CMPT 888: Computer Animation

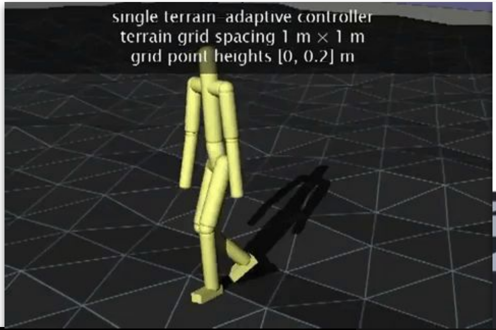
SIMBICON and GENBICON Physics-based Control Models for Locomotion.

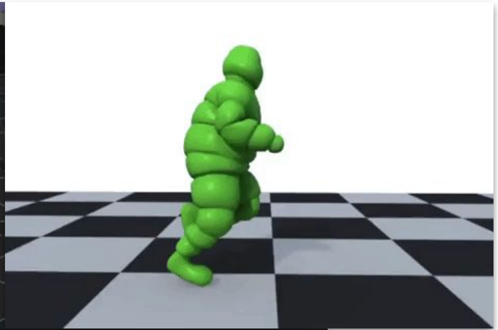
Presented By

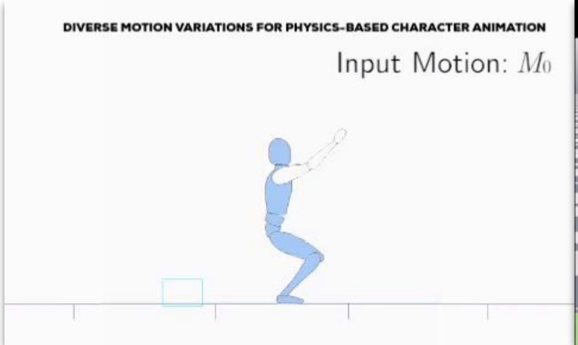
Anmol Sharma
Medical Image Analysis Lab
School of Computing Science
Simon Fraser University

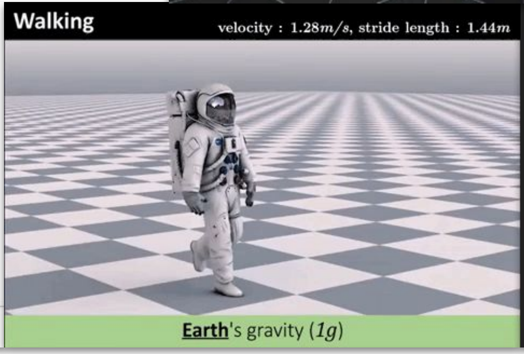
Locomotion

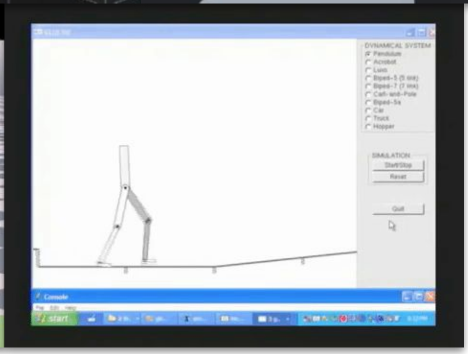
- What is locomotion?
- Why it’s difficult to model?
    - Unstable, underactuated,high-dimensional











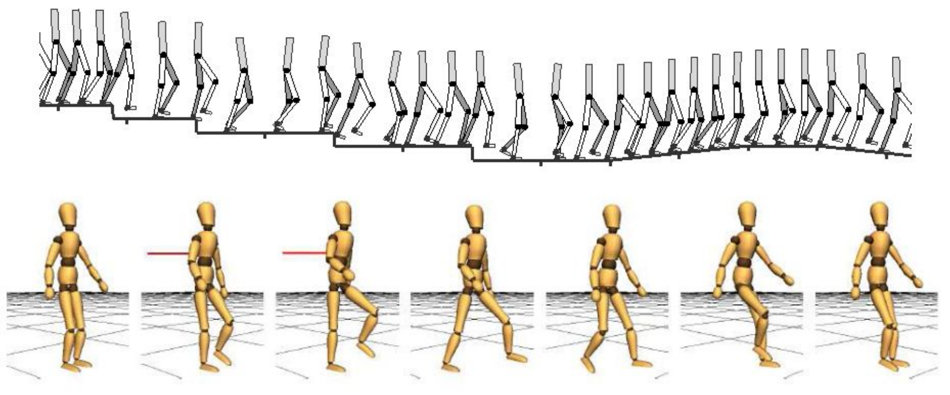
Libin Liu, KangKang Yin, Bin Wang, and Baining Guo. 2013. Simulation and Control of Skeleton-driven Soft Body Characters. ACM Trans. Graph. 32, 6 (SIGGRAPH Asia 2013), Article 215, 8 pages.
Y. h. Kim, T. Kwon, D. Song and Y. J. Kim, "Full-Body Animation of Human Locomotion in Reduced Gravity Using Physics-Based Control," in IEEE Computer Graphics and Applications, vol. 37, no. 6, pp. 28-39, November/December 2017. doi: 10.1109/MCG.2017.4031066
KangKang Yin, Kevin Loken, and Michiel van de Panne. 2007. SIMBICON: simple biped locomotion control. In ACM SIGGRAPH 2007 papers (SIGGRAPH '07). ACM, New York, NY, USA, Article 105 . DOI: https://doi.org/10.1145/1275808.1276509
Wu, Jia-chi, and Zoran Popović. "Terrain-adaptive bipedal locomotion control." ACM Transactions on Graphics (TOG)29.4 (2010): 72.

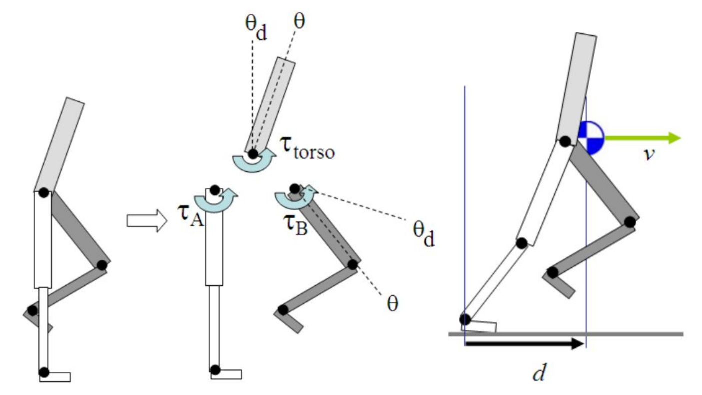
Moravec's Paradox

> “
> Moravec's paradox is the discovery by artificial intelligence and robotics researchers that, contrary to traditional assumptions, <u>high-level reasoning requires very little computation</u>, **but low-level sensorimotor skills require enormous computational resources.**
> ”

SIMBICON: Simple Biped Locomotion Control

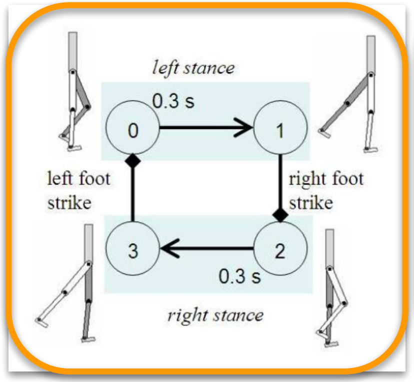
KangKang Yin, Kevin Loken, Michiel van de Panne
University of British Columbia

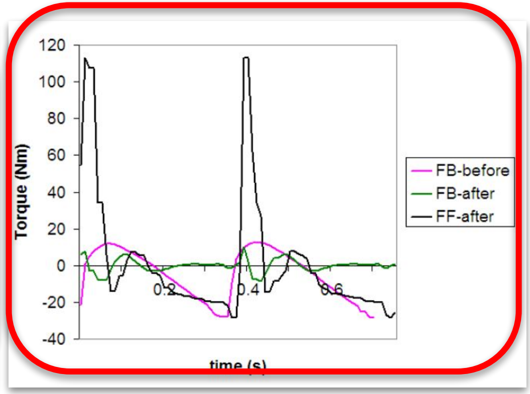


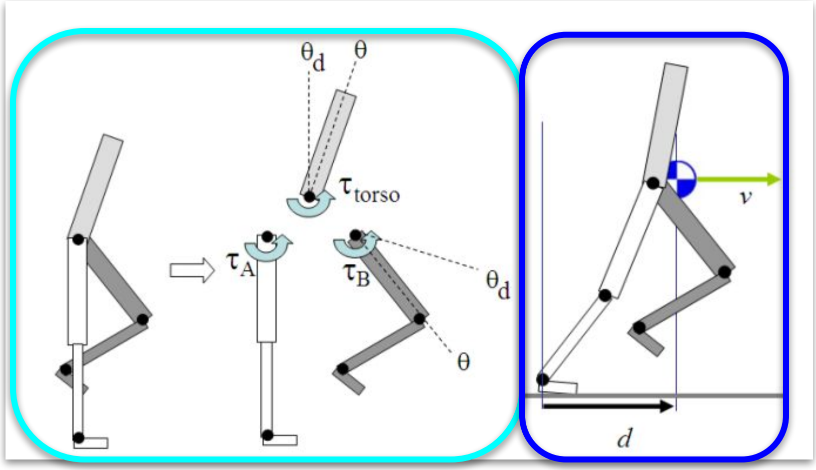


Overview

- Challenges?
- Proposed Control Model
  - Gaits as Finite State Machines
  - Torso and Swing-hip Control
  - Balance Feedback
  - Feedback Error Learning
- Results
- Conclusion
- Summary



