International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 5 Issue 11 ǁ November. 2017 ǁ PP. 32-36

Validation of an Unrestricted DMC Controller Implemented on Raspberry PI III

[1]Walter Luiz Ferreira dos Anjos, [1]Richard Junior Manuel Godinez Tello
[1]Postgraduate Program in Automation and Control Engineering - IFES Campus Serra, Brazil

ABSTRACT ***:*** *The goal of this work is theimplementation of a Predictive Controller DMC unrestricted (Dynamic Matrix Control) in Python programming language, shipped in Raspberry PI III, for the control of a water supply system. The implementation aims the reduction of project costs, use of free software, in addition to enabling the use of Raspberry hardware with Internet Protocol (IP) for outgoing interface.The goal of controller was acting in the control of motor speed of a motor pump set of a treated water lift, so that by monitoring the pressure at the critical point of supply, to control engine speed, ensuring the required minimum pressure of supply of 10mCA(metres of water column).*

Keywords -*Controller, DMC, Raspberry PI*

I. INTRODUCTION

Due to the growing demand for water and the reduction of water resources, water distribution companies, seek way to supply pressure control more and more effective, to make available to customers a minimum supply pressure established by standard, control the rate of leaks in the supply network and reduce energy consumption. The technique usually used to control the Lifting stations supply is the use of Proportional-Integral-derivative Controller (PID).which has the characteristics of simple and stable and has been widely used by engineers. PID control is still a stable and efficient method in the control of stable systems. These controllers are accepts to many control problems when the dynamics of the process is favorable, but are hard to tune in more complex situations. The use of a predictive controller for the control of supply, seeks to evaluate your performance.

The Dynamic matrix Control (DMC) is a type of MPC. He appeared in 1979, as a methodology able to handle operational constraints on the control variables and output systematically. This algorithm is suitable for the control of processes with a high degree of interaction between the variables, orders high and dominant delays and ability to handle constraints. The DMC algorithm uses a model obtained from the step response of the process.

Its implementation was done in Python programming language and a Raspberry PI III board. The output of the controller was used to trigger the frequency Inverter to control the pump speed, and consequently the supply pressure.

II. RASPBERRY PI III

A.The Raspberry PI III

The Raspberry PI III is shown in figure 1. It is a portable, single board microcomputer, with ARM processor 1.2 Ghz, quad-core, 64-bit produced by Raspberry Pi Foundation. Has 1 GHz RAM, 4 USB ports, full support for HDMI, wireless LAN and Bluetooth. Uses a linux-based operating system with boot from an SD card. It has a GPIO with 40 pins, with I2C communication, digital inputs and outputs, a USART, among others.

(a) Raspberry (b) GPIO

Figure. 1 Raspberry PI

B.The Python Language

The Raspberry PI III has installed Python version 2 and 3. Python is a language versatile, used in Web development and many other types of applications. The name "Python"is a tribute to the English comedy group Monty Python. It runs on servers of mega-sites like Google and YouTube, in the printers computing clusters Industrial Light and Magic, in NASA laboratories and AstraZeneca Pharmaceuticals, and in games such as Civilization IV and EVE-Online. Python offers the following features also available in more complicated languages such as Java and C++.
In Process control, Python has Control-specific libraries (Control) and manipulation of arrays (Numpy), with functions similar to Matlab Software. It Has diversified functions for plotting data in multiple libraries (matplotlib). These libraries contains a set of classes and functions that implement common operations for analysis and design of control systems. Python offers Compatibility Control Pack with Matlab (control.matlab) is available and offers a lot of functions corresponding to common commands in Matlab Control Systems Toolbox. Python-control must be imported into the python import command. Python-Control uses the Numpy, which is the basic package for scientific computing Python. It is a Python library which supports arrays and multidimensional arrays,owning a large collection of mathematical functions for quick operations with arrays,including mathematical, logical operations, handling forms, selections, i/o, Fourier transform, basic linear algebra, basic statistics operations, simulations, etc.

III. PREDICTIVE DMC CONTROL SYSTEM

A.Predictive DMC Controller

The model-based Predictive Control, MPC(Model Predictive Control) is a type of control algorithm, which uses the process model, to predict the behavior of the system controlled. Every sampling period, the optimization problem is solved and is calculated future actions to control. However,just the first control action is sent to the plant. In the next time, the parameters of the model are updated with the data of the plant and the optimization problem is recalculated. The use in the industry model-based Predictive control is growing, mainly because your feature to deal with the constraints of the process, as well as your anticipatory action to correct the error. The technique of type DMC is showing widely accepted due to the simplicity of the algorithms and the use of the step response model, which requires little information for the dynamic system characterization, used to predict the effect of future control actions on exit, determined by minimizing errors in the future.The approach of MPC controllers is presented in Figure 2.

