Gaussian Process Temporal Difference Learning - Theory and Practice

Yaakov Engel

Collaborators: Shie Mannor, Ron Meir, Peter Szabo,
Dmitry Volkinshtein, Nadav Aharony, Tzachi Zehavi

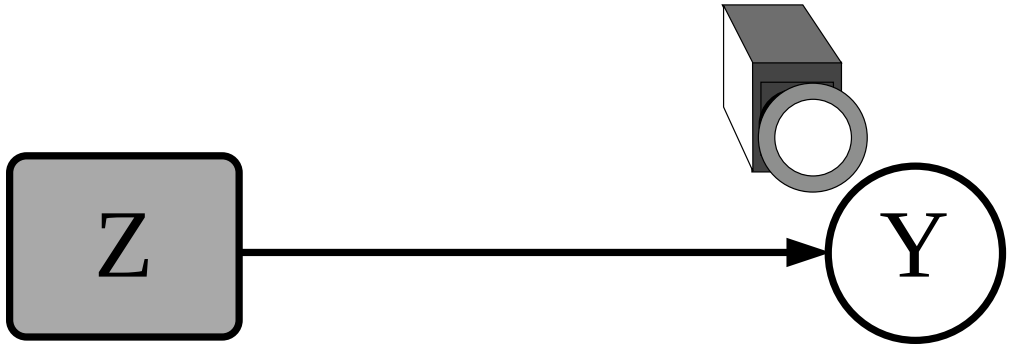
Timeline

- **ICML'03:** Bayes meets Bellman paper – GPTD model for MDPs with deterministic transitions
- **ICML'05:** RL with GPs paper – GPTD model for general MDPs + GPSARSA for learning control
- **NIPS'05:** Learning to control an Octopus Arm – GPTD applied to a high dimensional control problem
- **OPNET'05:** Network association-control with GPSARSA

Why use GPs in RL?

- A Bayesian approach to value estimation
- Forces us to to make our assumptions explicit
- Non-parametric – priors are placed and inference is performed directly in function space (kernels).
- But, can also be defined parametrically
- Domain knowledge intuitively coded in priors
- Provides full posterior, not just point estimates
- Efficient, on-line implementations, suitable for large problems

The Bayesian Approach



- Z – hidden process, Y – observable
- We want to infer Z from measurements of Y
- Statistical dependence between Z and Y known: $P(Y|Z)$
- Place prior over Z, reflecting our uncertainty: $P(Z)$
- Observe $Y = y$
- Compute posterior: $P(Z|Y = y) = \frac{P(y|Z)P(Z)}{\int dZ' P(y|Z')P(Z')}$

Gaussian Processes

Definition: "An **indexed** set of jointly Gaussian random variables"

Note: The index set $\mathcal{X}$ may be just about **any** set.

Example: $F(\mathbf{x})$, index is $\mathbf{x} \in [0,1]^n$

F's distribution is specified by its mean and covariance:

$$\mathbf{E}\big[F(\mathbf{x})\big] = m(\mathbf{x}) \;, \quad \mathbf{Cov}\big[F(\mathbf{x}), F(\mathbf{x}')\big] = k(\mathbf{x}, \mathbf{x}')$$

m is a function $\mathcal{X} \to \mathbb{R}$, k is a function $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

Conditions on k:

Symmetric, positive definite $\Rightarrow$ k is a **Mercer kernel**

GP Regression

Model equation:

$$Y(\mathbf{x}) = F(\mathbf{x}) + N(\mathbf{x})$$

Prior:

$$F \sim \mathcal{N}\{0, k(\cdot,\cdot)\}$$

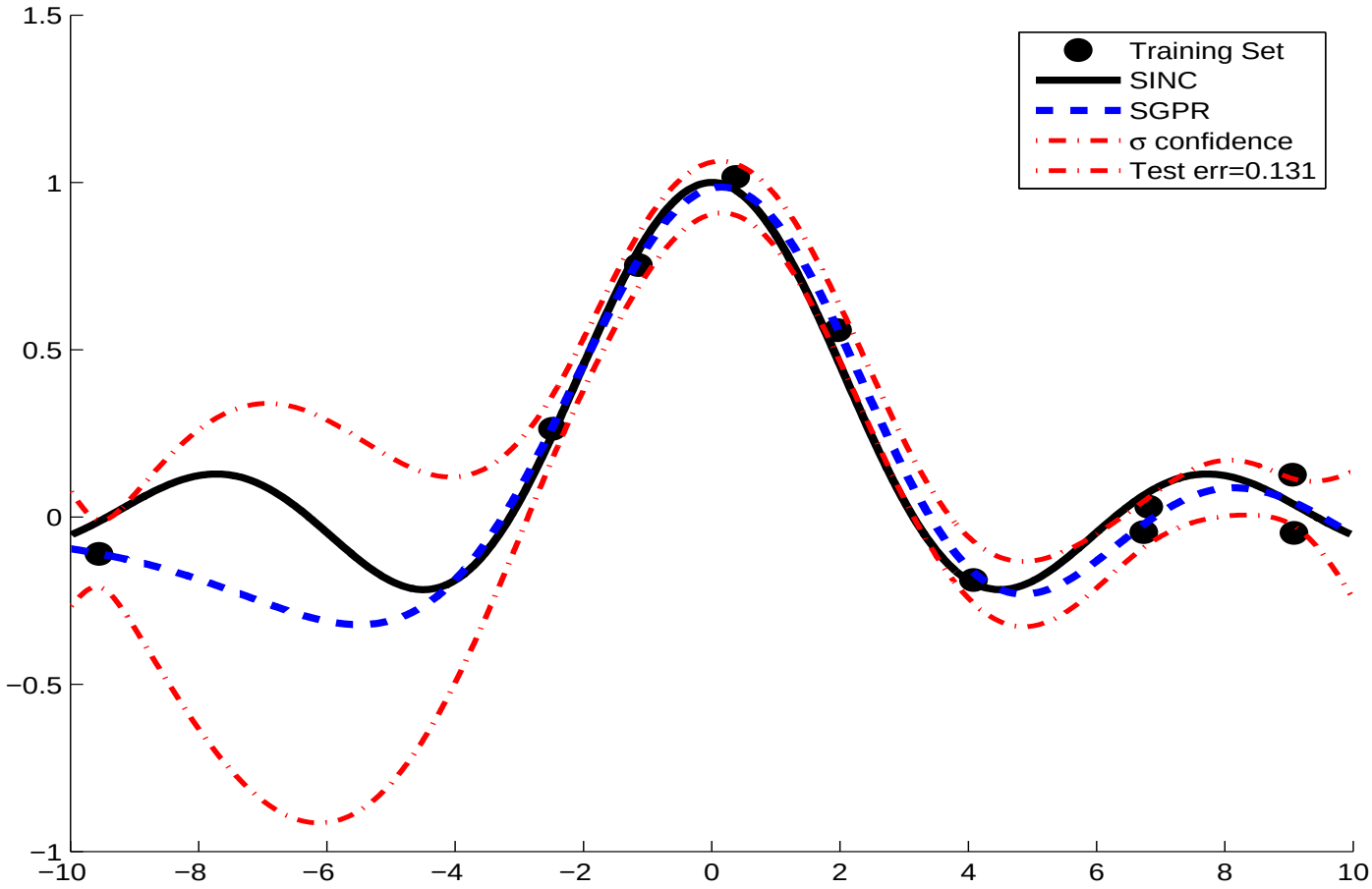
Noise:

$$N \sim \mathcal{N}\left\{0, \sigma^2\delta(\cdot - \cdot)\right\}$$

Goal:

Find the posterior distribution of F,
given a sample for Y (via Bayes' rule)

Example


Training Set
SINC
SGPR
σ confidence
Test err=0.131
1.5
1
0.5
0
−0.5
−1
−10
−8
−6
−4
−2
0
2
4
6
8
10

Markov Decision Processes

$\mathcal{X}$: state space

$\mathcal{U}$: action space

p: $\mathcal{X} \times \mathcal{X} \times \mathcal{U} \to [0,1], \quad \mathbf{x}_{t+1} \sim p(\cdot|\mathbf{x}_t, \mathbf{u}_t)$

q: $\mathbb{R} \times \mathcal{X} \times \mathcal{U} \to [0,1], \quad R(\mathbf{x}_t, \mathbf{u}_t) \sim q(\cdot|\mathbf{x}_t, \mathbf{u}_t)$

A Stationary policy:

μ: $\mathcal{U} \times \mathcal{X} \to [0,1], \quad \mathbf{u}_t \sim \mu(\cdot|\mathbf{x}_t)$

Discounted Return: $D^\mu(\mathbf{x}) = \sum_{i=0}^{\infty} \gamma^i R(\mathbf{x}_i, \mathbf{u}_i) | (\mathbf{x}_0 = \mathbf{x})$

Value function: $V^\mu(\mathbf{x}) = \mathbf{E}_\mu[D^\mu(\mathbf{x})]$

Goal: Find a policy μ^* maximizing $V^\mu(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{X}$

Bellman's Equation

For a fixed policy μ:

$$V^{\mu}(\mathbf{x}) = \mathbf{E}_{\mathbf{x}',\mathbf{u}|\mathbf{x}}\Big[R(\mathbf{x},\mathbf{u}) + \gamma V^{\mu}(\mathbf{x}')\Big]$$

