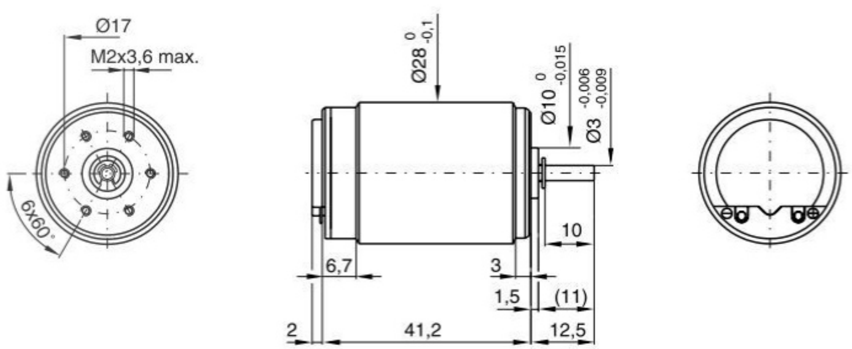
# SPEED OR POSITION CONTROL ON A DC BRUSHED MOTOR WITH A PARALLEL STRUCTURED PID ISSUED FROM Z-TRANSFORM

## 0-HOW TO CHOOSE THE GOOD SAMPLING TIME Ts AND THE PWM FREQUENCY?

You need to know some electrical and mechanical features on the motor. Here are some datasheets hard to find on the net about the ESCAP 28LT2R-416E

**escap 28LT12**
Graphite/copper commutation system - 9 segments

**D.C. Motor**
**21 Watt**

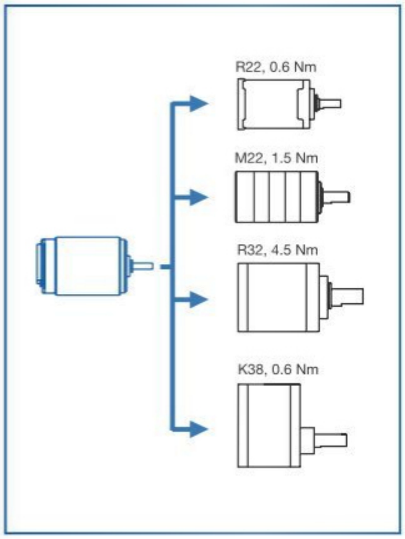



scale: 3:4
dimensions in mm
mass: 135 g

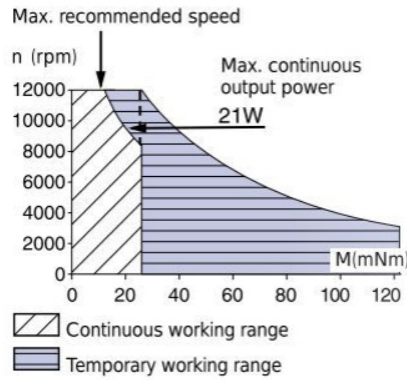
28LT12 ···· • 49

| Winding types | ···· | -219P | -219 | -416E |
|---|---|---|---|---|
| **Measured values** | | | | |
| 1 Measuring voltage | V | 9.0 | 18.0 | 32.0 |
| 2 No-load speed | rpm | 7900 | 7900 | 7400 |
| 3 Stall torque | mNm (oz-in) | 57 (8.1) | 63 (8.86) | 65 (9.26) |
| 4 Average no-load current | mA | 125.0 | 65.0 | 35.0 |
| 5 Typical starting voltage | V | -- | -- | -- |
| **Max. recommended values** | | | | |
| 6 Max. continuous current | A | 2.15 | 1.13 | 0.63 |
| 7 Max. continuous torque | mNm (oz-in) | 21.6 (3.06) | 22.8 (3.23) | 24.2 (3.42) |
| 8 Max. angular acceleration | $10^3$ rad/s$^2$ | 51 | 53 | 34 |
| **Intrinsic parameters** | | | | |
| 9 Back-EMF constant | V/1000 rpm | 1.12 | 2.24 | 4.26 |
| 10 Torque constant | mNm/A (oz-in/A) | 10.7 (1.51) | 21.4 (3.03) | 40.7 (5.76) |
| 11 Terminal resistance | ohm | 1.70 | 6.15 | 19.9 |
| 12 Motor regulation R/k$^2$ | $10^3$/Nms | 15 | 13 | 12 |
| 13 Rotor inductance | mH | 0.10 | 0.50 | 2.40 |
| 14 Rotor inertia | kgm$^2$ . $10^{-7}$ | 10.70 | 10.70 | 17.80 |
| 15 Mechanical time constant | ms | 16 | 14 | 21 |

Availability: see enclosed document at the end of the catalogue




- Thermal resistance:
  rotor-body 5°C/W
  body-ambient 12°C/W
- Thermal time constant - rotor / stator:
  17 s / 760 s
- Max. rated coil temperature: 155°C
- Recom. ambient temperature range:
  -10°C to +80°C (14°F to 176°F)
- Max. axial static force for press-fit: 250 N
- End play: ≤ 150 µm
  Radial play: ≤ 18 µm
  Shaft runout: ≤ 10 µm
- Max. side load at 5 mm from mounting face:
  - sleeve bearings 6 N
  - ball bearings 8 N
- Motor fitted with sleeve bearings
  (ball bearings optional)










## ***Sampling time:***

### 2.5. Choix de la période d'échantillonnage pour les systèmes dynamiques

Pour les cas des systèmes dynamiques, le choix de la période d'échantillonnage '$T$' doit pratiquement satisfaire les intervalles suivants :

$$5f_c \leq f_T \leq 25f_c \quad \Leftrightarrow \quad 2\pi/25\omega_c \leq T \leq 2\pi/5\omega_c \qquad (2.12)$$

où

$f_T$, $f_c$ et $\omega_c$ sont respectivement la fréquence d'échantillonnage, la fréquence de coupure et la pulsation du coupure du système à discrétiser.

- ***Cas d'un système du premier ordre***

Pour un système du premier ordre, la période d'échantillonnage doit pratiquement satisfaire l'intervalle suivant :

$$0.25\tau \leq T \leq 1.25\tau \qquad (2.13)$$

où $\tau$ est la constante de temps du système du premier ordre à discrétiser.

- ***Cas d'un système du second ordre***

Un système du second ordre peut être discrétisé si :

$$\frac{0.25}{w_0} \leq T \leq \frac{1.25}{w_0} \qquad (2.14)$$

où $w_0$ est la pulsation propre du système du second ordre à discrétiser.