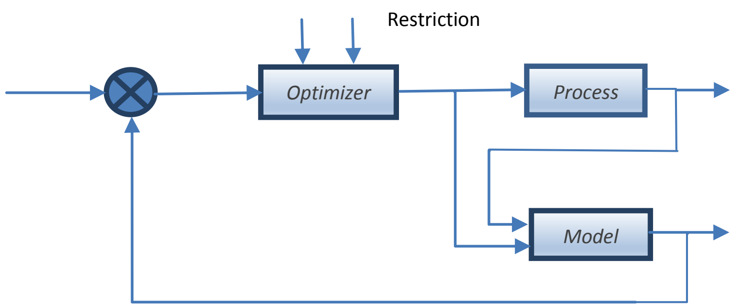

Figure. 2 Predictive MPC Controller

The future outputs for a given N, called horizon of prediction, are predicted for each time t, using the process template. the outputs predicted ŷ (t + k|t), for k = 1 ... N depend on the knowledge of the values until the moment "t" (past inputs and outputs) and signs of future control u (t + k) j|t), k = 0 .. N-1, which are provided by the system and calculated.
The set of futures control signals arecalculated by a certain optimization role to keep the process as close as possible to the path of reference w (t + k), that can be the set point, or an approximation thereof. This role usually takes the form of quadratic function the error, between the sign of predicted output and the reference predicted trajectory. The control effort is included in objective function in most cases.

Control signal u (t) is forwarded to the process, while others are rejected because in the next instant of time y(t + 1) be known, the steps be replayed.
The goal of the DMC controller is to calculate the actions of control, represented by the manipulated variables, increments through the minimization of cost equation (1):

$$J(t) = \sum_{k=N_1}^{N_2} \lambda_y [\hat{y}(t+k) - w(t+k)]^2 + \sum_{k=1}^{N_u} \lambda \, \Delta u^2(t + k - 1)$$

Equation 1

Where N_2 is the output forecast horizon, the horizon ($Nu \leq N_2$). λy and λu are the weights in the control and output signals, respectively. $\hat{y}(t + k)$ is the output forecast at the moment (t + k), using the information available at time t. w (t) is the reference at time t and Δu (t) is the increase in the control signal at time t.
The prediction along the horizon is given by (2):

$$\hat{y}(t+k|t) = \sum_{i=1}^{k} g_i \, \Delta u(t + k - i) + \sum_{i=k+1}^{N} g_i \, \Delta u(t + k - i)$$

Equation 2

Where $\sum_{i=k+1}^{N} g_i \, \Delta u\left(t + k - i\right)$ is the free response of the system. This and the part of the answer that does not depend on future control actions, and is given by (3):

$$f(t+k) = y_m(t) + \sum_{i=1}^{\infty} (g_{k+1} - g_i) \, \Delta u(t - i)$$

Equation 3

Where g_i is obtained from the system step response.

IV. DESIGN OF DMC CONTROL SYSTEM

A.Process Description

The area under test comprised a supply sector of a sanitation company, responsible for the supply of 1500 People. A booster shown in figure 3,consisting of a motor-pump set of 7.5 CVs is responsible for the supply of treated water. This booster has a frequency Inverter triggering a motor pump set. At the critical point of the supply sector was installed a pressure transmitter.

Figure. 3 Supply structure

The acquisition of the data was carried out in the frequency range lift operation 45 to 56hz. 1500 samples were taken, figure 4.

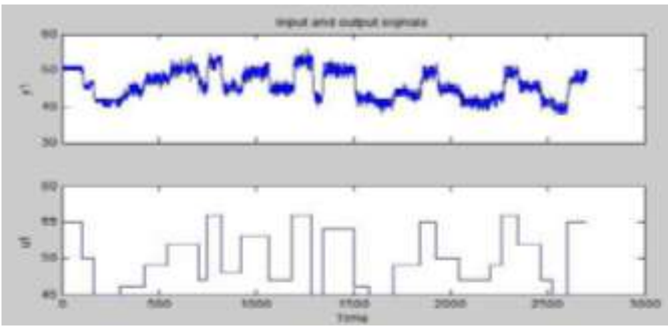

Figure. 4 Colected data

The identification and validation of transfer function relating the pressure at the critical point and the frequency of operation of the motor pump set, were made using the Ident tool of Matlab. (4)

