

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

Институт кибернетики

Кафедра проблем управления

Лабораторная работа №1

**Тема:** **«**Отладка программного обеспечения роботехнических систем с использованием виртуального моделирования»

Выполнил:

Студент 4-го курса

Группы: КРБО-01-17

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Хмелевский В.С.

Преподаватель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Морозов А.А г. Москва,

2020

**Цель работы:** получение навыков моделирования объекта управления в промышленных системах автоматического управления и создание функциональных блоков.

**Задание:** создать виртуальную систему управления (рис.1), включающую: модель объекта управления (рис.2), ПИ-регулятор (рис.3), сумматор и обратную связь. Передаточная функция объекта:

(1)

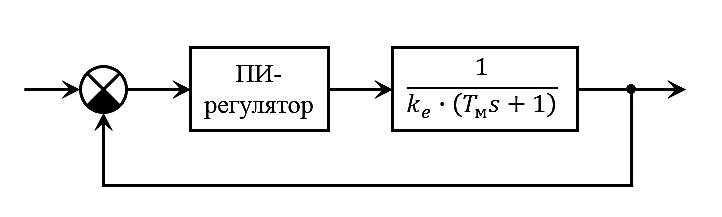


Рис. 1 – Структура системы управления

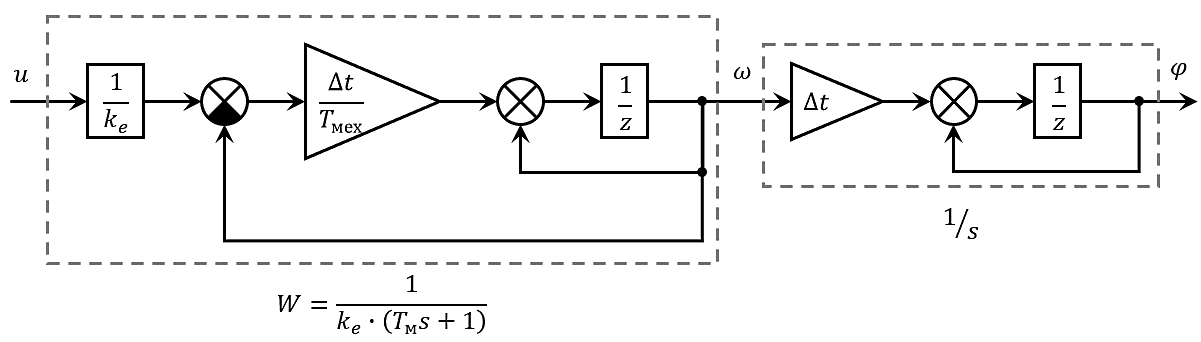


Рис. 2 – Структура объекта управления

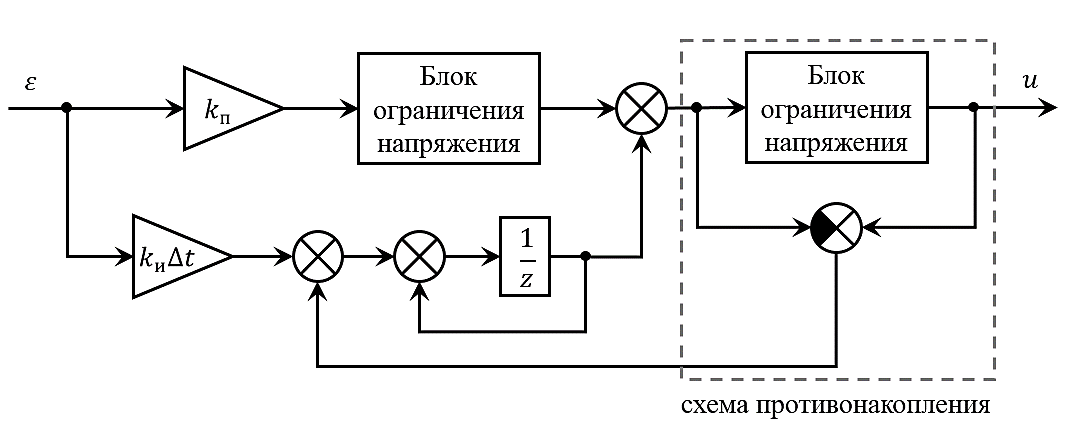


Рис. 3 – Структура ПИ-регулятора

**Создание проекта и реализация моделей:**

Был создан новый проект в программной среде Automation Studio, в который добавлены следующие объекты: ANSI C Library «MotorControl» и ANSI C Program. В ANSI C Library «MotorControl» добавлены следующие функциональные блоки:

А) FB\_Motor – блок, необходимый для реализации модели двигателя постоянного тока. На входе – напряжение. На выходе – частота вращения и положение.

