Practical numerical methods for stochastic optimal control of biological systems in continuous time and space

Alex Simpkins†, and Emanuel Todorov‡

***Abstract*— Previous studies have suggested that optimal control is one suitable model for biological movement. In some cases, solutions to optimal control problems are known, such as the Linear Quadratic Gaussian setting. However, more general cost functionals and nonlinear stochastic systems lead to optimal control problems to which direct solutions are presently unknown but these solutions would theoretically model behavioral processes. Additionally, in active exploration-based control situations, uncertainty drives control actions and therefore the separation principle does not hold. Thus traditional approaches to control may not be applicable in many instances of biological systems. In low dimensional cases researchers would traditionally turn to discretization methods. However, biological systems tend to be high dimensional, even in simple cases. Function approximation is an approach which can yield globally optimal solutions in continuous time and space. In this paper, we first describe the problem. Then, two examples are explored demonstrating the effectiveness of this method. A higher dimensional case which involves active exploration to learn unobservable parameters and the numerical challenges which arise will be addressed. Throughout this paper, multiple pitfalls and how to avoid then are discussed. This will help researchers to avoid spending large amounts of time merely attempting to solve a problem because a parameter is mistuned.**

I. Introduction

Modeling biological sensorimotor control and learning with optimal control[13][10], especially when uncertainties are considered in exploration/exploitation situations leads to strongly nonlinear and stochastic problems. These are difficult to solve, being often nonlinear, second order, high dimensional partial differential equations. Typical approaches to such problems involve discretizing the equations, defining a set of transition probabilities, and solving the new problem as a Markov Decision Process with Dynamic Programming[2][6]. However, since biological systems tend to operate in high dimensional (and often redundant) spaces, another approach which is gaining favor is to approximate a solution to the continuous problem using continuous function approximators[3]. In a previous paper we derived a function approximation-based nonlinear adaptive control scheme to model the exploration/exploitation tradeoff in biological systems. Here we will elaborate on the method of state augmentation to make a partially observable problem fully observable as well as to address redundancy (a common issue in modeling biological systems, advanced robotics, and decision making processes). In addition, several numerical issues arise when attempting to fit a continuous function in high dimensional space. These include fit quality, number of required functions and feature shapes (if using Gaussians, how one selects the center and variance of each Gaussian), the method of collocation (including rapid convergence of a solution in as little as one iteration and sparse matrices), numerical stability, and performance measures.

This work was supported by the US National Science Foundation

†Department of Mechanical and Aerospace Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0411, email: csimpkin@ucsd.edu

‡Department of Cognitive Science, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0515, email: todorov@cogsci.ucsd.edu

II. Stochastic optimal control problem formulation

Consider the following system (written as a stochastic differential equation) where $x \in \Re^{n_x}$ represents the state, $a(x)$ the (possibly nonlinear) system dynamics, $B(x)u(x)$ the controlled dynamics, with $u(x) \in \Re^{n_u}$, $C(x)$ the covariance matrix for the noise due to Brownian motion ω.

$$dx = \big(a(x) + B(x)u(x)\big)dt + C(x)d\omega, \tag{1}$$

with observation process

$$y = h(x) + d\nu, \tag{2}$$

where the observed quantity y is a function of additive noise driven the Brownian motion ν and $h(\cdot)$, the (possibly nonlinear, again) observation transformation of x. We will choose an infinite horizon formulation for the cost function, which will take on the following form (where the exponential term acts to discount future costs by a discount factor $\alpha > 0$, and keep the cost finite), with cost rate $\ell(x, u)$:

$$V^{\pi}(x) = \int_t^{\infty} e^{-\alpha(s-t)} \ell[x(s), u(s)]ds. \tag{3}$$

This has the advantage of not being dependent on time, and thus lends itself to creating function approximators which estimate the value function and optimal policy. The associated discounted Hamilton-Jacobi-Bellman equation, given by the principle of optimality is

$$\alpha V^*(x) = \min_u \Big\{ \ell(x,u) + (a(x) + B(x)u(x))V_x^*(x) + \frac{1}{2} Tr(C(x)C(x)^T V_{xx}^*(x)) \Big\}, \tag{4}$$

where $Tr(\cdot)$ represents the trace operator, and subscripts denote partial derivatives with respect to what is in the subscript (i.e. V_x reads 'the partial derivative of V with respect to x'). Thus, the problem is to find a control policy $u^*(x)$ which minimizes the right hand side of (4). In this paper we will consider cost rates of the form

$$\ell(x,u) = p(x) + \frac{1}{2}||u||^2, \tag{5}$$

where the first term represents the problem-specific cost contributions, and the second term represents an energy cost to keep control actions finite. With this cost rate defined, the minimization in (7) can be performed analytically, yielding the optimal feedback control law

$$\pi(x) = -B(x)V_x(x). \tag{6}$$

We then substitute (6) into (4) to arive at the minimized HJB equation,

$$\begin{aligned}\alpha V(x) =& p(x) + a(x)^T V_x(x) \\ &+ \frac{1}{2}Tr(C(x)C(x)^T V_{xx}(x)) - \frac{1}{2}||u||^2.\end{aligned} \tag{7}$$

The challenge lies in the following: how do we compute $V_x(x)$ for nontrivial problems?