$$G(s) = \frac{0.96e^{-2.59s}}{s(1 + 29.81s)}$$

Equation 4

B.Python Control DMC Implementation

For this project implementation in Python programming language was used mainly Control, matplotlib and numpy libraries.Due the Raspberry PI does not have analog inputs, we used the converter analog/digital PIC 16f877A as input interface, which makes reading theset-point and the process variable, converts to digital and passes to the Raspberry PI via I2C Protocol. How I2C communication each byte is transmitted at a time and the signal converted has two bytes, each variable is broadcast with two bytes. SPL/Sph are the bytes from the set-point and PvL/PvH are the bytes of the process variable, and are mounted in Raspberry after reception.
The prediction horizon, horizon, sampling interval control and weighting control signals and output have been set as follows:
HP = 7 (HP is the forecast horizon)
HC = 5 (HC is the control horizon)
TS =. 04 (sampling interval) Limit = 1800/Ts
lick =. 8 (Consideration of the control signal)
Alpha = 0.3 (factor reference forecast)
Delta = 0.6 (output signal Weighting)
The control signal output served directly on the PWM of the Raspberry PI, controlling the inverter. The range of PWM DutyCycle varies from 0 to 100%, where 0 represent 45Hz frequency and 100% represent 56Hz of operating frequency.
To test implemented controller, was generated a random reference signal (r - linein blue). Based on cost function, the control signal from the controller (u - line in purple) is generated controlling the output(y - line in green), figure 5.

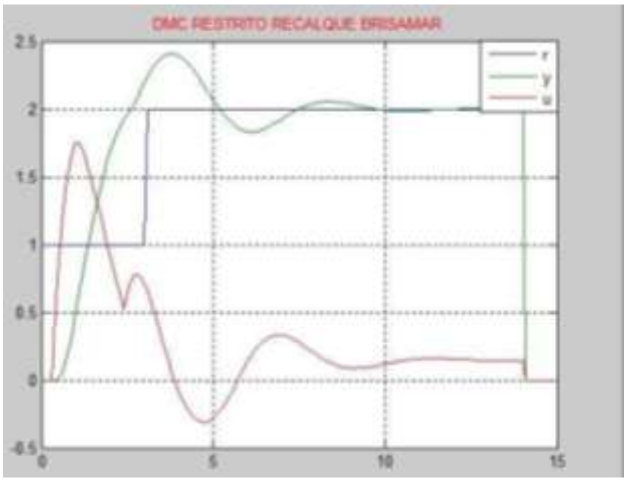

Fig. 5 Implemented DMCController

To evaluate the DMC controller, was implemented a PID controller and tuned by the Ziegler-Nichols method, where through comparison of performance index, settling time and overshoot.

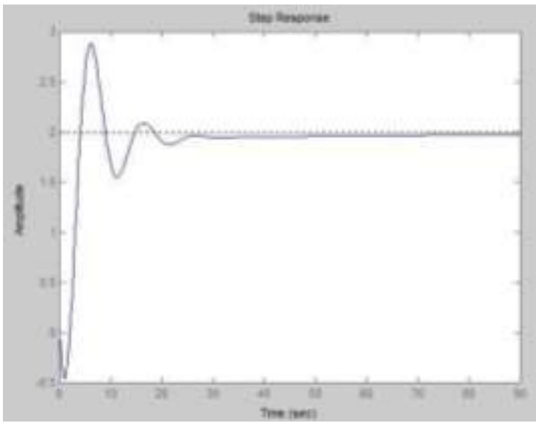

Fig. 6 Implemented PID signal.

V. CONCLUSION

The results obtained showed the feasibility of implementing the DMC in Python language. Although there are differences in implementation in relation to Matlab, the two languages control functions and a variety of compatible libraries, being the main advantage of implementing cost reduction, since Python is free.

Raspberry implementation IP enabled boarding the controller in a compact device with wide variety of features, which allowed the direct interface to the drive.

Both controllers in the test process stabilized, but note the anticipatory action in the DMC predictive control, providing a faster response of the front controller set-point variations.

REFERENCES

[1] G. M. Almeida and j.l. f. Salles,*Controle Preditivo. Sintonia e Aplicações na Siderurgia (Vol. 1, 75-95, 2016).*

[2] E.F. Camacho and c. Bordons, *Model Predictive Control, (2007).*

[3] L. Campestrini *Decentralized PID Controllers tuning, based on the method the critical point,* master's thesis in electrical engineering, Federal Universidade of Rio Grande do Sul, (2006).

[4] P. S. Ch, *Control to Distributed with a view to Minimizing Pumping energy costs applied to water supply systems*. 150F. Thesis (Doctorate in mechanical engineering)-Federal University of Paraíba (2012).

[5] F.R. Fonseca, *Model for automation of Water supply systems,* master's Thesis in electrical engineering-Politécnica School of University of São Paulo, (2009).

[7] K. Ogata, *Modern Control Engineering, vol. 5, (2010).*