Design and Development of an Intelligent Controller for a Pole-balancing Robot

Pedram Masajedi[1*], Tohid Yeganeh[2], Mehrdad Fouj Lali[3], Mehran Mohammadi[4], Saeed Hosseinkhah Soorkoohi[5]

[1.] MSc of Applied Design Engineering, Control and Robotics, Shahid Chamran University
[2.] BS in Accurate Instruments Engineering of Petroleum Industry, member of Young Researchers Club, Islamic Azad University of Urmia
[3.] Member of Young Researchers Club, Islamic Azad University of Urmia
[4.] Member of Young Researchers Club, Islamic Azad University of Urmia
[5.] Mechanical Engineer of Iran's Renewable Energy Investigation Office
pedram.masajedi@gmail.com

Abstract: In this paper, an intelligent controller capable of static balancing as well as dynamic balancing of a pole mounted amotorized robot is designed and developed. The brain of the intelligent controller lies in the Fuzzy Inference System, which receives as its input displacement, velocity and acceleration information. An embedded instrumentation system onboard the robot\ measures the displacement of the robot and the angle of inclination of the pole from the vertical position. For static balancing, the controller needs to maintain the pole in an upright position while the robot is free to move on a flat surface. For dynamic balancing, the robot needs to balance the pole while performing transitions up and down a ramp. Furthermore, the robot needs to steer itself back to the center to prevent it from falling off the ramp.

[Pedram Masajedi, Tohid Yeganeh, Mehrdad Fouj Lali, Mehran Mohammadi, Saeed Hosseinkhah Soorkoohi. **Design and Development of an Intelligent Controller for a Pole-balancing Robot.** *J Am Sci* 2012;8(8):555-562]. (ISSN: 1545-1003). http://www.jofamericanscience.org. 85

Keywords: Micro-controller; Fuzzy inference system; Pole-balancing robot

1. Introduction

The inverted pendulum model has proven to be an important precursor to more complex work on active balancing, particularly in the field of legged machines and locomotion studies for robotics. The first recorded attempt for this fundamental problem was by Ref. [1]. He used the parts from an erector set to build a machine that balanced an inverted pendulum on top of a small powered truck (Fig. 1). The truck drove back and forth in response to tipping movements of the pendulum as sensed by a pair of switches at its base. In order to move from one place to another, the truck first had to drive it away from the goal to unbalance the pendulum toward the goal. In order to balance again at the destination, the truck moved past the destination until the pendulum was again upright with no forward velocity. It then moved back to the goal. The inverted pendulum model would become the primary tool for studying balancing in legged systems and the importance of active balancing in legged locomotion has been widely recognized since 1938, see Refs. [2–12]. However, progress in building physical legged systems that employ such principles was slowed down by perceived difficulty of the task. It was not until the late 1970s that experimental work on balancing for legged systems gained momentum Ref. [13]. The major drawback of the previous work is that the algorithms are very complicated Ref. [13]. In this paper, a new algorithm for balancing the pole is presented. We show that by using a three-dimensional Fuzzy Logic Controller, the controller is capable of performing static and dynamic balancing with high consistency and mrepeatability.

2. Static balancing

2.1. Design and development of robot hardware

2.1.1. Mechanical design

The robot was designed to support an inverted pendulum that is free to swing about a horizontal axis with one degree of freedom. The inverted pendulum was secured on a polesupport mechanism and balanced by moving the pivoted support parallel to the plane of swing. Other than the robot’s wheels, no parts of the body touch the balancing platform. On the pole-support mechanism, ball bearings were used in the axle of rotation supporting the pole. A potentiometer was used to feedback the rotating pole angle. An optical encoder was mounted on one side of the front wheel instead of the back driving wheels in order to reduce skidding errors. The robot is driven by a DC motor which receives commands from the FIS (Fig. 2).

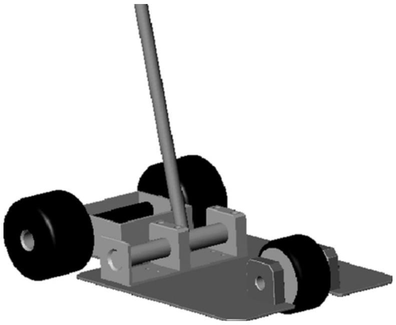


Fig. 1. CAD design of first prototype for static balancing.

