Frequency Space Representation and Transitions of Quadruped Robot Gaits

Jan Hoffmann **Uwe Düffert**

Institut für Informatik
LFG Künstliche Intelligenz
Humboldt-Universität zu Berlin
Unter den Linden 6, D-10099 Berlin, Germany
Email: jhoffman@informatik.hu-berlin.de
dueffert@informatik.hu-berlin.de

Abstract

The locomotion of a quadruped robot was examined with the emphasis on the creation, optimization and merging of motions and gaits. A Fourier Series Expansion was performed on the controller commands used in the robot gait. It was found that omitting higher order terms has very little, at times even positive effect on the robot's performance yet yields a greatly reduced parameter set describing the motion. The parameter reduction is desirable for creating, optimizing and adapting motions. Furthermore, the acquired frequency space representation allows for simple creation of new motions by merging of existing ones and also for creating transitions from one type of motion to another.

Keywords: quadruped robot, locomotion, gait pattern representation and generation

1 Introduction

Creation and optimization of quadruped robot locomotion is a challenging, highly complex task. The main reason for this is the difficulty to cope with the many degrees of freedom of a legged robot even in a relatively simple robot design. Such designs (may it be software and/or hardware) are often inspired by animal locomotion in terms of anatomy, gait patterns (Central Pattern Generators) and other concepts such as reflexes [5, 1, 11, 13]. Evolutionary approaches include the evolution of the controller alone (with fixed robot morphology [9, 14]) and simultaneous evolution of robot morphology and controller [16, 15]. Other approaches include teaching, learning, and inverse kinematics [10].

1.1 Creating Robot Motions

Inverse kinematics is a commonly used concept to calculate the motor commands necessary for robot motions: trajectories of the robot's paws are defined and then the corresponding limb movements (and thus joint angles as a functions of time) are calculated. This type of calculation is usually based on simple geometric modeling neglecting physical properties such as friction, weight of the robot as a whole and of individual components, moments of inertia, motor strengths, etc. These simplifications in the modeling tend to impair performance in real world environments. Two examples illustrate this:

- The mass of the robot weighs down its torso onto its joints, distorting the motion which may have looked promising "in theory" and renders it useless.
- Robot motions are highly sensitive to the properties of the surface they are being performed on. Friction (or lack of it) allows for certain motions to be executed on certain types of surfaces only, whereas the result is unpredictable on other surfaces.

Experience can help to come up with experimental setups that take these pitfalls into consideration while not explicitly modeling them, e.g. by conducting experiments on especially difficult surfaces [9]. Limiting possible motions to static ones (i.e. non-dynamic), robust robot gaits can be developed. These motions are reversible at any given point in time. To achive this, three of the robot's four legs always touch the ground at any point in time [6]. These motions trade off speed for robustness. Dynamic motions can produce faster locomotion [12, 4, 8]. Gaits can be found by trying to model all physical aspects of the robot, but bridging the gap between simulation and the real world remains challenging [7, 2].

1.2 Optimizing Robot Motions

Having found a motion, the process of optimization still proves to be a difficult and tedious task due to the many degrees of freedom and interdependencies involved. Evolutionary approaches include trying to optimize the parameters of a given type of motion model [9]. Common to these methods is the fact that a certain amount of knowledge of how locomotion is performed is implicitly present in the model, e.g. symmetries and/or the shape of the limb's trajectories. This narrows down the search space thus reducing the time needed for the optimization process (the search space can be reduced further by judging parameter sets a priori by a neural network and then measuring the fitness only of those that were labeled promising [3]). On the down side, other global maxima may not be found (which is also a matter of finding an appropriate fitness function). To allow all possible motions to be modeled, the description of the robot's motion needs to be more generic. The most complete description would be to denote the angle of each joint over the time of the motion. Given that quadruped locomotion is a periodic process, describing each joint's motion over one period fully describes the motion. This periodic behavior is described easily in frequency space. This paper deals with the possibilities of representing robot gaits in frequency space and examines possibilities of creating transitions between gaits.

2 Fourier Series Expansion

Periodic motions are conveniently described in frequency space. Since the servo motors used in the robot act as low pass filters, we expected to be able to omit higher order frequencies without decreasing performance because these frequencies cannot contribute to the motion that is actually being executed by the robot. Furthermore, generating curves through Fourier Transformation yields curves that are continuous and continuously differentiable by definition, which we were hoping would result in smooth overall motions of the robot.

