Motivation dynamics for autonomous systems

Paul Reverdy (joint with Daniel Koditschek, UPenn, and Craig Thompson, UA)
Aerospace and Mechanical Engineering
University of Arizona, USA

Mini-symposium on Nonlinear Decision-Making Dynamics
SIAM DS19
19 May 2019

COLLEGE OF ENGINEERING
Aerospace &
Mechanical Engineering

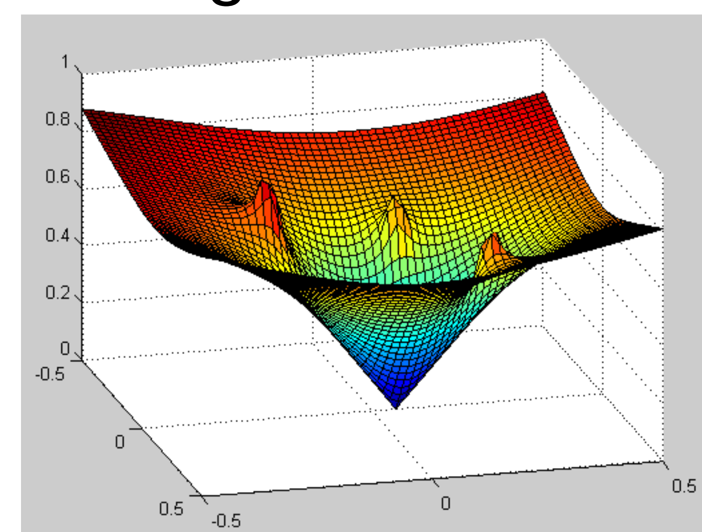
Goal: Autonomy for mobile robots

- Premise: autonomy = appropriately-coordinated behaviors
- Consider navigation as a prototypical behavior (go to a goal set while avoiding obstacles)
- So how to do the composition?
  - Like to encode navigation in vector fields

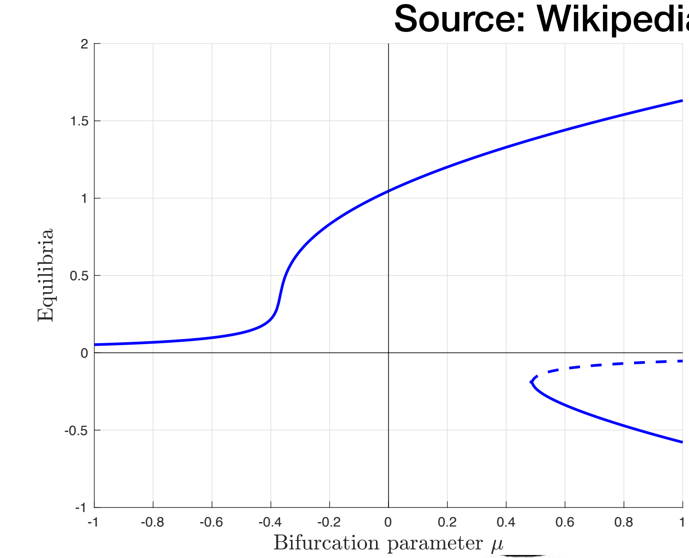
$$F = -\nabla \varphi$$

  - Can we do the same for composition?
- Idea: use pitchfork bifurcation as a switch

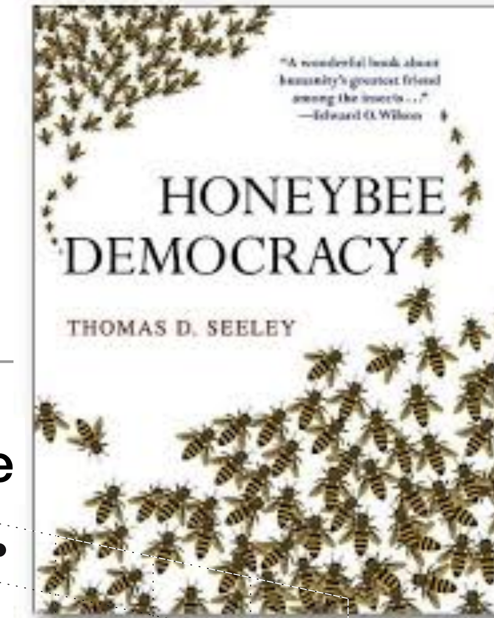
Navigation function



Source: Wikipedia

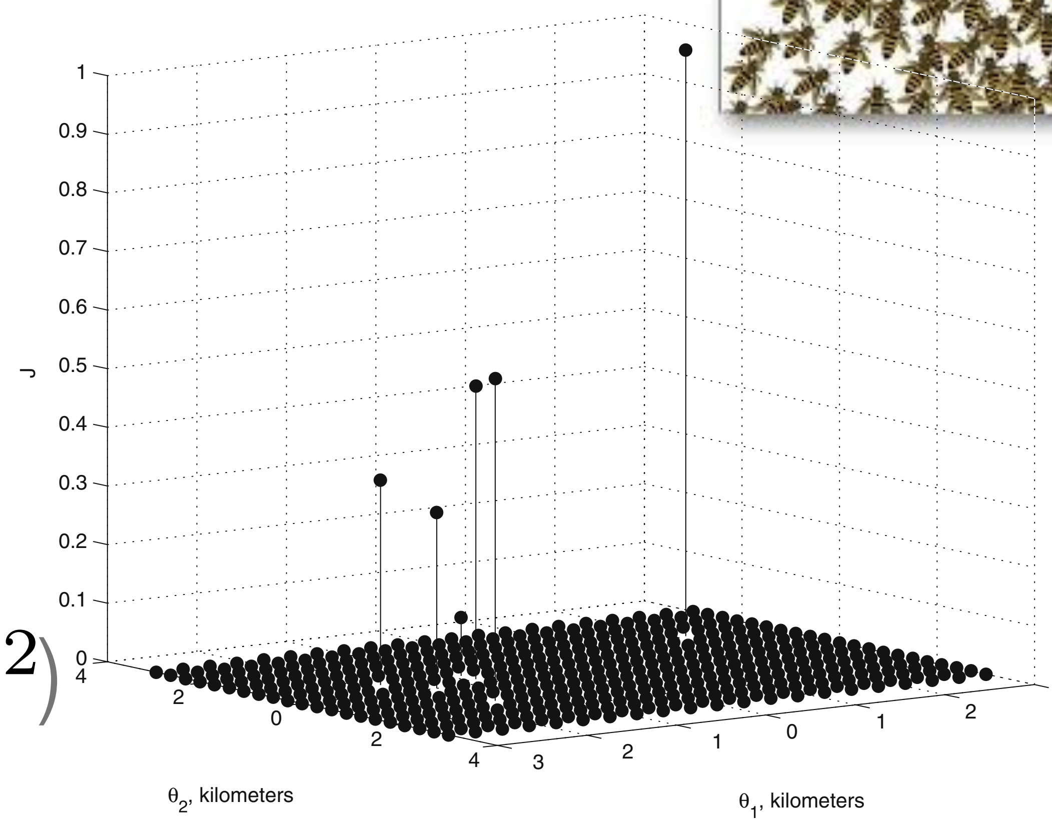


Honeybee Democracy



- Pick nest site
  - With high quality (value, v)
  - Quickly (avoid deadlock)
- Two-site model: (on simplex Δ^2)

Quality landscape

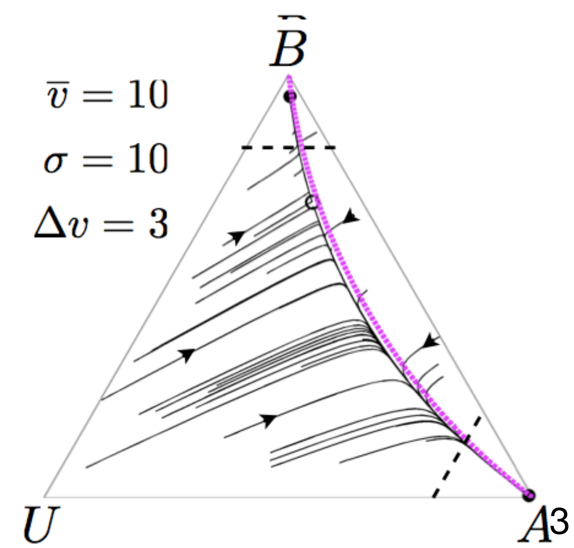


Source: Passino & Seeley

$$\dot{y}_A = \boxed{-\frac{1}{v_A} y_A} + \boxed{v_A y_U (1 + y_A)} - \boxed{\sigma y_A y_B}$$

$$\dot{y}_B = \boxed{-\frac{1}{v_B} y_B} + \boxed{v_B y_U (1 + y_B)} - \boxed{\sigma y_A y_B}$$

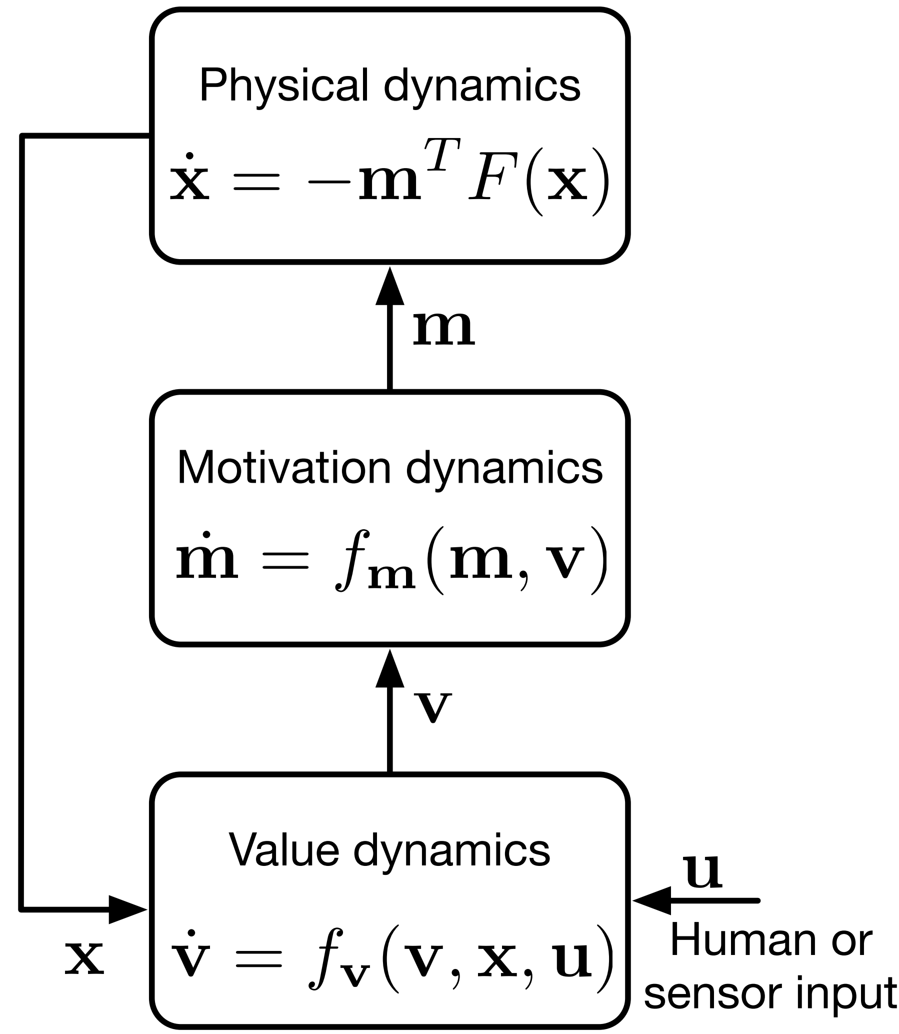
Inhibition Excitation Stop signal



Source: Pais et al.

Motivation system architecture

- Seeley *et al.* model embeds an unfolded pitchfork; converges to high-value option
- Values evolve as tasks are completed
- Physical dynamics are a linear combination of task vector fields
- Appropriate value dynamics yields repetitive two-point patrol



Value dynamics

- N goals (locations), each with navigation functions

$$\varphi_i : \mathcal{D} \to [0,1] \qquad\qquad \varphi_i, i \in \{1, \ldots, N\}$$

- Value $v_i > 0$ with dynamics

$$\dot{v}_i = \boxed{\lambda_i(v_i^* - v_i)} - \boxed{\lambda_i v_i^* (1 - \varphi_i(x))} \qquad \lambda_i, v_i^* > 0$$

