

EXCM calculations

Notice applicative pour l'utilisation de manipulateur en H type EXCM/EXCH .

Cette note d'application explique comment calculer les données dynamiques pour chaque moteur afin d'obtenir un mouvement linéaire dans le système cartésien 2D.

EXCM
EXCH

Title EXCM calculations
Version 1.10
Document no. 100691
Original fr
Author Festo

Last saved 14.08.2024

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.

Table of contents

1	Components	4
2	Introduction	5
2.1	Configuration matériel.....	5
2.2	Matrices de conversion	5
3	Gestion des trajectoires.....	7
4	Equations cinématique	8
4.1	Calcul des distances à parcourir par axes(mm) et par moteur (trs)	8
4.2	Calcul de la distance à parcourir [X,Y], (Pythagore)	9
4.3	Calcul des temps d'accélération et décélération	9
4.4	Calcul des distances d'accélération et de décélération	9
4.5	Calcul de la distance parcourue à Vmax	9
4.6	Détermination de l'angle de direction de mouvement.....	10
4.7	Projection sur les axes X et Y	10
4.8	Conversion des distances d'accélération et décélération rotation moteur	10
4.9	Détermination des accélération et décélération en rotation/s ²	11
4.10	Détermination de la vitesse en rotation/s	11
5	Commandes de mouvement.....	12

1 Components

Photo	Type/Name	Part n°	Links
	EXCM-30-	2226101	

Table 1.1: Components/Software used

2 Introduction

2.1 Configuration matériel

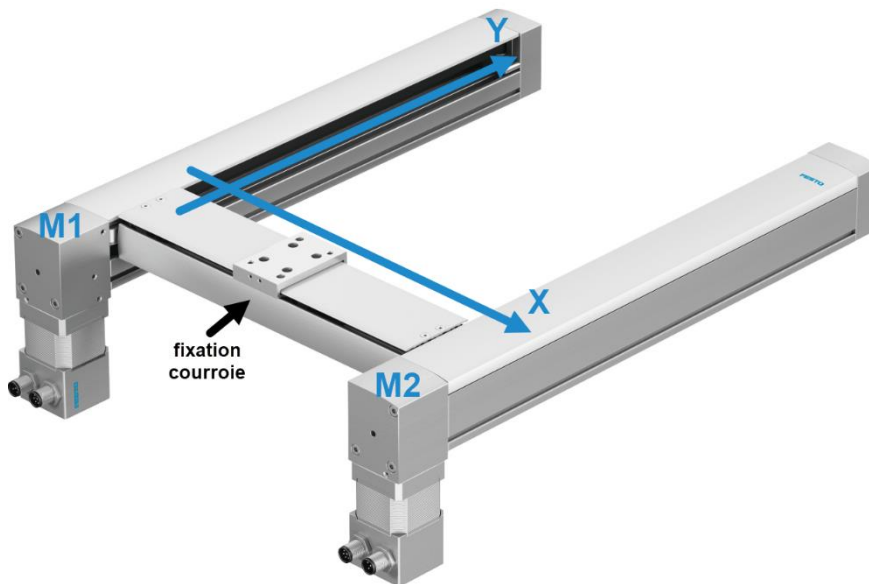


Fig. 2.1: EXCM-30

2.2 Matrices de conversion

Pour connaître les positions angulaires θ_1 et θ_2 (rad) de chaque moteurs à partir de la position cartésienne [X;Y] (mm)

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \frac{1}{rayon} * \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}$$

Pour déterminer la position cartésienne [X;Y] à partir des positions angulaire θ_1 et θ_2 (rad)

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \frac{rayon}{2} * \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

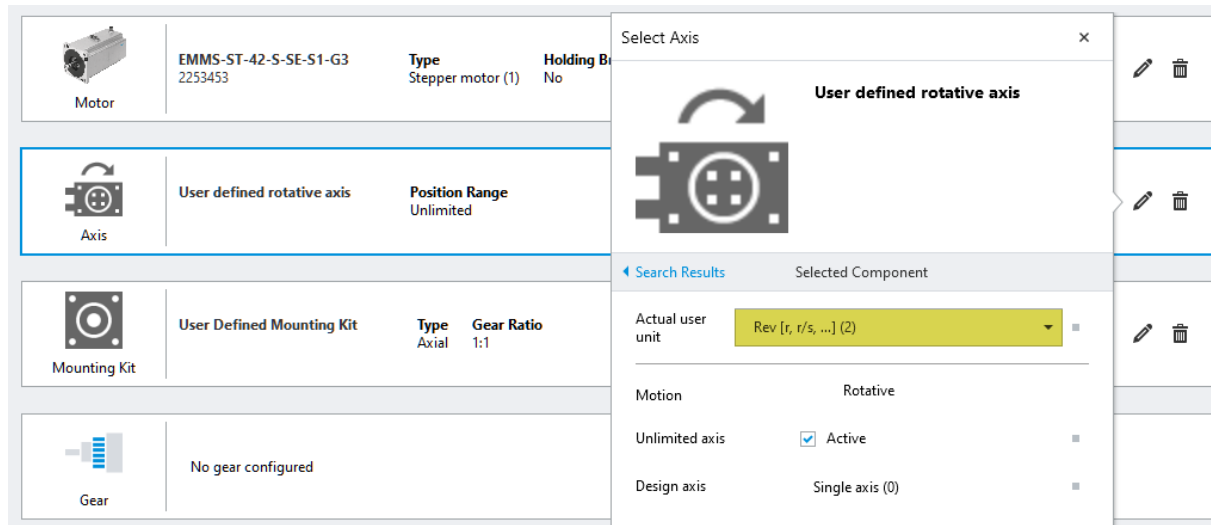
Les contrôleurs moteurs sont configurés chacun avec leur moteur et l'unité de mesure est le radian (rad ; rad/s ; rad/s²)

