Evolving a Locus Based Gait for a Humanoid Robot

Gordon Wyeth, Damien Kee
Information Technology and Electrical Engineering
University of Queensland, Brisbane, 4072, Australia
{wyeth, damien}@itee.uq.edu.au

Tak Fai Yik
School of Computer Science and Engineering
University of New South Wales, Sydney, 2052, Australia
tfyik@cse.unsw.edu.au

Abstract

This paper describes a process for evolving a stable humanoid walking gait that is based around parameterised loci of motion. The parameters of the loci are chosen by an evolutionary process based on the criteria that the robot's ZMP (Zero Moment Point) follows a desirable path. The paper illustrates the evolution of a straight line walking gait. The gait has been tested on a 1.2m tall humanoid robot (GuRoo). The results, apart from illustrating a successful walk, illustrate the effectiveness of the ZMP path criterion in not only ensuring a stable walk, but also in achieving efficient use of the actuators.

1 Introduction

Humanoid robots require a stable, reliable gait for locomotion. The generation of a suitable gait is a complex operation, with many possible combinations of joint position and trajectory. On one hand, there is no analytical solution to this underspecified problem, while on the other hand, there is a very large number of parameters to tune to explore a solution space by hand. This paper demonstrates the use of a constrained locus of motion that restricts the search space, and characterises the effect of the gait. Genetic algorithms are used to explore the parameter space within the constraints of the locus in order to generate a stable walk.

1.1 Related Techniques for Humanoid Gaits

Many humanoid gait generation techniques revolve around the notion of Zero Moment Point (ZMP) stability [Vukobratovic 1969]. The majority of ZMP stable gaits generated for humanoid robots are generated by torso stabilisation of a ZMP unstable gait. Yamaguchi *et al* [1993] generates arbitrary leg motion that mimics the human walking gait but is not ZMP stable in its own right. Resultant moments in the roll, pitch and yaw axis are calculated based on the assumed movements of the legs. A complementary torso motion is calculated offline, then executed in parallel with the gait to achieve ZMP stability. Nagasaka *et al* [1999] calculates a statically stable torso motion, which then becomes a ZMP stable torso motion through on-line learning.

The generation of a gait that is intrinsically ZMP stable is more difficult. Yamasaki [2002] applied a genetic algorithm to determine the necessary joint motions to achieve stable walking; where stability is defined as the successful completion of a walking task. A first stage fitness function maximised the total distance travelled by the robot. Kinematic constraints such as a parallel foot during the swing phase and mirrored movements between legs simplified the process. Once initial walking movements had been generated, the fitness function was modified to reflect the ratio between the distance walked and the energy consumed. Further generations of the genetic algorithm improved the walking gait. This approach is an indirect way of achieving a ZMP stable walk. As the ZMP is never directly calculated, it is assumed that as long as the robot remains upright, the ZMP lies within the support polygon.

1.2 Locus Based Gaits for Four Legs

The use of a parametised walking gait with four legged robots has been explored by Hengst *et al* [2002] for the highly successful "rUNSWift" RoboCup Legged League team. Using the Sony AIBO as a base, a walking gait consisting of 13 hand tuned parameters was applied. Rectangular loci define the motion of a leg in a plane orthogonal to the ground. The orientation of the plane of the loci for each leg, with respect to the body of the robot, defines the overall translation and/or rotation of the robot. Omnidirectional locomotion is possible by combining forward, sideways and rotational movements into one gait.

1.3 Contribution of Paper

This paper describes a genetic algorithm that explicitly generates a ZMP stable path. The torso is fixed during the gait; ZMP stability is inherent in the gait itself. The gait is based on parameterised key frames in a kinematically constrained locus of gait motion. This formulation is shown to produce rapid and reliable convergence of the evolution process. Experiments with the "GuRoo" platform show reliable performance of the evolved gait. It is proposed that the evolution of a range of locus based gaits can form the basis of omnidirectional motion for a humanoid robot.