Таблица 1 – Параметры функционального блока FBMotor

|  |  |  |  |
| --- | --- | --- | --- |
| **Конфигурация** | **Имя** | **Тип данных** | **Описание** |
| вход | u | REAL | входное напряжение [В] |
| выход | w | REAL | частота вращения [об/мин] |
| выход | phi | REAL | положение [рад] |
| внутреннее состояние | integrator | FB \_Integrator | интегратор |
| внутреннее состояние | Tm | REAL | электромеханическая постоянная времени [с] |
| внутреннее состояние | ke | REAL | постоянная ЭДС двигателя [В•мин/об] |
| внутреннее состояние | dt | REAL | шаг расчета [с] |

Б) FB\_Regulator – данный функциональный блок будет выполнять роль ПИ-регулятора с ограничением. На входе – значение рассогласования между задающим воздействием и реальной скоростью вала ДПТ. На выходе – напряжение, подаваемое на вход ДПТ.

Таблица 2. – Параметры функционального блока FB\_Regulator

|  |  |  |  |
| --- | --- | --- | --- |
| **Конфигурация** | **Имя** | **Тип данных** | **Описание** |
| вход | e | REAL | рассогласование между задающим воздействием и реальной скоростью вращения вала ДПТ [об/мин] |
| выход | u | REAL | напряжение, подаваемое на вход ДПТ [В] |
| внутреннее состояние | k\_p | REAL | пропорциональный коэффициент регулятора |
| внутреннее состояние | k\_i | REAL | интегральный коэффициент регулятора |
| внутреннее состояние | integrator | FB\_Integrator | интегратор |
| внутреннее состояние | iyOld | REAL | хранение предыдущего значения схемы противонакопления |
| внутреннее состояние | max\_abs\_value | REAL | граница блока ограничения [В] |
| внутреннее состояние | dt | REAL | шаг расчета [с] |

В) FB\_Integrator – блок, представляющий собой модель интегратора. На вход принимает значения с интегрирующего звена. Выход – соответственное значение с интегрирующего звена.

Таблица 3 – Параметры функционального блока FB\_Integrator

|  |  |  |  |
| --- | --- | --- | --- |
| **Конфигурация** | **Имя** | **Тип данных** | **Описание** |
| вход | in | REAL | вход интегрирующего звена |
| выход | out | REAL | выход интегрирующего звена |
| внутреннее состояние | dt | REAL | шаг расчета [с] |

Все переменные функциональных блоков двигателя, регулятора и интегратора изображены на Рис. 4.

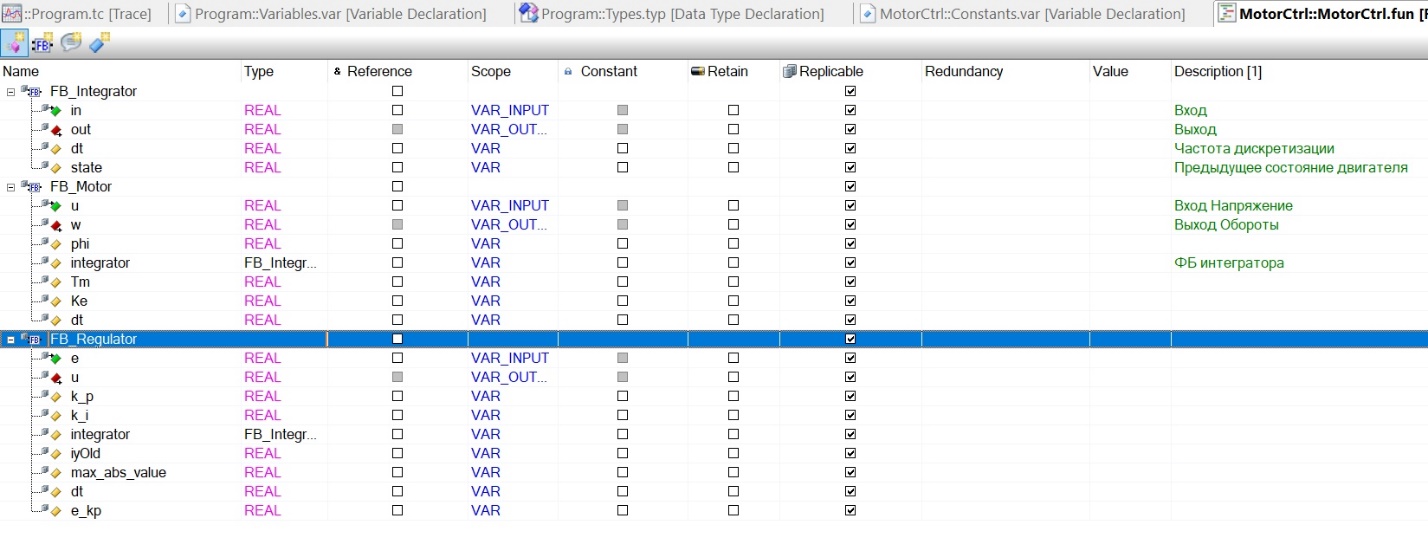


Рис. 4 – Переменные функциональных блоков

Характеристики для двигателя и регулятора были высчитаны в программе Octave:

* шаг расчета [с] ДПТ dt = 0.002;
* постоянная ЭДС двигателя [В•мин/об] Ke =2;
* электромеханическая постоянная времени [с] Tm = 0.4;
* желаемая постоянная времени [с] T\_ж = 0,1;
* значение integrator для FB\_Motor dt = 0.002;
* шаг расчета [с] регулятора =0.002с;
* cnt = 0;
* граница блока ограничения [В]
* max\_abs\_value = 60;
* speed 0-6
* пропорциональный коэффициент регулятора k\_p=8;
* интегральный коэффициент регулятора k\_i=20;

Необходимо создать экземпляры объектов в глобальных переменных программы: fb\_controller, fb\_motor2, fb\_motor. Создано два экземпляра мотора, чтобы снять показатели с графиков: для двигателя с регулятором и двигателя без регулятора. Глобальные переменные и экземпляры блоков представлены на Рис. 5.

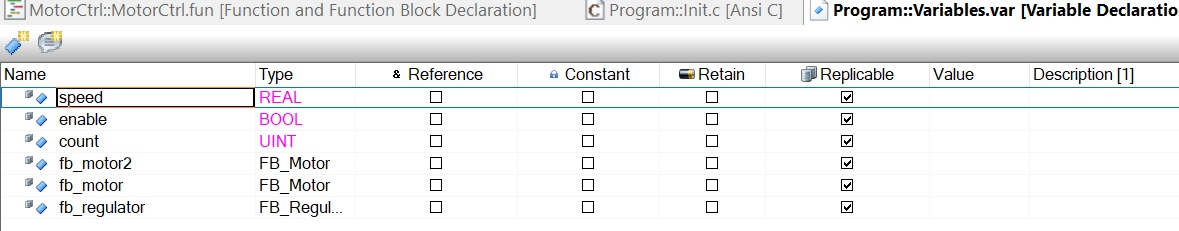


Рис. 5 – Переменные основной программы

Объединяем объект и регулятор в систему управления в основной программе с применением разработанных функциональных блоков, разрабатываем программу, подающую на систему управления ступенчатое воздействие.

**Результаты тестирования:**

После компилирования проекта и снятия показаний изменения с величин: speed, fb\_motor.w, fb\_motor2.w, был получен график изменения данных величин, представленный на Рис. 6. По показаниям графиков, скорость выхода на установленных оборотах и скорость остановки двигателя с регулятором - выше, чем у двигателя без регулятора.

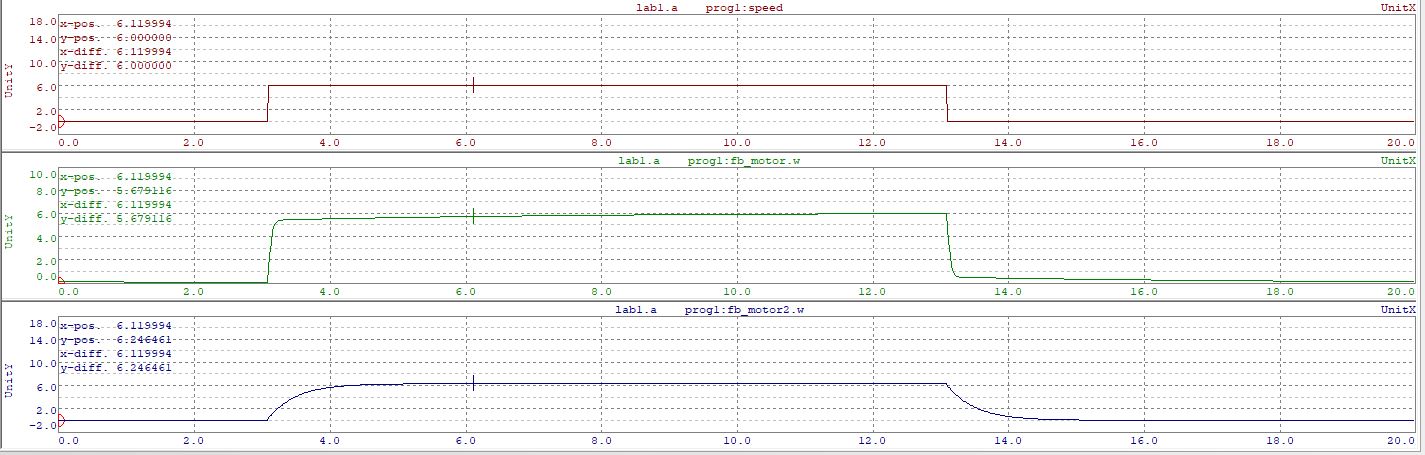


Рис. 6 – График уставки, скорости вращения двигателя без регулятора и скорости вращения двигателя с настроенным регулятором

**Итог:**

В ходе выполнения лабораторной работы была создана модель системы управления двигателями с регулятором и без регулятора. Преимущество системы управления с регулятором над системой управления без регулятора заключается в том, что регулятор позволяет контролировать некоторые параметры переходного процесса (время прихода в установившийся режим). Таким образом система с регулятором входит в установившийся режим за указанное время.