III. Solution Challenges and Radial Basis Function Approximation Defined

The resulting Hamilton-Jacobi-Bellman equation is typically a second order nonlinear partial differential equation, which is difficult to solve. In some cases it can be solved using methods such as a viscosity solution, minimax solution, and others. In general one has to presently resort to some sort of approximation - either approximating the problem then solving the approximation or approximating a solution via a continuous function fit. Continuous functions offer the advantage of differentiability, which is attractive.

In cases of exploration/exploitation tradeoffs, uncertainty is part of the decision-making process (exploration can be, in fact, *driven* by uncertainty), and so the separation principle does not hold. In these cases, augmenting the state with the filter dynamics results in a higher dimensional but fully observable problem which we formulate within the stochastic optimal control framework.

One way of fitting a function which approximates a solution to the problem at hand is a nonlinear black box method based on radial basis functions. A radial basis function is a function whose value depends on the difference of x from some center point c or from the origin. More specifically, any function that satisfies the relation $\phi(x) = \phi(||x||)$ is referred to as a radial basis function.

A typical function structure is a Gaussian which has the following form:

$$\theta(x, c, S, w) = w * exp\big[-\frac{1}{2}(x-c)^T S(x-c)\big]. \tag{8}$$

Where S is the matrix of the inverse covariances of the Gaussians in each dimension (this determines the 'width' of the Gaussian in a particular dimension). In the present paper we make the matrix S diagonal, thus making the basis functions orthogonal with respect to each dimension. c represents the location of the origin of the Gaussian, and w are weights which are the parameters to be fit. Now if we define the part of the Gaussian which includes everything but the weights as ϕ, given a particular point x, and inverse covariance matrix S, the output of the approximator is then given by a sum over all the basis functions (for each different center and weight),

$$y(x) = \sum_{i=0}^{N-1} w_i \phi_i(x, c_i, S) = \phi(x, c, S)^T \mathbf{w}. \tag{9}$$

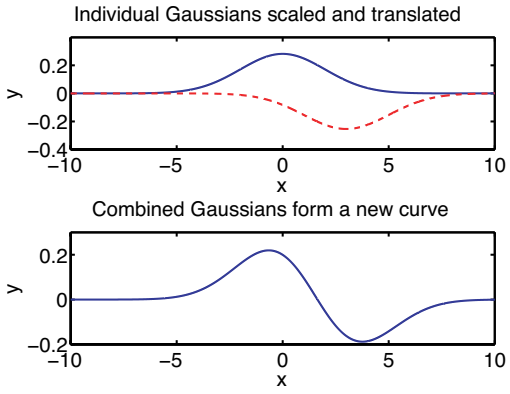

Fig. 1. The top figure displays two different standard Gaussians, with different scaling and centers. The bottom figure displays the sum of the two Gaussians, forming a new function. Extending this methodology to many many Gaussians, one can fit any function theoretically to any desired degree of accuracy.

For this type of approximator, with the c's and variance S fixed, the function is nonlinear in the state but linear in the parameters. Therefore the unknown weights can easily be computed directly by linear regression, given input-output data.

For a sufficient number of basis functions, any function can in theory be approximated to arbitrary accuracy.

One effective method for computing these weights (w_i) is to generate a set of input states (x_i), decide upon a set of centers (c_i) compute the exponential phi functions for all centers, then use linear regression to compute the weights. This is referred to as the method of collocation. This can be applied to compute a continuous approximation to the Hamilton-Jacobi-Bellman equation, and thus, the resulting optimal control policy for difficult problems.

IV. Solution Method - the FAS (Function Approximation Scheme)

Here we will discuss a general method (the reader is referred to [11] for an in-depth description and application of this method to an exploration/exploitation problem for biological control) for approximating an optimal control policy in continuous space and time using collocation. The basic idea follows.