2 The GuRoo Project

GuRoo is a 1.2 m tall, fully autonomous humanoid robot (see Figure 1) designed and built in the University of Queensland Robotics Laboratory [Kee *et al*, 2003]. The robot has a total mass of 34 kg, including on-board power and computation. *GuRoo* is currently capable of a number of demonstration tasks including balancing, walking, turning, crouching, shaking hands and waving. The robot has performed live demonstrations of combinations of these tasks at various robot displays. The intended challenge task for the robot is to play a game of soccer with or against human players or other humanoid robots.

GuRoo has been designed to replicate the human form, to such an extent as conflicting factors of function, power, weight, cost and manufacturability allow. Figure 1 shows the degrees of freedom contained in each joint area of the robot. In the cases where there are multiple degrees of freedom (for example, the hip) the joints are implemented sequentially through short links rather than as spherical joints.

2.1 Electro-Mechanical Design

The robot has 23 joints in total. The legs and abdomen contain 15 joints that are required to produce significant mechanical power, most generally with large torques and relatively low speeds. The other 8 joints drive the head and neck assembly, and the arms. The torque and speed requirements are significantly less.

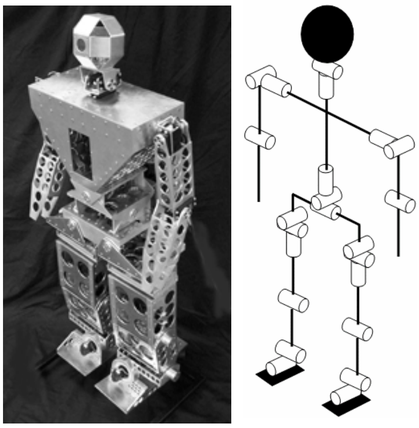

Figure 1: The GuRoo humanoid robot in its current form.

The motors that drive the roll axis of the hip joints are supplemented by springs with a spring constant of 1 Nm/degree. These springs serve to counteract the natural tendency of the legs to collide, and help to generate the swaying motion that is critical to the success of the walking gait.

The encoder feedback on each motor is currently the only sensing on the robot, although provision has been made for the addition of inertial and contact sensors.

2.2 Distributed Control Network

A distributed control network controls the robot, with a central computing hub that sets the goals for the robot, processes the sensor information, and provides coordination targets for the joints. The joints have their own control processors that act in groups to maintain global stability, while also operating individually to provide local motor control. The distributed system is connected by a CAN network. In addition, the robot requires various sensor amplifiers and power conversion circuits.

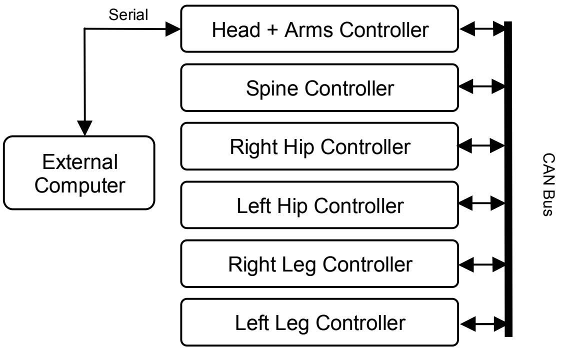

Figure 2: Block diagram of the distributed control system.

2.3 Software

The software consists of four main entities: the global movement generation code, the local motor control, the low-level code of the robot, and the simulator. The software is organised to provide a standard interface to both the low-level code on the robot and the simulator. This means that the software developed in simulation can be simply re-compiled to operate on the real robot. Consequently, the robot needs a number of standard interface calls that are used for both the robot and the simulator. Figure 3 shows modularisation of the software, and the common interfaces.

