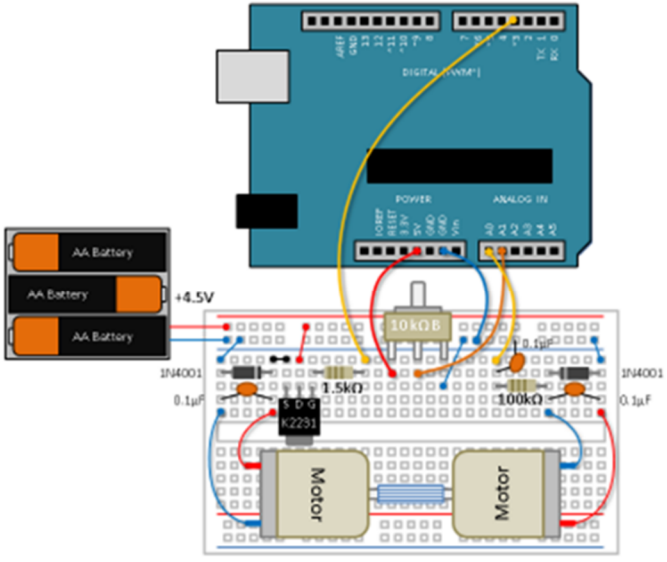
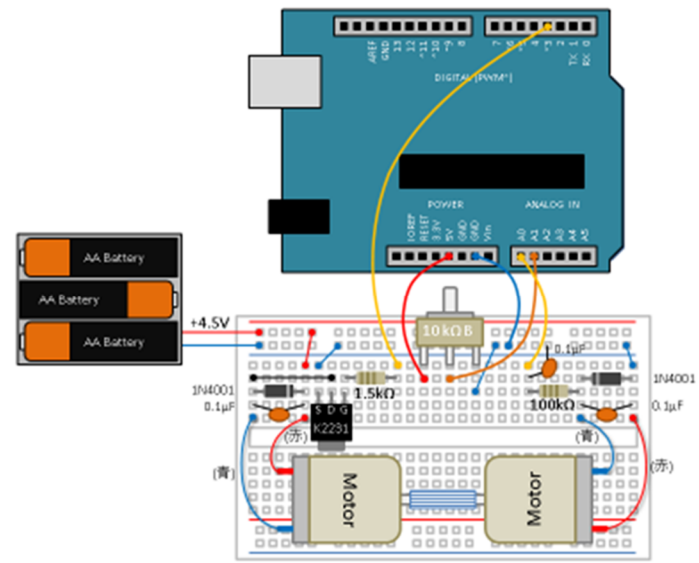


