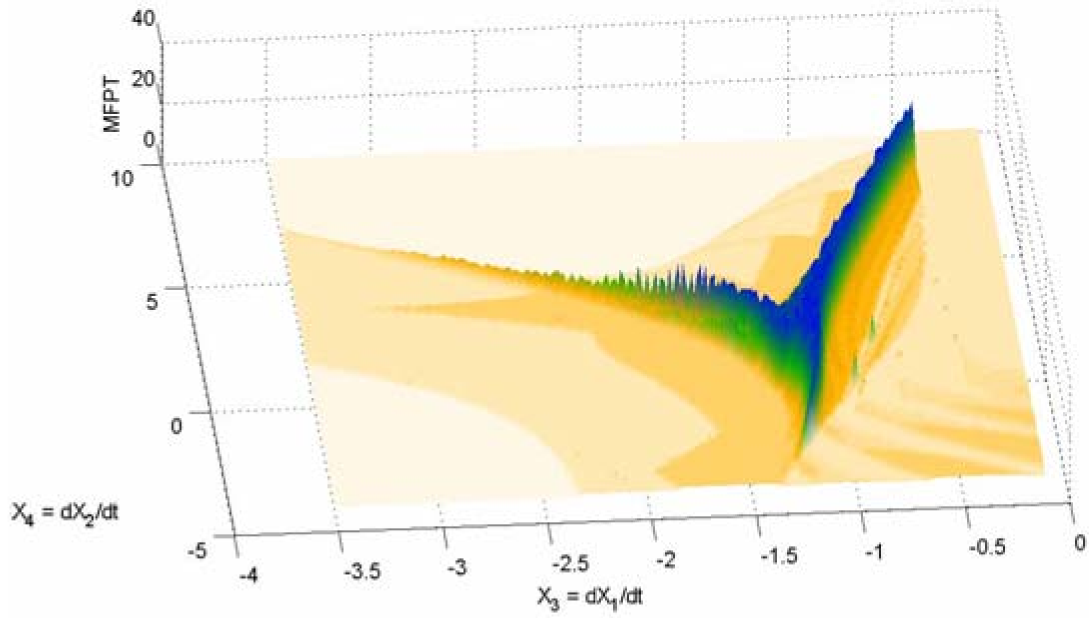
MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY



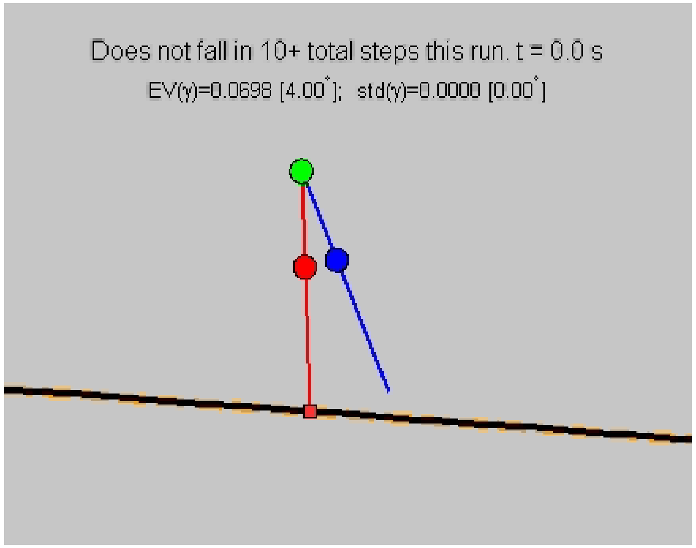
Stability of passive dynamic walking on uneven terrain

Katie Byl and Russ Tedrake

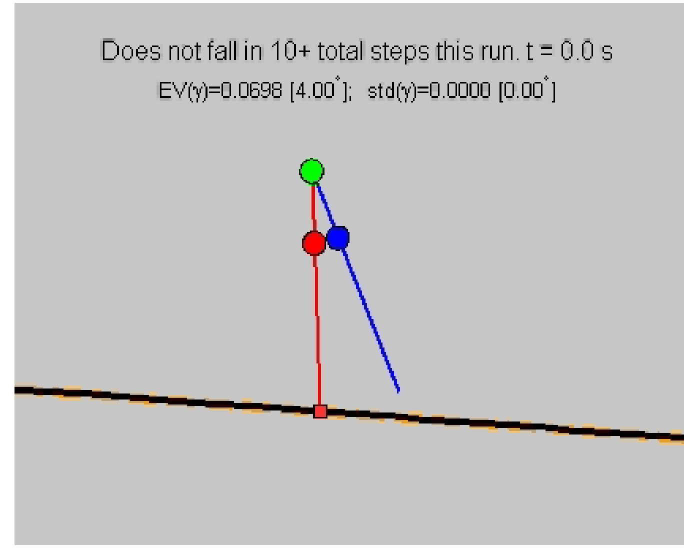
Robot Locomotion Group, MIT

Passive compass gait on uneven terrain

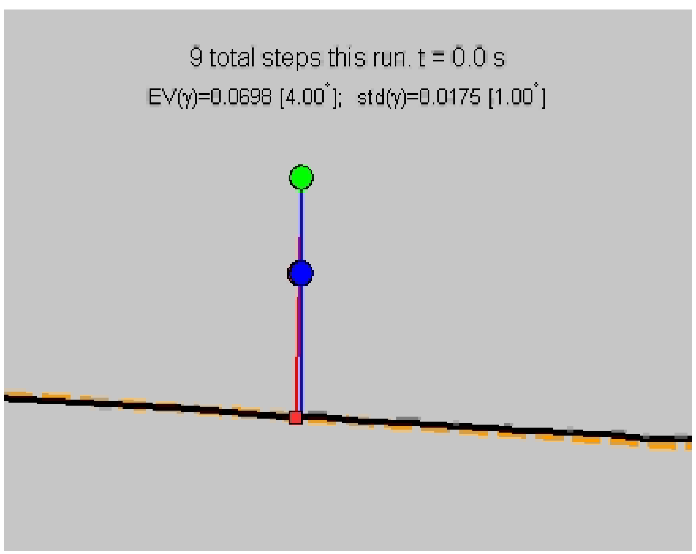
- **Is your walker stable?** **vs** ***How* stable is your walker?**

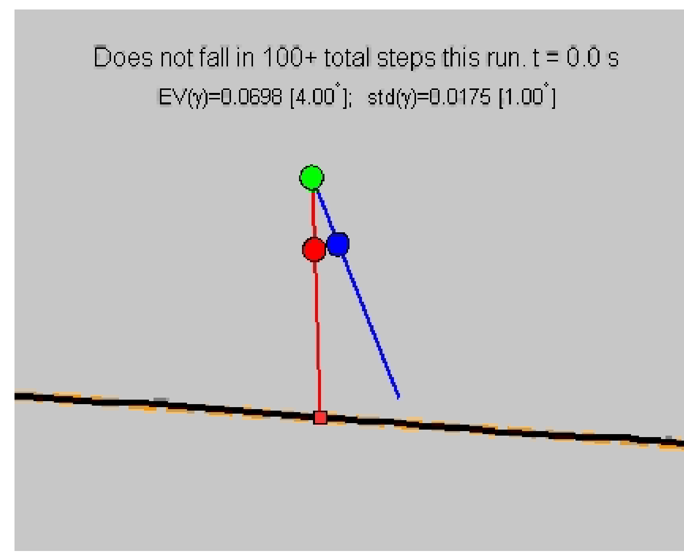


←Left: walker #1
Right: walker #2→



← Constant slope →
(upper movies)
Periodic gaits

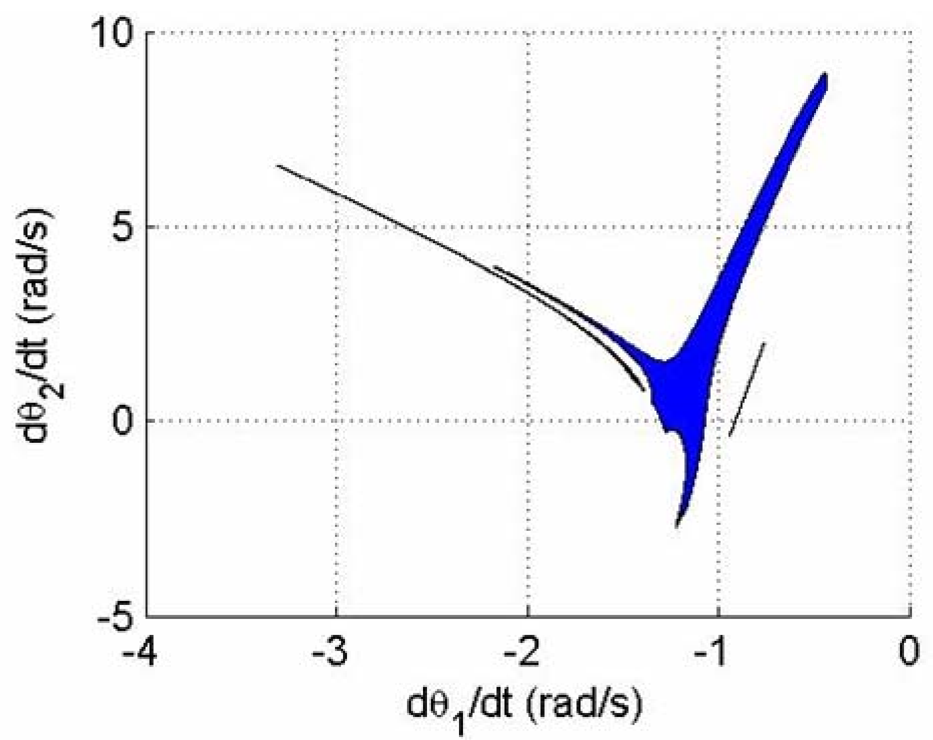


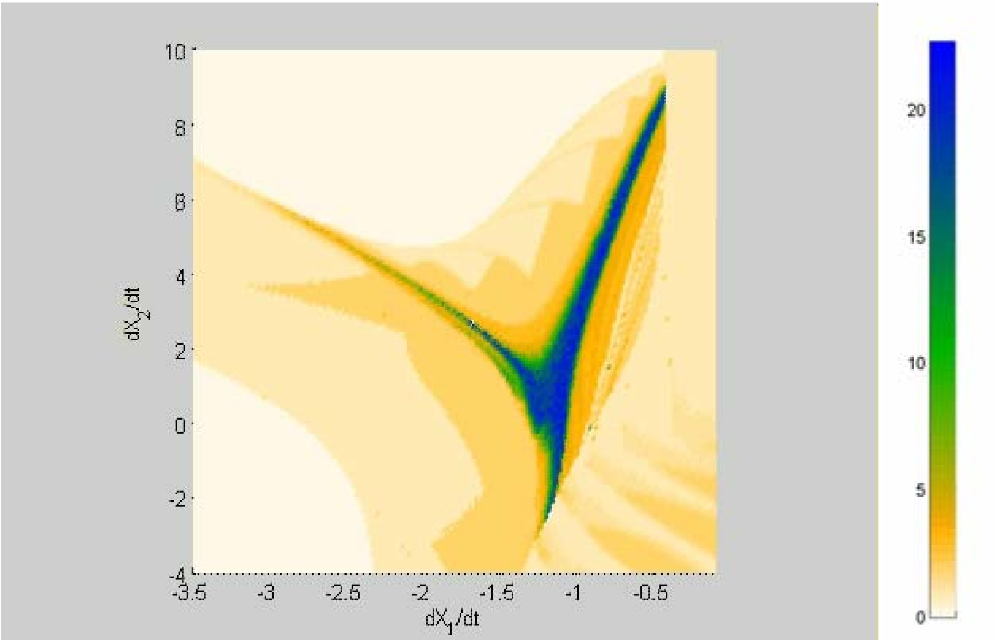


← Changing slope →
(lower movies)
***Aperiodic** gaits*

Stability metrics for dynamic walking

- **For <u>deterministic</u> systems：**
  - Global stability ：size and shape of deterministic (no noise) **basin of attraction**
  - Local stability ：**recovery from a single perturbation** about the fixed point
- **For <u>stochastic</u> systems：** *statistics of noise* map to *statistics* of failure
  - "**mean first passage time**" (**MFPT**) For walking, this is the expected number of steps taken before falling down. *[aka "mean time between failures"]*



