Application Note

FESTO

EXCM calculations

Notice regarding the use of the H-type EXCM/EXCH gantry without motion PLC.

This application note explain how to calculate the dynamics data for each motors to obtain a linear movement in the 2D cartesian system.

EXCM
EXCH

Title ..... EXCM calculation
Version 1.10
Document no. 100691
Original ..... en
Author ..... Festo
Last saved 13.08.2024

Copyright Notice

This documentation is the intellectual property of Festo SE & Co. KG, which also has the exclusive copyright. Any modification of the content, duplication or reprinting of this documentation as well as distribution to third parties can only be made with the express consent of Festo SE & Co. KG.

Festo SE & Co KG reserves the right to make modifications to this document in whole or in part. All brand and product names are trademarks or registered trademarks of their respective owners.

Legal Notice

Hardware, software, operating systems and drivers may only be used for the applications described and only in conjunction with components recommended by Festo SE & Co. KG.

Festo SE & Co. KG does not accept any liability for damages arising from the use of any incorrect or incomplete information contained in this documentation or any information missing therefrom.

Defects resulting from the improper handling of devices and modules are excluded from the warranty.

The data and information specified in this document should not be used for the implementation of safety functions relating to the protection of personnel and machinery.

No liability is accepted for claims for damages arising from a failure or functional defect. In other respects, the regulations with regard to liability from the terms and conditions of delivery, payment and use of software of Festo SE & Co. KG, which can be found at www.festo.com and can be supplied on request, shall apply.

All data contained in this document do not represent guaranteed specifications, particularly with regard to functionality, condition or quality, in the legal sense.

The information in this document serves only as basic information for the implementation of a specific, hypothetical application and is in no way intended as a substitute for the operating instructions of the respective manufacturers and the design and testing of the respective application by the user.

The operating instructions for Festo products can be found at www.festo.com .

Users of this document (application note) must verify that all functions described here also work correctly in the application. By reading this document and adhering to the specifications contained therein, users are also solely responsible for their own application.

