

PERANCANGAN *MODEL PREDICTIVE CONTROL* (MPC) PADA SISTEM PENDULUM TERBALIK

Cahyantari Ekaputri

Fakultas Teknik Elektro, Universitas Telkom
cahyantarie@telkomuniversity.ac.id

Abstrak

Model Predictive Control (MPC) memiliki ide dasar untuk menghasilkan nilai masukan suatu proses sebagai solusi masalah optimisasi yang real-time. Optimisasi tersebut dilakukan berdasarkan pada pemodelan suatu sistem. MPC digunakan untuk mengatasi masalah kendali multivariable yang memungkinkan ada lebih dari satu variable yang cukup signifikan terhadap proses.

Kelebihan MPC adalah dapat bekerja secara efektif saat menghadapi batasan-batasan (*constraints*) dari aktuator. Kelemahan dari MPC adalah perhitungan yang cukup rumit dan membutuhkan waktu yang cukup lama untuk pemrosesan data dibandingkan pengendali lain. Tulisan ini membahas mengenai aplikasi MPC pada sistem pendulum terbalik.

Tahapan yang dilakukan dimulai dari pemodelan sistem, perancangan sistem menggunakan MATLAB, simulasi perancangan menggunakan MATLAB dan analisis hasil. Setelah tahapan tersebut disimulasikan, dapat dilihat bahwa pengendali MPC yang dirancang dapat mengembalikan batang pendulum ke titik seimbangannya saat diberi gangguan.

Kata kunci : *MPC, pendulum terbalik.*

Abstract

Model Predictive Control (MPC) is based on the idea to produce control input as a solution to real-time optimization problem. Optimization itself is based on the system model. MPC is used to solve multivariable control problem which has more than one variable that may have significant effect on the process.

The advantage of MPC is that MPC can works effectively within constraints of the real actuator which are relatively narrow. The disadvantage of MPC lies on its complex algorithm that needs longer time than the other controller. This paper discuss the MPC application for inverted pendulum.

The general guideline of project are modeling the system, designing the system using MATLAB, simulating the design using MATLAB, and analyzing the result. After the guideline have been simulated, it can be seen that MPC can return the pendulum rod at the balance point by the time external force applied.

Keywords: *MPC, inverted pendulum.*

1. Pendahuluan [10 pts/Bold]

Model Predictive Controller (MPC) merupakan sebuah teori kendali yang cukup sering digunakan pada pengendalian dalam domain sistem kompleks. MPC sudah digunakan dalam beberapa tahun terakhir dalam proses industri. MPC sangat efektif untuk meningkatkan optimisasi dan menanggulangi masalah kendali yang biasanya berupa batasan-batasan (*constraints*) yang dimiliki oleh aktuator. Batasan-batasan ini merupakan suatu permasalahan dalam perancangan sistem karena umumnya sistem beroperasi secara optimal pada daerah kerja yang dekat pada batasan tersebut.

Prinsip utama dari MPC adalah bagaimana cara untuk memilih tindakan kontrol yang tepat dengan perulangan dalam menyelesaikan masalah kendali optimum.

Tujuannya adalah meminimasi kriteria kinerja dari suatu sistem sampai pada horizon tertentu, yang umumnya disebabkan adanya batasan-batasan masukan dan keluaran di masa depan yang perilakunya dikomputasi berdasarkan model proses. Masalah yang sering dihadapi dalam perancangan MPC ini adalah bagaimana menjamin kestabilan lup tertutup sistem, menanggulangi ketidakpastian dari model yang dirancang, dan mengurangi komputasi yang dilakukan.

Tulisan ini akan membahas tentang aplikasi pengendali MPC. Aktuator yang digunakan adalah pendulum terbalik. Hal ini dikarenakan pendulum terbalik menggambarkan ketidakstabilan sistem. Ketidakstabilan merupakan suatu permasalahan utama dalam perancangan sistem kendali, sehingga diharapkan aktuator ini dapat menggambarkan pemodelan suatu sistem secara umum.