Pour un moteur à courant continu: fonction de transfert du second ordre

La fonction de transfert $H(p)$ peut être représentée par :

$$H(p) = \frac{k}{1+\frac{2m}{\omega_0}p+\frac{p^2}{\omega_0^2}} = \frac{k}{\frac{1}{\omega_0^2}(p-p_1)(p-p_2)} = \frac{k}{(1+\tau_1 p)(1+\tau_2 p)}$$

avec $\omega0=\frac{1}{\sqrt{\tau1.\tau2}}=\frac{1}{\sqrt{\tau m.\tau elec}}=\frac{1}{\sqrt{0.021.\frac{0.021}{19.9}}}$=212.43 rad/s

Alors: $\frac{0.25}{\omega0}$ ≤Ts≤ $\frac{1.25}{\omega0}$ ce qui donne: 0.0012ms ≤Ts≤ 0.0059ms

**Conclusions:**

The tests were made for:

-50ms (20Hz), good results, easy to tune, tendences on SerialPlot not so accurate

-20ms (50Hz), good results, easy to tune, tendances on SerialPlot a little chaotic. This sampling time is the one to choose.

-10ms (100Hz), best results, disturbed signal, not easy to tune, tendances more « round », interfere with some functions used in the program (displays on LCD)

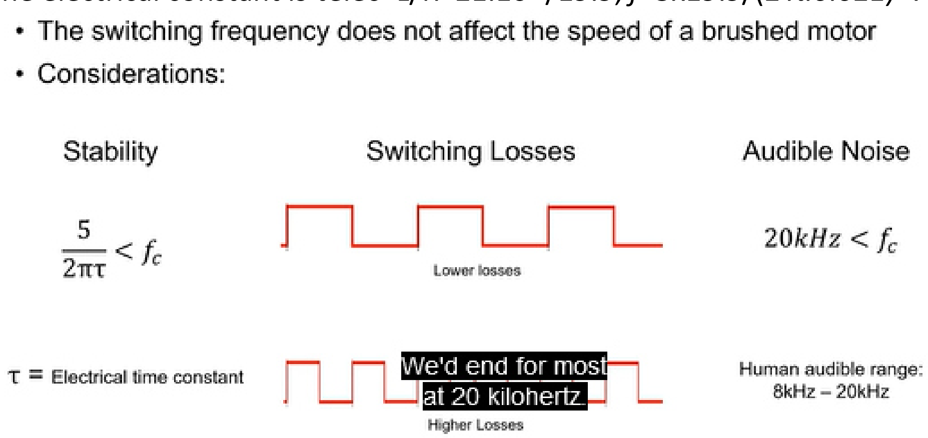
-8ms (125Hz), very good but not easy to tune, signal a little bit disturbed, problem on LCD: speed higher

Under 8ms a lot of problems appears about the LCD display and the H bridge PWM, too many interferences between I2C/timers/PWM : the microcontroller reach its limits, too many calculation is unusefull.

Whatever the sampling time, the response time remains the same but the graphical result is more or less accurate.

## ***PWM frequency:***

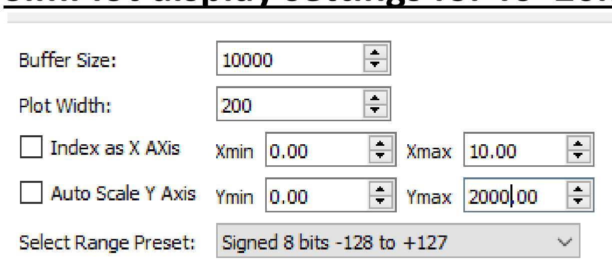
The electrical constant is $\tau elec=L/R=21.10^{-3}/19.9$, $f=5\times19.9/(2\pi.0.021)=754Hz$




In my examples 31250Hz (higher than the human audible range 20Hz-20KHz) and MOSFETs with no heat.

# 1-SPEED CONTROL WITH Ts=20ms, SP=1500tr/mn

## SimPlot display settings for Ts=20ms:




It means 200 samples on the graph and the distance between 2 points is 20ms.

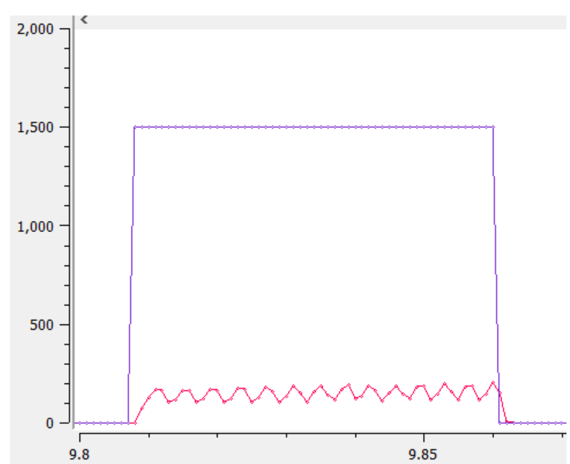
## PID settings examples

Derivative action unusefull because it 's a second order system.

## With Ts=20ms, only Proportional action: Kp =0;09

The overshoot must not exceed 10%


We will decrease Kp after numerous tests to Kp=0.03 and it displays (only proportional action)

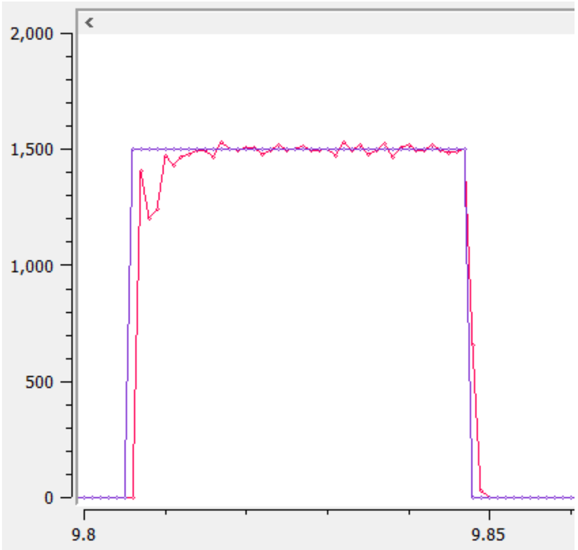



1-Start with Kp=0.9 and try to find the good Ki which gives no error and no oscillation