Arduino と MATLAB で制御系設計をはじめよう！ 「2012年11月30日 初版第1刷」改訂表

改訂箇所	改訂内容												
頁 v 目次	(旧)	1.5 本書で作成したソースコードについて											
	(新)	1.5 本書で作成したソースコードと補足情報について											
P006 1.4 本文	次の文章に書き改めました。												
	<p>本書執筆時の動作環境を表1.1に示します。MATLAB/Simulink のバージョンについて、RoTH を試す場合はR2012a 以降が必要です。また、Student Version での動作も確認しています。なお、ArduinoIOについては、バージョンアップにともなって古いバージョンは非公開になるようです。2014年3月現在の最新版は4.4 ですが、問題無く動作することを確認しています。</p> <p style="text-align: center;">表1.1 本書執筆時の作動環境</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tbody> <tr> <td>ホストPC のOS</td> <td>Windows 7 32bit</td> </tr> <tr> <td>MATLAB/Simulink のバージョン</td> <td>R2012a</td> </tr> <tr> <td>MATLAB/Simulink のツールボックス</td> <td>Control System Toolbox (必須)</td> </tr> <tr> <td>Arduino</td> <td>Uno R3 またはMega 2560 R3</td> </tr> <tr> <td>Arduino IDE のバージョン</td> <td>Version 1.01</td> </tr> <tr> <td>ArduinoIO のバージョン</td> <td>Version 4.2</td> </tr> </tbody> </table>		ホストPC のOS	Windows 7 32bit	MATLAB/Simulink のバージョン	R2012a	MATLAB/Simulink のツールボックス	Control System Toolbox (必須)	Arduino	Uno R3 またはMega 2560 R3	Arduino IDE のバージョン	Version 1.01	ArduinoIO のバージョン
ホストPC のOS	Windows 7 32bit												
MATLAB/Simulink のバージョン	R2012a												
MATLAB/Simulink のツールボックス	Control System Toolbox (必須)												
Arduino	Uno R3 またはMega 2560 R3												
Arduino IDE のバージョン	Version 1.01												
ArduinoIO のバージョン	Version 4.2												
P006 1.5 見出し	(旧)	本書で作成したソースコードについて											
	(新)	本書で作成したソースコードと補足情報について											
P006 1.5 本文	次の文章に書き改めました。												
	<p>本書で作成したMATLAB のm-file やSimulink モデルについては、下記のURL からダウンロードできます。また、最新のMATLAB/Simulink での動作状況といった補足情報も掲載していますので、是非ご参照ください。</p> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; text-align: center; margin: 10px auto; width: fit-content;"> http://books.techshare.co.jp/ </div>												

P006 1.6 本文	次の文章に書き改めました。	
	<p>本書で製作した実験装置は、街のホビーショップなどで一般的に入手できる部材を使っています。しかしながら、第5章で製作するモータ角度制御実験装置においてのみ、角度検出のポテンシオメータを取り付けるための専用プレートを使っています。アルミ板等を加工して製作できますが、加工が苦手な方のために、実験キットを用意しました。詳しくは下記Web サイトをご覧ください。</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> http://store.techshare.jp/ </div>	
P023 2.5.1 3行目	(誤)	本書執筆時点のバージョンは4.4です。
	(正)	本書執筆時点のバージョンは4.2です。
P025 2.5.3 4行目	(旧)	…次のスケッチを開きます。
	(新)	…次のスケッチを開きます ² 。
P025 補足欄	追加	² ArduinoIO のバージョンによって異なります。2014年3月現在、最新版(4.4)ではC:\Arduino\ArduinoIO\pde\adיותes\adיותes.pde になります。
P026 下から4行目	(旧)	…モータシールドに「Adafruit Motor/Stepper/Servo Shield」 ² が、…
	(新)	…モータシールドに「Adafruit Motor/Stepper/Servo Shield」 ³ が、…
P026 下から3行目	(旧)	…そのためのAFMotorライブラリ ³ を…
	(新)	…そのためのAFMotorライブラリ ⁴ を…
P026 補足欄	(旧)	² http://www.adafruit.com/products/81 ³ https://github.com/adafruit/Adafruit-Motor-Shield-library/zipball/master
	(新)	³ http://www.adafruit.com/products/81 ⁴ https://github.com/adafruit/Adafruit-Motor-Shield-library/zipball/master
P045 2行目	(誤)	a.digitalRead(0);
	(正)	a.analogRead(0);

P048 補足欄	(誤)	現在のバージョン (4.4) では、…
	(正)	本書執筆時のバージョン (4.2) では、…
P055 4.2 7行目	(旧)	また、実態配線図を図4.5に示します。
	(新)	また、モータとタコメータにはマブチモータ製FA-130RA-2270を使用しました。実態配線図を図4.5に示します。
P056 図 4.3	(誤)	2SK2231 とモータとの結線に誤りがありました。
	(誤)	
	(正)	

P057 図 4.5	2SK2231 とモータとの結線に誤りがありました。	
(誤)		
(正)		
P090 プログラム 4-4	⇒付属のソースコードをご参照下さい。	
P097 補足欄 下から3行目	(旧)	専用プレートを含む製作キットが TechShare Store から販売される予定です。
	(新)	専用プレートを含む実験キットがTechShare Store から販売されています。(http://store.techshare.jp/)

