The Development of Proportional-Integral-Plus Control Using Field Programmable Gate Array Technology Applied to Mechatronics System

Ayman A. Nada[a], E. M. Shaban[b]

[a]Benha Institute of Technology, Benha University, Benha 13512, Egypt, email: arobust@tedata.net.eg
[b]Mechanical Engineering Department, Jazan University, Jazan – KSA, On leave from Faculty of Engineering (Mataria), Helwan University – Egypt, email: modern3@hotmail.com

Abstract

Proportional-Integral-Plus (PIP) controller can be considered as an extension of classical discrete PI controller, in which, the proportional and the integral actions are enhanced by higher order of input and feedback compensators (PLUS terms). Conventionally, the PIP control design is carried out entirely in discrete time domain, for which open-loop data-based identification and estimation for the linearized transfer function model is performed off-line. This is followed by an on-line implementation for the control operations. In case of on-line estimation of the transfer function model, the overall execution time, at each step, increases considerably. Here, the overall execution time is the sum of each operation, i.e. estimation and control implementation. In such cases, digital processors with single resources may fail to capture relatively fast mechatronic systems properly, causing a decrease in the controller performance. This paper investigates the use of parallel processing facilities available within the Field Programmable Gate Arrays (FPGAs) technology and reconfigurable Input/output chips, PIP-FPGA, for which a fixed-point PIP control design algorithm is developed and implemented upon relatively fast Mechatronic system with high-speed control and high channel count on an FPGA target. Here, parallel loops for measuring feedback signals, model parameter estimation, updating of the control gains, and generating pulse train for actuator's drivers, are illustrated through experimental test-rig of simple automation line system. The experimental results shows valuable enhancement of the implemented PIP control upon such mechatronic systems.

Keywords: Proportional-Integral-Plus (PIP) control, system identification and estimation, Field Programmable Gate Array (FPGA)

{**Citation**: Ayman A. Nada, E. M. Shaban. The development of proportional-integral-plus control using field programmable gate array technology applied to Mechatronics system. American Journal of Research Communication, 2014, 2(4):14-27} www.usa-journals.com, ISSN: 2325-4076.

1. Introduction

Recently, the control design using State Variable Feedback (SVF) becomes a typical approach for many control applications [1, 2]. It is based on state space formulation of the system model, which allows the implementation of powerful control approaches, such as pole placement, Linear Quadratic Regulator (LQR), Linear Quadratic Gaussian (LQG) and other optimal approaches, e.g. H_2 and H_∞ control [3, 4].

One of the major difficulties in the state space approach is that the state vector is not normally available for direct measurement. Therefore, an estimate of the states is required for the implementation of the control system. However, this estimation process, in turns, decreases the control system robustness [4]. The problem remains unresolved in discrete control theories based on minimal state space formulation [5].

The Proportional-Integral-Plus (PIP) control design approach, which is firstly introduced by Young and Wang [6], present a simple solution to this problem by defining the Non-Minimal State Space (NMSS) formulation. Here, the state vector is composed only of those states that can be measured directly and stored for the use of control law. These states are the present and past sampled values of the output variable, and the past sampled value of the input variable. The PIP control method has tuning parameters represented by the weights of the LQR cost function. In this regard, adequate closed-loop performance is obtained by direct tuning of these parameters [7, 8]. NMSS-PIP controllers have been successfully designed for a wide range of real applications [10-23]. Preliminarily, PIP control is based on the discrete time transfer functions (TF), which normally identified and estimated from several open-loop system experiments, by means of Simplified Refined Instrumental Variable (SRIV) which has the ability to smooth the noise subjected to data collection [24-27].

Mechatronic systems are defined as a computer-controlled mechanical system, including both digital computer and electromechanical components. Some examples of fast mechatronic systems include: a DVD player, computer hard disc drive, intelligent mechanisms, mobile robots, packing machines, and smart materials applications. These applications are characterized by its relatively low inertia and quick response time, beside the reference tracking requirements in some situations [28, 29].

The aim of this paper is to study the effectiveness of implementing the PIP control system design upon such fast mechatronic systems. The practical applications in references [9-15] have large capacitances and therefore their responses are relatively slow, which, thereby require low sampling rates. Another concern is the use of single processor computers in order to implement the control action on some of these applications using LabVIEW real-time environment [14, 23, 30] and MATLAB/Simulink [31]. Here, the overall execution time for multiple control process operations is the sum of the execution times of each operation, i.e. estimation and control implementation. Jun Gu, et. al. [16] had used three embedded PC104 computers, each for separate control task in order to decrease the execution time.

In this work, the Field Programmable Gate Arrays (FPGAs) technology [32, 33] is proposed to execute simultaneous parallel control operations. It is worth to note here that FPGA is No-Processor device and therefore can be used effectively for control processes demanding very high loop cycle time. Consequently, this enables FPGAs to score over general purpose computing chips like, such as DSP chips. This reduction in the execution time expects the increase of the efficiency of NMSS-PIP control for such relatively fast mechatronic systems. The ready-made LabVIEW-FPGA module [34, 35] can be used to generate the code and compiling procedures to develop a standalone on-shelf PIP controller for such mechatronic systems.

2. PIP Control

In terms of backward shift operator, z^{-1}, the dynamic representation of discrete time TF model of single-input single-output (SISO) system takes the form

$$\begin{aligned} y(k) &= \frac{b_1 \mathrm{z}^{-1} + b_2 \mathrm{z}^{-2} + \ldots + b_m \mathrm{z}^{-m}}{1 + a_1 \mathrm{z}^{-1} + \ldots a_n \mathrm{z}^{-n}} u(k) \\ &= \frac{B\left(\mathrm{z}^{-1}\right)}{A\left(\mathrm{z}^{-1}\right)} u(k) \end{aligned} \tag{1}$$

where $y(k)$ is the system output, $u(k)$ is the system input, and $A(\mathrm{z}^{-1})$ and $B(\mathrm{z}^{-1})$ are the TF polynomials with $a_1 \ldots a_n$, $b_1 \ldots b_m$ are the TF parameters. An appropriate structure for the transfer function, equation (1) needs to be defined, i.e. the triad (n, m, δ), where δ is the pure time delay, typically represented by setting $b_1 \ldots b_{\delta-1} = 0$.

