

**COMPUTING THE AUTOPILOT CONTROL ALGORITHM USING
PREDICTIVE FUNCTIONAL CONTROL FOR UNSTABLE MODEL**

**H. A. KASDIRIN
DR J. A. ROSSITER**

(2009 International Conference of Soft Computing and Pattern Recognition)

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Computing The Autopilot Control Algorithm using Predictive Functional Control for Unstable Model

H. A. Kasdirin,
Faculty of Electrical Engineering,
Universiti Teknikal Malaysia Melaka, Malaysia
Phone: +606-555 2215, Fax: +606-555 2222,
Email: hyreil@utem.edu.my

Dr J. A. Rossiter
Department of Automatic Control & Systems
Engineering
The University of Sheffield, UK
Email: rossiter@shef.ac.uk

Abstract—This paper discusses the computing development of a control algorithm using Predictive Functional Control (PFC) for model-based that having one or more unstable poles. One basic Ballistic Missile model [10] is used as an unstable model to formulate the control law algorithm using PFC. PFC algorithm development is computationally simple as a controller and it is not very complicated as the function of a missile will explode as it reaches the target. Furthermore, the analysis and issues of the implementation relating linear discrete-time unstable process are also being discussed. Hence, designed PFC algorithm need to find the suitable tuning parameters as its play an important part of the designing the autopilot controller. Thus, the tuning of the desired time constant, Ψ and small coincidence horizon n_1 in a single coincidence point shows that the PFC control law is built better in the dynamic pole of the unstable missile mode. As a result, by using a trajectory set-point, some positive results is presented and discussed as the missile follow its reference trajectory via some simulation using MATLAB 7.0.

Keywords: *Predictive Functional Control (PFC), autopilot design, state-space models*

I. INTRODUCTION/BACKGROUND

Predictive Functional Control (PFC), which developed by Richalet [1] is one of Model Predictive Control (MPC) techniques that have been developed as a powerful algorithm for controlling process plants [6]. PFC is based on the same approach with all MPC strategies i.e., prediction of the future outputs, and calculation of the manipulated variables for an optimal control [4]. Therefore, PFC is also based on the same principles which are using an internal model, specification of a reference trajectory and determination of the control law. In this paper, the focus is on the computing implementation of the PFC algorithm on an unstable model-based automation application that is in fast response such as the missile dynamic models.

Based on the aim and objectives given above, this project is computing the control algorithm design an autopilot system of a guided missile using PFC controller. Thus, this paper will be started with the computation development theory of the PFC algorithm. The second section will be looked the problem formulation by introducing an unstable

model and the formulation of its control law. This section also will be discussing the way PFC handle the unstable process by pre-stabilise the unstable plants to implement the stabilizing linear PFC control formulation.

The last section will then jumped into the simulation and implementation result of PFC on the model-based. The section will be explained how PFC algorithm could work in given unstable model and then further the implementation of PFC whether PFC could work as a controller on fast process by given a specific set-point. The last section of this project tries to conclude the project as it developed from previous section. The summary of the project will be discussed and some recommendations will be noted for further analysis and research.

II. PREDICTIVE FUNCTIONAL CONTROL

This section will be discussing the theoretical part of control law so that it can be formulated and implemented in the following sections. However, at first, this section will describe the introduction of PFC algorithm and continued with the formulation of its control law.

A. Formation of Model-based controller

The model is the essential element of an MPC controller and hence, also for PFC controller. PFC can use many forms of model i.e.; internal model (IM), including state space, transfer function, Finite Impulse Response (FIR), fuzzy rules, and etc [5],[6]. The use of IM is important in PFC to capture the process dynamics of the system and also continue to calculate the PFC control law later on.

Hence, for this section, the model is developed in state-space form. The discussion of PFC and other MPC algorithms in state-space form has several advantages, including easy generalisation to multivariable systems, ease of analysis of closed-loop properties, and online computation [6].

