HEFAT2007
5th International Conference on Heat Transfer, Fluid Mechanics and Thermodynamics
Sun City, South Africa
Paper number: MN2

OPTIMIZATION OF PID CONTROL PARAMETERS USING THE POLE-PLACEMENT APPROACH

Naison Mazana* and Bernard Muller
*Author for correspondence
Thermo-Hydraulics Analysis Group,
Pebble Bed Modular Reactor Proprietary Ltd
Centurion 0046,
South Africa,
E-mail: naison.mazana@pbmr.co.za

ABSTRACT

This paper presents an analytical method of determining optimum PID parameters using the pole placement approach followed by the closed-loop analysis of Pebble-Bed Modular Reactor (PBMR) reactor temperature control employing thermo-fluid simulation software. The proposed method is a reliable alternative to the Ziegler Nichols, Cohen-and-Coon and other approaches that normally suggest initial controller parameters that would require further optimization. The pole placement approach forces the positions of the poles of the closed-loop characteristic equation to the stable region in the z-domain. In this approach the plant model is represented by an Auto-Regressive Moving Average (ARMA) black box model and the parameters of the model are determined by employing the Batch Least Squares approach. The pole-placement method reduces the determination of PID parameters to the setting of only a single parameter referred to as the tailoring coefficient of a first order tailoring polynomial. The controller settings can be selected with the value of the tailoring coefficient migrating from zero towards the unit circle at -1 in the *z*-domain. This progression results in closed loop responses that range from underdamped ($t_1 = 0$) to overdamped as t_1 approaches -1. Although the mathematics behind the approach is quite involved, in this work the methodology has been transformed into a user-friendly MATLAB© based calculation. The above method is applied to the control of pebble-bed modular nuclear reactor (PBMR) temperature by manipulating the reactor activity. The closed loop transients are generated from a Flownex© thermal-hydraulics modelling/simulation environment.

INTRODUCTION

When The main thrust of model identification involves the derivation of a model reflecting all the available information concerning the process, the nature of the load disturbance, whether the system is linear or nonlinear, whether the system is time varying or not, and as well as the correct choice of the model order [Mazana 1995]. Depending on the dynamics of the process and on the quality of the control required, the lowest possible model order is normally chosen. First and second order models are normally used for control design purposes since further increase in model order is generally found to provide little improvement in controller performance whilst substantially increasing the computational burden. In this work we have adopted a second order deterministic model as the basis for our investigations.

The most widely employed approach in model identification is the identification of an estimate of the unknown parameter vector which maximizes or minimizes a function determined by a chosen criterion, e.g. the sum of least squares of the residuals in the case of the least squares method. It may be sufficient to collect the input/output data and carry out an offline batch regression of the data but, for processes with time-varying dynamics, adaptive control algorithms should be applied which employ online inference of system properties through recursive parameter estimation.

The most important factors influencing the success of identification processes are [2]:

(i) The type of excitation applied to the system,
(ii) The choice of the output variables which can be measured and the precision of the measurements,

(iii) The signal to noise ratio,
(iv) The properties of the noise,
(v) The method of identification,
(vi) The numerical procedure underlying the calculation of the parameter estimates,
(vii) The redundancy of parameters in the regression model, and
(viii) The sampling period and the number of bits in computer word employed.

To the above should be added the experience of the operator with the system and the operator's own intuitive judgment during on-line identification procedures.

In this work we will employ the least squares (LS) parameter estimator for the identification of the controller parameters.

NOMENCLATURE

$A(z^{-1})$	[-]	Monic polynomial in the z-domain representing the poles of the discrete-time system
$B(z^{-1})$	[-]	Polynomial in the z-domain representing the zeros discrete-time system
$C(z^{-1})$	[-]	Monic polynomial in the z-domain representing the zeros of the discrete-time noise
A^{-1}	[-]	Inverse of matrix A
A^{T}	[-]	Transpose of matrix A
$\theta(t)$	[-]	Model parameter vector
$R(z^{-1})$	[-]	Differencing factor
$T(z^{-1})$	[-]	Tailoring polynomial
$\Phi(t)$	[-]	Matrix of past process outputs and inputs
$\xi(t)$	[-]	Noise term

Special characters

n_a	[-]	Process model order
n_b	[-]	Length of process input data
K_I	[-]	Integral gain
K_C	[-]	Proportional gain
K_D	[-]	Derivative gain