© (Festo SE & CO. KG, D-73726 Esslingen, 2024)
Internet: http://www.festo.com

Table of contents

1 Components

Photo	Type/Name	Part nº	Links
	EXCM-30-	2226101	

Table 1.1: Components/Software used

2 Introduction

2.1 Hardware Configuration

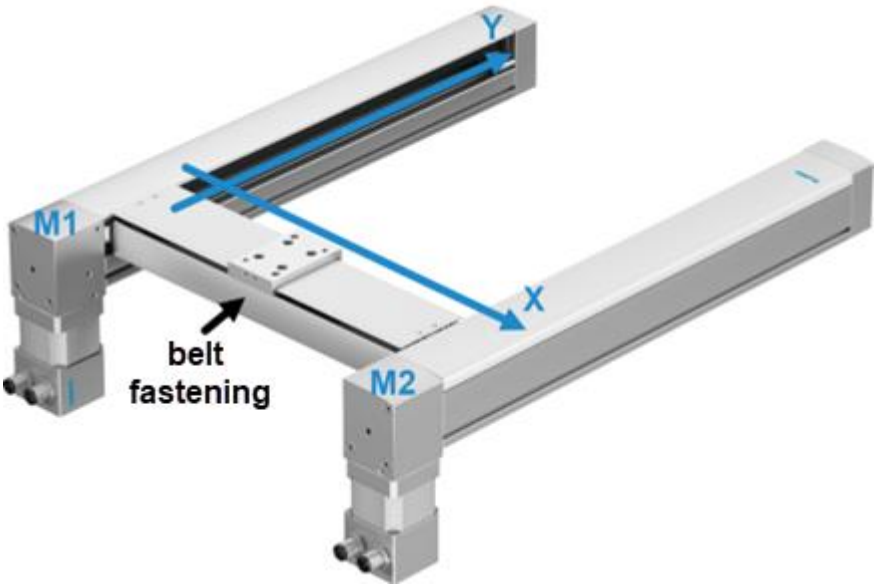


Fig. 2.1: EXCM-30

2.2 Conversion Matrices

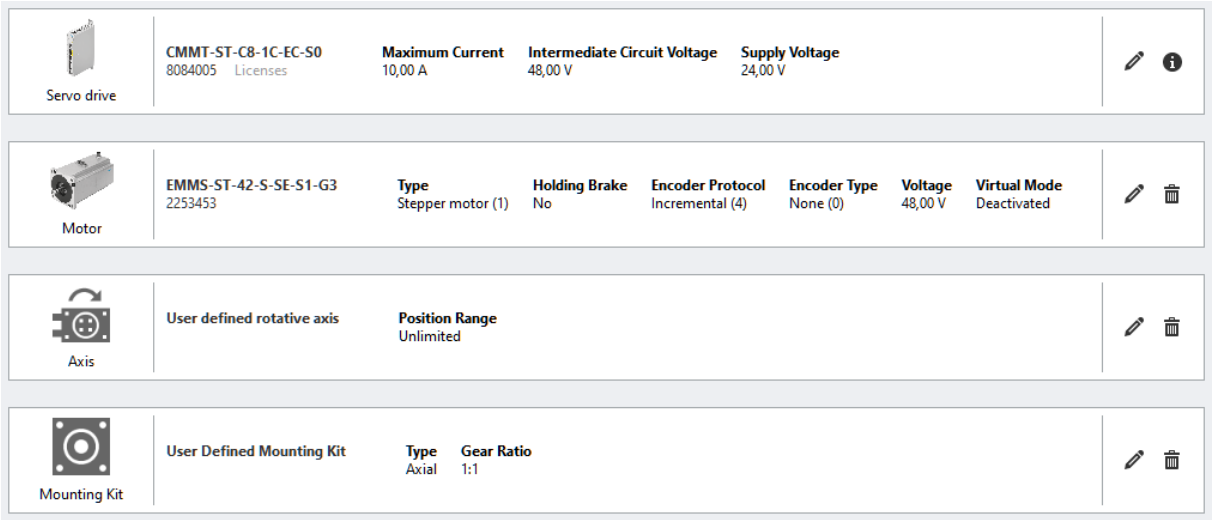
To know the angular positions θ1 and θ2 (rad) of each motor from the Cartesian position [X;Y] (mm)

$$\begin{bmatrix}\theta 1\\ \theta 2\end{bmatrix} = \frac{1}{radius} * \begin{bmatrix}-1 & -1\\ -1 & 1\end{bmatrix}\begin{bmatrix}X\\ Y\end{bmatrix}$$

To determine the cartesian position [X;Y] from the angular positions θ1 and θ2 (rad)

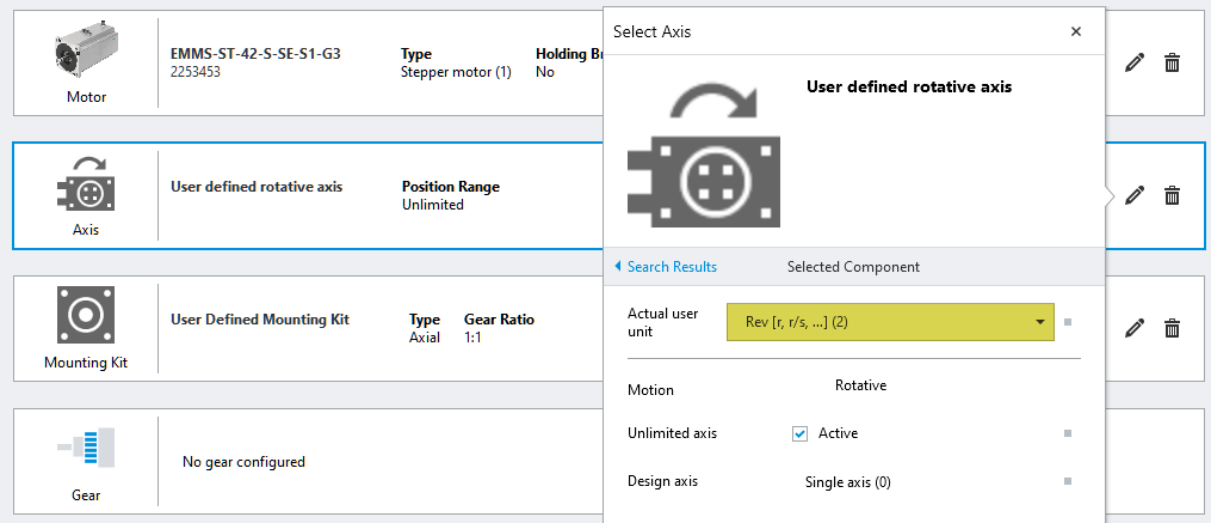
$$\begin{bmatrix}X\\ Y\end{bmatrix} = \frac{radius}{2} * \begin{bmatrix}-1 & -1\\ -1 & 1\end{bmatrix}\begin{bmatrix}\theta 1\\ \theta 2\end{bmatrix}$$

