Motor Learning at Intermediate Reynolds Number: Experiments with Policy Gradient on a Heaving Plate

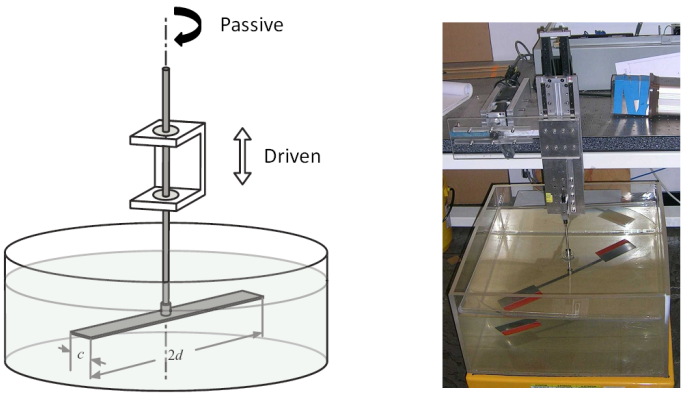
John W. Roberts, Jun Zhang and Russ Tedrake

Abstract This work describes the development of a model-free reinforcement learning-based control methodology for the heaving plate, a laboratory experimental fluid system that serves as a model of flapping flight. Through an optimized policy gradient algorithm, we were able to demonstrate rapid convergence (requiring less than 10 minutes of experiments) to a stroke form which maximized the propulsive efficiency of this very complicated fluid-dynamical system. This success was due in part to an improved sampling distribution and carefully selected policy parameterization, both motivated by a formal analysis of the signal-to-noise ratio of policy gradient algorithms. The resulting optimal policy provides insight into the behavior of the fluid system, and the effectiveness of the learning strategy suggests a number of exciting opportunities for machine learning control of fluid dynamics.

1 Introduction

The possible applications of robots that swim and fly are myriad, and include agile UAVs, high-maneuverability AUVs and biomimetic craft such as ornithopters and robotic fish. However, controlling robots whose behavior is heavily dependent upon their interaction with a fluid can be particularly challenging. Whereas in many regimes robots are able to make use of either accurate dynamical models or easy-to-stabilize dynamics, in the case of flapping and swimming robots neither of these conditions apply. The models available to swimming and flying robots tend to be very limited in their region of validity (such as quasi-steady models of fixed-wing aircraft), very expensive to evaluate (such as direct numerical simulation of the governing equations) or almost completely unavailable (as is the case with interactions between a complex flow and a compliant body).

Here we investigate the problem of designing a controller for a specific experimental system: the heaving plate (see Figure 1). This setup is a model of forward flapping flight developed by the Applied Math Lab (AML) of the Courant Institute



(A) (B)

Fig. 1: (A) Schematic of experimental flapping system. The white arrow shows the driven vertical motion determined by the controller, while the black arrow shows the passive rotational motion resulting from the fluid forces. Figure from (Vandenberghe et al., 2006), with slight modifications. (B) Original experimental flapping system. The wing shown is an earlier design used to study the fluid system, while the wing used for this work was a simple rigid rectangle.

of Mathematical Sciences at New York University (Vandenberghe et al., 2004; Vandenberghe et al., 2006). This system consists of a symmetric rigid horizontal plate that is driven up and down along its vertical axis and is free to rotate in the horizontal plane. Previous work demonstrated that driving the vertical motion with a sinusoidal waveform caused the system to begin rotating in a stable "forward flight" (Vandenberghe et al., 2004). Here we will investigate a richer class of waveforms in an attempt to optimize the efficiency of that forward flight. This system is an excellent candidate for control experiments as it is perhaps the simplest experimental model of flapping flight, is experimentally convenient for learning experiments[1], and captures the essential qualities of the rich fluid dynamics in fluid-body interactions at intermediate Reynolds numbers. While accurate Navier-Stokes simulations of this system do exist for the case of a rigid wing (Alben & Shelley, 2005), they require a dramatic amount of computation[2]. As such, model-based design of an effective controller for this system is a daunting task.

Model-free reinforcement learning algorithms, however, offer an avenue by which controllers can be designed for this system despite the paucity of the avail-

[1] Particularly when compared to a flapping machine that falls out of the sky repeatedly during control design experiments.

[2] At the time, this simulation required approximately 36 hours to simulate 30 flaps on specialized hardware (Shelley, 2007).

able models by evaluating the effectiveness of a controller directly on the physical system (Peters et al., 2003; Tedrake et al., 2004). In fact, learning through experimentally collected data can be far more efficient than learning via simulation in these systems, as high-fidelity simulations can take orders of magnitude longer than an experiment to obtain the same data. One of the limitations of this experimental approach, however, is the potential difficulty in directly measuring the state of the fluid, which, naively, is infinite-dimensional (a continuum model). Therefore, the problem is best formulated as a partially-observable Markov decision process (POMDP) (Kaelbling et al., 1998). In the experiment described here, we solve the POMDP with a pure policy gradient approach, choosing not to attempt to approximate a value function due to our poor understanding of even the dimensionality of the fluid state.

