Closed Loop Systems - Final Project

Tiernan Cuesta and Chris Amling
Rowan University

May 9, 2019

1 Design Overview

The main goal of this project is to apply system and control theory to appropriately and accurately compensate the step response of a plant using negative feedback loop design and a mixed signal design. The system or plant is a voltage source regulator that is supplying a certain load with current. Due to high current the system temperature must be regulated via a cooling system. With a small fan as the cooling system a PWM signal is used to maintain the set point. The duty cycle is based on the proportional, integral and derivative terms of the error signal. MatLab Simulink was helpful for calculating PID gain coefficients to optimized system specifications. An image of the assembled project can be seen in Fig.1.

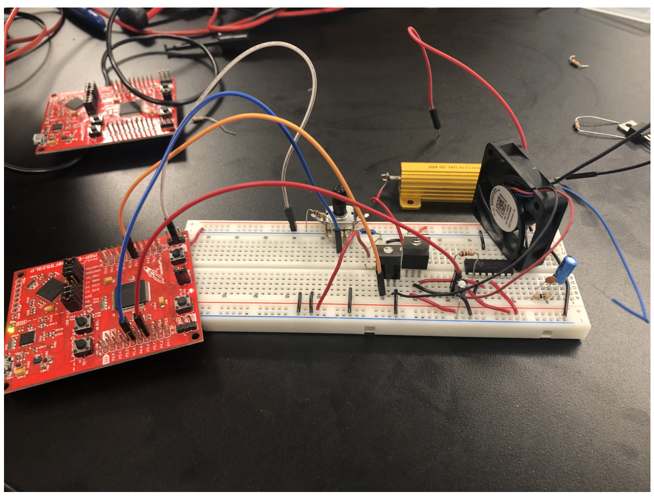

Figure 1: Fully-assembled system under test.

1.1 Design Features

These are the design features of the closed loop system:

- Serial Communication
- Multi-channel, multi-port analog to digital conversion sampling
- Dual low-pass filter design
- PID to PWM motor control

1.2 Featured Applications

These are some featured applications that the Control Feedback loop system could plausibly support:

- HVAC systems
- Motor/Generator temperature control
- Computer cooling systems
- Safety shutdown systems

1.3 Design Resources

This is a quicklink to the program files for the project:

`https://github.com/cuestat7/SaC_finalproject/tree/master/SAC_final_project`

1.4 Block Diagram

This very high level diagram explicitly shows that the system requires two converter subsystems where one converter is an output and another is an input to the micro-controller. This feedback loop is important for the overall design of the system since the temperature regulation of the system depends on it's current state.

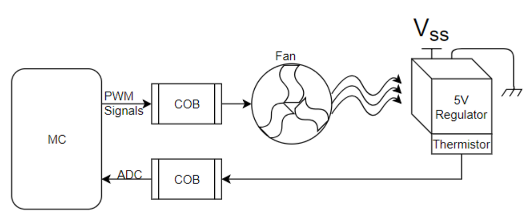

Figure 2: High-level block diagram of the closed loop system.

1.5 Board Image

In the figure below is the system design under test.

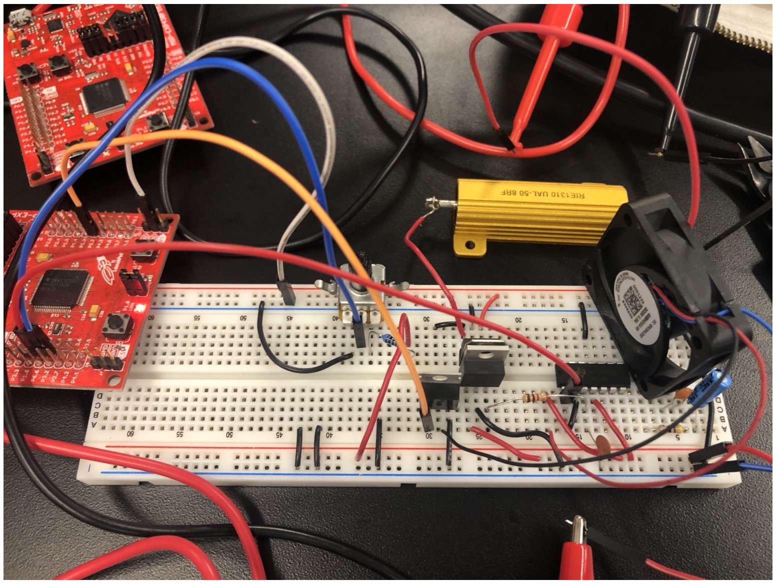

Figure 3: Photo of the protoboard setup.