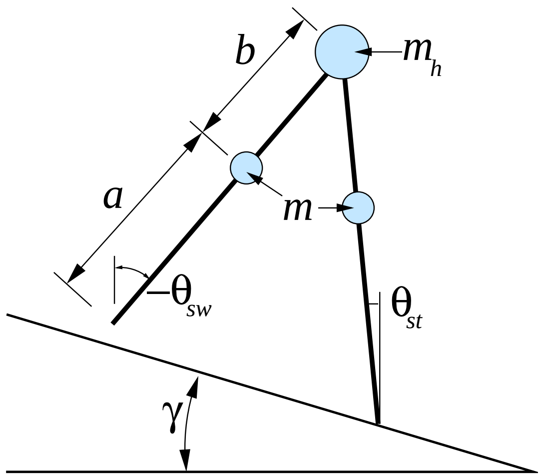


Above: average # of steps before falling

Slice of deterministic basin (left) and stochastic basin (right) for a CG

Methods: Monte Carlo simulations

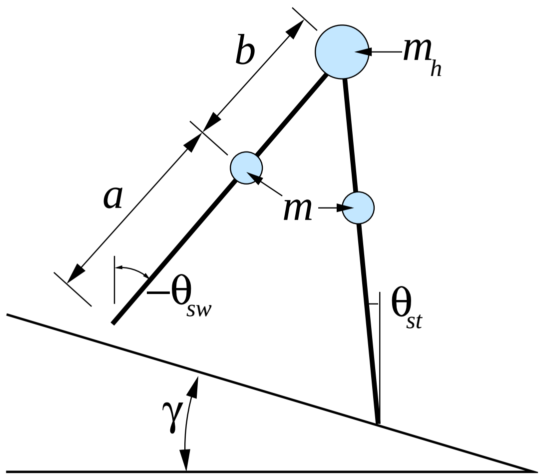
- **Example: passive compass gait on rough terrain**
  - Mean value (4 deg) for downhill slope
  - Gaussian distribution; testing std's of 0.5-2.0 deg
- **Set init. cond. and simulate dynamics over many trials**
- **Calculate "mean first passage time"** (MFPT) for each particular initial condition of interest
- Below are **MFPTs for init. cond. at the fixed point** for each respective walker



	$(.5m_h)/m$	a/(a+b)	MFPT .5 deg std	MFPT 1.0 deg std
Walker #1	**1**	**.6**	**20**	**6**
Walker #2	**.15**	**.7**	**>>100,000**	**150**

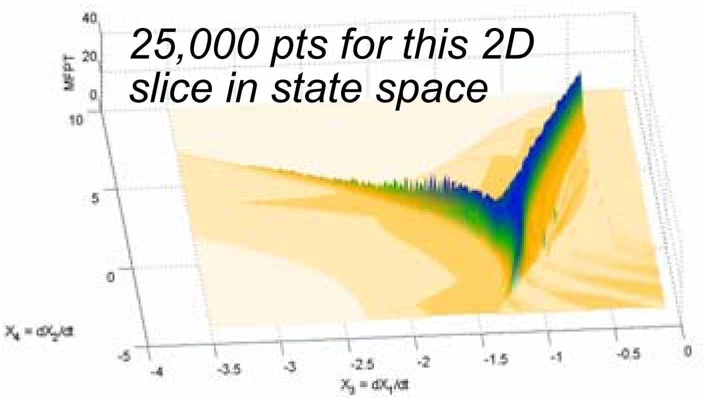
Methods: Monte Carlo simulations

- **Example: passive compass gait on rough terrain**
  - Mean value (4 deg) for downhill slope
  - Gaussian distribution; testing std's of 0.5-2.0 deg



- **Set init. cond. and simulate dynamics over many trials**
- **Calculate "mean first passage time"** (MFPT) for each particular initial condition of interest
- Below are **MFPTs for init. cond. at the fixed point** for each respective walker

	$(.5m_h)/m$	a/(a+b)	MFPT .5 deg std	MFPT 1.0 deg std
Walker #1	**1**	**.6**	**20**	**6**
Walker #2	**.15**	**.7**	**>>100,000**	**150**

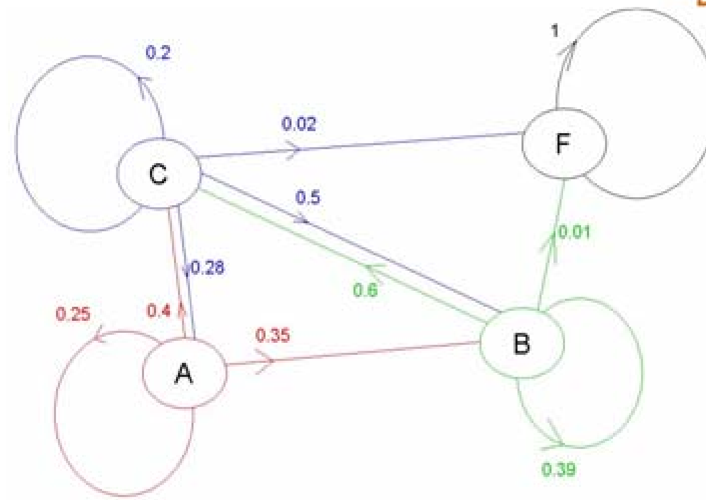


Monte Carlo method is computationally intense

- Estimating MFPT over the entire state space takes many, many trials
- **We present a more direct method to calculate this distribution…**

Modeling the system as a Markov chain:
step-to-step transition matrix, *f*

$$f = \begin{bmatrix} 0.25 & 0.35 & 0.4 & 0 \\ 0 & 0.39 & 0.6 & 0.01 \\ 0.28 & 0.5 & 0.2 & 0.02 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad f^{10} = \begin{bmatrix} 0.13915 & 0.38823 & 0.3665 & 0.10611 \\ 0.13714 & 0.38261 & 0.36118 & 0.11907 \\ 0.13655 & 0.38098 & 0.35966 & 0.12281 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



- **Non-iterative calculation of state-dependent MFPT, *m*** *(a vector)*
  - $m_i = \Sigma f_{ij} m_j + 1$, summed over all j s.t. $s_j \neq$ failed state
  - $[I - f']m = 1$ (eqn above in matrix form)

➔ $m = [I - f']^{-1} 1$ ***direct calculation of MFPT!***

  - ***m*** is a vector giving the MFPT at each discrete state (mesh node)
  - ***I*** is the identity matrix
  - ***f'*** contains the *non-absorbing rows and cols of **f***
  - ***1*** is the ones vector
- Gradient in ***m*** can be used as a metric for **remeshing**
- Note: for a deterministic system (no noise), ***m***=∞ in the basin of attraction

System-wide stochastic stability

- **Eigenvalue analysis of the transition matrix, *f***
  - Any initial condition is a weighted sum of the eigenvectors
  - Each corresponding eigenvalue shows how rapidly that part fades away
  - **Look for eigenvector(s) that persist**; i.e. describe long-term distribution

Calculate first 3 eigenvalues and eigenvectors of (sparse matrix) f^T

- $\lambda_1=1$ **failure is an absorbing state**; it persists for all time
  1st eigenvector: $[0,\ldots,0,1]^T$ shows to inevitability of a "failure" as $t\rightarrow\infty$

- λ_3 provides an estimate of **"mixing time"** to forget initial conditions.
  "Fast" mixing implies: $1/\tau_2=\log(1/|\boldsymbol{\lambda_2}|) \ll \log(1/|\boldsymbol{\lambda_3}|)= 1/\tau_3$,

so $(1-|\lambda_2|) \ll (1-|\lambda_3|)$ **implies separation of time scales.**

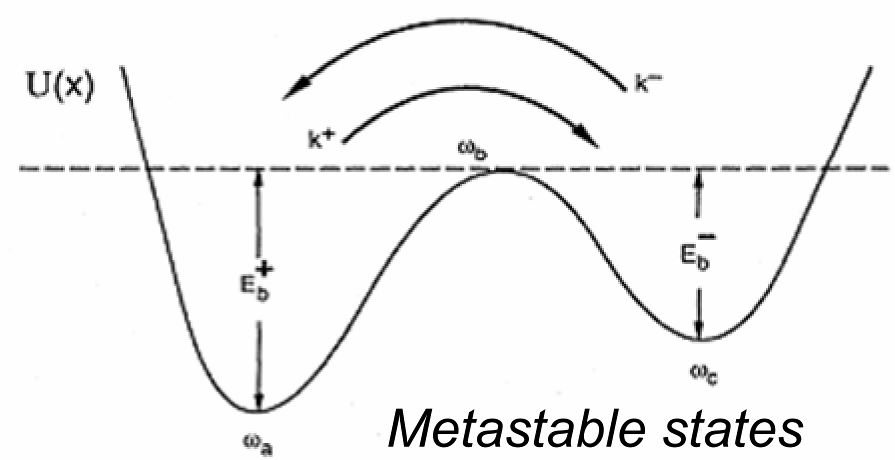
- $1-|\lambda_2| = r$; $r = 1/m$ ("leakage rate" is the inverse of the MFPT)
  2nd eigenvector renormalized (to exclude failure state) represents the **quasi-stationary distribution of the stochastic basin of attraction**.

System-wide stochastic stability

- **An elegant simplification emerges!**
  - For our simulations, the magnitude of λ_3 is about 0.5 (fast mixing), so walkers which have not failed will converge rapidly to a **quasi-stationary distribution of states**, which is given by the eigenvector associated with λ_2.
  - Failures (falling) occur at a slow, calculable **leakage rate,** $r \approx 1-|\lambda_2|$
  - $\lambda_1 = 1$ implies the robot will *eventually* fall, but a small leakage rate means we still expect aperiodic walking to persist for a long time before falling.

- **"Metastable" (i.e. long-living) states**
  - We should think of dynamic walking as convergence to a metastable limit cycle, with a slow leak rate, *r*, to an absorbing failure state (falling down).
  - **mfpt=1/r** gives a system-wide mean first passage time. It is a *scalar* quantity that characterizes the stability of the system and answers the question:

    "*How* stable is your walker?"



Metastable states

The End

- **Additional slides follow… (more video, et al)**

"for a deterministic system (no noise), *m*=∞ in the basin of attraction"

- In other words, if you set the noise to "zero", you are calculating the basin of attraction for the DETERMINISTIC system using the step-to-step transition matrix, ***f*** ; this basin is the region where MFPT (***m***) is "infinite".

- If you have a description of the equations of motion (to calculate the step-to-step state transition), you can identify whether or not stable limit cycles exist **w/out tweaking** (trial and error) by hand to search for appropriate initial conditions.

- You need to take care to do appropriate (iterative) **remeshing** (and de-meshing) of the state space to get good resolution!! (i.e. try some mesh; calculate MFPT; then put in more mesh elements where MFPT changes drastically… , calc MFPT,…

Review:

How to answer, "how stable is your walker?"

- **Monte Carlo approximation of MFPT from initial conditions**
  - computationally intense
- **Direct (non-iterative) calculation of vector MFPT, *m*, using the transition matrix, *f***
  - Vector ***m*** and its gradient can be used in refining mesh
- **System-wide stability analysis, by finding the largest eigenvalues and eigenvectors of f^T.**
  - scalar MFPT describes system
  - quasi-stationary distribution can be found
  - aperiodic walking can be modeled as a metastable limit cycle with a slow leakage rate.

