Automatic Gait Optimisation for Quadruped Robots

Min Sub Kim[*], **William Uther**[†*]
[*]School of Computer Science and Engineering, University of New South Wales
[†]National ICT Australia
{msk,willu}@cse.unsw.edu.au

Abstract

As part of the winning team in the Sony legged robot league of the RoboCup 2003 international robotic soccer competition, we used automatic gait optimisation. This was the first such technique seen in the competition and allowed an improvement in both speed and stability over previous gaits. Additionally, the technique was fast enough to run on-site at the competition, optimising the walk for the particular surface used. This paper describes the gait optimisation technique as well as some surprising features of the final gait discovered by the algorithm.

1 Introduction

It is clear that effective gaits have a significant impact on the overall effectiveness of legged robots in most environments. Having a fast, stable gait is an important part in allowing legged robots to accomplish larger tasks. Example applications include robotic pets, exploration robots, and more recent and popular interest in humanoid robots. In this research, the environment or problem context is robot soccer, in the form of the RoboCup competition [robocup-website, 2003].

RoboCup is an annual international robotic soccer competition that drives research in artificial intelligence and robotics by the motivation of competition. In practice, it provides an effective research test-bed for a variety of robotic techniques, as it allows direct comparison of complete robotic systems by playing them in matches.

The competition itself consists of a number of leagues, including the Sony legged robot league. In this league, teams compete using Sony ERS-210(A) quadruped robots [aibo-website, 2003] as a common hardware platform, and hardware modifications are forbidden. The authors were part of the rUNSWift team, a combined entry by the University of New South Wales (UNSW) and National ICT Australia (NICTA) in this league. The optimisation technique described in this paper was used by team rUNSWift at RoboCup 2003 to great effect.

From previous UNSW RoboCup entries in 2000 [Hengst *et al.*, 2002] and 2002 [Wang *et al.*, 2002] it is known that fast, stable gaits have a significant impact on the ability of a team to play effective robot soccer. The UNSW gaits from 2000 and 2002 were hand optimised, and were among the fastest gaits in the competition when introduced. However, hand optimisation is time-consuming and cumbersome. In this paper we describe an effective technique for automatic optimisation.

Previous work on automatically finding a gait [Hornby *et al.*, 1999, 2000] was very effective in finding the general form of a gait. Unfortunately, it was reported to be very stressful on the hardware of the robots used for learning, and still rather time-consuming. In this paper we introduce an automatic gait optimisation technique that searches over a constrained subset of possible gaits to optimise walk speed. By constraining the space of gaits over which our algorithm searches, we minimise both damage to the robot and optimisation time, while still retaining enough freedom to find an effective gait. As well as being fast and stable, the gait found by this technique at the RoboCup competition has a number of qualitatively surprising features. In particular, the front legs of the robot appear to walk backwards, or "moonwalk". We describe this gait in detail and our current understanding of why it is effective, as well as describing the optimisation technique itself.

2 Background

The locomotion module developed by the year 2000 UNSW team used a rectangular trajectory, or locus, for each foot of the robot [Hengst *et al.*, 2002]. The walk itself was parametrised by three movement parameters, one speed parameter, and eight stance parameters. These parameters were used to define the four corners of the rectangular shape to be traced out by the robot's feet, as well as the position of those rectangles relative to the robot's body.

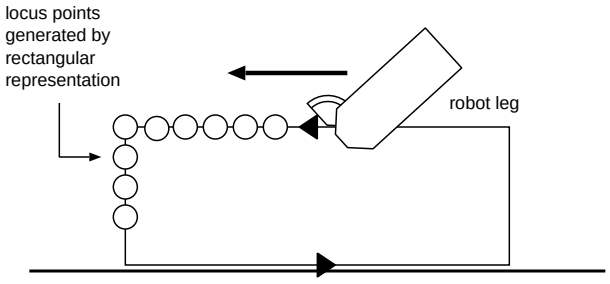


Figure 1: The original rectangular locus representation developed by the 2000 UNSW team, adapted from Hengst *et al.* [2002] and Wang *et al.* [2002].