Begin with (9), and compute the first and second derivatives with respect to x,

$$V(x, w) = \sum_i w_i \phi^i(x) = \phi^T(x) w, \tag{10}$$

$$V_x(x, w) = \sum_i w_i \phi_x^i(x) = \phi_x^T(x) w, \tag{11}$$

$$V_{xx}(x, w) = \sum_i w_i \phi_{xx}^i(x) = \phi_{xx}^T(x) w. \tag{12}$$

Substitute into (7) and simplify, then define

$$\mathbf{M} = \Big\{ M_{j,i} = \Big(\alpha \phi^i(x_j)^T - a(x_j)^T \phi_x^i(x_j)^T - \frac{1}{2} tr\{ C(x_j) C(x_j)^T \phi_{xx}^i(x_j)^T \} \Big), \forall i, j \Big\}, \tag{13}$$

$$\mathbf{d} = \Big\{ d_j = p(x_j) - \frac{1}{2} \|\pi(x_j)\|^2, \forall j \Big\}. \tag{14}$$

Now we have an iterative least squares problem,

$$\mathbf{M} w = \mathbf{d}. \tag{15}$$

The algorithm is initialized by generating a set of random states (x_i) which span the space of interest. There should be at least as many states as the number of features (Gaussians, quadratics, etc) and Gaussian centers c_j. Then compute $\phi(x_i, c_j)$, $\phi_x(x_i, c_j)$, and $\phi_{xx}(x_i, c_j)$ and all the results stored. The weights are initialized, the initial guess for the optimal policy computed and fixed, then the least squares is performed and a new policy is computed. The normalized difference between the left and right sides of (15) is used as the stopping criterion, along with some sanity checks for divergence.

V. Limitations of function approximation and computational methods for optimal solutions

Though the method of collocation with radial basis functions is an effective way to approximate an arbitrary function, there are limitations. One such limitation is that the number of Gaussians, given a fixed variance and location of centers required to approximate a given function may be very large[8]. This problem can be mitigated by including relevant features of the system being studied (to be described in depth below).

A second limitation is that, in the stochastic case, it is challenging to know when a true minimum is reached versus some local minimum far from the solution due to noise. Additionally, one does not wish to fit the noise of a system, only the relevant features. There are several approaches to mitigate these types of problems. This is the well-known issue of generalization and overfitting (namely, we want to increase the generalization of our function fit, and reduce overfitting).

A. Large number of parameters

If one uses some insight into the system being modeled and then creates some features which have an appropriate structure, function approximation can be performed with many less parameters, and thus less computational expense. In this way many less Gaussians can be used. A common feature to use if the system has global and local fluctuations is a polynomial term such as a quadratic (all terms $x_a * x_b$ and x_a)

It can be shown that there is a relationship between the number of required Gaussians to achieve appropriate level of coverage in all dimensions by

$$n_g = \prod_i \frac{p_i}{\sigma_i}, \tag{16}$$

where σ_i is the standard deviation of each Gaussian (assuming fixed widths) in dimension i and p_i is the function space width in dimension i. Thus, if the order of p is large and small fluctuations are expected as well as large global ones, it may prove useful to have a nonuniform Gaussian variance in each dimension (by having sets of variances), or include a global quadratic feature. The global quadratic's second order term weight matrix is symmetric, and thus uniquely determined by its upper triangular portion. The general equation is, with constant matrices D, F, and P of the appropriate sizes,

$$y_q = D + F^T x + x^T P x, \tag{17}$$

and

$$P = P^T. \tag{18}$$

To find the weights P, it helps numerically to compute the coefficients for the upper or lower triangular parts of P, then rebuild the matrix later for simulation by

$$P = P_L + P_U, \qquad P_L = P_U^T. \tag{19}$$

B. Generalization/overfitting

The traditional method of maximizing generalization and minimizing overfitting is by using validation sets. This consists of splitting the randomized collocation points (or randomly regenerating more sets), then testing the computed fit on the new data set, since a good fit over a space should be an effective minimum for all points within the space, unless noise has been fit. In this case, it is best to reduce the number of Gaussians or to compute an average over data points.

C. Numerical errors (differentiation, condition number, and sparsity)

Numerical errors are likely to arise in many stochastic problems due to the second order term. Differentiation is an un-smoothing process and so functions with analytical derivatives are always preferable to numerical differentiation if possible. Fortunately Gaussians and quadratics are analytically differentiable. One must take care with the second order derivatives however as dimensionality is such that the matrices become tensors, and during implementation this is typically a point where bugs delay results. Not only must the equations be correctly computed on paper, but a double check of the actual code will prevent many headaches.