P106 下から2行目	(誤)	$\zeta=1$ の不足減衰では、…
	(正)	$\zeta=0.2$ の不足減衰では、…
P111 プログラム 5-3	⇒付属のソースコードをご参照下さい。	
P124 9行目	(誤)	これらの式を K_p, K_i について…
	(正)	これらの式を K_p, K_D について…
P126 1行目	(旧)	$\omega_n=10, \zeta=0.6$ に…
	(新)	一例として、 $\omega_n=10, \zeta=0.6$ に…
P170 プログラム 6-3	⇒付属のソースコードをご参照下さい。	
P175 プログラム 6-4	⇒付属のソースコードをご参照下さい。	
P176 プログラム 6-4	⇒付属のソースコードをご参照下さい。	
P190 プログラム 7-1	⇒付属のソースコードをご参照下さい。	
P199 プログラム 7-2	⇒付属のソースコードをご参照下さい。	
P200 プログラム 7-1	⇒付属のソースコードをご参照下さい。	

4.6 極指定法によるPIゲイン設計 (p.90)

プログラム 4-4 [velo_pi_mbd.m]

```
1 %% velo_pi_mbd.m
2
3 %% Initialize & load data
4 close all
5 clear all
6 load model_data
7
8 %% PI design by pole placement
9 p1 = input('p1 = ');
10 p2 = p1; % 重根の場合
11 % p2 = input('p2 = '); % p2 も指定する場合
12
13 Kp = -((p1+p2)*T + 1)/K;
14 Ki = p1*p2*T/K;
15
16 %% Display PI parameters
17 disp('>>> PI parameters <<<')
18 fprintf('Kp = %f\n',Kp);
19 fprintf('Ki = %f\n',Ki);
20
21 %% Open simulink model
22 open_system('velo_pi_mbd_sl');
23 open_system('velo_pi_mbd_sl/Output');
24
25 %% Experiment
26 ts = 1/50;
27 sim('velo_pi_mbd_sl')
28
29 %% EOF of velo_pi_mbd.m
```

■ 5.3.3 ステップ応答実験 (p.111)

プログラム 5-3 [pos_id_step.m]

```
1 %% pos_id_step.m
2
3 %% Initialize
4 close all
5 clear all
6 load sim_param
7
8 %% Parameters for identification
9 r      = 60;
10 r_cyc = 8;
11 Kp_id = 0.12;
12 Ncyc  = 2;
13 tfinal = r_cyc*Ncyc;
14
15 %% ID Experiment
16 open_system('pos_id_step_sl')
17 open_system('pos_id_step_sl/Scope')
18 sim('pos_id_step_sl')
19
20 %% Data processing
21 y = yout.signals(1).values(:,2);
22 t = yout.time;
23
24 NN = length(y);
25 N  = r_cyc/ts;
26 yy = reshape(y(2:NN),N,(NN-1)/N);
27 yf = yy(1:N/2,2:end); % 最初のデータを除き&前進データのみとする
28
29 % 平均化と正規化処理
30 ym = mean(yf)';
31 y0 = ym(1); yN = ym(end);
32 ym = (ym-y0)/(yN-y0);
33
34 %% Plot figure
35 t = (0:N/2-1)*ts;
36 figure(1)
37 subplot(211)
38 plot(t,yf), grid
39 xlabel('Time [s]'),ylabel('Output [deg]')
40 subplot(212)
41 plot(t,ym), grid
42 xlabel('Time [s]'),ylabel('Output [deg]')
43
44 %% EOF of pos_id_step.m
```

■ 6.4.2 I-PD+FF 制御のための Simulink モデル (p.170)

