Omnidirectional Locomotion for Quadruped Robots

Bernhard Hengst, Darren Ibbotson, Son Bao Pham, and Claude Sammut

School of Computer Science and Engineering
University of New South Wales, UNSW Sydney 2052 AUSTRALIA
{bernhardh,s2207509,sonp,claude}@cse.unsw.edu.au

Abstract. Competing at the RoboCup 2000 Sony legged robot league, the UNSW team won both the challenge competition and all their soccer matches, emerging the outright winners for this league against eleven other international teams. The main advantage that the UNSW team had was speed. A major contributor to the speed was a novel omnidirectional locomotion method developed for the quadruped Sony ERS-110 robot used in the competition. It is believed to be the fastest walk style known for this type of robot. In this paper we describe the parameterised omnidirectional walk in detail. The walk also made a positive contribution to other robot tasks such as ball tracking and localisation while playing soccer. The authors believe that this omnidirectional locomotion could be applied more generally in other legged robots.

1 Introduction

In RoboCup 1999 our own robots, as well as those of our competitors, moved slowly and gingerly, often falling over, even when unprovoked. From our observations in that competition, we concluded that locomotion was a fruitful area for research.

In 1999, the UNSW team, Dalgliesh and Lawther [2], used what they called atomic actions to control the robots movements. The atomic actions were simple macro actions to perform specific effector sub-tasks such as turning, walking forward or walking sideways. These atomic actions needed to be activated sequentially and could not be interrupted once initiated. Consequently robots would often continue to the completion of an atomic action, long after it was appropriate, in the circumstances. The transition between atomic actions was problematic in that robots could lose their balance and fall over because the leg positions were mismatched at the point of transition. While a fall and the struggle to get up are entertaining for the audience, they lose valuable time during a soccer match.

For RoboCup 2000 we decided to address these shortcomings by redesigning the locomotion for the Sony ERS-110 entertainment robot. The rest of this paper will describe the approach and implementation of an omnidirectional walk style that characterised the University of New South Wales entry in the competition. The omnidirectional walk gave the UNSW team a decisive advantage

A. Birk, S. Coradeschi, and S. Tadokoro (Eds.): RoboCup 2001, LNAI 2377, pp. 368–373, 2002.
© Springer-Verlag Berlin Heidelberg 2002

over the other teams. It is also a good illustration of the advantage a simplifying representation gives in making a difficult task more easily achievable.

2 Gait Selection

Hornby, et al in [4] describe three of the most common gaits used by quadruped robots, the crawl, trot and pace gaits.

The crawl gait operates by moving each leg in turn lifting one at a time. This gait is static in the sense that it keeps the centre of gravity inside the triangle described by the three legs that touch the ground. The trot gait lifts the two diagonally opposite legs alternately. The robot balances along the diagonal joining the two feet that are on the ground. In practice the robot is steadied by either of the other two legs not supporting the robot's weight. That is, the robot falls onto a third leg. The pace gait lifts both legs on the same side of the body simultaneously, alternating between the sides. This is the least stable gait as it is easy for the robot to lose balance and fall over sideways.

We experimented with all three gaits. From the results of early prototype walks, we rejected the pace gait because the robot would not walk forward reliably. It was difficult to synchronise the shifting of weight from side to side with the leg movements and it easily became unstable. The crawl gait was stable but slow in comparison with the trot gait. We therefore decided on the trot gait for the competition.

3 Design Objectives

Our three primary design objectives were to:

1. Drive the robot as if controlled by a joystick with three degrees of freedom. We wanted to be able to move forward or backward, sideways left or right and to turn on the spot clockwise or counterclockwise. This requirement suggested itself because of the shortcomings described above. Ideally we would like to be able to infinitely vary the speed of movement for any of the three degrees of freedom rapidly and simultaneously to allow the robots to react quickly to environmental changes.
2. Move the robot over the ground at a constant speed. This should reduce the strain on the robot motors by not accelerating and decelerating the body. It should also allow a faster walking pace to be achieved after the robot has gained momentum.
3. Keep the camera as steady as possible. The camera is located in the nose of the robot. We observed that, using previous walks, images from the robot's camera showed wildly erratic movements due to the robots head and leg motions. We postulated that a steadier stream of images would assist in object tracking. When approaching a ball, for example, it is desirable not lose sight of it.