The non-minimal state vector $x(k)$ is defined in terms of the present and past sampled outputs and the past sampled inputs, plus the discrete-time integral of error between the output $y(k)$ and the command input $y_d(k)$. It takes the following form

$$x(k) = \left[y(k) \quad \ldots \quad y(k-n+1) \quad u(k-1) \quad \ldots \quad u(k-m+1) \quad z(k) \right]^T \tag{2}$$

where integral-of-error state variable, $z(k)$, is represented as

$$z(k) = z(k-1) + \left(y_d(k) - y(k) \right) \tag{3}$$

The NMSS formulation can be written as [7]

$$\begin{aligned} x(k) &= \mathbf{F}\mathbf{x}(k-1) + \mathbf{g}u(k-1) + \mathbf{k}_d y_d(k) \\ y(k) &= \mathbf{h}\mathbf{x}(k) \end{aligned} \tag{4}$$

where $\mathbf{F}$ is state transition matrix, $\mathbf{g}$ is input vector, $\mathbf{k}_d$ is the command input vector, and $\mathbf{h}$ is output vector. Since the SVF control law in the NMSS case involves only the measured input and output variables and their past values, it avoids the need for an explicit state estimation [7]. The SVF control law associated with the NMSS model takes the form of

$$u(k) = -\mathbf{K}^T \mathbf{x}(k) \tag{5}$$

for which the SVF control gain vector can be represented as

$$\mathbf{K}^T = \left[f_o \quad \ldots \quad f_{n-1} \quad g_1 \quad \ldots \quad g_{m-1} \quad -k_I \right] \tag{6}$$

Regarding the control law, equation (5), the block diagram of the closed loop control system, can be constructed as shown in figure (1). The figure shows that the PIP controller can be considered as an extension of the classical discrete PI controller. The proportional action f_o and the integral action $k_I \left(1 - \mathrm{z}^{-1}\right)^{-1}$ are enhanced by higher order input $G_1\left(\mathrm{z}^{-1}\right)$ and the feedback $F_1\left(\mathrm{z}^{-1}\right)$ compensators. These filters can be represented as:

$$\begin{aligned} G_1\left(\mathrm{z}^{-1}\right) &= g_1 \mathrm{z}^{-1} + \ldots g_{m-1} \mathrm{z}^{-m+1} \\ F_1\left(\mathrm{z}^{-1}\right) &= f_1 \mathrm{z}^{-1} + \ldots g_{n-1} \mathrm{z}^{-n+1} \end{aligned} \tag{7}$$

One possible approach to calculate the control gain vector, $\mathbf{K}$, is the optimal state variable feedback control which minimizes the Linear Quadratic (LQ) type of performance criterion, i.e.

$$J = \sum_{k=0}^{\infty} \left[\mathbf{x}^T(k) \mathbf{Q} \mathbf{x}(k) + r u^2(k) \right] \tag{8}$$

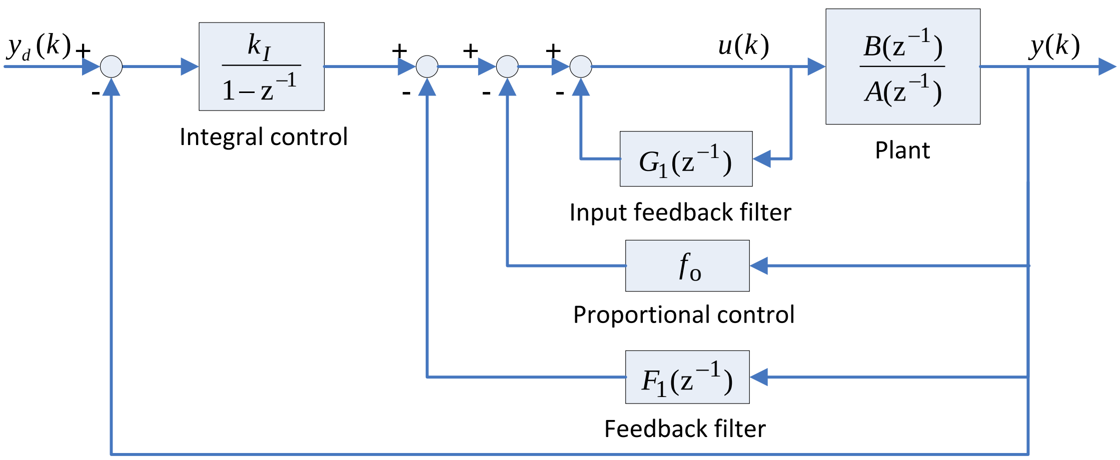

Figure. 1 PIP control system.

This represents the standard formulation of the infinite time optimal LQ servomechanism cost function for SISO system. The weighing square symmetric positive matrix of states, $\mathbf{Q}$, has the dimension $n+m$, and r is a positive scalar for weighing the control action. Considering the NMSS state vector, equation (2), the weighing state matrix takes $\mathbf{Q}=diag\lfloor \mathbf{q}_y \quad \mathbf{q}_u \quad q_e \rfloor$, where $\mathbf{q}_y=\left[q_1 \quad \dots \quad q_n\right]$, and $\mathbf{q}_u=\left[q_{n+1} \quad \dots \quad q_{n+m-1}\right]$ are the weighting measured output and input respectively, whereas q_e is the weighting constraint on the integral-of-error state variable $z(k)$.

