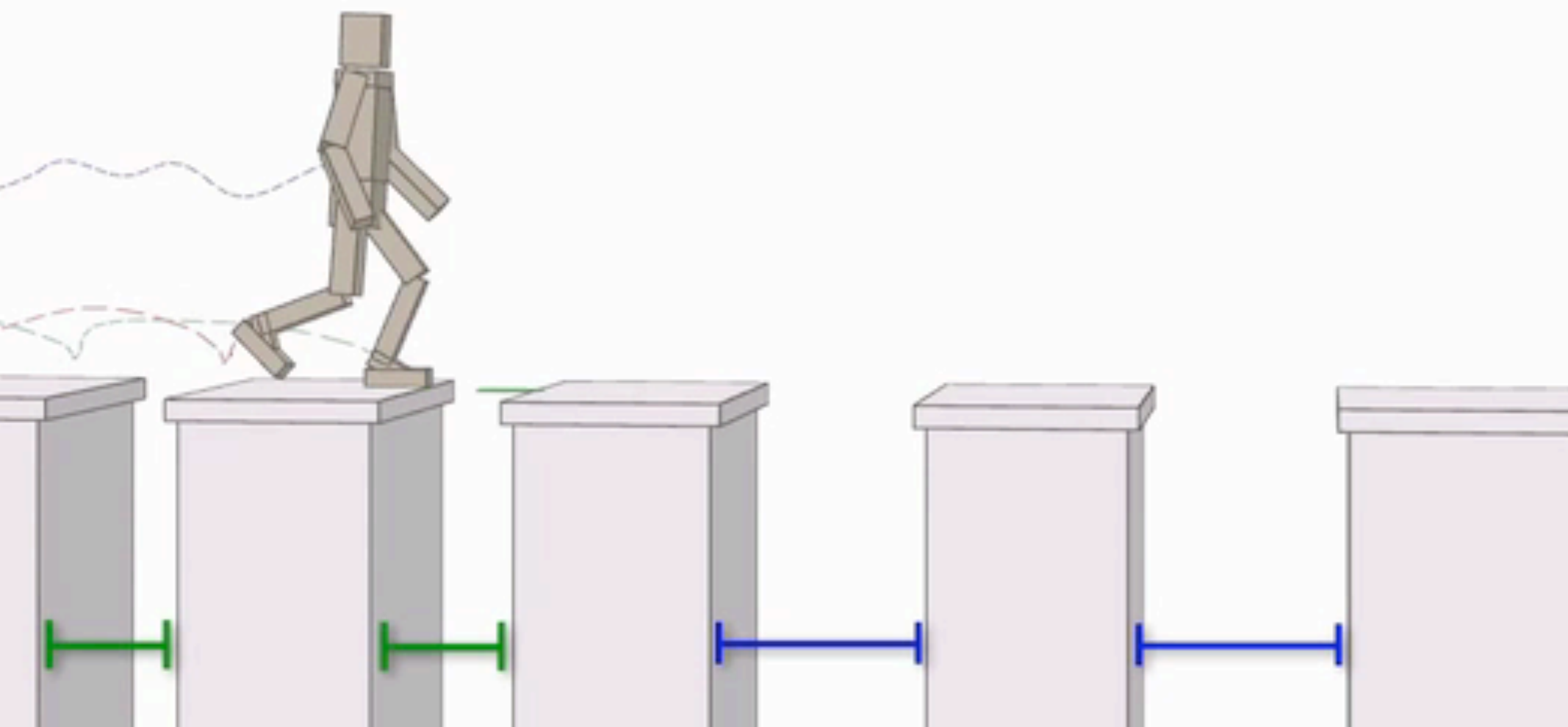


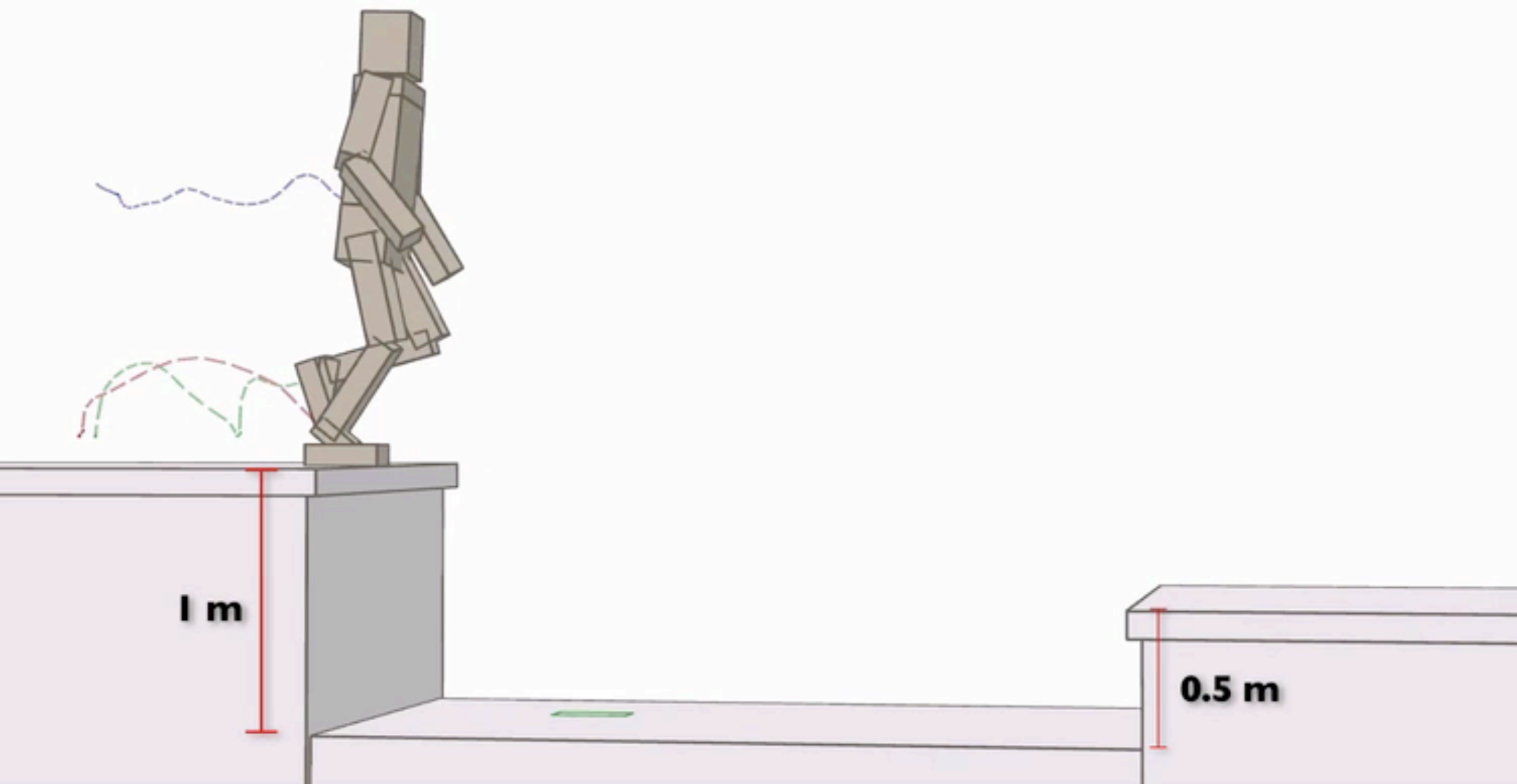
Robust Physics-Based Locomotion Using Low-Dimensional Planning

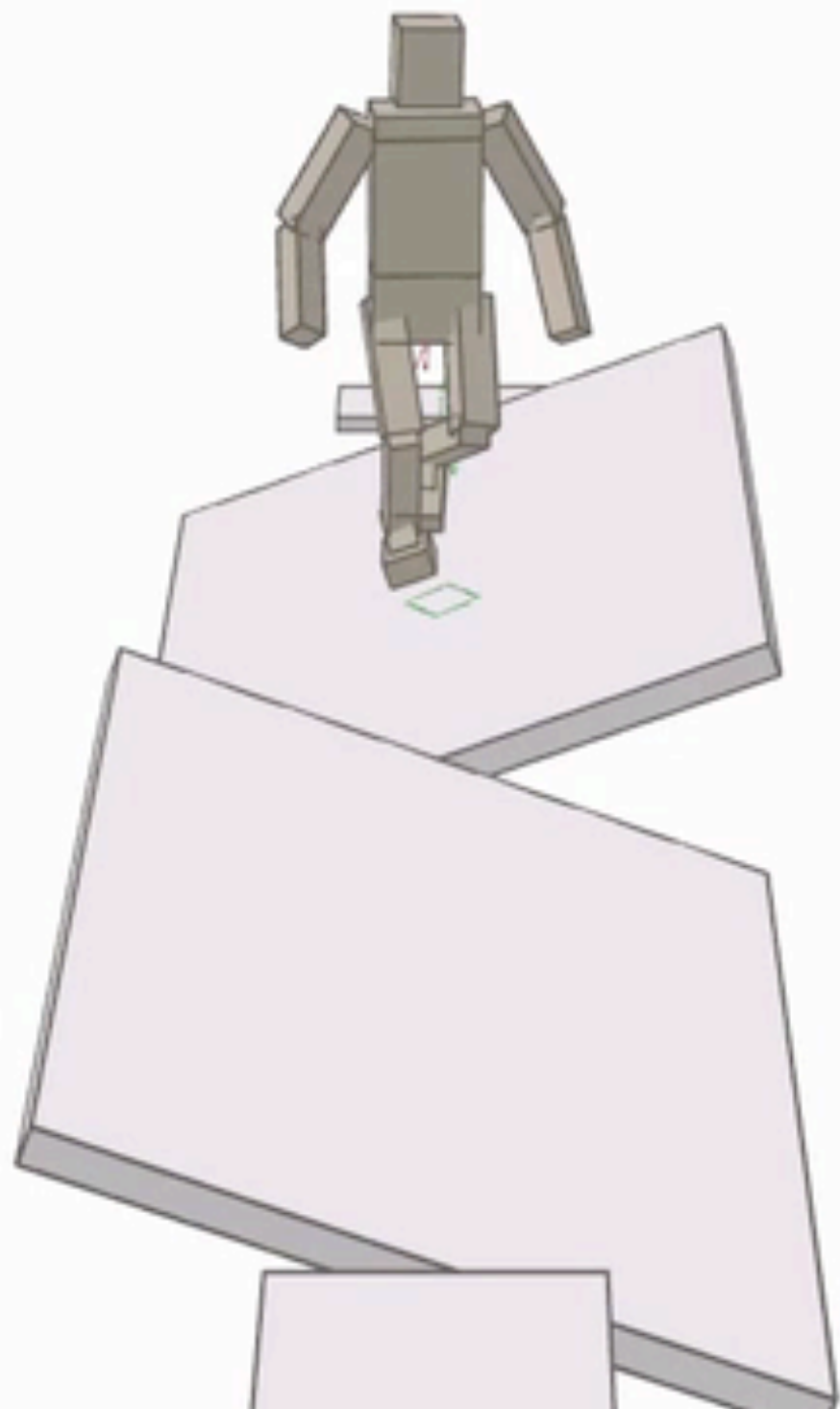
Igor Mordatch
Martin de Lasa
Aaron Hertzmann