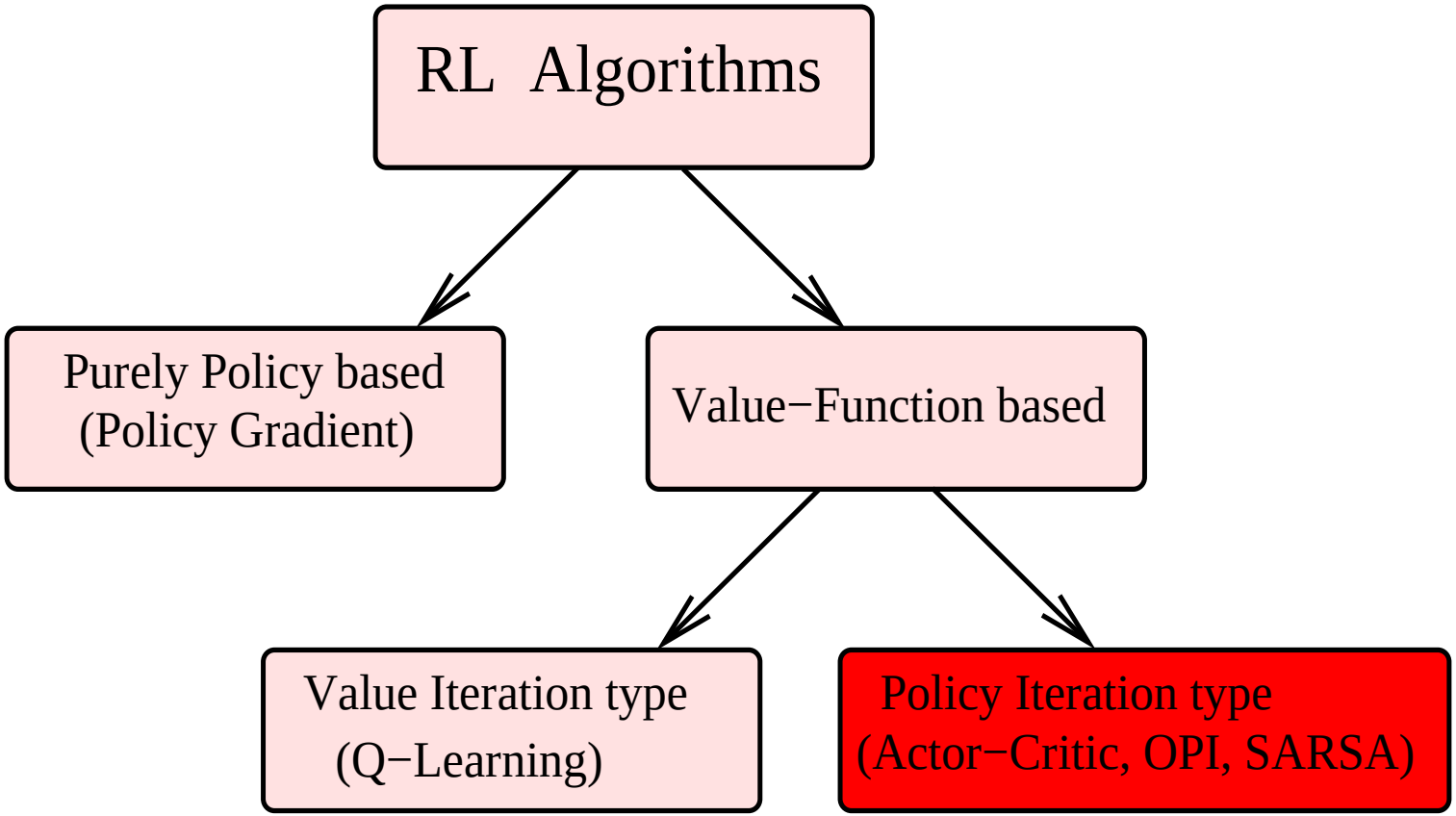
Optimal value and policy:

$$V^{*}(\mathbf{x}) = \max_{\mu} V^{\mu}(\mathbf{x}) \;, \quad \mu^{*} = \operatorname*{argmax}_{\mu} V^{\mu}(\mathbf{x})$$

How to solve it?

- Methods based on Value Iteration (e.g. Q-learning)
- Methods based on Policy Iteration (e.g. SARSA, OPI, Actor-Critic)

Solution Method Taxonomy



PI methods need a "subroutine" for policy evaluation

What's Missing?

Shortcomings of current policy evaluation methods:

- Some methods can only be applied to small problems
- No probabilistic interpretation - how good is the estimate?
- Only parametric methods are capable of operating on-line
- Non-parametric methods are more flexible but only work off-line
- Small-step-size (stoch. approx.) methods use data inefficiently
- Finite-time solutions lack interpretability, all statements are asymptotic
- Convergence issues

Gaussian Process Temporal Difference Learning

Model Equations:

$$R(\mathbf{x}_i) = V(\mathbf{x}_i) - \gamma V(\mathbf{x}_{i+1}) + N(\mathbf{x}_i, \mathbf{x}_{i+1})$$

Or, in compact form:

$$R_t = \mathbf{H}_{t+1} V_{t+1} + N_t$$

$$\mathbf{H}_t = \begin{bmatrix} 1 & -\gamma & 0 & \dots & 0 \\ 0 & 1 & -\gamma & \dots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & \dots & 1 & -\gamma \end{bmatrix}.$$

Our (Bayesian) goal:

Find the posterior distribution of V,
given a sequence of observed states and rewards.

Deterministic Dynamics

Bellman's Equation:

$$V(\mathbf{x}_i) = \bar{R}(\mathbf{x}_i) + \gamma V(\mathbf{x}_{i+1})$$

Define:

$$N(\mathbf{x}) = R(\mathbf{x}) - \bar{R}(\mathbf{x})$$

Assumption: $N(\mathbf{x}_i)$ are Normal, i.i.d., with variance σ^2.

Model Equations:

$$R(\mathbf{x}_i) = V(\mathbf{x}_i) - \gamma V(\mathbf{x}_{i+1}) + N(\mathbf{x}_i)$$

In compact form:

$$R_t = \mathbf{H}_{t+1} V_{t+1} + N_t \text{ , with } N_t \sim \mathcal{N}\left\{0, \sigma^2 \mathbf{I}\right\}$$

Stochastic Dynamics

The discounted return:

$$D(\mathbf{x}_i) = \mathbf{E}_\mu D(\mathbf{x}_i) + (D(\mathbf{x}_i) - \mathbf{E}_\mu D(\mathbf{x}_i)) = V(\mathbf{x}_i) + \Delta V(\mathbf{x}_i)$$

For a stationary MDP:

$$D(\mathbf{x}_i) = R(\mathbf{x}_i) + \gamma D(\mathbf{x}_{i+1}) \text{ (where } \mathbf{x}_{i+1} \sim p(\cdot|\mathbf{x}_i, \mathbf{u}_i),\ \mathbf{u}_i \sim \mu(\cdot|\mathbf{x}_i))$$

Substitute and rearrange:

$$\begin{array}{l} R(\mathbf{x}_i) = V(\mathbf{x}_i) - \gamma V(\mathbf{x}_{i+1}) + N(\mathbf{x}_i, \mathbf{x}_{i+1}) \\ N(\mathbf{x}_i, \mathbf{x}_{i+1}) \stackrel{\text{def}}{=} \Delta V(\mathbf{x}_i) - \gamma \Delta V(\mathbf{x}_{i+1}) \end{array}$$

Assumption: $\Delta V(\mathbf{x}_i)$ are Normal, i.i.d., with variance σ^2.