Once the algorithm has been implemented, a computational check of the $\mathbf{M}$ matrix condition number gives a measure of the probable accuracy of the least squares operation to be performed. If necessary, computational accuracy can be improved without a significant loss of efficiency in cases where the condition number is moderate to poor. See, for example, [9],[4].

Finally, if the widths of the Gaussians are too large, the weights will tend to alternate in a large and numerically (the more Gaussians that overlap and the wider the overlap) unstable fashion. Thus, considering Section (V-A) and the following section regarding simple visualization and numerical overlap computation will be significant. If the widths of the Gaussians are too narrow, there will be space that cannot be covered by the function approximator (no matter what weight is computed, the approximator will output zero). This results in a sparse coefficient matrix ($\mathbf{M}$), and so is easily checked. The general rule is that some overlap is preferable to open space if good approximation is desired.

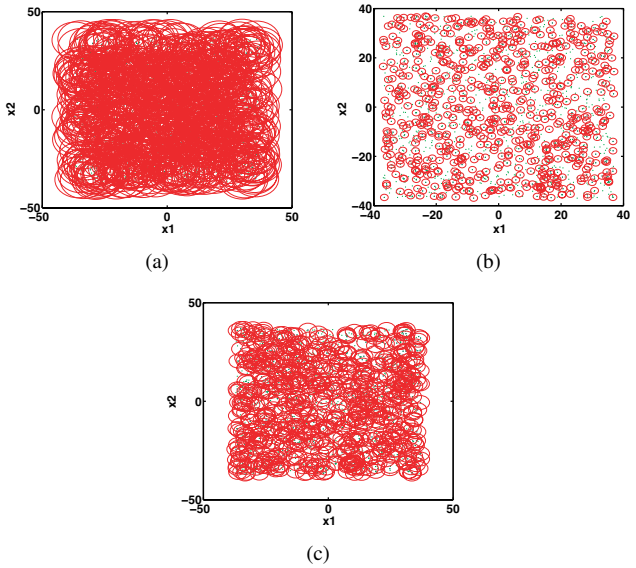

Fig. 2. A low dimensional visualization of the space coverable by the Gaussians for a function fit. The green dots represent the points covering the space, and the red circles represent the standard deviation of the centers of the Gaussians over the space scaled to minimize overlap. Too much redundancy results in an indeterminate problem.

Fig. 3. One typical colormap for Gaussian coverage - green is appropriate overlap, yellow is some overlap, red is too much overlap. In other words, black - no cover, green- one Gaussian, move towards red as more and more overlap occurs.

D. How to determine unkown basis function constant parameters - centers and widths

In order to determine if the state space is completely covered by the Gaussians, numerical computations can be confirmed with visualization methods. One effective scheme for setting the width of the Gaussians is to compute the mean absolute distance between centers and set the covariance to approximately 68% of the maximum value. The simplest way to check that the Gaussians cover all the space with no gaps is to plot the 2σ radius relative to the center of each Gaussian as a circle about center points, plotting two dimensions at a time (3). One can then extend this to include false color representation with natural mappings which use the natural processing of the human perceptual system to help - map the 'quantity' of covered space as colors (black - no cover, green - one Gaussian, move towards red as more and more overlap occurs, see figure 2).

E. Measures of numerical fit quality

Numerical consistency of the computed fit can be checked by using the weights to attempt to recompute the right side of the $Mw = d$ fit equation. The two results d and $\hat{d}$ are plotted and compared. The number

$$\left|\left|\frac{Mw - d}{max(||d||_1)}\right|\right|_2, \tag{20}$$

gives a single value measure of fit quality. The L_1 norm in the bottom of the fraction scales the fit to essentially a proportion of fit. Otherwise the numerical value returned by this check is deceiving since each iteration tends to grow.

F. Numerical stability

In order to assure numerical stability of the weights, one can normalize the weights to one or another constant maximum after computing the least squares fit. What this does is prevent the next iteration's d from exponentially increasing in the norm of the amplitude. Then the maximum amplitude scale of the w's can be determined by another least squares fit after convergence is achieved.

G. Visualize the value function and control

When a reasonable solution has been determined, or one that appears reasonable, it is important to visualize the resulting value function and control actions mapped over the state space. Though the system may be high dimensional, most real-world applications have a low dimensional interpretation which is possible. For example, in our recent work [11], we applied this methodology to an eight dimensional problem. A reasonable control solution could be visually confirmed versus an unreasonable solution by creating a 3d surface plot of the cost versus two-dimensional states in slices. An unreasonable or poor solution would yield no consistent shape, merely an appearance of noise, whereas a reasonable solution would yield farely smooth shapes. A smooth shape may not be the numerically optimal solution possible, but in practice a smooth shape was produced only when a good fit was found. Reasonable solutions tend to be logical.