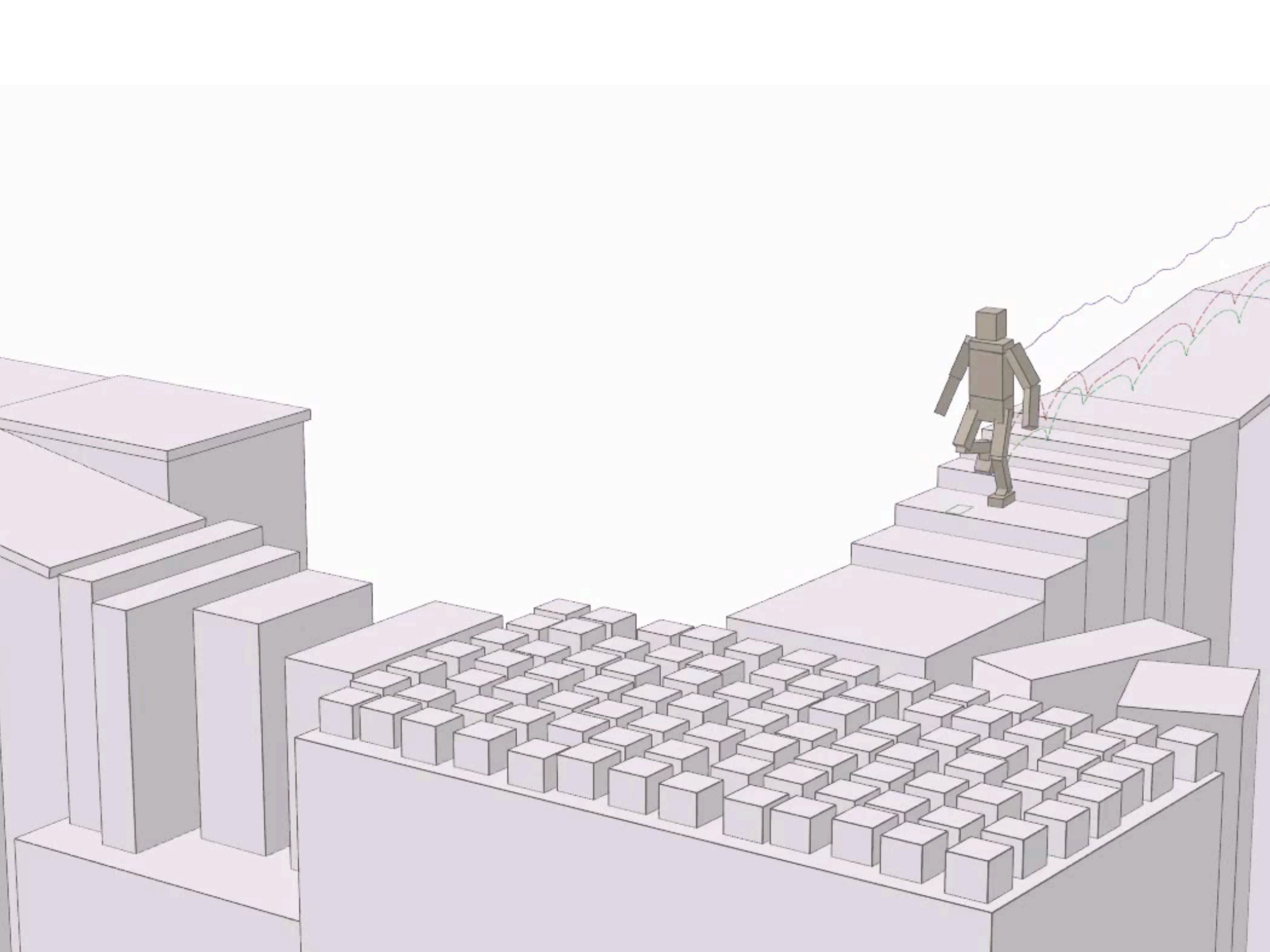
University of Toronto



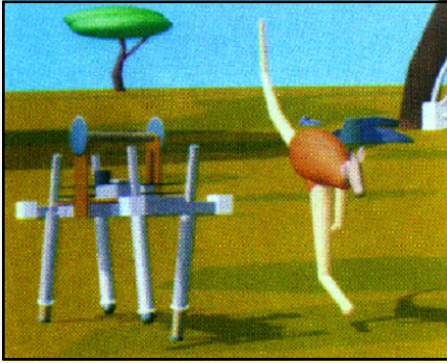




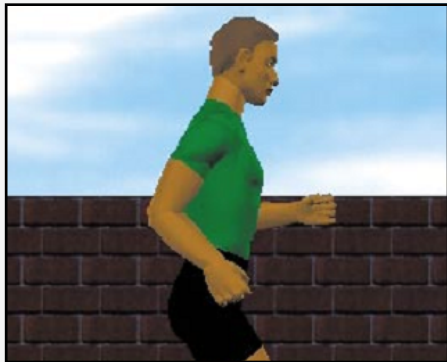




Related Work: PD Control

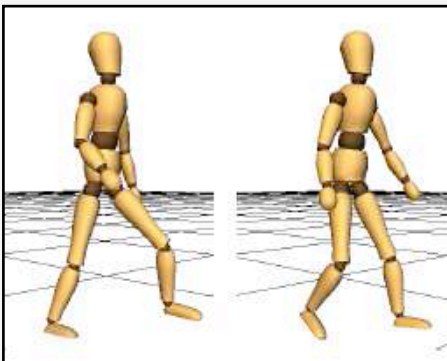


Raibert, Hodgins 1991



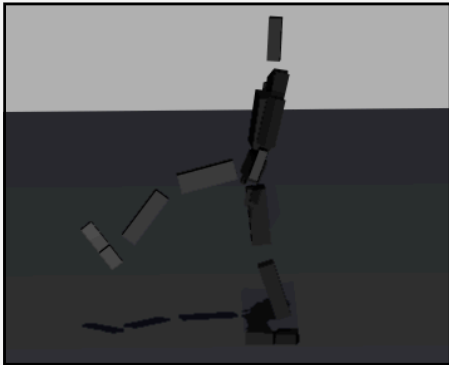
Hodgins et al 1995

Faloutsos et al 2001



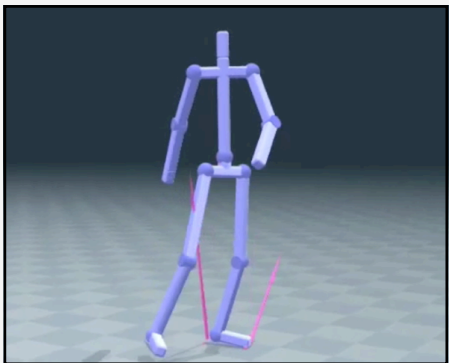
Yin, Loken, Van de Panne 2007

Related Work: Mocap Tracking



Zordan, Hodgins 2002

da Silva, Abe, Popovic 2008



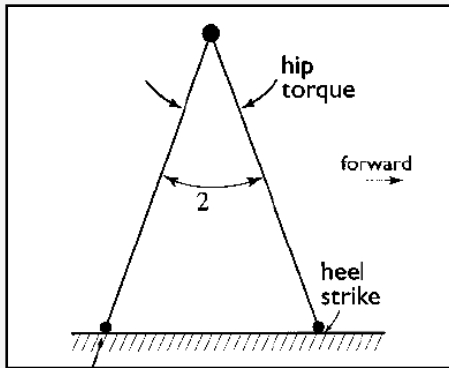
Macchieto 2009

Muico et al 2009

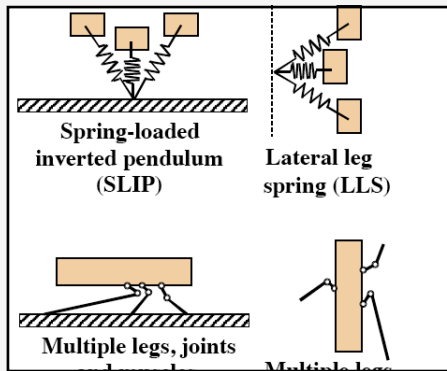


Tsai et al 2010

Related Work: Simplified Models

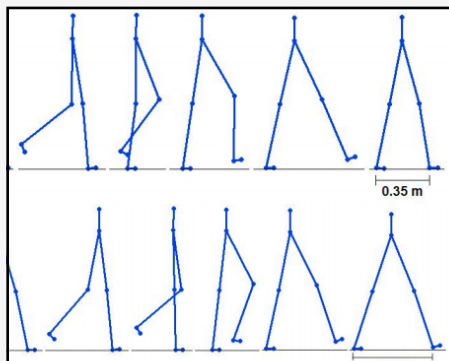


McGeer 1990



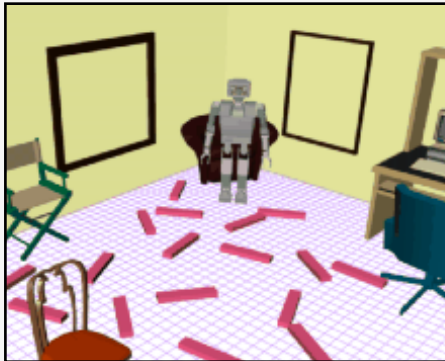
Full, Koditscheck 1999

Popovic et al 2004



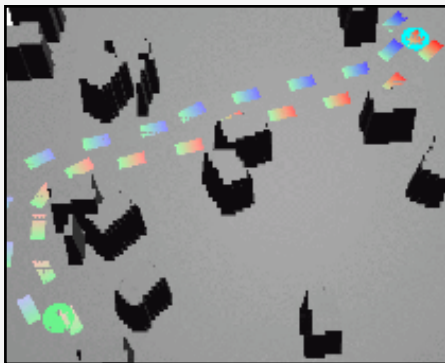
Ramamoorthy, Kuipers 2008

Related Work: Planning

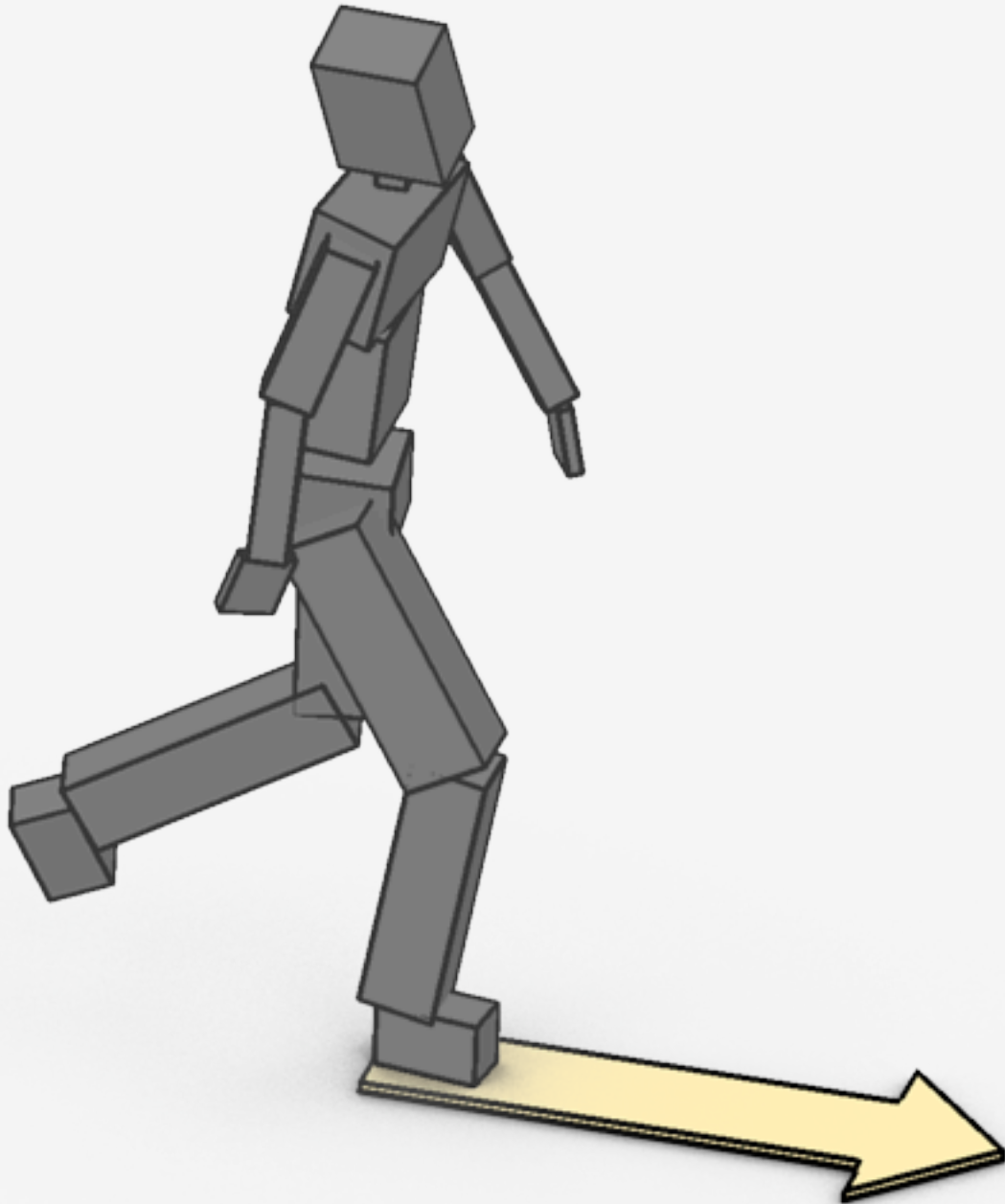


Kuffner et al 2003

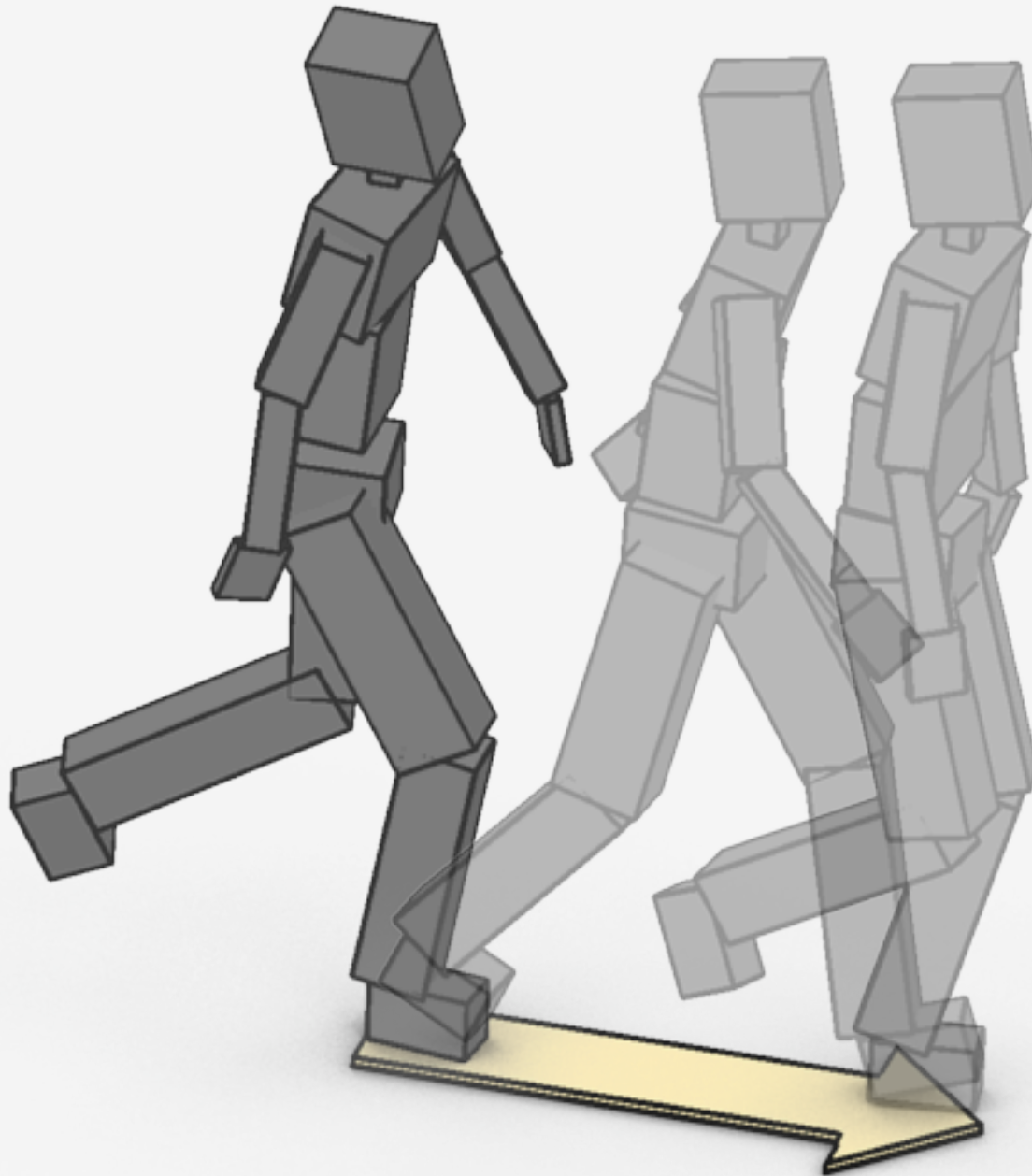
Chestnutt 2007



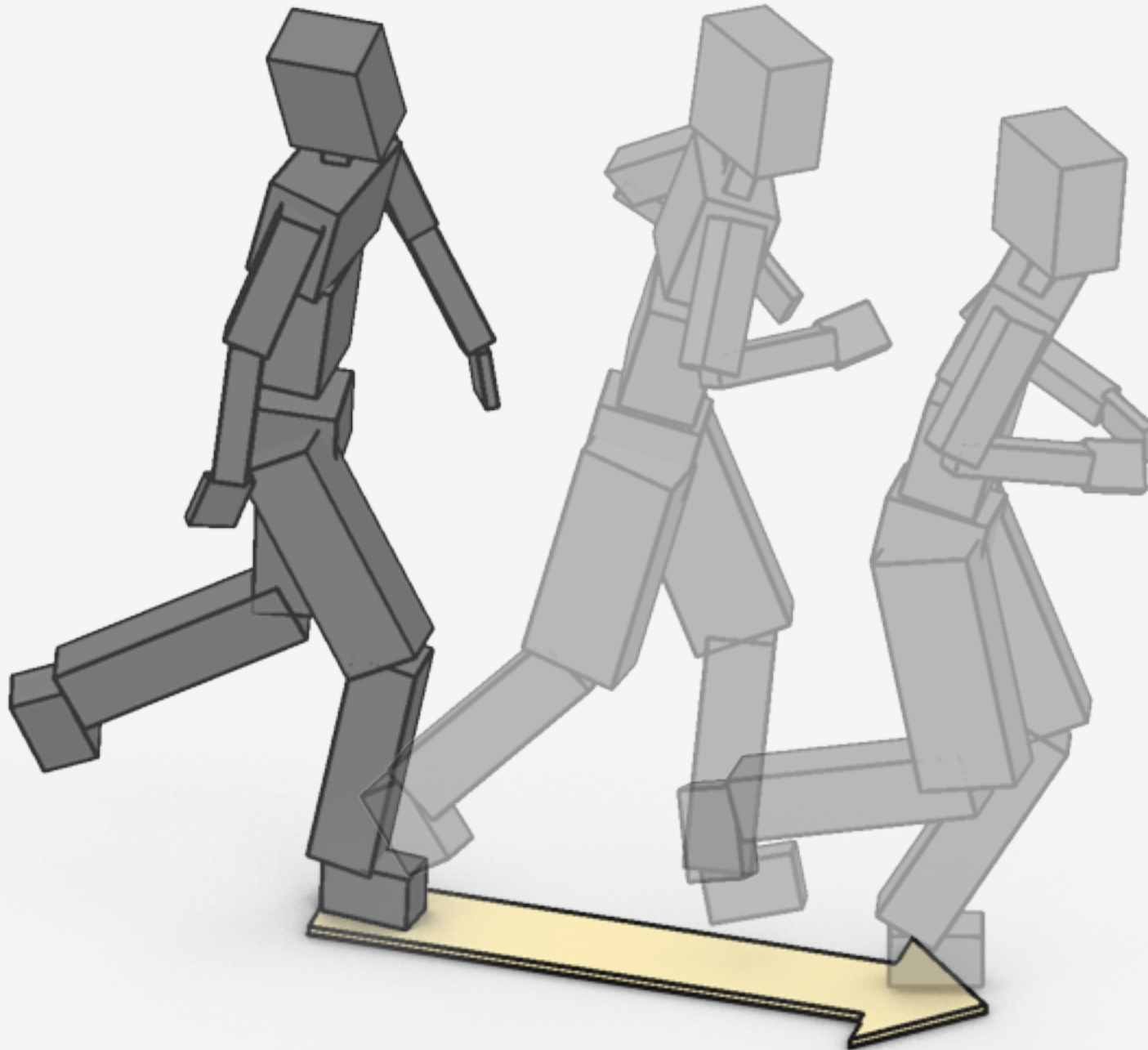
At any timestep, must determine character joint torques



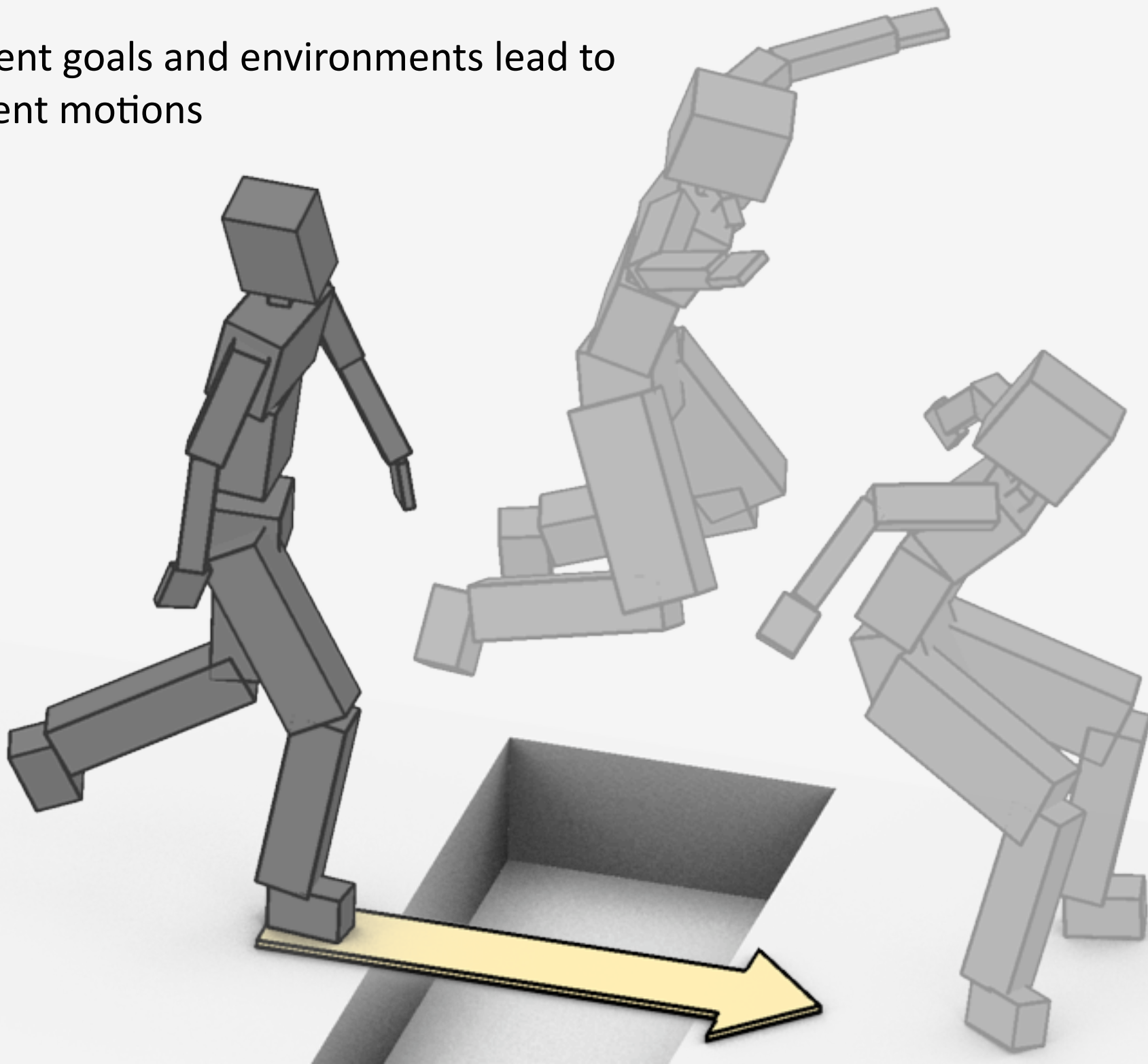
Different goals and environments lead to different motions



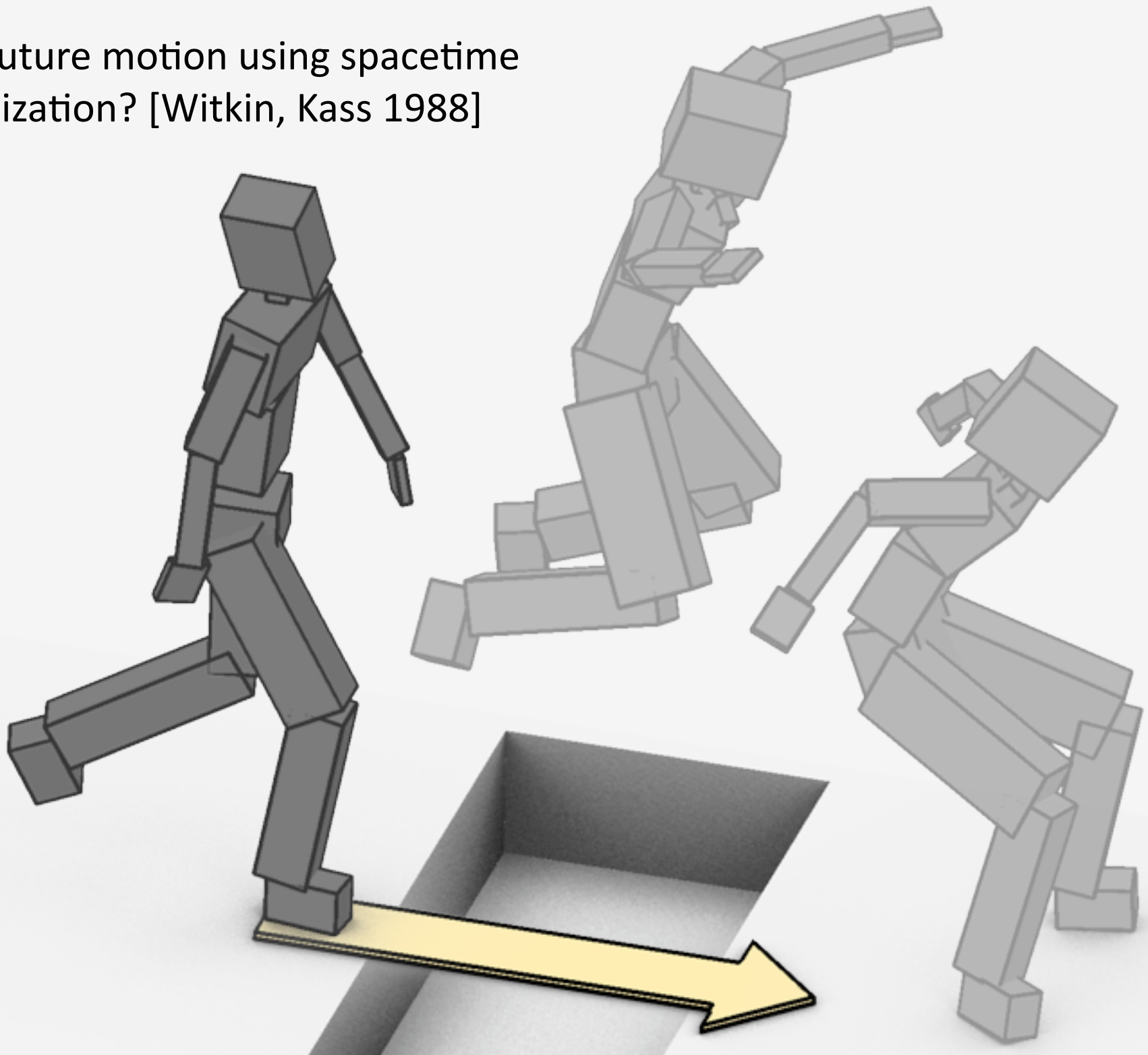
Different goals and environments lead to different motions



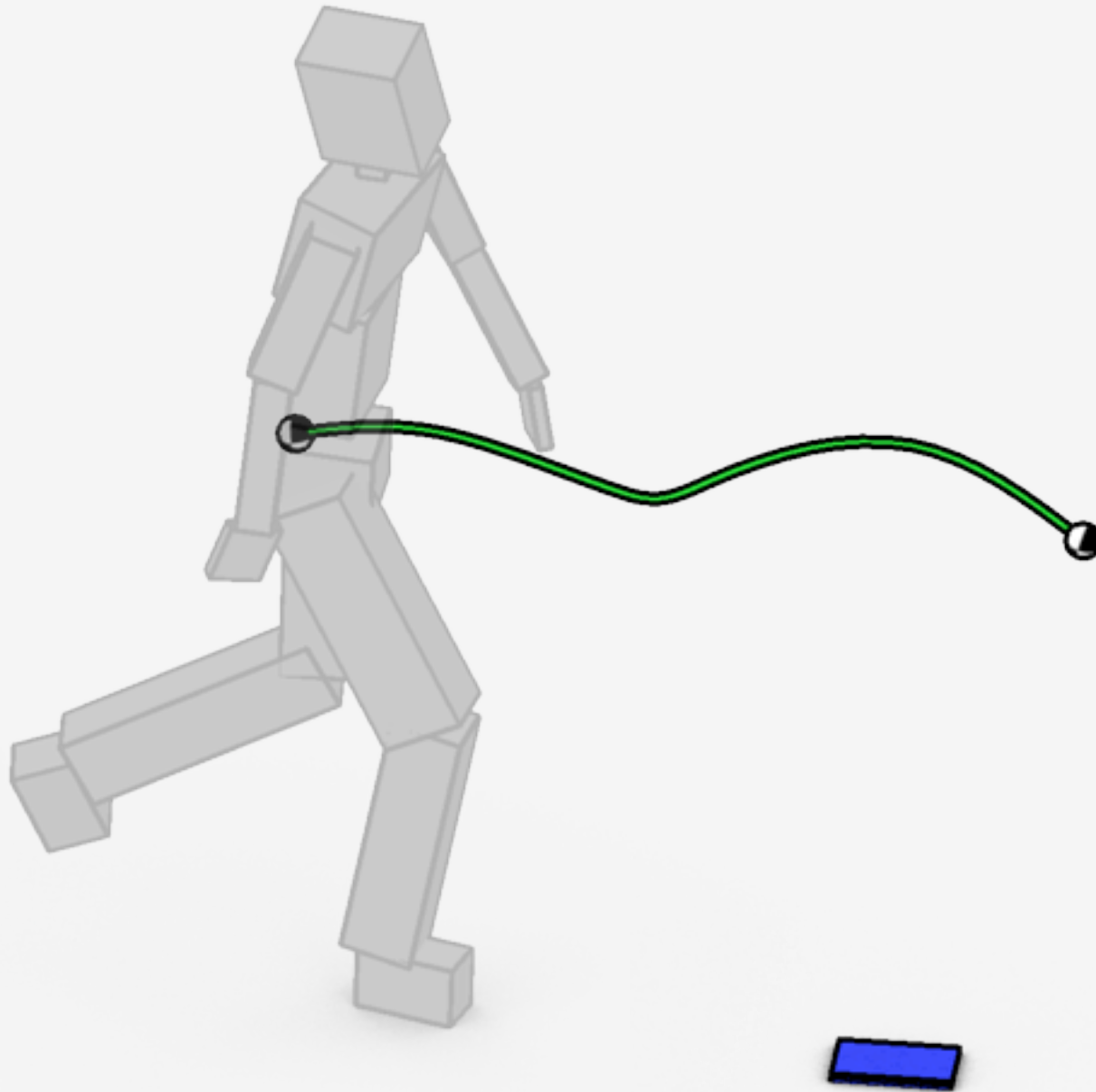
Different goals and environments lead to different motions



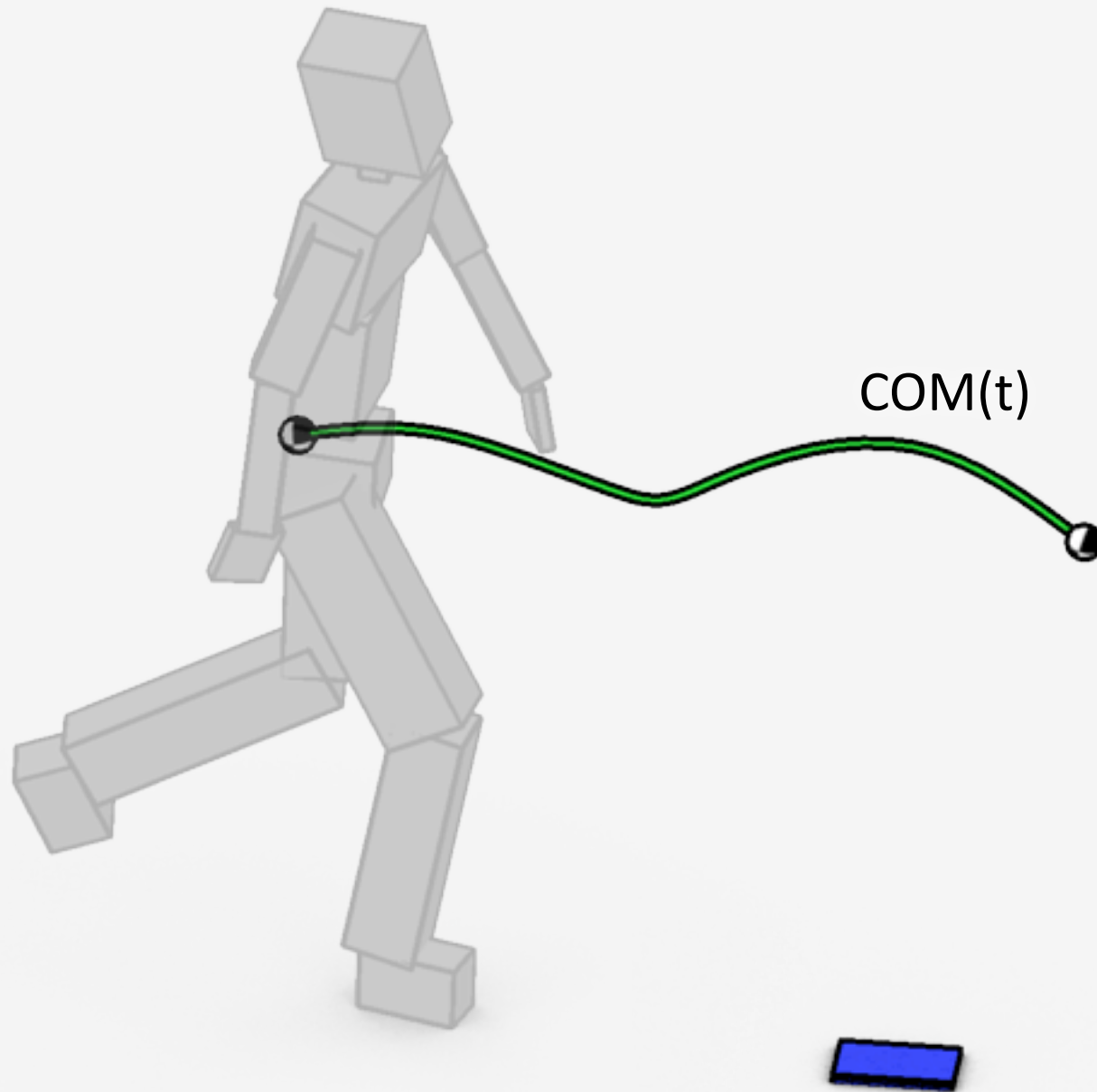
Plan future motion using spacetime optimization? [Witkin, Kass 1988]



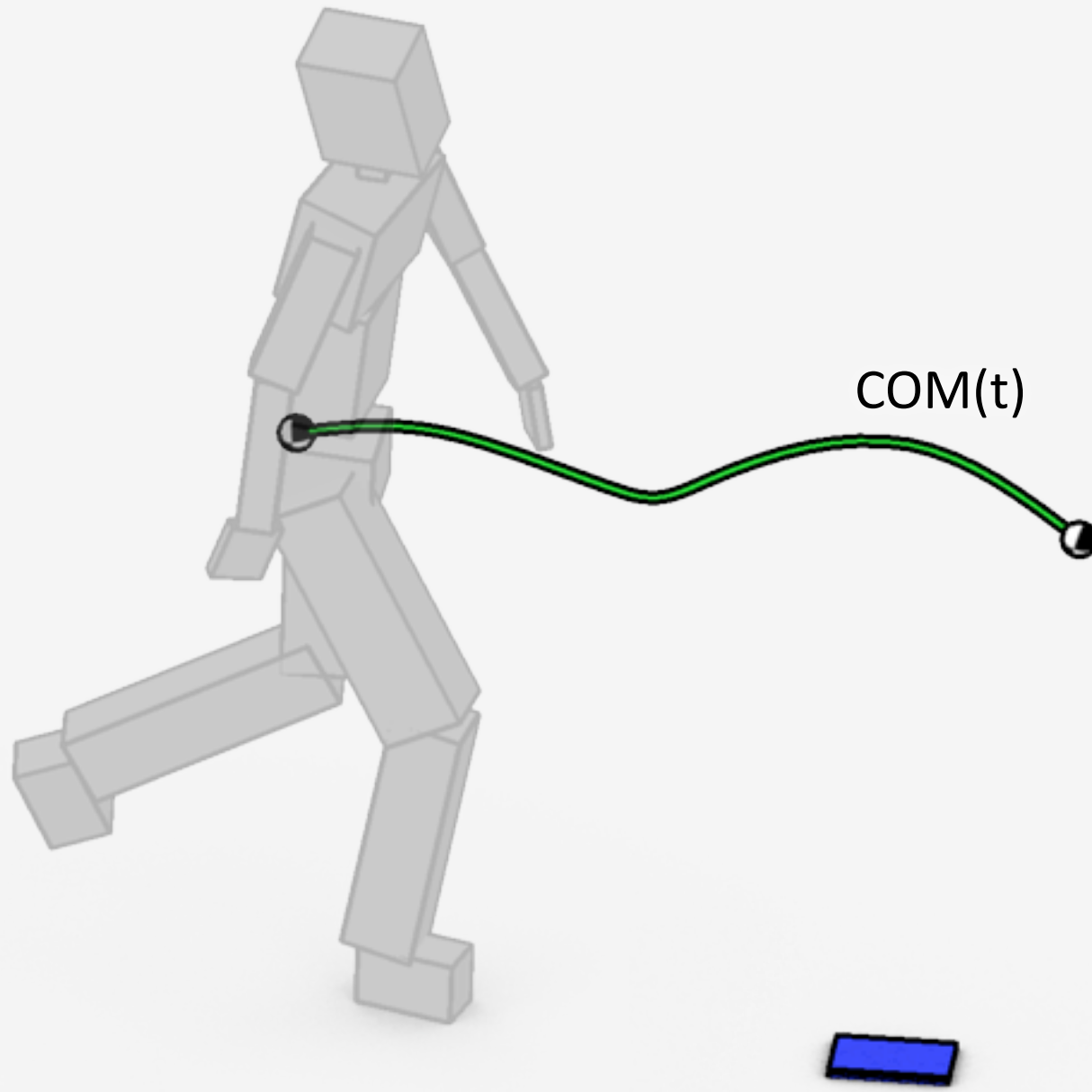
Optimize with a low-dimensional motion model



