Towards a general framework for optimizing human movement

Chris Dembia

EE 364b: Convex Optimization II

Stanford University

June 4, 2015
Analyzing human movement
Predicting human locomotion
Outline

The problem

A simple point-mass model of locomotion

A general framework
The optimal control problem

- Find the states $y(t) \in \mathbb{R}^{n_y}$ and controls $u(t) \in \mathbb{R}^{n_u}$ that solve:

$$\text{minimize } J(y(t), u(t), t)$$
$$\text{subject to } \dot{y}(t) = g(y(t), u(t), t) \quad t \in [t_i, t_f]$$

- The performance index $J$ often has the form

$$J(y, u, t) = \int_{t_i}^{t_f} f(y, u) dt = \int_{t_i}^{t_f} (f_u(u) + \gamma f_y(y)) dt.$$

  - The costs $f_u : \mathbb{R}^{n_u} \to \mathbb{R}$ and $f_y : \mathbb{R}^{n_y} \to \mathbb{R}$ are usually quadratic.
  - The scalar $\gamma$ balances the state and control costs.

- The system dynamics $g \in \mathbb{R}^{n_y}$ are in general non-convex.

- This continuous problem is infinite-dimensional.
The discretized problem

- Discretize the state and controls into $T$ mesh points $t_1, t_2, \ldots, t_T$.
- The variables $x \in \mathbb{R}^n$, $n = (n_y + n_u)T$, are:

$$x = (y_1, u_1, y_2, u_2, \ldots, y_T, u_T)^T.$$

- The finite-dimensional problem takes the form

$$\begin{align*}
\text{minimize} & \quad J_d \\
\text{subject to} & \quad y_{i+1} - y_i = h_i g_i(y_i, u_i, t_i) \quad i = 1, 2, \ldots, T - 1.
\end{align*}$$

- The step size $h_i = t_{i+1} - t_i$.
- The integral in the performance index is approximated via the trapezoidal rule:

$$J_d = \frac{1}{2} \sum_{i=1}^{T-1} h_i [f(y_{i+1}, u_{i+1}) - f(y_i, u_i)].$$

- The equality constraints are in general non-affine.
- This method is called direct collocation.
Solution via sequential convex programming

- At each iteration $k$, approximate as a QP about current iterate $x^{(k)}$.
- Performance index approximated as quadratic:
  \[ \nabla_x J_d^{(k)}(x - x^{(k)}) + (x - x^{(k)}) \nabla^2_x J_d^{(k)}(x - x^{(k)}). \]
  - The quantities $\nabla_x J_d^{(k)}$ and $\nabla^2_x J_d^{(k)}$ are evaluated at $x^{(k)}$.
- Constraints approximated as affine:
  \[ c_i^{(k)} + G_i^{(k)}(x_i - x_i^{(k)}) = 0 \quad i = 1, 2, \ldots, T - 1. \]
  - $c_i^{(k)} = y_{i+1} - y_i - h_i g_i(y_i, u_i, t_i)$: Constraint evaluated at $x^{(k)}$.
  - $G_i^{(k)} \in \mathbb{R}^{n_y \times (n_y + n_u)}$: Constraint jacobian for time $t_i$.
- Solution to QP becomes the next iterate $x^{(k+1)}$. 

The problem
Nature of the discretized problem

- Often around 1,000-5,000 variables and equality constraints.
- Extremely sparse (possibly only 2 entries per row in constraint jacobian).
- Most problems are very sensitive to initial guess.
  - May be difficult to get desired balance between feasibility and low cost.
- Usually do not have explicit derivatives (using black box models); decreases convergence rate.
Mesh refinement

- Even if the constraints are satisfied, the error in the ODE solution might be large.
- If the ODE error is above some threshold, e.g., $10^{-3}$, solve the problem again on a finer mesh.
- Alternatively, use a higher order ODE constraint (e.g., Hermite-Simpson).
- With SQP, the hot-start initial guess for subsequent problems doesn't need to be feasible.
Outline

The problem

A simple point-mass model of locomotion

A general framework
Model contains point mass and telescoping leg

(Srinivasan and Ruina, 2006)
The optimal control problem
(Srinivasan and Ruina, 2006)

_variables:
- \( x, y \): position of the mass.
- \( F(t), 0 < t < t_s \): force generated by the leg.
- \( t_s \): Duration of swing.

