A NEW APPROACH TO NON-LINEAR GENERALISED PREDICTIVE CONTROL AND DESIGN OF MATLAB/SIMULINK INTERFACE PROGRAM

TANER ARSAN

Istanbul Technical University
Electrical and Electronics Faculty
Control Systems Division
80626 Maslak-Istanbul
Turkey

E-mail: arsan@elk.itu.edu.tr

ATİLLA BİR

Istanbul Technical University
Electrical and Electronics Faculty
Control Systems Division
80626 Maslak-Istanbul
Turkey

E-mail: abir@elk.itu.edu.tr

Abstract: In this study, it is aimed to design a Matlab/Simulink interface program for Generalised Predictive Control (GPC). In addition to that, one possible extension of Generalised Minimum Variance (GMV) control as an alternative prediction-free version of Non-Linear Generalised Predictive Control (NGPC) is given. It is also shown that this new approach copes with nonlinear systems with unstable dynamics.

Keywords: Generalised Minimum Variance Control, Generalised Predictive Control, Non-Linear Generalised Predictive Control, Matlab/Simulink Interface Program.

INTRODUCTION

ieneralised Minimum Variance (GMV) control lgorithm [1] and Generalised Predictive Control (GPC) 2-12] has been recast in state-space form by Gawthrop, Demircioğlu and Siller-Alcala [13,14]. Unlike the transfer function approach, the state-space approach extends readily to the nonlinear case.

GPC is a natural extension of GMV and there are two ways to extend GMV to GPC. First method is to increase the time horizon so that maximum prediction horizon is greater than zero as in Gawthrop's GPC [13, 14]. Second method is to increase the degree of performance polynomial [15-16]. This method can be called as prediction-free version of Non-Linear GPC (NGPC).

In this study, using geometric interpretation of GMV, an alternative (prediction-free) version of NGPC has been explained and a twolink-manipulator with delayed measurement is used as an simulation example.

In this study, other important point is to design a Matlab/Simulink interface program for GPC, NGPC and prediction-free version of NGPC. State-space GPC algorithm has also been realised as an available Matlab tool-box by Gawthrop, Demircioğlu and Alcala [13, 14]. A new interface program has been designed for Matlab/Simulink using Matlab tool-box. GPC Algorithm has become easy to use and user friendly by means of proposed interface program. It is also possible to compare the efficiency of NGPC and prediction-free version of NGPC with other algorithms proposed by several researchers.

II. GMV ALGORITHM

This section considers nonlinear dynamic systems with the state-space representation:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{F}(\mathbf{x},\mathbf{u}) \\ \mathbf{y} &= \mathbf{H}(\mathbf{x}) \end{aligned} \tag{1}$$

where the function **F** and **H** are precise N_y times differentiable with respect to each argument and the system inputs **u**, system outputs **y** and system states **x**, dimensions are n_u, n_y and n_x respectively.

A special case of (1) is one where the control enters in a linear fashion:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}) \end{aligned} \tag{2}$$

where $\mathbf{f}(\mathbf{x})+\mathbf{g}(\mathbf{x})\mathbf{u}=\mathbf{F}(\mathbf{x},\mathbf{u})$ and $\mathbf{h}(\mathbf{x})=\mathbf{H}(\mathbf{x})$.

A column vector $\mathbf{y}_{N_y}(t)$ of output derivatives is defined as

$$\mathbf{y}_{N_y}(t) = [\mathbf{y} \quad \mathbf{y}^{[1]} \quad \mathbf{y}^{[2]} \quad . \quad . \quad . \quad \mathbf{y}^{[N_y]}]^T \tag{3}$$

where $^{[i]}$ indicates the i^{th} derivative with respect to time. $\mathbf{u}_{N_u}(t)$ is a column vector (n_u (N_u+1) × 1) of input derivatives:

$$\mathbf{u}_{N_p}(t) = [\mathbf{u} \quad \mathbf{u}^{[1]} \quad \mathbf{u}^{[2]} \quad . \quad . \quad . \quad \mathbf{u}^{[N_y]}]^T \tag{4}$$