Statistical metrics for stochastic stability

- **Goal: Quantify stability for a system with definable noise**

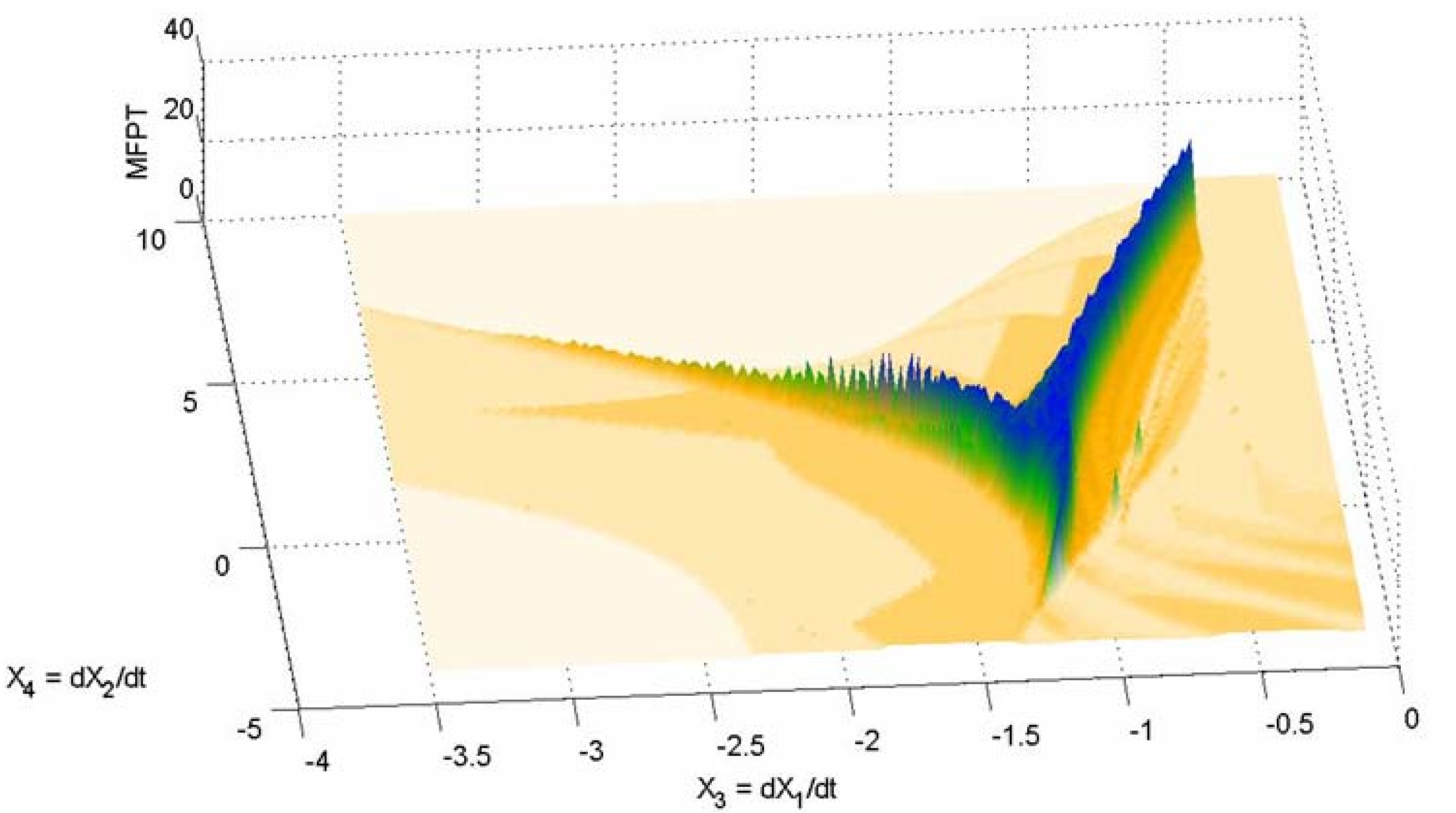
- **New stability metrics:**
  - Describe statistics of failure events
  - **MFPT : "mean first passage time"**
    * Also called **"mean time between failures" (MTBF)**
    * Longevity can also be measured in *number of steps* (rather than "time")
  - MFPT = 1/r (inverse of leakage rate)
  - $P_x(t)$: probability of falling by time t
  - ML (maximum likelihood) time to fall
  - time at which probability of having fallen exceeds some critical limit

Direct (Matrix) Calculation of MFPT

1) Discretize (mesh) the state space
2) Create the step-to-step (Poincare) transition matrix, $\boldsymbol{f}$
   - $\boldsymbol{f_{ij}} = Pr(\, s_{n+1}=j \;|\; s_n=i \,)$, given our dynamics and noise.
   - New states, s_{n+1}, modeled by probabilistic arrival at nearby mesh nodes.
   - "Failure" (falling) is a self-absorbing state in $\boldsymbol{f}$.
3) Calculate the 3 largest eigenvalues ($\lambda_1, \lambda_2, \lambda_3$) of $\boldsymbol{f}^T$
   - $\lambda_1=1$; 1st eigenvector: $[0,\ldots,0,1]^T$ shows inevitability of a "failure" as $t\rightarrow\infty$
   - $1-\lambda_2= r$; $r = 1/mfpt$ (metastable "leakage rate") ; 2nd eigenvector gives the quasi-stationary distribution of the metastable basin of attraction.
   - λ_3 provides an estimate of "mixing time" to forget initial conditions. "Fast" mixing implies: $1/\tau_2=\log(1/\boldsymbol{\lambda_2}) \ll \log(1/\boldsymbol{\lambda_3})= 1/\tau_3$, so $(1-\lambda_2) \ll (1-\lambda_3)$
4) Calculate the MFPT for each discrete node in the mesh
   - $\boldsymbol{m}=[\boldsymbol{I}-\boldsymbol{f}']^{-1}\boldsymbol{1}$, where $\boldsymbol{f}'$ contains the non-absorbing rows and cols of $\boldsymbol{f}$, and $\boldsymbol{1}$ is the ones vector
5) Refine mesh where the gradient in MFPT is most significant

Monte Carlo = computationally intense

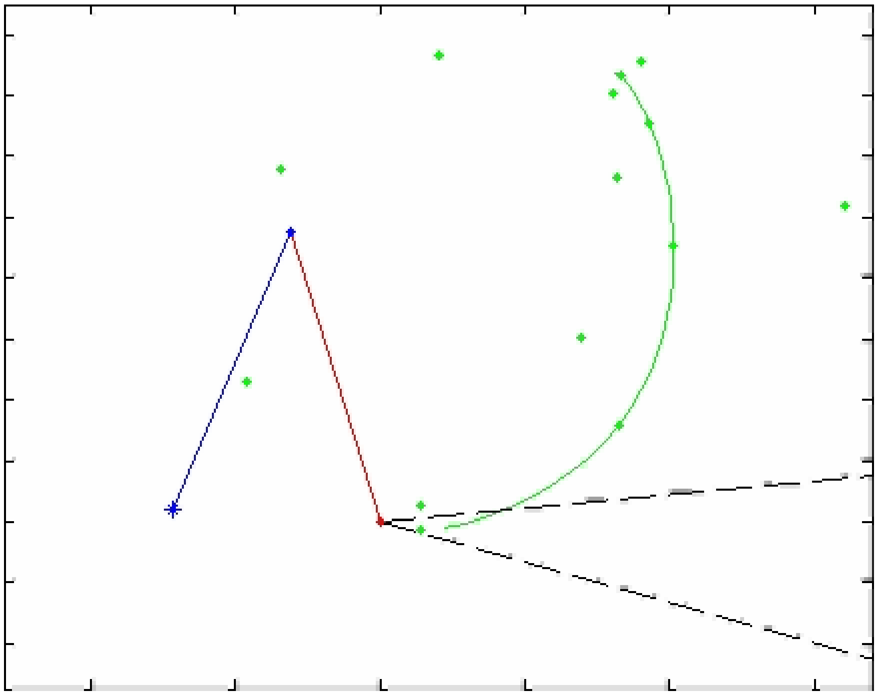
- Estimating the MFPT over the state space takes many, many trials
- Motivation for **efficient mathematical tools**
- We present a more direct method to calculate this distribution…

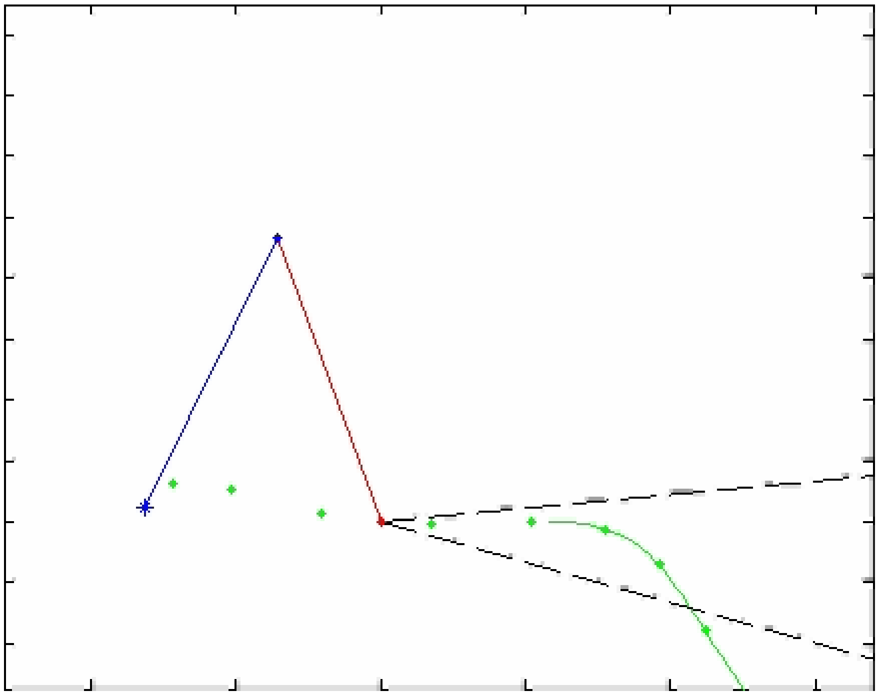


MFPT over a 2D slice of (3D) state space

Case Study: Passive Compass Gait on Rough Terrain

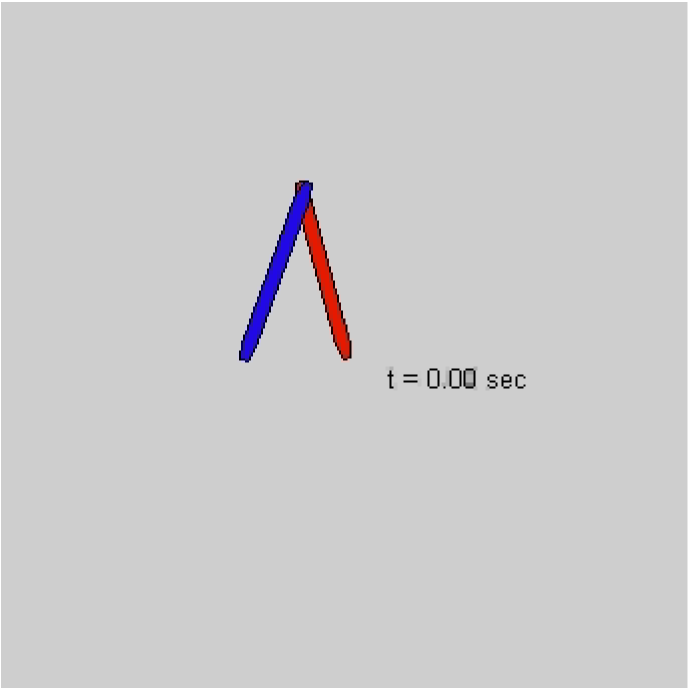
- Once walker begins a step, it follows a deterministic trajectory until it "hits the ground"
- Thus, we can pre-calculate and save trajectories; then interpolate to look up next step's initial condition (if any!) as a function of ground slope. Examples below…