The hardware of the robot consists of the following main modules:
1. DC Motor
2. Motor and wheel holders
3. Robot base plate
4. Pole pivot bracket
5. Potentiometer to feedback the pole angle
6. Optical encoder to feedback the robot position
7. Transceiver board with Intel 8751 micro-controller

The intelligence of the robot lies in the computer, which communicates with the Intel 8751 micro-controller reading sensory feedback signals and driving the motor. The algorithm on the computer is designed to control the robot throughout the duration of its operation without human intervention (Fig. 3).

2.1.2. PCB design of embedded 8751 system

The PCB was designed to provide an interface for capturing and processing readings of the optical encoder (16 bits) and potentiometer (12 bits). The data are sent via the serial communication link to the computer where they are processed to determine the required direction and magnitude of the force to drive the motor. The output data generated by the computer is the pulse-width-modulation (PWM) signal for the DC Motor.

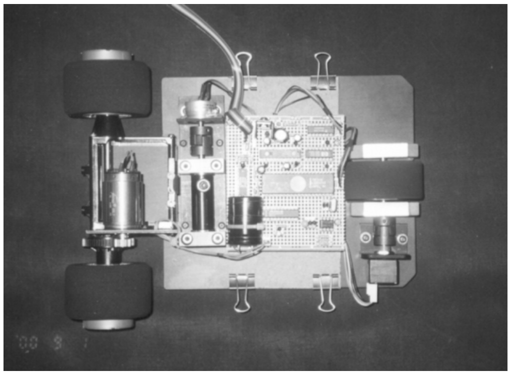


Fig. 2. Fabricated robot structure with the wire-wrapped PCB.

The components of the embedded system are:
1. Serial port communication
2. Optical encoder feedback circuitry
3. Potentiometer feedback circuitry
4. Motor driver
5. Interrupt generator
6. Micro-controller to handle individual sub-modules (Fig. 4)

2.2. Design and development of robot software

2.2.1. Proposed control strategy

The control system for the pole-balancing robot is separated into two main parts, namely the q/dq/ddq System and the x/dx/ddx System. The basic block diagram of the proposed control strategy is shown in Fig. 5. The three key parameters are the displacement, velocity and acceleration of the pole angle and the robot position. The flowchart of the 87C51/2 micro-controller routine is shown in Fig. 6.

2.2.2. q/dq/ddq System

This system is the primary control system to keep the pole upright at all times. The inputs are q, the pole angle with respect to the vertical position, dq, the velocity of the pole movement and ddq, the acceleration of the pole movement. Taking the sampling time into consideration, we can incorporate dq from q by using the relationship between distance, speed and time. This additional information enables the system to predict whether the pole swing is slowing down or vice versa. This is similar to adding derivative control in the classical PID controller. This additional input to the fuzzy system makes the system more intelligent. The same reasoning applies to ddq.

2.2.3. x/dx/ddx System

Besides trying to balance the pole, it is desired to maintain the balancing action at a particular position. This is done by the x/dx/ddx System. The inputs are x, robot displacement with respect to a predefined position, dx, velocity of the robot and ddx, acceleration of the robot. The output of the x/dx/ddx System is the required offset angle to compensate for the system so that the motor will return to the desired balancing point on the platform. Similarly, the additional parameters dx and ddx will provide anticipatory actions like what the q/dq/ddq System has achieved with dq and ddq.

2.2.4. Incorporation of fuzzy logic

The potentiometer and encoder readings are fuzzified before being read by the FIS, which is implemented using C programming language. A total of five membership functions for each variable are created and this results in 125 rules in either the q/dq/ddq System or the x/dx/ddx System. Implementation of the q/dq/ddq System is similar to that of the x/dx/ddx System except that the variables are different.

