Chapter 2

Background

This chapter reviews background needed to understand this work. Section 2.1 briefly summarizes some important facts about spinal cord injury (SCI), the injury which motivates this work. A discussion of therapeutic approaches for spinal cord injured patients follows in Section 2.2. Section 2.3 reviews the literature on active learning and bandit algorithms, machine learning techniques with the goal of either learning about or optimizing the performance of an unknown system. Gaussian processes (GPs), which will be used to model the responses of an injured spinal cord to electrostimulation, are introduced in Section 2.4. Covariance functions, which are at the heart of specifying a meaningful and problem-tailored GP model, are described in Section 2.5.

2.1 Spinal Cord Injury

Spinal cord injury (SCI) is a medical condition with broad impact. A 2009 survey of 33,000 households estimated that 0.40% of the population of the United States is living with spinal cord injury of some sort (Christopher and Dana Reeve Foundation, 2009). In general, these injuries tend to affect a population which is in the prime of life; the same survey found that over 50% of respondents who indicated they were living with an SCI were under 50 years old and more than 75% of were under 60 years old.

Spinal cord injury has been widely studied in both human patients (e.g., Harkema et al., 2011) and animal models (e.g., van den Brand et al., 2012). Several good reviews exist, including those by Thuret et al. (2006), Edgerton et al. (2006), and Gerasimenko et al. (2008), as well chapter 14 of the text by Watson et al. (2008). Damage resulting from the injury includes the primary damage, e.g., crushing and hemorrhage, directly caused by the insult itself, and secondary damage, including cell destruction and degeneration, which results from the remaining tissue's responses to the primary damage (Watson et al., 2008, p. 209). A field of considerable interest is the acute mitigation of this secondary damage (Zhang et al., 2013).

The primary, long-term result of this damage is a substantial loss of function in terms of motor

control and sensation, resulting in impaired mobility and independence. While the symptoms of some patients improve over the first one to one and one-half years after injury, these improvements eventually cease (see Fawcett et al., 2007). The remaining deficits in sensory and motor function are at present generally considered to be largely irreversible, i.e., there is no cure for SCI, though a number of approaches have been developed (e.g., locomotor therapy, see Wernig and Müller, 1992) which have produced gains for some patients. Interestingly, in incomplete injuries and within the general realm of motor control, the degree of recovery in the performance of individual motor tasks may be somewhat independent; this may be due to different levels of supraspinal control exercised in different motor functions, e.g., locomotion versus reaching (Courtine et al., 2005). Beyond loss of motor control and bodily sensation, a number of other problems commonly arise for SCI patients, particularly issues resulting from lack of exercise and from the disruption of the nervous system's internal communications. These deficits can include muscle atrophy and spasticity, as well as potentially life-threatening autonomic problems such as failures of temperature regulation and autonomic dysreflexia. For a discussion of the many and varied autonomic deficits which result from SCI, the symposium proceedings edited by Weaver and Polosa (2006) are an excellent resource. Advances in care have meant that SCI patients who do not die immediately tend to survive for a many years (see John F. Ditunno and Formal, 1994, which contrasts early and late 20th century prognoses for SCI patients), such that therapies which partially alleviate some of their symptoms are highly desirable. Anderson (2004) surveyed 681 SCI patients and found that, among the seven options presented on the survey instrument, a near-majority of quadriplegics believed that recovery of hand and arm function would produce the greatest improvement in their quality of life, while a plurality of paraplegics believed that recovery of sexual function would most greatly improve the quality of theirs. A very substantial number of both populations ranked the item comprised of bladder, bowel, and autonomic dysreflexia as one of their top two potential greatest gains in quality of life. Among paraplegics, walking movement, described by the survey's creator as inclusive of standing and other forms of exercise, also ranked highly, but its share of first or second votes was much lower among quadraplegics. Even given the limitations of the survey, it is striking that bladder, bowel, and autonomic dysreflexia concerns were so important, particularly as compared with walking and mobility.

Current and experimental therapies have begun to deliver this desirable alleviation of secondary symptoms; for example, the initial patient in the epidural electrostimulation studies conducted by our collaborators (Harkema et al., 2011), whose therapy program includes epidural electrostimulation, locomotor training, and stand training, reports that his gains have included improved mental wellbeing, improved bladder and bowel function, some improvement in sensory function, some improvement in sexual function, substantial gains in muscle mass, including gains in the legs, core, and upper body, and better postural control. Note that this patient had already participated in

an intensive locomotor training program, and that these gains are relative to his condition after that program. Additionally, this patient has recovered some gross voluntary motor control of his lower limbs; it has been suggested that this control is a result of use-dependent plasticity of spared supraspinal axonal projections (Courtine et al., 2011).

2.2 Existing Therapeutic Approaches

Since no cure currently exists for SCI, current practice focuses on therapy, applied in a variety of approaches. Some techniques attempt to directly create the pattern of muscle activation in the extremities which would ordinarily be associated with the desired activity. Other approaches focus on the spinal cord; Bradbury and McMahon (2006) describe these as attempting either to induce regeneration of damaged axons, repairing the damage to some extent, or to rehabilitate the spinal cord's ability to control the body without addressing the injury itself. A review of a number of approaches is presented in the following sub-sections, including epidural electrostimulation, the approach which is the particular focus of this dissertation.