Subscripts

k	Process dead time

BATCH LEAST SQUARES PARAMETER ESTIMATION

System parameter estimation entails the use of observed input/output process data online or offline to estimate the a_i and b_i coefficients of a CARMA model. The most widely employed approach is the identification of an estimate of the unknown parameter vector which maximizes or minimizes a function determined by a chosen criterion e.g. the sum of the squares of the residuals in the case of the least squares method. It may be sufficient, as is the case in the current work, to collect the input/output data and then carry out an offline batch regression of the data. The CARMA model that is adopted as the black box model representation of the process to be controlled is as follows [2], [3]:

$$A(z^{-1})y(t)=z^{-k}B(z^{-1})u(t)+C(z^{-1})\xi(t) \qquad (1)$$

Consider the ARMA model represented by the following equation [1]:

$$\begin{aligned} y(t) &= -a_1y(t-1)-....-a_{na}y(t-n_a)+b_0u(t-1)+...+b_{nb}u(t-n_b) \\ &= \theta(t)\phi(t)+\varepsilon(t) \end{aligned} \qquad (2)$$

where:

$y(t-i)$: $i=0..n_a$, are the actual present and past plant outputs,

$u(t-i)$: $i=0..n_b$, are the inputs to the plant,

$\varepsilon(t)$ is the error between the model and the plant outputs, and $\theta(t)\phi(t)$ is the estimated model.

For the ARMA model described by equation (1) we take:

$$\mathrm{Y}(t)=\left[y(t),....,y(t-n_a),u(t),....,u(t-n_b)\right]^{\mathrm{T}} \qquad (3)$$

$$\Phi(t)=\begin{bmatrix} y(n-1).........y(0) & :u(n).......u(0) \\ y(n) \quad ....y(1) & :u(n+1)...u(1) \\ y(t-1)..........y(t-n_a) & :u(t).........u(t-n_b) \end{bmatrix} \qquad (4)$$

$$\theta=\left[-a_1(t),....,-a_{na}(t),b_0(t),....,b_{na}(t)\right]^{\mathrm{T}} \qquad (5)$$

$$\mathrm{E}(t)=\left[\varepsilon(t),\varepsilon(t-1),....,\ \varepsilon(0)\right]^{\mathrm{T}} \qquad (6)$$

Given the input/output data for measurements carried out over time t, the estimate of the parameter vector $\theta(t)$ which minimizes the least squares optimization criterion is:

$$\begin{aligned} \theta(t) &= \left[\Phi^{\mathrm{T}}(t)\Phi(t)\right]^{-1}\Phi^{\mathrm{T}}(t)\mathrm{Y}(t) \\ &= \mathrm{P}(t)\Phi^{\mathrm{T}}(t)\mathrm{Y}(t) \end{aligned} \qquad (7)$$

provided that the matrix $\mathrm{P}(t)$ is positive definite or is capable of being inverted.

The ARMA model (2) employed in this work is adopted in its second order version which reduces the parameter estimation process to the identification of three parameters: a_1, a_2 and b_0 as shown in Equation (8) below:

$$y(t)=-a_1y(t-1)-a_2y(t-2)+b_bu(t-k) \qquad (8)$$

where $a_1=0.6866$, $a_2=-0.2982$,

and $b_k=28.2982$ are process model parameters and $k=1$ is the process dead time.

Past experience [7] has shown that the calculated parameters get closer to their ideal values as the length of the process response data matrix becomes larger. This is a reasonable observation, as the larger data matrix provides the estimator with more historical data for curve fitting. However, extremely large data vectors have been seen to cause deterioration of the parameter values as the effect of ancient historical data becomes more and more significant. The manipulation of matrices of such large order in equation (7) may also not be very attractive when using less powerful computers. The resultant deterioration of parameter convergence for certain large data vectors may indicate that the use of some form of forgetting factor of past data may be required in order to improve parameter convergence in the recursive form of the recursive least squares (RLS) routine. The reader is referred to [7] for an in-depth discussion of various versions of the least squares routine.

THE POLE-PLACEMENT ALGORITHM

Consider a process represented by a discrete-time CARMA model of the form [4]:

$$A(z^{-1})y(t) = z^{-k}B(z^{-1})u(t) + C(z^{-1})\xi(t) \tag{9}$$

where:

$$\left.\begin{aligned} &A(z^{-1}) + 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots\ldots + a_{na} z^{-na} \\ &B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots\ldots + b_{nb} z^{-nb} \end{aligned}\right\} \tag{10}$$

and $n\alpha = 2, n\beta = 0$ are the orders of the *A* and *B* polynomials.

