Proportional Integral Derivative versus Proportional Integral plus Control Applied to Mobile Robotic System

E. M. Shaban[1] and Ayman A. Nada[2]

[1]Faculty of Engineering, Jazan University, – Saudi Arabia,
On leave from Faculty of Engineering (Mataria), Helwan University – Egypt
[2]Benha Faculty of Engineering, Benha University, Benha – 13512, Egypt
modern3@hotmail.com, ayman.nada@bhit.bu.edu.eg

Abstract: An autonomous robot is an embedded intelligent system that operates independently with real time computing constraints. These embedded systems are dedicated to specific tasks, therefore most researchers and design engineers are focusing to optimize its size and cost, as well as increase its reliability and performance. The development of complex control procedures within computerized and integrated systems is crucial and attracts high prospect in research and industry. One of the primary factors considered in the navigation of autonomous robot is dynamic modeling and real time control. In this regard, the recently developed Proportional Integral Plus (PIP) control is implemented over an autonomous guided robot. In the PIP control, full identification process for the discrete time model is carried out and the control law is estimated according to the measured data. The PIP control performance is compared with the optimal Proportional Integral Derivative (PID) control methodology, for which the Internal Time Square Error (ITSE) index is used to design the controller constants. The advantages and limitations for these distinct approaches are identified in terms of both performance and design effort. A primitive semi-circle route is proposed for the robot when the two controllers are applied. Simple differential wheeled mobile robot is used with laptop host and LabView software.

[E. M. Shaban and Ayman A. Nada. **Proportionla Integral Derivative versus Proportional Integral plus Control Applied to Mobile Robotic System.** *J Am Sci* 2013;9(12):583-591]. (ISSN: 1545-1003). http://www.jofamericanscience.org. 76

Keywords: Proportionla Integral Derivative; Proportional Integral; Mobile Robotic System

1.Introduction

Control problem of autonomous robotic systems attracts a considerable attention, since they are used in a wide variety of applications. These systems have the advantage of small time cost, and most important, they enhance the safety of operators due to the inexistence of manual operation. To make autonomous robotic systems viable, it is essential that researchers develop their controller actions to improve response times on those of skilled human operators; otherwise the economic and practical benefits of autonomous are limited.

For this reason, this paper focuses on the implementation of two contrasting control approaches on the navigation of autonomous robot. A primitive semi-circle path is assumed, see Fig. (1), during the application of the two approaches. The first is called PIP approach which is presumed as a sort of True Digital Control (TDC). The PIP controller can be interpreted as a logical extension of conventional PI controllers, with additional dynamic feedback and input compensators, which introduced automatically when the process has second order, higher dynamics, or pure time delays greater than one sampling interval. Here, the Non-Minimal State Space (NMSS) models are formulated [1], so that full state variable feedback can be implemented directly from the measured input and output signals of the process, without resort to the design and implementation of a deterministic state reconstructor (observer) or stochastic Kalman filter [1, 2]. The structure of PIP exploits the power of state variable feedback, for which the vagaries of manual tuning are replaced by pole assignment or Linear Quadratic (LQ) design. Over the last few years, such NMSS/PIP control systems have been successfully employed in a wide range of applications, e.g. [3–6].

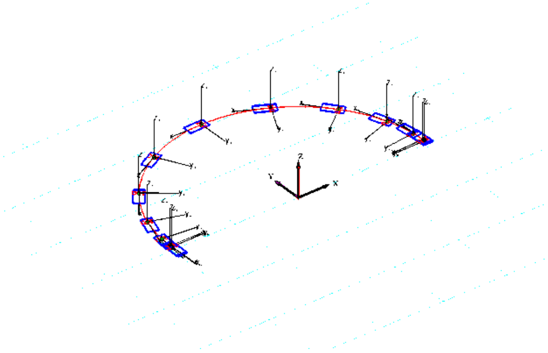


Figure 1. The semi-circle path assumed for the mobile robot.

Although modern design methods, such as NMSS/PIP, are often said to yield performance or robustness benefits, most industrial feedback control systems are nonetheless based on classical PID methods such as the Nichols chart [7]. There are perhaps two reasons why such classical techniques remain popular in industrial practice. In the first instance, they provide good designs in the face of uncertainty in the plant model. For example, if a system has poorly understood resonances at high frequency, the design can be compensated to alleviate their effects. Secondly, in the absence of a formal model of the system, experimental information can be used directly for design purposes – there is no need for intermediate processing of the data to arrive at a system model. In other words, measurements of the output amplitude and phase of a system exited by a sinusoidal input can be used directly to design the control system. Whilst the wide availability of powerful computers renders this second advantage less important than in the past, the design methods remain extremely effective and popular.

In order to gain insight into the design problem and to compare the true advantages of PIP over a properly tuned classical controller, the present paper develops and evaluates control systems based on both NMSS/PIP and classical PID methods. For the preliminary study reported here, a primitive semi-circle rout for the autonomous robot is considered. The control objective is to achieve smoother, more accurate steady state tracking response with ideally no overshoot even in the event of load disturbances. Finally, for consistent comparison, both controllers are implemented with a sampling time of 0.1 seconds, which is the fastest possible time interval permitted by the existing experimental kit.