2.2.1 Functional Electrical Stimulation

Functional Electrical Stimulation (FES, Liberson et al., 1961) attempts to treat the symptoms of paralysis via direct stimulation of the muscles themselves; in FES, electrical stimulators are placed on or within the skeletal muscles and are then activated in a pattern engineered to replicate a desired activity, e.g., the stride cycle. The pattern of muscle activation is directly controlled, such that FES can be used to treat foot drop in hemiplegic patients (Liberson et al., 1961), to aid sit-to-stand transition in paraplegic patients (Kralj and Grobelnik, 1973), or even to produce weight-bearing locomotion in paraplegics (Klose et al., 1997). Applied as an exercise therapy, FES has been shown to confer gains in a variety of cardiorespiratory and metabolic metrics (Davis et al., 2008). However, since FES is an open-loop control method, the stimulation pattern must be carefully designed and/or user-controlled if complex behaviors are desired (e.g., in hand control, as examined by Mangold et al., 2004). Further, the resulting muscle contractions do not respond directly to sensory feedback, an important consideration when considering activities which require significant feedback control, e.g., standing. Another important problem with FES is rapid fatigue; muscle contraction force typically decreases rapidly under FES (see Thrasher et al., 2005, for a discussion of fatigue and the difficulties in mitigating it).

2.2.2 Regenerative Therapies

Another major approach to SCI rehabilitation has been to attempt to induce the spinal cord to repair itself via the introduction of signaling molecules and/or the suppression of endogenous signaling

(Karimi-Abdolrezaee et al., 2012), or to introduce cells, exogenous or autologous, into the injury site which would promote or support regrowth (Coumans et al., 2001; Wu et al., 2012). Often, scaffolds are constructed from biomaterials and used to support regrowth by providing a stable and permissive environment (Pêgo et al., 2012). Regenerative therapies seem promising in the long-term, but have not to date met with substantial success in patients with complete SCI (Thuret et al., 2006). However, there is evidence that, in the case of some incomplete injuries, and even without therapeutic aid beyond exercise, the central nervous system can reroute connections through existing neurons which bypass the injury site to make some small functional gains (Bareyre et al., 2004; Courtine et al., 2008). If these gains can be further improved, they may provide the basis for substantial recovery in the future.

2.2.3 Cord-Rehabilitative Approaches

In contrast to FES and regenerative therapies, cord-rehabilitative approaches do not seek to directly drive the muscles or repair the spinal cord; rather, these approaches take a middle road and attempt to modify the function of the spinal cord in order to produce the desired motor behavior. This avenue of SCI therapy attempts to take advantage of the surviving spinal cord circuitry below the site of injury, which remains viable and adaptable (Edgerton et al., 2006). Specific targets include interneuron networks responsible for reflexes and the central pattern generator (CPG), the region of the spinal cord responsible for generating the overall pattern of muscle activation in walking (see, e.g., Dimitrijevic et al., 1998).

Methods of this type may attempt to use any of a variety of approaches to promote lower spinal cord activity. Pharmaceutical replacement of or substitution for neurotransmitters which would normally be delivered from the higher CNS has been shown to produce substantial gains in stepping performance (Antri et al., 2002). If made practicable by an incomplete motor injury or some other therapeutic approach, physical training is also very useful for recovering function, as it provides the task-appropriate input to which the spinal cord is being trained to respond appropriately (Wernig and Müller, 1992; Engesser-Cesar et al., 2007), which may induce plastic reorganization of the lower spinal cord. In order to reduce the need for human assistance of the patient during activity-based therapy, a number of efforts have concentrated on robotic locomotor training. Cai et al. (2006) considered how the controller which drives a robotic assistance system can affect the therapeutic outcome, showing that an assist-as-needed paradigm which enforced some interlimb coordination outperformed both rote training of the nominal trajectory and an assist-as-needed controller which did not enforce interlimb coordination. Emken et al. (2008) addressed a similar question of robotic gait training and appropriate control algorithms in humans. The phenomenon of learned helplessness, i.e., non-responsiveness to stimuli which cannot be avoided, is present in the rat spinal cord and can be manipulated by both pharmacology and linkage of lower limb position with

noxious stimulus (Crown and Grau, 2001). The results of Cai et al. (2006) may provide evidence that variability in the training paradigm is important for avoiding this outcome.

2.2.3.1 Epidural Electrostimulation

Another important cord-therapeutic technique for SCI therapy, and the focus of the applied portions of this dissertation, is epidural electrostimulation. While originally developed for chronic pain therapy (Shealy et al., 1967a,b), spinal electrostimulation can produce complex motor patterns (Dimitrijevic et al., 1998). A variety of methods for delivering the electrical stimulus have been suggested, including penetrating microelectrodes (the major topic of a review by Prochazka et al., 2001). In the animal studies described in Chapter 4, electrostimulation is applied using a non-penetrating electrode array implanted in the epidural space, between the outer membrane of the spinal cord, the dura, and the interior walls of the vertebrae (see Figure 1.2). Epidural spinal cord stimulation has been applied with similar results in spinal and decerebrate cats, spinalized rats, and humans, and when properly configured, can produce walking motions (described in the review by Gerasimenko et al., 2008). Herman et al. (2002) combined epidural electrostimulation with partial body weight support exercise training to produce substantial perceived, functional, and metabolic gains in locomotion in an incomplete quadraplegic patient. More recently, Harkema et al. (2011) have used epidural electrostimulation with training and demonstrated substantial gains in a motor-complete patient in a similar setting. This type of stimulation is believed to activate afferent fibers as they enter the spinal cord through the dorsal nerve roots (Minassian et al., 2007).

2.2.4 Combined Approaches

It is often the case that the therapeutic approaches outlined above can be combined for improved effects. For example, Courtine et al. (2009) examine serotonin agonists and electrostimulation for excitation of the spinal cord in treadmill walking and van den Brand et al. (2012) use electrostimulation, pharmacology, and a compliant robotic assist device, both in SCI rats. Both works show impressive functional gains, with the latter (in animals with two staggered lesions, producing a clinically complete injury, but sparing some bridiging tissue) showing a restoration of voluntary locomotion. As pointed out by Edgerton et al. (2006), it remains an open question as to how to optimally combine individual, disparate therapies into a therapeutic program. In this dissertation, the studies in rats, described in Chapter 4, use both electrostimulation and physical training.

