



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет: «Специальное машиностроение»

Кафедра: «Робототехнические системы и мехатроника»

Лабораторная работа №4

по курсу «Теория автоматического управления»

Вариант 4

Выполнил: Давыдов В.Ю.

Группа: СМ7-62Б

Проверил (-а):

Москва, 2024г.

1. Моделирование цифрового-аналогового преобразователя

Будем считать, что процессор 8-ми разрядный. На вход процессора будем подавать синусоиду с параметрами

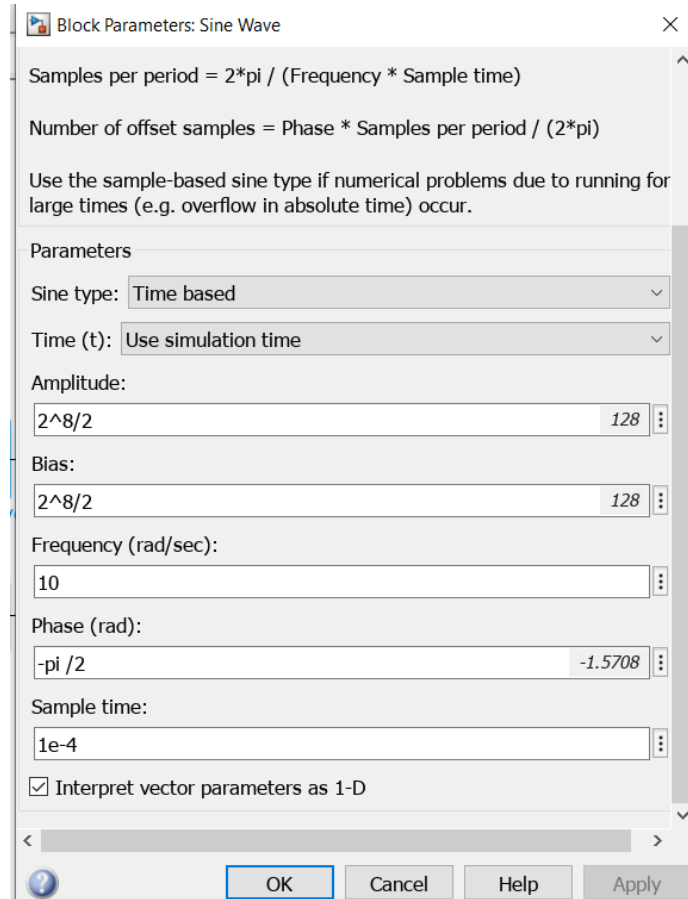


Рисунок 1

Структурная схема DAC, показана на рисунке 2.

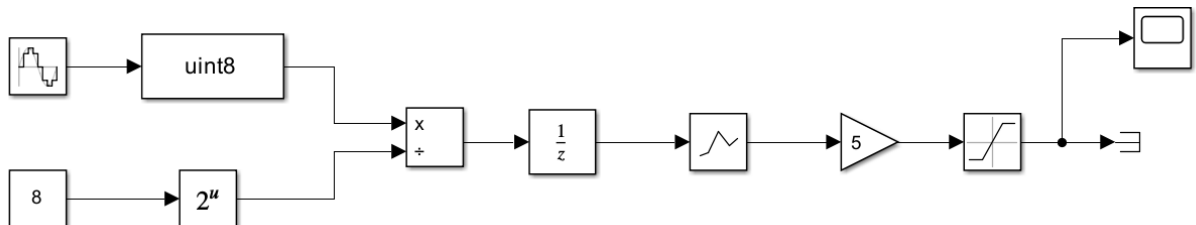


Рисунок 2, DAC

До блока продукт происходит деление синусоидального на 2^8 , это необходимо для того, чтобы получить число от 0 до 1, которое в дальнейшем

можно легко преобразовать в напряжение от 0 до 5V (блок Gain, ограничивает верхнюю границу).