N_y times repeated differentiation of the output y with respect to time, together with repeated substitution of the system equation (1) gives a nonlinear equation relating the output derivative vector $\mathbf{y}_{N_y}(t)$ to the state **x** and the input derivative vector $\mathbf{u}_{N_u}(t)$:

$$\mathbf{y}_{N_y}(t) = \mathbf{O}_{N_y}(\mathbf{x}(t), \mathbf{u}_{N_y}(t)) \tag{5}$$

However, we first consider the system where the state is not measurable and thus Equation (5) cannot be implemented. There is also no general theory of state estimation for non-linear systems. For this reason, a standard state observer [17] is used, given by:

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}) + \mathbf{L}(\hat{\mathbf{x}})\mathbf{e}(t) \tag{6}$$

$$\hat{\mathbf{y}}(t) = \mathbf{g}(\hat{\mathbf{x}}) \tag{7}$$

$$\mathbf{e}(t) = \mathbf{y}(t) - \hat{\mathbf{y}}(t) \tag{8}$$

where **L** is the ($n_x \times n_y$) observer gain matrix which is chosen to give observer poles (eigenvalues of **A-LC**) at appropriate places [13, 14]. Then we obtain the emulated value of $\mathbf{y}_{N_y}(t)$ which is denoted by $\hat{\mathbf{y}}_{N_y}(t)$ and defined as:

$$\hat{\mathbf{y}}_{N_y}(t) = \mathbf{O}_{N_y}(\hat{\mathbf{x}}(t), \mathbf{u}_{N_y}(t)) \tag{9}$$

Sometimes Equation (5) is affine in **u**(t) and can be rewritten as:

$$\hat{\mathbf{y}}(t) = \mathbf{O}_x(\hat{\mathbf{x}}(t)) + \mathbf{O}_u(\hat{\mathbf{x}}(t))\mathbf{u}(t) \tag{10}$$

The performance matrix polynomial **P**(s) defines a desired output $\mathbf{y}_d$ in terms of the setpoint **w**(t) model of the form,

$$\mathbf{P}(s)\ \mathbf{y}_d(t) \triangleq \mathbf{w}(t) \tag{11}$$

where

$$\mathbf{P}(s) = \sum_{k=0}^{n_p} \mathbf{p}_k s^k \tag{12}$$

where $\mathbf{p}_k$ is the k^th^ $n_y\times n_y$ matrix coefficient. It is convenient to define the matrix $\mathbf{\Pi}$ as,

$$\mathbf{\Pi} \triangleq [\mathbf{p}_0 \quad \mathbf{p}_1 \quad \mathbf{p}_2 \quad . \quad . \quad . \quad \mathbf{p}_k] \tag{13}$$

In ideal case, we would wish to make **y** = $\mathbf{y}_d$ and Equation (11) then becomes:

$$\mathbf{P}(s)\hat{\mathbf{y}}_d(t) \triangleq \mathbf{\Pi}\,\hat{\mathbf{y}}(t) = \mathbf{\Pi}\,\mathbf{O}_x(\hat{\mathbf{x}}(t), \mathbf{u}(t)) = \mathbf{w}(t) \tag{14}$$

where $\hat{\phi}(t)$ is an auxilary vector described as $\hat{\phi}(t) = \mathbf{\Pi}\,\hat{\mathbf{y}}(t)$ and the aim is to be $\hat{\phi}(t) = \mathbf{w}(t)$. In the special case of (10), this control law can be written as:

$$\mathbf{\Pi}\,\hat{\mathbf{y}}(t) = \mathbf{\Pi}\,\mathbf{O}_x(\hat{\mathbf{x}}(t)) + \mathbf{\Pi}\,\mathbf{O}_u(\hat{\mathbf{x}}(t))\mathbf{u}(t) = \mathbf{w}(t) \tag{15}$$

or

$$\mathbf{u}(t) = [\,\mathbf{\Pi}\,\mathbf{O}_u(\hat{\mathbf{x}}(t))]^{-1}[\mathbf{w}(t) - \mathbf{\Pi}\,\mathbf{O}_x(\hat{\mathbf{x}}(t))] \tag{16}$$

as GMV control signal [13-16].