2.3 Active Learning and Bandits

This section provides an overview of active learning and bandit algorithms, slanted toward the work in this dissertation. Intuitively, a learner which asks useful questions should be able to learn more

information and use fewer observations doing so than a learner which waits for informative data to arrive by chance. For a view of the traditional field of active learning, i.e., the field of actively querying algorithms which do not obtain reward or suffer regret, the interested reader may refer to the text by Settles (2012). Since this work combines ideas from bandits, Bayesian optimization, and batch selection, a brief review of the literature in each of these areas is included. Much of the material on these three topics is drawn from our recently submitted paper[1], which also provides the material in Chapter 3. Attempts to deal with the problem of time variation as faced by active learning systems are discussed in Section 2.3.4. A review of the literature on control algorithms and learning agents in biological applications follows in Section 2.3.5.

2.3.1 Bandit Algorithms

Many problems have a repeating game structure, in which there exist a set of alternatives and the agent must choose one among these at each round. The agent then receives the reward corresponding to this action. Crucially, only this (possibly noisy) reward is observed, while rewards corresponding to other actions are unrevealed; this suggests that, in order to obtain a good amount of reward, the agent must use a strategy which exploits knowledge of the reward function to obtain high reward, and explores the reward function thoroughly enough to be assured that the action which is apparently best is, in fact, the one which yields the highest reward. The balance between these competing imperatives is referred to as the exploration-exploitation tradeoff. The most crucial division among algorithms in the bandit class is in regard to the types of structural assumptions made about the reward function, i.e., whether the payoffs corresponding to individual actions are somehow related to one another, or if they are totally independent.

2.3.1.1 Classical Setting

Exploration-exploitation tradeoffs have been classically studied in context of the multi-armed bandit problem, in which, from among some finite set of candidate actions, a single action is chosen at each round, and the corresponding (possibly noisy) reward is observed. A recent monograph by Bubeck and Cesa-Bianchi (2012) describes a number of related bandit problems and several algorithms for solving each. Briefly, early work has focused on the case of a finite number of decisions and payoffs that are independent across the arms (Robbins, 1952). In this setting, under some strong assumptions, optimal policies can be computed (Gittins, 1979). Due to the difficulties inherent in doing so, however, a number of heuristic policies have been created. Optimistic allocation of actions according to upper-confidence bounds (UCB) on the payoffs has proven to be particularly effective (Auer et al., 2002). One important feature contributing to the success of UCB algorithms is their

[1] Desautels, Krause, and Burdick, 2013, "Parallelizing Exploration-Exploitation Tradeoffs in Bayesian Global Optimization."

explicit description of the posterior uncertainty about the reward and their direct incorporation of this uncertainty into the decision rule.

2.3.1.2 Making Large Problems Tractable: Structural Assumptions

Recently, approaches for coping with large (or infinite) sets of decisions have been developed. In these cases, since the number of candidate actions is very large compared to the number of actions to be allocated, the reward function cannot be adequately learned if the payoffs are independent. In order to achieve some level of tractability, the dependence between the payoffs associated with different candidate actions must be modeled and exploited. Examples include bandits with linear (Dani et al., 2008; Abernethy et al., 2008) or Lipschitz-continous payoffs (Kleinberg et al., 2008), or bandits on trees (Kocsis and Szepesvári, 2006; Bubeck et al., 2008). Chapter 3 pursues a Bayesian approach to bandits, where fine-grained assumptions on the regularity of the reward function can be imposed through proper choice of the prior distribution (Srinivas et al., 2010).

2.3.2 Bayesian Optimization

The exploration-exploitation tradeoff has also been studied in Bayesian global optimization and response surface modeling, where Gaussian process models are often used due to their flexibility in incorporating prior assumptions about the structure of the payoff function (Brochu et al., 2009). Several bandit-like heuristics, such as Maximum Expected Improvement (Jones et al., 1998), Maximum Probability of Improvement (Mockus, 1989), Knowledge Gradient (Ryzhov et al., 2012), and upper-confidence based methods (Cox and John, 1997), have been developed to balance exploration with exploitation and have been successfully applied in learning problems (e.g., Lizotte et al., 2007). In contrast, the Entropy Search algorithm of Hennig and Schuler (2012) considers the estimate of the location of the optimum at any given time and tries to take the action which will greedily decrease future losses, a less bandit-like and more optimization-focused heuristic. Recently, Srinivas et al. (2010) analyzed GP-UCB, an upper-confidence bound sampling based algorithm for this setting, and proved bounds on its cumulative regret, and thus convergence rates for Bayesian global optimization. This work builds on this foundation and generalizes it to the parallel setting.

2.3.3 Parallel Selection

To enable parallel selection, one must account for the delay between decisions and observations. Most existing approaches that can deal with such delay result in a multiplicative increase in the cumulative regret as the delay grows. Only recently, Dudik et al. (2011) demonstrated that it is possible to obtain regret bounds that only increase *additively* with the delay (i.e., the penalty becomes negligible for large numbers of decisions). However, the approach of Dudik et al. only applies to contextual

bandit problems with finite decision sets, and thus not to settings with complex (even nonparametric) payoff functions. In contrast, there has been heuristic work in parallel Bayesian global optimization using GPs, e.g., by Ginsbourger et al. (2010). The state of the art is the *simulation matching* algorithm of Azimi et al. (2010), which uses the posterior of the payoff function at the beginning of the batch to simulate observations that the sequential algorithm would encounter if it could receive feedback during the batch, creating some number of Monte Carlo samples from this posterior over future sequential algorithm behavior. These Monte Carlo samples are then aggregated into a batch of observations which is intended to "closely match" the set of actions which would be taken by the sequential algorithm if it had been run with sequential feedback. To our knowledge, no theoretical results regarding the regret of this algorithm exist. This dissertation compares with this approach in experimental settings in Section 3.6. Azimi et al. (2012b) also propose a very different algorithm which adaptively choses the level of parallelism it will allow. This done in a manner which depends on the expected prediction error between the posterior given the simulated observations in the batch in progress versus the true posterior which would be available assuming the observations had actually been obtained.