Similarly, the control response surface provides a positive measure of consistency. Both surfaces should be of reasonable maximum amplitude, though the control action is more intuitive in terms of the programmer's being able to recognize a reasonable state-dependent action (i.e. if a robot arm is being controlled, and given a particular state, if a control action which is near infinite is computed despite a penalty on control energy, it is likely that the designer is witnessing an inconsistent solution).

H. Simulation performance, and repeated measures

The most important measure of a control's goodness of fit may be in terms of performance. This means that a control should be computed, then a short simulation performed and a performance criterion evaluated, such as the mean-square-error or 2-norm of the error between reference and actual output.

Given a stochastic system, it is important to perform repeated solution attempts using the same parameters in order to confirm a particular solution. For example, one setting of initial conditions may lead to a poor solution in one solution attempt, and a good solution in another. But averaged over several solution attempts given the same parameters/initial conditions, a good versus bad solution is evident since in general the problem will be solved well for good settings. It is in this way that repeated measures are essential in stochastic problems for determining a solution's validity.

VI. Examples

Two examples will be considered, and a third referred to from another of the authors' papers. The first example is used to apply the function approximation solution to a well-studied problem. This problem is the limited torque pendulum swing-up problem. The second example demonstrates the reduced sensitivity of the function approximation-based solution to the curse of dimensionality by demonstrating a high dimensional solution to an active exploration problem. The problem is the same as the first pendulum example, but with the added challenge that an unobservable mapping is part of the cost functional, and must be estimated to make the problem fully observable.

A. Example 1: 1-DOF Pendulum swing-up problem

Consider a bar-shaped pendulum with limited torque. When the maximum torque the actuator can exert is less than $mgl/2$, (where m is the mass of the pendulum bar, l is the length, and g is the acceleration due to gravity), the pendulum must undergo multiple swings in order to attain the necessary momentum to swing to a vertical position. Additionally, the controller must anticipate the peak of the arc and begin deceleration at the appropriate time in order for the pendulum to avoid over-rotating and falling again.

The dynamics of the system are given by

$$J\ddot{\theta} + H\dot{\theta} + G(\theta) = \tau. \tag{21}$$

In this case $J = ml^2$ is the inertia of the link, $H\dot{\theta}$ is the velocity-dependent friction, $G = mglcos(\theta)$ is the torque due to gravitational force, and τ is the torque applied to the pendulum externally (via an actuator such as a motor). This can be arranged in the optimal control framework we set out in the following way. Define the state x as

$$x = \begin{bmatrix} \theta & \dot{\theta} \end{bmatrix}^T, \tag{22}$$

$$a(x) = \begin{bmatrix} \dot{\theta} \\ -J^{-1}(H\dot{\theta} + G) \end{bmatrix}, \tag{23}$$

$$B = \begin{bmatrix} 0 \\ J^{-1} \end{bmatrix}, \tag{24}$$

$$u = \tau. \tag{25}$$

Now we can write the dynamics in the standard form. Here we make the problem deterministic to simplify comparisons between 'what to do and what not to do steps:' This makes the second derivative drop out, leaving us with

$$\dot{x} = a(x) + Bu(x). \tag{26}$$

with parameters $m = 1kg$, $l = 1m$, $g = 9.81m/s^2$, and $H = 0$.

The cost rate was determined by a combination of a velocity penalty, a control energy penalty, and a position penalty (for angles other than $\pi/2$). All trials had random initial conditions, and were considered successful if the pendulum achieved a vertical orientation for an indefinite period of time ($t > 15sec$), achieving the vertical position in under 10 seconds (this time is arbitrary depending on the degree of under-actuation). The cost rate is then of the form:

$$\ell(x, u) = k_\theta(\theta - \pi/2)^2 + k_{\dot{\theta}}(\dot{\theta})^2 + \frac{1}{2}u^2, \tag{27}$$

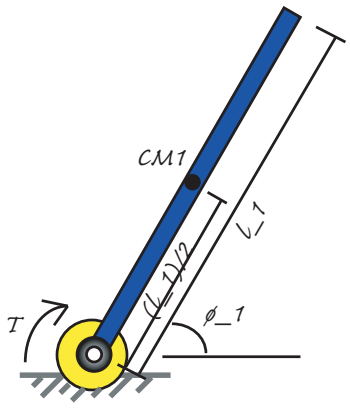

Fig. 4. Basic diagram of 1-link pendulum. The state space is a cylinder with position range $\pm\pi$

where the k's are gains which can be adjusted to tailor behaviors if desired. In all example problems presented in this paper, the constants are all the same value for simplicity.