Instead of creating new motions from scratch, existing motions were used to bootstrap our experiments. These motions where created using inverse kinematics. The robot's paws follow trapezoidal trajectories as described in [12, 4, 8]. The original motions were recorded and controller commands were then transformed into frequency space (see below).

The effect of omitting higher order terms (which is essentially a crude low pass filtering) when resynthesizing the motions was investigated.

Since the joint angles were recorded in discrete time intervals (in our case $\Delta t = 8$ msec), Discrete Fourier Transformation was used. It is the discrete version of the continuous Fourier Transformation:

Fourier Transform (FT):

$$f(t) = \int_{-\infty}^{\infty} F(\omega) e^{\mathrm{i}\omega t} \mathrm{d}\omega$$

$$F(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(t) e^{-\mathrm{i}\omega t} \mathrm{d}t$$

Discrete Fourier Transform (DFT):

$$f(t_k) = \sum_{n=1}^{M} c_n e^{2\pi \mathrm{i}(n-1)t_k/T}$$

$$c_k = \frac{1}{N} \sum_{n=1}^{N} f(t_n) e^{-2\pi \mathrm{i}(n-1)t_k/T}$$

$F(\omega)$ is the Fourier Transform of $f(t)$, c_n is the Fourier coefficient of $f(t_k)$, continuous time t, discrete time $t_k = k\Delta t$ (Δt being the grid interval, $k \in \mathbf{N}$), frequency $\omega = 2\pi/t$, period T, number of sample points N. M denotes the number of coefficients used in the Fourier Synthesis ($2 \leq M \leq N$).

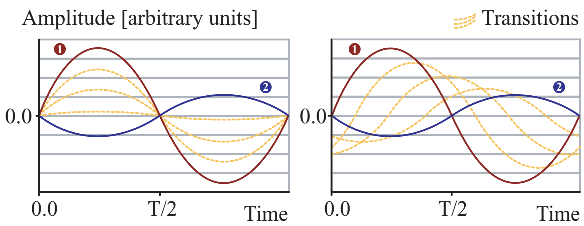

Figure 1: *Left:* Transition from one curve to the other using a weighted average of the amplitudes. *Right:* Transition generated using amplitude *and* phase information provided by the DFT.

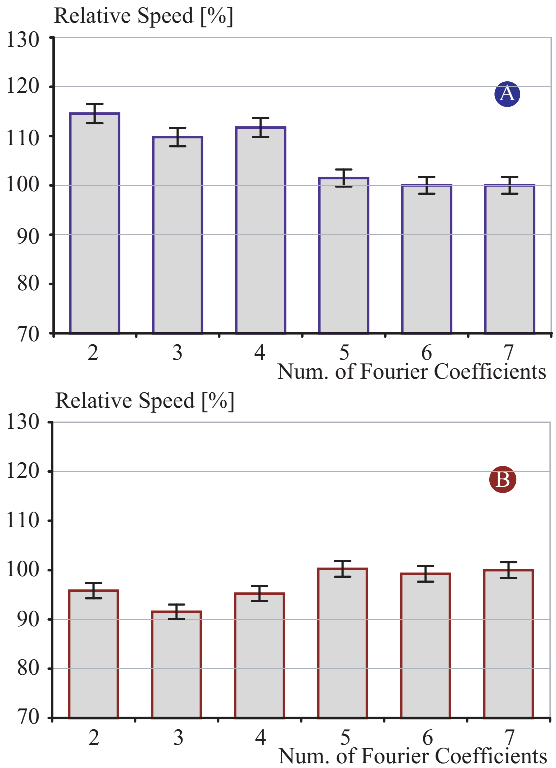

Figure 2: Relative speeds of two motions A and B that were Fourier synthesized as a function of the number of Fourier Coefficients used. In both cases, the absolute speed of the robot using the original motion was 10 cm/sec corresponding to a relative speed of 100%. This performance is also reached by Fourier synthesized motions using all available coefficients obtained by DFT. The first complex Fourier Coefficient gives only the offset of the motor command, so the minimum value to produce any kind of motion is 2 complex Fourier Coefficients. As would be expected, the speed of the approximated motions differs from the original ones for a small number of coefficients and converges toward 100% for a large number of coefficients. In the case of motion A an increase in speed was observed for the rough approximation generated using only the first 2 complex coefficients.