The general structure of the self-tuning PID (STPID) algorithm is given by:

$$R(z^{-1})u(t) = S(z^{-1})[w(t) - y(t)] = S\varepsilon(t) \tag{11}$$

where *R* is the differencing factor (i.e. $R = 1 - z^{-1} = \Delta u$) and *w(t)* is the reference set point. *S* is the error filtering polynomial of the form:

$$S(z^{-1}) = s_0 + s_1 z^{-1} + s_2 z^{-2} + \ldots..$$

$$\Delta u(t) = s_0\varepsilon(t) + s_1\varepsilon(t-1) + s_2\varepsilon(t-2) + \ldots\ldots.. \tag{12}$$

Taking *S* to be second order, and ε (t) = *w(t) - y(t)* Equation (12) becomes:

$$\Delta u(t) = s_0\varepsilon(t) + s_1\varepsilon(t-1) + s_2\varepsilon(t-2) \tag{13}$$

Substituting Equation (11) into Equation (9) yields the closed-loop equation:

$$y(t) = \frac{z^{-k}BS}{AR + z^{-k}BS}w(t) + \frac{CR}{AR + z^{-k}BS}\xi(t) \tag{14}$$

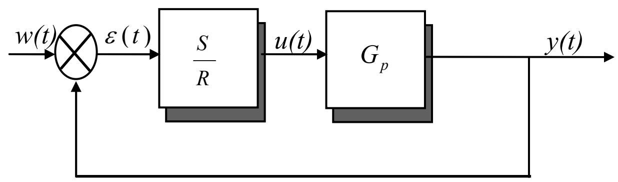

Figure 1 Structure of the feedback loop

The pole-placement approach implies the movement of the closed-loop poles of the characteristic equation in Equation (14) from their open loop locations to prescribed values given by a preset tailoring polynomial:

$$T = 1 + t_1 z^{-1} + t_2 z^{-2} + \ldots\ldots \tag{15}$$

and for the NNW-PID control strategy the following pole placement algorithm is used:

$$AR - z^{-k}BS = T \tag{16}$$

The polynomial *T* is chosen to be of order corresponding to the desired controlled response. The roots of this polynomial are set to within the unit circle in the *z*-domain for stable control. In this work a first order polynomial was found to be adequate.

The solution is arrived at by treating this as an identity and equating coefficients of like powers of *z*. For a maximum order of *T* equal to 3:

$n_a \le 2$; $n_b = 0$; $n_s = 2$; $k = 1$ [1].

where n_a, n_b, and n_s are the orders of the *A, B*, and *S* polynomials respectively, and k is the dead time. This gives the following polynomials in the identity in equation (16):

$$\left.\begin{aligned} &A(z^{-1}) = a_0 + a_1 z^{-1} + a_2 z^{-2} \\ &B(z^{-1}) = b_0 \\ &T(z^{-1}) = t_0 + t_1 z^{-1} + t_2 z^{-2} + t_3 z^{-3} \\ &R(z^{-1}) = 1 - z^{-1} \end{aligned}\right\} \tag{17}$$

The solution to Equation (16) gives the following set of simultaneous equations for *k=1*:

$$\left.\begin{aligned} s_0 &= (t_1 - a_1 + 1)/b_0 \\ s_1 &= (t_2 - a_2 + a_1)/b_0 \\ s_2 &= (t_3 + a_2)/b_0 \end{aligned}\right\} \tag{18}$$

The digital incremental PID control law has the form [6]:

$$\Delta u(t)=\left\{K_C+\frac{K_I T_C}{2}+\frac{K_d}{T_C}\right\}e(t)+\left\{\frac{K_I T_C}{2}-K_C-2\frac{K_D}{T_C}\right\}e(t-1)+\frac{K_D}{T_C}e(t-2) \quad (19)$$

where K_C, K_I, and K_D are the proportional, integral and derivative PID controller gains. T_C is the control period. Comparing Equations (13) and (18) we get:

$$\left.\begin{aligned} s_0 &= \left\{K_C+\frac{K_I T_C}{2}+\frac{K_d}{T_C}\right\} \\ s_1 &= \left\{\frac{K_I T_C}{2}-K_C-2\frac{K_D}{T_C}\right\} \\ s_2 &= \frac{K_D}{T_C} \end{aligned}\right\} \quad (20)$$

and hence:

$$\left.\begin{aligned} K_D &= s_2 T_C \\ K_C &= (s_0 - s_1 - 3s_2)/2 \\ K_I &= (s_0 + s_1 + s_2)/T_C \end{aligned}\right\} \quad (21)$$

