ME 305 Mechatronics

Department of Mechanical Engineering

Cal Poly

Fall Quarter -2022

Nathan Burtnett, Tom Taylor

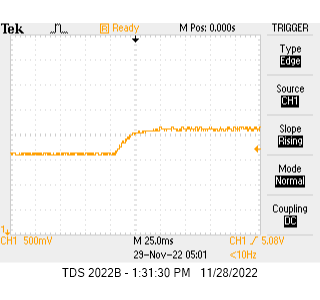
**Lab 5 Report**

# Objective and Overview

The goal of this lab is to integrate a keypad for user entry, an LCD for user display, interrupts to update variables at a set interval, a PI controller to adjust motor voltage and flywheel speed, and PWM to adjust motor voltage accordingly. The user is able to enter a desired angular velocity, proportional, and integral gain. The motor is able to be run with either open or closed loop control. The LCD is updated regularly to display the actual and desired velocity, the difference between the two, motor effort, KI and KP, open or closed loop mode, and display update on or off.

# Parameterization

## Tables and Figures



**Figure 1:** Oscilloscope Capture

**Table 1:** Open Loop Data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Kp,in** | **Kp** | **Vref** | **Vact** | **Vmin** | **Vmax** | **Vτ** | **τ** |
| 750 | 0.73 | 100 | 15 | 5 | 5.56 | 5.35 | 24 |
| 200 | 33 | 5 | 6.06 | 5.67 | 12.1 |
| 300 | 50 | 5 | 6.62 | 6.02 | 12.2 |
| 400 | 67 | 5 | 7.18 | 6.37 | 13 |
| 500 | 84 | 5 | 7.7 | 6.70 | 14.4 |
| 1250 | 1.22 | 100 | 27 | 5 | 5.82 | 5.52 | 11.6 |
| 200 | 55 | 5 | 6.76 | 6.11 | 10.4 |
| 300 | 84 | 5 | 7.68 | 6.69 | 14.4 |
| 400 | 112 | 5 | 8.56 | 7.24 | 12.8 |
| 500 | 141 | 5 | 9.52 | 7.85 | 12.6 |
| 1750 | 1.71 | 100 | 38 | 5 | 6.22 | 5.77 | 12 |
| 200 | 78 | 5 | 7.48 | 6.56 | 12.8 |
| 300 | 118 | 5 | 8.8 | 7.39 | 14 |
| 400 | 145 | 5 | 9.67 | 7.94 | 14.8 |
| 500 | 160 | 5 | 9.71 | 7.97 | 12.4 |

**Table 2**: Open Loop Results

|  |  |  |  |
| --- | --- | --- | --- |
| **Kp** | 0.73 | 1.22 | 1.71 |
| **Vref/Vact** | 0.17 | 0.285 | 0.311 |
| **K** | 0.23 | 0.23 | 0.18 |
| **Km** | 9.60 | 9.54 | 7.44 |
| **τ (ms)** | 15.14 | 12.36 | 13.2 |

**Figure 2**: Open Loop Vact vs. Vref

**Table 3**: Closed Loop Data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Kp,in** | **Kp** | **Vref** | **Vact** | **Vmin** | **Vmax** | **Vτ** | **τ** |
| 750 | 0.73 | 100 | 13 | 5 | 5.4 | 5.25 | 8.5 |
| 200 | 28 | 5 | 5.8 | 5.50 | 9.5 |
| 300 | 43 | 5 | 6.3 | 5.82 | 8.4 |
| 400 | 58 | 5 | 6.8 | 6.13 | 9.9 |
| 500 | 72 | 5 | 7.32 | 6.46 | 10 |
| 1250 | 1.22 | 100 | 21 | 5 | 5.66 | 5.42 | 8.9 |
| 200 | 43 | 5 | 6.36 | 5.86 | 8.6 |
| 300 | 66 | 5 | 7.06 | 6.30 | 10 |
| 400 | 87 | 5 | 7.84 | 6.79 | 10.1 |
| 500 | 109 | 5 | 8.52 | 7.22 | 10.1 |
| 1750 | 1.71 | 100 | 27 | 5 | 5.84 | 5.53 | 8.6 |
| 200 | 56 | 5 | 6.76 | 6.11 | 6.3 |
| 300 | 85 | 5 | 7.68 | 6.69 | 7.3 |
| 400 | 113 | 5 | 8.6 | 7.27 | 12 |
| 500 | 142 | 5 | 9.56 | 7.87 | 12.4 |

**Table 4**: Closed Loop Results

|  |  |  |  |
| --- | --- | --- | --- |
| **Kp** | 0.73 | 1.22 | 1.71 |
| **Vref/Vact** | 0.15 | 0.22 | 0.287 |
| **K** | 0.20 | 0.18 | 0.17 |
| **Km** | 8.26 | 7.37 | 6.87 |
| **τ (ms)** | 9.26 | 9.54 | 9.32 |