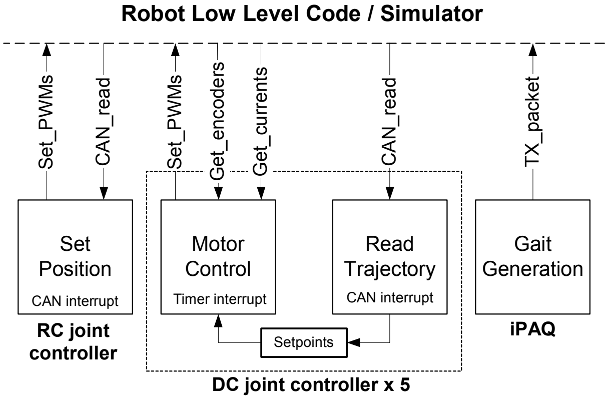

Figure 3: Block diagram of common software modules and the interface used to both the real robot and the simulator.

2.4 Simulator

The simulator is a high-fidelity dynamic simulation of the robot. The behaviour of the simulated robot accurately reflects the behaviour of the real robot, providing a great deal of confidence in simulated results. The simulator is based on the *DynaMechs* project [McMillan, 1995], with additions to simulate specific features of the robot such as the DC motors and motor drives, the RC servos, the sensors, the heterogeneous processing environment and the CAN network. These additions provide the same interface for the dynamic graphical simulation as for the joint controller and gait generation code.

3 Gait Generation

The gait generator yields walking gaits that are ZMP stable, and are based on a constrained locus of motion. The offline gait generator makes use of a low-resolution simulation and evolutionary computation. This approach involves learning a rough solution with an approximated model. Evolutionary computation is used for evolving walking gaits from a random population. Each gene is encoded to represents the motion sequence in terms of ankle positions. Limited information of the model on separation of legs and ratio of upper leg to lower leg for inverse kinematics calculation and rough weight distribution of the robot body is used. The ZMP of a motion sequence is estimated with the approximated model. The fitness is then measured by comparing the estimated ZMP trajectory of the playback of a motion sequence with a desired ZMP trajectory that meets stability criteria (with the projection of ZMP lying within the support polygon). After generations of evolution, a solution for a stable gait pattern for the approximated model results. This pattern can then be used with the high-resolution simulation or the actual robot.

3.1 Motion Sequence

The motion is described by a series of Cartesian positions of the ankle joints relative to the hips, termed *key frames*. The full locus is linearly interpolated from the four key frames generated by evolution. Note that only the locus of half of the walk-cycle (left swing leg) is generated. The full locus can be obtained by mirroring the left half-locus to become a right half-locus, and then concatenating the two loci.

In order to reduce the search space, some rules are applied to the motion set. The two legs of the robot must move in parallel, when viewed from the frontal plane. With this restriction, the two legs cannot collide and thus collision detection is redundant. The clearance of the foot from the floor is manually set. The position of the opposing ankle in the direction of motion is calculated by adding the specified stride length value to the hind leg except for the last key frame. In order to have the swing legs to swing forward, the stride length of the last key frame is set to 0, which means the legs must goes across each other for that frame. The hip base must travel horizontally, so the height of hip is fixed.

Since stride length, foot clearance and leg crossing are given, the shape of the ankle locus is set. The gait generator then finds the motion sequence that would satisfy this specification while following the desired ZMP trajectory through evolution.

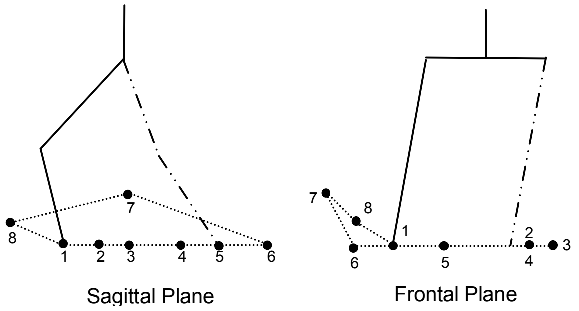

Figure 4: The locus traced by the left foot with respect to the hip, as generated by the GA. Heel strike of the left foot occurs at Stage 8, raised slightly from stage 1 to reduce impact disturbances. The stick figure is indicative and not to scale.

3.2 ZMP Estimation

The full robot dynamics are not simulated for gait evolution. Instead, the estimation of ZMP trajectory is based on a point mass model of the robot. This represents a significant saving in computational time. With a simplified robot model, ZMP trajectory and fitness of motion sequences can be estimated in a short period of time.

