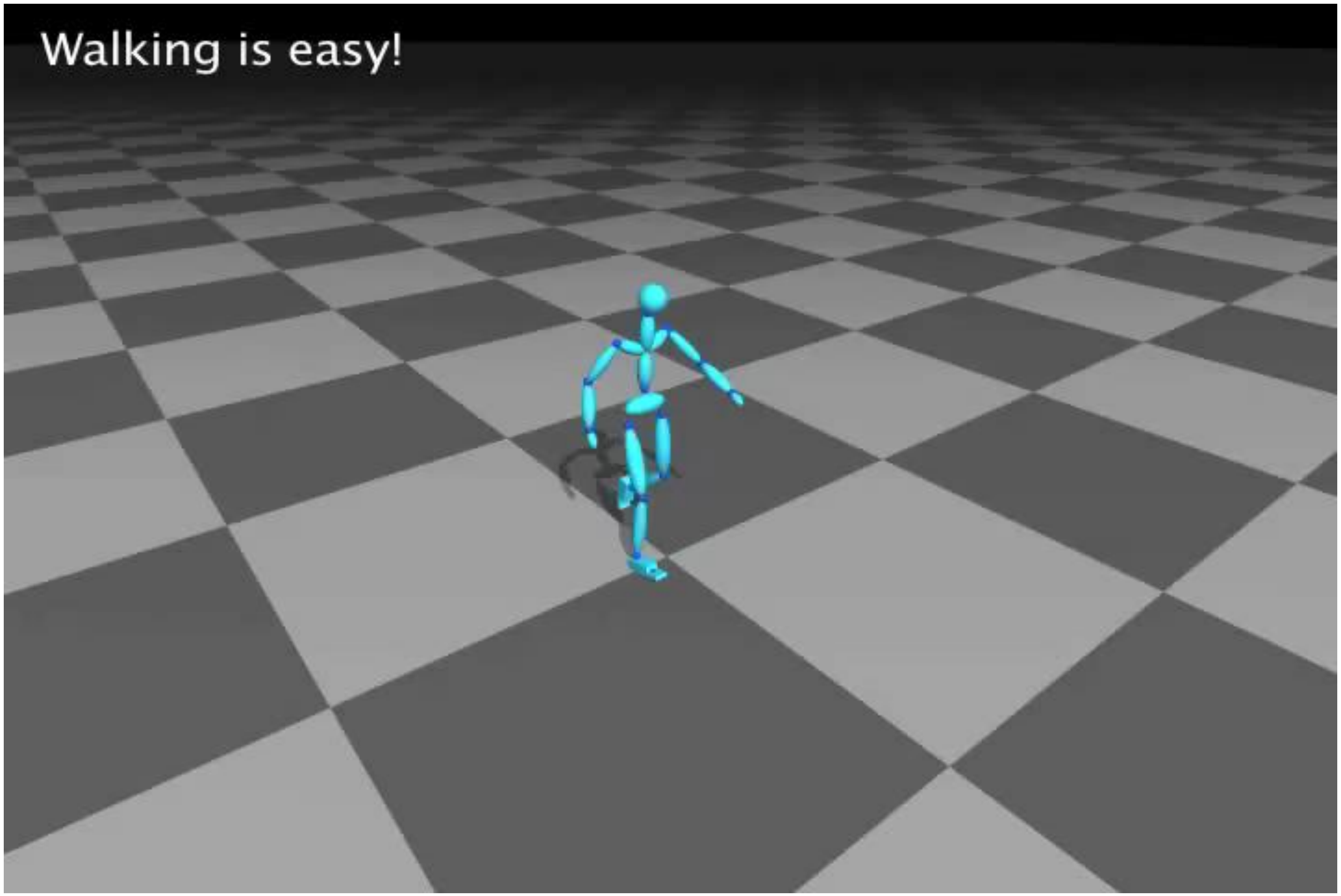
Optimizing Walking Controllers for Uncertain Inputs and Environments

서울대학교
박황필

2012. 8. 14.

