A Hybrid Software Platform for Sony AIBO Robots

Dragos Golubovic, Bo Li, and Huosheng Hu

Department of Computer Science, University of Essex
Wivenhoe Park, Colchester CO4 3SQ, United Kingdom
{dgolub,bli,hhu}@essex.ac.uk

Abstract. In the development of robotic systems, an interactive software platform plays an important role for control design and parameter optimisation. This paper presents a modular approach to the development of a hybrid software platform for Sony quadruped robots. Such a platform consists of an overhead vision system, a Sony AIBO robot and a desktop PC, and is designed to be interactive in order to speed up the development of various algorithms for object recognition and gait generation. Based on this platform, both the colour segmentation algorithm and the gait control algorithm are investigated. The experimental results are presented to show its operation and good performance.

1 Introduction

In the development of robotic systems, software development platforms play an important role for control design of robots, especially before a prototype system is available. A good software platform can provides simulation functions that speed up the development of different algorithms, including complex programming and huge data collection. There are mainly two types of software development platforms for robotics research. One is the pure simulation platform, in which only models of real robotic systems are presented. It is based on an assumption that the developed algorithms can be implemented in future. Another type is hybrid, which partially relies on the real robotic system. The parameters of this kind of systems are fully collected from real robots and the real environment, which is also based on real experiments. If the algorithm works well under such a hybrid platform, it can also work well on real robots. The benefit of developing a suitable software platform includes the ability to sample and store sensory data.

In this paper, we describe our research work based on Sony AIBO robots. In each AIBO, there are 20 motors for its motion control and over 30 sensors for feedback control and navigation. However, there are some difficulties in the development of control software and sensing algorithms based on AIBO robots. Firstly, programming Sony AIBO robots is complex since different algorithms and individual programs need to be developed and debugged on real robots. Such development cycle takes a long time and may have a risk to damage the robot. With a hybrid experimental platform, most of the work is done on a PC safely and efficiently. Secondly, the processor on the Sony AIBO robot is not powerful enough. Complex algorithms cannot be implemented on real robots in real time. It becomes necessary to develop complex algorithms on a PC during the development phase, then optimised in size for the real ro-

D. Polani et al. (Eds.): RoboCup 2003, LNAI 3020, pp. 478–486, 2004.

bots. Thirdly, the current interface between the Sony robot and human operators is not friendly. Each time when the experiment completed, all the results need to be downloaded from the memory stick manually, which is very time consuming. With the proposed hybrid platform, the progress of experiments is displayed on the screen directly and any problem with the experiment can be seen immediately, i.e. a convenient way to develop control and vision algorithms.

The rest of the paper is organized as follows. In section 2, a modular experimental platform is proposed, which consists of three parts: a real robot, an overhead camera and a PC. Section 3 describes the adaptive colour segmentation algorithm and some experimental results. Section 4 presents the gait generation algorithm being developed by using the proposed platform. Finally, section 5 presents a brief conclusion and future work.

2 A Hybrid Software Platform

The Sony AIBO robot is a dog-like robot mainly for entertainment. It can also be used for research on multi-agent systems and the robot soccer competition [4]. Each Sony AIBO robot has a quadruped design, approximately 30cm long and 30cm tall including the head. The neck and four legs of each robot have 3 degrees of freedom (DOFs). The neck can pan almost 90 degree to each side, allowing the robot to scan around the pitch for beacons, ball and other objects. And it has an onboard colour CCD camera and a hardware colour detection engine.