Adaptive batch size selection in active learning has also been a topic of some recent interest, e.g., by Chakraborty et al. (2011), who propose a method for selecting batches of key frames from video for classification. Azimi et al. (2012a) recently extended the simulation matching construction to the batch classification setting. Section 4.3 of the text by Settles (2012) discusses a number of methods for batch active learning in non-reward settings.

2.3.4 Active Learning in the Face of Time Variation

Allowing the reward function to vary with time creates some substantial problems for the active learning agent, as uncertainty cannot be said to monotonically decrease in this setting, and thus actions must be allocated to manage this increase in uncertainty, while still doing well with respect to a performance metric. In the bandit setting, Hazan and Kale (2009) develop an algorithm for solving a time-varying bandit problem with bounded total variation, given that the reward is linear over the actions, and bound its regret in terms of this bound on variation. Another bandit algorithm, contextual zooming (Slivkins, 2011), takes advantage of Lipschitz continuity on the space of actions and contexts to bound local variation in the reward. This algorithm then adaptively partitions the space, according to its own actions, such that higher resolution partitions are generated in regions of frequent context arrival and high action frequency. This is particularly advantageous when the contexts (e.g., time) arrive in some relatively benign fashion (e.g., in sequence). Slivkins (2011) is thus able to obtain bounds on the regret in a number of drifting or stochastically changing reward settings, including in cases of spatial constraints on this drift. The problem of how to minimize regret in the case of a dynamic, but structured reward function remains open, however.

2.3.5 Learning Systems and Control Algorithms in Biological Contexts

Some attempts to use algorithmic methods for managing the interaction of therapeutic systems with complex biological systems have been made in the past. Santaniello et al. (2011), for example, modeled the responses of simulated neurons in the ventral intermediate nucleus of the thalamus to deep brain electrical stimulation as a parametric dynamic model, with coefficients fitted online; they then controlled the application of the stimulator to attempt to disrupt tremor-like activity in this population of simulated cells. Another application of interest has been brain-computer interfaces (BCIs). Traditionally, BCIs use fairly simple decoding algorithms, which classify neural activity by comparison to pre-computed, possibly stereotyped patterns corresponding to putative volitional states. An interesting example in SCI rats is presented by Manohar et al. (2011). Before SCI, the animals are first trained to press a button with their hindlimbs on-cue and secondly trained to produce similar activity in the hindlimb motor cortex. After SCI, the reward is again triggered by decoded cortical activity, indicating that this cortical activity is robust to the changes in the motor cortex resulting from SCI. More sophisticated methods for choosing the code-book have been recently employed; Fruitet et al. (2012) developed and tested (Fruitet et al., 2013) in humans a bandit-based algorithm to create personalized BCIs. This algorithm adaptively chooses which action to ask the user to imagine performing, in order to eventually produce good discrimination between this neurological state and the resting state, thus creating a classifier for the state of a volitional "button-press" manifested in the patient's sensorimotor rhythms. While intended to ultimately work with much larger sets of imagined motor actions, these experiments used a menu of three to five possible actions. Gürel and Mehring (2012) use what is essentially an ϵ-greedy bandit as a meta-algorithm for online, continuing calibration of a BCI decoding process, following an initial supervised training stage. Vidaurre et al. (2011) employed a multi-phase calibration of such a system, including the user's immediate feedback responses to the online-decoded intention, in a study with 11 human volunteers. In contrast, the present work in SCI therapy uses more structure over the space of actions (over which the reward function is modeled as a Gaussian process) in order to enable the use of a very large decision set. Further, the goal of the GP-UCB-based algorithms described in this dissertation is the optimization of a reward function *during the course of the experiment*, somewhat different than the optimization of *terminal* classifier performance. This is dependent on (eventually) measuring the reward given by each action, and so the reward must be measurable online. In the BCI setting, this is not possible in the unsupervised phase. Finally, while the unsupervised phase of some BCI algorithms incorporates tracking of the features, the implementation of GP-BUCB in Chapter 4 explicitly models the time-variation of the response function and chooses actions to yield high reward.

2.4 Gaussian Processes

Gaussian processes (GPs) are a flexible model for capturing knowledge about functions from a variety of classes. Rasmussen and Williams (2006) provide an excellent introduction to GPs.[2] In this section and Section 2.5, I present a brief review of relevant parts of GP theory and practice.

Rasmussen and Williams (2006) define a Gaussian process as follows:

Definition 1. *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

Another way of thinking of a Gaussian processes is to describe it as a probability distribution over functions mapping from an arbitrary (possibly continuous) index set $\mathcal{S}$ to $\mathbb{R}$. To denote that such a function $f : \mathcal{S} \to \mathbb{R}$ is drawn from a GP over functions on $\mathcal{S}$, one may write

$$f \sim \mathcal{GP}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')), \tag{2.1}$$

where $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{S}$, $\mu(\boldsymbol{x})$ is the *mean function* and $k(\boldsymbol{x}, \boldsymbol{x}')$ is the *covariance function*. Any element $\boldsymbol{x} \in \mathcal{S}$ corresponds to the identity of a random variable in Definition 1, and the value of any function drawn from the GP at $\boldsymbol{x}$, $f(\boldsymbol{x})$, corresponds to a particular assignment of a value to the random variable identified by $\boldsymbol{x}$. Note that here and in the remainder of the text, we use the notation f to denote a function over $\mathcal{S}$ and $f(\cdot)$ to denote the value of that function at a finite collection of elements in $\mathcal{S}$. A GP is fully specified by the mean function and covariance function; for any collection of elements of the GP, these may be used to define the Gaussian joint distribution over the values of those random variables. For example, on any collection of $n \in \mathbb{N}^+$ elements of $\mathcal{S}$, where this collection of points is described as a column vector, $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t]^T$, the Gaussian joint distribution over this column vector of corresponding values of f is

$$f(\boldsymbol{X}) = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_t)]^T \sim \mathcal{N}(\mu(\boldsymbol{X}), K(\boldsymbol{X}, \boldsymbol{X})), \tag{2.2}$$

where $\mu(\boldsymbol{X})$ is the column vector of values of the mean function and $K(\boldsymbol{X}, \boldsymbol{X})$ is the covariance matrix, and where the entries of $K(\boldsymbol{X}, \boldsymbol{X})$ are

$$[K(\boldsymbol{X}, \boldsymbol{X})]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j), \forall\, i, j \leq t.$$