Полученные графики

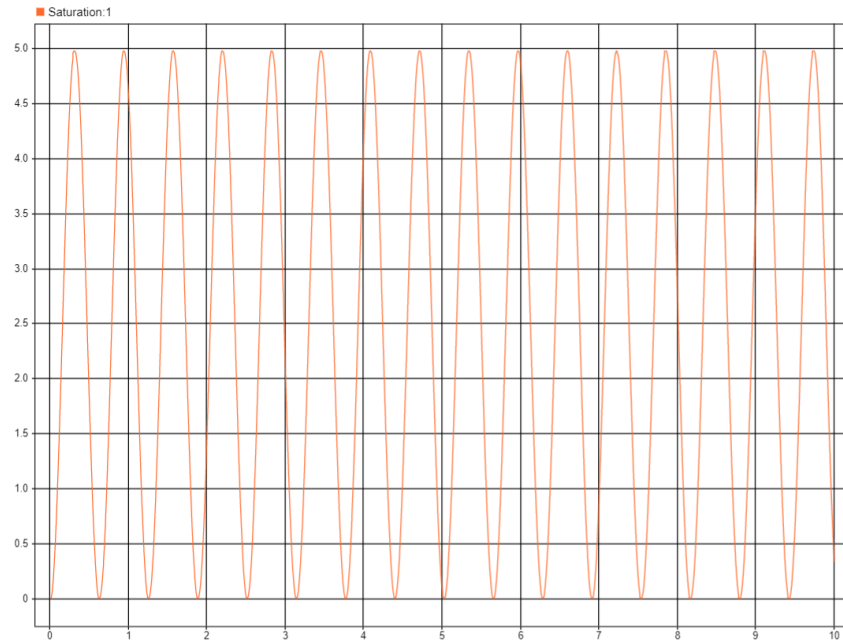


Рисунок 3, результаты

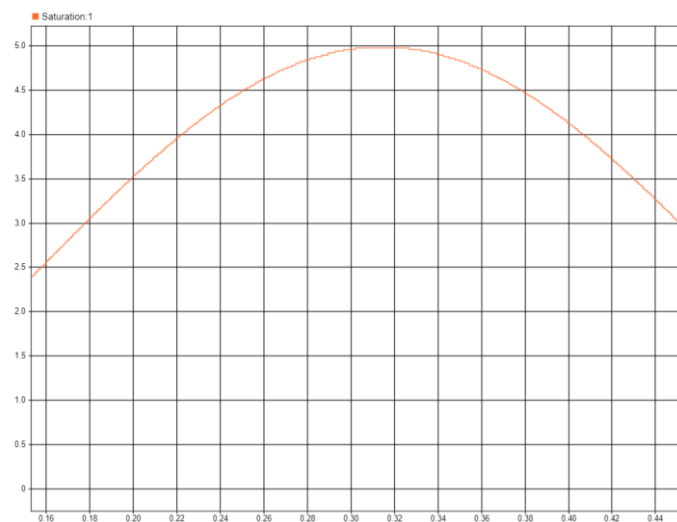


Рисунок 4, результаты в масштабе

2. ДЗ по курсу нелинейной ТАУ

Main.m

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

```

%% Constants parameters
b = 1;
c = 5;
x_max = 3;
y_max = c + 1;
COUNT = 5000;
a = 2;

%% Settings plot
figure
subplot(2, 2, 1);
hold on;
grid on;

%% First branch
x = linspace(-x_max, x_max, COUNT);
x_dot = ones([1 COUNT]);
y_1 = Nelineenost(b, c, x, x_dot);
plot(x, y_1, 'Color', 'Black');
plot(b, 0, '^', 'Color', 'Black');

%% Second branch
x = linspace(x_max, -x_max, COUNT);
x_dot = ones([1 COUNT]) .* -1;
y_2 = Nelineenost(b, c, x, x_dot);
plot(x, y_2, 'Color', 'Black');
plot(-b, 0, 'v', 'Color', 'Black');

xlim([-x_max x_max]);
ylim([-y_max y_max]);
xlabel('x');
ylabel('F(x)');

%% second graph
subplot(2, 2, 3);
hold on;
grid on;

% drawing a*sin(xsi);
xsi = linspace(0, 2 * pi, COUNT);
ax_x = a * sin(xsi);
plot(ax_x, xsi, 'Color', 'Black');
grid on
xlim([-x_max x_max]);
ylim([0-0.5 2*pi+0.5]);
axis ij;
xlabel('a*sin(xsi)');
ylabel('xsi');

%% third graph

```

```

subplot(2, 2, 2);
hold on;
grid on;

% drawing  $F(a*\sin(xsi)); dF = a* \cos(xsi)$ 

xsi = linspace(0, 2 * pi, COUNT);
f_a_sinx = a .* sin(xsi);
f_dot_a_sinx = a .* cos(xsi);

y_res = Nelineenost(b, c, f_a_sinx, f_dot_a_sinx);
plot(xsi, y_res, 'Color','Black');
xlabel('xsi');
ylabel('F(a*sin(xsi))');
xlim([-1 2*pi + 1]);
ylim([-y_max y_max]);

```

Reshen_yravnen.m

```

clc
clear all;
close all;

syms A w p

eqns = [-0.1 * w^2 + 40 * 5 / (pi * A) * sqrt(1 - 1 / A^2) == 0, -
0.0025 * w^3 + w - 40 * 5 / (pi * A^2) == 0];
vars = [A, w];

S = vpasolve(eqns, [A, w], [2,4])

X = -0.1 * w^2 + 40 * 5 / (pi * A) * sqrt(1 - 1 / A^2);
Y = -0.0025 * w^3 + w - 40 * 5 / (pi * A^2);

```

Third_task.m

```

clc;
clear all;
close all;

%% Constants
STEP = 1e-4;
MAX_A = 1e3;

% s = tf("s");

```

```

% W_1 = 10 / (s * (0.05 * s + 1)^2);
%
% figure("Name", "Bode diagram", "NumberTitle","off");

% bode(W_1);
% grid on;

plot_log_W_nelin(@q_a, @q_str_a);

function plot_log_W_nelin(func_q, func_q_str)
    figure("Name", "My lgscope", "NumberTitle", "off");
    s = tf("s");
    w = logspace(-1,3);
    W_1 = 10 ./ (1i .* w .* (0.05 .* 1i .* w + 1) .^ 2);
    RE_W = real(W_1);
    IM_W = imag(W_1);

    subplot(2,1,1);
    semilogx(w, 20 .* log(sqrt(RE_W .^ 2 + IM_W .^ 2)));
    hold on;
    xline(14.1);
    W_14 = 10 ./ (1i .* 14.1 .* (0.05 .* 1i .* 14.1 + 1) .^ 2);
    RE_14 = real(W_14);
    IM_14 = imag(W_14);
    yline(20 .* log(sqrt(RE_14 .^ 2 + IM_14 .^ 2)))
    xline(3.02);

    subplot(2,1,2);
    semilogx(w, (-pi + atan(IM_W ./ RE_W)) .* 180 ./ pi);
    hold on;
    xline(14.1);
    xline(3.02);
    yline(-160.63);

    a = logspace(-1, 3);

    subplot(2,1,1);
    semilogx(a, -20 .* log(sqrt(func_q(a) .^ 2 + func_q_str(a) .^
2)), "Color", "RED");
    hold on;
    grid on;

    subplot(2,1,2);
    semilogx(a, (-pi - atan(func_q_str(a) ./ func_q(a))) .* 180 ./
pi, "Color", "RED");
    grid on;
    hold on;
end

