

# Dual Control for Approximate Bayesian Reinforcement Learning

**Edgar D. Klenske**

*Max-Planck-Institute for Intelligent Systems  
Spemannstraße 38  
72076 Tübingen, Germany*

EDGAR.KLENSKE@TUEBINGEN.MPG.DE

**Philipp Hennig**

*Max-Planck-Institute for Intelligent Systems  
Spemannstraße 38  
72076 Tübingen, Germany*

PHILIPP.HENNIG@TUEBINGEN.MPG.DE

**Editor:** Manfred Opper

## Abstract

Control of non-episodic, finite-horizon dynamical systems with uncertain dynamics poses a tough and elementary case of the exploration-exploitation trade-off. Bayesian reinforcement learning, reasoning about the effect of actions and future observations, offers a principled solution, but is intractable. We review, then extend an old approximate approach from control theory—where the problem is known as *dual control*—in the context of modern regression methods, specifically generalized linear regression. Experiments on simulated systems show that this framework offers a useful approximation to the intractable aspects of Bayesian RL, producing structured exploration strategies that differ from standard RL approaches. We provide simple examples for the use of this framework in (approximate) Gaussian process regression and feedforward neural networks for the control of exploration.

**Keywords:** reinforcement learning, control, Gaussian processes, filtering, Bayesian inference

## 1. Introduction

The exploration-exploitation trade-off is a central problem of learning in interactive settings, where the learner’s actions influence future observations. In episodic settings, where the control problem is re-instantiated repeatedly with unchanged dynamics, comparably simple notions of exploration can succeed. E.g., assigning an *exploration bonus* to uncertain options (Macready and Wolpert, 1998; Audibert et al., 2009) or acting optimally under one sample from the current probabilistic model of the environment (*Thompson sampling*, see Thompson, 1933; Chapelle and Li, 2011), can perform well (Dearden et al., 1999; Kolter and Ng, 2009; Srinivas et al., 2010). Such approaches, however, do not model the effect of actions on future beliefs, which limits the potential for the balancing of exploration and exploitation. This issue is most drastic in the non-episodic case, the control of a single, ongoing trial. Here, the controller cannot hope to be returned to known states, and exploration must be carefully controlled to avoid disaster.

A principled solution to this problem is offered by *Bayesian reinforcement learning* (Duff, 2002; Poupart et al., 2006; Hennig, 2011): A probabilistic belief over the dynamics and cost of the environment can be used not just to simulate and plan trajectories, but also to reason about changes to the belief from future observations, and their influence on future decisions. An elegant formulation is to combine the physical state with the parameters of the probabilistic model into an augmented dynamical description, then aim to control this system. Due to the inference, the augmented system invariably has strongly nonlinear dynamics, causing prohibitive computational cost—even for finite state spaces and discrete time (Poupart et al., 2006), all the more for continuous space and time (Hennig, 2011).

The idea of augmenting the physical state with model parameters was noted early, and termed *dual control*, by Feldbaum (1960–1961). It seems both conceptual and—by the standards of the time—computational complexity hindered its application. An exception is a strand of several works by Meier, Bar-Shalom, and Tse (Tse et al., 1973; Tse and Bar-Shalom, 1973; Bar-Shalom and Tse, 1976; Bar-Shalom, 1981). These authors developed techniques for limiting the computational cost of dual control that, from a modern perspective, can be seen as a form of approximate inference for Bayesian reinforcement learning. While the Bayesian reinforcement learning community is certainly aware of their work (Duff, 2002; Hennig, 2011), it has not found widespread attention. The first purpose of this paper is to cast their dual control algorithm as an approximate inference technique for Bayesian RL in parametric Gaussian (general least-squares) regression. We then extend the framework with ideas from contemporary machine learning. Specifically, we explain how it can in principle be formulated non-parametrically in a Gaussian process context, and then investigate simple, practical finite-dimensional approximations to this result. We also give a simple, small-scale example for the use of this algorithm for dual control if the environment model is constructed with a feedforward neural network rather than a Gaussian process.

## 2. Model and Notation

Throughout, we consider discrete-time, finite-horizon dynamic systems (POMDPs) of form

$$x_{k+1} = f_k(x_k, u_k) + \xi_k \quad (\text{state dynamics}) \qquad y_k = Cx_k + \gamma_k \quad (\text{observation model}).$$

At time  $k \in \{0, \dots, T\}$ ,  $x_k \in \mathbb{R}^n$  is the state,  $\xi_k \sim \mathcal{N}(0, Q)$  is a Gaussian disturbance. The control input (continuous action) is denoted  $u_k$ ; for simplicity we will assume scalar  $u_k \in \mathbb{R}$  throughout. Measurements  $y_k \in \mathbb{R}^d$  are observations of  $x_k$ , corrupted by Gaussian noise  $\gamma_k \sim \mathcal{N}(0, R)$ . The generative model thus reads  $p(x_{k+1} | x_k, u_k) = \mathcal{N}(x_{k+1}; f_k(x_k, u_k), Q)$  and  $p(y_k | x_k) = \mathcal{N}(y_k; Cx_k, R)$ , with a linear map  $C \in \mathbb{R}^{d \times n}$ . Trajectories are vectors  $\mathbf{x} = [x_0, \dots, x_T]$ , and analogously for  $\mathbf{u}, \mathbf{y}$ . We will occasionally use the subset notation  $\mathbf{y}_{i:j} = [y_i, \dots, y_j]$ . We further assume that dynamics  $f_k$  are not known, but can be described up to Gaussian uncertainty by a general linear model with nonlinear features  $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$  and uncertain matrices  $A_k, B_k$ .

$$x_{k+1} = A_k \phi(x_k) + B_k u_k + \xi_k, \qquad A_k \in \mathbb{R}^{n \times m}; B_k \in \mathbb{R}^{n \times 1}. \qquad (1)$$

To simplify notation, we reshape the elements of  $A_k$  and  $B_k$  into a parameter vector  $\theta_k = [\text{vec}(A_k); \text{vec}(B_k)] \in \mathbb{R}^{(m+1)n}$ , and define the reshaping transformations  $A(\theta_k) : \theta_k \mapsto A_k$

and  $B(\theta_k) : \theta_k \mapsto B_k$ . At initialization,  $k = 0$ , the belief over states and parameters is assumed to be Gaussian

$$p\left(\begin{bmatrix} x_0 \\ \theta_0 \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} x_0 \\ \theta_0 \end{bmatrix}; \begin{bmatrix} \hat{x}_0 \\ \hat{\theta}_0 \end{bmatrix}, \begin{bmatrix} \Sigma_0^{xx} & \Sigma_0^{x\theta} \\ \Sigma_0^{\theta x} & \Sigma_0^{\theta\theta} \end{bmatrix}\right). \quad (2)$$

The control response  $B_k u_k$  is linear, a common assumption for physical systems. Nonlinear mappings can be included in a generic form  $\phi(x_k, u_k)$ , but complicate the following derivations and raise issues of identifiability. For simplicity, we also assume that the dynamics do not change through time:  $p(\theta_{k+1} | \theta_k) = \delta(\theta_{k+1} - \theta_k)$ . This could be relaxed to an autoregressive model  $p(\theta_{k+1} | \theta_k) = \mathcal{N}(\theta_{k+1}; D\theta_k, \Xi)$ , which would give additive terms in the derivations below. Throughout, we assume a finite horizon with terminal time  $T$  and a quadratic cost function in state and control

$$\mathcal{L}(\mathbf{x}, \mathbf{u}) = \left[ \sum_{k=0}^T (x_k - r_k)^\top W_k (x_k - r_k) + \sum_{k=0}^{T-1} u_k^\top U_k u_k \right],$$

where  $\mathbf{r} = [r_0, \dots, r_T]$  is a target trajectory.  $W_k$  and  $U_k$  define state and control cost, they can be time-varying. The goal, in line with the standard in both optimal control and reinforcement learning, is to find the control sequence  $\mathbf{u}$  that, at each  $k$ , minimizes the *expected cost* to the horizon

$$J_k(\mathbf{u}_{k:T-1}, p(x_k)) = \mathbb{E}_{x_k} \left[ (x_k - r_k)^\top W_k (x_k - r_k) + u_k^\top U_k u_k + J_{k+1}(\mathbf{u}_{k+1:T-1}, p(x_{k+1})) | p(x_k) \right], \quad (3)$$

where past measurements  $\mathbf{y}_{1:k}$ , controls  $\mathbf{u}_{1:k-1}$  and prior information  $p(x_0)$  are incorporated into the belief  $p(x_k)$ , relative to which the expectation is calculated. Effectively,  $p(x_k)$  serves as a bounded rationality approximation to the true information state. Since the equation above is recursive, the final element of the cost has to be defined differently, as

$$J_T(p(x_T)) = \mathbb{E}_{x_T} \left[ (x_T - r_T)^\top W_T (x_T - r_T) | p(x_T) \right]$$

(that is, without control input and future cost). The optimal control sequence minimizing this cost will be denoted  $\mathbf{u}^*$ , with associated cost

$$J_k^*(p(x_k)) = \min_{u_k} \mathbb{E}_{x_k} \left[ (x_k - r_k)^\top W_k (x_k - r_k) + u_k^\top U_k u_k + J_{k+1}^*(p(x_{k+1})) | p(x_k) \right]. \quad (4)$$

This recursive formulation, if written out, amounts to alternating minimization and expectation steps. As  $u_k$  influences  $x_{k+1}$  and  $y_{k+1}$ , it enters the latter expectation nonlinearly. Classic optimal control is the linear base case ( $\phi(x) = x$ ) with known  $\theta$ , where  $\mathbf{u}^*$  can be found by dynamic programming (Bellman, 1961; Bertsekas, 2005).

### 3. Bayesian RL and Dual Control

Feldbaum (1960–1961) coined the term *dual control* to describe the idea now also known as Bayesian reinforcement learning in the machine learning community: While adaptive control only considers past observations, dual control also takes future observations into account. This is necessary because all other ways to deal with uncertain parameters have substantial

drawbacks. Robust controllers, for example, sacrifice performance due to their conservative design; adaptive controllers based on *certainty equivalence* (where the uncertainty of the parameters is not taken into account but only their mean estimates) do not show exploration, so that all learning is purely passive. For most systems it is obvious that more excitation leads to better estimation, but also to worse control performance. Attempts at finding a compromise between exploration and exploitation are generally subsumed under the term “dual control” in the control literature. It can only be achieved by taking the future effect of current actions into account.

It has been shown that optimal dual control is practically unsolvable for most cases (Aoki, 1967), with a few examples where solutions were found for simple systems (e.g., Sternby, 1976). Instead, a large number of *approximate* formulations of the dual control problem were formulated in the decades since then. This includes the introduction of perturbation signals (e.g., Jacobs and Patchell, 1972), constrained optimization to limit the minimal control signal or the maximum variance, serial expansion of the loss function (e.g., Tse et al., 1973) or modifications of the loss function (e.g., Filatov and Unbehauen, 2004). A comprehensive overview of dual control methods is given by Wittenmark (1995). A historical side-effect of these numerous treatments is that the meaning of the term “dual control” has evolved over time, and is now applied both to the fundamental concept of optimal exploration, and to methods that only approximate this notion to varying degree. Our treatment below studies one such class of practical methods that aim to approximate the true dual control solution.

The central observation in Bayesian RL / dual control is that both the states  $x$  and the parameters  $\theta$  are subject to uncertainty. While part of this uncertainty is caused by randomness, part by lack of knowledge, both can be captured in the same way by probability distributions. States and parameters can thus be subsumed in an *augmented state* (Feldbaum, 1960–1961; Duff, 2002; Poupart et al., 2006)  $z_k^\top = (x_k^\top \ \theta_k^\top) \in \mathbb{R}^{(m+2)n}$ . In this notation, the optimal exploration-exploitation trade-off—relative to the probabilistic priors defined above—can be written compactly as optimal control of the augmented system with a new observation model  $p(y_k | z_k) = \mathcal{N}(y_k; \tilde{C}z_k, R)$  using  $\tilde{C} = [C \ \mathbf{0}]$  and a cost analogous to Eq. (3).