III. EXTENDING GMV TO NGPC

GPC is a natural extension of GMV which overcomes two problems. First problem is that the system relative degree must be precisely known, second is that the system must have stable zero dynamics [13, 14].

Given the special-case conditions, there are two ways to extend GMV to GPC [15, 16]:

1. Increase the time horizon so that $\tau_2 > 0$ or
2. Increase the degree of P (deg(P)> ρ).

The first is an obvious approach and it is proposed by Gawthrop and Demircioğlu [8-14]. The latter choice is also proposed by Gawthrop and Arsan in a technical report [15]. In this section, first approach has been explained. There are four concepts associated with the NGPC considered here [13, 14]:

1. Prediction via Taylor series,
2. Moving-horizon control,
3. Control constraints,
4. Optimisation.

In non-linear case [13, 14], prediction is accomplished via a Taylor series expansion of the system output y, here this concept is extended slightly to use an corresponding expansion of auxilary vector $\hat{\phi}(t) = \mathbf{\Pi}\,\hat{\mathbf{y}}(t)$. It is useful to define a predicted $\hat{\phi}(\tau, t)$:

$$\hat{\phi}(\tau,t) \triangleq \mathbf{T}(\tau)\,\mathbf{\Pi}\,\hat{\mathbf{y}}(t) \tag{17}$$

where $\mathbf{T}(\tau)$ is a row vector with ($n_\phi\times n_\phi$) matrix elements and given by

$$\mathbf{T}(\tau) = \begin{pmatrix} \mathbf{I}_{n_\phi\times n_\phi} & \theta & \ldots & \dfrac{\theta^i}{i!} & \ldots & \dfrac{\theta^{N_\phi}}{N_\phi!} \end{pmatrix} \tag{18}$$

where θ is the ($n_\phi\times n_\phi$) diagonal matrix with $\theta_{ii} = \tau_i$ and $\mathbf{I}_{n_\phi\times n_\phi}$ is the ($n_\phi\times n_\phi$) unit matrix. Using (9), (17) can be rewritten as:

$$\hat{\phi}(\tau,t) \triangleq \mathbf{T}(\tau)\,\mathbf{\Pi}\,\mathbf{O}(\hat{\mathbf{x}}(t), \mathbf{u}^*(t,0)) \tag{19}$$

The predicted reference value is also described as:

$$\mathbf{w}^*(\tau) \triangleq \mathbf{T}(\tau)\mathbf{\Pi}\,\hat{\mathbf{y}}(t) = \mathbf{T}(\tau)\,\mathbf{w}(t) \tag{20}$$

The aim to minimize the error signal between predicted value and predicted set point value given by

$$\tilde{\mathbf{e}}^*(\tau) = \phi^*(\tau,t) - \mathbf{w}^*(t) \tag{21}$$

The NGPC cost averages the error over the time intervals specified by τ_1 and τ_2 [13, 14]. To allow for the different time intervals corresponding to each ϕ in the context of a single integral, τ is parameterised by the scalar $\lambda \in [0\ 1]$:

$$\tau \triangleq \tau_1 + \lambda(\tau_2 - \tau_1) \tag{22}$$

NGPC cost function is given by

$$
\begin{aligned}
&J_{GPC}(\mathbf{u}^{*}_{N_u}(t,\tau)) = \\
&= \int_0^1 [\phi^{*}(\tau,t) - \mathbf{w}^{*}(t)]^T [\phi^{*}(\tau,t) - \mathbf{w}^{*}(t)] d\lambda \\
&= \int_0^1 [\Pi \mathbf{O}_{N_y} \hat{\mathbf{x}}(t) + \Pi \mathbf{H}_{N_y,N_u} \mathbf{u}^{*}_{N_u}(t,0) - \omega(t)]^T \mathbf{T}^T(\tau). \qquad (23) \\
&\qquad \mathbf{T}(\tau)[\Pi \mathbf{O}_{N_y} \hat{\mathbf{x}}(t) + \Pi \mathbf{H}_{N_y,N_u} \mathbf{u}^{*}_{N_u}(t,0) - \omega(t)] d\lambda \\
&= [\Pi \mathbf{O}_{N_y} \hat{\mathbf{x}}(t) + \Pi \mathbf{H}_{N_y,N_u} \mathbf{u}^{*}_{N_u}(t,0) - \omega(t)]^T \overline{\mathbf{T}}(\tau_1,\tau_2). \\
&\qquad [\Pi \mathbf{O}_{N_y} \hat{\mathbf{x}}(t) + \Pi \mathbf{H}_{N_y,N_u} \mathbf{u}^{*}_{N_u}(t,0) - \omega(t)]
\end{aligned}
$$

where $\overline{\mathbf{T}}(\tau_1,\tau_2) \triangleq \int_0^t \mathbf{T}^T(\tau)\mathbf{T}(\tau)d\lambda$. Using (18) the ij^th^ element of $\mathbf{T}^T(\tau)\mathbf{T}(\tau)$ is $\frac{\tau^{i-1}\tau^{j-1}}{(i-1)!(j-1)!}$. It follows that the ij^th^ elements of $\overline{\mathbf{T}}(\tau_1,\tau_2)$ is:

$$\overline{\mathbf{T}}_{ij}(\tau_1,\tau_2) = \frac{\tau_2^{i+j-1} - \tau_1^{i+j-1}}{(i-1)!(j-1)!(i+j-1)(\tau_2-\tau_1)} \qquad (24)$$

It is also very important to notify that this is non-dynamic optimisation problem and the equation must be solved numerically for $\mathbf{u}^{*}_{N_u}(t,0)$ at each time t [13,14].

IV.GEOMETRIC INTERPOLATION OF GMV AND A NEW APPROACH TO NGPC

The control signal of GMV control can be given as in Equation (16). Considering the system of Equation (2), differentiating the output y with respect to time gives:

$$
\begin{aligned}
y^{[1]}(t) &= \frac{\partial h}{\partial \mathbf{x}} \dot{\mathbf{x}}(t) = \frac{\partial h}{\partial \mathbf{x}} f(\mathbf{x}(t)) + \frac{\partial h}{\partial \mathbf{x}} g(\mathbf{x}(t))u(t) \\
&= o_1(\mathbf{x}(t)) + h_1(\mathbf{x}(t))u(t)
\end{aligned} \qquad (25)
$$

In this equation if $h_1(\mathbf{x}(t)) \triangleq \frac{\partial h}{\partial \mathbf{x}} g(\mathbf{x}(t)) \neq 0$, the procedure terminates; otherwise the second derivative of the output y with respect to time is written as:

$$
\begin{aligned}
y^{[2]}(t) &= \frac{\partial o_1}{\partial \mathbf{x}} \dot{\mathbf{x}}(t) = \frac{\partial o_1}{\partial \mathbf{x}} f(\mathbf{x}(t)) + \frac{\partial o_1}{\partial \mathbf{x}} g(\mathbf{x}(t))u(t) \\
&= o_2(\mathbf{x}(t)) + h_2(\mathbf{x}(t))u(t)
\end{aligned} \qquad (26)
$$

This procedure is repeated until $h_i \neq 0$ $(i=1,2,...,\rho)$. This is the definition of relative order ρ in the nonlinear context. At this point Equation (9) can be rewritten (with $N_y=\rho$) as:

$$\mathbf{y}_\rho(t) = \mathbf{O}_\rho(\mathbf{x}(t)) + \mathbf{H}_{\rho\rho}(\mathbf{x}(t))\mathbf{u}_\rho \qquad (27)$$

where

$$\mathbf{O}_\rho(\mathbf{x}) \triangleq \begin{bmatrix} o_1(\mathbf{x}) \\ \dots \\ o_\rho(\mathbf{x}) \end{bmatrix} , \quad \mathbf{H}_{\rho\rho}(\mathbf{x}) \triangleq \begin{bmatrix} 0 \\ 0 \\ \dots \\ h_\rho(\mathbf{x}) \end{bmatrix} \qquad (28)$$

In Equation (28), $\mathbf{O}_\rho(\mathbf{x})$ contains the elements which depend on states of the system and $\mathbf{H}_{\rho,\rho}(\mathbf{x})$ contains the elements which depend on control signal.