2 Key System Specifications

PARAMETERS	SPECIFICATIONS	DETAILS
Set Point	30-70°C	Desired temperature of system
True Point	20-80°C	Recorded temperature of system
Fan Speed	0-100% duty cycle	PWM control of the fan, cooling system
Voltage Max	12V	Expected max plant voltage
Current Max	1.0A	Current draw from plant and cooling system
Rise Time	10s	The average rise time observed during testing
Error	±1°C	Difference in current Sys. temp and desired Sys. temp

Table 1: Various parameter types required by the system

3 System Description

The plant being compensated consists of a voltage source regulator that is supplied with 12 volts and has a load of 8Ω drawing on average 0.6A. A schematic drawing of the plant being controlled is in Fig.4. The regulator is the component that must be maintained at a desired temperature. There are no inputs or outputs from/to the microcontroller for this circuit.

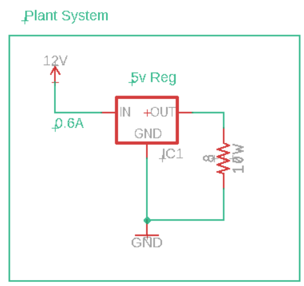

Figure 4: System that requires compensation

For the sensor feedback loop a PTAT is used to measure the temperature of regulator. That temperature is read as a voltage by ADC12 after being filtered through a buffered RC low-pass filter with a cut-off frequency of about 100Hz. The circuit diagram for this section of the control system is in Fig.5.

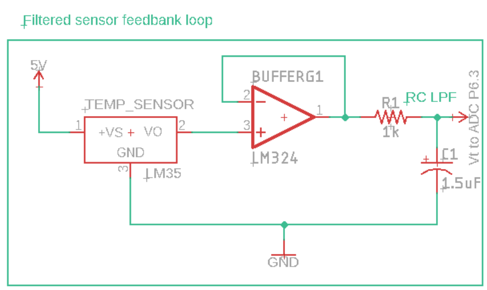

Figure 5: Feedback loop with signal conditioning

Next, a digital average filter further conditions the signal and that signal is converted to a temperature by some gain. A set point is generated using a rotary potentiometer as a voltage source that is again converted to a temperature by a tuned gain. Once, a new set point is recorded the PID controller sends PWM to a power NMOS transistor that acts as a low-side switch for the cooling fan to achieve an appropriate duty cycle. The circuit diagram is observed below in Fig.6

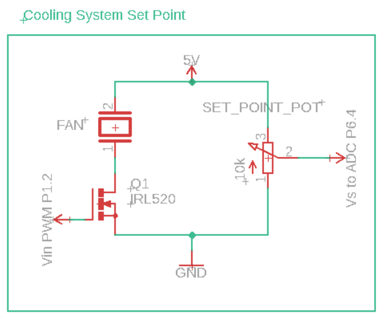

Figure 6: Cooling System and set-point

3.1 Highlighted Devices

- MSP430F5529: The microprocessor contains the protocols for the ADC, PWM, and UART for the fan speed and temperature conversions for the set point and current system temperatures.
- 5V 1A Linear Voltage-Reg: A voltage regulator is a system designed to automatically maintain a constant voltage level. The voltage regulator emulated the plant under compensation since it was a reliable heat source.
- 50W 8Ω Power Resistor: The power resistor was used to draw current from the voltage regulator to dissipate power in the form of heat.
- IRL520 Power NMOS: The IRL520 is a Power fet typically used for low-side switching. It was used to produce a duty cycle that could supply enough current for the fan.
- LM35: The LM35 is a linear temperature sensor used to measure the current state of the system.

3.2 Device/IC 1

MSP430F5529 Microprocessor:

The MSP430F5529, not seen in Fig. 4,5, or 6, role was to perform PID control algorithms based on a desired set point temperature and current temperature of the system. In addition to an analog filter, a moving average filter was implemented on the MSP to further condition the signal. This device has two inputs and one output. ADC12 P6.3 and P6.4 are inputs that record the current and set point temperature respectively. The output is a PWM signal that has a duty cycle based on a PID control algorithm.

4 SYSTEM DESIGN THEORY

The open loop system is characterized by a first-order system, represented by a tf (transfer function) of the form $\frac{b}{s+a}$. The open loop system was characterized by applying a known duty cycle to the fan and measuring the steady-state temperature for that PWM input. PWM and temperature were converted to a voltage to find the aforementioned tf in Fig.7 below.