プログラム 6-3 [bb_pid_rc.m]

```
1 %% bb_pid_rc.m
2
3 %% Initialize & load data
4 close all
5 clear all
6 load sim_param
7
8 %% サーボ 1 度 あたりのアームの傾き
9 % RC サーボの場合
10 K_theta_rc = (pi/180)*(1.5/11.5); % [rad/deg]
11
12 %% ビーム傾き角 [rad] -> ボール位置 [cm] までの 1/s^2 のゲイン
13 K_b = (3/5*9.8)*100;
14
15 Pb = K_b/s^2;
16 Pbd = c2d(Pb,ts,'zoh');
17 [numbd,denbd] = tfdata(Pbd,'v');
18
19 %% PID パラメータ for Ball 位置制御
20 omega_n = 1.5;
21 zeta = 0.6;
22 alpha = 0.5;
23
24 p1 = (-zeta + j*sqrt(1-zeta^2))*omega_n;
25 p2 = (-zeta - j*sqrt(1-zeta^2))*omega_n;
26 p3 = -alpha;
27
28 kp = (p1*p2 + p2*p3 + p3*p1)/K_b;
29 ki = -p1*p2*p3/K_b;
30 kd = -(p1+p2+p3)/K_b;
31 % kff = 0;
32 kff = ki/alpha;
33
34 disp('>>> PID parameters for Ball and Beam <<<')
35 fprintf('kp = %f\n',kp);
36 fprintf('ki = %f\n',ki);
37 fprintf('kd = %f\n',kd);
38 fprintf('kff = %f\n',kff);
39
40 %% Reference for ball position
41 r1 = 20-5;
42 r2 = 20+5;
43
44 %% LPF カットオフ周波数
45 wf = 2*pi*2;
46
47 %% LPF for ball position sensor
48 Fc = wf^2/(s^2+2*0.7*wf*s+wf^2);
49 Fd = c2d(Fc,ts,'tustin');
50 [numlpf,denlpf] = tfdata(Fd,'v');
51
52 %% Open simulink model
53 open_system('bb_pid_rc_sl');
54 open_system('bb_pid_rc_sl/Ref angle')
55 open_system('bb_pid_rc_sl/Ball position and reference')
56
57 %% EOF of bb_pid_rc.m
```


■ 6.5.3 実験してみよう (p.175-176)

プログラム 6-4 [bb_pid_hg.m]

```
1 %% bb_pid_hg.m
2
3 %% Initialize & load data
4 close all
5 clear all
6 load sim_param
7
8 %% Set identified parameters
9 K = 300;
10 T = 0.28;
11
12 %% Discrete-time plant model
13 P = K/(T*s^2 + s);
14 Pd = c2d(P,ts,'zoh');
15 [numpd,denpd] = tfdata(Pd,'v');
16
17 %% PD パラメータ for 自作サーボ
18 omega_n_hg = 10;
19 zeta_hg = 0.65;
20 p1 = (-zeta_hg + j*sqrt(1-zeta_hg^2))*omega_n_hg;
21 p2 = (-zeta_hg - j*sqrt(1-zeta_hg^2))*omega_n_hg;
22
23 %% Set PD parameters
24 Kp = p1*p2*T/K;
25 Kd = -((p1+p2)*T + 1)/K;
26 Ki = 0;
27
28 %% Display PID parameters
29 disp('>>>> PID parameters for HG Servo <<<')
30 fprintf('Kp = %f\n',Kp);
31 fprintf('Ki = %f\n',Ki);
32 fprintf('Kd = %f\n',Kd);
33
34 %% サーボ 1度あたりのアームの傾き
35 % 自作サーボの場合
36 K_theta_hg = (pi/180)*(2.1/15); % [rad/deg]
37
38 %% ビーム傾き角 [rad] -> ボール位置 [cm] までの 1/s^2 のゲイン
39 K_b = (3/5*9.8)*100;
40
41 Pb = K_b/s^2;
42 Pbd = c2d(Pb,ts,'zoh');
43 [numbd,denbd] = tfdata(Pbd,'v');
44
45 %% PID パラメータ for Ball 位置制御
46 omega_n = 1.5;
47 zeta = 0.6;
48 alpha = 0.5;
49
50 p1 = (-zeta + j*sqrt(1-zeta^2))*omega_n;
51 p2 = (-zeta - j*sqrt(1-zeta^2))*omega_n;
52 p3 = -alpha;
53
54 kp = (p1*p2 + p2*p3 + p3*p1)/K_b;
55 ki = -p1*p2*p3/K_b;
56 kd = -(p1+p2+p3)/K_b;
57 % kff = 0;
58 kff = ki/alpha;
59
60 disp('>>>> PID parameters for Ball and Beam <<<')
61 fprintf('kp = %f\n',kp);
62 fprintf('ki = %f\n',ki);
63 fprintf('kd = %f\n',kd);
64 fprintf('kff = %f\n',kff);
65
66 %% Reference for ball position
67 r1 = 20-5;
68 r2 = 20+5;
69
70 %% LPF カットオフ周波数
71 wf = 2*pi*5;
72
73 %% LPF for ball position sensor
```

```
74 Fc = wf^2/(s^2+2*0.7*wf*s+wf^2);
75 Fd = c2d(Fc,ts,'tustin');
76 [numlpf,denlpf] = tfdata(Fd,'v');
77
78 %% Open simulink model
79 open_system('bb_pid_hg_sl');
80 open_system('bb_pid_hg_sl/Servo angle')
81 open_system('bb_pid_hg_sl/Ball position and reference')
82
83 %% EOF of bb_pid_hg.m
```