Unfortunately, the dynamics of this new system are nonlinear, even if the original physical system is linear. This is because inference is always nonlinear and future states influence future parameter beliefs, and vice versa. A first problem, not unique to dual control, is thus that inference is not analytically tractable, even under the Gaussian assumptions above (Aoki, 1967). The standard remedy is to use approximations, most popularly the linearization of the extended Kalman filter (e.g., Särkkä, 2013). This gives a sequence of approximate Gaussian likelihood terms. But even so, incorporating these Gaussian likelihood terms into future dynamics is still intractable, because it involves expectations over rational polynomial functions, whose degree increases with the length of the prediction horizon. The following section provides an intuition for this complexity, but also the descriptive power of the augmented state space.

As an aside, we note that several authors (Kappen, 2011; Hennig, 2011) have previously pointed out another possible construction of an augmented state: incorporating not the actual *value* of the parameters  $\theta_k$  in the state, but the parameters  $\mu_k, \Sigma_k$  of a Gaussian belief  $p(\theta_k | \mu_k, \Sigma_k) = \mathcal{N}(\theta_k; \mu_k, \Sigma_k)$  over them. The advantage of this is that, if the state  $x_k$  is observed without noise, these belief parameters follow stochastic differential equations—more

precisely,  $\Sigma_k$  follows an ordinary (deterministic) differential equation, while  $\mu_k$  follows a stochastic differential equation—and it can then be attempted to solve the control problem for these differential equations more directly.

While it can be a numerical advantage, this formulation of the augmented state also has some drawbacks, which is why we have here decided not to adopt it: First, the simplicity of the directly formalizable SDE vanishes in the POMDP setting, i.e. if the state is not observed without noise. If the state observations are corrupted, the exact belief state is not a Gaussian process, so that the parameters  $\mu_k$  and  $\Sigma_k$  have no natural meaning. Approximate methods can be used to retain a Gaussian belief (and we will do so below), but the dynamics of  $\mu_k$ ,  $\Sigma_k$  are then intertwined with the chosen approximation (i.e. changing the approximation changes their dynamics), which causes additional complication. More generally speaking, it is not entirely natural to give differing treatment to the state  $x_k$  and parameters  $\theta_k$ : Both state and parameters should thus be treated within the same framework; this also allows extending the framework to the case where also the parameters do follow an SDE.

### 3.1 A Toy Problem

To provide an intuition for sheer complexity of optimal dual control, consider the perhaps simplest possible example: the linear, scalar system

$$x_{k+1} = ax_k + bu_k + \xi_k, \quad (5)$$

with target  $r_k = 0$  and noise-free observations ( $R = 0$ ). If  $a$  and  $b$  are known, the optimal  $u_k$  to drive the current state  $x_k$  to zero in one step can be trivially verified to be

$$u_{k,\text{oracle}}^* = -\frac{abx_k}{U + b^2}.$$

Let now parameter  $b$  be uncertain, with current belief  $p(b) = \mathcal{N}(b; \mu_k, \sigma_k^2)$  at time  $k$ . The naïve option of simply replacing the parameter with the current mean estimate is known as *certainty equivalence (CE)* control in the dual control literature (e.g., Bar-Shalom and Tse, 1974). The resulting control law is

$$u_{k,\text{CE}}^* = -\frac{a\mu_k x_k}{U + \mu_k^2}.$$

It is used in many adaptive control settings in practice, but has substantial deficiencies: If the uncertainty is large, the mean is not a good estimate, and the CE controller might apply completely useless control signals. This often results in large overshoots at the beginning or after parameter changes.

A slightly more elaborate solution is to compute the expected cost  $\mathbb{E}_b[x_{k+1}^2 + Uu_k^2 | \mu_k, \sigma_k^2]$  and then optimize for  $u_k$ . This gives *optimal feedback (OF)* or “cautious” control (Dreyfus, 1964)<sup>1</sup>:

$$u_{k,\text{OF}}^* = -\frac{a\mu_k x_k}{U + \sigma_k^2 + \mu_k^2}. \quad (6)$$

---

1. Dreyfus used the term “open loop optimal feedback” for his approach, a term that is misleading to modern readers, because it is in fact a closed-loop algorithm.

This control law reduces control actions in cases of high parameter uncertainty. This mitigates the main drawback of the CE controller, but leads to another problem: Since the OF controller decreases control with rising uncertainty, it can entirely prevent learning. Consider the posterior on  $b$  after observing  $x_{k+1}$ , which is a closed-form Gaussian (because  $u_k$  is chosen by the controller and has no uncertainty):

$$p(b | \mu_{k+1}, \sigma_{k+1}^2) = \mathcal{N}(b; \mu_{k+1}, \sigma_{k+1}^2) = \mathcal{N}\left(b; \frac{\sigma_k^2 u_k (bu_k + \xi_k) + \mu_k Q}{u_k^2 \sigma_k^2 + Q}, \frac{\sigma_k^2 Q}{u_k^2 \sigma_k^2 + Q}\right) \quad (7)$$

( $b$  shows up in the fully observed  $x_{k+1} = ax_k + bu_k + \xi_k$ ). The dual effect here is that the updated  $\sigma_{k+1}^2$  depends on  $u_k$ . For large values of  $\sigma_k^2$ , according to (6),  $u_{k,\text{OF}}^* \rightarrow 0$ , and the new uncertainty  $\sigma_{k+1}^2 \rightarrow \sigma_k^2$ . The system thus will never learn or act, even for large  $x_k$ . This is known as the “turn-off phenomenon” (Aoki, 1967; Bar-Shalom, 1981).

However, the derivation for OF control above amounts to minimizing Eq. (3) for the myopic controller, where the horizon is only a single step long ( $T = 1$ ). Therefore, OF control is indeed optimal for this case. By the optimality principle (e.g., Bertsekas, 2005), this means that Eq. (6) is the optimal solution for the last step of every controller. But since it does not show any form of exploration or “probing” (Bar-Shalom and Tse, 1976), a myopic controller is not enough to show the dual properties.

In order to expose the dual features, the horizon has to be at least of length  $T = 2$ . Since the optimal controller follows Bellman’s equation, the solution proceeds backwards. The solution for the second control action  $u_1$  is identical to the solution of the myopic controller (6); but after applying the first control action  $u_0$ , the belief over the unknown parameter  $b$  needs an update according to Eq. (7), resulting in

$$u_1^* = - \left[ U + \frac{\sigma_0^2 Q}{u_0^2 \sigma_0^2 + Q} + \left( \frac{\sigma_0^2 u_0 (bu_0 + \xi_0) + \mu_0 Q}{u_0^2 \sigma_0^2 + Q} \right)^2 \right]^{-1} \left[ a \frac{\sigma_0^2 u_0 (bu_0 + \xi_0) + \mu_0 Q}{u_0^2 \sigma_0^2 + Q} x_1 \right]. \quad (8)$$

Inserting into Eq. (4) gives

$$\begin{aligned} J_0^*(x_0) &= \min_{u_0} \mathbb{E}_{x_0} \left[ Wx_0^2 + Uu_0^2 + \min_{u_1} \mathbb{E}_{x_1} \left[ Wx_1^2 + Uu_1^2 + \mathbb{E}_{x_2} [Wx_2^2] \right] \right] \\ &= \min_{u_0} \left[ Wx_0^2 + Uu_0^2 + \mathbb{E}_{\xi_0, b} [Wx_1^2 + U(u_1^*)^2 + \mathbb{E}_{\xi_1, b} [W(x_1 + bu_1^* + \xi_1)^2 | \mu_1, \sigma_1] | \mu_0, \sigma_0] \right]. \end{aligned} \quad (9)$$

Since  $u_1^*$  from Eq. (8) is already a rational function of fourth order in  $b_0$ , and shows up quadratically in Eq. (9), the relevant expectations cannot be computed in closed form (Aoki, 1967). For this simple case though, it is possible to compute the optimal dual control by performing the expectation through sampling  $b, \xi_0, \xi_1$  from the prior. Fig. 1 shows such samples of  $\mathcal{L}(u_0)$  (in gray; one single sample highlighted in orange), and the empirical expectation  $J(u_0)$  in dashed green. Each sample is a rational function of even leading order. In contrast to the CE cost, the dual cost is much narrower, leading to more cautious behavior of the dual controller. The average dual cost has its minima not at zero, but to either side of it, reflecting the optimal amount of exploration in this particular belief state.

While it is not out of the question that the Monte Carlo solution can remain feasible for larger horizons, we are not aware of successful solutions for continuous state spaces

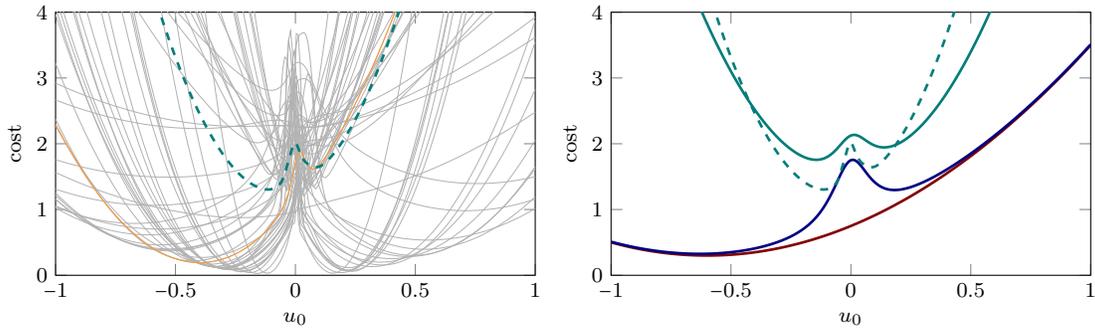


Figure 1: **Left:** Computing the  $T = 2$  dual cost for the simple system of Eq. (5). Costs  $\mathcal{L}(u_0)$  under optimal control on  $u_1$  for sampled parameter  $b$  (thin gray; one sample highlighted, orange). Expected dual cost  $J(u_0)$  under  $u_1^*$  (dashed green). The optimal  $u_0^*$  lies at the minimum of the dashed green line. **Right:** Comparison of sampling (dashed green; thin gray: samples) to three approximations: CE (red) and CE with Bayesian exploration bonus (blue). The solid green line is the approximate dual control constructed in Section 4. See also Sec. 6.1 for details.

(however, see Poupart et al., 2006, for a sampling solution to Bayesian reinforcement learning in discrete spaces, including notes on the considerable computational complexity of this approach). The next section describes a tractable *analytic* approximation that does not involve samples.

#### 4. Approximate Dual Control for Linear Systems

In 1973, Tse et al. (1973) constructed theory and an algorithm (Tse and Bar-Shalom, 1973) for approximate dual (AD) control, based on the series expansion of the cost-to-go. This is related to differential dynamic programming for the control of nonlinear dynamic systems (Mayne, 1966). It separates into three conceptual steps (described in Sec. 4.1–4.3), which together yield what, from a contemporary perspective, amounts to a structured Gaussian approximation to Bayesian RL:

- ① Find an optimal trajectory for the deterministic part of the system under the mean model: the *nominal* trajectory under certainty equivalent control. For linear systems this is easy (see below), for nonlinear ones it poses a nontrivial, but feasible nonlinear model predictive control problem (Allgöwer et al., 1999; Diehl et al., 2009). It yields a nominal trajectory, relative to which the following step constructs a tractable quadratic expansion.
- ② Around the nominal trajectory, construct a local *quadratic expansion* that approximates the effects of future observations. Because the expansion is quadratic, an optimal control law relative to the deterministic system—the *perturbation control*—can be constructed by dynamic programming. Plugging this perturbation control into the residual dynamics of the approximate quadratic system gives an approximation for the cost-to-go. This step adds the cost of uncertainty to the deterministic control cost.

