

Московский Государственный Университет
им. М. В. Ломоносова

Физический факультет

Кафедра физико-математических методов управления

Отчет по физическому практикуму
Исследование системы управления рукой робота

Работу выполнил:
студент 442 группы
Пауль Эдуард Вячеславович

Преподаватели:
Митришкин Юрий Владимирович
Коньков Артем Евгеньевич

Москва 2020

Содержание

1	Введение	3
2	Постановка задачи	4
3	Анализ линейной модели объекта	6
4	Синтез регулятора обратной связи	8
5	Вывод	10
6	Приложение	11
6.1	Matlab программа	11
6.2	Simulink схема	15

1 Введение

Роботизированная рука - механизм, который обладает функциями, аналогичными человеческой руке. С ее помощью можно управлять положением предметов в пространстве. Зачастую робо-рука является составной частью более сложного механизма. Имеет широкое применение в промышленности и космической инженерии.

В данной работе проведено исследование управления положением руки робота в средах разработки MATLAB и Simulink. Проанализированы системы управления, на основе ПД- и ПИД-регуляторов. Получены выводы об эффективности соответствующих систем управления.

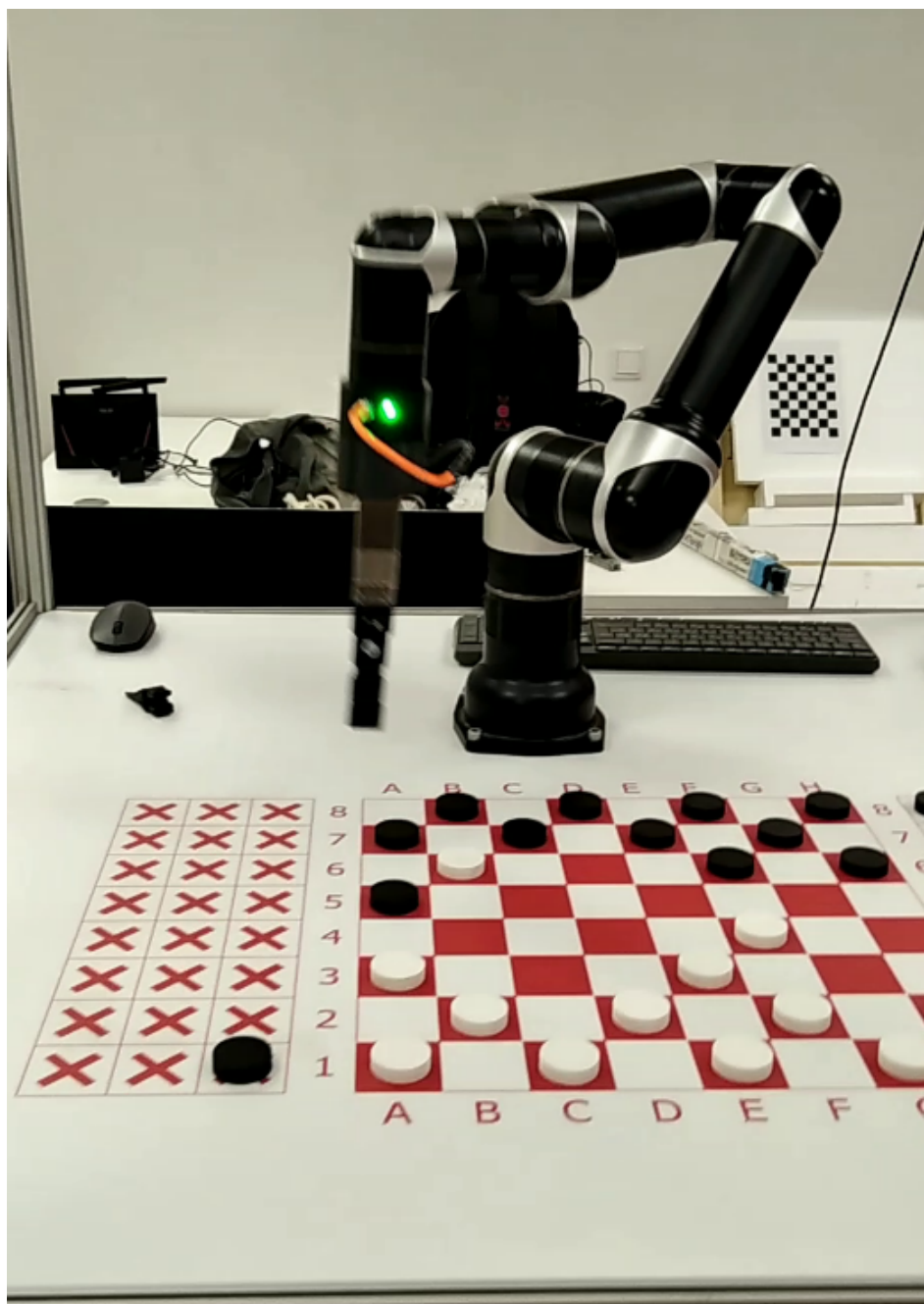