Introduction



Sources of Uncertainty

- External forces
  - Wind on a gusty day
  - Pushes suddenly
- User inputs
- Transition between controllers
  - Transition from high to low speeds
- Motor noise

Goal

- Optimize 3D locomotion controllers under uncertainty
  - Find optimized control parameters

- Optimize expected objective function instead of objective function
  - Assume that all unknown quantities are modeled by probability distribution

Controllers

State #	0	1	2	3
Left Foot	Swing	Swing Contact	Stance	Heel-off
Right Foot	Stance	Heel-off	Swing	Swing Contact

$$\tau = k_p \left(\theta_d - \theta\right) + k_d \dot{\theta}$$

- Control parameters
  - PD gains, target angles, start states

Objective function

- Objective(return) function $R(s_{1:T})$
- $R(s_{1:T}) = -\sum_t \sum_i \gamma_i E_i$

$$E_{power} = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{m}\tau_{tj}^2$$

$$E_{ang} = Q(L_x; 0.04) + Q(L_y; 0.05) + Q(L_z; 0.01)$$

$$E_{user} = Q(v_x - \hat{v}_x; 0.05) + Q(s - \hat{s}; 0.05)$$

$$E_{fail} = failed_t$$

Deterministic Cases

- $s_{t+1} = f(s_t, \tau_t, e_t)$

  $$\tau_t = \pi(s_t, c_t; w\ ) = \ k_p(\theta_d - \theta) + k_d\ \dot{\theta}$$

  - s_t : states
  - τ_t : joint torques
  - e_t : external forces
  - c_t : user inputs
  - w : control parameters

- Finding w which maximize $R(s_{1:T})$

- But, e_t, c_t, s_1 are usually unknown

Random Environments

- External forces

$$\mathbf{e}_t \sim p(\mathbf{e})$$

- User inputs

$$\mathbf{c}_t \sim p(\mathbf{c})$$

- A start state of the controller

$$\mathbf{s}_1 \sim p(\mathbf{s}_1)$$

- Motor noise

$$\boldsymbol{\tau}_t \sim p(\boldsymbol{\tau} | \mathbf{s}_t, \mathbf{c}_t, \mathbf{w})$$

- By sampling e_t, c_t, s_1 and τ_t, we can sample an animation sequence

$$\boldsymbol{\tau}_t \sim p(\boldsymbol{\tau} | \mathbf{s}_t, \mathbf{c}_t, \mathbf{w})$$

$$\mathbf{s}_{t+1} = f(\mathbf{s}_t, \boldsymbol{\tau}_t, \mathbf{e}_t)$$

$$\mathbf{s}_{1:T}^{(i)} \sim p(\mathbf{s}_{1:T} | \mathbf{w})$$

Expected Return

- Optimizing *expected return V(w)* instead of $R(s_{1:T})$

$$V(\mathbf{w}) \equiv E_{p(\mathbf{s}_{1:T}|\mathbf{w})}\left[R(\mathbf{s}_{1:T})\right]$$
$$= \int p(\mathbf{s}_{1:T}|\mathbf{w})\, R(\mathbf{s}_{1:T})\, d\mathbf{s}_{1:T}$$

Optimization

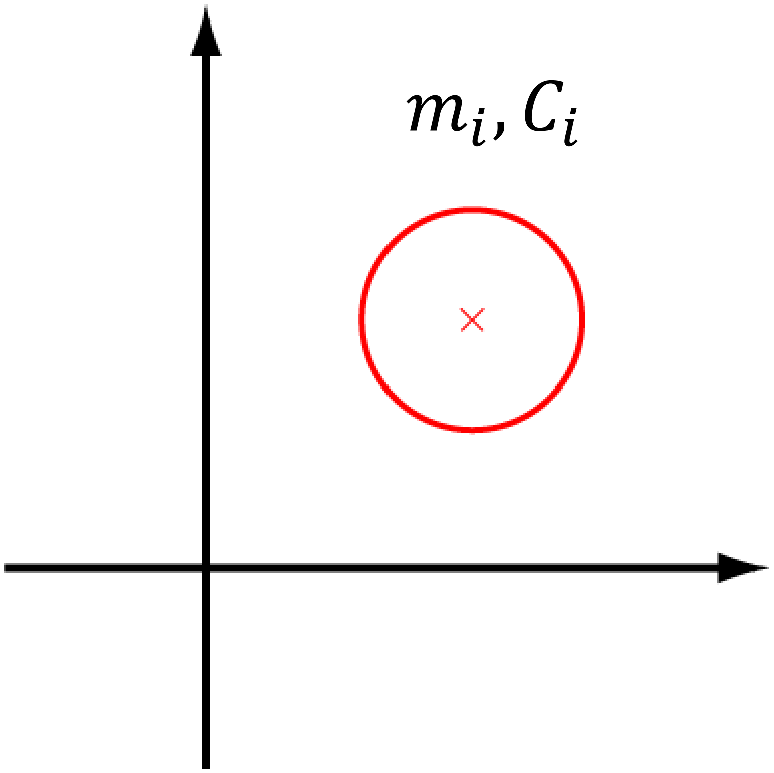
- Since V(w) cannot be computed analytically, we use $\hat{V}(w)$

