Steering Controller Tester Documentation

Bruno Tibério

June, 2018

CONTENTS:

CHAPTER ONE

EPOS CONTROLLER TESTER CLASS

Simple class to perform a few tests with EPOS device in the car It is a wrap around Epos class to include additional functions.

class steering_controller.**Epos_controller**(*_network=None*, *debug=False*)

Bases: `EPOS_Canopen.epos.Epos`

emcyErrorPrint(*EmcyError*)

Print any EMCY Error Received on CAN BUS

getDeltaAngle(*qc*)

Converts qc of steering wheel to angle of wheel

Given the desired qc steering position, convert the requested value to angle of bicycle model in degrees.

Parameters **qc** – an int with desired qc position of steering wheel.

Returns estimated angle of wheels in degrees or None if not possible

Return type double

getQcPosition(*delta*)

Converts angle of wheels to qc

Given the desired angle of wheels, in degrees of the bicycle model of car, convert the requested value to qc position of steering wheel using the calibration performed at beginning.

Parameters **delta** – desired angle of wheels in degrees.

Returns a rounded integer with qc position estimated or None if not possible

Return type int

moveToPosition(*pFinal*, *isAngle=False*)

Move to desired position.

Plan and apply a motion profile to reduce with low jerk, max speed, max acceleration to avoid abrupt variations. The function implement the algorithm developed in[1]

Parameters

- **pFinal** – desired position.
- **isAngle** – a boolean, true if pFinal is an angle or false if is qc value

Returns

[1] Li, Huaizhong & M Gong, Z & Lin, Wei & Lippa, T. (2007). Motion profile planning for reduced jerk and vibration residuals. 10.13140/2.1.4211.2647.

`readFromFile` (*filename=None*, *useAngle=False*)

Read qc positions from file and follow them

The file must contain time and position in quadrature positions of steering wheel and angle (degrees) of "center" wheel of bicycle model in a csv style

time	position	angle
t1	p1	a1
…	…	…
tN	pN	aN

If the calibration value is not the same, user should set useAngle flag. Because the calibration when the file was created can differ from current calibration, user should set the useAngle flag to calculate the position reference to send for device via the given angle.

Parameters

- **`filename`** – csv file to be read.
- **`useAngle`** – use the angle value instead of position.

`saveToFile` (*filename=None*, *exitFlag=None*)

Record qc positions into a csv file

The following fields will be recorded

time	position	angle
t1	p1	a1
…	…	…
tN	pN	aN

An additional file with same name but with ext TXT will have the current calibration parameters

- minValue
- maxValue
- zeroRef

If filename is not supplied or already used, the current asctime() will be used as filename.

Parameters

- **`filename`** – name of the file to save the data
- **`exitFlag`** – threading event flag to signal exit.

`startCalibration` (*exitFlag=None*)

Perform steering wheel calibration

This function is expected to be run on a thread in order to find the limits of the steering wheel position and find the expected value of the zero angle of wheels.

Parameters **`exitFlag`** – threading.Event() to signal the finish of acquisition