The procedure of implementing the Pole-Placement PID algorithm is therefore as follows:

1. Carry out an LS estimation to determine the *A* and *B* polynomial parameters in Equation (9).
2. Calculate the values of the *S* polynomial coefficients s_0, s_1 and s_2 from the *A* and *B* polynomial parameters and the preset *T* parameters employing Equations (20). In the case of a first order tailoring polynomial $T = 1 + t_1 z^{-1}$ the value of t_1 is fixed between 0 and –1 depending on the desired closed loop response. The response becomes more damped as the value of t_1 approaches *–1*.
3. Determine the controller gains K_C, K_D, and K_I from the values of the *S* polynomial coefficients and the control period T_C using Equations (21).
4. Calculate the control increment $\Delta u(t)$ using the velocity PID expression in Equation (19).

THE PEBBLE BED MODULAR REACTOR MODEL

The PBMR is a high temperature reactor (HTR), with a closed Brayton thermodynamic cycle and gas turbine power conversion system. The system has been designed to generate electricity without compromising the high levels of passive safety expected of advanced nuclear designs.

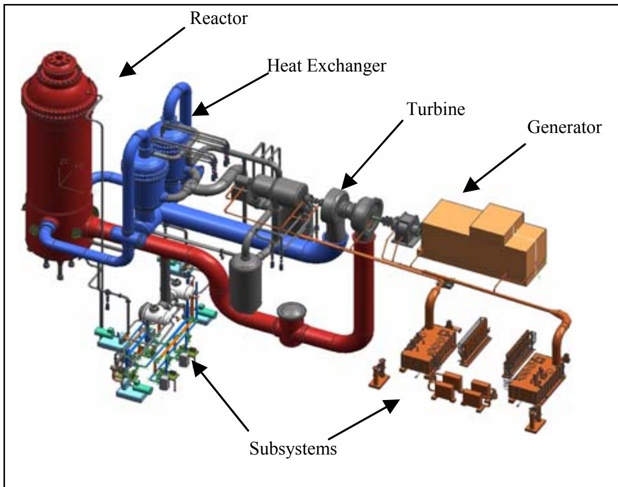

Figure 2 PBMR Main Power System Cycle

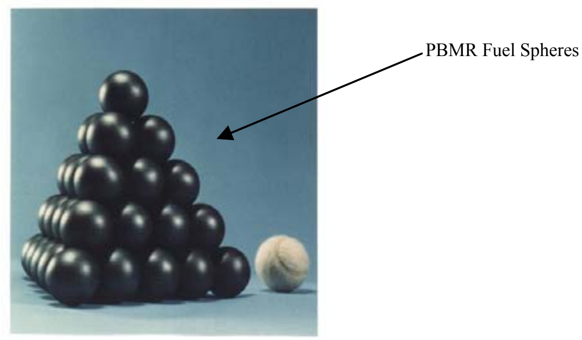

Figure 3 PBMR Fuel Spheres

Main components and processes

The PBMR [5] essentially comprises of a steel pressure vessel which holds about 450 000 fuel spheres. The fuel consists of low enriched uranium triple-coated isotropic particles contained in a molded graphite sphere. A coated particle consists of a kernel of uranium dioxide surrounded by four coating layers. The PBMR system is cooled with helium. The heat that is transferred by the helium to the power conversion system, is converted into electricity through a turbine. To remove the heat generated by the nuclear reaction, helium coolant enters the reactor vessel at a temperature of about 500 ºC (932 ºF) and a pressure of 9 MPa or 1 323 pounds per square inch (psi). The gas moves down between the hot fuel spheres, after which it leaves the bottom of the vessel having been heated to a temperature of about 900 ºC (1 652 ºF). The hot gas then enters the turbine which is mechanically connected to the generator through a speed-reduction gearbox on one side and the gas compressors on the other side. The coolant leaves the turbine at about 500 ºC (932 ºF) and 2.6 MPa (377 psi), after which it is cooled, recompressed, reheated and returned to the reactor vessel.

