Data-driven Extraction of Drive Functions for Legged Locomotion: A Study on Cheetah-cub Robot

Mostafa Ajallooeian, Alexander Sproewitz, Alexandre Tuleu and Auke J. Ijspeert
Biorobotics Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{*first.last*}@*epfl.ch*

1 Introduction

The process of finding working gaits for legged robots always, to different extents, includes manual tuning, systematic search, or optimization of control parameters. This process populates a dataset of control parameter vectors and respective robot behavior factors including body rotations, speed, duty factor, etc. Normally, targeting actuated robots and not simulated ones, these datasets are sparse, unless systematic search with small parameter changes is applied.

The dataset obtained from a tuning process can include many gaits which share a similar performance in one behavior factor, e.g. speed, but differ in the control parameter vectors used. Our question here is, using the tuning dataset, how a continuous drive function can be calculated which takes the desired behavior, e.g. speed, and maps that to a control parameter vector[1]. If this question is answered properly, then the robot operator (or a higher level controller) will have a single control knob to continuously change the desired behavior factor. Here in this contribution we address the case were a Central Pattern Generator (CPG) [3] is used as the locomotion controller and the desired behavior factor to control is the locomotion speed.

There are model-based approaches like [4, 5, 6] which explore speed control using closed-loop control of the step length. We address the question of the speed drive function from a model-free open-loop[2] control perspective when a parameter-speed dataset is given. We do our experiments with the compliant quadruped robot Cheetah-cub (Figure 1) [7], use trot gaits, and go up to a speed of more than 4 $BodyLength/s$ which gives dynamic locomotion with a froud number $fr \approx 1$.

2 Drive function extraction

Our hypothesis is that if the desired continuous change in the behavior is small, then the change in control parameter vector should be small as well. So, using the collected tuning dataset, one should choose an array of parameter vectors such that the successive changes between them are minimal and results in a small change in the desired behavior factor

[1]This is different from experiment design approaches like Doehlerts [1, 2] where prior knowledge about the form of the drive functions and independence of control parameters is available.

[2]no sensing other than for low-level motor control.

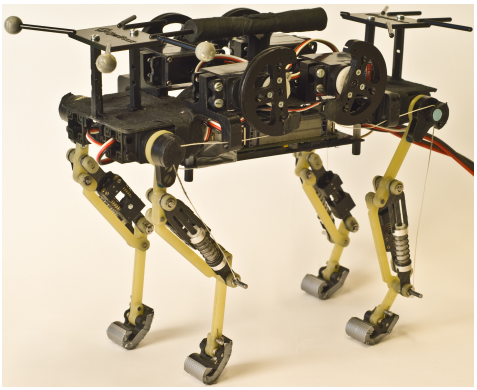


Figure 1: Cheetah-cub robot

(e.g a small increase in speed). Then function approximation tools can be used to fit continuous functions on the chosen parameter vectors.

We cluster all the control parameter vectors based on the speed that they give to the robot in equally spaced bins. So the i-th bin $\mathscr{B}_i$ contains:

$$\mathscr{B}_i = \{\mathbf{x}_k \ : \ \|v(\mathbf{x}_k) - v_i\| < \delta\} \ \forall k = 1..K \tag{1}$$

where K is the number of control parameter vectors, $\mathbf{x}_k$ is the k-th parameter vector, $v(.)$ is the obtained speed, and v_i values are the center of the bins equally space on the speed axis and δ determines the bin width. Now if one candidate is chosen from each bin, then the total length of the multi-segment line passing through all candidates is:

$$d = \sum_{i=1}^{N-1} \|\mathbf{x}_{k_i} - \mathbf{x}_{k_{i+1}}\| \tag{2}$$

Table 1: CPG parameters and the obtained drive functions

Name	Param.	Obtained drive function
Frequency	f	$-1.7v^2+4.7v+0.3$
Desired duty factor	D	$-0.18v+0.66$
Fore/hind hip amplitude	A_{H}	$12v^3-9.3v^2+48v+19$
Fore knee swing amplitude	A_{FK}	$-0.74v^3+2.3v^2-1.4v+0.71$
Hind knee swing amplitude	A_{HK}	$-0.44v^3+1.7v^2-1.2v+0.71$
Fore hip offset	O_{FH}	$1.8v^3-6.8v^2+5v+14$
Hind hip offset	O_{HH}	$4.4v^3-17v^2+12v+13$
Fore knee offset	O_{FK}	$-0.14v^2+0.12v+0.38$
Hind knee offset	O_{HK}	$-0.28v^2+0.24v+0.16$

