L. Odilon Petra I.
Tugas minggu 6

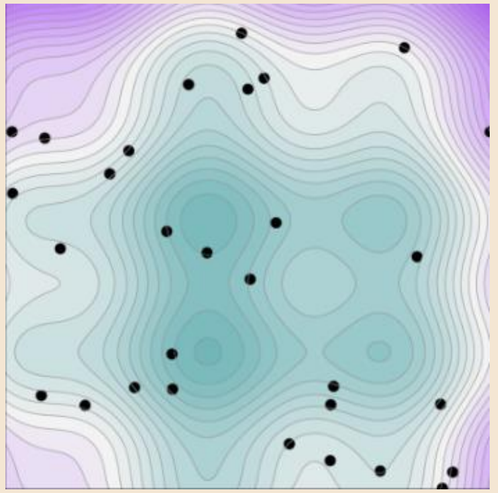
Link Github: https://github.com/dilonpetra/Sistem-Kendali.git

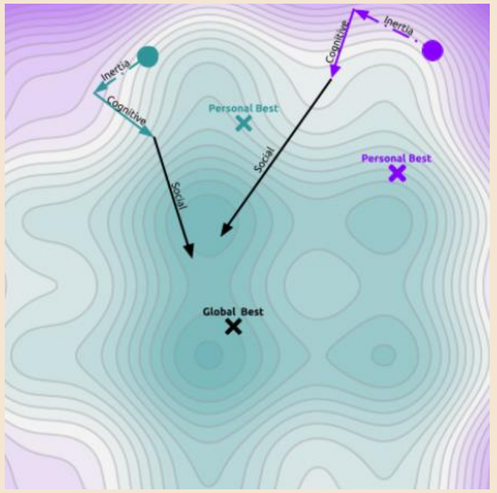
Membandingkan metode grid search dengan PSO pada proses tuning pid motor DC

Tuning PID pada Motor DC Menggunakan Metode PSO dan Grid Search

Arina Muthmainnah	19/442365/PA/19114
Ferio Astika Mahatma	19/442373/PA/19122
lukas Odilon Petra	19/442378/PA/19127
Novelio Putra Indarto	19/442386/PA/19135
Raihan Fadhil Maftuh	19/439114/PA/18937

Particle Swarm Optimization





PSO

Langkah-langkah

- Membuat script pada aplikasi MATLAB
- Mendefinisikan parameter yang digunakan untuk membuat trasnfer function
- Membuat transfer function
- Menentukan jumlah iterasi
- Menentukan nilai random untuk mencari parameter terbaik, dengan matriks 3*N
- Merumuskan fitness function
- Tuning parameter PID dengan nilai random terbaik, ditentukan oleh fitness function
- Menggunakan hasil PID yang didapatkan

PSO

Kode Program

```
close all; clear all; clc;
```

```
%% System
J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;

s = tf('s');
sys = K/((J*s+b)*(L*s+R)+K^2)
```

```
%% PSO parameters
N = 10;
max_iter = 40;

X = rand(3,N);
V = rand(3,N);

pbest = X;
pbest_obj = fitness_function(sys, X, N);
[~, idx_best] = min(pbest_obj);
gbest = pbest(:, idx_best);
gbest_obj = min(pbest_obj);
```

```
%% Initial control system
KP = gbest(1)
KI = gbest(2)
KD = gbest(3)

cont = pid(KP, KI, KD);
final = feedback(cont*sys, 1);

figure
step(final);
stepinfo(final)
```

```
%% PSO update
for iter = 1:max_iter
    r1 = rand(3);
    r2 = rand(3);

    w = 0.4 * iter/(max_iter^2) + 0.4;
    c1 = -3 * iter/max_iter + 3.5;
    c2 = 3 * iter/max_iter + 0.5;

    V = w*V + c1*r1*(pbest-X) + c2*r2*(gbest-X);
    X = X + V;
```