\( l = \sqrt{x^2 + y^2} \): leg length.

_fixed quantities:
- \( d \): distance to travel in one gait cycle.
- \( v \): walking speed.
- \( t_{\text{step}} = d/v \): duration of gait cycle.

_vary\( d, v \) to explore different gaits.
The optimal control problem
(Srinivasan and Ruina, 2006)

minimize \[ C = \int_0^{t_{\text{step}}} \frac{[F(t)\dot{l}]^+}{mgd}dt \]

subject to

\[
\begin{align*}
    m\ddot{x} &= F\frac{x - x_c}{l} & 0 \leq t \leq t_s \\
    m\ddot{y} &= F\frac{y}{l} - mg & 0 \leq t \leq t_s \\
    m\ddot{x} &= 0 & t_s < t \leq t_{\text{step}} \\
    m\ddot{y} &= -mg & t_s < t \leq t_{\text{step}} \\
    0 \leq l(t) \leq l_{\text{max}} & 0 \leq t \leq t_s \\
    0 \leq F(t) \leq F_{\text{max}} & 0 \leq t \leq t_s \\
    0 < t_s \leq t_{\text{step}} & \\
    x(t_{\text{step}}) - x(0) &= d \\
\end{align*}
\]

periodicity
continuity

\[ 0 < t_s \leq t_{\text{step}} \]

\[ x(t_{\text{step}}) - x(0) = d \]

Solutions obtained with single shooting.

A simple point-mass model of locomotion
Solutions yield three distinct gaits  
(Srinivasan and Ruina, 2006)

- **a) Some possible gaits**
  - Inverted pendulum walk
  - Impulsive run
  - Hybrid intermediate gait: pendular run

- **b) Inverted pendulum walk**
  - Heel-strike
  - Push-off

- **c) Impulsive run**
  - Flight
  - Bounce

- **d) Hybrid intermediate gait: pendular run**
  - Inverted pendulum
  - Flight

A simple point-mass model of locomotion
Solutions yield three distinct gaits
(Srinivasan and Ruina, 2006)

b) Pendular walk  c) Impulsive run  d) Pendular run

A simple point-mass model of locomotion
Solution via direct collocation

- Used the PSOPT direct collocation package.
- PSOPT transcribes the continuous problem into a discretized non-convex problem.
- SNOPT solves the discretized problem using SQP.
- Choosing a smooth approximation to the integrand of cost $C$ improves convergence.
def endpoint_cost(states, controls):
    return 0

def integrand_cost(states, controls):
    L = sqrt(x^2 + y^2)
    return max(F * L, 0) / (m * g * d)

def differential_algebraic_equations(states, controls):
    derivatives[0] = xd
    derivatives[1] = yd
    derivatives[2] = 1 / m * (F * x / L)
    derivatives[3] = 1 / m * (F * y / L) - g
    path[0] = L
    return derivatives, path

def linkages(initial_states, final_states):
    # continuity, periodicity
Recovered same three gaits as Srinivasan and Ruina

inverted pendulum walking
\[ D = 0.5, V = 0.5 \]

impulsive run
\[ D = 0.5, V = 1.5 \]

pendular running
\[ D = 1.5, V = 0.5 \]

A simple point-mass model of locomotion
Outline

The problem

A simple point-mass model of locomotion

A general framework
A more complex walking model

- 7 degrees of freedom.
- Contact with the ground.
- Modeled in OpenSim.
Multibody dynamics

- State $y$ consists of joint angles $q(t)$ and joint velocities $v(t)$.
- The system’s number of degrees of freedom is the length of $q$.
- Second order dynamics ($F = m\ddot{x}$).

$$g(y(t), u(t), t) = \begin{bmatrix} \dot{q}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} M(q)^{-1}(F_g(q) + F_c(q) + u - V(q,v)) \end{bmatrix}$$

- $M$: mass matrix; describes mass distribution.
- $F_g$: forces from gravity.
- $F_c$: forces from contact (with the ground).
- $u$: control inputs (motor torques).
- $V$: Coriolis acceleration; quadratic in $v$. 