The difficulty in applying policy gradient techniques to physical systems stems from the fact that model-free algorithms often suffer from high variance and relatively slow convergence rates (Greensmith et al., 2004), resulting in the need for many evaluations. As the same systems on which one wishes to use these algorithms tend to have a high cost of policy evaluation, much work has been done on maximizing the policy improvement from any individual evaluation (Meuleau et al., 2000; Williams et al., 2006). Techniques such as Natural Gradient (Amari, 1998; Peters et al., 2003) and GPOMDP (Baxter & Bartlett, 2001) have become popular through their ability to converge on locally optimal policies using fewer policy evaluations.

During our experiments with policy gradient algorithms on this system, we developed a number of optimizations to the vanilla policy gradient algorithm which provided significant benefits to learning performance. The most significant of these is the use of non-Gaussian sampling distributions on the parameters, a technique which is appropriate for systems with parameter-independent additive noise (a common model for sensor-driven observation noise). We make these observations concrete by formulating a signal-to-noise ratio (SNR) analysis of the policy gradient algorithms, and demonstrate that our modified sampling distributions improve the SNR.

With careful experiments and improvements to the policy gradient algorithms, we were able to reliably optimize the stroke form (i.e., the periodic vertical trajectory of the plate through time) of the heaving plate for propulsive efficiency to the same optima from different initial conditions (in policy space) in less than ten minutes of trial-and-error experiments on the system. The remainder of this chapter details these experiments and the theory behind them.

2 Experimental Setup

The heaving plate is an ideal physical system to use as a testbed for developing control methodologies for the broader class of problems involving fluid-robot interactions. The system consists of a symmetric rigid titanium plate pinned in the horizontal plane. It is free to rotate about the point at which it is pinned with very

little non-fluid related friction. The control input is the vertical position of the plate z (note that the input is the kinematic position, not the force on the plate). This vertical driven flapping motion is coupled through the fluid with the passive angular rotation such that once the system begins to flap, fluid forces cause it to spin. To ensure that the rotational dynamics are dominated by fluid forces, and not friction due to the slip ring and bearings used to mount the wing, the decay in angular speed was measured for both a submerged and non-submerged wing. Figure 2 shows the result of this experiment, demonstrating that the fluid forces are far more significant than the bearing friction in the regime of operation.

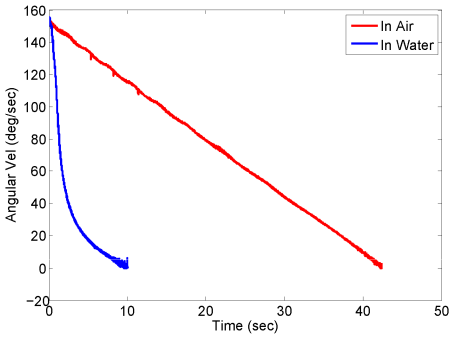


Fig. 2: Comparison of fluid and bearing friction for large rigid wing. The foil (not undergoing a heaving motion) was spun by hand in both air and water, with the angular velocity measured as a function of time. Five curves for both cases were truncated to begin at the same speed then averaged and filtered with a zero-phase low-pass filter to generate these plots. The quick deceleration of the wing in water as compared to air indicates that fluid viscous losses are much greater than bearing losses. At the speeds achieved at the convergence of the learning, the frictional forces in water were over six times that seen in air.

The only measurements taken were the vertical position of the plate (z) and angular position of the plate (x) by optical encoders and the axial force applied to the plate (F_z), obtained by an analog tension-compression load cell. While these measurements were sufficient for formulating interesting optimization problems, note that the fluid was never measured directly. Indeed, the hidden state of the fluid, including highly transient dynamic events such as the vortex pairs created on every flapping cycle, provide the primary mechanism of thrust generation in this regime (Vandenberghe et al., 2004). Sensing the state of the flow is possible using local flow sensors or even real-time far-field optical flow measurement (Bennis et al., 2008), but experimentally more complex. However, this work demonstrates such sensing is not necessary for the purpose of learning to move effectively in the fluid, despite the critical role the fluid state plays in the relevant dynamics.

The setup was originally used to study the fluid dynamics of forward flapping flight, and possess a Reynolds number of approximately 16,000, putting it in the same regime as dragonflies and other small biological flapping fliers[3]. The plate is unalloyed titanium (chosen for its corrosion resistance), 72 cm long, 5 cm wide and .3175 cm thick. The vertical motion was produced by a scotch yoke, which converted the rotational motion of the motor into the desired linear driven motion of the plate. Due to the high gear ratio between the motor and the scotch yoke, the system was not back-drivable, and thus the plate was controlled be specifying a desired kinematic trajectory which was followed closely using tuned high-gain linear feedback. While the trajectories were not followed perfectly (e.g., there is some lag in tracking the more violent motions), the errors between specified trajectory and followed trajectory were small (on the order of 5% of the waveform's amplitude).

3 Optimal Control Formulation

We formulate the goal of control as maximizing the propulsive efficiency of forward flight by altering the plate's stroke form. We attempt to maximize this efficiency within the class of strokeforms that can be described by a given parameterization. In this section we discuss both the reward function used to measure performance, and the parameterization used to represent policies (i.e., stroke forms).

