Proceedings of the World Congress on Engineering and Computer Science 2011 Vol I
WCECS 2011, October 19-21, 2011, San Francisco, USA

Evolving a Three Dimensional Lookup Table Controller for a Curved Ball and Beam System

Mark. Beckerleg, John. Collins

Abstract—.**This paper presents a novel approach to the use of a genetic algorithm to evolve a three dimensional lookup table which acts as a robotic controller to balance a ball on a beam. The lookup table translates three ball-beam states, ball position, ball speed and beam position, into the motor speed and direction required to maintain the ball in balance. A population comprising these lookup tables was evolved by applying a genetic algorithm using tournament selection, two-point crossover and a mutation rate of two percent. Four different ranges of motor speeds within the lookup table were successfully evolved, each capable of maintaining the ball in balance for over five minutes.**

Index Terms—**Evolvable robotics, evolution of lookup table, lookup table based robotic controllers, ball and beam controller, genetic algorithms**

I. Introduction

This paper investigates the use of a genetic algorithm (GA) to evolve a ball and beam controller by evolving a population of lookup tables (LUT) used to control the beam. The system developed for this paper (Fig 1) consists of four parts: i) the graphical user interface (GUI), which displays the motion of the ball and beam with control and data logging capabilities, ii) the GA, which evolves a population of LUTs, iii) the simulation, which models the characteristics of the ball-beam system and iv) the LUT, which provides the new beam motor speed and direction depending on the current ball-beam state.

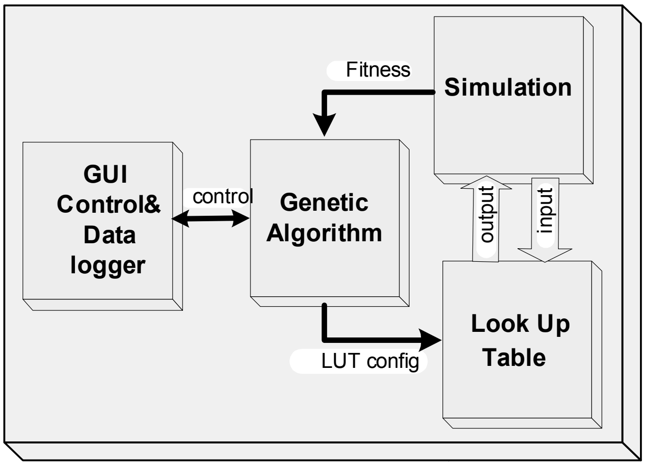

Fig 1. Block representation and connections between the four units that were implemented on a computer.

Manuscript received July 19, 2011; revised August 12, 2011.

Mark Beckerleg is with the School of Engineering, AUT University, New Zealand. phone: +64-9-9219999, fax: +64-9-9219973 (e-mail mark.beckerleg@aut.ac.nz)

John Collins is with the School of Engineering, AUT University, New Zealand. (e-mail john.collins@aut.ac.nz)

Typically a straight beam is used for a ball and beam apparatus, as it simplifies the control system algorithms that are required to balance the ball. However in this paper the beam is curved as this provides a more complex simulation model and algorithm and also means that the ball will never reach a static stable state with the motor stopped.

The simulation was modeled around a ball and beam system that was developed at AUT University for a student project (Fig 2). The physical beam was curved and had 19 infrared detectors that could determine the position of the ball, and a stepper motor that could alter the angle of the beam. The angular velocity of the beam was controlled by the number of pulses fed into the stepper motor per second. The maximum angular velocity was determined by the maximum pulse rate that the stepper motor could respond to. This was 125 pulses per second. The angular movement of 0.22 degrees per pulse gave a maximum angular velocity of the beam as 27.5 degrees per second.

Fig 2. The physical beam that the simulated ball and beam was modeled on.

This paper is organized as follows. In section two, background information about the ball and beam and the use of LUT in evolutionary computation is provided. Section three describes the mathematical model of the beam. In section four, the GA with its associated reproduction and selection schemes is explained. In section five, the simulation derivation and implementation is detailed. Section six contains the results of the evolution process and section seven contains the summary and conclusions.