In particular, for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{S}$, the covariance of $f(\boldsymbol{x})$ and $f(\boldsymbol{x}')$ is $k(\boldsymbol{x}, \boldsymbol{x}')$, hence the designation "covariance function." Importantly, the existence of a covariance function automatically grants a consistency property, that is, the distribution of any subcollection of random variables from the GP

[2] This text, along with accessible software for using GPs in MATLAB, are provided by the authors at `www.gaussianprocess.org/gpml/`

is identical, whether that subcollection is considered on its own or as part of a larger collection of variables. In much of the analysis in Chapter 3, this dissertation will assume without loss of generality that the mean function $\mu(\boldsymbol{x})$ is zero everywhere in $\mathcal{S}$; as can be seen by an examination of Equations (2.3) and (2.6) below, it is mathematically equivalent to perform regression on deviations from $\mu(\boldsymbol{x})$, expressed as $f(\boldsymbol{x}) - \mu(\boldsymbol{x})$, rather than on the actual value of the function $f(\boldsymbol{x})$, and thus the corresponding change of definitions is preferred for simplicity of presentation and calculation. The key object which defines the structure of the model is then the covariance function, $k(\boldsymbol{x}, \boldsymbol{x}')$, examined in more detail in Section 2.5.

2.4.1 Regression Using Gaussian Processes

In this dissertation, it is of greatest interest to use the GP model to make predictions about $f(\boldsymbol{x}^*)$, the value of a function drawn from the GP at a test point $\boldsymbol{x}^*$, given some finite set of observations $\mathbf{y}$ corresponding to the set $\boldsymbol{X}$. Assuming i.i.d. Gaussian noise on these observations with noise variance σ_n^2, and denoting the size of $\boldsymbol{X}$ as t, the individual observations corresponding to $\boldsymbol{x}_i \in \boldsymbol{X}$ may be written as

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i, \forall i \in \{1, \ldots, t\}, \tag{2.3}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$, $\forall i \in \{1, \ldots, t\}$. The joint distribution over the observations $\mathbf{y} = [y_1, \ldots, y_t]^T$ and $f(\boldsymbol{x}^*)$ is thus

$$\begin{bmatrix} \mathbf{y} \\ f(\boldsymbol{x}^*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\boldsymbol{X}) \\ \mu(\boldsymbol{x}^*) \end{bmatrix}, \begin{bmatrix} K(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 I & K(\boldsymbol{X}, \boldsymbol{x}^*) \\ K(\boldsymbol{x}^*, \boldsymbol{X}) & k(\boldsymbol{x}^*, \boldsymbol{x}^*) \end{bmatrix} \right). \tag{2.4}$$

Since these variables are jointly Gaussian, the posterior distribution over $f(\boldsymbol{x}^*)$ conditioned on $\mathbf{y}$ is

$$f(\boldsymbol{x}*) \mid \mathbf{y} \sim \mathcal{N}(\mu_t(\boldsymbol{x}^*), \sigma_t^2(\boldsymbol{x}^*)), \tag{2.5}$$

where the posterior mean and variance are denoted $\mu_t(\boldsymbol{x}^*)$ and $\sigma_t^2(\boldsymbol{x}^*)$ and respectively have the forms

$$\mu_t(\boldsymbol{x}^*) = \mu(\boldsymbol{x}^*) + K(\boldsymbol{x}^*, \boldsymbol{X})(K(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 I)^{-1}(\mathbf{y} - \mu(\boldsymbol{X})) \tag{2.6}$$

$$\sigma_t^2(\boldsymbol{x}^*) = k(\boldsymbol{x}^*, \boldsymbol{x}^*) - K(\boldsymbol{x}^*, \boldsymbol{X})(K(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 I)^{-1} K(\boldsymbol{X}, \boldsymbol{x}^*). \tag{2.7}$$

These forms represent the uncertainty over which function from the Gaussian process explains the observations, and capture the marginalization over all functions which could be drawn from the GP; this implicit marginalization is a manifestation of the consistency property. Notice that for $\sigma_n^2 \neq 0$, the matrix inversion used to calculate both the posterior mean and variance is possible even if there are some repeated observations in $\boldsymbol{X}$, such that $K(\boldsymbol{X}, \boldsymbol{X})$ is singular.

Of crucial importance for the algorithms described in Chapter 3 is the observation that while Equation (2.6) is dependent on the observations $\mathbf{y}$, Equation (2.7) is not. This second fact allows an algorithm to use knowledge of how uncertain it *will be* after receiving the observations which are known to be pending. This in turn enables the assembly of batches of experiments which are constructed for simultaneous execution and observation, yet are only redundant to a controlled degree.

2.5 Covariance Functions

The previous discussion has assumed the availability of a covariance function $k(\cdot, \cdot)$, but covariance functions are themselves a topic of significant interest. It should be noted that the terms "kernel function" and "covariance function" are often used interchangeably in the context of GPs; apart from the formal definitions in this section, this dissertation also uses them somewhat flexibly, and usages of "kernel" or "kernel function" should be understood to mean a valid covariance function. In applications, the choice of covariance function is a major opportunity to specify the structure of the problem, as expert knowledge can be used to choose a covariance function which encodes a great deal of problem-specific knowledge. Such choices result in relatively stronger or weaker links between the values of f at various pairs of elements $\boldsymbol{x}, \boldsymbol{x}'$ from within the chosen domain $\mathcal{S}$ of the covariance function (which is thus the domain of functions drawn from the corresponding GP). Further, some regions of $\mathcal{S}$ could be specified to have larger variances than others, encoding the knowledge that some particular region of the space is known to produce more variable behavior. Similarly, in $\mathbb{R}^d$, a covariance function could be constructed to produce draws from the GP which vary more slowly in certain directions than others; this can be very useful, e.g., if the system being modeled is known to be relatively insensitive to one of the variables describing the location in $\mathbb{R}^d$, whereas it is more sensitive to others.

