Visuomotor Tracking

Ray Luo
rluo@cory.eecs.berkeley.edu
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720

Abstract

Motor planning is modeled as a tracking problem in the case of rapid motor executions as in swatting a fly. Visual feedback and commanded motor movement are coupled in a linear dynamical systems graphical model of the task. Parameters learned by the EM Algorithm on an artificial data set reflect underlying system dynamics.

1 Introduction

How does motor planning and execution incorporate visual information when tracking a fast-moving object in space? One hypothesis involves the activity of an internal model consisting of a forward predicting and an inverse kinematic module [10]. Reviews by Jordan ([4] and [3]) give more details. A recent attempt to model selection of motor programs learns an HMM-based switch model for different types of objects [1]. A similar approach is advocated by Shademehr and Thoroughman [9].

Our approach here is simpler. We treat the motor planning problem as a tracking problem in which the nervous system estimates the positions of both the target and the hand position. The resulting graphical model is a coupled linear dynamical system that can be learned using the EM Algorithm. The approach is similar to what Wu and Huang calls co-inference tracking [11]. In their paper, they gave a sequential Monte Carlo algorithm that utilizes a

bottom up EM step to track both color and shape distributions in a video sequence. We will begin wih a similar model adapted, in our case, to track a target in 2D with a 3D representation of hand position and velocity. We hope to extend the model by incorporating a switching motor control module within the arm location estimator.

2 Graphical model

Every node in the model is continuous, with equations that look like

$$\begin{aligned} H_s(t+1) &= BV_s(t+1) - H_s(t) + W_H(t), \\ V_s(t+1) &= AV_s(t) + W_V(t), \\ H_o(t) &= C_H H_s(t) + U_H(t), \\ V_o(t) &= C_V V_s(t) + U_V(t), \end{aligned}$$

where H_s is the control module associated with the hand, H_o is the estimated hand parameters (in particular, hand position), V_s and V_o are the estimated and observed target motion characteristics, A, B, and C are corresponding output matrices, and the Ws and Us are noise. H_o and V_o are observed. Note that the first equation accounts for sensory correction due to visual feedback.