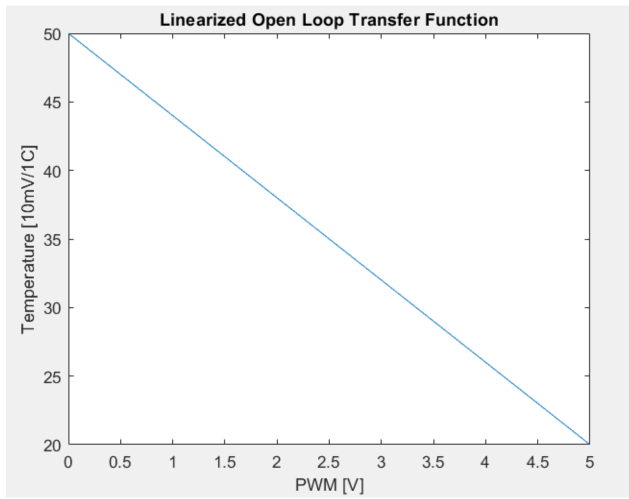

Figure 7: Characterization of the Open Loop system

4.1 Control Theory Design

The control system design is based upon a typical non-unity gain negative feedback loop of a PID compensated first-order system discussed in the previous section. Some noise is observed at the feedback loop system some high frequencies are introduced by the fan's motor. The sensor feedback and the noise is filtered through LPF with a certain gain back to the PID controller, or the microcontroller in this case.

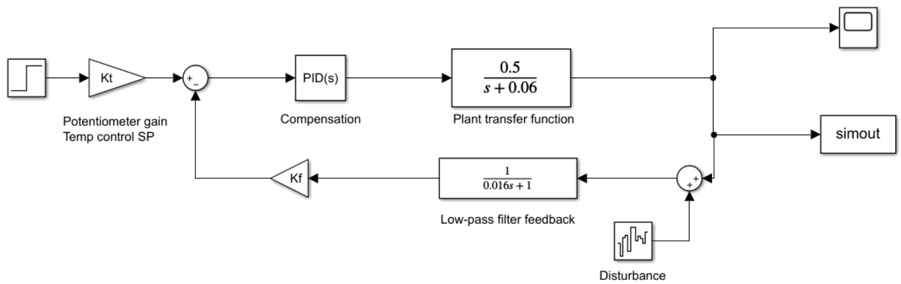

Figure 8: System that requires compensation

Utilizing the Simulink PID tuner the coefficients K_p,K_i, and K_d were approximated to 0.4816, 0.1208, and 0.1208 respectively. For these coefficients to be useful the coefficient were scaled to accommodate the on-board timer specifications of the MC that handled duty cycle calculations. The only way to know if the coefficients were appropriate was to put the control system under test. Several iterations were done with varying scaled coefficients. It was realized that higher integral terms caused the controller to saturate if there was a bump in the system. Also, if the K_p was too low the fan wouldn't pass the dead-zone even though there was an error in the system set point. The derivative term seemed to work well when the magnitude was larger than integral term. The final values that seemed to produce the best results were $K_p = 25$, $K_i = 0.0408$, and $K_d = 0.2416$.

5 Code Review

5.1 Initialization

The backbone of the program begins with configuring the required system features. This includes a multi-channel ADC, UART serial communication capabilities, and timer set-up for duty cycle generation. The multi-channel ADC configuration used supports up to 4 ADC ports, the script for this can be seen below.

```
void configureADC()
{
    P6SEL = 0x1F;          // Enable A/D channel inputs
    ADC12CTL0 = ADC12ON+ADC12MSC+ADC12SHT0_2; // Set sampling time
    ADC12CTL1 = ADC12SHP+ADC12CONSEQ_1;  // Multi-channel single sequence
    ADC12MCTL0 = ADC12INCH_0;            // ref+=AVcc, channel = A0
    ADC12MCTL1 = ADC12INCH_1;            // ref+=AVcc, channel = A1
    ADC12MCTL2 = ADC12INCH_2;            // ref+=AVcc, channel = A2
    ADC12MCTL3 = ADC12INCH_3;            // ref+=AVcc, channel = A3
    ADC12MCTL4 = ADC12INCH_4 + ADC12EOS; // ref+=AVcc, channel = A4, end se
    ADC12IE = 0x10;                          // Enable ADC12IFG.4
    ADC12CTL0 |= ADC12ENC;                   // Enable conversions
```

}

The ADC configuration initializes 4 usable ADC12MEM, sampling at 16 clock cycles per sample. In other words, the converter is sampling every 0.025 seconds which is the period used for I and D controller terms. A timer setup was used to produced a PWM signal. To initialize this configuration the code in the listing below describes the on-timer specifications.

```
{
    P1DIR |= BIT2;  // Sets P1.2 as output driver for pwm
    P1SEL |= BIT2;  // Selects the port 1.2 as the timer A output
    TA0CTL = TASSEL_1 | MC_1 | TACLR;  // Sets timerA_0 to SMCLK, up-mode
    TA0CCR0 = 255;  // Sets CCR0 max pwm
    TA0CCR1 = 0;    // Sets CCR1 to initial value of 0% Duty Cycle
    TA0CCTL1 = OUTMOD_7;  // Output mode 7 reset/set
}
```