As covariance functions are a crucial topic in understanding GPs, Rasmussen and Williams (2006) also provide a thorough description of this topic. A brief distillation of the relevant details is presented here.

A real *kernel function* k is a function which maps pairs of elements of $\mathcal{S}$ into $\mathbb{R}$, i.e., $k : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$. A kernel which is symmetric in its arguments, i.e., $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}', \boldsymbol{x})$, is referred to as a *symmetric kernel*. In Euclidean spaces, the properties of stationarity and isotropy are of interest. If k is solely a function of $\boldsymbol{x} - \boldsymbol{x}'$, k is *stationary*, and it is invariant to translations of the inputs. If k is a function of only the (vector) magnitude of this difference, $|\boldsymbol{x} - \boldsymbol{x}'|$, it is *isotropic*.

Covariance functions are a sub-class of symmetric kernel functions. For any kernel function k

and collection $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ of n elements of $\mathcal{S}$, the *Gram matrix* is $K(\boldsymbol{X}, \boldsymbol{X})$, where

$$[K(\boldsymbol{X}, \boldsymbol{X})]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j), \forall i, j \leq n.$$

If for a symmetric kernel k, the Gram matrix is positive semi-definite (i.e., given any vector $v \in \mathbb{R}^n$, $v^T K(\boldsymbol{X}, \boldsymbol{X}) v \geq 0$) for all $n \in \mathbb{N}^+$, k is termed *positive semi-definite*. If $v^T K(\boldsymbol{X}, \boldsymbol{X}) v = 0$ if and only if $v = 0$, the kernel is termed *positive definite*. If a kernel k is symmetric and positive semi-definite, k is also a valid *covariance function*, and any Gram matrix corresponding to k is referred to as a *covariance matrix*. The covariance function in essence defines similarity between the values of $f(\boldsymbol{x})$ and $f(\boldsymbol{x}')$ for any two elements $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{S}$, and does so by reference to $\boldsymbol{x}$ and $\boldsymbol{x}'$, rather than the function itself. This is useful for a variety of reasons:

- For a finite collection of points $D \in \mathcal{S}$, the covariance matrix $K(D, D)$ of the jointly Gaussian values of f at the elements of D can be precomputed and, conditioned on (noisy, and possibly repeated) observations of elements of D, the posterior over f at D can be computed using this matrix. This fact is useful for the bandit algorithms considered in Chapter 3.

- Complex representations of $\boldsymbol{x}$ and $\boldsymbol{x}'$ in feature spaces, even infinite-dimensional feature spaces, can be encoded by using a kernel (covariance) function which operates (typically very simply) on $\boldsymbol{x}$ and $\boldsymbol{x}'$; this is commonly known in machine learning as the "kernel trick," and is employed to leverage a simple technique into a much more complex and expressive suite of techniques while incurring very little computational expense. In this sense, Gaussian process regression is actually Bayesian linear regression, extended via the kernel trick.

- Perhaps most importantly from a practical perspective, careful choices of the representation of the inputs $\boldsymbol{x} \in \mathcal{S}$, the kernel function, and the corresponding hyperparameters can allow expert knowledge to be encoded into the GP model very simply and intuitively. As an expert works with a system, they might plausibly acquire some intuition of which variables are functionally important and which are less so. It might be that there are many ways to describe the objects in $\mathcal{S}$, but some may be more convenient or meaningful than others; in essence, this is an extension of the kernel trick. The choice of covariance function also is an opportunity for expert intuition to be expressed; linear or squared-exponential kernels imply quite different things about the functions drawn from the corresponding GPs. Similarly, the choice of the hyperparameters of selected kernel function encodes information like the relative sensitivity of the responses to variation in any of the chosen features.

As an example of the last point, in Chapter 4, a variant of the GP-BUCB algorithm is applied to choosing electrical stimuli $\boldsymbol{x}$ for SCI therapy from a set D, using a GP model of the responses. There are discretely many pairs of electrodes which can be chosen to stimulate the spinal cord;

these could be labeled essentially arbitrarily, e.g., by the integers, but there is not an obvious way of doing so without destroying the useful information given by knowing which electrodes are which. The kernel functions in Chapter 4 are chosen such that each element $\boldsymbol{x} \in D$ is represented in terms of the physical locations of the corresponding pair of epidural electrodes; since function within the spinal cord is spatially organized, this spatial representation is an intuitive way to capture the functional similarity of the evoked potentials which will result. The choice of stationary covariance functions over this Euclidian parameterization of the stimulus space also encodes information, in particular, that the uncertainty about how strongly the spinal cord and muscles will respond is the same everywhere, and that this response strength is expected to be of a particular sensitivity to the same movement of the cathode or anode, regardless of where the electrodes are on the cord. Finally, the hyperparameters of the kernel function are chosen to quantify the greater or lesser spatial sensitivity of the evoked potential to the rostro-caudal or lateral movements of each of the anode and cathode. As will be demonstrated in Chapter 4, this series of choices allows the GP model to express meaningful information about the spinal cord and thus gives the algorithm the ability to make intelligent choices about which stimuli to apply.