II. Back Ground

Historically the ball and beam has been used as a standard laboratory apparatus to demonstrate control systems. It has also been used as a benchmark for research in control systems owing to its non-linear dynamics and behaviour. Several different control systems such as

ISBN: 978-988-18210-9-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

proportional integral differential (PID) control [1, 2], fuzzy logic [3, 4], and neural networks [5, 6] have been studied using the ball and beam. The application of a GA to evolve ball and beam controllers has also been investigated. In particular a GA has been used in robotic controllers to evolve the rules and classes of a fuzzy logic controller [7, 8], the weightings and connectivity of artificial neural networks [9, 10], and the coefficients of a PID controller [11, 12]. However to the author's knowledge the use of a LUT for the beam controller evolved by a GA has not been investigated.

LUT's have been used in evolutionary computation in a variety of applications, although not as a robotic controller. Robotic simulations have been replaced by a LUT, reducing the amount of computation required when running the simulation and thus the evolution time [13, 14]. Researchers have evolved cellular automata by performing a GA on a LUT that held the cellular automata rules, to create two and three dimensional shapes [15]. Robotic controllers have been evolved with a LUT that was encoded with simulated DNA sequences in order to create robotic motion that drew motifs [16]. Research has also been performed on evolving the LUT found within a FPGA's functional elements using custom software to avoid destructive configurations [17].

The use of a GA to evolve a robotic controller based on a LUT has been performed by the authors on two robotic systems including a mobile inverted pendulum [18] and the gait of a hexapod robot [19].

III. Mathematical Model

In the model of the beam (Fig 3), the beam position is measured as an angle φ from horizontal, and the ball position is measured as an angle θ from the centre of the beam. The full derivation for the mathematical model has previously been described by the authors [20]. The final equations for the ball acceleration are given in equations (1) and (2).

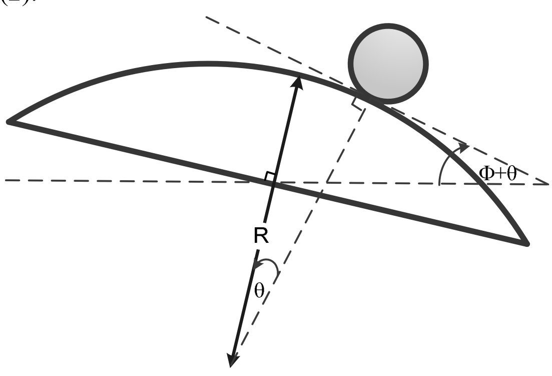

Fig 3. The ball and beam showing the relationships between the angles and motion.

$$\ddot{\theta} = A(\theta + \emptyset) \quad (1)$$

$$A = \frac{g}{R(1+\frac{I}{mr^2})} \quad (2)$$

Where

g - gravitational acceleration
I – moment of inertia of the ball
R - radius of curvature of the beam
m - mass of the ball
r - radius of the ball
θ - ball position (angle from the centre)
Ø - beam position (angle from horizontal)
x – ball position
v – ball velocity
b – beam position
a - acceleration of the ball

From physical experimentation on the beam, the value for acceleration (a) of the ball was determined as a factor of the ball position (x) and beam position (b) in equation (3).

Placing this into the mechanical modeling we can determine the new position of the ball, depending on its current position, velocity (v) and acceleration in equation (4), and the new speed of the ball dependant on its current speed and acceleration in equation (5). The simulation was set to a time period of 1 ms in equations (6) and (7).

$$a = 12x + 2.8b \quad (3)$$

$$x_{new} = x + vt + \frac{at^2}{2} \quad (4)$$

$$v_{new} = v + at \quad (5)$$

$$x_{new} = x + \frac{v}{10^3} + \frac{12x+2.8b}{2\text{x}10^6} \quad (6)$$

$$v_{new} = v + \frac{12x+2.8b}{10^3} \quad (7)$$

IV. Genetic Algorithm

Evolutionary computation is an optimization process that autonomously searches through a sequence of possible solutions to a problem to find a solution that will adequately solve the problem. It is modelled on Darwinian evolution, 'survival of the fittest', where a population of solutions is evolved. Each solution is evaluated and given a fitness, and the solutions with a higher fitness are retained and used to create new solutions. These solutions are often referred to as individuals or chromosomes, and can be in many forms depending on the problem to be solved. A group of solutions is called a population. There are several forms of evolutionary computation, one of which is the GA.