Given the general state space model, of the form:

$$\begin{aligned}\underline{x}_{k+1} &= A\underline{x}_k + B\underline{u}_k \\ \underline{y}_k &= C\underline{x}_k + D\underline{u}_k\end{aligned}\quad (2.1)$$

Prediction with a strictly proper system ($D = [0]$), so

$$\begin{aligned}\underline{x}_{k+2} &= A\underline{x}_{k+1} + B\underline{u}_{k+1} \\ \underline{y}_{k+2} &= C\underline{x}_{k+2}\end{aligned}\quad (2.2)$$

From Equation 2.1 and 2.2 above, it shows the model used is a linear one that could represent by;

$$\begin{aligned}\underline{x}_k &= P_{xx}\underline{x}_k + H_x\underline{u}_{k-1} \\ \underline{y}_{-k} &= P\underline{x}_k + H\underline{u}_{k-1}\end{aligned}\quad (2.3)$$

where \underline{x}_k is the state model, \underline{u}_k is the input model, \underline{y}_k is the measured output model, P_{xx} , H_{xx} , P and H are respectively, matrices or vectors of the right dimension by using the state-space approach.

The advantages of developing the PFC algorithm are that by its intuitive approach. However, before deriving the PFC control law, there are some criteria needed to look at which are the formulation its reference trajectory, the coincidence points (if occurs) as well as its future control trajectory. Hence, assuming at a single coincidence point and $\underline{u}_{k-i} = \underline{u}_k$, the PFC control law can be computed by rewriting the Equation 2.3 and obtained;

$$\begin{aligned}\underline{u}_k &= -H^T [P\underline{x}_k + (r_k - (r_k - y_k)\Psi^i)] \\ \underline{u}_k &= -K\underline{x}_k + \beta r_k\end{aligned}\quad (2.4)$$

where; $K = -H^T (P - \Psi^i y_k)$
 $\beta = -H^T (1 - \Psi^i)$

Thus, it can easily express as a fixed linear feedback law in the form of prediction algorithm. Hence, the conventional a posterior stability and sensitivity analysis can be applied in straightforward manner.

B. Unstable Model Using PFC Algorithm

PFC algorithm is defined in the previous subsection is basically open-loop process control applications. In the contrary, in industry applications, open-loop unstable processes do also occur. Yet, these systems are difficult to control. Therefore, systematic control design tools are needed to handle complex instability without a high on-line computational load. ADERSA have successfully applied PFC on many unstable systems [3]. This section will discuss the theoretical tools to pre-stabilise the unstable plants to implement the stabilizing linear PFC control formulation.

PFC, as well as other MPC algorithms are weak with both non-minimum phase problems and some unstable process [3], or called prediction mismatch. If the process open-loop is unstable, PFC is ill-posed because prediction cannot match desired behaviour of the process, i.e. diverging. Therefore, divergence

open-loop prediction is the main cause of the prediction mismatch. Hence, there is a must to stabilise the prediction. There are two ways of pre-stabilising the predictions which are inserting a stabilising loop and another by shaping the future inputs, algebraically so that the outputs are stable. However, this section only focused on solving algebraically the unstable process as discussed by [7] and [8].

C. Predictive Stabilisation

Removal of the prediction mismatch is essential for PFC to work. Hence, the model needs to have prediction stabilisation. One method is by cancelling the unstable modes and starts working with stable predictions [8]. This means that the PFC control law must be modified. Therefore, in solving this problems PFC will lead to good closed-loop performance if the predictions used are a good match to the consequent closed-loop behaviour.

The illustration below shows the state space method of predictions to cancel the unstable modes [3]. From Equation 2.1, let the state-space matrix have some unstable eigenvalues. Decompose the system into stable and unstable modes using eigenvalue decomposition;

$$A = [W_s, W_u] \text{diag} [A_s, A_u] \begin{bmatrix} \overline{V_s^T} \\ \underline{V_u^T} \end{bmatrix}\quad (2.5)$$

where; subscript s is used for stable and u for unstable.

The control law becomes,

$$\begin{aligned}\underline{u}_{k-i} &= - [W_s A_s^i V_s^T]^{-1} V_u^T A^i \underline{x}_k + H\beta \\ \underline{u}_k &= -K\underline{x}_k + H\beta\end{aligned}\quad (2.6)$$

where $K = [W_s A_s^i V_s^T]^{-1} V_u^T A^i$ and β could be choose freely.

Consequently, the main concepts of PFC and its algorithm have been discussed. This section is really important before introducing an example of fast system which is missile models and the implementation of the algorithm elaborated in this section.

III. PROBLEM FORMULATION OF THE AUTOMATION FAST-RESPONSE MODEL

The choice of prediction model is essential for PFC to work. The model-based of a missile design is selected as an example of automation fast-response application model. Therefore, the research will be using based on ballistic type missile. This missile is guided during powered flight by deflecting the thrust

vector and become a free-falling body after engine cut-off [10].