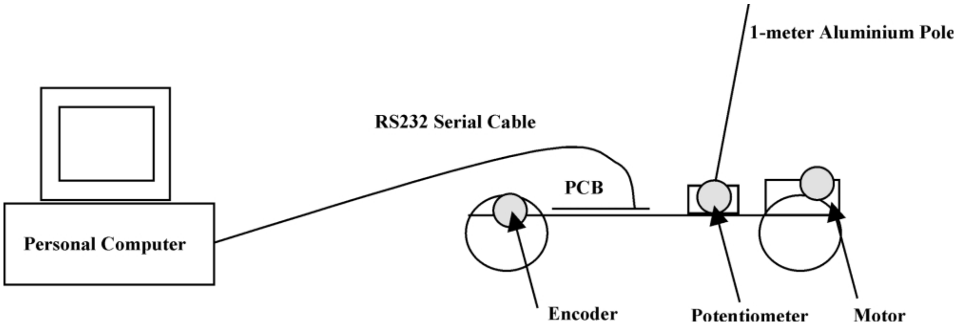


Fig. 3. Setup of the entire pole balancing system

Membership functions of the fuzzy engine are shown in. The above block diagram (Fig. 7) applies to the q/dq/ ddq System as well as the x/dx/ddx System. The maxproduct- COA (overlapping) defuzzifying technique was used to compute thePWMoutput of the robot.

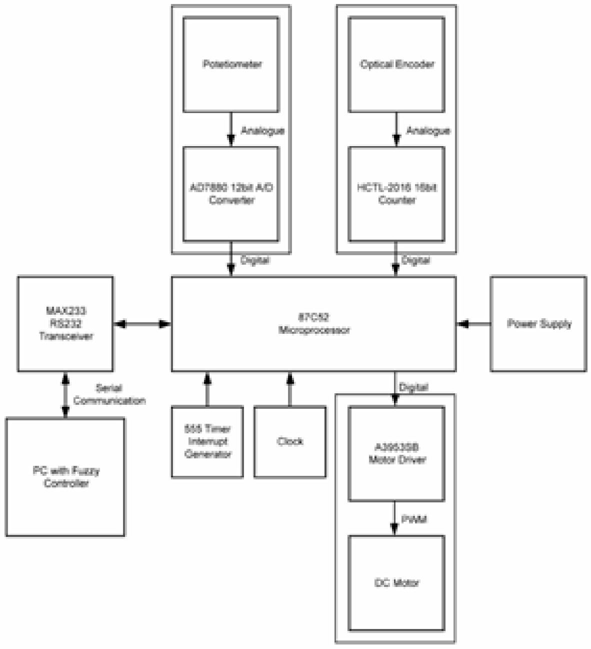


Fig. 4. Block diagram of the embedded 87C51/2 system.

The following flowchart explains the working principle of a Fuzzy Inferencing Engine (Fig. 8). As illustrated in Fig. 9, the inverted pendulum consists of an arm attached to the motor at a particular point. The arm rotates in the plane OXY, and the motor moves along the X-axis forward and backward under the influence of a force. The input parameters are the angle, u and the velocity, du. The output parameter is the force. The control objective is the following: for any pair (u, du), which specifies the position and velocity of the system, find and apply a corresponding force to the robot such that the pole is balanced, i.e. does not fall over the motor. The force depends on (u, du).

The Fuzzy Logic Controller is given in the form of If–Then rules as shown:
(1) If u is medium negative and du is medium negative then force is large negative
(2) If u is small negative and du is medium negative then force is large negative
(3) If u is zero and du is medium negative then force is medium negative
(4) If u is small positive and du is medium negative then force is small negative
(5) If u is medium positive and du is medium negative then force is zero
(6) If u is medium negative and du is small negative then mforce is large negative
(7) If u is small negative and du is small negative then force is medium negative
(8) If u is zero and du is small negative then force is small negative
(9) If u is small positive and du is small negative then force is zero
(10) If u is medium positive and du is small negative then force is small positive
(11) If u is medium negative and du is zero then force is medium negative
(12) If u is small negative and du is zero then force is small negative
(13) If u is zero and du is zero then force is zero