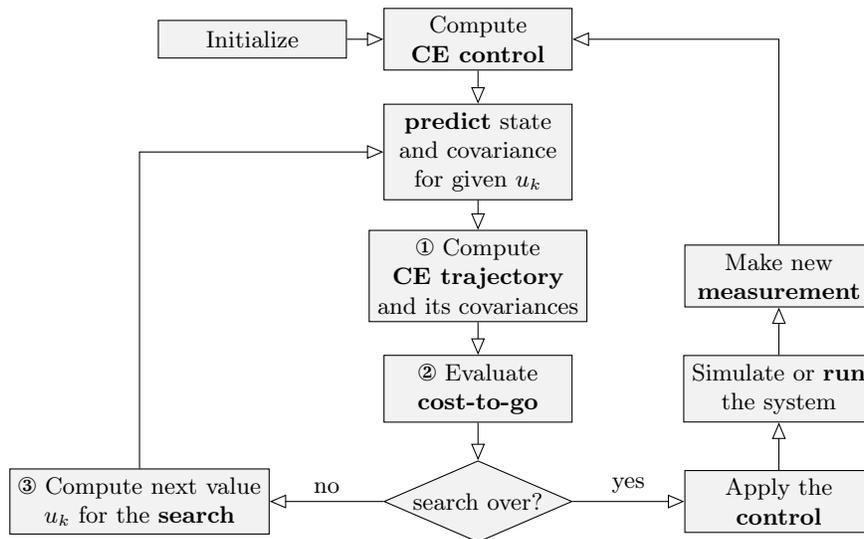


Figure 2: Flow-chart of the approximate dual control algorithm to show the overall structure. Adapted from Tse and Bar-Shalom (1973). The left cycle is the inner loop, performing the nonlinear optimization.

- ③ In the current time step  $k$ , perform the prediction for an arbitrary control input  $u_k$  (as opposed to the analytically computed control input for later steps). Optimize  $u_k$  numerically by repeated computation of steps ① and ② at varying  $u_k$  to minimize the approximate cost.

These three steps will be explained in detail in the subsequent sections. The interplay between the different parts of the algorithm is shown in Figure 2.

The abstract introductory work Tse et al. (1973) is relatively general, but the explicit formulation in Tse and Bar-Shalom (1973) only applies to linear systems. Since both works are difficult to parse for contemporary readers, the following sections thus first provide a short review, before we extend to more modern concepts. In this section, we follow the more transparent case of a linear system from Tse and Bar-Shalom (1973), i.e.  $\phi(x) = x$  in Eq. (1). For the augmented state  $z$ , this still gives a nonlinear system, because  $\theta$  and  $x$  interact multiplicatively

$$z_{k+1} = \begin{pmatrix} x_{k+1} \\ \theta_{k+1} \end{pmatrix} = \begin{pmatrix} A(\theta_k) & 0 \\ 0 & \mathbf{I} \end{pmatrix} z_k + \begin{pmatrix} B(\theta_k) \\ 0 \end{pmatrix} u_k + \begin{pmatrix} \xi_k \\ 0 \end{pmatrix} =: \tilde{f}(z_k, u_k). \quad (10)$$

The parameters  $\theta$  are assumed to be deterministic, but not known to the controller. This uncertainty is captured by the distribution  $p(\theta)$  representing the lack of knowledge.

#### 4.1 Certainty Equivalent Control Gives a Nominal Reference Trajectory

The certainty equivalent model is built on the assumption that the uncertain  $\theta$  coincide with their most likely value, the mean  $\hat{\theta}$  of  $p(\theta)$ , and that the system propagates deterministically without noise. This means that the nominal parameters  $\bar{\theta}$  are the current mean values  $\hat{\theta}$ ,

which decouples  $\theta$  entirely from  $x$  in Eq. (10), and the optimal control for the finite horizon problem can be computed by dynamic programming (DP) (Aoki, 1967), yielding an optimal linear control law

$$\bar{u}_j^* = -(\bar{B}^\top \bar{K}_{j+1} \bar{B} + U_j)^{-1} \bar{B}^\top [\bar{K}_{j+1} \bar{A} \bar{x}_j + \bar{p}_{j+1}],$$

where we have momentarily simplified notation to  $\bar{A} = A(\bar{\theta}_j)$ ,  $\bar{B} = B(\bar{\theta}_j)$ ,  $\forall j$ , because the  $\bar{\theta}_j$  are constant. The  $\bar{K}_j$  and  $\bar{p}_j$  for  $j = k+1, \dots, T$  are defined and computed recursively as

$$\begin{aligned} \bar{K}_j &= \bar{A}^\top \left( \bar{K}_{j+1} - \bar{K}_{j+1} \bar{B} (\bar{B}^\top \bar{K}_{j+1} \bar{B} + U_j)^{-1} \bar{B}^\top \bar{K}_{j+1} \right) \bar{A} + W_j & \bar{K}_T &= W_T \\ \bar{p}_j &= \bar{A}^\top \left( \bar{p}_{j+1} - \bar{K}_{j+1} \bar{B} (\bar{B}^\top \bar{K}_{j+1} \bar{B} + U_j)^{-1} \bar{B}^\top \bar{K}_{j+1} \right) \bar{p}_{j+1} - W_j r_j & \bar{p}_T &= -W_T r_T, \end{aligned}$$

where  $r$  is the reference trajectory to be followed. This CE controller gives the *nominal trajectory* of inputs  $\bar{\mathbf{u}}_{k:T-1}$  and states  $\bar{\mathbf{x}}_{k:T}$ , from the current time  $k$  to the horizon  $T$ . The true future trajectory is subject to stochasticity and uncertainty, but the deterministic nominal trajectory  $\bar{\mathbf{x}}$ , with its optimal control  $\bar{\mathbf{u}}^*$  and associated nominal cost  $\bar{J}_k^* = \mathcal{L}(\bar{\mathbf{x}}_{k:T}, \bar{\mathbf{u}}_{k:T}^*)$  provides a base, relative to which an approximation will be constructed.

## 4.2 Quadratic Expansion Around the Nominal Defines Cost of Uncertainty

The central idea of AD control is to project the nonlinear objective  $J_k(\mathbf{u}_{k:T-1}, p(x_k))$  of Eq. (3) into a quadratic, by locally linearizing around the nominal trajectory  $\mathbf{x}$  and maintaining a joint Gaussian belief.

To do so, we introduce small perturbations around nominal cost, states, and control:  $\Delta J_j = J_j - \bar{J}_j$ ,  $\Delta z_j = z_j - \bar{z}_j$ , and  $\Delta u_j = u_j - \bar{u}_j$ . These perturbations arise from both the stochasticity of the state and the parameter uncertainty. Note that a change in the state results in a change of the control signal, because the optimal control signal in each step depends on the state. Even though the origin of the uncertainties is different ( $\Delta x$  arises from stochasticity and  $\Delta \theta$  from the lack of knowledge), both can be modeled in a joint probability distribution.

Approximate Gaussian filtering ensures that beliefs over  $\Delta z$  remain Gaussian:

$$p(\Delta z_j) = \mathcal{N} \left[ \begin{pmatrix} \Delta x_j \\ \Delta \theta_j \end{pmatrix}; \begin{pmatrix} \Delta \hat{x}_j \\ 0 \end{pmatrix}, \begin{pmatrix} \Sigma_j^{xx} & \Sigma_j^{x\theta} \\ \Sigma_j^{\theta x} & \Sigma_j^{\theta\theta} \end{pmatrix} \right].$$

Note that shifting the mean to the nominal trajectory does not change the uncertainty. Note further that the expected perturbation in the parameters is nil. This is because the parameters are assumed to be deterministic and are not affected by any state or input.

Calculating the Gaussian filtering updates is in principle not possible for future measurements, since it violates the causality principle (Glad and Ljung, 2000). Nonetheless, it is possible to use the *expected* measurements to simulate the effects of the future measurements on the uncertainty, since these effects are deterministic. This is sometimes referred to as preposterior analysis (Raiffa and Schlaifer, 1961).

To second order around the nominal trajectory, the cost is approximated by

$$J_k(\mathbf{u}_{k:T-1}, p(x_k)) = \bar{J}_k^* + \Delta J_k \approx \bar{J}_k^* + \Delta \tilde{J}_k,$$

where  $\tilde{J}_k^*$  is the optimal cost for the nominal system and  $\Delta\tilde{J}_k$  is the approximate additional cost from the perturbation:

$$\Delta\tilde{J}_k := \mathbb{E}_{\mathbf{x}_{k:T}} \left[ \sum_{j=k}^T \left\{ (\bar{x}_j - r_j)^\top W_j \Delta x_j + \frac{1}{2} \Delta x_j^\top W_j \Delta x_j \right\} + \sum_{j=k}^{T-1} \left\{ \bar{u}_j^\top U_j \Delta u_j + \frac{1}{2} \Delta u_j^\top U_j \Delta u_j \right\} \right]. \quad (12)$$

Although the uncertain parameters  $\theta$  do not show up explicitly in the above equation, this step captures dual effects: The uncertainty of the trajectory  $\Delta\mathbf{x}$  depends on  $\theta$  via the dynamics. Higher uncertainty over  $\theta$  at time  $j-1$  causes higher predictive uncertainty over  $\Delta x_j$  (for each  $j$ ), and thus increases the expectation of the quadratic term  $\Delta x_j^\top W_j \Delta x_j$ . Control that decreases uncertainty in  $\theta$  can lower this approximate cost, modeling the benefit of exploration. For the same reason, Eq. (12) is in fact still not a quadratic function and has no closed form solution. To make it tractable, Tse and Bar-Shalom (1973) make the ansatz that all terms in the expectation of Eq. (12) can be written as  $g_j + p_j^\top \Delta z_j + 1/2 \Delta z_j^\top K_j \Delta z_j$ . This amounts to applying dynamic programming on the perturbed system. Expectations over the cost under Gaussian beliefs on  $\Delta z$  can then be computed analytically. Because all  $\Delta\theta$  have zero mean, linear terms in these quantities vanish in the expectation. This allows analytic minimization of the approximate optimal cost for each time step

$$\Delta\tilde{J}_j^*(p(x_j)) = \min_{\Delta u_j} \left\{ (x_j - r_j)^\top W_j \Delta \hat{x}_{j|j} + \frac{1}{2} \Delta \hat{x}_{j|j}^\top W_j \Delta \hat{x}_{j|j} + u_j^\top U \Delta u_j + \frac{1}{2} \Delta u_j^\top U \Delta u_j \right. \\ \left. + \frac{1}{2} \text{tr} \left[ W_j \Sigma_{j|j}^{xx} \right] + \mathbb{E}_{\Delta x_{j+1}} \left[ \Delta\tilde{J}_{j+1}^*(\mathbf{y}_{1:j+1}) | p(x_j) \right] \right\}, \quad (13)$$

which is feasible given an explicit description of the Gaussian filtering update. It is important to note that, assuming extended Kalman filtering, the update to the mean from *expected* future observations  $y_{j+1}$  is nil. This is because we expect to see measurements consistent with the current mean estimate. Nonetheless, the (co-)variance changes depending on the control input  $u_j$ , which is the dual effect.

Following the dynamic programming equations for the perturbed problem, including the additional cost from uncertainty, the resulting cost amounts to (Tse et al., 1973)

$$\Delta\tilde{J}_k^*(p(x_k)) = \tilde{g}_{k+1} + \tilde{p}_{k+1}^\top \Delta \hat{z}_k + \frac{1}{2} \Delta \hat{z}_k^\top \tilde{K}_{k+1} \Delta \hat{z}_k \\ + \frac{1}{2} \text{tr} \left\{ W_T \Sigma_{T|T}^{xx} + \sum_{j=k}^{T-1} \left[ W_j \Sigma_{j|j}^{xx} + (\Sigma_{j+1|j} - \Sigma_{j+1|j+1}) \tilde{K}_{j+1} \right] \right\}$$

(where we have neglected second-order effects of the dynamics). Recalling that  $\Delta \hat{z} = 0$  and dropping the constant part, the dual cost can be approximated to be

$$J_k^d = \frac{1}{2} \text{tr} \left\{ W_T \Sigma_{T|T}^{xx} + \sum_{j=k}^{T-1} \left[ W_j \Sigma_{j|j}^{xx} + (\Sigma_{j+1|j} - \Sigma_{j+1|j+1}) \tilde{K}_{j+1} \right] \right\} \quad (= \Delta\tilde{J}_k^* - \text{const})$$

where the recursive equation

$$\tilde{K}_j = \tilde{A}^\top \left( \tilde{K}_{j+1} - \tilde{K}_{j+1} \tilde{B} (B^\top K_{j+1}^{xx} B + U_j)^{-1} \tilde{B}^\top \tilde{K}_{j+1} \right) \tilde{A} + \tilde{W}_j \quad \tilde{K}_T = \tilde{W}_T$$

is defined for the augmented system (10), with  $\tilde{A} = \frac{\partial}{\partial z} \tilde{f}$ ,  $\tilde{B} = \frac{\partial}{\partial u} \tilde{f}$  and  $\tilde{W}_j = \text{blkdiag}(W_j, \mathbf{0})$ . The approximation to the overall cost is then  $\bar{J}_k^* + J_k^d$ , which is used in the subsequent optimization procedure.