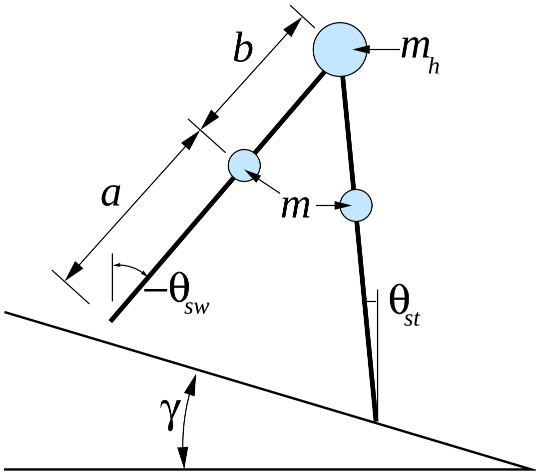


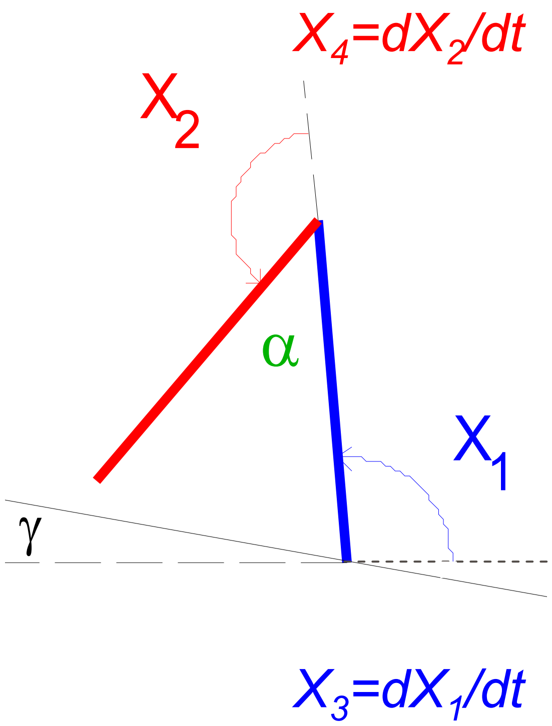


Case Study: Passive Compass Gait on Rough Terrain

- **Using "acrobot" (Spong) definition for states**
  - Continuous equations of motion are identical to the acrobot between the discrete impacts
  - 4 states variable: Angles X_1 and X_2, and their derivatives (X_3 and X_4)



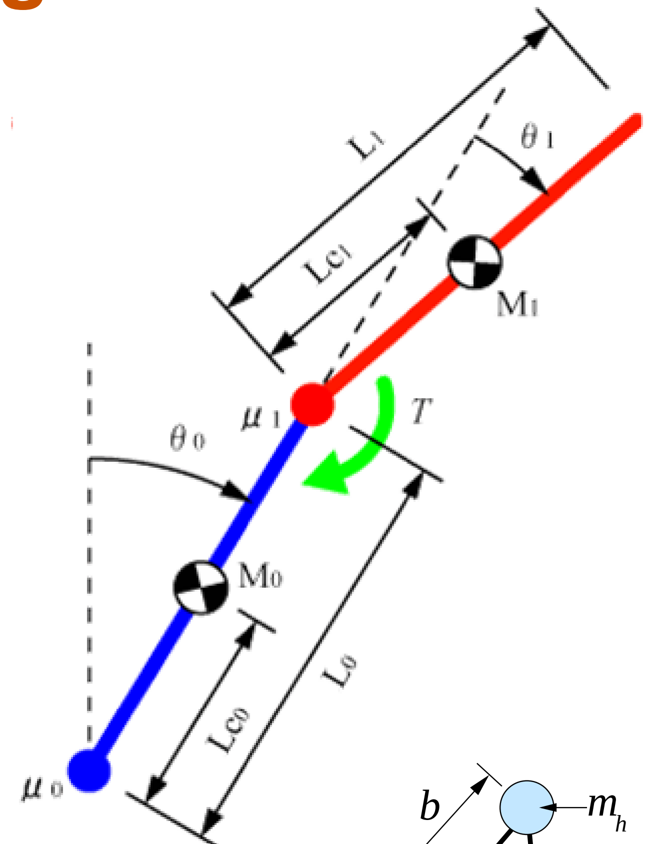


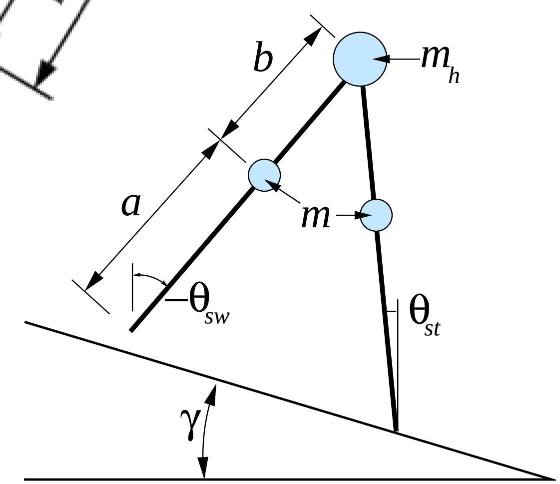


Case Study: Passive Compass Gait on Rough Terrain

- **Absolute mass not import: it's how the mass is distributed!**
- **Dimensionless inertia: $I/(mL^2)$**
- **Intuitively, want low inertia swing leg. (Mass toward upper part of leg.)**
- **Three walkers analyzed:**

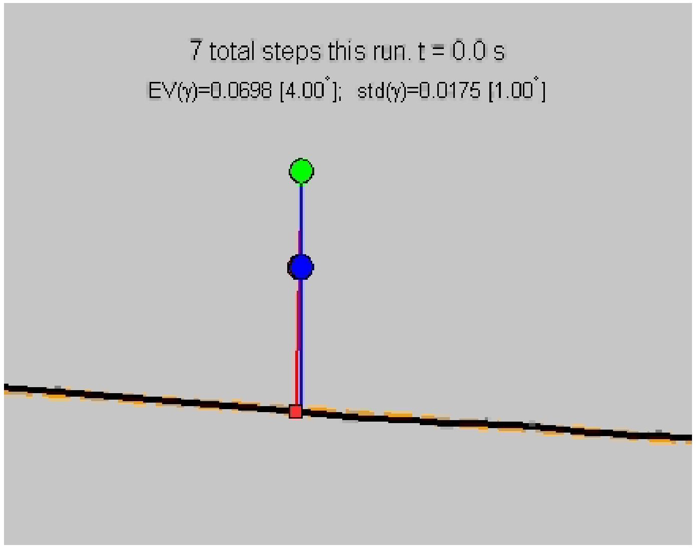
	(.5mh)/ m	a/(a+b)	$I/(mL^2)$	Lco/L
Mid-size	1	.6	.0400	.8
Low-inertia	.15	.7	.0102	.74
Beam-leg	1/3	1/3	.0833	.5

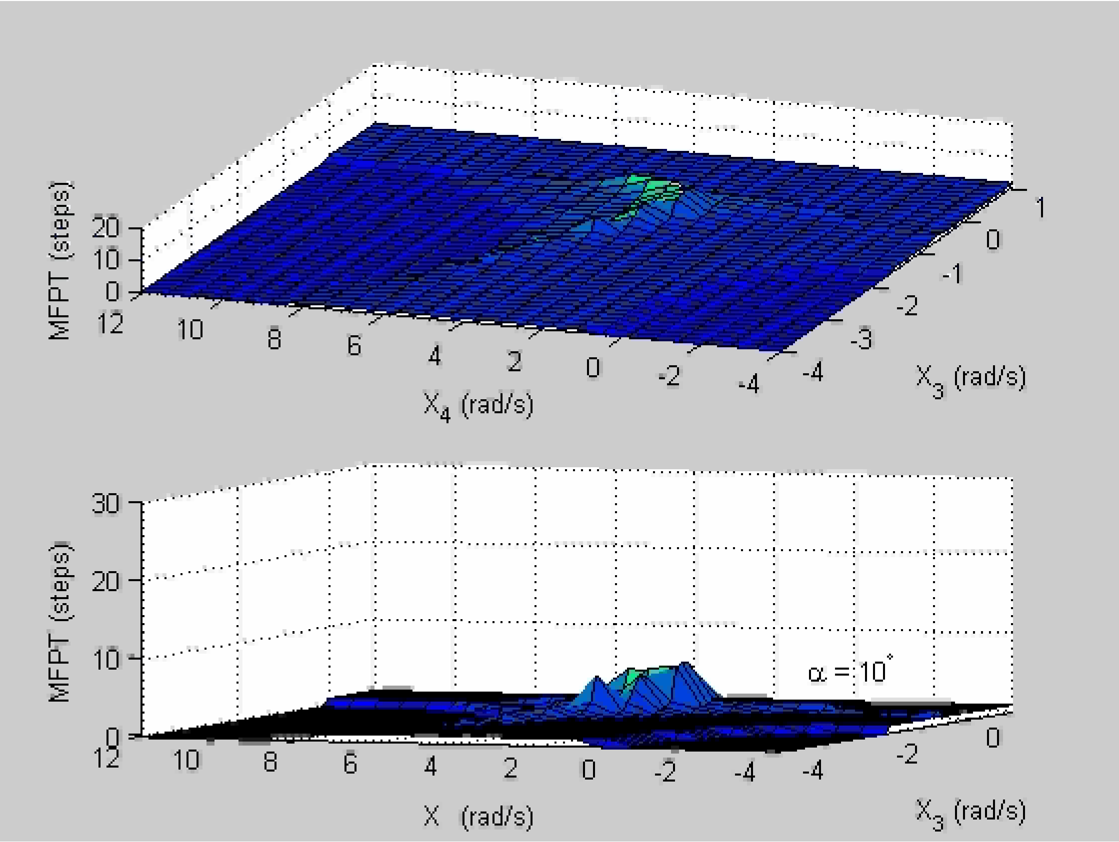


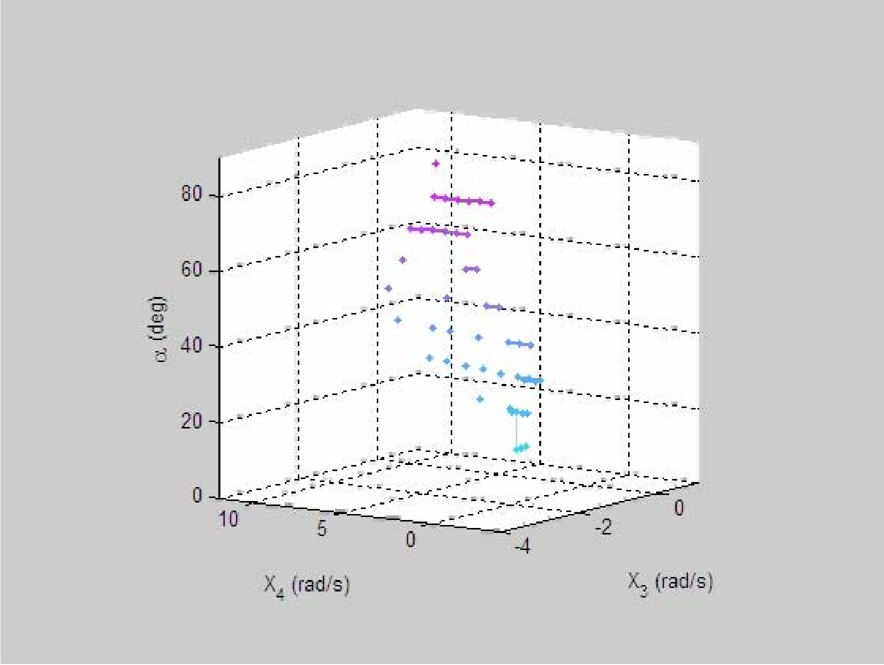


Initial walker design ("mid-size")