2.1 Performance of Fourier Synthesized Motions

The relative speed as compared to the original motion was used to measure gait performance (other possible criteria are reproducibility, robustness, steadiness and stability, surface independence, etc.). Fourier approximations were generated by DFT from existing motions. The number of terms used in the series approximation was varied. It was started using the first two complex Fourier Coefficients (the first being the constant offset with no phase information). The number of terms was then increased. It was expected that the speed would be different from the original speed for a small number of terms and would then converge towards the level of performance of the orignial motor command as the number of coefficients used was increased. Figure 2 shows such a measurement for two different gaits A and B. These motions were developed and used by teams in the 2002 Fukuoka RoboCup competition in the Sony Four Legged League, A by the Carnegie Mellon University and B by the GermanTeam. It can be seen that the relative speed differs for small M and then approaches the original speed. The difference in speed is less than 15%. Note that in the case of motion A,

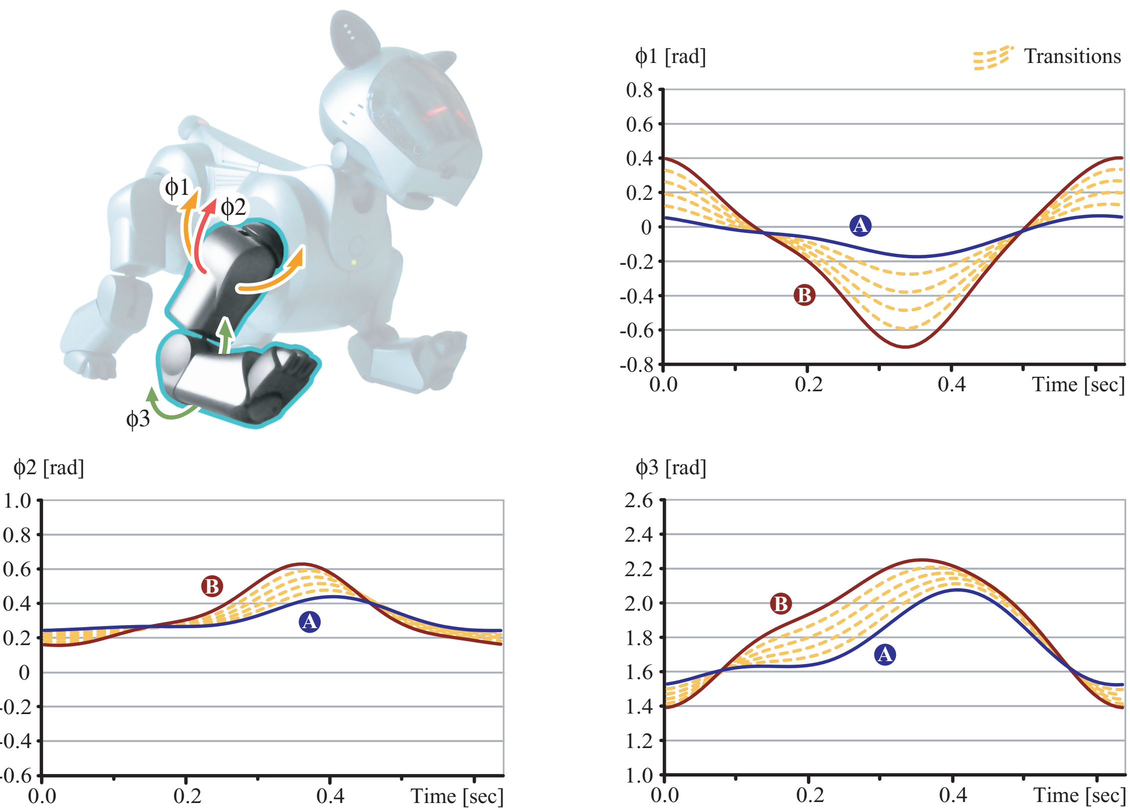

Figure 3: *Top left:* Image of Aibo with degrees of freedom superimposed for the robot's front right leg ($\phi 1, \phi 2, \phi 3$). *Diagrams:* The diagrams show the joint angles of two motions A and B as a function of time over one period T. The dashed lines show the transitional motions. They are obtained in the following way: a DFT is performed on the original motions A and B. The obtained coefficients are merged (using weighted averaging) and then the curve is synthesized from the newly generated coefficient set. (Number of Fourier Coefficients $M = 4$)

there is actually a speed increase in the Fourier synthesized motion over the original. Subjectively, the approximated gaits were as robust and steady as the original ones, also being perceived as smoother by the experimenter.