Given the NMSS system description $\{\mathbf{F},\, \mathbf{G}\}$, weighing matrix $\mathbf{Q}$ and the control input weighing r; the SVF gain vector, $\mathbf{K}^T$, is given by standard LQ theory as

$$\mathbf{K}^T=\left(r+\mathbf{g}^T\mathbf{P}\mathbf{g}\right)^{-1}\mathbf{g}^T\mathbf{P}\mathbf{F} \tag{9}$$

where $\mathbf{P}$ is the steady-state solution of the following *discrete-time Riccati equation*

$$\mathbf{P}-\mathbf{F}^T\mathbf{P}\mathbf{F}+\mathbf{F}^T\mathbf{P}\mathbf{g}\left(r+\mathbf{g}^T\mathbf{P}\mathbf{g}\right)^{-1}\mathbf{g}^T\mathbf{P}\mathbf{F}-\mathbf{Q}=0 \tag{10}$$

Because of the singularity of the state transition matrix $\mathbf{F}$, the SVF gains are obtained recursively [38] as follows

$$\begin{aligned}\mathbf{K}^T(i)&=\left(r+\mathbf{g}^T\mathbf{P}(i+1)\mathbf{g}\right)^{-1}\mathbf{g}^T\mathbf{P}(i+1)\mathbf{F}\\ \mathbf{P}(i)&=\mathbf{Q}+\mathbf{F}^T\mathbf{P}(i+1)\mathbf{F}-\mathbf{F}^T\mathbf{P}(i+1)\mathbf{g}\mathbf{K}^T(i)\end{aligned} \tag{11}$$

Here, the initial value of matrix $\mathbf{P}$ takes the same value of $\mathbf{Q}$, i.e. $\mathbf{P}(N)=\mathbf{Q}$ and $\mathbf{K}(N)=\mathbf{0}$.

3. Field Programmable Gate Arrays (FPGA)

An FPGA is analogous to a printed circuit board that has a number of unconnected devices on it. The connection in an FPGA circuit can be dynamically defined by appropriate software programming. The program causes semiconductor switches to turn ON or OFF, thereby defining the connections between gates. The FPGA is defined as a programmable chip composed of three basic components: logic blocks, programmable interconnects, and Input/Output (I/O) blocks. A single FPGA chip can replace thousands of discrete components by incorporating millions of logic gates. Figure (2) shows an FPGA as a reconfigurable digital architecture with a matrix of configurable logic blocks with horizontal and vertical routing channels surrounded by a periphery of IO blocks. Signals can be routed within the FPGA matrix in any arbitrary manner by programmable interconnect switches and wire routes.

However, programming these FPGAs require experienced digital designers and hardware engineers. NI[1] has simplified programming of these devices via visual programming system design with LabVIEW-FPGA module, hence the advantages of these powerful reconfigurable chips can be explicitly implemented.

The LabVIEW-FPGA module is used to define the FPGA logic, similar to some low-level languages, such as very high speed integrated circuit hardware description language (VHDL). Here, LabVIEW-FPGA generates the VHDL code and passes it to the Xilinx[2] compiler. Then the Xilinx compiler synthesizes the VHDL and routes all synthesized components into a “bitfile”. The compiled “bitfile” is downloaded to the FPGA chip. FPGA logic provides timing, triggering, processing and custom IO measurements. One can implement multi-loop analog control systems at loop rates exceeding $100\,\text{KS/s}$, and digital control systems at loop rates up to $1\,\text{MS/s}$, and it is possible to evaluate multiple rungs of Boolean logic using single-cycle “while” loops at $40\,\text{MHz}$.

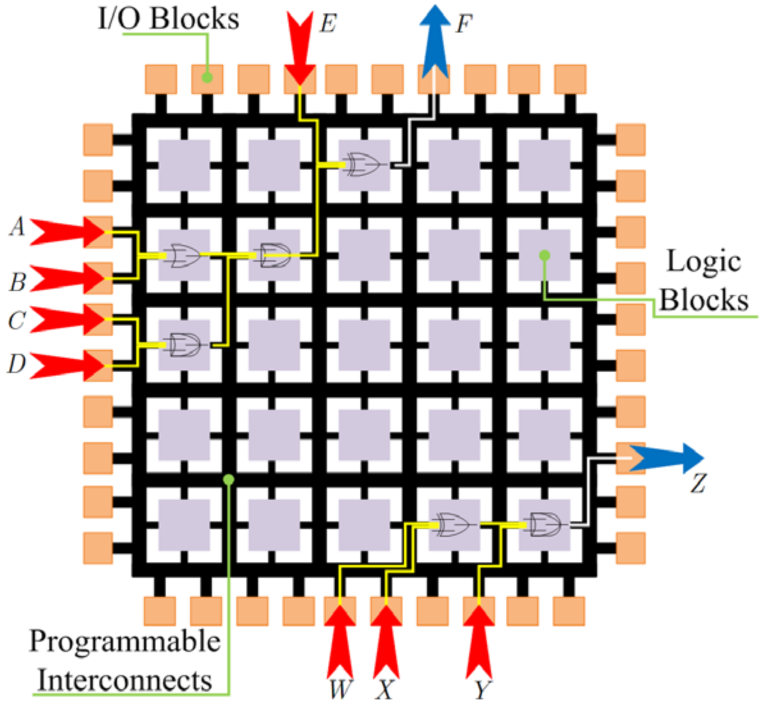

Figure 2 Reconfigurable FPGA chip showing the ability of parallel processing.

LabVIEW and FPGA, together, can implement synchronous or asynchronous parallel tasks in hardware to process and generate synchronized analog and digital signals rapidly and deterministically. FPGA module creates dedicated hardware for each independent function in FPGA VIs. Besides, there is no operating system on the FPGA module that divide the Central Processing Unit (CPU) time between several tasks. This parallel nature of the Reconfigurable Input/Output (RIO) core provides additional computation without reduction in the speed of the FPGA application.

[1] National Instruments Corporation, www.ni.com
[2] www.xilinx.com

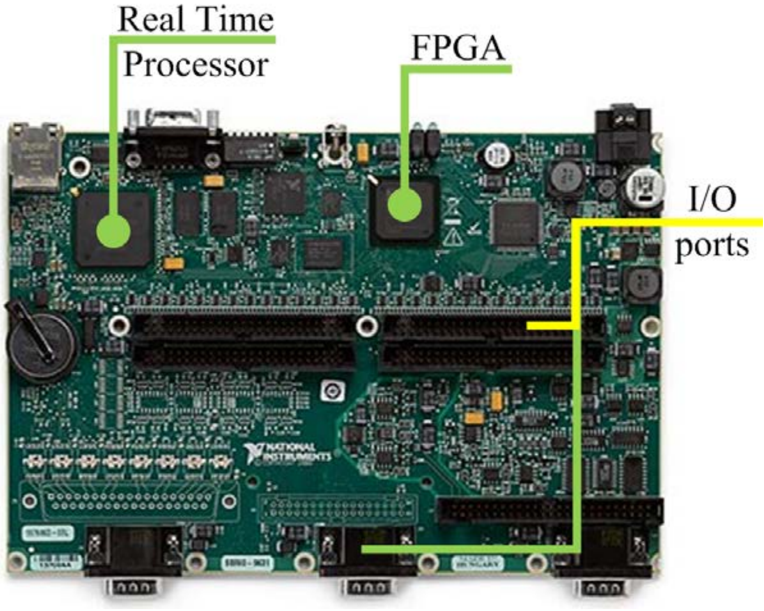

Figure 3 Reconfigurable Input/Output Single Board (sbRIO-9631) with built-in real time processor, FPGA and I/O blocks. www.ni.com.

