

Teaching Control Using NI Starter Kit Robot

Payman Shakouri, Member IEEE, (Research student)
Gordana Collier, Member IEEE,
Andrzej Ordys, Senior Member IEEE
School of Mechanical and Automotive Engineering
Kingston University London, UK, SW15 3DW
E-mail: P.Shakouri@kingston.ac.uk

Abstract-Teaching engineering concepts using demonstrations and experiments on real hardware is always engaging and well received by students. This paper provides reference materials (both theoretical and test results), to be used in Control teaching and assessment using a laboratory experiment, with a real-time single board computer based robotic vehicle (National Instruments Robotics Starter Kit). This robotic vehicle is programmed using a graphical programming environment. The Adaptive Cruise Control (ACC) algorithm based on Proportional-Integral (PI) and Proportional – Integral – Derivative (PID) Control are deployed on a field programmable gate array (FPGA), included in the robot’s architecture. The robot model (based on a given second order transfer function) is controlled using the same method. The results obtained are compared for the simulation model and a real robot. The performance comparison demonstrates a good correlation between theory and implementation, whilst demonstrating problems and discrepancies introduced by a real system.

Keywords: *System modelling, Adaptive Cruise Control (ACC), PID Controller, Educational robotics, Real-time implementation, Single board computer, FPGA programming.*

I. INTRODUCTION

Lately, Engineering at Kingston University London has been running a project to develop new Electronics and Control teaching methods. As a result, simulation is not the tool of choice for this task any more. The applied approach to teaching and learning, using latest technology is gaining popularity and interest in the faculty.

This paper proposes an experimental method for teaching control concepts to senior undergraduate engineering students and then assessing the learning by setting a group assignment with the objective to design and implement the software to

control the performance of a robotic vehicle with on board Adaptive Cruise Control (ACC), implemented using a PI and a PID loop. Graphical programming language (LabVIEW) was used for code implementation, as it does not deter from the main learning goal by enabling a quick completion of the programming task, allowing students to focus on tuning and testing. The target hardware was a Starter Kit Robot from National Instruments; where two robots were used as ‘the leader’ and ‘the follower’. ACC [1], [2] was used to maintain a set distance between them. These experiments not only illustrate the theoretical concepts but also offer an excellent opportunity for students to programme a real-time, single board computer by directly accessing on-board FPGA, which is likely to open new opportunities for final year project implementation.

The teaching methodology, overall teaching plan (including the formal lectures on control design for distance tracking based on the classical controllers [3], hands-on exercises, and associated assessment task) are presented in Section II. The rest of the paper provides all of the information required by instructors for course delivery, including the solution of the assessment, as follows. Section III describes the hardware and associated system transfer function (given to students), while Section IV deals with the theoretical concepts required. Section V deals with system test, offering all of the results required by the instructor. Section VI discusses the design, tuning and testing of a performance of classical controllers, implemented in ACC. Overall, it verifies the correlation between theoretical expectations and real-life system performance.

II. UNDERLYING TEACHING STRATEGY

This course is aimed at senior undergraduates, who have moved on from the ‘system approach’ used in first year to more ‘discrete’ learning. In year two they study particular concepts, in order to acquire sufficient knowledge in a range of disciplines (programming, electronics and control). They can then amalgamate these concepts and work for the first time with a complex system, applying this knowledge and reinforcing understanding through implementation.

Students are given a set of formal lectures, covering relevant control concepts as shown in Table I:

TABLE I

LECTURE SCHEDULE AND MILESTONES

Week	Milestones
1	Introduction to modelling of dynamic systems
2	Use of Laplace transform and transfer functions
3	Time and feedback domain analysis
4	Frequency domain analysis
5	Introduction to PI and PID control including PID loop tuning
6	ACC and switching rules

This is followed by a set of tutorials and laboratory exercises as shown in Table II.

TABLE II

LABORATORY SCHEDULE, MILESTONES AND MODE

Week	Milestones	Mode
1	Introduction to control concepts	Individual
2	PID tuning using Matlab	Individual
3	Introduction to LabVIEW and FPGA toolbox	Individual
4	Introduction to hardware, 'walk-through' exercises (NI single board RIO including FPGA module)	Group (3)
5 - 6	Implementation and testing (work on group assignment)	Group (3)

Introduction to the software and hardware is done individually to ensure that every student gets the chance to learn the software and become equipped to work independently. However collaborative learning used in later weeks is highly beneficial when new complex concepts are to be adopted.