Description of the Autonomous Robot

This research is based on applying two distinctive controller types on DaNI robotic kit, see Fig. 2. It includes an assembled robot with frame, wheels, drive train, DC motors with encoders, and host computer (not shown in the figure). The control laws are developed in LabView software via a remote host computer, and downloaded to the robot's real-time processor for implementation.

The NI robotics starter kit [8], uses an NI Reconfigurable I/O (RIO IO) single board with embedded control platform and ultrasonic range radar finder (not considered in this study). The RIO single board controllers integrates with a real time processor, configurable field-programmable gate array (FPGA), and analogue-to-digital (I/O) terminals. It is powered by LabView real-time and LabView FPGA modiles. The built-in I/O can be expanded using C series modules. The robot has two DC motors for each side, installed on the front wheels with a 400-tick encoder. This independency of the two motors allows not only driving but also maneuvering the robot. The mechanical components are produced by PITSCO education [9]. Ethernet crossover cable is used for direct connection between the demonstrator and the host computer with all required preloads of LabView and robotics real-time and FPGA modules.

Finally, the two TETRIX DC motors are working in the range $0-4.66\ \mathrm{A}$. Sabertooth $2\times 10\mathrm{RC}$ motor driver is included in the DaNI components to provide the power needed to drive the motor. However the sbRIO specification shows that the current output is only $3\ \mathrm{mA}$, the motor driver connects to both batteries and sbRIO to supply $10\ \mathrm{A}$ for each DC motor. The two optical encoders are attached to the DC motors, they need $5\ \mathrm{V}$ as power supply, and produce $100\ \mathrm{CPR}$ with $400\ \mathrm{PPR}$.

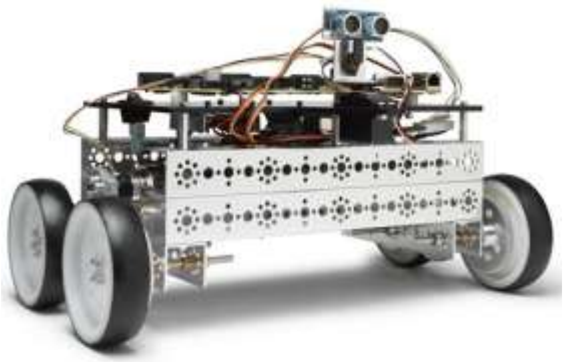


Figure 2. A snapshot of DaNI robot.

Pip Control

The PIP approach to control system design is carried out entirely in discrete time, starting from the identification and estimation of a suitable linearized model to the real-time implementation of the final control law. Therefore, as a first step, it is required to find a linearized single-input single output (SISO) representation of the dynamic system based on a transfer function (TF) model, with parameters identified and estimated from the measured data. In terms of backward shift operator, z^{-1}, this TF model takes the form of (1).

$$y_k = \frac{B(z^{-1})}{A(z^{-1})} u_k \tag{1}$$

where y_k is the output variable, u_k the control input and,

$$A(\mathrm{z}^{-1}) = 1 + a_1 \mathrm{z}^{-1} + a_2 \mathrm{z}^{-2} + \cdots + a_n \mathrm{z}^{-n}$$
$$B(\mathrm{z}^{-1}) = b_1 \mathrm{z}^{-1} + b_2 \mathrm{z}^{-2} + \cdots + b_m \mathrm{z}^{-m} \tag{2}$$

in which $a_1 \cdots a_n$ and $b_1 \cdots b_m$ are the TF parameters, while z^{-1} is the backward shift operator, i.e. $\mathrm{z}^{-i} y_k = y_{k-i}$.

An appropriate structure for the transfer function (1) needs to be defined, i.e. the triad $\{n,m,\delta\}$ where δ is the pure time delay, typically represented by setting $b_1 \cdots b_{\delta-1} = 0$. The two main statistical measures employed to help determine these values are the coefficient of determination R_T^2, based on the response error, which is a simple measure of model fit; and the Young Identification Criterion (YIC), which provides a combined measure of fit and parametric efficiency, with large negative values indicating a model which explains the output data well, without over-parameterization [10]. The present work utilizes the Simplified Refined Instrumental Variable (SRIV) algorithm to estimate the model parameters [10,11]. These statistical tools and associated estimation algorithms have been assembled in the Matlab® software as the CAPTAIN toolbox. It could be found at http://www.es.lancs.ac.uk/cres/captain/.