PSO

Kode Program

```
    obj = fitness_function(sys, X, N);
    pbest(:, (pbest_obj >= obj)) = X(:,(pbest_obj >= obj));
    pbest_obj = min(pbest_obj, obj);

    [~, idx_best] = min(pbest_obj);
    gbest = pbest(:, idx_best);
    gbest_obj = min(pbest_obj);

    fprintf("Best Global at Iteration %i : %.20f \n", iter, gbest_obj);
end
```

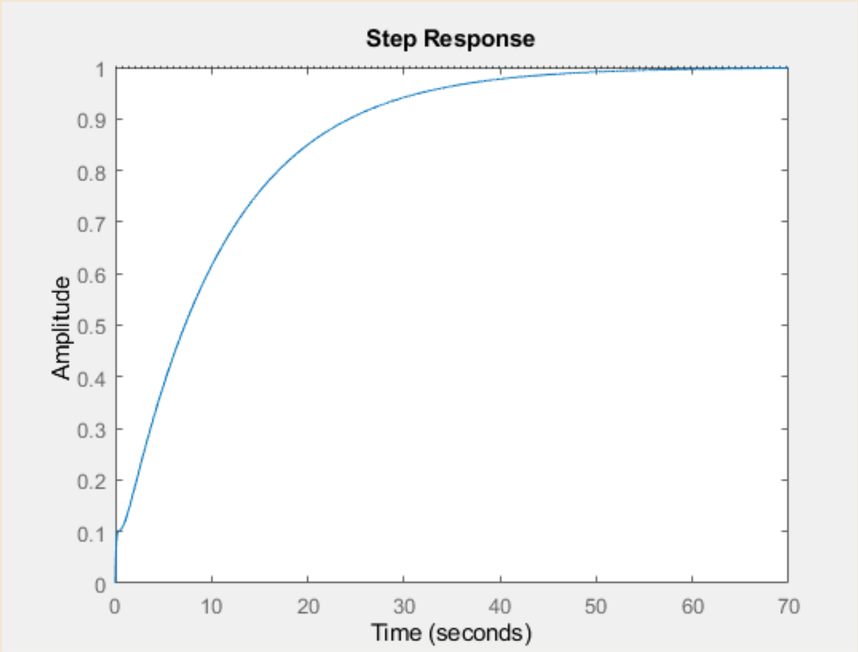
```
%% Final result
KP = gbest(1)
KI = gbest(2)
KD = gbest(3)

cont = pid(KP, KI, KD);
final = feedback(cont*sys, 1);

figure
step(final);
stepinfo(final)
```

```
%% Fitness function based on rise time, overshoot, and settling time
function f = fitness_function(sys, K, N)
    for i = 1:N
        c = pid(K(1,i), K(2,i), K(3,i));
        fb = feedback(c*sys, 1);
        si = stepinfo(fb);
        f(i) = si.RiseTime + si.Overshoot + si.SettlingTime;
    end
end
```