Stable growth   Decay at goal

- Motivation state $m = (m_1, \ldots m_N, m_U) \in \Delta^N$

$$\dot{m}_i = v_i m_U - m_i \left(1/v_i - v_i m_U - \sigma(1 - m_i m_U\right))$$

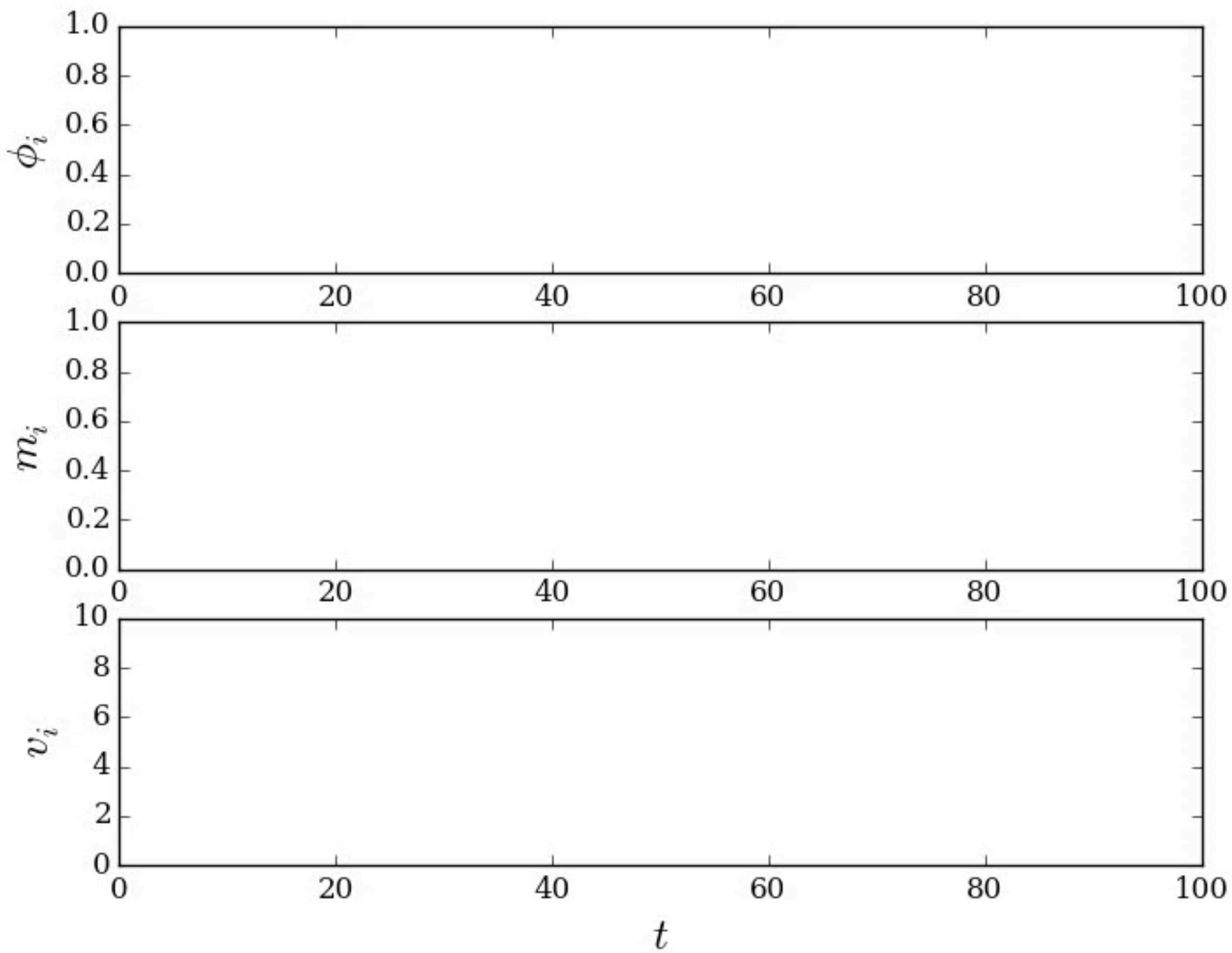
- Physical dynamics

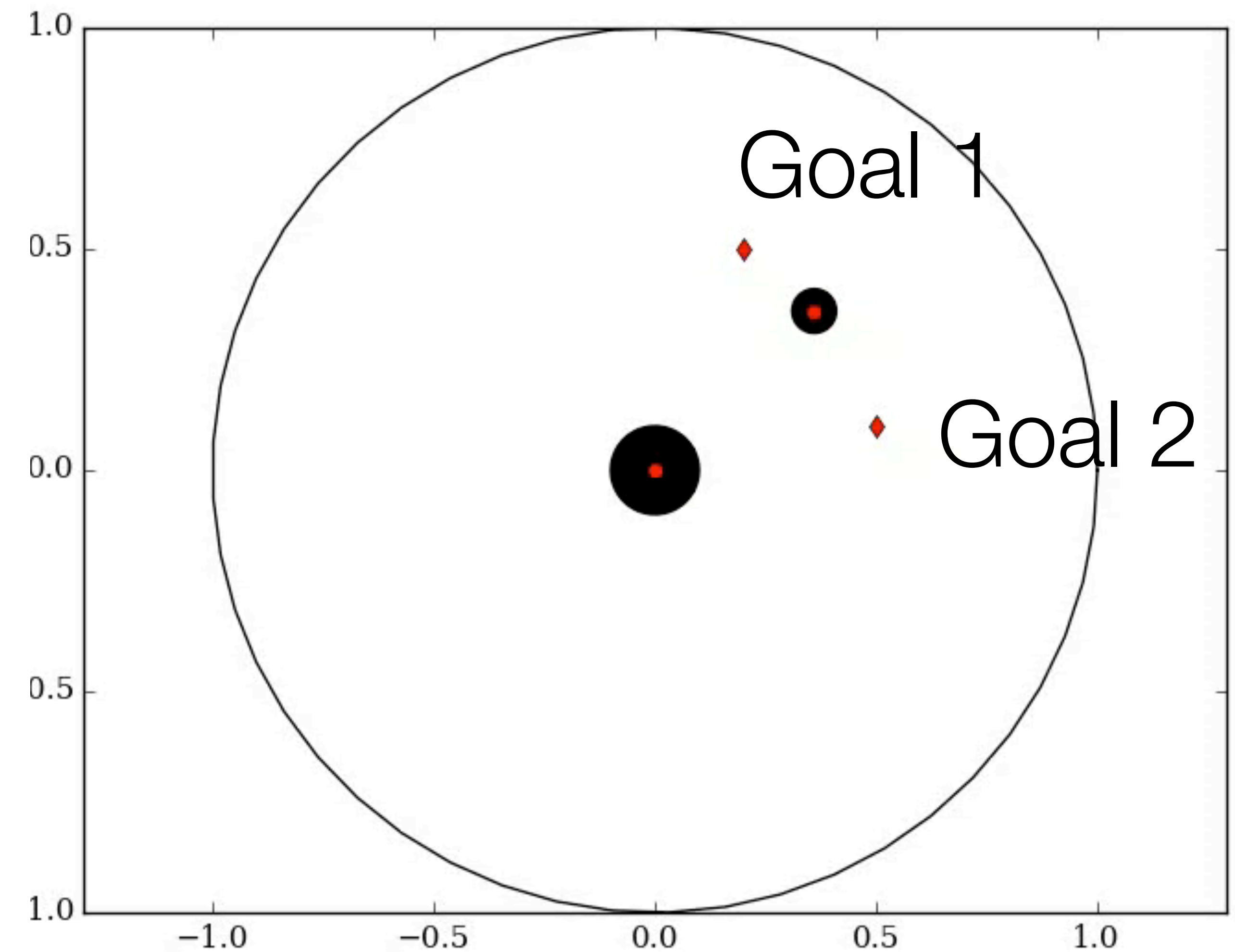
$$\begin{aligned} \dot{\mathbf{x}} &= -m^T D_x \Phi \\ &= -(m_1 \nabla\varphi_1 + \cdots + m_N \nabla\varphi_N) \end{aligned}$$

combination of vector fields

Example

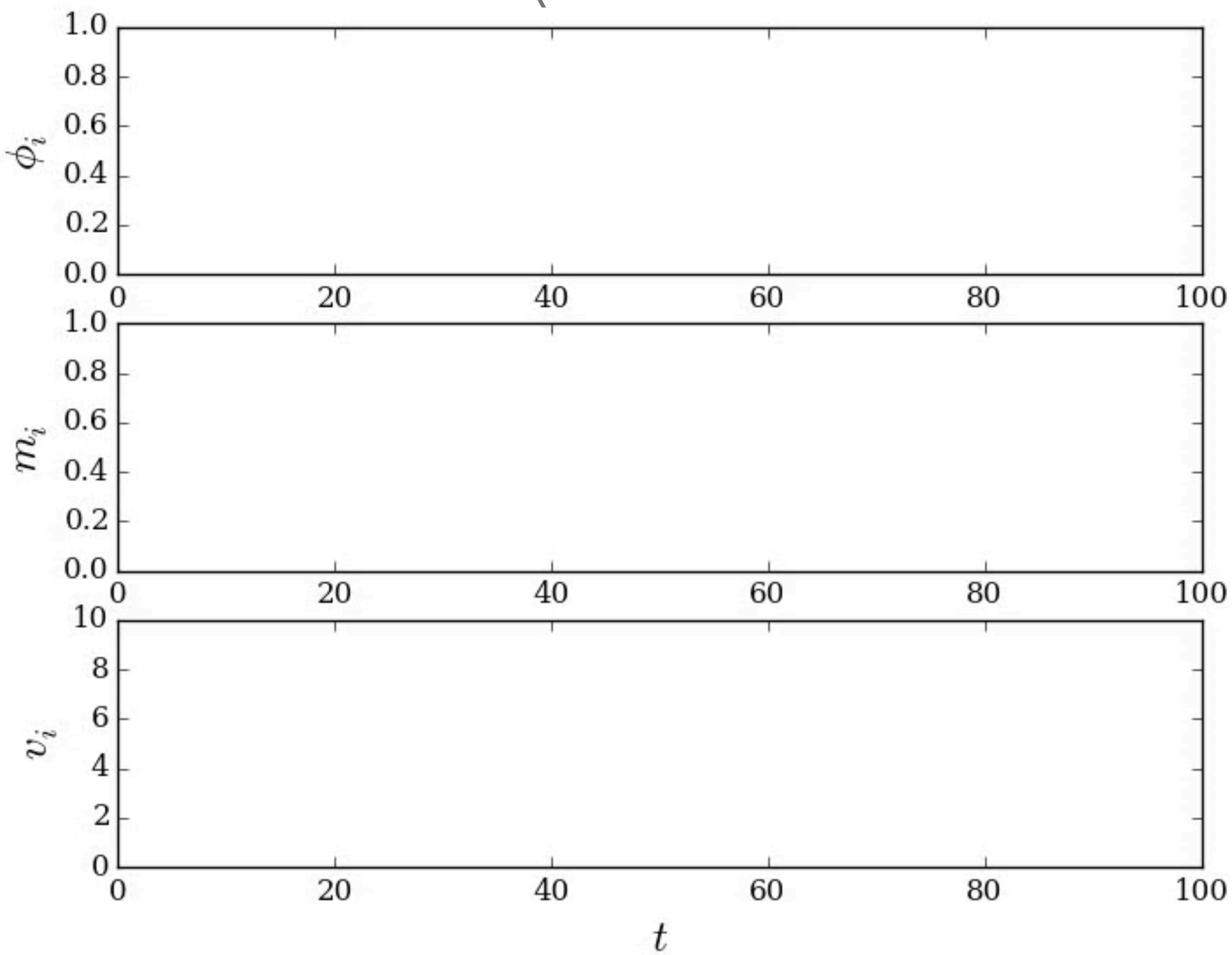
- Numerically, we find a limit cycle

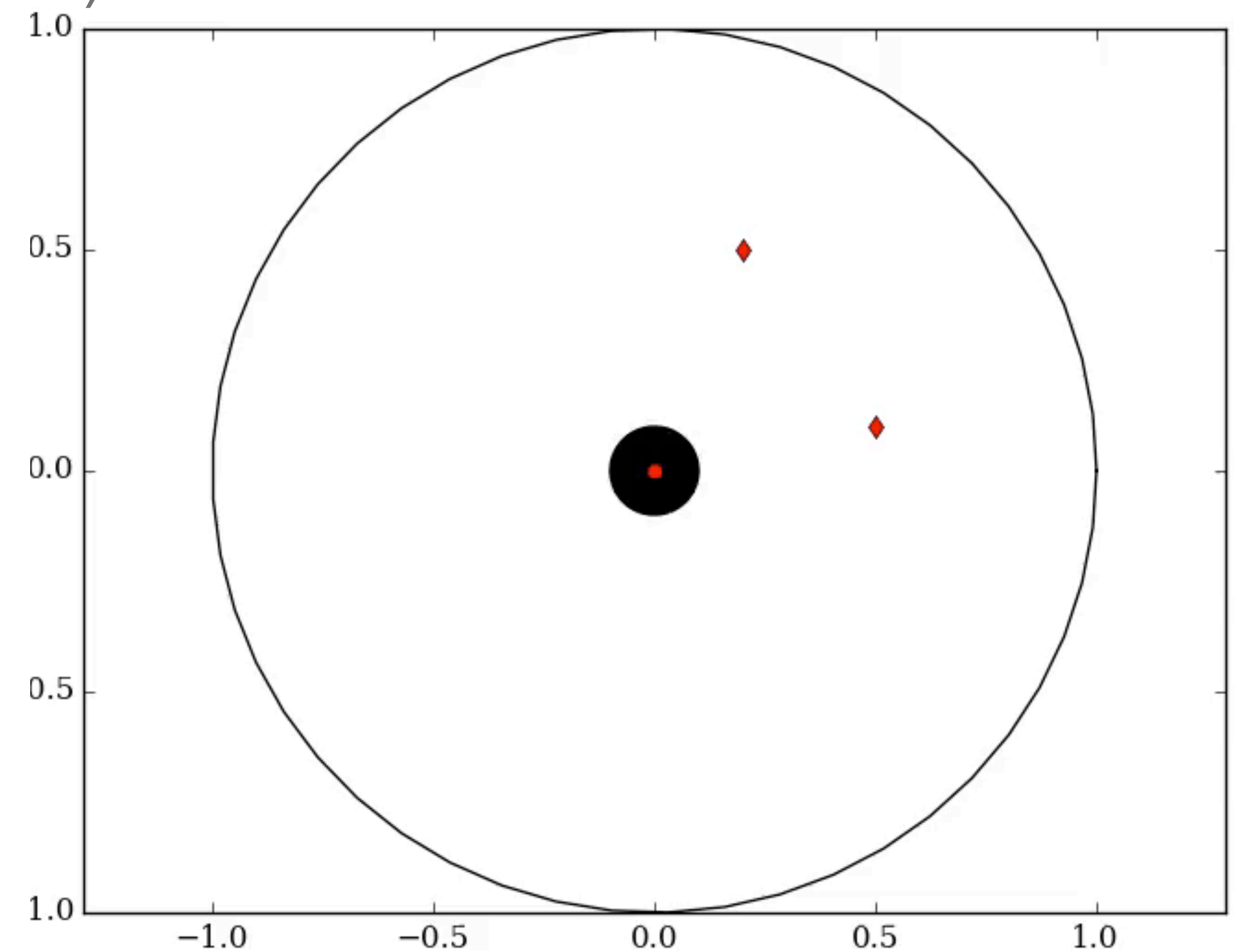




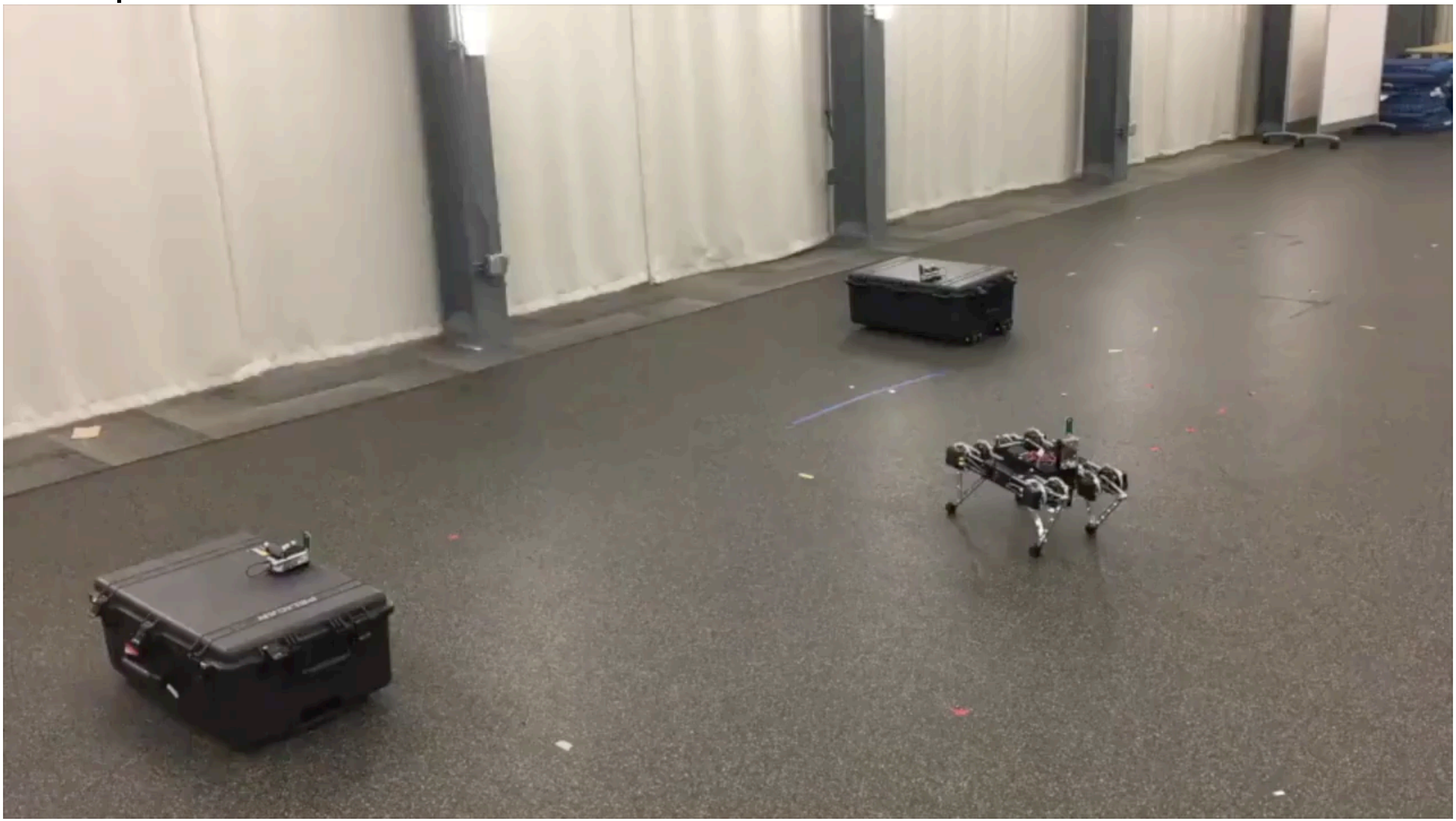
The limit cycle is quite robust!