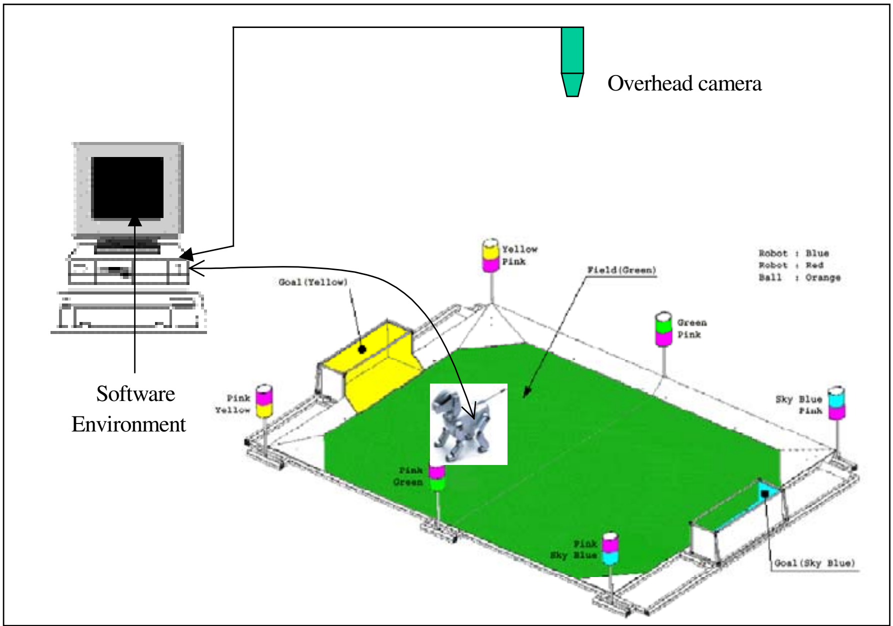

Fig. 1. System configuration of the proposed hybrid development platform

The proposed platform is based on three sub-systems: a Sony robot, a PC, and an overhead camera with one frame grab card. Figure 1 shows the system configuration in which wireless LAN is used for communication between the robot and the PC. In contrast, Figure 2 shows the basic software functions. More specifically, the program on the robot is designed to read instructions and implement basic behaviours. Its diagram is shown at the lower part of Figure 2. In contrast, the program on the PC is designed to provide a human-machine interface for the development vision and robot control algorithms. There are four main modules in it, namely,

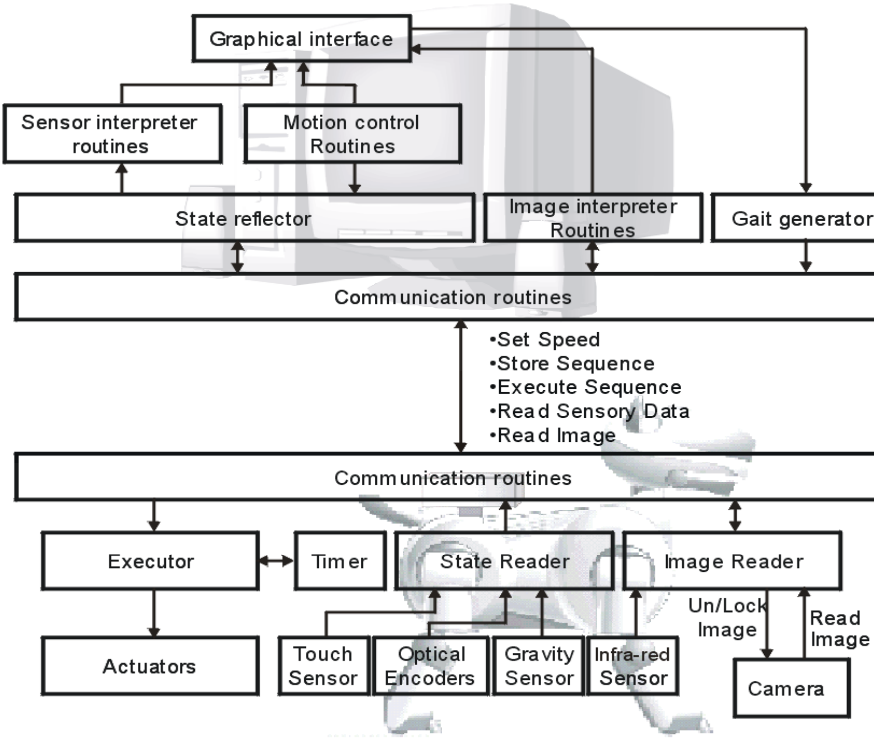

Fig. 2. Configuration of the proposed software development tool

State Reflector – An internal state reflector has been incorporated to mirror the robot's state on the host computer, which is an abstract view of the actual robot's internal state, such as sensor information from the robot and control commands from the host computer. More details can be seen in [2].