It is possible to show that the SISO model (1) can be represented by the following linear Non–Minimal State Space (NMSS) equations,

$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}\mathbf{x}_{k-1} + \mathbf{g}u_{k-1} + \mathbf{d}y_{d,k} \\ y_k &= \mathbf{h}\mathbf{x}_k \end{aligned} \tag{3}$$

for which the matrix **F**, and the vectors **g**, **d** and **h** are defined by [12]. Here, the $n+m$ *non-minimal* state vector $\mathbf{x}_k$, consists of the present and past sampled values of the input and output variables, i.e.,

$$\mathbf{x}_k = [y_k \quad y_{k-1} \quad \cdots \quad y_{k-n+1} \quad u_{k-1} \quad \cdots \quad u_{k-m+1} \quad z_k]^T \tag{4}$$

Here, the state $z_k = z_{k-1} + \{r_k - y_k\}$ is the *integral-of-error* state between the command input r_k and the sampled output y_k. The control law associated with the NMSS model (3) takes the usual State Variable Feedback (SVF) form,

$$u_k = -\mathbf{k}\,\mathbf{x}_k \tag{5}$$

Where $\mathbf{k} = [\,f_0 \;\; f_1 \cdots f_{n-1} \;\; g_1 \cdots g_{m-1} \;\; -K_I\,]$ is the SVF control gain vector. In more conventional block-diagram terms, the SVF controller (5) can be implemented as shown in Fig. 3, where it is clear that it can be considered as one particular extension of the ubiquitous PI controller, where the PI action is enhanced by the higher order forward path and feedback compensators $1/G(z^{-1})$ and $F(z^{-1})$ where,

$$\begin{aligned} F(z^{-1}) &= f_0 + f_1 z^{-1} + \cdots + f_{n-1} z^{-(n-1)} \\ G(z^{-1}) &= 1 + g_1 z^{-1} + \cdots + g_{m-1} z^{-(m-1)} \end{aligned} \tag{6}$$

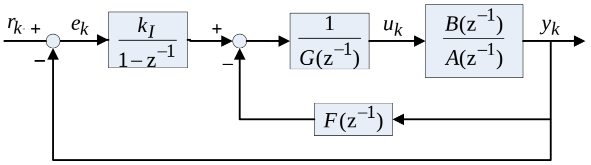


Figure 3. The typical PIP control block diagram.

However, because it exploits fully the power of SVF within the NMSS setting, PIP control is inherently much more flexible and sophisticated, allowing for well-known SVF strategies such as closed loop pole assignment, with decoupling control in the multivariable case; or optimization in terms of a Linear-Quadratic (LQ) cost function of the form,

$$J = \frac{1}{2}\sum_{i=0}^{\infty}\left\{\mathbf{x}_i^{\mathrm{T}}\mathbf{Q}\mathbf{x}_i + Ru_i^{\,2}\right\} \tag{7}$$

where $\mathbf{Q} = diag[q_1 \ldots q_n \;\; q_{n+1} \ldots q_{n+m-1} \;\; q_{n+m}]$ is a diagonal state weighting matrix and R is an additional scalar weight on the input. The resulting SVF gains are then obtained recursively from the steady state solution of the Algebraic Riccati Equation (ARE), derived from the standard LQ cost function (7) as follows [13],

$$\begin{aligned} \mathbf{k} &= \left[\mathbf{g}^T\,\mathbf{P}^{(i+1)}\,\mathbf{g} + R\right]^{-1}\mathbf{g}^T\,\mathbf{P}^{(i+1)}\,\mathbf{F} \\ \mathbf{P}^{(i)} &= \mathbf{F}^T\,\mathbf{P}^{(i+1)}\left[\mathbf{F} - \mathbf{g}\,\mathbf{k}\right] + \mathbf{Q} \end{aligned} \tag{8}$$

for which $\mathbf{P}$ is a symmetrical positive-definite matrix with the initial value, $\mathbf{P}^{(i+1)}$, equal to the weighting matrix $\mathbf{Q}$ and $\mathbf{k}$ is the control gain vector.

System Identification and Estimation:

The demonstrator and its hardware peripherals limit the sampling time to 100 ms. This shows sufficient sampling rate for controlling the autonomous robot regarding similar dynamic systems [3–7].

In the first instance, data were collected during the normal working conditions of the robot. The data file contains the two input signals used to maneuver and drive both left and right DC motors, and the speed of each of them. The SRIV algorithm together with the YIC and R_T^2 identification criteria reveals that a first order TF model with one numerator having one sample delay provides the best estimated model with optimum fit to the data collected. Hence, the TF model for both DC motors, based on (1), is

$$y_k = \frac{b_1\,z^{-1}}{1 - a_1\,z^{-1}}\,u_k \tag{9}$$

where y_k is the angular velocity of the dc motor, in rad/sec, and u_k is the normalized voltage applied to the dc motor scaled from –1000 to 1000. The parameters $a_1 = -0.9759$ and $b_1 = 0.00066$ for the right dc motor yields the best fit for the given range of operating conditions. Similarly, the parameters $a_1 = -0.9416$ and $b_1 = 0.0017$ are used for the left dc motor, see Fig. 4.