- Purely reactive: No model of obstacle behavior, just good sensors (and no actuation limits)





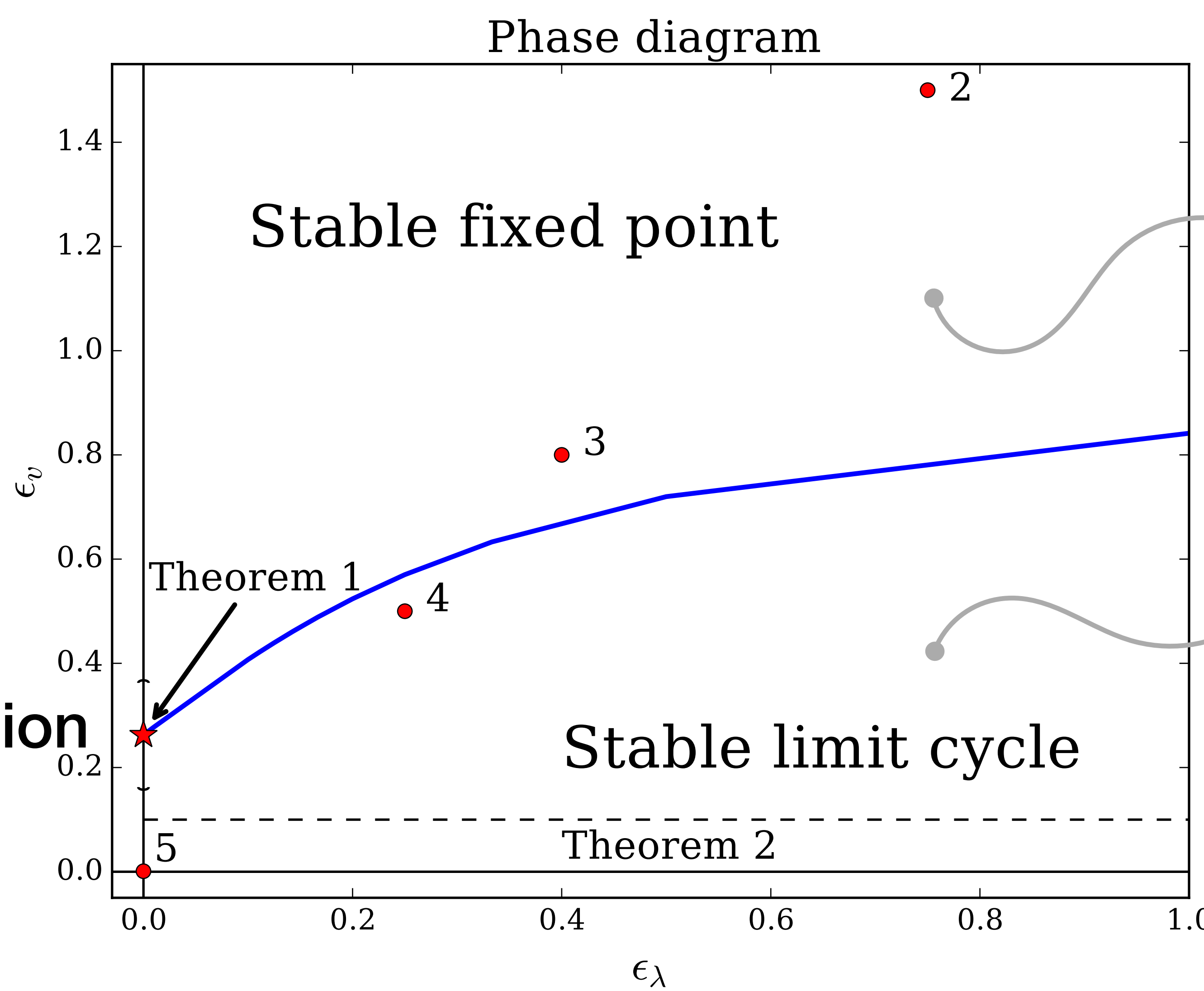
Implementation: it works!



With Vasilis Vasilopoulos, D. E. Koditschek

(application paper under revision)

Analytical results



Hopf bifurcation

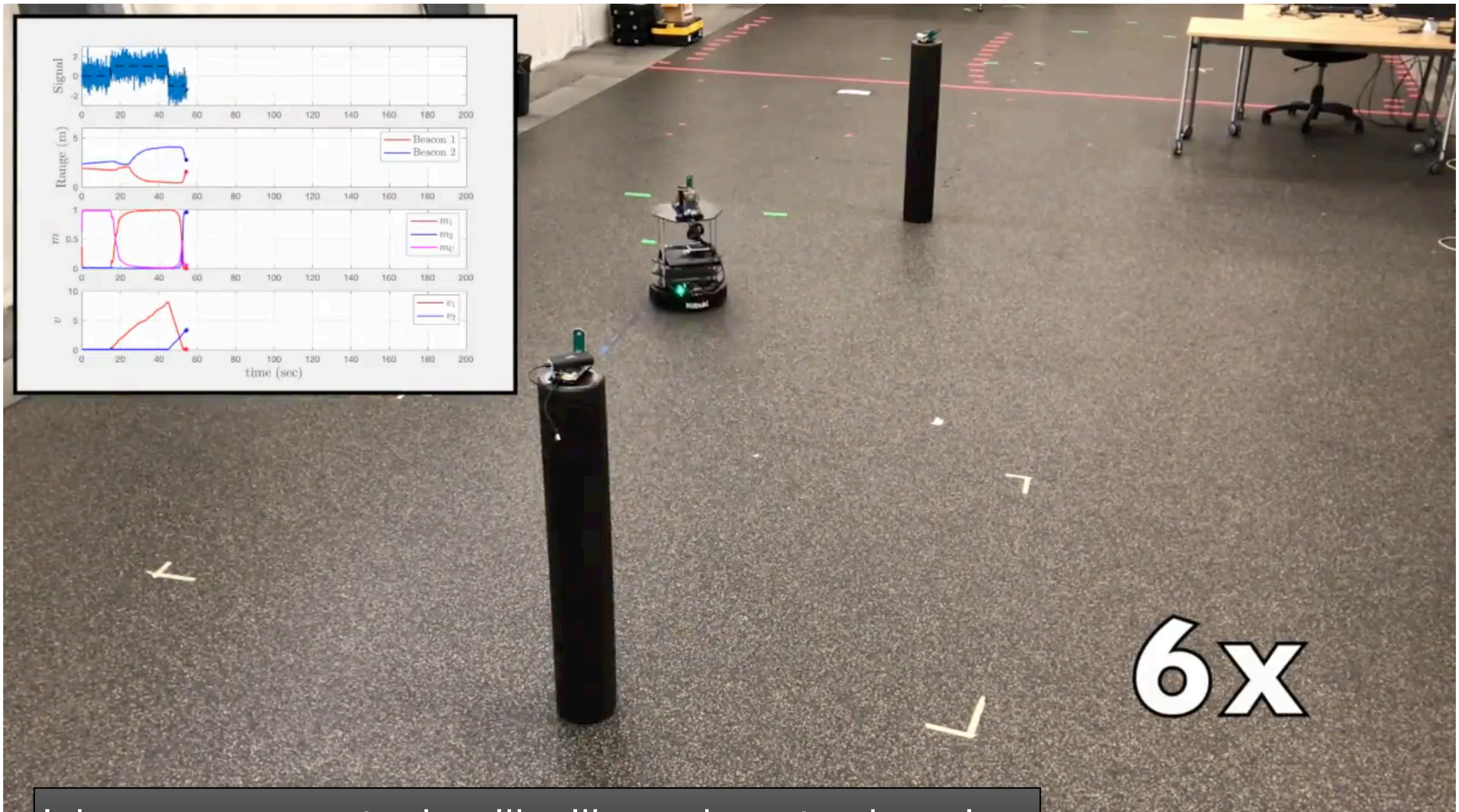
Stable deadlock

Stable recurrent behavior

Geometric singular perturbation

PBR, Kod, *SIAM J. Appl. Dyn. Systems*, 2018

Adding sensors



Idea: use posterior likelihood as task value

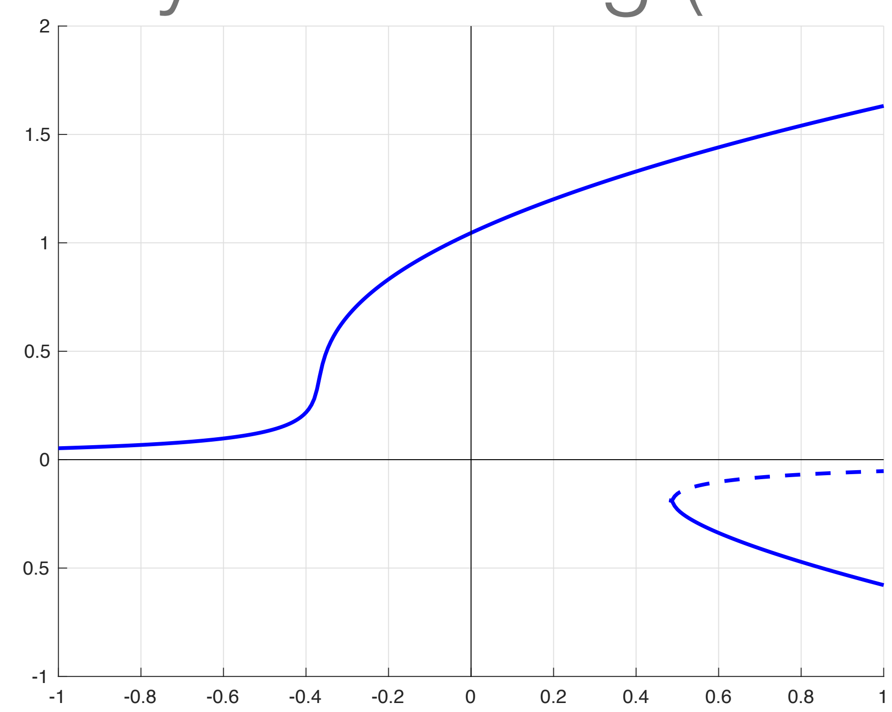
Next steps

- More general tasks (e.g., patrol around a region)

Craig's talk: CP18, Tuesday 3:10 pm

- Control of limit cycle geometry and timing (control the unfolding of the pitchfork)

ACC '19, CDC '19 papers



- Synthesis of general strategies from logical (e.g., LTL) behavioral specifications

Thank you!

preverdy@email.arizona.edu
http://www.paulreverdy.com/

Funding: Air Force Research Laboratory

Craig's talk: CP18, Tuesday 21 May 3:10 pm

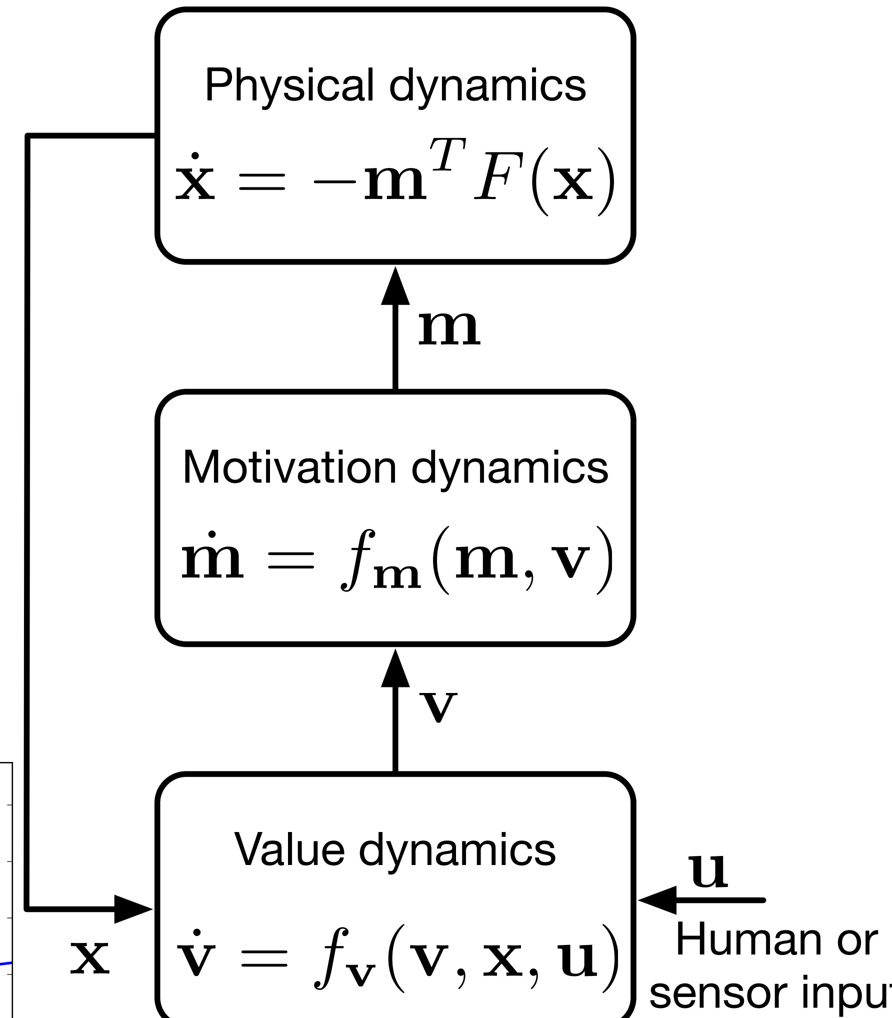
Continuous action selection

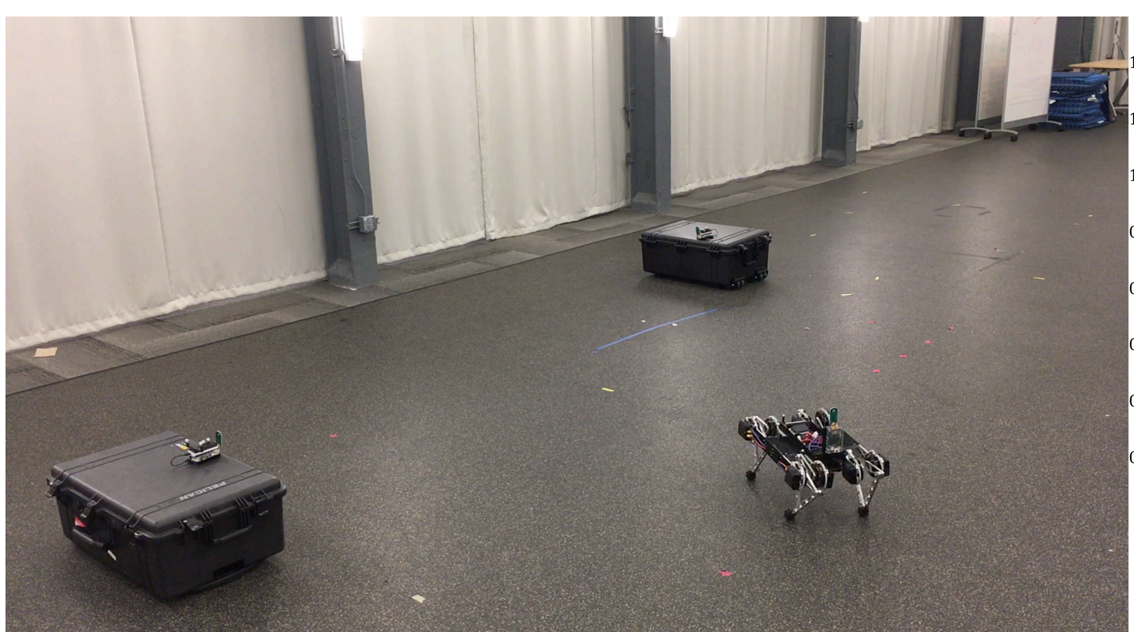
- Simultaneous action selection and movement planning
- Ability to update smoothly
- Dynamics $\dot{x} = m_1 F_1(x) + m_2 F_2(x)$