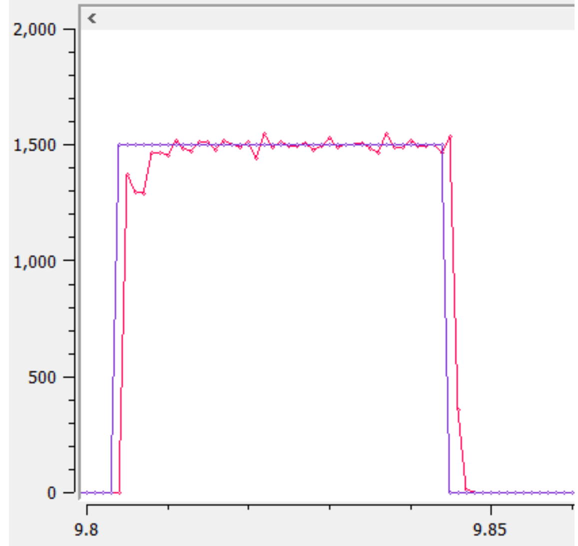
2-Decrease Kp to be as quick as possible with no high overshoot.

It will take time to find the good settings with no calculation.

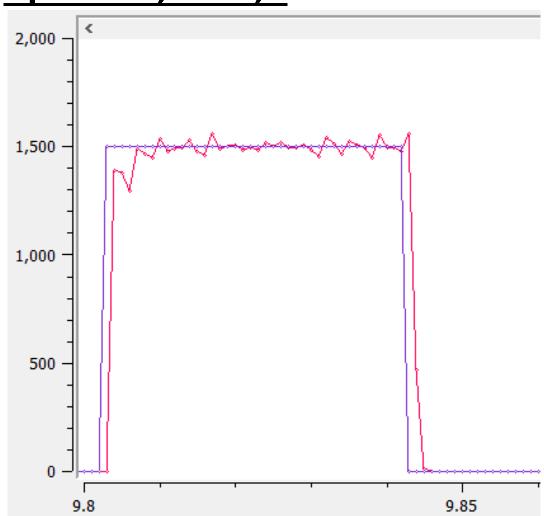
**Kp =0.09, Ki=0,5**



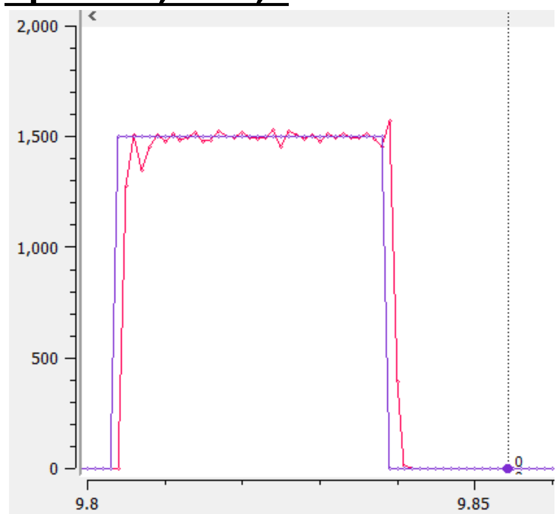

**Kp =0.08, Ki=0,5**



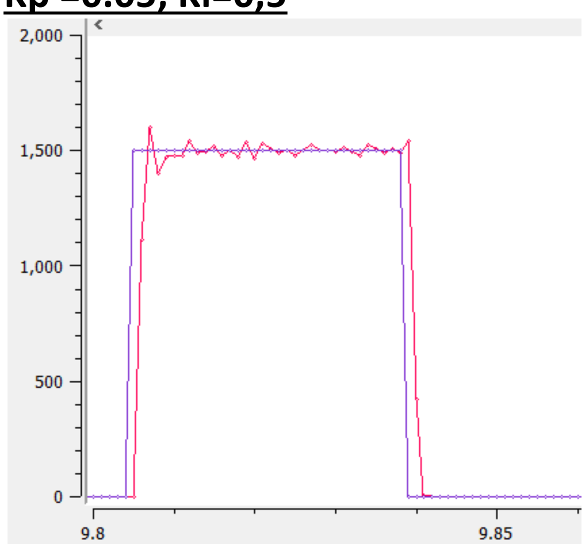

**Kp =0.07, Ki=0,5**



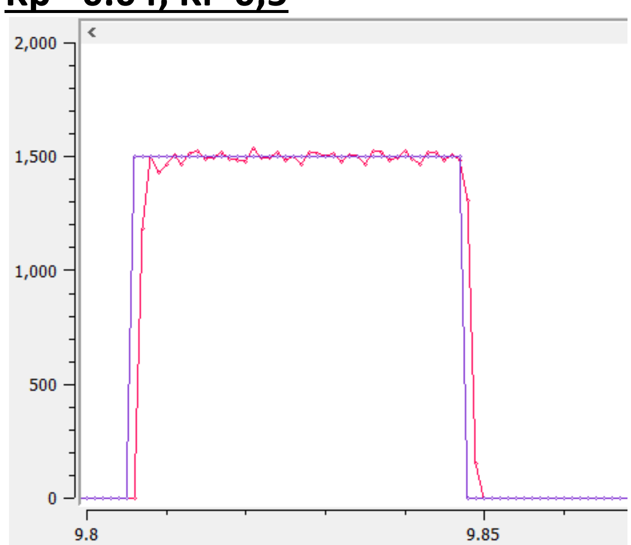

**Kp =0.06, Ki=0,5**



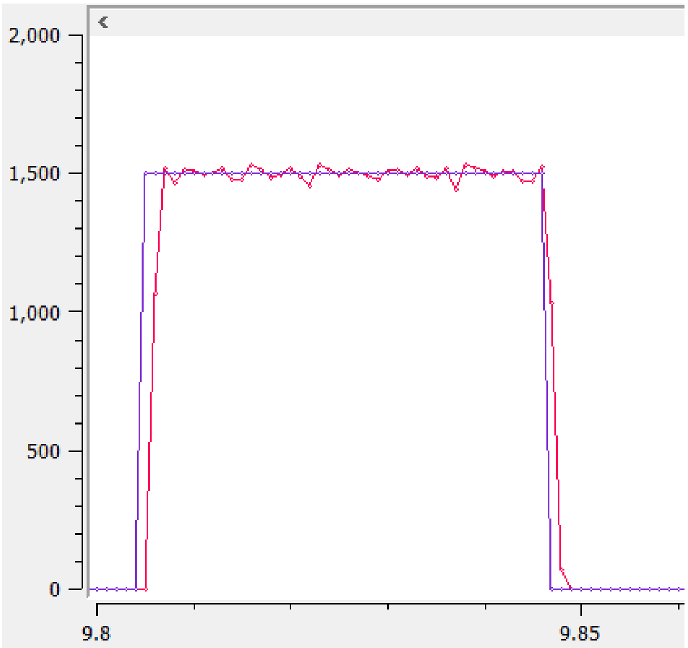

**Kp =0.05, Ki=0,5**



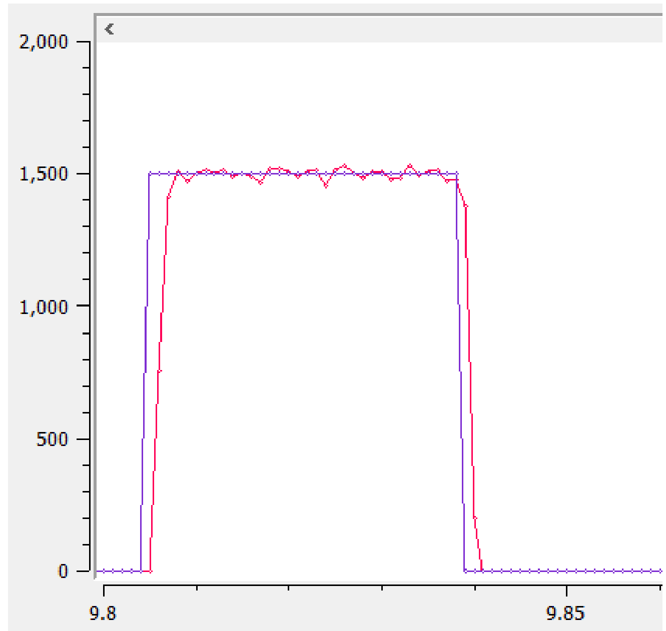