$$\hat{V}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} R(\mathbf{s}_{1:T}^{(i)})$$

$$\text{where } \mathbf{s}_{1:T}^{(i)} \sim p(\mathbf{s}_{1:T} | \mathbf{w})$$

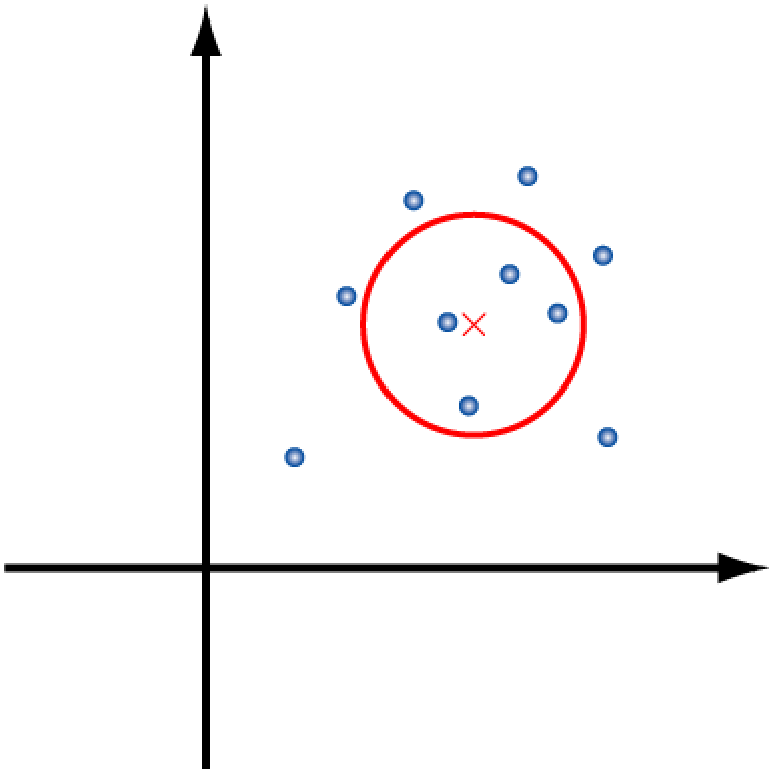
- To optimize $\hat{V}(w)$, we use the CMA (Covariance Matrix Adaptation)