Student assessment task aims to:

1. Design an Adaptive cruise control algorithm using MATLAB/SIMULINK using the dynamic model of the robot (given) and obtain optimised solution using PI and PID strategy, in order to perform 'the leader' and 'the

follower' action (maintain a certain distance between the leader and follower).

2. Investigate the construction and operation of the real robot, including the incorporated sensors and actuators.
3. Investigate a programming method for a real-time embedded FPGA based system.
4. Apply the Adaptive cruise control strategy developed in 1, to a real robot.
5. Compare system performance in order to discuss the correlation between theory and implementation in terms of PID tuning.
6. Compare the system performance to discuss the correlation between the performance of the robot model and a real robot. Discuss the problems and discrepancies introduced by a real system.

In addition to the written report including the graphs (similar to the ones shown in Sections IV and V), a demo and a viva are also introduced as part of the assessment, to identify the level of learning for individuals and curb plagiarism. Additional benefits of this assessment task are development of 'soft skills' including teamwork, project management and public speaking, with the added benefit of peer support.

III. HARDWARE CHARACTERISTICS AND SYSTEM TRANSFER FUNCTION

The LabVIEW starter kit robot (Fig. 1) is based on a NI single- board RIO, with Ethernet connectivity. The robot 'follower' uses ultrasonic sensors for velocity and distance tracking of the 'leader' (2cm - 3m sensor range) and two 12V DC brushed motors, (linear velocity of approximately 0.9 [m/s]). The robots can be programmed either by using the high-level LabVIEW robotic starter kit API or by using the LabVIEW FPGA module [4].

The robot model transfer function approximated through the model identification process can be presented as:

$$\frac{v}{v_{sp}} = \frac{1}{0.48s^2 + s + 1} \quad (1)$$

where v and v_{sp} respectively denote the robot's velocity and velocity setpoint. Units are in [m/s].

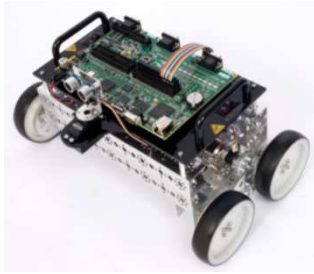


Fig.1. NI LabVIEW starter kit robot [4]

IV. CONTROL DESIGN

The Adaptive Cruise Control is based on two controllers; Proportional-Integral-Derivative (PID) control for velocity tracking, and PI control for distance tracking. The distance tracking controller calculates the velocity setpoint using the difference of the current inter-distance between the leading and following robots from the desired distance-headway. The velocity tracking controller computes the velocity setpoint, to maintain the constant desired cruising speed. ACC concept will be discussed in the next section, after tuning of the PID parameters.

A. PID parameters

The PID controller defined in the FPGA module has the configuration illustrated in Fig. 2. The controller adjusts the speed of the DC motors (in the robot) based on the given set-point speed. The gains of the PID control are set to zero by default; therefore the set-point speed will be directly sent to the DC motor. In this work, we design our controller to calculate the set-point speed which can fulfil the requirement of the ACC system.

