

Heterarchical Control in Sensorimotor Processing

Spencer R. Wilson

February 22, 2021

0.1 Internal Model Adaptation for Linear Quadratic Control

Here I investigate the effects of approximating internal dynamical models for movement and using the resulting endpoint error to update this approximation over trials.

Our state space is denoted x and our control space u where $\dim(x) < \dim(u)$. Each trial, we move from state $x(0)$ to $x(N)$ in N timesteps. Each trial, we have a goal state x^* and a resulting endpoint error $e(N) = |x(N) - x^*|^2$.

We use a deterministic linear dynamical system to model our within-trial state dynamics:

$$x(t) = Ax(t-1) + Bu(t-1).$$

For this system, we assume there exists a linear feedback control law optimal under a given quadratic state and control cost:

$$u(t) = Kx(t).$$

We can write the controlled, closed-loop system dynamics for the final time step N :

$$\begin{aligned} x(N) &= (A - BK)x(N-1) = Cx(N-1) \\ x(N) &= Cx(N-1) = C(Cx(N-2)) \\ x(N) &= C^N x(0). \end{aligned}$$

where C^N might be called the trajectory dynamic. If the trajectory dynamic C^N is an approximation to the true trajectory dynamic C^{N*} , we can use the error of a given trajectory to find an incremental update. The error at the final time step N for trial r is

$$e(r) = |C^N(r)x(0) - x^*|^2.$$

This error may be due to several sources. Our internal dynamics model A might have error relative to the true dynamic A^* . Our control gain K may be optimal relative to our internal model A but not with respect to the true dynamic A^* . Finally, we might have an approximate model A and a suboptimal control gain K . Note that since this is still deterministic system, we have yet to include any source of variability in state or control.

If we assume that our computation of the control gain K is optimal for our approximate internal model A (we can compute a controller given only our internal representation of the system dynamic being controlled), we can use our endpoint error to derive a gradient descent update for A on trial r :

$$A(r+1) = A(r) - \eta \frac{\partial e(r)}{\partial A}$$

We might think about this as an internal simulation of trial r 's trajectory, and a subsequent post hoc evaluation of the movement. To compute δ , we must take the gradient with respect to A of the error:

$$\frac{\partial e(r)}{\partial A} = \frac{\partial}{\partial A} |C^N(r)x(0) - x^*|^2$$

Since the gradient with respect to A is the same as the gradient with respect to C , we can compute the gradient with respect to C to find:

$$\frac{\partial e}{\partial A_{ij}} = 2 \sum_{k=1}^N [(M^N x(0) - x^*)^T M^{k-1}]_i [M^{N-k} x(0)]_j$$

Below is a figure showing LQR simulations across gradient descent updates to the A matrix after it is corrupted by Gaussian noise. Each trajectory is a single run of the LQR controlled for 200 time steps. The star shows the target state, the colored circles show the endpoints of the trajectories. The red circle is the initial state.

The descent is converging in endpoint error in position, velocity, and force space. Unfortunately, this optimization is causing the dynamics to change. The routine is very fragile to parameter changes. Next steps:

- Corrupt the A matrix in a more principled way, working to alter the passive dynamics in a physically realistic manner.
- Explore the action of the resulting gradient through it's eigenvalues and vectors. This can be done in two dimensions as a starting point.
- Compute second-order derivatives and work towards a Newton's method.
- Compute derivatives with respect to the control law K as a comparison.
- Analyze results of the routine in comparison with the reaching adaptation literature.

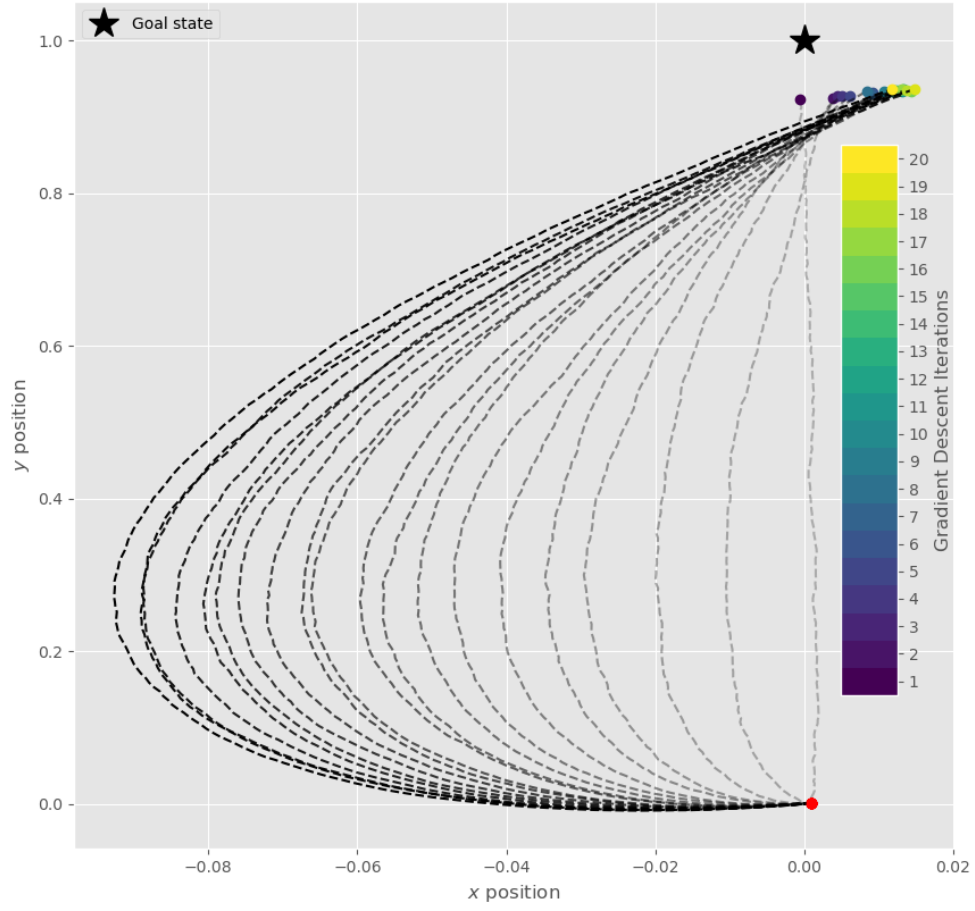


Figure 1: Iterations of gradient descent on the A matrix of an infinite-horizon LQR. Each dotted line in a trajectory with a different A matrix. The red circle denotes the initial state, the star denotes the goal state, and the colored circles denote the endpoints of each iteration.

0.2 Bibliography