Communication Routines – The designed controller communicates with the robot using a handshake mechanism, and sends an appropriate command to the robot.

Gait Generator – The gait generator communicates with the robot by passing a sequence of arrays that are transformed into a sequence of robot movements. It creates different gaits in a form of a sequence of arrays.

Image Interpreter – Gathering image snapshots and processing images can be done completely independently from the rest of the application. The size of a captured image is 144 x 76 pixels and each pixel has three bytes for colour information. A locking mechanism has been adopted to allow the transfer of the current image to be safely completed before the new snapshot image can be grabbed.

3 Adaptive Colour Segmentation Algorithm

The task of extracting a colour object and estimating its position in images depends on colour segmentation [1][5], which needs to be robust to the varying lighting conditions and adapt to dynamic changes in the environment. The first step is to adjust the camera's shutter, aperture and amplifying rate settings, which is difficult for the Sony-legged robot. Such adjustments may cause other problems, such as change of the depth of view in the image. Therefore the implementation of different thresholds in different lighting conditions is necessary. Three main parts in our colour segmentation algorithm is shown in Figure 3.

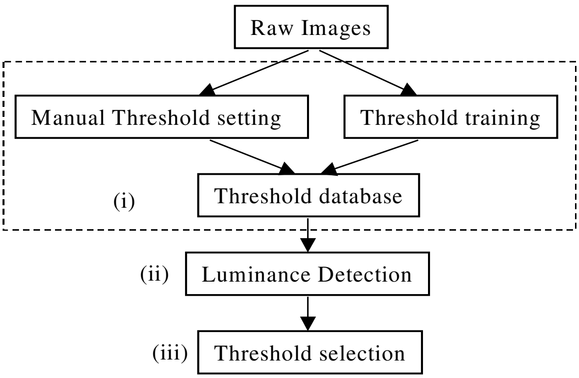

Fig. 3. The colour segmentation algorithm

3.1 Manual Threshold Setting

The threshold for the colour segmentation depends on the colour space in which the operation implemented. Commonly, HSV colour space is suitable for the colour segmentation, but the camera of a Sony AIBO robot outputs the images in YUV format and has a hardware colour detection table (CDT). The GCD method was developed by an Australia team, UNSW, to solve such a problem, which is very similar to the LUT methods presented by Schroter [8], i.e. a statistical approach. The thresholds of the GCD method can be any shapes in the UV space. The main difficulty of the GCD method is how to construct the thresholds under different lighting conditions.

Based on the proposed experimental platform, we can grab images from the on-board camera of the connected Sony AIBO. Figure 4 shows the raw image and the segmented image by using the CDT table at AGC=112. In contrast, Figure 5 presents the raw image and the segmented image by using the GCD table at AGC=112. Note that AGC represents "Auto Gain Control" here. As can be seen, the GCD method works better than the CDT method.

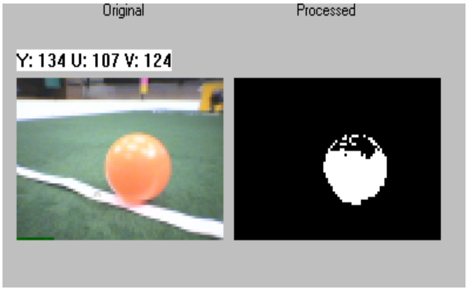

Fig. 4. Raw and segmented images (CDT)

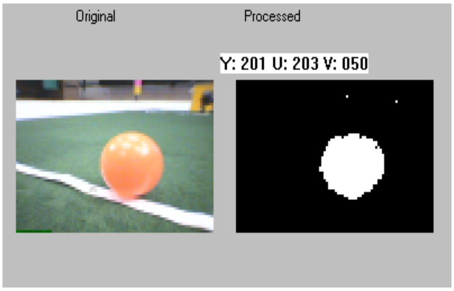

Fig. 5. Raw and segmented images (GCD)