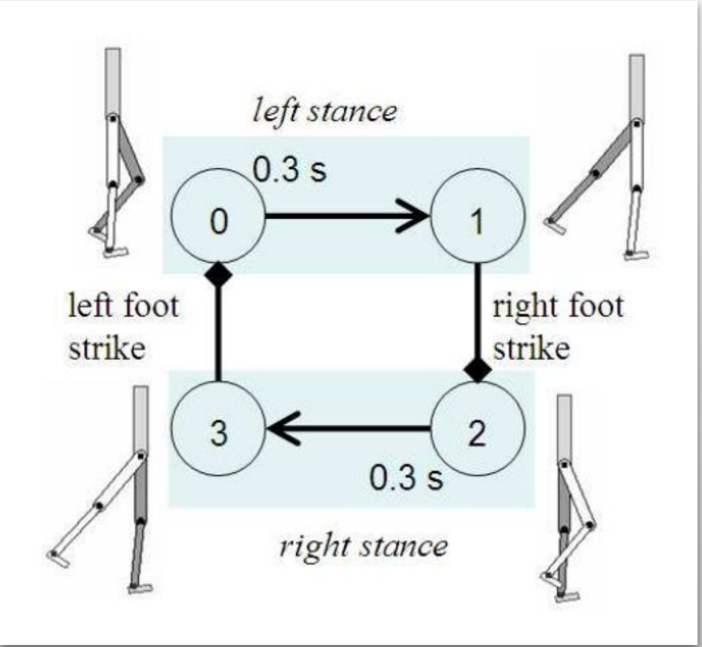




Challenges

- Previous approaches have been very complex, in terms of the number of modelling decision that were taken.
    - How to make the approach simpler, which in turn would mean easy implementation, and manageability.
- How to make control models generalizable to different simulated biped gaits?
    - Can the same model walk backwards? Or Scissor-hop?
    - Can it transition between different skills? (Useful in gaming where character walk/runs/climbs in a continuous motion).
- How to induce the notion of balance to the biped?
- How to ensure only physically-viable torques are generated in the simulation?
    - The leg/arm doesn’t fly off with an arbitrary torque value?

Gaits as Finite State Machines

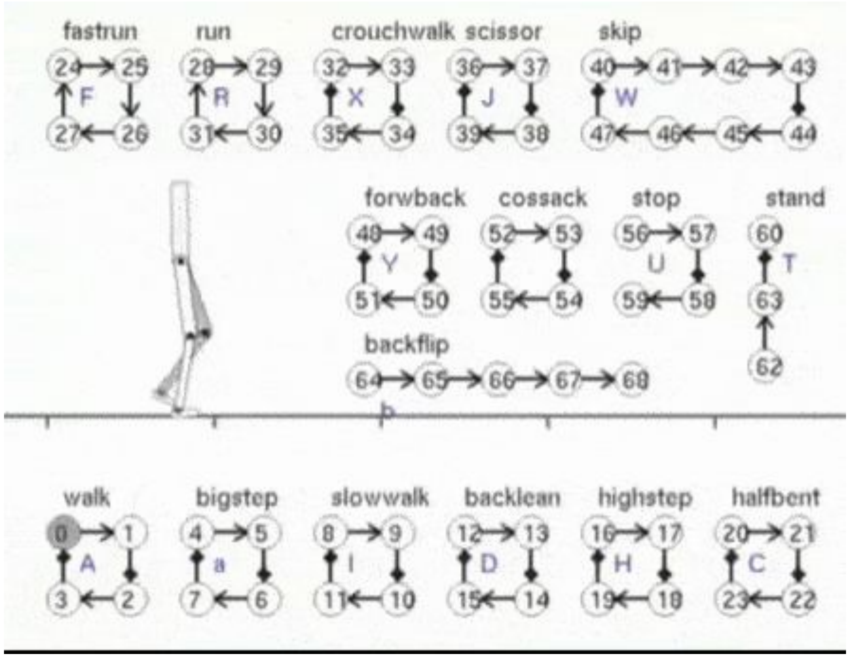


Walking Gaits as Finite State Machines

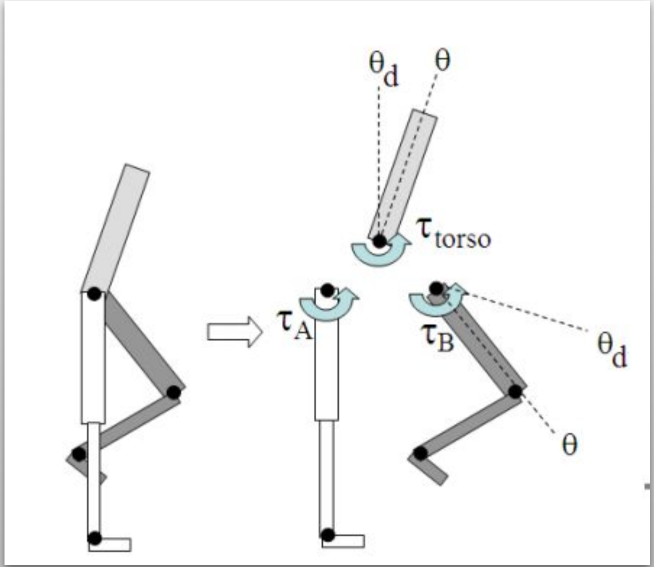
- Encode each walking gait as a finite state machine.
- Each state has its own target pose for internal joint angles.
- Transitions happen between states after an elapsed time, or a foot contact.
- Torques are calculated by PD-controller using:

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

And applied to target joints.

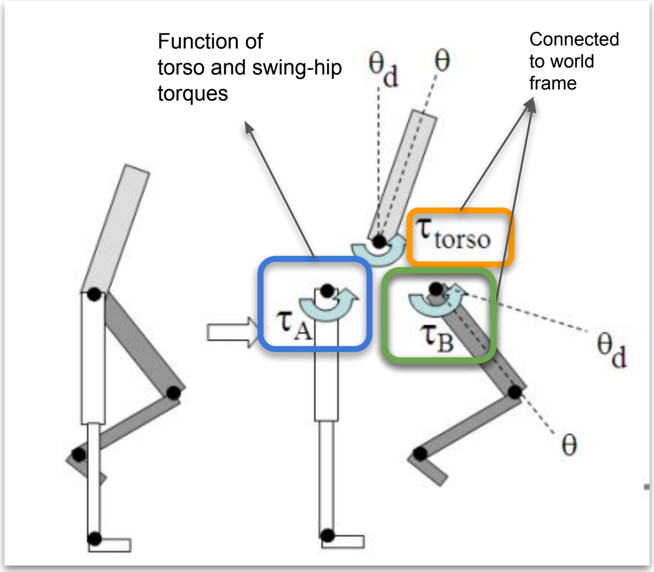


Torso and Swing Hip Control

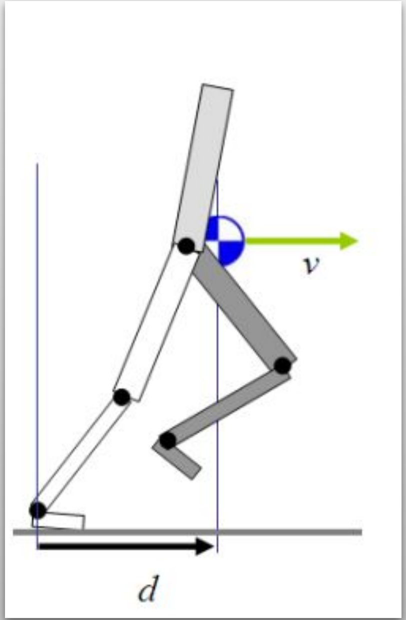


Torso and Swing-Hip Control

- Swing-Hip and torso handled separately.
  - Orientation of torso controlled in world frame managed using PD controller-computed τ_{torso}
- Decouple swing-foot position from torso pitch angle.
  - Compute τ_B using PD-controller.
- Ensure virtual torques are realizable using only internal torques (physically realizable)
  - Decouple stance hip torque as: $\tau_A = -\tau_{torso} - \tau_B$



Balance Feedback



Balance Feedback

- Induce notion of balance in control model.
- Employ a simple feedback loop to handle it.
- Why have two gain parameters c_v and c_d?
  - Combination of (d,v) provides complete information about current position in gait cycle.
  - Example: When v = 0 but d_a=+10cm and d_b=-10cm.
  - In the case where only v is used to balance, no feedback will be triggered.

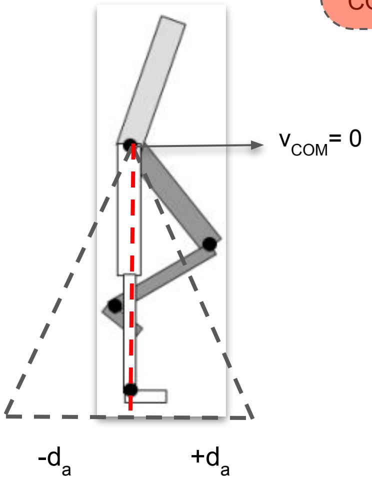
$$\boxed{\theta_d} = \boxed{\theta_{d0}} + \boxed{c_d \boxed{d}} + c_v \boxed{v}$$

Target angle used for PD control

Default target angle in FSM

Distance between stance ankle to COM

Velocity of COM



No response will be triggered!