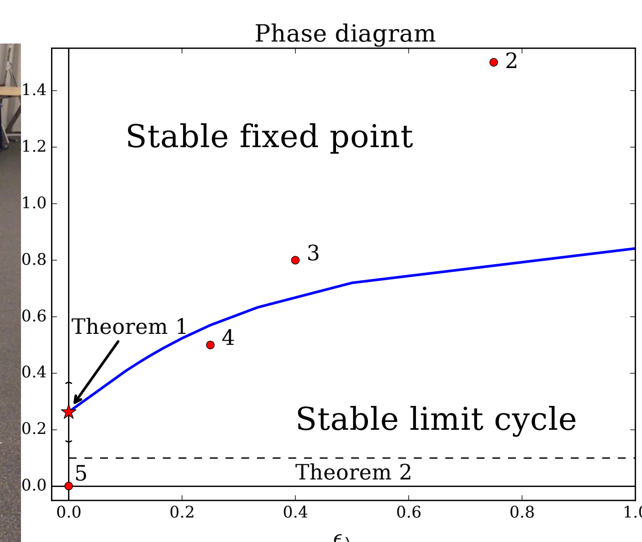
Weights

Vector fields

- Vary weights to pick high-value actions





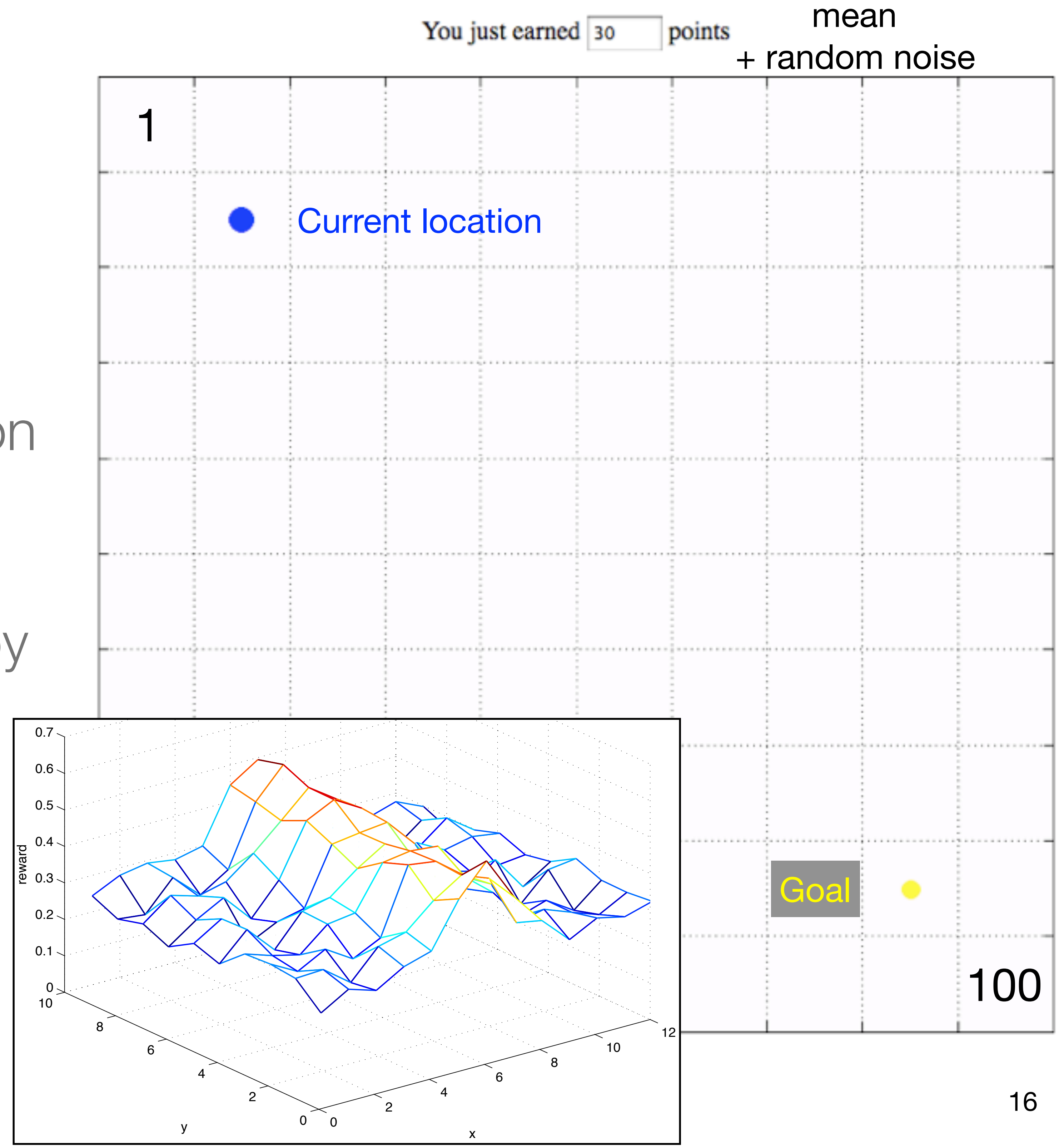


PBR, Koditschek,
SIAM J. Appl. Dyn. Systems (2018)

Additional slides

Grid task: abstraction of spatial search

- Study human behavior in spatial search tasks
- Discretize space
- Earn points based on location (unknown to subject a priori)
- Subject's goal: earn points by navigating through the grid (i.e., find peak quickly)
- Restricted movement or allow jumping in space



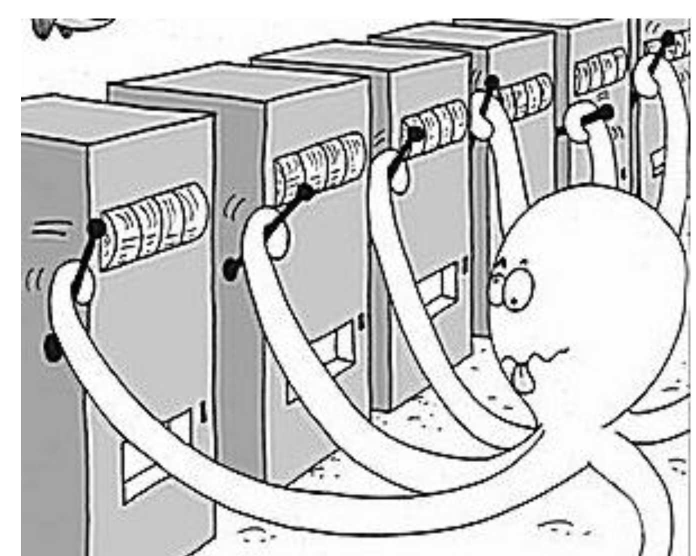
Spatial multi-armed bandit task

The multi-armed bandit problem

- A canonical representation of the *explore-exploit* tradeoff
- N options (arms), indexed by i
- Each arm has an associated distribution $p_i(r)$ with mean m_i (unknown)
- For each sequential decision time $t \in \{1, \dots, T\}$, pick arm i_t, receive reward $r_t \sim p_{i_t}(r)$
- Objective: maximize cumulative expected reward

$$\max_{\underset{\uparrow}{\{i_t\}}} J, \quad J = \mathbb{E}\left[\sum_{t=1}^{T} r_t\right]$$

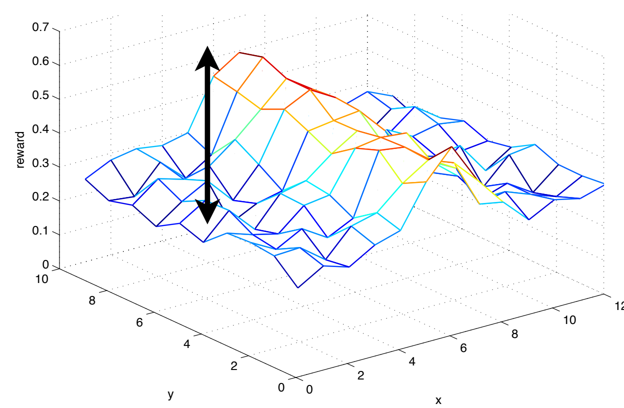
Sequential decisions



research.microsoft.com

Regret

- Bounds on optimal performance more easily formulated in terms of *regret*:
- Define $m_* = \max_i m_i$ and $R_t = m_* - m_{i_t}$, expected regret at time t



- Objective: minimize cumulative expected regret (analytical quantity)

Omniscient optimal

Mean value of decisions made

$$J_R = \sum_{t=1}^{T} R_t = T m_* - \sum_{t=1}^{T} m_{i_t}$$

Sum over decisions

$$= \sum_{i=1}^{N} \Delta_i \mathbb{E}[n_i^T]$$

Sum over options

$\Delta_i = m_* - m_i$: Expected regret

n_i^T : Number of times option i chosen

Bounds on optimal performance

- A fundamental result of Lai and Robbins (1985) shows

$$p_i = \mathcal{N}(m_i, \sigma_s^2)$$
$$p_{i^*} = \mathcal{N}(m_{i^*}, \sigma_s^2)$$

$$\mathbb{E}\left[n_i^T\right] \geq \left(\frac{1}{D(p_i||p_{i^*})} + o(1)\right) \log T$$

Horizon

$$D(p_i||p_{i^*}) = \frac{\Delta_i^2}{2\sigma_s^2}$$

Kullback-Liebler divergence

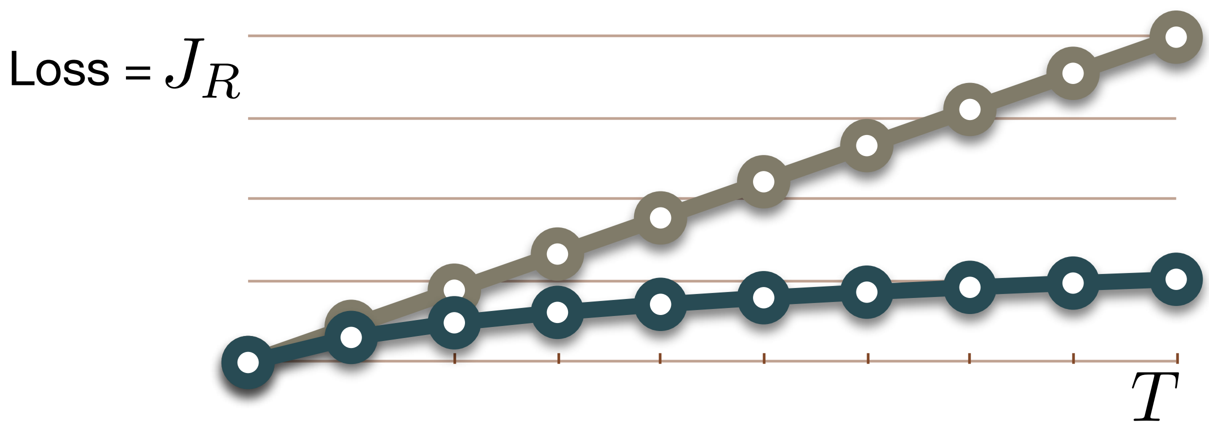
- So regret grows at least logarithmically in time:

$$J_R(T) \geq \mathcal{C} \log T$$

- Lai-Robbins is an asymptotic result; the literature seeks uniform bounds (in T)
- Uniform logarithmic regret is considered optimal

$$J_R(T) < \mathcal{C}' \log T$$

$\mathcal{C}'$, $\mathcal{C}$ differ by a constant factor

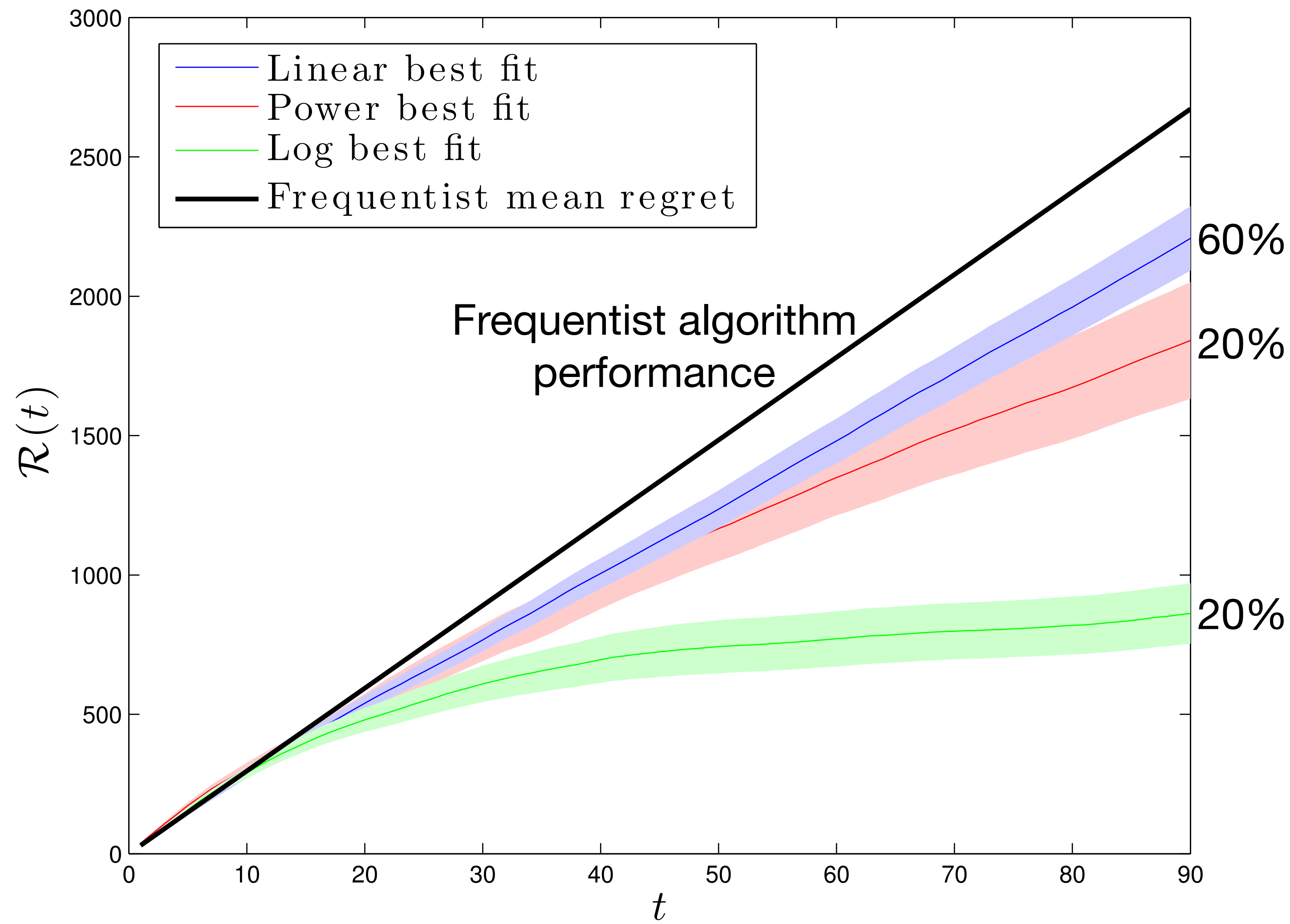


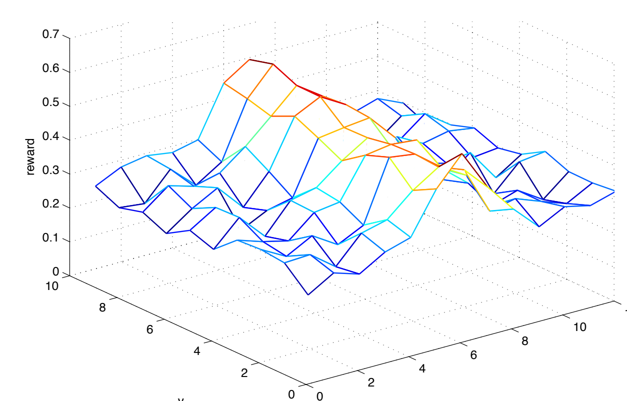
Observed human performance phenotypes

- Data from grid task; short horizon
- Fit models to observed regret:

$$\mathcal{R}(t) = a + bt$$
$$\mathcal{R}(t) = at^b$$
$$\mathcal{R}(t) = a + b\log t$$

- This set of models captures most observed performance
- Some people display logarithmic regret: "optimal" performance!
- Can we capture these three classes in a model?