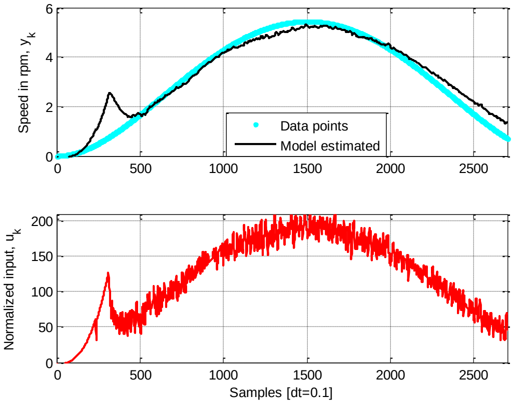
PIP Controller Design and Implementation:

For the given plant, the linear NMSS equations are given by (3) with state vector $x_k = [y_k \quad z_k]^T$. The system is controllable since the two conditions of controllability are applied for the whole range of linearized input [14]. The SVF-PIP control law for the two DC motors is given by

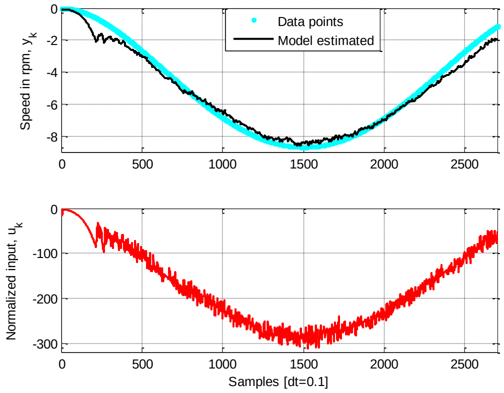
$$u_k = -\mathbf{k}\,\mathbf{x}_k = -[k_p \quad -k_I]\begin{bmatrix} y_k \\ z_k \end{bmatrix} \qquad (10)$$

The problem of optimization is defined as follows: for a linear SISO discrete NMSS form defined in (3), it is required to find the control law in equation (9) with optimal SVF control gain vector that minimizes the quadratic cost function (7). The resulting SVF gains are then obtained recursively from the steady state solution of the ARE (8) derived from the standard LQ cost function (7). Trial and error experimentation suggests that setting $Q = diag\,[1 \quad 1]$ and $R = 0.1$ for right DC motor, and $Q = diag\,[1 \quad 1]$ and $R = 0.25$ for left dc motor yield a suitably fast PIP–SVF gain vectors,

$$\mathbf{k} = [65.1952 \quad -3.0919] \quad \text{for right dc motor}$$
$$\mathbf{k} = [23.3042 \quad -1.9575] \quad \text{for left dc motor} \qquad (11)$$



(a) Right hand side DC motor, $R_T^2 = 0.95$



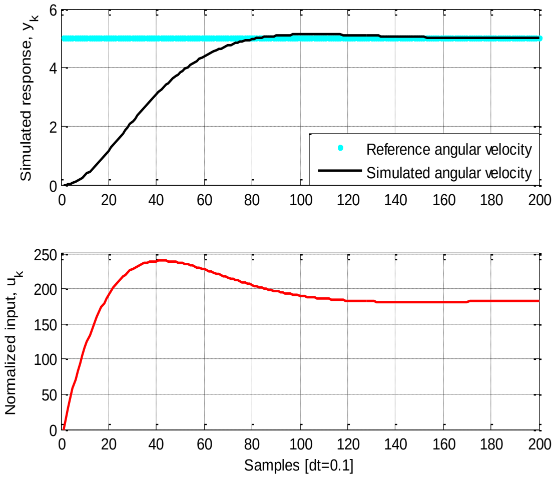
(b) Left hand side DC motor, $R_T^2 = 0.98$

Figure 4. The response of the model (9) [solid], with respect to the data collected. The sampling rate is 10 samples/second.

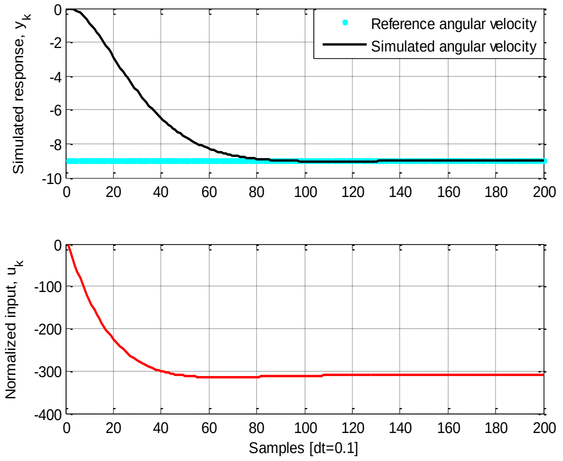
The simulation of the model (9) for both right and left motors, using gain vectors in (11), is shown in Fig. 5, for which the simulation shows satisfactory tracking performance with acceptable settling time for both left and right wheels. As shown in Fig. 5, suitable synchronization for both motor responses is also achieved, theoretically.

