ANALISIS SISTEM DIGITAL PADA SISTEM ROTATIONAL/TRANSLATIONAL ACTUATOR

LAPORAN TUGAS AKHIR

TEKNIK KENDALI DIGITAL

(TKEE163231)

Disusun Oleh:

Resha Dwika Hefni Al-Fahsi

16/394959/TK/44251

PROGRAM STUDI TEKNIK ELEKTRO

DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI

FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA

2019

Abstrak

Laporan tugas akhir ini membahas tentang analisis sistem digital pada sistem *rotational/translational actuator*. Kinematis dari sistem tersebut dirumuskan dari model non-linearnya. Kemudian untuk mempermudah dalam analisis kendali digital dilakukan linearisasi dan diskretisasi sistem sehingga dapat dicari *state space*-nya. Dilakukan juga analisis *controllable* dan *observable* pada sistem tersebut. Menggunakan LQR untuk mencari nilai matriks *gain* yang tepat untuk kestabilan. Semua simulasi perhitungan pada sistem dilakukan menggunakan Matlab® [1] R2016a.

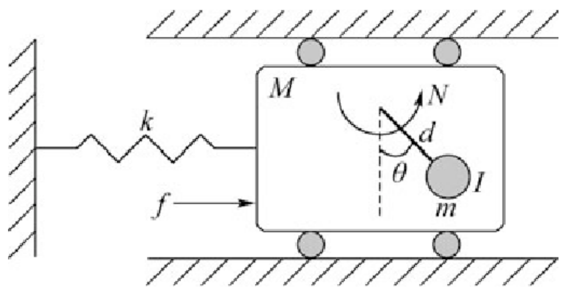
[1] Matlab and Simulink are trademarks of MathWorks Inc. USA.

Daftar Isi

Bab 1

Pendahuluan

Tujuan dari pembuatan laporan ini untuk memberikan penjelasan tentang perancangan *robust* $\mathcal{H}_\infty$ *control* pada sistem *rotational/translational actuator*. Sistem *rotational/translational actuator* atau biasa disingkat RTAC merupakan sebuah sistem berupa gabungan dua aktuator yang salah satunya bergerak secara rotasi dan aktuator lainnya bergerak secara translasi[2]. Dalam praktiknya aktuator tersebut berupa sistem osilasi yaitu pendulum yang berperan dalam gerakan rotasi dan pegas yang berperan dalam gerakan translasi. Sistem RTAC sendiri sering digambarkan dengan sebuah gerobak yang terpasang sistem pegas yang dipasang secara tegak lurus pada dinding dan di dalam gerobak tersebut terdapat pendulum yang berosilasi (bergerak secara rotasi) sehingga akan menyebabkan gerakan translasi pada gerobak[3].



Gambar 1. Pemodelan Sistem RTAC

Pada bab 2, kita mendefinisikan sistem RTAC dalam *state-space*. Sistem RTAC sendiri pada dasarnya merupakan sistem non-linear sehingga diperlukan proses linearisasi untuk mendapatkan bentuk *state-space*-nya.
Pada bab 3 dilakukan proses diskretisasi pada sistem sehingga dapat dilakukan analisis kendali digital untuk mendapatkan *state-space* dalam ranah diskret. Dengan mengetahui bentuk *state-space* tersebut dapat diperiksa apakah sistem tersebut *observable* dan *controllable*.
Pada bab 4 dilakukan perhitungan menggunakan LQR untuk mendapatkan nilai matriks *gain* K. Matriks tersebut digunakan untuk membuat sistem stabil dengan membuat sistem menjadi *closed-loop*.

Bab 2

Sistem *Rotational/Translational Actuator*

Seperti yang dijelaskan sebelumnya sistem RTAC merupakan sistem non-linear. Sistem non-linear sendiri memiliki *state* yang *dependent* dengan *state* lainnya. Pada sistem yang akan dibahas, terdapat 4 *state* yang didefiniskan pada *state-space*. Untuk mendapatkan *state-space* sistem tersebut dilakukan linearisasi dengan metode linearisasi Jacobian.