At this point, second special-case condition in extending GMV to NGPC is considered. This case means an increase in the degree of P (deg(P)> ρ) [15, 16].

The procedure given by Equations (25) and (26) is repeated until $h_i \neq 0$ $(i=1,2,...,\rho)$ in GMV, but this procedure repeated until $h_i \neq 0$ (i=1,2,...,deg(P)) in the alternative version of NGPC. It is also very important to notify that (deg(P)> ρ). Choosing performance polynomial degree greater than relative order of the system means increasing the control constraint as:

$$u = \sum_{i=0}^{N_u} u_i t^i \qquad (29)$$

Control signal u is constrained to be a polynomial function of time with degree N_u such as:

$N_u = 0$: u is constant,

$N_u = 1$: u is constant + ramp,

$N_u = 2$: u is constant + ramp + quadratic.

This approach can be called prediction-free version of NGPC. In this work, twolink-manipulator which has a non-linear characteristics is considered as a plant to be controlled.

V. SIMULATION EXAMPLE AND INTERFACE PROGRAM FOR NGPC

This example provides an illustration of nonlinear Generalised Predictive Control (NGPC) operating on a twolink-manipulator with a modified sensor. The sensor is modified in a rather contrived way to give a nonlinear system with an unstable inverse. The idea is that the sensor approximates a pure delay of τ as $e^{-s\tau} \approx 1 - s\tau$. The states of the system are:

$$
\begin{aligned}
\dot{x}_1 = &(-(2(\cos(x_2 - x_4)j_r^2 u_2 + \sin(2(x_2 - x_4))l_r^2 m_r x_1^2)l_r^2 m_r - \\
&-(j_r + l_r^2 m_r)(u_1 - u_2)j_r^2 + 2(j_r + l_r^2 m_r)\sin(x_2 - x_4)l_r^2 m_r x_3^2)) \\
&/((j_r^2 + 6j_r l_r^2 m_r + 3l_r^4 m_r^2 - 2\cos(2(x_2 - x_4))l_r^4 m_r^2)j_r)
\end{aligned}
$$

$$\dot{x}_2 = \frac{x_1}{j_r}$$

$$
\begin{aligned}
\dot{x}_3 = &((j_r + 5l_r^2 m_r)j_r^2 u_2 + 2\sin(2(x_2 - x_4))l_r^2 m_r^2 x_3^2 - 2(u_1 - u_2) \\
&\cos(x_2 - x_4)j_r^2 l_r^2 m_r + 2(j_r + 5l_r^2 m_r)\sin(x_2 - x_4)l_r^2 m_r x_1^2) \\
&/((j_r^2 + 6j_r l_r^2 m_r + 3l_r^4 m_r^2 - 2\cos(2(x_2) - x_4))l_r^4 m_r^2)j_r
\end{aligned}
$$

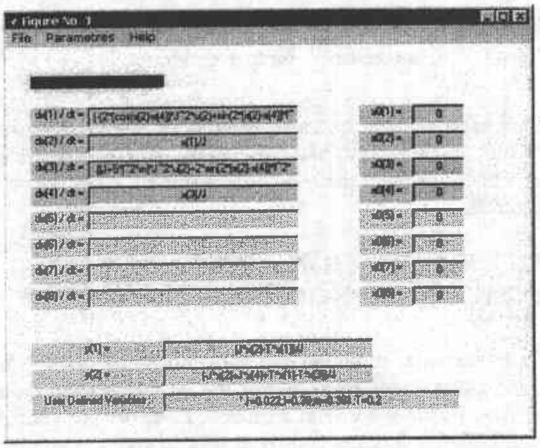
$$\dot{x}_4 = \frac{x_3}{j_r}$$

and system outputs are:

$$y_1 = \frac{j_r x_2 - x_1 \tau}{j_r} \qquad y_2 = \frac{-j_r x_2 + -j_r x_4 + x_1 \tau - x_3 \tau}{j_r}$$

where x_1, x_2, x_3, x_4 are states of the system, m_r, l_r and j_r are the mass, length and inertia of the uniform rod forming each identical link. y_1 and y_2 are outputs of the twolink-manipulator. Initial values for twolink-manipulator are zero and the set point values are $[\pi/3, -\pi/3]$. Performance polynomial is chosen as the transfer function of the system is $\frac{\bar{y}}{w} = \frac{1}{0{,}5s^2 + s + 1}$.

When "Parameters" pull-down menu is selected, there are four selection such as: System, Controller, Observer and Simulation. State space equations, outputs of the system and initial values of states can be defined in "System Equation" window as shown in Figure 1 using Matlab/Simulink interface program. Then controller type, control order and prediction order need to be defined in "Controller Selection" window of interface program as N_u=2 and the prediction horizon is 0.2-0.3seconds for NGPC as shown in Figure 2.

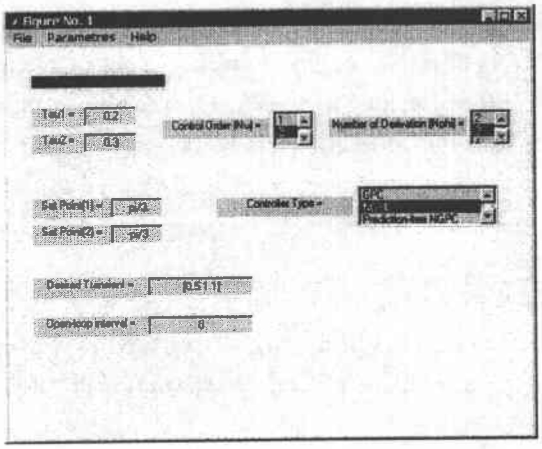


dx(1) / dt = [illegible]
dx(2) / dt = x(1)/J
dx(3) / dt = [illegible]
dx(4) / dt = x(3)/J
dx(5) / dt =
dx(6) / dt =
dx(7) / dt =
dx(8) / dt =

x0(1) = 0
x0(2) = 0
x0(3) = 0
x0(4) = 0
x0(5) = 0
x0(6) = 0
x0(7) = 0
x0(8) = 0

y(1) = [illegible]
y(2) = [illegible]
User Defined Variables = [illegible] T=0.2

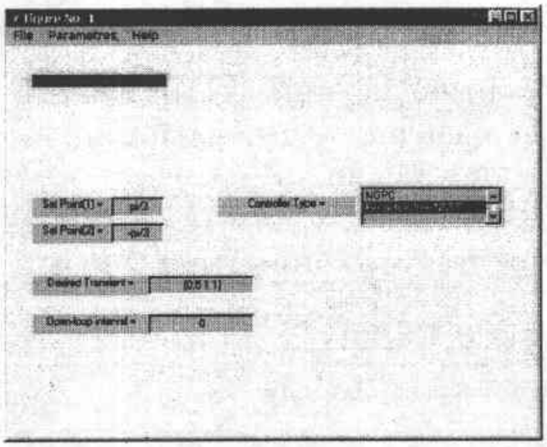
Figure 1 - "System Equation" window of Matlab/Simulink interface program for GPC tool-box.



Tau1 = 0.2
Tau2 = 0.3
Control Order (Nu) = 1
Number of Derivation (Nohi) = 2
Set Point(1) = pi/3
Set Point(2) = -pi/3
Controller Type = GPC / NGPC / Prediction-free NGPC
Desired Transient = [0.5 1 1]
Open-loop interval = 0

Figure 2 - "Controller Selection" window for Matlab/Simulink interface program for NGPC.