2. Model Predictive Control (MPC)

Model Predictive Control (MPC) merupakan suatu teknik pengendalian dalam bentuk waktu diskrit yang menggunakan model proses dari suatu sistem. MPC bertujuan untuk memprediksikan kelakuan proses di masa depan untuk memperoleh tindakan kendali lup tertutup yang optimal dengan meminimalisasikan fungsi objektif dalam suatu rentang horizon waktu yang terbatas dalam batasan tertentu. Perhitungan MPC akan menghasilkan suatu nilai yang akan dipergunakan untuk suatu proses masukan sebagai solusi masalah optimasi yang *real-time*.

Pada dasarnya MPC memprediksi keluaran berikutnya dari suatu proses dalam suatu rentang horizon waktu tertentu dengan menggunakan masukan dan keluaran dari proses sebelumnya. Setelah itu, akan dilakukan perhitungan untuk menghasilkan sinyal kendali dalam rentang horizon waktu tersebut dan melakukan optimasi. Ilustrasi tersebut dapat dilihat pada Gambar 1.

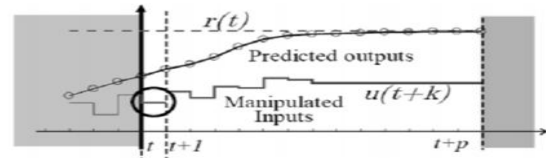
Fungsi utama dari MPC adalah mengoptimalkan kelakuan proses dengan memanipulasi atau mengendalikan masukan sistem yang diminimalisasi oleh fungsi objektif kuadratik dalam batasan proses aktuator. Struktur dasar dari MPC digambarkan seperti gambar 2 di bawah ini.

Gambar 3 memberikan ilustrasi konsep dasar MPC bekerja dalam memprediksi perilaku dari suatu kendalian untuk horizon sepanjang M. Dengan mempergunakan model yang akurat, MPC dapat memprediksikan bagaimana perilaku keluaran dari proses pengendalian untuk setiap sinyal kendali yang diberikan kepada kendalian sehingga dapat diperoleh sinyal kendali yang paling optimal dengan memperhitungkan batasan kendalian di dalamnya.

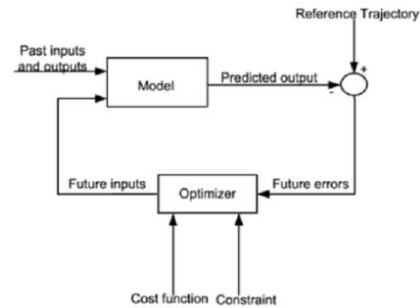
Respon dari sinyal kendali di masa depan dikomputasi sebagai kumpulan solusi masalah optimisasi lup terbuka sebanyak horizon yang ditentukan dengan elemen pertama dari respon sinyal kendali optimal yang dipergunakan sebagai masukan proses selanjutnya, sedangkan elemen yang lain tidak akan digunakan. Kemudian horizon waktu akan bergeser dan akan dikomputasi kembali sehingga mendapatkan kumpulan solusi optimisasi yang baru dan akan digunakan berikutnya. Proses ini terus berulang sehingga menghasilkan keluaran yang optimal.

Gambar 4 merepresentasikan secara sederhana diagram alir dari algoritma MPC. Langkah awal dimulai dengan membuat model proses yang paling mendekati kendalian.

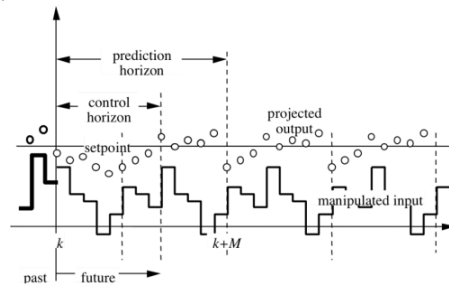
Apabila masukan kendalian, u, dan keluaran proses, y, dapat diamati, maka perilaku sistem di masa depan dapat diprediksi hanya dengan mengetahui model proses dan sinyal kendali di masa depan uk+j.