(14) If u is small positive and du is zero then force is small positive
(15) If u is medium positive and du is zero then force is medium positive
(16) If u is medium negative and du is small positive then force is small negative
(17) If u is small negative and du is small positive then force is zero
(18) If u is zero and du is small positive then force is small mpositive
(19) If u is small positive and du is small positive then force is medium positive
(20) If u is medium positive and du is small positive then force is large positive
(21) If u is medium negative and du is medium positive then force is zero
(22) If u is small negative and du is medium positive then force is small positive
(23) If u is zero and du is medium positive then force is medium positive
(24) If u is small positive and du is medium positive then force is large positive
(25) If u is medium positive and du is medium positive then force is large positive

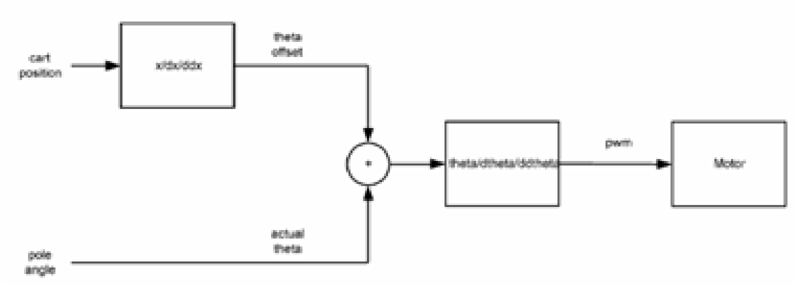


Fig. 5 Setup of the entire pole balancing system

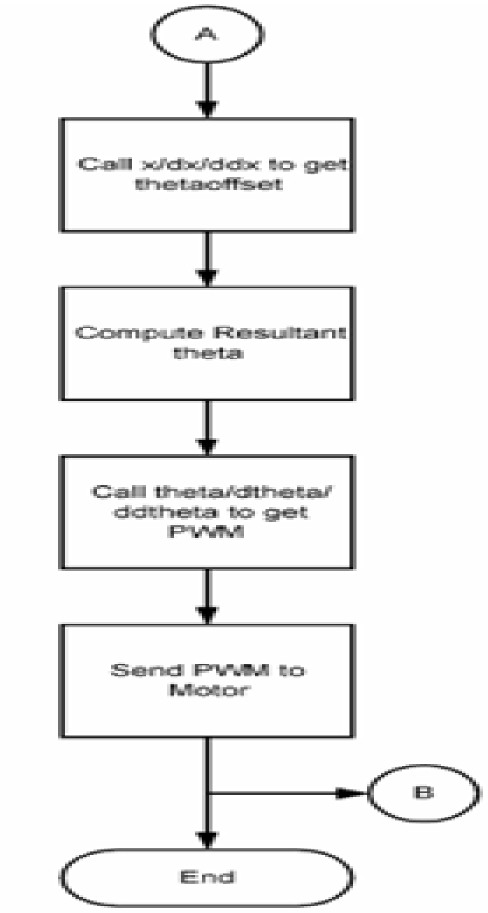
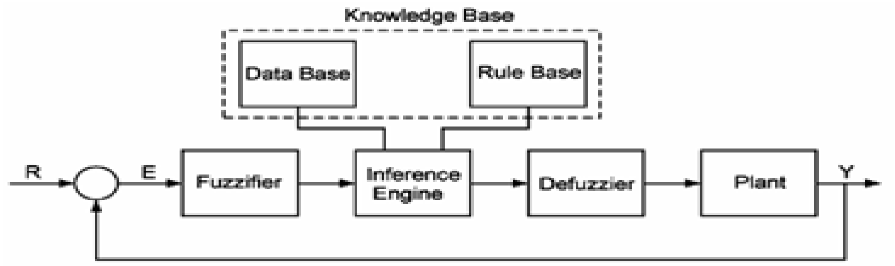

A
Call x/dx/ddx to get thetaoffset
Compute Resultant theta
Call theta/dtheta/ ddtheta to get PWM
Send PWM to Motor
B
End