LabView® program is constructed for the sake of implementation, for which the angular velocity feedback, for both right and left wheels, is always compared to the reference angular velocity. This error action together with the gradient of output angular velocity constitute the control action, according to the following incremental form

$$u_k = u_{k-1} + k_I(r_k - y_k) - k_p(y_k - y_{k-1}) \qquad (12)$$



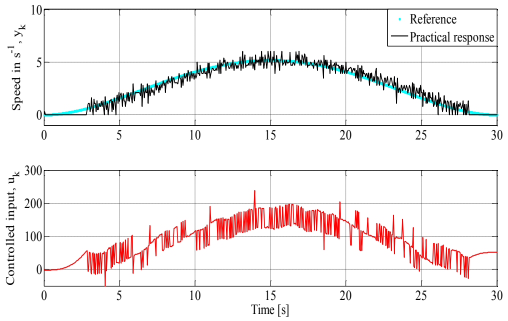
(a) Right hand side DC motor.



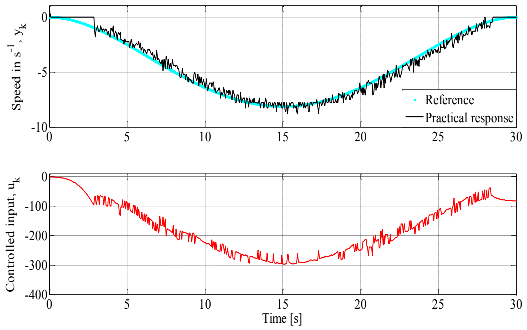
(b) Left hand side DC motor.

Figure 5. The simulation response of the model (9) for both DC motors using the gain vector defined in (10).

The implementation of PIP controller is depicted in Fig. 6, for which the right and left motors' angular velocity tracks their calculated reference values according to the path planned. A simple coefficient of determination can be used here to measure the controller efficiency. According to the given experiment, the PIP controller for right and left wheels give a coefficient of determination of 0.95 and 0.98 respectively over the whole time of implementation. This is good enough tracking responses for both DC motors.



(a) Implementation of right hand side DC motor.



(b) Implementation of left hand side DC motor.

Figure 6. The implementation results of the autonomous robot.

The practical responses of both DC motors usingPIP gain vectors defined in (11).

Pid Control

The most majority of the controllers used in industry are PID controllers, because when properly tuned they generally achieve satisfactory performance. These types of controllers are more convenient for users owing to their simplicity of implementation. The PID controllers have many forms, the most widely used form is written as

$$u(t) = K_p e(t) + K_i \int e(t)\,\mathrm{d}t + K_d \frac{\mathrm{d}e(t)}{\mathrm{d}t}$$

$$U(\mathrm{s}) = K_p\left(1+\frac{1}{\tau_i\,\mathrm{s}}+\tau_d\,\mathrm{s}\right)E(\mathrm{s}) \qquad (13)$$

where K_p, K_i, K_d are the proportional, integral and derivative gain, respectively, and τ_i, τ_d are the reset and derivative time.

The development of adaptive PID control algorithms is characterized by the search for appropriate methods of estimation of system model parameters as well as methods of control law synthesis. The mathematical theory of adaptive control is well established provided that the primary control design is putting the closed loop into the stability boundary and determining the system gain, time constant and the dead time period. In the case of controlling motor velocities, it is assumed that the system model is first-order with dead time, i.e.

$$G(\mathrm{s}) = \frac{K_c e^{-L\mathrm{s}}}{T\mathrm{s}+1} \qquad (14)$$

where K_c is the system gain, T is the time constant, and L is the dead time. The following PID tuning algorithms are considered for primary control design, see Table 1:

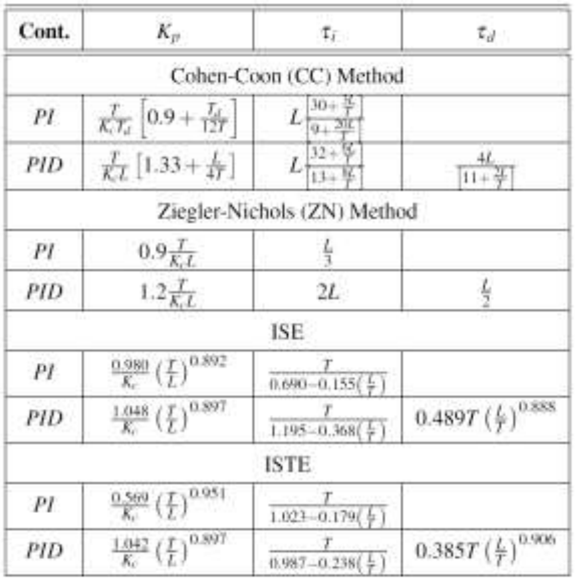
1- Ziegler–Nichols (Z–N) method [15]. There are two versions of Z–N method, one depends on the reaction curve, whereas the other rely on the ultimate gain K_u and the ultimate period T_u.
2- Cohen–Coon (CC) method [16], which is based on reaction curve. Here, a model with one tangent and point is derived first to tune the PID controller. For first-order system with dead time, the PID parameters can be directly related to model parameters. This method is similar to Z–N method, nevertheless provides better results when the controller has a large dead time.
3- Optimum integral error for load disturbance (IAE–load, ITAE–load, ISE–load, ISTE–load), and for set point change (IAE–set point, ITAE–set point, ISE–set point, ISTE–set point) methods. There are many versions of the integral–error based methods. Here, ISE, ISTE optimal tunings are adopted from [17].