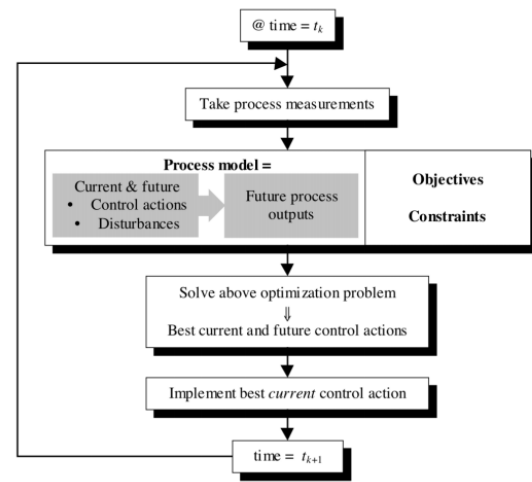
Gambar 1 Ilustrasi MPC



Gambar 2 Struktur Dasar MPC



Gambar 3 Konsep Dasar MPC



Gambar 4 Skema MPC

2.1 MPC Tracking Problem

Apabila diketahui suatu fungsi transfer dari kendalian yang ditulis dalam bentuk *state space* seperti berikut:

$$x_{k+1} = \tilde{A}x_k + \tilde{B}u_k \tag{1}$$

$$y_k = \tilde{C}x_k \tag{2}$$

dengan $x_k \in R^n$, $y_k \in R^l$, dan $u_k \in R^m$.

Keluaran y_k dikehendaki untuk dapat mencapai suatu nilai masukan referensi $W_k \in R^l$. Untuk

memperoleh *zero tracking error* maka sebuah integrator dalam bentuk

$$u_k = u_{k-1} + \Delta u_k \quad (3)$$

ditambahkan pada kendalian sehingga diperoleh kendalian yang baru yang telah ditambahkan komponen integrator (*augmented plant*) dengan persamaan matematis:

$$\bar{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = A\bar{x}_k + \Delta B u_k \quad (4)$$

$$y_k = C\bar{x}_k \quad (5)$$

dimana $A := \begin{bmatrix} \bar{A} & \bar{B} \\ 0 & I_m \end{bmatrix}$, $B := \begin{bmatrix} \bar{B} \\ I_m \end{bmatrix}$, dan $C := [\bar{C} \ 0]$.

Fungsi objektif didefinisikan sebagai berikut:

$$J = e_{k+N_y}^T \bar{Q}_N e_{k+N_y} + \sum_{t=0}^{N_y-1} e_{k+t}^T \bar{Q} e_{k+t} + \sum_{t=0}^{N_u-1} \Delta u_{k+t}^T \bar{R} \Delta u_{k+t} \quad (6)$$

dimana $e_k = y_k - w_k$, N_y = horizon prediksi, N_u = horizon kendali, $N_u \leq N_y$, $\bar{Q} = \bar{Q}^T \geq 0$ dan $\bar{R} = \bar{R}^T \geq 0$. Untuk menyederhanakan, dianggap $N = N_u = N_y$ untuk mendefinisikan

$$Y = \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+N} \end{bmatrix}, W = \begin{bmatrix} w_{k+1} \\ w_{k+2} \\ \vdots \\ w_{k+N} \end{bmatrix}, U_d = \begin{bmatrix} u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+N} \end{bmatrix} \quad (7)$$

Sehingga didapat:

$$Y = \Phi \bar{x}_k + \Gamma U_d \quad (8)$$

dimana Φ dan Γ merupakan fungsi yang bersesuaian terhadap matriks A , B , dan C .

Apabila batasan amplitude pada masukan kendali dipertimbangkan dalam persamaan, maka terlebih dahulu harus didefinisikan

$$U_d = DU - E_1 u_{k-1} \quad (9)$$

$$\text{dengan } D = \begin{bmatrix} I_m & 0 & \cdots & 0 & 0 \\ -I_m & I_m & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I_m & 0 \\ 0 & 0 & \cdots & -I_m & I_m \end{bmatrix} \quad (10)$$

untuk mendapatkan persamaan fungsi yang ekuivalen.