Covariance Matrix Adaptation



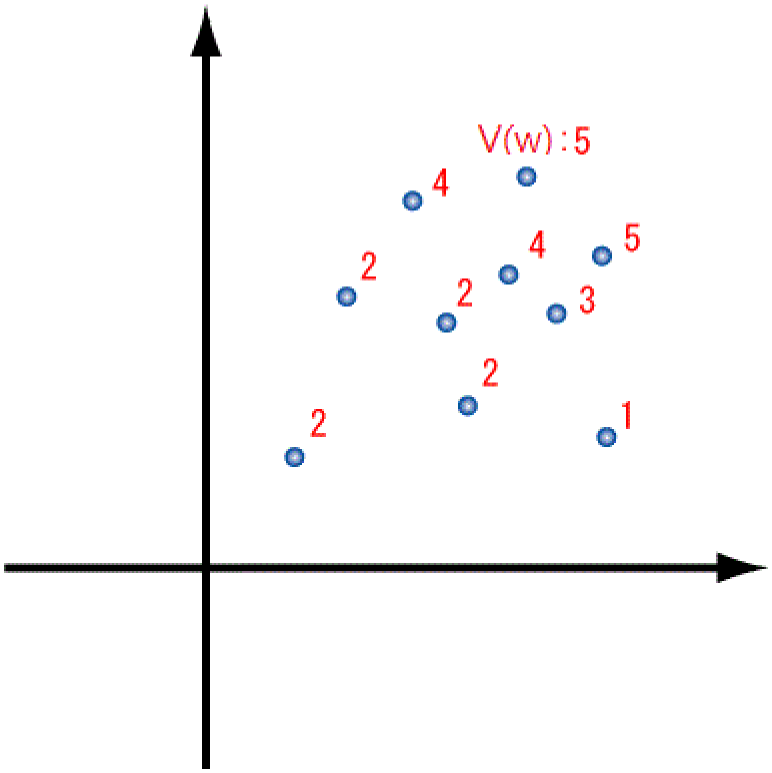
- Sample
- Select elites
- Update mean
- Update covariance
- iterate

Covariance Matrix Adaptation



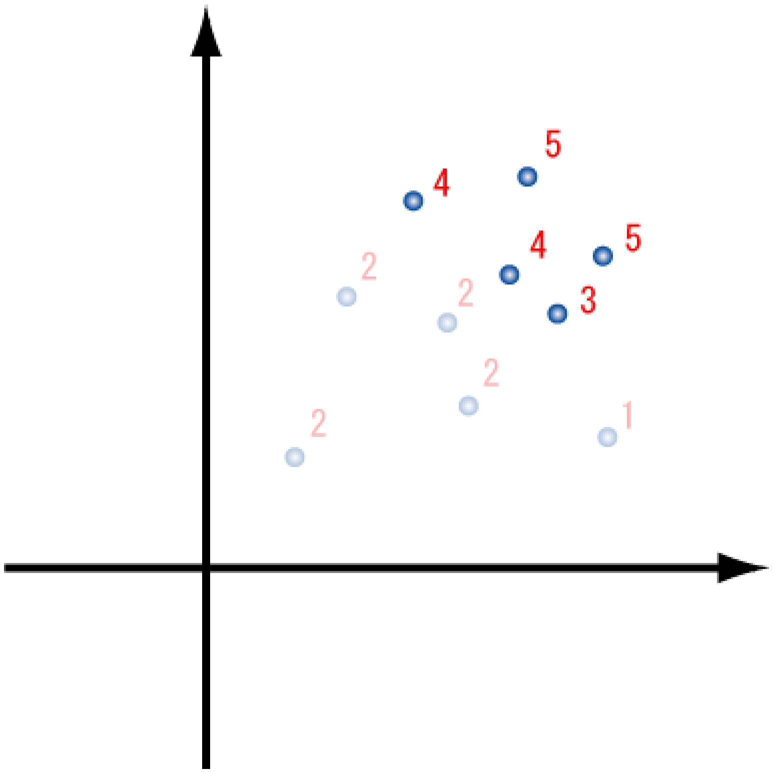
- Sample
- Select elites
- Update mean
- Update covariance
- iterate