For consistent comparison with PIP control, the optimum PID control algorithm based on ISTE-set point is used.

Identification and PID Control Design:

The system parameters can be identified by plotting the open loop system step response, and estimating the system measures. The identification process can be carried out computationally, using the LABVIEW software, to estimate the parameters K_p, τ_i, and τ_d according to Table 1.

TABLE 1: PID tuning formulae [17]



Cont.	K_p	τ_i	τ_d
Cohen-Coon (CC) Method			
PI	$\frac{T}{K_c T_d}\left[0.9+\frac{T_d}{12T}\right]$	$L\frac{30+\frac{3L}{T}}{9+\frac{20L}{T}}$	
PID	$\frac{T}{K_c L}\left[1.33+\frac{L}{4T}\right]$	$L\frac{32+\frac{6L}{T}}{13+\frac{8L}{T}}$	$\frac{4L}{11+\frac{2L}{T}}$
Ziegler-Nichols (ZN) Method			
PI	$0.9\frac{T}{K_c L}$	$\frac{L}{3}$	
PID	$1.2\frac{T}{K_c L}$	$2L$	$\frac{L}{2}$
ISE			
PI	$\frac{0.980}{K_c}\left(\frac{T}{L}\right)^{0.892}$	$\frac{T}{0.690-0.155\left(\frac{L}{T}\right)}$	
PID	$\frac{1.048}{K_c}\left(\frac{T}{L}\right)^{0.897}$	$\frac{T}{1.195-0.368\left(\frac{L}{T}\right)}$	$0.489T\left(\frac{L}{T}\right)^{0.888}$
ISTE			
PI	$\frac{0.569}{K_c}\left(\frac{T}{L}\right)^{0.951}$	$\frac{T}{1.023-0.179\left(\frac{L}{T}\right)}$	
PID	$\frac{1.042}{K_c}\left(\frac{T}{L}\right)^{0.897}$	$\frac{T}{0.987-0.238\left(\frac{L}{T}\right)}$	$0.385T\left(\frac{L}{T}\right)^{0.906}$

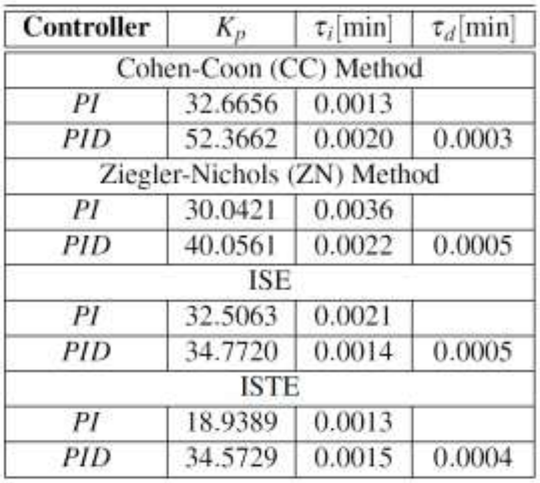
An open loop step input function is used with half-load input signal. Here, the open-loop response for the left motor, shows that the system gain $K_c = 0.031764$, time constant $T = 0.068865\text{s}$, and time delay $L = 0.0649496\text{s}$. The normalized time delay (the ratio between the system time delay and the time constant) is very important factor. Here, the normalized delay is $L/T = 0.943$, which is less than unity. However, if the normalized delay is greater than unity, the process will be regarded as having a large time delay, and PID control structure is not recommended for such processes. Nevertheless, to achieve good performance, dead time compensator structure, such as a Smith predictor must be used.

Repeating the same steps for the right motor, a very close behavior to first order plus dead time is obtained. The following system characteristics can then be achieved: The proportional gain $K_c = 0.0298$, time constant $T = 0.0766\,\text{s}$, and time delay or dead time $L = 0.05339\,\text{s}$. Based on the identified parameters for both motors, the PID control gains are depicted in Table (2).

Implementation of PID controller using LabView®:

Similarly, LabView® program is constructed for implementing the PID control, for which several PID control laws have been tried out. It is found that the best control performance has been achieved using ISTE designed controller.

