

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САУ

ОТЧЕТ
по практической работе №7
по дисциплине «МОПСУ»
ТЕМА: Разработка программ для программируемых логических
контроллеров (PLC) и микропроцессоров
ВАРИАНТ №9

Студент гр. 9492

Викторов А.Д.

Преподаватель

Игнатович Ю.В.

Санкт-Петербург

2024

Задание

На основе рассмотренных в лекции методов разработки программ для программируемых логических контроллеров (PLC) и микропроцессоров и результатов предыдущих лабораторных и практических работ, выполнить разработку дискретных ПИД-регуляторов для своего варианта следящей системы управления с ДПТ независимого возбуждения. При этом должны быть выполнены все рассмотренные варианты построения регуляторов

Цель работы

Цель данной работы заключается в изучении методов разработки для ПЛК и МК.

Исходные данные

Паспортные данные двигателя постоянного тока приведены в таблице 1:

Таблица 1

Марка двигател	P_n , Вт	ω_n , рад/с	U_n , В	I_n , А	M_n , Н·м	$J_{дв} \cdot 10^{-4}$, кг·м ²	$R_{я}$, Ом	$L_{я}$, мГн
СЛ-121	77	315	110	1,07	0,245	1,67	8,5	58

Содержание практической работы

1. При помощи листинга 1 построим зададим параметры двигателя, на рисунке 1 представлена исходная модель системы:

Листинг 1 – Задание параметров двигателя:

```
clc;
clear;
close all;
%СЛ-121 Вариант 9
Pn = 77; %Номинальная мощность, Вт
wn = 315; % Номинальная скорость вращения, рад/с
Un = 110; % Номинальное напряжение, В
In = 1.07; % Номинальный ток якоря, А
Mn = 0.245; % Номинальный момент, Н*м
Jd = 10^-4 * 1.67; % Момент инерции двигателя, кг*м^2
R = 8.5; % Сопротивление якоря, Ом
```

```

L = 10^-3 * 58; % Индуктивность якоря, Гн
Jn=2*Jd; % Приведенный момент инерции на валу двигателя
km=Mn/In; % Коэффициент между током и моментом
ke=(Un-R*In)/wn; % Коэффициент противо-ЭДС
Te=L/R; % Постоянная времени якорной цепи
k1=1/R;
k2=km/Jn; % k1, k2 вспомогательные коэффициенты
ku=20; % Коэффициент усиления усилителя
numdv=ku*1/ke; % Числитель ПФ
dendv=[Te/(k1*k2*ke) 1/(k1*k2*ke) 1]; % Знаменатель ПФ
Wdpt=tf(numdv,dendv)% Передаточная функция ДПТ по скорости

```

1. При помощи листинга 1 построим зададим параметры двигателя, на рисунке 1 представлена исходная модель системы:

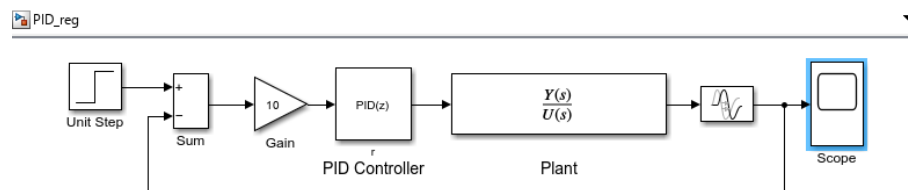


Рисунок 1 – Исходная модель системы

Модель системы управления на рисунке 2:

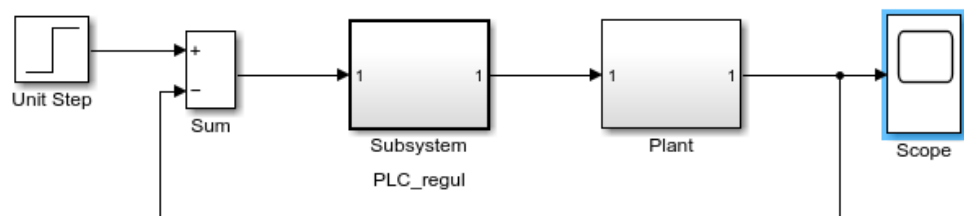


Рисунок 2 – Модель системы управления

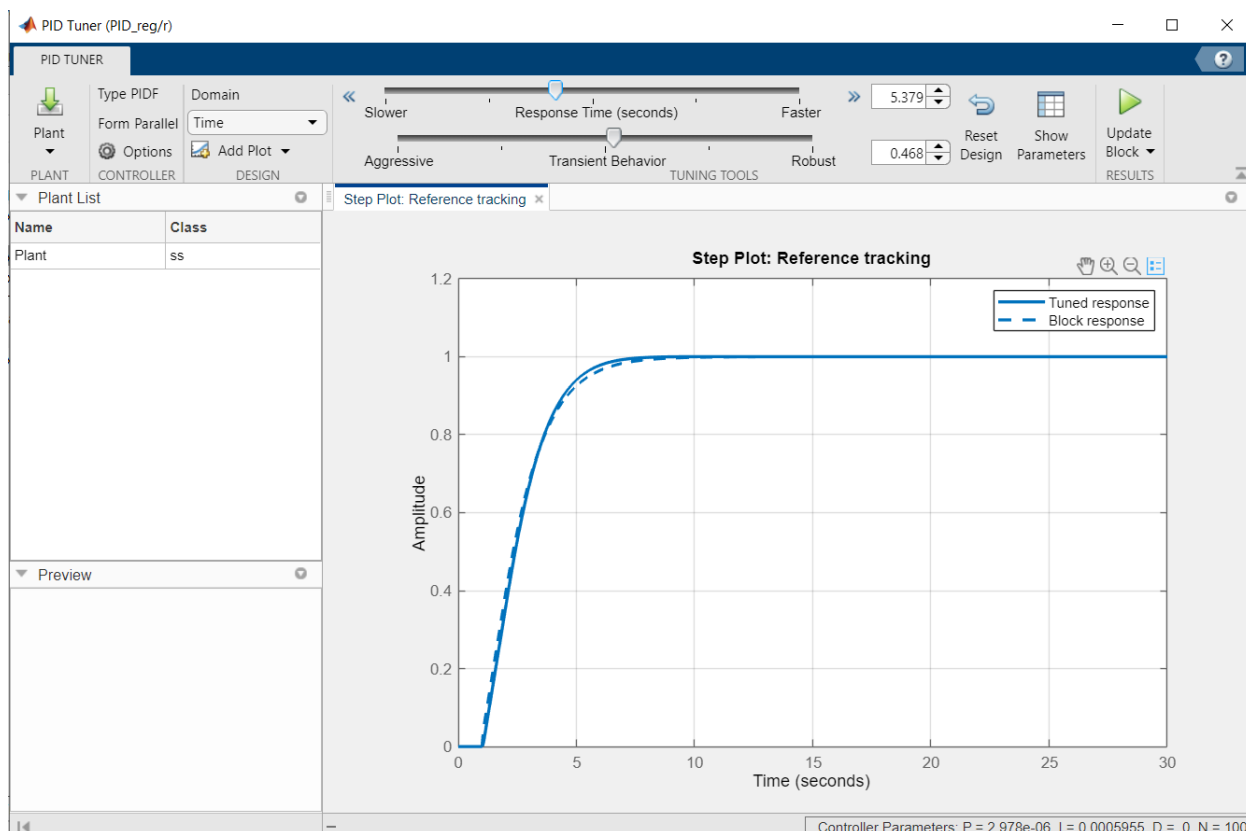


Рисунок 3 – Подбор коэффициентов

Настройка параметров PLC_regul на рисунке 4:

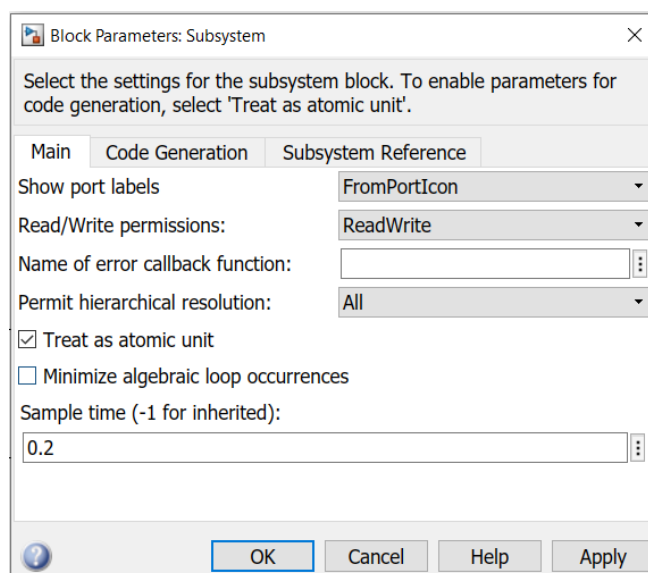


Рисунок 4 – Настройка параметров

Сгенерированный код представлен в листинге 2.

```
FUNCTION_BLOCK Subsystem
VAR_INPUT
    ssMethodType: SINT;
    In1: LREAL;
END_VAR
VAR_OUTPUT
    Out1: LREAL;
END_VAR
VAR
    Filter_DSTATE: LREAL;
    Integrator_DSTATE: LREAL;
END_VAR
VAR_TEMP
    rtb_Gain: LREAL;
    rtb_FilterCoefficient: LREAL;
END_VAR
CASE ssMethodType OF
    SS_INITIALIZE:
        (* SystemInitialize for Atomic SubSystem: '<Root>/Subsystem' *)
        (* SystemInitialize for Atomic SubSystem: '<S1>/PLC_reg' *)
        (* InitializeConditions for DiscreteIntegrator: '<S30>/Filter' *)
        Filter_DSTATE := 0.0;
        (* InitializeConditions for DiscreteIntegrator: '<S35>/Integrator' *)
        Integrator_DSTATE := 0.0;
        (* End of SystemInitialize for SubSystem: '<S1>/PLC_reg' *)
        (* End of SystemInitialize for SubSystem: '<Root>/Subsystem' *)
    SS_STEP:
        (* Outputs for Atomic SubSystem: '<Root>/Subsystem' *)
        (* Gain: '<S1>/Gain' *)
        rtb_Gain := 10.0 * In1;
        (* Outputs for Atomic SubSystem: '<S1>/PLC_reg' *)
        (* Gain: '<S38>/Filter Coefficient' incorporates:
        *   DiscreteIntegrator: '<S30>/Filter'
        *   Gain: '<S29>/Derivative Gain'
        *   Sum: '<S30>/SumD' *)
        rtb_FilterCoefficient := (0.0 - Filter_DSTATE) * 100.0;
        (* Outport: '<Root>/Out1' incorporates:
        *   DiscreteIntegrator: '<S35>/Integrator'
        *   Gain: '<S40>/Proportional Gain'
        *   Sum: '<S44>/Sum' *)
        Out1 := ((2.97757805905239e-06 * rtb_Gain) + Integrator_DSTATE) +
rtb_FilterCoefficient;
        (* Update for DiscreteIntegrator: '<S30>/Filter' *)
        Filter_DSTATE := (0.2 * rtb_FilterCoefficient) + Filter_DSTATE;
        (* Update for DiscreteIntegrator: '<S35>/Integrator' incorporates:
        *   Gain: '<S32>/Integral Gain' *)
        Integrator_DSTATE := ((0.000595515611810478 * rtb_Gain) * 0.2) +
Integrator_DSTATE;
        (* End of Outputs for SubSystem: '<S1>/PLC_reg' *)
        (* End of Outputs for SubSystem: '<Root>/Subsystem' *)
END_CASE;
END_FUNCTION_BLOCK
VAR_GLOBAL CONSTANT
    SS_INITIALIZE: SINT := 0;
    SS_STEP: SINT := 1;
END_VAR
```

```

function [Out1, Integrator_DSTATE, Filter_DSTATE]...
=fcn_PID(In1, Integrator_DSTATE, Filter_DSTATE)

    rtb_Gain = 10.0 * In1;
    rtb_FilterCoefficient = (0.0 - Filter_DSTATE) * 100.0;
    Out1 = ((2.97757805905239e-06 * rtb_Gain) + Integrator_DSTATE) +
rtb_FilterCoefficient;
    Filter_DSTATE = (0.2 * rtb_FilterCoefficient) + Filter_DSTATE;

    Integrator_DSTATE = ((0.000595515611810478 * rtb_Gain) * 0.2) +
Integrator_DSTATE;
End