In order to attain the parallel processing, the code should be divided into several independent segments “loops”. For example, in mechatronic demonstrator, independent loops can be created for acquiring the measured analog data with distinct loop rates.

The FPGA module restricts the use of mathematical operations in FPGA VIs to integer numeric data types and it is not possible to use the floating point operations. Also, even as integer math is used, the results may overflow, when the result of a mathematical operation exceeds the range of the output data type. One possible approach to avoid the integer overflow is scaling down the magnitude of inputs by power of 2, or the use of larger output data type. Nevertheless, the scaling minimize the amount of space used in FPGA device, it reduces the precision of the mathematical operations. On the other hand, larger output data type take more space on FPGA device, yet perform the processing more quickly and receive more accurate data. In this investigation, scaling down of control action has been used with acceptable control performance. Furthermore, only fixed-size, one dimensional arrays in FPGA VIs can be used, i.e., the arithmetic manipulation that returns a variable-size array cannot be used. Since arrays consume significant amounts of space on FPGA, therefore, arrays larger than 32 elements should be avoided.

In the current experimental work, single board RIO (sbRIO), see figure (3), is used. Here sbRIO-9631 integrates a $266\,\mathrm{MHz}$ real-time processor with $1\mathrm{M}$ gate Xilinx Spartan FPGA, 110 digital IO lines. It provides $64\,\mathrm{MB}$ of DRAM for embedded operation and $128\,\mathrm{MB}$ of nonvolatile memory for storing programs and data logging. This device features a built-in $10/100\,\mathrm{Mbits/s}$ Ethernet port which enables to conduct signals communication over the network.

A simple Mechatronic demonstrator is constructed using the following main components: actuator (DC motor), sensor (Encoder), driver (linear/pulses) and belt conveyer as a simple automation line for controller evaluation, see figure (4). For the sake of controlling the mechatronic system, FPGA layout should include two basic independent loops; the first loop concerns measuring and feedback the angular velocity by the optical quadrature encoder, whereas the second is responsible for generating the necessary pulse width modulation (PWM) to drive the DC motor. These two loops are also used, preliminarily, for TF parameters estimation process. During the implementation process, some other loops are incorporated with these two basic loops, such as control action loop, and another loop data collection and transmission to the real-time processor for TF parameters updating process. Figure (5) shows the layout of the parallel processing loops. The details of measuring and driving loops are found in [37].

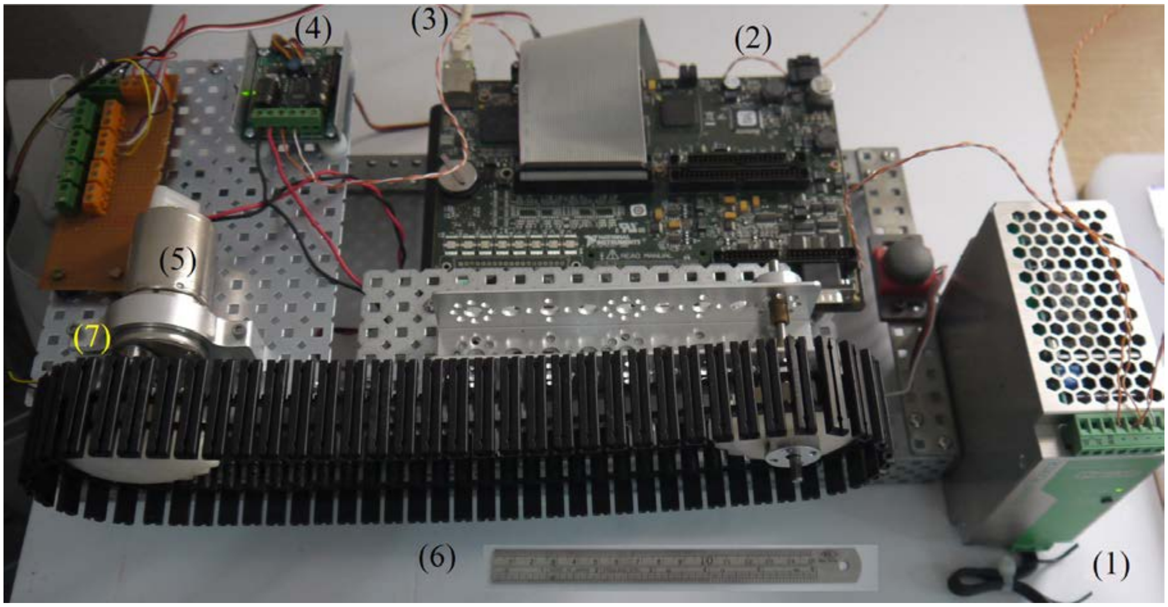

Figure 4 Test-Rig: Mechatronic system: (1) Power supply, (2) sbRIO 9631, (3) Ethernet cable, (4) Motor PWM driver, (5) DC motor, (6) Simple Automation line, (7) Encoder.

4. Development of FPGA-PIP Controller

In order to develop the procedure of fixed-point PIP control for FPGA reconfigurable chip, the limitation of the available space in the FPGA chip as well as the numerical manipulation using only fixed-size integer numbers should be considered. Therefore, in this work, the identification of the TF model and preliminarily estimation of its parameters are carried out offline. This is followed by the development of fixed-point PIP controller.