Balance Feedback

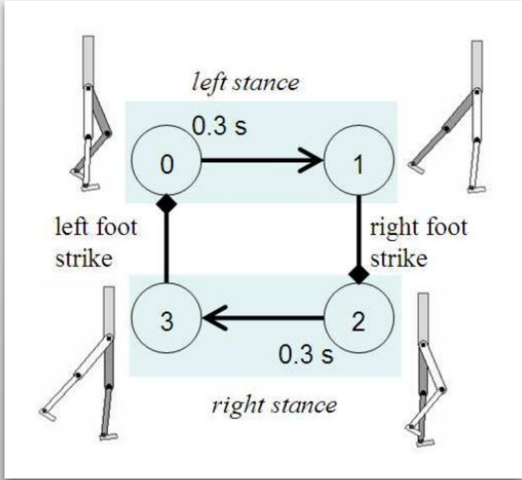
- Same model can be extended to 3D using a general form of the previous equation.
- Equation is applied in both sagittal and coronal plane.
- General form of equation for all joints in the biped is:

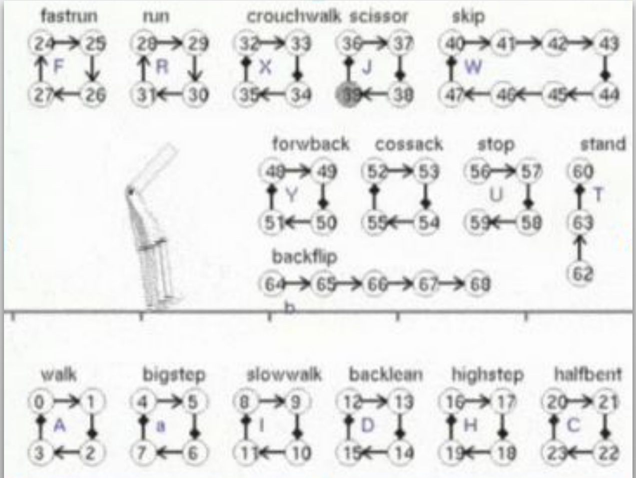
$$\theta_{\mathbf{d}} = \theta_{\mathbf{d0}} + \boxed{\mathbf{F}} \begin{bmatrix} d \\ v \end{bmatrix}$$

Nx2 matrix
with feedback
coefficients

$$\begin{bmatrix} c_{d1} & c_{v1} \\ c_{d2} & c_{v2} \\ c_{d3} & c_{v3} \\ .. & .. \end{bmatrix}$$

Controller Design

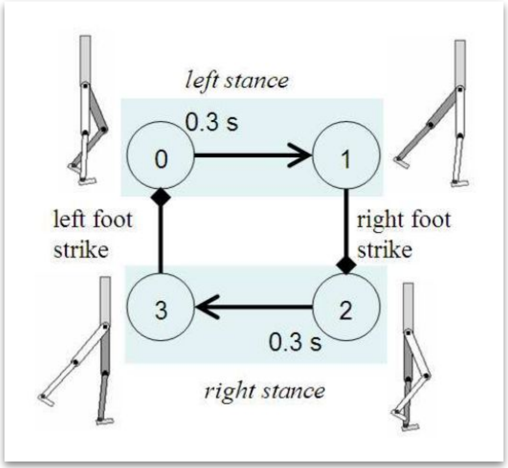




Manual Controller Design

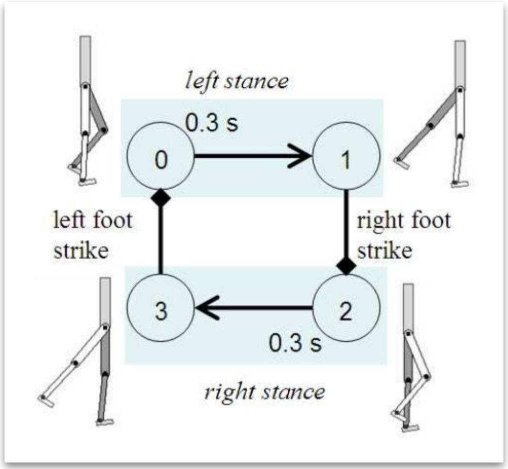
- To design controllers, many different control parameters need to be specified.
- Control parameters are:
  - Number of states and state transition parameters.
  - Balance feedback gains (c_v and c_d) for each joint.
  - Target poses for each state.
  - Initial state of the controller.
  - Joint limits, torque limits, and PD-controller gains.

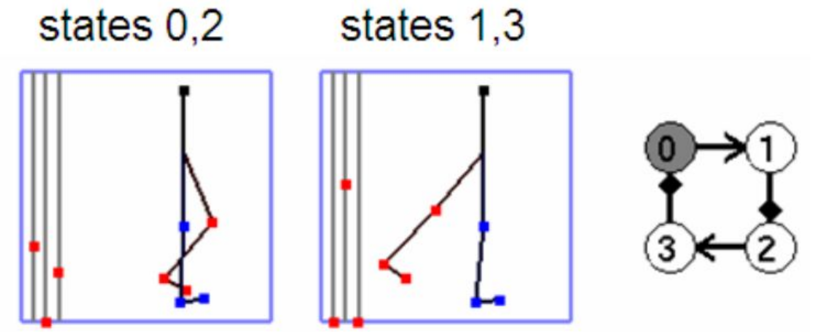
Fixed



Manual Controller Design

- Walking gaits are modeled with **4 states.**
  - Consists of two symmetric walking steps.
  - Each step has two states:
    - Lift swing foot up and forward.
    - Drive swing foot to ground until contact is made.
  - Switching between gaits:
    - Transition from state n of first gait to state n+1 of second gait.
- GUI is used to design new controllers.
  - Sliders to change:
    - State duration, c_v and c_d.
    - WYSIWYG type editor.
  - Handle points on joints to specify target poses.
    - Torso and swing femur are in world coordinates.
    - Remaining are defined w.r.t their parent's coordinate frame.





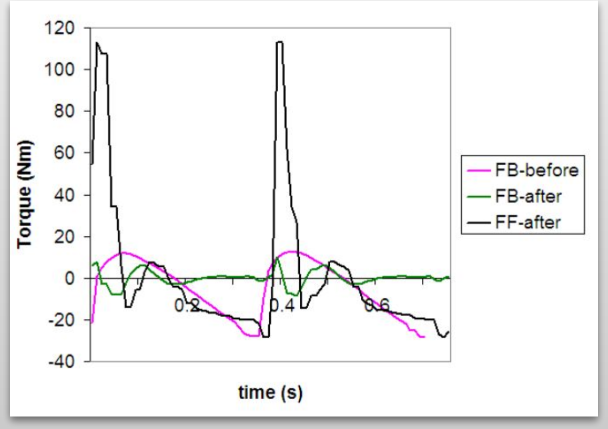
Controllers from Motion Capture Data

- Alternative to manual design of controllers.
- Import complex kinematic motion from MOCAP data.
  - Original MOCAP data cannot handle uncertainties in the scene.
- Given 3-7 cycles of MOCAP data:
  - Apply Fourier analysis to a joint, extract time period ***T*** of cycle, using frequency ***o*** of the Fourier analysis.
  - Reconstruct motion using largest fourier coefficients.
  - This filters the data to smooth periodic function (***theta***) , which is a representation of the average motion from all cycles.
- ***theta*** serves as target trajectory in place of target poses used in manually-designed controllers.
  - PD controller is changed to: $\tau = k_p(\theta_d - \theta) - k_d(\dot{\theta} - \dot{\theta}_d)$
- Balance feedback is applied to swing-hip and stance ankle (in slow walk).

Controllers from Motion Capture Data

- Controllers won't perfectly mimic the motion capture reference, due to the following reasons:
    - Original MOCAP is error prone, both in acquisition and processing.
    - Simulation biped parameters may not match MOCAP actor's parameters.
        - Link dimensions, joint placements, mass and inertial parameters, and joint gains.
    - No stance hip tracking (in order to provide balance feedback and allow physically realizable torques).
- Tracking control requires high-gain PD controllers.

Feedback Error Learning



Current Issues

- Unnecessary bobbing motion in the torso due to a “reactive” response to motion of hip, rather than “anticipatory”.
- Following trajectories in controller designed through MOCAP requires high-gain PD-controllers.
    - Also leads to “stiff” motions of the physical biped.
- How can we allow:
    - Low impedance (low gain) control when environment is predictable.
    - High impedance (high gain) control when it is not.
- Feedback Error Learning (FEL) is employed to address these issues.

Feedback Error Learning (FEL)

- FEL was first proposed by Kawato et al. 1987.
  - Proposed from a biological perspective, to establish computational model of the cerebellum.
  - Use it to learn motor control with the internal models in the central nervous system (CNS).
- General form.
- In SIMBICON, a different formulation is used.

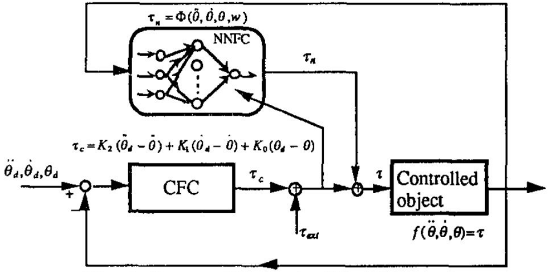


Fig. 1 Inverse Dynamics Model Learning

$$\boxed{\tau = f(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}})}$$

position

velocity

commanded acceleration

$$\boxed{\tau = f(\phi) \quad \Big| \quad \phi = t / \hat{T}}$$

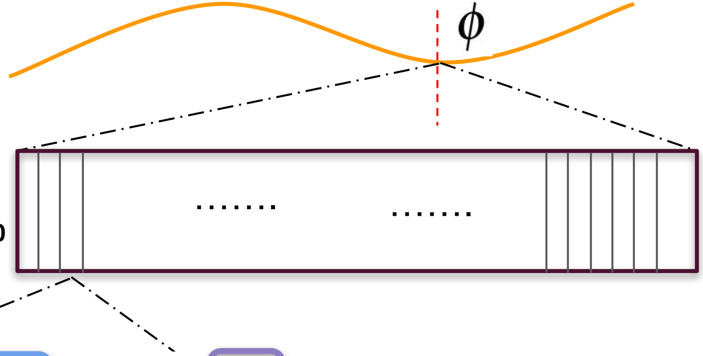
elapse
d time

Estimate of motion period

Kawato, M. (1987). A hierarchical neural-network model for control and learning of voluntary movement. Biological Cybernetics, 57, 169–185.