In compact form:

$$R_t = \mathbf{H}_{t+1} V_{t+1} + N_t \text{ , with } N_t \sim \mathcal{N}\left\{0, \sigma^2 \mathbf{H}_{t+1} \mathbf{H}_{t+1}^\top\right\}$$

The Posterior

General noise covariance: $\mathbf{Cov}[N_t] = \mathbf{\Sigma}_t$

Joint distribution:

$$\begin{bmatrix} R_{t-1} \\ V(\mathbf{x}) \end{bmatrix} \sim \mathcal{N}\left\{ \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{H}_t\mathbf{K}_t\mathbf{H}_t^\top + \mathbf{\Sigma}_t & \mathbf{H}_t\mathbf{k}_t(\mathbf{x}) \\ \mathbf{k}_t(\mathbf{x})^\top\mathbf{H}_t^\top & k(\mathbf{x},\mathbf{x}) \end{bmatrix} \right\}$$

Invoke Bayes' Rule:

$$\mathbf{E}[V(\mathbf{x})|R_{t-1} = \mathbf{r}_{t-1}] = \mathbf{k}_t(\mathbf{x})^\top \boldsymbol{\alpha}_t$$

$$\mathbf{Cov}[V(\mathbf{x}), V(\mathbf{x}')|R_{t-1} = \mathbf{r}_{t-1}] = k(\mathbf{x},\mathbf{x}') - \mathbf{k}_t(\mathbf{x})^\top \mathbf{C}_t \mathbf{k}_t(\mathbf{x}')$$

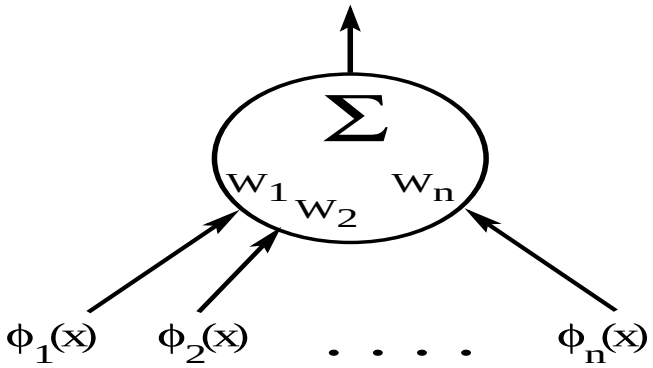
$$\mathbf{k}_t(\mathbf{x}) = (k(\mathbf{x}_0,\mathbf{x}),\ldots,k(\mathbf{x}_t,\mathbf{x}))^\top, \qquad \mathbf{K}_t = [\mathbf{k}_t(\mathbf{x}_0),\ldots,\mathbf{k}_t(\mathbf{x}_t)]$$

$$\boldsymbol{\alpha}_t = \mathbf{H}_t^\top\left(\mathbf{H}_t\mathbf{K}_t\mathbf{H}_t^\top + \mathbf{\Sigma}_t\right)^{-1}\mathbf{r}_{t-1}, \quad \mathbf{C}_t = \mathbf{H}_t^\top\left(\mathbf{H}_t\mathbf{K}_t\mathbf{H}_t^\top + \mathbf{\Sigma}_t\right)^{-1}\mathbf{H}_t.$$

A Parametric Gaussian Process Model

A linear combination of features:

$$V(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top W$$



Prior on W: Gaussian, with $\mathbf{E}[W] = \mathbf{0}$, $\mathbf{Cov}[W, W] = \mathbf{I}$

Prior on V: Gaussian, with

$$\mathbf{E}[V(\mathbf{x})] = \mathbf{0}, \quad \mathbf{Cov}[V(\mathbf{x}), V(\mathbf{x}')] = \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$$

Comparison of Models

	Parametric	**Nonparametric**
Parametrization	$V(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top W$	None, V is V
Prior	$W \sim \mathcal{N}\{\mathbf{0}, \mathbf{I}\}$	$V \sim \mathcal{N}\{0, k(\cdot,\cdot)\}$
$\mathbf{E}[V(\mathbf{x})]$	0	0
$\mathbf{Cov}[V(\mathbf{x}), V(\mathbf{x}')]$	$\boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$	$k(\mathbf{x}, \mathbf{x}')$
We seek	$W \vert R_{t-1}$	$V(\mathbf{x}) \vert R_{t-1}$

If we can find a set of basis functions satisfying $\boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}') = k(\mathbf{x}, \mathbf{x}')$, the two models become equivalent.

In fact, such a set **always** exists [Mercer].

However, it may be infinite

Relation to Monte-Carlo Estimation

In the stochastic model: $\boldsymbol{\Sigma}_t = \sigma^2 \mathbf{H}_{t+1} \mathbf{H}_{t+1}^\top$

Also, let: $(Y_t)_i = \sum_{j=i}^{t} \gamma^{j-i} R(\mathbf{x}_i, \mathbf{u}_i)$

Then:

$$\mathbf{E}[W|R_t] = \left(\boldsymbol{\Phi}_t \boldsymbol{\Phi}_t^\top + \sigma^2 \mathbf{I}\right)^{-1} \boldsymbol{\Phi}_t Y_t$$

$$\mathbf{Cov}[W|R_t] = \sigma^2 \left(\boldsymbol{\Phi}_t \boldsymbol{\Phi}_t^\top + \sigma^2 \mathbf{I}\right)^{-1}$$

That's the solution to GP regression on Monte-Carlo samples of the discounted return.