Reverdy et al. *Proc. IEEE* (2014)

The Upper Credible Limit Algorithm (UCL)

Covariance belief: smoothness e.g., length scale λ

- Prior belief $\mathbf{m} \sim \mathcal{N}(\boldsymbol{\mu_0}, \Sigma_0)$ Update: Kalman filter, no dynamics

Mean reward values

Mean belief

ΔI_i^t Info gain

A Ambiguity bonus: value of information

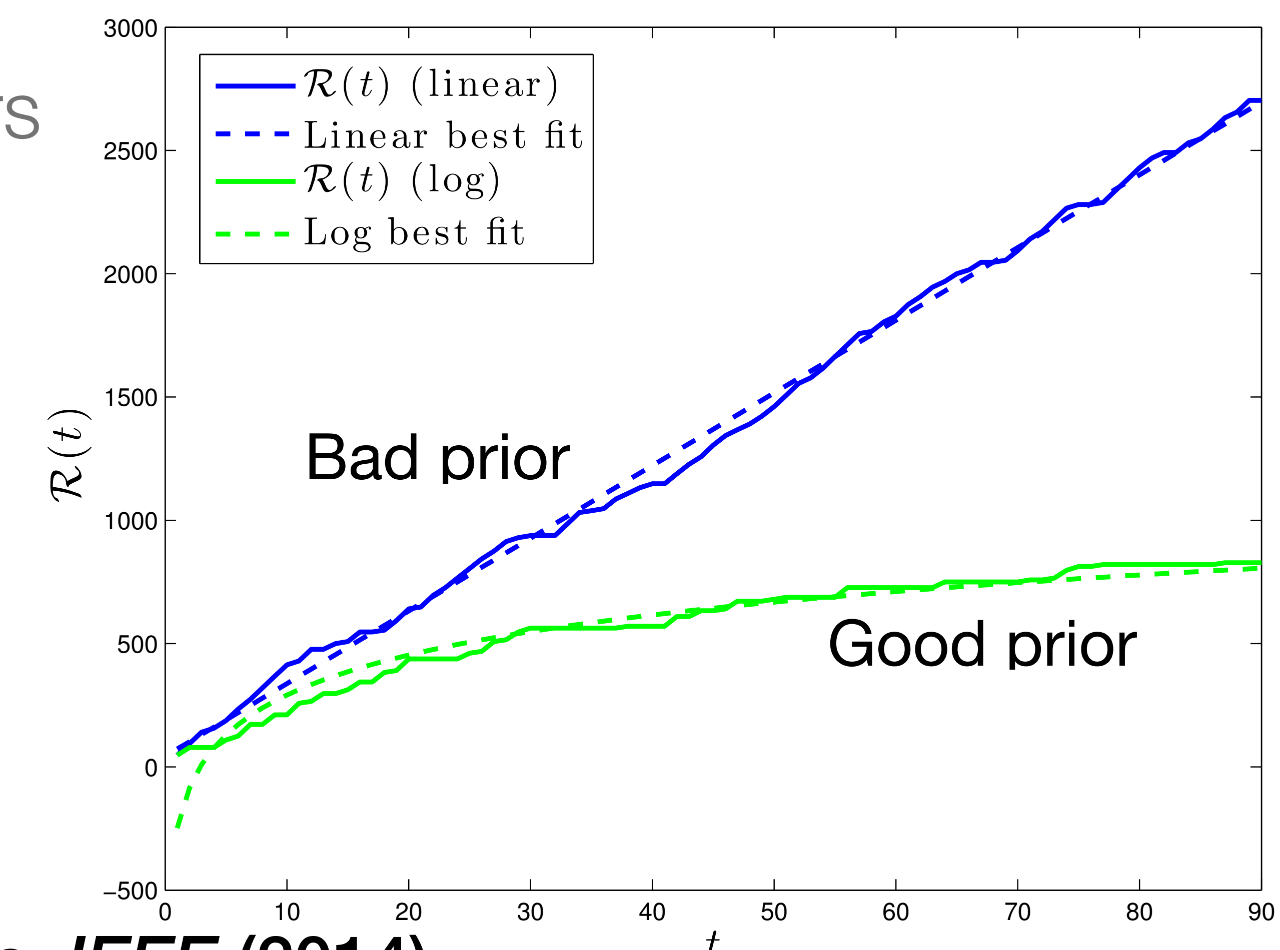
- Heuristic $$Q_i^t = \mu_i^t + \underbrace{\sigma_i^t \Phi^{-1}(1 - \alpha_t)}_{C_i^t}$$

C_i^t Uncertainty

- For $\alpha_t = 1/(\sqrt{2\pi e}t)$, achieve logarithmic regret for good priors

- And linear regret for bad priors

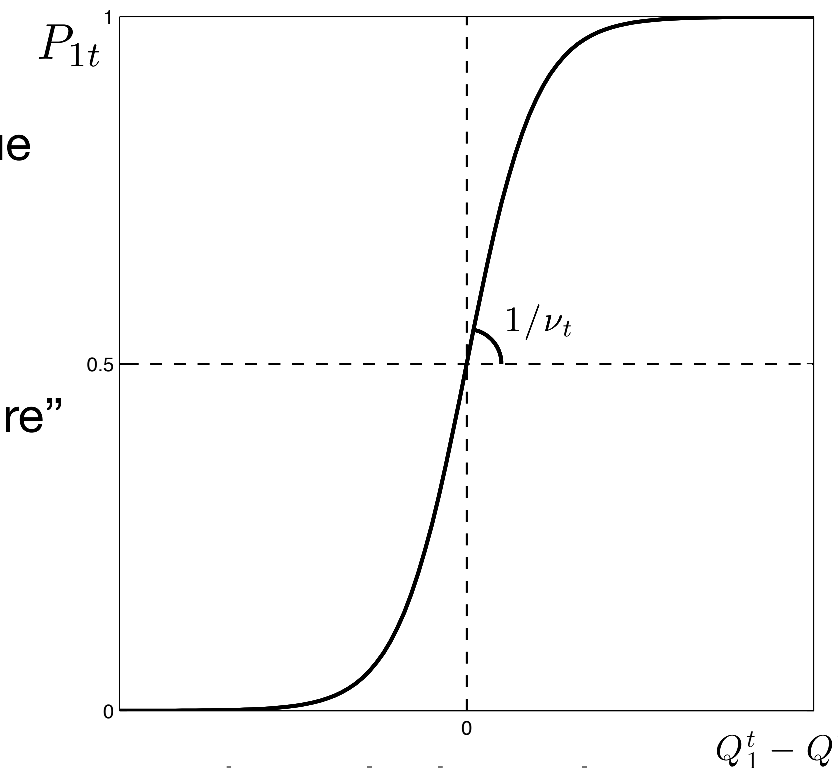
- Prior quality depends on accuracy and certainty



Reverdy et al. *Proc. IEEE* (2014)

Stochastic UCL

- Human decision making is stochastic, so extend UCL to stochastic policies



- Use Boltzmann/softmax action selection

Heuristic value

$$P_{it} = \frac{\exp(Q_i^t/\upsilon_t)}{\sum_{j=1}^{N} \exp(Q_j^t/\upsilon_t)}$$

Selection probability

"Temperature"

- Use dynamic temperature parameter

$$\upsilon_t = \frac{\Delta Q_{\min}^t D}{2 \log t}$$

where $\Delta Q_{\min}^t = \min_{i \neq j} |Q_i^t - Q_j^t|$ is the minimum gap between heuristic values, $D > 0$

- Stochastic UCL achieves logarithmic regret with a slightly larger constant

- But gains potential robustness to wrong priors

Reverdy et al. *Proc. IEEE* (2014)

Parameter estimation for UCL

- Have a model; need an observer
- Stochastic UCL defines a maximum likelihood estimator; requires solving hard non-convex optimization problem
- If the heuristic is a linear function of the unknown parameters, we get a generalized linear model (GLM)

$$P_{it} = \frac{\exp(\theta^T \mathbf{x}_i^t)}{\sum_{j=1}^{N} \exp(\theta^T \mathbf{x}_j^t)}$$

- Reduces to convex problem$\Rightarrow$ estimators with provable convergence
- Can be applied to stochastic UCL via linearization

Reverdy and Leonard, *TASE* 2016

Parameter estimates

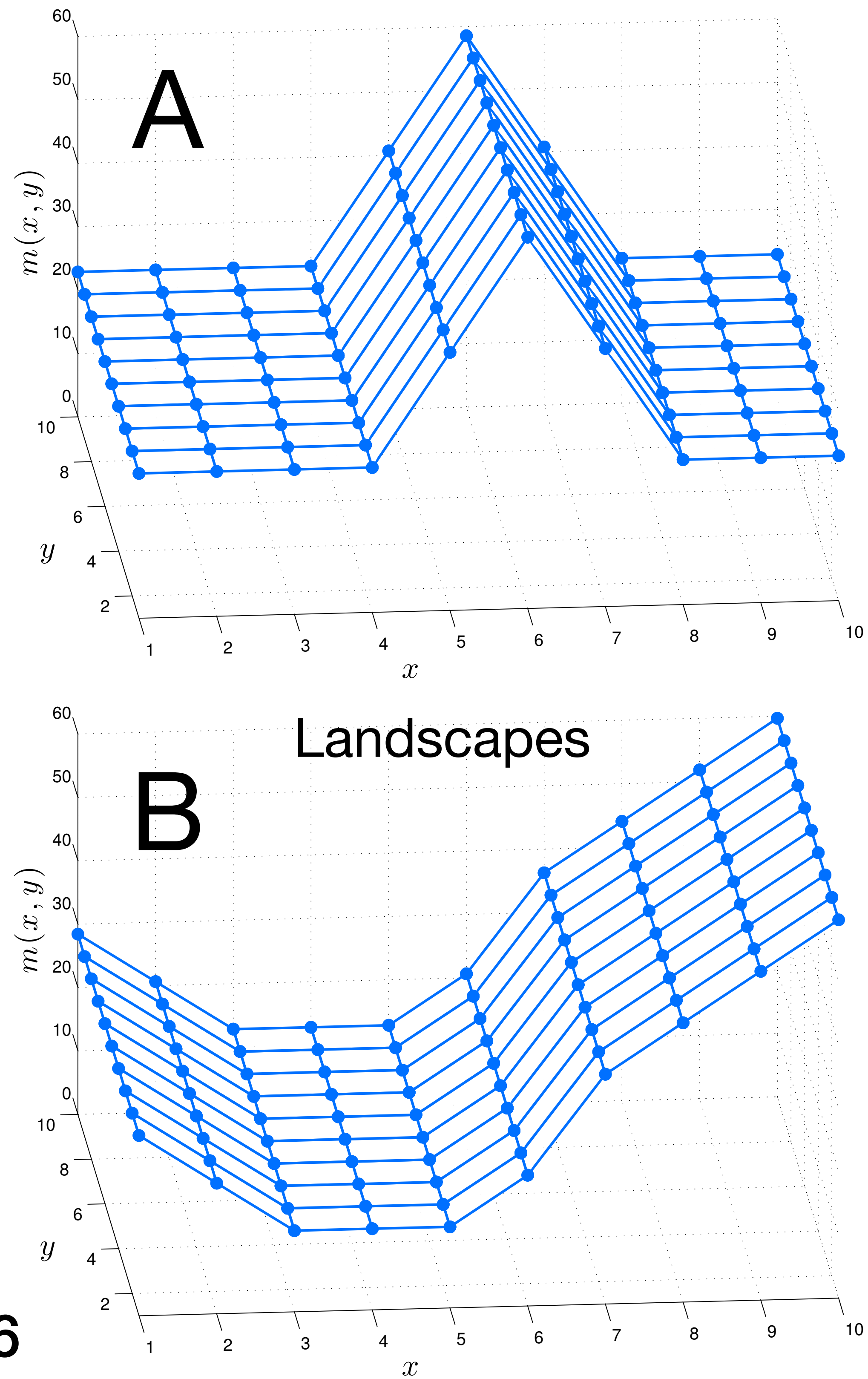
- Data from subjects with high performance
- Use GLM-based estimator
- Find statistically-significant difference between parameters for different landscapes
- Evidence for adapted strategies/priors

ν	25.5
μ_0	25.3
σ_0^2	3.32E+05

53 subjects

ν	29.5
μ_0	6.08
σ_0^2	3.35E+05

17 subjects



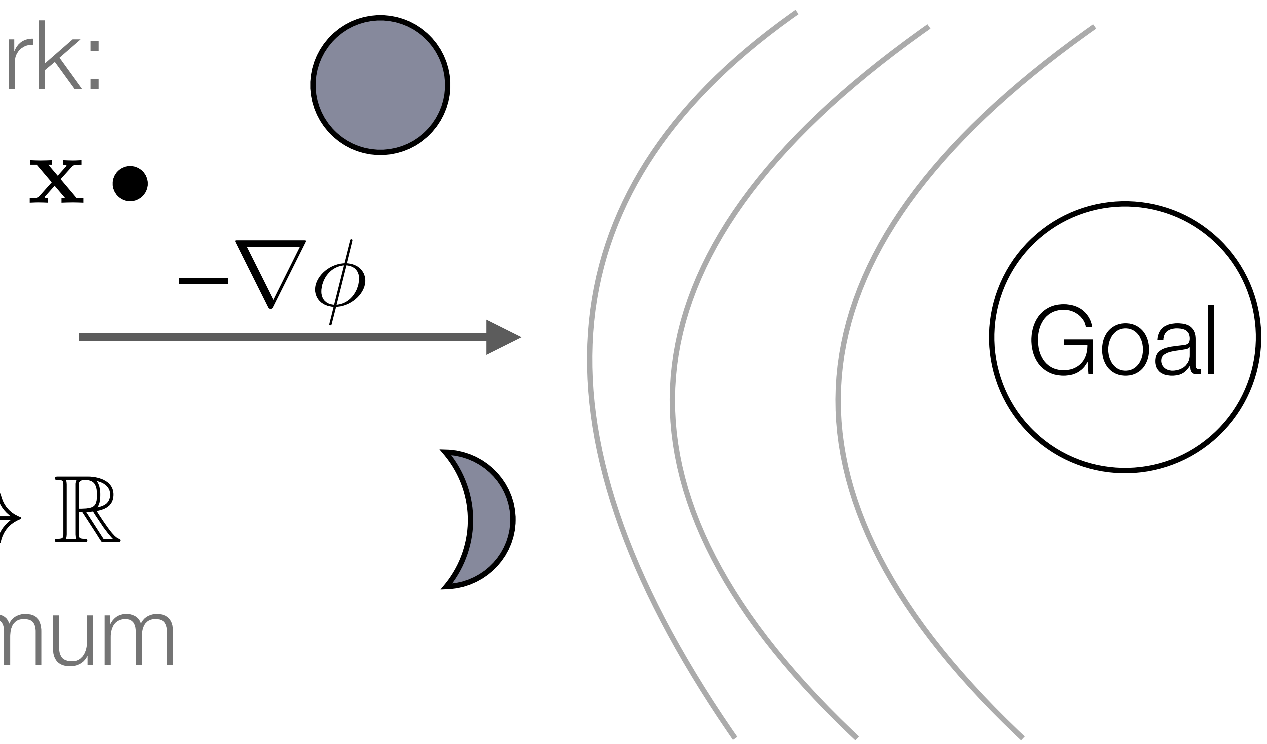
PR and Leonard, *TASE* 2016

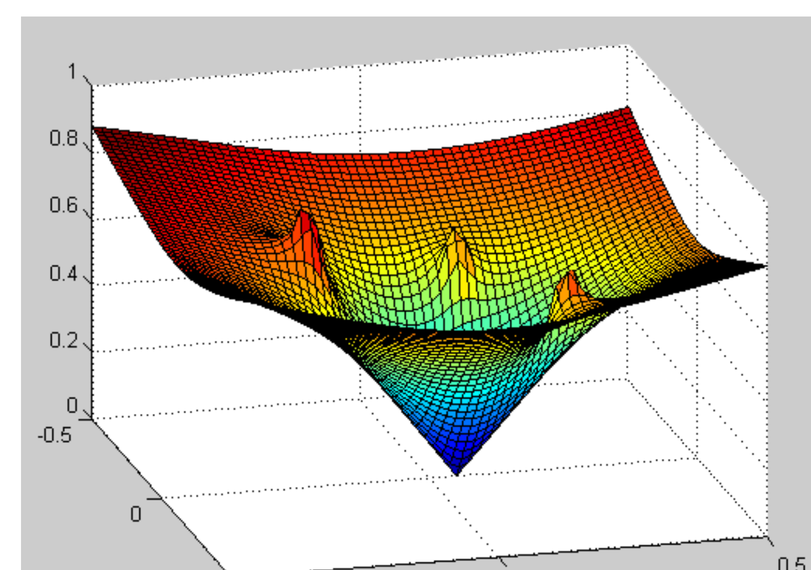
Navigation: a prototypical task

- Navigation function framework:
  - $\mathbf{x}(t) \in \mathcal{D} \subseteq \mathbb{R}^2$
  - Potential function $\phi : \mathcal{D} \to \mathbb{R}$ differentiable, unique minimum
  - Task: $\lim\limits_{t \to +\infty} \mathbf{x}(t) = \arg\min\limits_{\mathbf{x}} \phi(\mathbf{x})$
  - Ideal dynamics:

$\dot{\mathbf{x}} = -u\nabla\phi, u \in \mathbb{R}_{++}$ (~potential flow)