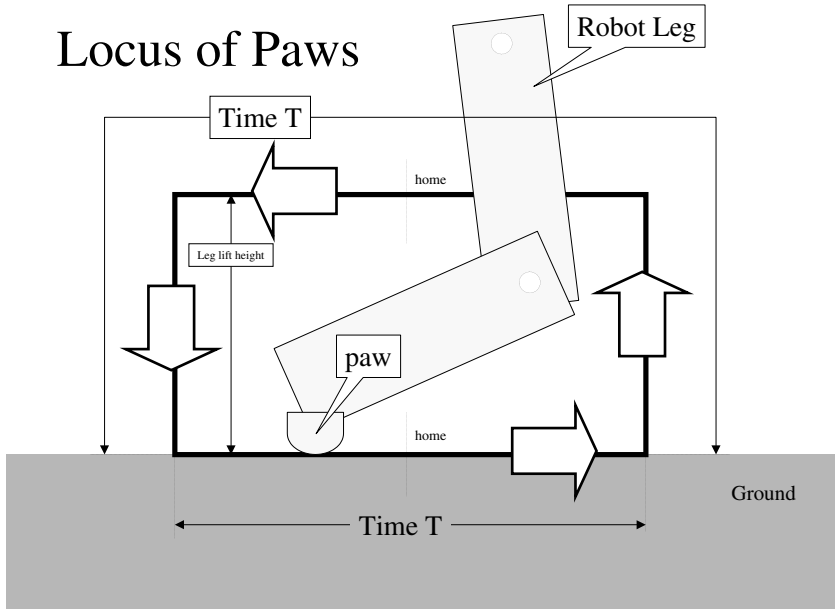

Fig. 1. Rectangular locus of each robot paw

4 The Representation

We chose to describe the leg action of the robots by a rectangular paw motion. The bottom edge of the rectangle describes that part of the path during which the paws make contact with the ground. The sides and top of the locus describe the path used to lift the paw back ready for it to take the next step. The legs that are touching the ground exert the forces that move the robot. In the trot gait diagonally opposite legs touch the ground alternately. If the paws that touch the ground move at a constant velocity, the robot should move at that same constant velocity. This requires that the time taken to move the paw along the bottom edge of the rectangle is equivalent to the total time taken to move the paw along the other three edges as shown in figure 1. Another way to think of the locus action is to see that its effect is similar to a wheel turning.

Design objectives 2 and 3 were achieved in this way. The speed over the ground is constant as long as the size of the rectangular locus does not change and its traversal is at a constant frequency. The robots camera is steadied because the bottom edge of the rectangular locus is a straight line lying in the ground plane. When the robot looses balance in the trot walk there is camera movement until it is arrested by falling on one of the legs that is off the ground. This movement can be minimised by lifting the legs as little as possible during the walk. Unfortunately in practice it was necessary to specify a significant leg

lift height to ensure that the spring-loaded claws would clear the carpet. This introduced some unwanted camera movement.

5 Omnidirectional Control

We will now address design objective 1, that is, how to control which way the robot moves. The plane containing the rectangular locus for the paw is always perpendicular to the ground. By changing the inclination of this plane relative to the sides of the robot we can determine whether the robot moves forward/backwards or sideways. For example if the locus plane is parallel to the robot sides, the robot will move forwards or backwards. Whether the robot moves forward or backward is determined by the direction in which the paw moves around the locus. It helps to imagine that the paw moving around the locus acts similarly to a rotating wheel. If we look at the robot side on and it is facing left, then if the paws move in a anti-clockwise direction around the rectangular locus the robot will move forward. If the paws are made to move clockwise the robot will walk backwards. If we incline the locus plane perpen-

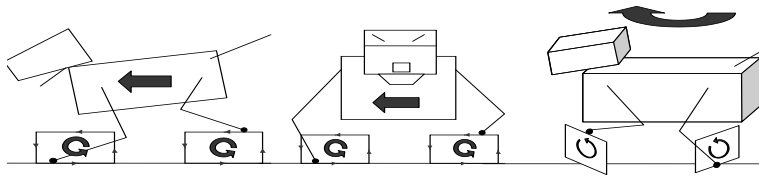

Fig. 2. Locomotion forward, sideways, and turning by moving the paws of the legs in a rectangular locus

dicular to the sides of the robot it will move either left or right in a similar fashion to the way it moves forward/backward. An inclination of the locus plan at another angle will cause the robot to move both forward or backward and left or right calculated by adding the vectors of both components. So far we have assumed that all the locus planes are kept parallel to each other and all the paws move either clockwise or counterclockwise. Before we move on to explain how the robot is made to turn, we should also note that the width of the rectangular locus and the speed at which it is traversed by the paw determines the speed at which the robot moves. There are limitations to these parameters and we will see later how they were tuned to achieve a locally optimal performance.