### 4.3 Optimization of the Current Control Input Gives Approximate Dual Control

The last step ③ amounts to the outer loop of the overall algorithm. A gradient-free black-box optimization algorithm is used to find the minimum of the dual cost function. In every step, this algorithm proposes a control input  $u_k$  for which the dual cost is evaluated.

Depending on  $u_k$ , approximate filtering is carried out to the horizon. The perturbation control is plugged into Eq. (13) to give an analytic, recursive definition for  $\tilde{K}_j$ , and an approximation for the dual cost  $J_k^d$ , as a function of the current control input  $u_k$ .

Nonlinear optimization—through repetitions of steps ① and ② for proposed locations  $u_k$ —then yields an approximation to the optimal dual control  $u_k^*$ . Conceptually the simplest part of the algorithm, this outer loop dominates computational cost, because for every location  $u_k$  the whole machinery of ① and ② has to be evaluated.

## 5. Extension to Contemporary Machine Learning Models

The preceding section reviewed the treatment of dual control in linear dynamical systems from Tse and Bar-Shalom (1973). In this section, we extend the approach to inference on, and dual control of, the dynamics of *nonlinear* dynamical systems. This extension is guided by the desire to use a number of popular, standard regression frameworks in machine learning: Parametric general least-squares regression, nonparametric Gaussian process regression, and feedforward neural networks (including the base case of logistic regression).

### 5.1 Parametric Nonlinear Systems

We begin with the generalized linear model mentioned in Eq. (1). The nonlinear features  $\phi$  can in principle be any function (popular choices include sines and cosines, radial basis functions, sigmoids, polynomials and others), with the caveat that their structure crucially influences the properties of the model. From a modeling perspective, this approach is quite standard for machine learning. However, the dynamical learning setting requires a few adaptations: First, to allow the modeling of higher-order dynamical systems, the original states must be included. This gives features of the form  $\phi(x)^\top = (x^\top \quad \varphi(x)^\top)$ , consisting of the linear representation, augmented by general features  $\varphi$ .

The next challenge is that the optimal control for nonlinear dynamical systems cannot be optimized in closed form using dynamic programming, not even for the deterministic nominal system. Instead, we find the nominal reference trajectory using nonlinear model predictive control (Allgöwer et al., 1999; Diehl et al., 2009). In our case, we begin with dynamic programming on a locally linearized system, then optimize nonlinearly with a numerical method across the trajectory. This adds computational cost, and requires some care to achieve stable optimization performance for specific system setups.

Filtering from observations is also more involved in the case of nonlinear dynamics. In the experiments reported below, we stayed within the extended Kalman filtering framework

to retain Gaussian beliefs over the states and parameters. Extensions of this approach to more elaborate filtering methods are an interesting direction for future work. This includes relatively standard options like unscented Kalman filtering (Uhlmann, 1995), but also more recent developments in machine learning and probabilistic control, such as analytic moment propagation where the features  $\varphi$  allow this (e.g., Deisenroth and Rasmussen, 2011).

The final problem is the generalization of the derivations from the preceding sections to the nonlinear dynamics. We take a relatively simplistic approach, which nevertheless turns out to work well. A linearization gives locally linear dynamics whose structure closely matches Eq. (10):

$$\begin{aligned} z_{k+1} &= \begin{pmatrix} \bar{x}_{k+1} + \Delta x_{k+1} \\ \bar{\theta}_{k+1} + \Delta \theta_{k+1} \end{pmatrix} = \begin{pmatrix} A(\theta_k) & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \phi(\bar{x}_k + \Delta x_k) \\ \bar{\theta}_k + \Delta \theta_k \end{pmatrix} + \begin{pmatrix} B(\theta_k) \\ 0 \end{pmatrix} u_k + \begin{pmatrix} \xi_k \\ 0 \end{pmatrix} \\ &\approx \begin{pmatrix} A(\bar{\theta}_k) & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \phi(\bar{x}_k) \\ \bar{\theta}_k \end{pmatrix} + \begin{pmatrix} B(\bar{\theta}_k) \\ 0 \end{pmatrix} u_k + \begin{pmatrix} \xi_k \\ 0 \end{pmatrix} \\ &\quad + \begin{pmatrix} A(\bar{\theta}_k) \frac{\partial}{\partial x_k} \phi(\bar{x}_k) & \frac{\partial}{\partial \theta_k} (A(\bar{\theta}_k) \phi(\bar{x}_k) + B(\bar{\theta}_k) u_k) \\ 0 & I \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta \theta_k \end{pmatrix}. \end{aligned}$$

This essentially amounts to extended Kalman filtering on the augmented state. Using this linearization, the approximation described in Sec. 4 can be applied analogously.

## 5.2 Nonparametric Gaussian Process Dynamics Models

The above treatment of parametric linear models makes it comparably easy to extend the description from finitely many feature functions to an infinite-dimensional feature space defining a Gaussian process (GP) dynamics model: Assume that the true dynamics function  $f$  is a draw from a Gaussian process prior  $p(f) = \mathcal{GP}(f; m, \bar{\kappa})$  with prior mean function  $m: \mathbb{R}^n \rightarrow \mathbb{R}^n$ , and prior covariance function (kernel)  $\bar{\kappa}: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n \times \mathbb{R}^n$ . This is using the widely used notion of “multi-output regression” (Rasmussen and Williams, 2006, § 9.1), i.e. formulating the covariance as

$$\text{cov}(f_i(x), f_j(x')) = \bar{\kappa}_{ij}(x, x').$$

To simplify the treatment, we will assume that the covariance factors between inputs and outputs, i.e.  $\bar{\kappa}_{ij}(x, x') = V_{ij} \kappa(x, x')$  with a univariate kernel  $\kappa: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  and a positive semi-definite matrix  $V \in \mathbb{R}^{n \times n}$  of output covariances. By Mercer’s theorem (e.g., König, 1986; Rasmussen and Williams, 2006), the kernel can be decomposed into a converging series over eigenfunctions  $\phi(x)$ , as

$$\kappa(x, x') = \sum_{\ell=1}^{\infty} \lambda_{\ell} \phi_{\ell}(x) \phi_{\ell}^*(x'), \tag{14}$$

where  $\phi_{\ell}: \mathbb{R}^n \rightarrow \mathbb{R}$  are functions that are orthonormal relative to some measure  $\mu$  over  $\mathbb{R}^n$  (the precise choice of which is irrelevant for the time being), with the property

$$\int \kappa(x, x') \phi_{\ell}(x') d\mu(x') = \lambda_{\ell} \phi_{\ell}(x).$$

Precisely in this sense, Gaussian process regression can be written as “infinite-dimensional” Bayesian linear regression. We will use the suggestive, and somewhat abusive notation  $f_k(x_k) = L\Omega_k\Phi(x_k)$  for this generative model, defined as

$$f_k^i(x_k) = \sum_{j=1}^T L_{ij} \sum_{\ell=1}^{\infty} \Omega_k^{j\ell} \phi_i(x_k) \quad (15)$$

where  $L$  is a matrix satisfying  $LL^\top = V$  (e.g., the Cholesky decomposition), and the elements of  $\Omega$  are draws from the “white” Gaussian process  $\Omega_k^{j\ell} \sim \mathcal{N}(0, \lambda_\ell)$ . Because of Mercer’s theorem above, Eq. (15) exists in  $\mu^2$  expectation, and is well-defined in this sense. This notation allows writing Eq. (2) as a nonparametric prior with mean  $\hat{\theta}_0$  and covariance  $\Sigma_0^{\theta\theta} = V \otimes (\Phi\Lambda\Phi^\top)$  where  $\Lambda$  is an infinite diagonal matrix with diagonal elements  $\Lambda_{\ell\ell} = \lambda_\ell$  (the matrix multiplication  $\Phi\Lambda\Phi^\top$  is here defined as in Eq. (15)).

Using this notation, a tedious but straightforward linear algebra derivation (see Appendix A) shows that the posterior over  $z^\top = (x^\top \ \theta^\top)$  after a number  $k$  of EKF-linearized Gaussian observations is a tractable Gaussian process, for which the Gram matrix

$$\mathcal{G} = \mathcal{P} + \mathcal{Q} + \mathcal{K} + \mathcal{F}^{-1}\mathcal{R}\mathcal{F}^{-\top}$$

consists of the parts

$$\mathcal{P} = \begin{bmatrix} P_0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathcal{Q} = (Q \otimes I) \quad \mathcal{K} = \kappa(\mathbf{y}_{1:m}, \mathbf{y}_{1:m}) \quad \mathcal{R} = (R \otimes I)$$

of appropriate size, depending on the current time  $k$ . The multi-step state transition matrix

$$\mathcal{F} = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ A_1 & I & 0 & \cdots & 0 \\ A_2 A_1 & A_2 & I & \cdots & 0 \\ \vdots & & & \ddots & \\ A_m \cdots A_1 & A_m \cdots A_2 & \cdots & & I \end{bmatrix} \quad \text{with} \quad \mathcal{F}^{-1} = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ -A_1 & I & 0 & \cdots & 0 \\ 0 & -A_2 & I & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & -A_m & I \end{bmatrix}$$

is needed to account for the effect of the measurement noise  $R$  over time. The  $A$ -matrices are the Jacobians  $\nabla_x f(x)|_{x_k}$ .

The posterior mean now evaluates to

$$\begin{bmatrix} \hat{x}_k \\ \hat{\theta}_k \end{bmatrix} = \begin{bmatrix} \hat{x}_{k-1} \\ 0 \end{bmatrix} + \begin{bmatrix} \Phi(\hat{x}_{k-1})\Lambda\Phi(\mathbf{y}_{1:k-1})^\top \\ \Lambda\Phi(\mathbf{y}_{1:k-1})^\top \end{bmatrix} \mathcal{G}^{-1} [\Phi(\mathbf{y}_{1:k-1})\Lambda\Phi(\hat{x}_{k-1})^\top \quad \Phi(\mathbf{y}_{1:k-1})\Lambda] \quad (16)$$

and the posterior covariance is comprised of

$$\bar{\Sigma}_k^{xx} = A_k \Sigma_k^{xx} A_k^\top + Q + \Phi_k \Sigma_k^{\theta x} A_k^\top + A_k \Sigma_k^{x\theta} \Phi_k^\top + \Phi_k \Sigma_k^{\theta\theta} \Phi_k^\top \quad (17a)$$

$$\Sigma_k^{xx} = \bar{\Sigma}_{k-1}^{xx} - \bar{\Sigma}_{k-1}^{xx} [\bar{\Sigma}_{k-1}^{xx} + R]^{-1} \bar{\Sigma}_{k-1}^{xx} \quad (17b)$$

$$\Sigma_k^{x\theta} = \mathcal{F}_{k,:} \Phi(\mathbf{y}) \Lambda - \mathcal{F}_{k,:} [\mathcal{P} + \mathcal{K} + \mathcal{Q}] \mathcal{G}^{-1} \Phi(\mathbf{y}) \Lambda \quad (17c)$$

$$\Sigma_k^{\theta x} = (\Sigma_k^{x\theta})^\top \quad (17d)$$

$$\Sigma_k^{\theta\theta} = \Lambda - \Lambda \Phi(\mathbf{y})^\top \mathcal{G}^{-1} \Phi(\mathbf{y}) \Lambda. \quad (17e)$$

This formulation, together with the expositions in the preceding sections, defines a nonparametric dual control algorithm for Gaussian process priors. It is important to stress that this posterior is indeed “tractable” in so far as it depends only on a Gram matrix of size  $nT \times nT$ , and the posterior over any  $f(x)$  can be computed in time  $\mathcal{O}((nT)^3)$ , despite the infinite-dimensional state space.