B. Example 2: 1-DOF Pendulum problem with an uncertain wandering mapping. Solving partially observable nonlinear exploration/exploitation problems to which the separation principle does not apply

Consider now the same problem with a slight change- there is an unobservable and continuously wandering mapping between the observation of the base angle and the actual angle. In physical terms this could be interpreted as the base to which the pendulum is attached undergoing a continuously random rotation about a parallel axis with the pendulum base. This is similar to our previous study in [11]- except that the motion is not damped, but instead undamped Brownian motion. Now we have a partially observable problem, since the cost function includes the angle. By taking the expectation of the uncertain term, and augmenting the state with the mean and covariance of the estimated quantity, one can create a fully observable, but higher dimensional problem solvable with our FAS scheme.

$$\begin{aligned} \ell(x, u) \approx & E(k_\theta||\theta - \pi/2||^2 + k_{\dot{\theta}}||\dot{\theta}||^2 + \frac{1}{2}||u||^2) \\ = & k_\theta(||\widehat{m}\theta - \pi/2||^2 + \theta^2\Sigma) + k_{\dot{\theta}}||\dot{\theta}||^2 + \frac{1}{2}||u||^2. \end{aligned} \tag{28}$$

The observation process is given by

$$dy = m(t)\theta(t)dt + d\omega_y. \tag{29}$$

Assuming the prior over the initial state of the mapping is Gaussian, with mean $\widehat{m}(0)$ and covariance $\Sigma(0)$, then the posterior remains Gaussian for all $t > 0$ for $m(t)$. Given the additive white noise model (where the properties of the noise and disturbances do not change over time, and $d\omega$ is a white, zero-mean Gaussian noise process with covariance Ω_y), the optimal map estimate is propagated by the Kalman-Bucy filter[5][1][12],

$$\begin{aligned} d\widehat{m} &= K\left(dy - \widehat{m}(t)\theta(t)dt\right), \\ K &= \Sigma(t)\theta(t)^\intercal\Omega_y^{-1}, \\ d\Sigma &= \Omega_m dt - K(t)\theta(t)\Sigma(t)dt. \end{aligned} \tag{30}$$

The mean of the estimate is $\widehat{m}(t)$ and the covariance is $\Sigma(t)$. Ideally we would like to augment our state with $m(t)$, but we can only estimate m, so now our composite state vector will be

$$x(t) = [\theta(t); \dot{\theta}(t); \widehat{m}(t); \Sigma(t)], \tag{31}$$

and our stochastic dynamics can be written in the form of (1), with uncontrolled dynamics representing the pendulum and the evolution of the covariance matrix,

$$a(x) = \begin{bmatrix} \dot{\theta} \\ J^{-1}(-H\dot{\theta} - G(\theta, \dot{\theta})) \\ 0 \\ \Omega_m - \Sigma^2\theta^2\Omega_y^{-1} \end{bmatrix}, \tag{32}$$

controlled dynamics,

$$Bu = \begin{bmatrix} 0 & J^{-1}\tau & 0 & 0 \end{bmatrix}, \tag{33}$$

and finally the noise-scaling matrix,

$$C(x) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \Sigma\theta\Omega_y^{-1} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{34}$$

Now we have a nonlinear stochastic optimal control problem defined by (1), (28), and (31)-(34). An approximation to the optimal control policy can be created using our FAS algorithm.

VII. Results

A. 1-link pendulum swing-up

The single link pendulum swing-up task results show that the FAS can indeed perform nonlinear control for a nontrivial problem quite effectively. Given one hundred trials with random start points, the average time to vertical was under five seconds, with a final error under 1e-3 Radians. A typical policy function representation is shown in Figure 5(a), using one hundred basis functions.

The techniques presented for fitting the Gaussians were used in order to determine the constant parameters of the basis functions. The covariance of the Gaussians was determined by computing over the state space the average distance between randomly generated Gaussian centers, as discussed in Section (V-D), then computing the variance in each dimension required to make the average overlap occur at 68% of the value of the Gaussians. This works well in practice, and a quick visualization in each pair of dimensions (or a subset if one has high dimensional space) gives a sense of how good the coverage is. One could automate this process by computing some metric such as the two-norm of the volume of empty space, then iterating over the variance to minimize that value. The solution only takes, if well vectorized in Matlab, a few seconds to compute, and several variables can be pre-computed and stored, then accessed repeatedly for optimizations. Generally it is fairly clear when there is a large quantity of space, if one generates a representative circle around the center of all the Gaussians and visualizes the results as seen in Figure 2. The discount factor is another parameter which must be carefully tuned. However, the problem will tend to be minimally sensitive to small differences in the discount factor. It is key to consider how far into the future should be incorporated into the control policy. If one wishes a more 'greedy' response to the current state and goal, a heavily discounted problem will achieve this. The opposite is true for incorporating the future into the present control state. Here we want the controller to swing the pendulum back and forth in order to gain enough momentum to achieve a vertical state with minimal energy input, rather than just act at the moment to minimize instantaneous cost. Thus a problem which is not too heavily discounted is required. Intuition, and quick checks with computing the solution the two ways (minimal/maximal discounting) and observing performance were clear enough to converge quickly upon a discounting factor which was effective to produce the required behavior.