Covariance Matrix Adaptation



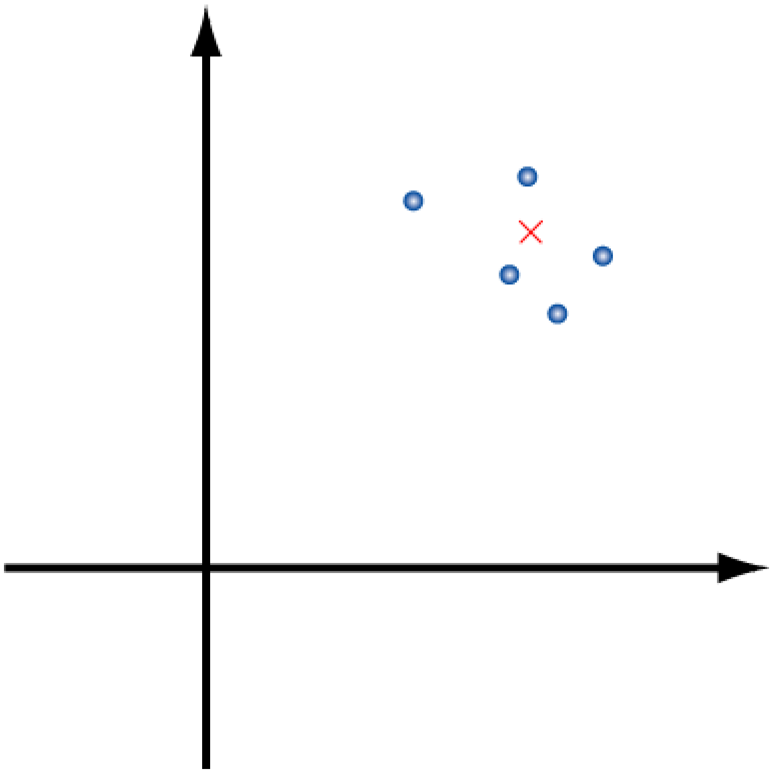
- Sample
- Select elites
- Update mean
- Update covariance
- iterate

Covariance Matrix Adaptation



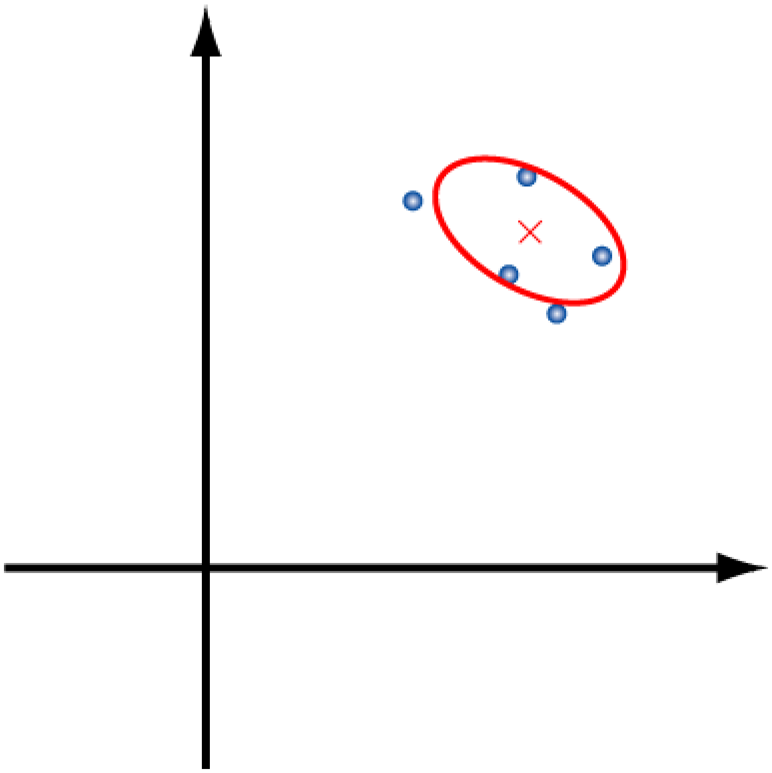
- Sample
- Select elites
- Update mean
- Update covariance
- iterate