The GA is a repetitive process with three parts including a) reproduction, where the genetic operators crossover and mutation are used to generate new individuals from the surviving population of individuals, b) fitness evaluation, which determines how well each individual within the population performs, and c) selection, which is the process that determines which individuals within the population (based on their fitness) will survive to the next generation.

A. Chromosome

The chromosome that was used for the ball and beam controller was a three dimensional LUT. The LUT's axial co-ordinates connected to the current ball-beam states of ball position (nineteen positions), beam position (ten positions) and ball speed (three positions). The parameter at each co-ordinate within the LUT was the desired motor speed and direction required to move the beam into a position that would maintain the balance of the ball (Fig 4).

The LUT was used to control the beam's motor depending on the beam states. This was achieved by connecting the simulation's current ball-beam states to the axis of the LUT. The parameter at that location was sent back to the simulation to control the simulation's motor

ISBN: 978-988-18210-9-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

speed and direction. The parameters within the LUT were eleven discrete values ranging from 0 to 250, in steps of 25 giving a maximum of eleven motor speeds. If less speeds were required, these numbers were then broken into ranges, i.e. for two motor speeds, values less than 128 the motor was reversed, while numbers greater than 128 the motor was forward.

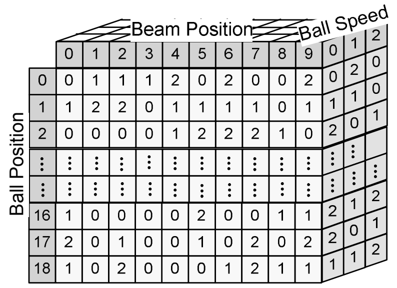

Fig 4. Three dimensional LUT showing 19 ball positions, 10 beam positions, 3 ball speeds.

The search space of a chromosome is the total number of combinations that the chromosome can have. The fitness landscape is the fitness level of each one of these chromosomes. In this study, the search space within the three dimensional LUT was dependant on the number of locations within the LUT, and the number of speeds that were employed at each location. The experiments were repeated with four ranges of motor speeds. These were two (left and right), three (left, stopped and right), five (two left, stopped and two right) and eleven speeds (five left, stopped and five right).

The total search space that the GA was required to search through was calculated using equation [8] and illustrated in Table I. It can be seen that the search space rapidly increased as the number of speeds increased. The exponent 570 was derived from the size of the LUT (19x10x3).

$$Search\ space = speeds^{\text{size of LUT}} = speeds^{570} \qquad [8]$$

TABLE I SEARCH SPACE WITHIN THE LUT DEPENDENT ON THE NUMBER OF MOTOR SPEEDS.

Speeds	Search Space
2	3.9×10^{171}
3	9.1×10^{271}
5	2.6×10^{398}
11	3.9×10^{593}

B. Reproduction and Selection

The aim of the reproduction and selection scheme is to provide a high selection pressure, i.e. to move rapidly up the fitness landscape whilst maintaining population diversity. This is dependent on both the selection scheme and the reproduction method employed.

The reproduction method used two point crossover with a mutation rate of 2%. Two point crossover selects two random points within both parent's chromosomes and transposes the chromosome information at these points to create two new individuals. In this case the crossover points were selected only on the ball position axis (Fig 5).

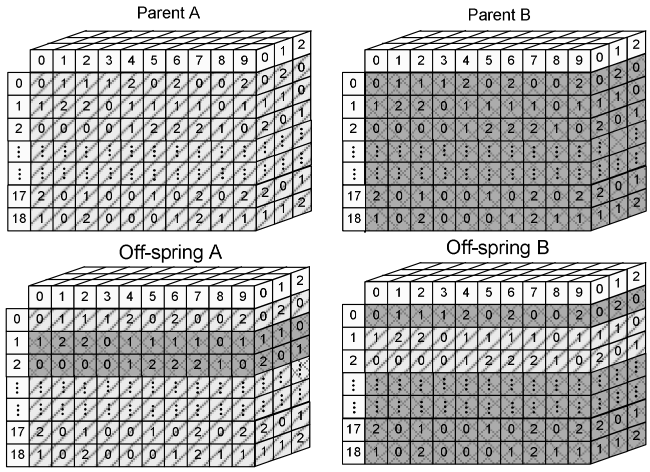

Fig 5. Two point crossover cut along the ball position axis.