3.1 Reward Function

We desire our reward function to capture the efficiency of the forward motion produced by the stroke form. To this end, we define the (mechanical) cost-of-transport over one period T as:

$$c_{mt} = \frac{\int_T |F_z(t)\dot{z}(t)|dt}{mg\int_T \dot{x}(t)dt}. \tag{1}$$

where x is the angular position, z is the vertical position, F_z is the vertical force, m is the mass of the body and g is gravitational acceleration. The numerator of this quantity is the energy used, while the denominator is the weight times distance traveled. It was computed on our system experimentally by filtering and integrating the force measured by the load cell and dividing by the measured angular displacement, all over one period. This expression is the standard means of measuring transport cost for walking and running creatures (Collins et al., 2005), and thus seems a sensible place to start when measuring the performance of a swimming system.

This cost has the advantage of being dimensionless, and thus invariant to the units used. The non-dimensionalization is achieved by dividing by a simple scalar

[3] This Reynolds number is determined using the forward flapping motion of the wing, rather than the vertical heaving motion. The vertical heaving motion possesses a Re of approximately 3,000

(in this case mg), and thus does not change as the policy changes. While alternatives to the mass such as using the difference in mass between the plate and the displaced fluid were debated (to take into account the importance of the fluid to the motion of the plate), changes such as these would affect the magnitude of the cost, but not the optima and learning behavior, as these are invariant to a scaling. Therefore, implementing this change would not effect the found optimal policy.

Another possibility would be non-dimensionalizing by dividing by expected energy to travel through the fluid (as opposed to weight times distance), but this would depend upon the speed of travel as drag is velocity dependent, and thus would have a more complicated form. While this could obviously still be implemented, and new behavior and optima may be found, rewarding very fast flapping gaits strongly (as this would tend to do) was undesirable simply because the experimental setup struggled mechanically with the violent motions found when attempting to maximize speed. The cost function selected often produced relatively gentle motions, and as such put less strain on the setup.

Finally, note that our learning algorithm attempts to maximize the cost of transport's *inverse* (turning it from a cost into a reward), which is equivalent to minimizing the energy cost of traveling a given distance. This was done for purely practical considerations, as occasionally when very poor policies were tried the system would not move significantly despite flapping, which resulted in an infinite or near-infinite cost of transport. The inverse, however, remained well-behaved.

3.2 Policy Parameterization

The parameterization chosen took the following form: the vertical heaving motion of the wing was represented by a 13-point periodic cubic spline with fixed amplitude and frequency, giving height z as a function of time t (see Figure 3). There were five independent parameters, as the half-strokes up and down were constrained to be symmetric about the t axis (i.e., the first, seventh and last points were fixed at zero, while points 2 and 8, 3 and 9 etc. were set to such that they were equal in absolute value but opposite in sign which were determined by the control parameters).

This parameterization represented an interesting class of waveforms, had a relatively small number of parameters, and was both smooth and periodic. We will also see that many of the properties are desirable when viewed through the SNR (these advantages are discussed in greater detail in Section 6.1).

4 The Learning Algorithm

In light of the challenges faced by learning on a system with such dynamic complexity and partial observability, we made use of the weight-perturbation (WP) algorithm: a model-free policy gradient method that has been shown empirically to be

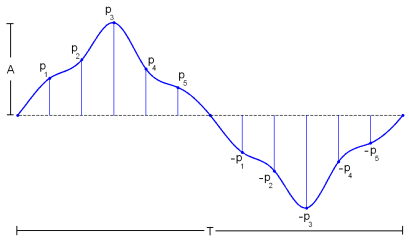


Fig. 3: A schematic of the parameterization of policies used in this work. Note the symmetry of the up and down strokes, and the fact that five independent parameters are used to encode the shape of the waveform.

well-suited to these sorts of problems due to its robustness to noise and insensitivity to the complexity of the system dynamics.

4.1 The weight perturbation update

Consider minimizing a scalar function $J(\mathbf{w})$ with respect to the parameters $\mathbf{w}$ (note that it is possible that $J(\mathbf{w})$ is a long-term cost and results from running a system with the parameters $\mathbf{w}$ until conclusion). The weight perturbation algorithm (Jabri & Flower, 1992) performs this minimization with the update:

$$\Delta\mathbf{w} = -\eta\left(J(\mathbf{w}+\mathbf{z}) - J(\mathbf{w})\right)\mathbf{z}, \tag{2}$$

where the components of the "perturbation", $\mathbf{z}$, are drawn independently from a mean-zero distribution, and η is a positive scalar controlling the magnitude of the update (the "learning rate"). Performing a first-order Taylor expansion of $J(\mathbf{w}+\mathbf{z})$ yields:

$$\Delta\mathbf{w} = -\eta\left(J(\mathbf{w}) + \sum_i \frac{\partial J}{\partial \mathbf{w}_i} z_i - J(\mathbf{w})\right)\mathbf{z} = -\eta\sum_i \frac{\partial J}{\partial \mathbf{w}_i} z_i \cdot \mathbf{z}. \tag{3}$$