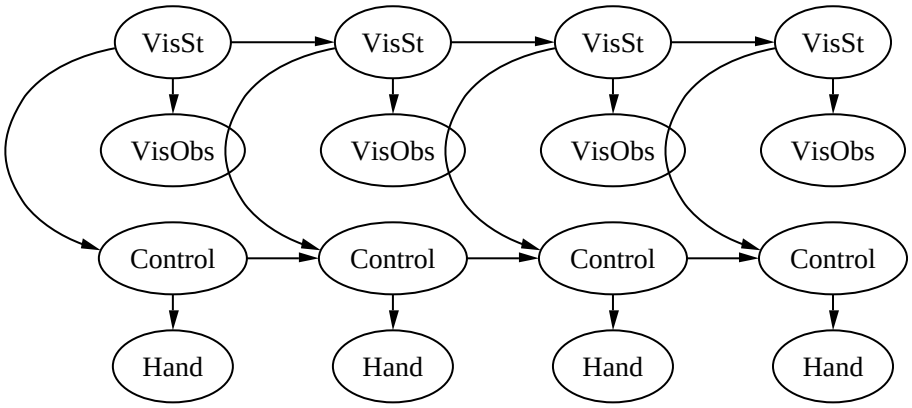


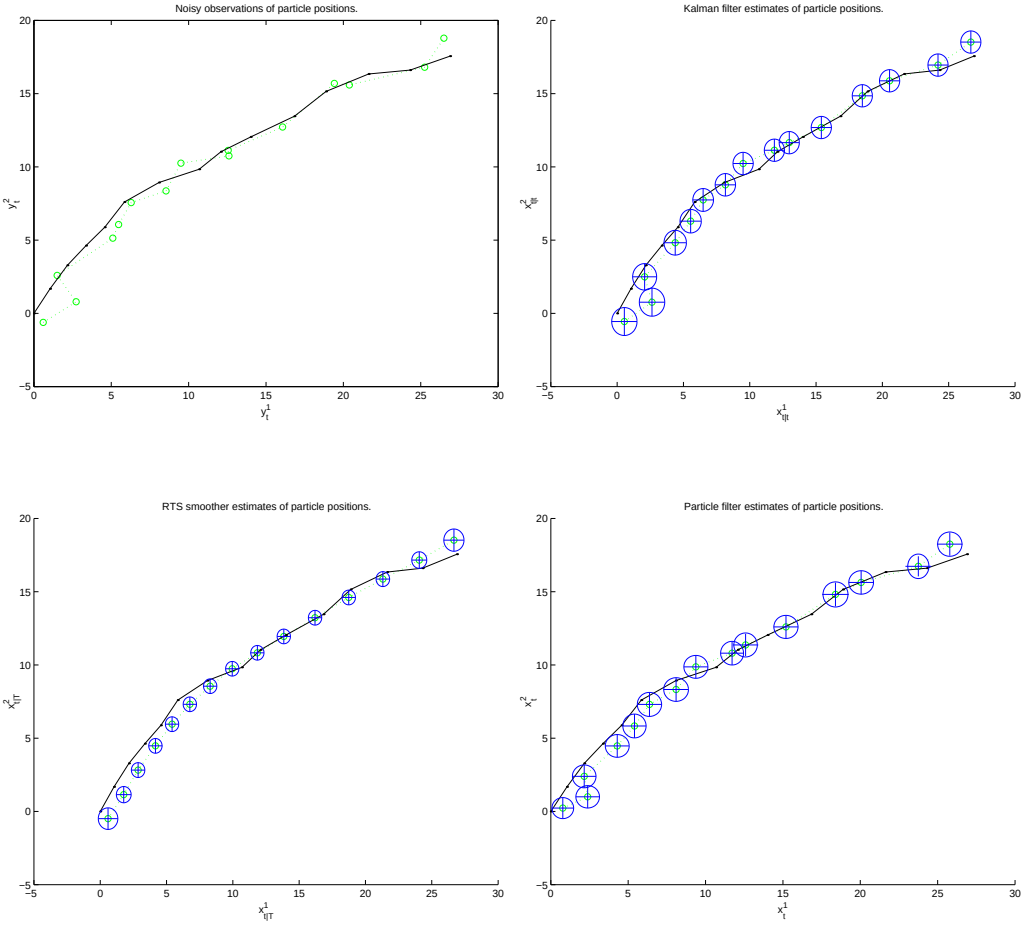
Figure 1: Graphical model of visuomotor tracking. Note that $VisSt$ and $Hand$ nodes are observed in our case.

From Figure 1, we see that visual information enters the system by correcting the control signal at the next time step (the alignment of the time

slice is arbitrary). Given the controller, the observed motor characteristics are independent of all visual information.

3 Implementation and results

I first built a Kalman filter with RTS smoothing ([5]) for tracking positions and velocities, for the V_s and V_o cluster of the graphical model. I implemented a sequential MCMC ([8], [7]) algorithm known as the Particle filter [2], which can handle the case of non-Gaussian emission and transition probabilities. I tested the two algorithms on data sampled from a linear dynamical system with Gaussian emissions (see the file `track.m`; availability discussed in the appendix).



The upper left figure is the observed data, the upper right is the Kalman filtered trajectory with Gaussian confidence elipse, the lower left is the RTS smoothed trajectory, the lower right is the particle filtered trajectory with variance calculated across samples for each dimension (and zero covariance). Note that the Particle filter has nearly constant variance across time. The major disadvantage of the Particle filter is that it is not obvious how to learn the parameters of the sample propagation (transition probabilities) and likelihood weighting (emission probabilities). We could, for example, run the algorithm many times to learn the optimal parameters. Here, I set the parameters equal to the initial Kalman filtering parameters.

Next I implemented the model of Figure 1 using linear dynamical components. I generated some test data from reasonable dynamical considerations. For example, subjects usually move slowly initially, speeds up in the middle of the trajectory, and slow down again towards the end of the movement.