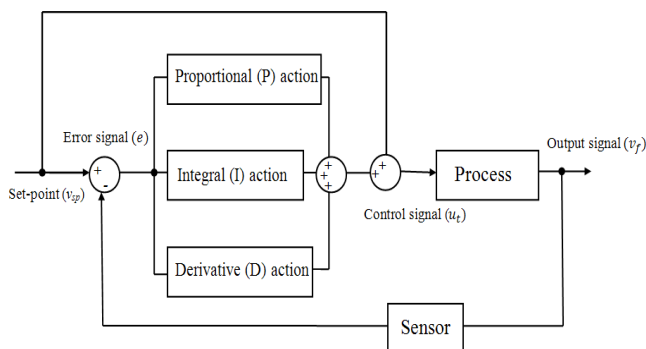
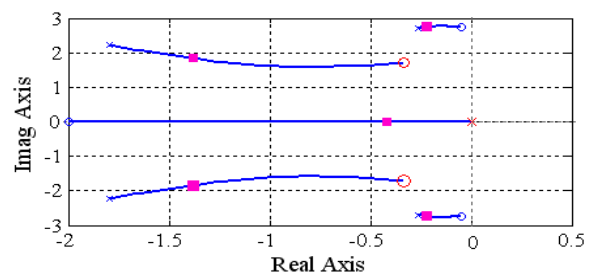
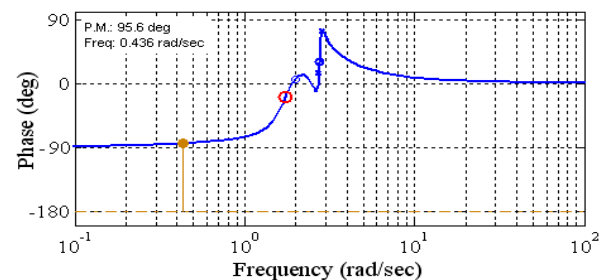
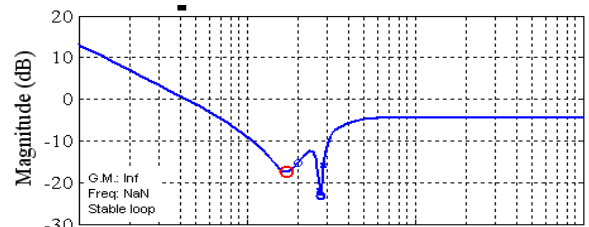


Fig. 2. Configuration of the controller including PID control in the FPFA

Different tuning rules to obtain the Proportional-Integral-Derivative (PID) control gains were introduced in literature [5]; such as, step response (frequency response method), Ziegler-Nichols etc. “Control and Estimation Tools Manager” toolbox in MATLAB/SIMULINK [10] is used, to tune the PI/PID controllers. By using this toolbox, students can easily design the controllers through the interactive plots such as root-locus, Bode, or Nichols, within the SISO design tool. The controller can be graphically tuned by manually moving, adding, or deleting poles and zeros of the model’s tunable blocks, and observing the closed loop response of the system in the analysis plots such as, step response, impulse response, pole/zero response etc. Root locus and open-loop bode diagrams resulted from the tuning process are illustrated in figure 3. These results were obtained using the Control and Estimation toolbox for a single-input-single-output system and selecting singular frequency based tuning for the tuning algorithm. PID controller gains for the three controllers, used for an ACC system, are given in Table III.



(a)



(b)

Fig. 3. (a) Root locus for open loop system, (b) Open-loop bode diagram

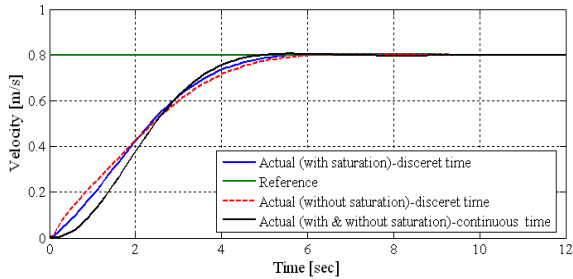
TABLE III

DISCRETE TIME PID CONTROLLERS' GAINS

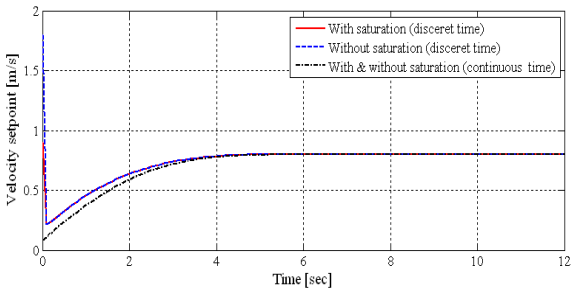
	k_p	k_i	k_d
Default PID controller	0	0	0
Distance Tracking (ACC)	0.9	0.5	0
Velocity Tracking (CC)	0.2	0.48	0.2

The closed-loop responses of the system given by Eq.1 are presented in Fig. 4. The responses of the closed-loop system in both continuous-time and discrete-time have been compared. The control action (velocity setpoint v_{sp}) has to be constrained between $[0,0.9]$, reflecting the physical limitation of the system i.e. motor velocity. The results obtained by saturating the control output have been compared to those without saturation, as illustrated in Fig. 4. Fig. 4. (b) shows that the control action initially violates the upper bound limit if it is not constrained. In most of the system, saturation can degrade the performance, especially when using the PID controller.