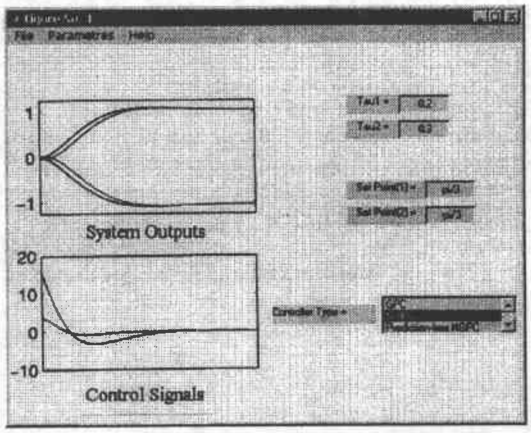
When prediction-free NGPC controller is selected from the pull-down menu, prediction horizon and controller order information are automatically removed from the window as shown in Figure 3.



Set Point(1) = pi/3
Set Point(2) = -pi/3
Controller Type = NGPC
Desired Transient = [0.5 1 1]
Open-loop interval = 0

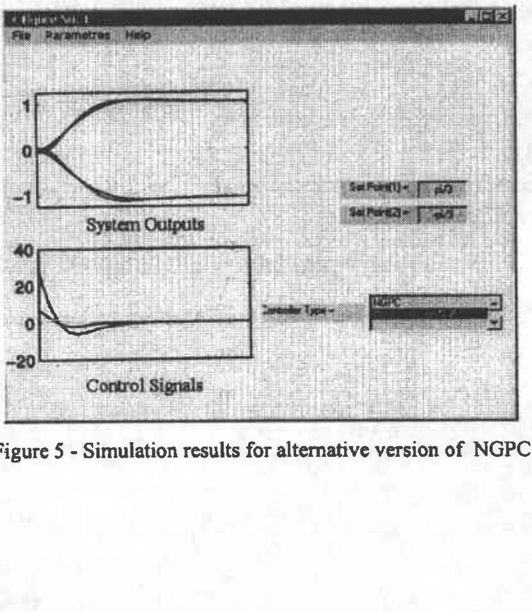
Figure 3 - "Controller Selection" window for Matlab/Simulink interface program for Prediction-free NGPC.

The simulations start running, when "File" menu and then "Run" is selected. Simulation results for NGPC and prediction-free version of NGPC are given in Figure 4 and Figure 5, respectively.



Tau1 = 0.2
Tau2 = 0.3
Set Point(1) = pi/3
Set Point(2) = -pi/3
Controller Type = GPC / NGPC / Prediction-free NGPC

Figure 4 - Simulation results for NGPC.



Set Point(1) = pi/3
Set Point(2) = -pi/3
Controller Type = NGPC

Figure 5 - Simulation results for alternative version of NGPC.

VI- CONCLUSION

In this work, an alternative version of Non-linear Generalised Predictive Control has been shown and a new Matlab/Simulink interface program has been designed.

Twolink-manipulator with a modified sensor, which has a non-linear characteristics and unstable inverse dynamics, is considered as a plant to be controlled. The simulations were performed on a Sun Sparc Ultra 1 using Matlab/Simulink and the symbolic-algebra package called Reduce. NGPC Algorithm needs on-line optimization of performance index given by Equation (23). Therefore, simulation time is 1296.15 seconds for the reference value of $[\pi/3,-\pi/3]$ and the prediction horizon is 0.2-0.3 seconds. An alternative (prediction-free) version of NGPC Algorithm completes the same simulation example in only 11.24 seconds for the reference value of $[\pi/3,-\pi/3]$ with no prediction horizon. As a result of simulations, it can easily be said that an alternative version of NGPC can achieve almost same performance 100 times faster than NGPC. It is also very important to notify that it is impossible to apply GMV controller to such a unstable inverse dynamics system, but extending GMV to an alternative version of GPC makes it possible.

Future and parallel works are related to enhance fast control action in GPC and the aim is to make GPC performance better and faster[18-20].

VII- REFERENCES

[1] Aström, K.J., B. Wittenmark – "On Self-tuning Regulators" – Automatica 9, 185-199, 1973.

[2] Clarke, D.W., C. Mohtadi, P.S.Tuffs–"Generalised Predictive Control-Part I. The Basic Algorithm"-Automatica - 23(2), 137-148,1987.