Fig. 6. Flowchart of the 87C51/2 micro-controller routine



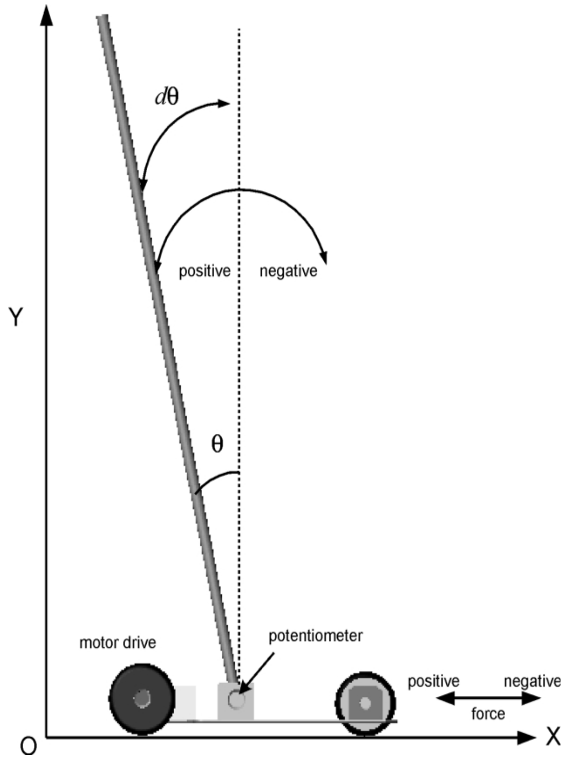


Fig. 9. Diagram showing system parameters

2.2.5. Evaluation of system performance

Real-time data taken during a successful run have been recorded and plotted using Matlab. Each variable can be analyzed individually and trends of instability in the system can be identified. Fig. 10 shows the variation of q, i.e. the pole angle from the vertical position. The working range of q in the q/dq/ ddq Fuzzy Inference Engine (FIE) is ^58. From the graph, the pole angle never exceeds ^18. It is evident that the three-dimensional rule base has improved the system reliability and accuracy significantly. Also, Fig. 11 shows the deviation of the robot position from the desired spatial point on the platform.

Here, the working range of x in the x/ dx/ddx FIE is ^5 cm. The graph shows that the robot displacement fluctuates between 3 to 27 cm, with a mean position of 22 cm. This is due to the uneven ground on which the platform is placed. Unless the controller compensates for this uncertainty, the robot will always tend to bias to one side of the balancing point. This is the main idea of balancing the robot on a slope. The variable used in the C language code is 't_slope'. This variable is added to each defuzzification membership function of both the q/dq/ddq and the x/dx/ddx FIEs. Note that there are other possible causes to this observation. These include uneven distribution of weights on the robot itself and the driving wheels positioned at the rear of the robot causing an uneven push–pull effect. Despite the above uncertainties and the fact that the robot dynamics is changing with time, the FIE is able to control the effects of these unknowns using simple tuning parameters, namely 'K_T', 'K_DT', 'K_X', 'K_DX' and other parameters that decide the range of physical inputs to be fuzzified and manipulated by the controller (Fig. 12).

The terms 'K_PWM' and 'K_THETAOFFSET' will decide the range of defuzzified output to be delivered to the next stage of the control system. For example, 'K_PWM' determines the range of effective PWM signal computed by the FIE. If it is set too high, the PWM signal computed will always be saturated at its peak. On the other hand, if it is set too low, the robot may not move at all despite the fact that there are some small variations in the PWM signal computed. The aforementioned observations are evident in the following plots of theta_offset and PWM. The working range of theta_offset varied within ^0.48 because 'K_THETAOFFSET' was set to 0.4. This argument only holds for the x/dx/ddx System because its output is added directly to the fuzzified theta input. As such, the output of the q/dq/ddq System is the scaled PWM signal which is not recorded. The plot in Fig13 shows the actual count for the PWM duty cycle. This is sent to the 8751 micro controller to produce the corresponding PWM signal. It has been observed that the fast varying PWM signal is due to high sensitivity of the q/dq/ddq System.