2.1 Pemodelan

Pemodelan sistem RTAC didefinisikan dengan *state-space* non-linearnya[4]:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \dfrac{(I+mr^2)(mrx_4^2 \sin x_3 - kx_1) - (mr\cos x_3)N + (I+mr^2)F}{(M+m)(I+mr^2) - m^2r^2\cos^2 x_3} \\ x_4 \\ \dfrac{mr\cos x_3(kx_1 - mrx_4^2 \sin x_3 - F) + (M+m)N}{(M+m)(I+mr^2) - m^2r^2\cos^2 x_3} \end{bmatrix}$$

Di mana:

x_1= perpindahan gerobak dari titik setimbang, m
x_2= kecepatan linear gerobak, m/s
x_3= posisi sudut pendulum, rad
x_4= kecepatan sudut pendulum, rad/s
M = massa gerobak, 1.3608 kg
m = massa pendulum, 0.096 kg
r = panjang tali pendulum, 0.0592 m
k = konstanta/kekakuan pegas, 186.3 N/m
F = gaya translasi, gaya gangguan, 5 N
N = torsi pada pendulum, kontrol pada sistem, 0.41861 Nm
I = inersia pendulum, 0.0002175 kg/m^2

2.2. Linearisasi

Linearisasi Jacobian sendiri menggunakan turunan parsial setiap *state* terhadap suatu fungsi yang mendefinisikan salah satu turunan *state* pada sistem. Dengan linearisasi Jacobian menggunakan *operating point*, $[x_1 \quad x_2 \quad x_3 \quad x_4] = \begin{bmatrix}0 & 0 & \frac{\pi}{4} & 0.785\end{bmatrix}$ dan $[F \quad N] = [0 \quad 0]$, didapatkan matriks-matriks *state-space*-nya:

Untuk matriks A-nya:

$$A = \begin{bmatrix} x_2 \\ \dfrac{(I+mr^2)(mrx_4^2 \sin x_3 - kx_1) - (mr\cos x_3)N + (I+mr^2)F}{(M+m)(I+mr^2) - m^2r^2\cos^2 x_3} \\ x_4 \\ \dfrac{mr\cos x_3(kx_1 - mrx_4^2 \sin x_3 - F) + (M+m)N}{(M+m)(I+mr^2) - m^2r^2\cos^2 x_3} \end{bmatrix} \begin{bmatrix} \partial x_1 \\ \partial x_2 \\ \partial x_3 \\ \partial x_4 \end{bmatrix}^T$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k(I + mr^2)}{(M+m)(I + (0.5mr^2))} & 0 & -\frac{m^2r^3(0.555((M+m)I + (Mmr^2)) + m(0.616I + 0.555mr^2))}{((M+m)(I + (0.5mr^2)))^2} & \frac{1.57\, Imr + 1.11\, m^2r^3}{(M+m)(I + (0.5mr^2))} \\ 0 & 0 & 0 & 1 \\ \frac{0.707\, mrk}{(M+m)(I + (0.5mr^2))} & 0 & 0 & -\frac{0.785\, m^2r^2}{(M+m)(I + (0.5mr^2))} \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -183.66 & 0 & -0.00279 & 0.00723 \\ 0 & 0 & 0 & 1 \\ 1332.1 & 0 & 0 & -0.04512 \end{bmatrix}$$

Sedangkan untuk matriks B-nya:

$$B = \begin{bmatrix} x_2 \\ \frac{(I + mr^2)(mrx_4^2 \sin x_3 - kx_1) - (mr \cos x_3)N + (I + mr^2)F}{(M+m)(I + mr^2) - m^2r^2 \cos^2 x_3} \\ x_4 \\ \frac{mr \cos x_3 (kx_1 - mrx_4^2 \sin x_3 - F) + (M+m)N}{(M+m)(I + mr^2) - m^2r^2 \cos^2 x_3} \end{bmatrix} \begin{bmatrix} \partial F \\ \partial N \end{bmatrix}^T$$

$$B = \begin{bmatrix} 0 & 0 \\ \frac{(I + mr^2)}{(M+m)(I + (0.5mr^2))} & -\frac{0.707\, mr}{(M+m)(I + (0.5mr^2))} \\ 0 & 0 \\ -\frac{0.707\, mr}{(M+m)(I + (0.5mr^2))} & \frac{M+m}{(M+m)(I + (0.5mr^2))} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0.98581 & -7.1505 \\ 0 & 0 \\ -7.1505 & 2592.5 \end{bmatrix}$$

Dengan matriks C = $\mathbf{I}_{4\times4}$ dan D = $\mathbf{0}_{4\times2}$. Sehingga bentuk *state-space*, $\dot{x} = Ax + Bu$ dan $\dot{x} = Cx + Du$, dari sistem tersebut telah terbentuk.

Bab 3

Sistem Digital

Pada praktiknya, sistem yang ada di dunia ini dikendalikan menggunakan komputer. Komputer sendiri merupakan suatu sistem yang bekerja pada ranah diskret. Sedangkan sistem yang ada di alam berbentuk kontinyu. Sehingga perlu dilakukan proses diskretisasi yaitu mengubah sistem pada ranah analog menjadi ranah digital.