Reactor Temperature Control using Reactivity

The controller reactivity is amongst others a function of the control rod position. When the control rods are inserted, more neutrons will be absorbed and thus the reactivity will be reduced. The opposite will happen if the control rods are withdrawn. If the reactivity of the reactor is reduced, less heat will be generated and the reactor outlet temperature will decrease. Similarly, an increase in the reactivity will increase the temperature. In practice, the control rod position will therefore be used to control the reactor outlet temperature. Since in the model used for the simulation the reactivity can be manipulated directly, the reactivity was used to control the reactor outlet temperature.

Flownex® Thermo-Hydraulic Modelling

Flownex® is a software package developed by M-Tech Industrial used for solving thermal-fluid systems using one-dimensional CFD (computational fluid dynamics) calculations. It has a wide range of build-in components that can be used to build up a network to represent a complex circuit. Steady state as well as transients simulations (simulations over time) can be performed. The transient simulation option is useful in testing the dynamic behaviour of the circuit required for developing and testing controllers. This functionality, together with the fact that Flownex® can easily be linked with Simulink® (Matlab®) [10], was used to design and test the controller for this paper.

RESULTS

The data employed in the determination of the parameters was obtained from an open loop transient generated from a Flownex® [8] model after inducing a 0.2978 step change in the reactivity. The original open loop response of 4001 samples has been employed for model parameter estimation. The open loop response is shown in Figure 4.

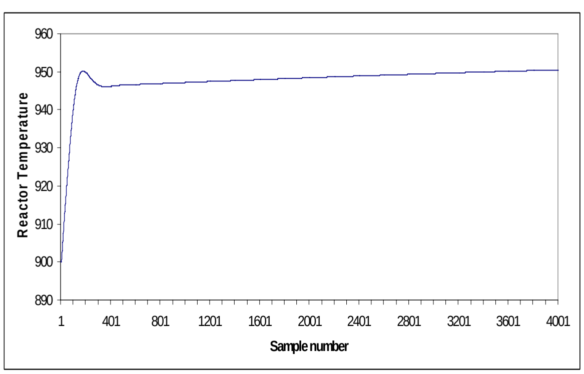

Figure 4: Reactor open-loop response: ΔT = 1s

Table 1 shows the second order model parameters and controller parameters which are obtained for a PBMR reactor outlet temperature as the process variable versus reactivity as the control input for different values of the tailoring coefficient t_1.

Table 1 Pole Placement PID Parameters

Option	t_1	K_C	K_I	K_D
1	0	0.05169	0.03481	-0.010382
2	-0.2	0.04821	0.02785	-0.010382
3	-0.4	0.04473	0.02089	-0.010382
4	-0.6	0.04125	0.01393	-0.010382
5	-0.8	0.03777	0.006963	-0.010382
6	-0.9	0.03603	0.003481	-0.010382
7	-0.95	0.03516	0.001741	-0.010382
8	-0.99	0.03446	0.0003481	-0.010382
9	-0.9999	0.03429	0.000003482	-0.010382

The closed loop responses depicted in Figure 5 show that the response becomes more damped as the tailoring coefficient approaches the unit circle at -1. The controller designer therefore has at his disposal the facility to select any desired stable response depending on the value of t_1. The parameter optimization problem is thus simplified to the pegging of a single parameter (t_1) in order to obtain the corresponding set of PID controller parameters. Figure 5 shows the effect on the controller parameters: K_C, K_I and K_D as the tailoring coefficient t_1 approaches the unit circle at -1. Whilst the tailoring coefficient has no effect on the derivative time constant and a noticeable but relatively weak lowering effect on the proportional gain, it has the effect of substantially lowering the integral component K_I (Figure 6). This effect explains the resultant damping of the closed loop responses as t_1 approaches -1.

