Trio Calibration and Pathtypes

Frederik Hartig & Jeppe Fjederholdt Nielsen

June 2020

Contents

1 Introduction

This document is a guide on how the program calibrates the Robot.

2 Calibration

This section explains how the Trio in cooperation with the robostacker program, calibrates the robot.

2.1 Robot file

For calibration some variables in the .robot file are used seen in figure 1.

```
VERSION 3
m_name=340
m_gantryLength=3500.00,m_legHeight=735.00,m_shoulderGearB=822.000000
m_Link[0]=0.0000,126.0000,0.2000,-0.6800,0.0400,8.2500,0.5700,0.2100,0.5700,4.0400,1.6800,0.2100,1.6800,7.3200
m_Link[1]=1000.0000,58.2200,0.3500,0.0000,0.1400,0.3900,0.0000,-0.0600,0.0000,9.5700,0.0000,-0.0600,0.0000,9.8600
m_Link[2]=1055.0000,65.6500,0.1700,0.0000,0.0200,0.6100,-0.0200,-0.6400,-0.0200,14.3500,0.0100,-0.6400,0.0100,14.0200
m_Link[3]=171.0000,3.6600,0.0001,-0.0564,0.0072,0.0200,0.0000,0.0000,0.0000,0.0100,0.0000,0.0000,0.0000,0.0300
m_Link[4]=0.0000,1.6900,0.0000,0.0000,0.1167,0.0058,0.0000,0.0000,0.0000,0.0058,0.0000,0.0000,0.0000,0.0010
m_State[0]=0.0000,1.6900,0.0000,0.0000,0.1167,0.0058,0.0000,0.0000,0.0000,0.0058,0.0000,0.0000,0.0000,0.0010
m_State[1]=0.0000,11.3280,-0.1900,0.0000,0.0900,0.5000,0.0000,0.1100,0.0000,0.3400,0.0000,0.1100,0.0000,0.3800
m_State[2]=0.0000,21.3280,-0.4000,0.0000,0.1900,0.9000,0.0000,0.2100,0.0000,0.7400,0.0000,0.2100,0.0000,0.7800
m_sign.a=-1,b=-1,c=1,d=1,e=-1,f=1
m_ratedNmGear.a=40.00,b=1568.00,c=491.00,d=55.00,e=108.00,f=108.00
m_ratedRpmGear.a=680,b=27,c=42,d=20,e=40,f=40
m_ratedRPM.a=3000,b=3000,c=3000,d=3000,e=3000,f=3000
m_ratedNm.a=7.50,b=18.40,c=7.50,d=2.20,e=2.20,f=2.20
m_maxForceNm.a=75.00,b=3716.80,c=1575.00,d=220.00,e=110.00,f=110.00
m_pulses.a=65536.0000,b=65536.0000,c=65536.0000,d=65536.0000,e=65536.0000,f=65536.0000
m_gearing.a=5,b=101,c=105,d=135,e=50,f=7
m_unit.a=157.0796,b=360.0000,c=360.0000,d=360.0000,e=360.0000,f=157.0796
m_units.a=2086.0759,b=18386.4883,c=19114.6660,d=24576.0000,e=9102.2227,f=2920.5063
m_p_gain.a=0.1000,b=0.1000,c=0.1200,d=0.1500,e=0.2000,f=0.0000
m_i_gain.a=0.0000,b=0.0000,c=0.0000,d=0.0000,e=0.0000,f=0.0000
m_d_gain.a=0.1500,b=0.1500,c=0.1500,d=0.2000,e=0.2500,f=0.0000
m_vff_gain.a=8.0000,b=8.0000,c=8.0000,d=8.0000,e=8.0000,f=0.0000
m_speed.a=50.0000,b=4.0000,c=4.0000,d=4.0000,e=4.0000,f=0.0000
m_accel.a=100.0000,b=8.0000,c=8.0000,d=8.0000,e=8.0000,f=0.0000
m_decel.a=100.0000,b=8.0000,c=8.0000,d=8.0000,e=8.0000,f=0.0000
m_creep.a=0.1000,b=0.1000,c=0.1000,d=0.1000,e=0.1000,f=0.0000
m_jogspeed.a=0.5000,b=0.5000,c=0.5000,d=0.5000,e=0.5000,f=0.0000
m_felimit.a=10.0000,b=1.0000,c=1.0000,d=1.0000,e=1.0000,f=0.0000
m_min.x=-9999.00,y=-9999.00,z=-9999.00,v=-9999.00,w=-9999.00,u=-9999.00
m_max.x=9999.00,y=9999.00,z=9999.00,v=9999.00,w=9999.00,u=9999.00
m_localTrans.x=0.00,y=-1557.00,z=0.00,v=0.00,w=0.00,u=0.00
m_jointCalib.a=0.00,b=47.80,c=89.99,d=-17.70,e=-110.55,f=0.00
m_mpnhome.a=4861499.00,b=0.00,c=0.00,d=0.00,e=0.00,f=0.00
m_tcpTrans[case].x=157.00,y=0.00,z=0.00,v=0.00,w=0.00,u=0.00
m_tcpTrans[bag].x=200.00,y=0.00,z=0.00,v=0.00,w=0.00,u=0.00
m_tcpTrans[zero].x=-171.00,y=0.00,z=0.00,v=0.00,w=0.00,u=0.00
```

Figure 1: Robot file

Now the usable variables will be explained

2.1.1 Version

This is the Version of the robot, version 3 is generally used.