- **Mean = 4 deg slope**
- **STD = 1 deg**
- **MFPT ≈ 6 steps**

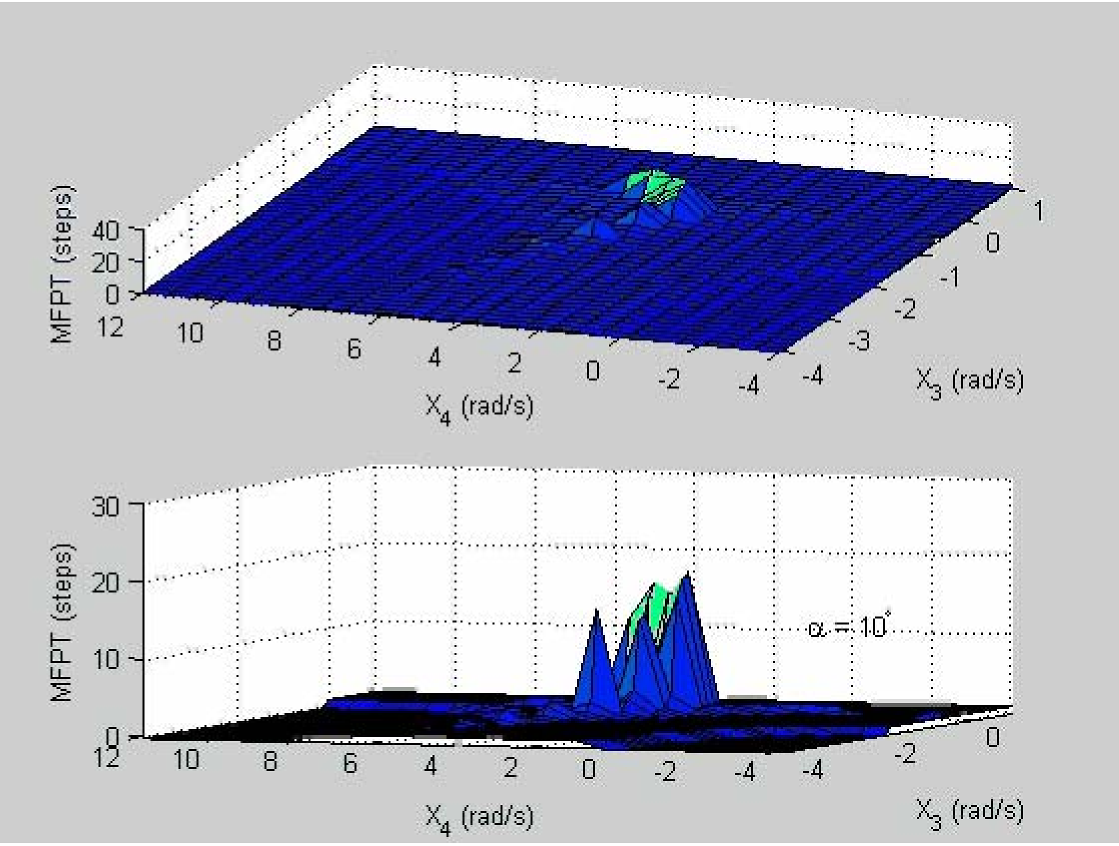


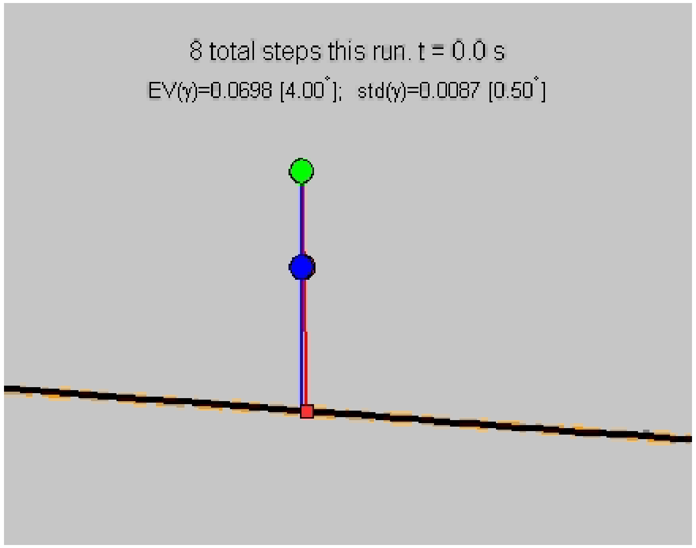


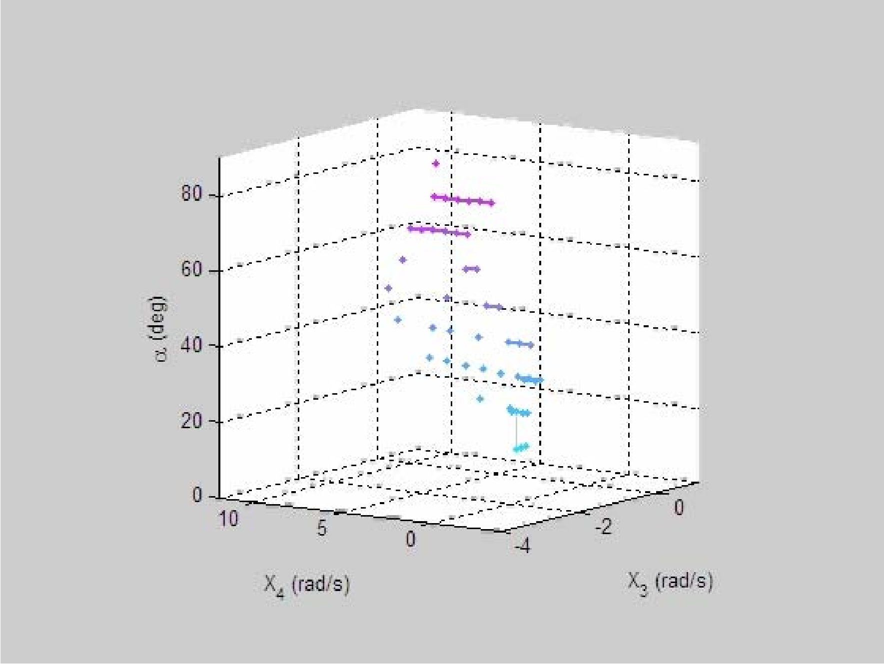


Initial walker design ("mid-size")

- **Mean = 4 deg slope**
- **STD = 0.5 deg deg**
- **MFPT ≈ 12 steps**

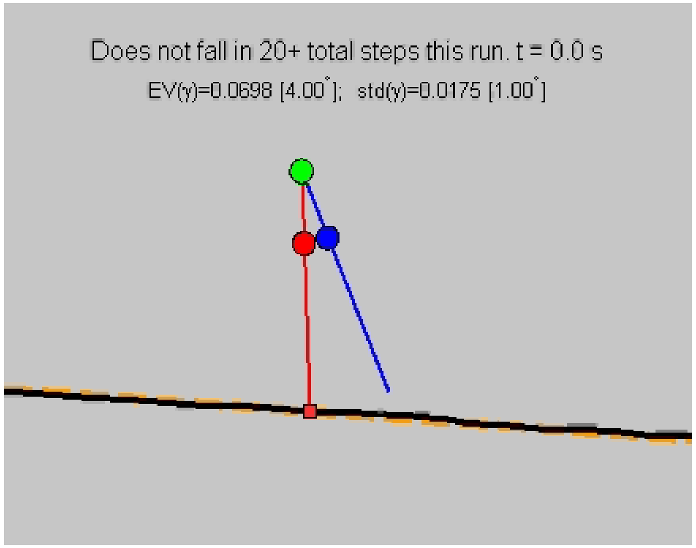


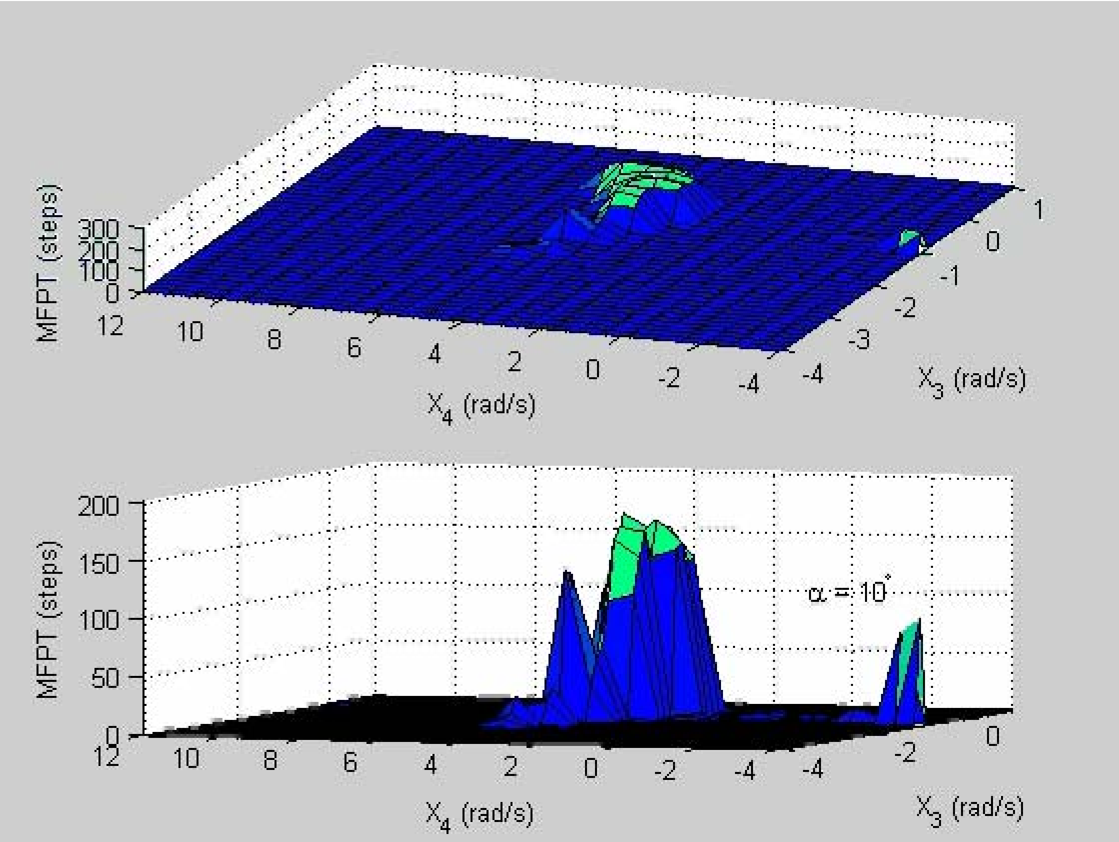


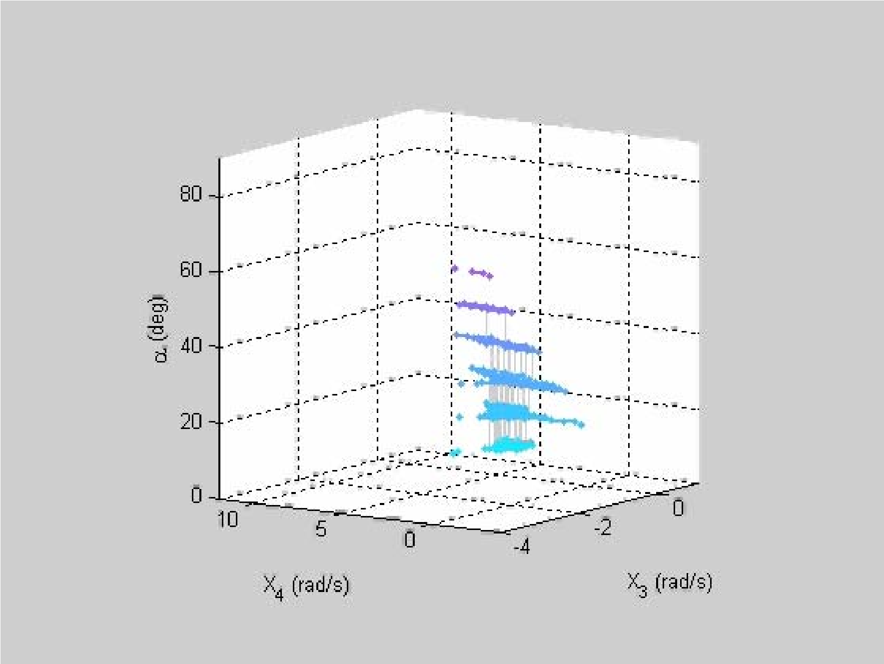


Low-inertia walker (more stable)