The main feature of the missile as shown in Figure 1 is that it is roll-stabilised; thus there is no coupling between longitudinal and lateral mode which simplifies the analysis. Another feature is that its trajectory is planned to maintain the missile at a zero angle of attack. Based on the assumption made above, the Ballistic Missile Model can be represented by its dynamic equation of transfer function equation [10]. Furthermore, noted that the deflection of the thrust chamber, δ is being controlled by the change of θ (s) along its missile body. The component of thrust normal to the X axis is proportional to the sine δ . However if δ is small the sine δ can be replaced by δ in radians.

The values of the rest of the quantities in equation above are given for the time of maximum dynamic pressure and follow the characteristics tabulated in the Table II in Appendix 1 below. Then, the model need to be in the form of discrete-time model representation as PFC only works with discrete model. By using MATLAB 7.0, the change of model from continuous model to discrete model is trivial; both prediction models are in state-space discrete-time model.

A. Development of PFC Algorithm for The Fast-Response Unstable Model

From Equation 2.6, the improved unstable stable system is in linear discrete state-space model form. Therefore, based on the Equation 2.1, 2.2 and adding the PFC control law (Equation 2.4), it could be defined the control law of the system by;

$$\underline{u}_k = -K_{new} \underline{x}_k + \beta_{new} r_k \quad (2.7)$$

where; $K_{new} = -(CAB + CB)^{-1} (CA^2 - C\Psi^2)$
 $\beta_{new} = -(CAB + CB)^{-1} (1 - \Psi^2)$

The control algorithm is set and simulated with set-point trajectory, r_k . However, the PFC control algorithm need to be tune to get the best prediction result.

B. Tuning Parameters of PFC

Later, tuning of the parameters need to collaborate with the control law. The tuning parameters of PFC are generally the coincidence horizon, e.g. $n_1 = 1, 2, \dots$ and the desired time constant, Ψ . The typical procedure with one coincidence point [3] would be as follows:

- Choose the desired Ψ .
- Do a search for $n_1 = 1, \dots$ large and find the associated control law for each n_1 .
- Select the n_1 which gives closed-loop dynamics closest to the chosen Ψ .

- Simulate the proposed law. Otherwise, reselect Ψ and go to step 2.

Hence, the tuning reduces to a global search, but this requires only relatively trivial computations and hence would be quite quick. With two coincidence points, the global search would be more involved but should still be quick.

IV. SIMULATION RESULT

A. The Implementation of PFC for unstable process

Unstable process can be difficult to control. Yet, this section will see whether PFC can be implemented for unstable process. The Ballistic Model [10] will be used as an example of unstable process as it has one unstable pole in its transfer function. For this reason, the same missile model missile will be used and therefore has one unstable pole. Therefore, open-loop response by using root locus and bode diagram analysis have been investigated regarding the model to ensure it is unstable loop.

B. Solving the Prediction Mismatch by Pre-stabilised Prediction

In order to do the pre-stabilised method, it needs to eliminate the unstable pole of the unstable system. The method to eliminate the unstable poles has been discussed in earlier. Therefore, by using the missile model, in order to pre-stabilise the prediction mismatch of the model, modification of PFC algorithm is needed. Before implementing the modified PFC control law, the choice of tuning parameters also could be figured by the relation desired time constant, Ψ with the algorithm formed from Equation 2.7.

Hence, the dynamic response of the model is based on pre-stabilised pole of the model from Equation 2.6 and 2.7. Based on some simulation outputs, the tuning parameters of the PFC controller is performed by setting the dynamic pole of the model or the desired time constant, $\Psi = 0.98$, with comparing some variation of $n_1 = 3, 6$ and 10 (large), the illustrations of the controller outputs are as shown in Figure 2 and 3. Hence, the result shows that the method is successfully performed with the unstable system. The performance and response of the controller and output response is shown good result. In addition, as the coincidence horizon of n_1 become larger, the figures show that the response is fast and quick to the stable condition.

C. Implementation PFC as Missile Autopilot Control

Based on the result of the implementation of PFC algorithm above, the real implementation to use for

the missile scenario should give the same result. The result above shows that the best of PFC algorithm can be configured if the appropriate tuning parameters used.