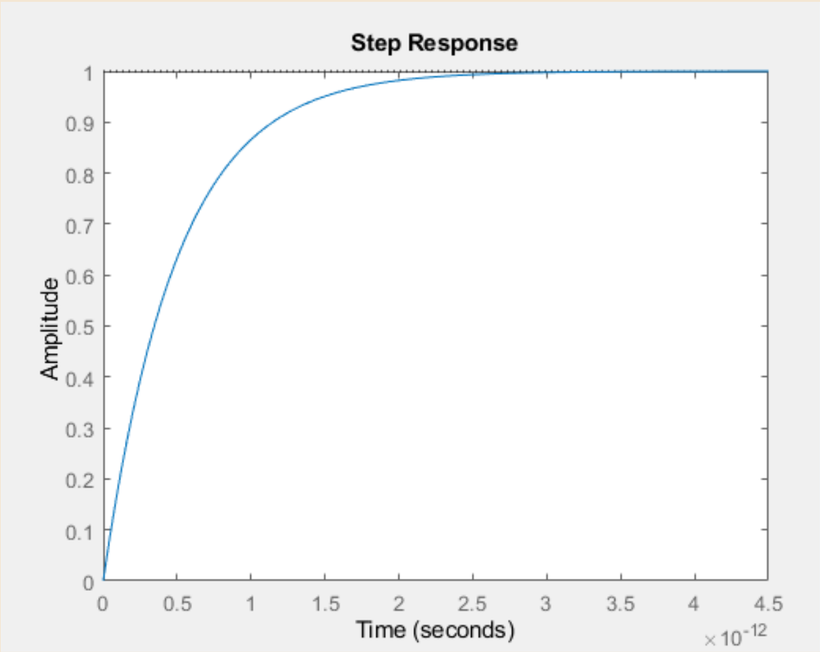
PSO

Grafik step response

Sebelum Tuning



Sesudah Tuning



PSO

Nilai Hasil

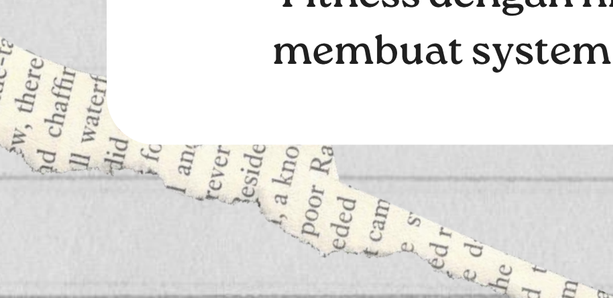
Kp	Ki	Kd	Rise time	Settling time	Overshoot
5.7916e + 12	6.5294e + 12	8.1926e + 12	1.3409e - 13	2.3876e - 13	0

Grid Search

Grid Search

Langkah-langkah

- Membuat script pada aplikasi MATLAB
- Mendefinisikan parameter yang digunakan untuk membuat trasnfer function
- Membuat transfer function
- Menginisialisasi Kp =0.1, KI = 0, KD= 0, fitness =1000
- Membuat sistem dengan PID dan dilakukan proses feedback
- Menginisialisasi grid search parameter
- Melakukan proses increment nilai KP, KI dan KD di dalam looping
- Pada proses looping, akan dihitung nilai fitness terbaru berdasarkan perhitungan parameter stepinfo dikalikan dengan weighnya dan dijumlahkan semuanya
- Hasil perhitungan fitness terbaru akan dibandingkan dengan nilai fitness , jika fitness terbaru lebih kecil dari fitness, maka fitness terbaru akan menjadi nilai dari fitness beserta dengan nilai KP, KI dan KDnya
- Fitness dengan nilai tertinggi akan digunakan parameter Kp, KI dan KDnya untuk digunakan dalam membuat system PID



Grid Search

Kode Program

```
close all; clear all; clc;

J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;

s = tf('s');
sys = K/((J*s+b)*(L*s+R)+K^2)

% initial value
KP = 0.1;
KI = 0;
KD = 0;

cont = pid(KP, KI, KD);
final = feedback(cont*sys, 1);
info = stepinfo(final);
```

```
figure (3)
step(final);

fitness = 1000;

% griid search parameter
max_kp = 3;
max_ki = 3;
max_kd = 3;
increament = 0.1;
weight_risetime = 0.6;
weight_overshoot = 0.3;
weight_settlingtime = 0.1;
```