To move a foot around this locus, a series of way-points about this locus were generated by the algorithm, the exact number being specified by the movement parameters. These points were divided into two groups of equal size; one group distributed evenly along the ground stroke of the locus, and the other group distributed evenly along the other three sides of the locus rectangle. Every 8ms the joint angles required to move the foot to the next way-point were calculated and sent to the robot's PID controllers. In this way, each foot moves smoothly around the locus, and the time taken for the foot to move along the ground stroke is equal to the recovery time – the time taken for the foot to lift off the ground, move forward and be placed back on the ground for the next ground stroke. A visualisation of the locus generation is shown in Figure 1.

This rectangular motion was used for each foot in a trot gait, where diagonally opposite legs are synchronised and the two sets of synchronised legs are 180° out of phase. The stance parameters were configured to produce a low forward-leaning stance that allowed the robot to take long strides while remaining stable. This gait was further improved with an additional canter motion. This adjusted the height of the shoulder and hip joints of the robot in a smooth, periodic sinusoidal fashion, which had the effect of increasing the forward speed of the walk. In addition, the robot was able to turn or walk sideways by rotating the rectangles relative to its body.

The 2002 UNSW team used a variation of the walk locus implemented by the 2000 team. The rectangular locus that had been used in the past was replaced with a trapezoidal locus, with a modified cycle timing around the locus and slightly adjusted stance [Wang *et al.*, 2002]. The change in the locus was designed to reduce the slowing effect caused by the claws on the robot's rear feet catching on the carpet surface. This change resulted in a significant speed increase of approximately 12% over the rectangular locus.

With the introduction of the trapezoidal locus in 2002, it was discovered that while this gait was faster than the rectangular gait, it was not very manoeuvrable and it was not very smooth. For these reasons, the 2002 rUNSWift team used a combination of gaits and switched between them where appropriate.

This work builds on the previous rUNSWift gaits by searching over the space of quadrilateral loci to find an effective gait. This automation has two effects: we are able to find a more effective gait than used previously, and we are able to optimise that gait for a particular walking surface, *e.g.* the surface in use at the RoboCup competition. As with the 2002 architecture, we used the new walk only for walking forward, with no sideways component to the walk and less than 15° turn per step.

3 Representing the Locus

The problem representation is formulated with the intent of restricting exploration to small local changes in the walk locus. We do not consider large changes as it was known from Hengst *et al.* [2002] and Wang *et al.* [2002] that effective walks existed in the space being searched, and small changes both increase search speed and minimise disruption to other tasks, such as localisation, that may be dependent on characteristics of the previous walking system.

The new quadrilateral walk locus is described by four offsets from the original rectangular locus, and the speed of the robot foot around the locus is constant. Using the original locus as a base allows all of the current speed and turn controls to carry over to the new gait. In order to explore this search space, each of the offsets can be moved in three dimensions - either forward or backward, sideways, and up or down. Given four points, this produces a 12-dimensional vector as a search state representation.

Preliminary optimisation over this space revealed a walk that was not significantly different to, or faster than, the trapezoidal walk used in 2002. We then extended the search space to explore different locus shapes for front and rear legs. Thus, a search state is represented with a 12-dimensional vector for each of the front and rear pairs of legs, combined as a single 24-dimensional vector. The results for this 24-dimensional space are reported.

The basis rectangular locus is positioned relative to the robot body, with the same stance parameters as used by the trapezoidal locus walk in 2002, shown in Figure 2. Specifically, this places the centre of the front locus 60mm in front and 5mm out sideways from the shoulder joint, and the centre of the rear locus is positioned 55mm behind and 10mm out sideways from the hip joint. The shoulder and hip joints are raised 70mm and 110mm from the ground respectively. The result of

Figure 2: The low forward-leaning stance, adopted from existing walks.

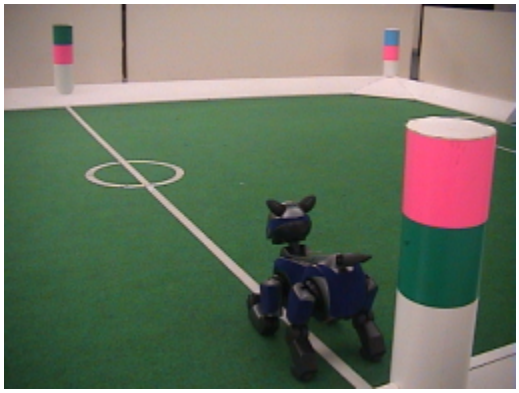


Figure 3: The experiment environment. The pink and green beacons serve as landmarks; by walking from one landmark to the other, the robot walks a path of fixed distance, thereby allowing evaluation of gaits by measuring the time needed to cover that distance.

this is a quadrilateral locus shape in three dimensional space.