In expectation, this becomes the gradient times a (diagonal) covariance matrix, and reduces to

$$E[\Delta\mathbf{w}] = -\eta\sigma^2\frac{\partial J}{\partial \mathbf{w}}, \tag{4}$$

an unbiased estimate of the gradient, scaled by the learning rate and σ^2, the variance of the perturbation. However, this unbiasedness comes with a very high variance, as the direction of an update is uniformly distributed. It is only the fact that updates

near the direction of the true gradient have a larger magnitude than do those nearly perpendicular to the gradient that allows for the true gradient to be achieved in expectation. Note also that all samples parallel to the gradient are equally useful, whether they be in the same or opposite direction, as the sign of the change in cost does not affect the resulting update.

The WP algorithm is one of the simplest examples of a policy gradient reinforcement learning algorithm, and in the special case when $\mathbf{z}$ is drawn from a Gaussian distribution, weight perturbation can be interpreted as a REINFORCE update (Williams, 1992).

4.2 The Shell Distribution

Rather than the Gaussian noise which is most commonly used for sampling, in our work we used a distribution in which $\mathbf{z}$ (the perturbation) is uniformly distributed in direction, but always has a fixed magnitude. We call this the shell distribution. This style of sampling was originally motivated by the intuitive realization that when a Gaussian distribution produced a small noise magnitude, the inherent noise in the system effectively swamped out the change in cost due to the policy perturbation, preventing any useful update from taking place. When the SNR was studied in this domain (noisy policy evaluations and possibly poor baselines), it was found to support these conclusions (as discussed in Section 5.4). For these reasons, the shell distribution was used throughout this work, and as Section 5.5 demonstrates, tangible benefits were obtained.

5 Signal-to-Noise Ratio Analysis

Our experiments with sampling distributions quickly revealed that significant performance benefits could be realized through a better understanding of the effect of sampling distributions and measurement noise on learning performance. In this section we formulate a signal-to-noise ration (SNR) analysis of the policy gradient algorithms. This analysis formalized a number of our empirical observations about WP's performance, and gave insight in several improvements that offered real benefits to the speed of convergence.

5.1 Definition of the Signal-to-Noise Ratio

The SNR is the expected power of the signal (update in the direction of the true gradient) divided by the expected power of the noise (update perpendicular to the true gradient). Taking care to ensure that the magnitude of the true gradient does not

effect the SNR, we have:

$$\text{SNR} = \frac{E\left[\Delta \mathbf{w}_{\parallel}^T \Delta \mathbf{w}_{\parallel}\right]}{E\left[\Delta \mathbf{w}_{\perp}^T \Delta \mathbf{w}_{\perp}\right]}, \tag{5}$$

$$\Delta \mathbf{w}_{\parallel} = \left(\Delta \mathbf{w}^T \frac{\mathbf{J}_{\mathbf{w}}}{\|\mathbf{J}_{\mathbf{w}}\|}\right) \frac{\mathbf{J}_{\mathbf{w}}}{\|\mathbf{J}_{\mathbf{w}}\|}, \quad \Delta \mathbf{w}_{\perp} = \Delta \mathbf{w} - \mathbf{w}_{\parallel}, \tag{6}$$

and using $\mathbf{J}_{\mathbf{w}}(\mathbf{w}_0) = \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}\Big|_{(\mathbf{w}=\mathbf{w}_0)}$ for convenience.

Intuitively, this expression measures how large a proportion of the update is "useful". If the update is purely in the direction of the gradient the SNR would be infinite, while if the update moved perpendicular to the true gradient, it would be zero. As such, all else being equal, a higher SNR should generally perform as well or better than a lower SNR, and result in less violent swings in cost and policy for the same improvement in performance. For a more in depth study of the SNR's relationship to learning performance, see (Roberts & Tedrake, 2009).

5.2 Weight perturbation with Gaussian distributions

Evaluating the SNR for the WP update in Equation 2 with a deterministic $J(\mathbf{w})$ and $\mathbf{z}$ drawn from a Gaussian distribution yields a surprisingly simple result. If one first considers the numerator:

$$\begin{aligned} E\left[\Delta \mathbf{w}_{\parallel}^T \Delta \mathbf{w}_{\parallel}\right] &= E\left[\frac{\eta^2}{\|\mathbf{J}_{\mathbf{w}}\|^4}\left(\sum_{i,j} J_{w_i} J_{w_j} z_i z_j\right) \mathbf{J}_{\mathbf{w}}{}^T \cdot \left(\sum_{k,p} J_{w_k} J_{w_p} z_k z_p\right) \mathbf{J}_{\mathbf{w}}\right] \\ &= E\left[\frac{\eta^2}{\|\mathbf{J}_{\mathbf{w}}\|^2} \sum_{i,j,k,p} J_{w_i} J_{w_j} J_{w_k} J_{w_p} z_i z_j z_k z_p\right] = Q, \end{aligned} \tag{7}$$

where we have named this term Q for convenience as it occurs several times in the expansion of the SNR. We now expand the denominator as follows:

$$E\left[\Delta \mathbf{w}_{\perp}^T \Delta \mathbf{w}_{\perp}\right] = E\left[\Delta \mathbf{w}^T \Delta \mathbf{w} - 2\Delta \mathbf{w}_{\parallel}^T (\Delta \mathbf{w}_{\parallel} + \Delta \mathbf{w}_{\perp}) + \Delta \mathbf{w}_{\parallel}^T \Delta \mathbf{w}_{\parallel}\right] = E\left[\Delta \mathbf{w}^T \Delta \mathbf{w}\right] - 2Q + Q \tag{8}$$