Optimize with a low-dimensional motion model



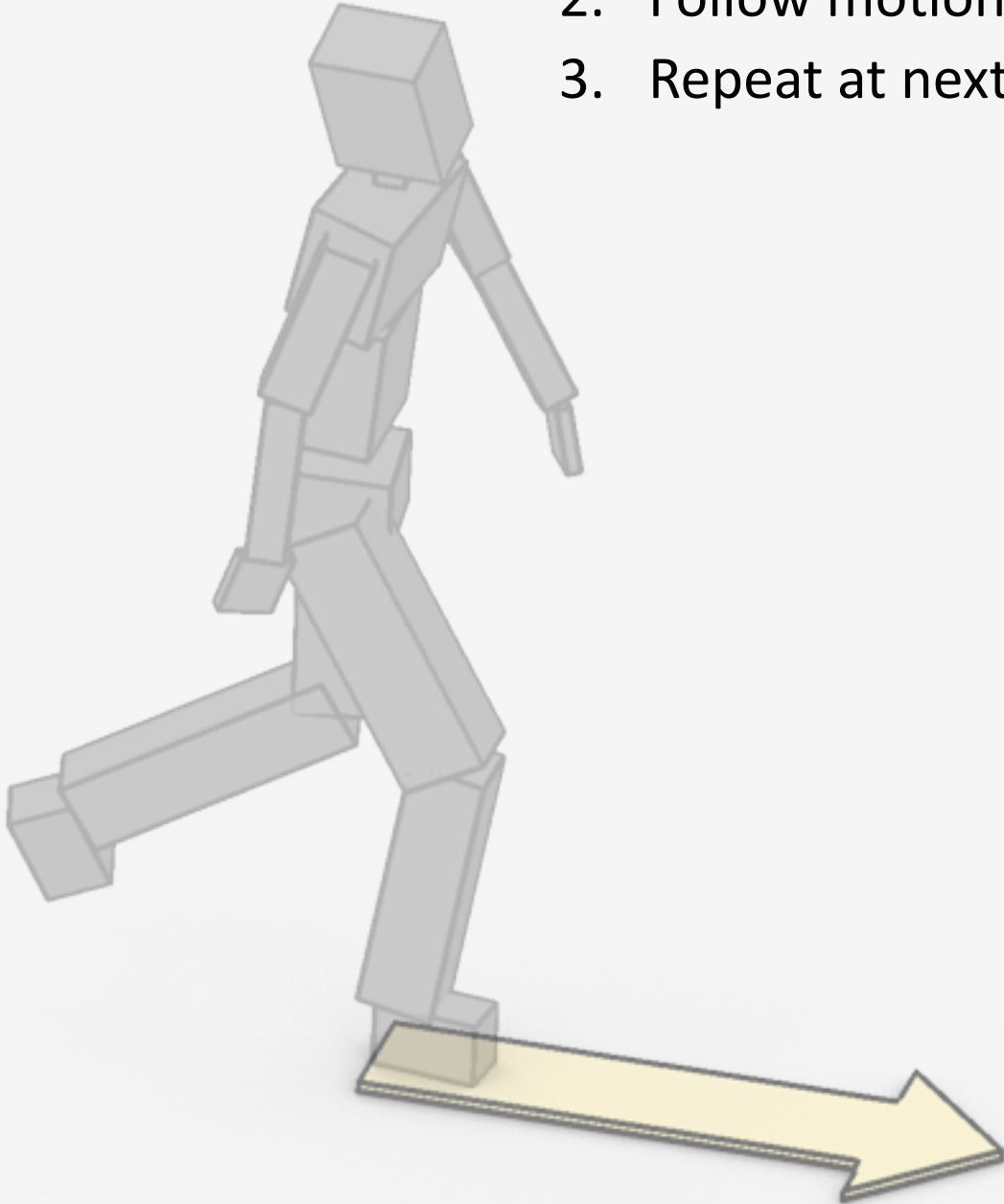
Optimize with a low-dimensional motion model



Next foot contact

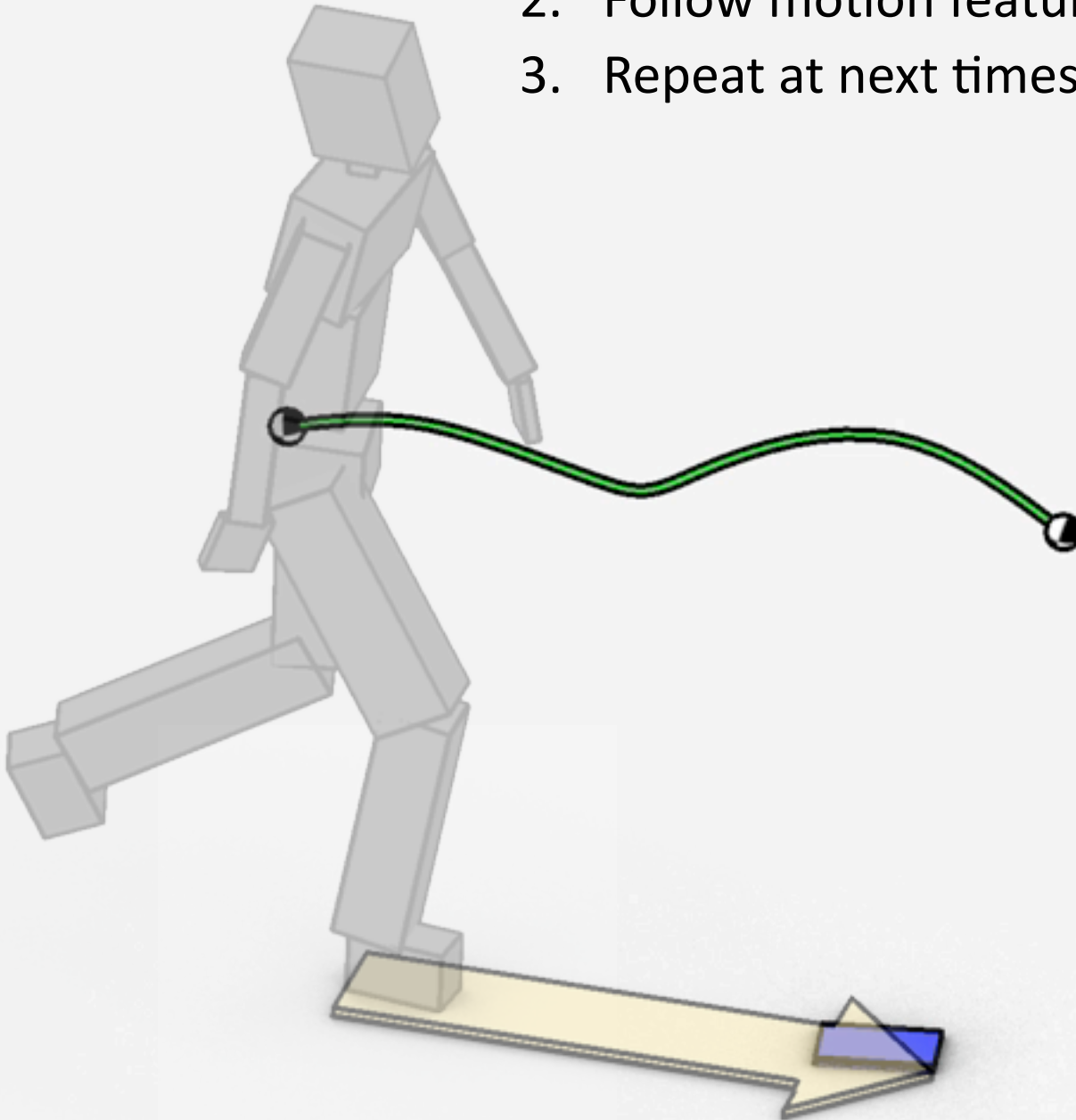
Algorithm

1. Optimize motion trajectory
2. Follow motion features with full character
3. Repeat at next timestep



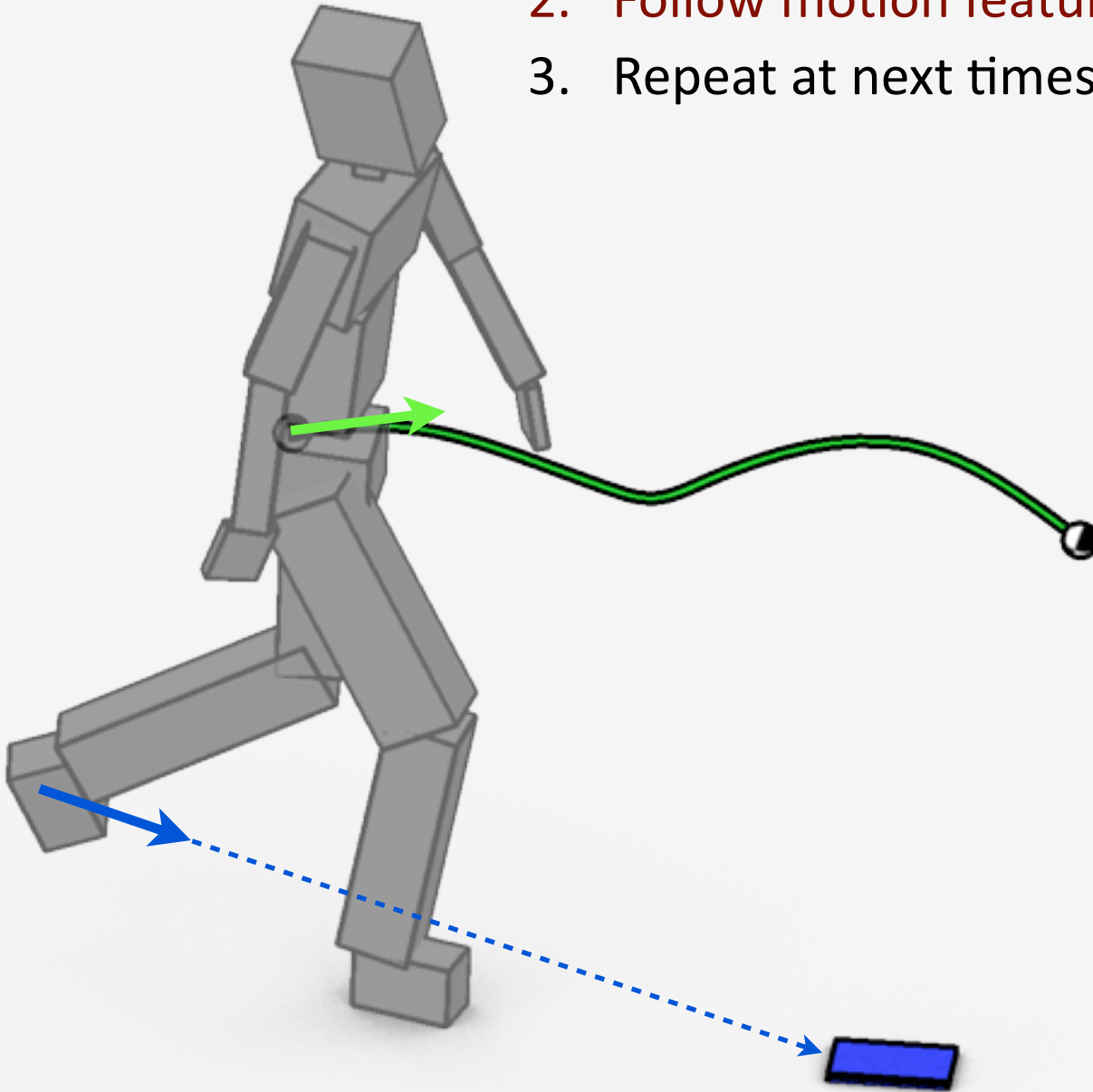
Algorithm

1. Optimize motion trajectory
2. Follow motion features with full character
3. Repeat at next timestep



Algorithm

1. Optimize motion trajectory
2. Follow motion features with full character
3. Repeat at next timestep

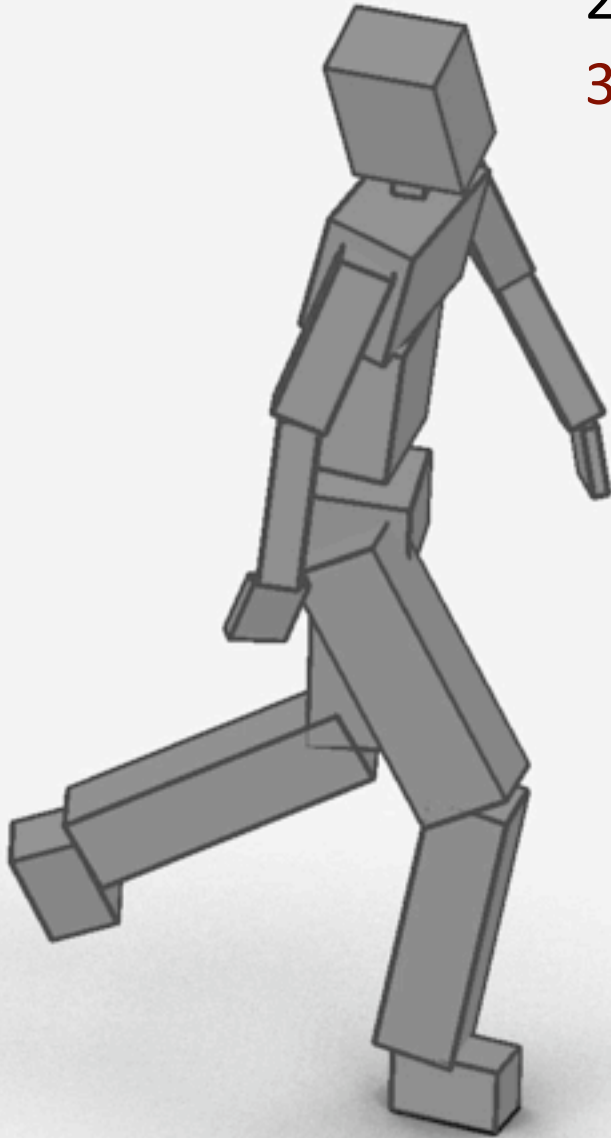


Algorithm

1. Optimize motion trajectory
2. Follow motion features with full character
3. Repeat at next timestep

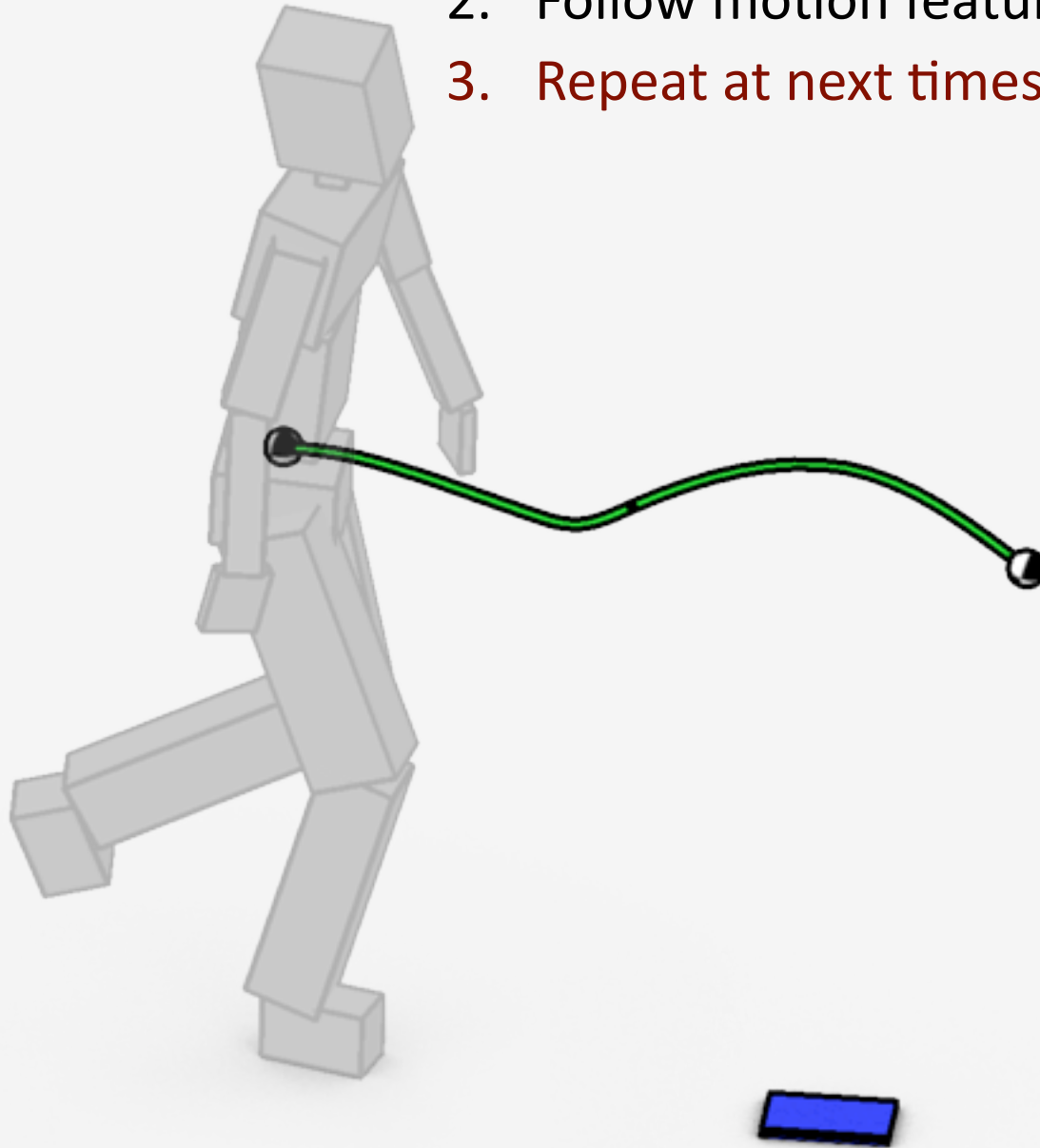
Algorithm

1. Optimize motion trajectory
2. Follow motion features with full character
3. Repeat at next timestep



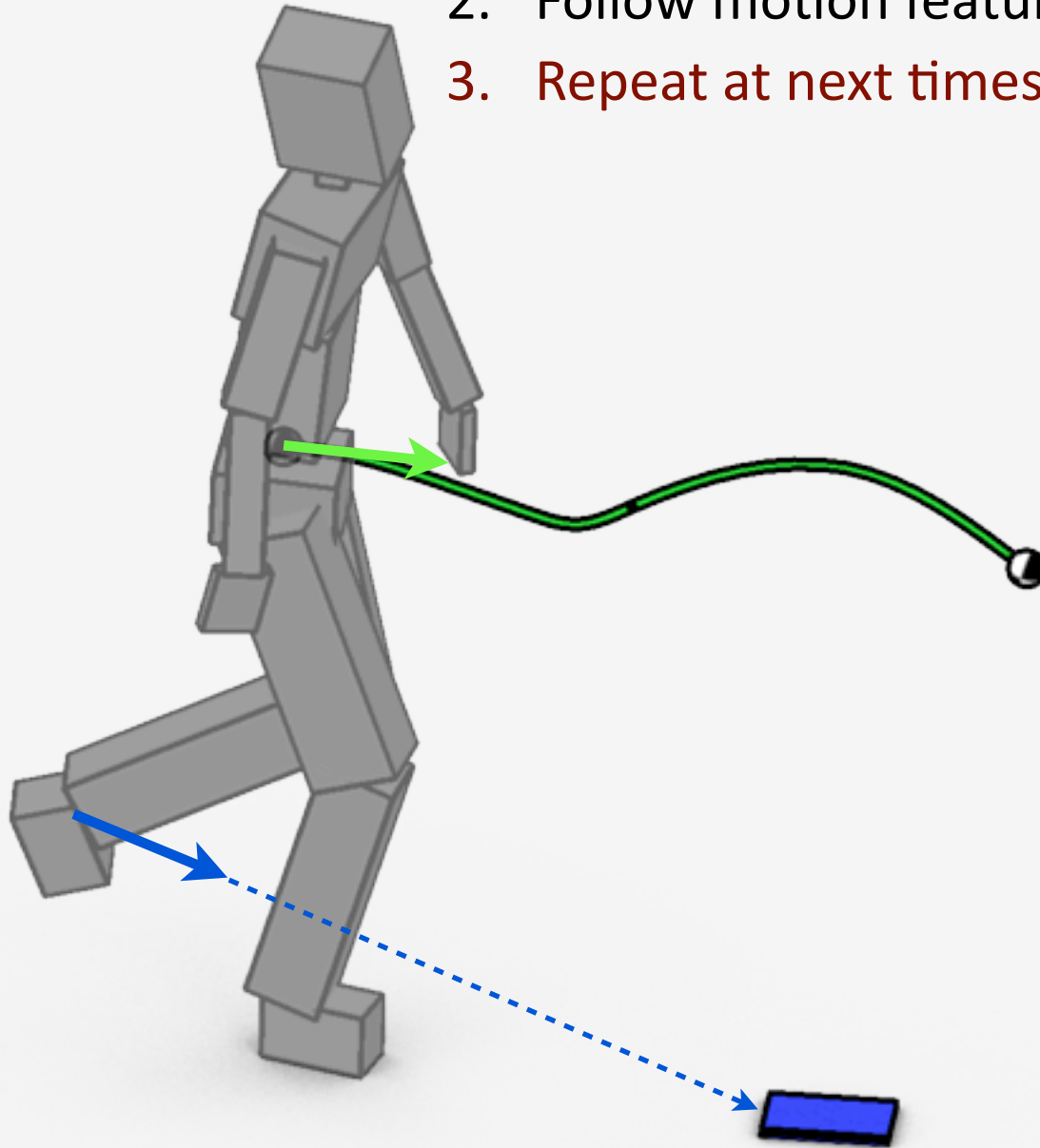
Algorithm

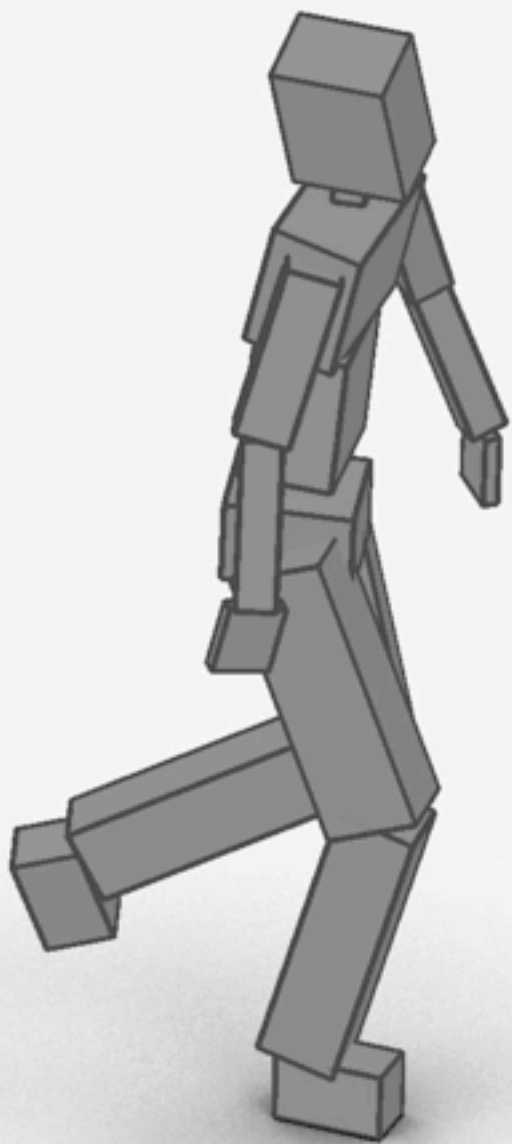
1. Optimize motion trajectory
2. Follow motion features with full character
3. Repeat at next timestep



Algorithm

1. Optimize motion trajectory
2. Follow motion features with full character
3. Repeat at next timestep





What is the low-dimensional motion model?

How to optimize the model?

How to map motion to full-dimensional character?

