Automatic Gait Optimization with Gaussian Process Regression

Daniel Lizotte, Tao Wang, Michael Bowling, Dale Schuurmans
Department of Computing Science
University of Alberta
{dlizotte,trysi,bowling,dale}@cs.ualberta.ca

Abstract

Gait optimization is a basic yet challenging problem for both quadrupedal and bipedal robots. Although techniques for automating the process exist, most involve local function optimization procedures that suffer from three key drawbacks. Local optimization techniques are naturally plagued by local optima, make no use of the expensive gait evaluations once a local step is taken, and do not explicitly model noise in gait evaluation. These drawbacks increase the need for a large number of gait evaluations, making optimization slow, data inefficient, and manually intensive. We present a Bayesian approach based on Gaussian process regression that addresses all three drawbacks. It uses a global search strategy based on a posterior model inferred from all of the individual noisy evaluations. We demonstrate the technique on a quadruped robot, using it to optimize two different criteria: speed and smoothness. We show in both cases our technique requires dramatically fewer gait evaluations than state-of-the-art local gradient approaches.

1 Introduction

Legged robot platforms offer many advantages over traditional wheeled robots. In addition to their ability to traverse a wide variety of terrain, walking robots are basically a requirement for performing useful tasks in our human-centric world. Despite these advantages, walking is also one of the fundamental challenges for legged robots.

Optimizing a robot's gait is not a simple control problem. An open loop gait consists of a sequence of joint values for an already high degree of freedom system of leg joints. Simplified parametric representations of leg trajectories can result in a manageable number of parameters. However, these parameters will likely have complicated interactions making manual tuning of gait parameters time-consuming for simple robots and nearly impossible for the increasing complexity of humanoid platforms.

Even worse, no single gait will be effective in all circumstances. The walking surface is critical and can vary in terms of friction, softness, and height variation (e.g., compare concrete, linoleum, carpet, and grass). Robot platforms themselves also vary due to manufacturing imperfections and general joint and motor wear. Lastly, even the criterion for determining an effective gait is likely to be situation specific. Although velocity may seem like the obvious choice, relative stability of the robot's sensor hardware can also be important. Although tailoring a robot's gait to the environmental, robot, and task specific circumstances would be ideal, constant manual retuning is impractical. Automatic gait optimization is an attractive alternative to this laborious manual process.

Although walk learning is not a new idea, existing techniques are based on the common approach of local function optimization. Hence, existing algorithms also share key drawbacks including local minima and inefficient use of gait evaluations. These drawbacks combine to increase the number of gait evaluations required to find effective parameters. In this work, we propose a different approach based on Gaussian process regression. Not only does the approach not suffer drawbacks inherent in local methods, but it has the additional advantages of providing confidence estimates on the gait's performance, which is useful for exploration, and allows for a natural inclusion of prior knowledge. We demonstrate the effectiveness of the approach on the Sony AIBO quadruped robot. We show that under two separate criteria the Gaussian process regression not only finds effective gait parameters, but also does so with an order of magnitude fewer evaluations than a local gradient competitor.

We begin in Section 2 by examining some of the recent approaches to gait optimization. We note the common drawbacks in these approaches as motivation for our proposed technique. In Section 3 we present relevant background material on Gaussian process regression. We then, in Section 4 describe how we apply Gaussian processes to the problem of gait optimization. In Section 5 we show results of our technique applied to two different gait optimization scenarios, demonstrating dramatically more efficient learning than both simple and state-of-the-art gait optimization techniques. Finally, we discuss some future extensions of our approach.

2 Motivation

The Sony AIBO, a commercially available quadruped robot, has spurred recent interest in gait optimization. Its use in robot soccer within the RoboCup Legged League gives an immediate application where gait velocity and uncertain competition conditions reward well tuned walks. Since the AIBO is our evaluation platform as well, we will examine a number of recent approaches pioneered on this robot.