Substituting Equation (2) into Equation (8) and simplifying results in:

$$E\left[\Delta \mathbf{w}_{\perp}^T \Delta \mathbf{w}_{\perp}\right] = \frac{\eta^2}{\|\mathbf{J}_{\mathbf{w}}\|^2} E\left[\sum_{i,j,k} J_{w_i} J_{w_j} z_i z_j z_k^2\right] - Q. \tag{9}$$

We now assume that each component z_i is drawn from a Gaussian distribution with variance σ^2. Taking the expected value, it may be further simplified to:

$$Q = \frac{\eta^2}{\|\mathbf{J_w}\|^4}\left(3\sigma^4\sum_i {J_{w_i}}^4 + 3\sigma^4\sum_i {J_{w_i}}^2\sum_{j\neq i}{J_{w_j}}^2\right) = \frac{3\sigma^4}{\|\mathbf{J_w}\|^4}\sum_{i,j}{J_{w_i}}^2{J_{w_j}}^2 = 3\sigma^4, \tag{10}$$

$$E\left[\Delta\mathbf{w}_\perp^T\Delta\mathbf{w}_\perp\right] = \frac{\eta^2\sigma^4}{\|\mathbf{J_w}\|^2}\left(2\sum_i {J_{w_i}}^2 + \sum_{i,j}{J_{w_i}}^2\right) - Q = \sigma^4(2+N) - 3\sigma^4 = \sigma^4(N-1), \tag{11}$$

where N is the number of parameters. Canceling σ results in:

$$\text{SNR} = \frac{3}{N-1}. \tag{12}$$

Thus, for small noises and constant σ the SNR and the parameter number have a simple inverse relationship. This is a particularly concise model for performance scaling in PG algorithms.

5.3 SNR with parameter-independent additive noise

In many real world systems, the evaluation of the cost $J(\mathbf{w})$ is not deterministic, a property which can significantly affect learning performance. In this section we investigate how additive noise in the function evaluation affects the analytical expression for the SNR. We demonstrate that for very high noise WP begins to behave like a random walk, and we find in the SNR the motivation for the shell distribution; an improvement in the WP algorithm that will be examined in Section 5.4.

Consider modifying the update seen in Equation (2) to allow for a parameter-independent additive noise term v and a more general baseline $b(\mathbf{w})$, and again perform the Taylor expansion. Writing the update with these terms gives:

$$\Delta\mathbf{w} = -\eta\left(J(\mathbf{w}) + \sum_i J_{w_i}z_i - b(\mathbf{w}) + v\right)\mathbf{z} = -\eta\left(\sum_i J_{w_i}z_i + \xi(\mathbf{w})\right)\mathbf{z}. \tag{13}$$

where we have combined the terms $J(\mathbf{w})$, $b(\mathbf{w})$ and v into a single random variable $\xi(\mathbf{w})$. The new variable $\xi(\mathbf{w})$ has two important properties: its mean can be controlled through the value of $b(\mathbf{w})$, and its distribution is independent of parameters $\mathbf{w}$, thus $\xi(\mathbf{w})$ is independent of all the z_i.

We now essentially repeat the calculation seen in Section 5.2, with the small modification of including the noise term. When we again assume independent z_i, each drawn from identical Gaussian distributions with standard deviation σ, we obtain the expression:

$$\text{SNR} = \frac{\phi+3}{(N-1)(\phi+1)}, \quad \phi = \frac{(J(\mathbf{w}) - b(\mathbf{w}))^2 + \sigma_v^2}{\sigma^2\|\mathbf{J_w}\|^2} \tag{14}$$

where σ_v is the standard deviation of the noise v and we have termed the error component ϕ. This expression depends upon the fact that the noise v is mean-zero and independent of the parameters, although the assumption that v is mean-zero is not limiting. It is clear that in the limit of small ϕ the expression reduces to that seen in Equation (12), while in the limit of very large ϕ it becomes the expression for the SNR of a random walk (see (Roberts & Tedrake, 2009)). This expression makes it clear that minimizing ϕ is desirable, a result that suggests two things: (1) the optimal baseline (from the perspective of the SNR) is the value function (i.e., $b^*(\mathbf{w}) = J(\mathbf{w})$) and (2) higher values of σ are desirable as they reduce ϕ by increasing the size of its denominator. However, there is clearly a limit on the size of σ due to higher-order terms in the Taylor expansion; very large σ will result in samples which do not represent the local gradient. Thus, in the case of noisy measurements, there is some optimal sampling distance that is as large as possible without resulting in poor sampling of the local gradient. This is explored in the next section.