3.2 Threshold Training Algorithm

Learning algorithms can be adopted to produce a suitable GCD or other kind of thresholds automatically [7][8]. Figure 6 shows the learning network used for GCD training, in which Y, U, and V are inputs, and output is the colour mark of the GCD. L1 is the luminance decided by the meter, and L2 is the luminance measured from the image. Training data are current GCD tables that are manually constructed with a set of lighting conditions. There are many different methods to measure the luminance of the images. Basically, most methods need a patterned object for measuring, which may be not reliable in the application of Sony legged robots. The field colour of a football pitch is green, which is not sensitive to the change of the lighting condition. However, the colour of other objects such as ball and goal is affected by light refection. With different angles, the measured results can vary dramatically under same lighting condition.

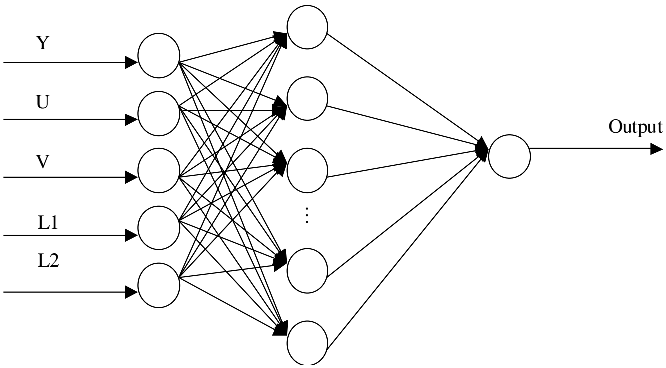

Fig. 6. A learning network for GCD training

Experiments were carried out using the real Sony robots to show the performance of the algorithm. Three different lighting conditions were adopted in the experiments: 405 Lux, 455 Lux, 535 Lux. GCD tables under 405 and 535 Lux were constructed manually. GCD tables for 455 Lux were constructed by the supervised learning. With large data sets, the speed of learning is very slow. For the learning process in a HSV colour space, the network was trained by 32MBytes test data over many hours. With the parameters such as the learning speed 5, 1 hidden layer, and 32 hidden neurons, it produced an inadequate GCD table with two many binary values 0 and the learning progress was too slow. On the other hand, with the parameters: learning speed 50, 1 hidden layer, and 8 hidden neurons, it produced a GCD table with too many binary values 1. The learning speed seemed very fast. Figure 7 shows the results obtained by using suitable GCD tables from training.

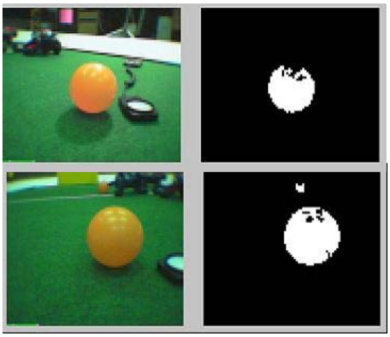

Fig. 7. Some Processed Images

4 Gait Generation Algorithm

4.1 Generation of Wheel-Like Motion

For a Sony AIBO, a trajectory refers to both the path of the movement of the tip of a limb (paw), and the velocity along the path. Thus, a trajectory has both spatial and temporal aspects. The spatial aspect is the sequence of the locations of the endpoint from the start of the movement to the goal, and the temporal aspect is the time dependence along the path. The essential conditions for stable dynamic walking on irregular terrain and on the flat ground can be itemized with physical terms:

- the swinging legs not be prevented from moving forward during the former period of the swinging phase,
- the swinging legs must be landed reliably on the ground during the latter period of the swinging phase,
- the angular velocity of the supporting legs during their pitching motion around the contact points at the moment of landing or leaving should be kept constant,
- the phase differences between the legs should be maintained in spite of delay of motion of a leg receiving disturbance from irregular terrain.