4.1. System identification and estimation

Consider the following revised TF model in which the sample delay δ is explicitly acknowledged,

$$y(k)=\frac{b_{\delta}\mathrm{z}^{-1}+b_{\delta+1}\mathrm{z}^{-(\delta+1)}+\ldots+b_{m}\mathrm{z}^{-m}}{1+a_{1}\mathrm{z}^{-1}+\ldots+a_{n}\mathrm{z}^{-n}}u(k) \tag{12}$$

Recalling that n is the number of output parameters, and $m-\delta+1$ is the number of input parameters. Equation (12) can be represented incrementally as

$$y(k) = -a_1 y(k-1) - \ldots - a_n y(k-n) + b_\delta u(k-\delta) + b_{\delta+1} u(k-(\delta+1)) + \ldots + b_m u(k-m) \tag{13}$$

for which the TF parameters, a_i $(i = 1, 2, \ldots, n)$ and b_j $(j = \delta, \delta+1, \ldots, m)$ are explicitly appeared. Equation (13) can be written in following form

$$y(k) = \mathbf{\Phi}^T(k)\boldsymbol{\xi} \tag{14}$$

given that:

$$\mathbf{\Phi}(k) = \begin{bmatrix} -y(k-1) & \ldots & -y(k-n) & u(k-\delta) & u(k-(\delta+1)) & \ldots & u(k-(\delta+m)) \end{bmatrix}^T$$
$$\boldsymbol{\xi} = \begin{bmatrix} a_1 & \ldots & a_n & b_\delta & b_{\delta+1} & \ldots & b_{\delta+m} \end{bmatrix}^T$$

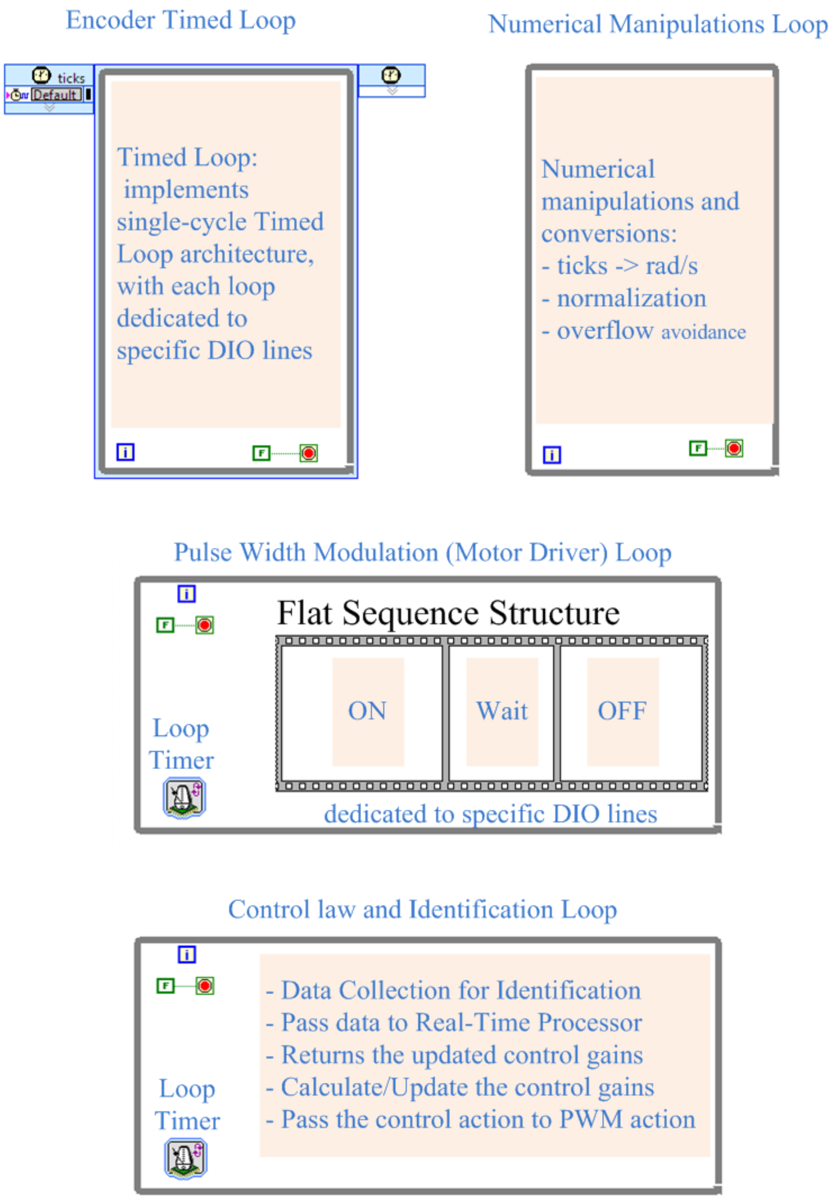
Encoder Timed Loop
ticks
Default
Timed Loop:
implements
single-cycle Timed
Loop architecture,
with each loop
dedicated to
specific DIO lines
Numerical Manipulations Loop
Numerical
manipulations and
conversions:
- ticks -> rad/s
- normalization
- overflow avoidance
Pulse Width Modulation (Motor Driver) Loop
Flat Sequence Structure
ON
Wait
OFF
Loop
Timer
dedicated to specific DIO lines
Control law and Identification Loop
- Data Collection for Identification
- Pass data to Real-Time Processor
- Returns the updated control gains
- Calculate/Update the control gains
- Pass the control action to PWM action
Loop
Timer

Figure 5 Preliminary FPGA layout with measuring, driving and control loops.

Linear Least Squares (LLS) can be used to estimate $\hat{\xi}$, based on the data samples collected; by the minimization of the error between the measured output $y(k)$ and the estimated model response $\mathbf{\Phi}^T(k)\hat{\xi}$. Therefore, the cost function, $J(\xi)$, takes the following form:

$$J(\xi) = \sum_k \left[y(k) - \mathbf{\Phi}^T(k)\hat{\xi} \right]^2 \tag{15}$$

The concept of solving, equation (15) for $\hat{\xi}$ is based on the partial differentiation of the cost function $J(\xi)$ with respect to each element in the vector ξ, i.e., $\partial J(\xi)/\partial \xi = 0$. The resulting $n+m$ system of equations is solved simultaneously to estimate the vector of TF parameters $\hat{\xi}$ as follows

$$\begin{aligned} \hat{\xi} &= \left[\sum_k \mathbf{\Phi}(k)\mathbf{\Phi}^T(k) \right]^{-1} \left[\sum \mathbf{\Phi}(k) y(k) \right] \\ &= \mathbf{\Gamma\beta} \end{aligned} \tag{16}$$