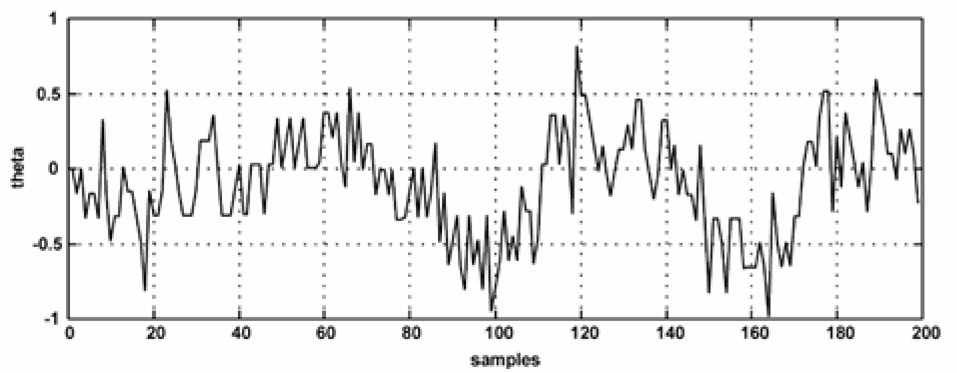


Fig10. Real-time plot of theta

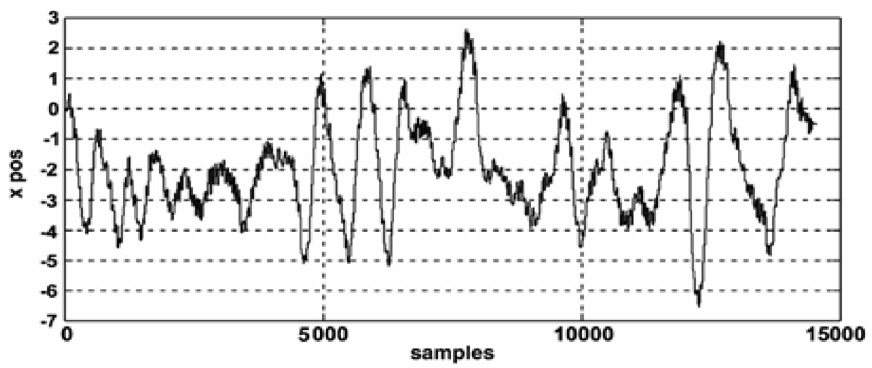


Fig. 11. Real-time plot of x

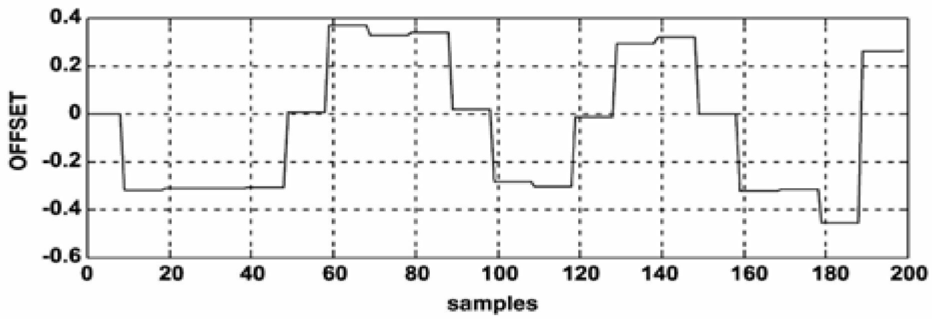


Fig. 12. Real time plot of theta_offset

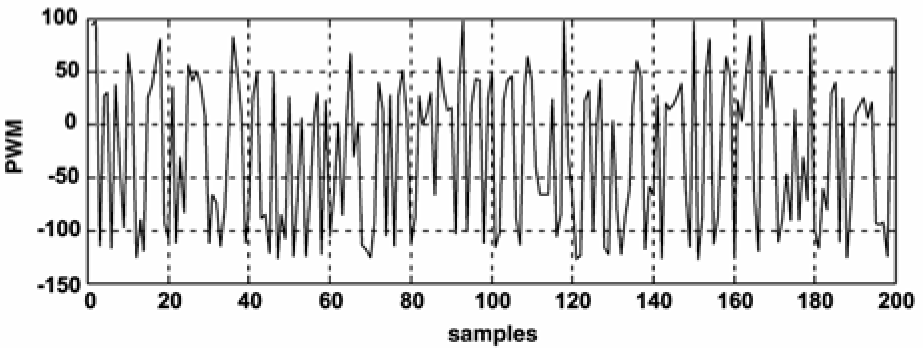


Fig. 13 Real time plot of PWM signal

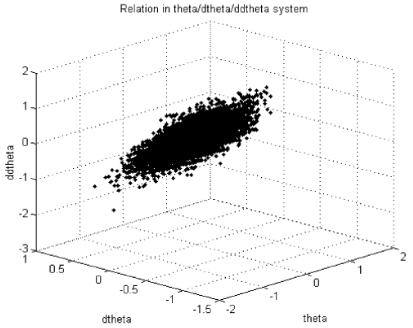


Fig. 14. Relationship of the displacement, velocity and acceleration
parameters

Undoubtedly, this causes the output of the x/dx/ddx System to be masked by the higher prioritized q/dq/ddq System. In fact, this is an inherent problem within our chosen control strategy. However, there are many reasons to stick to the current strategy. This will be explained further in the section for dynamic balancing.

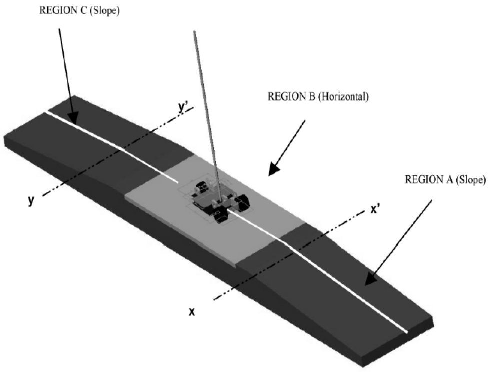


Fig. 15 Dynamic balancing platform

In addition to the two main inputs and the two main outputs of the system, i.e. theta, x_pos, theta_offset and PWM, the following plot shows how the inputs relate to their derivatives. The values of all the derivatives are relatively consistent about their means, which are near zero. This further proves the robustness of both the q/dq/ddq System and the x/dx/ddx System. One important observation here is to relate the distribution of the variables with their corresponding membership functions. This will allow accurate analysis of the effectiveness of the chosen tuning parameters. To illustrate this idea, the data of q and dq can be analyzed as shown in Fig. 14. 3. Dynamic balancing 3.1. Design and development of robot hardware Dynamic balancing of the pole will take place on the top of the balancing platform depicted in Fig. 15. The platform has a slight slope at the beginning (Region A) and the end (Region C). The surface of the platform is covered by a layer of rubber in order to increase the friction on the robot's wheels. A reflective tape is placed along the centre of the platform for the robot to sense its alignment using infrared detectors.

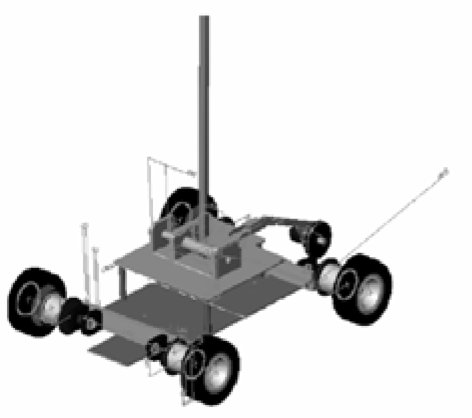


Fig. 16. CAD drawing of proposed design for dynamic balancing.