A common foundation for all of these approaches, including ours as well, is the notion of a "walk engine", or parameterized gait generation system. Since the number of degrees of freedom in legged robots is large, optimizing the angle of each joint at a finely discretized time scale would involve searching over a thousand parameters. The walk engine reduces the number of parameters by focusing on leg trajectories that are both physically possible and intuitively plausible. These parameters usually define properties of each leg's trajectory such as the "distance the foot is lifted off the ground" and "the period of the walk". The space of parameterized walks obviously has a large impact on the final quality of any automatic gait optimization, but the optimization problem itself is the same regardless of the walk engine. Optimization in all cases requires finding a point in parameter space, ranging from eight to fifty parameters, that results in an effective gait. All of the cited work below are based on different walk engines. Hence, because of the variation in walk engine, surfaces, and robots themselves, reported walk speeds are largely incomparable.[1]

A second common feature of the approaches is the experimental setup. All of the approaches involve evaluating specific parameter settings by having the AIBO walk in a structured arena using the parameters to be evaluated. Local sensor readings on the AIBO–either camera images of visual landmarks or the IR sensor readings of walls–are then used to compute a noisy estimate of the gait's average speed. The procedure may be replicated and each estimate averaged to compute a more accurate evaluation.

2.1 Evolutionary Approaches

An evolutionary approach was the first proposed method for gait optimization on the AIBO. Hornby et al. [1999; 2000] used a fairly standard evolutionary search on an early prototype AIBO. A population of parameters were maintained at each evolutionary generation. A new population was formed through mutation and crossover of individuals from the previous generation, replacing parameters in the population that evaluated poorly. Their procedure showed slow gait improvement over 500 generations, requiring approximately twenty-five robot-hours of learning.

The evolutionary approach was revisited by Chernova and Veloso [Chernova and Veloso, 2004]. They used similar mutation and crossover operations to generate candidate gaits. Unlike the work of Hornby, their parametric space of walks included a measurement of the possibility that the AIBO could physically perform the gait. This allowed them to throw out poor gait parameters without requiring an empirical evaluation. In addition, they used a "radiation" procedure to disperse clusters of similar parameters in the population, forcing further search. They demonstrated that the technique learned competitive walks using only 4,000 evaluations and a total running time of approximately five hours distributed across four robots for a total of twenty robot-hours.

As evaluations are noisy, both approaches must deal with the possibility that an inaccurate evaluation will cause poor gait parameters to incorrectly remain in the population. Both used targeted reevaluation to reduce this possibility, reevaluating either the parameters that remained for multiple generations or the ones that performed disproportionately well.

2.2 Function Optimization

The second family of approaches that has been explored involves adapting techniques for multidimensional function optimization to the gait optimization problem. Kim and Uther [2003] used Powell's method [Press *et al.*, 1992], which performs line search along a chosen search direction based on the effectiveness of previously chosen search directions. Kohl and Stone [2004] used a hill climbing approach. Although the gradient is not known, a set of random perturbations are evaluated empirically, and these evaluations are used to approximate the gradient. The parameters are then adjusted by a fixed step size in the direction of this estimated gradient. Kohl and Stone reported the fastest learning result in the literature, requiring only three hours distributed across three robots for a total of nine robot-hours. It is important to note that the reported experiments for both techniques involved initializing the optimization with a known set of reasonable parameters. This differs from evolutionary approaches, which begin with a random initial population.

2.3 Drawbacks

All of the previous approaches share three key drawbacks. First, they can get stuck at local optima. Kim and Uther reported actual experiences of local optima and Kohl and Stone noted the importance of starting from good initial parameters, having found considerably poorer performance under different starting conditions. There are techniques to deal with local optima, such as random restarts for local function optimization approaches and radiation for evolutionary approaches. However, both involve a considerable increase in the required number of gait evaluations. Furthermore, the approaches forget previously evaluated gaits once they either die out of the population or after the gradient step is taken. Not only is this an inefficient use of expensive gait evaluations, but certain parameter settings may be unnecessarily reevaluated. Finally, none of the approaches explicitly model the noise in the evaluation process. Hence they all involve long evaluations or even repeated evaluations that are averaged to compute a less noisy estimate. In summary, if the goal is to reduce the total number of robot-hours these drawbacks severely hurt the usefulness of the approaches. Our approach addresses these disadvantages and consequently requires considerably fewer gait evaluations.

3 Background

At the core of our proposed method is a Gaussian process regression model that describes our knowledge about the function we are optimizing. Before discussing its use for gait opti-

[1]Despite the lack of basis for comparison, the three most recent techniques discussed below all achieve a similar walk speed in the range of 0.27–0.29 m/s.

mization we give a brief overview of Gaussian processes and Gaussian process regression.