 Servo drive	CMMT-ST-C8-1C-EC-S0 8084005 Licenses	Maximum Current 10,00 A	Intermediate Circuit Voltage 48,00 V	Supply Voltage 24,00 V	 			
 Motor	EMMS-ST-42-S-SE-S1-G3 2253453	Type Stepper motor (1)	Holding Brake No	Encoder Protocol Incremental (4)	Encoder Type None (0)	Voltage 48,00 V	Virtual Mode Deactivated	 
 Axis	User defined rotative axis	Position Range Unlimited	 					
 Mounting Kit	User Defined Mounting Kit	Type Axial	Gear Ratio 1:1	 				

Introduction

L'utilisation des radians n'est pas imposé, il est tout à fait possible de changer l'unité de mesure et adapter les matrices en conséquence.

Par exemple en tours Rev [r, r/s, r/s² ...]



$$\begin{bmatrix} \theta 1 \\ \theta 2 \end{bmatrix} = \frac{1}{\text{avance par tour}} * \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \frac{\text{avance par tour}}{2} * \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \theta 1 \\ \theta 2 \end{bmatrix}$$

L'avance par tour d'un EXCM-30 est de 38mm/tour.

Si l'on applique la matrice pour connaitre la position cartésienne cela donne 2 équations :

$$\theta 1 = PosMot1$$

$$\theta 2 = PosMot2$$

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

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

3 Gestion des trajectoires

Avec ces conversions il est alors possible d'envoyer des consignes cartésiennes $[X,Y]$, mais le mouvement ne sera pas rectiligne, et l'on obtiendra un déplacement similaire à la trajectoire rouge.

Pour obtenir un déplacement rectiligne (trajectoire verte) il est nécessaire de déterminer la dynamique de chaque moteur de manière à ce que leur temps de déplacement soit égaux.

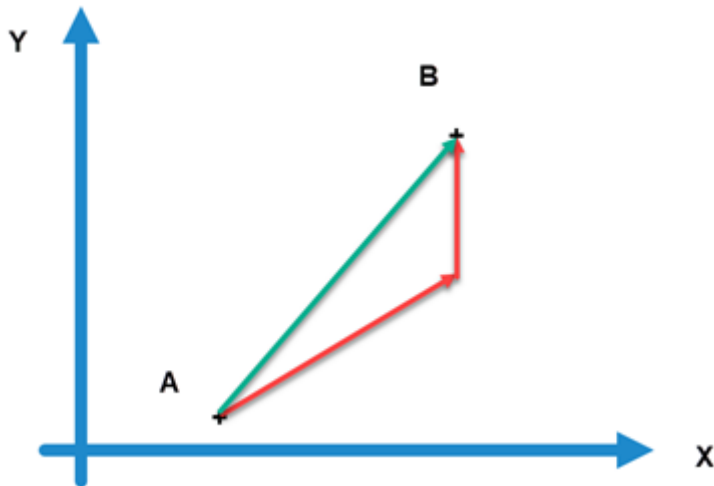


Fig. 3.1: Trajectoires des mouvements

Des équations supplémentaires permettront de calculer les vitesses, accélérations et décélérations pour chaque moteur.