3.1. Diskretisasi

Pemodelan sistem pada bab sebelumnya dilakukan pada ranah kontinyu waktu. Sehingga sistem tersebut perlu diubah ke bentuk sistem dalam waktu diskret. Berikut persamaan *state-space* waktu diskret[1].

$$\mathbf{A_d} = \mathbf{\Phi} = \mathcal{L}^{-1}[(s\boldsymbol{I} - \boldsymbol{A})^{-1}]$$

$$\boldsymbol{B_d} = \int_0^T \mathbf{\Phi}(\lambda)\, d\lambda\, \boldsymbol{B}$$

$$\boldsymbol{C_d} = \boldsymbol{C}$$

$$\boldsymbol{D_d} = \boldsymbol{D}$$

Persamaan di atas akan mengubah model *state-space* dalam ranah kontinyu waktu menjadi diskret.

3.2. *Contrabillity* dan *Observability*

Dengan mengetahui bentuk lengkap dari *state-space* dapat ditentukan *observability* dan *controllability* dari sistem tersebut. Untuk mencari *observability* dari suatu sistem digunakan matriks $O = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$ dan mencari *observability* dari suatu sistem digunakan matriks $C = [B \quad AB \quad \cdots \quad A^{m-1}B]$. Syarat sistem *observable* ketika matriks O merupakan matriks *full-column rank* dan sistem *controllable* ketika matriks C merupakan matriks *full-row rank*.

3.3. Program Matlab®

Dengan menggunakan Matlab® dapat dicari:

```
%Diskretisasi Sistem
fs = 250; %Hz
Ts = 1/fs; % periode sampling (sekon)
sys_ss = ss(A,B,C,D); % representasi state space
sys_ss_d = c2d(sys_ss, Ts)
[Ad, Bd, Cd, Dd] = ssdata(sys_ss_d)

Ob=obsv(Ad,Cd);
[n,m]=size(Ob);
unob = m-rank(Ob); %observable
if(unob==0)
   disp('Given System is Observable.');
else
   disp('Given System is Unobservable');
end

Co = ctrb(Ad,Bd);
[n,m]=size(Co);
unco=n-rank(Co); %controllable
if(unco==0)
   disp('Given System is Controllable.');
else
   disp('Given System is Uncontrollable');
end
```

Dari hasil simulasi didapatkan *state-space* waktu diskret dengan frekuensi cuplik 250 Hz didapatkan.

$$A_d = \begin{bmatrix} 0.9985 & 0.004 & 0 & 0 \\ -0.7342 & 0 & 0 & 0 \\ 0.0107 & 0 & 1 & 0.004 \\ 5.3253 & 0.0107 & 0 & 0.9998 \end{bmatrix}$$

$$B_d = \begin{bmatrix} 0 & -0.0001 \\ 0.0039 & -0.0284 \\ -0.0001 & 0.0207 \\ -0.0286 & 10.369 \end{bmatrix}$$

Dengan matriks $C_d = \mathbf{I}_{4\times4}$ dan $D_d = \mathbf{0}_{4\times2}$. *State-space* waktu diskret tersebut merupakan sistem yang *controllable* dan *observable*.

Bab 4

Perancangan Pengendali Digital

Pada teori *optimal control* suatu sistem dinamis berjalan dengan membuat minimal suatu *cost function*. Sistem dinamis tersebut dimodelkan dalam persamaan diferensial sedangkan *cost function* berupa fungsi kuadrat. Solusi dari persamaan kuadrat tersebut diselesaikan dengan metode LQR atau Linear Qudratic Regulator. LQR berguna untuk mencari kestabilan dalam sistem.

4.1. *Linear Quadratic Regulator*

Pada sistem yang akan dianalisis bukanlah sebuah sistem yang stabil, apabila dicari *eigenvalue*-nya, $det(sI - A) = 0$, memiliki nilai lebih dari 0. Di mana kestabilan sistem diindikasikan bahwa *pole-pole*-nya berada di sisi sebelah kiri bidang y. Dengan adanya nilai positif pada *eigenvalue* matriks A maka sistem tersebut bukanlah sistem yang stabil. Dengan menyelesaikan persamaan aljabar Riccati kontinyu waktu:

$$A^T X + XA - XBR^{-1}B^T X + Q = 0,$$

di mana R dan Q berturut-turut didefinisikan sebagai:

$$R_{m\times m} = \sigma I_{m\times m}$$
$$Q_{n\times n} = C_{n\times n}^T C_{n\times n}$$

Di mana σ merupakan sebuah konstanta yang diatur sedemikian sehingga dapat dicari matriks *gain* K yang membuat sistem stabil dengan menghitung:

$$K = R^{-1}B^T X,$$

dengan X merupakan solusi dari persamaan aljabar Riccati kontinyu waktu. Sehingga didapatkan matriks A baru dengan sistem *closed-loop*:

$$A_{new} = A - BK$$

Sistem akan stabil jika dan hanya jika semua *eigenvalue* dari matriks A benar-benar berada di dalam *unit circle* ($|\lambda(\mathbf{A_d})| < 1$).