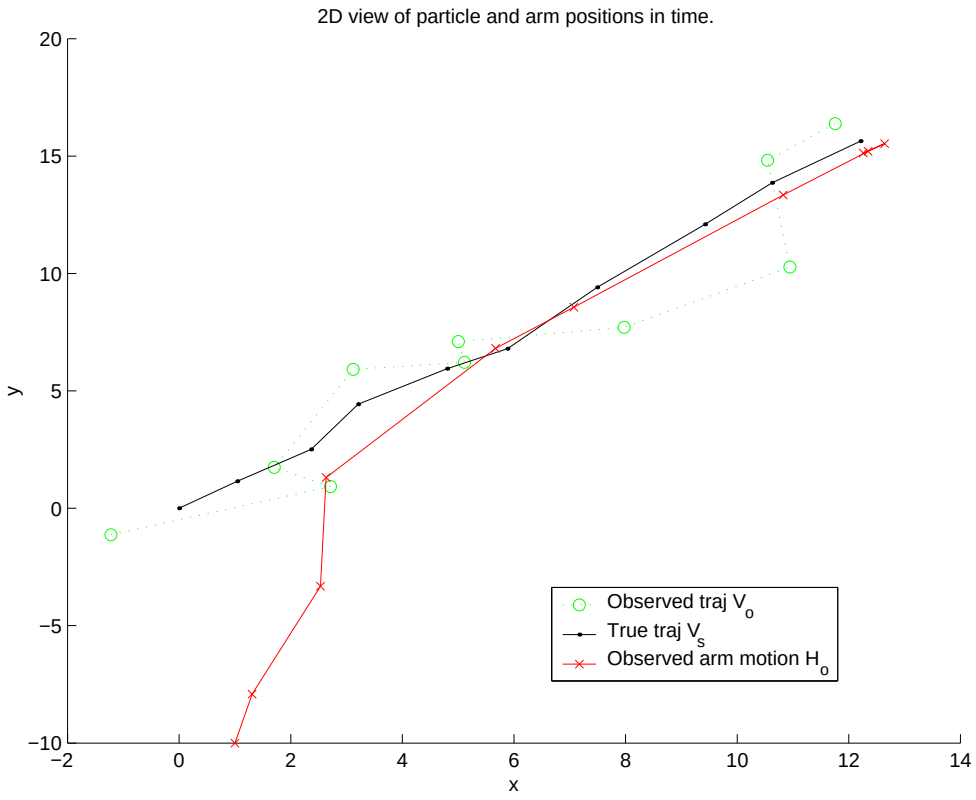


Figure 2: Artificial data used to train the graphical model, shown in 2D.

Figure 3 shows the V_o as green dots and H_o as red lines against the true target trajectory in black. Figure 3 shows the 3D representation. Our model assumes a 6D H_s representing the three spatial dimensions and their derivatives, a 4D V_s for visual tracking of 2D coordinates and their time derivatives, and 3D H_o and 2D V_o, observations of spatial locations. The parameters learned are given in the appendix. A log likelihood trace of EM is shown in Figure 3.