Feedback Error Learning (FEL) in SIMBICON

$$\tau = f(\phi) \;\Big|\; \phi = t/\hat{T}$$



Bins
N = 20 - 1000

Low-pass filter $v'_{ff} = (1-\alpha)\boxed{v_{ff}} + \alpha(v_{ff} + \boxed{v_{fb}})$

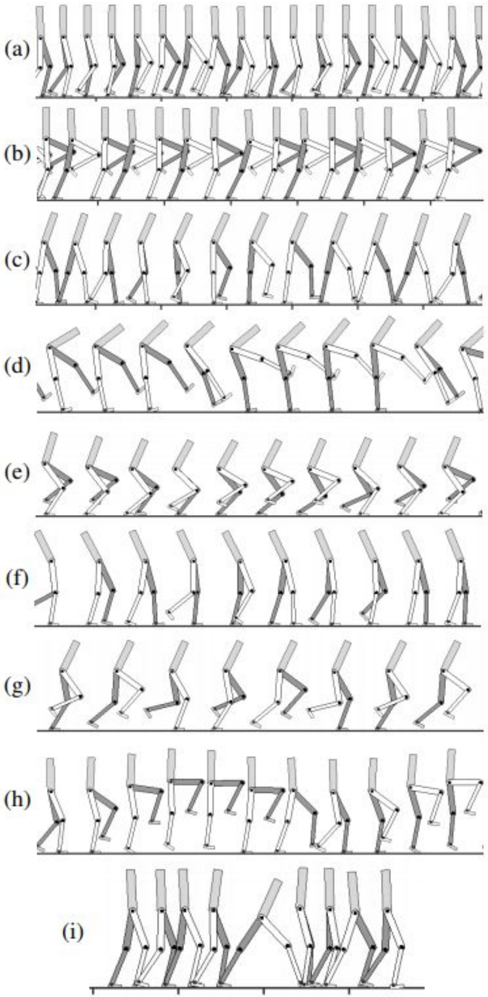
feed-forward torque

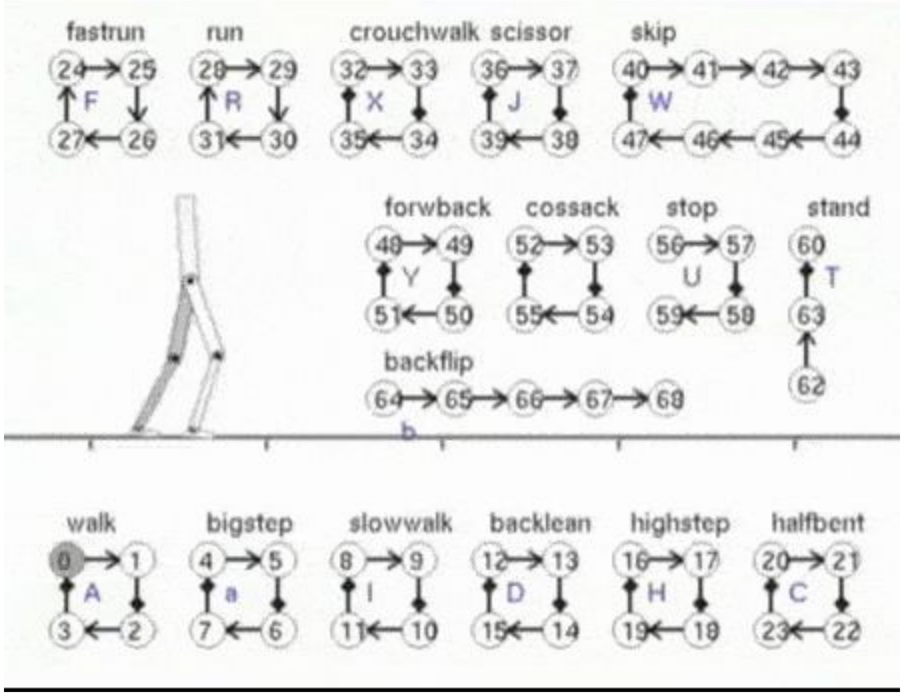
feed-back torque

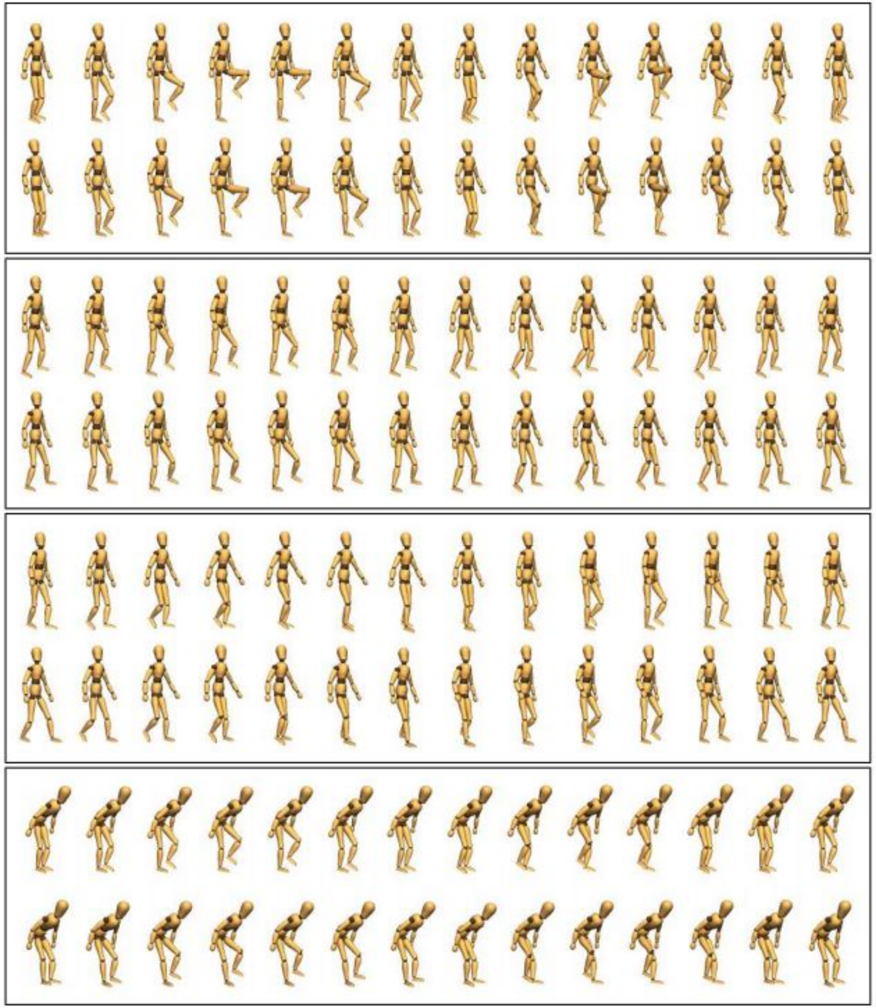
Final Torque **=** $v_{ff} + v_{fb}$

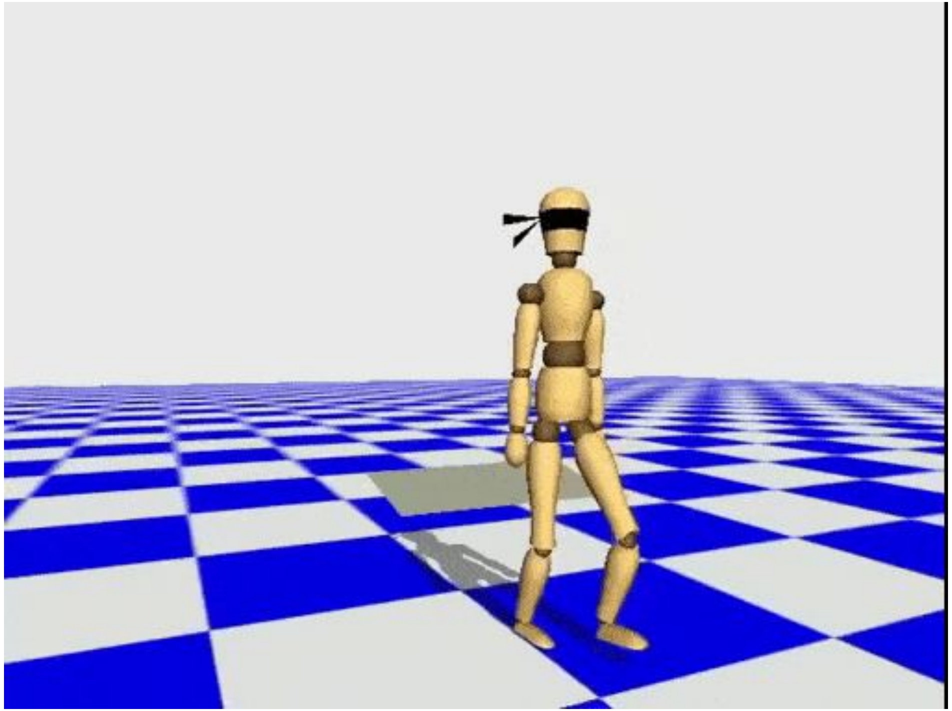
Gomi, Hiroaki, and Mitsuo Kawato. "Learning control for a closed loop system using feedback-error-learning." *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on*. IEEE, 1990.