The motor controllers are each configured with their own motor and the unit of measurement is the radian (rad; rad/s; rad/s²)



The use of radians is not mandatory, it is quite possible to change the unit of measurement and adapt the matrices accordingly.

For example in rev turns [r, r/s, r/s² ...]



$$\begin{bmatrix}\theta 1\\ \theta 2\end{bmatrix} = \frac{1}{feed\ constant} * \begin{bmatrix}-1 & -1\\ -1 & 1\end{bmatrix}\begin{bmatrix}X\\ Y\end{bmatrix}$$

$$\begin{bmatrix}X\\ Y\end{bmatrix} = \frac{feed\ constant}{2} * \begin{bmatrix}-1 & -1\\ -1 & 1\end{bmatrix}\begin{bmatrix}\theta 1\\ \theta 2\end{bmatrix}$$

The feed constant per revolution of an EXCM-30 is 38mm/revolution.

If we apply the matrix to know the cartesian position, this gives 2 equations:

$\theta 1 = PosMot1$
$\theta 2 = PosMot2$

$$X = \frac{38}{2} * (-PosMot1 - PosMot2)$$

$$Y = \frac{38}{2} * (-PosMot1 + PosMot2)$$

3 Trajectory management

With these conversions it is then possible to send Cartesian instructions [X,Y], but the movement will not be rectilinear, and we will obtain a displacement similar to the red trajectory.

To obtain a straight displacement (green trajectory), it is necessary to determine the dynamics of each motor so that their travel time is equal.

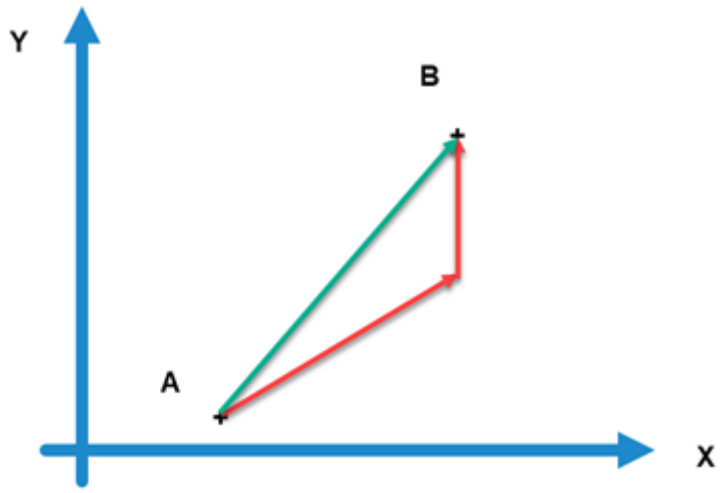


Fig. 3.1: Trajectories of Movements

Additional equations will be used to calculate the speeds, accelerations and deceleration for each engin