Grid Search

Kode Program

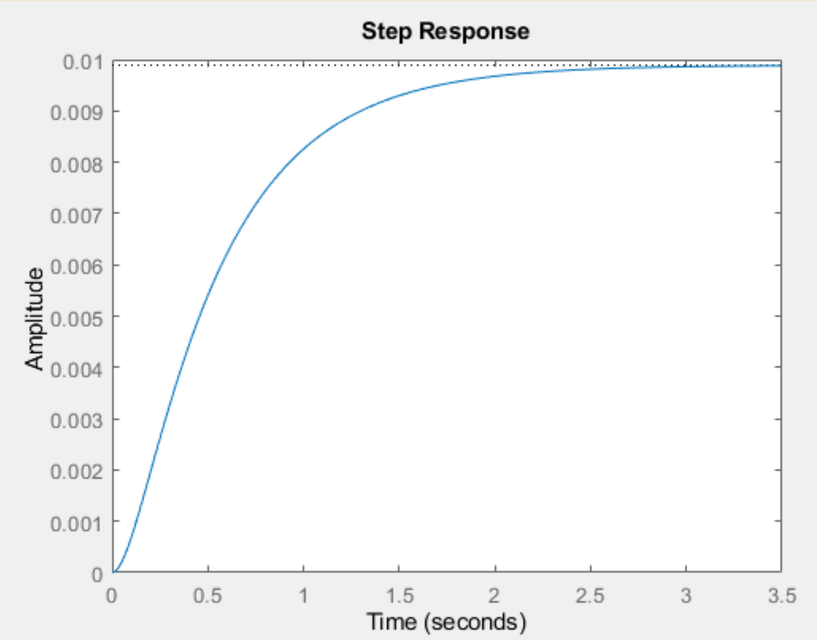
```
for kp = KP : increament : max_kp
    for ki = KI : increament : max_ki
        for kd = KD : increament : max_kd
            cont = pid(kp, ki, kd);
            final = feedback(cont*sys, 1);
            info = stepinfo(final);

            % calculate fitness function
            fitness_new = weight_risetime*info.RiseTime + weight_overshoot*info.Overshoot
            + weight_settlingtime*info.SettlingTime;

            fprintf("kp:%.1f ki:%.1f kd:%.1f rise:%.2f ovs:%.2f set:%.2f fitness:%f\n",
            kp, ki, kd, info.RiseTime, info.Overshoot, info.SettlingTime, fitness);

            % determine is new fitness is better than before
            if fitness > fitness_new
                fitness = fitness_new;
                KP = kp;
                KI = ki;
                KD = kd;
            end
        end
    end
end
```

```
cont = pid(KP, KI, KD);
final = feedback(cont*sys, 1);

figure (4)
step(final);
stepinfo(final);
```

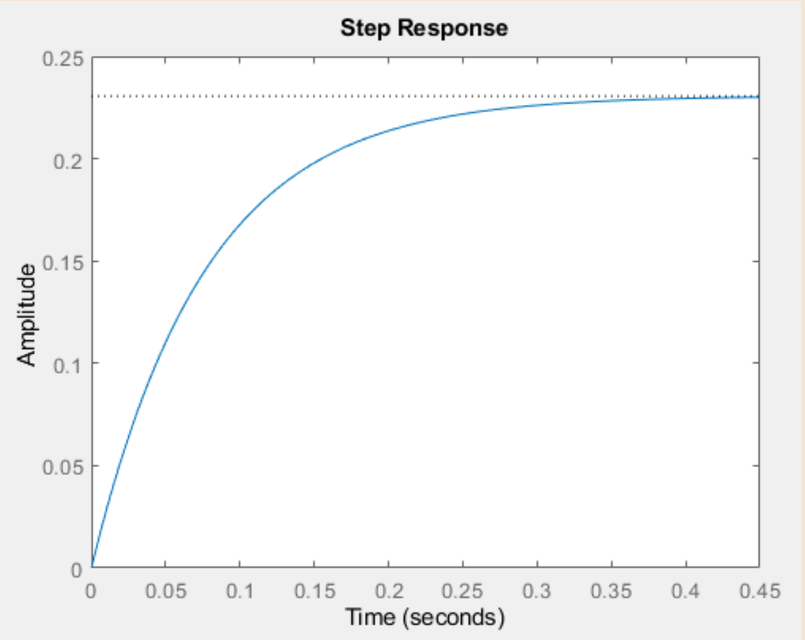
Grid Search

Grafik step response

Sebelum Tuning



Sesudah Tuning



Grid Search

Nilai Hasil

Kp	Ki	Kd	Rise time	Settling time	Overshoot	Fitness
3	3	3	8.17	13.81	0	0.131

Terima kasih.