The missile moving scenario is set to be its reference trajectory of the missile mission. For this implementation, the scenario is developed by turning of missile up to 3 radians per second and turning it back at 2 radians per second before coming back to its original location. Using the PFC control law developed from Equation 2.7, the tuning parameters of PFC control law are stated in Table I below;

TABLE I: PARAMETERS USED FOR DEVELOPED OF PFC CONTROL LAW FOR BALLISTIC MODEL MISSILE

Parameter	Sampling time, T	Desired Time Constant, ψ	Coincidence horizon, n	Coincidence point (n_1, n_2, n_3, \dots)
	0.01	0.98	6	1 (n_1)

From the result obtained in Figure 3, it shows that the output response follows the reference trajectory as the control output gives stable condition and found no ill-posed result. By observing the deflection of thrust chamber, the thrust chamber deflected shows some overshoot before quickly perform into the stable condition as the reference trajectory change its course. This condition is true for the chamber as it perform very fast turns to follow according to the reference trajectory. Overall result shows that the response is fast and quick to the stable condition. Therefore, removal the prediction mismatch is essential to improve the performance of PFC.

CONCLUSION & RECOMMENDATIONS

In conclusion, this paper has implemented a controller for an autopilot missile using Predictive Functional Control (PFC) method in state-space form, then coded to simulate using MATLAB 7.0 environment. In all, based on the achievement/result the implementation of PFC algorithm seems intuitive and computationally simple. This truly important as the missile controller need not to be very complicated as it will explode as it reaches the target.

On the other hand, this section has also briefly shows that one can modify PFC so that it can handle unstable system more reliably. It is a very good motivation as any developments to extent would be useful. Also, as PFC is so simple to implement for fast system such as autopilot missile, this would be an opportunity for industries to exploit PFC.

This report tried to cover the main intent as more as possible. However this report does not mention how the PFC algorithm operates in nonlinear cases. Because of the limitation of the time this it was not achievable. With clearly illustrated the computation PFC model in Section III and its

implementation on Section IV, the idea of PFC methodology work on a fast system such as autopilot missile is explained.

Sometimes it is better to use a fast sampling rate (fast update of the receding horizon) with some prediction mismatch than to use slower sampling rate and less prediction mismatch. PFC allows the former, as it allows fast sampling rates. Moreover, due to the algorithm simplicity it is more straightforward to adapt for nonlinear models. Nonetheless, despite its apparent simplicity, PFC often gives very good performance, with constraint handling, quite similar to that achievable with a far more complex MPC algorithm.

ACKNOWLEDGMENT

Thanks are extended to all the colleague and staff of Automatic Control & System Engineering Department in the University of Sheffield, UK for their support in providing motivation and idea used for this project. Not forget from Universiti Teknikal Malaysia Melaka (UTeM) for their strong commitment towards academic excellence that encourages the researcher to go forward.

REFERENCE

- [1] J. Richalet, "Pratique de la Commande Predictive." Editions Hermes, Paris, 1993.
- [2] Clarke, D. W., Mothadi, C., & Tuffs, P. S., "Generalized Predictive Control. Part I: the basic algorithm. Part II: Extensions and interpretations." *Automatica*, 23(2), pp.137-160, 1987.
- [3] J A Rossiter, "Model-Based Predictive Control: A Practical Approach", CRC Press, 2003.
- [4] E F Camacho and C Bordons, "Model Predictive Control", Springer, 2004.
- [5] J.M Maciejowski, 2002. "Predictive Control With Constrains" Prentice Hall, 2002.
- [6] J. B. Rawlings, "Tutorial Overview of Model Predictive Control. Special Edition: Industrial Process Control." *IEEE Control Systems Magazine*. pp 38-52, 2000.
- [7] J. B. Rawlings, and K. R. Muske, "Stability of Constrained Receding Horizon Control". *IEEE Transactions on Automatic Control*. AC-38(10), 1512-1516, 1993.
- [8] Muske, K.R. and J.B. Rawlings, "Linear Model Predictive Control of Unstable Processes." *J.Proc. Cont.*, 3(2), pp. 85-96, 1993.
- [9] B. Kouvaritakis, J.A. Rossiter and A.O.T Chang, "Stable Generalised Predictive Control: An Algorithm with Guaranteed Stability", *Proceedings IEE, Pt. D*, 139(4), pp.349-262, 1992.
- [10] J H Blakelock, "Automatic Control of Aircraft and Missiles", Wiley, 1991.