3.1 Gaussian Processes

A *Gaussian process* (GP) [Williams, 1999; Rasmussen, 2004; Rasmussen and Williams, 2006] is a collection $\mathcal{F}$ of random variables $F_{x_1}, F_{x_2}, \ldots$ for which any finite subset of the variables has a joint multivariate Gaussian distribution. The variables are indexed by elements x of a set $\mathcal{X}$. For any finite length vector of indices $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$, we have a corresponding vector $\boldsymbol{F_x} = [F_{x_1}, F_{x_2}, \ldots, F_{x_n}]^T$ of variables that has a multivariate Gaussian (or *normal*) distribution,

$$\boldsymbol{F_x} \sim \mathcal{N}\left\{\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x})\right\}, \tag{1}$$

where the elements of $\mu(\boldsymbol{x})$ are given by a prior mean function $\mu(x_i)$, and k is the *kernel* function. The kernel takes two indices x_i and x_j, and gives the covariance between their corresponding variables F_{x_i} and F_{x_j}. Given vectors of indices $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, k returns the matrix of covariances between all pairs of variables where the first in the pair comes from $F_{\boldsymbol{x}_i}$ and the second from $F_{\boldsymbol{x}_j}$. Note that each F_{x_i} is marginally Gaussian, with mean $\mu(x_i)$ and variance $k(x_i, x_i)$.

3.2 Gaussian Process Regression

Suppose we have a function $f(x)$ that we would like to optimize. Further, suppose that we cannot observe f directly, but that we can observe a random variable F_x that is indexed by the same domain as f and whose *expected value* is f, i.e., $\forall x \in \mathcal{X},\ E[F_x] = f(x)$. In particular, we assume that our prior belief about the function f conforms to a Gaussian process with prior mean μ and kernel k, as described above. Suppose that F_x is an observation of $f(x)$ that has been corrupted by zero-mean, i.i.d. Gaussian noise, i.e., $F_x = f(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Hence, $f(x)$ is a hidden variable whose posterior distribution we can infer after observing samples of F_x at various locations in the domain. The resulting inference is called Gaussian process regression.

Let $\boldsymbol{x}$ be the set of observations points and $\boldsymbol{F_x}$ be the resulting real-valued observations. We want to compute the posterior distribution of some new point $\hat{x} \in \mathcal{X}$. The distribution will be Gaussian with mean and variance,

$$\mu(\hat{x}|\boldsymbol{x}) = \mu(\hat{x}) + k(\hat{x}, \boldsymbol{x})k(\boldsymbol{x}, \boldsymbol{x})^{-1}\left(\boldsymbol{F_x} - \mu(\boldsymbol{x})\right) \tag{2}$$

$$\sigma^2(\hat{x}|\boldsymbol{x}) = k(\hat{x}, \hat{x}) - k(\hat{x}, \boldsymbol{x})k(\boldsymbol{x}, \boldsymbol{x})^{-1}k(\boldsymbol{x}, \hat{x}). \tag{3}$$

Note that the inverse applies to the kernel matrix of observed domain points, and so can be computed once and used to evaluate the posterior at many points in the domain.

Since the problem is to find an optimum of the unknown function, the final step is to compute the optimum of the resulting posterior mean $x^R = \mathrm{argmax}_{\hat{x} \in \mathcal{X}}\, \mu(\hat{x}|\boldsymbol{x})$. This, in general, cannot be computed in closed form, and requires the application of some technique for function optimization. Although not optimal, another technique is to simply return the x_i from $\boldsymbol{x}$ with the largest observed value F_{x_i}. This has the practical side-effect that the technique will not return a point that has never been evaluated, adding some protection from an incorrect prior.

3.3 Observation Selection

Given a Gaussian process model and a set of observations we have shown how to incorporate the observations to create the function posterior and then find the expected optimum. The remaining challenge, then, is to decide for which points x_i to receive observations. This is, in fact, a sequential decision making task, as each observation gives additional information that can be used when selecting future points.

Optimal Selection. Assume we have a fixed horizon, i.e., a limited number of remaining observations, the optimal action given our model can theoretically be computed. This requires considering all possible points in the domain, all possible real-valued outcomes of this function evaluation, and then taking expectations and maximizations for all possible future observation sequences out to the the horizon. We then make the next observation at the point which maximizes the *eventual* posterior maximum assuming future optimal decisions. This enumeration for exploration is discussed in our previous work [Wang *et al.*, 2005], but in domains with significant size, it is impractical. Instead, we will focus on myopic search strategies.