**Листинг программы 1**

#include &lt;bur/plctypes.h&gt;

#ifdef \_DEFAULT\_INCLUDES

#include &lt;AsDefault.h&gt;

#endif

void \_INIT ProgramInit(void)

{

enable=1;

count=0;

fb\_motor.dt = 0.01;

fb\_motor.ke=2;

fb\_motor.Tm=0.4;

fb\_motor2.dt=0.01;

fb\_motor2.ke=2;

fb\_motor2.Tm=0.4;

fb\_controller.dt=0.01;

fb\_controller.k\_p=8;

fb\_controller.k\_i=20;

fb\_controller.max\_abs\_value=24;

}

**Листинг программы 2**

#include &lt;bur/plctypes.h&gt;

#ifdef \_DEFAULT\_INCLUDES

#include &lt;AsDefault.h&gt;

#endif

void \_CYCLIC ProgramCyclic(void)

{

if(enable)

{

count += 10;

if(count&lt;=5000)

{

speed=0;

}

else

{

speed=6;

if(count&gt;=10000)

{

count=0;

}

}

}

fb\_controller.e=speed-fb\_motor.w;

fb\_motor2.u=speed\*fb\_motor2.ke;

FB\_Regulator(&amp;fb\_controller);

fb\_motor.u=fb\_controller.u\*fb\_motor.ke;

FB\_Motor(&amp;fb\_motor);

FB\_Motor(&amp;fb\_motor2);

}

**Листинг программы 3**

#include &lt;bur/plctypes.h&gt;

#ifdef \_\_cplusplus

extern &quot;C&quot;

{

#endif

#include &quot;MotrContr.h&quot;

#ifdef \_\_cplusplus

};

#endif

void FB\_Integrator(struct FB\_Integrator\* inst)

{

inst-&gt;out += inst-&gt;in;

}

#include &lt;bur/plctypes.h&gt;

#ifdef \_\_cplusplus

extern &quot;C&quot;

{

#endif

#include &quot;MotrContr.h&quot;

#ifdef \_\_cplusplus

};

#endif

void FB\_Motor(struct FB\_Motor\* inst)

{

inst-&gt;integrator.in = (inst-&gt;u / inst-&gt;ke - inst-&gt;w) \* inst-&gt;dt /

inst-&gt;Tm;

FB\_Integrator(&amp;(inst-&gt;integrator));

inst-&gt;w = inst-&gt;integrator.out;

inst-&gt;integrator.in = inst-&gt;w \* inst-&gt;dt;

FB\_Integrator(&amp;(inst-&gt;integrator));

inst-&gt;phi = inst-&gt;integrator.out;

}

#include &lt;bur/plctypes.h&gt;

#ifdef \_\_cplusplus

extern &quot;C&quot;

{

#endif

#include &quot;MotrContr.h&quot;

#ifdef \_\_cplusplus

};

#endif

void FB\_Regulator(struct FB\_Regulator\* inst)

{

inst-&gt;integrator.in=inst-&gt;e \* inst-&gt;k\_i \* inst-&gt;dt + inst-&gt;iyOld;

FB\_Integrator(&amp;(inst-&gt;integrator));

inst-&gt;e\_kp=inst-&gt;e\*inst-&gt;k\_p;

inst-&gt;e\_kp=(inst-&gt;e\_kp &gt; inst-&gt;max\_abs\_value || inst-&gt;e\_kp &lt; - inst-

&gt;max\_abs\_value)?((inst-&gt;e\_kp&gt;0)?inst-&gt;max\_abs\_value:-inst-

&gt;max\_abs\_value):inst-&gt;e\_kp;

inst-&gt;e\_kp+=inst-&gt;integrator.out;

inst-&gt;u=(inst-&gt;e\_kp &gt; inst-&gt;max\_abs\_value || inst-&gt;e\_kp &lt; - inst-

&gt;max\_abs\_value)?((inst-&gt;e\_kp&gt;0)?inst-&gt;max\_abs\_value:-inst-

&gt;max\_abs\_value):inst-&gt;e\_kp;

inst-&gt;iyOld=inst-&gt;u-inst-&gt;e\_kp;

}

**Ссылка на репозиторий:**

**https://github.com/VladislavHmelevskiy/labwork1**