- **Mean = 4 deg slope**
- **STD = 1 deg**
- **MFPT >= 110 steps**

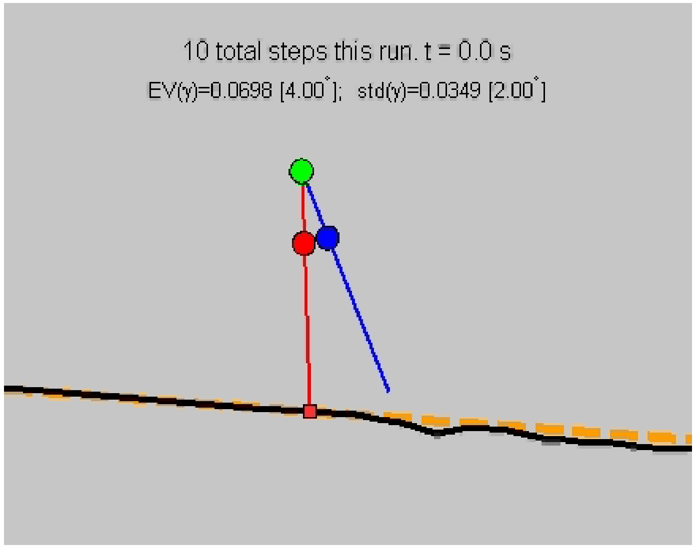


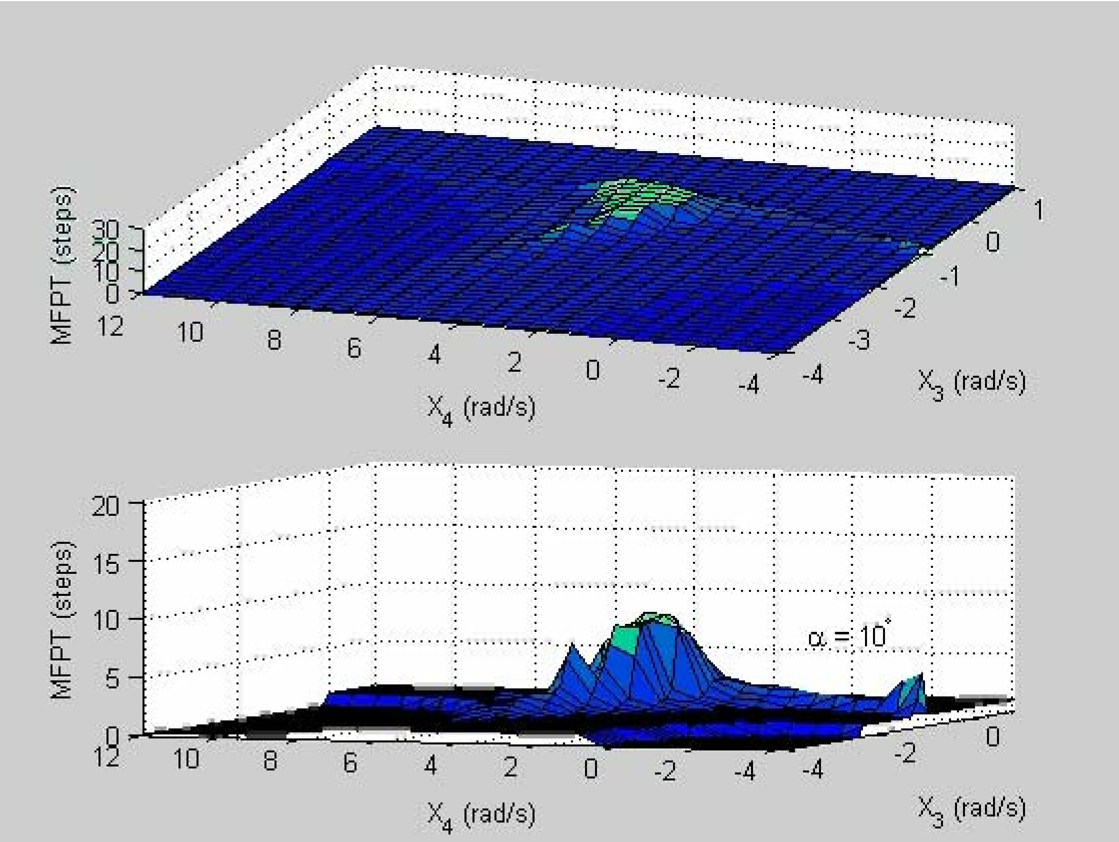


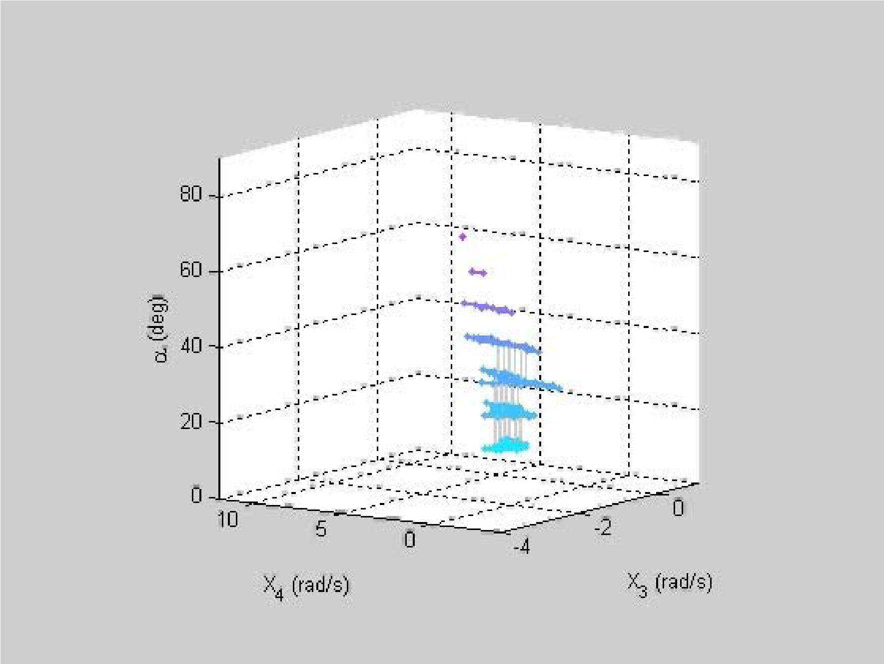


Low-inertia walker (more stable)

- **Mean = 4 deg slope**
- **STD = 2 deg**
- **MFPT ≈ 8 steps**

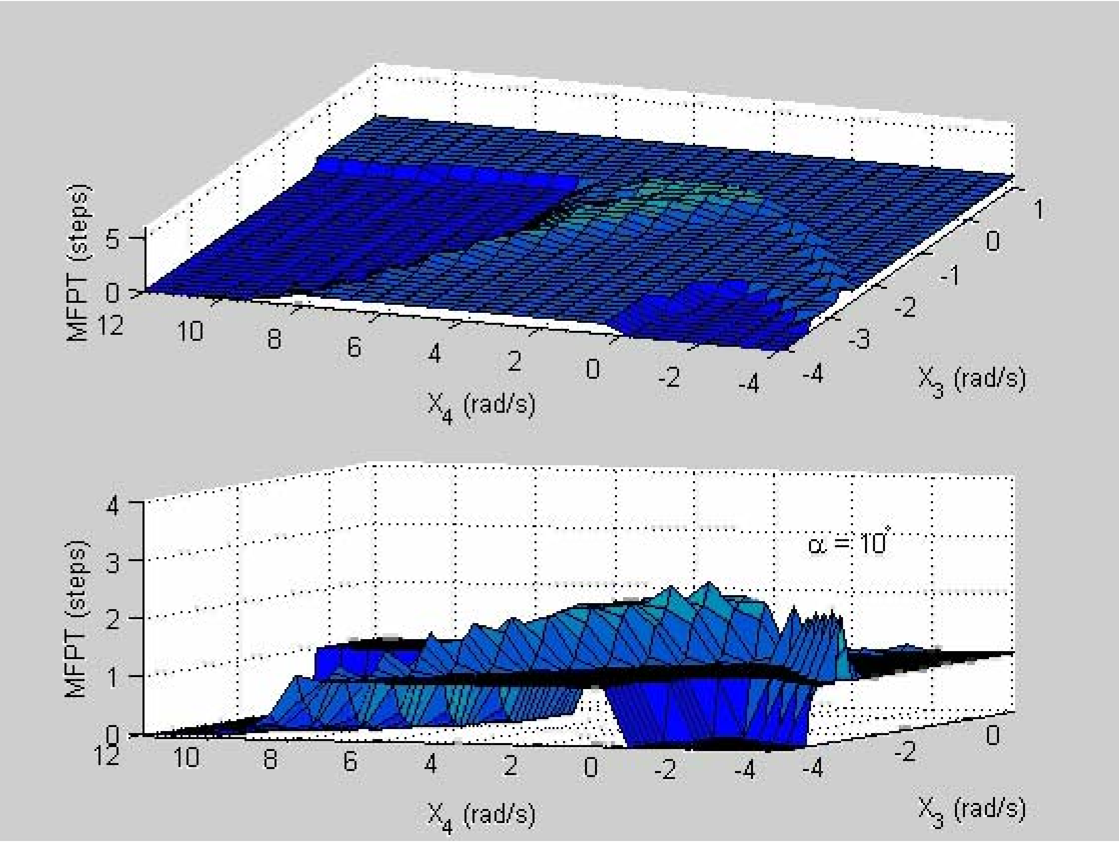


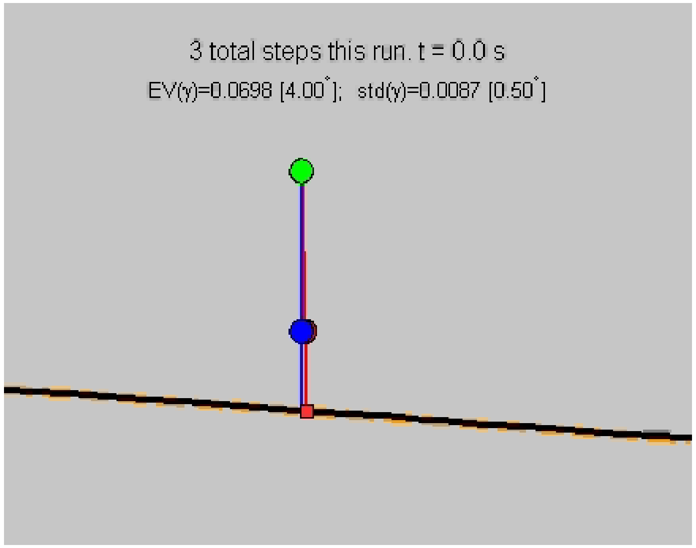




Beam-legged walker

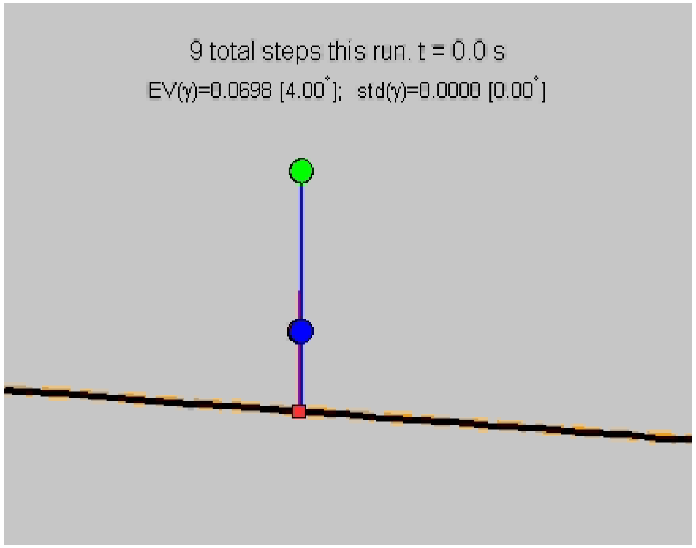
- **Mean = 4 deg slope**
- **STD = 1 deg**
- **MFPT ≈ 2 steps**





