Learning control for a communicating mobile robot

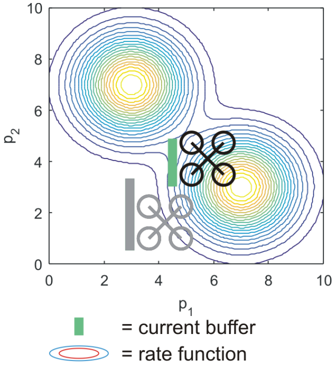
L. Buşoniu, V.S. Varma, I.-C. Morărescu, S. Lasaulce

LB – Technical University of Cluj-Napoca, Romania
VSV, I-CM – CNRS & Université de Lorraine - Nancy, France
SL – CNRS & Université Paris Sud, France

ACC, 10 July 2019

Presented by Aris Kanellopoulos, GATech

Problem statement



- Robot with position p, moves with known dynamics: $p_{k+1} = g(p_k, u_k)$
- Buffer of size b transmitted with: $b_{k+1} = \max\{0, b_k - R(p_k)\}$
- Position-dependent rates $R(p)$ **unknown**

Objective: Transmit buffer in minimum time

Requires learning about the rate function!

Problem statement (continued)

Motivation:

Robot must quickly upload (e.g. remote survey) data over an ad-hoc network with unknown rates

Optimal control formalization:

$$\min_h \quad V^h(x_0) := \sum_{k=0}^{\infty} \rho(b_k)$$

with stage cost: $\rho(b) = \begin{cases} 1 & \text{if } b > 0 \\ 0 & \text{if } b = 0 \end{cases}$ and control law $h(x)$

Note overall state $x = [p^\top, b]^\top$ with dynamics $f(x, u)$

Dynamic programming

To reduce clutter, define DP backup operator:

$$T(x, V) := \min_u [\rho(b) + V(f(x, u))]$$

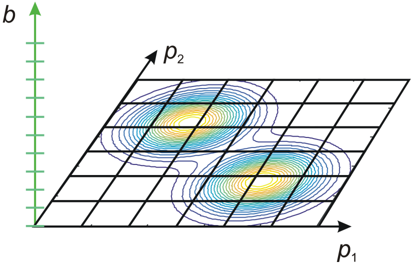
From $V_0 = 0$, iterate until convergence to V^*:

$$V_{\ell+1}(x) = T(x, V_\ell) \text{ for all } x$$

then use control: $h(x) \in \arg T(x, V^*)$

with "arg T..." meaning "$\arg\min_u$ ..."

Interpolated dynamic programming



Interpolate on a grid over p and b

$\Rightarrow$ approximate value function: $\widehat{V}_\theta(x) = \varphi^\top(x)\theta$

Discretize actions $u \Rightarrow$ approximate DP update:

$$\theta_{\ell+1,i} = T(x_i, \widehat{V}_{\theta_\ell}) \text{ for all grid points } x_i$$

High-level algorithm

Learning for the communicating robot

repeat at each time step k

(i) sample $R(p_k)$, update rate approximator $\widehat{R}$

(ii) using $\widehat{R}$, run local DP around current state, starting from current parameters θ

(iii) choose and apply action u_k

(iv) perform Q-learning-like update

until $b_{k+1} = 0$

Key features:

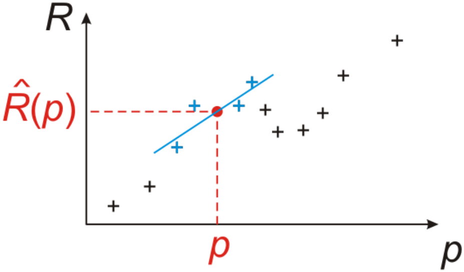
- Single-trajectory learning
- Mixes model-based with model-free
- Learning focuses on unknown part, R

(i) Rate function learning

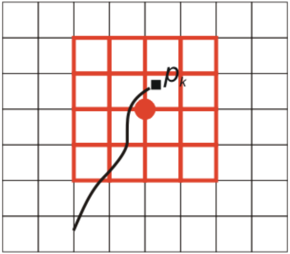
Local linear regression, LLR:

- Given database of points $(p_k, R(p_k))$
- For given p, finds the K nearest neighbors
- $\widehat{R}(p)$ found with linear regression on neighbors

Illustration in 1D:



(ii) Local DP sweeps



Run ℓ_{DP} approximate DP updates on a **subgrid** centered on point closest to x_k and extending r_{DP} in each direction (also on b, not pictured)

Avoids extrapolating too far ahead, since usually samples of R are behind on the trajectory