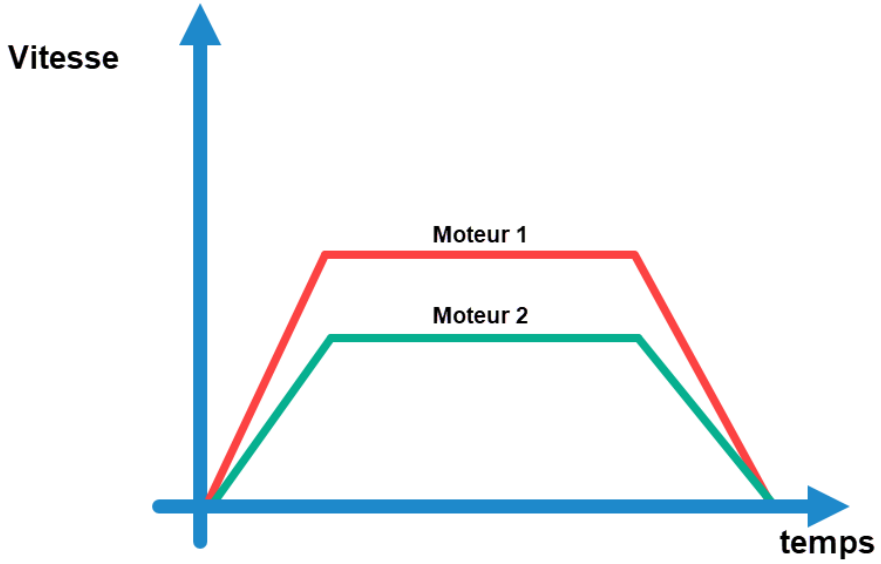


Fig. 3.2: Combined motor speeds

4 Kinematic equations

Starting position	Setpoint	Unknowns to be determined
Angular position M1 ($\Theta 1_A$)	Position X_B (mm)	Angular position M1 ($\Theta 1_B$)
Angular position M2 ($\Theta 2_A$)	Position Y_B (mm)	Angular position M2 ($\Theta 2_B$)
Position X_A (mm)	Velocity (mm/s)	Velocity M1
Position Y_A (mm)	Acceleration (mm/s²)	Velocity M2
	Deceleration (mm/s²)	Acceleration M1
		Deceleration M1
		Acceleration M2
		Deceleration M2

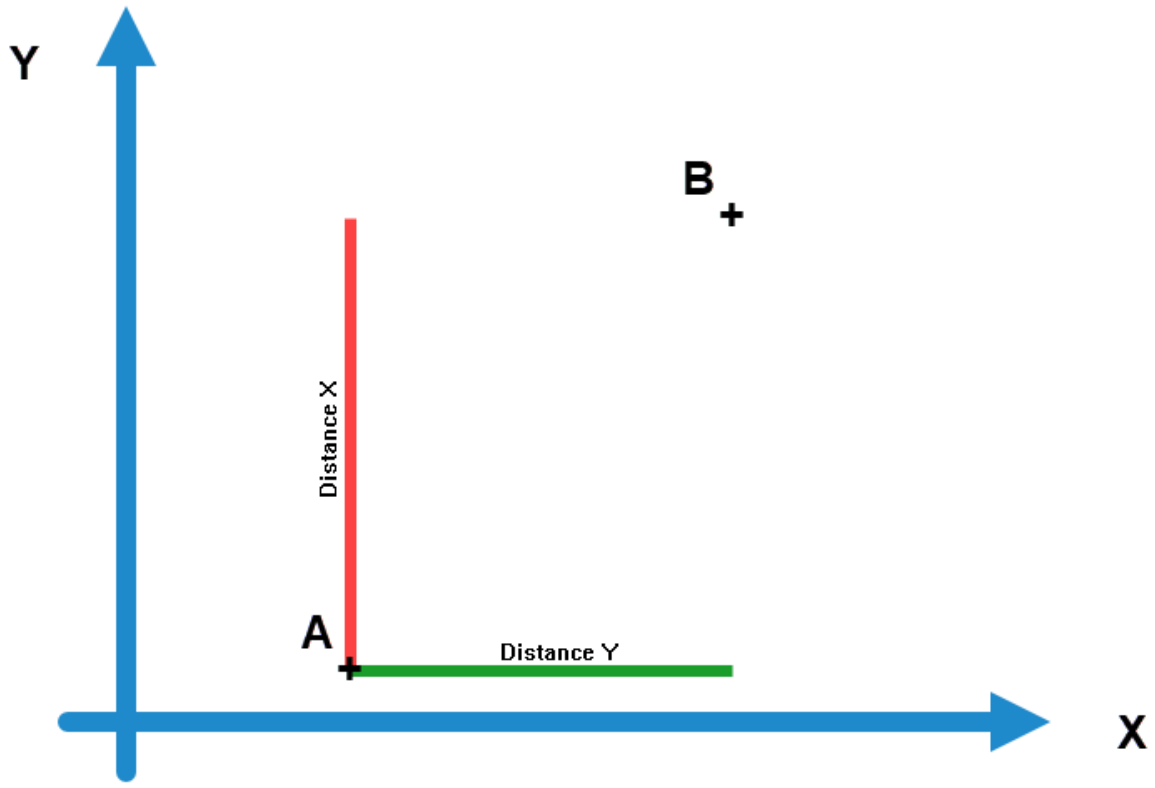
Table 4.1: Input data

The units of measurement of the unknowns to be determined are based on the parameterization made in the Festo Automation Suite software.