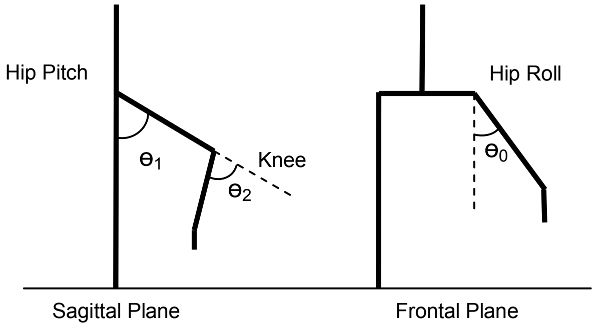

Figure 5: Approximated configuration of GuRoo used to simplify ZMP estimation during gait evolution.

The point mass model consists of a simplified 3D robot stick figure with an approximated weight distribution and configuration. The centre of mass of each component of the robot is flattened to a single plane and the hip pitch and hip roll joint treated as a single point, which is different from the configuration of GuRoo (Figure 5). At this stage, joints in upper body (waist, head and arms) and leg twist are not used to reduce the search space and simplify the inverse kinematics and ZMP calculations.

An inverse kinematics function works out θ_0, θ_1 and θ_2 based on the Cartesian position of the ankle given relative to the hip base. One assumption made for this simulation is that the feet of the robot are always parallel to the floor surface, and that the torso is vertical to the ground plane. With these joint angles, and information on weight distribution, the positions of centre of mass for body components with respect to the hip base can be found.

Figure 6 shows an example of the desired ZMP trajectory and the estimation of the ZMP for an evolved gait on y-axis. A cosine function is used to smooth the transition of the ZMP and the amplitude is chosen to ensure that the ZMP lies within the single support foot for heel-strike and lift-off.

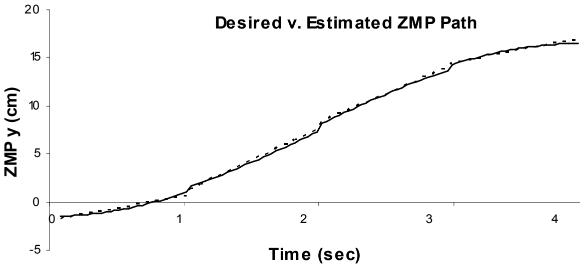

Figure 6: The estimated (solid line) and desired (dotted line) *y*-component of the ZMP for an evolved gait.

3.3 Gait Evolution

The gene contains the information of the slope and offset of the forward axis of ZMP trajectory and Cartesian positions of the motion key frames. Floating point numbers are used to store the information encapsulated by the gene. A multi-point crossover operation is used, which swaps the values in the rest of the series at nine random crossover points. The mutation operation is applied to the x and y coordinates of ankle positions and the offset of desired trajectory only, while keeping the predefined relations between entries. Gaussian mutation adds a random number from a Gaussian distribution to every corresponding entry of the gene. As a result, the gene is moved to a new position close to its current position instead of shifted to a random position in the state space. The parameters used for the genetic algorithm are summarised in Table 1. The error function is defined as the difference between desired ZMP trajectory and actual ZMP trajectory at each sample point over *n* sample points for the motion sequence. The fitness function is the reciprocal of average error.

$$\text{fitness} = \frac{1}{\sum_{k=1}^{n}(x_{ZMP}(k) - x_{DES}(k))^2 + (y_{ZMP}(k) - y_{DES}(k))^2}$$

Results show that it takes less than 500 generation to converge using multi-point crossover and Gaussian mutation operations (Figure 7). In terms of time, a ZMP stable gait is generated after approximately 3 minutes on a Pentium 4 1.8GHz machine.

Table 1: Summary of the parameters used in the evolutionary process.