4.2. Program Matlab®

Dengan menggunakan Matlab® dapat dicari:

```
%Mengecek kestabilan (melihat apakah pole berada di dalam unit circle)
pole = eig(Ad)
%Membuat stabil dengan menggunakan LQR
Qd=Cd'*Cd;
R=9999999999999999999*eye(2);
p=care(Ad,Bd,Qd,R);
K=inv(R)*Bd'*p
Anew=Ad-Bd*K; Bnew=Bd; Cnew=Cd; Dnew=Dd;
polenew=eig(Anew)
```

Dari hasil simulasi didapatkan semua nilai *eigenvalue*-nya berada di dalam *unit circle* sehingga sistem menjadi stabil. Perlu diperhatikan untuk mencapai kestabilan diperlukan konstanta σ atau input sangat besar (menuju tak hingga) sehingga dalam praktiknya sangat sulit untuk diwujudkan.

Daftar Pustaka

[1] Ogata, K. *Discrete Time Control Systems*. Upper Saddle River, NJ: Prentice Hall, 1995.
[2] Tavakoli, Mahdi, Hamid D. Taghirad, and Mehdi Abrishamchian. "Identification and robust H∞ control of the rotational/translational actuator system." *International Journal of Control, Automation, and Systems* 3.3 (2005): 387-396.
[3] Adlgostar, R., H. Azimian, and H. D. Taghirad. "Robust H∞, H 2/H∞ controller for rotational/translational actuator (RTAC)." *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*. IEEE, 2006.
[4] Rosales, Andres, et al. "Controller designed by means of numeric methods for a benchmark problem: RTAC (Rotational Translational Actuator)." *Electronics, Robotics and Automotive Mechanics Conference, 2006*. Vol. 1. IEEE, 2006.

Appendix A

Lampiran Program

Berikut lampiran program yang digunakan untuk simulasi sistem RTAC ini. Program dibuat menggunakan Matlab® dan tersedia secara online di https://github.com/reshalfahsi/rtacsystem.

```
clear all;
close all;

M = 1.3608;
m = 0.096;
R = 0.0592;
I = 2.175*10^-4;
k = 186.3;

%pembagi di persamaan
h = (M+m)*(I+0.5*m*R^2);

%nilai di matriks A
a11 = 0;
a12 = 1;
a13 = 0;
a14 = 0;
a21 = (-k*(I+m*R^2))/h;
a22 = 0;
a23 = -
(((m^2)*(R^3)*(0.555*((M+m)*I+(M*m*R^2))+m*(0.616*I+0.555*m*R^2)))/(h^2));
a24 = (1.57*I*m*R+1.11*(m^2)*(R^3))/h;

a31 = 0;
a32 = 0;
a33 = 0;
a34 = 1;

a41 = (0.707*m*R*k)/h;
a42 = 0;
a43 = 0;
a44 = -((0.785*(m^2)*(R^2))/h);

%nilai di matriks B
b11 = 0;
b12 = 0;
b21 = (I+m*R^2)/h;
b22 = (-0.707*m*R/h);
b31 = 0;
b32 = 0;
b41 = (-0.707*m*R/h);
b42 = (M+m)/h;
```

```
%matriks A,B,C, dan D
A = [a11 a12 a13 a14; a21 a22 a23 a24; a31 a32 a33 a34; a41 a42 a43 a44];
B = [0 0; b21 b22; 0 0; b41 b42];
C = eye(4);
D = zeros(4,2);

sys_ss = ss(A,B,C,D); % representasi state space

%Diskretisasi Sistem
fs = 250; %Hz
Ts = 1/fs; % periode sampling (sekon)

sys_ss_d = c2d(sys_ss, Ts)
[Ad, Bd, Cd, Dd] = ssdata(sys_ss_d)
Ob=obsv(Ad,Cd);
[n,m]=size(Ob);
unob = m-rank(Ob); %observable
if(unob==0)
   disp('Given System is Observable.');
else
   disp('Given System is Unobservable');
end

Co = ctrb(Ad,Bd);
[n,m]=size(Co);
unco=n-rank(Co); %controllable
if(unco==0)
   disp('Given System is Controllable.');
else
   disp('Given System is Uncontrollable');
end

%Mengecek kestabilan (melihat apakah pole berada di dalam unit circle)
pole = eig(Ad)
%Membuat stabil dengan menggunakan LQR
Qd=Cd'*Cd;
R=9999999999999999999*eye(2);
p=care(Ad,Bd,Qd,R);
K=inv(R)*Bd'*p
Anew=Ad-Bd*K; Bnew=Bd; Cnew=Cd; Dnew=Dd;
polenew=eig(Anew)
```