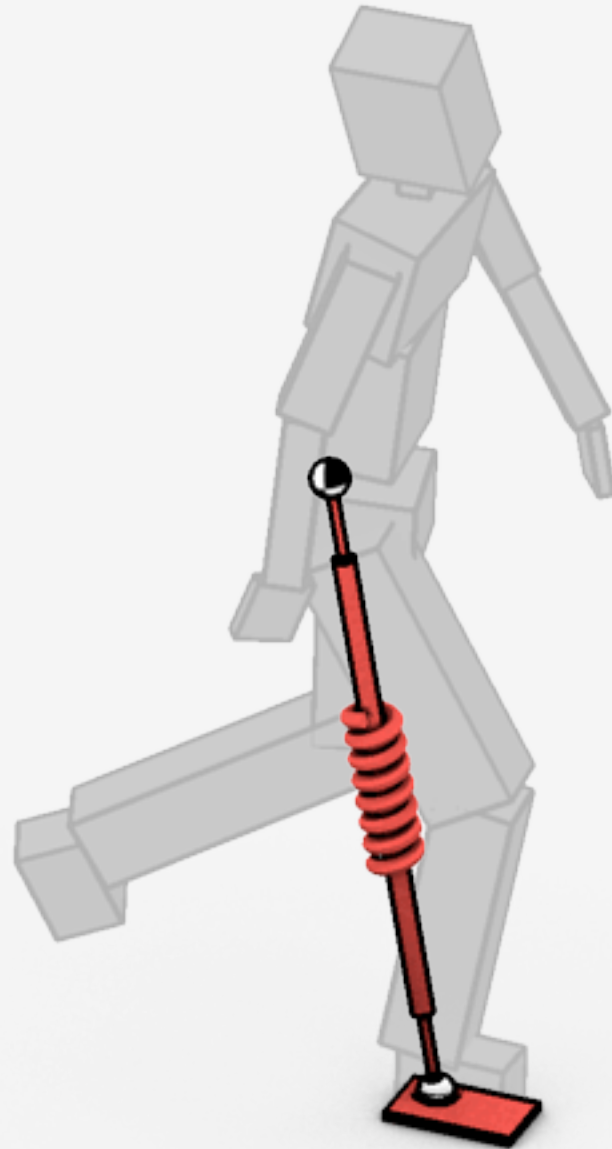
What is the low-dimensional motion model?

How to optimize the model?

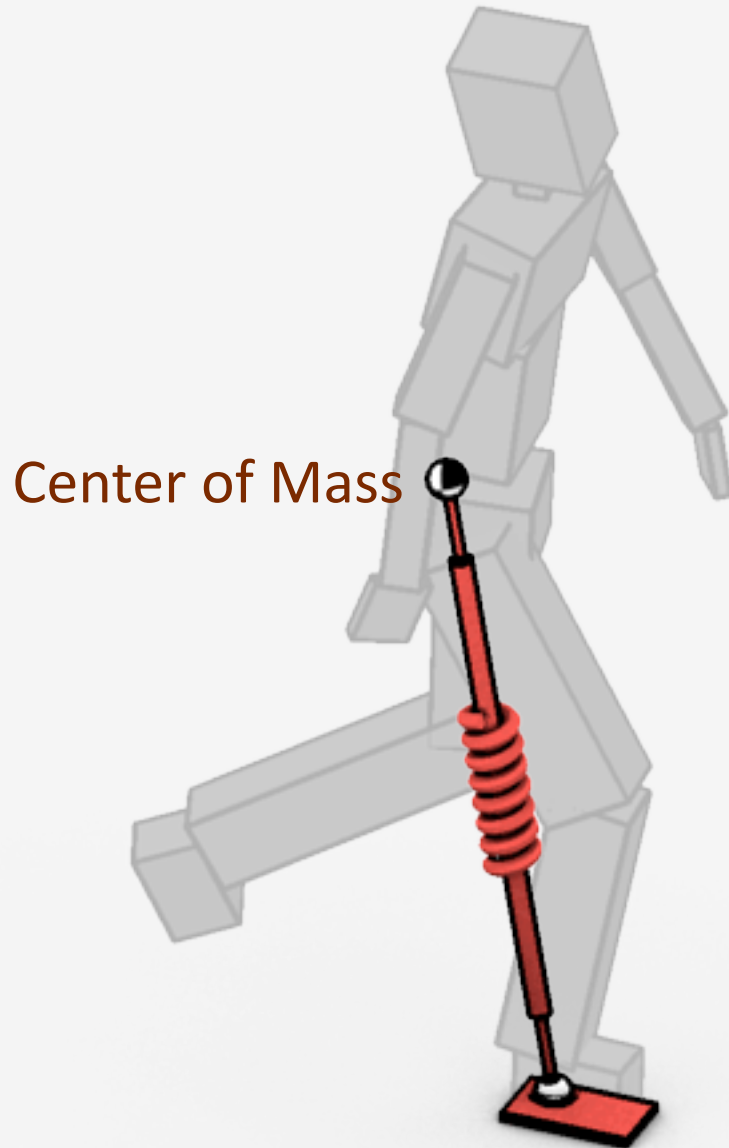
How to map motion to full-dimensional character?

Spring-Loaded Inverted Pendulum

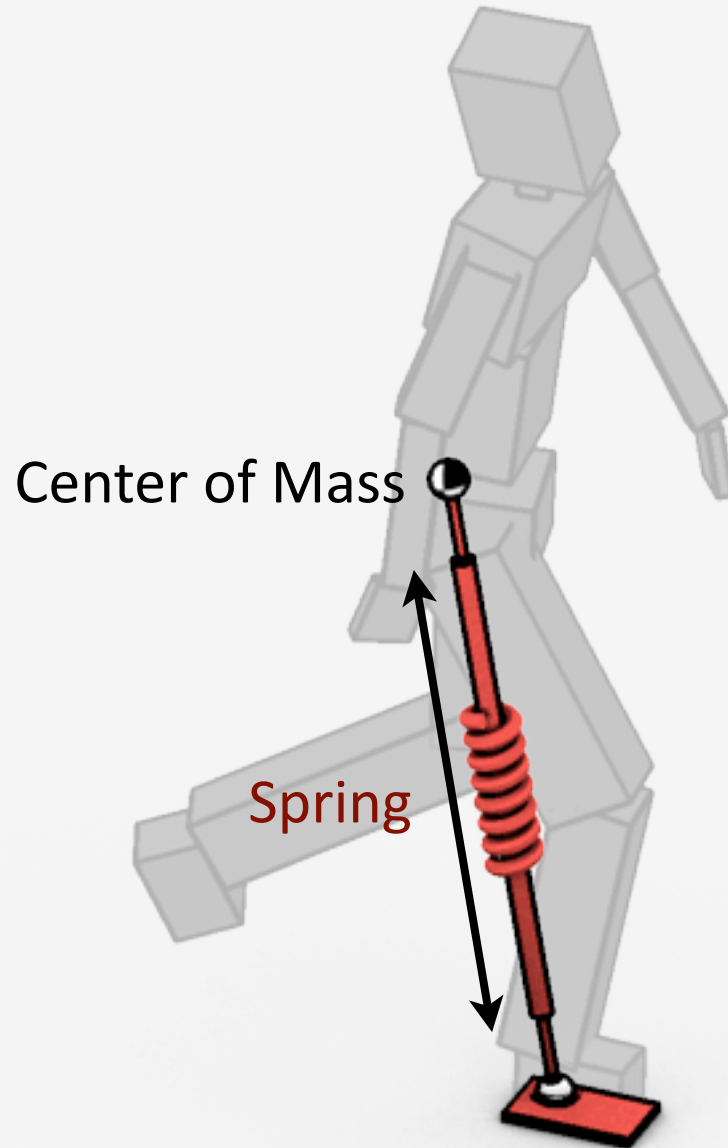
(SLIP)



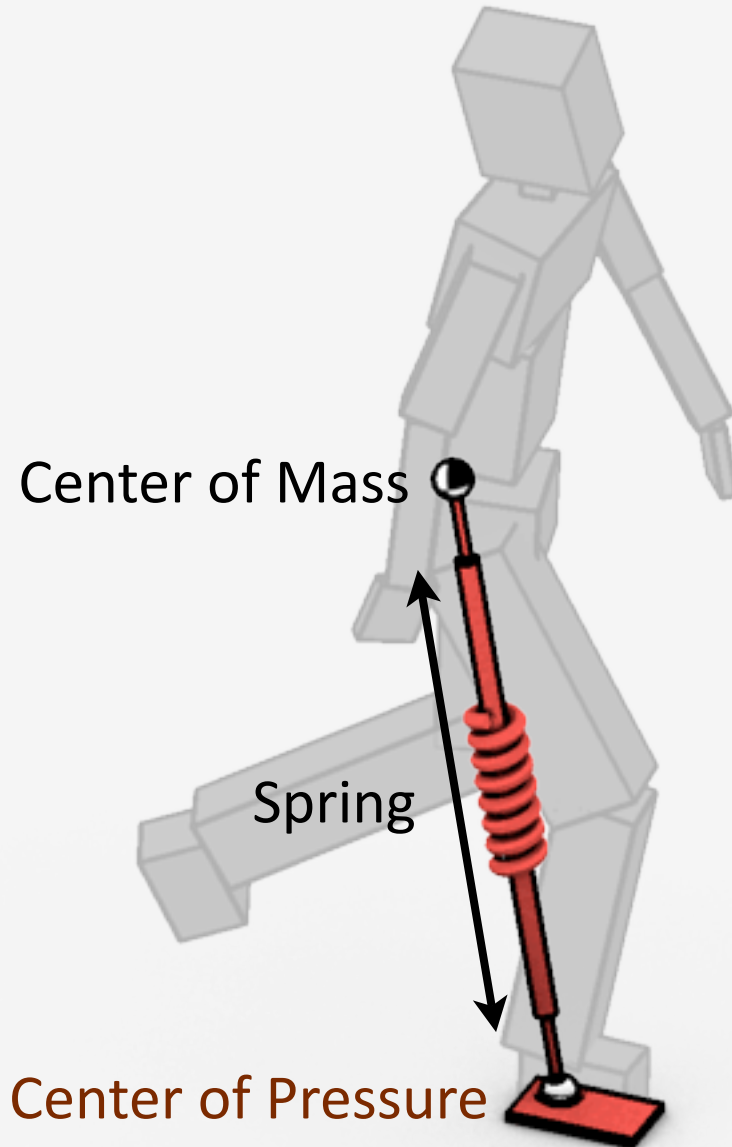
Spring-Loaded Inverted Pendulum



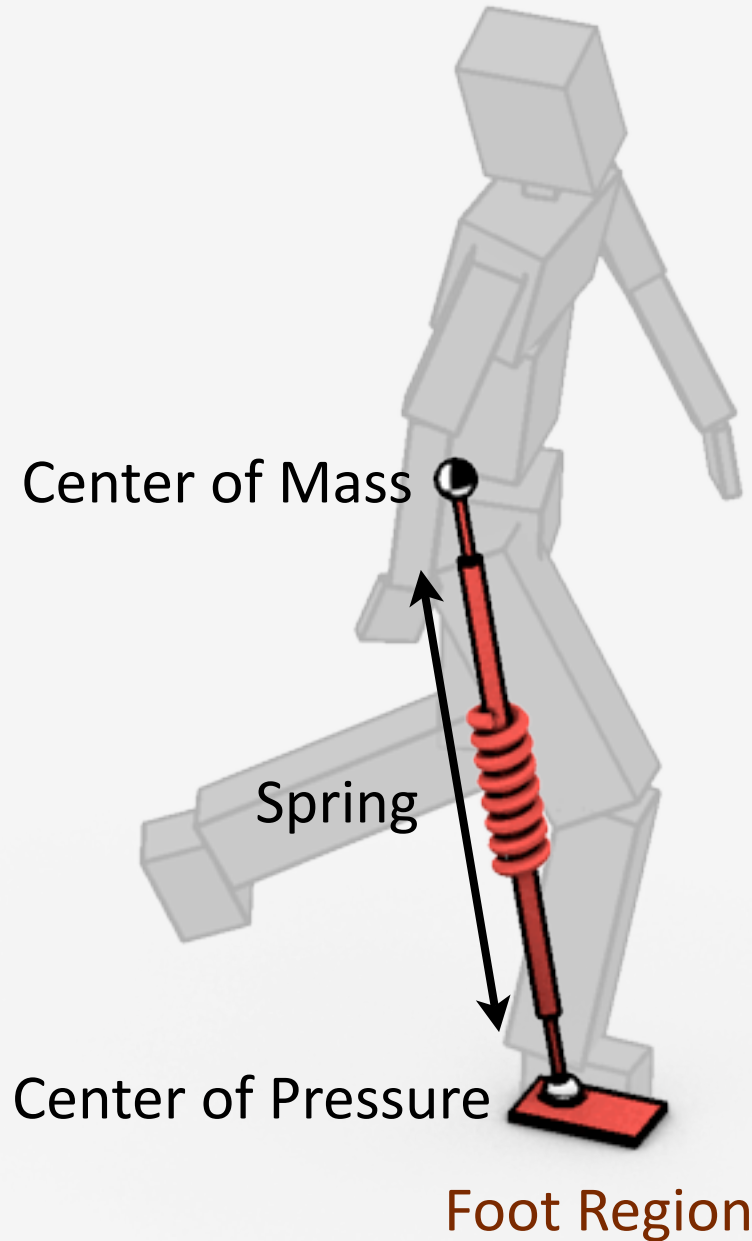
Spring-Loaded Inverted Pendulum



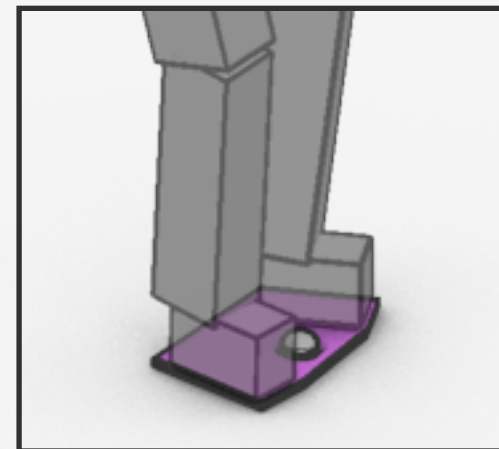
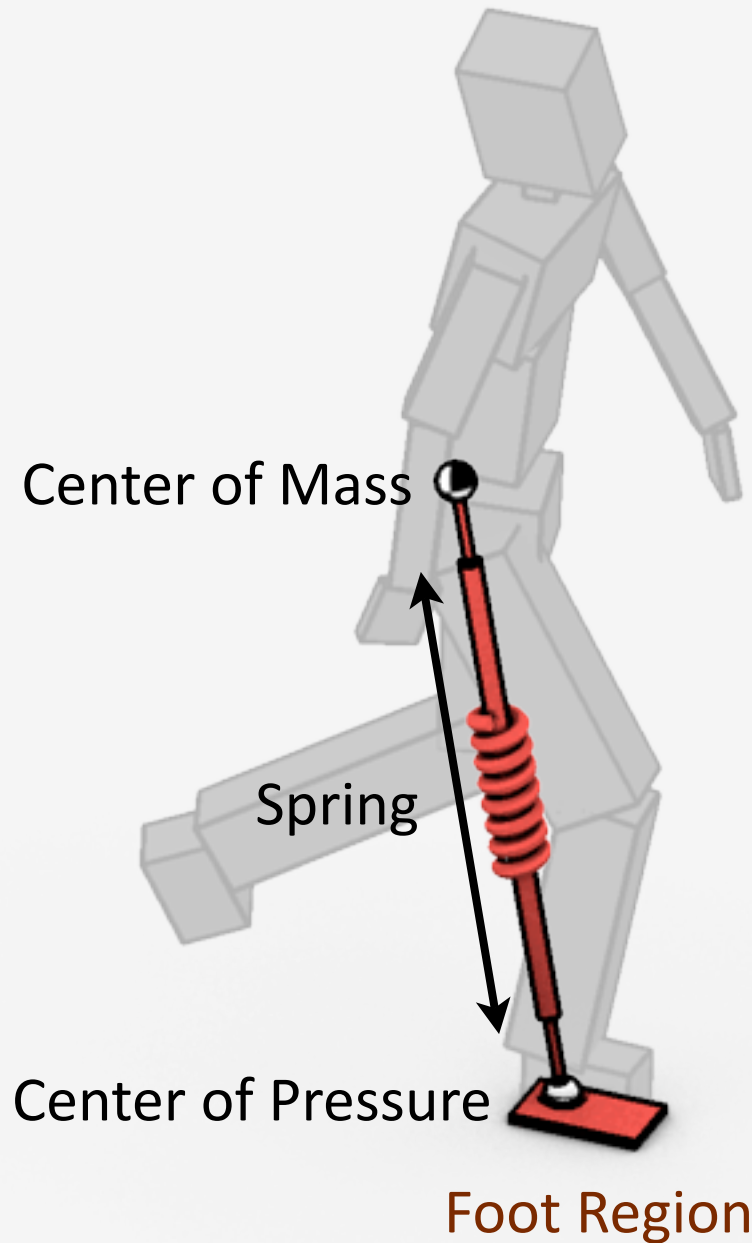
Spring-Loaded Inverted Pendulum



Spring-Loaded Inverted Pendulum



Spring-Loaded Inverted Pendulum



Spring-Loaded Inverted Pendulum

Center of Pressure

Spring-Loaded Inverted Pendulum

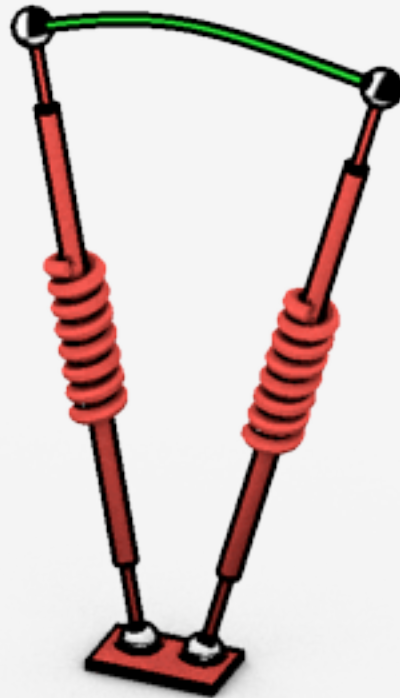


Spring-Loaded Inverted Pendulum

Spring Rest Length

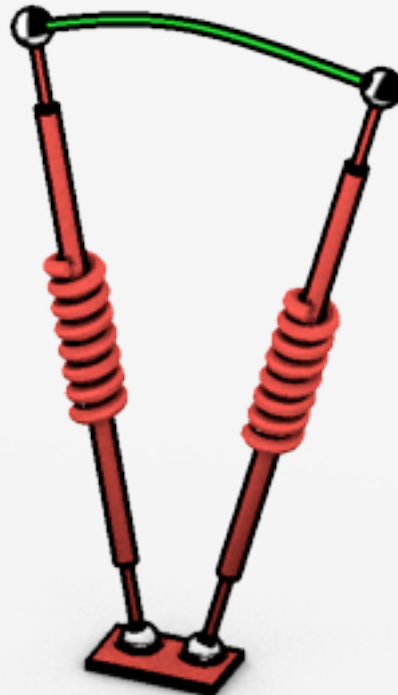


$$(s_0, u) \longrightarrow s(t)$$



$$\left(\underset{\substack{\nearrow \\ \text{Initial state}}}{s_0}, u \right) \longrightarrow s(t)$$

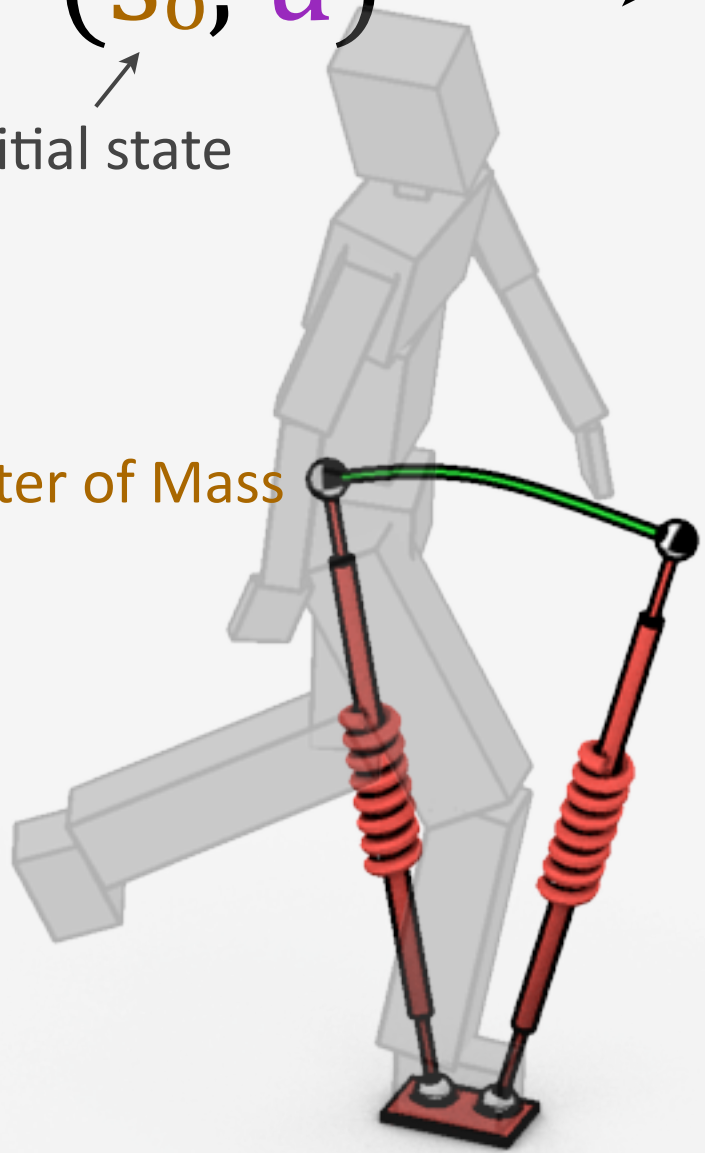
Center of Mass



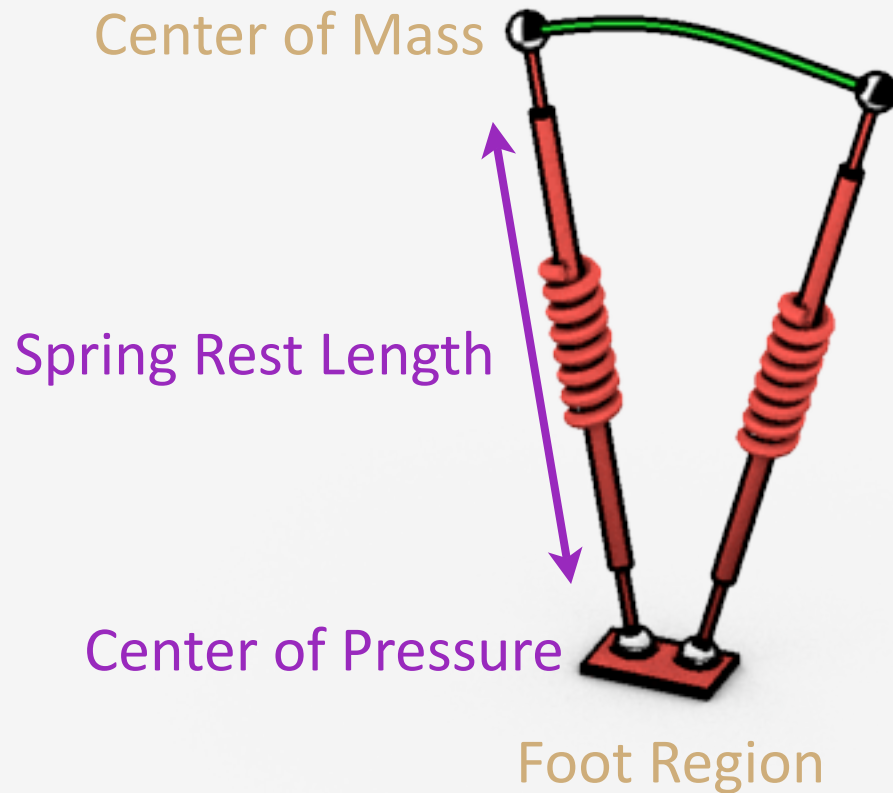
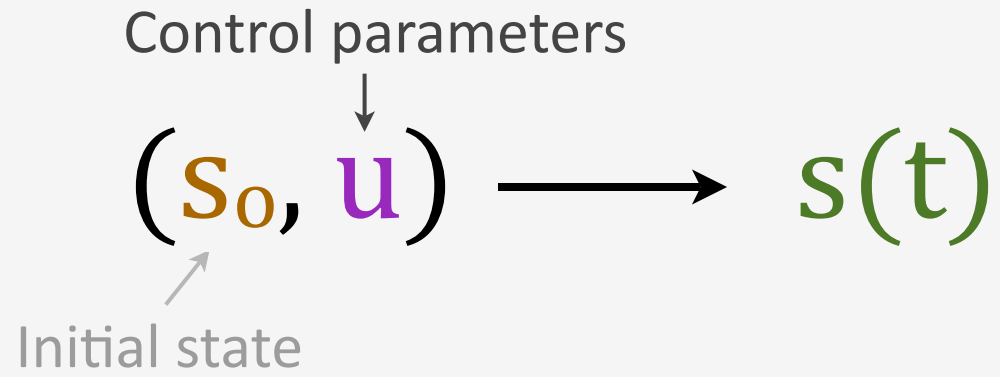
Foot Region