Parameters for Gait Evolution	
Population Size	100
Generation	500
Crossover Ratio	0.7
Mutation Ratio	0.2

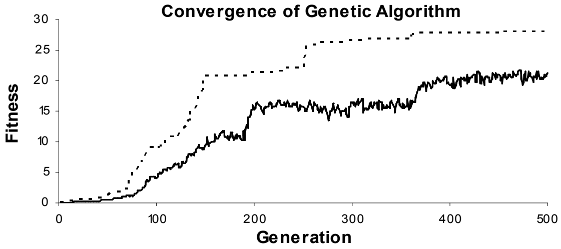

Figure 7: Convergence of the evolutionary process. The dotted line shows the maximum fitness in each generation, while the solid line shows the average fitness.

4 Walking

The gait generator creates a gait that is approximately ZMP stable, where the approximation is due to the modelling assumptions used for the evolutionary simulation. The robot should therefore walk stably, so long as the joints actually follow the trajectories proscribed by the gait generator. The ability of the joints to follow the position commands is principally determined by the saturation of motor torque. It is proposed that the motors are unlikely to become saturated provided that the gait follows through a smooth transition of ZMP stable points.

Based on this argument, the gait is best analysed by comparing the desired velocity from the gait generation module to the actual velocity at each joint. The comparison of desired to actual shows the loading on each joint as the control loop exerts torque to match the command. Figure 8 shows the comparison for the pitch motion of the hip, knee and ankle while Figure 9 shows the roll of the hip and ankle. These graphs form the basis of the following analysis.

The walking gait has a step rate of 0.245 Hz using a step length of 105mm. The gait is broken into eight distinct stages, each 1.02 seconds in length. As described in the gait generation section, a complete walking cycle commences and terminates when the left heel strikes the ground. A study of the left leg follows, with results for the right leg identical save the appropriate phase difference. The period of time from 0 to 1.02 seconds, is labelled Stage 1, with subsequent stages at 1.02 second intervals.

4.1 Analysis of Results

During stages 1 and 2, the GuRoo is in the double support phase, with ZMP transferring from the right leg to the left. The left hip roll and left ankle roll actuator accurately track the desired position trajectory. The spring in parallel with the hip roll actuator compresses during this movement. The hip pitch, knee and ankle pitch are all working together to propel the robot onto the right foot.

At the start of Stage 3, the left leg becomes the support leg. Both roll actuators continue to follow the desired trajectory initially, but with the ankle roll motor unable to reach the maximum angle requested. This is due to the ZMP not quite reaching the centre of the foot, and the robot collapsing slightly inwards. The hip roll actuator is aided by the weight of the robot as it travels to its maximum angle. All three pitch actuators drive towards a straight leg configuration as reflected in each graph tending towards 0 degrees. The ankle pitch struggles as the weight of the robot moves forward.

During Stage 4, the robot tries to drive its weight back towards the right, in anticipation of the right heel strike at the start of stage 5. At this point, the hip roll motor, now having to drive against the weight of the entire robot, strains considerably and results in significant positional error. The left leg bends slightly in the pitch axis as the right extends to meet the ground. Position error in the ankle pitch actuator is resolved by the start of stage 5, as the right heel strikes the ground.

Stages 5 and 6 are the double support phase as the weight of the robot is transferred back to the right. The left leg straightens along the pitch axis, driving the weight forward. While the right leg was the support leg, sag in the hip roll actuators caused the swing leg to drift towards the centre line of the robot in the frontal plane. As a result, the foot makes contact with the ground prematurely. The left leg then extends to reach the ground. In reality, the foot is already on the ground and the act of extending it pushes the robot further towards the supporting leg. This position error is maintained on the supporting leg as the friction between the foot and ground is too much for the roll error to be resolved.