4.1 Calculation of distances by Axis (mm) and motor (rev)

$$Distance\ X_{AB} = actual\ position\ X_A - setpoint\ X_B$$
$$Distance\ Y_{AB} = actual\ position\ Y_A - setpoint\ Y_B$$

$$angular\ distance\ M1 = actual\ position\ \Theta 1_A - setpoint\ \Theta 1_A$$
$$angular\ distance\ M2 = actual\ position\ \Theta 2_A - setpoint\ \Theta 2_B$$

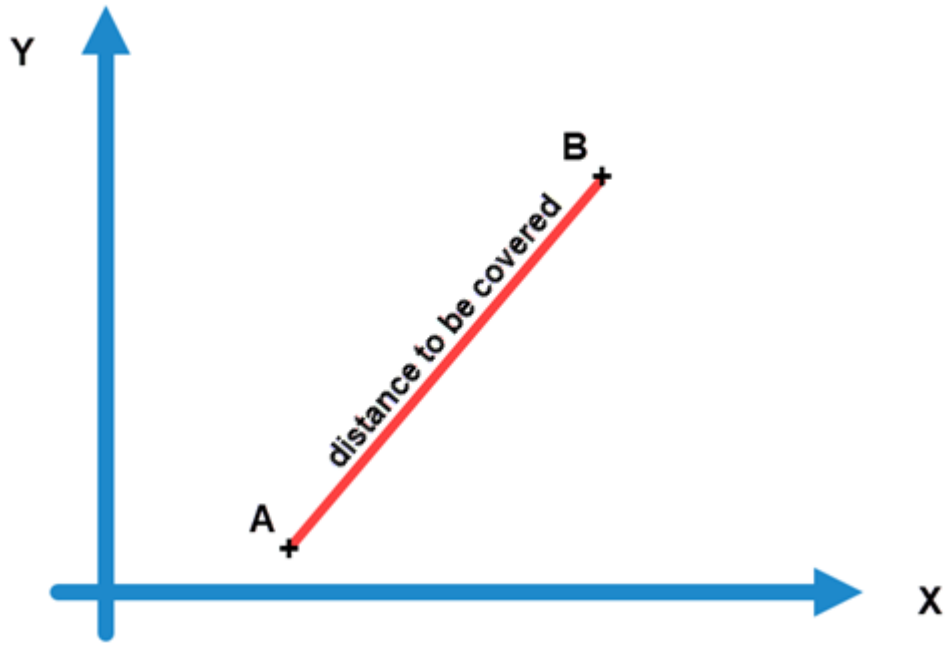


A = current position.

B = position setpoint to be reached.