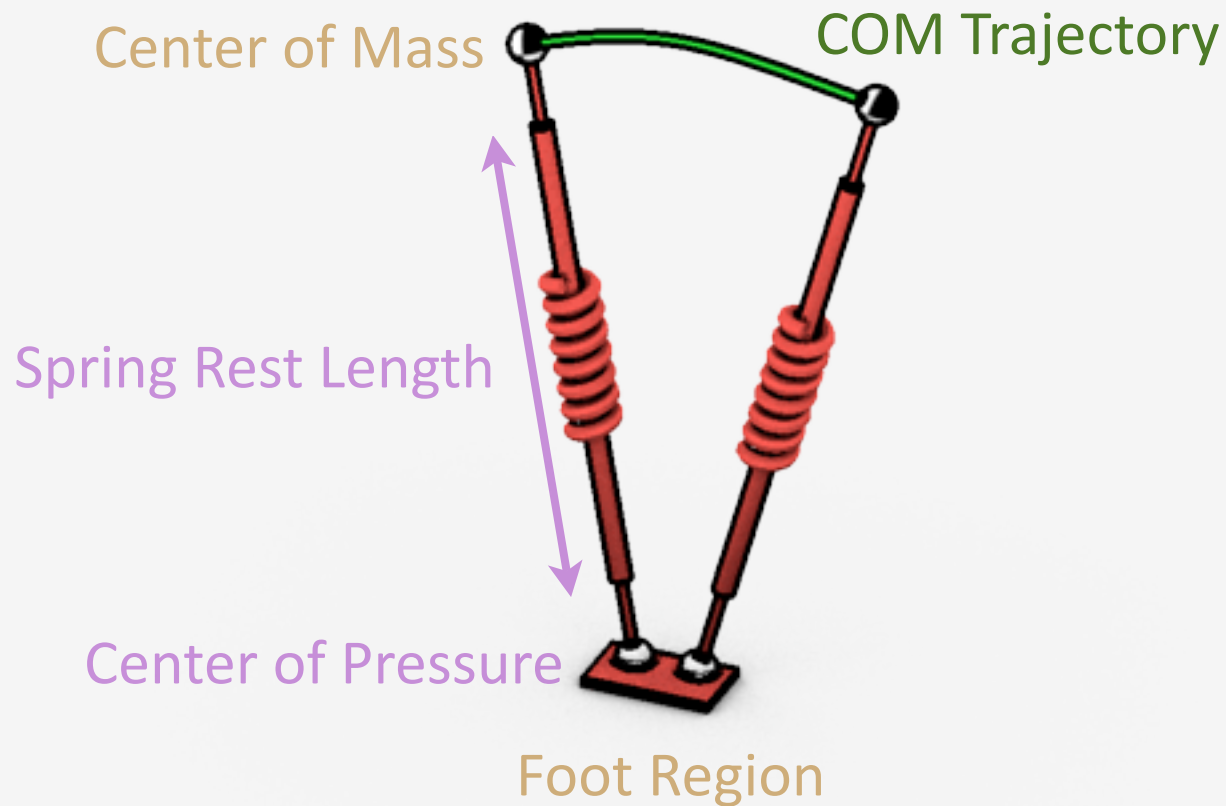
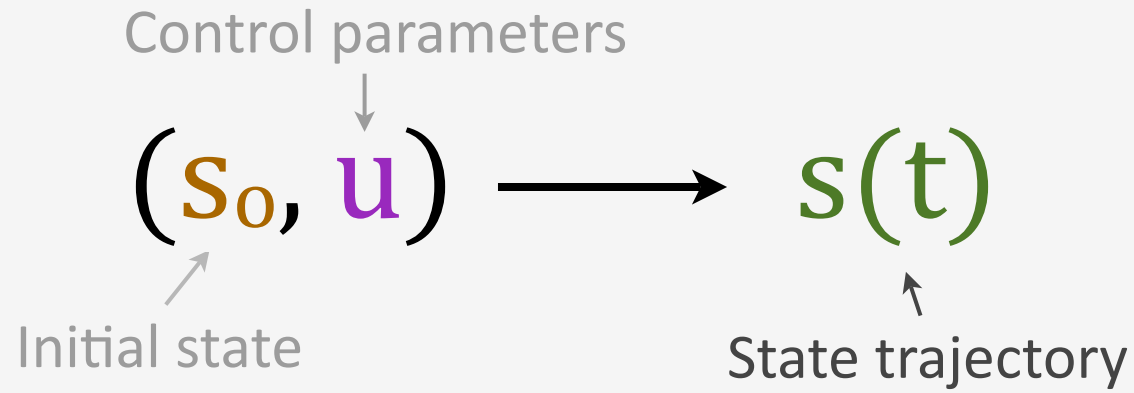
(s_0, u) \longrightarrow $s(t)$
Initial state