2.5.1 Reproducing Kernel Hilbert Spaces

A *reproducing kernel Hilbert space* (RKHS) is a Hilbert space of functions over a set $\mathcal{S}$ associated with a particular kernel function. More precisely, an RKHS is defined by Rasmussen and Williams (2006) as follows:

Definition 2. *Let $\mathcal{H}$ be a Hilbert space of real functions f defined on an index set $\mathcal{S}$. Then $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS) endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ (and norm $||f||_{\mathcal{H}} = \sqrt{\langle f, f \rangle_{\mathcal{H}}}$) if there exists a function $k : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ with the following properties:*

- *for every $\boldsymbol{x}$, $k(\boldsymbol{x}, \boldsymbol{x}')$ as a function of $\boldsymbol{x}'$ belongs to $\mathcal{H}$, and*
- *k has the* reproducing property *$\langle f(\cdot), k(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}} = f(\boldsymbol{x})$.*

For any kernel function, there exists a unique RKHS, and for each RKHS, there exists a unique corresponding kernel function (Aronszajn, 1950, Part I, Section 2)

The RKHS can also be viewed as the set of functions

$$\left\{ f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i) \ : \ n \in \mathbb{N},\, \boldsymbol{x}_i \in \mathcal{S},\, \alpha_i \in \mathbb{R} \right\}, \tag{2.8}$$

with the associated inner product

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^{n} \sum_{j=1}^{n'} \alpha_i \alpha'_j k(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{2.9}$$

where $g(\boldsymbol{x}) = \sum_{j=1}^{n'} \alpha'_j k(\boldsymbol{x}, \boldsymbol{x}'_j)$. Note that, with the assumption that $\mu(\boldsymbol{x}) = 0, \forall\, \boldsymbol{x} \in D$ and the definition α_i as the ith entry of the column vector $(K(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 I)^{-1}\mathbf{y}$, where $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, the predictive mean of a GP model, given by Equation (2.6), can be written as a linear combination of kernel evaluations; this is precisely the form of Equation (2.8), and so the RKHS may be regarded as the space of possible posterior means for any collection of data, given that GP regression is conducted with a particular kernel. This is known as the *reproducing kernel map* perspective.

For the purposes of this dissertation, it is most important to note that the RKHS norm of a function f provides a measure of how closely f matches the possible posterior means which would be constructed from a GP model using the corresponding kernel and a finite amount of data, i.e., how well f can be captured by the model; a small value for $||f||_{\mathcal{H}}$ implies that f is much like a linear combination of relatively few copies of the kernel function, whereas a large or infinite value for $||f||_{\mathcal{H}}$ implies that this is not the case. Alternatively, a finite value of $||f||_{\mathcal{H}}$ could be viewed as the rapid decay of the eigenvalues of f with respect to an eigenfunction basis of the RKHS. This use of the RKHS norm to quantify the complexity of a function with respect to a GP model underlies the performance guarantee given by the third case of Theorem 1 in Chapter 3; one expects that it should be easier to make decisions about a function which is well-captured by the model (has a low RKHS norm) than one which is only poorly captured by the model.

For a somewhat different treatment than the above (again, drawn from Rasmussen and Williams, 2006), see Wahba (1990), Sections 1.1 and 1.4.

2.5.2 Stationary Covariance Functions on $\mathbb{R}^d$

A number of stationary functions on $\mathbb{R}^d$ will be used in the remainder of this work.

The first of these is the Squared Exponential covariance function, which may be written as

$$k_{SE}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp(-1/2 \cdot (\boldsymbol{x} - \boldsymbol{x}')^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{x}')), \tag{2.10}$$

where Σ is a symmetric, positive definite matrix in $\mathbb{R}^{d \times d}$. Often, Σ is chosen as $\Sigma = \frac{1}{2} l^2 I$, such that the covariance function is isotropic and can be rewritten as

$$k_{SEiso}(r) = \sigma^2 \exp\left(\frac{-r^2}{l^2}\right), \tag{2.11}$$

where $r = |\boldsymbol{x} - \boldsymbol{x}'|$. Another common choice for Σ is to have a separate lengthscale $l_i > 0$ for each dimension, such that

$$\Sigma_{i,j} = \begin{cases} 0 & i \neq j \\ l_i^2 & i = j \end{cases}, \tag{2.12}$$

such that samples from the GP tend to be rougher in dimensions with small values of l_i and larger

in dimensions with large values of l_i. A GP with any version of the squared exponential covariance function is infinitely many times mean square differentiable (i.e, very smooth; see Rasmussen and Williams, 2006, Sections 4.1.1 & 4.2.1 for details).

Another class of covariance functions of interest is the Matérn class. Following Rasmussen and Williams (2006), assuming isotropy, and thus $r = |\boldsymbol{x} - \boldsymbol{x}'|$, the general form of the Matérn covariance is

$$k_{Matérn}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{l} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu} r}{l} \right), \tag{2.13}$$

where ν and l are positive parameters and K_{ν} is a modified Bessel function. The parameter ν describes the smoothness of samples from the resulting GP; for a given ν, the resulting process is k times mean square differentiable if and only if $\nu > k$ (again, see Rasmussen and Williams, 2006, for details). In the limit as $\nu \to \infty$, the isotropic squared exponential covariance function is recovered from Equation (2.13). From a practical perspective, ν gives a useful means of expressing expert knowledge about the system being modeled and its characteristics, such that it can be chosen to give an appropriate degree of smoothness; small values of ν result in predicted functions which are extremely jagged, whereas larger values give smoother predicted functions. Certain values for ν are commonly used for GP regression, specifically $\nu = 1/2$, $3/2$, and $5/2$; these three values yield a practically useful range of graduated roughness. Again using the isotropic form, these values of ν yield a simplified form of the covariance function consisting of a polynomial in r multiplied by an exponential in r (as do any others such that $\nu = p + 1/2$, where $p \in \mathbb{N}$):

$$k_{\nu=1/2}(r) = \exp\left(-\frac{r}{l}\right), \tag{2.14}$$

$$k_{\nu=3/2}(r) = \left(1 + \frac{\sqrt{3} r}{l}\right) \exp\left(-\frac{\sqrt{3} r}{l}\right), \tag{2.15}$$

$$k_{\nu=5/2}(r) = \left(1 + \frac{\sqrt{5} r}{l} + \frac{5r^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5} r}{l}\right). \tag{2.16}$$