This is due to a phenomenon called “integrator windup” which causes a large overshoot in response. To solve this, an anti-windup can be used. Since there was not any serious issue in our design, we did not use anti-windup.



(a)



(b)

Fig.4 (a) the closed-loop responses of the system given by Eq.1 (b) using continuous-time and discrete-time PID controllers, (b) the control actions

B. ACC switching rules – distance (ACC) and velocity (CC) tracking

ACC systems operate in two different modes depending on the situation in the front - distance tracking or velocity tracking. If the ultrasonic sensor of the follower (ACC equipped) detects any obstacle, or a slower moving robot in front, the controller adjusts the velocity to maintain the clearance inter-distance (desired distance). If the inter-distance measured by the ultrasonic sensor is greater than the desired distance, it will switch to velocity tracking mode, known as cruise control (CC) mode, to track the desired velocity.

The switching logic was devised to implement automated switching between the two modes of operation for the ACC system, i.e. velocity tracking (CC) and distance tracking [3], [6], [7]. The switching rules for transition between CC and ACC modes are illustrated in Table IV.

TABLE IV

LOGICAL RULE FOR SWITCHING BETWEEN CC AND ACC (Shakouri, Ordys and Laila, et al. 2011)

	$v_f < v_{des}$	$v_f \geq v_{des}$
$d < d_{des}$	ACC	CC
$d \geq d_{des}$	CC	CC

A schematic block diagram of this switching algorithm is underlined in close-up of the switching logic CC/ACC in Fig. 5.

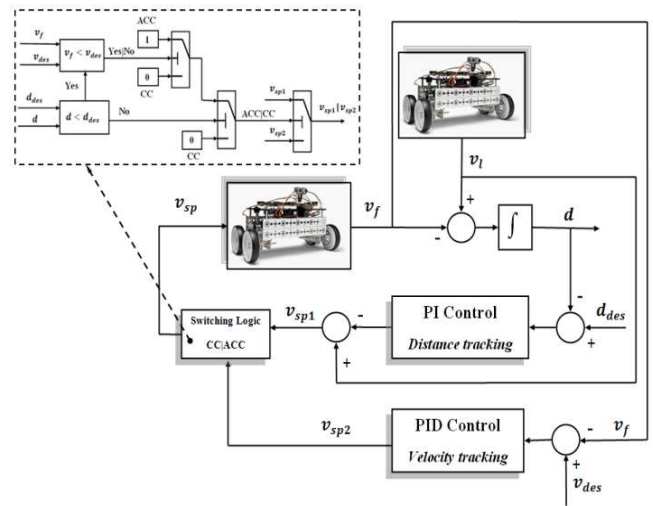


Fig. 5. A schematic block diagram of an ACC system developed in the simulation including a PI control for distance tracking and a PID control velocity tracking.

The desired headway distance d_{des} can be computed using the equation (2), assuming that the follower and leader robot have the same speed ($v_f = v_l$), known as Constant-Time Headway policy [8], [9]:

$$d_{des} = l + \underbrace{d_s}_{d_0} + T_h v_f \quad (2)$$

l - robot length, d_s - additional distance between two robots, v_f - follower robot's velocity and T_h is the constant-time headway (approximates system reaction time) [s].

Distance between real robots is measured by an ultrasonic sensor. In simulation, the distance is determined by taking the integral of their relative velocity:

$$d = \int_0^t (v_f - v_l) dt \quad (3)$$

The Schematic block diagram from Fig 5, implementing ACC in LabVIEW is shown on Fig. 6.