```

```

function [res] = q_a(a)
    res = 20 ./ (pi .* a) .* sqrt(1 - 1 ./ (a .^ 2));
end

function [res] = q_str_a(a)
    res = -20 ./ (pi .* (a .^ 2));
end

```

Были

получены

графики:

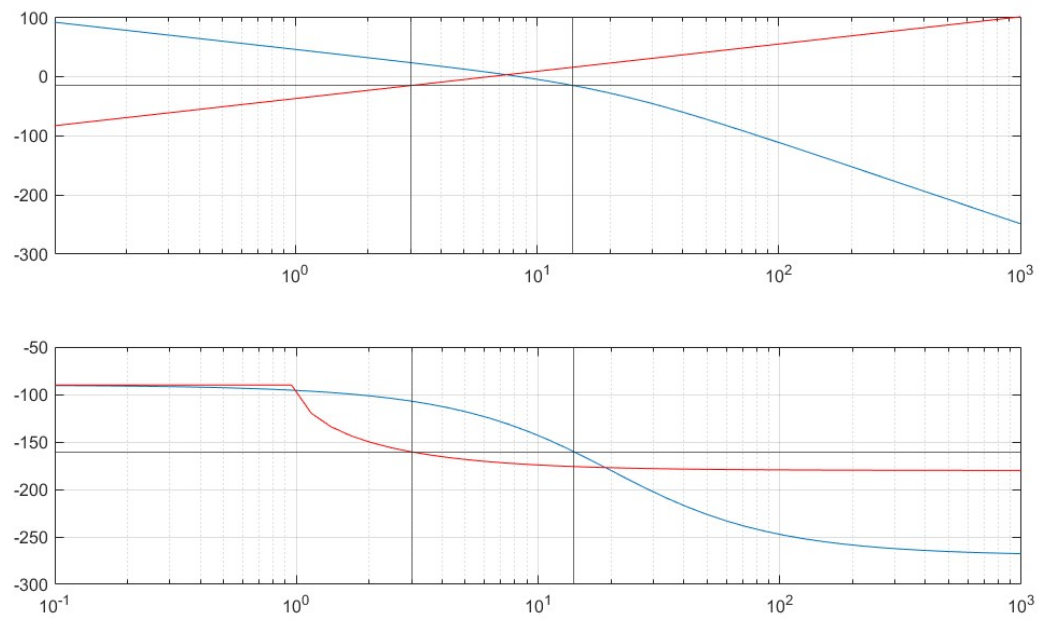


Рисунок 5, ЛАЧХ

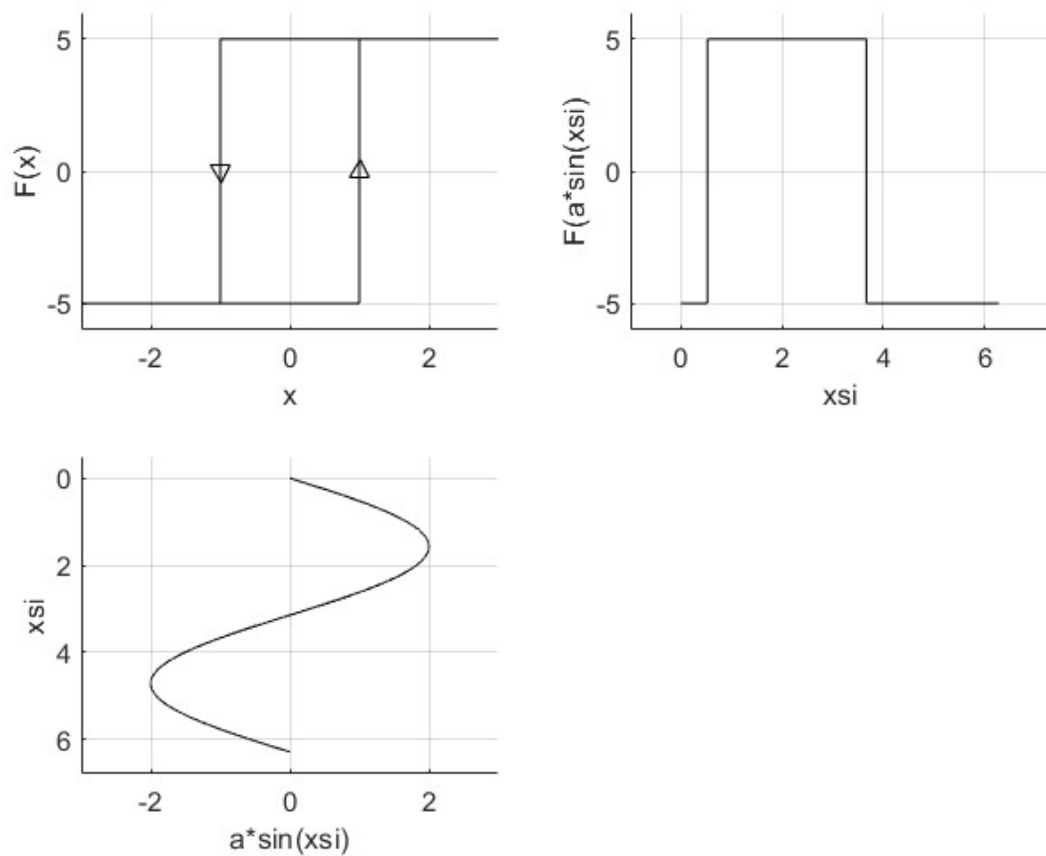


Рисунок 6, нелинейность

S =

struct with fields:

A: 3.0006488381405513578873168255766

w: 14.143282385835841778119853634816

Рисунок 7, решение уравнений

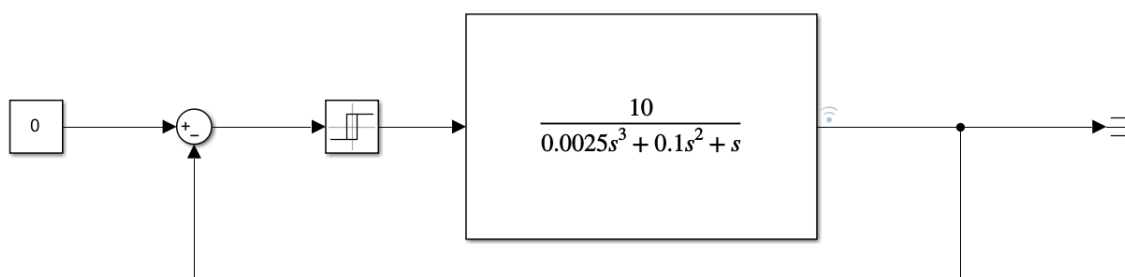


Рисунок 8, система в сборе

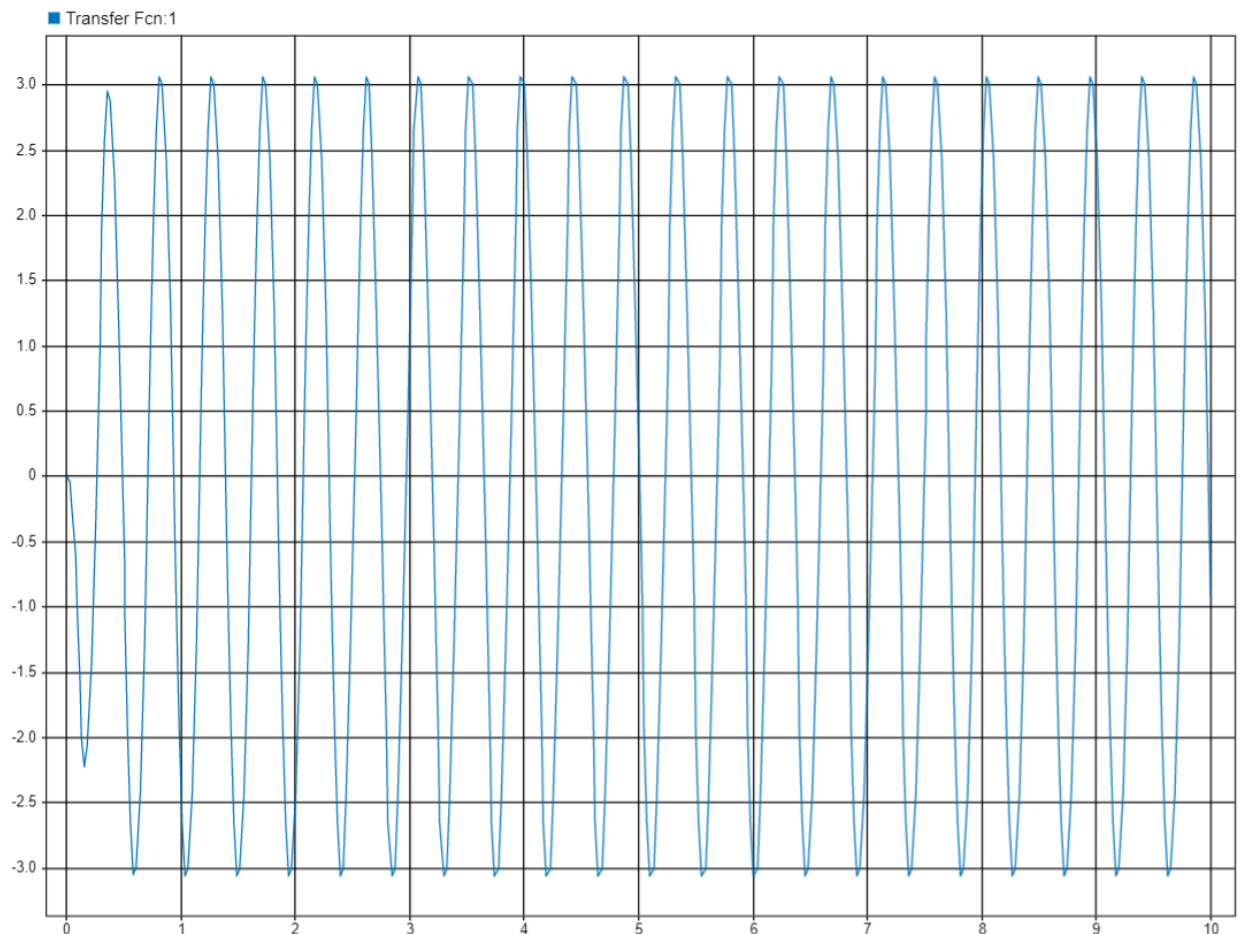


Рисунок 9, полученные графики

3. Синтез системы с дискретным регулятором

Была смоделирована система, представленная ниже.

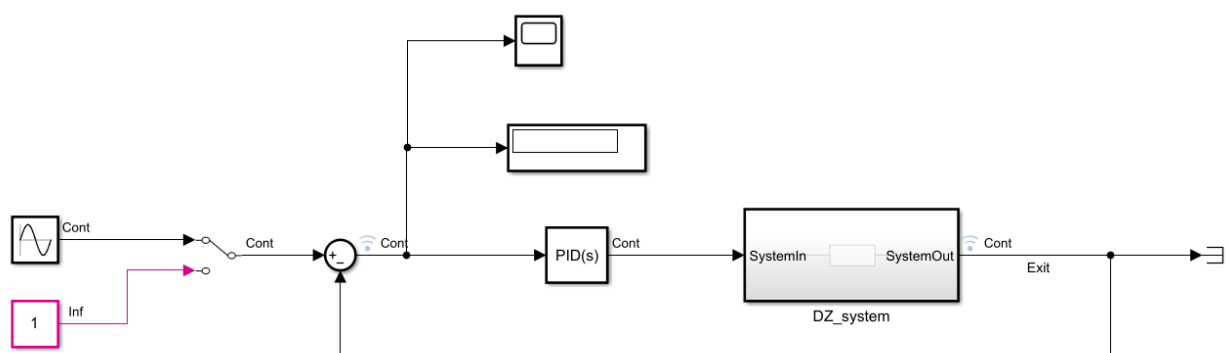


Рисунок 10

Настроим PID, так чтобы он удовлетворял требованиям:

- Перерегулирование не более 30% при подаче на вход единицы;
- Динамическая ошибка 1 градус при подаче на вход синуса частотой 0.5 рад/с и амплитудой 50 градусов.