7.3.2 最適状態フィードバックベクトルの設計 (p.190)

プログラム 7-1 [bb_lqr_test.m]

```
1 %% bb_lqr_test.m
2
3 %% Initialize & load data
4 close all
5 clear all
6 load sim_param
7
8 %% Set identified parameters
9 K = 300;
10 T = 0.28;
11
12 %% ビーム傾き角 [rad] -> ボール位置 [cm] までの  $1/s^2$  のゲイン
13 K_b = (3/5*9.8)*100;
14
15 %% サーボ 1 度 あたりのアームの傾き
16 % 自作サーボの場合
17 K_theta_hg = (pi/180)*(2.1/15); % [rad/deg]
18
19 %% 状態方程式
20 A = [ 0  1  0          0  ;
21       0  0  K_b*K_theta_hg  0  ;
22       0  0  0              1  ;
23       0  0  0              -1/T ];
24 B = [ 0  ;
25       0  ;
26       0  ;
27       K/T ];
28 C = [ 1  0  0  0 ];
29 D = 0;
30 Pbb = ss(A,B,C,D);
31
32 %% LQ design 1
33 Q1 = diag([100 20 20 1]);
34 R1 = 10000;
35 F1 = lqr(Pbb,Q1,R1);
36
37 %% LQ design 2
38 Q2 = diag([500 20 20 1]);
39 R2 = 10000;
40 F2 = lqr(Pbb,Q2,R2);
41
42 %% Simulation
43 t = 0:ts:5;
44 x0 = [10,0,0,0]'; % Set initial state
45
46 % Closed-loop system
47 Pc11 = ss(A-B*F1,zeros(4,1),C,D);
48 Pc12 = ss(A-B*F2,zeros(4,1),C,D);
49
50 % Initial state response
51 [y1,tt1,X1] = initial(Pc11,x0,t);
52 [y2,tt2,X2] = initial(Pc12,x0,t);
53
54 %% Plot figure
55 figure(1)
56 subplot(211)
57 plot(t,X1(:,1:2),t,X2(:,1:2),'--')
58 xlabel('Time [s]'), ylabel('Ball position and velocity')
59 legend('Design 1 x','Design 1 d/dt x','Design 2 x','Design 2 d/dt x')
60 subplot(212)
61 plot(t,X1(:,3:4),t,X2(:,3:4),'--')
62 xlabel('Time [s]'), ylabel('Servo angle and angular velocity')
63 legend('Design 1 \phi','Design 1 d/dt \phi','Design 2 \phi','Design 2 d/dt \phi')
64
65 %% EOF of bb_lqr_test.m
```