**Figure 3**: Closed Loop Vact vs. Vref

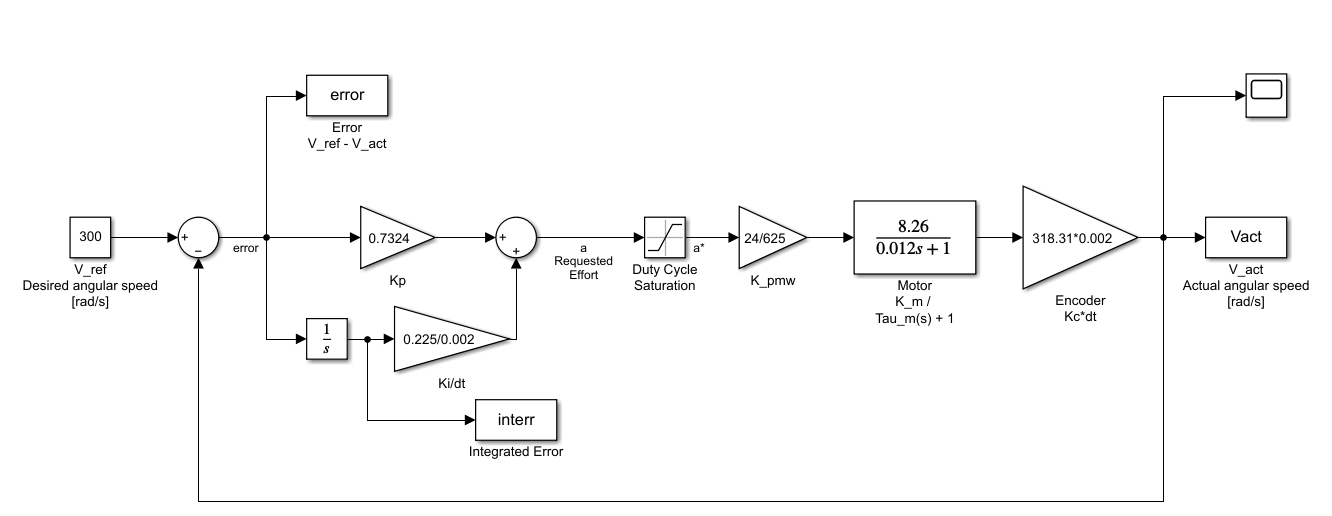
**Table 5**: Key Results

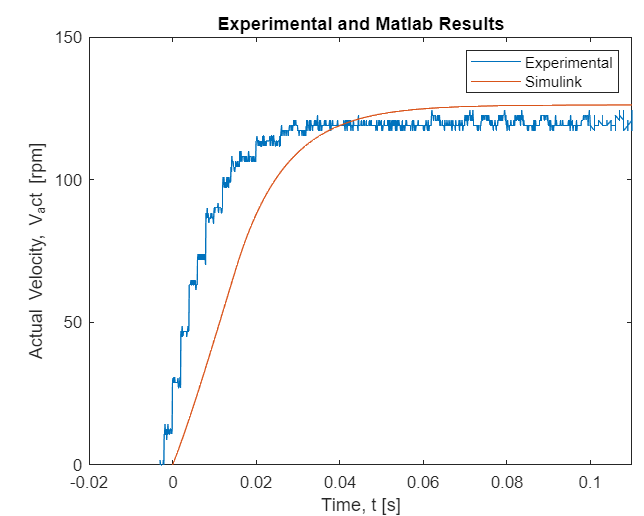
|  |  |
| --- | --- |
| Parameterization | |
| **Km** | 8.18 |
| **τ (ms)** | 11.47 |
| Nominal Gains | |
| **ωn** | 44.3 |
| **ζ** | 1.12 |
| Revised Gains | |
| **Ki** | 0.63 |
| **Kp** | 3.05 |

## Constants

## Calculations

# 2nd Order Model





# Variables

|  |  |
| --- | --- |
| Run | Boolean flag for motor on or off |
| Cl | Boolean flag for open or closed loop |
| V\_act | Actual velocity |
| V\_ref | Desired velocity |
| Theta\_OLD | Previous encoder reading |
| Theta\_NEW | New encoder reading |
| KP | Proportional gain |
| KPRES | Proportional gain result |
| KI | Integral gain |
| KIRES | Integral gain result |
| APRE | Raw PI Controller output |
| ASTAR | Saturation checked PI Controller output |
| ERROR | Current error in system |
| ESUM | Reiman sum of errors |
| EFF | Current motor effort |
| ERR | Current motor error |
| TEMP | Reserved space for saturated addition |
| UPDATE\_COUNT | Flag for count to update |
| UPDATE\_FLG1 | Flag for update display line 1 |