4.2 Calculation of the distance to be covered [X,Y], (Pythagoras)

$$distance\ to\ be\ covered\ = \sqrt{distance\ on\ X^2 + distance\ on\ Y^2}$$



4.3 Calculation of acceleration and deceleration times

$$time_{acc} = \frac{Velocity}{acceleration}$$

$$time_{dec} = \frac{Velocity}{deceleration}$$

4.4 Calculation of acceleration and deceleration distances

$$acc\ distance = \frac{1}{2} * acceleration * {time_{acc}}^2$$

$$acc\ distance = \frac{1}{2} * deceleration * {time_{dec}}^2$$

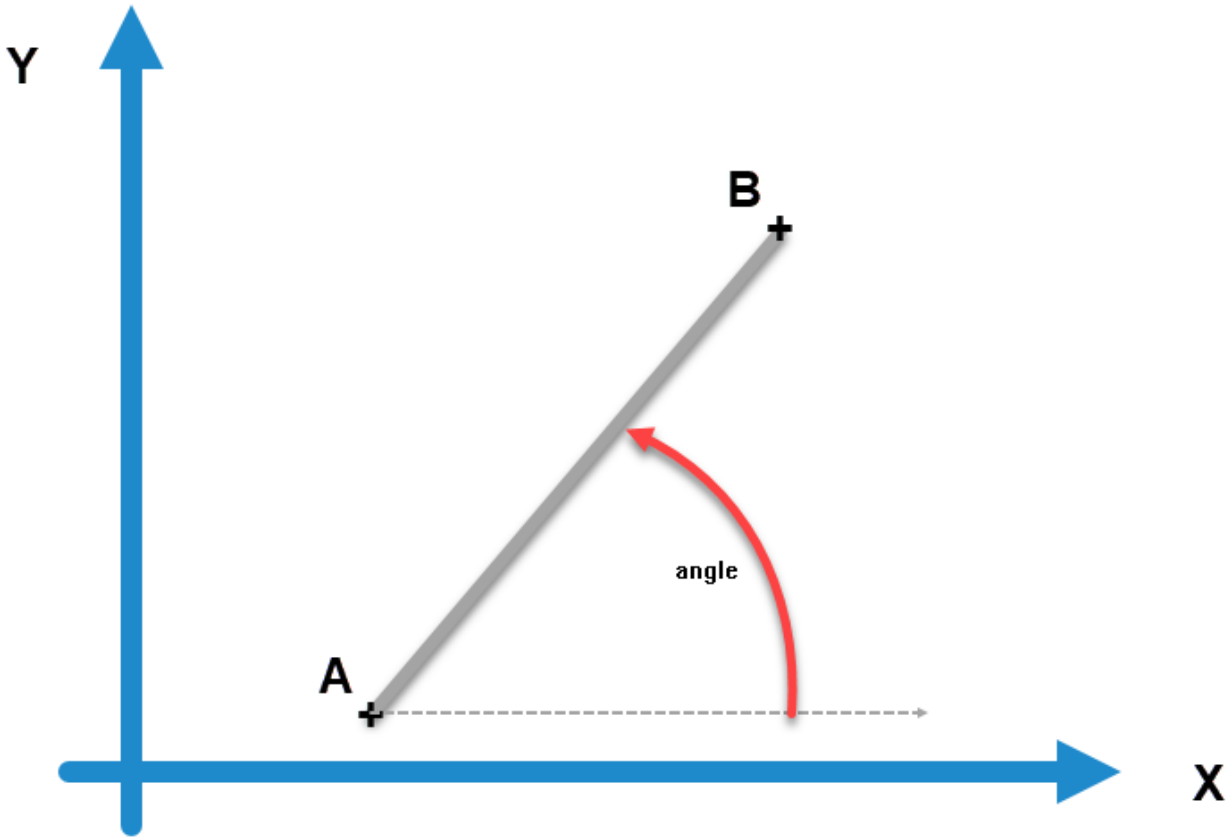
4.5 Calculating the distance traveled at Vmax

$$Vmax\ distance = \ distance\ to\ be\ covered - (\ Acc\ distance + Dec\ distance\ )$$

Optional formula, to determine if the maximum speed can be reached $Vmax\ distance > 0$

4.6 Determination of the Direction of Motion Angle

$$angle = \tan^{-1} * \left(\frac{Distance\ Y_{AB}}{Distance\ X_{AB}}\right)$$



4.7 X & Y axis projection

α = angle

$$Acc\ distance\ on\ X = \sin\alpha\ *\ Acc\ distance$$
$$Dec\ distance\ on\ X = \sin\alpha\ *\ Dec\ distance$$
$$Acc\ distance\ on\ Y = \cos\alpha\ *\ Acc\ distance$$
$$Dec\ distance\ on\ Y = \cos\alpha\ *\ Dec\ distance$$

4.8 Conversion of Acceleration Distances and Engine Rotation Deceleration

We consider that the parameterized unit is: Rev [r, r/s, r/s² ...] and that the feed constance = 38mm.

$$Acc\ distance\ M1 = \frac{1}{38} * (\ Acc\ distance\ X + \ Acc\ distance\ Y)$$
$$Dec\ distance\ M1 = \frac{1}{38} * (\ Dec\ distance\ X + \ Dec\ distance\ Y)$$
$$Acc\ distance\ M2 = \frac{1}{38} * (\ Acc\ distance\ X - \ Acc\ distance\ Y)$$
$$Dec\ distance\ M2 = \frac{1}{38} * (\ Dec\ distance\ X - Dec\ distance\ \ Y)$$