Covariance Matrix Adaptation



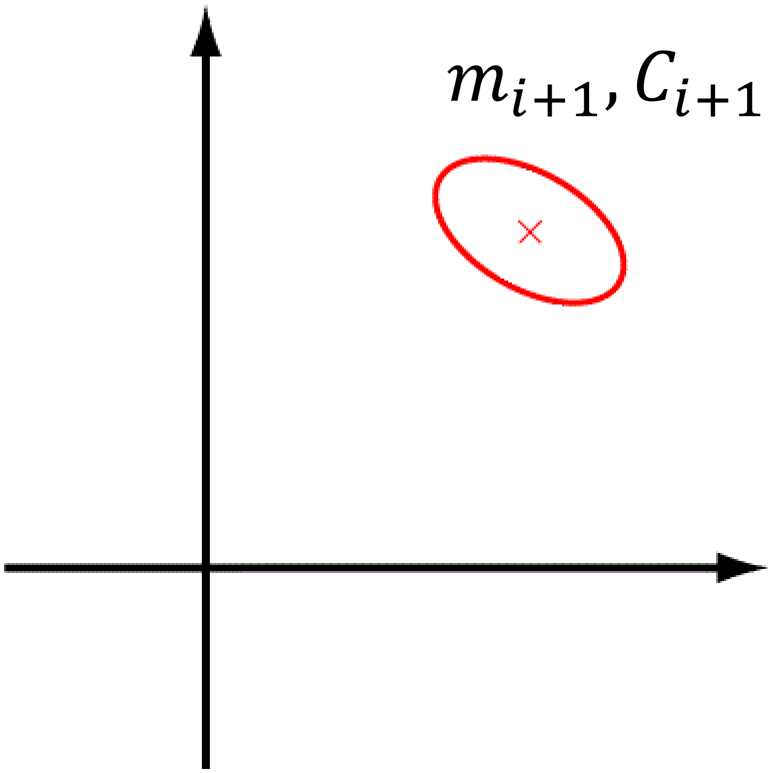
- Sample
- Select elites
- Update mean
- Update covariance
- iterate

Covariance Matrix Adaptation



- Sample
- Select elites
- Update mean
- Update covariance
- iterate

Covariance Matrix Adaptation



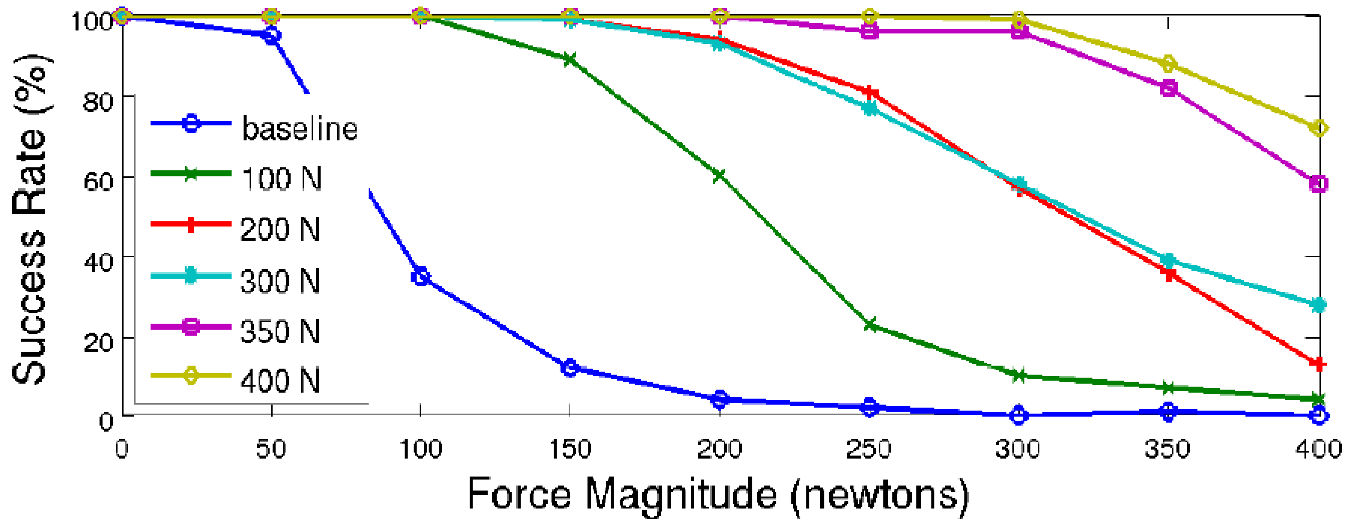
- Sample
- Select elites
- Update mean
- Update covariance
- iterate