Komputasi bertujuan untuk menemukan suatu nilai U yang meminimisasi

$$J_T := \frac{1}{2} (U + M^{-1} F_X X)^T M (U + M^{-1} F_X X) \quad (11)$$

dengan batasan $1u_{\min} \leq U \leq 1u_{\max}$

$$X := [x_k^T \ u_{k-1}^T \ -W^T]^T \quad (12)$$

$$M := \bar{\Gamma}^T Q \bar{\Gamma} + D^T R D \quad (13)$$

$$F_X := \bar{\Gamma}^T Q \bar{\Phi} + D^T R G \quad (14)$$

Matriks $\bar{\Phi}$, $\bar{\Gamma}$, Q , R , dan G merupakan fungsi dari matriks A , B , C , D , E_1 , \bar{Q} , dan \bar{R} .

Dengan terlebih dahulu mendefinisikan

$$U_0 := -M^{-1} F_X X = -M^{-1} [F \ H] \begin{bmatrix} \bar{x}_k \\ -Z \end{bmatrix} \quad (15)$$

dan sesuai dengan pembuktian ekuivalensi *quadratic programming* dengan lup aljabar yang ada, maka

MPC tracking dapat diimplementasikan sesuai diagram blok pada gambar 5.

2.2 Algoritma 3 MPC

Algoritma yang digunakan adalah algoritma 3 yang merupakan algoritma iterasi *quadratic programming*. Algoritma ini dapat direpresentasikan dalam diagram blok seperti gambar 6.

Algoritma iterasi dapat ditunjukkan dengan adanya umpan balik (*feedback*). Persamaan matematis dari diagram blok diatas didefinisikan:

$$W_s = F_f U_0 + F_b W \quad (16)$$

$$W = \Psi_b(W_s) \quad (17)$$

Sehingga implementasi iterasi dapat didefinisikan:

$$W_{k+1} = \Psi_b(F_f U_0 + F_b W_k) \quad (18)$$

dengan Ψ_b merupakan fungsi non linear dari saturasi aktuator.

$$\Psi_b(W_s) := \Psi - \frac{1}{2} I := \Psi(W_s) - \frac{1}{2} W_s \quad (19)$$

$$F_f := 2(I + M)^{-1} M \quad (20)$$

$$F_b := 2(I + M)^{-1} (I - M) \quad (21)$$

Persamaan iterasi di atas (18) bersifat konvergen, sehingga solusi dari *quadratic programming* di atas dapat didefinisikan:

$$U^* = W^* + \frac{1}{2} (F_f U_0 + F_b W^*) \quad (22)$$

3. Perancangan Sistem

Secara umum, sistem dari pengendalian pendulum terbalik yang akan dirancang mengikuti diagram blok kendalian pada gambar 7.

Referensi input sistem yang digunakan adalah nilai setpoint dari state θ (sudut saat batang pendulum berdiri tegak/posisi vertikal). Pengendali yang digunakan adalah algoritma 3 MPC (*Model Predictive Control*). Aktuator pada sistem ini adalah pendulum terbalik model 505 yang diproduksi oleh ECP (*Educational Control Product*). Sensor yang digunakan adalah sensor potensiometer yang ditempelkan pada penunjuk sudut batang pendulum. Nilai keluaran sistem yang digunakan untuk umpan balik adalah state θ (sesuai dengan referensi input sistem). State space kendalian sebagai berikut:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

Pada sistem ini, terdapat 4 buah state, sebuah masukan/input dan sebuah keluaran/output. Masukan yang digunakan pada sistem ini adalah besarnya gaya yang diperlukan untuk menggerakkan batang pendulum agar mencapai kestabilan saat berdiri vertikal ($F(t)$), sedangkan keluaran sistem pendulum ini adalah besarnya sudut/posisi batang pendulum terhadap sumbu vertikal (θ). State yang digunakan pada sistem ini:

$$x = [\theta \ \dot{\theta} \ x \ \dot{x}]^T$$

Keterangan:

θ = sudut/posisi batang pendulum terhadap sumbu vertical

$\dot{\theta}$ = kecepatan sudut batang pendulum saat bergerak
 x = posisi batang luncur terhadap batang pendulum
 \dot{x} = kecepatan batang luncur saat bergerak

State space dari sistem pendulum terbalik dapat dimodelkan secara matematis persamaan ruang dalam waktu kontinu sebagai berikut:

$$\dot{x} = A_{\text{bar}C} x + B_{\text{bar}C} u$$

$$y = C_{\text{bar}C} x + D_{\text{bar}C} u$$

State space dari sistem pendulum terbalik dapat dimodelkan secara matematis persamaan ruang dalam bentuk diskrit sebagai berikut:

$$x_{k+1} = A_{\text{bar}} x_k + B_{\text{bar}} u_k$$

$$y_k = C_{\text{bar}} x_k + D_{\text{bar}} u_k$$

Dengan menggunakan nilai parameter waktu sampling 0,05 detik, persamaan matematis sistem dalam bentuk diskrit menjadi:

$$x_{k+1} = \begin{bmatrix} 0.9390 & 0.1959 & -1.0796 & 0.0723 \\ -0.6033 & 0.9390 & -10.6847 & 1.0796 \\ 0 & 0 & 1 & -0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.0688 \\ 0.6547 \\ 0.0142 \\ -0.1418 \end{bmatrix} u_k$$

$$y_k = [-0.6039 \ 0 \ 0.7971 \ 0] x_k + [0] u_k$$

Input dan output persamaan diatas telah dilakukan penyeimbangan realisasi state space berbasis Gramian.

4. Simulasi dan Analisis

Simulasi dilakukan dengan menggunakan *simulink* MATLAB. Simulasi dilakukan dengan mengubah horizon. Tujuan dari perancangan simulink ini adalah untuk menentukan nilai horizon yang optimal dari sistem ini untuk nilai iterasi yang terbatas. Hal tersebut dilihat dari respon waktu sistem dan sinyal kendalian yang dihasilkan.

Parameter yang digunakan pada simulasi ini antara lain:

- Waktu Sampling = 0.04 detik
- Nilai $\bar{Q} = 1$ dan $\bar{R} = 1$
- Setpoint yang digunakan = 0
- Waktu simulasi = 5 detik

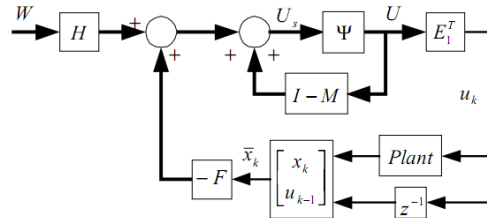
Pada perancangan, diambil waktu sampling sebesar 0,04 detik. Pemilihan ini didasarkan pada rule of thumb, yang menyatakan bahwa waktu sampling yang ideal dari suatu sistem adalah sepersepuluh dari konstanta waktu fungsi alih kendalian. Ada pula teori Nyquist-Shannon yang menyatakan bahwa waktu sampling minimal adalah setengah dari konstanta waktu fungsi alih kendalian.

Setelah menentukan parameter yang diperlukan, dilakukan perancangan pengendali MPC. Perancangan MPC dilakukan dengan menghitung nilai-nilai matriks yang menjadi dasar perhitungan algoritma 3 MPC. Kemudian dilakukan simulasi untuk mengetahui horizon yang digunakan.

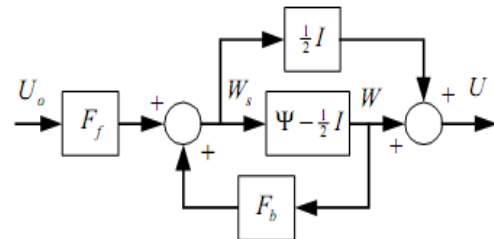
Dari hasil simulasi didapat respon waktu sistem yang paling baik terjadi saat nilai horizon 6. Sehingga diambil nilai horizon 6 sebagai nilai horizon yang optimal.