A general framework
General biomechanics framework

- OpenSim provides differential equations
- PSOPT transcribe continuous to discrete
- SNOPT solve discrete problem
OpenSim implements the PSOPT interface

```python
def endpoint_cost(states, controls):
    return opensim.endpoint_costs(states, controls)

def integrand_cost(states, controls):
    return opensim.integrand_cost(states, controls)

def differential_algebraic_equations(states, controls):
    return opensim.differential_algebraic_equations(
        states, controls)

def linkages(initial_states, final_states):
    return opensim.linkages(initial_stats, final_states)
```
Desired capabilities of a biomechanics optimal control package

- Trajectory optimization.
- Minimize effort.
- Track desired joint angles.
- Track desired Cartesian trajectories.
- Permit (physical) constraints between rigid bodies.
- Permit contact between rigid bodies.
- Permit multiple-phase problems.
Double pendulum minimum time

minimize \( t_f \)
subject to dynamics
\[ y_1 = 0 \]
\[ r_T(q_T) = \text{red dot} \]
\[ v_T = 0 \]
\[ -100 \leq u_i \leq 100 \quad i = 1, \ldots, T \]

A general framework
Double pendulum minimum time: motion solution
Double pendulum minimum time: control solution

- Bang-bang control.

A general framework
Double pendulum minimum effort

minimize \[ \int_0^{t_f} u^T u \, dt \]
subject to dynamics
\[ y_1 = 0 \]
\[ r_T(q_T) = \text{red dot} \]
\[ v_T = 0 \]
\[ -100 \leq u_i \leq 100 \quad i = 1, \ldots, T \]
\[ t_f \leq 1 \]

- \( r(q) \) is Cartesian location of end effector.
Double pendulum minimum effort: motion solution
Double pendulum minimum time: control solution

A general framework
Double pendulum joint angle tracking

minimize \[ \int_0^{t_f} \| y(t) - \hat{y}(t) \|^2 dt \]
subject to dynamics
\[ -100 \leq u_i \leq 100 \quad i = 1, \ldots, T \]
Double pendulum joint angle tracking: motion solution

$\gg p^* \approx 10^{-12}$
Double pendulum Cartesian tracking

- End effector tracks a circle.

\[
\begin{align*}
\text{minimize} & \quad \int_0^{t_f} \| r(q(t)) - \hat{r}(q(t)) \|^2 dt \\
\text{subject to} & \quad \text{dynamics} \quad -100 \leq u_i \leq 100 \quad i = 1, \ldots, T
\end{align*}
\]
Double pendulum Cartesian tracking: motion solution
Stick with constraint

- Stepping stone for foot-ground contact.
- State \((x_c, y_c, \theta)\).
- Must apply forces (multipliers) \(F_x, F_y\) to enforce the constraint.

\[
\begin{align*}
\text{minimize} & \quad 0 \\
\text{subject to} & \quad \text{dynamics} \\
& \quad (x_c(0), y_c(0), \theta(0)) = (0.5L, 0, 0) \\
& \quad (x_c - 0.5L \cos \theta, y_c - 0.5L \sin \theta) = (0, 0) \quad 0 \leq t \leq t_f
\end{align*}
\]
Stick with constraint: motion solution
Walking with “hand-of-God” forces

- Tracking experimentally measured walking motion.
- No ground contact.
- Optimize only half the gait cycle, with symmetry constraints.

\[
\begin{align*}
\text{minimize} & \quad \int_0^{t_f} ||y(t) - \hat{y}(t)||^2 dt \\
\text{subject to} & \quad \text{dynamics} \\
& \quad u_L \leq u \leq u_U \\
& \quad y_{\text{left}}(t_f) = y_{\text{right}}(t_0) \\
& \quad y_{\text{left}}(t_0) = y_{\text{right}}(t_f)
\end{align*}
\]
Walking with “hand-of-God” forces: motion solution

A general framework
Other elements of a human movement problem

- Muscle actuation.
- Neural delays.
- Metabolic energy expenditure.