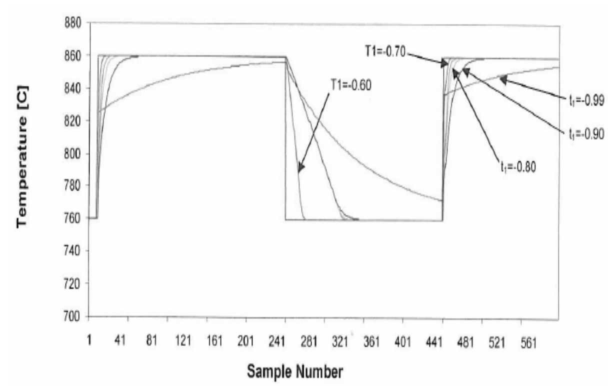

Figure 5 Closed loop responses

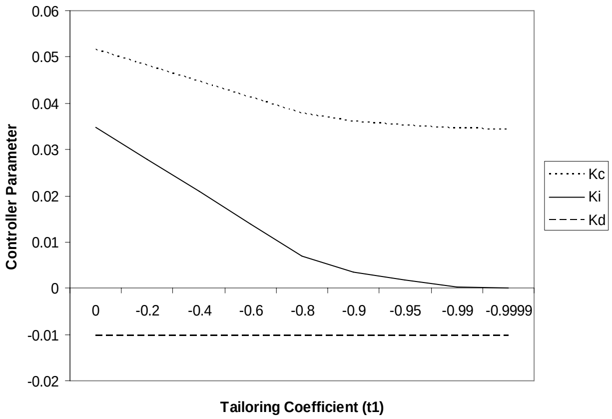

Figure 6: Effect of t_1 on PID parameters

CONCLUSIONS

This paper has presented a mathematical approach to the optimization of PID controller parameters employing the pole-placement approach in conjunction with process model parameter identification by means of a batch least squares routine. The closed loop transients were generated from a Flownex© thermal-hydraulics modelling and simulation environment. An example has been included for optimizing rector-outlet-temperature/reactivity controller parameters and a program to execute the calculation using MATLAB is also included in the Appendix.

This method is particularly important as a scientific tool that can be used by engineers to analytically set PID parameters with a high degree of accuracy and confidence. As a result the design engineer will be able to tune the PID parameters to achieve a preset closed loop response.

REFERENCES

[1] Mazana, N; The self tuning control of a tubular fixed-bed catalytic reactor, *PhD Thesis, University of Wales – Swansea, UK,* 1995.
[2] Streijc, V., Least squares parameter estimation, *Automatica, V16, pp535 - 550,* 1980.
[3] Astrom, K.J., Witternmark, B., Self-Tuning Controllers Based on Pole-Zero placement, *Proc. IEE, V127, N3, D, pp120-130*, 1980.
[4] Wellstead, P.E., Prager, D., Zanker,P., Pole-Assignment Self-Tuning Regulator, Proc. *IEE, V126, N8, pp781-790*, 1979.
[5] www.pbmr.com
[6] Jacquot, R.G., Modern Control Systems, Dekker N.Y., 1981.
[7] Mazana, N., Practical model identification for adaptive control using variants of the least squares routine, WCCE, Glasgow, 2005.
[8] www.flownex.com
[9] Flownex 6.9 User Manual – Part, M-Tech Industrial (Pty) Ltd., www.flownex.com Copyright 2005.
[10] www.mathworks.com

APPENDIX A

```
%% It is assumed that the following historical data
%% vectors have been
%% constructed in Excel and can be imported to the
%% MATLAB© workspace:
```

%% $X = \theta(t)$; $F = \Phi^T(t)$; $Y = Y(t)$

%% Set the sampling period for calculation
SamplingPeriod = 1

%% Set the tailoring polynomial coefficients for first order model
%% t_1 is set to between 0 and -1 in order to force the poles of the
%% characteristic equation (15) to the stable region within the unit
%% circle in the z-domain

t1=-0.80; %% The value of t_1 is manipulated between 0 and -1
t2=0; %% t_2 = 0 for a first order tailoring polynomial
t3=0; %% t_3 = 0 for a first order tailoring polynomial

%% Determine the model parameters for the first order ARMA
%% process model in equation (7)
%% $\theta(t) = \left[\Phi^T(t)\Phi(t)\right]^{-1}\Phi^T(t)Y(t) = P(t)\ \Phi^T(t)Y(t)$

X =inv(F'*F)*F'*Y; %% X = $\theta(t)$; F = $\Phi^T(t)$; Y = Y(t)

%% The vector $\theta(t)$ gives the following as the first order model
%% parameters (see equations (1) and (8)

A1=-X(1,1)
A2=-X(2,1)
B0=X(3,1)

%% The controller parameters are determined as follows from equations
%% (20) and (21):

S0=(T1-A1+1)/B0;
S1=(T2-A2+A1)/B0;
S2=(T3+A2)/B0;

KC = (S0-S1-3*S2)/2 %% Proportional gain
KI = (S0+S1+S2)/SamplingPeriod %% Integral constant
KD = S2*SamplingPeriod %% Derivative constant