Equation (16) can be used for preliminary estimate of the system parameters, $\hat{\xi}$**,** in an offline manner, provided that the matrix, $\mathbf{\Gamma}$ exists. In order to update the system parameters during the real-time operation; Recursive Least Squares (RLS) algorithm is used instead. At a given sample k, the matrix $\mathbf{\Gamma}^{-1}(k)$ and the vector $\boldsymbol{\beta}(k)$ can be estimated recursively as follows

$$\begin{aligned} &\mathbf{\Gamma}^{-1}(k) = \mathbf{\Gamma}^{-1}(k-1) + \mathbf{\Phi}(k)\mathbf{\Phi}^T(k) \\ &\boldsymbol{\beta}(k) = \boldsymbol{\beta}(k-1) + \mathbf{\Phi}(k) y(k) \end{aligned} \tag{17}$$

Using the matrix inversion lemma, which provides the inverse of the accumulated `cross-product' matrix $\mathbf{\Gamma}(k)$,

$$\mathbf{\Gamma}(k) = \mathbf{\Gamma}(k-1) - \mathbf{\Gamma}(k-1)\mathbf{\Phi}(k)\left[1 + \mathbf{\Phi}^T(k)\mathbf{\Gamma}(k-1)\mathbf{\Phi}(k)\right]^{-1}\mathbf{\Phi}^T(k)\mathbf{\Gamma}(k-1) \tag{18}$$

Regarding equation (16), the updated TF parameters vector at instant k is $\hat{\xi}(k) = \mathbf{\Gamma}(k)\boldsymbol{\beta}(k)$, which yields

$$\hat{\xi}(k) = \hat{\xi}(k-1) + \mathbf{\Gamma}(k-1)\mathbf{\Phi}(k)\left[1 + \mathbf{\Phi}^T(k)\mathbf{\Gamma}(k-1)\mathbf{\Phi}(k)\right]^{-1}\left[y(k) - \mathbf{\Phi}^T(k)\hat{\xi}(k-1)\right] \tag{19}$$

Equations (18 and 19) summarize the RLS algorithm used to update the TF parameters in the LabVIEW VI.

Primarily, the collected data of the normalized input, scaled from -1000 to 1000, applied to the DC motor driver, and the corresponding measured angular velocity of the motor' shaft shows that the best form of the TF that best describe the dynamic behavior of the mechatronic system is

$$\frac{y(k)}{u(k)} = \frac{b_4 z^{-4}}{1 + a_1 z^{-1}} \tag{20}$$

As shown in equation (20), the system is first order, with 4-samples delay, for which $y(k)$ is the angular velocity of the DC motor in rad/s. Also, TF parameters values $a_1 = -0.8993$ and $b_4 = 0.0024$ return a coefficient of determination $R_T^2 = 0.96$.

4.2 PIP controller design and implementation

Based on the identified TF, equation (20), the NMSS formulation takes the form of equation (4), for which the state vector $\mathbf{x}(k)$, the state transition matrix $\mathbf{F}$, and the associated input vector $\mathbf{g}$ are presented as

$$\mathbf{x}(k) = \begin{bmatrix} y(k) & u(k-1) & u(k-2) & u(k-3) & z(k) \end{bmatrix}^T$$

$$\mathbf{F} = \begin{bmatrix} -a_1 & 0 & 0 & b_4 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ a_1 & 0 & 0 & -b_4 & 1 \end{bmatrix} \tag{21}$$

$$\mathbf{g} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T$$

According to equation (6), the SVF gain vector has 5 elements and takes the following form

$$\mathbf{K} = \begin{bmatrix} f_o & g_1 & g_2 & g_3 & -k_I \end{bmatrix}^T \tag{22}$$

The incremental form of the control law can then be written as

$$\begin{aligned} u(k) &= u(k-1) - f_o(y(k) - y(k-1)) - g_1(u(k-1) - u(k-2)) \\ &\quad - g_2(u(k-2) - u(k-3)) - g_3 u(k-3) - u(k-4)) + k_I(y_d(k) - y(k)) \end{aligned} \tag{23}$$

The control law depicted in equation (23) should be constructed upon the FPGA chip, taking into consideration the space occupied by mapping the variable into fixed-point numbers before converting it into integer data types within LabVIEW environment, such that the control action sub-VIs avoids values overflow. This PIP control action is then fed-forward to the PWM motor driver via sbRIO-9631 for implementation process. The compilation process is carried out, using LabVIEW-FPGA compiler.

Three values for the weightings r and $\mathbf{Q}$ have been tried out, the corresponding control gain vectors, $\mathbf{K}$, are evaluated using equation (11) and Table (1) summarizes their numerical values. The simulation of the three controllers give acceptable closed loop responses with little to choose between them, see Figure (6), however the third controller is chosen since it gives relatively fast closed loop response.

Table (1) Three trials of PIP control gains using different weightings

#	r	**Q**	$\mathbf{K}^T$
I	0.25	$\begin{bmatrix}1 & 0.25 & 0.25 & 0.25 & 1\end{bmatrix}$	$\begin{bmatrix}8.2538 & 0.0214 & 0.0216 & 0.0218 & -0.9893\end{bmatrix}$
II	0.25	$\begin{bmatrix}1 & 0.25 & 0.25 & 0.25 & 10\end{bmatrix}$	$\begin{bmatrix}23.1323 & 0.0584 & 0.0599 & 0.0611 & -3.0710\end{bmatrix}$
III	0.25	$\begin{bmatrix}1 & 0.25 & 0.25 & 0.25 & 20\end{bmatrix}$	$\begin{bmatrix}30.9837 & 0.0772 & 0.0797 & 0.0819 & -4.3023\end{bmatrix}$