There are many selection schemes that can be used within a GA, such as roulette, rank based, or tournament, with each method having its advantages and disadvantages. The selection process used in this GA was tournament, where the population was divided into groups. One individual within that group was selected for reproduction, depending on its fitness compared to the others within that group. The larger the group size, the higher the selection pressure; however with a larger group size diversity could be quickly lost. In this GA, a group size of two individuals was used.

C. Fitness Criteria

The fitness was determined by how long the ball remained balanced on the beam before hitting either end-stop. At the start of each test, the beam was placed in the horizontal position and the ball was at rest. The simulation was then run until either the ball hit an end-stop or 60 seconds had passed. Each individual was tested seven times with the ball positioned at seven different locations on the beam, giving a total maximum fitness of 420 seconds.

V. SIMULATION

The simulation used the equations as shown in equations [6] and [7]. The simulation was modeled on a 1ms time period. A new ball position and speed was calculated every 1ms time period. Correspondingly the beam movement was calculated over a similar period. The maximum beam movement was calculated from the real beam system, using two maximum motor speeds of 125 and 250 pulses per second, or a beam angular velocity of 22.7 and 45.4 degrees per second. The motor speed and direction was feed into the simulation which was used to calculate the new beam position. The new ball speed and position for the next 1ms was then calculated and fed back to the LUT. The 1ms time period of the simulation was used to give the real time that the ball was in motion.

ISBN: 978-988-18210-9-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

VI. Graphical User Interface

The GUI (Fig 6) provided simple control of the GA and allowed the maximum fitness, average fitness, and duration of the simulation time to be recorded. The graphics of the beam could be physically turned on or off allowing the motion of the ball and beam to be observed. However when the graphics were enabled, the graphical drawing of the ball and beam on the display severely slowed down the computer program, therefore this feature was usually disabled.

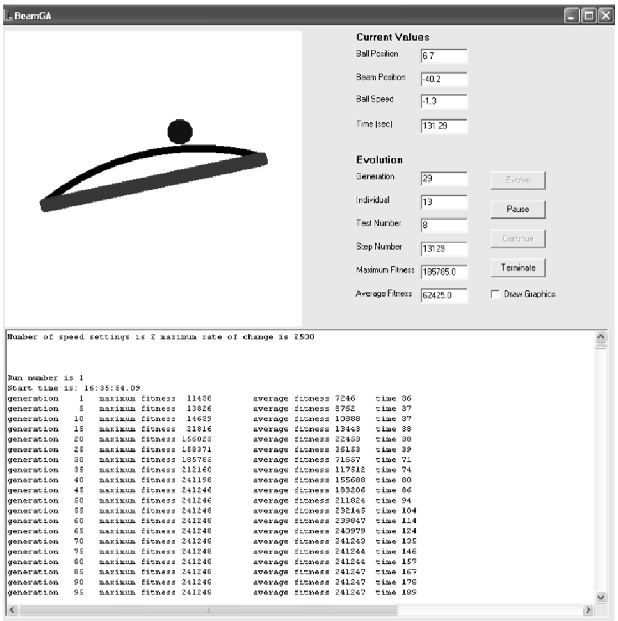

Fig 6. The graphical user interface

VII. Results

Initial experiments used a two dimensional LUT which used only the beam and ball positions. It was found that this information alone was not enough to provide a successful evolution. The LUT was modified to provide for a third parameter incorporating speed.

Two ranges of experiments were performed with two maximum stepper motor pulse rates. The first used 125 pulses per second which equated to a maximum beam angular velocity of 22.7 degrees per second. This is the speed of the actual beam motor and was at the limit at which the beam could control the ball. The second was 250 pulses per second which equated to a maximum beam angular velocity of 45.4 degrees per second. For each experiment, four ranges of motor speeds (two, three, five and eleven speeds) were evaluated.