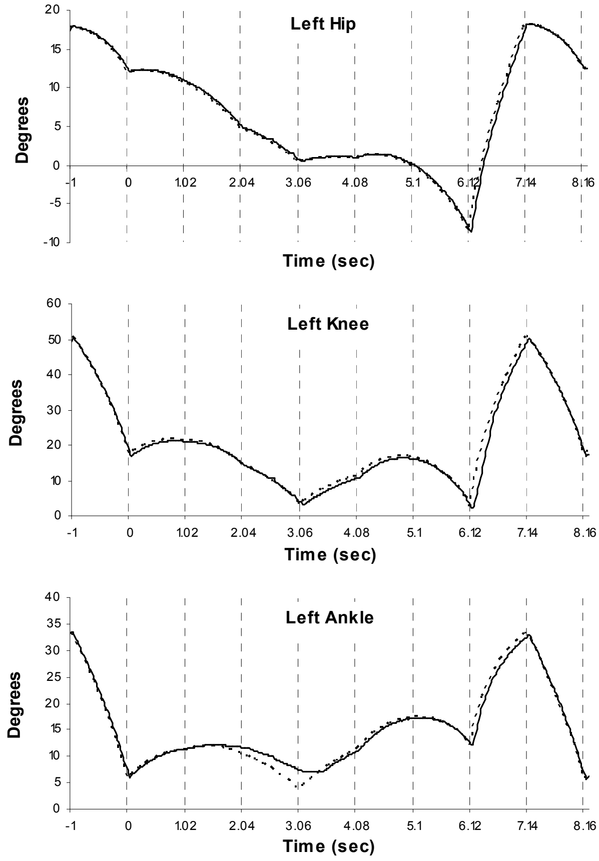

Figure 8: Positions of the pitch axis for the left leg hip, knee and ankle for a complete walking cycle. The dotted line shows the desired position, and the solid line is the actual position.

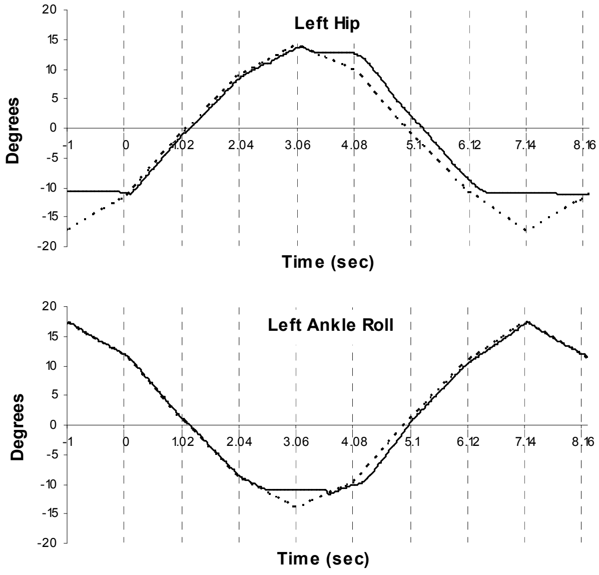

Figure 9: Positions of the roll axis of the hip and ankle of the left leg for a complete walking cycle.

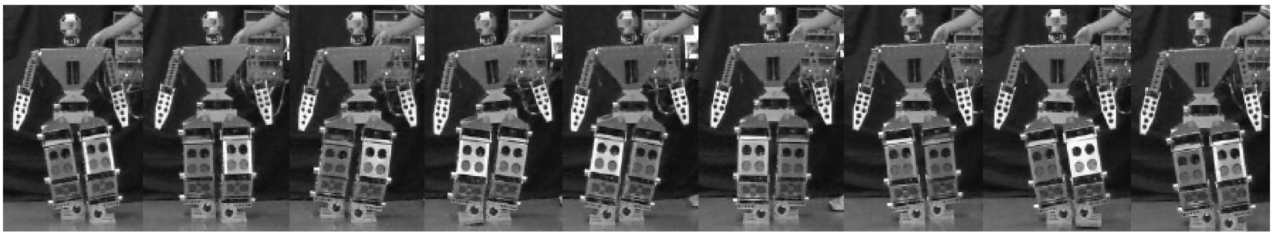

Figure 10: Frontal view of the walking process. The series starts with stage 1 and continues for a complete walking cycle.