Most Probable Improvement. Consider our proposal to return the domain point with the largest observed value. If we want to maximize the largest observed value, one approach would be to search for points that are likely to be better than our current best observation. Let F_{x^+} be the maximum observed value. Choose x^{mpi} to maximize the posterior probability $P(F_{x^{\mathrm{mpi}}} > F_{x^+})$. We call this the *most probable improvement* (MPI) criterion.

Computing the posterior probability of improvement at a point $\hat{x}$ in the domain is a simple computation since the posterior distribution of $F_{\hat{x}}$ is Gaussian as given in Equations 2 and 3. Finding the point x^{mpi} then requires a search through the domain to maximize the probability of improvement. This is equivalent to maximizing,

$$\phi(\hat{x}) = \sigma(\hat{x}|\boldsymbol{x}) / (F_{x^+} - \mu(\hat{x}|\boldsymbol{x})). \tag{4}$$

This criterion for deciding where to evaluate next was suggested by Mockus [1989]. Fortunately, the posterior mean and standard deviation in this ratio are continuous and differentiable functions, so we can apply function optimization techniques *in the model* to find x^{mpi}.

4 Gait Optimization

Our proposed approach to gait optimization is basically the application of the above described Gaussian process optimization procedure. Like previous approaches, we assume that we already have some parameterized walk engine with k parameters. In addition, we can empirically take noisy evaluations of a gait's velocity (or other feature of the gait). Unlike previous approaches, we model the stochastic velocity function, $f : \mathbb{R}^k \rightarrow \mathbb{R}$, which maps walk parameters to velocity, as a Gaussian process. The exact mean and kernel function of the Gaussian process are chosen based on prior domain knowledge. We use the most probable improvement selection rule to decide which parameters to empirically evaluate based on the previous observations. After some number of gait evaluations, the parameters that generated the fastest observed walk are returned.

In place of choosing mutation and crossover rates or initial parameters and step sizes, our approach requires the specification of the mean and kernel of the Gaussian process. This provides a very natural way to encode domain knowledge, if it exists, or one can just use "generic" priors. We first describe our walk engine and then detail exactly how priors were chosen for our AIBO experiments.

4.1 Walk Engine

As discussed earlier, a complete joint trajectory for a gait could have thousands of parameters. All gait optimization techniques parameterize this walk space using a walk engine. For this work, we have used Carnegie Mellon University's Tekkotsu[2] software to control the AIBO, which includes a walk parameterization (as of release 2.4.1) originating from the Carnegie Mellon CMPack 2002 robot soccer team. This is an early version of the CMWalk engine used in the work by Chernova et al. [2004]. In consultation with a domain expert, we identified 15 walk parameters along with reasonable bounds on each parameter to define our domain, i.e., $X \subset \mathbb{R}^{15}$.

4.2 Priors

Defining the prior for the Gaussian process means defining a mean function $\mu : X \rightarrow \mathbb{R}$ and kernel function $k : X \times X \rightarrow \mathbb{R}$. We also need to specify the variance σ_ϵ^2 of the noise that is believed to be added to the observation F_x. The priors that we use have a very simple form. For the mean, we use a constant function $\mu(x) = \mu_f$, which means we have no a priori belief about any specific parameters' velocities. For the covariance function, it is hypothesized that, in general, parameter vectors that are close in terms of Euclidean distance are likely to have similar walk velocities, and therefore a large positive covariance. Furthermore, we expect that some parameters may have wide-reaching consequences, and so parameters far apart should still have some small positive correlation. We therefore chose to use the radial basis function kernel,

$$k(x_i, x_j) = \sigma_f^2 \cdot \mathrm{e}^{-\frac{1}{2}(x_i - x_j)^T S (x_i - x_j)}, \tag{5}$$

where S is a scaling matrix that has along its diagonal the inverse of the range of each dimension, i.e., the distance from one end to the other of one dimension's range is scaled to equal one. We now need to simply choose the constants σ_ϵ^2, μ_f, and σ_f^2. Setting these to appropriate values depends on the feature of the gait to be optimized. In Section 5, we present results both for optimizing the gait's velocity and its smoothness. So we examine each of these cases in turn.