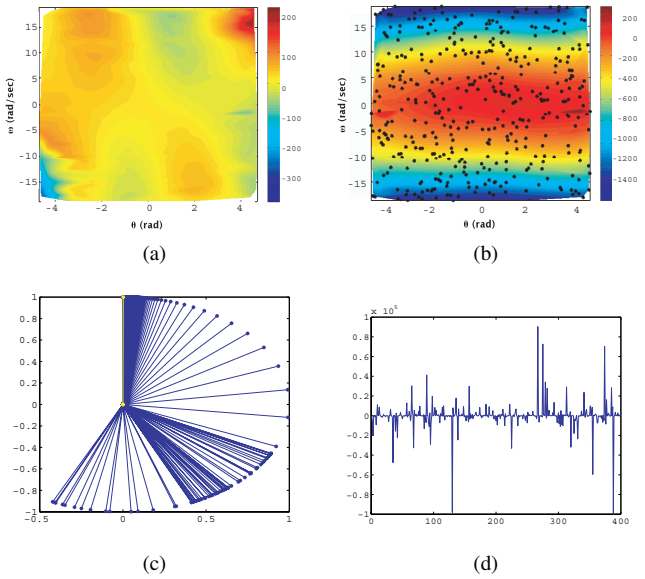

Fig. 5. (a) Shows the surface of the control action space. (b) Shows the random cloud of points used to fit the weights and thus a surface defining the cost function in the method of collocation. (c) Shows a typical swing-up trial, with the initial position in this case at -0.89 Radians. The final error is 1e-3 Radians, and the swing-up time is under 5 seconds (measured as $t_{s.u.t.} = t(error < 0.001Rad)$ (d) Shows a plot of typical weight values if the Gaussian variance is slightly too large for the distance between Gaussian centers. Note that the weights have large opposing values to balance each other.

B. 1-link pendulum swing-up with uncertainty

The pendulum was still able to perform the experimental task when driven by the uncertain mapping, in addition to the pendulum swing-up challenge. This problem is four dimensional, which is difficult to solve with discretization methods, yet our FAS could solve this with one hundred basis functions (this suggests fewer bases could be effectively used on the previous problem, but minimal basis function application was not the goal). The Kalman filter effectively estimated the unobservable mapping (Figure 6(a)), keeping the pendulum swing-up task possible. The parameter is only shown fluctuating in a small range, but positive or negative values are acceptable and posed little problem for the FAS algorithm during experimentation.

Figure 6(d) shows exploratory actions being injected into the system by the policy after convergence to the vertical position. This is done to highlight the pseudorandom behavior triggered

by the covariance term. By this time the map parameter was being tracked well by the FAS algorithm, and so the actions are small due to the covariance term being small (Figure 6(b)).