TABLE 2: PID tuning gains



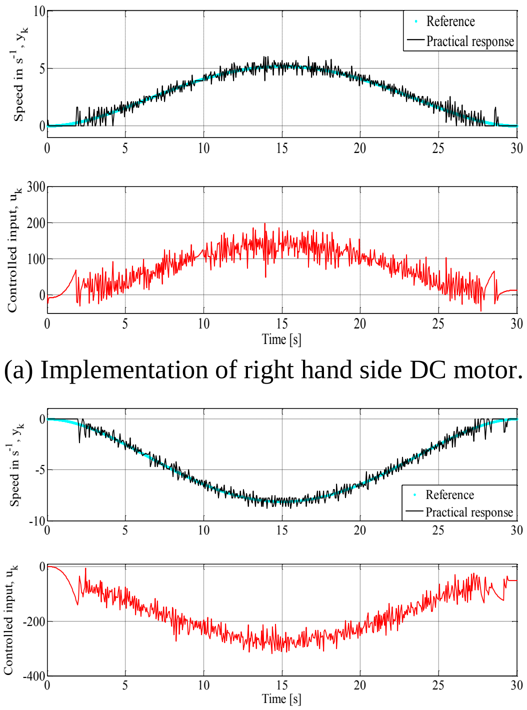
Controller	K_p	τ_i[min]	τ_d[min]
Cohen-Coon (CC) Method			
PI	32.6656	0.0013	
PID	52.3662	0.0020	0.0003
Ziegler-Nichols (ZN) Method			
PI	30.0421	0.0036	
PID	40.0561	0.0022	0.0005
ISE			
PI	32.5063	0.0021	
PID	34.7720	0.0014	0.0005
ISTE			
PI	18.9389	0.0013	
PID	34.5729	0.0015	0.0004

The implementation of PID/ISTE controller is shown in Fig. 7. As shown in this figure, both motors' angular velocity tracks their calculated reference values with a coefficient of determination of 0.96 and

0.98 for right and left motors respectively, which shows good enough tracking responses for both DC motors.

3.Results and Discussion

The responses of the two DC motors of the mobile robot for both PIP and PID controllers are illustrated in Figs. 6 and 7 respectively. As would be expected both controllers yield zero steady state error, with very close tracking performance. However, PID/ISTE control shows a slight enhancement over PIP control in terms of response time. Moreover PIP control shows relatively less control input noise especially in the left DC motor.



(a) Implementation of right hand side DC motor.

(b) Implementation of left hand side DC motor. Figure 7. The implementation results of the autonomous robot.

The practical responses of both DC motors using PID/ISTE gains as defined in Table (2).

Finally, Fig. 8 shows a comparison between the two control systems regarding tracking a semi-circle path. A typical result shows very close performance between both controls. It is worth to note that the performance of PIP control could be enhanced by simply adjusting the weighting terms and repeat the design for faster gains.

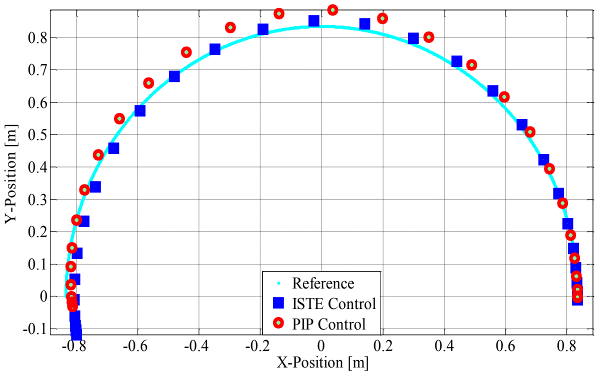


Figure 8. Experiment showing the response of the PIP and PID/ISTE controllers when tracking the semi-circle path.

Moreover, PIP control shows better performance at the end of the path. It seems that the PIP gains compensate the inertia of the mobile robot better than PID/ISTE gains, therefore the plant using PIP stops precisely at the end of the path.

Indeed, the reason for the differences between the two approaches during the implementation results, which are likely to be caused by the different control structures utilized, requires further investigation: a decision as to which control is "better" for this application remains ambiguous. This is clear from Figs 9, 10 and 11.

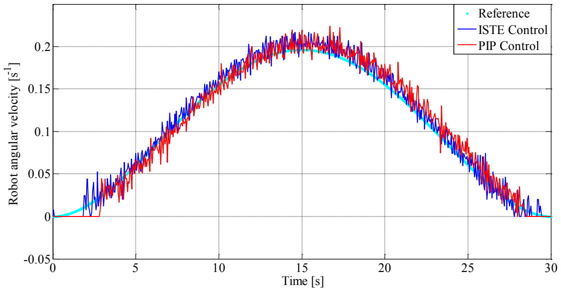


Figure 9. The response of the PIP and PID/ISTE controllers to mobile robot angular velocity.

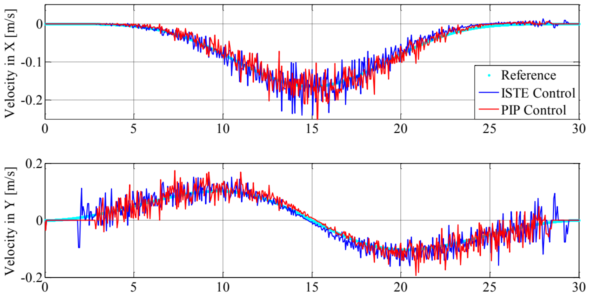


Figure 10. The response of the PIP and PID/ISTE controllers to mobile robot in terms of its resolved velocity in X- and Y-direction.