MAP / ML Solutions

Since the posterior is Gaussian:

$$\hat{\mathbf{w}}_{t+1}^{MAP} = \mathbf{E}[W|R_t] = \left(\mathbf{\Phi}_t \mathbf{\Phi}_t^\top + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{\Phi}_t Y_t$$

Performing ML inference using the same model we get:

$$\hat{\mathbf{w}}_{t+1}^{ML} = \left(\mathbf{\Phi}_t \mathbf{\Phi}_t^\top\right)^{-1} \mathbf{\Phi}_t Y_t$$

That's the LSTD(1) (Least-Squares Monte-Carlo) solution.

Policy Improvement

How can we perform policy improvement?

State values? Not without a transition model (even then tricky).

State-action (Q-) values? Yes!

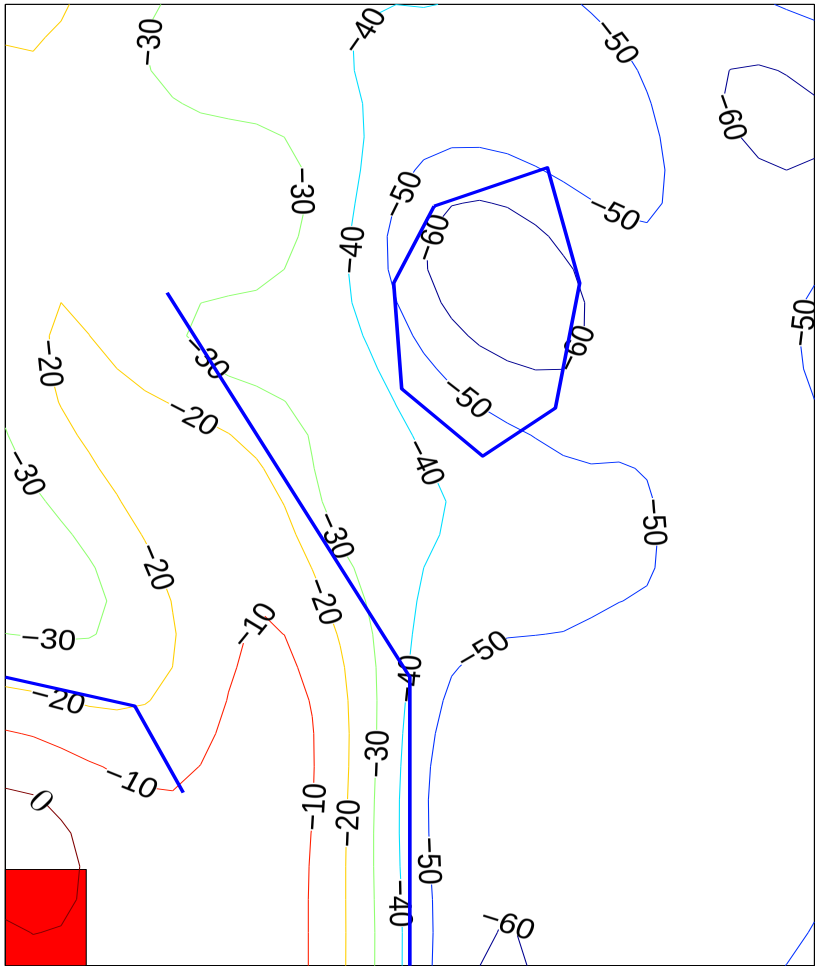
Idea: Use a state-action value GP

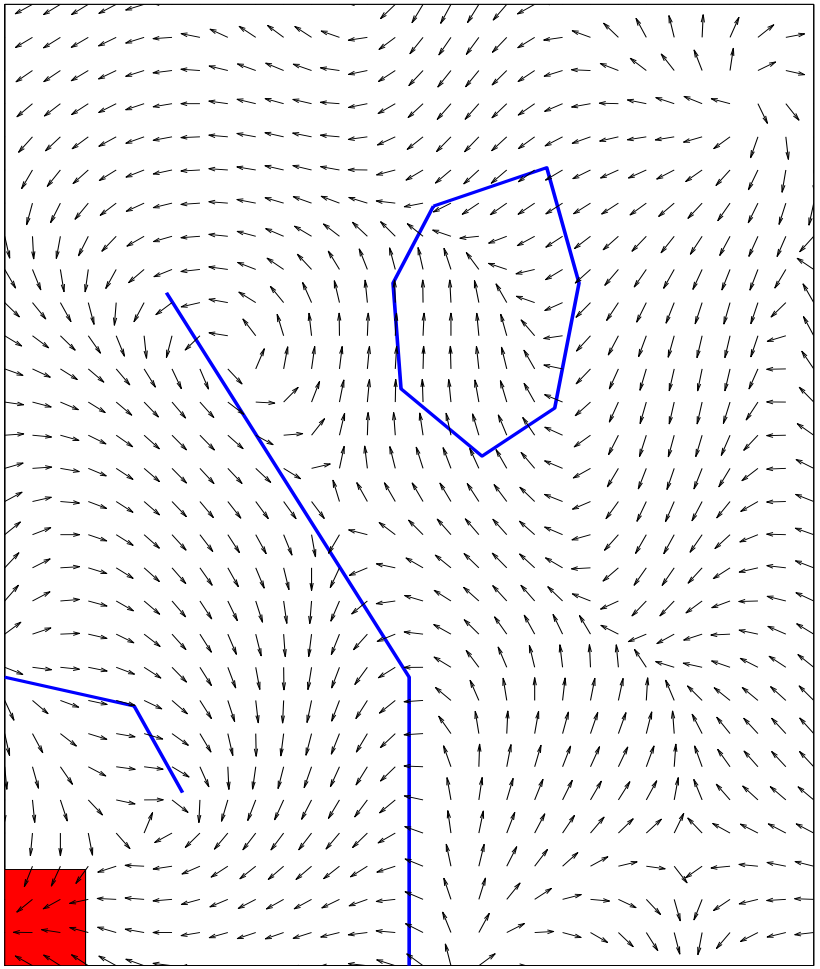
How?

