

Тестовое задание C/C++

Окружение

Linux

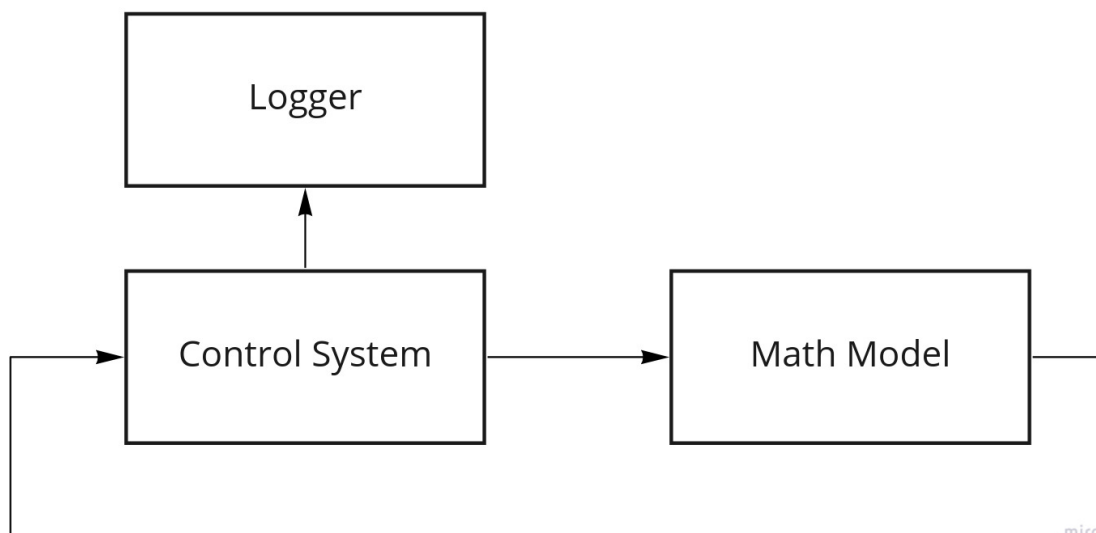
C++03

Задание

В данном тестовом задании необходимо разработать три взаимодействующих между собой приложения:

- Математическая модель (Math Model)
- Система управления (Control System)
- Система записи телеметрической информации (Logger)

Система управления по заданному входному сигналу стабилизирует величину на выходе математической модели и отправляет данные в систему записи телеметрической информации. Данные приложения взаимодействуют между собой через TCP соединение. Ниже на схеме отображено взаимодействие между приложениями:



Picture 1 - Structure diagram

Архитектура приложений должна основываться на принципах ООП: каждое приложение должно состоять из соответствующего класса, унаследованного от базового, с виртуальными методами инициализации (init) и обновления (update). Поток с методом update должен вызываться с заданной частотой и блокироваться до следующей итерации. Для обмена данными между приложениями вводится понятие сигнала. Сигнал представляет собой структуру, состоящую из наименования, ID и значения.

1. Математическая модель (Math Mode)

Математическая модель имеет вход x и выход y и описывается следующим дифференциальным уравнением:

$$T_m \frac{d}{dt} y(t) + y(t) = K_m x(t),$$

где

$y(t)$ - выходной сигнал,

$x(t)$ - входной сигнал,

T_m - постоянная времени модели,

K_m - коэффициент усиления модели;

Данная математическая модель представляет собой апериодическое звено и упрощенно описывает динамику двигателя, сервопривода. Необходимо реализовать функцию, которая по входному сигналу и параметрам математической модели рассчитывает выходной сигнал.

В качестве аргументов приложение принимает входной порт, выходной порт, постоянную времени, коэффициент усиления и период итерирования:

```
math_model -i [in_port] -o [out_port] --Tm [time_const] --Km [gain] -t [period]
```

пример:

```
math_model -i 8004 -o 8005 --Tm 0.5 --Km 10 -t 0.01
```

2. Система управления (Control System)

Система управления должна содержать регулятор с пропорциональным и интегральным законом управления.

$$e(t) = fb(t) - ref(t)$$

$$c(t) = k_p e(t) + k_i \int_0^t e(t) dt ,$$

где

$fb(t)$ - сигнал обратной связи,

$ref(t)$ - входной сигнал,

$c(t)$ - сигнал управления,

$e(t)$ - сигнал ошибки,

k_p - коэффициент усиления пропорциональной составляющей,

k_i - коэффициент усиления интегральной составляющей;

Данный регулятор - пример модуля пилотажного уровня. Необходимо реализовать функцию, которая по входному сигналу, сигналу обратной связи и параметрам регулятора рассчитывает сигнал управления.

В качестве аргументов приложение принимает входной порт, выходной порт, порт логгера, значение входного сигнала, коэффициенты регулятора и период итерирования:

```
control_system -i [in_port] -o [out_port] -l [log_port] --ref  
[reference_value] --kp [prop] --ki [integ] -t [period]
```

пример:

```
control_system -i 8005 -o 8004 -l 8006 --ref 10 --kp 1 --ki 0.1 -t 0.01
```

3. Система записи телеметрической информации (Logger)

Система записи телеметрической информации принимает данные от системы управления и записывает их в файл в csv формате. Пример содержания csv файла показан ниже:

```
math_in    math_out  control_in  control_fb  control_out  
5          0.3     10          0.3         5
```

Разделители - символ табуляции \t, перенос на новую строку - \n (LF).

Структура телеметрического кадра должна содержать:

- входной сигнал математической модели,
- выходной сигнал математической модели,
- входной сигнал системы управления,
- сигнал обратной связи,
- сигнал управления;

В качестве аргументов приложение принимает входной порт и путь к файлу, в который нужно записывать данные.

```
logger -i [in_port] -p [path]
```

пример:

```
logger -i 8006 -p /home/user/log.csv
```