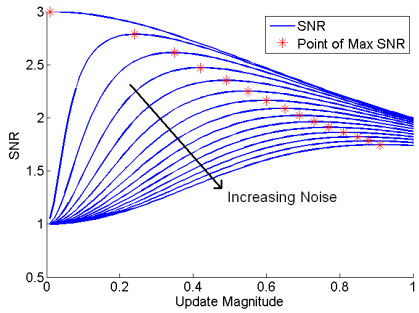


Fig. 4: SNR vs. update magnitude for a 2D quadratic cost function. Mean-zero measurement noise is included with variances from 0 to .65 (the value function's Hessian was diagonal with all entries equal to 2). As the noise is increased, the sampling magnitude producing the maximum SNR is larger and the SNR achieved is lower. Note that the highest SNR achieved is for the smallest sampling magnitude with no noise where it approaches the theoretical value (for 2D) of 3. Also note that for small sampling magnitudes and large noises the SNR approaches the random walk value of $1/N - 1$ (see (Roberts & Tedrake, 2009)).

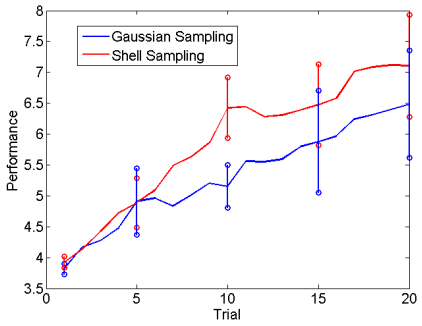
5.4 Non-Gaussian Distributions

The analysis in Section 5.3 suggests that for a function with noisy measurements there is an optimal sampling distance which depends upon the local noise and gradient as well as the strength of higher-order terms in that region. For a two-

dimensional cost function that takes the form of a quadratic bowl in parameter space, Figure 4 shows the SNR's dependence upon the radius of the shell distribution (i.e., the magnitude of the sampling). For various levels of additive mean-zero noise the SNR was computed for a distribution uniform in angle and fixed in its distance from the mean (this distance is the "sampling magnitude"). The fact that there is a unique maximum for each case suggests the possibility of sampling *only* at that maximal magnitude, rather than over all magnitudes as is done with a Gaussian, and thus improving SNR and performance. While determining the exact magnitude of maximum SNR may be impractical, by choosing a distribution with uniformly distributed direction and a constant magnitude close to this optimal value, performance can be improved.

5.5 Experimental Evaluation of Shell Distributions

To provide compelling evidence that the shell distribution could improve convergence in problems of interest, the shell distribution was implemented directly on the heaving plate, and the resulting learning curves were compared to those obtained using Gaussian sampling. For the purposes of this comparison, policy evaluations were run for long enough to reach steady state (to eliminate any issues relating to the coupling between consecutive evaluations). As can be seen in Figure 5, the shell distribution provided a real advantage in convergence rate on this system of interest, when dealing with the full dynamical complexity of a laboratory experimental system.



(a)

Fig. 5: Five averaged runs on the heaving plate using Gaussian or Shell distributions for sampling. The error bars represent one standard deviation in the performance of different runs at that trial.

5.6 Implications for Learning at Intermediate Reynolds Numbers

The SNR demonstrates many of the properties of WP that make it so well suited to learning on partially observable and dynamically complicated system, such as fluid systems in the intermediate Reynolds number regime. The SNR shows that the system's dynamical complexity does not (locally) effect the difficulty of learning, as the dynamics appear nowhere in the expression. Instead, learning performance is locally effected by the number of parameters in the policy (N), the level of stochasticity in policy evaluations (σ_v), the quality of the baseline and the steepness of local gradients.

The SNR does not take into account the effects of higher-order behavior such as the roughness of the value function in policy space, which is in general a function of the system, the choice of parameterization and the choice of the cost function. These properties can be extremely important to the performance of WP, affecting both number and depth of local minima and the rate of learning, but are not analytically tractable in general.

6 Learning Results

6.1 Policy Representation Viewed Through SNR

Due to the importance of the policy parameterization to the performance of the learning, it is important to pick the policy class carefully. Now armed with knowledge of the SNR, some basic guidelines for choosing the parameterization, previously justified heuristically, become more precise. Finding a rich representation with a small number of parameters can greatly improve convergence rates. Furthermore, certain parameterizations can be found with fewer local minima and smoother gradients than others, although determining these properties a priori is often impractical. A parameterization in which all parameters are reasonably well-coupled to the cost function is beneficial, as this will result in less anisotropy in the magnitude of the gradients, and thus larger sampling magnitudes and greater robustness to noise can be achieved. Prior to the periodic cubic spline, several parameterizations were tried, with all taking the form of encoding the z height of the wing as a function of time t over one period T, with the ends of the waveform constrained to be periodic (i.e., the beginning and end have identical values and first derivatives).