Results


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)


fastrun
24 25 27 26
F
run
28 29 31 30
R
crouchwalk
32 33 35 34
X
scissor
36 37 39 38
J
skip
40 41 42 43 47 46 45 44
W
forwback
48 49 51 50
Y
cossack
52 53 55 54
stop
56 57 59 58
U
stand
60 63 62
T
backflip
64 65 66 67 68
b
walk
0 1 3 2
A
bigstep
4 5 7 6
a
slowwalk
8 9 11 10
I
backlean
12 13 15 14
D
highstep
16 17 19 18
H
halfbent
20 21 23 22
C





Limitations

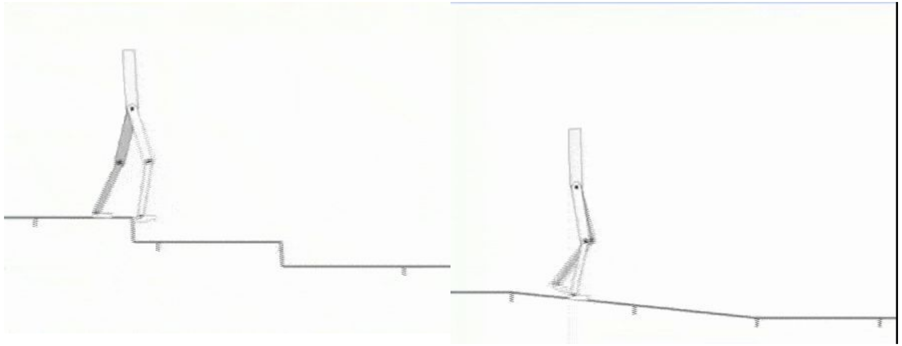
Limitations

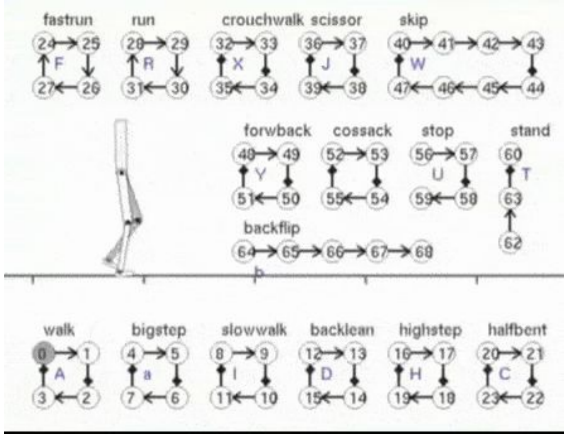
- Controller design from motion capture data not fully automated.
  - Balance feedback parameters have to be tuned sometimes.
- Inability to move between arbitrary states.
  - The basis of attraction of some states do not overlap, hence transitioning between them is not possible.
- Manual gait designs are not optimized for energy efficiency.
- Models for CNS reaction times absent.
  - Leads to overly stable motions, where a human might obviously fail.
- Convergence proof for FEL not discussed.
  - Although the authors did not have any issues with the choice of learning rate.
- Cannot generalize to biped of varied proportions

Summary

Summary

- SIMBICON tries to address these challenges through a simple physics-based controller.
- The controller generalizes between different walking skills, without any modification.
    - Also generalizes to 3D from 2D.
- Induces notion of balance through a simple feedback loop.
- Produced only physically-valid torques by decoupling stance leg hip torque.
- Reduced PD-controller gains through Feedback Error Learning.



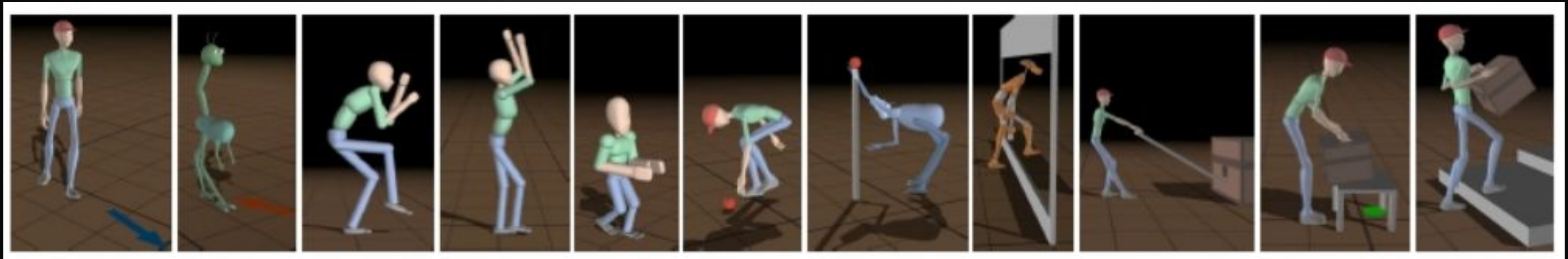


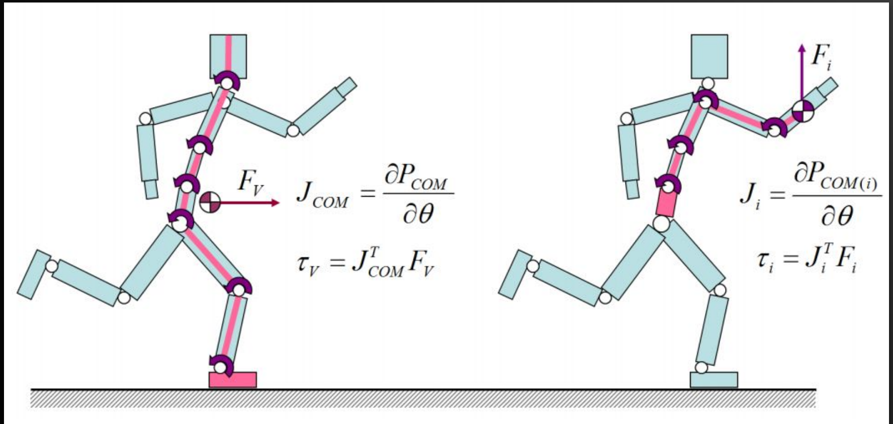
Questions?

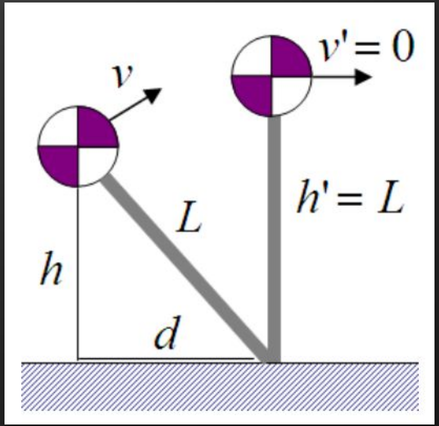
Thank you!

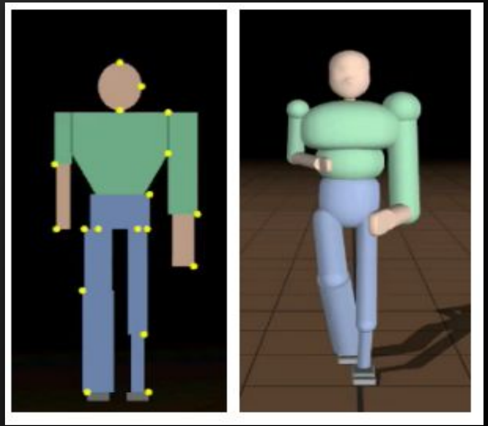
Generalized Biped Walking Control

Stelian Coros, Philippe Beaudoin, Michiel van de Panne
University of British Columbia



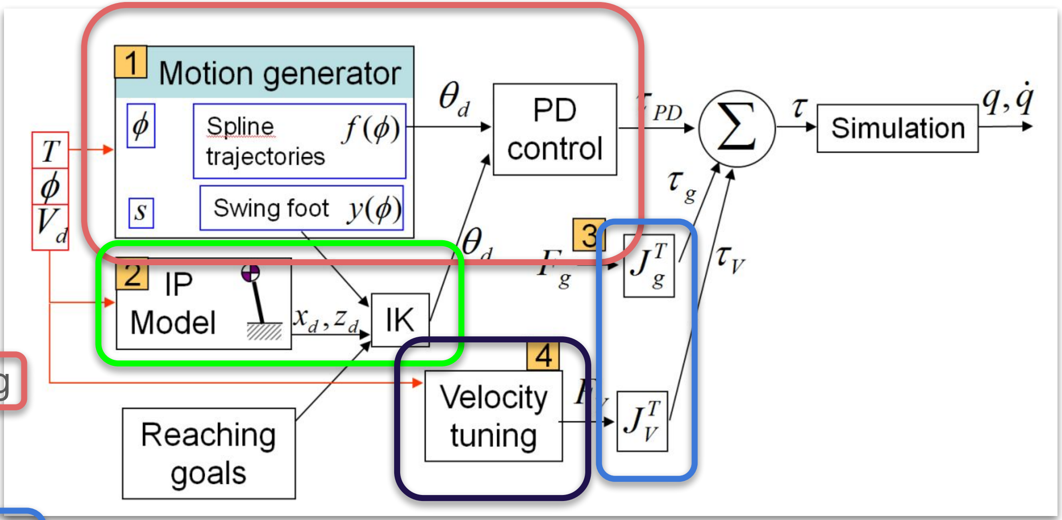






Outline

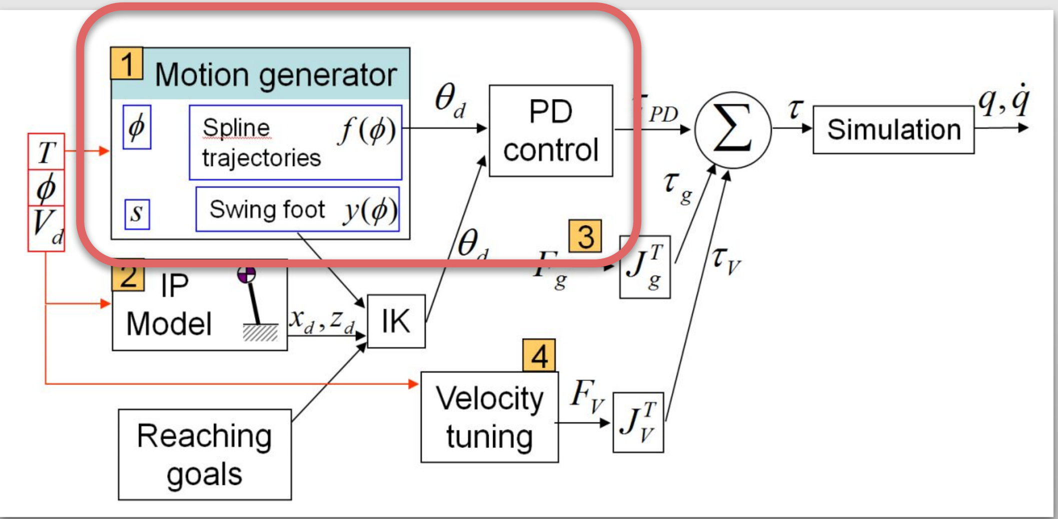
- GENBICON: Generalizable physics based control model for walking.
- Current Issues
- Control Model
  - Motion Generator with PD Tracking
  - Inverted Pendulum and Inverse Kinematics Model
  - Continuous Balance Adjustment and Gravity Compensation
  - Velocity Tuning
- Results