2.2 Mixing Motions, Transitions

Encouraged by the good performance, possibilities of "mixing" or merging different motions were investigated. Two goals were pursued: to create new motions from existing ones and to create intermediate motions to allow transitions from one type of motion to another.

When trying to interpolate between two motions we felt it was necessary to retain certain characteristics of the limb motion (such as frequency). This can be achieved by interpolating the Fourier Coefficients and then synthesizing the intermediate curve from these newly found coefficients. Figure 1 shows a simplified example of a transition: the result of amplitude averaging and resulting curve when averaging Fourier Coefficients (it can be argued that in some cases weighted averages may be more useful). The effect of aligning the phases of the two curves (i.e. shifting curve 2 by $T/2$) before merging is also illustrated. The phase information is readily available after spectral analysis (DFT) of the curves and was used to improve transitions.

Figure 3 shows the calculated intermediate curves used in the later experiment for each joint of the robot's left fore leg for the transition from walking forward to walking backward (note that in this case forward and backward motions are not symmetrical). Figure 4 illustrates the three transitions that were examined:

a. Transition from walking forward at a slow pace to walking forward at a faster pace using a different motion type: a speed increase can be investigated as the transition shifts towards the faster motion (as one would expect). One of the transition steps turns out to be faster than the fastest of the original motions from which it was derived.

b. Transition from backward motion to forward motion: in the middle of the transition the robot is neither moving forward nor backward. But it is not standing still either: the amplitudes of the joint motions are non-zero, but the phase shift is such that the robot itself does not move forward or backward.

c. Transition from turning counterclockwise to turning clockwise: Similar to b., there is no net locomotion of the robot in the case of a 50 : 50 mixing of the two motions.

Qualitatively speaking, the three transitions are stable and robust. The speeds of the interpolated motions are roughly proportional to the percentage of the motions they were derived from. The dependencies are non-linear, which is not surprising given the complexity of such motions.

3 Conclusion

Robot gaits can be represented in frequency space. Low pass filtering has very little effect on gait performance (in some cases even results in an increase in speed) and yields a parameter reduction. Frequency

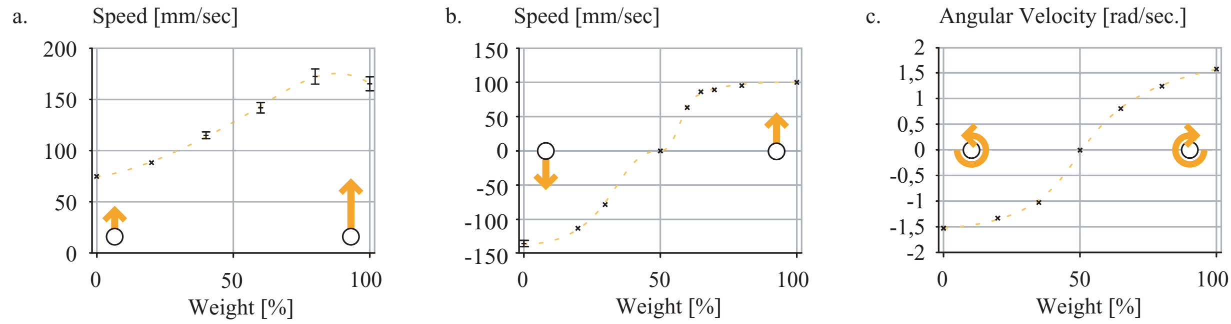

Figure 4: Three types of transitions between gait patterns were examined: *a.* From a slow forward motion to a fast forward motion *b.* From a forward to a backward motion *c.* From turning counterclockwise to turning clockwise. In all cases, the speed was proportional to the percentages of the motions that were mixed. Note that in *a.*, one of the transitional motions is faster than the two it was generated from.

space representation makes it easy to align the phases of different kinds of motion for the purpose of interpolating between them. Transition calculation in frequency space can take this phase information into account thus creating stable and robust intermediate motions. Manual creation and optimization of gaits was successfully performed using frequency space representation.