Рис. 1: Пример использования руки робота

2 Постановка задачи

Схематично руку робота можно представить следующим образом.

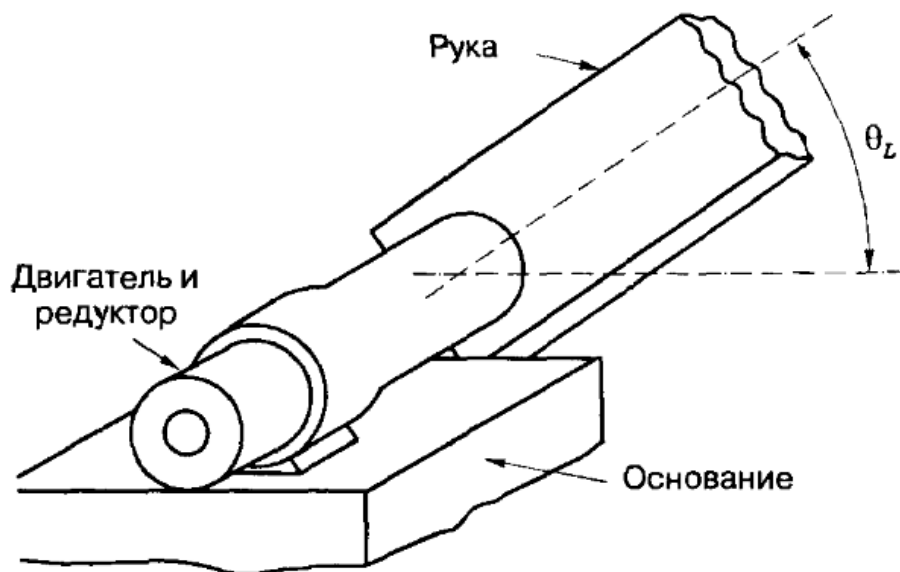


Рис. 2: Схема устройства руки робота

Исполнительным устройством является двигатель постоянного тока, управляемый по цепи якоря. Предполагается, что двигатель соединен с рукой посредством редуктора.

Структурная схема системы управления показана на рисунке

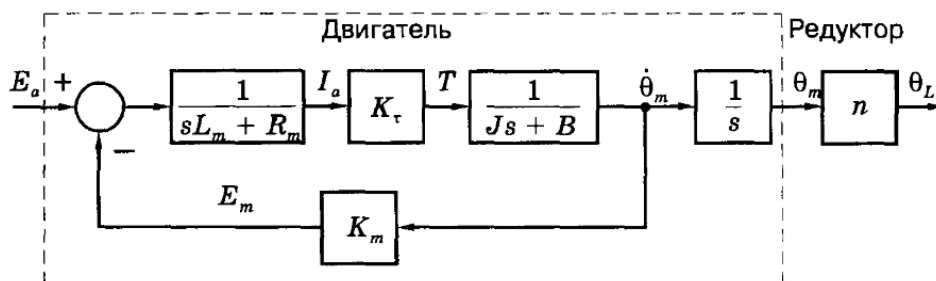


Рис. 3: Структурная схема системы управления рукой робота

Здесь E_a - входное напряжение; I_a - ток в цепи якоря; T - развиваемый момент; J - сумма моментов инерции, приведенных в валу двигателя; B - коэффициент, определяемый силами трения, приведенными к валу двигателя; L_m - индуктивность цепи якоря; R_m - сопротивление цепи якоря; θ_m - угол поворота вала двигателя; θ_L - угол поворота руки робота.

Значения параметров нашей системы возьмем из примера, который приводится в литературе¹:

$$L_m = 2; R_m = 21; K_\tau = 18; J = 2; B = 1; K_m = 0.5; n = 30$$

¹Филлипс Ч., Харбор Р. Системы управления с обратной связью.

Система управления с обратной связью изображена на рис.4. Здесь θ_c - желаемое положение руки робота.