Prior for Velocity. Gait velocity has been studied relatively extensively both on the AIBO and with our particular walk engine. In consulting with our domain expert, we easily solicited useful prior information. In particular, we chose $\mu_f = 0.15$ and $\sigma_f^2 = 0.066$, which correspond to the intuition, "For some random gait parameters, we expect the observed velocity to be 0.15 meters per second and within about 0.2 meters per second 99% of the time." For observation noise, our domain expert was not familiar with our particular experimental setup. Instead, we computed the sample variance of a small number of observations of one particular setting of the walk parameters. This gave us a value of $\sigma_\epsilon^2 = 0.01$, which seemed to work well in practice.

As an interesting test of the stability of our method with respect to the model parameters, we also ran the velocity optimization with a prior variance parameter of $\sigma_f^2 = 0.66$ (ten times larger).

Prior for Smoothness. Since smoothness had not been evaluated before, we had no domain expert to consult. Instead, we took a small number of samples (about 30) and used sample means and variances to estimate the parameters of the prior $\mu_f = -30$, $\sigma_f^2 = 100$, $\sigma_\epsilon^2 = 2.25$. As we will show in the next section, even this simple uninformed method for specifying the Gaussian process model can be very effective.

4.3 Implementation Details

In order to compute the most probable improvement point as described in Section 3.3, we used the generic constrained hill climber in MATLAB (`fmincon`) supplied with the function and gradient of $\phi(\hat{x})$ from Equation 4. We used as default starting points the two best parameters found so far, and 13 drawn uniformly random within the bounded domain for a total of 15 starting points. In addition, we forced the first gait evaluation selection by choosing the center point of the domain. Since the Gaussian process model starts with a uniform belief over the domain, all function points are equally good to the most expected probable rule.

5 Results

We have applied our Gaussian process approach to two gait optimization tasks on the Sony AIBO ERS-7. We first look at the standard problem of maximizing walk velocity, and we also examine the problem of optimizing a gait's smoothness. The goal of any gait optimization technique is to find a near-optimal gait in as few evaluations as possible. Therefore, we'll want to compare techniques using this criterion.

Since previous gait learning has not involved the same walk engine, experimental setup, or robots, comparing directly with previously reported results can be somewhat problematic. For a more direct comparison, we have implemented the hill climbing method of Kohl and Stone [2004] as described in Section 2.2 and applied it in identical circumstances as our approach. The Kohl and Stone algorithm is the most data efficient technique for the AIBO from the literature, demonstrating effective walks with only nine robot-hours of training. We replicated their experimental setup, using 15 random evaluations to approximate the gradient at each "iteration" and using a step size of 2. The empirical epsilon used in estimating the gradient was 1/15 of the parameters' range, which seemed to be an adequate change in performance. As with the Gaussian process model, we started the hill climber from the point in the center of the space.

[2] http://www.tekkotsu.org

5.1 Physical Setup

To evaluate each parameter choice, we had the robot walk between two visual landmarks while focusing the head on the center of the landmark. The robot determined its distance from a landmark as it walked toward it based on the landmark's apparent width in the camera's field of view, and that change is used to estimate velocity. To determine a gait's smoothness, we measured the time-averaged distance from the center of the landmark to the center of the robot's field of view, and negated this. Unstable walks that result in a large amount of head movement yield negative smoothness values, since it is difficult for the robot to keep the head and the camera aimed at the target. More fluid walks allow the robot to aim the camera more directly at the target, resulting in a smoothness much nearer to zero.

Each observed measurement is the result of three "traversals" from one landmark to the other. The average time for three traversals, including time to turn around, was approximately one minute. This was chosen to be consistent with Kohl and Stone's hill climbing experiments. We could have easily only used a single traversal and compensated by increasing the observation variance used in the model.

5.2 Gait Velocity

A graph showing the result of 321 observations is shown in Figure 1(a). We chose this number of observations a priori to allow the hill climber 20 "steps" or iterations with 15 test points for each. Both our Gaussian process technique and the Kohl and Stone hill climbing technique are shown, as well as the simple baseline of choosing gaits uniformly at random. The solid lines represent the maximum achieved walk speed over the accumulating observations, and the corresponding isolated markers show the maximum velocity achieved over the most recent 15 observations.