Future work will investigate the possibilities of automated optimization, learning and adaptation to changes in the environment. Adaptation could be achieved by having a set of standard gait patterns (e.g. one being fast on flat surfaces and another one being slower but more robust in general); as an decrease in performance for a gait pattern is observed in a given environment, a gradual transition to a different gait pattern can be triggered.

4 Acknowledgments

The project is funded by the Deutsche Forschungsgemeinschaft, Schwerpunktprogramm "Kooperierende Teams mobiler Roboter in dynamischen Umgebungen" ("Cooperative Teams of Mobile Robots in Dynamic Environments").

Program code was developed by the GermanTeam, a collaboration four German univeristies: the Humboldt University (Berlin), the University of Bremen, the University of Dortmund, and the Technical University of Darmstadt. Source code is publicly available at http://www.aiboteamhumboldt.com.

CMU walking engine is a courtesy of Carnegie Mellon University.

References

[1] A. Billard and A. J. Ijspeert. Biologically inspired neural controllers for motor control in a quadruped robot. 2000.

[2] X. Chen, K. Watanabe, K. Kiguchi, and K. Izumi. Optimal force distribution for the legs of a quadruped robot. *Machine Intelligence and Robotic Control*, 1(2):87–93, 1999.

[3] I. Dahm and J. Ziegler. Using artificial neural networks to construct a meta-model for the evolution of gait patterns. In *Proceedings of the 5th International Conference on Climbing and Walking Robots (CLAWAR 2002)*, September 2002.

[4] U. Düffert, M. Jüngel, T. Laue, M. Lötzsch, M. Risler, and T. Röfer. GermanTeam 2002. In *RoboCup 2002 Robot Soccer World Cup VI, Gal A. Kaminka, Pedro U. Lima, Raul Rojas (Eds.)*, number 2752 in Lecture Notes in Artificial Intelligence. Springer, 2003. More detailed in http://www.tzi.de/kogrob/papers/GermanTeam2002.pdf.

[5] J. Duysens, H. V. de Crommert, B. Smits-Engelsman, and F. V. der Helm. A walking robot called human: lessons to be learned from neural control of locomotion. *Journal of Biomechanics*, 2000.

[6] M. Fujita, S. Zrehen, and H. Kitano. A quadruped robot for RoboCup legged robot challenge. In *Proceedings of the second RoboCup Workshop*. Springer, 1998.

[7] M. Hardt and O. von Stryk. The role of motion dynamics in the design, control and stability of bipedal and quadrupedal robots.

[8] B. Hengst, D. Ibbotson, S. B. Pham, J. Dalgliesh, M. Lawther, P. Preston, and C. Sammut. The UNSW RoboCup 2000 Sony Legged League team. In *RoboCup 2000: Robot Soccer World Cup IV*, pages 70–72 (64–75). Springer, 2001.

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

[10] V. Hugel and P. Blazevic. Towards efficient implementation of quadruped gaits with duty factor of 0.75. In *Proceedings of the IEEE International Conference On Robotics and Automation*, 1999.

[11] H. Kimura, Y. Fukuoka, Y. Hada, and K. Takase. 3d adaptive dynamic walking of a quadruped robot by using neural system model. 2001.

[12] S. Lenser, J. Bruce, and M. Veloso. CMPack: A complete software system for autonomous legged soccer robots. 2001.

[13] M. A. Lewis. Gait adaptation in a quadruped robot. *Autonomous Robots*, 12(3):301–312, 2002.

[14] S. Nolfi and D. Floreano. Learing and evolution. *Autonomous Robots*, 7(1):89–113, 1998.

[15] S. Nolfi and D. Floreano. *Evolutionary Robotics.* MIT Press, 2000.

[16] K. Sims. Evolving 3D morphology and behavior by competition. In *Proceedings in Artificial Life IV, R. Brooks and P. Maes (editors)*, pages 28–39. MIT Press, 1994.