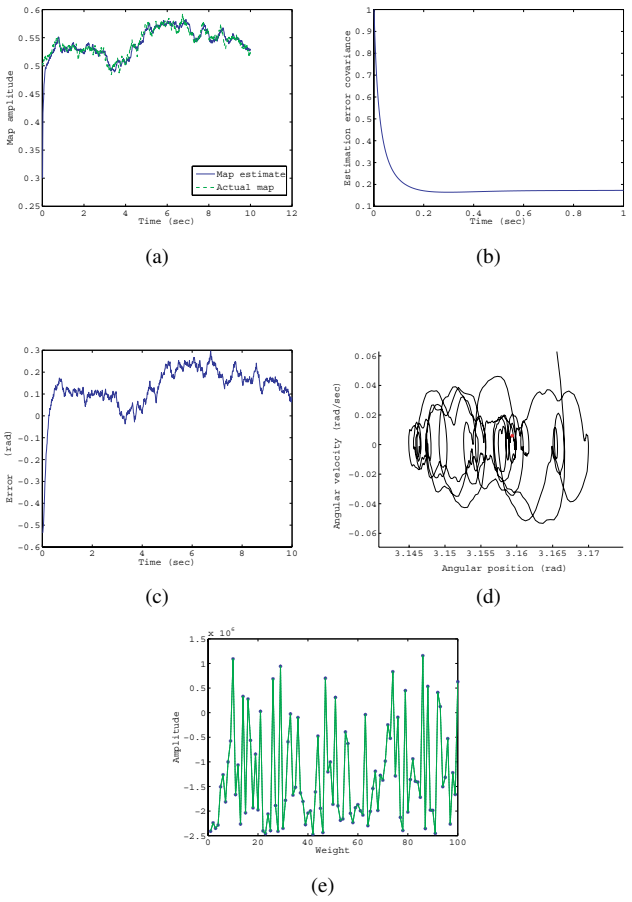

Fig. 6. (a) Map versus estimate for pendulum swing-up problem with uncertainty. (b) Estimation error covariance. Note the rapid drop in uncertainty during this trial. (c)position error $(m(t)\theta - \pi/2)$. (d) Small exploratory movements in position-velocity space. (e) An example of the typical numerical fit achieved in two iterations. The normalized fit error is 1.1e-14.

VIII. Conclusion

In this paper we addressed several topics related to function approximation in continuous time and space applied to nonlinear stochastic optimal control problems suitable for modeling biological systems. The FAS algorithm can effectively approximate optimal policies for active exploration-type problems, as we demonstrated with the pendulum on a randomly rotating base problem. The fact that the FAS algorithm can produce a viable control policy in the latter case, using the state augmentation method will be very useful for modeling sensorimotor learning and control. In our previous paper we showed that this method can effectively deal with redundancy, a common issue in motor control, as well as higher dimensional systems.

The shortcomings of these methods are due to the significant human effort (parameter adjustment) that must take place to implement them effectively. This paper also deals with many of these shortcomings and suggests, where possible, numerical methods such as using performance criteria to pose optimizations over those tuning parameters. This can reduce the manual tuning that makes implementing many reinforcement learning and approximately optimal control policies difficult and time consuming.

Another future direction of this work is to combine global and local methods. Previously, iterative quadratic approximation methods were developed in our laboratory [14], [7]. The local methods suffer from the need for good initialization, but are very effective when in moderately close proximity to a solution. Thus it is reasonable to suggest that an effective algorithm would use the global method to initialize the local method and provide a check at each time step.

In the near future, we will be implementing these control policies in several novel robots which the authors have developed to further explore the benefits of active exploration and to model human sensorimotor learning.

In some senses, all learning can be reduced to the estimation of observable or unobservable functions and parameters. Optimal control has been successfully applied in many simple settings for modeling sensorimotor control. This extension to redundant and unobservable systems is very powerful. In this context, estimation and control not only coexist, but they are also intermixed, driving each other to achieve an otherwise impossible control objective. A methodology such as presented here which specifically makes use of rather than attempting to average out the uncertainty allows a more broad range of problems to be addressed.

References

[1] B. Anderson and J. Moore. *Optimal Filtering*. Prentice Hall, 1979.

[2] D. Bertsekas. *Dynamic programming and optimal control*. Athena Scientific, Bellmont, MA, 2nd edition, 2001.

[3] K. Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12:219–245, 2000.

[4] J. Ferziger. *Numerical Methods for Engineering Application*. John Wiley and Sons, New York, NY, 2nd edition edition, 1998.

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

[6] H. Kushner and P. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer-Verlag, New York, New York, 1992.

[7] W. Li and E. Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. *Proc. of the 1st International Conference on Informatics in Control, Automation and Robotics*, 1:222–229, August 2004.

[8] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, PTR, Upper Saddle River, NJ, 2nd edition edition, 1999.

[9] C. Pozrikidis. *Numerical Computation in Science and Engineering*. Oxford University Press, 1998.

[10] S.H. Scott. Optimal feedback control and the neural basis of volitional motor control. *Nat Rev Neurosci*, 5:532–546, 2004.

[11] A. Simpkins and E. Todorov. Optimal tradeoff between exploration and exploitation. American Control Conference, IEEE Computer Society, 2008.

[12] H. W. Sorenson, editor. *Kalman Filtering: Theory and Application*. IEEE Press, 1985.

[13] E. Todorov. Optimality principles in sensorimotor control. *Nature Neuroscience*, 7:907–915, 2004.

[14] E. Todorov and W. Li. A generalized iterative lqg method for locally-optimal feed-back control of constrained nonlinear stochastic system. *Proc. of American Control Conference*, pages 300–306, June 2005.