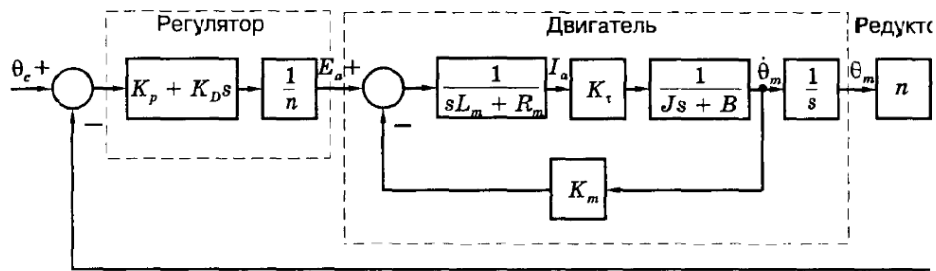


Рис. 4: Системы управления рукой робота с обратной связью

На данной схеме фигурирует ПД-регулятор. Необходимо сравнить предложенный ПД-регулятор с системами управления, использующими ПИД-регуляторы.

3 Анализ линейной модели объекта

Проследим, как ведет себя система без регулирования.

Передаточную функцию электродвигателя получим, воспользовавшись формулой Мейсона:

$$G(s) = \frac{G_1(s)G_2(s)K_\tau}{1 + K_\tau G_1(s)G_2(s)H(s)} \quad (1)$$

Применительно к нашей задаче:

$$G_1(s) = \frac{1}{sL_m + R_m} = \frac{1}{2s + 21};$$

$$G_2(s) = \frac{1}{Js + B} = \frac{1}{2s + 1};$$

$$K_\tau = 18;$$

$$H(s) = K_m = 0.5;$$

В итоге приходим к следующему выражению для $G(s)$:

$$G(s) = \frac{18}{(2s + 21)(2s + 1) + 9} \quad (2)$$

Выясним, является ли система устойчивой. Рассмотрим положение полюсов передаточной функции. Из рис.5 видно, что все полюса передаточной функ-

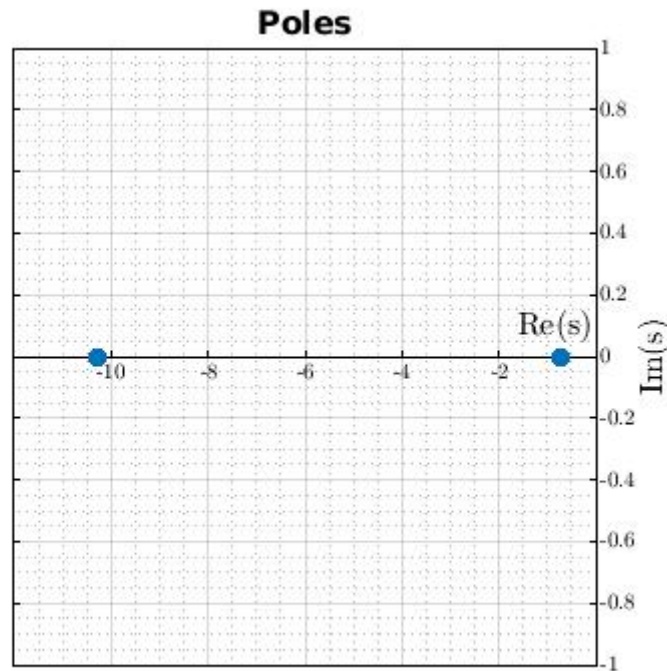


Рис. 5: Положение полюсов передаточной функции

ции располагаются на левой полуплоскости. Из чего можно сделать вывод, что система устойчива.

Также любопытный читатель может проверить выполнение критерия Калмана нашей системы. Критерий Калмана выполняется, следовательно наша система является управляемой и наблюдаемой.

Проанализируем отклик системы с данной передаточной функцией на единичный сигнал. Из графика видно, что система не обнаруживает перерегули-