4.9 Determination of Rotational Acceleration and Deceleration/s²

$$\mathbf{Acc\ M1} = 2 * \frac{Acc\ distance\ M1}{Acc\ time^{\ 2}}$$

$$\boldsymbol{Dec\ M1} = 2 * \frac{Dec\ distance\ M1}{Dec\ time^{\ 2}}$$

$$\boldsymbol{Acc\ M2} = 2 * \frac{Acc\ distance\ M2}{Acc\ Time^{\ 2}}$$

$$\boldsymbol{Dec\ M2} = 2 * \frac{Dec\ distance\ M2}{Dec\ Time^{\ 2}}$$

4.10 Determination of velocity, rev/s

$$\boldsymbol{M1\ Velocity} = M1\ Acc * Acc\ time$$
$$\boldsymbol{M2\ Velocity} = M2\ Acc * Acc\ time$$

5 Motion Controls

In this sample, the Festo point to point Library for Codesys3.5 is used, it's available on Festo Support Portal

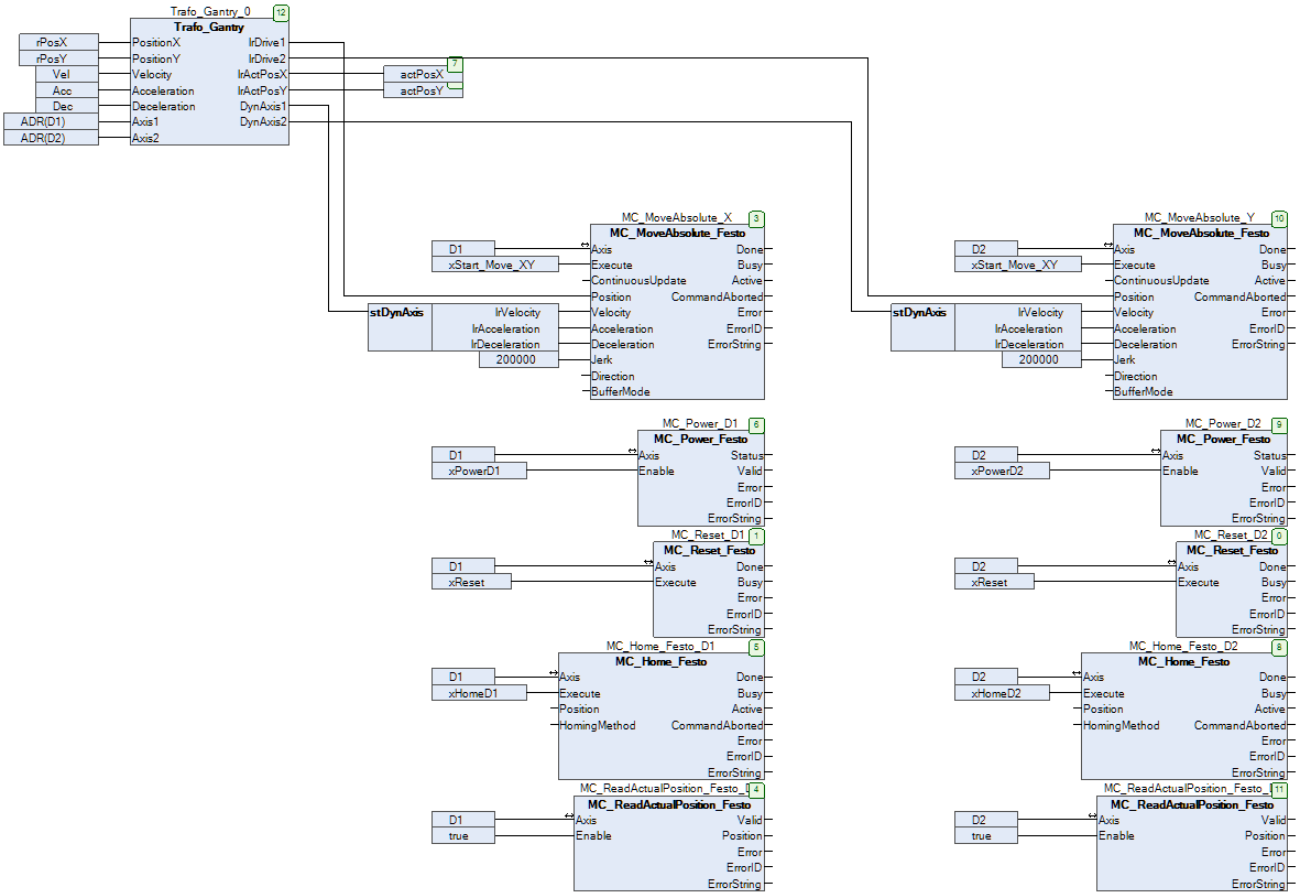
Function blocks CODESYS

PtP function blocks for Codesys 3.5
Point-to-point (PtP) library for servo drives with EtherCAT (CiA402) in the Codesys V3.5 development environment.

All the calculation are implemented in an function bloc named Trafo_Gantry, the source code is shown at the next page.

Each motor can then be controlled individually and simultaneously by applying the calculated setpoints for each one

Position, Acceleration, Speed and Deceleration.



```
// needed for axis Position, could be replaced by MC_ReadActualPosition_Festo
D1:=Axis1^;
D2:=Axis2^;

 //Hgantry Conversion Matrices
lrDrive1:=(1/lrRadius)*((-1*PositionX)+(-1*PositionY));
lrDrive2:=(1/lrRadius)*((-1*PositionX)+(1*PositionY));

lrActPosX:=(lrRadius/2)*((-1*D1.ActualPosition)+(-1*D2.ActualPosition));
lrActPosY:=(lrRadius/2)*((-1*D1.ActualPosition)+(1*D2.ActualPosition));

// Calculation of distances by Axis (mm) and motor (rev)
