

# PID Simulator (C++)

## General Requirements:

- Use Linux/GNU toolchain
- Use the make command to build (create Makefiles)
- Use C++
- Demonstrate good software organization and OOP techniques

Be sure to...

- state any assumptions that were made.
- "defend" your implementation as you see fit.

## Part 1

Write a C++ library that simulates a PID controller. The library should provide the following interfaces:

- Interface for the user to configure desired PID gains
- Interface for the user to provide an abstracted plant model (M)
- Interface for the user to retrieve the current *control signal* value at any moment.
- Interface for the user to retrieve the current *feedback signal* value at any moment.
- Interface for the user to specify a low pass filter frequency for the "D" signal in Hz.

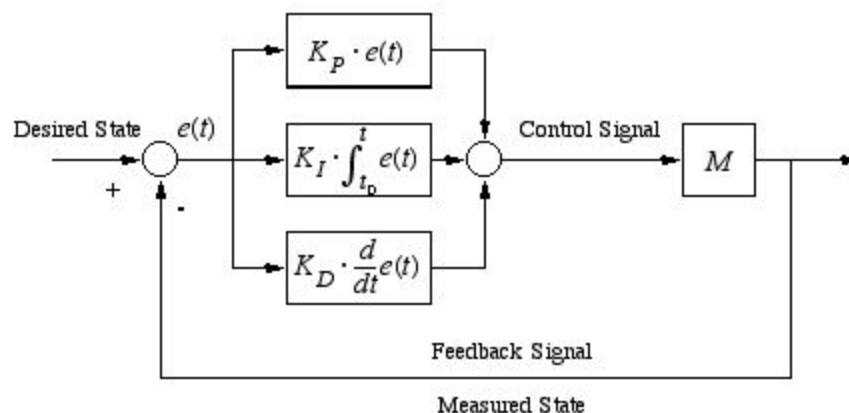
In addition, the library should allow the user to specify a sampling frequency with a minimum of 10 Hz.

## Part 2

Using the library implemented in part 1, write a program that demonstrates usage of the library.

## Requirements

- The program should output results of a 5+ second trial run.
- The output should be in CSV format (other output formats are optional).
- The output should include the error signal, the control signal, and the feedback signal.
- The plant model should have some form of randomized variance injected into it.



(Source: <http://cs.brown.edu/~tld/courses/cs148/02/sensors.html>)

# IP Packet Analysis

Write a function in C (C99) with the following signature:

```
bool is_local_address(uint8_t *ipv4_header, uint32_t address, uint32_t subnet_mask);
```

This function should return true if both of the following conditions are true:

- The 'address' is part of the same local network as the source address provided in the header (use the 'subnet\_mask' argument to determine this).
- The checksum in 'ipv4\_header' is valid.

The byte buffer 'ipv4\_header' represents an Internet Protocol header as defined in RFC 791 (<http://www.ietf.org/rfc/rfc791.txt>)

The 'address' and 'subnet\_mask' arguments are in network byte order.

Next, write a program that demonstrates correct implementation of 'is\_local\_address'.

Be sure to...

- state any assumptions that were made.
- "defend" your implementation as you see fit.

Do not link to any external libraries.

# Firmware Programming

Describe at least three guidelines to follow while writing firmware for a microcontroller that do not apply to other types of software development.

For each of the guidelines be sure to...

- Explain why the guideline is important.
- Give an example of what could go wrong if the guideline is broken.