7.4.4 実験してみよう (p.198-200)

プログラム 7-2 [bb_lqr_servo_hg.m]

```
1 %% bb_lqr_servo_hg.m
2
3 %% Initialize & load data
4 close all
5 clear all
6 load sim_param
7
8 %% Set identified parameters
9 K = 300;
10 T = 0.28;
11
12 %% Discrete-time plant model
13 P = K/(T*s^2 + s);
14 Pd = c2d(P,ts,'zoh');
15 [numpd,denpd] = tfdata(Pd,'v');
16
17 %% ビーム傾き角 [rad] -> ボール位置 [cm] までの 1/s^2 のゲイン
18 K_b = (3/5*9.8)*100;
19
20 Pb = K_b/s^2;
21 Pbd = c2d(Pb,ts,'zoh');
22 [numbd,denbd] = tfdata(Pbd,'v');
23
24 %% サーボ 1 度 あたりのアームの傾き
25 % 自作サーボの場合
26 K_theta_hg = (pi/180)*(2.1/15); % [rad/deg]
27
28 %% 状態方程式
29 A = [ 0 1 0 0 ;
30       0 0 K_b*K_theta_hg 0 ;
31       0 0 0 1 ;
32       0 0 0 -1/T ];
33 B = [ 0 ;
34       0 ;
35       0 ;
36       K/T ];
37 C = [ 1 0 0 0 ];
38 D = 0;
39 Pbb = ss(A,B,C,D);
40
41 %% LQI design
42 Qe = diag([100 20 20 1]);
43 ru = 10000;
44 rv = 1000;
45
46 % Error system
47 Abar = [A, B ;
48         zeros(1,4), 0 ];
49 Bbar = [ zeros(4,1) ;
50         1 ];
51 Pbar = ss(Abar,Bbar,[],[]);
52
53 Q = [ Qe, zeros(4,1) ;
54       zeros(1,4), ru ];
55 Fbar = lqr(Pbar,Q,rv);
56 E = [A, B;
57       C, zeros(1,1) ];
58 Ftmp = Fbar*inv(E);
59 F = Ftmp(:,1:4);
60 Ki = Ftmp(:,5);
61 Faug = [F, -Ki];
62
63 %% 閉ループ系の構成とステップ応答
64 % 拡大システム
65 Aaug = [ A, zeros(4,1) ;
66         -C, 0 ];
67 Baug = [ B ;
68         0 ];
69 Caug = [ C, 0 ];
70 Daug = [ 0 ];
71 Br = [ zeros(4,1) ;
72        1 ]; % 目標値に対する入力行列
73
```

```

74 % 閉ループシステム
75 Paug = ss(Aaug-Baug*Faug,Br,Caug,Daug);
76
77 % ステップ応答計算
78 t=0:ts:5;
79 [y,tt,x] = step(Paug,t);
80
81 %% Plot figure
82 figure(1)
83 subplot(211)
84 plot(t,x(:,1:2))
85 xlabel('Time [s]'), ylabel('Ball position and velocity')
86 legend('x','d/dt x')
87 subplot(212)
88 plot(t,x(:,3:4))
89 xlabel('Time [s]'), ylabel('Servo angle and angular velocity')
90 legend('\phi','d/dt \phi')
91
92 %% Reference for ball position
93 r1 = 20-5;
94 r2 = 20+5;
95
96 %% LPF カットオフ周波数
97 wf = 2*pi*5;
98
99 %% LPF for ball position sensor
100 Fc = wf^2/(s^2+2*0.7*wf*s+wf^2);
101 Fd = c2d(Fc,ts,'tustin');
102 [numlpf,denlpf] = tfdata(Fd,'v');
103
104 %% Open simulink model
105 open_system('bb_lqr_servo_hg_sl');
106 open_system('bb_lqr_servo_hg_sl/Ball position and motor angle')
107
108 %% EOF of bb_lqr_servo_hg.m

```