulators have discontinuous dynamics at contacts
  - Policies must be in temporal alignment
  - Objective functions must be manually specified
  - Reference trajectories must be derived from common objective functions
Linear Bellman Combination for Simulation of Human Motion

Jovan Popović
Adobe Systems
University of Washington