Cf. Lyapunov functions





Source: Wikipedia

A simple multi-goal task

- Say the robot has several goals

- Task: stay close to all of them
- Let $f_i(x)$ measure distance to each goal; close = $f_i(x) \leq \epsilon$
- Pose as a constraint satisfaction problem:

$$\min_{x \in X} \; 0$$
$$\text{s.t.} \; f(x) \leq 0$$

- Solve using saddle-point algorithm

Optimization problem

- Suppose $f_0 : X \subseteq \mathbb{R}^n \to \mathbb{R}$ is an objective function
- N constraints $f_i : X \subseteq \mathbb{R}^n \to \mathbb{R}, i \in \{1, \ldots, N\}$
- Solve problem

$$\min_{x \in X} \; f_0(x)$$

$$\text{s.t. } f(x) \leq 0$$

- Introduce Lagrange multipliers $\lambda \in \Lambda = \mathbb{R}^N_+$ and define the Lagrangian

$$\mathcal{L}(x, \lambda) = f_0(x) + \lambda^T f(x)$$

Nonlinear dynamics can yield limit cycles

- Seek a new system for Lagrange multiplier dynamics
- Specialize to $N = 2$ constraints, use bio-inspired dynamics from Passino and Seeley, 2012:

$$\dot{y}_1 = -1/(Kf_1(x)) + Kf_1(x)y_0(1+y_1) - \sigma y_1 y_2$$
$$\dot{y}_2 = -1/(Kf_2(x)) + Kf_2(x)y_0(1+y_2) - \sigma y_1 y_2$$
$$y \in \Delta^2 = \{x \in \mathbb{R}^3 : x_i \geq 0, \sum_i x_i = 1\} \quad K, \sigma > 0$$

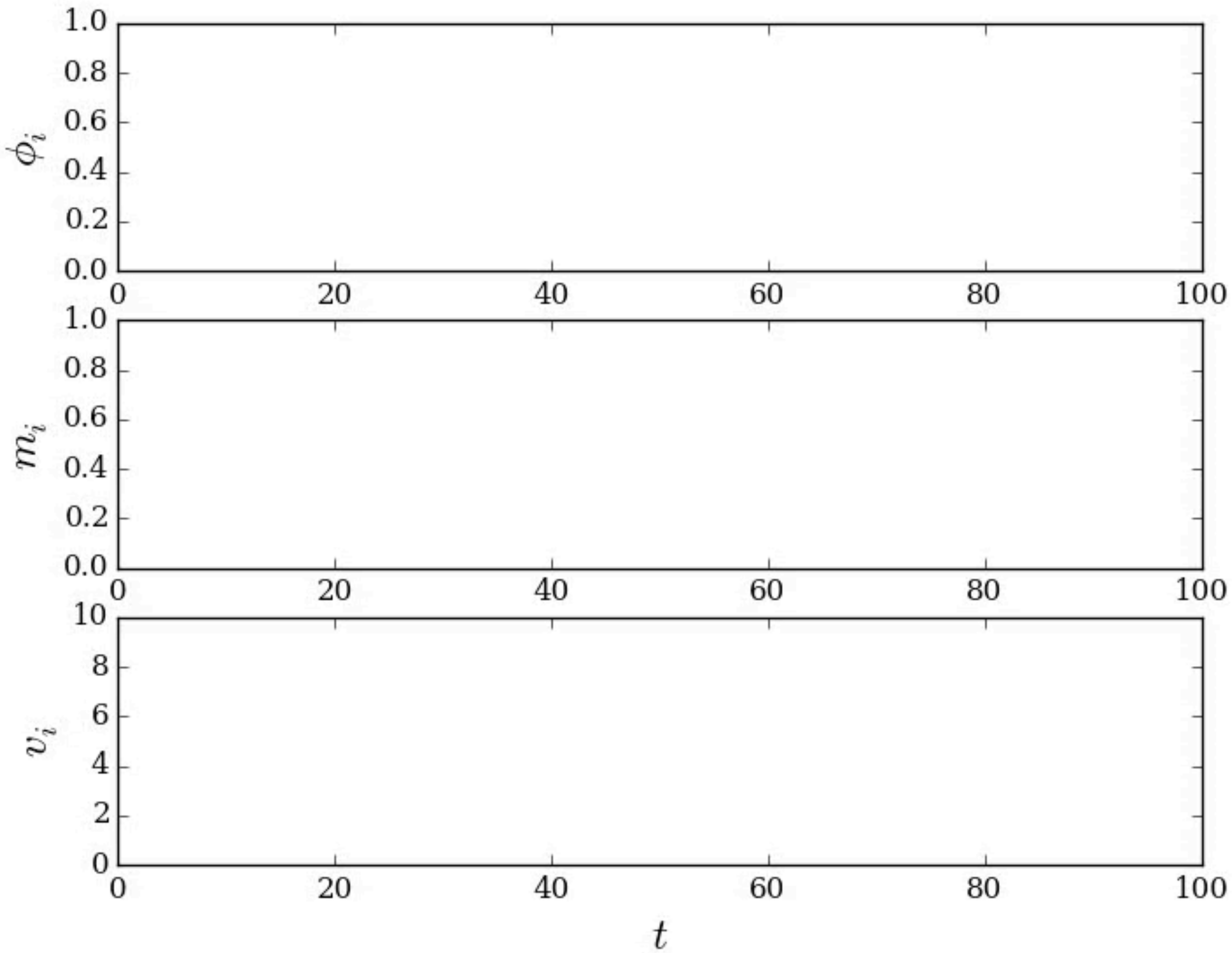
- Same decision variable dynamics as saddle point:

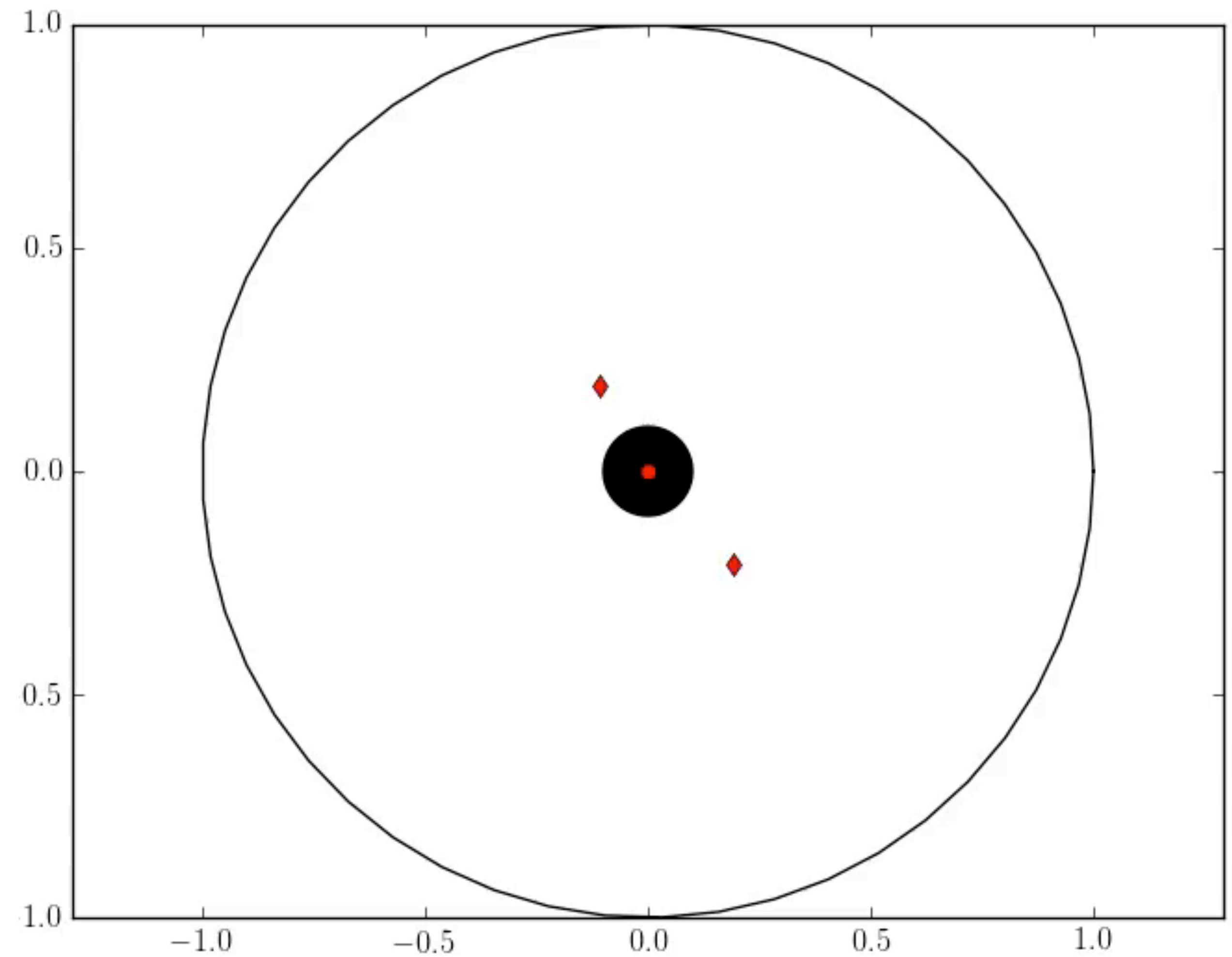
$$\dot{x} = y_1 f_{1,x}(x) + y_2 f_{2,x}(x)$$

- This system can exhibit limit cycles!

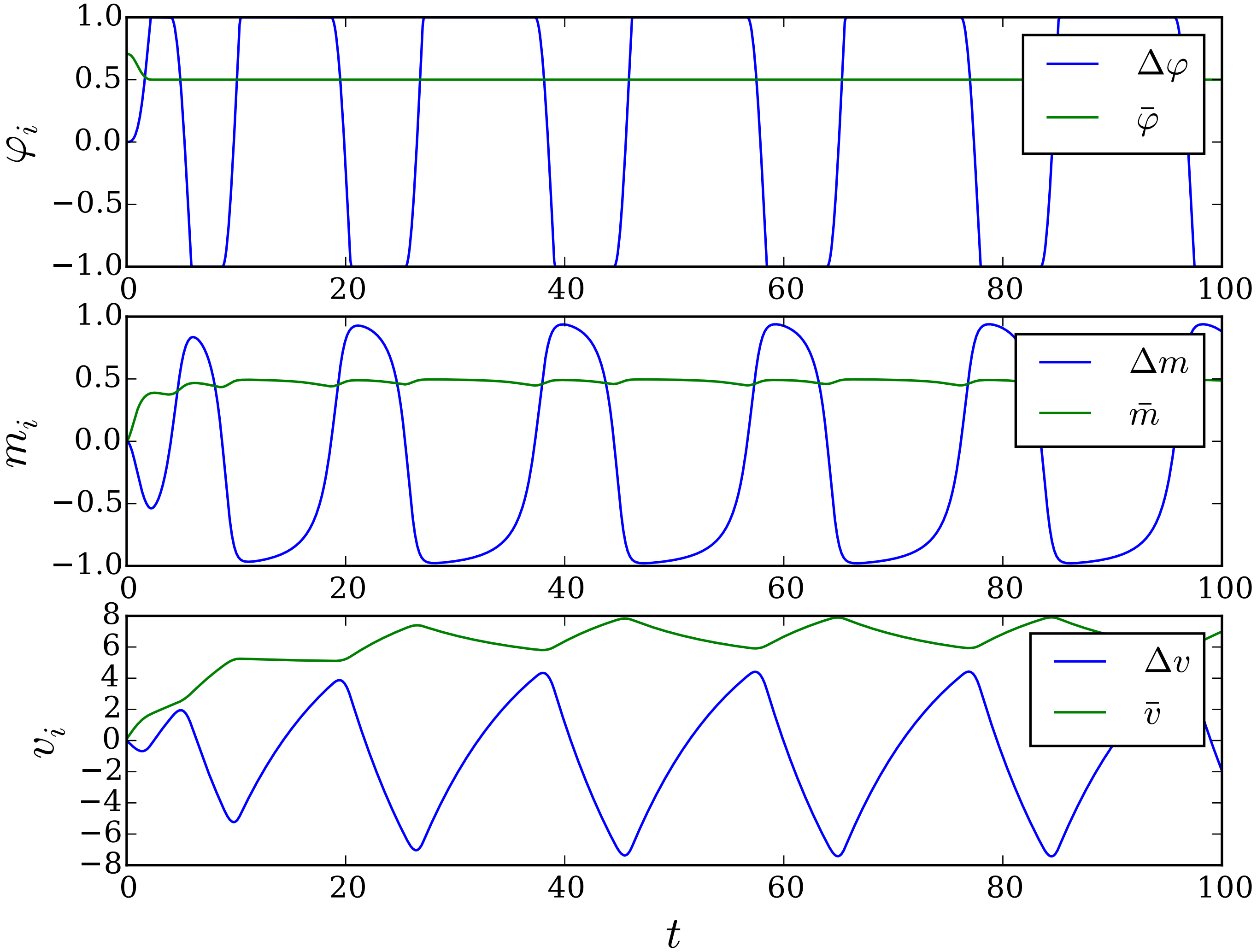
The limit cycle is quite robust! (2)

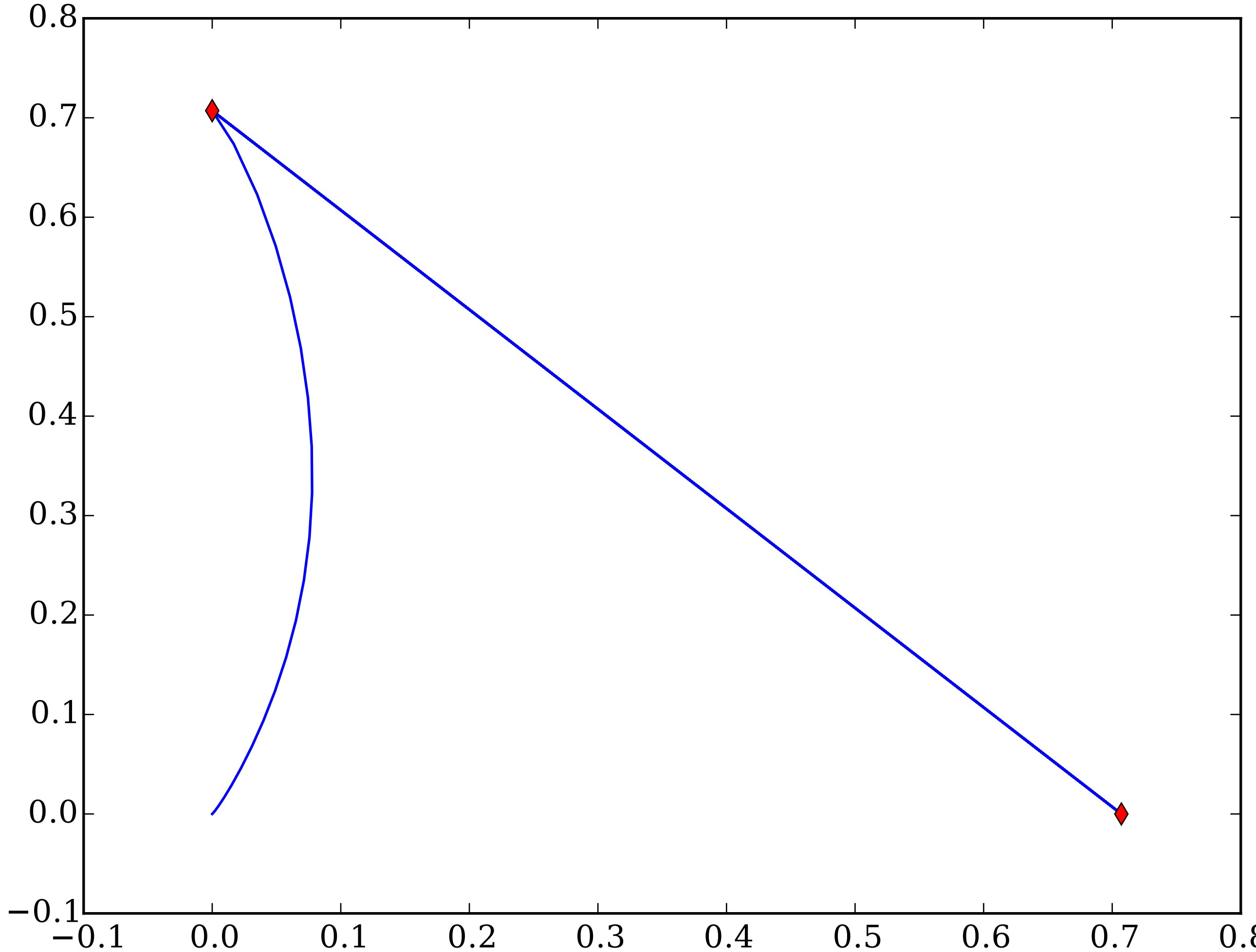
- Purely reactive: No model of goal behavior, just good sensors (and no actuation limits)