The first experiments used eleven start positions lying between ±18 degrees from the top of the beam. The fitness level for these experiments never reached the maximum fitness. Under investigation it was found that the motor was not fast enough to prevent those balls starting at the extremes from hitting an end-stop. The experiment was changed to seven ball start positions lying between ±12 degrees from the top of the beam, with each individual tested seven times at each start position. A successful run occurred when the ball was balanced for 60 seconds; giving a maximum fitness of 420 seconds.

A. *Evolved motion of the ball.*

The graphs presented in figures 7 to 10 show the relationship between the fitness of the best individual within the population and the number of generations for the four ranges of motor speeds.

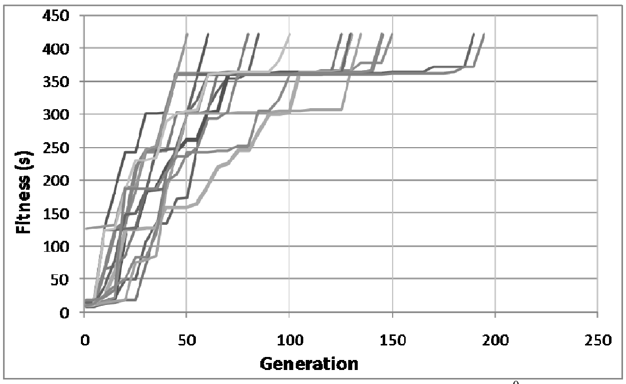

Fig 7. Two motor speeds, maximum angular velocity of 22.7^0 per second

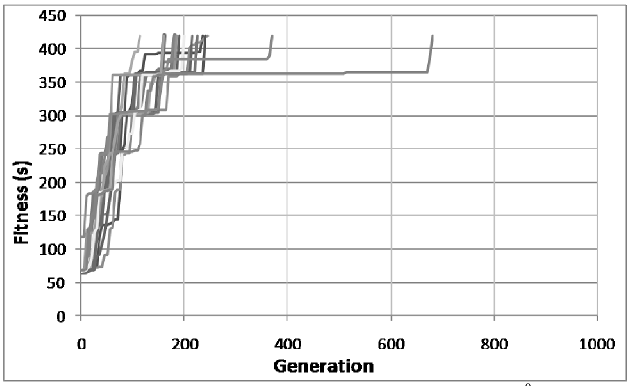

Fig 8. Three motor speeds, maximum angular velocity of 22.7^0 per second

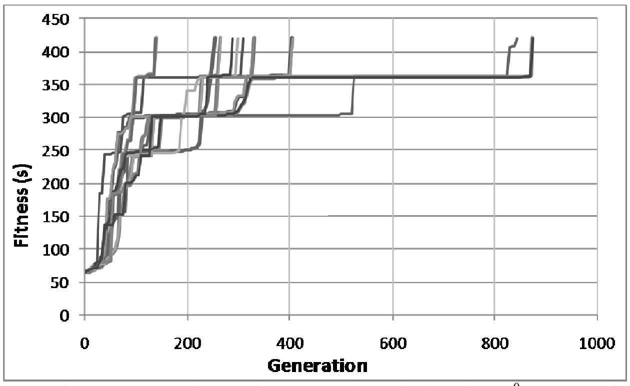

Fig 9. Five motor speeds, maximum angular velocity of 22.7^0 per second

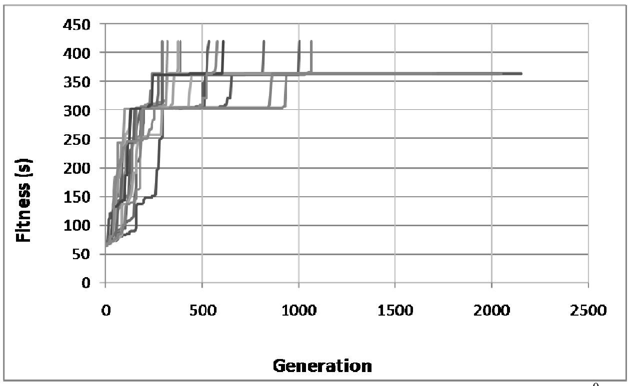

Fig 10. Eleven motor speeds, maximum angular velocity of 22.7^0 per second

ISBN: 978-988-18210-9-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The graphs show a step function in the fitness level as the evolution had to independently evolve seven start positions. It can be seen that the experiments with two motor speeds evolved to successful solutions in less generations and time than the other speeds.