3.1 Experiment Environment

The experiment environment was largely based on the setup used by Hornby *et al.* [1999, 2000]. Two landmarks are present, and the robot walks from one to the other, before turning around to repeat the process. The aim of the optimisation is to minimise the time required to cover the distance.

The landmarks used in the experiment were the coloured beacons used in the RoboCup competition. An advantage of this approach was that since our development was on top of the 2003 UNSW/NICTA team's entry, visual recognition and distance and heading estimation of the landmarks were readily available. To determine when the robot has reached a landmark, a combination of visual distance estimation and the robot's infrared sensor are used to gauge the distance. Once the robot determines that it is within a pre-defined distance of a landmark, it stops, turns around and lines up to face the opposite landmark, and begins the next trial run. The environment is shown in Figure 3.

The time measurement was based on the camera frame rate and execution cycle of the robot. The robot's camera receives 25 frames per second; a count of the number of camera frames processed is kept as the robot walks from one landmark to the other. The goal of the optimisation process then is to minimise this amount. Given the fixed distance walked, minimising time corresponds to maximising speed.

There is significant stochasticity in this experimental setup. Specifically:

- Error in distance measurement from sensors - this may cause the robot to stop either too close or too far in front of the target landmark. Consequently, the distance that the robot walks on each trial is not exactly the same.
- Error in timing - it is possible, although extremely rare, for the decision cycle of the robot to fall behind the camera. This causes the robot to skip a camera frame, and consequently, the number of camera frames processed would no longer be an accurate measure of time.
- Error in positioning - apart from the distance measurement error described above, the distance covered by the robot on different trials could also come out to be different if the robot ventured slightly off course. While the robot was programmed to head straight towards the target landmark, some error in sideways shifting was present, and difficult to avoid with our approach as points of the walk locus could be shifted sideways, resulting in walks that do not move in a straight line.

In particular, the error in distance measurement was most significant when the robot produced a bouncy walk. A bouncy walk had the side effect of shaking the robot's head, and as a result, the sensors would produce inaccurate distance readings and would tend to end trials prematurely. To work around this problem, the robot stopped briefly after each trial to settle, and then re-measured its distance to the target landmark; trials that fell short by more than a certain threshold were penalised.

To reduce the effect of stochasticity, a single measurement of the speed of a gait was implemented as four separate runs between the landmarks. The highest and lowest readings from the four runs were then discarded, and the average of the remaining two was used as the evaluated time. This had the effect of making the measurement require more time, but allowed us to use a

non-stochastic optimisation technique.

It should also be noted that while the robot was mostly walking straight forward, it was applying small corrections to its heading to reach the destination landmark. This had the important effect of introducing small amounts of turn into the evaluation of the walk. The final walk was quite robust to small amounts of turn.

3.2 Optimisation Method

With the problem representation and experiment environment established, the problem has been reduced to one of multidimensional optimisation and many algorithms are applicable. A major restriction is that gradient information is not available.

We expected the space to be relatively smooth, and the optimisation method that we chose to employ was Powell's (direction set) method for multidimensional minimisation, as outlined in Press *et al.* [1992]. Powell's minimisation method uses a set of directions in the multidimensional space, minimises along each one, and notes the effectiveness of each direction. It then uses that to derive new directions, with the intention of finding directions that minimise the function quickly.

Search Basis Directions

As stated previously, the problem representation consists of four points describing a walk locus, using different loci for front and rear legs, and each point can be moved in three dimensions, producing a 24-dimensional search space. Given this, an obvious approach would be to use each dimension as a direction. This would move each point in turn, trying to find an optimal position for it before moving on to the next point. However, Powell's method does not restrict the directions to being unit vectors in each dimension; in fact, the method explores different directions to try to find effective ones to minimise along. With that in mind, we define our initial directions differently, with the intent of exploring the search space in a more efficient manner. The set of directions we chose is a linear transform of the set of unit vectors and forms a complete basis set of the space, *i.e.* we can reach any point in the space using a linear combination of this set of vectors.