Center of Mass

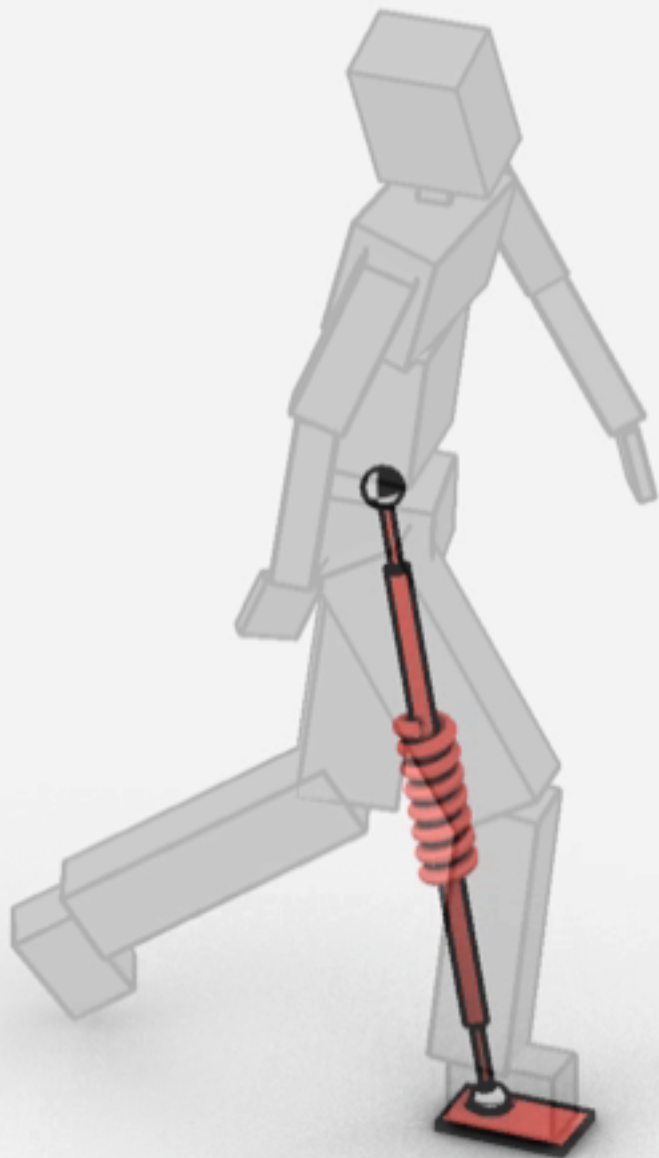


Foot Region

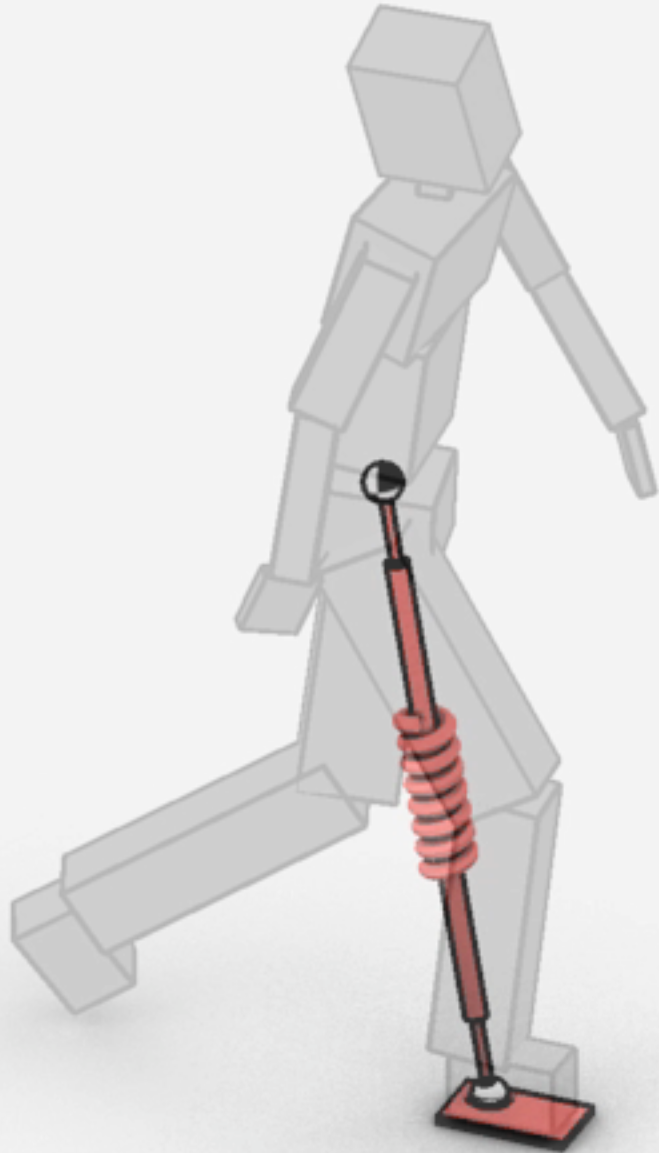




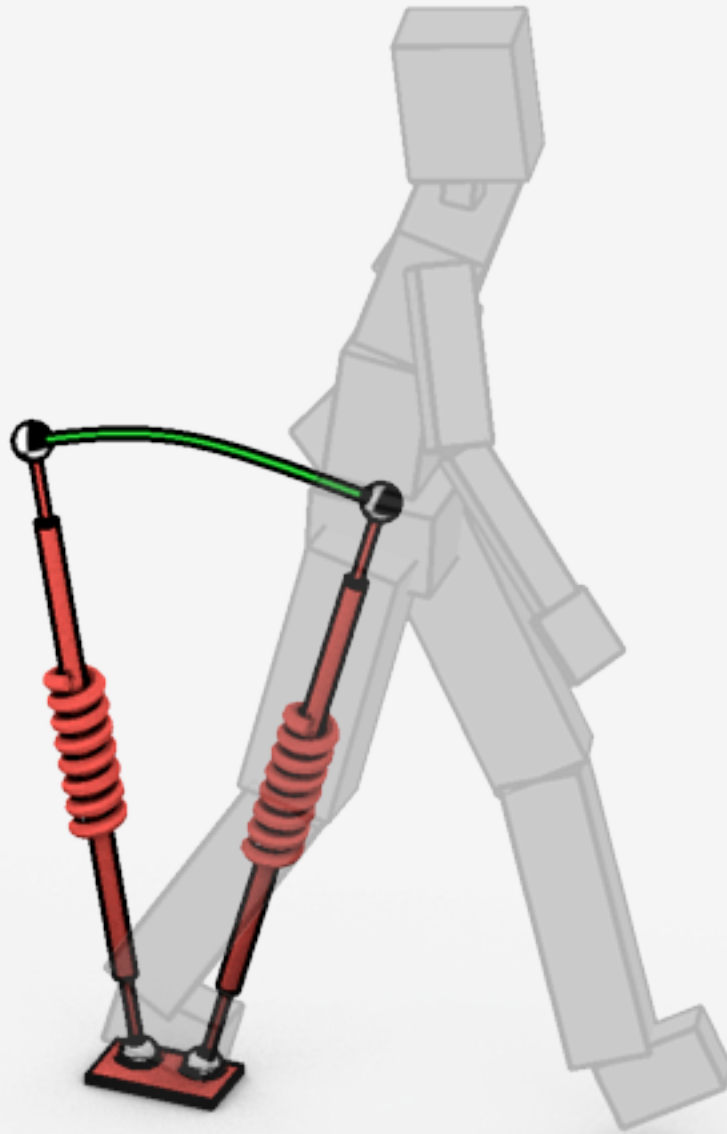
Walking



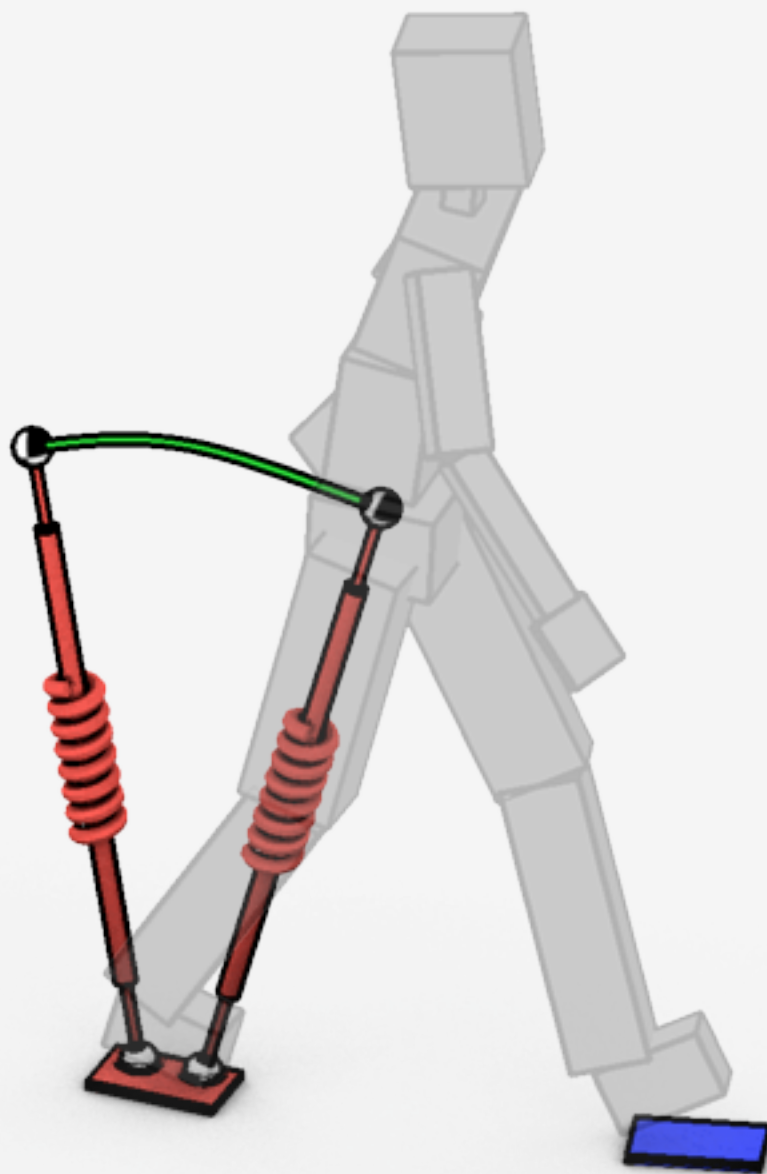
Walking



Walking

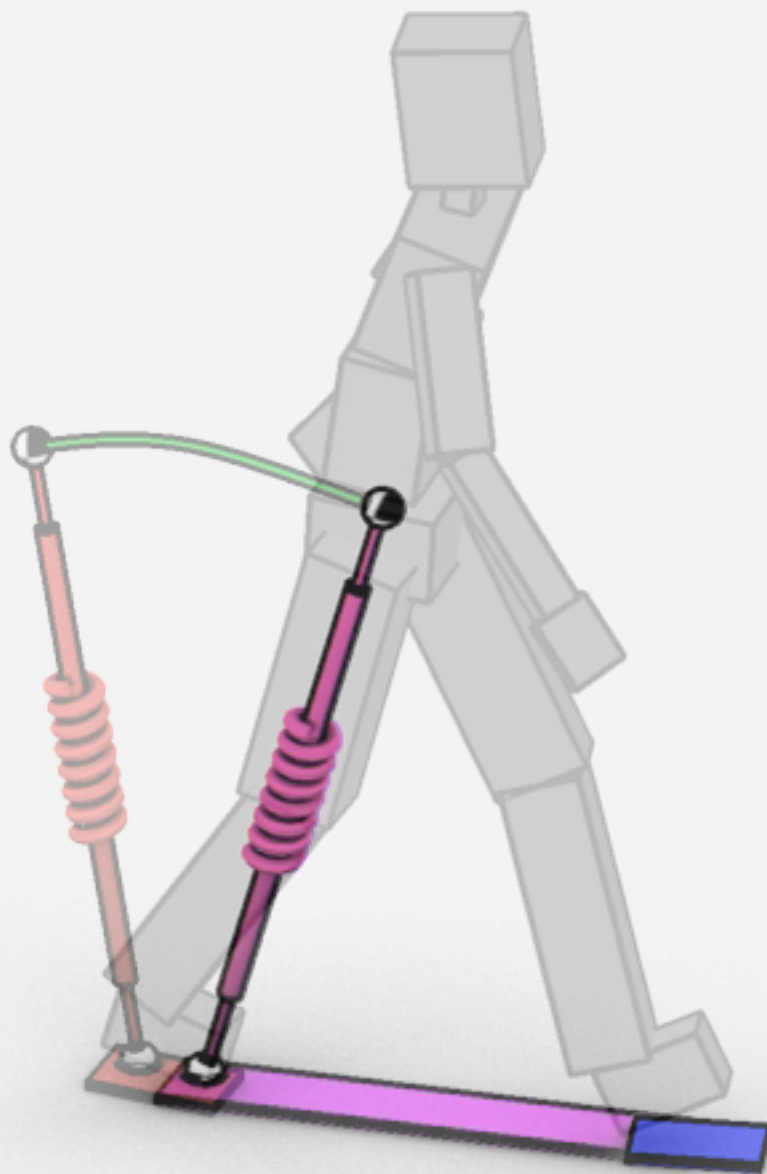


Walking



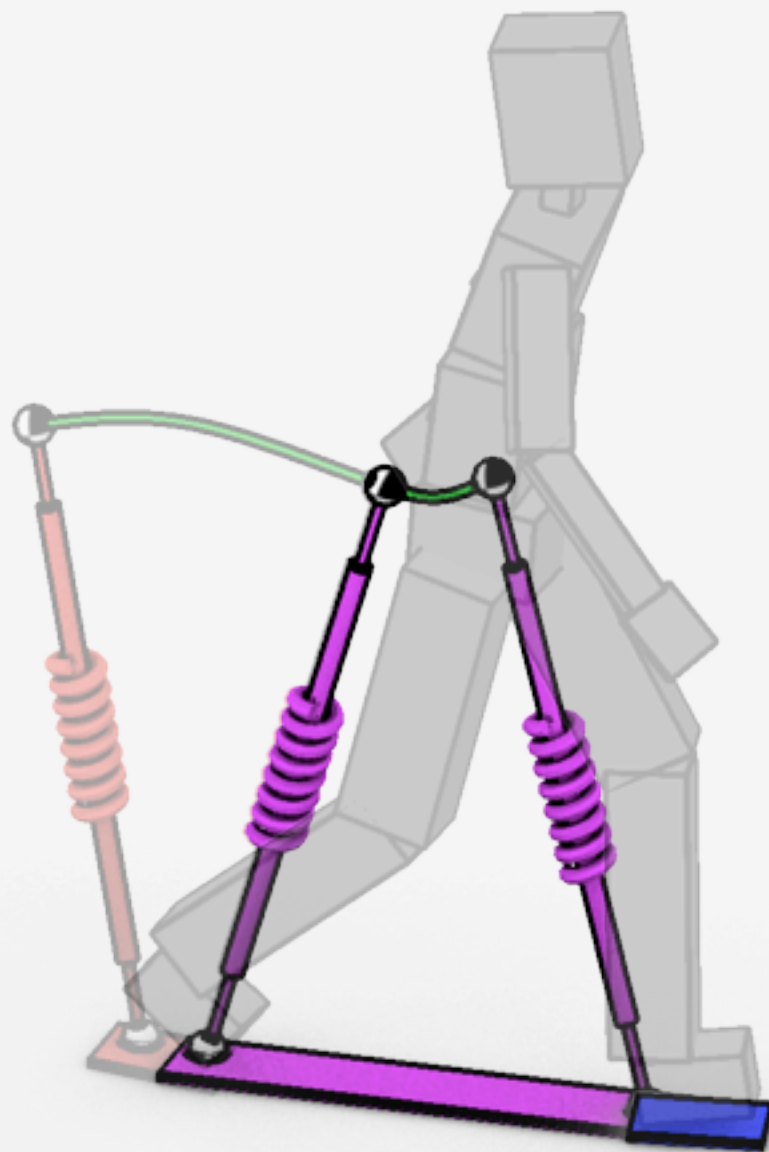
Next Foot Contact

Walking



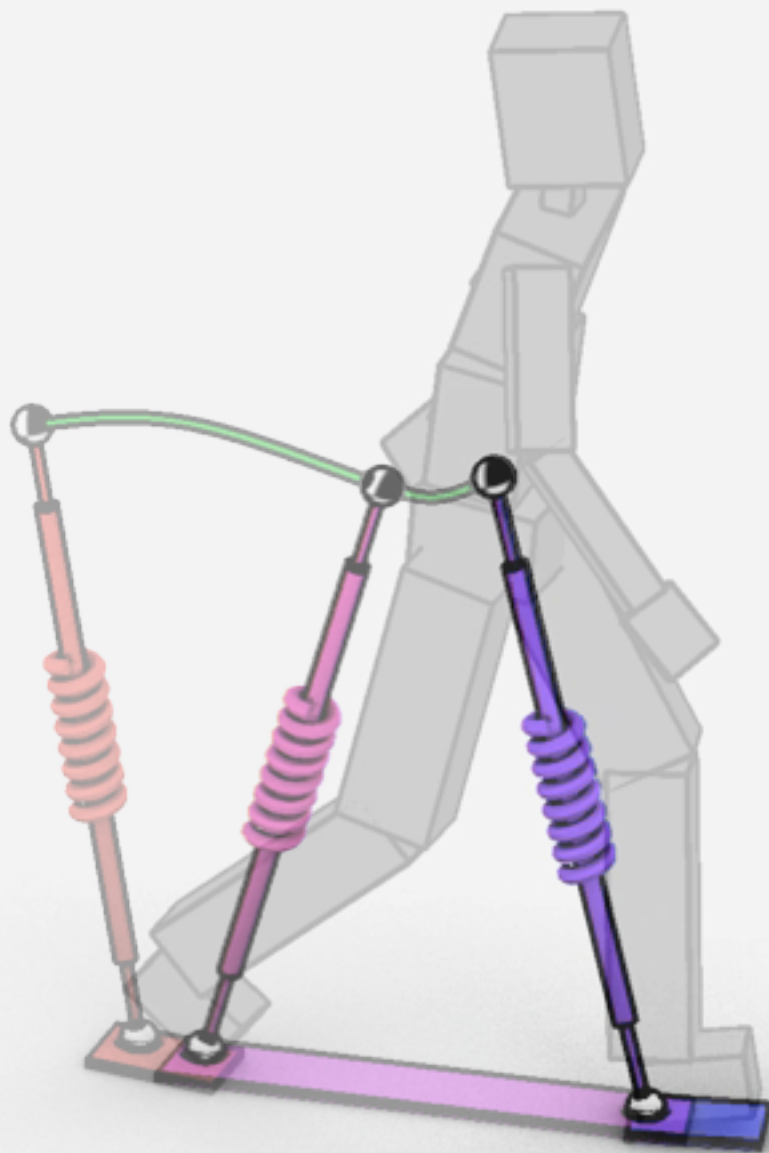
Next Foot Contact

Walking

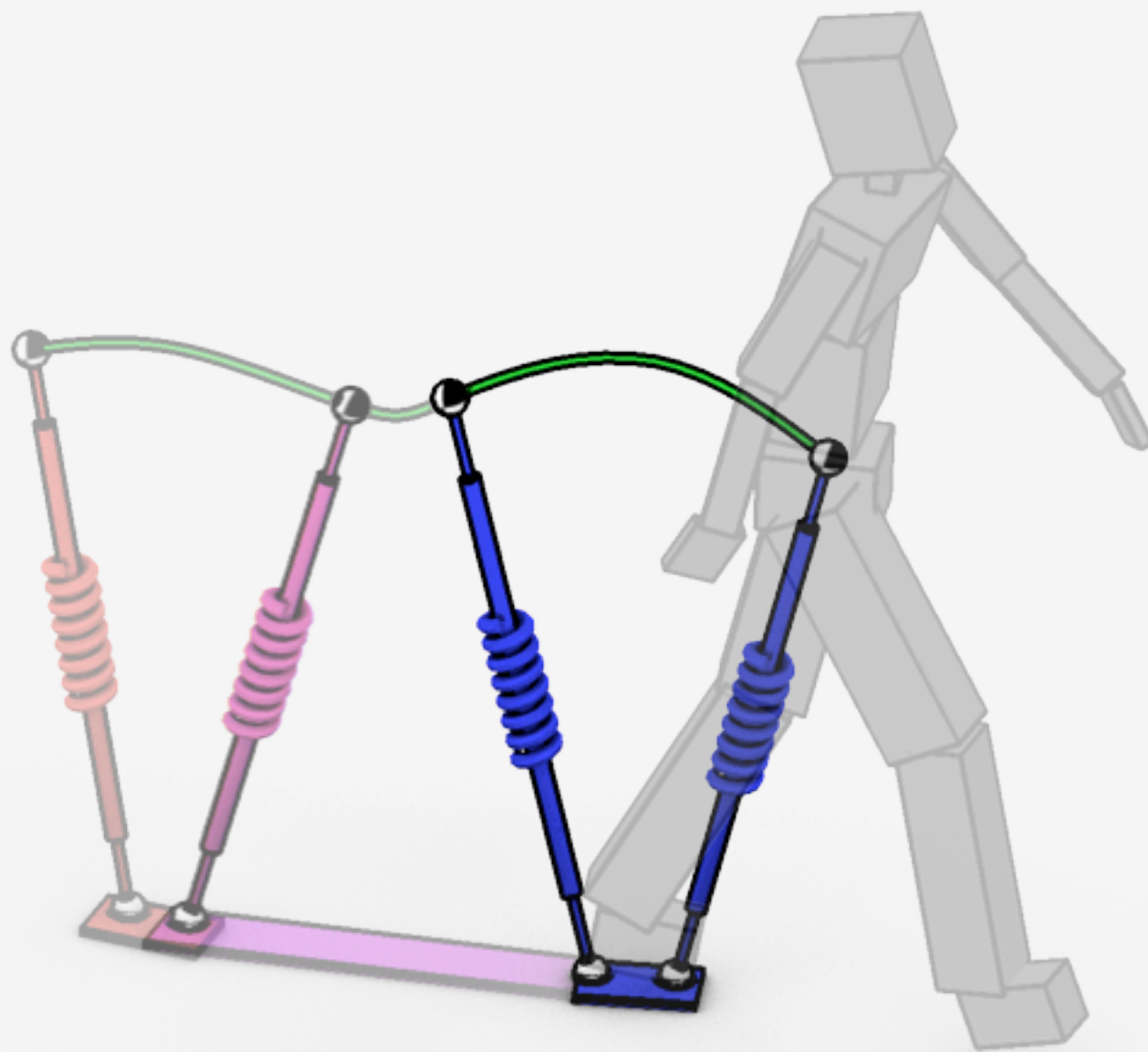


Next Foot Contact

Walking

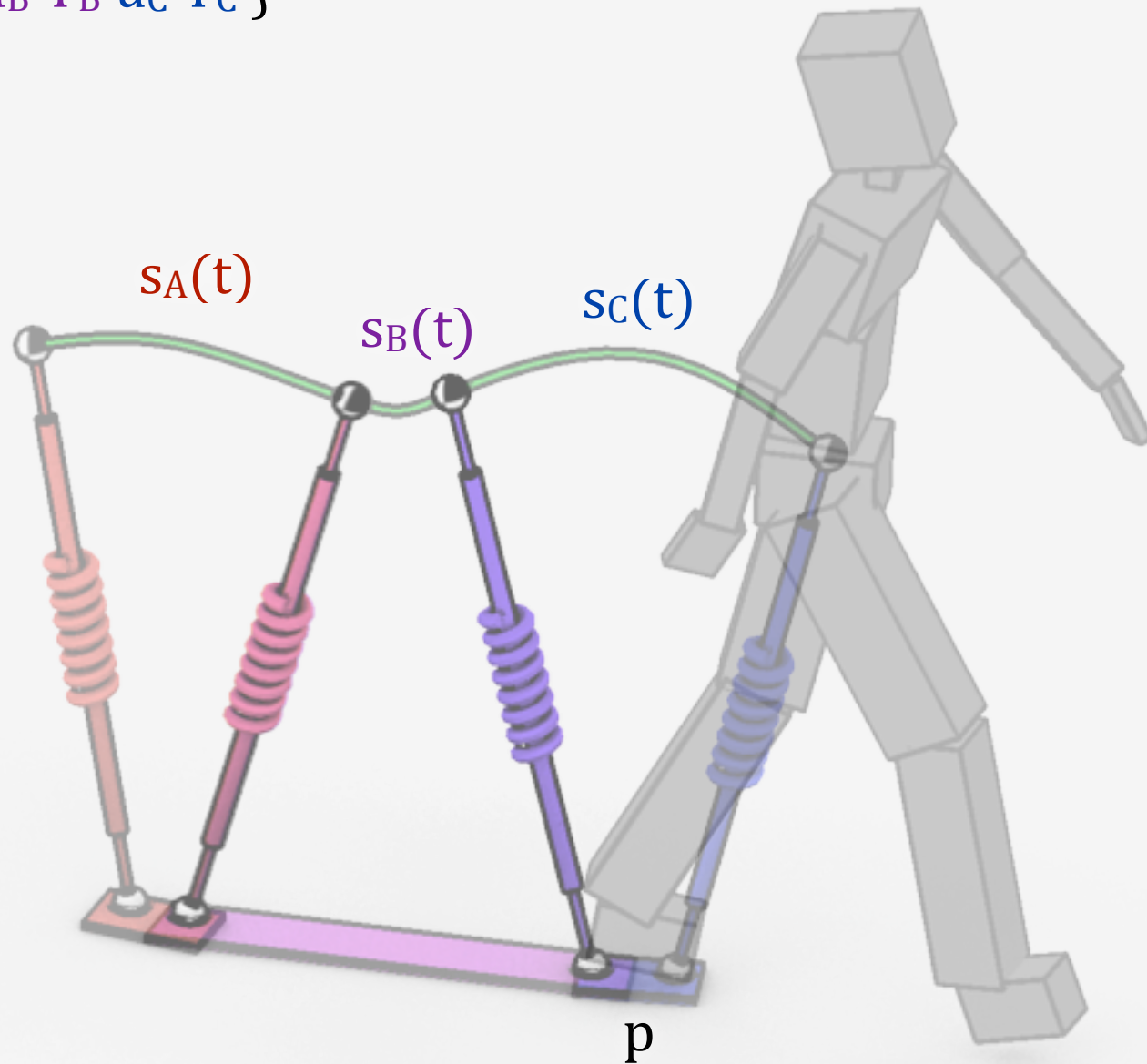


Walking



Walking

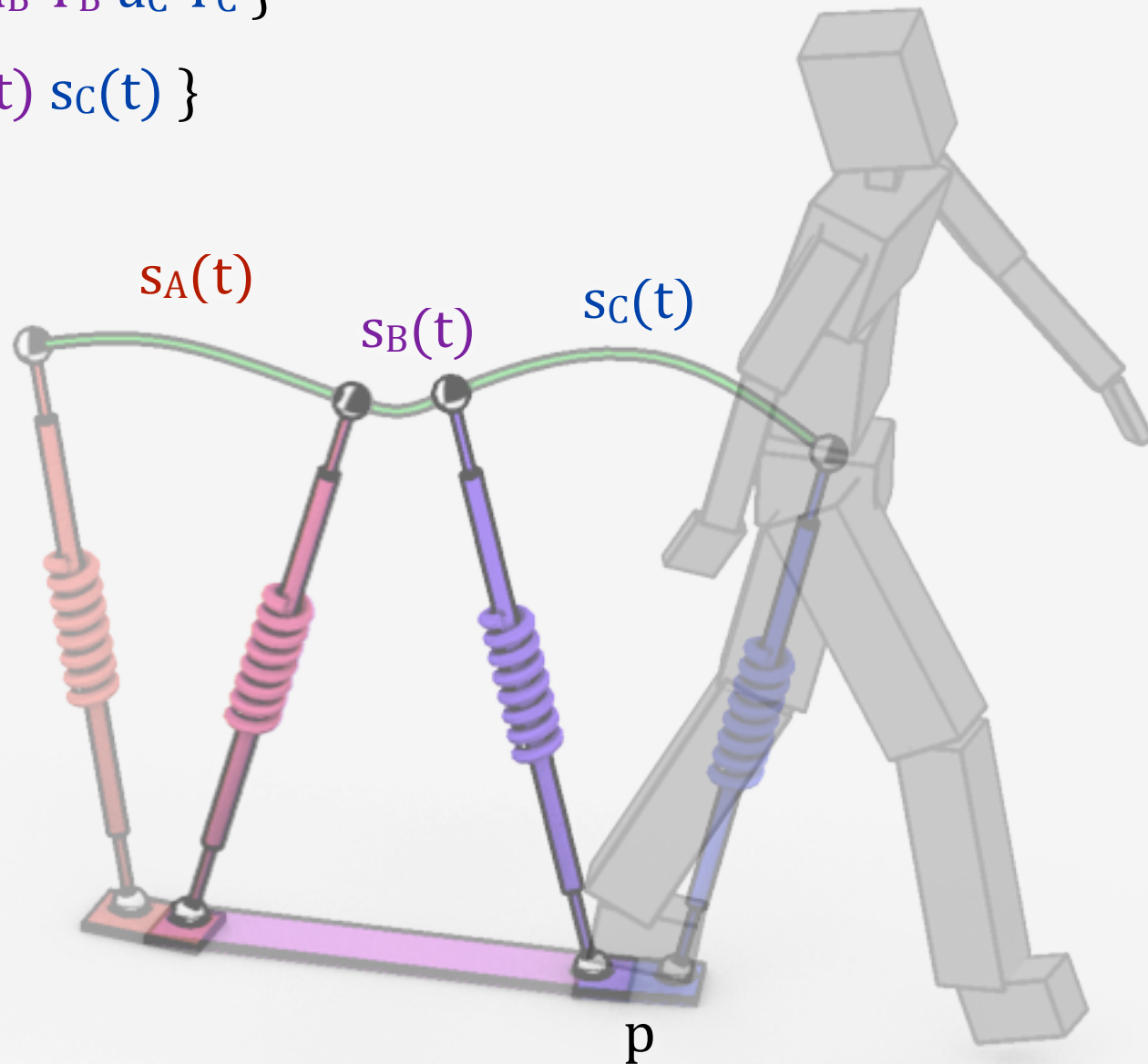
$$U = \{ \mathbf{u}_A \mathbf{T}_A \mathbf{p} \ \mathbf{u}_B \mathbf{T}_B \ \mathbf{u}_C \mathbf{T}_C \}$$



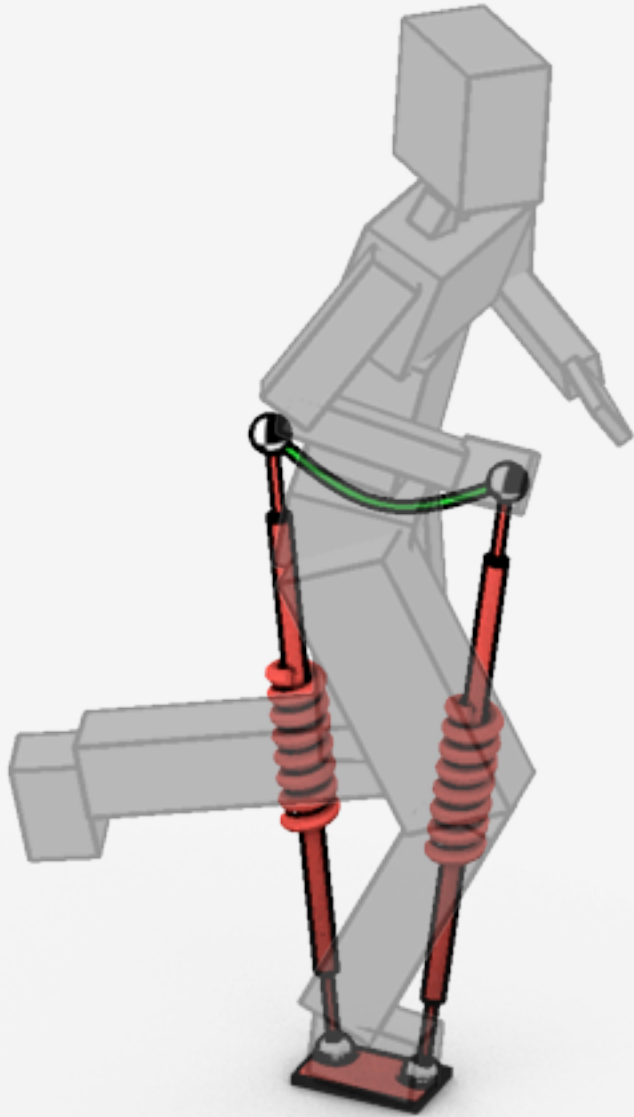
Walking

$$U = \{ \mathbf{u}_A \mathbf{T}_A \mathbf{p} \ \mathbf{u}_B \mathbf{T}_B \ \mathbf{u}_C \mathbf{T}_C \}$$

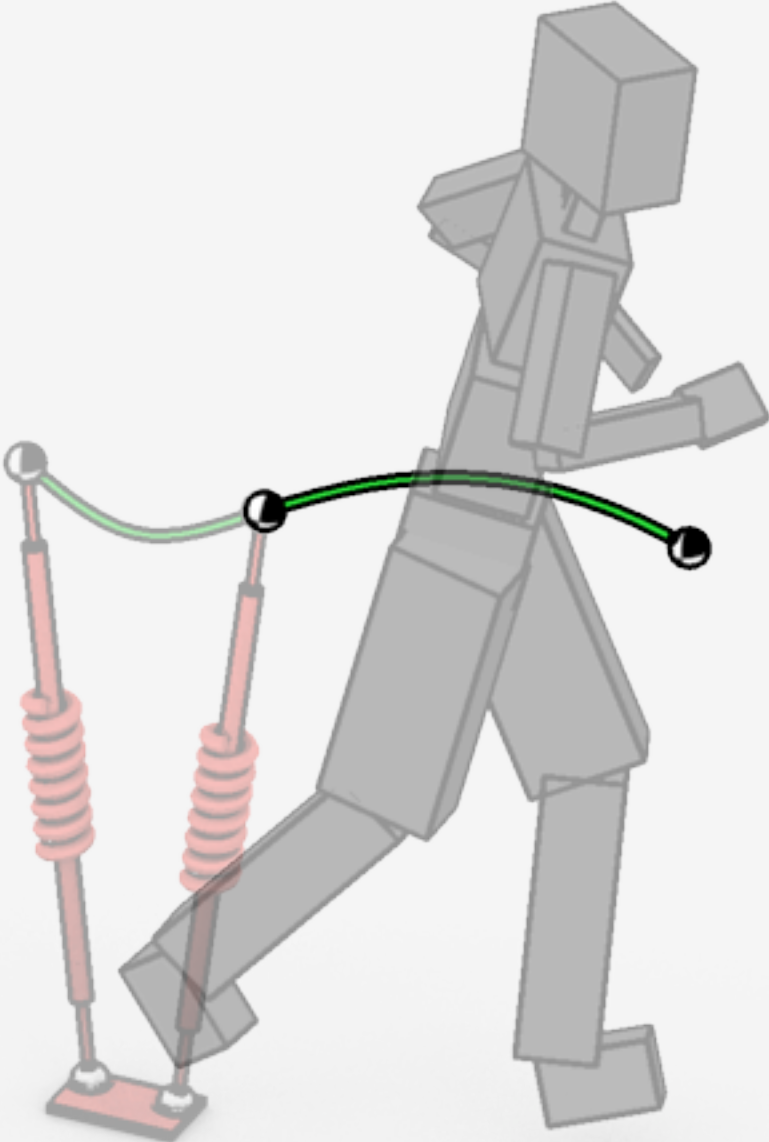
$$S(t) = \{ s_A(t) \ s_B(t) \ s_C(t) \}$$



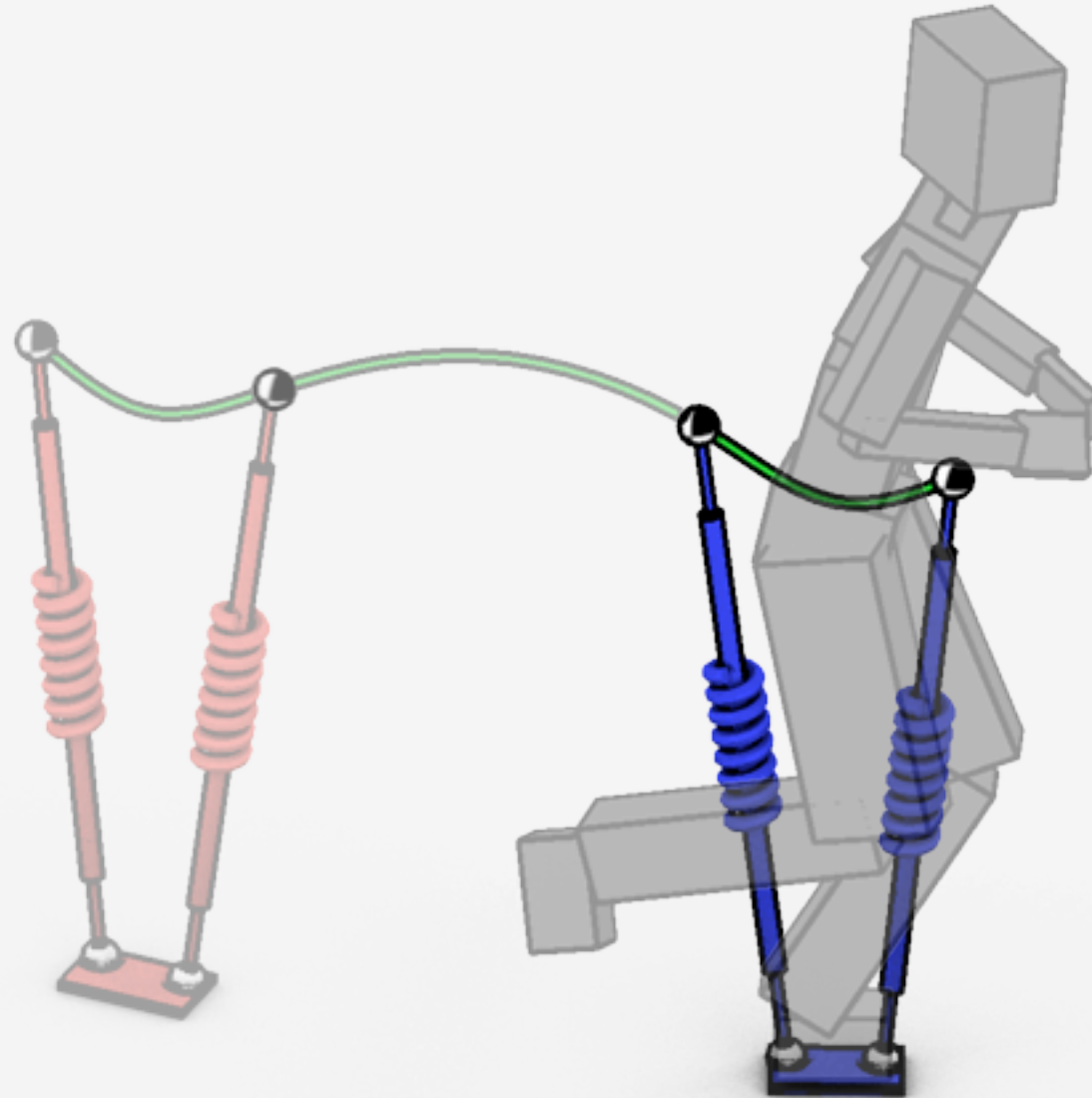
Running



Running



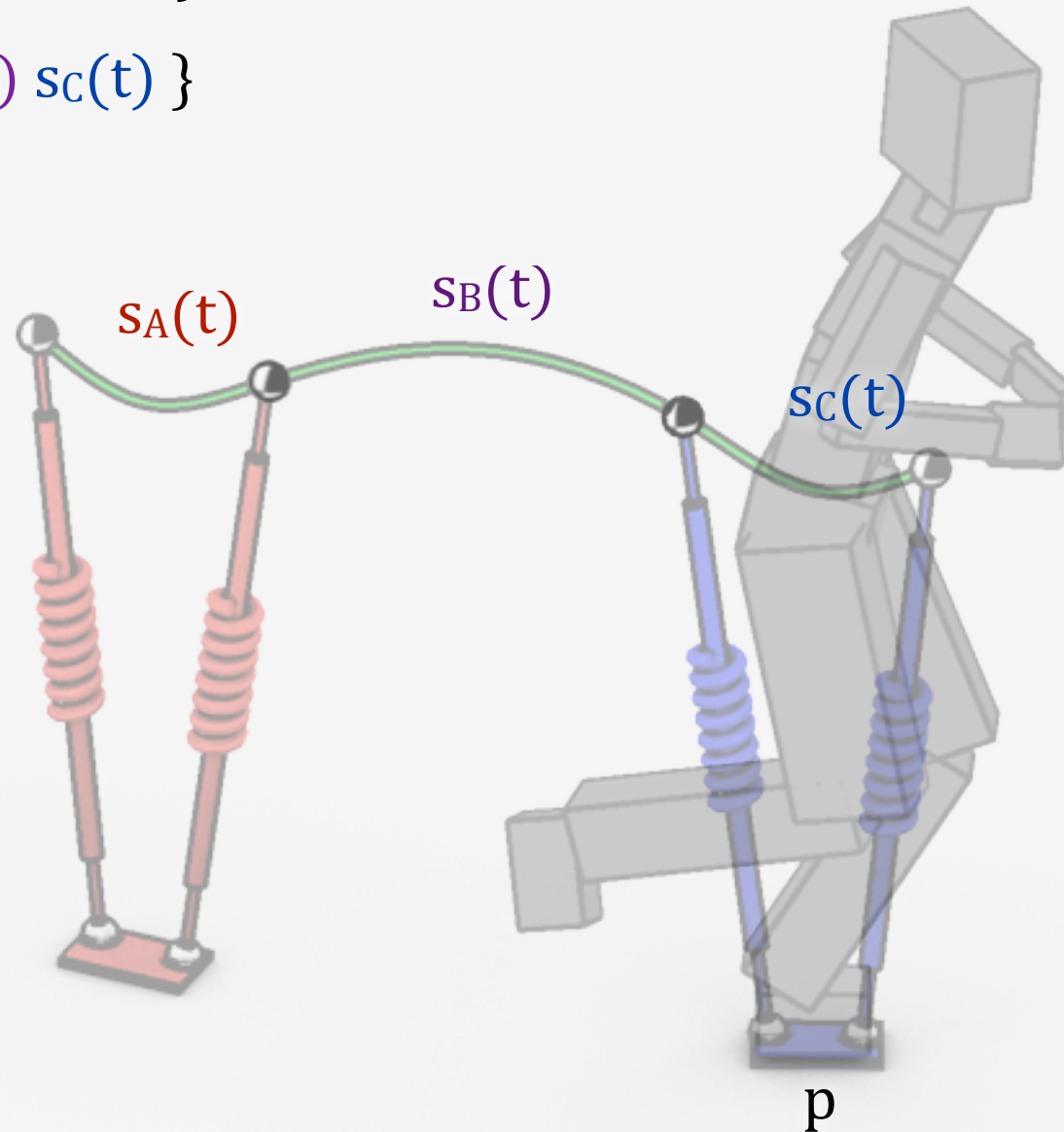
Running



Running

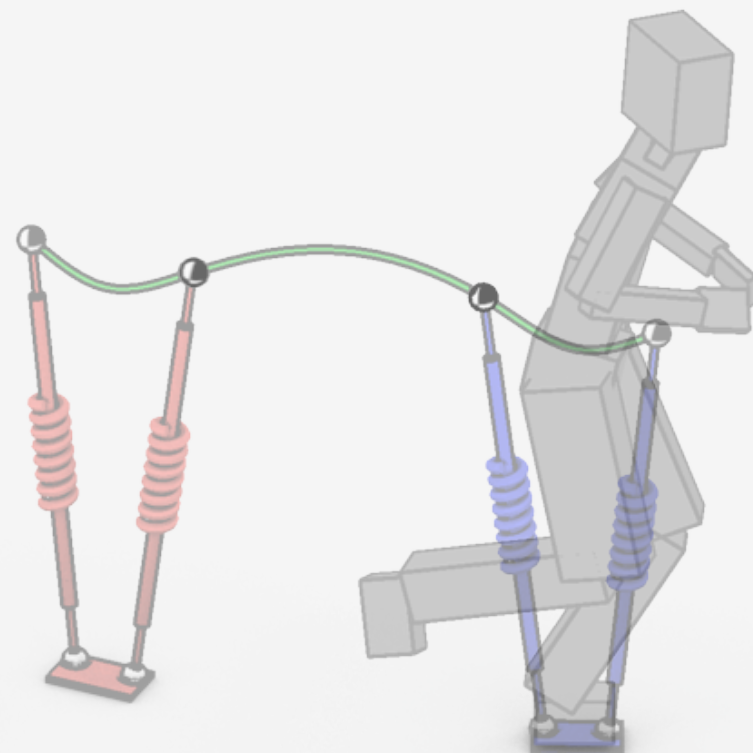
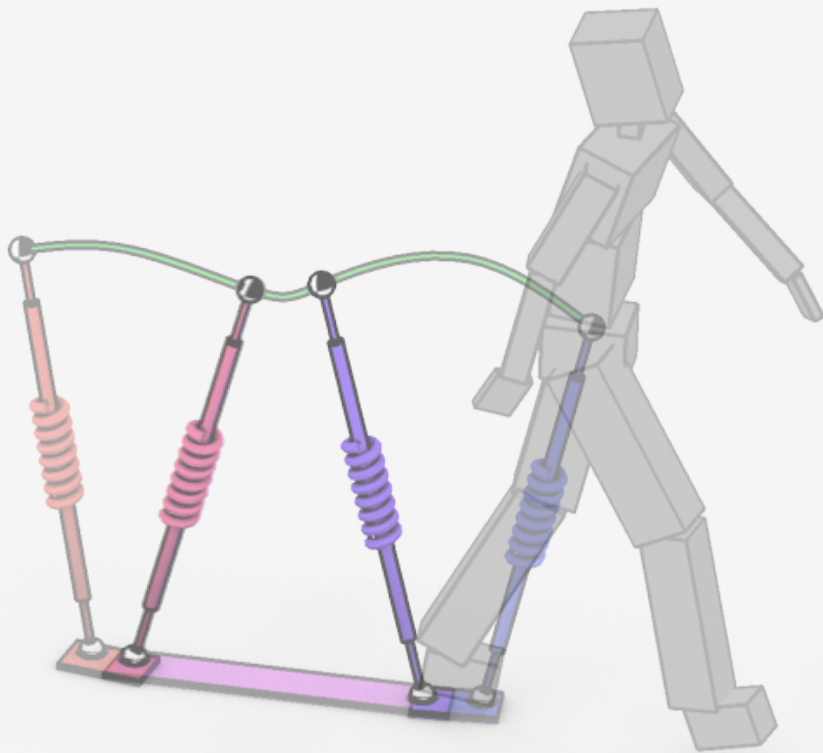
$$U = \{ u_A T_A T_B p u_C T_C \}$$