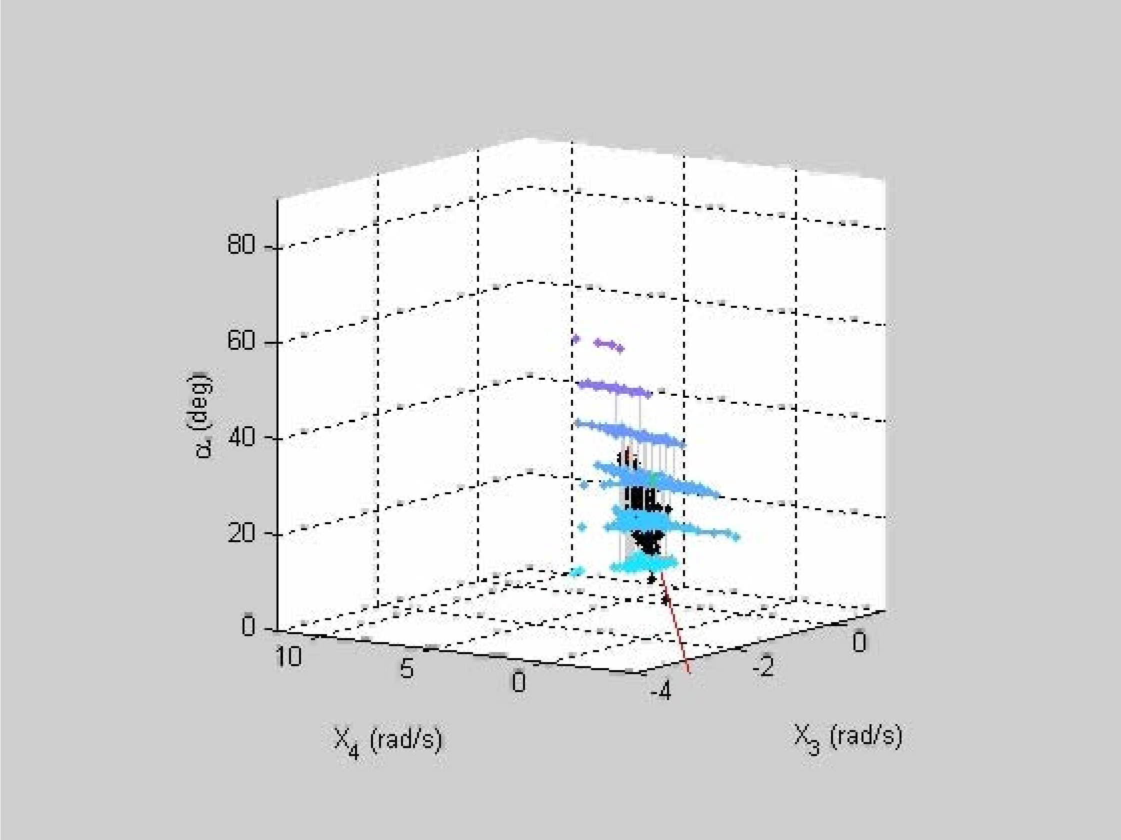
Above: SD = 1 deg

Below: SD = 0 deg (even)



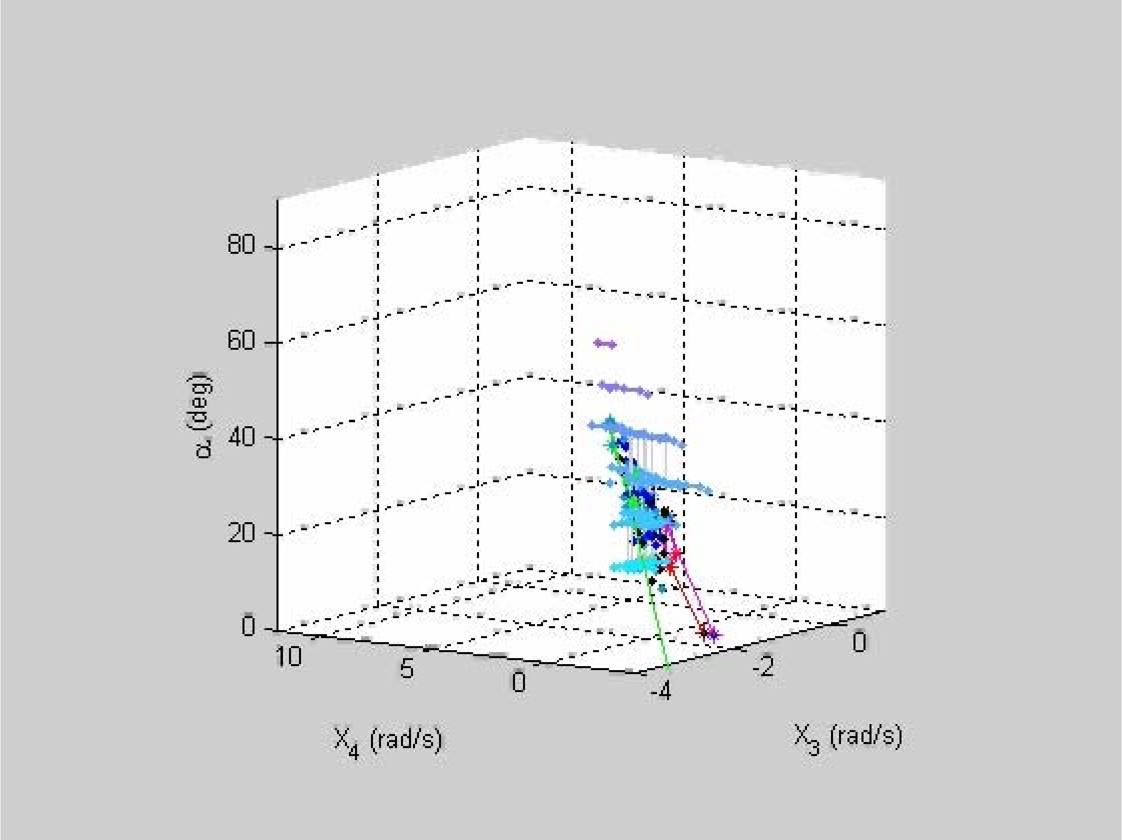
What (metastable) “neighborhood” in phase space is visited most often?

- **Most stable walker (low-inertia version) shown here**
- **MFPT of about 110 steps (STD of terrain = 1 deg)**
- **Black points indicate post-hit states (X3,X4 and alpha)**



What (metastable) “neighborhood” in phase space is visited most often?

- Same (low-inertia) walker with STD = 2 deg (double)
- MFPT of about 8 steps
- 3 trials plotted (as points) on same axes here



MFPT relates to probability of a catastrophic (n-sigma) event (?)

- **As the level of noise decreases, a "failure" may essentially correspond to the probability of a single large-gamma step on the terrain…**

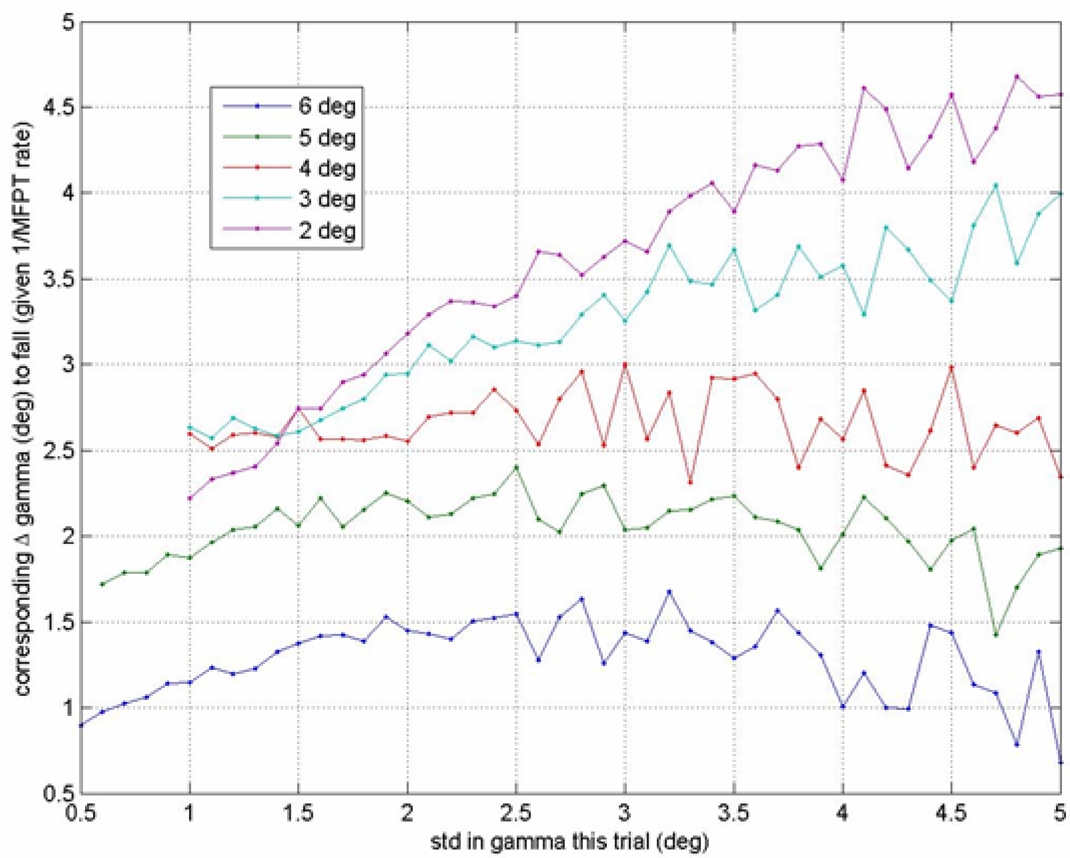
At right:

-MFPT recorded

-For a given std, what value "jump" in gamma corresponds to the leakage rate, 1/MFPT?

-Flat lines would indicate the walker is essentially waiting for a particularly bad one-time event

-Requires more run-time to make a conclusion here



Hip-Actuated Compass Gait Robot

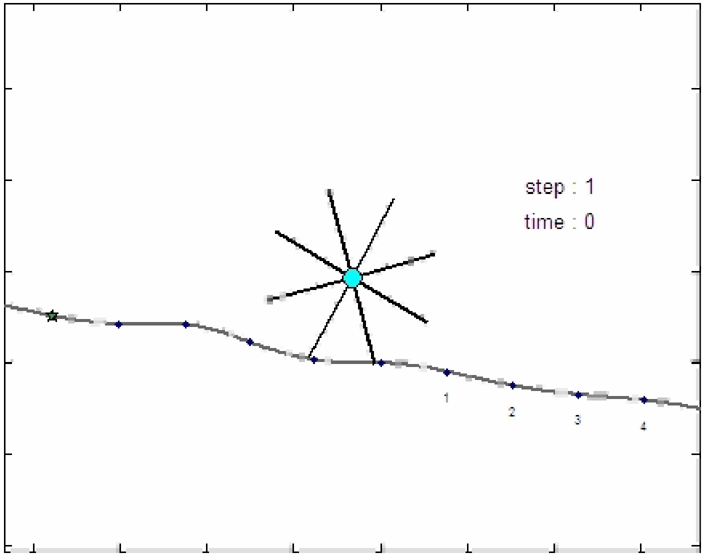
- **Robot under construction:**
  - CPU: PC/104, with MATLAB (Simulink)
  - Single actuator (motor w/ gearbox) at "hip"
  - Brake used as clutch to (dis)engage motor coupling between the legs.
  - 3 rate gyros; 2 encoders; 2 accelerometers
  - Reinforcement learning

- **Future modifications:**
  - Retractable (telescoping) "point" feet
  - Rugged terrain
  - Replace power-hungry PC/104?
  - Direct drive motor!