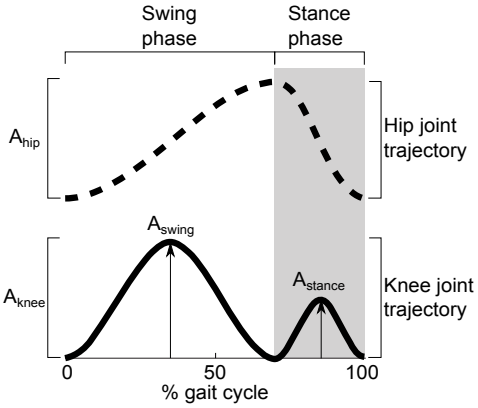


Figure 2: Parametric joint angles profiles generated by CPG.

with N being the number of bins, and $\mathbf{x}_{k_i}$ the k-th member of the i-th bin. Finally, the problem of finding the drive function knot points can be formalized as:

$$\min_{k_i,\ i=1..N} d \tag{3}$$

Solving the aforementioned problem, we obtain an array of control parameter vectors with respectively increasing robot speeds, which then should be approximated by fitting tools to obtain the drive functions.

3 Results

We implemented our approach on the Cheetah-cub robot, a small ($\sim 1Kg$) quadruped robot with compliant pantograph legs (Figure 1). In order to find working gaits for Cheetahcub, a series of manual trials was done to find working gaits. A collection of 110 control parameter vectors and respective locomotion speeds was obtained from the manual tuning process, which includes only the cases where the robot did not fall during locomotion on a flat terrain. This data collection is depicted in Figure 3 (cross markers).

The CPG model used for the control of locomotion is defined by four coupled phase oscillators generating the desired joint angles profiles for hip and knee joints, depicted in Figure 2. The hip joint angle profile is a skewed sine (skewed based on the desired duty factor), and the knee joint angle profile consists of a flexion during the swing phase (to obtain foot clearance), and an additional flexion during the stance phase to actively control the leg length. The CPG control parameters are given in Table 1.

Solving[3] the problem in equation (3) gave the selection of the knot points per bins which are depicted with bold markers in Figure 3. We then used first to third order polynomials to fit a function on these points and thus obtained the drive functions as depicted in Figure 3 and given in Table 1.

[3]Using brute force and checking all possible solutions. This takes about five minutes using MATLAB running on a quad-core PC. For bigger datasets, one can instead use discrete valued optimization techniques like Genetic Algorithm or [8].

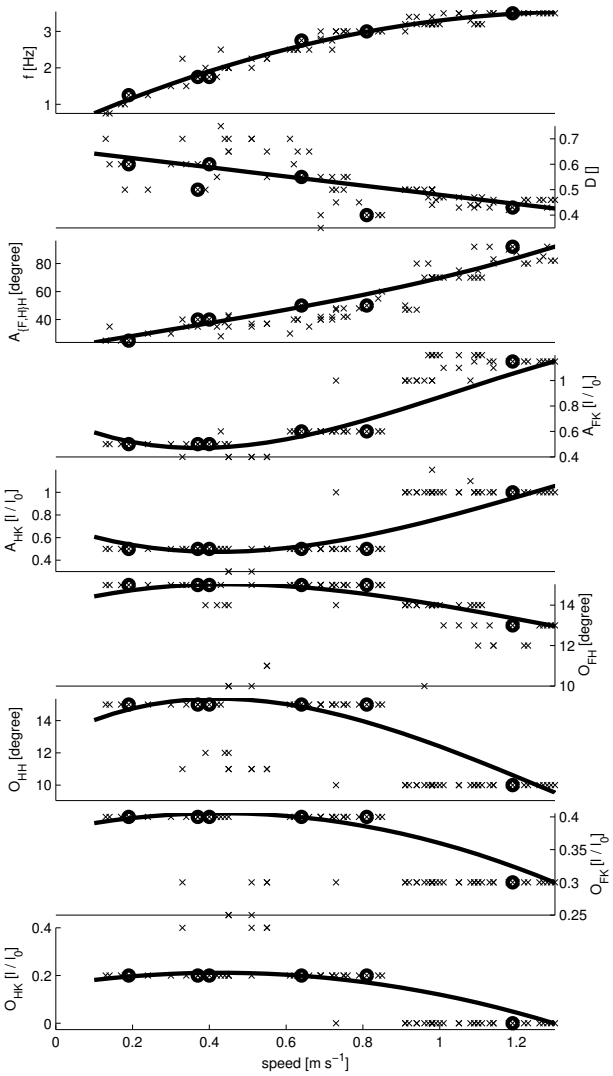
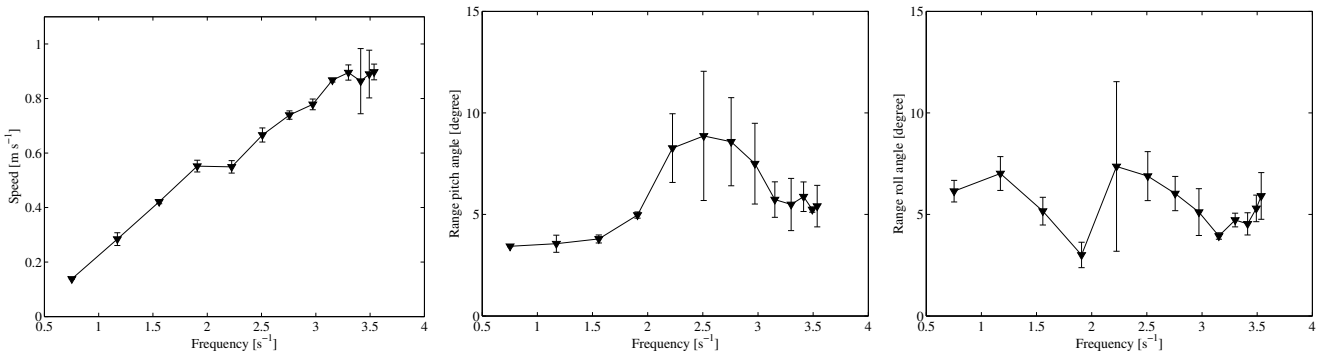