Untuk nilai horizon 6, nilai matriks algoritma 3 MPC didapat sebagai berikut:

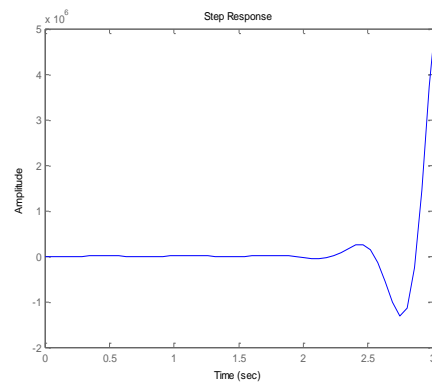
$$K = \begin{bmatrix} -0.0452 & -0.0843 & -0.0256 & 0.1063 & 0.3111 & 0.6849 \\ 0.0160 & -0.1102 & -0.4240 & -0.6361 & -0.3488 & 0.9960 \\ -0.0068 & -0.1347 & -0.6212 & -1.2412 & -1.0912 & 1.1023 \\ -0.0446 & -0.2428 & -0.7212 & -1.4384 & -1.4896 & 1.0767 \\ -0.0676 & -0.3200 & -0.8292 & -1.4629 & -1.5155 & 0.9924 \\ -0.0738 & -0.3429 & -0.8670 & -1.4857 & -1.4544 & 0.9473 \end{bmatrix}$$



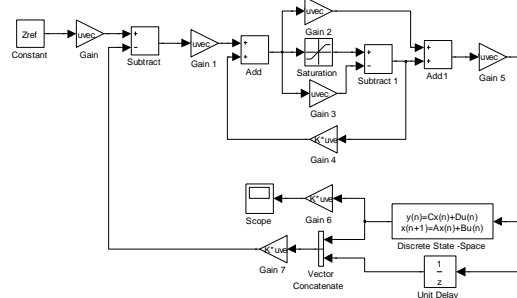
Gambar 5 Diagram Blok Implementasi MPC



Gambar 6 Algoritma 3 MPC



Gambar 8 Respon Sistem terhadap Masukan Undakan (Step)



Gambar 9 Diagram Blok Sistem dengan Algoritma 3 MPC pada Simulink

Tabel 1 Hasil Simulasi Algoritma 3 MPC

Horizon	Settling Time	Maximum Peak	Jumlah Iterasi QP
6	12 s	5 %	7
7	18 s	10 %	7
8	15 s	8 %	17
9	16 s	7 %	38
10	18 s	10 %	78
12	17 s	8 %	225
15	20 s	8 %	690

$$L = \begin{bmatrix} 0.3066 & -0.3124 & 16.3172 & -7.7974 & -0.1494 \\ 0.5005 & 0.1884 & 3.0351 & -3.8213 & 0.0528 \\ 0.5951 & 0.6805 & -12.3626 & 1.6161 & -0.0225 \\ 0.6641 & 0.9481 & -21.1034 & 4.8929 & -0.1475 \\ 0.7187 & 1.0424 & -24.0782 & 6.0627 & -0.2236 \\ 0.7392 & 1.0599 & -24.5622 & 6.2619 & -0.2443 \end{bmatrix}$$

$$F_f = \begin{bmatrix} 0.9343 & 0.1657 & 0.0535 & -0.0457 & -0.0562 & -0.0253 \\ 0.1657 & 0.4831 & -0.0893 & -0.0255 & -0.0293 & -0.0144 \\ 0.0535 & -0.0893 & 0.3374 & -0.1369 & -0.0182 & -0.0057 \\ -0.0457 & -0.0255 & -0.1369 & 0.3180 & -0.1378 & -0.0122 \\ -0.0562 & -0.0293 & -0.0182 & -0.1378 & 0.3140 & -0.1514 \\ -0.0253 & -0.0144 & -0.0057 & -0.0122 & -0.1514 & 0.1692 \end{bmatrix}$$