Stages 7 and 8 represent the swing phase of the left leg. The pitch actuators lift the leg quickly and as the robot is propelled forward by the right leg, the left leg is extended to meet the ground. The left ankle roll actuator tracks the desired position perfectly, as it must only drive the foot during the swing phase, as opposed to the whole robot during the support phase. The left hip roll actuator again struggles during this stage, as it must now fight the spring loaded hip, without the additional mass of the robot.

The motion of the robot can be seen from the front in Figure 10, and from the side of the robot in Figure 11. The figures illustrate that the movement of the torso mass relative to the support platform provides significant movement of the ZMP.

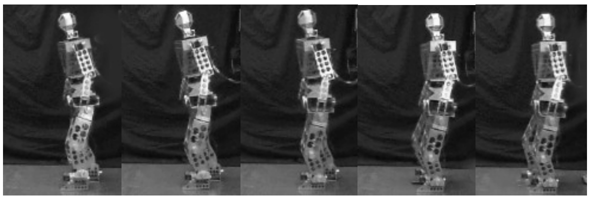

Figure 11: Stages 1, 2, 3, 4 and 5 of the left leg, and stages 5, 6, 7, 8 and 1 of the right leg.

5 Conclusions

The paper has presented a method of generating a ZMP stable gait that has been verified on a real humanoid robot platform. The method revolves around the use of parameterised loci of foot movement, where the parameters of motion have been determined by evolutionary computation. The loci evolve based on the criteria of how closely the ZMP follows a desired path.

A walking gait with 8 seconds per step cycle and a step length of 105 mm was generated. The results indicate that the method produces gaits that are realisable by actuators with limited torque production ability, indicating that the criteria of ZMP stability tends to lead to a gait that is not only stable, but also efficient in its use of available actuator torque. The ability to maintain a stable walk in controlled conditions with a fixed torso is beneficial as it provides scope for an additional means to reject unforseen disturbances.

Future work involves the evolution of other modes of walking (such as walking sideways) using the same method, with the aim of achieving omnidirectional locomotion for the GuRoo platform. The system is to be augmented with a global inertia sensor and balance system to improve overall gait stability and robustness with respect to disturbances.

References

[Endo *et al*, 2002] K. Endo, F. Yamasaki, T. Maeno and H. Kitano. *A Method for Co-Evolving Morphology and Walking Pattern of Biped Humanoid Robot*, Proceedings of the IEEE International Conference on Robotics and Automation, May 2002, Washington, DC

[Hengst *et al*, 2002] B. Hengst, S. Pham, D. Ibottson and C. Sammut, *Omnidirectional Locomotion for Quadruped Robots*, RoboCup 2001: Robot Soccer World Cup V, Springer, Berlin, 2002, pp 368-373.

[Nagasaka *et al*, 1999] K. Nagasaka, H. Inoue and M. Inaba. *Dynamic walking pattern generation for a humanoid robot based on optimal gradient method*. Systems, Man, and Cybernetics, 1999. IEEE SMC '99

[McMillan, 1995] S. McMillan, *Computational Dynamics for Robotic Systems on Land and Underwater*, PhD Thesis, Ohio State University, 1995.

[Vukobratovic 1969] M. Vukobratovic and D. Juricic, *Contribution to the Synthesis of Biped Gait.* IEEE Trans. on Bio-Medical Engineering, 1969. 16.

[Kee *et al*, 2003] D. Kee, G. Wyeth, A. Hood and A. Drury, *GuRoo: Autonomous Humanoid Platform for Walking Gait Research*, Autonomous Minirobots for Research and Edutainment,Feb2003, pp. 13-22.

[Yamaguchi *et al*, 1993] J. Yamaguchi, A. Takanishi, and I. Kato, *Development of a Biped Walking Robot Compensating for Three-Axis Moment by Trunk Motion.* International Conference on Intelligent Robots and Systems, 1993: pp 561 -566.

[Yamasaki *et al*, 2002] F. Yamasaki, K. Endo, H. Kitano and M. Asada. *Acquisition of Humanoid Walking Motion Using Genetic Algorithm – Considering Characteristics of Servo Modules*, Proceedings of the IEEE International Conference on Robotics and Automation, May 2002, Washington, DC