Current Issues

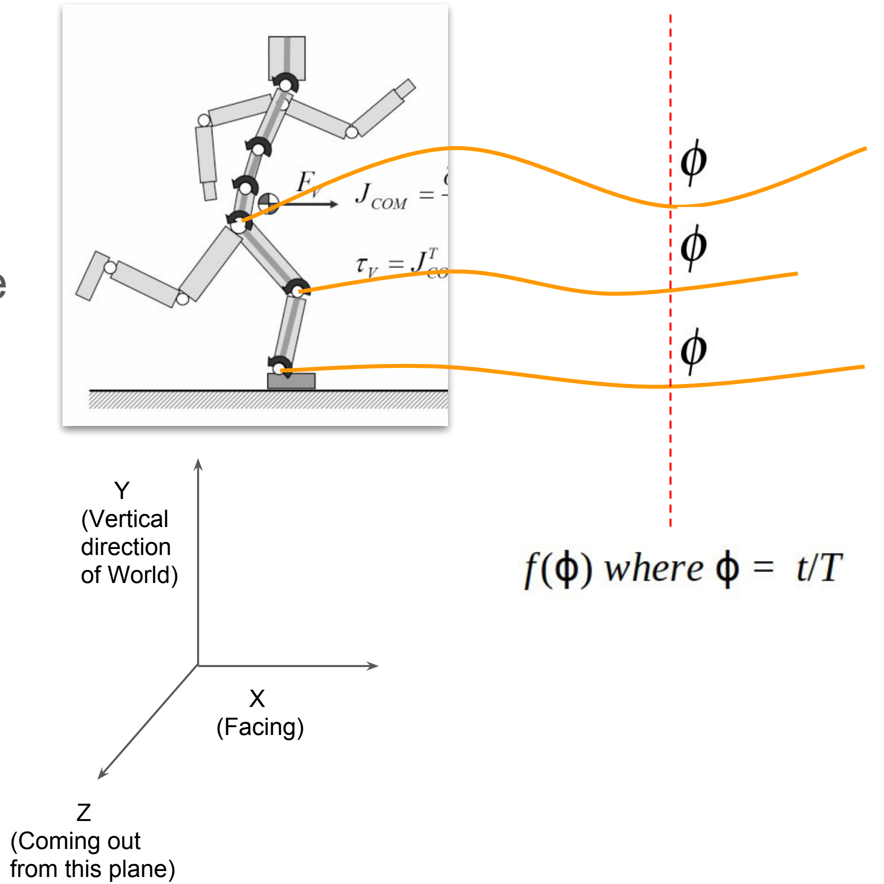
- “Weight” of the motion
  - How do you author motions that the byproduct is induced (and graceful?).
- Balance awareness.
  - Induce balance awareness to the model
- Authoring new motions.
  - Allow animators to easily author motions.
- Generalizable motions across:
  - Gait parameters
  - Character proportions
  - Motion styles
  - Walking skills
- Compute constraints.

Motion Generator

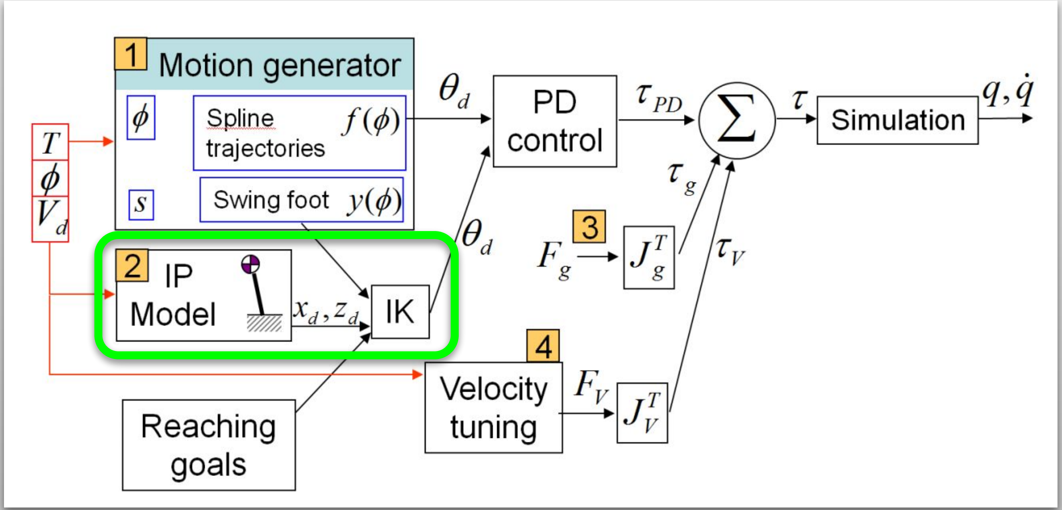


Motion Generator

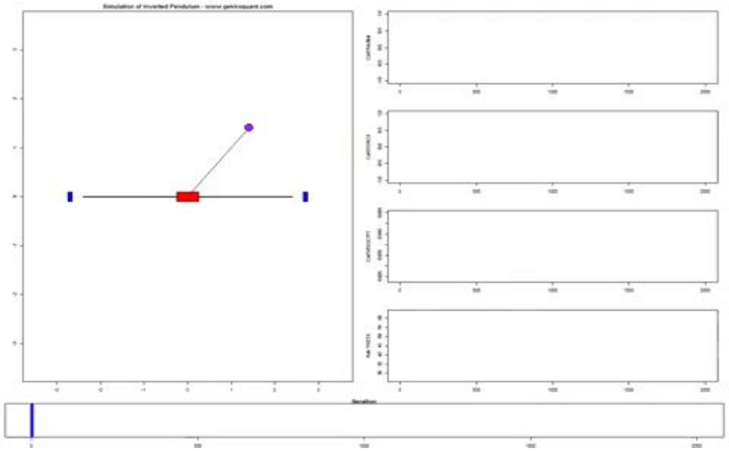
- Produce desired trajectories to track and create desired motion styles.
- Trajectories are modeled as spline functions.
- Angles can be relative to parent joints, or to character coordinate frame.
- No stance hip target trajectories (like SIMBICON).
- Desired joint angles are given to PD controller which generates tracking torques.

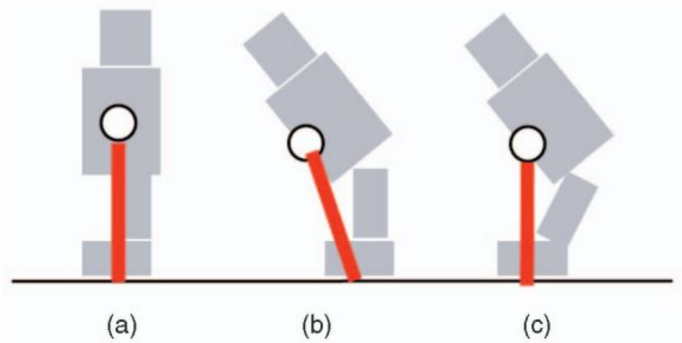


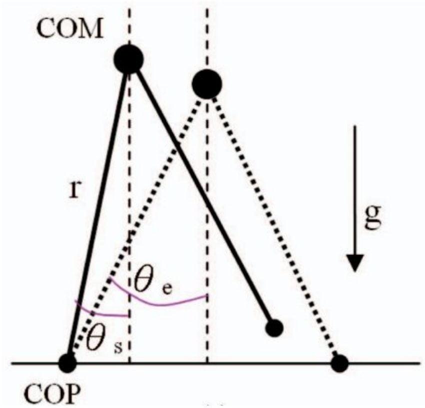
Inverted Pendulum and Inverse Kinematics