2.1.2 m_name

The name of the robot, can be a number but an describing abbreviation is preferred, like PS for PalletStacker.

2.1.3 m_gantryLength, m_legHeight, m_shoulderGearB

These variables determine what length the robot gantry, the gantry's leg height and the shoulder height of the B-axis is. The calibration of the TCP takes the leg and shoulder height in account. When the robot is calibrated, the TCP y coordinate would be leg height plus shoulder height. The leg and shoulder height are calculated and put into **m_localTrans.y** from section 2.1.8.

2.1.4 m_Link

This is the dimensions of the robot links.

```
m_Link[link from gantry=0]=length,mass,massDisplacement(ms),ms,ms,inertia(i),i,i,i,i,i,i,i,i
```

There are 6 links in total:

Link number	Axis
0	A
1	B
2	C
3	D
4	E
5	F

Table 1: Link and Axis

The links are used to perform kinematics and find the TCP in xyz coordinate system.

2.1.5 m_sign

This variable decides which way the motors turn, e.g:

```
m_sign.a=-1
```

Then the robot on the gantry moves from left being 0 to right.

2.1.6 m_min

m_min describes the minimum value all the Axis can be, on the robot.

2.1.7 m_max

m_min describes the maximum value all the Axis can be, on the robot.

2.1.8 m_localTrans

m_localTrans has the offset for the y-axis. The offset is equal to the gantry's leg height, and the shoulder height of the B-axis

2.1.9 m_jointCalib

m_jointCalib is the offset in degrees the zero point have from the calibration position.
The offset can change from robot to robot depending on the homesensors placement.

2.2 AutoHome

AutoHome is the calibration function that moves B-axis until the homesensor is off, and then moves the C-axis until the homesensor is off. it then sets the **m_jointCalib** values accordingly. When the autohome is performed, the A, D and E-axis also needs to be calibrated.

2.3 Zeroing axis

When zeroing an axis, the coordinate values related to the axis will be set to zero. this is used to calibrate the A-axis so the z coordinate is zero.

2.4 ZeroTool vs Tool90

ZeroTool and Tool90 is both functions that calibrate the D- and E-Axis.

1. ZeroTool is calibrating the Tool at point 0, the point were the D-axis is pointing at a vertical line away from the Robot.

2. Tool90 is calibrating the Tool with a 90 degrees angle, when the tool is in a horizontal line.

3 Path description

This section explains how pathtypes are executed. The following list is based on the Mihada trio code.

- pathtype 10 → path10
- pathtype 11 → path11
- pathtype 12 → path12
- pathtype 13 → path13
- pathtype 14 → path14
- pathtype 16 → path16
- pathtype 20 → path20 (Move to pallet pick up)
- pathtype 21 → path21 (Move to conveyor from pallet pick up)
- pathtype 22 → path22 (Move to home from conveyor)
- pathtype 23 → path23 (Move to conveyor pick up)
- pathtype 24 → path24 (Move to pallet from conveyor pick up)
- pathtype 25 → path25 (Move to home from pallet)
- pathtype 43 → path43 (Move to home from conveyor)
- pathtype 44 → path44 (Move to conveyor pick up from pallet)

There are currently no inherent correlation between the pathtype number and the execution of the pathtype. This needs standardization.

Pathtype	**startFrame task**	**Running test**	**endFrame task**	Outdated
10	N/A	crash and box-drop	N/A	Yes
11	Flap **OFF**, Trap **OFF**	N/A	N/A	Yes
12	Flap **OFF**, Trap **OFF**	crash and box check	N/A	Yes
13	Flap **ON**, Trap **OFF**	crash and box check	Flap **ON**, Trap **ON**	Yes
14	N/A	crash and box-drop	Flap **OFF**, Trap **OFF**	Yes
16	Grip **OFF**, Box support **OFF**	crash	Grip **OFF**, Box support **OFF**	No
20	Grip **OFF**, Box support **OFF**	crash	Grip **ON**	No
21	Grip **ON**	crash and double box	Grip **OFF**, Box support **OFF**	No
22	Grip **OFF**, Box support **OFF**	crash	Grip **OFF**, Box support **OFF**	No
23	Grip **OFF**, Box support **OFF**	crash	Grip **ON**, Box support **ON**	No
24	Grip **ON**, Box support **ON**	crash	Grip **OFF**, Box support **OFF**	No
25	Grip **OFF**, Box support **OFF**	crash	Grip **OFF**, Box support **OFF**	No
43	Grip **OFF**, Box support **OFF**	crash	Grip **OFF**, Box support **OFF**	No
44	Grip **OFF**, Box support **OFF**	crash	Grip **ON**, Box support **ON**	No

Table 2: Pathtype description