$$F_b = \begin{bmatrix} 0.1314 & -0.3313 & -0.1071 & 0.0913 & 0.1124 & 0.0506 \\ -0.3313 & 1.0338 & 0.1786 & 0.0509 & 0.0585 & 0.0287 \\ -0.1071 & 0.1786 & 1.3252 & 0.2739 & 0.0364 & 0.0115 \\ 0.0913 & 0.0509 & 0.2739 & 1.3639 & 0.2757 & 0.0243 \\ 0.1124 & 0.0585 & 0.0364 & 0.2757 & 1.3720 & 0.3028 \\ 0.0506 & 0.0287 & 0.0115 & 0.0243 & 0.3028 & 1.6616 \end{bmatrix}$$

$$E_f^T = [1 \ 0 \ 0 \ 0 \ 0 \ 0]$$

Saturasi: $\Psi(x) = \begin{cases} \Psi(x) = -0.5, & \text{untuk } x < -0.5 \\ \Psi(x) = x, & \text{untuk } -0.5 < x < 0.5 \\ \Psi(x) = 0.5, & \text{untuk } x > 0.5 \end{cases}$

Deadzone: $\Psi_a(x) = \begin{cases} \Psi_a(x) = -0.5 - 0.5x, & \text{untuk } x < -0.5 \\ \Psi_a(x) = 0.5x, & \text{untuk } -0.5 < x < 0.5 \\ \Psi_a(x) = 0.5 - 0.5x, & \text{untuk } x > 0.5 \end{cases}$

Gambar 10, 11, dan 12 menunjukkan hasil simulasi sistem pendulum terbalik dengan menggunakan pengendali MPC algoritma 3 dengan nilai horizon 6. Gambar 10 merupakan respon waktu sistem. Gambar 11 merupakan sinyal kendali sistem. Gambar 12 merupakan jumlah iterasi lup *quadratic programming*.

5. Kesimpulan dan Saran

Berdasarkan penelitian ini, dapat ditarik beberapa kesimpulan sebagai berikut :

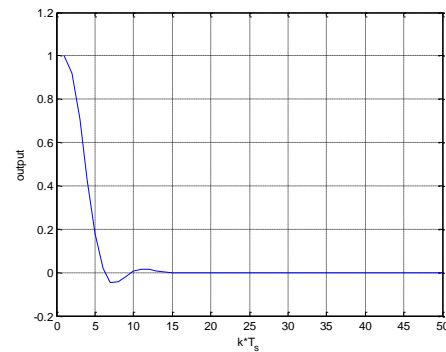
- a) Algoritma 3 MPC dapat dirancang sebagai pengendali pada sistem pendulum terbalik ECP model 505 dengan spesifikasi SISO (Single Input Single Output) dan floating point.
- b) Algoritma 3 MPC dapat menghasilkan suatu sinyal kendali yang optimal berdasarkan prediksi dari respon sistem. Algoritma MPC juga dapat memperhitungkan adanya batasan (constraints).
- c) Prediksi yang dilakukan pada algoritma Model Predictive Control dilakukan untuk suatu nilai horizon yang terbatas. Nilai horizon yang digunakan adalah horizon 6, karena berdasarkan simulasi nilai horizon ini memiliki respon waktu yang paling baik. Untuk nilai horizon dibawah 6, sistem bersifat tidak stabil (terjadi osilasi).