The selection of the initial minimisation directions is particularly important in this problem as each line minimisation takes a significant amount of time. Powell's method does not start to derive new directions until it has minimised along each of the initial directions, so it is very desirable that the initial directions allow for immediate improvement instead of working through all the initial directions before deciding that they were not effective.

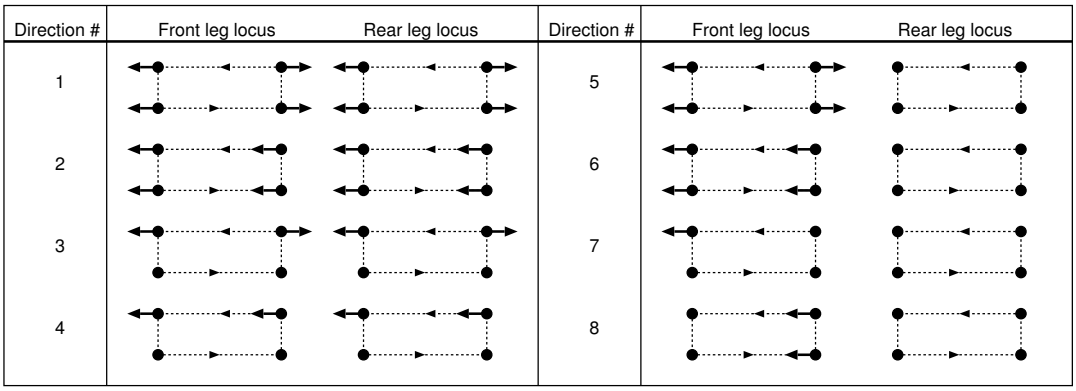
The first eight directions can be seen in Figure 4a. These minimisation directions affect the horizontal length of the locus. At this point, it is worth noting that when minimising along each direction, the values found for each direction may be positive or negative. In other words, the physical movement of the locus points for each minimisation direction may be in the directions shown in Figure 4a, or they may be reversed. The effect of the first eight directions are described below:

- Direction 1: stretch both loci outwards or shrink both loci inwards
- Direction 2: translate both loci forwards or backwards
- Direction 3: stretch the top of both loci outwards or shrink the top of both loci inwards
- Direction 4: translate the top of both loci forwards or backwards
- Direction 5: stretch the front locus outwards or shrink the front locus inwards
- Direction 6: translate the front locus forwards or backwards
- Direction 7: translate the front top point of the front locus forwards or backwards
- Direction 8: translate the rear points of the front locus forwards or backwards

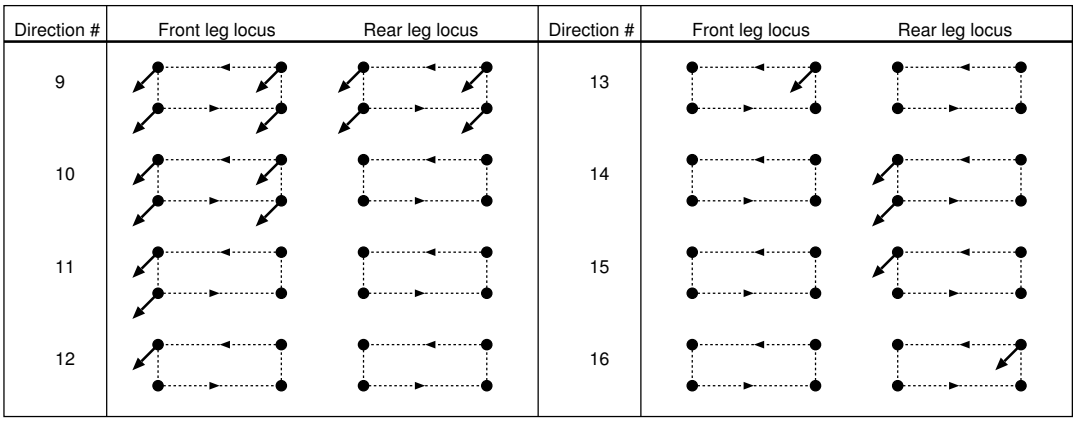
The next eight minimisation directions are shown in Figure 4b. These directions shift parts of the loci closer towards or further outwards from the body, and are individually described below:

- Direction 9: translate both loci towards or away from the body (or in other words, widen or reduce the sideways spread of both legs)
- Direction 10: translate the front locus towards or away from the body
- Direction 11: translate the front points of the front locus towards or away from the body
- Direction 12: translate the front top point of the front locus towards or away from the body
- Direction 13: translate the rear top point of the front locus towards or away from the body
- Direction 14: translate the front points of the rear locus towards or away from the body
- Direction 15: translate the front top point of the rear locus towards or away from the body
- Direction 16: translate the rear top point of the rear locus towards or away from the body

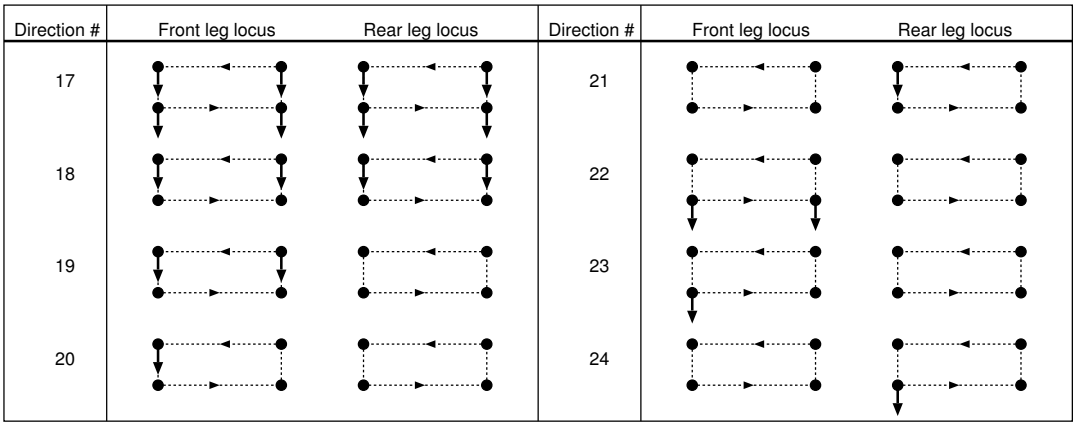
The final eight minimisation directions deal with the height of the walk locus. This affects the height that the robot lifts its legs when walking forwards, and are detailed below:



(a) Forward/backward directions: these directions have the effect of modifying the horizontal length of the locus.



(b) Sideways directions: these directions have the effect of either bringing the locus closer towards or further outwards from the robot body.



(c) Vertical directions: these directions have the effect of either raising or lowering the height of the locus.

Figure 4: Search directions used for manipulating the locus with Powell's method.

- Direction 17: translate both loci higher or lower (or in other words, raise or lower the height of the body)
- Direction 18: translate the top points of both loci (or raise or lower the height of both loci)
- Direction 19: translate the top points of the front locus
- Direction 20: translate the front top point of the front locus
- Direction 21: translate the front top point of the rear locus
- Direction 22: translate the bottom points of the front locus
- Direction 23: translate the front bottom point of the front locus
- Direction 24: translate the front bottom point of the rear locus

In coding terms, our implementation of the minimisation method itself was heavily based on the sample code found in *Numerical Recipes in C* [Press *et al.*, 1992]. The only significant difference in code was that the algorithm flow needed to be rolled out, as it takes many decision cycles on the robot before it can evaluate a particular locus.

The individual direction minimisations used in Powell's method were also implemented as per the sample code in *Numerical Recipes in C* [Press *et al.*, 1992]. Specifically, this involved a combination of a golden ratio search that exponentially increases an interval size to initially bracket a minimum, and a parabolic interpolation technique known as Brent's method to find the minimum in that interval.