For the creation of wheel-like leg motion shown in Figure 8, we used six parameters for front and six parameters for rear legs. These twelve parameters determine the spatial aspect of a robot trajectory. The 13^{th} parameter is bound to temporary aspect and determines the speed of paw movement.

In total, there are 13 real-valued parameters are used to determine a gait for the locomotion module. Table 1 lists some of these parameters, which are also the genes for individuals evolved by the evolutionary algorithm. These parameters specify the position and orientation of the body, the swing path and the swinging rate of the robot legs, the amplitude of oscillation of the body's location and orientation.

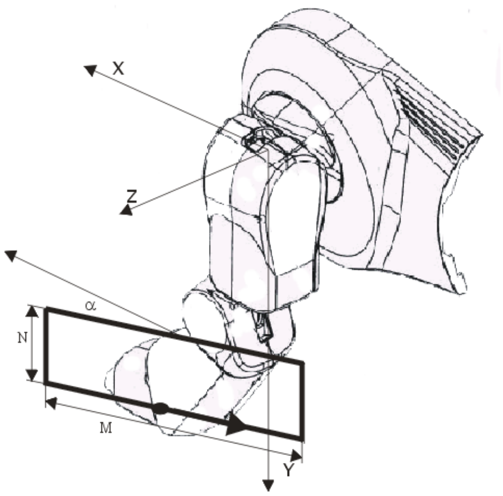

Fig. 8. Paw trajectory–Wheel-like motion

Table 1. Control Parameters for Gait Generation

	Parameter	Max value (mm)	Min value (mm)
Paw motion length	m	60	20
Paw motion width	n	50	10
X coordinate of COR	Xo	70	-20
Y coordinate of COR	Yo	140	50
Z coordinate of COR	Zo	40	-10
Paw rotation angle	α	90	-90
Paw moving speed	s	600ms/step	300ms/step

4.2 Implementation of Evolutionary Algorithms

We model natural processes, such as selection, recombination, mutation, migration, locality and neighbourhood. Figure 9 shows the structure of a genetic algorithm we implemented. Evolutionary algorithms work on populations of individuals instead of single solutions. In this way the search is performed in a parallel manner.

4.3 Evolving Results

The evaluation takes place inside of AIBO robot's football pitch. Each generation member produces a gait that runs for 5 steps. The 13th gene specifies the speed. If the robot falls over, it is fully capable of getting up and continuing its execution. After executing one member evaluation takes place and the speed and stability parameters are produced which qualitatively determines fitness values. The overhead camera records offset from the starting position and the changes when the robot to return to the starting position. In order to do that, an appropriate number of steps, e.g. forwards and backwards, are executed by the robot. We adopted a population size of 20 and run it for 50 generations, and results are presented in Figure 10.