$$S(t) = \{ s_A(t) s_B(t) s_C(t) \}$$



General Gaits:

sequence of single/double SLIP and flight phases



General Gaits:

sequence of single/double SLIP and flight phases

Concatenation of
phase parameters:

$$\{ u_i T_i p \}$$

$$(s_0, U) \longrightarrow S(t)$$

Initial conditions

Sequence of
state trajectories:

$$\{ s_i(t) \}$$

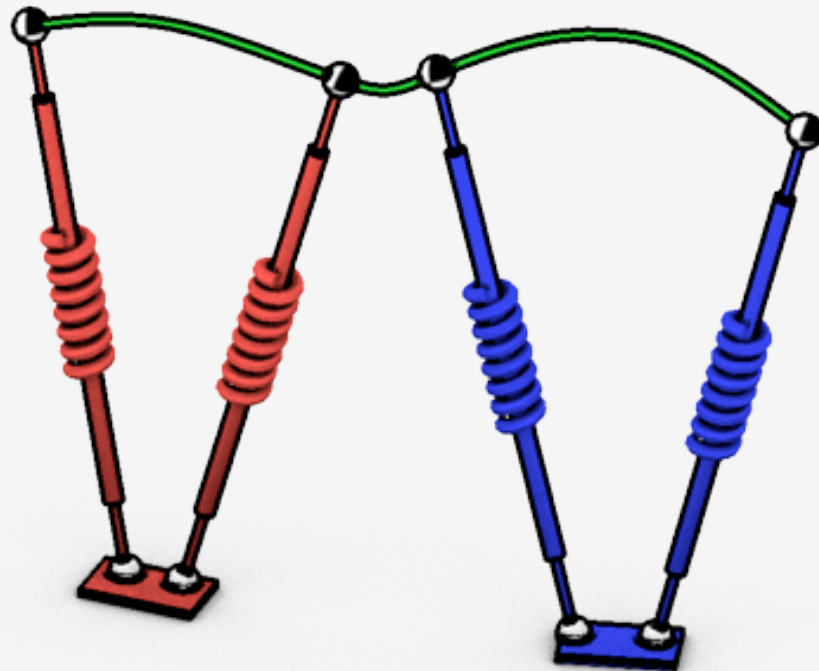
What is the low-dimensional motion model?

How to optimize the model?

How to map motion to full-dimensional character?

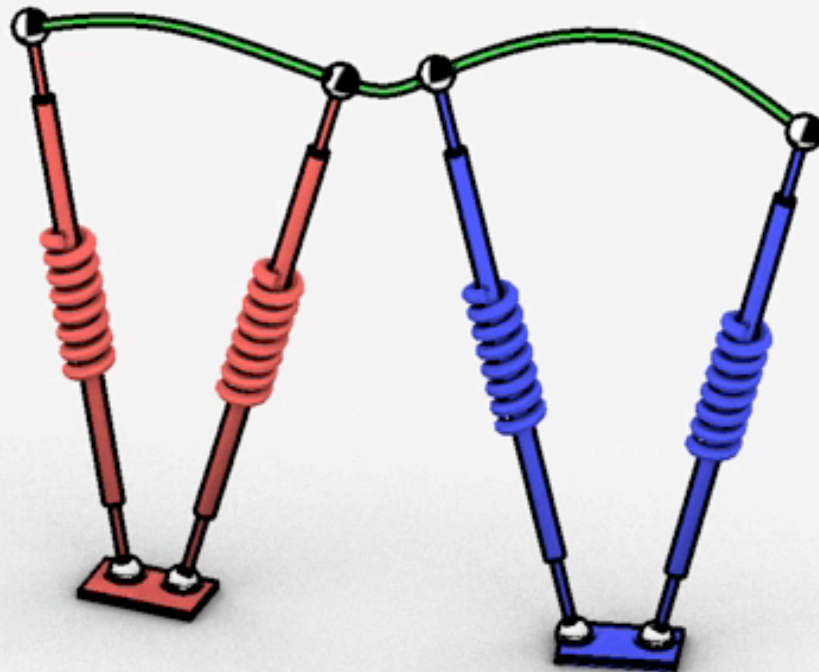
Optimize U^* to satisfy goals:

$$U^* = \operatorname{argmin}_U \sum_i w_i g_i(S(t))$$



Optimize U^* to satisfy goals:

$$U^* = \operatorname{argmin}_U \sum_i w_i g_i(S(t))$$



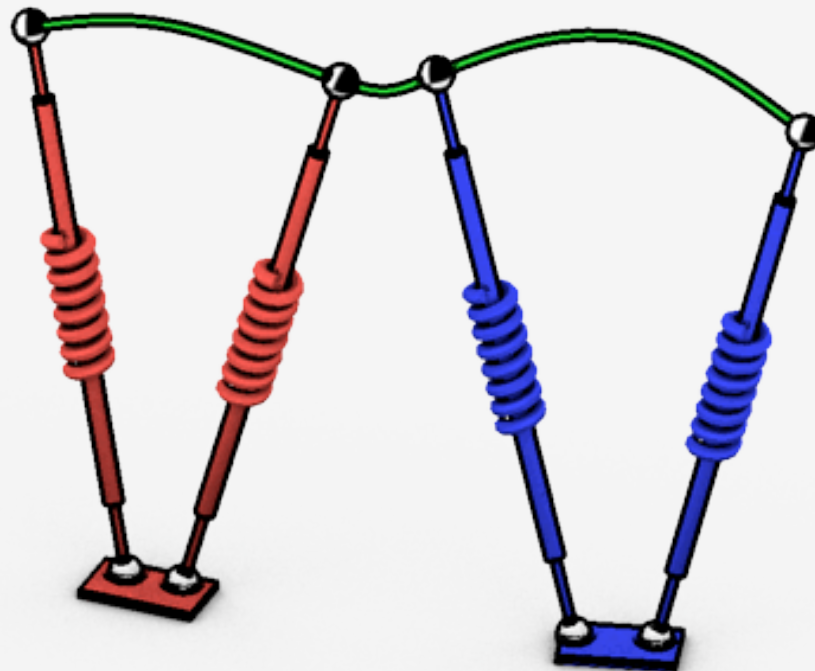
Optimize U^* to satisfy goals:

$$U^* = \operatorname{argmin}_U \sum_i w_i g_i(S(t))$$

General nonlinear optimization

23-dimensional

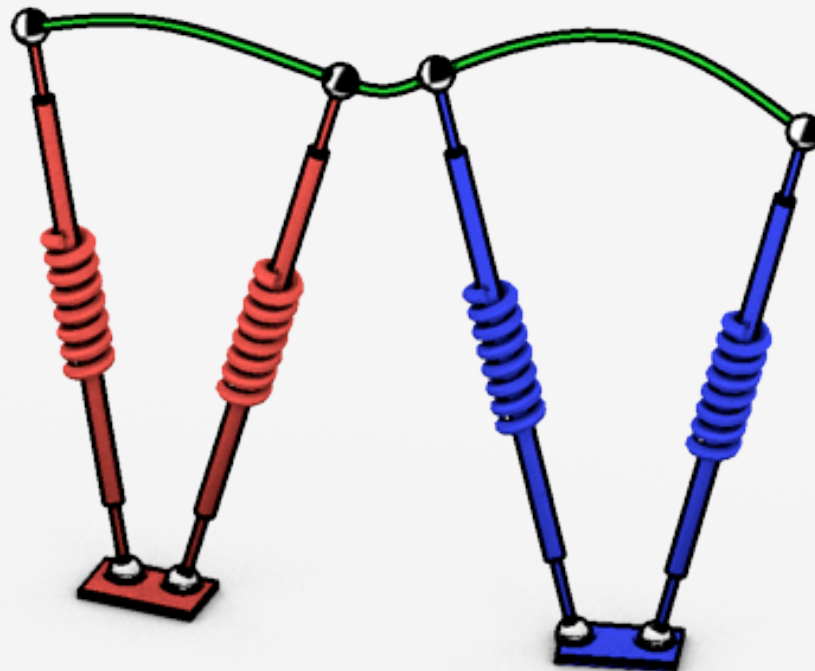
Solved using CMA [Hansen 2006]



Optimize U^* to satisfy goals:

$$U^* = \operatorname{argmin}_U \sum_i w_i g_i(S(t))$$

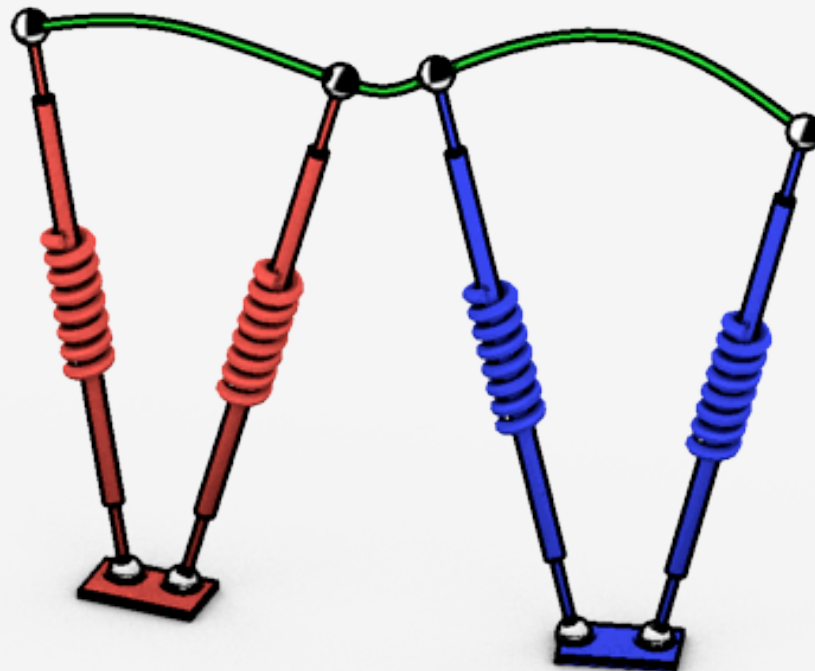
- Step Distance
- Step Duration
- Heading
- Jump Height
- Smoothness



Optimize U^* to satisfy goals:

$$U^* = \operatorname{argmin}_U \sum_i w_i g_i(S(t))$$

- Step Distance
- Step Duration
- Heading
- Jump Height
- Smoothness
- **Your Goal Here!**

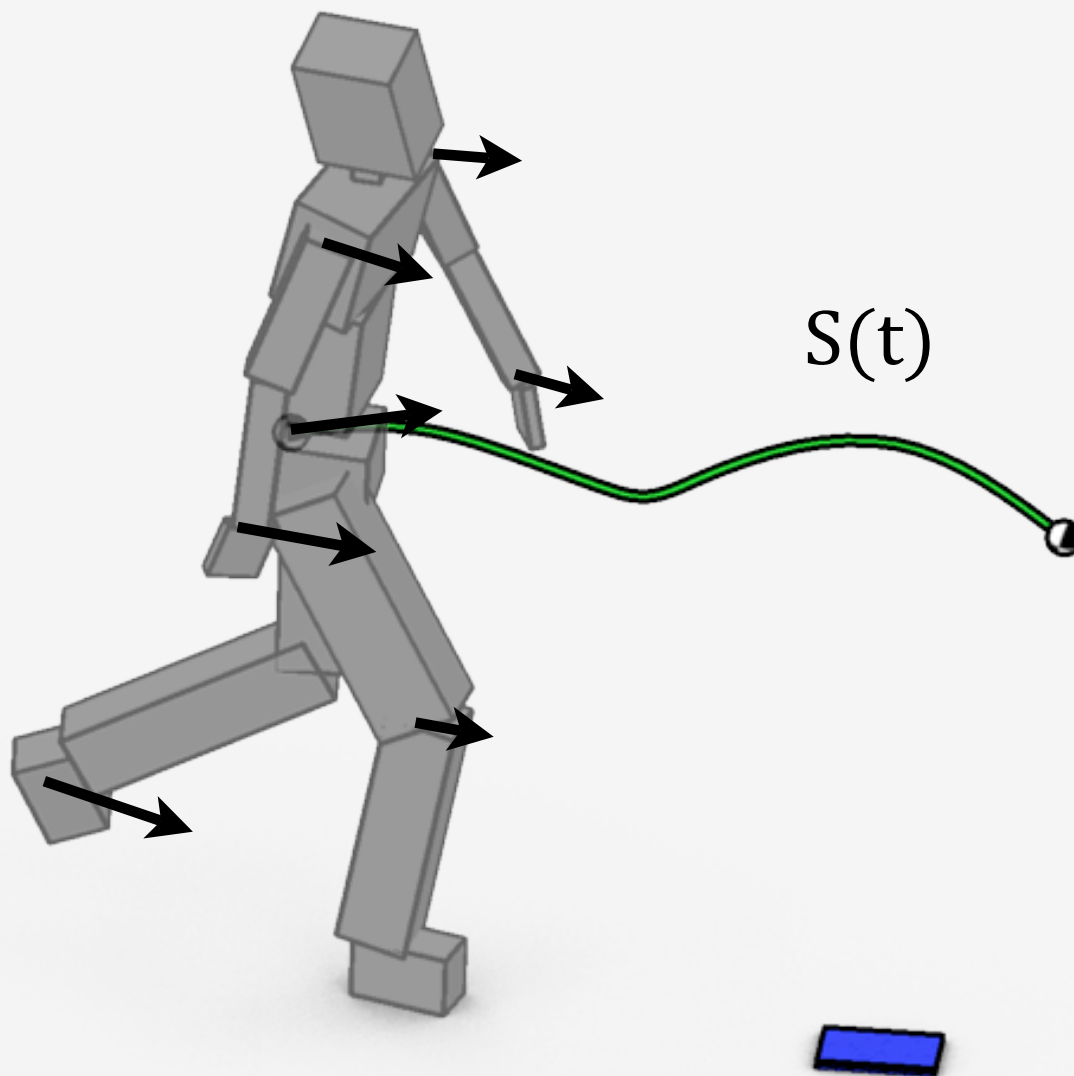


Given low-dimensional motion $S(t)$



Given low-dimensional motion $S(t)$

Want character torques T

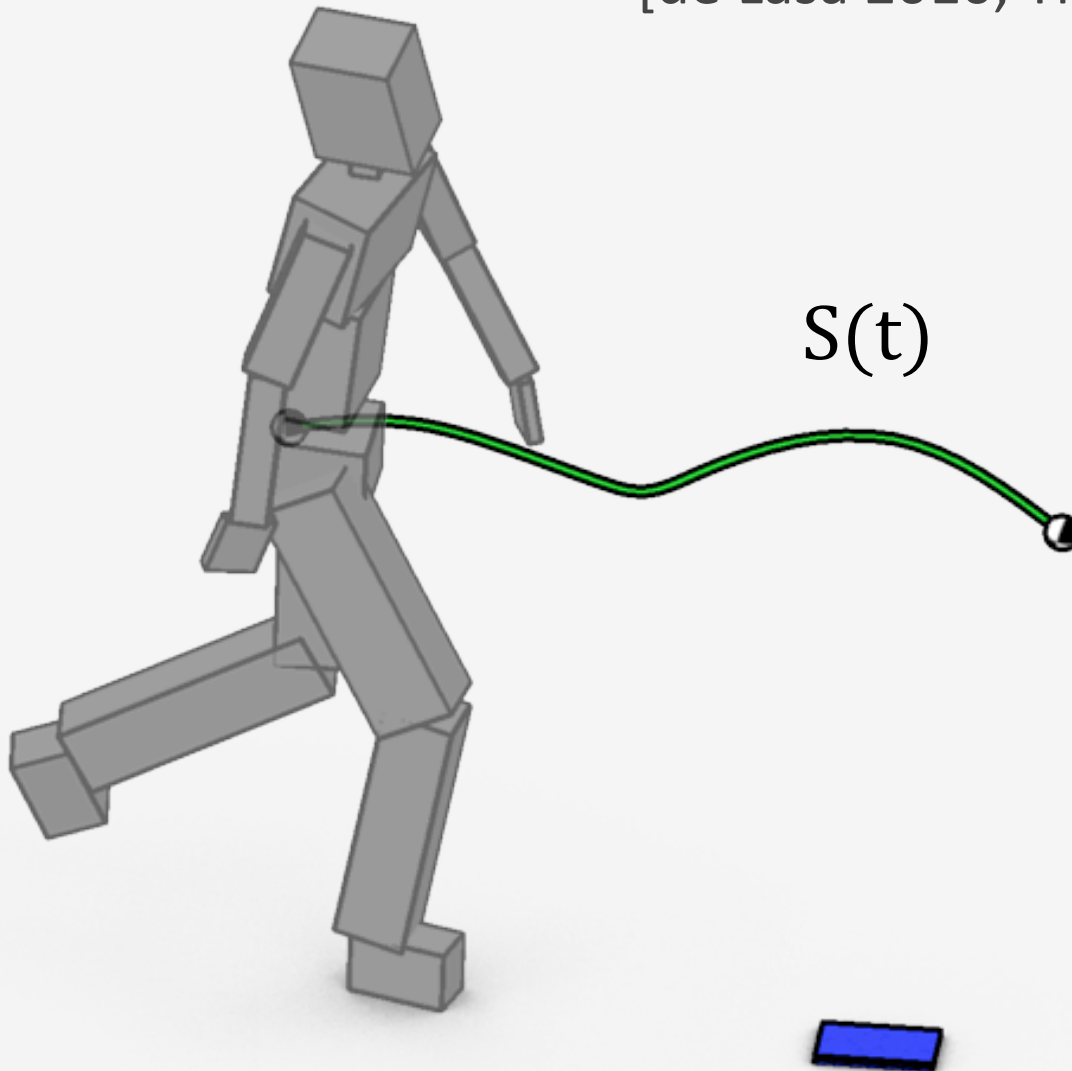


Given low-dimensional motion $S(t)$

Want character torques T

Use Feature-Based Locomotion Controller

[de Lasa 2010, Thurs. 4pm]

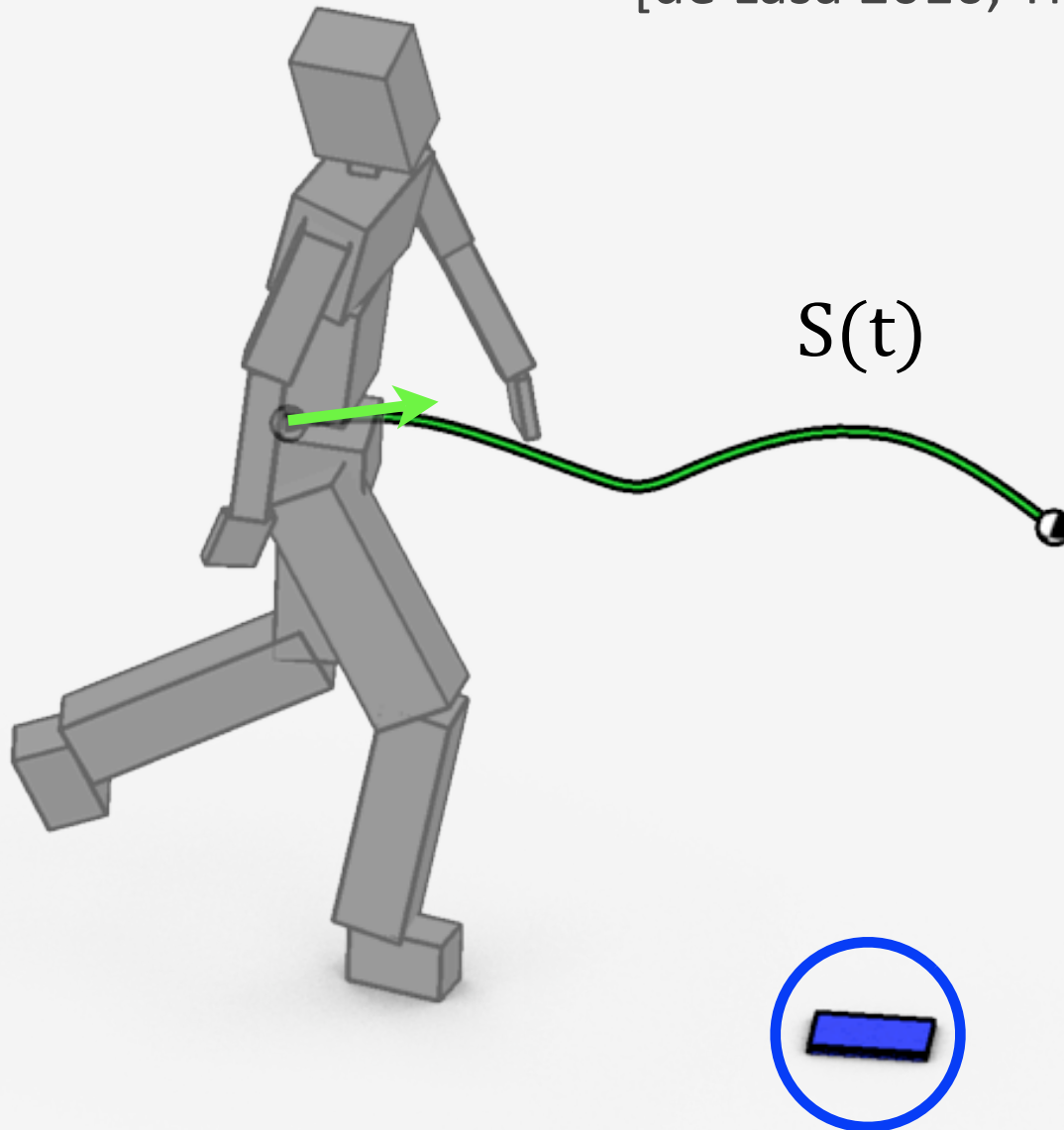


Given low-dimensional motion $S(t)$

Want character torques T

Use Feature-Based Locomotion Controller

[de Lasa 2010, Thurs. 4pm]