The motion of the ball and beam was observed in the various stages of the evolutionary process using the graphical display. In the first stage, the ball would roll towards the beam end-stops with little reaction from the beam itself. In the next stage the beam would react to the ball movement, reversing the motion of the ball; however the ball would then roll to the opposite end-stop. During the following stage in the process, the beam moved in an oscillating pattern, causing the ball to stay balanced in between two points. However after five to ten seconds the ball would break free and gather too much speed for the beam to prevent the ball from hitting an end stop. In the final stage of evolution, the beam was able to keep the ball trapped between two points for the full 60 seconds.

This characteristic oscillation of the beam was seen in all motor speed ranges. With only two speeds, the beam moved in rapid oscillations to keep the ball steady. However with a larger number of speeds, the beam would move at a slower pace. Eventually the beam evolved to keep the ball motionless, with the use of an oscillation pattern for all seven start positions.

It can be seen from the graphs that there are two main plateaus in the fitness level near 320 and 360 seconds. These plateaus can be explained by the two start positions at the furthest point from the center of the beam. These are the most difficult points bring the ball to a stable oscillating condition, as the ball tends to gather a high speed and is difficult to capture. This plateau was more noticeable in the five and eleven motor speeds.

For the five and eleven motor speed range, the ball would not be balanced in the middle of the beam. Instead it would be gently moved to either end of the beam, and kept centered around that point. This trait can be explained by the placement of the ball sensors on the beam. When the beam was designed, the ball sensors were unevenly spaced with the sensors placed closer together at the ends of the beam and further apart in the middle of the beam. This was because it was thought that determining the ball's position and speed was more critical near the beam ends. Unintentionally however, this gave the evolved controller the best location of the ball and its speed near either end of the beam. Subsequently the evolved controller used the end locations to balance the ball. This characteristic was not seen with the two and three motor speeds experiments.

Because a simulation was used, when a test was started with the ball motionless in the center of the upright beam, the evolved solution kept the motor off, so the ball stayed perfectly balanced for the duration of the test. In the case of the two speeds as the motor could not be stopped, it would move the ball to a stable position.

B. Evolved chromosome

An investigation of successfully evolved chromosomes and the corresponding sequence of beam and ball motions showed different patterns for each evolved chromosome. This is because there were multiple ways of successfully balancing a ball. A successful evolution did not use a large part of the parameters in the LUT, especially at the extreme values of beam and ball positions. The ball simply tracked to a position on the beam, and beam oscillations around that point kept it in place.

A comparison of the maximum and average fitness (Fig 11) shows the maximum fitness increased in steps with the average fitness converging when the maximum fitness reached a plateau. At each plateau it was thought that as all the population had the same fitness, the population diversity had been lost. However an investigation of each chromosome revealed that this was not the case. This was backed by observation of the beam and ball motion at the plateau points. The evolution produced multiple solutions, although no individual chromosome had found a solution that would balance the ball when started in either, or both its first and last start position. Eventually this solution was found and the evolution was completed.

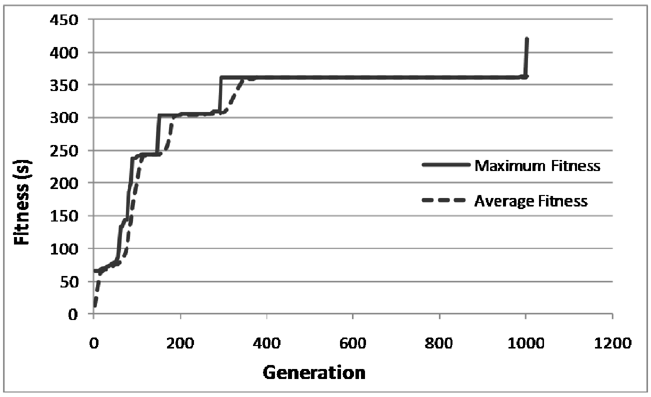

Fig 11. Comparison between the maximum and average fitness, with eleven motor speeds and a maximum angular velocity of 22.7^0/sec