3.3 Other Experimental Details

The minimisation method was run at the 2003 RoboCup competition in Padua, Italy. Unfortunately, due to preparation time constraints at the venue, we were not able to run the method through to complete convergence. We were able to get through minimisation along 22 of the 24 dimensions. At one point during the minimisation we noticed that the robot seemed to be stuck in a poor local minima, and so we restarted that particular line minimisation. It did not get stuck in the local minima the second time. We believe the minima was an artifact of noise in the robot's measurements. Also, a programming error in the code resulted in some different minimisation directions; the positions of the third and fourth locus points were reversed. This inverts the two points on the right hand side of the rectangles in Figure 4. Note that this is a linear change in basis and the modified basis set can still reach every point in the search space.

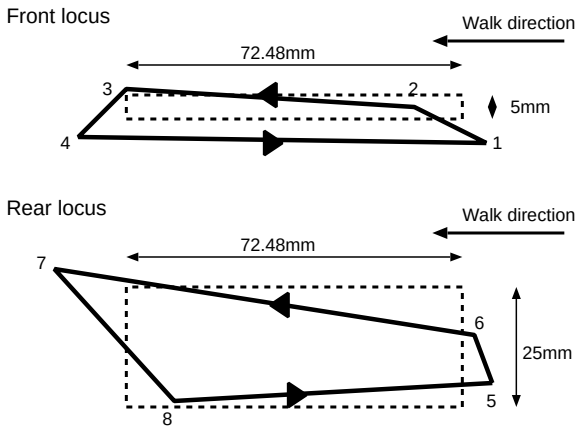


Figure 5: The loci produced by the minimisation technique in the sagittal plane, shown in bold. The dotted lines show the basis rectangular loci.

Locus Point	Forwards offset (mm)	Sideways offset (mm)	Vertical offset (mm)
1	-3.64829	4.65264	5.44244
2	9.40879	-2.16859	1.80812
3	0.50342	2.12128	-0.298782
4	7.92933	10.6357	4.47817
5	-6.30445	-4.70413	-5.22602
6	-2.58385	14.6642	10.4322
7	13.8908	2.5354	-4.09243
8	-8.3251	-1.80062	-1.51026

Table 1: The offsets learned by our optimisation procedure.

Another problem encountered was that the robots were operating on lithium ion battery packs, with operating durations of around 20 to 30 minutes. A tether to supply power to the robot, as in Hornby *et al.* [1999, 2000], is a feasible alternative; however this was neither readily available nor desired as the tether could affect the balance of the robot. Consequently, the entire internal state of Powell's algorithm was logged each time minimisation along a single dimension was complete, and the robot could then be restarted from a logged state on a fresh battery.

4 Results

The loci produced by the minimisation technique are shown in Figure 5 and Table 1. The offsets in Table 1 are added to the $x - y - z$ positions of corresponding corner points in the original rectangular loci to produce the final loci. For example, locus point 1 corresponds to the bottom right corner point of the front locus. To calculate the shifted position of locus point 1, the bottom right corner point of the front locus in the basis rectangular

locus is shifted back 3.64829mm, 4.65264mm outward, and 5.44244mm downwards.

In general, the walk tended to take long forward strides with the front legs, with the sideways spread reducing in towards the body when stretching forwards and spreading outwards from the body when pulling back. The rear legs took large, high steps forward, with sideways spread reducing in towards the body when pushing back and spreading outwards from the body as the legs were raised and brought forward for the next step.

The speed of the gait was measured at $27\text{cm}\,\text{s}^{-1}$. This was significantly faster than the previous hand-developed trapezoidal walk locus on the particular surface used in the RoboCup 2003 competition. In addition, the minimised locus was considerably smoother than the trapezoidal locus, and kept the robot's camera steadier. This had helpful effects on other tasks required to play soccer effectively, such as distance and velocity estimation and localisation. To put the result in context, the 2003 German Team informed us at the competition that they were walking at $23\text{cm}\,\text{s}^{-1}$.

To evaluate the effectiveness of the technique under more controlled conditions, the walk produced by the technique at RoboCup was compared with the hand-crafted trapezoidal locus and rectangular locus walks in our lab. The comparison was run on a similar surface to that used at RoboCup, timing the robot over a distance of 1 metre, over 50 trials for each walk. The rectangular locus was the slowest, as expected, with an average speed of $22.69 \pm 0.45\text{cm}\,\text{s}^{-1}$. The trapezoidal locus was next, measured at $25.42 \pm 0.50\text{cm}\,\text{s}^{-1}$. The walk produced at the competition came out as the fastest, measured at $26.99 \pm 0.50\text{cm}\,\text{s}^{-1}$. A graphical comparison of the average speeds is shown in Figure 6.