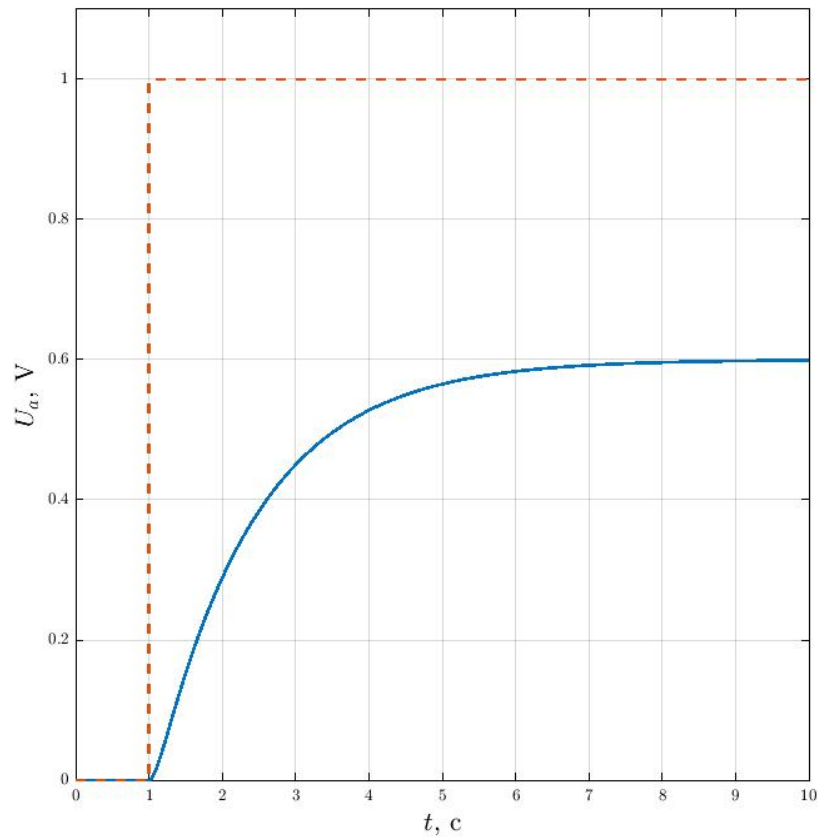


Рис. 6: Отклик линейной системы на единичную ступеньку

рования, но сигнал на выходе составляет лишь 60% от желаемого. Поэтому для корректной работы данной системы требуется разработать регулятор обратной связи.

4 Синтез регулятора обратной связи

В данном разделе рассматривается синтез пропорционально-интегрально-дифференциальных регуляторов. ПИД-регулятор - это наиболее общий тип регуляторов, используемых в системах управления.

Если $e(t)$ - входной сигнал регулятора, а $m(t)$ - его выходной сигнал, то связь между ними определяется уравнением:

$$m(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (3)$$

Преобразование Лапласа от этого уравнения позволяет найти передаточную функцию:

$$M(s) = (K_p + \frac{K_I}{s} + K_D s) E(s) \quad (4)$$

Откуда:

$$G(s) = \frac{M(s)}{E(s)} = K_p + \frac{K_I}{s} + K_D s \quad (5)$$

В литературе², посвященной данной теме, рекомендуется использовать ПД-регулятор. Для этого достаточно положить $K_I = 0$ и подобрать коэффициенты K_p и K_D .

Проведем сравнение ПД-регулятора с коэффициентами $K_p = 3.2$ и $K_D = 4.2$ с регулятором, настроенным с помощью PID-Tuner, который реализован в среде Simulink ($K_p = 2.53$; $K_I = 0.3$; $K_D = 4.03$).

Отклик систем на единичную ступеньку показан на графике:

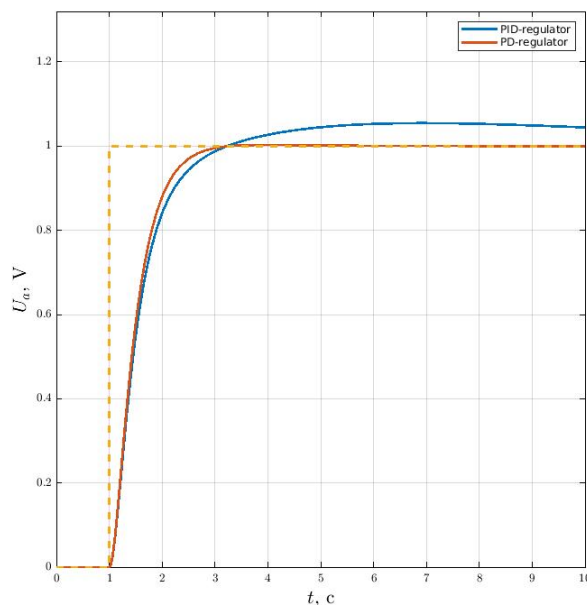


Рис. 7: Отклик систем с регуляторами на единичную ступеньку

²Филлипс Ч., Харбор Р. Системы управления с обратной связью.