How can the robot be made to turn? Using the analogy of each rectangular locus as a wheel we can imagine that if we were able to turn each locus plane around a vertical axis we would have the freedom of movement of a supermarket trolley with each of its wheels motorised. This large degree of freedom may introduce conflicts where one leg can work against another. If we take a plan view

of the robot and describe a circle that passes through each of the upper leg joints, then by inclining the locus planes tangentially at each joint we can coordinate them to turn the robot clockwise or anti-clockwise. This constraint gives us the third property of our omnidirectional locomotion, that is, being able to turn on the spot. Figure 2 illustrates the three components of the omnidirectional locomotion.

Again, it is possible to combine components of each of the three movement dimensions (forward/backward, sideways and turning) creating complex waltz like movements in which the robot moves forward, sideways and turns simultaneously.

6 Optimisation

We used 3 control parameters, 1 speed parameter and 8 stance parameters that influence the leg movements for a particular walk style. A machine learning solution to finding the best speed and stance parameter setting suggested itself. We could perform a gradient ascent on the performance measure (for example forward speed) incrementally adjusting the parameters over many test runs. The problem with automating this approach was that it would take considerable time and resources to set up the learning environment. We were on a tight development time-line. Also of concern was the wear and tear on the robot motors and leg joints given the long periods of training that would be required. Hornby, et al [4] report continuous training times of 25 hours per evolutionary run using their genetic algorithm (GA).

The approach we adopted was to manually adjust the parameters after a number of runs of observing and measuring the performance. Unlike gradient ascent or a GA we were able to adjust the parameters using our judgement and knowledge about the robots dynamics. These considerations include:

1. The home position of the legs needed to be adjusted so that the robot will dynamically balance on the two legs that touch the ground.
2. A longer stride required the robot body to be closer to the ground to allow the legs to reach further.
3. The front and back legs do not interfere with each other during the cycle of the gait.
4. The legs needed some sideways spread to allow the robot to move its legs laterally so it could walk sideways.

Besides ground speed we also judged the reliability of the walk and the deviation from the intended path (accuracy) as a part of the performance measure. We found that there was a trade-off between the frequency of the gait and the length of the stride. Maximum ground speed was achieved with a longer stride and a slower gait frequency. This manual approach to tuning parameters used about 10 minutes of robot running time and only about 12 iterations of parameter adjustments to find a good performance setting resulting in the characteristic low forward leaning stance of the UNSW robots.

A further refinement that increased ground speed considerably was the introduction of a canter action. The canter action sinusoidally raises and lowers the robots body by 10mm synchronised with the trot cycle. The parameters were then manually tuned so that the robot was able to reach speeds of 1200cm/min. This compares with 900cm/min achieve in [4] using a GA approach which itself was reported to have improved on the previously fastest hand developed gait of 660cm/min. The camera is not as steady in this type of walk because of the additional canter movement.

7 Conclusions and Future Development

As evidenced from the results of the Sony legged robot league in RoboCup 2000, the parameterised omnidirectional walk performed very well. Often UNSW robots would reach the ball first at kick-off despite the significant distance handicap in the defending position. At no time during any of the competition matches did any of the UNSW robots fall over.

Gradient descent can only discover local optima. It is therefore possible that other combination of parameters may result in better performance. Lack of time prevented a more extensive search. Machine learning using either GA as in [4] or gradient descent could automate this effort given adequate resources. The geometry used to model the robots limbs and the inverse kinematic equations did not take into consideration the interference of the ornamental claws on the Sony ERS-110 entertainment robot. They have the effect of an unwanted lift on the hind legs in particular. Including their geometry may further improve the steadiness of the walk.

Because of the generality of the underlying dynamics of the locomotion, we believe it could be applied to any legged robot that has a similar geometry and capability.

References

1. Beer, R.D., Quinn, R.D., Chiel, H.J. and Ritzmann, R.E. Biologically Inspired Approaches to Robotics, Communications of the ACM, Vol. 40, No. 3, March 1997.
2. Dalgliesh, J. and Lawther, M. (1999). Playing Soccer with Quadruped Robots. Computer Engineering Thesis, University of New South Wales.
3. Hengst, B., Ibbotson, D., Pham, S.B., Dalgliesh, J. Lawther, M., Preston, P., Sammut, C. The UNSW RoboCup 2000 Sony Legged Team, (to be published for RoboCup 2001 by Springer), 2001.
4. Hornby, G. S., Fujita, M., Takamura, S., Yamamoto, T., Hanagata, O. Evolving Robust Gaits with Aibo. IEEE International Conference on Robotics and Automation. pp. 3040-3045. 2000.
5. Stone, P., RoboCup-2000 The Fourth Robotic Soccer World Championships, AI Magazine, Vol. 22, No. 1, pp. 11-38, Spring 2001