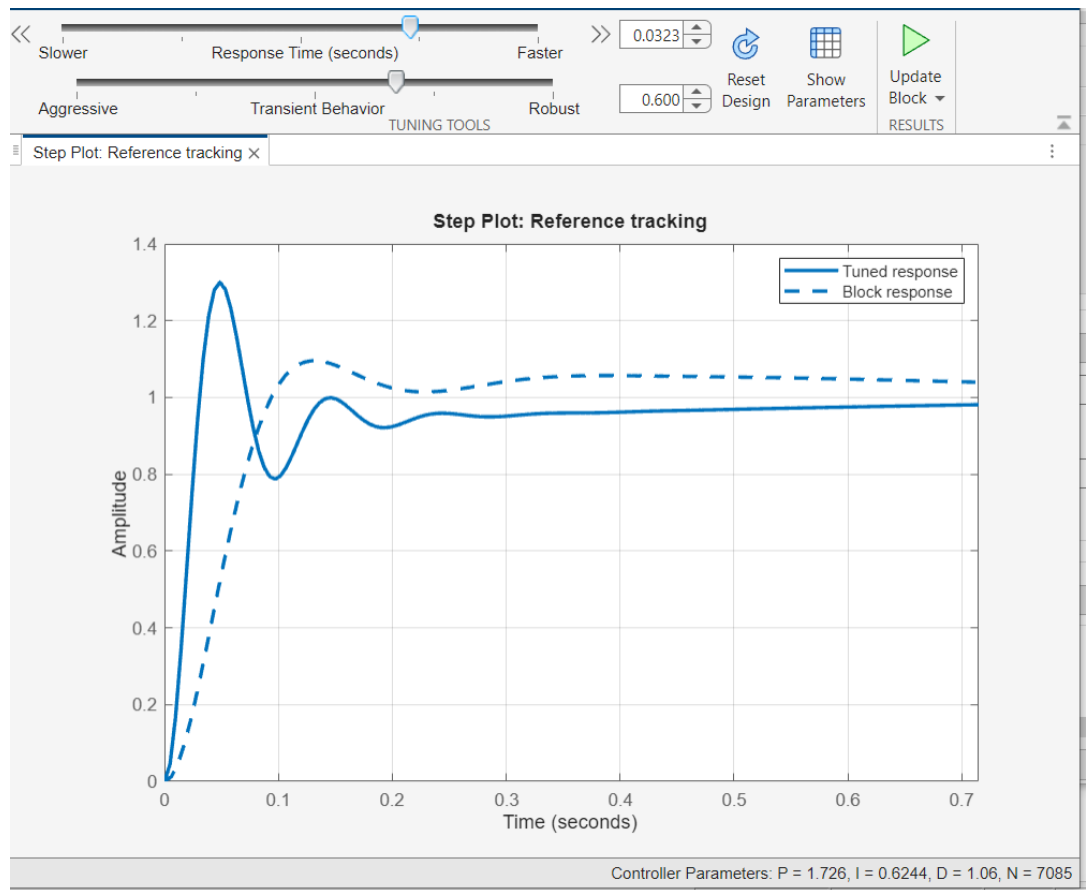


Рисунок 11, синтез PID

Controller Parameters			
	Tuned	Block	
P	1.7263	1.9839	
I	0.62436	3.2117	
D	1.0605	0.1986	
N	7085.1083	2437.0409	
Performance and Robustness			
	Tuned	Block	
Rise time	0.021 seconds	0.0625 seconds	
Settling time	0.692 seconds	1.05 seconds	
Overshoot	30 %	9.56 %	
Peak	1.3	1.1	
Gain margin	36.2 dB @ 522 rad/s	39.4 dB @ 271 rad/s	
Phase margin	33.8 deg @ 62 rad/s	60 deg @ 21.3 rad/s	
Closed-loop stability	Stable	Stable	