Optimizing Walking Controllers for Uncertain Inputs and Environments

Jack M. Wang
David J. Fleet
Aaron Hertzmann

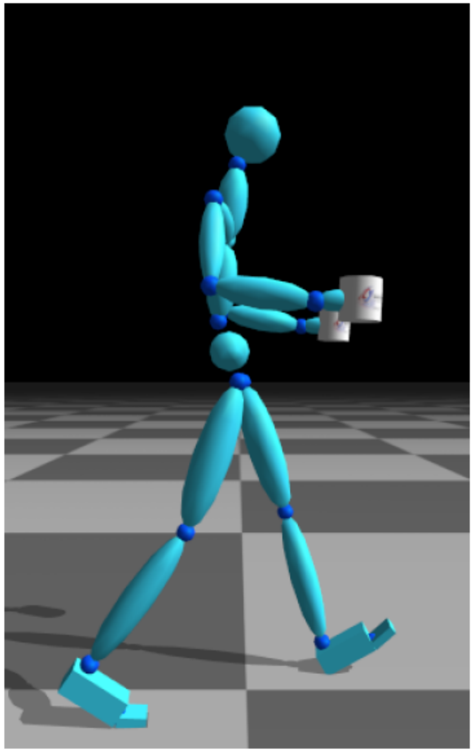
University of Toronto

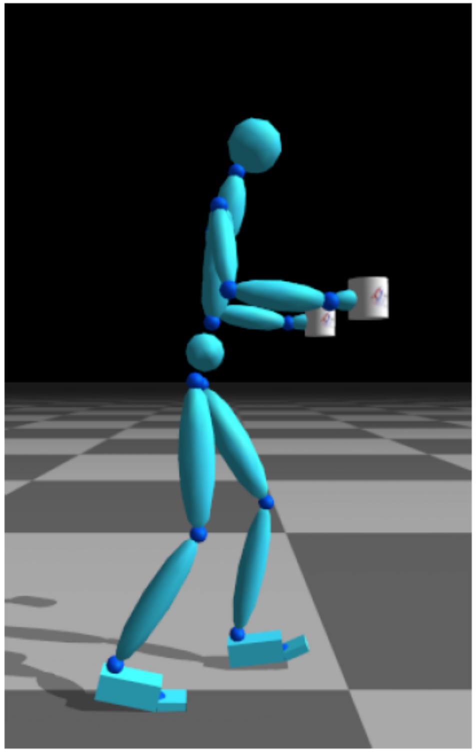
Result

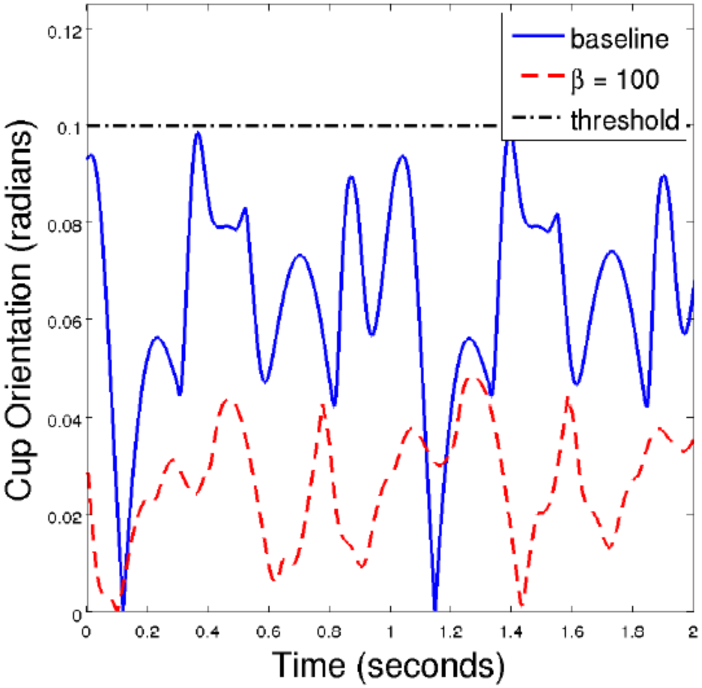


direction	0 N (baseline)	100 N	200 N	300 N	350 N	400 N
$(1, 0)$	125	175	275	375	350	425
$(1, \pm 1)$	75	125	225	275	300	325
$(0, \pm 1)$	75	125	325	375	425	475
$(-1, \pm 1)$	25	100	175	225	250	275
$(-1, 0)$	75	200	350	325	375	375

Result






baseline
β = 100
threshold
Cup Orientation (radians)
Time (seconds)
0
0.02
0.04
0.06
0.08
0.1
0.12
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2