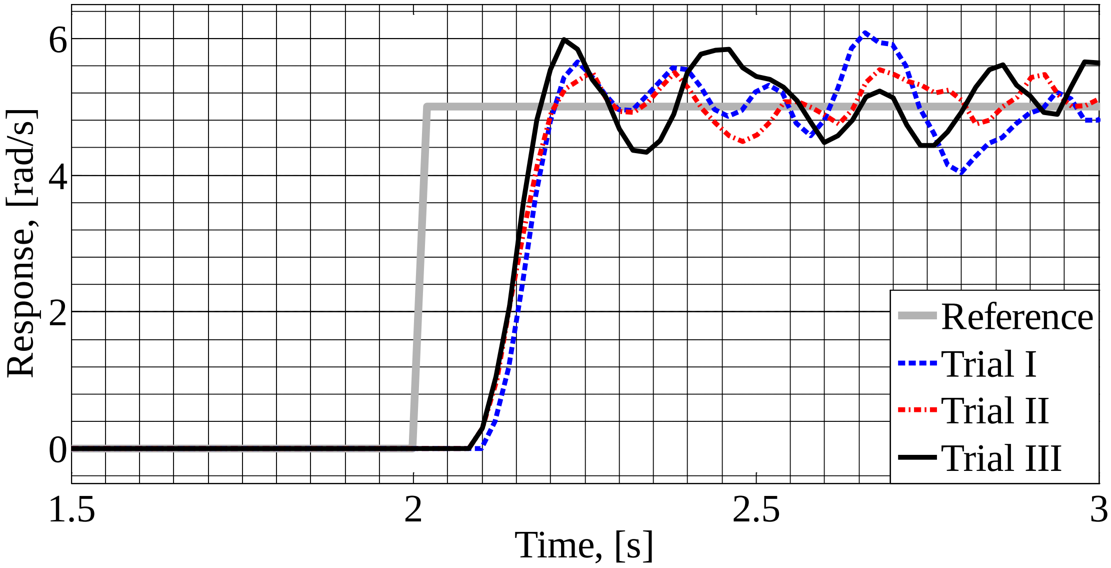

Figure 6 The simulation of the closed loop responses of the three selected controller, see Table (1).

The selected controller has been implemented twice to the mechatronic demonstrator; once using the real-time processor of the sbRIO-9631 and another using the FPGA device. The two practical responses are plotted in Figure (7). It is obvious that the reconfigurable PIP-FPGA control system is faster by about a factor of 1.5 than PIP real-time. This is expected since the parallel processing effectively reduces the calculation time. On other words, the PIP-FPGA calculates the control action faster than PIP/real-time; this allows the PIP-FPGA control action to take place earlier than PIP real-time. This causes the demonstrator to interact with PIP-FPGA control action before the end of the time step with a considerable time.

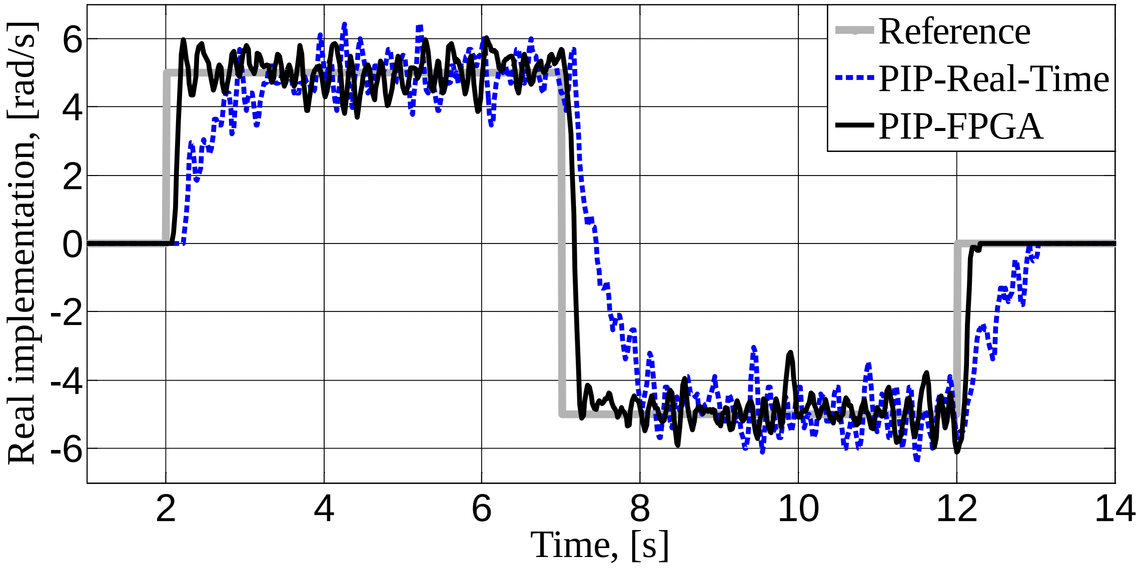

Figure 7 Practical implementation of PIP control using both real-time (dashed) and FPGA (solid) applied on mechatronic demonstrator.

5. Summary and Conclusion

The work shows a successful implementation of PIP control algorithm within Reconfigurable IO board with FPGA technology. Both transfer function parameters estimation step and calculating the control of the updated transfer function parameters are employed to a fast mechatronic demonstrator. The control law has been applied twice via real-time processor and FPGA device.

The implementation of PIP over reconfigurable IO board with FPGA technology shows a considerable enhancement in the response time by a factor of 1.5 over the real-time processor. This is due the parallel processing available within the FPGA chip. The construction of parallel circuits for the control algorithm minimizes the calculation time. This fact allows the control action to take place before the end of the time step by a considerable time. However, the disadvantages are limited in the memory size and the arithmetic manipulation. Signed-Fixed-point math with appropriate range is used in order to simplify the computations and hold the suitable memory size.

Acknowledgement

The authors are grateful to Jazan University, Automatic Control Laboratory, for using the licensed LabVIEW software package.