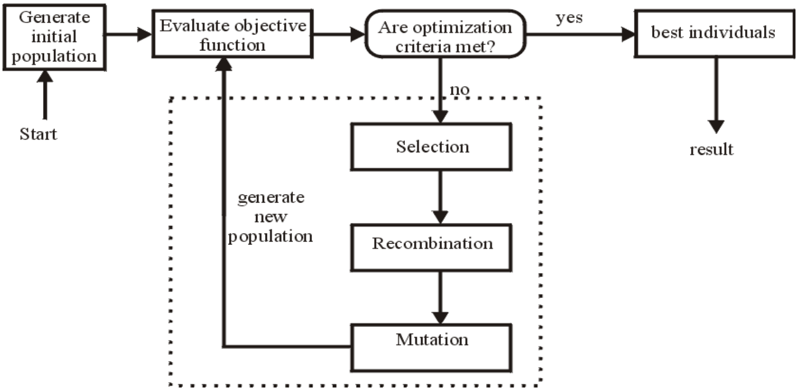

Fig. 9. Structure of a single population evolutionary algorithm

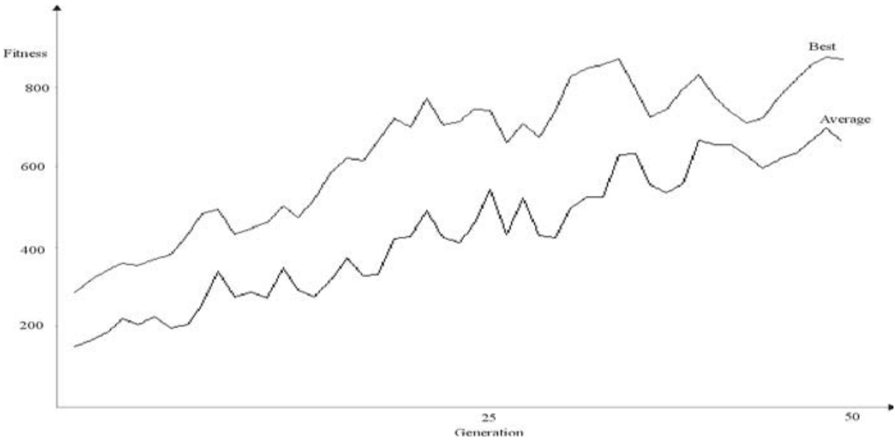

Fig. 10. The results of gait generation based on GA

At the beginning, most of the population members didn't perform very well. Some of them didn't even walk in the straight line or they even walked backward. Some of them were also causing the robot to fall. By the end of the experiment, the performance of the members significantly improved. The best member manages to walk in the straight line with good stability with the speed of 7m per minute.

We also investigated the improvement of speed and stability over a sequence of generations and the affects it has on the overall fitness functions, see [2] for details.

5 Conclusions and Future Work

A hybrid experimental platform for Sony AIBO robots is developed in this paper, which is a useful tool in the development of vision and gait generation algorithms in the RoboCup domain. Based on this platform, we can develop different vision algorithms for colour segmentation, object recognition and object tracking. Sony AIBO robots can learn good walking behaviours with little or no interaction with the designers. Once the learning method is put into place, the module can learn through its interaction with the world. The mutating and combination behaviours of the Genetic Algorithms allow the process to develop to a useful behaviour over time.

Our future work will be concentrated on three aspects:

- how to achieve the efficient operation and evaluation of tracking algorithms;
- how to improve vision system in terms of speed and accuracy;
- how to build robust and adaptive algorithms for gait generation.

Acknowledgements

We wish to thank Prof. Pierre Blazevic and his L.R.P research team for the code of the earlier version experimental platform, which gave us a starting point and motivation to develop this interactive software platform.

References

1. Y. Cui and J. Weng, "View-Based Hand Segmentation and Hand-Sequence Recognition with Complex Backgrounds", Proc. 13th Int. Conf. on Pattern Rec., pp. 617-621, 1996
2. D. Golubovic and H. Hu, An Interactive Software Environment for Gait Generation and Control Design of Sony Legged Robots, Proceedings of the 5th International Symposium on RoboCup, Fukuoka, Japan, 24-25 June 2002
3. D. Golubovic and H. Hu, A Hybrid Evolutionary Algorithm for Gait Generation of Sony Legged Robots, The 28th Annual Conference of the IEEE Industrial Electronics Society, Sevilla, Spain, November 5 - 8 2002
4. H. Hu and D. Gu, "A Multi-Agent System for Co-operative Quadruped Walking Robots", Proc. of the IASTED Int. Conference Robotics and Application, pp. 182-186, Aug. 2000
5. M. Isard and A. Blake, "Condensation -- conditional density propagation for visual tracking," Int. Journal of Computer Vision 29(1), pp. 5--28, 1998
6. B. Li, H. Hu and L. Spacek, An Adaptive Colour Segmentation Algorithm for Sony Legged Robots, Proceedings of the 21st IASTED Int. Conference on Applied Informatics, pp. 126-131, Innsbruck, Austria, 10-13 February 2003
7. T. Rendas and J. Folcher, "Image Segmentation by Unsupervised Adaptive Clustering in the Distribution Space for AUV guidance along sea-bed boundaries using Vision", 2001
8. S. Schröter, "Automatic Calibration of Lookup-Tables for Colour Image Segmentation", Proc. 3D Image Analysis and Synthesis, pp. 123-129, 1997