Inverted Pendulum





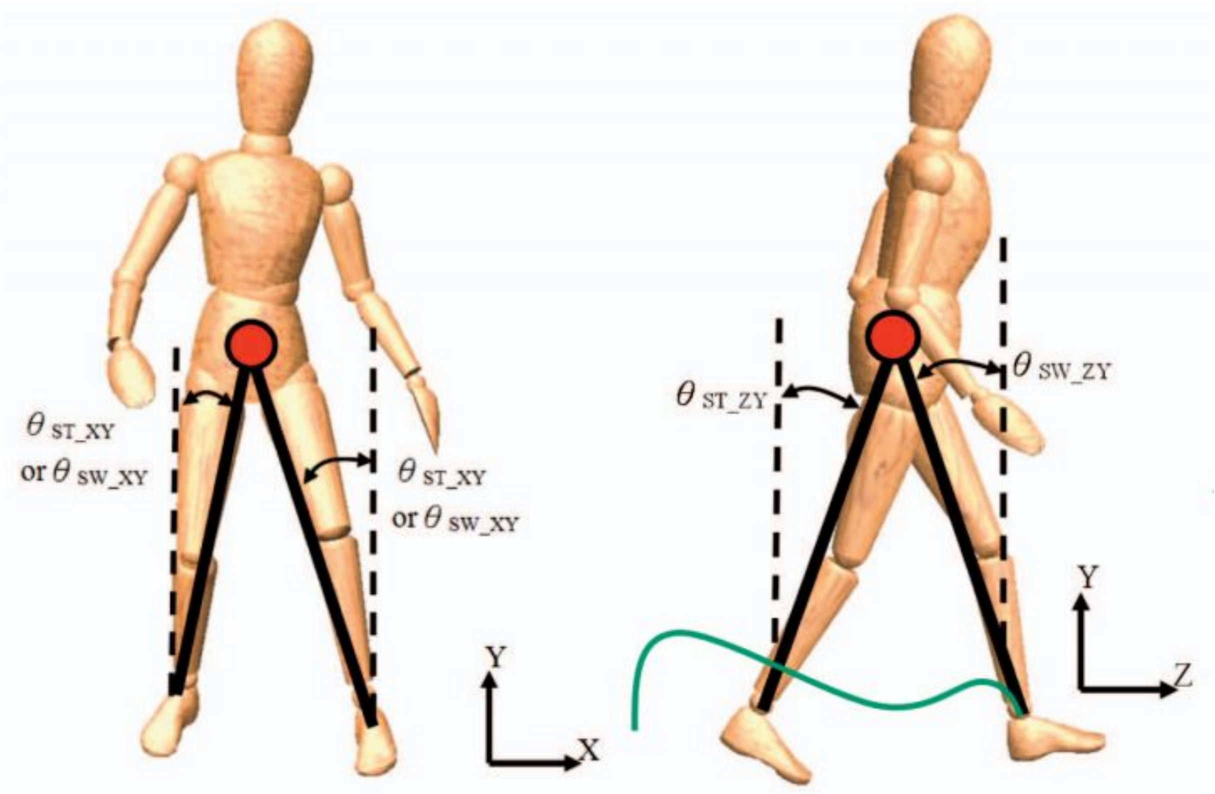


$$\frac{1}{2}I\omega_s^2 + \int_{\theta_s}^{\theta_e} \tau(\theta)d\theta = \frac{1}{2}I\omega_e^2$$

$$\frac{1}{2}I\omega_s^2 - mgr(\cos\theta_e - \cos\theta_s) = \frac{1}{2}I\left(\frac{d\theta_e}{dt}\right)^2$$

Tsai, Yao-Yang, et al. "Real-time physics-based 3d biped character animation using an inverted pendulum model." *IEEE transactions on visualization and computer graphics* 16.2 (2010): 325-337.

Inverted Pendulum



Inverted Pendulum Foot Placement

- Computing where to place the swing foot (x_d, z_d)
  - Small steps/Large steps.
  - Recovering from external push/pull.
- Assumes constant leg length.
- Desired velocity is computed by:

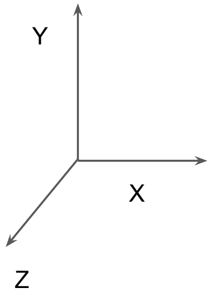
$$\frac{1}{2}mv^2 + mgh = \frac{1}{2}mv'^2 + mgh'$$

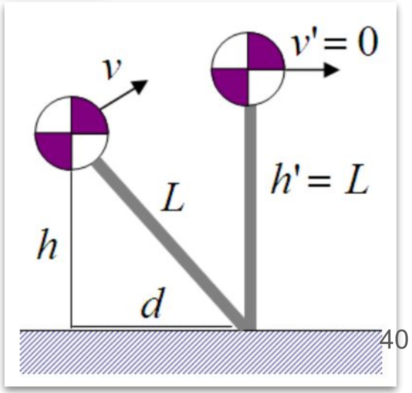
$$v' = 0 \text{ and } h' = L = \sqrt{h^2 + d^2}$$

$$d = v\sqrt{h/g + v^2/(4g^2)}$$

$$d' = d - \alpha \boxed{V_d}$$

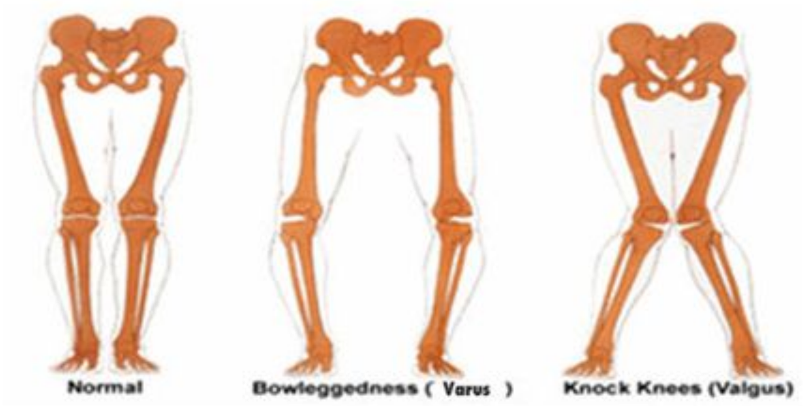
Desired Velocity



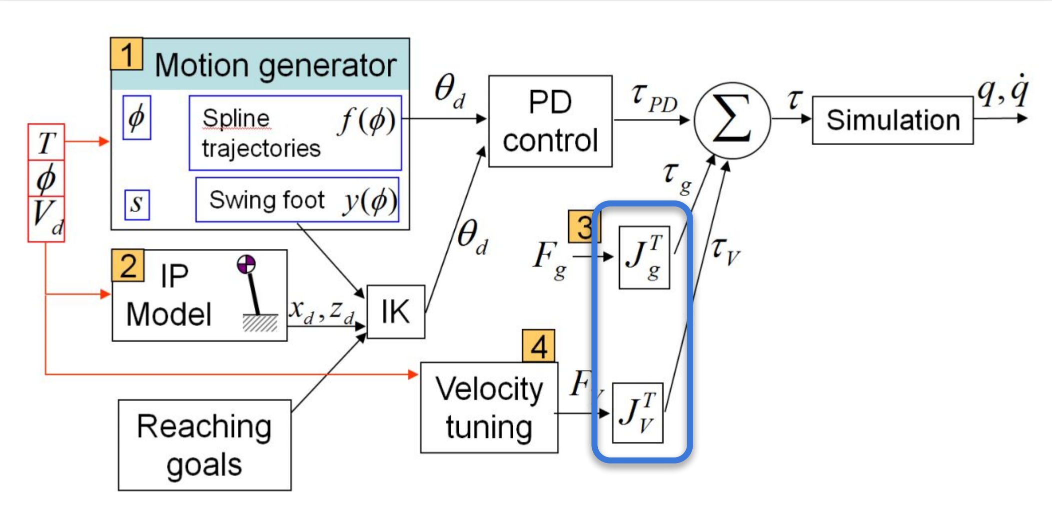


Inverse Kinematics

- Swing leg motion is synthesized as desired trajectory of ankle.
  - Height ankle w.r.t ground is modeled by the function: $y(\phi)$
- Once the ankle motion is generated, **Inverse Kinematics** is used to compute target joint angles for swing hip and knee.
  - One DOF exposed as "Twist angle parameter" which allows knock-kneed and bow legged motion.



Gravity Compensation

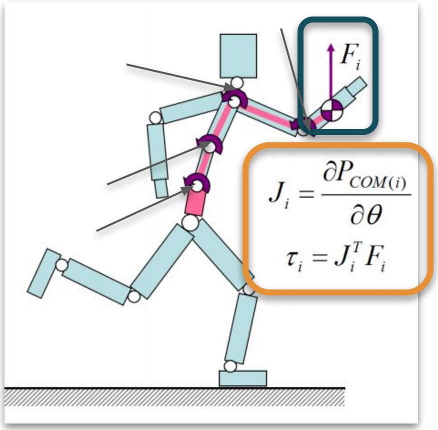


Gravity Compensation

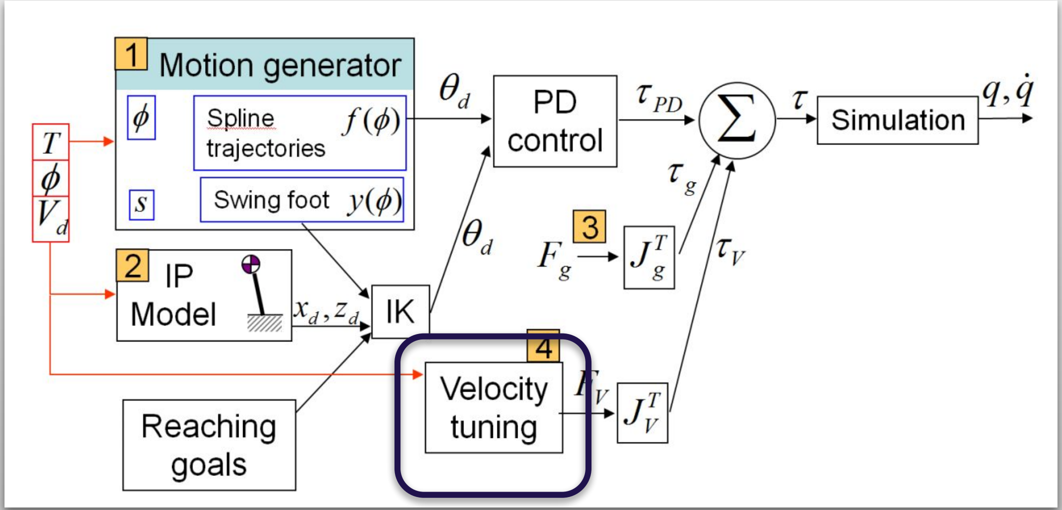
- Gravity compensation (GC) allow low-gain PD controllers to be used for each link (joint) of the simulated subject.
- GENBICON applies GC using a Jacobian Transpose method.
- Torques are computed for each joint and applied in the opposite direction (hence the negative sign).

Gravity Compensation

- Gravity compensation (GC) allow low-gain PD controllers to be used for each link (joint) of the simulated subject.
- GENBICON applies GC using a Jacobian Transpose method.
- Torques are computed for each joint and applied in the opposite direction (hence the negative sign).