We found that both the Gaussian process and hill climbing methods performed appreciably better than random evaluations. The best walk velocity found was 0.285 m/s, which was found by the Gaussian process with the over-estimated prior variance, $\sigma_f^2 = 0.66$, although the walk found by the Gaussian process with the "sensibly" initialized variance is nearly as fast. Despite having already warned of the difficulties in comparing walk speeds, note that this speed is comparable to other learned gaits. More impressively, though, is the fact this speed was attained after only 120 observations, which took approximately two robot-hours. This is nearly a five-fold improvement in the required number of robot-hours[3]

It is interesting that the best walks found by the two Gaussian process models are in widely separated parts of the space. The walk found by the sensible setting of $\sigma_f^2 = 0.066$ has a low period and shorter stride, with a parameter vector far from the center of the space. On the other hand, the experiment at $\sigma_f^2 = 0.66$ found a similarly fast walk near the center of the space with longer, slower strides that cover a similar distance.

Although the results of the hill climbing approach were not poor, we were somewhat surprised that the performance was not better. The method tended to take very poor steps once it reached a value of about 0.230 m/s. There are two natural explanations. One may simply be local optima, which is in line with Kohl and Stone's noted importance of the initial parameter vector. Alternatively, based on the frequency of taking poor gradient steps, the step size may have been too large along certain dimensions. Random restarts and a variable step size for each dimension could mitigate these unimpressive results.

5.3 Gait Smoothness

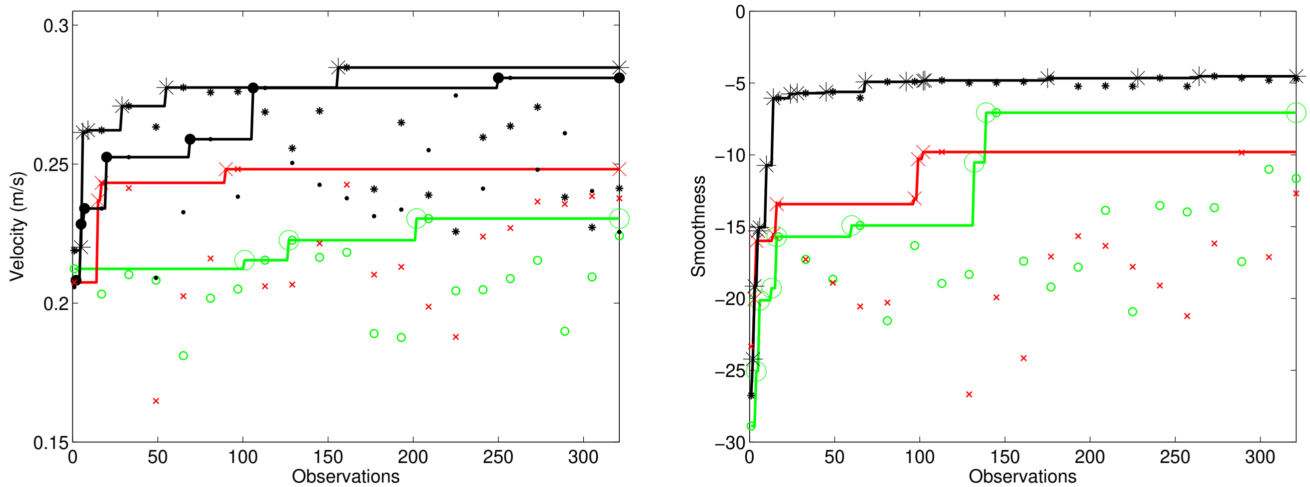
A similar graph for gait smoothness is shown in Figure 1(b). The Gaussian process optimization found the smoothest walk of all three methods and did so within the first 20 observations, or only twenty minutes of robot time. Later improvement was only incremental. In a post-mortem analysis, the gait smoothness task is apparently much simpler. The impact of parameters on our measure of smoothness ended up being quite simple as a wide range of walks with a short period (below about 310 ms) and a moderate requested walk speed (between 210 and 240 m/s) resulted in a smooth gait measurement. Scores of such walks were typically greater than -10. Due to the independence of this pair of parameters from the rest, it is not unlikely that even random search would find a smooth gait as choosing parameters in this range will occur on average every 250 gait evaluations. Our random search trial did happen to find one such point, giving it a moderate win over the hill climbing technique. Again, we suspect the unimpressive hill climbing result to be due to initial conditions and local maxima.