lrDistX:=lrActPosX-PositionX;
lrDistY:=lrActPosY-PositionY;

lrDistD1:=D1.ActualPosition-lrDrive1;
lrDistD2:=D2.ActualPosition-lrDrive2;

// Calculation of the distance to be covered [X,Y], (Pythagoras)
lrDistCart:=SQRT(EXPT(lrDistX,2)+EXPT(lrDistY,2));

// Calculation of acceleration and deceleration times
lrTpsAcc:=Velocity/Acceleration;
lrTpsDec:=Velocity/Deceleration;

// Calculation of acceleration and deceleration distances
lrDistAcc:=0.5*Acceleration*EXPT(lrTpsAcc,2);
lrDistDec:=0.5*Deceleration*EXPT(lrTpsDec,2);

// Determination of the Direction of Motion Angle
lrAngle:=ATAN(lrDistY/lrDistX);

// X & Y axis projection
lrDistAccX:=SIN(lrAngle)*lrDistAcc;
lrDistDecX:=SIN(lrAngle)*lrDistDec;
lrDistAccY:=COS(lrAngle)*lrDistAcc;
lrDistDecY:=COS(lrAngle)*lrDistDec;

// Conversion of Acceleration Distances and Engine Rotation Deceleration
lrDistAccD1:=ABS((1/lrRadius)*((1*lrDistAccX)+(1*lrDistAccY)));
lrDistDecD1:=ABS((1/lrRadius)*((1*lrDistDecX)+(1*lrDistDecY)));
lrDistAccD2:=ABS((1/lrRadius)*((1*lrDistAccX)+(-1*lrDistAccY)));
lrDistDecD2:=ABS((1/lrRadius)*((1*lrDistDecX)+(-1*lrDistDecY)));

// Determination of Rotational Acceleration and Deceleration/s²
DynAxis1.lrAcceleration:=2*lrDistAccD1/(lrTpsAcc*lrTpsAcc);
DynAxis1.lrDeceleration:=2*lrDistDecD1/(lrTpsDec*lrTpsDec);
DynAxis2.lrAcceleration:=2*lrDistAccD2/(lrTpsAcc*lrTpsAcc);
DynAxis2.lrDeceleration:=2*lrDistDecD2/(lrTpsDec*lrTpsDec);

// Determination of velocity, rev/s
DynAxis1.lrVelocity:=DynAxis1.lrAcceleration*lrTpsAcc;
DynAxis2.lrVelocity:=DynAxis2.lrAcceleration*lrTpsAcc;
```