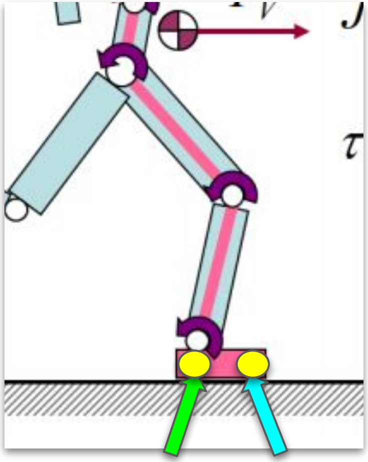


Velocity Tuning



Velocity Tuning

- **Foot placement**: Provides robustness for the gait
  - **However it's enacted only once per step.**
  - **Can't use manipulation of COP or GRPs to maintain balance.**
- Manipulating COP and GRPs allow finer level control and balancing.
- Another method is to use "**Virtual Forces**" on the COM, as described in [1].



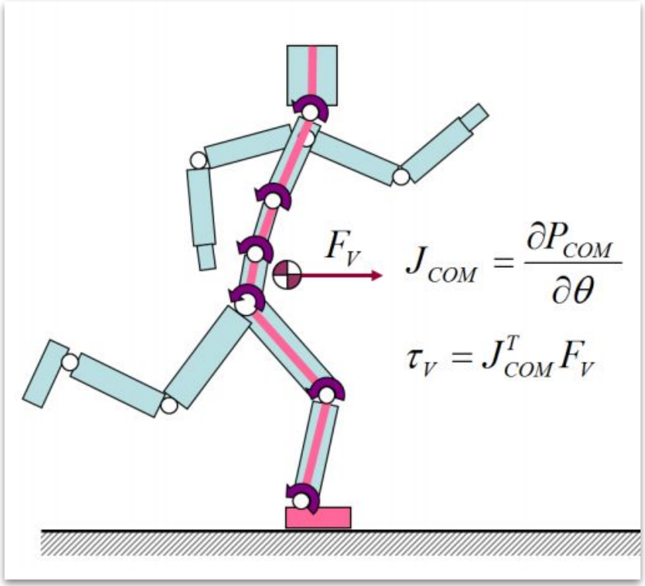
COP towards back: Accelerate forward

COP towards toes: Slow down forward progression

[1] Pratt, Jerry, et al. "Virtual model control: An intuitive approach for bipedal locomotion." *The International Journal of Robotics Research* 20.2 (2001): 129-143.

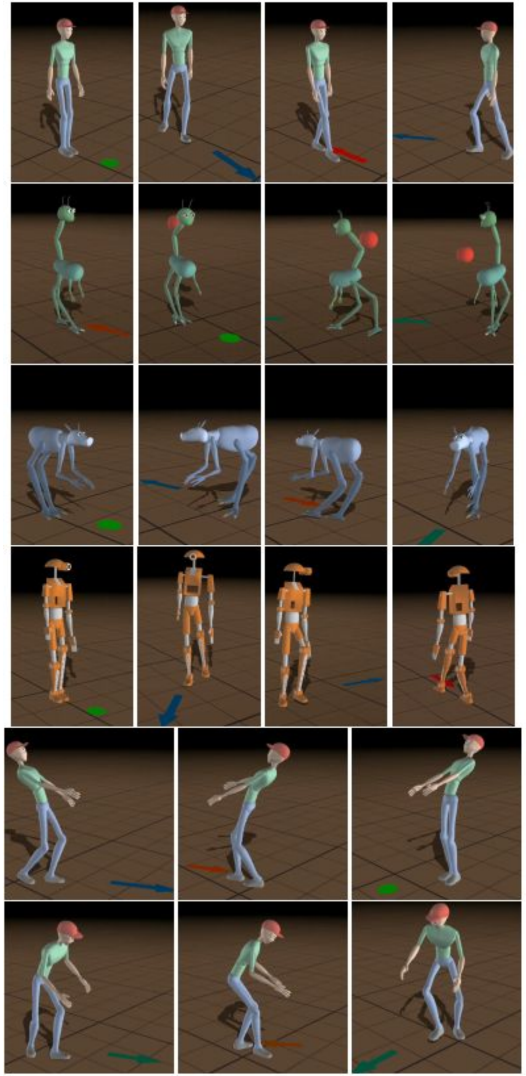
Velocity Tuning

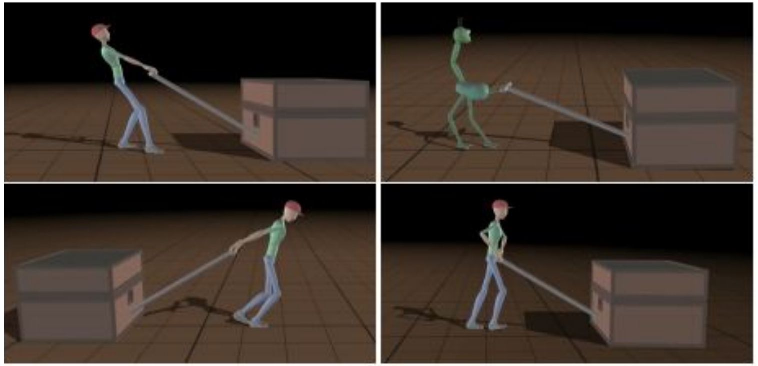
- **GENBICON uses Virtual velocity-tuning force to finely control COM velocity**.
  - Similar in essence to manipulating GRPs and COP position.
- Algorithm:
  - Compute COM velocity V
  - In Sagittal Plane: Compute $F_v = k_v(V_d - V)$
  - In Coronal Plane: Compute virtual force using PD-controller tracking a desired COM position laterally.
  - Compute $\tau_v = J_v{}^T F_v$

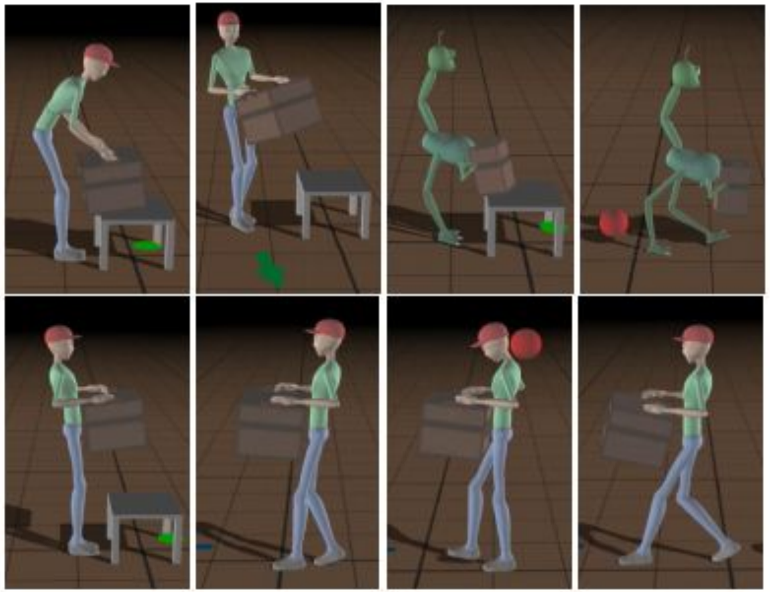


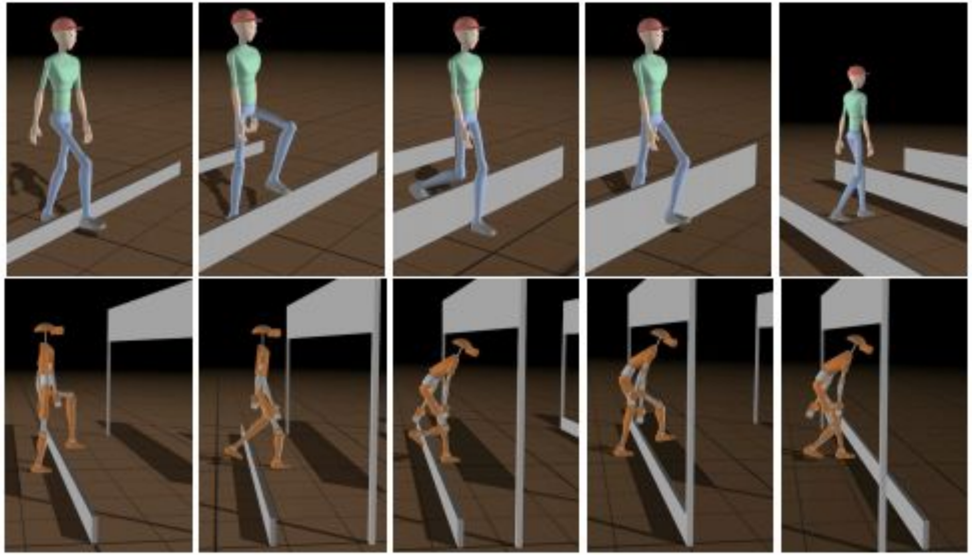
[1] Pratt, Jerry, et al. "Virtual model control: An intuitive approach for bipedal locomotion." *The International Journal of Robotics Research* 20.2 (2001): 129-143.

Results









Limitations

Limitations

- Does not work on fast, highly dynamic motions.
- Does not support authoring the “push recovery” styles.
  - This is governed by inverted pendulum.
- Swing and stance legs intersect (collide) in some instances.
- Does not generalize to characters with more than 2 legs.

Summary

Summary

- GENBICON proposes a generalizable physics based controller for locomotion, which generalizes across:
    - Gait parameters
    - Character proportions
    - Motion styles
    - Walking skills
- Supports multiple walking gaits like
    - forward-backward walking, different walking speeds, idling, walk to stop, stop to walk.
- Control further works with other walking related tasks, such as:
    - Picking up objects at a height.
    - Lifting/walking with heavy crates.
    - Pushing pulling crates.

SIMBICON vs GENBICON Comparison

SIMBICON vs GENBICON

Task	SIMBICON	GENBICON
User defined walking/running velocity	*No*	*Yes*
Center of Mass	*Pelvic Region*	*True*
Gravity Compensation	*No*	*Yes*
PD Gain Values	*Fixed*	*Scaled*
Interact with payloads (push/pull/carry)	*No*	*Yes*
Handle varied character proportions and weights	*No*	*Yes*

Questions?

Thank you!