C. Comparison of two maximum motor speeds

Several hundred experiments were performed on both maximum motor pulse rates and speed ranges. Table II provides a comparison of these results showing the average fitness, number of generations and time the evolution was in progress at the end of the evolution. From this table it can be seen that the faster motor and minimum number of motor speeds had the best results in terms of the number of generations and the time taken to come to a successful evolution. It was noted that the time taken for the five and eleven motor speeds to successfully evolve was also acceptable despite the much larger search space. This was due to the constrained motion of the beam and the path that the ball took, with only a limited part of the chromosome being used for the beam control.

TABLE II COMPARISON OF THE AVERAGE FITNESS, AVERAGE NUMBER OF GENERATIONS AND THE AVERAGE TIME TAKEN TO EVOLVE

22.7 degrees/second			45.4 degrees/second		
Generation	Av fitness	Time (s)	Generation	Av fitness	Time (s)
118	347726	197	42	268456	35
268	364240	592	56	327891	76
398	357240	3624	98	351811	297
861	359427	25794	103	349563	467

ISBN: 978-988-18210-9-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

A comparison of the four motor speeds within each maximum motor pulse rate is shown in figures 11 and 12.

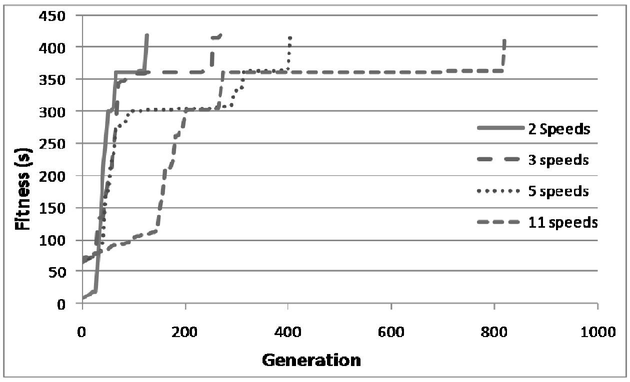

Fig 12. Four motor speeds with maximum beam angular velocity of 22.7^0 per second

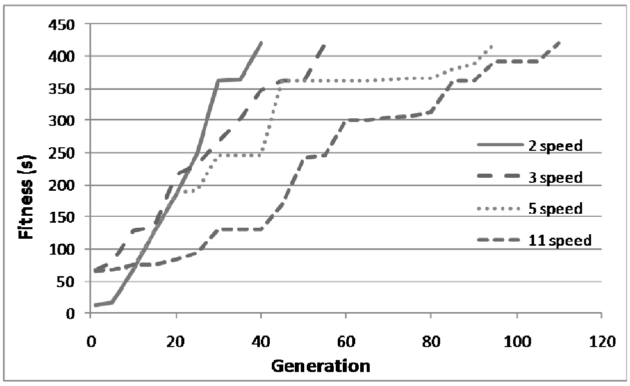

Fig 13. Four motor speeds with maximum beam angular velocity of 45.4^0 per second

From these graphs it can be seen that doubling the motor pulse rate had a significant improvement on the ability of the system to evolve, especially at the five and eleven speed range. The fitness plateau at 320 and 360 seconds can clearly be seen. All the solutions had difficulty with either one or both of the extreme starting points.

VIII. Conclusion

It has been demonstrated that a robotic controller for a ball and beam system based on a three dimensional lookup table can be evolved. While both motor pulse rates and all motor speed ranges were capable of being evolved to keep a ball balanced more than five minutes, the best evolutionary performance was achieved using a limited number of motor speeds and a higher motor pulse rate.

References

[1] C. Ka and L. Nan, "A ball balancing demonstration of optimal and disturbance-accomodating control," *Control Systems Magazine, IEEE,* vol. 7, pp. 54-57, 1987.

[2] F. G.-E. F.Gordillo, R.Ortega, and J.Aracil, "On the ball and beam problem: regulation with guaranteed transient performance and tracking periodic orbits," presented at the Proc of the International Symposium on Mathematical Theory of Networks and Systems, University of Notre Dame, IN, USA, 2002.