- Define a state-action kernel: $k\big((\mathbf{x}, \mathbf{u}), (\mathbf{x}', \mathbf{u}')\big)$
- Run GPTD on state-action pairs
- Use some semi-greedy action selection rule

We call this GPSARSA.

A Simple Experiment





The Octopus Arm

Can bend and twist at any point

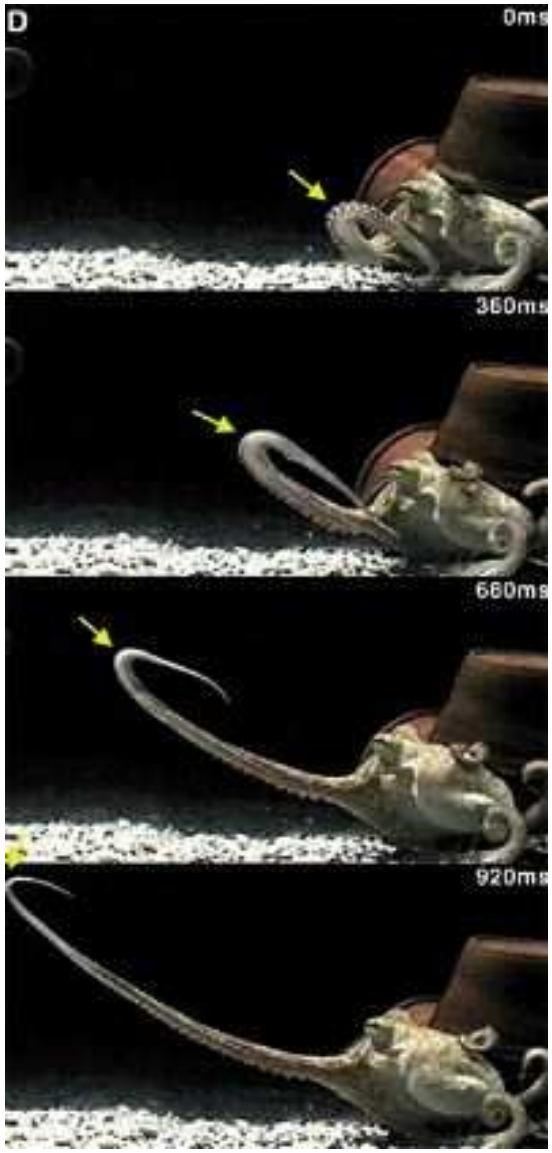
Can do this in any direction

Can be elongated and shortened

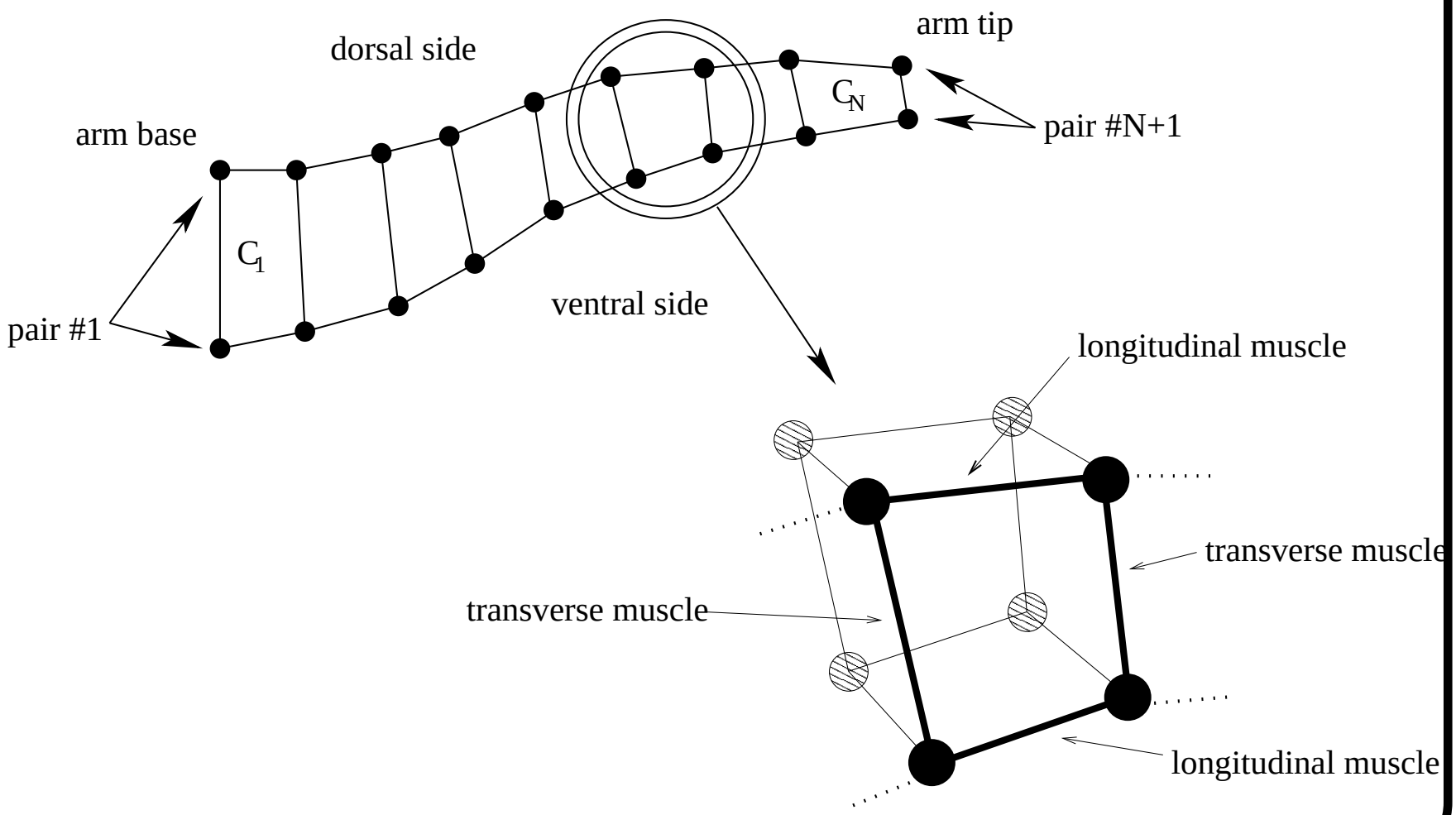
Can change cross section

Can grab using any part of the arm

Virtually infinitely many DOF