An interesting observation made of the minimised loci was that the actual locus traced out by the front feet when the robot moves along the field is different to the locus traced out when the robot is held in the air. In general, the front legs tended to walk much more on the elbow with the foot raised slightly off the ground. As a result, it appears as though the front feet are actually tracing out a locus going backwards. An approximation of the motion is shown in Figure 7. The effect of this locus is that the front forearms are used as skis during the recovery leading to a very stable walk. The robot then puts weight on its elbow during the ground stroke.

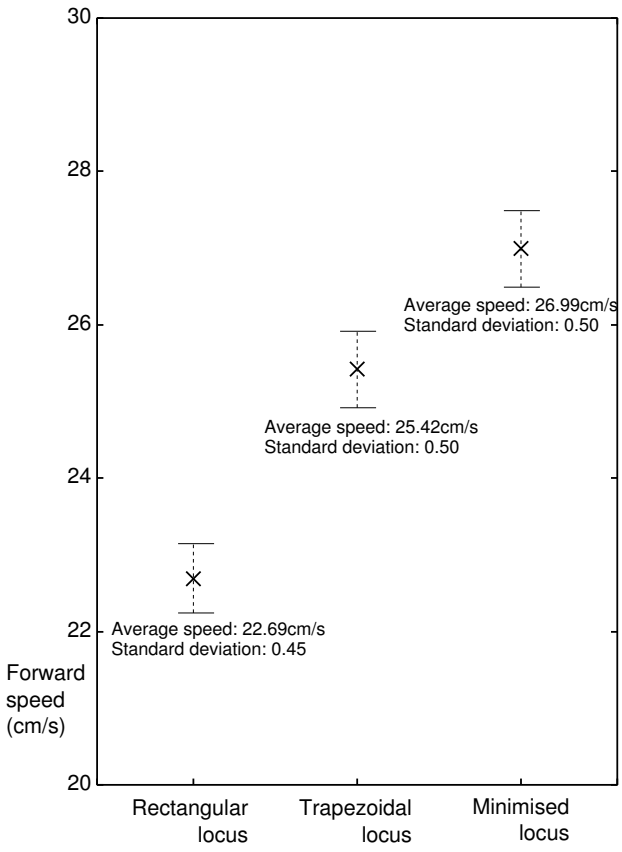


Figure 6: A comparison of the average speed of the hand-crafted rectangular and trapezoidal loci, and the minimised locus produced by this technique.

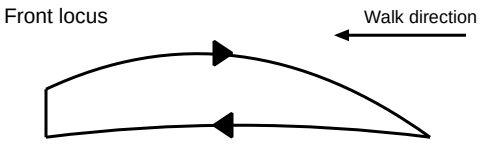


Figure 7: The observed locus traced out by the front foot when the robot walks along the field.

5 Discussion

It is clear that this technique is an effective method of gait optimisation. Even though we did not run the optimisation technique to completion, the gait found is 6% faster than the previous hand optimised gait. Additionally, the new gait is more stable than the previously

hand optimised gait. We might expect a small amount of further improvement were the optimisation given more time, although we believe that the larger part of any improvement possible has already been achieved.

The observed behaviour of the locus traced out by the front feet on the field was significantly different to that when the robot was held up off the ground. As shown in Figure 7, the front foot actually seems to move backwards along the locus. We suspect that the cause of this was that the inverse kinematics used to translate the feet positions on the locus to joint angles for the motors does not ensure that the foot will be below the elbow. As a result, the method produced a front locus that actually had the robot walking on the front elbows: along the bottom edge of the observed locus, the robot would lift its elbow slightly and step forward with it, and along the top edge, the robot was actually pushing back with its elbow, and raised its foot in the process. We believe that the locus appears to be going backwards because of the rocking motion of the body; that is, the locus relative to the body without the rocking motion is that shown in Figure 5, whereas the locus relative to the ground with the rocking motion is that shown in Figure 7. In effect, the front locus did not act so much as a walk locus, but rather acted as a walk parametrisation that was optimised to suit the surface and the body motion, a result that would have been unintuitive to develop by hand. The motion is shown and described in Figure 8.