Saran untuk proses penelitian berikutnya antara lain:

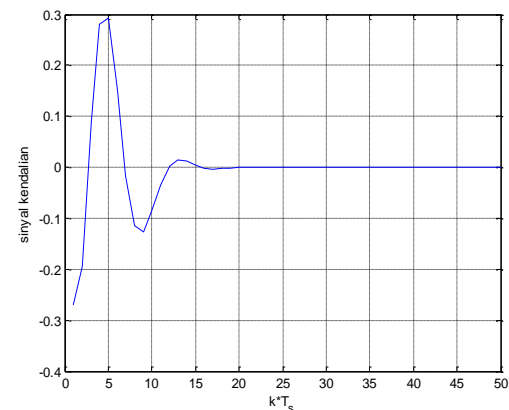
- a) Algoritma MPC merupakan jenis teknik kendali yang mampu menghasilkan kinerja optimal bahkan untuk kendali yang

- kompleks sekalipun seperti MIMO (Multi Input Multi Output).
- b) kendali yang kompleks dengan prosesor yang lebih kompleks.
- c) Perancangan pemodelan harus sesuai dengan keadaan sebenarnya, karena pengendali MPC berbasis pada pemodelan.

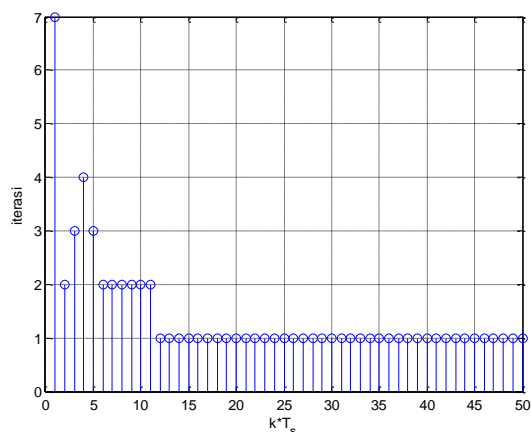
Implementasi pengendali MPC yang mampu melakukan komputasi yang kompleks dalam waktu yang singkat untuk



Gambar 10 Respon Waktu Model Predictive Control Horizon 6



Gambar 11 Sinyal Kendali Model Predictive Control Horizon 6



Gambar 12 Iterasi Model Predictive Control Horizon 6

Daftar Pustaka:

- [1] Ekaputri, Cahyantari and Syaichu-Rohman, Arief. *Implementation Model Predictive Control (MPC) Algorithm-3 for Inverted Pendulum*. IEEE Control and System Graduate Research Colloquium (ICSGRC). 2012. Shah Alam, Selangor, Malaysia.
- [2] Franklin, G. F. and J. D. Powell, and A. Emami-Naeni (1997), *Feedback Control on Dynamic Systems, 3rd ed*, Addison Wesley, New York.
- [3] Gorinevsky, Dimitry. *Model Predictive Control: Industrial MPC*, Stanford University. www.stanford.edu/class/ee392m/Lecture15_MPC.pdf.
- [4] Nikolaou, Michael. *Model Predictive Controllers: a Critical Synthesis of Theory and Industrial Needs*. www.chee.uh.edu/faculty/nikolaou/MPCtheoryRevised.pdf.
- [5] Ogata, Katsuhiko (1995), *Discrete Time Control System, 2nd ed*, Prentice-Hall, New Jersey.
- [6] Ogata, Katsuhiko (1997), *Modern Control Engineering, 3rd ed*, Prentice-Hall, New Jersey
- [7] Orukpe, P.E. *Basics of Model Predictive Control*, Imperial College, London. www3.imperial.ac.uk/portal/pls/portal/1/50918.PDF.
- [8] Syaichu-Rohman, Arief dan Richard H. Middleton - *A Multivariable Nonlinear Algebraic Loop*. <http://www.nt.ntnu.no/users/skoge/prost/proceedings/ecc03/pdfs/001.pdf>
- [9] Syaichu-Rohman, Arief, and Middleton, Richard H. *Convergence Study of Some Fixed Point Iteration QP Algorithms*. In Proceedings of the 43rd Conference on Decision and Control. Pradise Island, Bahamas.
- [10] Syaichu-Rohman, Arief. *Plant Performance Optimization*. The 6th Asian Control Conference. July 18-21, 2006. Inna Grand Bali Beach Hotel, Sanur, Bali, Indonesia.
- [11] Syaichu-Rohman. 2006. *Introduction to Model Predictive Control*. Indonesia: Workshop of the 6th Asian Control Conference.