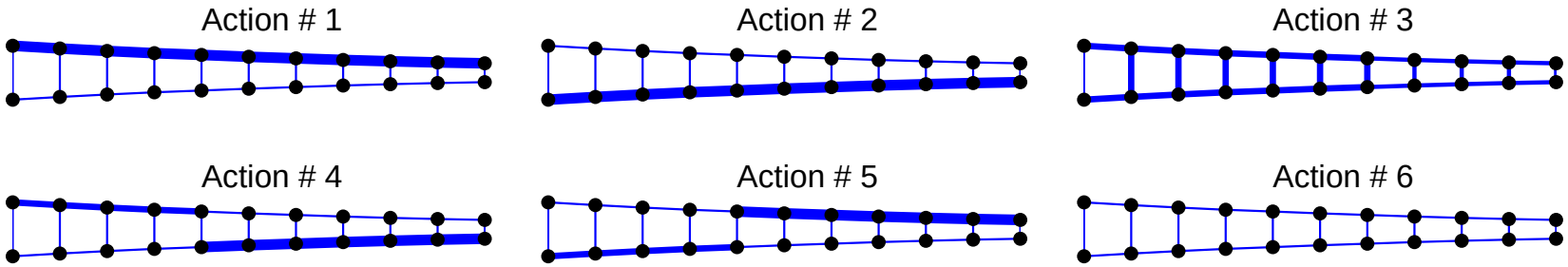


Our Arm Model



Actions

Each action specifies a set of fixed activations – one for each muscle in the arm.



Base rotation adds duplicates of actions 1,2,4 and 5 with positive and negative torques applied to the base.

The Control Problem

Starting from a random position, bring {any part, tip} of arm into contact with a goal region, **optimally**.

Optimality criteria:

Time, energy, obstacle avoidance

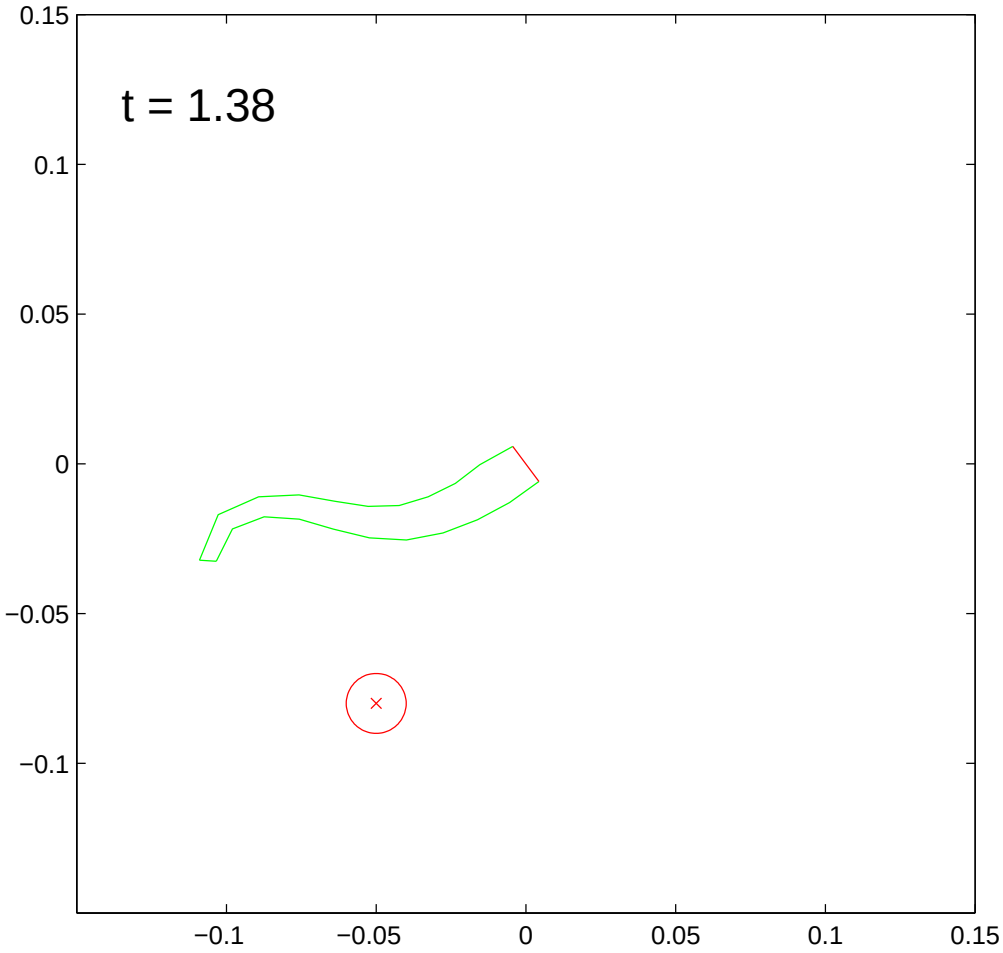
Constraint:

We only have access to sampled trajectories

Our approach:

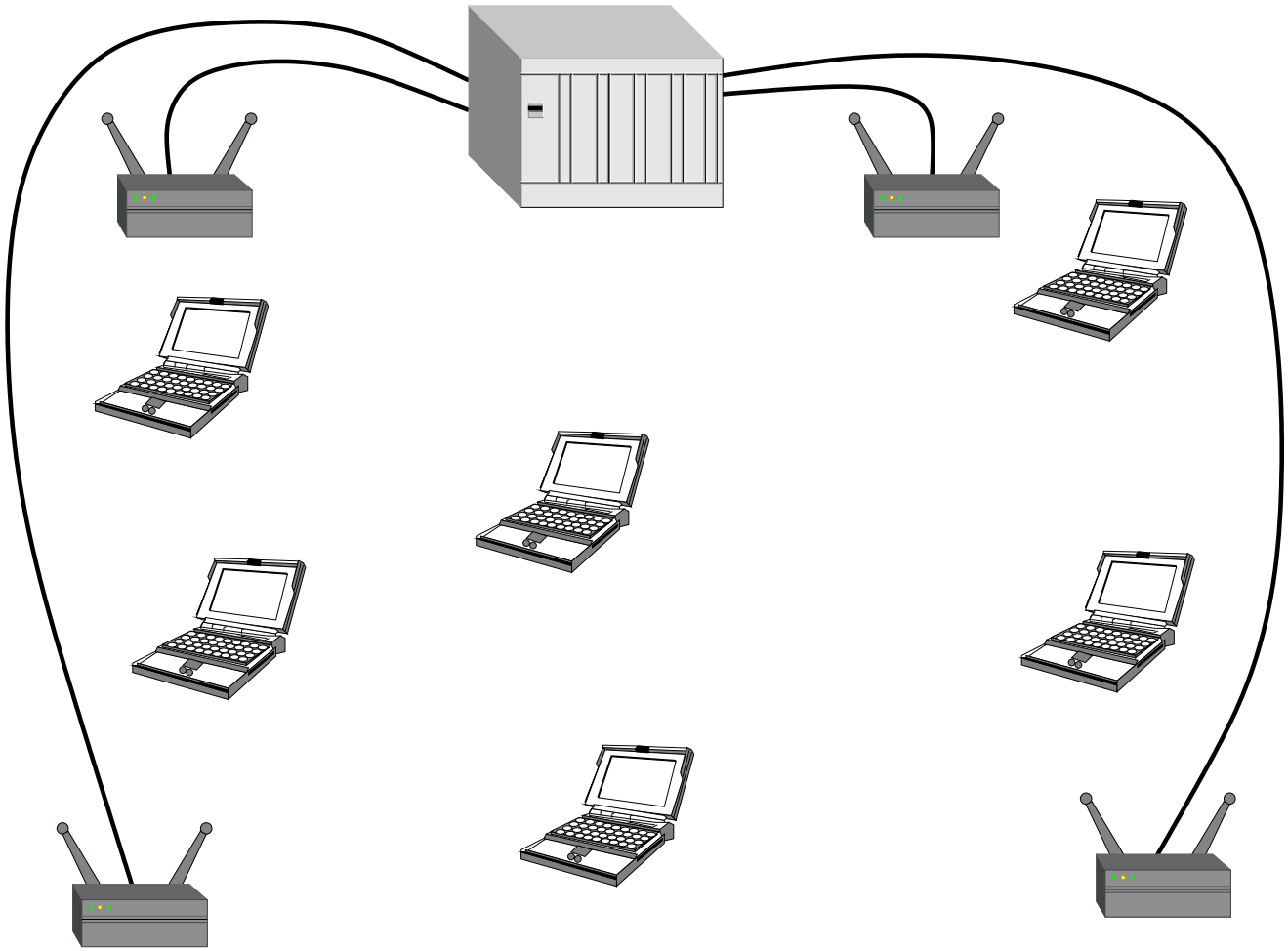
Define problem as a MDP
Solve using a GPTD algorithm

The Task



Movies

Association Control in WLANs



Association Control in WLANs

Setting: n users, $m \ll n$ access points (APs),

The problem: Associate users with APs, optimally.

Complications: Users are not the same, they move around, change their behavior over time, what is meant by "optimally"? etc.

Idea: Model the system as a MDP, solve using GPSARSA

Results:

- Tested on simple networks using the OPNET simulator
- Preliminary results look promising
- More work is needed

Challenges

- How to use value uncertainty?
- What's a disciplined way to select actions?
- What's the best noise covariance?
- Bias, variance, learning curves
- POMDPs
- More complicated tasks

Questions?