**Kp =0.04, Ki=0,5**



## Kp =0.03, Ki=0,5




## Kp =0.02, Ki=0,6




The best setting is: **Kp =0.03, Ki=0,5** because:

-no static error, accurate

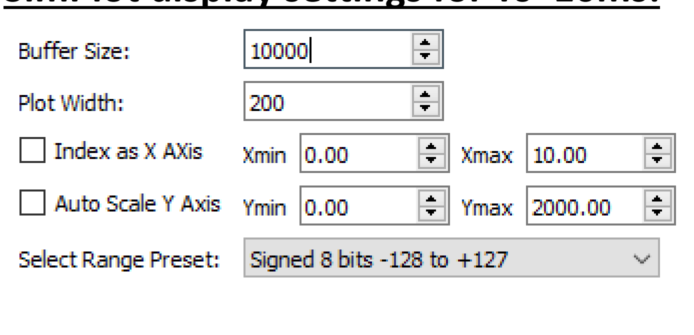
-no overshoot

-the shorter response time

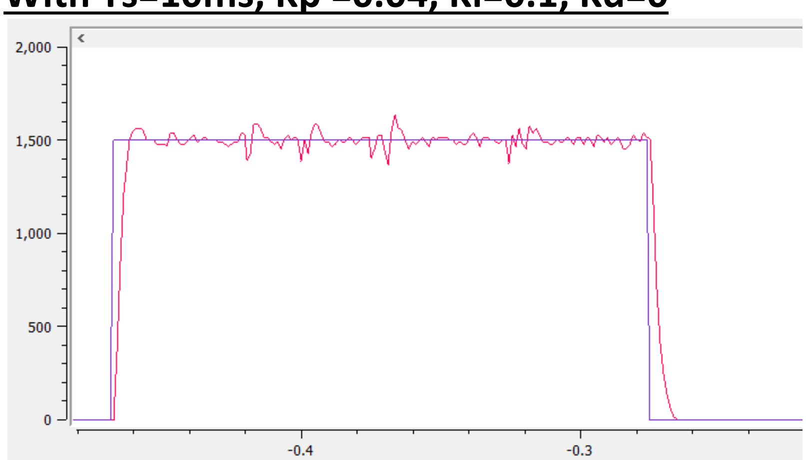
Here, time between 2 points is 20ms, so the response lasts about 50ms

# 2-SPEED CONTROL WITH Ts=10ms, SP=1500tr/mn

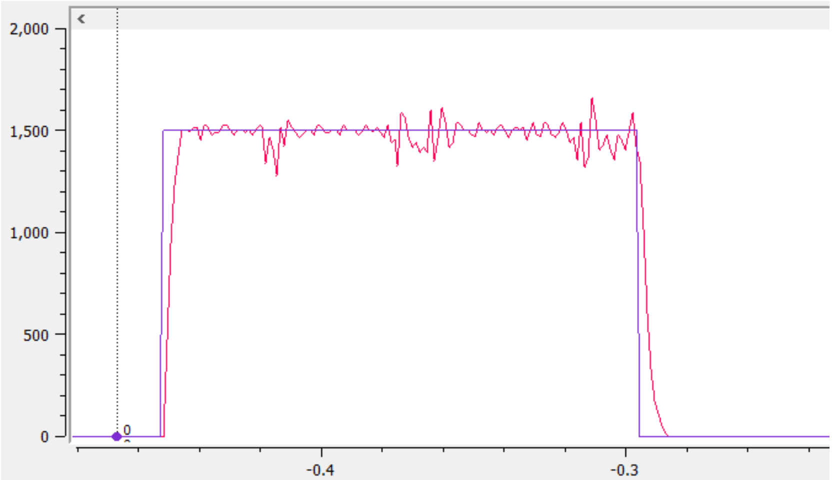
## SimPlot display settings for Ts=10ms:



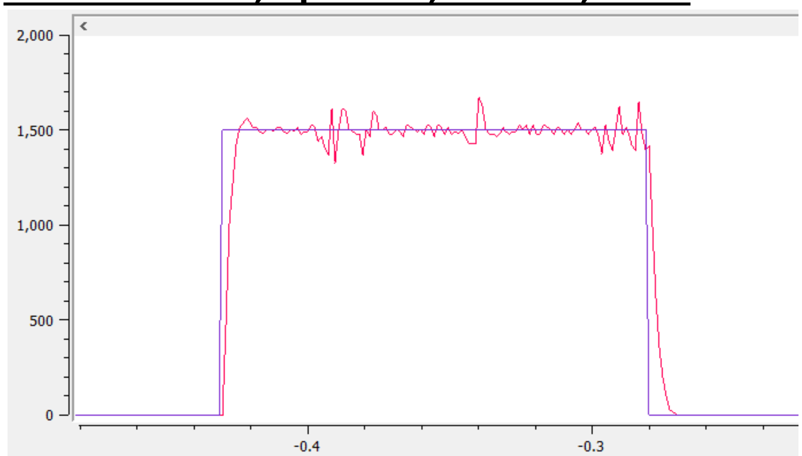

## With Ts=10ms, Kp =0.04, Ki=0.1, Kd=0



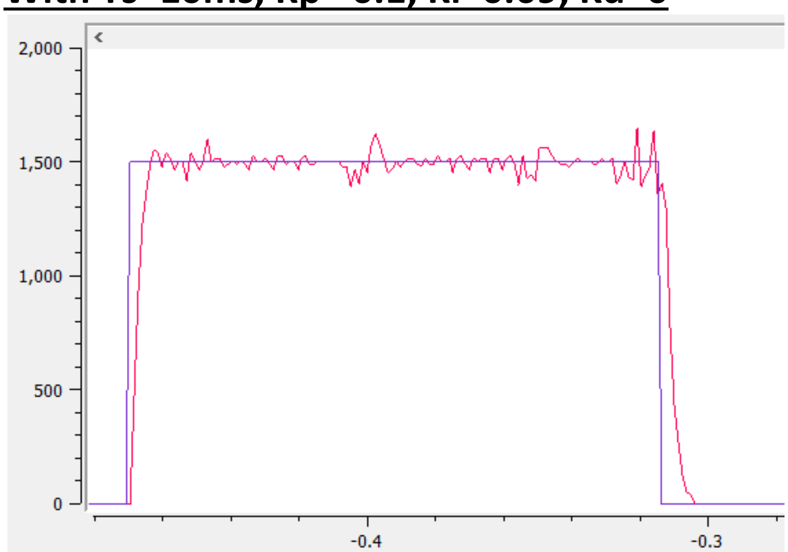
## With Ts=10ms, Kp =0.06, Ki=0.06, Kd=0.001




## With Ts=10ms, Kp =0.07, Ki=0.09, Kd=0

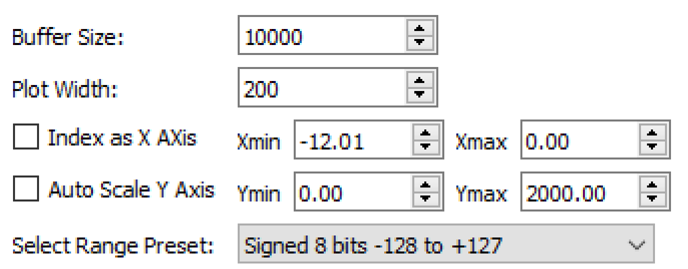



## With Ts=10ms, Kp =0.1, Ki=0.09, Kd=0



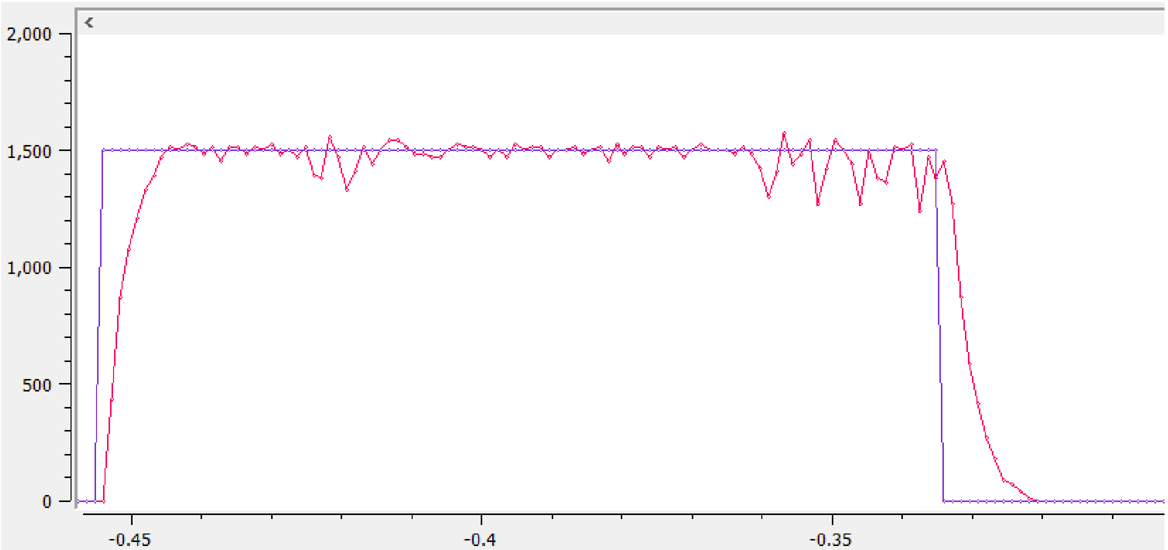
# 3-SPEED CONTROL WITH Ts=8ms, SP=1500tr/mn