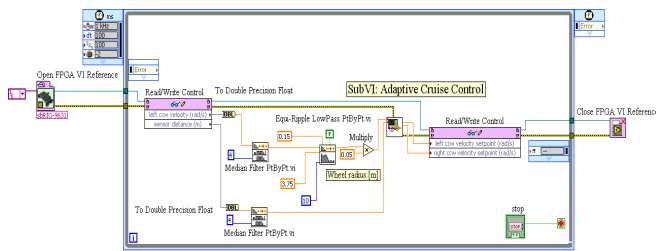
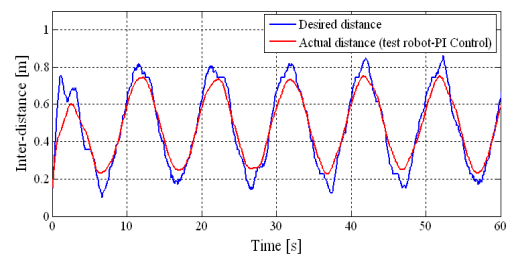


Fig. 6. The configuration of the LabVIEW block functions for implementing the ACC on a robot

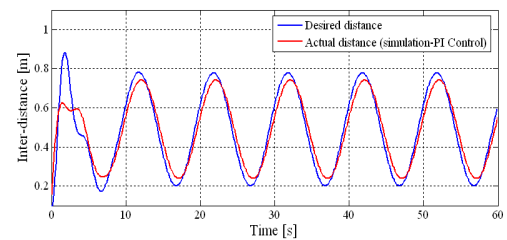
V. TESTING AND RESULTS

The ACC system was developed and tested using Matlab for PID tuning and LabVIEW for hardware implementation. The programming of the controller has been implemented in LabVIEW MathScript Node. The ACC system was designed and tested in the simulation and implemented on the robot's LabVIEW FPGA module, for comparison. In the initial experiment the leader robot is virtualised, the sinusoidal signal represents the leader robot's velocity and the distance between robots can be calculated from Eq. 8. The final stage of the test used two test robots; and the distance between them is measured by an ultrasonic sensor. The program developed in LabVIEW for real-time implementation is depicted in Fig. 6.

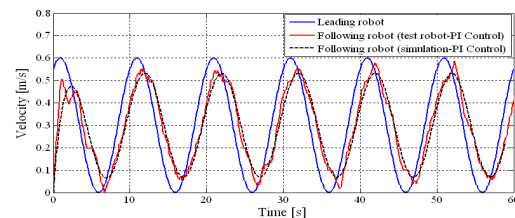
Fig. 7 illustrates the results for distance tracking (ACC) mode for the test executed by using the PID controller. The initial distance between the robots was setup at 0.1 [m]; both robots start from zero velocity. Based on the test, the minimum safe distance d_o and time-constant headway T_h were chosen to be 0.1 [m] and 1.3 [s], respectively. Some slight undulations in the results on the desired distance curve are due to the desired distance being a function of the follower robot velocity; therefore its variation depends on the velocity. However, the ACC system operates very well for distance tracking.



a)



(b)

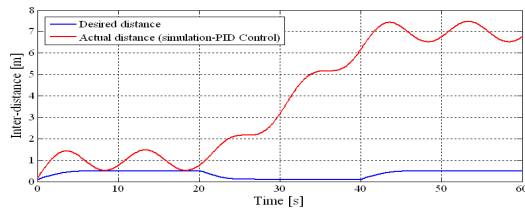


(c)

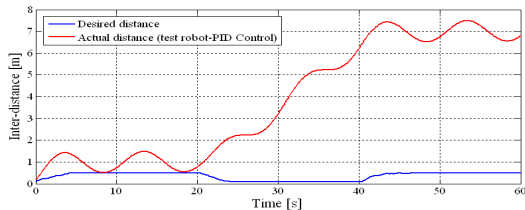
Fig. 7. Distance tracking (ACC) mode using PI control- (a) The inter-distance between the robots using test robot, (b) the inter-distance between the robots from simulation, (c) the velocities obtained during distance tracking using test robot and from simulation implementation

Fig. 8 shows the results for velocity tracking (CC) mode accomplished by PID Control with the same initial conditions. However, the desired cruising speed has been selected as a step form; from 0-20 [s], the value of the