Figure 3: The tuning dataset (cross markers), selected candidates (bold markers), and the calculated drive functions. Bin parameters are $v_i = 0.1 + 0.2i,\ i = 0..6$ and $\delta = 0.1$.

We interpolated the obtained drive functions in the speed range of $v = 0.1 + 0.1i[m/s],\ i = 0..12$ and extracted the respective control parameter vectors. Then we ran the robot with these parameter vectors (5 runs for each parameter vector) and recorded the locomotion speed. Results of these evaluation runs are depicted in Figure 4. No post-processing was done on the obtained drive functions. The robot did not fall in any of these experiments, and the average pitch and roll angles were always less than 10 degrees.

4 Discussion

We proposed a simple way to extract drive functions from existing data obtained from manual tuning of a robot's locomotion controller. These drive functions act as a single control knob which give the proper control parameters for a certain desired behavior (speed in this paper). The introduced method differs from a model fitting on the whole dataset which is not the proper way to extract the drive func-



(a) Average speed versus frequency (b) Max pitch versus frequency (c) Max roll versus frequency

Figure 4: Evaluation of the obtained drive functions. The error bars show the minimum and maximum values.

tions because it will average different trails out instead of finding a correct path along them. We believe that the introduced drive function extraction method is useful in many cases where robots are going through a tuning process and a dataset of behavior versus control parameters is collected.

The introduced approach is not dependent to the meaning of the control parameters. One can use a different control strategy, like a different CPG controller, or even a model-based controller, and obtain a parameter-behavior dataset while tuning the robot's locomotion. Such dataset can then be utilized to obtain a data-driven drive function using the method introduced in this paper.

Our drive-function experiments with the cheetah-cub robot where limited to testing the obtained parameter vectors one at a time, and we have not yet explored the online change of the control parameters during locomotion. We expect a slow change of the control parameters to give a smooth change in the locomotion speed, but this has to be further tested, especially to inspect the transient behavior and balance of the robot during the parameter change.

There are a number of open questions that we aim to explore in the future: 1) Is there a common parametric representation of the drive functions for a class of similar quadrupeds? 2) Can we also extract drive functions for all control parameters including both feedforward and feedback parameters? 3) Can we use the method here to extract drive functions at a different level of control, e.g. at the muscle activation level?

Acknowledgement

The research leading to these results has received funding from the European Communitys Seventh Framework Programme FP7/2007 - 2013 - Challenge 2- Cognitive Systems, Interaction, Robotics-under grant agreement No. 248311 (AMARSi).

References

[1] D. Doehlert, "Uniform shell designs," *Applied statistics*, pp. 231–239, 1970.

[2] D. Vojnovic, D. Chicco, and H. El Zenary, "Doehlert experimental design applied to optimization and quality control of a granulation process in a high shear mixer," *International journal of pharmaceutics*, vol. 145, no. 1, pp. 203–213, 1996.

[3] A. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks*, vol. 21, no. 4, pp. 642 – 653, 2008.

[4] J. Hodgins and M. Raibert, "Adjusting step length for rough terrain locomotion," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 3, pp. 289–298, 1991.

[5] A. Kuo, "A simple model of bipedal walking predicts the preferred speed–step length relationship," *Ann Arbor*, vol. 1001, pp. 48109–2125, 2001.

[6] S. Pouya, M. Ajallooeian, and A. Ijspeert, "a closed-loop optimal control approach for online control of a planar monopod hopper," in *15th International Conference on Climbing and Walking Robots (CLAWAR 2012), USA*, 2012.

[7] A. Sproewitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. Ijspeert, "Towards dynamic trot gait locomotion – design, control, and experiments with a compliant quadruped robot," *Interational Journal of Robotics Research (IJRR)*, 2012, Submitted.

[8] J. Kennedy and R. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, vol. 5, pp. 4104–4108, IEEE, 1997.