## SimPlot display settings for Ts=10ms:

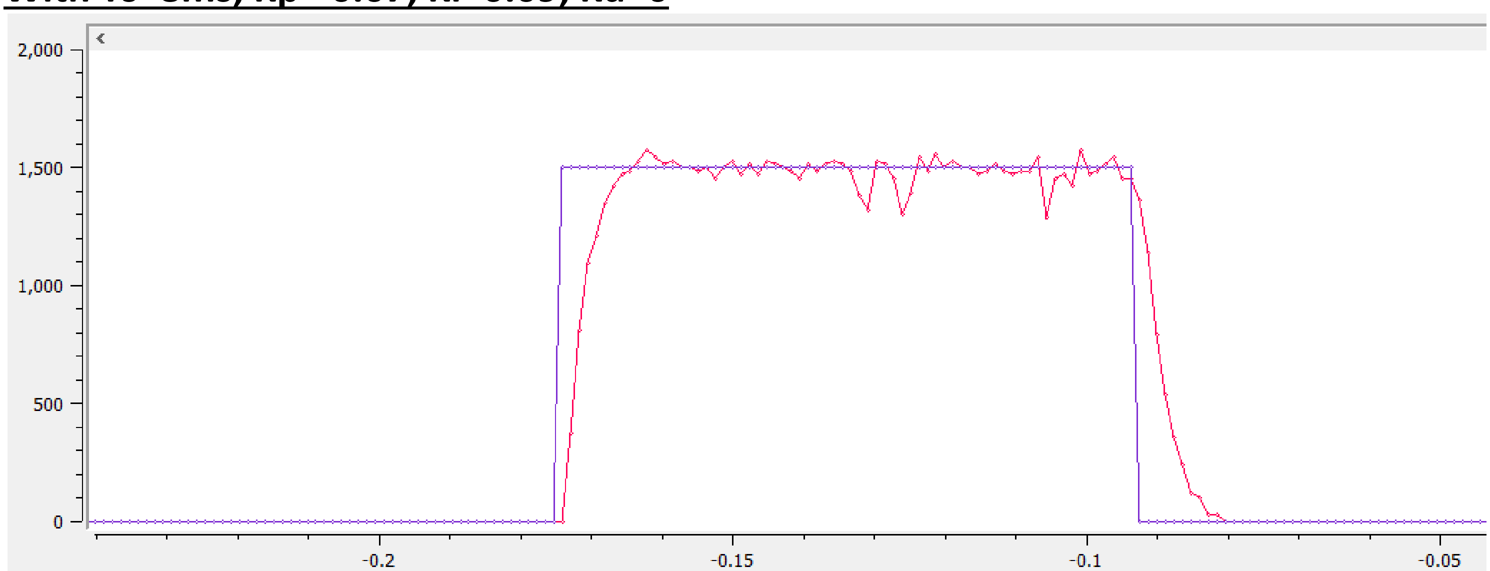



8ms between 2 points

## With Ts=8ms, Kp =0.1, Ki=0.05, Kd=0.00009

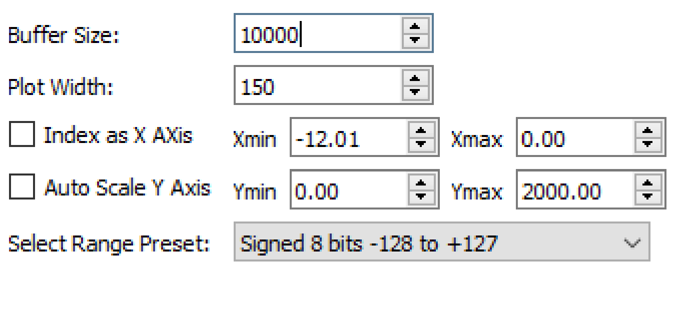



## With Ts=8ms, Kp =0.07, Ki=0.09, Kd=0



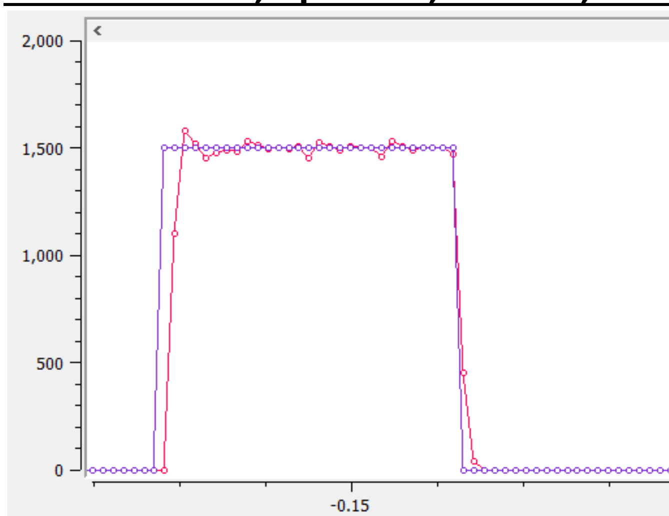
# 4-SPEED CONTROL WITH Ts=50ms, SP=1500tr/mn

## SimPlot display settings for Ts=50ms:



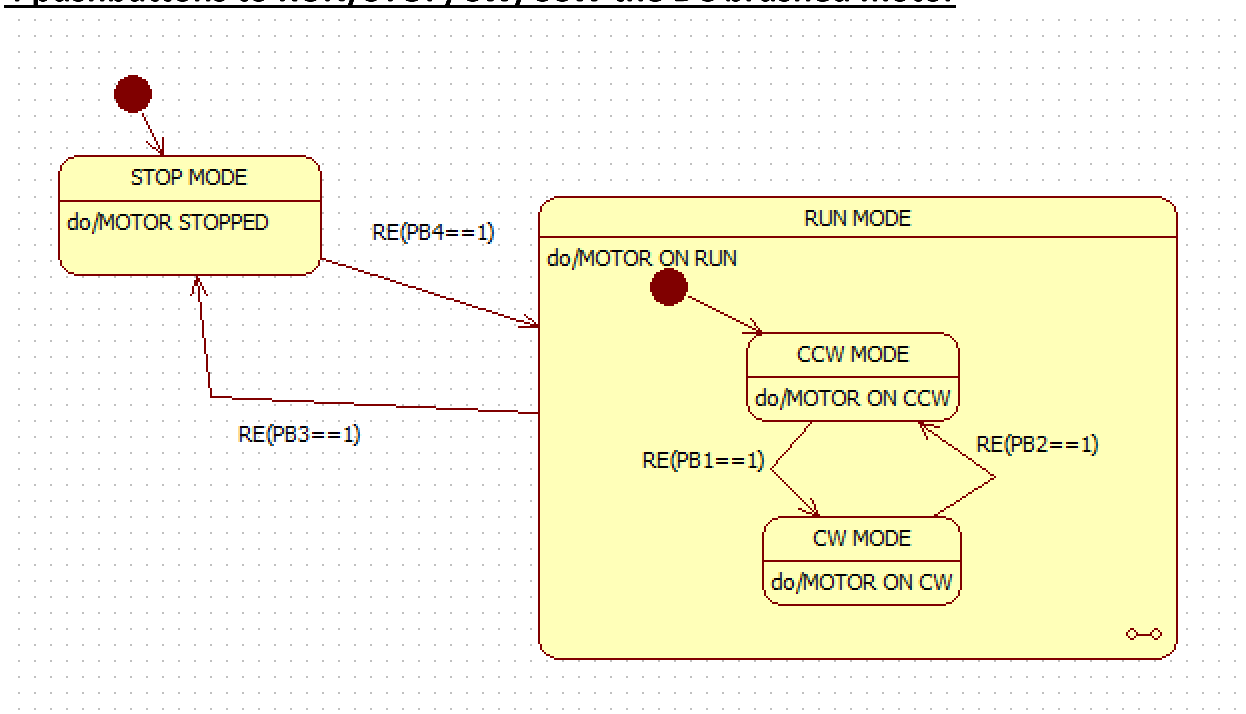

50ms between 2points.