References

1. Friedland, B. (1986), Control System Design: An Introduction to State-Space Methods, McGraw-Hill, Inc., New York.
2. Maciejowski, J. (1989), Multivariable feedback design, Addison-Wesley Publishers Ltd., Cornwall, UK.
3. Kemin Zhou, John Doyle, (1997), Essentials of Robust Control, Prentice Hall.
4. Skogestad, S. and Postlethwaite, I. (2005), Multivariable Feedback Control: Analysis and Design, Second Edition, John Wiley & Sons, Ltd, London.
5. Maciejowski, J. (2002) Predictive Control with Constraints, Prentice Hall, Harlow, UK.
6. 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.
7. C. James Taylor, Arun Chotai, Peter C. Young, 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.
8. James Taylor, C.; Chotai, A.; Young, P.C, 2001, "Design and application of PIP controllers: robust control of the IFAC93 benchmark", Transactions of the Institute of Measurement & Control, 2001, Vol. 23 Issue 3, p183-200
9. M.J. Lees, C.J. Taylor, P.C. Young, A. Chotai, (1998), "Modelling and PIP control design for open-top chambers", Control Engineering Practice, Volume 6, Issue 10, October 1998, Pages 1209-1216, ISSN 0967-0661, http://dx.doi.org/10.1016/S0967-0661(98)00089-6.

10. C.J. Taylor, P.C. Young, A. Chotai, A.R. McLeod, A.R. Glasock, (2000), "Modelling and Proportional-integral-plus Control Design for Free Air Carbon Dioxide Enrichment systems", Journal of Agricultural Engineering Research, Volume 75, Issue 4, April 2000, Pages 365-374, ISSN 0021-8634, http://dx.doi.org/10.1006/jaer.1999.0515.
11. P.C. Young, M.J. Lees, A. Chotai, W. Tych, Z.S. Chalabi, (1994), "Modelling and PIP control of a glasshouse micro-climate", Control Engineering Practice, Volume 2, Issue 4, August 1994, Pages 591-604, ISSN 0967-0661, http://dx.doi.org/10.1016/0967-0661(94)90003-5.
12. Laura Price, Peter Young, Daniel Berckmans, Karl Janssens, James Taylor, (1999), "Data-based mechanistic modelling (DBM) and control of mass and energy transfer in agricultural buildings", Annual Reviews in Control, Volume 23, 1999, Pages 71-82, ISSN 1367-5788, http://dx.doi.org/10.1016/S1367-5788(99)90063-9.
13. C.J. Taylor, P. Leigh, L. Price, P.C. Young, E. Vranken, D. Berckmans, (2004), "Proportional-integral-plus (PIP) control of ventilation rate in agricultural buildings", Control Engineering Practice, Volume 12, Issue 2, February 2004, Pages 225-233, ISSN 0967-0661, http://dx.doi.org/10.1016/S0967-0661(03)00060-1.
14. 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.
15. Dixon, R, Chotai, A, Young, PC, Scott, JN (1997), "The Automation of Piling Rig Positioning Utilising Multivariable Proportional-Integral-Plus (PIP) Control". In Proc. 11th International Conference on Systems Engineering, Coventry University, Great Britain, pp.211-216, ISBN: 0905949633.
16. Gu, Jun, Taylor, James, Seward, Derek, 2004, "Proportional-Integral-Plus Control of an Intelligent Excavator", Computer-Aided Civil and Infrastructure Engineering, Vol. 19, Issue 1, PP.16-27 , 2004. http://dx.doi.org/10.1111/j.1467-8667.2004.00334.x
17. 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.
18. 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.
19. C J Taylor, A P McCabe, P C Young, and A Chotai, 2000, "Proportional-integral-plus (PIP) control of the ALSTOM gasifier problem" Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, September 1, 2000; vol. 214, 6: pp. 469-481.
20. 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.
21. 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.
22. C.J. Taylor, A. Chotai, P.C. Young, 2009, "Non-linear control by input--output state variable feedback pole assignment", International Journal of Control Vol. 82, Iss. 6, 2009.
23. C. James Taylor, David Robertson, 2013, "State-dependent control of a hydraulically actuated nuclear decommissioning robot", Control Engineering Practice, Volume 21, Issue 12, December 2013, Pages 1716-1725.
24. C.J. Taylor, P.C. Young, A. Chotai, (2013), True digital control Statistical modelling and non-minimal state space design, John Wiley and Sons Ltd.
25. 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).

26. Young, P.C., 1998. Data-Based mechanistic modeling of engineering systems. Journal of Vibration and Control, (4), pp. 5--28.
27. P.C. Young, H. Garnier, 2006, "Identification and estimation of continuous-time, data-based mechanistic (DBM) models for environmental systems", Environmental Modelling & Software, Volume 21, Issue 8, August 2006, Pages 1055-1072, ISSN 1364-8152, http://dx.doi.org/10.1016/j.envsoft.2005.05.007.
28. M. Nakamura, S. Goto, N. Kyura, (2004), Mechatronic Servo System Control Problems in Industries and their Theoretical Solutions, Springer-Verlag Berlin Heidelberg, NewYork.
29. Bogdan M. Wilamowski and J. David Irwin, (2011), Control and Mechatronics, Second Edition, CRC Press, Taylor and Francis Group, NW.
30. C.J. Taylor, A. Chotai, P. Cross, 2012, "Non-minimal state variable feedback decoupling control for multivariable continuous-time systems", International Journal of Control, Vol. 85, Iss. 6, 2012.
31. Cross, Philip and Taylor, C. James and Aggidis, George A., 2010, "Continuous-time feed-forward proportional-integral-plus control", In: UKACC International Conference Control 2010, pp. 229-234. ISBN 978-184600-0386.
32. Roger Woods, John McAllister, Gaye Lightbody, Ying Yi, (2008), FPGA-based Implementation of Signal Processing Systems, John Wiley & Sons Ltd, UK.
33. Rahul Dubey, (2009), Introduction to Embedded System Design Using Field Programmable Gate Arrays, Springer-Verlag, London.
34. National Instruments Corporation, (2004), LabVIEW - FPGA Module User Manual, USA. www.ni.com .
35. National Instruments Corporation, (2010), LabVIEW - FPGA Course Manual, Part Number 372510D-01, USA. www.ni.com .
36. Astrom, K.J., Wittenmark, B. (1984) Computer-Controlled Systems Theory and Design, Prentice Hall, Englewood Cliffs, NJ.
37. O'Dwyer, Aiden, (2009). Handbook of PI and PID controller tuning rules, 3rd edition, World Scientific.
38. Taylor, C.J., Pedregal, D.J., Young, P.C. and Tych, W. (2007), "Environmental Time Series Analysis and Forecasting with the Captain Toolbox", Environmental Modelling and Software, 22,797-814 (http://dx.doi.org/doi:10.1016/j.envsoft.2006.03.002).