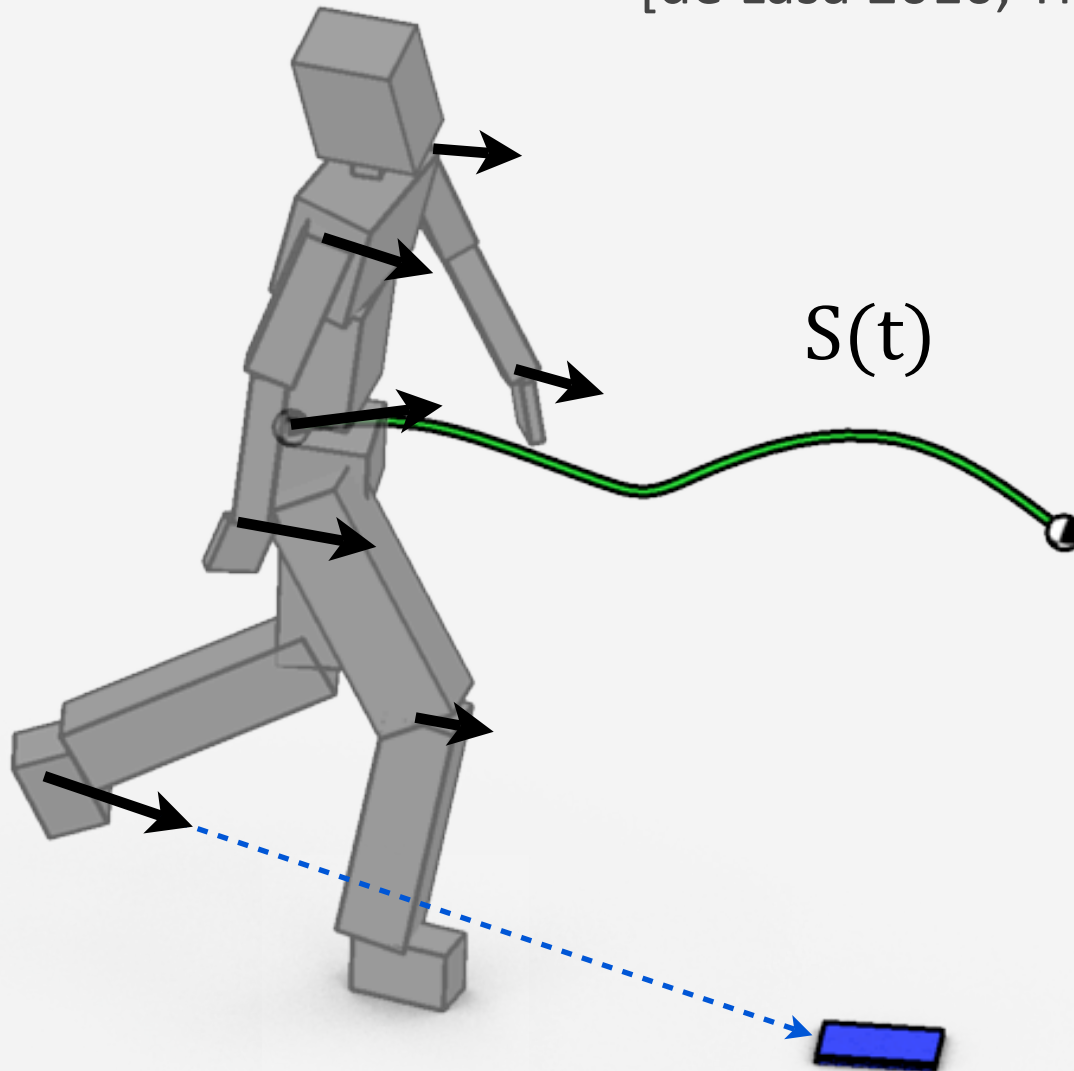


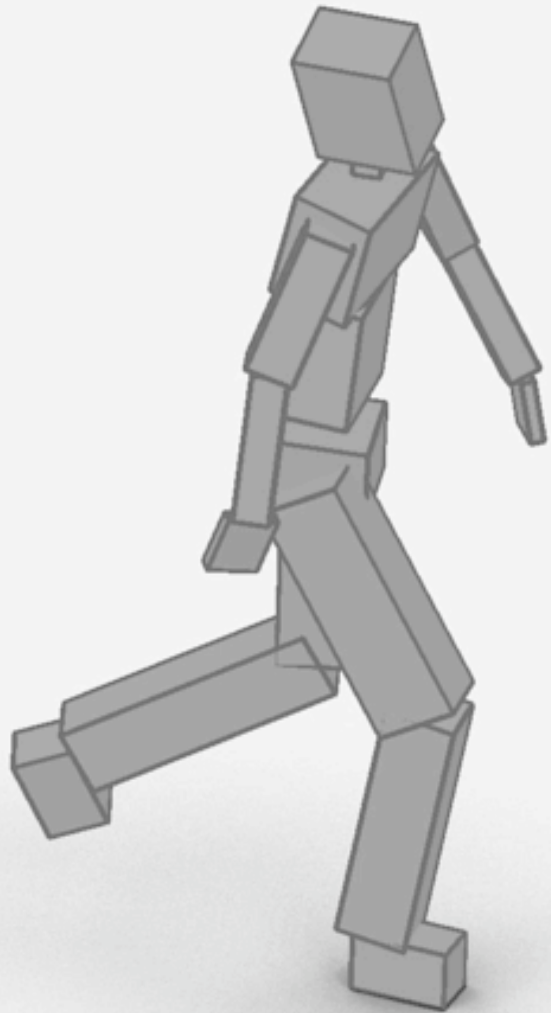
Given low-dimensional motion $S(t)$

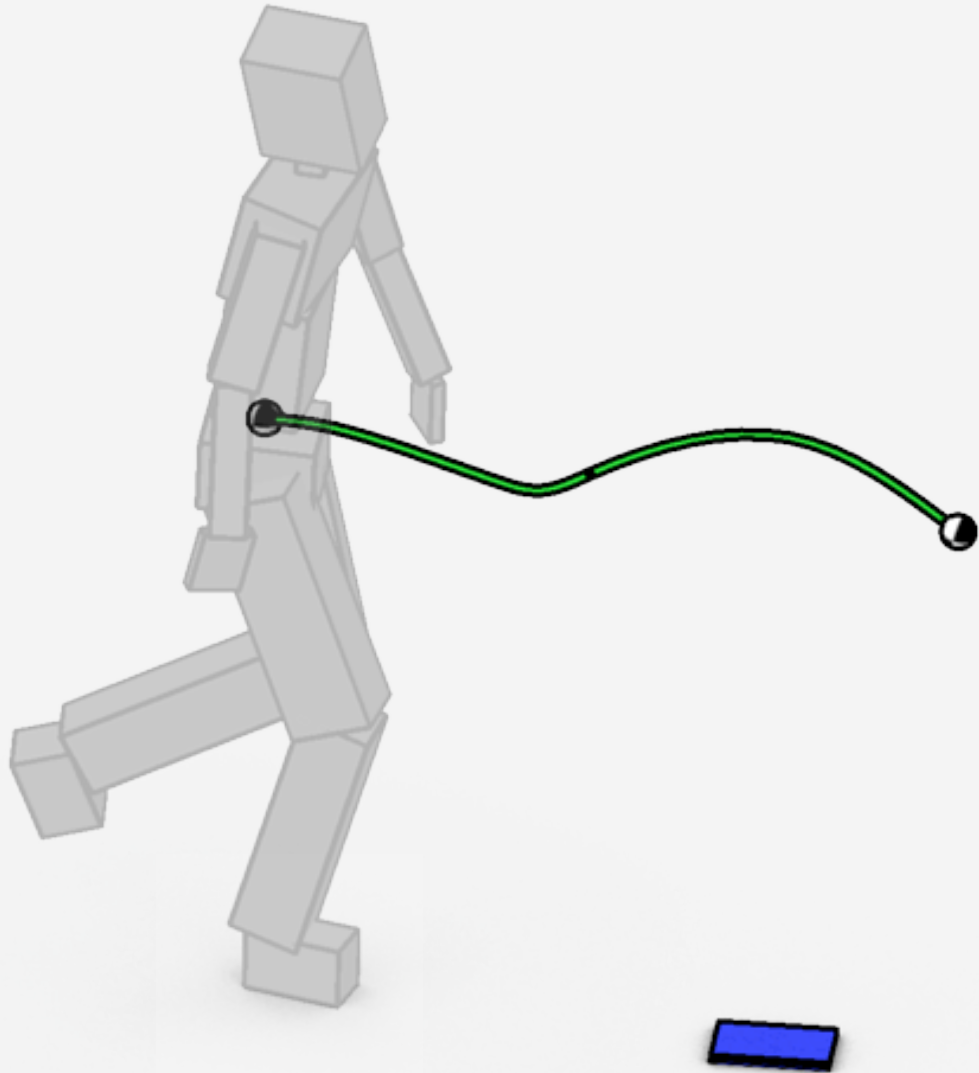
Want character torques T

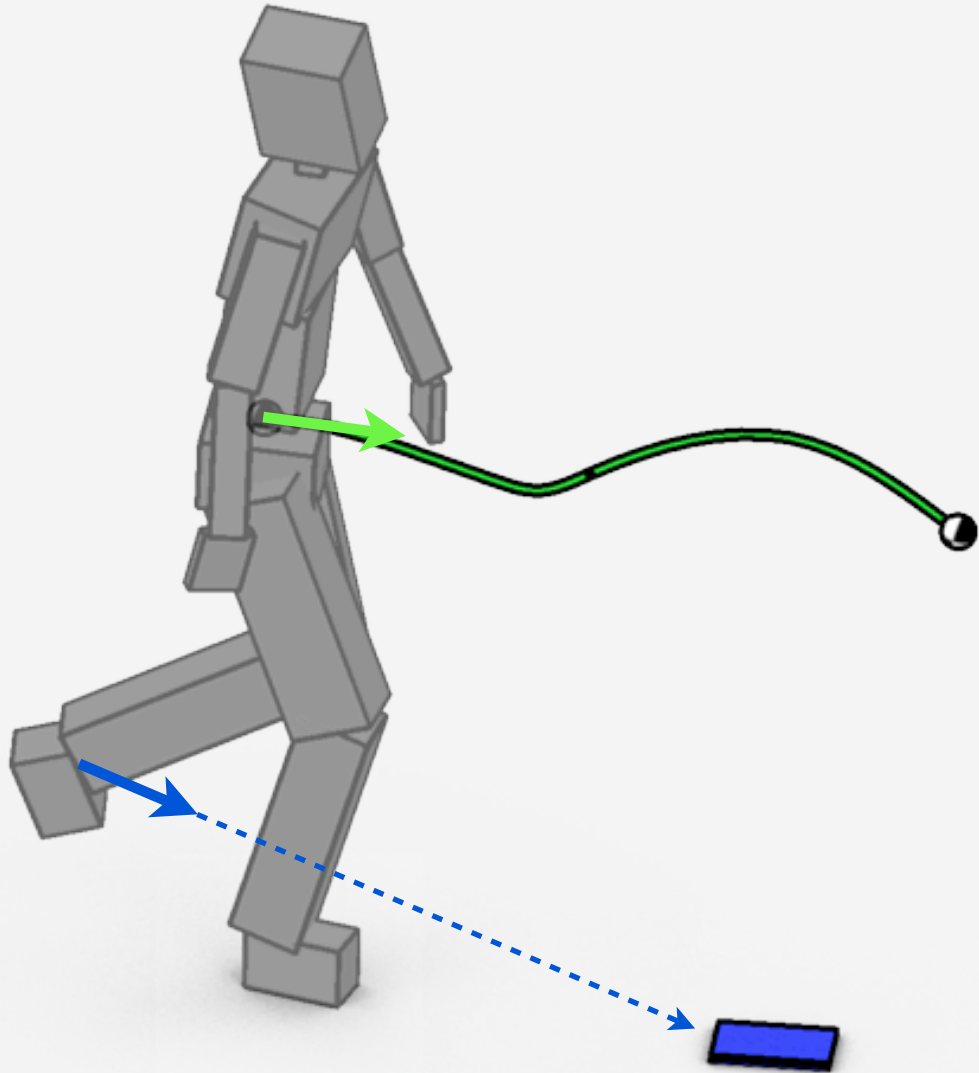
Use Feature-Based Locomotion Controller

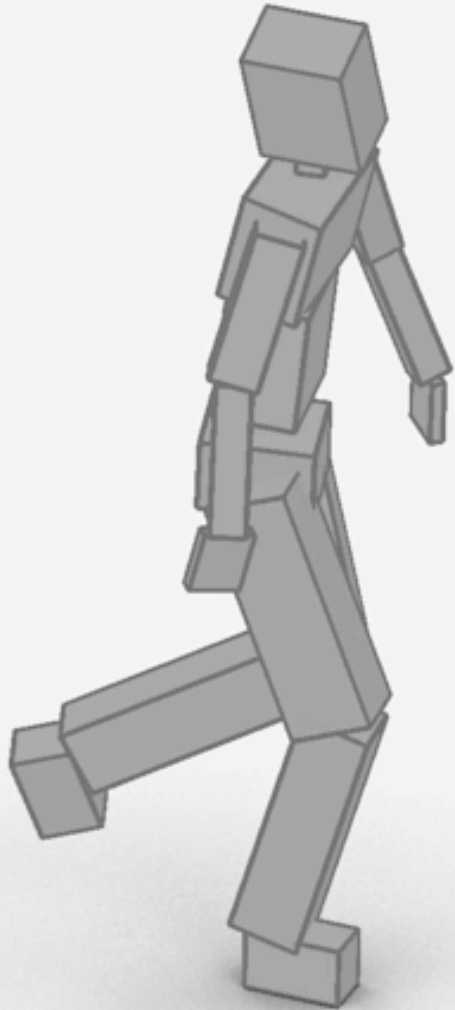
[de Lasa 2010, Thurs. 4pm]



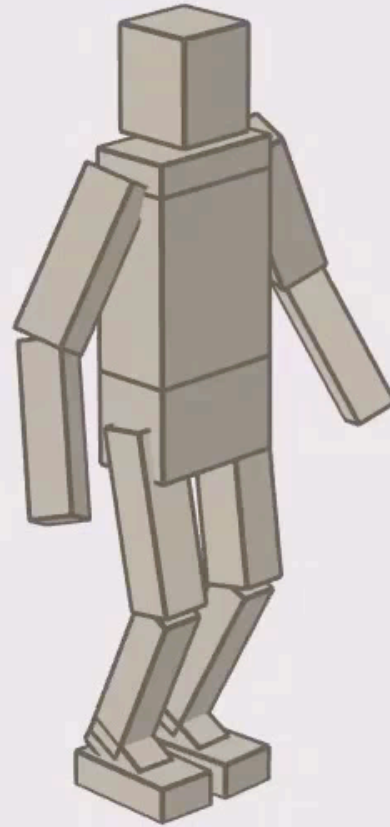








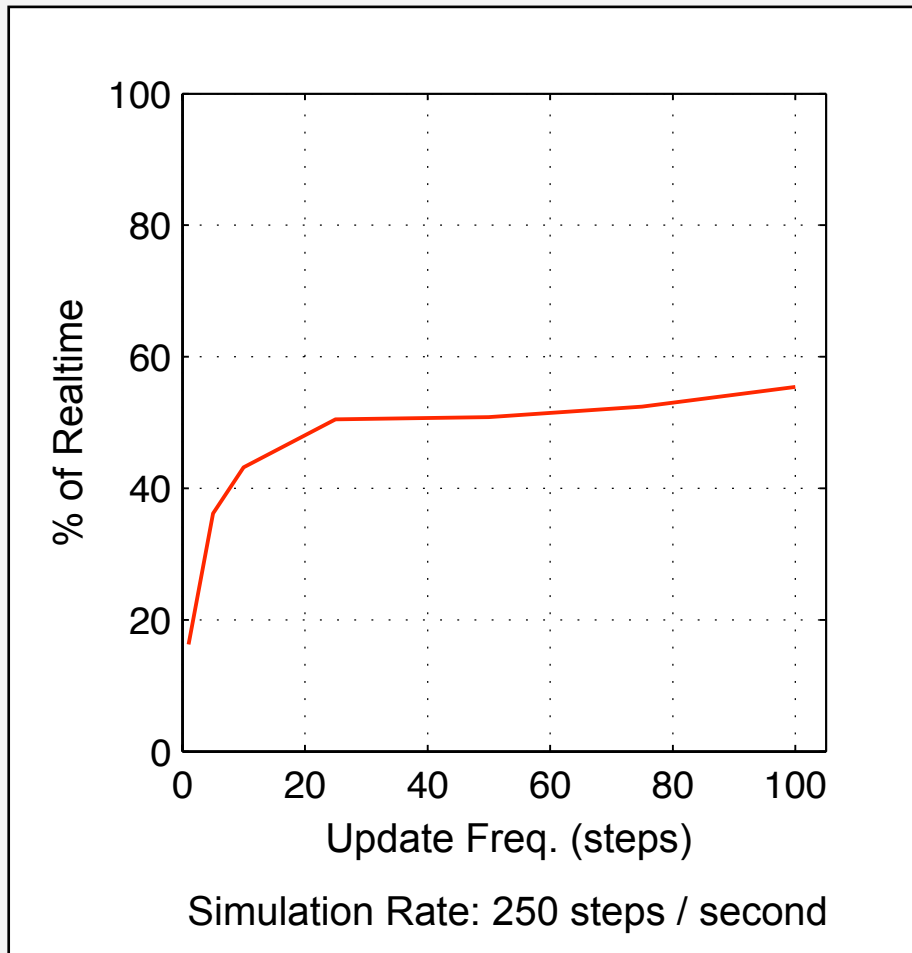
Results



- Gcomapex
- Gheading
- Gstepdir
- Gstepdist
- Gsteptime
- Wacceleration
- Wcomapex
- Wheading
- Whip
- Wleg
- Wstepdist
- Wsteptime
- Wterrain
- pushdir
- pushforce
- stepheight

Step Distance

Performance



18% of realtime performance

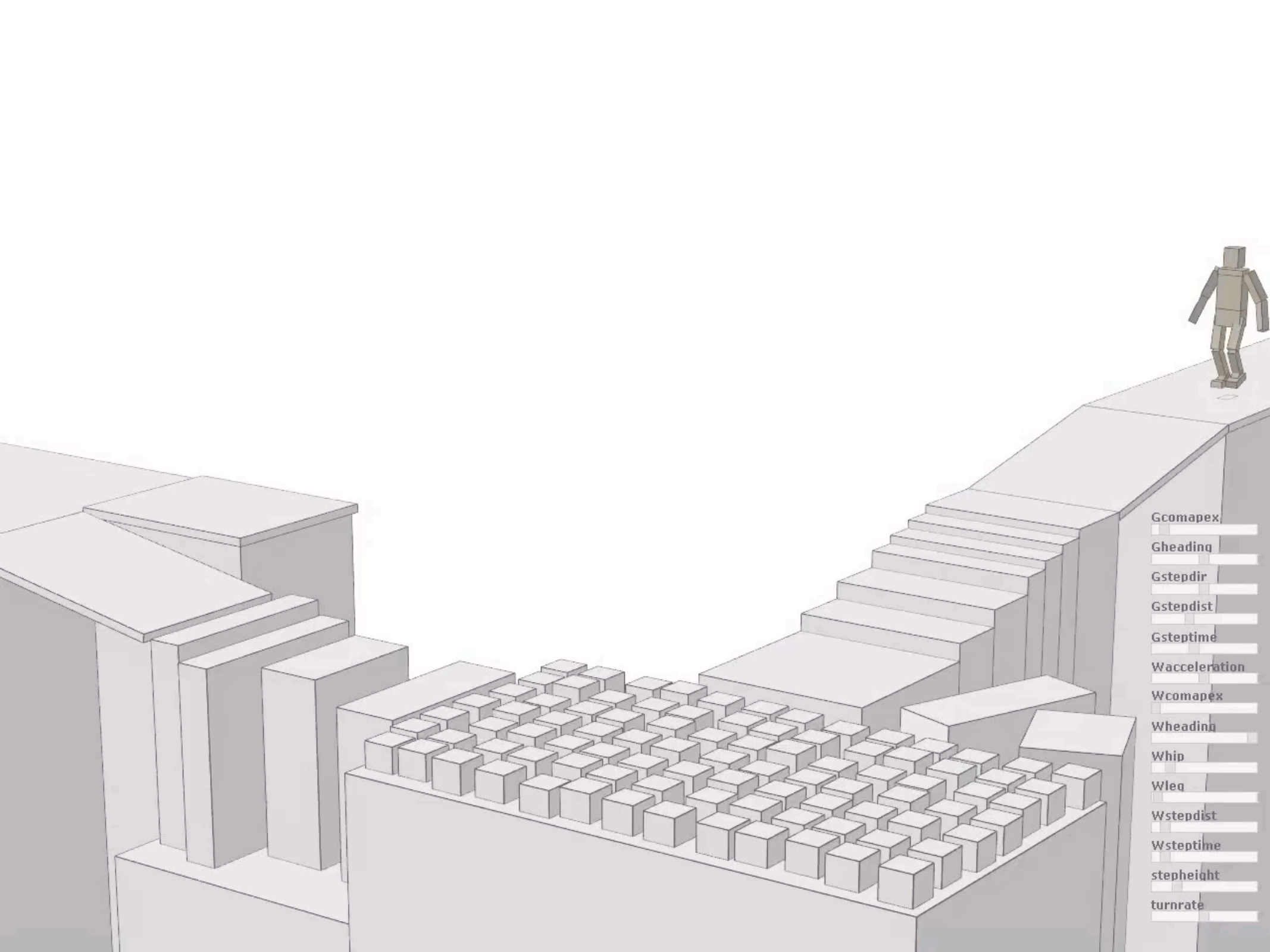
Don't need to replan every timestep

- Replanning every 1/10 sec gives 50% realtime performance
- Controller still robust (without perturbations)

Pushes

Uneven Terrain

Projectile Avoidance



- Gcomapex
- Gheading
- Gstepdir
- Gstepdist
- Gsteptime
- Wacceleration
- Wcomapex
- Wheading
- Whip
- Wleq
- Wstepdist
- Wsteptime
- stepheight
- turnrate

Limitations

Motion style

Accuracy of the low-dimensional model

Performance

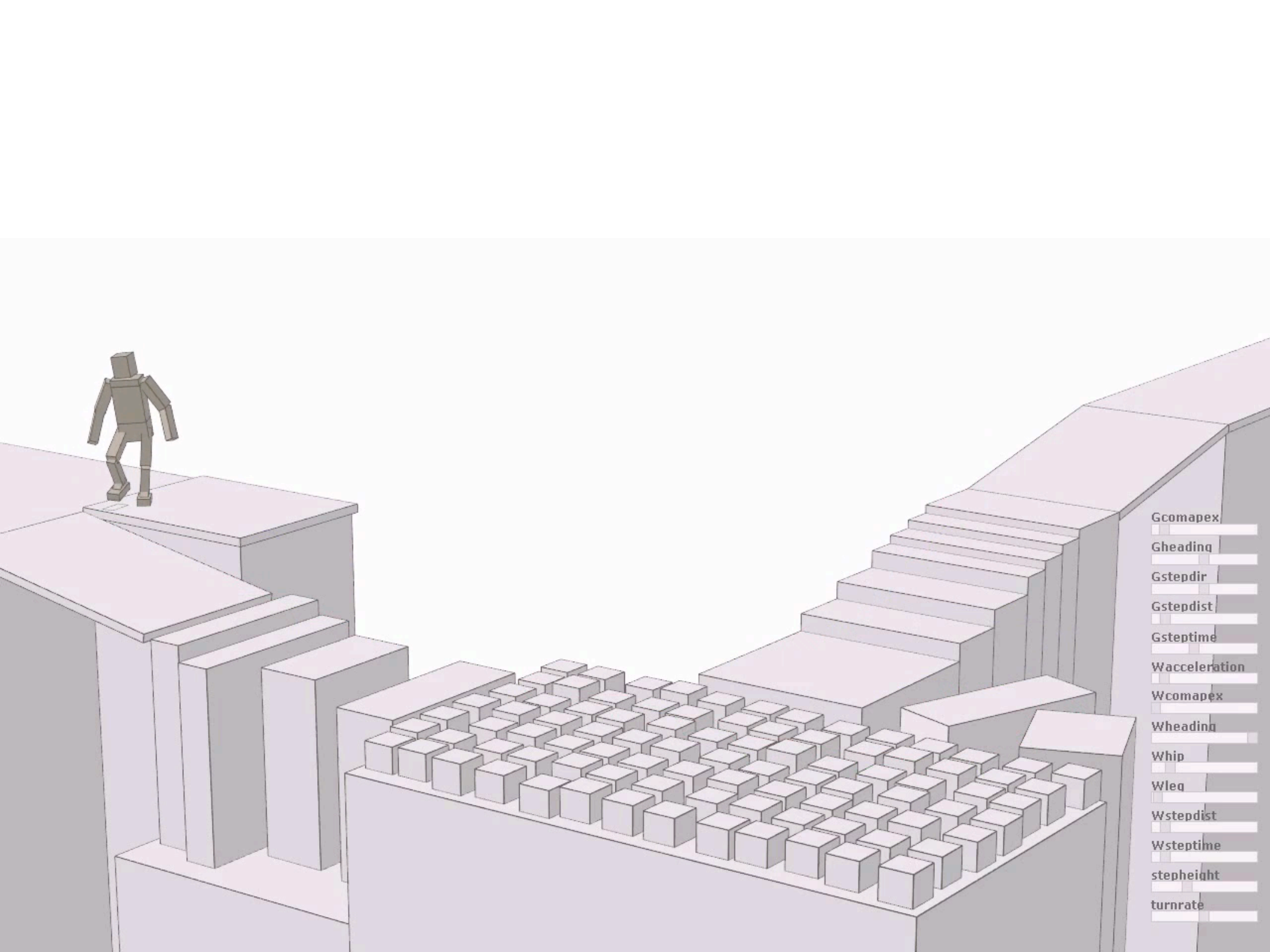
Conclusion

Controller for complex dynamic environments
with disturbances

Single parameterization for all motions

Flexible goal specification

No preprocessing or motion capture



- Gcomapex
- Gheading
- Gstepdir
- Gstepdist
- Gsteptime
- Wacceleration
- Wcomapex
- Wheading
- Whip
- Wleg
- Wstepdist
- Wsteptime
- stepheight
- turnrate