### 5.2.1 AN APPROXIMATION OF CONSTANT COST

In practical control applications, continuously rising inference cost is rarely acceptable. It is thus necessary to project the GP belief onto a finite representation, replacing the infinite sum in Eq. (14) with a finite one, to bound the computational cost of the matrix inversion in Eqs. (16), (17c) and (17e). We do so by projecting into a pre-defined finite basis of functions drawn from the eigen-spectrum of the kernel with respect to the Lebesgue measure. This approach has been recently popular elsewhere in regression (Rahimi and Recht, 2008). For readers unaware of this line of work, here is a short, self-contained introduction:

By Bochner’s theorem (e.g., Stein, 1999; Rasmussen and Williams, 2006), the covariance function  $k(r)$  (with  $r = |x - x'|$ ) of a stationary  $\mu^2$  continuous random process can be represented as the Fourier transform of a positive finite measure and, if that measure has a density  $S(s)$ , as the Fourier dual of  $S$ :

$$\kappa(r) = \int S(s)e^{2\pi isr} ds,$$

This means that the eigenfunctions of the kernel are trigonometric functions, and stationary covariance functions, like the commonly used square exponential kernel

$$\kappa_{\text{SE}}(x, x') = \exp\left(-\frac{(x - x')^2}{2\lambda^2}\right),$$

can be approximated by sine and cosine basis functions as

$$\kappa(x, x') \approx \tilde{\kappa}(x, x') = \sqrt{\frac{2}{F}} \sum_{i=1}^{F/2} \sin(\omega_{2i-1}|x - x'|) + \cos(\omega_{2i}|x - x'|),$$

where the frequencies  $\omega_i$  of the feature functions is sampled from the power spectrum of the process. An example of such kernel approximation is shown in Fig. 3. With increasing number of features, the approximation can be chosen as closely to the true covariance function as needed, while keeping the number of features in a range that is still feasible within the time constraints of the control algorithm.

### 5.3 Dual Control of Feedforward Neural Networks

Another extension of the parametric linear models of Section 5.1 is to allow for a nonlinear parametrization of the dynamics function:

$$f(x; \theta) = \sum_i^F \theta_i^{\text{lin}} \phi_i(x; \theta_i^{\text{nonlin}}).$$

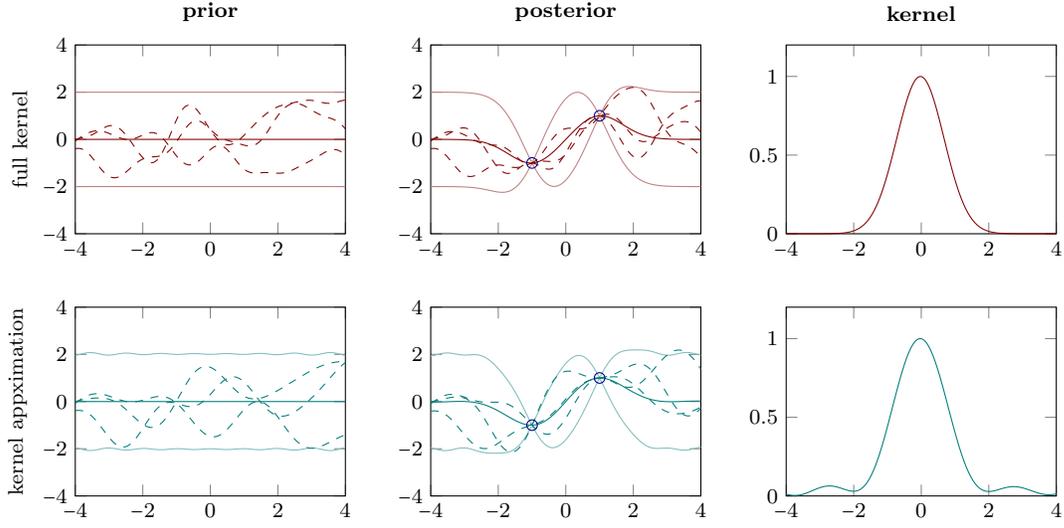


Figure 3: Prior (left), posterior (middle) and kernel function (right) of both the full kernel function (top row) and the approximate kernel (bottom row). The thick lines represent the mean and the thin lines show two standard deviations. The dashed lines are samples from the shown distributions.

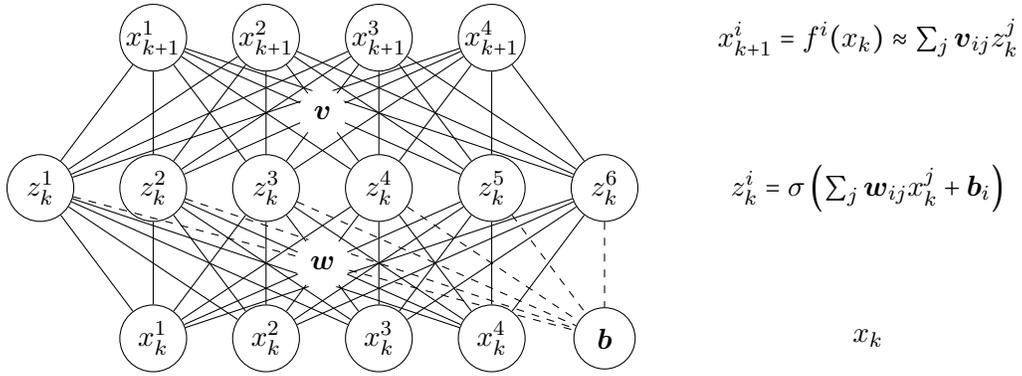


Figure 4: A two-layer feedforward neural network. Sketch to illustrate the structure of Eq. (18).

A particularly interesting example of this structure are multilayer perceptrons. Consider a two-layer network with logistic link function

$$f(x) = \sum_i \mathbf{v}_i \sigma(\mathbf{w}_i x + \mathbf{b}_i), \quad (18)$$

where  $\mathbf{v}$  are the weights from the latent to the output layer,  $\mathbf{w}$  are the weights from input to hidden units, and  $\mathbf{b}$  are the biases of the hidden units (see Fig. 4).

Neural networks are used in control quite regularly, see e.g., Nguyen and Widrow (1990). Instead of using backpropagation and stochastic gradient descent as in most applications of neural networks (Rumelhart et al., 1986; Robbins and Monro, 1951), the EKF inference procedure can be used to train the weights as well (Singhal and Wu, 1989). This is possible because the EKF linearization can also be applied for the nonlinear link function, e.g., the logistic function. Speaking in terms of feature functions, not only the weight of each feature but also the shape (steepness) can be inferred. A limiting factor for this inference naturally is the number of data points: the more features and parameters are introduced, the more data points are necessary to learn.

Using the state augmentation  $z^\top = (x^\top \ v^\top \ w^\top)$ , and linearizing w.r.t. all parameters in each step, the EKF inference on the neural network parameters allows us not only to apply relatively cheap inference on them, but also to use the dual control framework to plan control signals, accounting for the effect of future observations and the subsequent change in the belief. This means the adaptive dual controller described in Sec. 4 can identify those parts of the neural net that are relevant for applying optimal control to the problem at hand. In Sec. 6.3, we show an experiment with these properties.

## 6. Experiments

A series of experiments on single-episode tasks with continuous state space highlights qualitative differences between the adaptive dual (AD) controller and three other controllers: An oracle controller with access to the true parameters, which provides an unattainable lower bound (LB) on the achievable performance, a certainty equivalent (CE) controller as described in Sec. 4.1, and a controller minimizing the sum of CE cost and the Bayesian exploration bonus (BEB) (see Kolter and Ng, 2009):

$$\ell_{\text{BEB}} = \tau [\text{sqrt}(\text{diag}(\Sigma^{\theta\theta}))]^\top [\text{sqrt}(\text{diag}(\Sigma^{\theta\theta}))]$$

( $\tau$  is a scalar exploration weight). The additional cost term  $\ell_{\text{BEB}}$  is evaluated for the predicted parameter covariance where the prediction time is chosen according to the order of the system such that the effect of the current control signal shows up in the belief over the parameters. This type of controller is sometimes also called dual control, while being referred to as *explicit dual control*, where the dual features are obtained by a modified cost function (Filatov and Unbehauen, 2000).

Every experiment was repeated 50 times with different random seeds, which were shared across controllers for comparability. All systems presented below are very simple setups. Their primary point is to show qualitative differences of the controllers' behavior. The experiments were done with different approximations from the preceding section to show experimental feasibility for each of them.

The feature set used for a specific application is part of the prior assumptions for that application. Large uncertainty requires flexible models (which take longer to converge, and require more exploration). Feature selection is important, but since it is independent of the dual control framework itself and a broad topic on its own, it is beyond the scope of this paper. In the following experiments, different feature sets are used both as examples for the flexibility of the framework, but also to model different structural knowledge about the problems at hand.

### 6.1 On a Simple Scalar System, AD Control Matches Exact Dual Control Well

For the noise-free linear system of Sec. 3.1, ( $a = 1$  (known),  $b = 2$ ,  $p(b) = \mathcal{N}(b; 1, 10)$ ,  $Q = 10^{-1}$ ,  $R = 0$ ,  $W = 1$ ,  $\Lambda = 1$ ,  $T = 2$ ), Fig. 1, right, compares the cost functions of the various controllers and the approximately exact sampling solution (which is only available for this very simple setup). All cost functions are shifted by an irrelevant constant. The CE cost is quadratic and indifferent about zero. The BEB ( $\tau = 0.1$ ) gives additional structure near zero that encourages learning. While qualitatively similar to the dual cost, its global minimum is almost at the same location as that of CE. The dual control approximates the sampling solution much closer.

### 6.2 Faced with Time-Varying State Cost, AD Control Holds Off Exploration Until Suitable

A cart on a rail is a simple example for a dynamical system. Combined with a nonlinearly varying slope, a simple but nonlinear system can be constructed. The dynamics, prior beliefs, and true values for the parameters are chosen to be

$$x_{k+1} = \begin{bmatrix} 1 & 0.4 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 & 0 \\ \theta^1 & \theta^2 \end{bmatrix} \begin{bmatrix} \varphi^1(x_k^1) \\ \varphi^2(x_k^1) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k \quad \theta \sim \mathcal{N} \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \quad \theta_{\text{true}} = \begin{bmatrix} 0.8 \\ 0.4 \end{bmatrix},$$

where superscripts denote vector elements. The nonlinear functions  $\varphi$  are shifted logistic functions of the form

$$\varphi^1(x) = -\frac{1}{1 + e^{(x+5)}} \quad \varphi^2(x) = \frac{1}{1 + e^{-(x-5)}}, \quad (19)$$

and disturbance/noise is chosen to be  $R = Q = 10^{-2}I$ . We use this setup as a testbed for a time-structured exploration problem. The actual system and its dynamics are relatively irrelevant here, as we will focus on a complication caused by the cost function: The reference to be tracked is

$$\mathbf{r}_{0:11} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{r}_{12:14} = \begin{bmatrix} 10 \\ 0 \end{bmatrix} \quad \mathbf{r}_{15} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{r}_{16:18} = \begin{bmatrix} -10 \\ 0 \end{bmatrix} \quad \mathbf{r}_{19:20} = \begin{bmatrix} 0 \\ 0 \end{bmatrix};$$

it is also shown in each plot of Fig. 5 as dashed orange line. The state weighting is time-dependent

$$\mathbf{W}_{0:5} = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{W}_{5:10} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{W}_{11:20} = \begin{bmatrix} 100 & 0 \\ 0 & 0 \end{bmatrix},$$