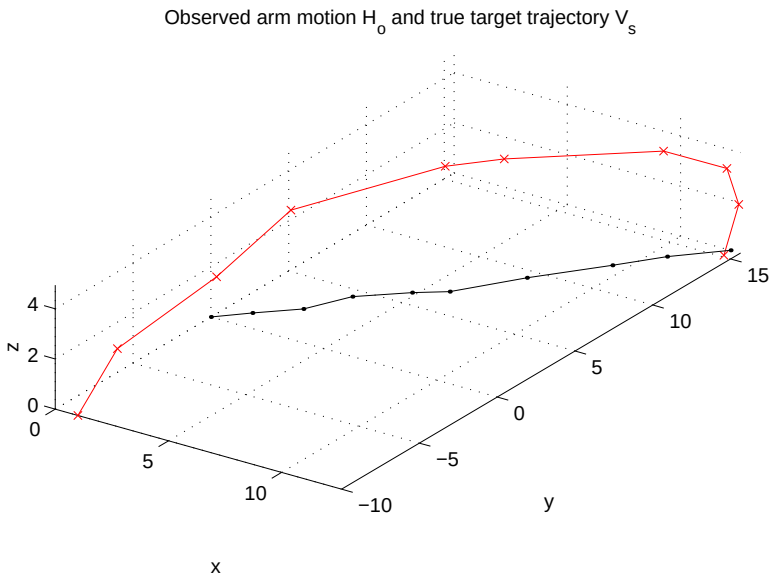


Figure 3: Trajectory of arm movement used to train the model.

4 Future work

It would be fruitful to conduct a linear systems analysis of the stability of the system. We have the visual signal as input, the motor signal as feedback, and the arm as the plant; output is the controller parameters.

The EM Algorithm takes a long time on this model. It would be useful to implement variational inference for this factorized model, as is done in [11]. More analysis of the results is necessary. In particular, we can look into the velocity profiles of the learned model. We investigate how spatial cueing effects come about by looking for consistencies among parameters learned

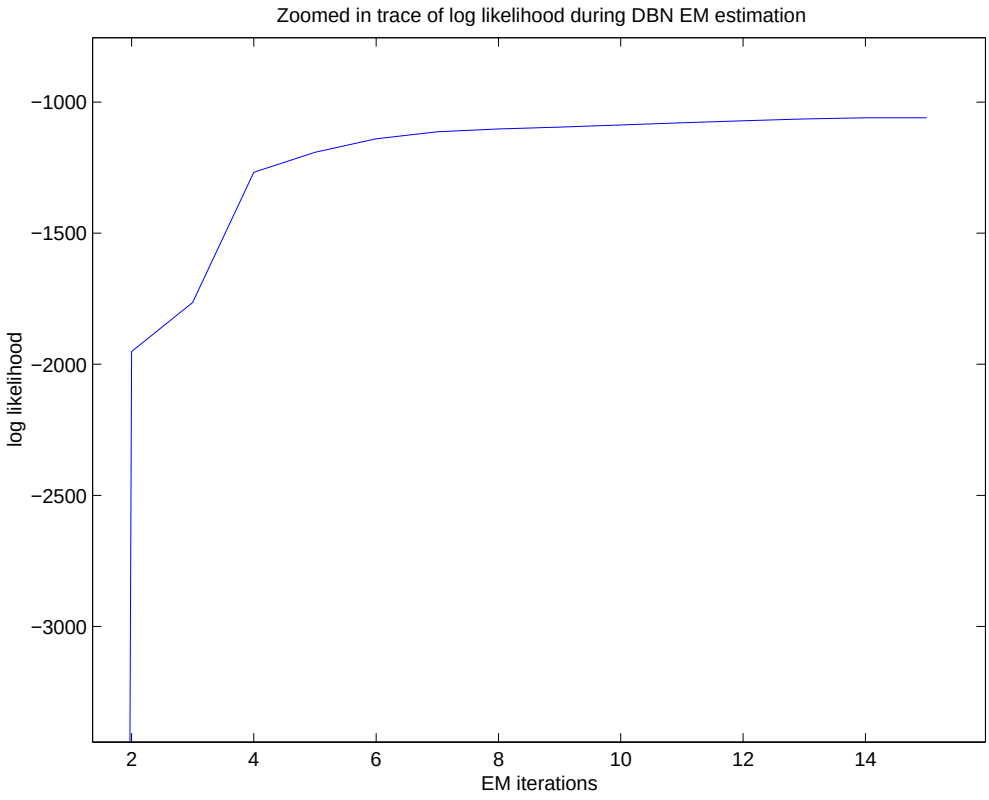


Figure 4: Trace of log likelihood during EM.

for different conditions in an experiment. We can examine the smoothness condition that is learned by the model.

The next step is to build a modular control system within the visuomotor tracking framework. This allows us to recognize and track different objects using a switch model. For example, swatting a fly is much more difficult than catching a baseball. We want to capture this difficulty in parameters of the model learned.