[3] E. P. Dadios, R. Baylon, R. De Guzman, A. Florentino, R. M. Lee, and Z. Zulueta, "Vision guided ball-beam balancing system using fuzzy logic," in *Industrial Electronics Society, 2000. IECON 2000. 26th Annual Confjerence of the IEEE*, 2000, pp. 1973-1978 vol.3.

[4] J. Iqbal, M. A. Khan, S. Tarar, M. Khan, and Z. Sabahat, "Implementing ball balancing beam using digital image processing and fuzzy logic," in *Electrical and Computer Engineering, 2005. Canadian Conference on*, 2005, pp. 2241-2244.

[5] K. C. Ng and M. M. Trivedi, "Neural integrated fuzzy controller (NiF-T) and real-time implementation of a ball balancing beam (BBB)," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, 1996, pp. 1590-1595 vol.2.

[6] P. H. Eaton, D. V. Prokhorov, and D. C. Wunsch, II, "Neurocontroller alternatives for ;fuzzy; ball-and-beam systems with nonuniform nonlinear friction," *Neural Networks, IEEE Transactions on,* vol. 11, pp. 423-435, 2000.

[7] A. G. B. Tettamanzi, "An evolutionary algorithm for fuzzy controller synthesis and optimization," in *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, 1995, pp. 4021-4026 vol.5.

[8] K. Sung Hoe, P. Chongkug, and F. Harashima, "A self-organized fuzzy controller for wheeled mobile robot using an evolutionary algorithm," *Industrial Electronics, IEEE Transactions on,* vol. 48, pp. 467-474, 2001.

[9] R. Bianco and S. Nolfi, *Evolving the Neural Controller for a Robotic Arm Able to Grasp Objects on the Basis of Tactile Sensors*, 2829 ed., 2003.

[10] K.-J. Kim and S.-B. Cho, "Dynamic selection of evolved neural controllers for higher behaviors of mobile robot," in *Computational Intelligence in Robotics and Automation, 2001. Proceedings 2001 IEEE International Symposium on*, 2001, pp. 467-472.

[11] Z. Yi and Y. Xiuxia, "Design for beam-balanced system controller based on chaos genetic algorithm," in *Information Acquisition, 2004. Proceedings. International Conference on*, 2004, pp. 448-451.

[12] G. K. M. Pedersen and M. V. Butz, "Evolving robust controller parameters using covariance matrix adaptation," presented at the Proceedings of the 12th annual conference on Genetic and evolutionary computation, Portland, Oregon, USA, 2010.

[13] H. H. Lund and J. Hallam, "Evolving sufficient robot controllers," in *Evolutionary Computation, 1997., IEEE International Conference on*, 1997, pp. 495-499.

[14] H. H. Lund, "Co-evolving Control and Morphology with LEGO Robots.," in *Proceedings of Workshop on Morpho-functional Machines*, 2001.

[15] A. Chavoya and Y. Duthen, "Using a genetic algorithm to evolve cellular automata for 2D/3D computational development," presented at the Proceedings of the 8th annual conference on Genetic and evolutionary computation, Seattle, Washington, USA, 2006.

[16] G. Greenfield, "Evolved Look-Up Tables for Simulated DNA Controlled Robots," presented at the Proceedings of the 7th International Conference on Simulated Evolution and Learning, Melbourne, Australia, 2008.

[17] Y. Z. Renato A. Krohling, and Andy M. Tyrrell, "Evolving FPGA-based robot controllers using an evolutionary algorithm," 2002.

[18] M. Beckerleg and J. Collins, "An Analysis of the Chromosome Generated by a Genetic Algorithm Used to Create a Controller for a Mobile Inverted Pendulum," *Studies in Computational Intelligence,* vol. 76, 2007.

[19] J. Currie, M. Beckerleg, and J. Collins, "Software Evolution Of A Hexapod Robot Walking Gait," in *Mechatronics and Machine Vision in Practice, 2008. M2VIP 2008. 15th International Conference on*, 2008, pp. 305-310.

[20] M. Beckerleg and J. Collins, "Evolving Electronic Circuits for Robotic Control.," presented at the 15th International Conference on Mechatronics and Machine Vision in Practice, Auckland, New Zealand, 2008.