[3] Clarke, D.W., C. Mohtadi, P.S.Tuffs–"Generalised Predictive Control-Part II. Extensions and Interpretations"-Automatica– 23(2), 149-160,1987.

[4] Clarke, D.W., C. Mohtadi – "Properties of Generalised Predictive Control" – Automatica - 25(2), 859-875, 1989.

[5] Gawthrop, P.J. – "Continuous-time Self-tuning control - a unified approach. In K.J. Aström and B. Wittenmark (Eds.)" – Preprints of the 2nd IFAC workshop on Adaptive Systems in Control ans Signal Processing, Lund, Sweden, 1986.

[6] Gawthrop,P.J. – "Self-tuning PID controllers: Algorithms and implementation" – IEEE Transactions on Automatic Control, AC-31(3), 201-209,1986.

[7] Demircioğlu,H. – "Continuous-time Self-tuning Algorithms" – Ph.D.Thesis, Glasgow University, Faculty of Engineering, Supervised by P.J. Gawthrop, 1989.

[8] Gawthrop,P.J., H.Demircioğlu – "Continuous-time Generalised Predictive Control" – The 3rd IFAC workshop on Adaptive Systems in Control and Signal Processing, Glasgow, U.K., 1989.

[9] Gawthrop,P.J.,H.Demircioğlu – "GPC in a Continuous-time context" – Proceedings of the 1st European Control Conference, ECC91, pp55-74, 1991.

[10] Gawthrop,P.J.,H.Demircioğlu – "Continuous-time Generalised Predictive Control" – Automatica, 27(1), 55-74, 1991.

[11] Gawthrop,P.J.,H. Demircioğlu – "Multivariable Continuous-time Generalised Predictive Control" – Automatica, 28(4), 697-713, 1992.

[12] Demircioğlu,H. – "Sürekli Zaman Genelleştirilmiş Öngörülü Denetim (SÜGÖNDE)" – Otomatik Kontrol Bilimsel Toplantısı,TOK'94, s.409-418, 6-7 Nisan 1994.

[13] Gawthrop, P.J., H. Demircioğlu, I.Siller-Alcala – "Multivariable Continuous-time Generalised Predictive Control:A state-space approach to linear and nonlinear systems." – CSC Research Report CSC98001, Centre for Systems and Control, University of Glasgow, 1998.

[14] Gawthrop, P.J.,H. Demircioğlu, I.Siller-Alcala – "Multivariable Continuous-time Generalised Predictive Control:A state-space approach to linear and nonlinear systems." – Proc.IEE Pt.D: Control, Theory and Applications, 145(3), 241-250, 1998.

[15] Gawthrop, P.J., T. Arsan – "Exact Linearisation is a Special Case of Non-linear GPC" – CSC Research Report CSC98005, Centre for Systems and Control, University of Glasgow, 1998.

[16] Gawthrop, P.J., T. Arsan – "Exact Linearisation is a Special Case of Non-linear GPC" – Workshop on Non-linear Model Predictive Control-Assessment and Future Directions, Ascona, Switzerland, June 3-5, 1998.

[17] Kwakernaak, H. and R.Sivan - "Linear optimal Control Systems"-, Wiley, 1972.

[18] Ronco, E., Arsan, T., and Gawthrop, P. J. –"Open-loop Intermittent Feedback Control: Practical Continuous-time GPC" – *Centre for Systems and Control Research Report,* CSC-98015, University of Glasgow, U.K, 1998.

[19] Ronco, E., Arsan, T., and Gawthrop, P. J., –"Open-loop Intermittent Feedback Control: Practical Continuous-time GPC", *IEE Control, Theory and Applications*, accepted, 1998.

[20] Ronco, E., Arsan, T., Gawthrop, P. J..and Hill, D.J., 1998. – "A Globally Valid Continuous-time GPC through Successive System Linearizations" – *Laboratory for dynamical Systems and Control Technical Report",* EE-98003, The University of Sydney, Australia, 1998.