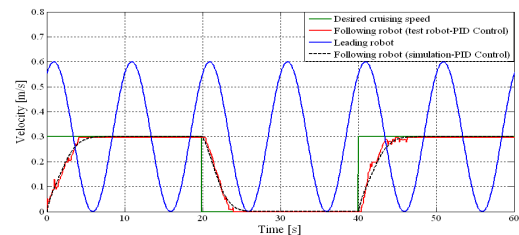
desired speed is 0.3 [m/s], its value reduces to 0 [m/s] between 20-40 [s] and then it reverts to 0.3 [m/s] from 40 [s] onward. Since the actual distance is far away from the desired distance, the ACC system operates in CC mode, so that the follower robot's velocity tracks the desired cruising speed (see Fig. 8.c), rather than tracking the desired distance. Consequently, it causes the distance between the robots to be increased (see Fig. 8.a-b).



(a)



(b)



(c)

Fig. 8. Velocity tracking (CC) using PID control- (a) The inter-distance between the robots using test robot, (b) the inter-distance between the robots from simulation implementation, (c) the velocities obtained during distance tracking using test robot and from simulation implementation

VI. CONCLUSIONS

The experiment described here delivers a range of robotics and control concepts, and control strategies, including PI and PID, PID tuning and switching strategies. It requires students to program these strategies and to implement in a real device a target real-time microcontroller board that has been configured as a robot, which is an excellent target system, capturing students' interest and imagination. This paper proposes an assessment brief for the experiment to contain:

1. System modelling and simulation.
2. Implementation in hardware.
3. Testing of the ACC system.

This paper deals with the implementation of only one type of control algorithm and there is scope to implement others e.g. fuzzy logic, or even Model Predictive Control, which can be used for post graduate level teaching. There is also potential for system hardware enhancement/expansion, to include further sensors, for example global positioning systems (GPS), inertial navigation systems (INS), camera etc. (all presently available off the shelf for the specified target board), opening a range of additional opportunities for further control feedback but also experiments and assignments in the field of digital imaging, signal processing, system fusion, robotics, autonomous driving and more.

The teaching materials described in this paper can be made available on request to the conference participants.

REFERENCES

- [1] Martinez, J. J., and C. Canudas-de-Wit. *A Safe Longitudinal Control for Adaptive Cruise Control and Stop-and-Go Scenarios*. Vol. 15. IEEE Transactions on Control Systems Technology, 2007.
- [2] Xiao, Lingyun, and Feng Gao. "A comprehensive review of the development of adaptive cruise control system." *Vehicle System Dynamics* 48: 10 (April 2010): 1167 — 1192.
- [3] Shakouri, P., A. Ordys, D. S. Laila, and M. R. Askari. "Adaptive Cruise Control System: Comparing Gain-Scheduling PI and LQ Controllers." *18th IFAC World Congress*. Milano, Italy, 2011.
- [4] *National Instrument*. 15 September 2011. <http://zone.ni.com/devzone/cda/tut/p/id/10464>.
- [5] Ogata, K. *Modern Control Engineering*. 3rd. The United State of America: Prentice-Hall, 1997.
- [6] Shakouri, P., and A. Ordys. "Application of the State-Dependent Nonlinear Model Predictive Control In Adaptive Cruise Control System." *14th International IEEE Conference on Intelligent Transportation Systems - ITSC 2011*. Washington, DC, USA, at The George Washington University, October 5-7, 2011.
- [7] Gerdes, J C, and J K Hedrick. "Vehicle Speed And Spacing Control Via Coordinated Throttle And Brake Actuation." (Control Eng. Practice) 5 (1997): 1607-1614.
- [8] van den Bleek, R. A. P. M. *Design of a Hybrid Adaptive Cruise Control Stop-&-Go system*. Master's Thesis, TNO Science & Industry , Technische Universiteit Eindhoven, 2007.
- [9] Zhou, W., and S. Zhang. "Analysis of Distance Headways." *Proc. Of the Eastern Asia Society for Transportation Studies*. 2003.
- [10] *Control system Toolbox User's Guide*. The MathWorks, Inc., 1999.