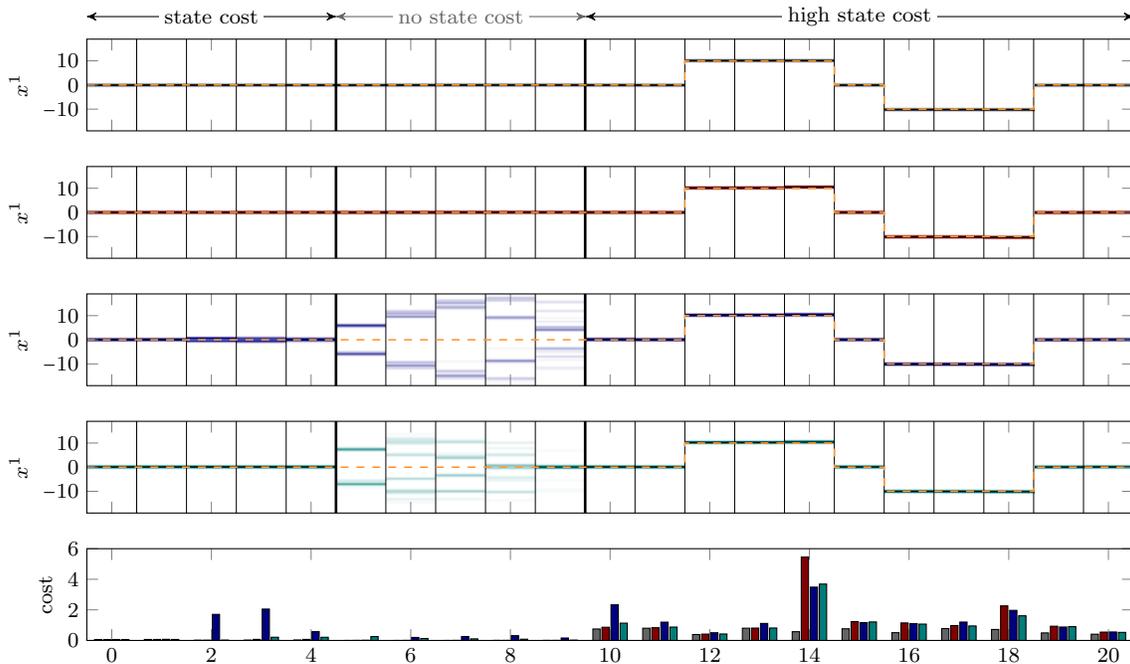


Figure 5: **Top four:** Density estimate for 50 trajectories (first state). From top to bottom: optimal oracle control (gray), certainty equivalent control (red), CE with Bayesian exploration bonus (blue), approximate dual control (green). Reference trajectory in dashed orange. **Bottom:** The mean cost per time step is shown in the bottom plot, with colors matching the controllers noted above.

and control cost is relatively low:  $\Lambda = 10^{-3}$ . The task, thus, is to first keep the cart fixed in the starting position to high precision, for the first 4 time steps. This is followed by a “loose” period between time steps 5 and 10. Then, the cart has to be moved to one side, back to the center, to the other side, and back again, all at high cost. A good exploration strategy in this setting is to act cautiously for the first 5 time steps, then aggressively explore in the “loose” phase, to finally be able to control the motion with high precision.

The inference model is a GP with approximated SE kernel, as described in Sec. 5.2.1. We use 30 alternating sine and cosine features that are distributed according to the power spectrum of the full SE kernel. Since the true nonlinearity of Eq. (19) is not of this form, the approximation is out of model and the lower bound controller only represents a perfectly learned, but still not exact, model.

Fig. 5 shows a density estimated from 50 state trajectories for the four different controllers. The lower bound controller (top) controls precisely at times of high cost, and does nothing for times with zero cost, controlling perfectly up to the measurement and state disturbances. The certainty equivalent controller (second from top) never explores actively, it only learns “accidentally” from observations arising during the run. Since the initial trajectory requires little action, it is left with a bad model when the reference starts to move at time step 12. The exploration bonus controller (second from bottom) continuously explores, because it

has no way of knowing about the “loose” phase ahead. Of course, this strategy incurs a higher cost initially. The dual controller (bottom) efficiently holds off exploration until it reaches the “loose” phase, where it explores aggressively.

### 6.3 AD Control Distinguishes Necessary and Unnecessary Parameter Exploration

The system including nonlinearities for this experiment is the same as before, although with noise parameters  $R = Q = 10^{-3}I$ . The reference trajectory and state weighting are much simpler, though:

$$\mathbf{r}_{0:11} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{r}_{12:18} = \begin{bmatrix} 10 \\ 0 \end{bmatrix} \quad \mathbf{r}_{18:20} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

with the time-dependent weighting

$$\mathbf{W}_{0:10} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{W}_{11:20} = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix},$$

allowing for identification in the beginning, while penalizing deviations of the first state in later time steps.

Important to note here is that the reference trajectory only passes areas of the state space where  $\varphi^1$  is strong, and  $\varphi^2$  is negligible. Good exploration thus will ignore  $\theta^2$ , but this can only be found through reasoning about future trajectories.

In this experiment, the learned model is of the neural network form described in Sec. 5.3. We use 4 logistic features (see Eq. (18)) with two free parameters each ( $w_i$  and  $v_i$ ) and equally spaced  $b_i$  between  $-5$  and  $5$ , the locations of the true nonlinear features. This means it is possible to learn the perfect model in this case.

Fig. 6 shows a density estimated from 50 state trajectories for the four different controllers. Because of symmetry in the cost function and feature functions, BEB (with  $\tau = 1$ ) cannot “decide” between the relevant  $\theta^1$  and the irrelevant  $\theta^2$ , choosing the exploration direction stochastically. It thus sometimes reduces the uncertainty on  $\theta^2$ , which does not help the subsequent control. The AD controller ignores  $\theta^2$  completely and only identifies  $\theta^1$  in early phases, leading to good control performance.

### 6.4 AD Control Maintains Only Useful Knowledge

The last experiment is again similar to Sec. 6.2, but uses a different set of nonlinear functions, namely shifted Gaussian functions (a.k.a. radial basis functions)

$$\varphi^1(x) = e^{-\frac{(x-2)^2}{2}} \quad \varphi^2(x) = e^{-\frac{(x+2)^2}{2}} \quad \theta_{\text{true}} = \begin{bmatrix} 1.0 \\ 0.8 \end{bmatrix}.$$

For this experiment, the model is learned with parametric linear regression, according to Sec. 5.1. The fundamental difference to the other experimental setups is that the model now assumes *parameter drift*. This results in growing uncertainty for the parameters over time. (The true parameters are kept constant for simplicity.)

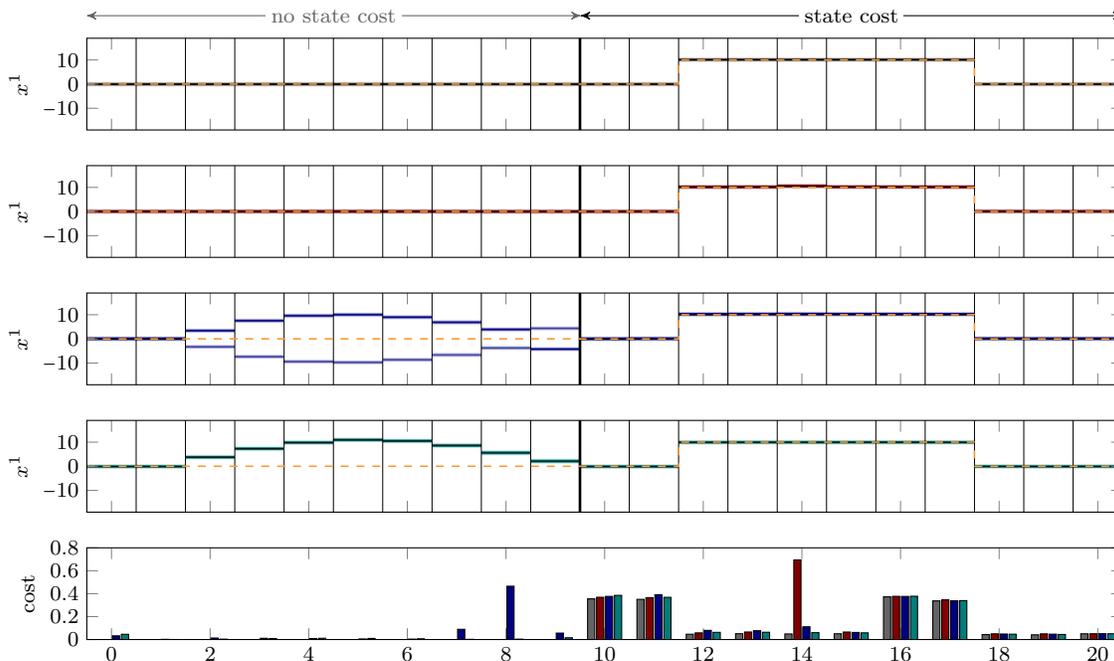


Figure 6: **Top four:** Density estimate for 50 trajectories (second state). From top to bottom: optimal oracle control (gray), certainty equivalent control (red), CE with Bayesian exploration bonus (blue), approximate dual control (green). Reference trajectory in dashed orange. **Bottom:** The mean cost per time step is shown in the bottom plot, with colors matching the controllers noted above.

The reference to be tracked passes through both nonlinear features but then stays at one of them:

$$\mathbf{r}_{0:6} = \begin{bmatrix} -5 \\ 0 \end{bmatrix} \quad \mathbf{r}_7 = \begin{bmatrix} -4 \\ 0 \end{bmatrix} \quad \mathbf{r}_8 = \begin{bmatrix} -2 \\ 0 \end{bmatrix} \quad \mathbf{r}_9 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{r}_{10:20} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$

The cost structure is

$$\mathbf{W}_{0:5} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{W}_{6:20} = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix},$$

such that there is cost starting with the linear reference trajectory at time instant 6.

Fig. 7 shows the parameter belief and relevant state of a single run of this experiment over time. It shows clearly that in the beginning necessary parameter  $\theta^1$  is learned early by BEB and the AD controller, while CE learns only “accidentally”. The BEB controller also learns the second parameter in the beginning, even though the knowledge will be lost over time. When the trajectory reaches the zone of the second parameter, the BEB controller tries to lower the growing uncertainty every now and then (visible by the drops in state  $x^1$ ), incurring high cost. AD control completely ignores the growing uncertainty on  $\theta^1$  after reaching the area of  $\theta^2$ , thus preventing unnecessary exploration.

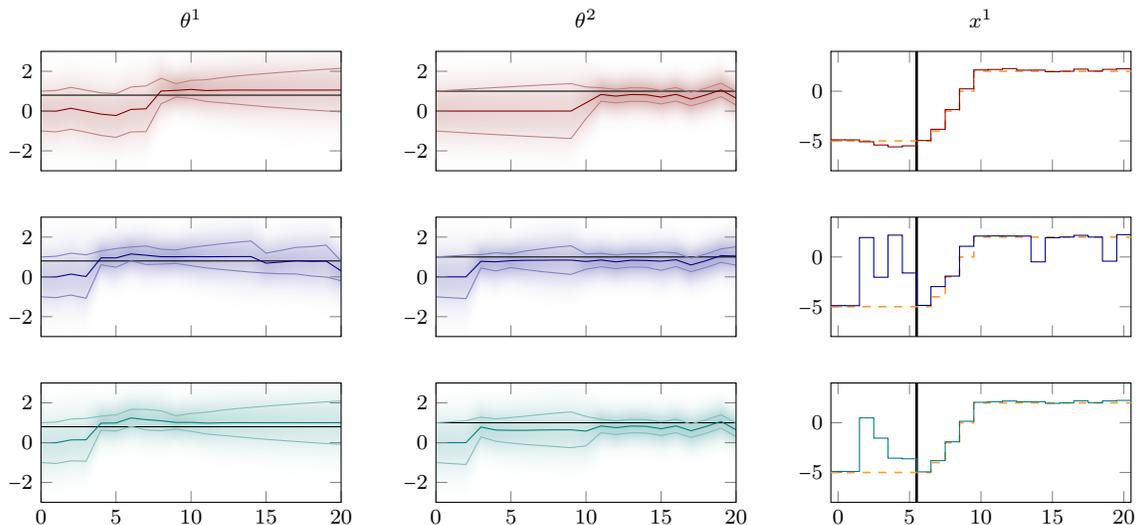


Figure 7: Parameter knowledge (left, middle) and state trajectory (right) for different controllers. From top to bottom: certainty equivalent control (red), CE with Bayesian exploration bonus (blue), approximate dual control (green). The true parameters are the black lines.

	Exp. 6.1		Exp. 6.2		Exp. 6.3		Exp. 6.4	
	mean	std	mean	std	mean	std	mean	std
Oracle	0.67	0.23	1.75	0.33	7.15	3.85	0.66	0.51
CE	0.84	0.73	2.49	0.74	15.72	5.20	1.76	0.90
CE-BEB	0.99	0.92	2.64	0.37	20.88	6.74	84.91	6.77
AD	0.77	0.43	1.96	0.34	14.33	5.40	1.62	0.56

Table 1: Average and standard deviation of costs in the experiments for 50 runs.