Заметны следующие отличия системы с ПИД-регулятором и с ПД-регулятором:

	ПИД-регулятор	ПД-регулятор
Перерегулирование, %	4	0
Время переходного процесса, с	>8	1.5
Отклонение от желаемого сигнала	нет	нет

Видно, что ПД-регулятор, предложенный в литературе, более эффективно справляется с задачей установления и поддержания сигнала, поданного на вход системе.

5 Вывод

В ходе данной работы была исследована система управления рукой робота. Было установлено, что для корректной работы данной системы требуется регулятор обратной связи. Произведен синтез ПД-регулятора и сравнение его с ПИД-регулятором, предложенным ПО PID-Tuner. Из сравнения этих двух регуляторов можно сделать вывод, что ПД-регулятор эффективно справляется с поставленной задачей.

6 Приложение

6.1 Matlab программа

Реализация модели

```
clear all;
close all;
clc;

%% Задание системы
s = tf('s'); % Ввод передаточной функции (transfer function)
H = 18 / ((2*s + 21)*(2*s + 1) + 9); %Передаточная функция
Ha = rade(H); % линейная аппроксимация
G_lin = ss(Ha); % объект statespace
G1 = ss(H);

%% Проверка наблюдаемости и управляемости системы
check_kalman(G1);

%% Вывод графика полюсов системы
param.fileName = strcat('poles_',datestr(datetime('now'),
'yyyy-mm-dd_HH-MM-SS'));

param.fileFormat = "pdf";
param.pictUnit = 'centimeters';
param.pictSize = [10 10];
param.tickFontSize = 10;
param.mainFontSize = 16;
param.MarkerSize = 25;

print_poles(G1, param)

%% Вывод графиков из Simulink модели
open('manipulator.slx');

sim_data = sim('manipulator.slx');

param.fileName = strcat('sim_', datestr(datetime('now'),
'yyyy-mm-dd_HH-MM-SS'));

param.pictUnit = 'centimeters';
param.pictSize = [20 20];
```

```

param.tickFontSize = 10;
param.mainFontSize = 16;
param.LineWidth = 2;
param.Ylabel = '$U_a$, V';

```

```

print_sim(sim_data.tout,[sim_data.Y1.data, sim_data.Y2.data,
sim_data.Yref.data],param)

```

Функция для проверки критерия Калмана

```

function [rControl rObserve] = check_kalman(H)

A = H.A;
B = H.B;
C = H.C;

n = size(A,1);

controlMatrix = [];
observeMatrix = [];
for i = 1:n
controlMatrix = [controlMatrix A^(i-1)*B];
observeMatrix = [observeMatrix; C*A^(i-1)];
end

rControl = rank(controlMatrix);
rObserve = rank(observeMatrix);
if rControl == n
disp(["The system is controllable", "rank:",num2str(rControl)]);
else
error(["The system is not controllable", "rank:",num2str(rControl)]);
end
if rObserve == n
disp(["The system is observable", "rank:",num2str(rObserve)]);
else
error(["The system is not observable", "rank:",num2str(rObserve)]);
end
end

```

Функция для вывода графика полюсов

```

function print_poles(G, param)
A = G.A;
poles = complex(eig(A));
%%
fig = figure;

```

```

ax = axes;
ax.Parent = fig;
p = plot(ax, poles);
%%
fig.Name = 'Poles';
fig.Color = 'white';
fig.Units = param.pictUnit;
fig.PaperUnits = param.pictUnit;
fig.Position=[fig.OuterPosition(1) fig.OuterPosition(2) param.pictSize];
fig.PaperPositionMode='manual';
fig.PaperPosition=[fig.OuterPosition(1) fig.OuterPosition(2) param.pictSiz
fig.PaperSize=param.pictSize;
%%
ax.XAxisLocation = 'origin';
ax.YAxisLocation = 'origin';
ax.Box = 'on';
ax.XColor = 'black';
ax.YColor = 'black';
ax.XGrid = 'on';
ax.YGrid = 'on';
ax.XMinorGrid = 'on';
ax.YMinorGrid = 'on';
ax.GridColor='black';
ax.XLabel.String = 'Re(s)';
ax.YLabel.String = 'Im(s)';
ax.FontName = 'Times New Roman';
ax.XLabel.FontName = 'Times New Roman';
ax.YLabel.FontName = 'Times New Roman';
ax.FontSize = param.tickFontSize;
ax.XLabel.FontSize = param.mainFontSize;
ax.YLabel.FontSize = param.mainFontSize;
set(ax,'TickLabelInterpreter', 'latex')
ax.YLabel.Interpreter = 'latex';
ax.XLabel.Interpreter = 'latex';
ax.FontWeight = 'normal';
ax.Title.String = 'Poles';
ax.Title.FontSize = param.mainFontSize;
%%
p.LineStyle= 'none';
p.Marker = '.';
p.MarkerSize = param.MarkerSize;
%%
switch param.fileFormat
    case "png"