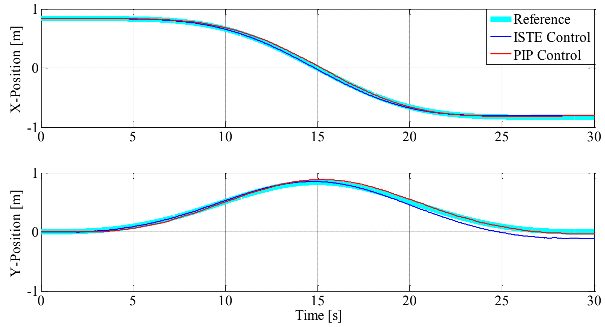


Figure 11. The response of the PIP and PID/ISTE controllers to mobile robot in terms of its resolved position in X- and Y-direction.

Conclusions

The aim of the research is to provide a balanced comparison between the chosen modern and classical approaches to control system design, when applied to a mobile robot. The two control schemes considered are output feedback. Both approaches yield acceptable results with little to choose between them. However, PID-ISTE uses one more gain over PIP control.

In general, the plant has one sample delay; this could be a reason for the slight difference between the two approaches. Note, however, that the results here are limited to primitive semi-circle route; other trajectories and/or input/output disturbances may well alter things.

Acknowledgment

The authors are grateful for the support of the College Engineering, Jazan University, Saudi Arabia.

References

1. Young, P.C., Behzadi, M.A., Wang C.L., and Chotai, A., 1987. “Direct digital and adaptive control by input-output, state variable feedback pole assignment”. Int. J. Control, 46, 1867–1881.
2. Taylor, C.J., Chotai, A., Young, P.C., 2000. “State space control system design based on non-minimal state-variable feedback”. Further generalization and unification results, International J. Control, 73, pp. 1329–1345.
3. Shaban, E.M., Elsayed, M., 2009. “Design, simulation, and implementation of a class of true digital control (TDC) applied to natural gas burner”. Computer Engineering and Systems, Appears on ICCES 2009, International Conference: 14–16 December, p. 109–114.
4. Shaban, E.M., Ako, S., Taylor, C.J. Seward, D.W., 2008. “Development of an automated verticality alignment system for a vibro-lance”. Automation in Construction, 17(5), p. 645–655.
5. Taylor, C.J., Shaban, E.M., Stables, M.A., Ako, S., 2007. “Proportional-Integral-Plus control applications of state dependent parameter models”. Journal of System and Control Engineering, 221(7), p. 1019–1031.
6. Sidiropoulou, E., Shaban, E.M., Taylor, C.J., Tych, W., Chotai, A., 2006. “Linear, nonlinear, and classical control of 1/5th scale automated excavator”. 18th International Conference on System Engineering, ICSE-06, Convey, UK.
7. Dixon, R., Taylor, C.J., Shaban, E.M., (2005). “Comparison of classical and modern control applied to an excavator arm”. International Federation of Automatic Control, 16th Triennial World Congress (IFAC-05), 16(1), p. 1367–1372, Prague, Czech Republic.
8. National Instrument Cooperation, www.ni.com\.
9. PITSCO Education, http://www.pitsco.com.
10. Young, P.C., 1991. “Simplified Refined Instrumental Variable (SRIV) estimation and True Digital Control (TDC)”. A tutorial introduction, 1st European Control Conference, pp. 1295–1306 (Grenoble).
11. Young, P.C., 1998. “Data-Based mechanistic modeling of engineering systems”. Journal of Vibration and Control, (4), pp. 5–28.
12. Taylor, C.J., Chotai, A., Young, P.C., 2000. “State space control system design based on non-minimal state-variable feedback: further generalization and unification results”. International Journal of Control, (73), pp. 1329–1345.
13. Astrom, K.J., Wittenmark, B., 1984. “Computer controlled systems: Theory and Design”. Prentice-Hall Information and System Sciences Series.
14. Chotai, A., Wang, C.L., Young, P.C., 1992. “Conditions for arbitrary pole assignability by input/output state-variable feedback”. Appears in N.K. Nichols and D.H. Owens (eds), The Mathematics of Control Theory, Oxford University Press, Oxford, pp. 87–96.

15. Ziegler, J. G., Nichols, N. B., (1942). "Optimum settings for automatic controllers". Transactions of ASME, 62, p. 759–768.
16. Cohen, G. H., Coon G. A., (1953). "Theoretical consideration of retarded control". Transactions of ASME, 75, p. 827–834.
17. O'Dwyer, Aiden, (2009). "Handbook of PI and PID controller tuning rules", 3rd edition, World Scientific. Online version available at http://www.knovel.com/web/portal/browse/display?_EXT_KNOVEL_DISPLAY_bookid=3428&VerticalID=0

11/12/2013