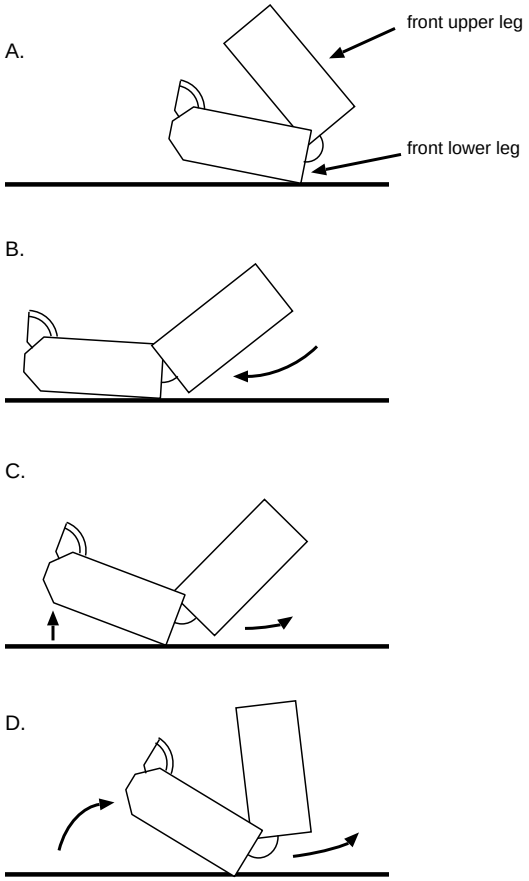


Figure 8: Four points of the front leg motion. At (A), the front leg is wound far back, resting on the lower side of the elbow. Then in (B), the upper leg is rotated forward and the lower leg is dropped slightly, reaching far forward. As a result, the height of the body drops a little. In (C), the upper leg starts to rotate back, and the lower leg lifts a little. This has the effect of bringing the robot body forward slightly, using the lower side of the elbow to push forward. Finally, in (D), the upper leg continues to rotate back to push the body forward, and the cycle resumes again from (A).

6 Conclusion

We have described an effective problem representation and technique for optimising the gait for a quadruped robot. The result of applying the technique at the RoboCup 2003 competition was a walk that improved modestly on the speed and significantly on the steadiness of previously hand-developed walks. Furthermore, the application of the technique on-site produced a walk that was adapted to the particular surface at the venue.

Acknowledgments

We would like to acknowledge the other members of the 2003 UNSW/NICTA team, as well as members of the UNSW team of previous years. In particular, we thank Bernhard Hengst who originally crafted the locomotion module on which this work was developed, and who contributed many useful ideas. We would also like to thank Claude Sammut for his helpful advice.

References

Sony AIBO website. `http://www.aibo.com/`, 2003.

Bernhard Hengst, Darren Ibbotson, Son Bao Pham, and Claude Sammut (2001). Omnidirectional Locomotion for Quadruped Robots. In A. Birk, S. Coradeschi,

and S. Tadokoro, editors, *Lecture Notes in Computer Science, RoboCup 2001: Robot Soccer World Cup V*, pages 368–373. Springer, 2002.

G. S. Hornby, M. Fujita, S. Takamura, T. Yamamoto, and O. Hanagata. Autonomous Evolution of Gaits with the Sony Quadruped Robot. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1297–1304, Orlando, Florida, USA, 13-17 1999. Morgan Kaufmann.

G. S. Hornby, M. Fujita, S. Takamura, T. Yamamoto, and O. Hanagata. Evolving Robust Gaits with AIBO. In *IEEE International Conference on Robotics and Automation*, pages 3040–3045, 2000.

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, 1992.

RoboCup website. `http://www.robocup.org/`, 2003.

David Wang, James Wong, Timothy Tam, Benjamin Leung, Min Sub Kim, James Brooks, Albert Chang, and Nik Von Huben. The UNSW RoboCup 2002 Legged League Team. Undergraduate thesis in computer and software engineering, University of New South Wales, 2002.