This specific timer set-up uses a 1 MHz clock to generate a PWM signal for a 0-100 duty cycle range base on the current value of CCR1, which is updated by the PID algorithm. Since an NMOS transistor is used for a low-side switch a reset/set toggle is used for the timer to emulate the proper PWM signal for fan speed control. Furthermore, for testing purposes and data acquisition a UART configuration was initialized to easily read the system response. The listing below shows the UART setup for the F5529.

```
{
void configureUART()
    P1DIR |= BIT0;
    P1OUT &= ~BIT0;

    P4SEL |= BIT5;              // Enables RX and TX buffer
    P4SEL |= BIT4;
    UCA1CTL1 |= UCSWRST;        // Clears the UART control register 1
    UCA1CTL1 |= UCSSEL_2;       // Sets SMCLK
    UCA1BR0 = 104;              // For baud rate of 9600
    UCA1BR1 = 0;                // For baud rate of 9600

    UCA1MCTL |= UCBRS_2;        // set modulation pattern to high on bit 1 & 5
    UCA1CTL1 &= ~UCSWRST;       // initialize USCI
    UCA1IE |= UCRXIE;           // enable USCI_A1 RX interrupt
    UCA1IFG &= ~UCRXIFG;        // clears interrupt flags

}
```

The UART configuration above supports Firmware serial communication with a baud rate of 9600.

5.2 Sensor Acquisition

The previously discussed ADC12 was used to record the voltage output of the temperature sensor, V_t. Since the ADC converts the voltage into a binary number the value must be converted to a readable decimal value. This conversion is done not only to the sensor output voltage but also for the set point, or the voltage output of the rotary potentiometer.

5.2.1 Signal Conditioning

Due to high frequency disturbances introduced by the fan the LM35's output signal must be conditioned and filtered. In this project a dual filter designed was implemented. First, the voltage output V_t was passed through an RC filter, converted by the ADC12, and passed through another simple digital moving average filter to clean the signal up. The averager implemented can be seen in the listing below.

```
{
volatile float Filtervolt;
float LPF_Beta = 0.95; // 0<B<1
Filtervolt = Filtervolt - (LPF_Beta * (Filtervolt - ADC12MEM4));
}
```

Where ADC12MEM4 is the ouput signal of the PTAT after it has passed through the analog filter. The LPF beta value basically is the weight of the difference previous value and the current value. The addition of the moving average filter was pertinent for getting smooth and consistent values from the sensor. Otherwise, the current temperature reading would bounce all over causing the PID controller to not work properly.

5.3 Error Calculation

The calculation of the error signal is easy to do after the signals have been converted and conditioned. After every sample a new error is recorded and the last error is saved for further calculations.

5.4 Control Calculation

The PID controller implemented for this project follows the fundamentals of Proportional Integral and Derivative control algorithms.

```
{
    Proportion = Kp * Error;
    Integral += Ki * (Error * 0.025);
    Derivative = Kd * ((Lasterror - Error)/0.025);

    PID = (Proportion + Integral + Derivative);
```

```
    if (PID > 255)  PID = 255;
    if (PID < 0)    PID = 0;
    else if (0 < PID < 255) PID = PID;
}
```

The PID controller is also clamped at the max and min values of the CCR0 timer specifications. This ensures that the PID output will not go above saturation or below 0.The 0.025 term in both the integral and derivative controller is the period between each sample, since those two terms are dependant on time.

5.5 Actuation

Subsequently, after the PID algorithm performs the calculations it sends it's new value to CCR1, which adjusts the new duty cycle required for the fan to heat/cool the system. The new required duty cycle is updated every time the ADC sample ticks.

5.6 Bill of Materials

- 1 - 10kΩ 1/4W resistor
- 1 - 8Ω 50W Power Resistor
- 1 - LM324 Operational Amplifier
- 1 - IRL520 Power-fet NMOS
- 1 - LM35 Temperature Sensor
- 1 - MSP430F5529 Launchpad kit
- 1 - 5V DC brushless fan
- 1 - 5V 1A Voltage Regulator

6 Appendix

6.1 Appendices A

Below is a quick link to the datasheet of the LM35, PTAT, used in this project.
`http://www.ti.com/lit/ds/symlink/lm35.pdf`

6.2 Appendices B

Below is a quick link to the datasheet of the switching transistor used in this project.
`https://www.vishay.com/docs/91298/91298.pdf`

6.3 Appendices C

Below is a quick link to the users guide for the MSP430F5529 launchpad.
`http://dev.ti.com/tirex/#/DevTool/MSP-EXP430F5529/?link=Device20Documentation2FMSP430F`
`6XX2FMSP430F55292FUsers20Guide`