Parameterizing the policy in the seemingly natural fashion of a finite Fourier series was ruled out due to the difficulty in representing many intuitively useful waveforms (e.g., square waves) with a reasonable number of parameters. A parameterization using the sum of base waveforms (i.e., a smoothed square wave, a sine wave and a smoothed triangle wave) was used and shown to learn well, but was deemed a too restrictive class which predetermined many features of the waveform. Learning the base period T and the amplitude A of the waveform was also tried, and

shown to perform well without significant difficulty. However, it was discovered that periods as long as possible and amplitudes as small as possible were selected, and thus these extra parameters were determined to not be of interest to the learning (this result was useful in determining how the system's dynamics related to the reward function, and is discussed in detail in Section 7).

6.2 Reward Function Viewed Through SNR

The SNR also gives some greater insight into the formulation of the reward function. The form of the reward was discussed in Section 3.1, and justified by its connection to previous work in locomotive efficiency. We may now, however, understand more of what makes it advantageous from the perspective of learning performance and SNR. Its integral form performs smoothing on the collected data, which effectively reduces the noise level for a given trial, and intuitively it should differentiate meaningfully between different policies (e.g., a reward function that has no gradient with respect to the parameters in some region of policy space will find it very difficult to learn in that region).

6.3 Implementation of Online Learning

The SNR analysis presented here is concerned with episodic trials (i.e., a series of distinct runs), but this does not preclude online operation, in which trials follow one another immediately and the system runs continuously. While at first we ran longer policy evaluations (on the order of 40 or more flaps, averaged together) to reduce noise, and gave the system plenty of time (20 or more flaps) between trials to reach steady state and avoid inter-trial coupling, as we became more proficient at dealing with the high noise levels of the system we began to attempt to learn more aggressively. Through techniques such as sampling from the shell distribution, we were able to reduce trial length to a single flap (requiring just 1 second), and by reducing noise levels (ultimately choosing a shell distribution with a radius of approximately 4% of the parameter value) were able to eliminate the inter-trial settling time. Inter-trial correlation was explored, and while present, we found it did not significantly hamper learning performance, and that including eligibility traces did not greatly improve the rate of convergence.

6.4 Performance of Learning

In its final form, the SNR-optimized WP update was able to learn extremely efficiently. Using the policy parameterization described above, along with shell sam-

pling and one-flap trials, the optimal policy was found within 7 minutes (around 400 flaps) even when starting far away in state space (see Figure 6). This quick convergence in the face of inter-trial coupling and high variance in policy evaluations (resulting from running them for such a short period of time) demonstrates the WP algorithm's robustness to these complex but very common difficulties.

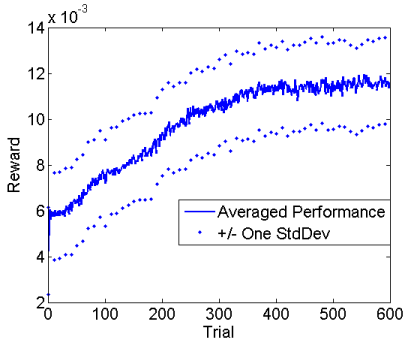


Fig. 6: The average of five learning curves using online learning (an update every second, after each full flapping cycle), with markers for +/- one standard deviation. The high variance is the result of large inter-trial variance in the cost, rather than large differences between different learning curves.

7 Interpretation of Optimal Solution

Once the learning had been successfully implemented, and repeatable convergence to the same optimum was achieved, it is interesting to investigate what the form of the solution suggests about the physical system. Figure 7 shows an example of an initial, intermediate and final waveform from a learning trial, starting at a smoothed out square wave and ending at the triangle wave which was found to be optimal.

The result is actually quite satisfying from a fluid dynamics point of view, as it is consistent with our theoretical understanding of the system, and indeed was the predicted solution given our reward function by experts in the field of flapping flight. If one considers the reward function used in this work (see 3.1), the basis of this behavior's optimality becomes clear.

Consider the cost of transport c_{mt}. As drag force is approximately quadratic with speed in this regime, the numerator behaves approximately as:

$$\int_T |F_z(t)\dot{z}(t)|dt \sim \frac{1}{2}\rho C_d \langle V^2 \rangle T, \tag{15}$$

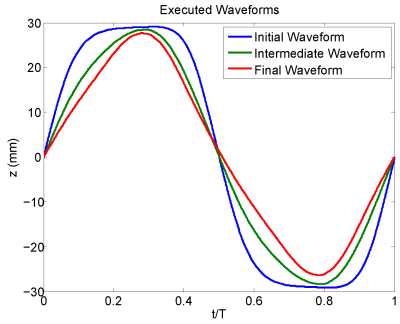


Fig. 7: A series of waveforms (initial, intermediate and final) seen during learning on the rigid plate.