APPENDIX 1

TABLE II: CHARACTERISTICS AND COEFFICIENT OF BASIC BALLISTIC MISSILE MODEL

Altitude	=	36 000 ft
Velocity	=	1285 ft/sec
Mass	=	445 slugs
$C_{m\delta}$	=	$-Tl/(Sq)$
	=	-34.25
$C_{z\delta}$	=	$-T/(Sq)$
	=	- 4.63
$C_{z\alpha}$	=	- 3.13
$C_{m\alpha}$	=	11.27

I	=	27 ft ²
Θ_0	=	68.5°
d	=	3.75 ft
q	=	585 lb/sqft
S	=	11.04 sq ft
$d/(2U) C_{mq}$	=	- 0.321 sec
$mU/(Sq)$	=	88.5 sec
$mg / (Sq)$	=	- C_w
	=	2.22
I_y	=	115 000 slug ft ²
$I_y / (Sq)$	=	4.75 sec ²

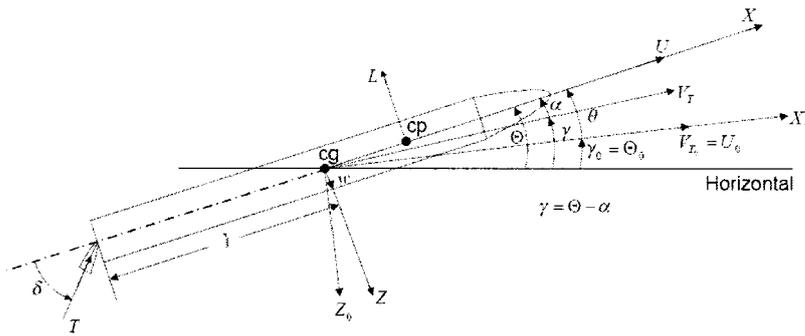


Figure 1: Ballistic missile axis system. [10]

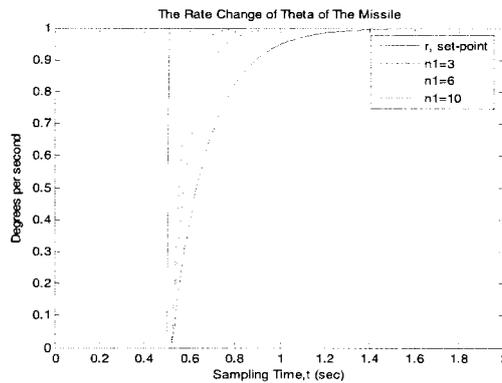
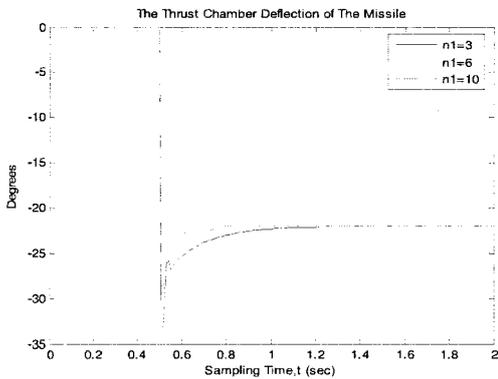


Figure 2: The Controller Response of the Computing Algorithm for $\Psi = 0.94$

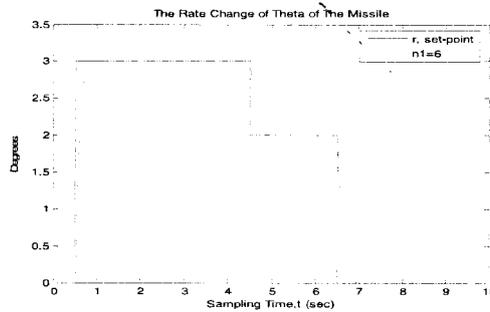
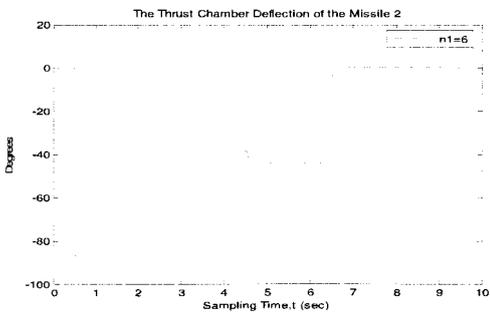


Figure 3: The Missile Controller Response of the Ballistic Missile