(iii) Action selection

Easiest option – take the greedy action (as if $\widehat{V}$ were optimal):

$$\arg T(x_k, \widehat{V}_\theta)$$

Optimistic initialization of $\widehat{V}$ to a lower bound
– forces algorithm to explore

(iv) Q-learning-like update

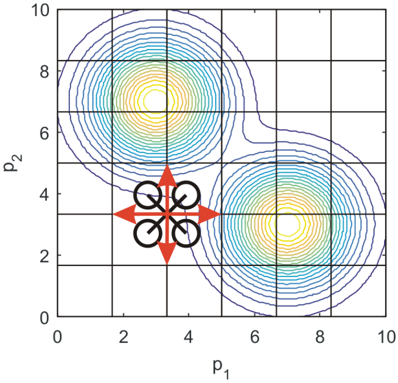
Adapted to V-function and approximator used:

$$\theta \leftarrow \theta + \alpha \varphi(x_k) \left[T(x_k, \widehat{V}_\theta) - \widehat{V}(x_k; \theta) \right]$$

with learning rate α

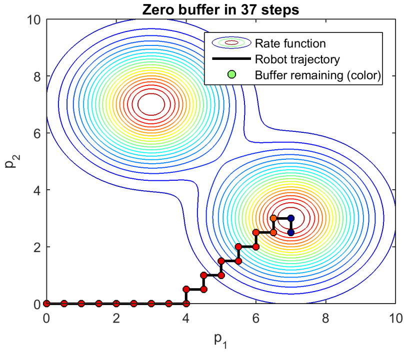
Extracts a bit of extra information from each transition

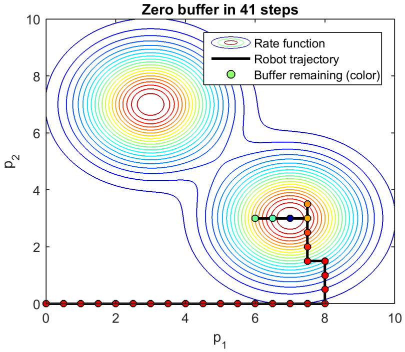
Simple example



- Integrator dynamics; actions chosen to move on a grid of 21×21 points (same as position interpolation grid)
- $b \in [0, 2]$, 21 grid points
- Rate $R(p)$ sum of two Gaussians with amplitude 0.1

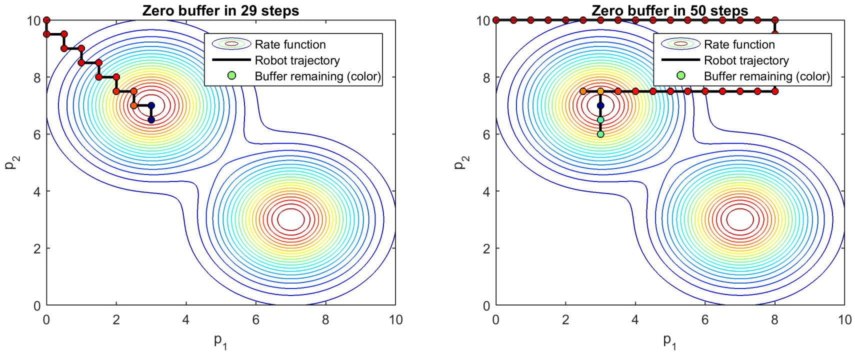
Typical good trajectory





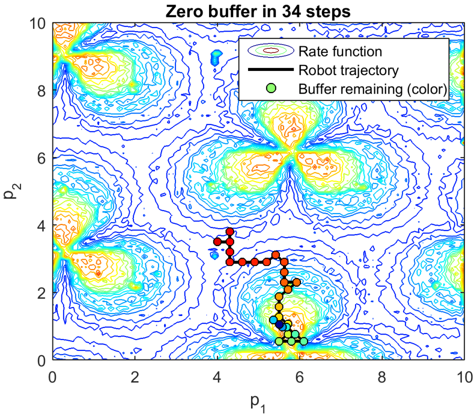
- Left model-based (R known), right model-free (R unknown)
- Tuned params: $K = 4$ in LLR, local DP range $r_{DP} = 4$ with $\ell_{DP} = 10$ iterations, learning rate $\alpha = 1$
- Learning only requires 10% more steps to transmit buffer
- Any DP range works well; no DP (only Q-learning) does not

Typical bad trajectory



- Learning under twice the number of model-based steps

More realistic example



- Realistic radio rates
- Unicycle-like dynamics that include heading as an action
- Same parameters as above, algorithm works

Conclusions

Learning approach for mobile robot to communicate under unknown rates

Next steps:

- Stochastic rates
- Experiments
- Analysis

Thank you! (and thanks Aris!)

Contact: lucian@busoniu.net