To begin with, the robot will be placed within Region A, the pole will be held in an upright position and released after the system is turned on. The robot will try to balance the pole for a minimum duration of 20 s without the pivot point of the pole crossing the line x–x0. After this initial balancing act, the robot will move across the line x–x0, through Region B, until the pivot point of the pole crosses the line y–y0. After crossing the line y–y0 the robot will retrace its initial original path, through Region B until the pivot point of the pole crosses the line x–x0 again. When the robot returns to its original position, it will be counted as one lap. During this dynamic transition across Region B, the robot need not stay for any length of time in any regions. The robot will continue to repeat these cycles within 5 min. However, for all these laps to be counted as successful cycles, the robot must perform at least 20 s of static balancing in Region A before the end of 5 min. 3.1.1. Mechanical design Based on requirements for dynamic balancing and the experience gained from static balancing, the idea of including steering in the new design was mooted. This calls for an innovative way to drive the robot such that it can steer itself in all directions. As such, two motors are used to drive the left and right wheels. Using feedback signals from the infrared sensors, the robot will be able to orientate itself back to the center to avoid falling off the platform after numerous transitions. An initial design using CAD software is shown in Fig. 16. Further details on the necessary changes following this initial idea will be described in the sequel. Photographs of the first prototype capable of dynamic balancing are shown in Figs. 17. In particular, note the changes made for the DC motor drivers. The prototype above uses direct drives for both DC motors, but it turns out that there is not enough starting torque to move the robot. As a result, the structure was modified to include gear trains that will produce a much larger torque.