The covariance function in the first case, for which $\nu = 1/2$, is also known as the exponential covariance function; the resulting GP is also known as the Ornstein-Uhlenbeck process (Uhlenbeck and Ornstein, 1930, Section IV). As with other isotropic covariance functions on $\mathbb{R}^d$, it is possible to use a positive definite lengthscale matrix Σ and substitute $\sqrt{(\boldsymbol{x} - \boldsymbol{x}')^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{x}')}$ for r, yielding anisotropic covariances of some desired form. As mentioned above, the parameter ν has a relationship to the mean square differentiability of the resulting GP, but for $\nu = p + 1/2$, $p \in \mathbb{N}$, this relationship can be put on firmer footing. It can be shown that, with particular choices of the necessary parameters, the covariance function corresponding to a special case of the steady state of a continuous-time, one-dimensional, auto-regressive process of order p with a Gaussian disruption, i.e., an AR(p) process, is the Matérn covariance with $\nu = p + 1/2$ (Rasmussen and Williams, 2006,

Appendix B). Thus, the order 1 AR(p) process corresponds to a special case of the exponential covariance function ($\nu = 1/2$) and order 2 corresponds to a special case of the Matérn covariance with $\nu = 3/2$. Using this fact, the choice of a Matérn covariance for $\nu = 3/2$ can be loosely considered to correspond to modeling the system in question as a second-order linear dynamical system in $\boldsymbol{x}$, such that the "position" and "velocity" are states of the system, but no higher order derivatives. This might also be thought of as specifying the degree of "memory" the system retains about past states.

2.5.3 Non-stationary Covariance Functions on $\mathbb{R}^d$

A non-stationary covariance function on $\mathbb{R}^d$ is one which, while holding the difference between the inputs $\boldsymbol{x} - \boldsymbol{x}'$ constant, changes value with translation of $\boldsymbol{x}, \boldsymbol{x}'$ within $\mathbb{R}^d$. While many others exist, one simple non-stationary covariance function for $\mathcal{S} \in \mathbb{R}^d$ is the linear covariance function, $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x} \cdot \boldsymbol{x}' + \sigma_0^2$, that is, the dot product of the vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ with some offset σ_0^2. If this covariance function is used, Gaussian process regression on a collection of data reduces to Bayesian linear regression. By changing from a simple dot product to a product using some positive semi-definite weight matrix Σ_p, one may produce a general linear covariance function, which corresponds to a dot product under a transformation of $\mathbb{R}^d$ by scaling and rotation. Note that scaling $\boldsymbol{x}$ by a constant c produces an increase of $k(\boldsymbol{x}, \boldsymbol{x}')$ by the same factor; linear covariance functions make sense for cases where variability can be expected to increase with distance from the origin or where $\mathcal{S}$ lies entirely on a hypersphere centered on the origin (after any transformation of coordinates implicit in Σ_p), e.g., $|\boldsymbol{x}| = 1, \forall \boldsymbol{x} \in \mathcal{S}$ if $\Sigma_p = I$. Further, since the covariance matrix K corresponding to any finite collection $\boldsymbol{X}$ in $\mathcal{S}$ of size n consists of the matrix of ones, multiplied by σ_0^2, summed with a matrix composed of the product of two $n \times d$ matrices containing the coordinates of these individual locations in $\mathbb{R}^d$, K is of rank no more than $d + 1$. This is consistent with the statement above that a dot product covariance function corresponds to Bayesian linear regression, since the model corresponds to a hyperplane in $\mathbb{R}^{d+1}$.

2.5.4 Constructing Covariance Functions

It is also possible to construct more complicated covariance functions by using compositions of simpler covariance functions. In particular, the sum of two covariance functions is a covariance function, and a sample from the GP corresponding to this covariance function corresponds to a sum of independent samples from the two GPs which correspond to the two original covariance functions. Similarly, the product of two covariance functions is also a covariance function, such that draws from the GP corresponding to the product covariance function can be thought of as being the product of two independent draws from the GPs corresponding to the individual factor covariance functions.

Finally, two covariance functions $k_1(\boldsymbol{x}_1, \boldsymbol{x}_1')$ and $k_2(\boldsymbol{x}_2, \boldsymbol{x}_2')$ over different spaces $\mathcal{S}_1$ and $\mathcal{S}_2$ may be combined as either a sum $k(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}_1, \boldsymbol{x}_1') + k_2(\boldsymbol{x}_2, \boldsymbol{x}_2')$ or product $k(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}_1, \boldsymbol{x}_1') \cdot k_2(\boldsymbol{x}_2, \boldsymbol{x}_2')$ to form a covariance function k for $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{S}_1 \times \mathcal{S}_2$ via the sum and product methods above. This allows the construction of covariance functions from individual covariance functions over subspaces, e.g., different dimensions of $\mathbb{R}^d$, or even radically different sets; $\mathcal{S}_1$ might be $\mathbb{R}$, whereas $\mathcal{S}_2$ could be nodes in a graph, words in a corpus, or something more exotic.

While it is possible to construct covariance functions which natively represent covariance over a space $\mathcal{S}$ which is not a subset of $\mathbb{R}^d$, another way to construct covariance functions for such $\mathcal{S}$ is to find a mapping $g : \mathcal{S} \to \mathbb{R}^d$ and then use a covariance function $\tilde{k} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ to construct a covariance function, such that $k(\boldsymbol{x}, \boldsymbol{x}') = \tilde{k}(g(\boldsymbol{x}), g(\boldsymbol{x}'))$.

The combination of all of these techniques allows a great deal of flexibility in terms of modeling assumptions; some of this flexibility is used in Chapter 4 and further possibilities are discussed in Chapter 5.