```

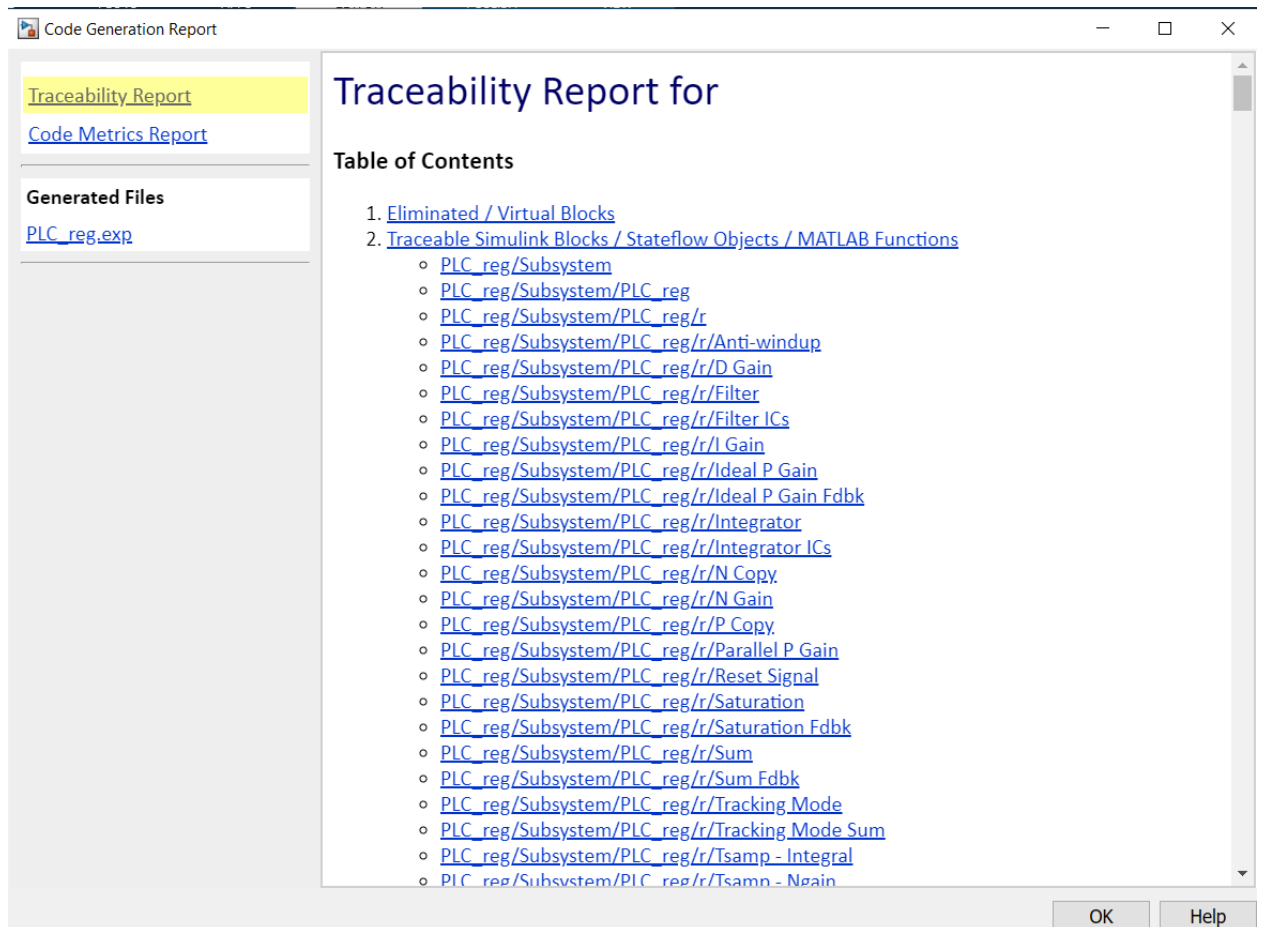


Рисунок 7 – Отчет о завершении компиляции

Чтобы проверить правильность кода была использована модель с рисунка 8, код был вставлен в блок fcn_PID, код приведен в Листинге 3:

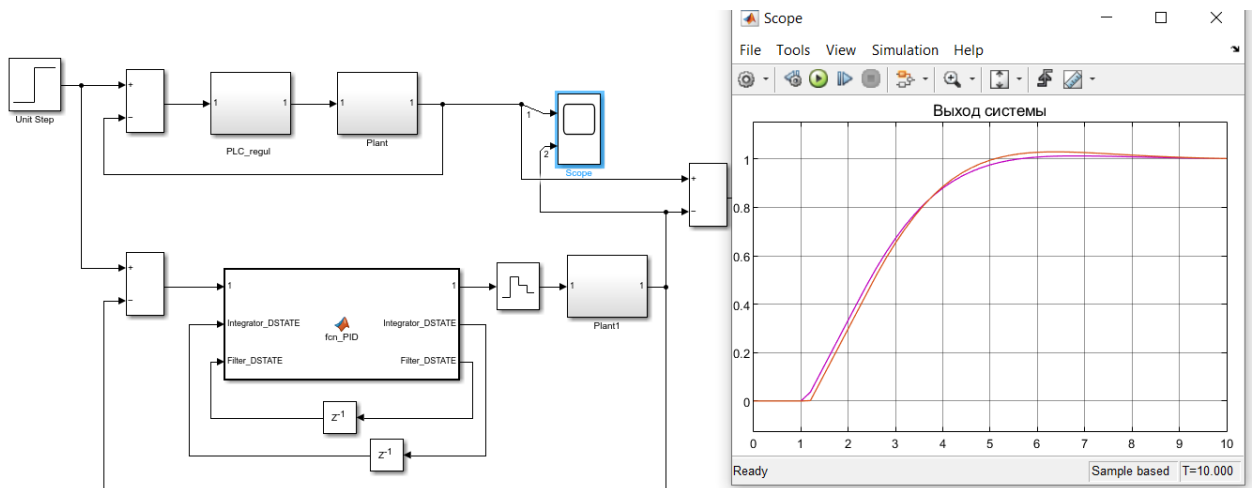


Рисунок 8 Сравнение ST кода и модели Simulink
Модель сравнивающая C и Simulink представлена на рисунке 9:

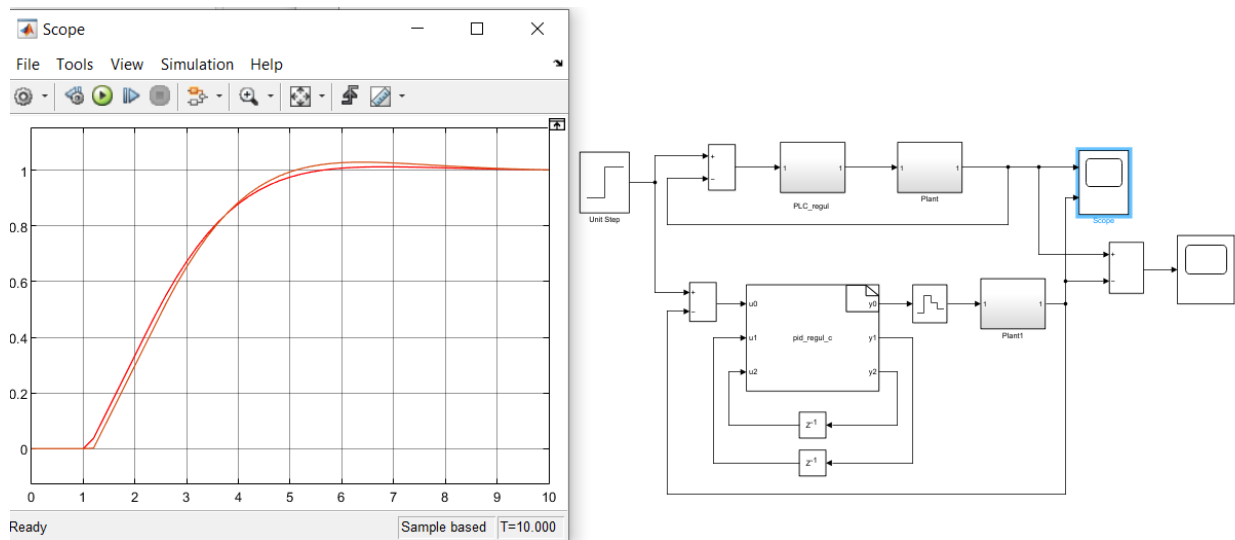


Рисунок 8 Сравнение кода на C и Simulink модели

Листинг 3:

```
void pid_regul_c_Outputs_wrapper(const real_T *u0,
                                const real_T *u1,
                                const real_T *u2,
                                real_T *y0,
                                real_T *y1,
                                real_T *y2)
{
    /* Output_BEGIN */
    float static input0=0;
    float static input1=0;
    float static input2=0;
    float static output=0;
    float static gain=0;
    float static filtercoefficient=0;
    float static input_prev=0;
    float static filter=0;
```

```

float static filter_out=0;
float static integrator=0;
float static integrator_out=0;
input0=u0[0];
input1=u1[0];
input2=u2[0];
gain = 10.0 * input0;
filter=input2;
integrator=input1;
filtercoefficient = (0.0 - filter) * 100.0;
output = ((2.97739e-06 * gain) + integrator) + filtercoefficient;
filter_out = (0.2 * filtercoefficient) + filter;
integrator_out = ((0.0005955 * gain) * 0.2) + integrator;
y0[0]=output;
y1[0]=integrator_out;
y2[0]=filter_out;
/* Output_END */
}
void pid_regul_c_Terminate_wrapper(void)
{
/* Terminate_BEGIN */
/*
 * Custom Terminate code goes here.
 */
/* Terminate_END */
}

```

Вывод

При выполнении данной практической работы были проведены различные способы генерации кода, который может быть использован в ПЛК (ST) и МК (C), результаты использования кода показывают, что применение иных программных средств от Simulink не приводит к изменению управления.