# Code

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* Lab 5 Code [includes LibV2.2] \*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\* Summary: \*

;\* - This is code that runs the controls for a PI controlled PWM motor \*

;\* \*

;\* Author: Nathan Burtnett, Tom Taylor \*

;\* Cal Poly University \*

;\* Spring 2022 \*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;/------------------------------------------------------------------------------------\

;| Include all associated files |

;\------------------------------------------------------------------------------------/

; The following are external files to be included during assembly

XDEF main

XDEF Theta\_OLD, RUN, CL, V\_ref, KP, KI, UPDATE\_FLG1

;/------------------------------------------------------------------------------------\

;| External References |

;\------------------------------------------------------------------------------------/

; All labels from other files must have an external reference

XREF ENABLE\_MOTOR, DISABLE\_MOTOR

XREF STARTUP\_MOTOR, UPDATE\_MOTOR, CURRENT\_MOTOR

XREF STARTUP\_PWM, STARTUP\_ATD0, STARTUP\_ATD1

XREF OUTDACA, OUTDACB

XREF STARTUP\_ENCODER, READ\_ENCODER

XREF DELAY\_MILLI, DELAY\_MICRO

XREF INITLCD, SETADDR, GETADDR, CURSOR\_ON, DISP\_OFF

XREF OUTCHAR, OUTCHAR\_AT, OUTSTRING, OUTSTRING\_AT

XREF INITKEY, LKEY\_FLG, GETCHAR

XREF LCDTEMPLATE, UPDATELCD\_L1, UPDATELCD\_L2

XREF LVREF\_BUF, LVACT\_BUF, LERR\_BUF,LEFF\_BUF, LKP\_BUF, LKI\_BUF

XREF Entry, ISR\_KEYPAD

XREF V\_act\_DISP, ERR\_DISP, EFF\_DISP

XREF FREDENTRY

;/------------------------------------------------------------------------------------\

;| Assembler Equates |

;\------------------------------------------------------------------------------------/

TFLG1 EQU $004E

TC0 EQU $0050

C0F EQU %00000001 ; timer channel 0 output compare bit

PORTT EQU $0240 ; PORTT pin 8 to be used for interrupt timing

LOWER\_LIM EQU -625 ; number for max reverse duty cycle

UPPER\_LIM EQU 625 ; number for max forward duty cycle

INTERVAL EQU $4E20 ; number of clock pulses that equal 2ms from 10.2MHz clock

;/------------------------------------------------------------------------------------\

;| Variables in RAM |

;\------------------------------------------------------------------------------------/

DEFAULT\_RAM: SECTION

RUN: DS.B 1 ; Boolean indicating controller is running

CL: DS.B 1 ; Boolean for closed-loop active

V\_act: DS.W 1 ; read velocity

V\_ref: DS.W 1 ; reference velocity

Theta\_OLD: DS.W 1 ; previous encoder reading

Theta\_NEW: DS.W 1 ; new encoder reading

KP: DS.W 1 ; proportional gain

KPRES: DS.W 1 ; proportional gain result

KI: DS.W 1 ; integral gain

KIRES: DS.W 1 ; integral gain result

APRE DS.W 1 ; pi controller output

ASTAR DS.W 1 ; saturation checked pi controller output

ERROR: DS.W 1 ; the current error of the system

ESUM: DS.W 1 ; Reiman sum of errors

EFF: DS.W 1 ; current effort of motor

ERR: DS.W 1 ; current error of motor

TEMP: DS.W 1 ; a temp variable used in saturated addition

UPDATE\_COUNT: DS.B 1 ; a flag for updating the count

UPDATE\_FLG1 DS.B 1 ; Boolean for display update for line one

;/------------------------------------------------------------------------------------\

;| Main Program Code |

;\------------------------------------------------------------------------------------/

MyCode: SECTION

main:

;CLEAR VARIABLES

clr RUN

clr CL

clr UPDATE\_FLG1

clr UPDATE\_COUNT

clrw V\_act

clrw V\_ref

clrw Theta\_NEW

clrw Theta\_OLD

clrw KP

clrw KPRES

clrw KI

clrw KPRES

clrw ESUM

clrw ERR

clrw EFF

clrw APRE

clrw ASTAR

jsr FREDENTRY ; good ol' fred

spin: bra spin ; endless horizontal loop

TC0ISR:

bset PORTT, $80 ; turn on PORTT pin 8 to begin ISR timing

inc UPDATE\_COUNT ; unless UPDATE\_COUNT = 0, skip saving

bne measurements ; display variables

movw V\_act, V\_act\_DISP ; take a snapshot of variables to enable

movw ERR, ERR\_DISP ; consistent display

movw EFF, EFF\_DISP

movb #$01, UPDATE\_FLG1 ; set UPDATEFLG1 when appropriate

measurements: ;reads the encoder and updates values

;read encoder value

jsr READ\_ENCODER ;read encoder position

std Theta\_NEW ;store it

;compute 2-point difference to get speed

subd Theta\_OLD ;compute displacement since last reading

std V\_act ;store displacement as actual speed

movw Theta\_NEW, Theta\_OLD ;move current reading to previous reading

olcl: ;tests whether open loop or closed

ldd V\_ref ;loads desired velocity

tst CL ;tests for a closed loop

bne clerr ;if not go to closed loop error

olerr: ;open loop error

std ERR ;store desired velocity into ERR

bra kpcalc ;jump past clerr

clerr: ;closed loop error

subd V\_act ;subtract actual velocity from desired velocity

std ERR ;store the result in ERR

kpcalc: ;compute proportional control

ldd ERR ;load D with error (redundant)

ldy KP ;load Y with Kp (entered by user)

emuls ;multiply acc D and Y, store in D

ldx #$0400 ;load X with HEX $0400 (decimal 1024, DAC resolution)

edivs ;divide D by X, result in Y

sty KPRES ;store result into new KP (actual value used in calcs)

kicalc: ;compute integral control

ldy ESUM ;load Y with sum of errors

ldd ERR ;load D with current error

jsr DSATADD ;add current error to the sum of errors

std ESUM ;stores result in new ESUM

ldy KI ;load Y with Ki (entered by user)

emuls ;multiply Y with D

ldx #$0400 ;load X with HEX $0400 (decimal 1024)

edivs ;divide D by x

sty KIRES ;store result into new KP (actual value used in calcs)

pisum: ;add Kp and Ki

ldd KPRES

ldy KIRES

jsr DSATADD ; saturated addition

std APRE ; pre-saturated motor voltage request

pisatcheck: ;satcheck

jsr SATCHECK

std ASTAR ; store ASTAR from the satcheck

effcalc: ;calculate effort

ldy #$0064 ; decimal 100

emuls ; multiply D x Y -> Y:D

ldx #$0271 ; decimal 625

edivs ; (Y : D) ÷ (X) -> Y; Remainder -> D

sty EFF ; store result into new EFF (actual value used in calcs)

runtst: ;test for run condition and whether motor needs to be set

tst RUN

beq motoroff

ldd ASTAR

bra motorset

motoroff: ;turns motor off

ldd #$0000

clrw ESUM

clrw ERR

motorset: ;set motor speed

jsr UPDATE\_MOTOR

ldd TC0

addd #INTERVAL

std TC0

bset TFLG1, C0F

measureout: ;outdacs the resultant velocity

ldd V\_act

ldy #13

emuls

addd #2048

jsr OUTDACA

rti

;/------------------------------------------------------------------------------------\

;| Subroutines |

;\------------------------------------------------------------------------------------/

SATCHECK:

pshx

pshc

cpd #$0000

bmi NEGATIVE

bgt POSITIVE

bra SATCHECK\_exit

POSITIVE:

cpd #$0271

ble SATCHECK\_exit

ldd #$0271

bra SATCHECK\_exit

NEGATIVE:

cpd #$FD8F

bge SATCHECK\_exit

ldd #$FD8F

SATCHECK\_exit:

pulc

pulx

rts

DSATADD:

pshx ;pushes x to stack

pshc ;pushes c to stack

sty TEMP ;stores acc y value in TEMP

addd TEMP ;adds acc D to TEMP

bvs DSATADD\_OF ;if Overflow is set then branch, otherwise just fall to exit

DSATADD\_exit:

pulc ;puls c from stack

pulx ;puls x from stack

rts ;end subroutine

DSATADD\_OF: ;tests overflow

cpy #$0000 ;compares TEMP with hex 0

bmi DSATADD\_OFN ;branches if value is negative

ldd #$7FFF ;loads acc D with positive overflow

bra DSATADD\_exit ;goes to end subroutine

DSATADD\_OFN: ;sets negative overflow

ldd #$8000 ;loads acc D with negative overflow

bra DSATADD\_exit ;goes to end subroutine

;/------------------------------------------------------------------------------------\

;| Vectors |

;\------------------------------------------------------------------------------------/

; Add interrupt and reset vectors here

ORG $FFEE ; Timer channel 0 vector address

DC.W TC0ISR

ORG $FFFE ; reset vector address

DC.W Entry