Analysis: Shift to mean-difference coordinates





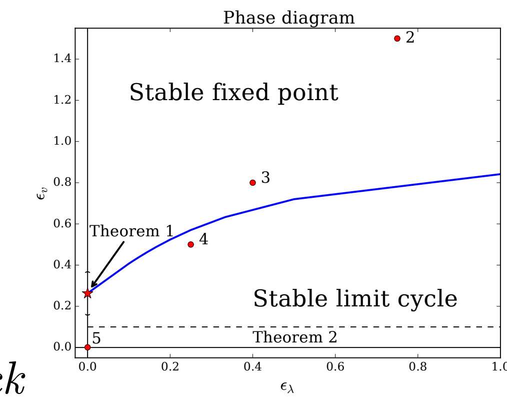
- Remove obstacles, workspace is $\mathbb{R}^2$

- Consider mean-difference coordinates, e.g.,

$$\Delta\varphi = \varphi_1 - \varphi_2, \bar{\varphi} = \frac{\varphi_1 + \varphi_2}{2}$$

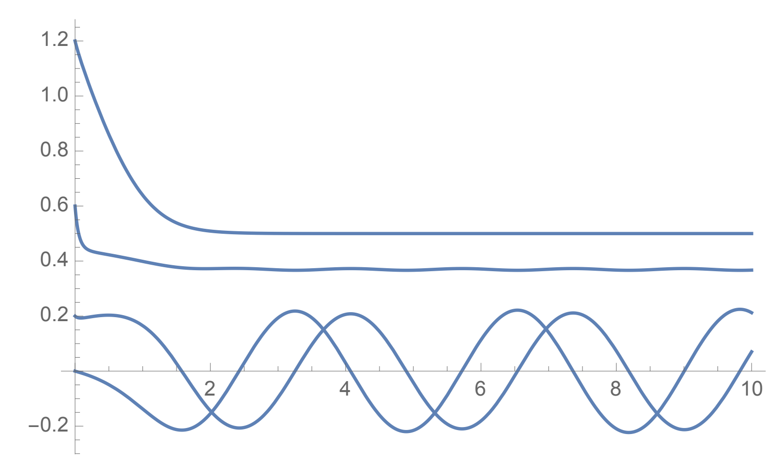
Limit cycles via Hopf analysis

- First route to limit cycles: find a Hopf bifurcation as gain $K = 1/\epsilon_v$ is increased



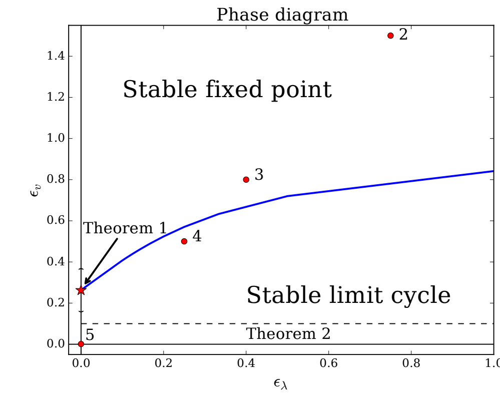
THEOREM 1. *Set $\sigma = 4$. The system $\dot{z}_r = f_r(z_r, \epsilon_v)$ defined by (19) has a deadlock equilibrium z_{rd} given by (22). For sufficiently small $\eta > 0$, the dynamics undergo a Hopf bifurcation resulting in stable periodic solutions at $(z_{rd}, \epsilon_{v,0}(\eta))$, where $\eta \ll 1$ is the saturation constant. In the limit $\eta \to 0$, $\epsilon_{v,0}(0) \approx 0.262$ is the smaller of the two real-valued solutions of $(1 - 4\epsilon_v^2)^2 - 2\epsilon_v = 0$.*

- Sufficiently high gain = limit cycle



PBR, Kod, *SIAM J. Appl. Dyn. Systems*, 2018

Limit cycles via singular perturbation

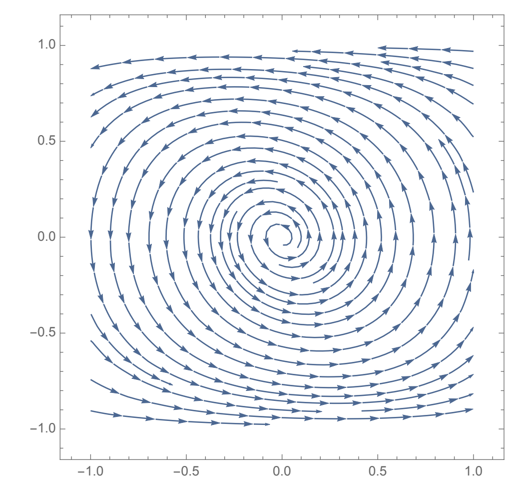


- The limit cycle is in fact structurally stable $\dot{x} = f(x, \mu = \epsilon_v)$

THEOREM 2. *Accepting Conjecture 21, below, for $\sigma = 4$, there exists a stable limit cycle of (12) for sufficiently small, but finite, values of ϵ_λ and ϵ_v. Equivalently, fixing λ, there exists a stable limit cycle of (12) for sufficiently large, but finite, values of v^*.*

High gain = limit cycle

- Proof sketch:
  - Start with 6-D system in $\Delta m, \bar{m}, \Delta\varphi, \bar{\varphi}, \Delta v, \bar{v}$
  - Eliminate $\bar{\varphi}, \bar{v}$ using asymptotic stability
  - Eliminate $\Delta v, \bar{m}$ by singular perturbation in $\epsilon = 1/v^*$
  - Resulting planar system has a limit cycle (Poincare-Bendixson)
  - Fenichel lets us relax away from the limit $\epsilon \to 0$



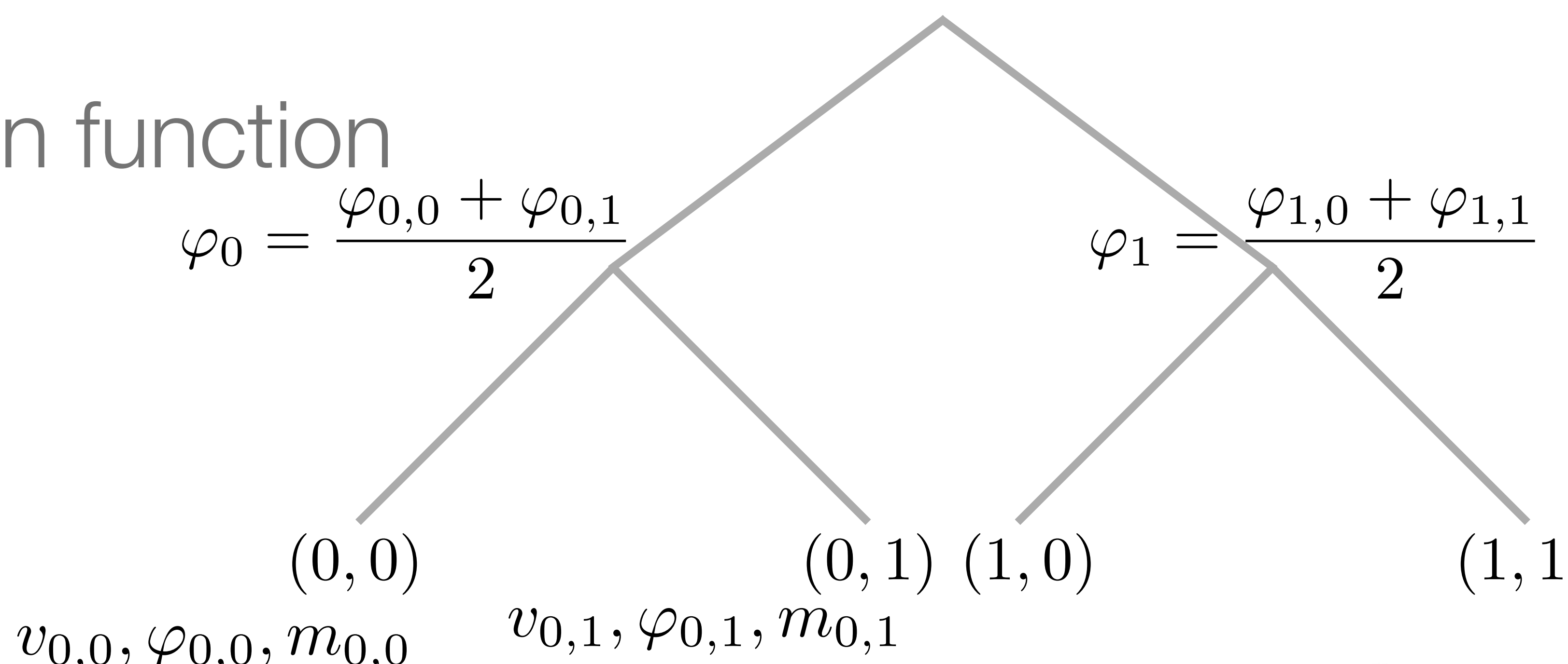
$\Delta m, \Delta\varphi$

Limit cycle = repeatedly visit goals

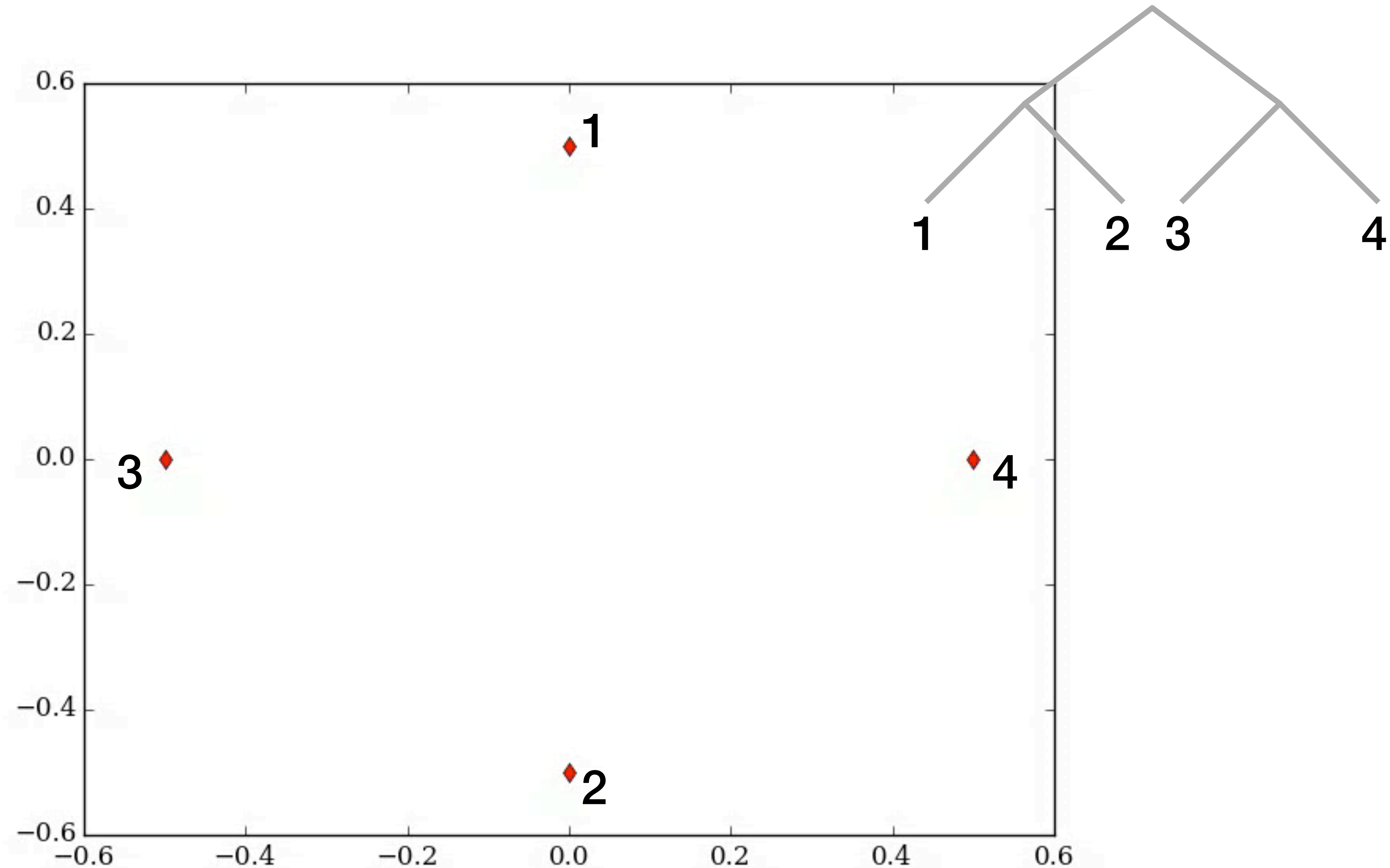
PBR, Kod, *SIAM J. Appl. Dyn. Systems*, 2018

Multiple tasks via trees

- The decision mechanism only accounts for $N = 2$ goals
- The case $N \geq 3$ is significantly harder; need bifurcations on the N-simplex
- One feasible workaround: use binary trees
- Feed mean navigation function of child nodes back to parent

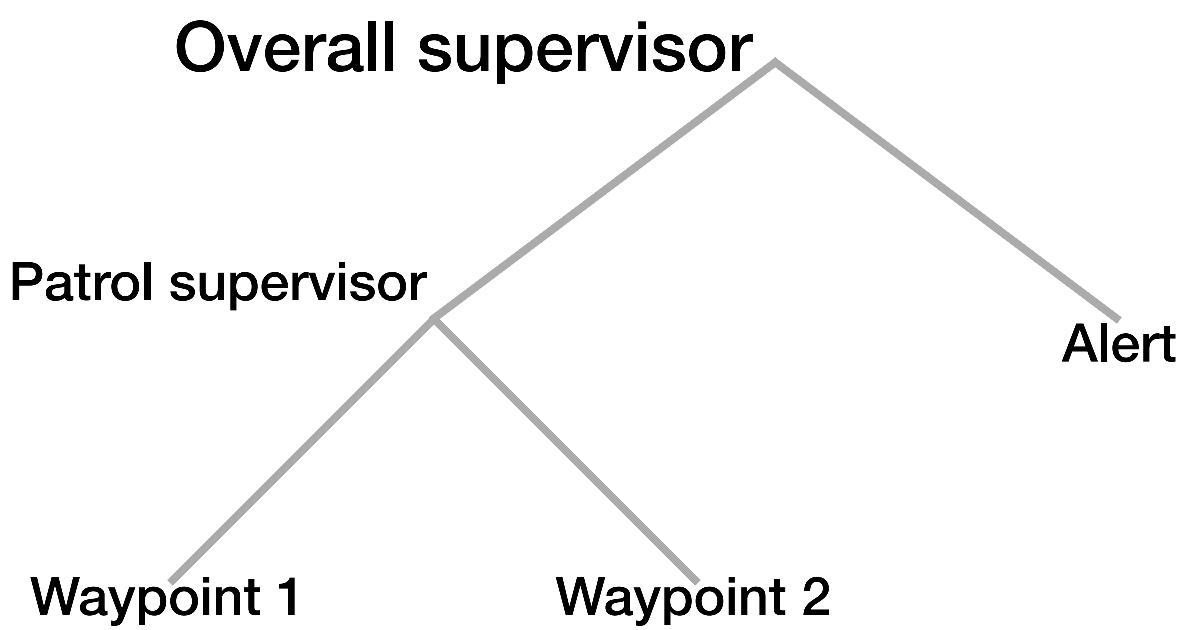


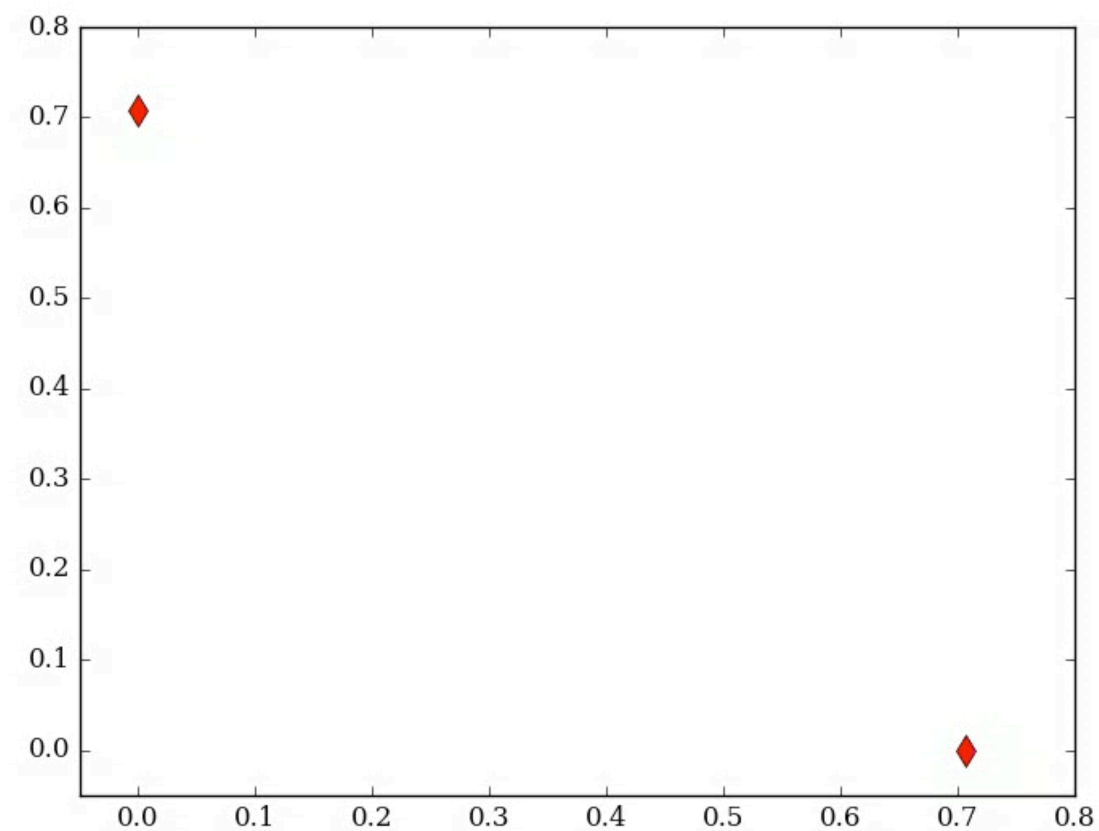
Four tasks



Topology of limit set ~ binary tree topology?