```

```

        param.fileFormatKey = '-dpng';
    case "eps"
        param.fileFormatKey = '-deps';
    case "pdf"
        param.fileFormatKey = '-dpdf';
    otherwise
        param.fileFormatKey = '-dpdf';
end
print(fig,param.fileName,'-painters','-r300','-fillpage',param.fileFormatKey)
end

```

Функция для вывода графиков из simulink-схемы

```

function print_sim(x,y,param)
%%
fig = figure;
ax = axes;
ax.Parent = fig;
p = plot(ax, x, y);
%%
fig.Name='Simulation results';
fig.Color='white';
fig.Units = param.pictUnit;
fig.PaperUnits = param.pictUnit;
fig.Position=[fig.OuterPosition(1) fig.OuterPosition(2) param.pictSize];
fig.PaperPositionMode='manual';
fig.PaperPosition=[fig.OuterPosition(1) fig.OuterPosition(2) param.pictSize];
fig.PaperSize=param.pictSize;
%%
ax.XAxisLocation = 'bottom';
ax.YAxisLocation = 'left';
ax.Box = 'on';
ax.XColor = 'black';
ax.YColor = 'black';
ax.XGrid = 'on';
ax.YGrid = 'on';
ax.GridColor='black';
ax.YLim = ax.YLim * 1.1;
ax.XLabel.String = '$t$, c';
ax.YLabel.String = param.Ylabel;
ax.FontName = 'Times';
ax.XLabel.FontName = 'Times';
ax.YLabel.FontName = 'Times';
ax.FontSize = param.tickFontSize;
ax.XLabel.FontSize = param.mainFontSize;

```

```

ax.YLabel.FontSize = param.mainFontSize;
set(ax,'TickLabelInterpreter', 'latex')
ax.YLabel.Interpreter = 'latex';
ax.XLabel.Interpreter = 'latex';
ax.FontWeight = 'normal';
%%
for i = 1:size(p,1)
    p(i).LineWidth = param.LineWidth;
end
p(end).LineStyle = '--';
%%
legend('PID-regulator', 'PD-regulator');
print(fig,param.fileName,'-painters','-r300','-fillpage','-dpdf')
end

```

6.2 Simulink схема

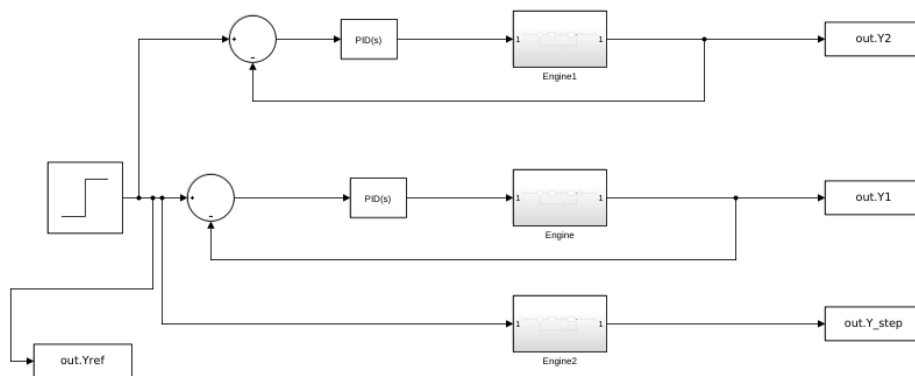


Рис. 8: Общая Simulink-схема

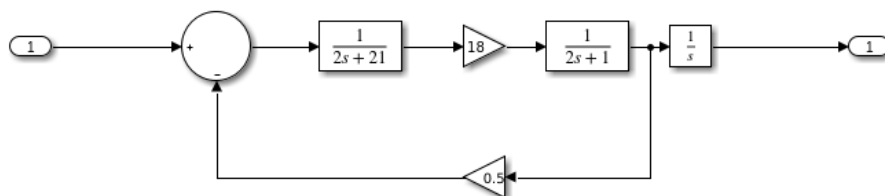


Рис. 9: Simulink-схема двигателя