6 Conclusion

We proposed a new approach to gait learning based on Gaussian process optimization. The approach overcomes many drawbacks of previous techniques, effectively avoiding local optima, efficiently using all gait evaluations, and explicitly modelling noisy observations. Even with all of the caveats associated with comparing gait velocities and training time, we have demonstrated that the approach is not only effective in our high dimensional, noisy, non-convex, optimization problem, but also requires drastically fewer evaluations. This was also accomplished with a minimum of parameter tuning, demonstrating effective performance when using prior settings from a domain expert, incorrect settings, and even data derived settings.

There are three main directions we still feel should be explored from this point. First, our model is quite simple in comparison to Gaussian process models in the recent machine learning literature. Many interesting innovations involving kernel optimization, hyper-parameter models, and dimensionality reduction seem well suited to this problem and are worth investigating. It would also be compelling to build models that incorporate knowledge from previous gait optimization runs with similar robots or surfaces. Second, it would be interesting to explore how knowledge of an imposed budget on the number of samples or compute time could be

[3] Although the work of Kohl and Stone used multiple AIBOs in parallel to reduce their total time to three hours, our approach could also easily benefit by evaluating multiple sample points simultaneously.



(•) GP w/MPI	(∗) GP w/MPI	(×) H.Clmb	(○) U.Rand
0.281 m/s	0.285 m/s	0.248 m/s	0.230 m/s
$\sigma_f^2 = 0.06$	$\sigma_f^2 = 0.6$		

(a)

(∗) GP w/MPI	(×) H.Clmb	(○) U.Rand
-4.53	-9.80	-7.07

(b)

Figure 1: Results for (a) gait velocity and (b) gait smoothness. Solid lines represent cumulative maximum, and the small markers indicate the maximum observation from the last 15 observations.

efficiently incorporated into this approach. Third, because of the generality of the method, it could also easily be applied in other application areas. We are currently investigating its use in the problem of "graph-cut stereo matching" [Kolmogorov, 2004], which has a number of continuous parameters that must be tuned. Initial reports from our colleagues in vision are that Gaussian process regression with the most probable improvement rule is also working very well on this problem.

Acknowledgments

We would like to sincerely thank our domain expert Sonia Chernova, and our team of programmers: Nelson Loyola, Eric Coulthard, Wesley Loh, James Neufeld, and Sam Vincent. This research is supported by Alberta Ingenuity, iCORE, and NSERC. Daniel Lizotte is supported by the Killam Trusts.

References

[Chernova and Veloso, 2004] Sonia Chernova and Manuela Veloso. An evolutionary approach to gait learning for four-legged robots. In *Intelligent Robots and Systems*, 2004.

[Hornby *et al.*, 1999] G. S. Hornby, M. Fujita, S. Takamura, T. Yamamoto, and O. Hanagata. Autonomous evolution of gaits with the sony quadruped robot. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1297–1304, 1999.

[Hornby *et al.*, 2000] G. Hornby, S. Takamura, J. Yokono, O. Hanagata, T. Yamamoto, and M. Fujita. Evolving robust gaits with AIBO. In *IEEE International Conference on Robotics and Automation*, pages 3040–3045, 2000.

[Kim and Uther, 2003] M. S. Kim and W. Uther. Automatic gait optimisation for quadruped robots. In *Australasian Conference on Robotics and Automation*, 2003.

[Kohl and Stone, 2004] Nate Kohl and Peter Stone. Machine learning for fast quadrupedal locomotion. In *The Nineteenth National Conference on Artificial Intelligence*, pages 611–616, July 2004.

[Kolmogorov, 2004] Vladimir Kolmogorov. *Graph Based Algorithms for Scene Reconstruction From Two or More Views*. PhD thesis, Cornell University, January 2004.

[Mockus, 1989] Jonas Mockus. *Bayesian Approach to Global Optimization*. Kluwer Academic Publishers, The Netherlands, 1989.

[Press *et al.*, 1992] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1992.

[Rasmussen and Williams, 2006] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[Rasmussen, 2004] C. E. Rasmussen. *Advanced Lectures in Machine Learning: ML Summer Schools 2003*, chapter Gaussian Processes in Machine Learning. Springer-Verlag, 2004.

[Wang *et al.*, 2005] Tao Wang, Daniel Lizotte, Michael Bowling, and Dale Schuurmans. Bayesian sparse sampling for on-line reward optimization. In *ICML 2005*, Bonn, 2005.

[Williams, 1999] C. Williams. Prediction with Gaussian processes. In *Learning in Graphical Models*. MIT, 1999.