## With Ts=50ms, Kp =0.03, Ki=0.52, Kd=0




# 5-The State machine/the timers/the interruptions/LCD I2C/serial plot used in speed control:

## 4 pushbuttons to RUN/STOP/CW/CCW the DC brushed motor



**Pin map choice:**

You can't choose either timer 0 or timer 1 because it causes malfunctions and interferes with progamed functions on arduino.

It's suitable to put an atmega1284 or 1284 or a 644 on your clone board because only these microcontrollers are able to run all the functions needed here.

-timer 2: to change PWM frequency on pin 15 to 31250Hz

-timer 3 to generate timer tick interrupt: the sampling time which runs the PID

-interrupt pin 10 and pin 11 to count pulses on A and B channels: sense of drive/speed calculation/position calculation/turns and degres calculation

-serial lines RX and TX (pin 0 and pin 1): to read the speed or the position on the usb serial plotter.

-I2C lines for LCD display (pin 16 and pin 17.

-the ATOMIC BLOCK is a way to measure the pulses or whatever you want without any disturbs. It's like a non intrusive way to have accurate measurements and an increase of the read/write/calculation time.

# 6-Position control:

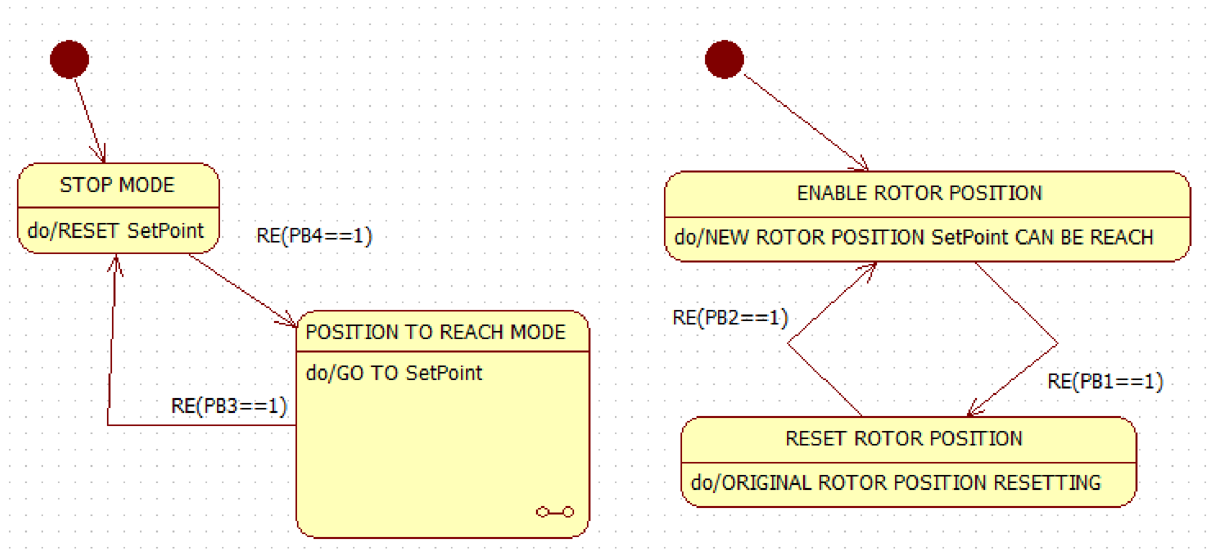
The setpoint of the position is divided in 2 parts for more accuracy:

-define the amount of degres: 0° ≤SetPointDeg ≤360°

-define the number of desired integer turns: SetPointTurn

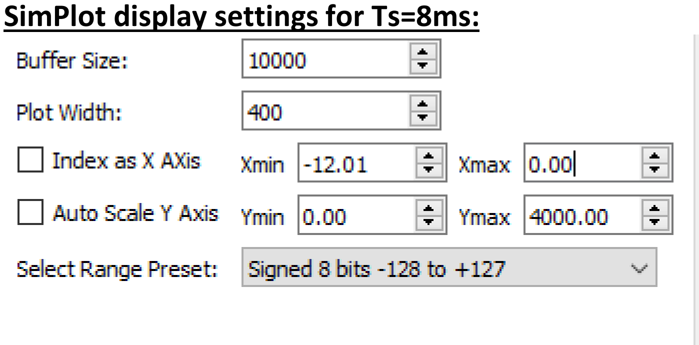
Then the position is calculated in degres: SPpositionDEG2 = SetPointDeg + (SetPointTurn * 360)

A new state machine to proceed measurements:

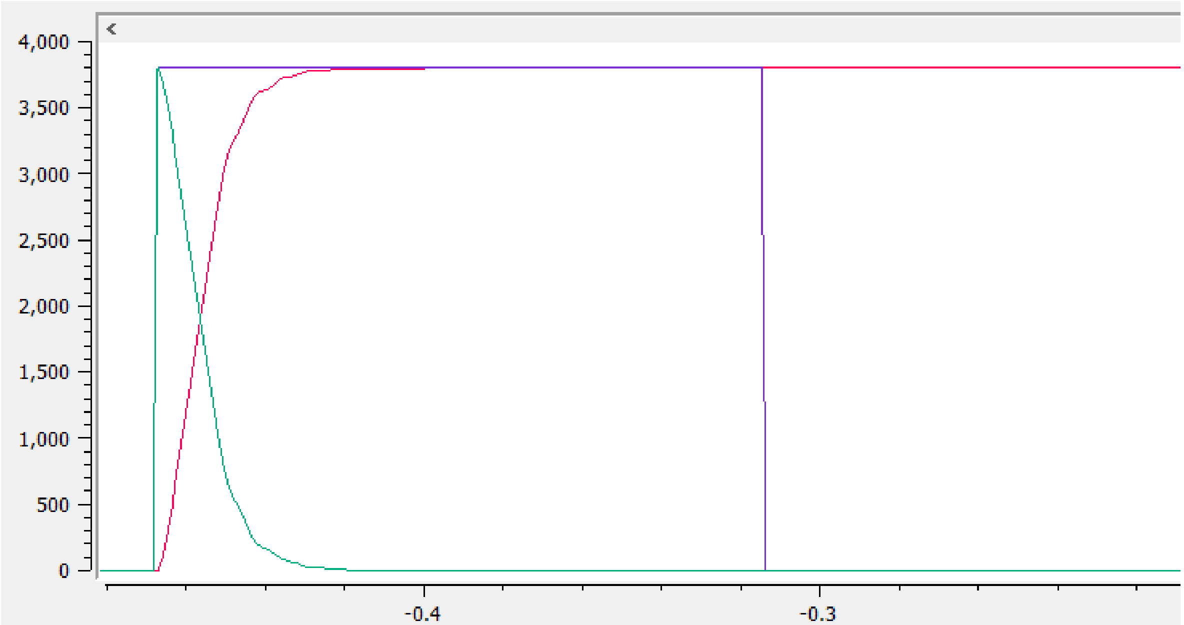


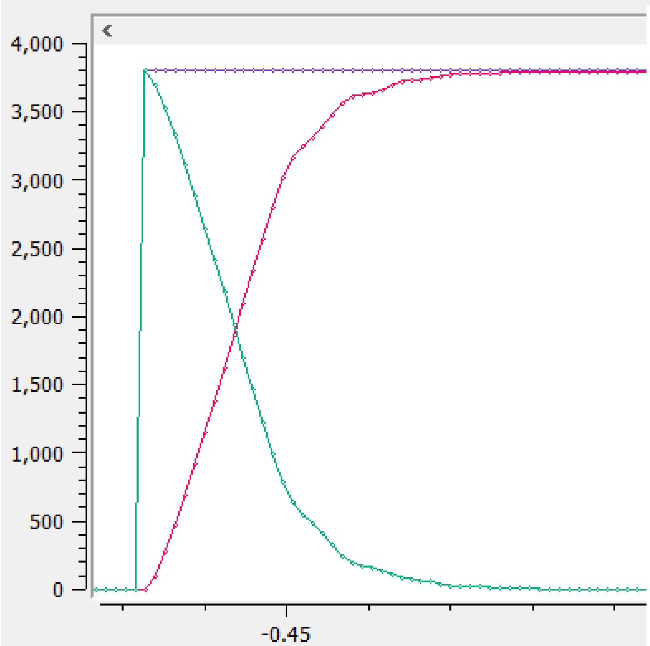

## POSITION CONTROL WITH Ts=8ms, SetPointDeg=200°, SetPointTurn=10 turns

**SimPlot display settings for Ts=8ms:**



## With Ts=8ms, Kp =1, Ki=0.07, Kd=0.01, SetPointDeg2=3800°

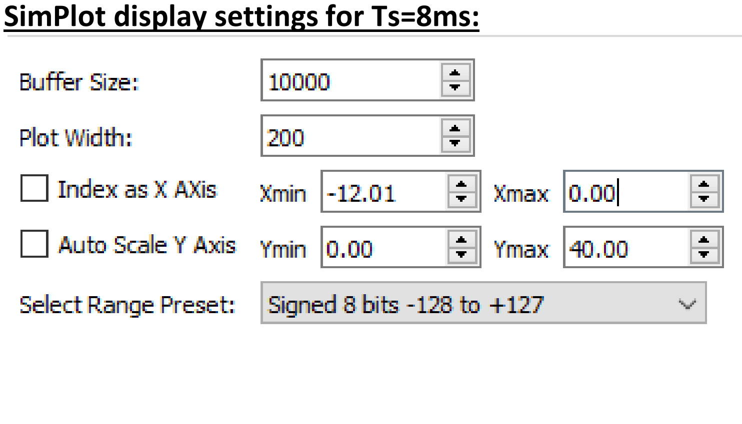


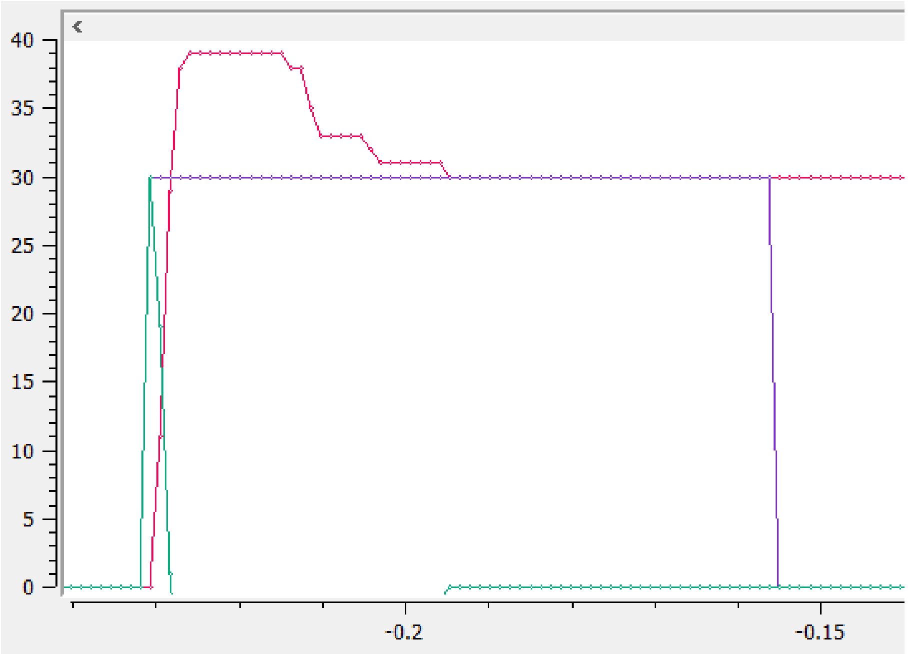

Response time: tr=28x8ms=224ms

## POSITION CONTROL WITH Ts=8ms, SetPointDeg=30°, SetPointTurn=0 turns

### SimPlot display settings for Ts=8ms:



## With Ts=8ms, Kp =1, Ki=0.07, Kd=0.01, SetPointDeg2=30°




Response time: tr=20x8ms=160ms