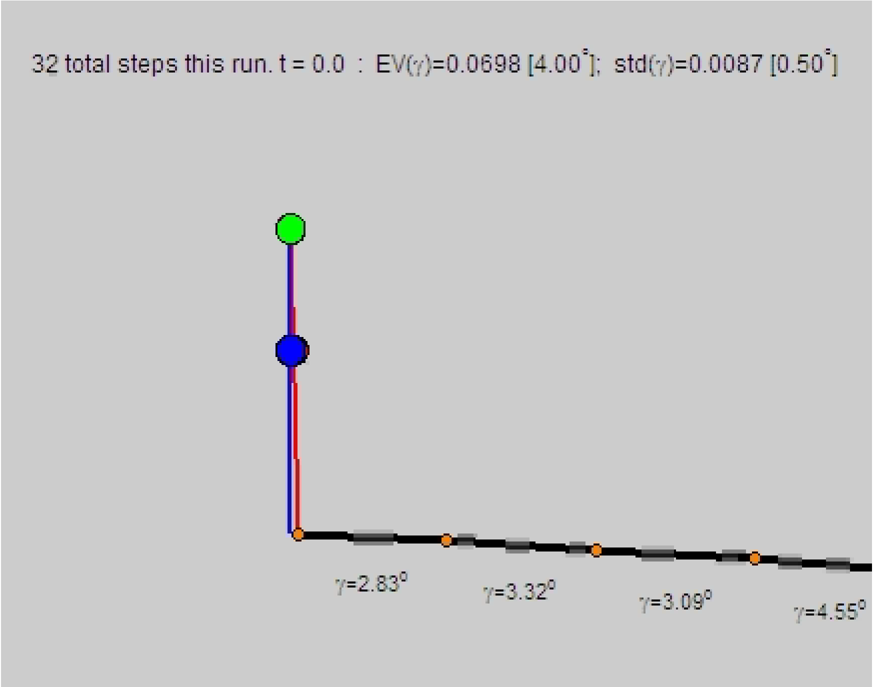
- **Thanks to Arlis Reynolds (UROP) and Stephen Proulx (staff) !**

Simple Biped Models

- **Rimless Wheel**
  - Simplest "walker"
  - Hybrid dynamics:
    - * **continuous inverted pendulum**
    - * **discrete state change at impact**
  - Analogous to dynamics of a biped with all mass at hips

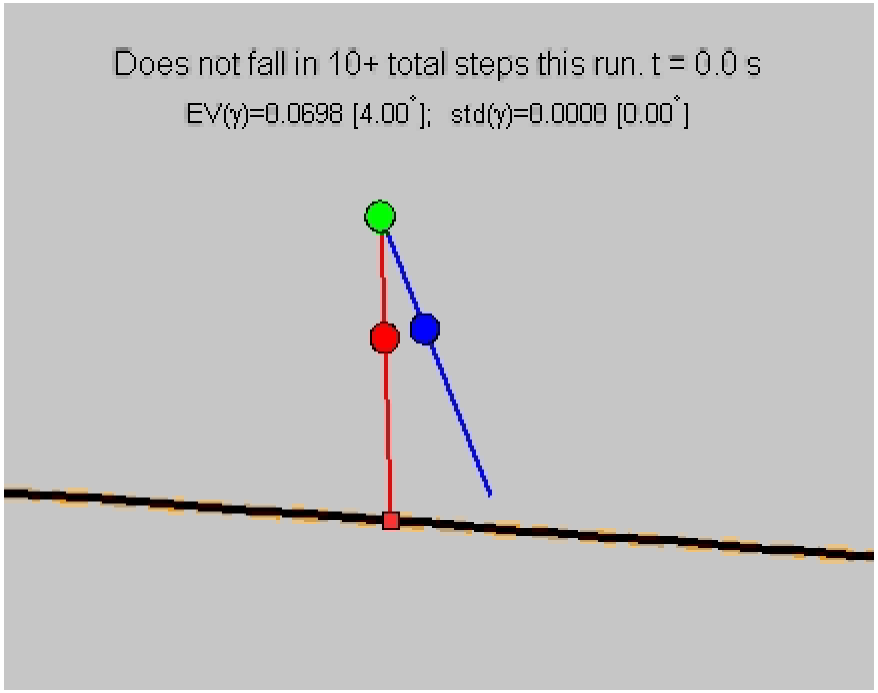


- **Compass Gait**
  - Resembles a compass
  - Stable limit cycles exist for particular downhill slopes
  - Idealized CG model ignores:
    - * **lateral stability**
    - * **ignores foot scuffing (no knees)**



Traditional Stability Margin for Walkers

- **Standard stability margins :**
  - Zero-moment point (ZMP)
- **…but a stable compass gait is always "falling forward"!**



Stable compass gait on even terrain

Asimo

Robot Locomotion Group
CSAIL, MIT

- **Lab focus:**
  - Robot locomotion
  - Control of underactuated systems
  - Reinforcement learning

- **Examples:**
  - “Toddler” (ankles actuated)
  - Hip-actuated CG walkers
  - Kneed walkers
  - RC airplanes
  - Ornithopter
  - Soap film flow between filaments
  - Acrobot
  - DARPA “Little Dog” project








Outline

- **Introduce the concept of *stochastic* stability**
  - Given a particular noise input, how often (statistically) will a walker fall?
  - Long-living, aperiodic gaits can be modeled as **"metastable"** states
  - Use statistics of failure such as the **"mean first passage time"** (MFPT) to define the relative degree of stability for a walker that will rarely, but inevitably, fall

- **Discuss methods for determining failure statistics**
  1. Monte Carlo simulations
  2. Calculations on the (probabilistic) step-to-step transition matrix, **f**, to obtain failure statistics from any ***particular initial condition***
  3. Characterize stochastic stability using ***system-wide*** stability measures:
     * quasi-stationary distribution of states visited in the metastable basin
     * mixing time (to converge to basin) and system-wide failure rate

- **Examples using a purely passive compass gait (CG) walker**
  - Gaussian variation in slope of terrain at each step

Modeling the system as a **Markov chain:** step-to-step transition matrix, ***f***

- **Iterative calculation of MFPT**
  - ***f***^{n} is the n-step transition matrix
  - Calculate $\Sigma n(f^n)_{ij}$ to get MFPT from state, *i*, to the failure state, j.
  - Infinite sum (as n goes to ∞) can be calculated non-iteratively (below)

- **Non-iterative calculation of MFPT, *m***
  - $m_i = \Sigma f_{ij} m_j + 1$, summed over all *j* s.t. $s_j \neq$ failed state
  - $[I-f']m=1$ (eqn above in matrix form)

  → $m=[I-f']^{-1}1$ ***direct calculation of MFPT!***
    - ***m*** is a vector giving the MFPT at each discrete state (mesh node)
    - ***I*** is the identity matrix
    - ***f'*** contains the *non-absorbing rows and cols of* ***f***
    - ***1*** is the ones vector
  - Gradient in ***m*** can be used as a metric for **remeshing**

Analysis: *System-wide* stochastic stability

- **Eigenvalue analysis of the transition matrix, *f***
  - Calculate first 3 eigenvalues and eigenvectors of (sparse matrix) f^T

1) $\lambda_1=1$ (failure is an absorbing state; it persists for all time)
   1st eigenvector: $[0,\ldots,0,1]^T$ shows to inevitability of a "failure" as $t\to\infty$

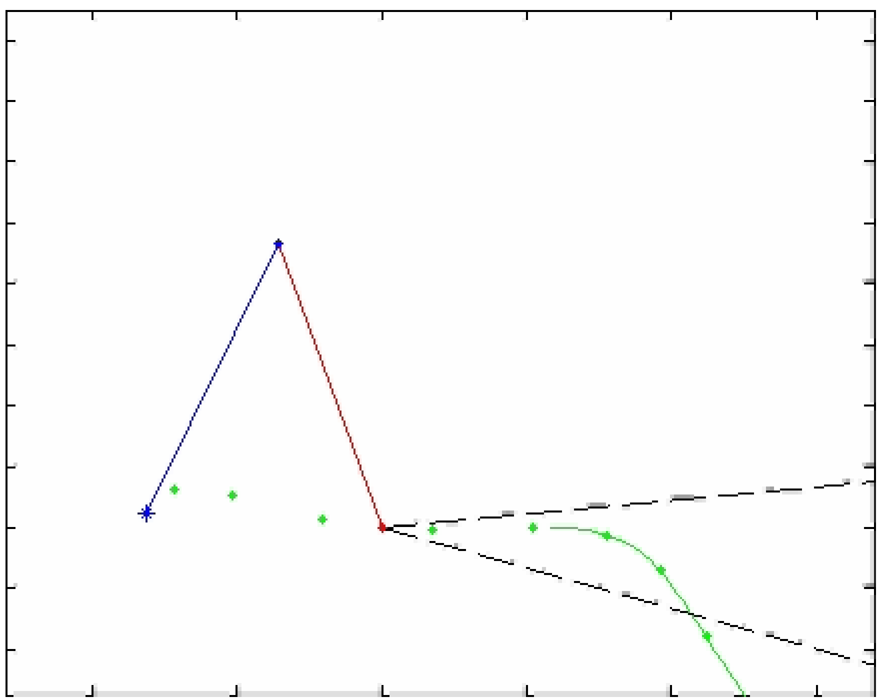
2) $1-|\lambda_2| = r$; $r = 1/m$ ("leakage rate" is the inverse of the MFPT)
   2nd eigenvector renormalized (to exclude failure state) represents the quasi-stationary distribution of the stochastic basin of attraction.

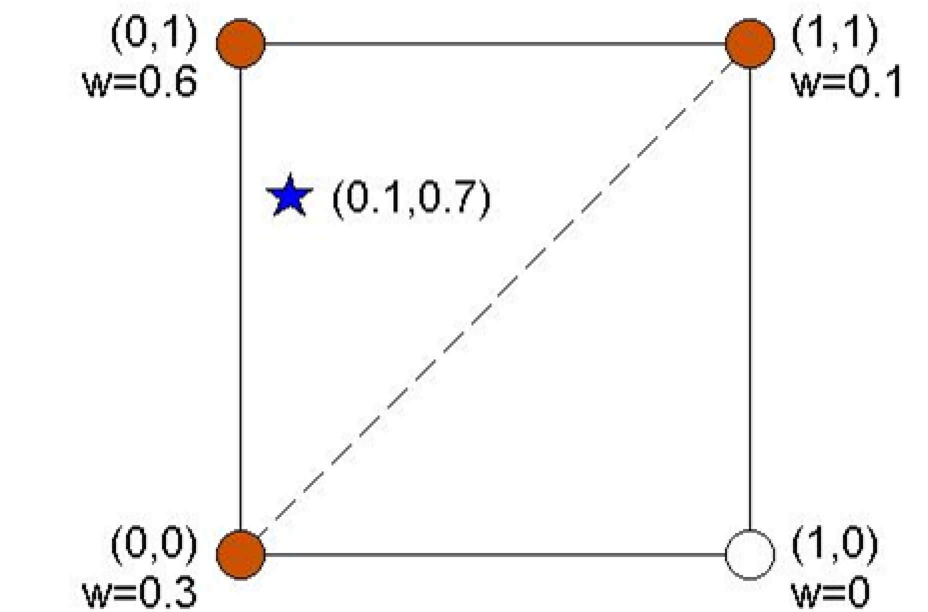
3) λ_3 provides an estimate of "**mixing time**" to forget initial conditions.
   "Fast" mixing implies: $1/\tau_2=log(1/|\boldsymbol{\lambda_2}|) << log(1/|\boldsymbol{\lambda_3}|)= 1/\tau_3$,

so $(1-|\lambda_2|) << (1-|\lambda_3|)$ implies separation of time scales.

Creating the step-to-step transition matrix

- **Discretize (mesh) the state space**
- **For each mesh node, simulate continuous dynamics**
  - Solve for post-impact state for each of many (finite) slopes
- **Use interpolation to approximate each new state**
  - Remesh to improve estimates





Above: ***barycentric*** *interpolation. (Using N+1 out of the 2^N nodes in an N-dimensional box-type element.)*