Рисунок 12

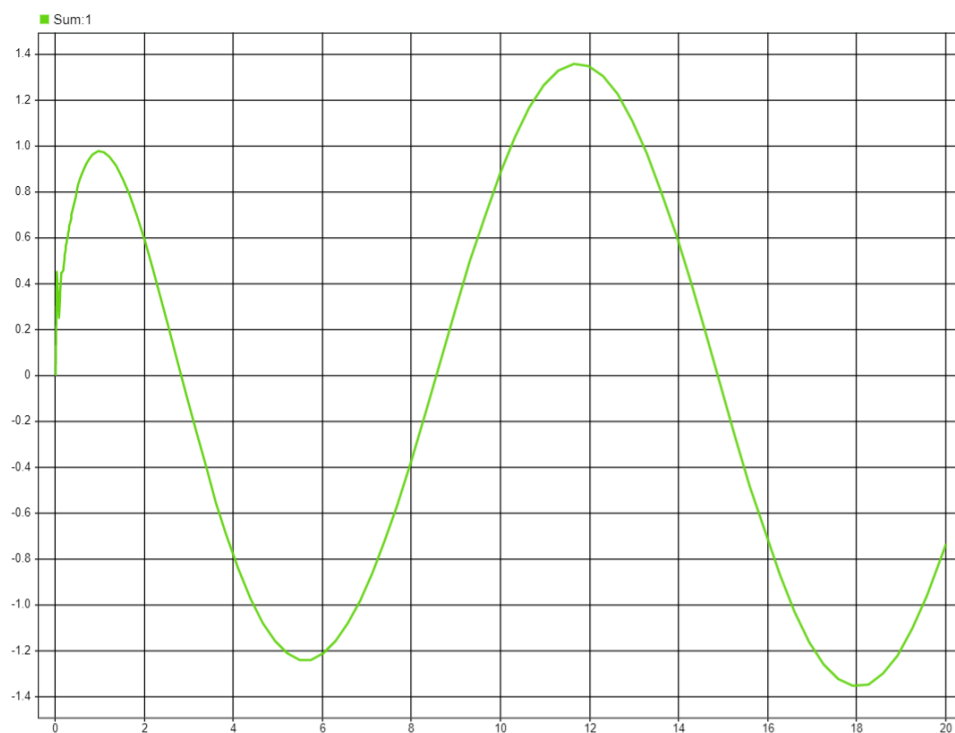


Рисунок 13, график ошибки

Как видно из графика динамической ошибки в 1 градус достичь не удалось

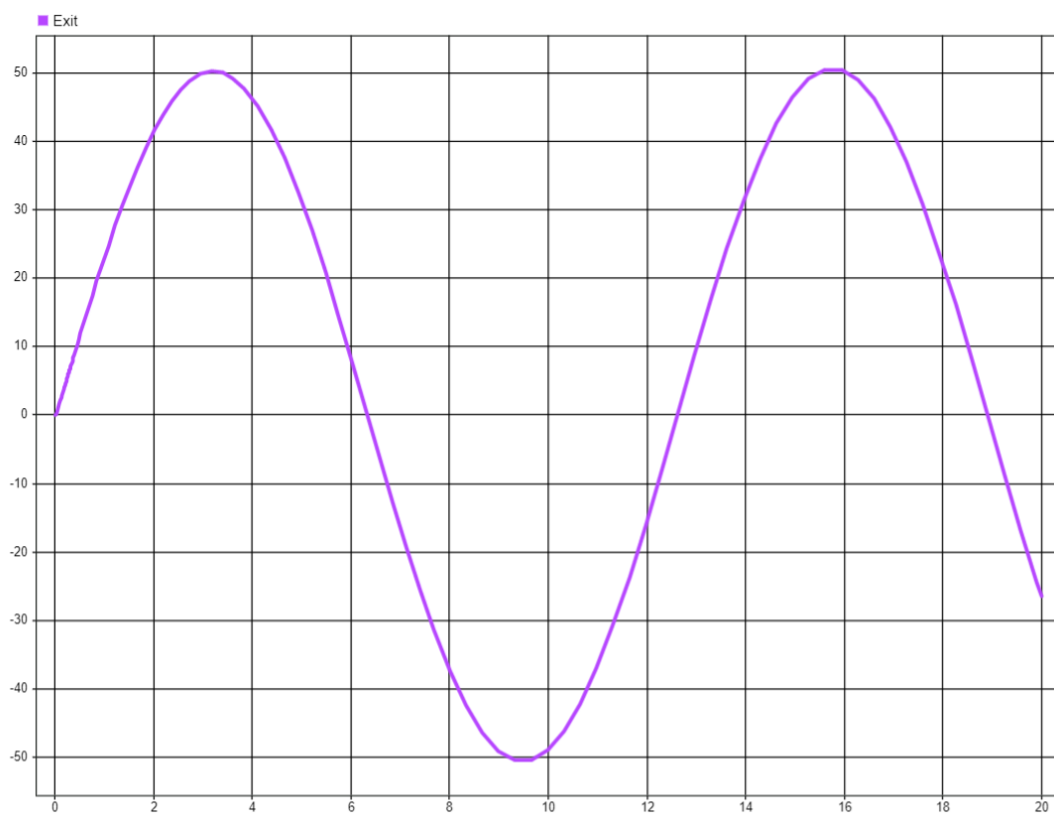


Рисунок 14, выходная синусоида

Заменим PID регулятор дискретным. Для этого введём передаточную функцию PID в файл matlab и используем команду c2d.

```
addpath("Doinig_lab\");  
  
s = tf('s');  
  
Peredat_Regul = 1.72626593892633 + 0.624359128150613 * (1 / s) +  
1.06047859947846 * (7085.10834495725 / (1 + 7085.10834495725 * (1 /  
s)))  
  
Z_Regul = c2d(Peredat_Regul, 1e-4, 'zoh')
```

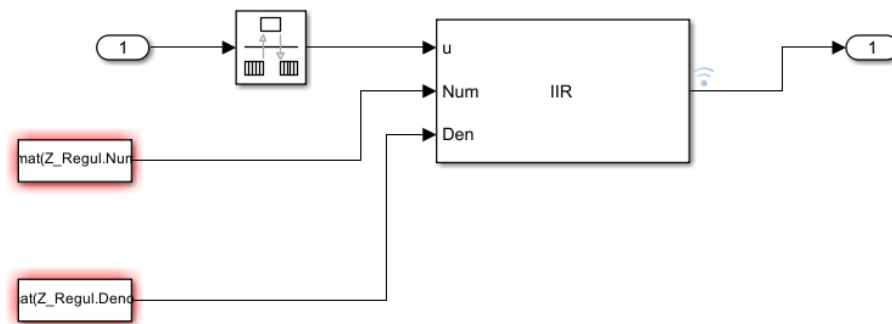


Рисунок 15, реализация дискретного PID

Полученные графики:

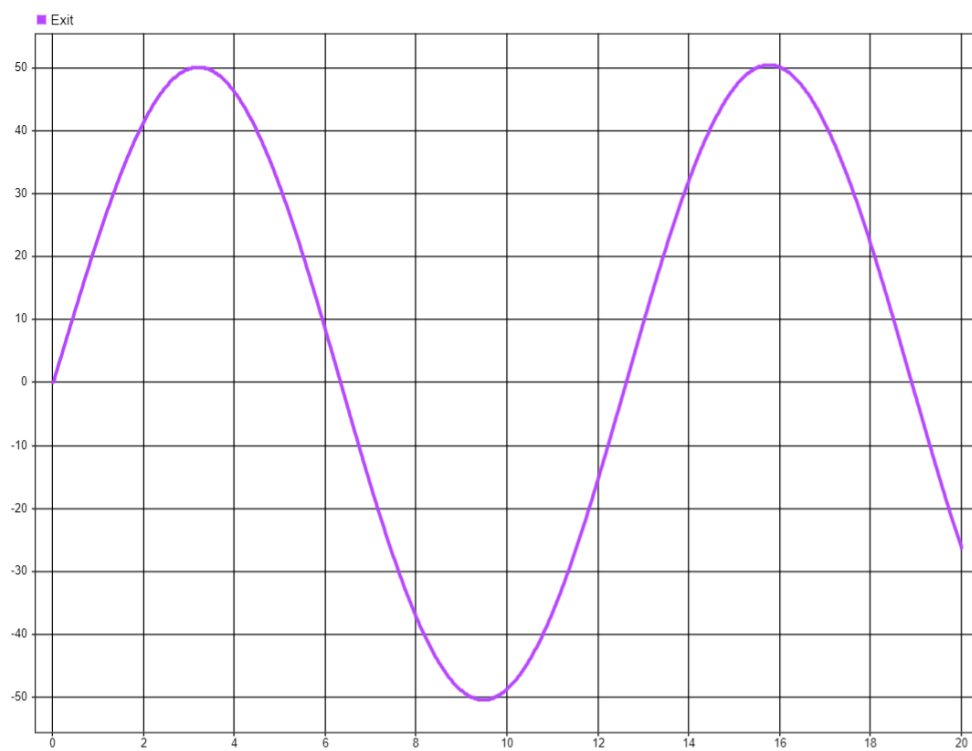


Рисунок 16, выход при синусоиде

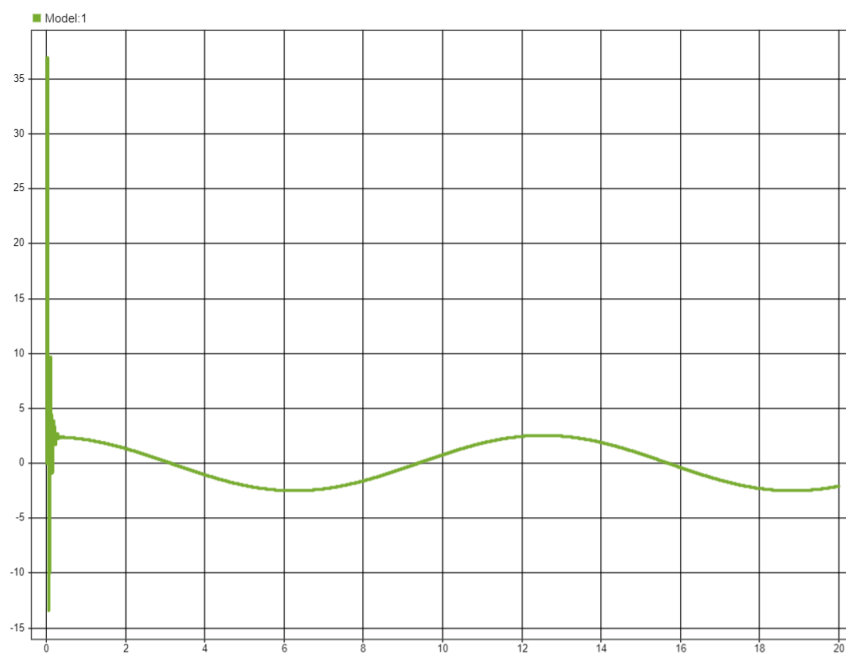


Рисунок 17, управляющий сигнал при синусоиде