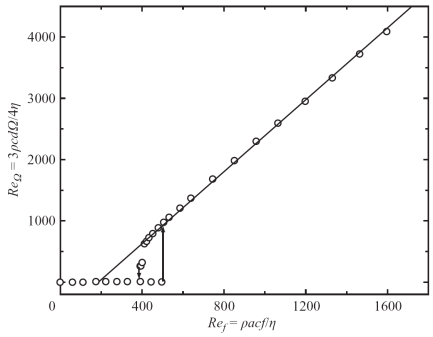


Fig. 8: Linear growth of forward speed with flapping frequency. The axes of this curve have been non-dimensionalized as shown, and the data was taken for a sine wave. Figure from (Vandenberghe et al., 2004).

where C_d is the coefficient of drag and $\langle V^2 \rangle$ is the mean squared heaving speed. However, forward rotational speed was found to grow linearly with flapping frequency (see (Vandenberghe et al., 2004) and Figure 8), thus the denominator can be written approximately as:

$$mg \int_T \dot{x}(t)dt \sim C_f \langle V \rangle T, \tag{16}$$

where C_f is a constant relating vertical speed to forward speed, and $\langle V \rangle$ is the mean vertical speed. Therefore, higher speeds result in quadratic growth of the numerator of the cost and linear growth of the cost's denominator. This can be seen as the reward r (the inverse of c_{mt}) having the approximate form:

$$r \sim C \frac{\langle V \rangle}{\langle V^2 \rangle}, \tag{17}$$

with C a constant. This results in lower speeds being more efficient, causing lower frequencies and amplitudes to be preferred. If period and amplitude are fixed, however, the average speed is fixed (assuming no new extrema of the stroke form are produced during learning, a valid assumption in practice). A triangle wave, then, is the means of achieving this average speed with the minimum average *squared* speed.

The utility of this control development method now becomes more clear. For systems that, despite having complicated dynamics, can be reasonably described with lumped-parameter or quasi-steady models, learning the controller directly serves dual purposes: it suggests lines of reasoning that inform the creation of relatively simple models, and it gives confidence that the modeling captures the aspects of the system relevant to the optimization. Furthermore, on systems for which tractable models are unavailable, the learning methodology can be applied just as easily while model-centric techniques will begin to fail.

8 Conclusion

This work has presented a case study in how to produce efficient, online learning on a complicated fluid system. The techniques used here were shown to be effective, with convergence being achieved on the heaving plate in approximately seven minutes. The algorithmic improvements presented have additional applications to many other systems, and by succeeding on a problem possessing great dynamic complexity, a reasonably large dimensionality, partial observability and noisy evaluations, they have been shown to be robust and useful. We believe the complexity of flow control systems is an excellent match for the capabilities of learning control, and expect to see many more applications for this domain in the future.

Bibliography

Alben, S., & Shelley, M. (2005). Coherent locomotion as an attracting state for a free flapping body. *Proceedings of the National Academy of Science*, *102*, 11163–11166.

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation*, *10*, 251–276.

Baxter, J., & Bartlett, P. (2001). Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, *15*, 319–350.

Bennis, A., Leeser, M., Tadmor, G., & Tedrake, R. (2008). Implementation of a highly parameterized digital PIV system on reconfigurable hardware. *Proceedings of the Twelfth Annual Workshop on High Performance Embedded Computing (HPEC)*. Lexington, MA.

Collins, S. H., Ruina, A., Tedrake, R., & Wisse, M. (2005). Efficient bipedal robots based on passive-dynamic walkers. *Science*, *307*, 1082–1085.

Greensmith, E., Bartlett, P. L., & Baxter, J. (2004). Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, *5*, 1471–1530.

Howard, M., Klanke, S., Gienger, M., Goerick, C., & Vijayakumar, S. (2009). Methods for learning control policies from variable-constraint demonstrations. In *From motor to interaction learning in robots*. Springer.

Jabri, M., & Flower, B. (1992). Weight perturbation: An optimal architecture and learning technique for analog VLSI feedforward and recurrent multilayer networks. *IEEE Trans. Neural Netw.*, *3*, 154–157.

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*.

Kober, J., Mohler, B., & Peters, J. (2009). Imitation and reinforcement learning for motor primitives with perceptual coupling. In *From motor to interaction learning in robots*. Springer.

Meuleau, N., Peshkin, L., Kaelbling, L. P., & Kim, K.-E. (2000). Off-policy policy search. *NIPS*.

Peters, J., Vijayakumar, S., & Schaal, S. (2003). *Policy gradient methods for robot control* (Technical Report CS-03-787). University of Southern California.

Roberts, J. W., & Tedrake, R. (2009). Signal-to-noise ratio analysis of policy gradient algorithms. *Advances of Neural Information Processing Systems (NIPS) 21* (p. 8).

Shelley, M. (2007). Personal Communication.

Tedrake, R., Zhang, T. W., & Seung, H. S. (2004). Stochastic policy gradient reinforcement learning on a simple 3D biped. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)* (pp. 2849–2854). Sendai, Japan.

Vandenberghe, N., Childress, S., & Zhang, J. (2006). On unidirectional flight of a free flapping wing. *Physics of Fluids*, *18*.

Vandenberghe, N., Zhang, J., & Childress, S. (2004). Symmetry breaking leads to forward flapping flight. *Journal of Fluid Mechanics*, *506*, 147–155.

Williams, J. L., III, J. W. F., & Willsky, A. S. (2006). Importance sampling actor-critic algorithms. *Proceedings of the 2006 American Control Conference*.

Williams, R. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, *8*, 229–256.