Appendix

Matlab code is found at `http://inst.eecs.berkeley.edu/~rluo/cs281`. See in particular the file `vismotor.m` to start things off. Some portions of the code needs the BNT toolkit written by Kevin Murphy.

The parameters learned by EM for the model in Figure 1 are:

```
node H_s0
m: 0.2125
   -0.0174
   2.0652
   0.2764
   -0.3802
   -0.8041
s: 0.0058 -0.0018 0.0000 0.0000 0.0000 -0.0000
   -0.0018 0.0025 0.0000 0.0000 -0.0000 -0.0000
   0.0000 0.0000 0.0370 -0.0445 0.0802 -0.0502
   0.0000 0.0000 -0.0445 0.1391 -0.2566 0.1239
   0.0000 -0.0000 0.0802 -0.2566 0.4787 -0.2293
   -0.0000 -0.0000 -0.0502 0.1239 -0.2293 0.1175
node V_s0
m: 2.0652
   0.2764
   -0.3802
   -0.8041
s: 0.0370 -0.0445 0.0802 -0.0502
   -0.0445 0.1391 -0.2566 0.1239
   0.0802 -0.2566 0.4787 -0.2293
   -0.0502 0.1239 -0.2293 0.1175
```

```
node H_o
m: 0.0000
   0.0000
   0.0000
s: 0.7829 0.2654 1.2374
   0.2654 1.8653 -0.0461
   1.2374 -0.0461 2.4727
node V_o
m: 0.0000
   0.0000
s: 0.3569 -0.3596
   -0.3596 1.0751
node H_s1
m: 0.0000
   0.0000
   0.0000
   0.0000
   0.0000
   0.0000
s: 0.3214 -0.0319 0.0612 0.1263 0.0506 0.1083
   -0.0319 0.0901 -0.0219 -0.1511 -0.0948 -0.0554
   0.0612 -0.0219 4.6232 8.6694 7.0972 6.5681
   0.1263 -0.1511 8.6694 17.0857 13.6294 12.7710
   0.0506 -0.0948 7.0972 13.6294 11.4500 10.2067
   0.1083 -0.0554 6.5681 12.7710 10.2067 9.7093
node V_s1
m: 0.0000
   0.0000
   0.0000
   0.0000
s: 0.0724 -0.0086 -0.0062 -0.0170
   -0.0086 0.0501 -0.0154 0.0176
   -0.0062 -0.0154 0.0824 -0.0183
   -0.0170 0.0176 -0.0183 0.0369
```

References

[1] M. Haruno, D. M. Wolpert, and M. Kawato. MOSAIC model for sensorimotor learning and control. *Neural Computation*, 13:2201–2220, 2001.

[2] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *European Conference on Computer Vision*, pages 343–356, Cambridge, UK, 1996.

[3] M. I. Jordan. Computational aspects of motor control and motor learning. In H. Heuer and S. Keele, editors, *Handbook of Perception and Action: Motor Skills*. Academic Press, NY, 1996.

[4] M. I. Jordan and D. M. Wolpert. Computational motor control. In M. S. Gazzaniga, editor, *The Cognitive Neurosciences*. MIT Press, MA, 1995.

[5] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the American Society of Mechanical Engineers - Journal of Basic Engineering*, 82(D):35–45, 1960.

[6] A. Leon-Garcia. *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, Reading, MA, 2nd edition, 1994.

[7] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamical systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.

[8] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, June 1953.

[9] K. A. Thoroughman and R. Shadmehr. Learning of action through adaptive combination of motor primitives. *Nature*, 407:742–747, October 2000.

[10] D. M. Wolpert, Z. Ghahramani, and M. I. Jordan. An internal model for sensorimotor integration. *Science*, 269:1880–1882, September 1995.

[11] Y. Wu and T. S. Huang. A co-inference approach to robust visual tracking. In *International Conference on Computer Vision*, Vancouver, Canada, 2001.