## 6.5 Quantitative Comparison

The above experiments aim to emphasize qualitative strengths of AD control over simpler approximations. It is desirable for a controllers to deal with flexible models of many parameters, many of which will invariably be superfluous. For reference, Table 6.5 also shows quantitative results: Averages and standard deviations of the cost, from the 50 runs for each controller. The AD controller shows good performance overall; interestingly, it also has low variance. CE and BEB were more prone to instabilities.

## 7. Conclusion

Bayesian reinforcement learning, or dual control, offers an elegant answer to the exploration-exploitation trade-off, relative to prior probabilistic beliefs. Its intricate, intractable structure requires approximations to balance another kind of trade-off, between computation and performance. This work investigated an old approximate framework from control, re-phrased

it in the language of reinforcement learning, and extended it to apply to contemporary inference methods from machine learning, including approximate Gaussian process regression and multi-layer networks. The result is a tractable approximation that captures notions of structured exploration, like the value of waiting for future exploration opportunities, and distinguishing relevant from irrelevant model parameters.

The dual control framework, in its now clearer form, offers interesting directions for research in reinforcement learning, including its combination with recent new developments in learning and planning. Following this conceptual work, the main challenge for further development is the still comparably high (but tractable) numerical load of dual control, particularly in problems of higher dimensionality.

## Appendix A. Nonparametric EKF Form

The standard Kalman filter (KF) can be found in many textbooks (e.g., Särkkä, 2013) and therefore will not be restated here. Starting from the standard equations, we derive a general multi-step formulation of the classic KF with

$$p(z_k) = \mathcal{N}(z_k, m_k, P_k).$$

From there, state augmentation with an infinite-dimensional weight vector gives the expected result.

### A.1 Derivation of the Multi-Step KF Formulation

Assuming that the result of the KF and the Gaussian process framework should be identical under certain circumstances, we wish to transform the KF to a formulation with full Gram matrix. Therefore, the prediction and update step have to be combined to

$$\begin{aligned} P_1 &= (A_0 P_0 A_0^\top + Q) - (A_0 P_0 A_0^\top + Q) H^\top S_1^{-1} H (A_0 P_0 A_0^\top + Q)^\top \\ S_1 &= H (A_0 P_0 A_0^\top + Q) H^\top + R, \end{aligned}$$

which is pretty straightforward. We're adopting a standard notation, where  $P_k$  is the covariance at time step  $k$ ,  $A_k$  is the Jacobian,  $Q$  is the drift and  $R$  is the measurement covariance. The same can be done for the second time step, but it is beneficial introducing a compact notation for the predictive covariance first

$$\begin{aligned} &(A_1 P_1 A_1^\top + Q) \\ &= \left( A_1 \left[ (A_0 P_0 A_0^\top + Q) - (A_0 P_0 A_0^\top + Q) H^\top S_1^{-1} H (A_0 P_0 A_0^\top + Q)^\top \right] A_1^\top + Q \right) \\ &= A_1 \underbrace{(A_0 P_0 A_0^\top + Q) A_1^\top + Q}_{:=g_{11}} - A_1 \underbrace{(A_0 P_0 A_0^\top + Q) H^\top}_{:=g_{10}} \underbrace{(S_1^{-1})^{-1}}_{:=g_{00}} H \underbrace{(A_0 P_0 A_0^\top + Q) A_1^\top}_{:=g_{01}} \\ &= g_{11} - g_{10} H^\top g_{00}^{-1} H g_{01}. \end{aligned}$$

Using the compact notation and defining  $S_2$  analogously to  $S_1$ , we can write the two-step update as

$$\begin{aligned} P_2 &= (g_{11} - g_{10} H^\top S_1^{-1} H g_{01}) - (g_{11} - g_{10} H^\top S_1^{-1} H g_{01}) H^\top S_2^{-1} H (g_{11} - g_{10} H^\top S_1^{-1} H g_{01}) \\ P_2 &= g_{11} - g_{10} H^\top S_1^{-1} H g_{01} - g_{11} H^\top S_2^{-1} H g_{11} - g_{10} H^\top S_1^{-1} H g_{01} H^\top S_2^{-1} H g_{10} H^\top S_1^{-1} H g_{01} \\ &\quad + g_{11} H^\top S_2^{-1} H g_{10} H^\top S_1^{-1} H g_{01} + g_{10} H^\top S_1^{-1} H g_{01} H^\top S_2^{-1} H g_{11} \\ P_2 &= g_{11} - \underbrace{\begin{bmatrix} g_{10} H^\top & g_{11} H^\top \end{bmatrix} \begin{bmatrix} S_1^{-1} + S_1^{-1} H g_{01} H^\top S_2^{-1} H g_{10} H^\top S_1^{-1} & -S_1^{-1} H g_{01} H^\top S_2^{-1} \\ -S_2^{-1} H g_{10} H^\top S_1^{-1} & S_2^{-1} \end{bmatrix}}_{:=G^{-1}} \begin{bmatrix} H g_{01} \\ H g_{11} \end{bmatrix}. \end{aligned}$$

Application of Schur's lemma gives

$$G = \begin{bmatrix} H g_{00} H^\top + R & H g_{01} H^\top \\ H g_{10} H^\top & H g_{11} H^\top + R \end{bmatrix}.$$

Assuming full state measurement ( $H = I$ ) for compactness of notation, the two-step update is

$$\begin{aligned}
 P_2 &= g_{11} - [g_{10}^\top \quad g_{11}^\top] \begin{bmatrix} g_{00} + R & g_{01} \\ g_{10} & g_{11} + R \end{bmatrix}^{-1} \begin{bmatrix} g_{01} \\ g_{11} \end{bmatrix} \\
 &= A_1 (A_0 P_0 A_0^\top + Q) A_1^\top + Q - [A_1 (A_0 P_0 A_0^\top + Q) \quad (A_1 (A_0 P_0 A_0^\top + Q) A_1^\top + Q)] \\
 &\quad \cdot \begin{bmatrix} (A_0 P_0 A_0^\top + Q) + R & (A_0 P_0 A_0^\top + Q) A_1^\top \\ A_1 (A_0 P_0 A_0^\top + Q) & (A_1 (A_0 P_0 A_0^\top + Q) A_1^\top + Q) + R \end{bmatrix}^{-1} \begin{bmatrix} (A_0 P_0 A_0^\top + Q) A_1^\top \\ (A_1 (A_0 P_0 A_0^\top + Q) A_1^\top + Q) \end{bmatrix}, \tag{20}
 \end{aligned}$$

which already looks similar to GP inference. We can now generalize this two-step result to the general form by building the Gram matrix according to

$$G = \mathcal{F} \mathcal{P} \mathcal{F}^\top + \mathcal{F} \mathcal{Q} \mathcal{F}^\top + \mathcal{R},$$

where the individual parts are

$$\mathcal{P} = \begin{bmatrix} P_0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathcal{Q} = (Q \otimes I) \quad \mathcal{R} = (R \otimes I)$$

of appropriate size, depending on the current time  $k$ . The multi-step state transition matrix

$$\mathcal{F} = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ A_1 & I & 0 & \cdots & 0 \\ A_2 A_1 & A_2 & I & \cdots & 0 \\ \vdots & & & \ddots & \\ A_m \cdots A_1 & A_m \cdots A_2 & \cdots & & I \end{bmatrix}, \quad \text{with} \quad \mathcal{F}^{-1} = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ -A_1 & I & 0 & \cdots & 0 \\ 0 & -A_2 & I & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & -A_m & I \end{bmatrix},$$

is also needed so shift the initial covariance and drift covariances through time. Put together, this results in

$$P_k = \mathcal{F}_{k,:} (\mathcal{P} + \mathcal{Q}) \mathcal{F}_{k,:}^\top - \mathcal{F}_{k,:} (\mathcal{P} + \mathcal{Q}) \mathcal{F} (\mathcal{F} \mathcal{P} \mathcal{F}^\top + \mathcal{F} \mathcal{Q} \mathcal{F}^\top + \mathcal{R})^{-1} \mathcal{F}^\top (\mathcal{P} + \mathcal{Q}) \mathcal{F}_{:,k}.$$

A more compact notation can be achieved by using  $\mathcal{F}^{-1}$  to obtain

$$P_k = \mathcal{F}_{k,:} (\mathcal{P} + \mathcal{Q}) \mathcal{F}_{k,:}^\top - \mathcal{F}_{k,:} (\mathcal{P} + \mathcal{Q}) (\mathcal{P} + \mathcal{Q} + \mathcal{F}^{-1} \mathcal{R} \mathcal{F}^{-\top})^{-1} (\mathcal{P} + \mathcal{Q}) \mathcal{F}_{:,k}. \tag{21}$$

Calculating the mean prediction is done analogously:

$$m_k = \mathcal{F}_{k,0} m_0 + \mathcal{F}_{k,:} (\mathcal{P} + \mathcal{Q}) (\mathcal{P} + \mathcal{Q} + \mathcal{F}^{-1} \mathcal{R} \mathcal{F}^{-\top})^{-1} \mathbf{y}.$$

## A.2 Augmenting the State

Instead of tracking only the state covariance, in the GP setting also the dynamics function has to be inferred. The system equations of the nonlinear system are now

$$\begin{aligned}
 x_k &= f(x_{k-1}) + q_{k-1} & q_{k-1} &\sim \mathcal{N}(0, Q) \\
 y_k &= H x_k + r_k & r_k &\sim \mathcal{N}(0, R),
 \end{aligned}$$

where  $f \sim \mathcal{GP}(0, k)$ . The inference in this model can be done through the EKF with augmented state. We adopt the weight-space view with  $f = \Phi^\top w$  (see Rasmussen and Williams, 2006) to augment the state with the infinite-dimensional weight vector  $w$ :

$$z = \begin{pmatrix} x \\ w \end{pmatrix} \quad \Sigma = \begin{pmatrix} P & \Sigma^{xw} \\ \Sigma^{wx} & \Sigma^{ww} \end{pmatrix} \quad J_A = \begin{pmatrix} \frac{\partial \bar{f}}{\partial x} & \frac{\partial \bar{f}}{\partial w} \\ 0 & I \end{pmatrix} = \begin{pmatrix} A & \Phi \\ 0 & I \end{pmatrix},$$

where, in (20), the original  $x$  is replaced by the augmented  $z$ ,  $P$  by  $\Sigma$  and  $A$  by  $J_A$ .

Choosing  $H = [I \ 0]$  so that  $H$  recovers the original states from the augmented state vector, we obtain, after calculations similar to those above, a Gram matrix with additional terms including feature functions and the prior on them:

$$G^* = G + \begin{pmatrix} \Phi_0 \Sigma_0^{ww} \Phi_0^\top & \Phi_0 \Sigma_0^{ww} (\Phi_0^\top A_1^\top + \Phi_1^\top) \\ (A_1 \Phi_0 + \Phi_1) \Sigma_0^{ww} \Phi_0^\top & (A_1 \Phi_0 + \Phi_1) \Sigma_0^{ww} (\Phi_0^\top A_1^\top + \Phi_1^\top) \end{pmatrix}.$$

At this point it is important to note that the infinite inner product  $\Phi_k \Sigma_0^{ww} \Phi_k^\top$  corresponds to an evaluation of the kernel

$$\Phi(x) \Sigma_0^{ww} \Phi(x)^\top = \sum \phi_i(x) \Sigma_{0,ij}^{ww} \phi_j(x') = \kappa(x, x').$$

This means we can write the Gram matrix as

$$G^* = G + \begin{pmatrix} \kappa_{00} & \kappa_{00} A_1^\top + \kappa_{01} \\ A_1 \kappa_{00} + \kappa_{10} & A_1 \kappa_{00} A_1^\top + \kappa_{10} A_1^\top + A_1 \kappa_{01} + \kappa_{11} \end{pmatrix},$$

where we have written  $\kappa_{..}$  for  $\kappa(\cdot, \cdot)$  to save space. In total, the Gram matrix is then

$$G^* = \mathcal{F} \mathcal{P} \mathcal{F}^\top + \mathcal{F} \mathcal{Q} \mathcal{F}^\top + \mathcal{F} \mathcal{K} \mathcal{F}^\top + \mathcal{R},$$

with  $\mathcal{K} = \kappa(\mathbf{y}_{1:m}, \mathbf{y}_{1:m})$ . Since inference is more compact and numerically stable if we absorb  $\mathcal{F}$  into the Gram matrix as in Eq. (21), we define

$$\mathcal{G} = \mathcal{P} + \mathcal{Q} + \mathcal{K} + \mathcal{F}^{-1} \mathcal{R} \mathcal{F}^{-\top}.$$

Inference is done according to

$$P_k = \mathcal{F}_{k,:} (\mathcal{P} + \mathcal{Q} + \mathcal{K}) \mathcal{F}_{k,:}^\top - \mathcal{F}_{k,:} (\mathcal{P} + \mathcal{Q} + \mathcal{K}) \mathcal{G}^{-1} (\mathcal{P} + \mathcal{Q} + \mathcal{K}) \mathcal{F}_{:,k}$$

$$\Sigma_k^{wx} = (\Sigma_k^{xw})^\top = \mathcal{F}_{k,:} \Phi(\mathbf{y}) \Sigma_0^{ww} - \mathcal{F}_{k,:} (\mathcal{P} + \mathcal{K} + \mathcal{Q}) \mathcal{G}^{-1} \Phi(\mathbf{y}) \Sigma_0^{ww}$$

$$\Sigma_k^{ww} = \Sigma_0^{ww} - \Sigma_0^{ww} \Phi(\mathbf{y})^\top \mathcal{G}^{-1} \Phi(\mathbf{y}) \Sigma_0^{ww}$$

for the covariance and

$$m_k = \mathcal{F}_{k,0} m_0 + \mathcal{F}_{k,:} (\mathcal{P} + \mathcal{Q} + \mathcal{K}) \mathcal{G}^{-1} \mathbf{y}$$

$$\bar{w}_k = \bar{w}_0 + \Sigma_0^{ww} \Phi(\mathbf{y})^\top \mathcal{G}^{-1} \mathbf{y}$$

for the mean.