Patrol and inspection (alerts)





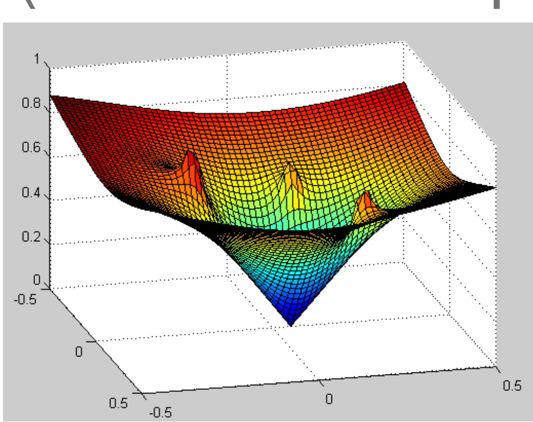
- Use trees again
- When an event occurs, spike alert value, robot visits it
- Once visited, returns to patrol

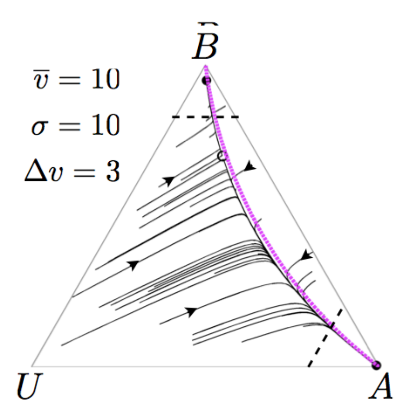
Questions

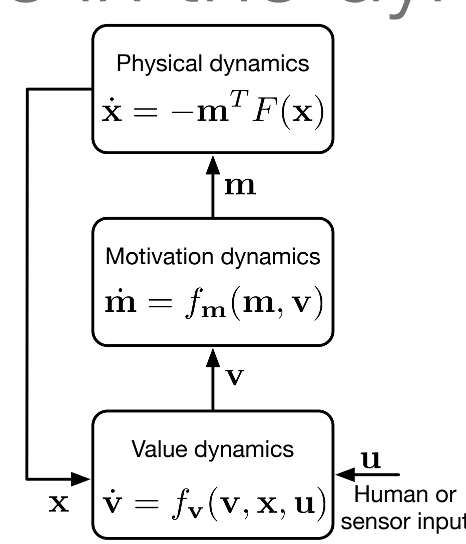
- How can we program this thing?
- In the multi-task case, how are the tree topology and the limit cycle topology related?
- How to connect this with formal synthesis methods?
    - We have a way to express (Eventually)(Always)(Go to location 1 (And) Go to location 2).
- How to incorporate external stimuli? Multiple agents?

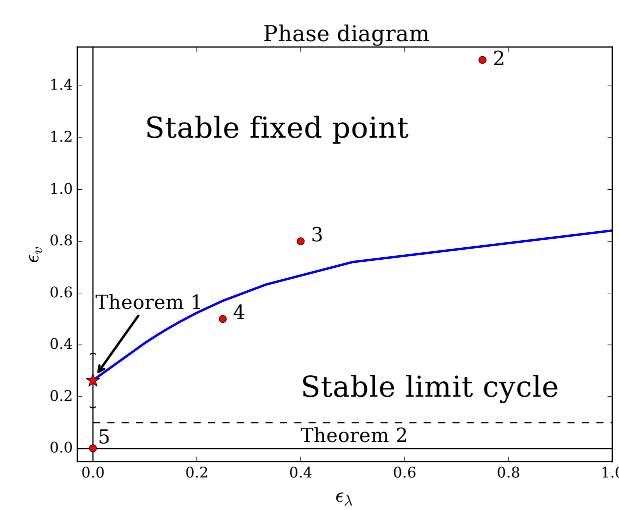
Conclusions

- Defined autonomy as prioritized behaviors
- Adopted navigation as prototypical behavior, encoded in vector fields
- Developed bio-inspired dynamical system to compose multiple vector fields
- Proved existence of limit cycle in the dynamical system (=recurrent patrol)

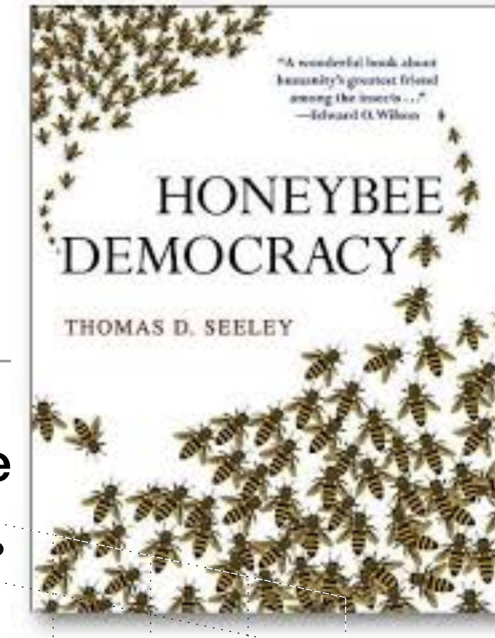






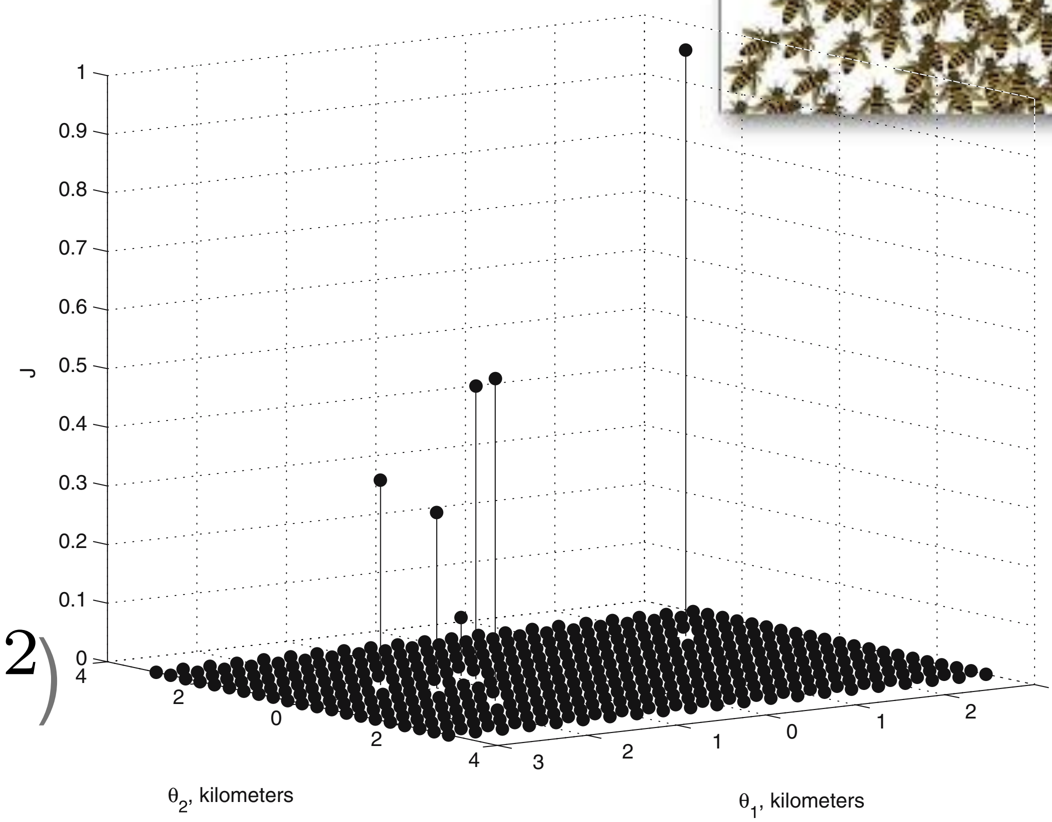


Honeybee Democracy



- Pick nest site
    - With high quality (value, v)
    - Quickly (avoid deadlock)
- Two-site model: (on simplex Δ^2)

Quality landscape

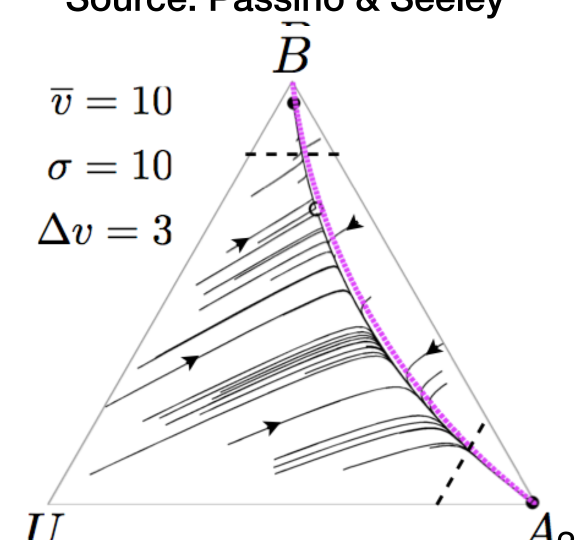


Source: Passino & Seeley

$$\dot{y}_A = \boxed{-\frac{1}{v_A} y_A} + \boxed{v_A y_U (1 + y_A)} - \boxed{\sigma y_A y_B}$$

$$\dot{y}_B = \boxed{-\frac{1}{v_B} y_B} + \boxed{v_B y_U (1 + y_B)} - \boxed{\sigma y_A y_B}$$

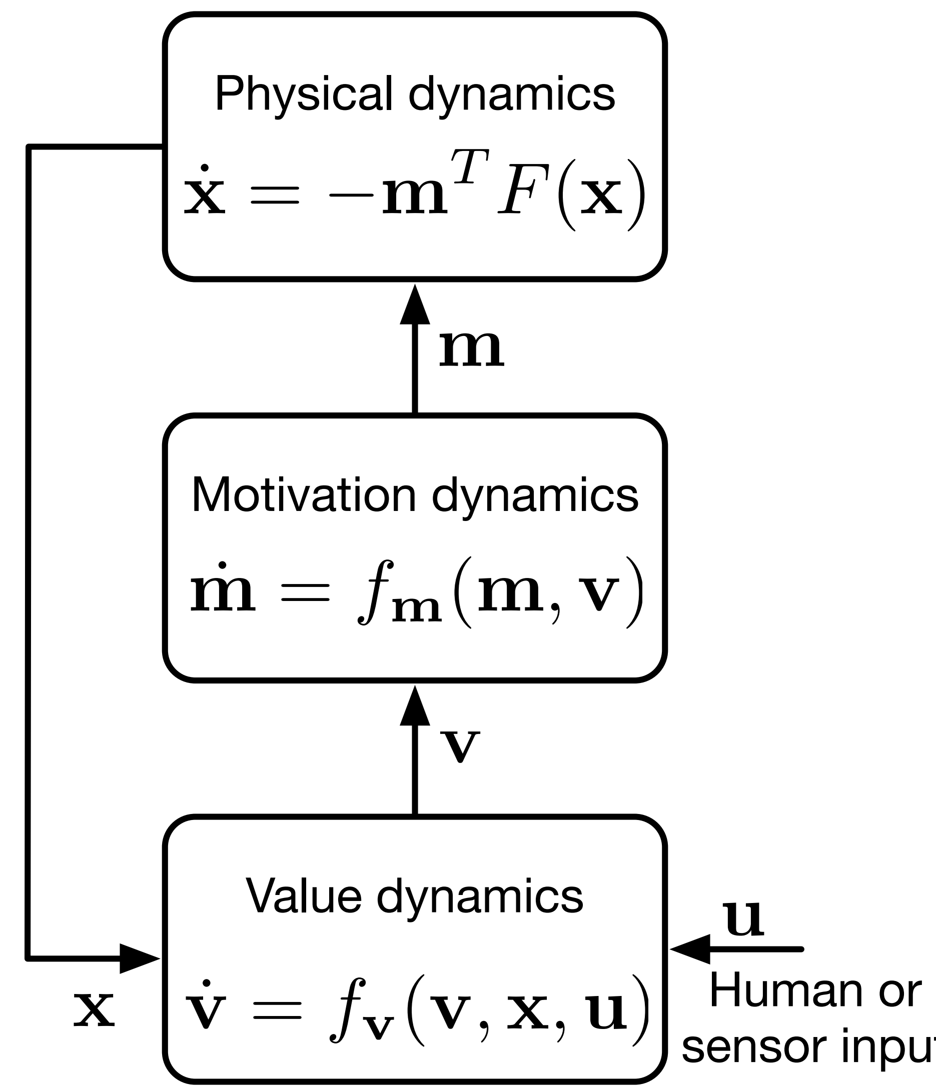
Inhibition Excitation Stop signal



Source: Pais et al.

Motivational *dynamics*

- Pais et al. models one-off decisions: “value” is static
- Value associated with a goal should:
    - Increase when far from goal
    - Decay once reached (satiation)
- **Idea:** nav. function modulates value
- Then use value as input to motivation
- Want to encode recurrent patrol tasks in limit cycles



Value dynamics

- N goals (locations), each with navigation functions

$$\varphi_i : \mathcal{D} \to [0,1] \qquad\qquad \varphi_i, i \in \{1, \ldots, N\}$$

- Value $v_i > 0$ with dynamics

$$\dot{v}_i = \underbrace{\boxed{\lambda_i(v_i^* - v_i)}}_{\text{Stable growth}} - \underbrace{\boxed{\lambda_i v_i^*(1 - \varphi_i(x))}}_{\text{Decay at goal}} \qquad \lambda_i, v_i^* > 0$$

- Motivation state $m = (m_1, \ldots m_N, m_U) \in \Delta^N$

$$\dot{m}_i = v_i m_U - m_i \left(1/v_i - v_i m_U - \sigma(1 - m_i m_U\right))$$

- Physical dynamics

$$\begin{aligned}\dot{\mathbf{x}} &= -m^T D_x \Phi \\ &= -(m_1 \nabla\varphi_1 + \cdots + m_N \nabla\varphi_N)\end{aligned}$$

combination of vector fields