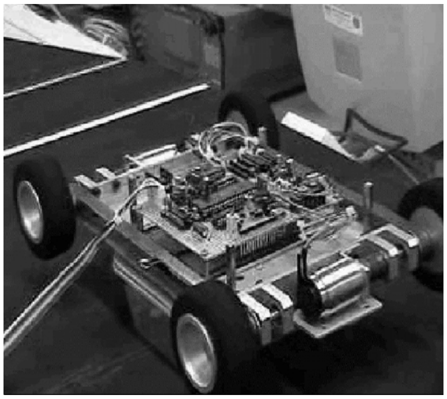


Fig17. Isometric view showing the new belt-drive system.

4. Conclusions

In this paper, an intelligent controller capable of performing static balancing and dynamic balancing of an inverted pendulum mounted on a motorized vehicle has been successfully designed and developed. Experimental studies show that the robot is capable of performing dynamic balancing as well as static balancing with great consistency and repeatability. Currently, other advanced control algorithms are being developed and will be implemented in the near future.

References

[1] B. Shannon, Personal Communication, 1985.

[2] H. Hemami, F.C. Weimer, Further considerations of the inverted pendulum, Proceedings of Fourth Iranian Conference on Electrical Engineering, Pahlavi University, Shiraz, Iran (1974) 697–708.

[3] M. Vukobratovic, Y. Sepaneko, On the stability of anthropomorphic systems, Mathematical Biosciences 14 (1972) 1–38.

[4] M. Vukobratovic, Dynamics and control of anthropomorphic active mechanisms, in: A. Morecki, G. Bianchi, K. Kedzior (Eds.), First Symposium on Theory and Practice of Robots and Manipulator Systems, Elsevier, Amsterdam, (1973), pp. 313–332.

[5] H. Hemami, C.L. Golliday Jr., The inverted pendulum and biped stability, Mathematical Biosciences 34 (1977) 95–110.

[6] T. Kato, A. Takanishi, H. Jishokawa, K. Kato, The realization of the mquasi-dynamic walking by the biped walking machine, in: A. Morecki, G. Bianchi, K. Kedzioir (Eds.), Fourth Symposium on Theory and Practice of Robots and Manipulators, Polish Scientific Publishers, Warsaw, (1983), pp. 341–351.

[7] H. Miura, I. Shimoyama, Dynamic walk of a biped, International Journal of Robotics Research 3 (1984) 60–74.

[8] J. Manter, Dynamics of quadruped walking, Journal of Experimental Biology 15 (1938) 522–539.

[9] R.B. McGhee, M.B. Kuhner, On the dynamic stability of legged locomotion systems, in: M.M. Gavrilovic, A.B. Wilson Jr. (Eds.), Advances in External Control of Human Extremities, Jugoslav Committee for Electronics and Automation, Belgrade, (1969), pp. 431–442.

[10] A.A. Frank, Automatic Control Systems for Legged Locomotion Machines, PhD Thesis, University of Southern California, Los Angeles, California, 1968.

[11] F. Gubina, H. Hemami, R.B. McGhee, On the dynamic stability of biped locomotion, IEEE Transactions on Biomedical Engineering BME-21 (1974) 102–108.

[12] V.V. Beletskii, Biped locomotion dynamics I, Izvestiya AN SSSR Mekhanika Tverdogo Tela 10 (3) (1975) 3–14.

[13] M.H. Raibert, Special Issue on legged systems, International Journal of Robotic Research 3 (1984) 75–92.

7/13/2012