## Appendix B. Gradients and Hessians of Dynamics Functions

### B.1 Neural Network Basis Functions

The neural network dynamics function is

$$f(x) = \sum_{i=1}^F v_i \sigma(w_i(x - b_i)), \quad \sigma(a) = \frac{1}{1 + e^{-a}},$$

with the well-known derivatives of the logistic

$$\frac{\partial}{\partial a} \sigma(a) = \sigma(a)(1 - \sigma(a)), \quad \frac{\partial^2}{\partial a^2} \sigma(a) = \sigma(a)(1 - \sigma(a))(3 - 2\sigma(a)).$$

The gradient of  $f(x)$  can easily found to be

$$\nabla f(x) = \begin{bmatrix} \sum_{i=1}^F v_i w_i \sigma(w_i(x - b_i))(1 - \sigma(w_i(x - b_i))) \\ \sigma(w_1(x - b_1)) \\ \vdots \\ \sigma(w_F(x - b_F)) \\ v_1(x - b_1)\sigma(w_1(x - b_1))(1 - \sigma(w_1(x - b_1))) \\ \vdots \\ v_F(x - b_F)\sigma(w_F(x - b_F))(1 - \sigma(w_F(x - b_F))) \end{bmatrix}.$$

The Hessian, written in parts, using  $a_i = w_i x + b_i$ , is:

$$\begin{aligned} \nabla_x^2 f(x) &= \sum_{i=1}^F v_i w_i^2 \sigma(a_i)(1 - \sigma(a_i))(3 - 2\sigma(a_i)) \\ \nabla_x \nabla_{v_i} f(x) &= w_i \sigma(a_i)(1 - \sigma(a_i)) \\ \nabla_x \nabla_{w_i} f(x) &= v_i \sigma(a_i)(1 - \sigma(a_i)) + (x - b_i) w_i v_i \sigma(a_i)(1 - \sigma(a_i))(3 - 2\sigma(a_i)) \\ \nabla_{v_i} \nabla_{v_i} f(x) &= 0 \\ \nabla_{v_i} \nabla_{w_i} f(x) &= (x - b_i) \sigma(a_i)(1 - \sigma(a_i)) \\ \nabla_{w_i} \nabla_{w_i} f(x) &= v_i (x - b_i)^2 \sigma(a_i)(1 - \sigma(a_i))(3 - 2\sigma(a_i)). \end{aligned}$$

### B.2 Fourier Basis Functions

The Fourier approximation to the dynamics function has the form

$$f(x) = \sqrt{\frac{2}{F}} \sum_{i=1}^{F/2} v_{2i-1} \sin(\omega_{2i-1} x) + v_{2i} \cos(\omega_{2i} x).$$

The gradient of  $f(x)$  can easily verified to be

$$\nabla f(x) = \begin{bmatrix} \sqrt{\frac{2}{F}} \sum_{i=1}^{F/2} v_{2i-1} \omega_{2i-1} \cos(\omega_{2i-1} x) - v_{2i} \omega_{2i} \sin(\omega_{2i} x) \\ \sqrt{\frac{2}{F}} \sin(\omega_1 x) \\ \sqrt{\frac{2}{F}} \cos(\omega_2 x) \\ \vdots \\ \sqrt{\frac{2}{F}} \sin(\omega_{F-1} x) \\ \sqrt{\frac{2}{F}} \cos(\omega_F x) \end{bmatrix}.$$

The Hessian, written in parts, using  $c = \sqrt{2/F}$  for normalization, is:

$$\begin{aligned}\nabla_x^2 f(x) &= c \sum_{i=1}^{F/2} -v_{2i-1} \omega_{2i-1}^2 \sin(\omega_{2i-1} x) - v_{2i} \omega_{2i}^2 \cos(\omega_{2i} x) \\ \nabla_x \nabla_{v_i} f(x) &= \begin{cases} c \omega_i \cos(\omega_i x) & i \text{ odd} \\ -c \omega_i \sin(\omega_i x) & i \text{ even} \end{cases} \\ \nabla_{v_i} \nabla_{v_i} f(x) &= 0.\end{aligned}$$

### B.3 Radial Basis Functions

With radial basis function features, the dynamics function is

$$f(x) = \sum_{i=1}^F v_i \exp\left(-\frac{(x - c_i)^2}{2\lambda^2}\right).$$

The gradient of  $f(x)$  is

$$\nabla f(x) = \begin{bmatrix} \sum_{i=1}^F v_i \exp\left(-\frac{(x-c_i)^2}{2\lambda^2}\right) \frac{(c_i-x)}{2\lambda^2} \\ \exp\left(-\frac{(x-c_1)^2}{2\lambda^2}\right) \\ \vdots \\ \exp\left(-\frac{(x-c_F)^2}{2\lambda^2}\right) \end{bmatrix}.$$

The Hessian, written in parts, is:

$$\begin{aligned}\nabla_x^2 f(x) &= \sum_{i=1}^F v_i \exp\left(-\frac{(x - c_i)^2}{2\lambda^2}\right) \left[ \left(\frac{c_i - x}{2\lambda^2}\right)^2 - \frac{1}{\lambda^2} \right] \\ \nabla_x \nabla_{v_i} f(x) &= \exp\left(-\frac{(x - c_i)^2}{2\lambda^2}\right) \frac{c_i - x}{2\lambda^2} \\ \nabla_{v_i} \nabla_{v_i} f(x) &= 0\end{aligned}$$

### References

- F. Allgöwer, T.A. Badgwell, J.S. Qin, J.B. Rawlings, and S.J. Wright. Nonlinear predictive control and moving horizon estimationan introductory overview. In *Advances in Control*, pages 391–449. Springer, 1999.
- M. Aoki. *Optimization of Stochastic Systems*. Academic Press, New York - London, 1967.
- J.-Y. Audibert, R. Munos, and C. Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.
- Y. Bar-Shalom. Stochastic dynamic programming: Caution and probing. *IEEE Transactions on Automatic Control*, 26(5):1184–1195, 1981.

- Y. Bar-Shalom and E. Tse. Dual effect, certainty equivalence, and separation in stochastic control. *IEEE Transactions on Automatic Control*, 19(5):494–500, 1974.
- Y. Bar-Shalom and E. Tse. Caution, probing, and the value of information in the control of uncertain systems. *Annals of Economic and Social Measurement*, 5(3):323–337, 1976.
- R.E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.
- D.P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 3rd edition, 2005.
- O. Chapelle and L. Li. An Empirical Evaluation of Thompson Sampling. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2249–2257, 2011.
- R. Dearden, N. Friedman, and D. Andre. Model based Bayesian exploration. In *Uncertainty in Artificial Intelligence (UAI)*, volume 15, pages 150–159, 1999.
- M.P. Deisenroth and C.E. Rasmussen. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *International Conference on Machine Learning (ICML)*, 2011.
- M. Diehl, H.J. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear MPC and moving horizon estimation. In *Nonlinear Model Predictive Control*, volume 384, pages 391–417. Springer, 2009.
- S.E. Dreyfus. Some types of optimal control of stochastic systems. *Journal of the Society for Industrial & Applied Mathematics, Series A: Control*, 2(1):120–134, 1964.
- M.O.G. Duff. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massachusetts, Amherst, 2002.
- A.A. Feldbaum. Dual Control Theory I-IV. *Avtomatika i Telemekhanika*, 21(9), 21(11), 22(1), 22(2), 1960–1961.
- N.M. Filatov and H. Unbehauen. Survey of adaptive dual control methods. *IEEE Proceedings on Control Theory and Applications*, 147(1):118–128, 2000.
- N.M. Filatov and H. Unbehauen. *Adaptive Dual Control*. Springer Verlag, Berlin, 2004.
- T. Glad and L. Ljung. *Control theory: Multivariable and Nonlinear Methods*. Taylor and Francis, New York, London, 2000.
- P. Hennig. Optimal Reinforcement Learning for Gaussian Systems. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.
- O.L.R. Jacobs and J.W. Patchell. Caution and probing in stochastic control. *International Journal of Control*, 16(1):189–199, 1972.
- H.J. Kappen. Optimal control theory and the linear Bellman equation. *Inference and Learning in Dynamic Models*, pages 363–387, 2011.

- J.Z. Kolter and A.Y. Ng. Near-Bayesian exploration in polynomial time. In *International Conference on Machine Learning (ICML)*, 2009.
- H. König. *Eigenvalue Distribution of Compact Operators*. Birkhäuser, 1986.
- W.G. Macready and D.H. Wolpert. Bandit problems and the exploration/exploitation tradeoff. *IEEE Transactions on Evolutionary Computation*, 2(1):2–22, 1998.
- D.Q. Mayne. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *International Journal of Control*, 3(1):85–95, 1966.
- D.H. Nguyen and B. Widrow. Neural networks for self-learning control systems. *IEEE Control Systems Magazine*, 10(3):18–23, 1990.
- P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete Bayesian reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2006.
- A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1177–1184, 2008.
- H. Raiffa and R. Schlaifer. *Applied statistical decision theory*. Studies in managerial economics. Harvard University, Boston, 1961.
- C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT, 2006.
- H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, Sep. 1951.
- D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
- S. Särkkä. *Bayesian filtering and smoothing*. Cambridge University Press, 2013.
- S. Singhal and L. Wu. Training multilayer perceptrons with the extended kalman algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, pages 133–140, 1989.
- N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *International Conference on Machine Learning (ICML)*, 2010.
- M.L. Stein. *Interpolation of spatial data: some theory for Kriging*. Springer Verlag, 1999.
- J. Sternby. A simple dual control problem with an analytical solution. *IEEE Transactions on Automatic Control*, 21(6):840–844, 1976.
- W.R. Thompson. On the Likelihood that one unknown probability exceeds another in view of two samples. *Biometrika*, 25:275–294, 1933.
- E. Tse and Y. Bar-Shalom. An actively adaptive control for linear systems with random parameters via the dual control approach. *IEEE Transactions on Automatic Control*, 18(2):109–117, 1973.

- E. Tse, Y. Bar-Shalom, and L. Meier III. Wide-sense adaptive dual control for nonlinear stochastic systems. *IEEE Transactions on Automatic Control*, 18(2):98–108, 1973.
- J. Uhlmann. *Dynamic Map Building and Localization: New Theoretical Foundations*. PhD thesis, University of Oxford, 1995.
- B. Wittenmark. Adaptive dual control methods: An overview. In *IFAC symposium on Adaptive Systems in Control and Signal Processing*, pages 67–72, 1995.