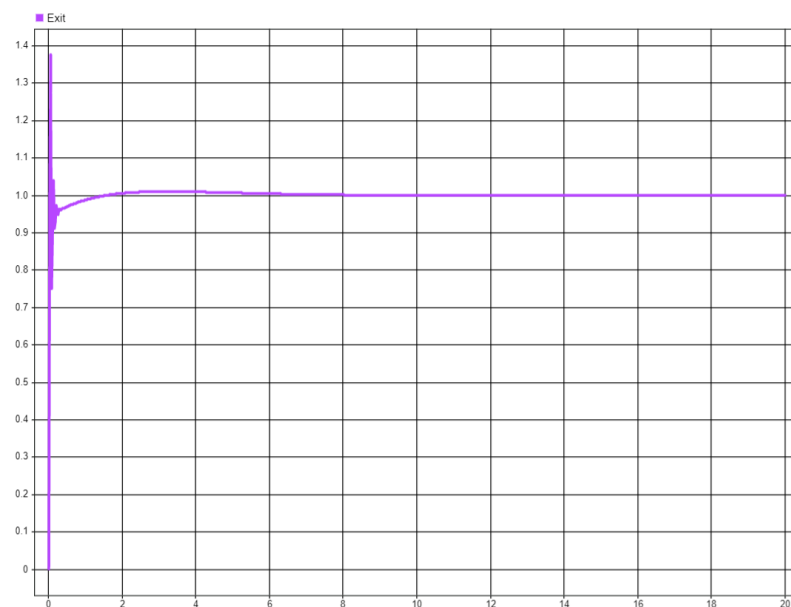


Рисунок 18, выход при единице

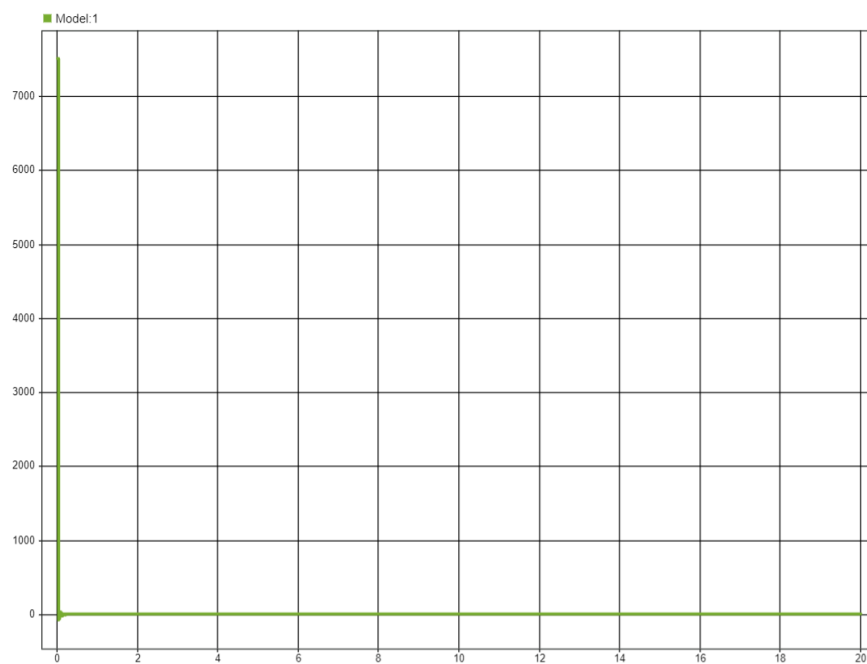


Рисунок 19, управление при единице