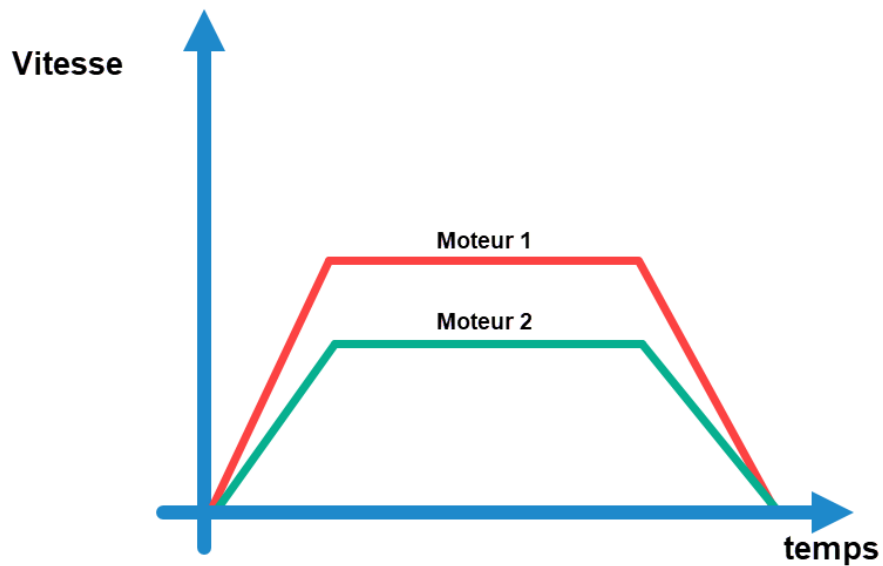


Fig. 3.2: Vitesses combinées des moteurs

4 Equations cinématique

Position de départ	Consignes	Inconnues à déterminer
Position angulaire M1 (θ_{1A})	Position X_B (mm)	Position angulaire M1 (θ_{1B})
Position angulaire M2 (θ_{2A})	Position Y_B (mm)	Position angulaire M2 (θ_{2B})
Position X_A (mm)	Vitesse (mm/s)	Vitesse M1
Position Y_A (mm)	Accélération (mm/s ²)	Vitesse M2
	Déccélération (mm/s ²)	Accélération M1
		Déccélération M1
		Accélération M2
		Déccélération M2

Table 4.1: données d'entrées

Les unités de mesures des inconnues à déterminer sont fonction du paramétrage fait dans le logiciel Festo Automation Suite.

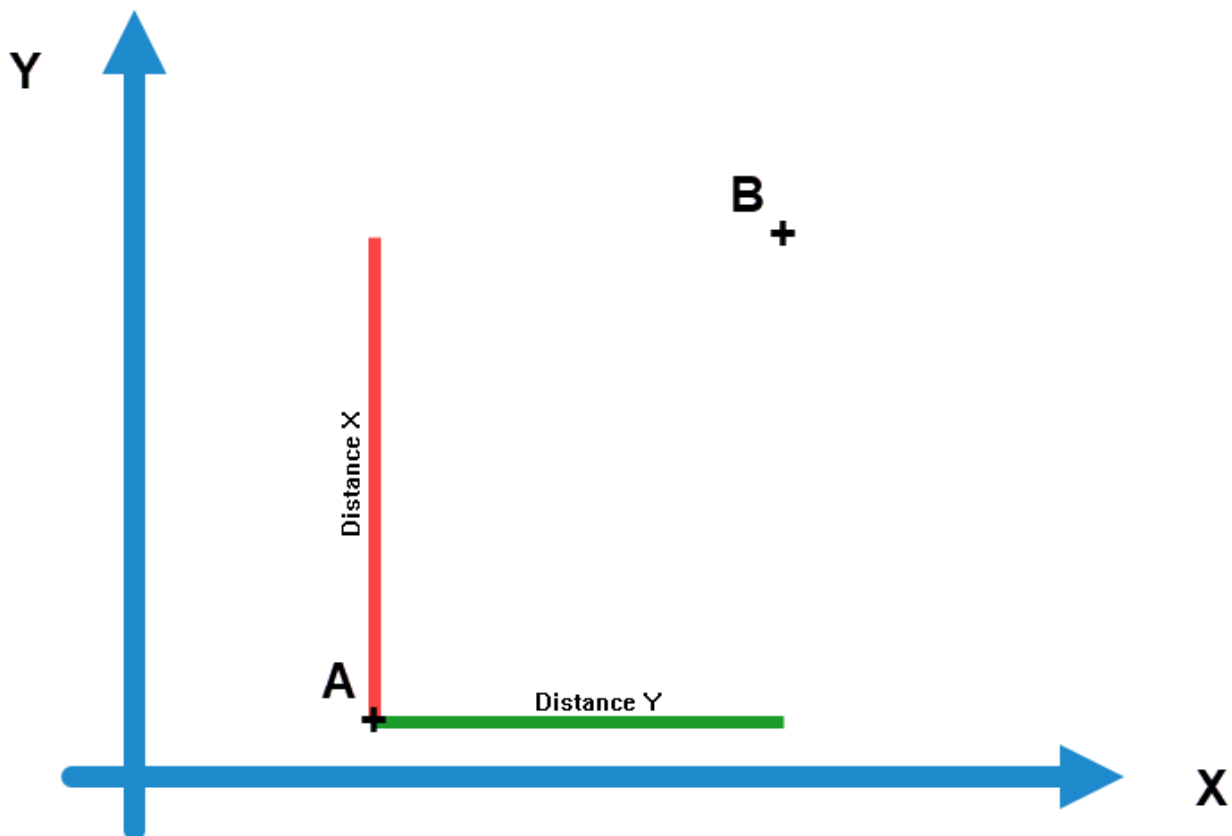
4.1 Calcul des distances à parcourir par axes(mm) et par moteur (trs)

Distance X_{AB} = position actuelle X_A - consigne à atteindre X_B

Distance Y_{AB} = position actuelle Y_A - consigne à atteindre Y_B

Distance angulaire M1 = position actuelle θ_{1A} - consigne à atteindre θ_{1B}

Distance angulaire M2 = position actuelle θ_{2A} - consigne à atteindre θ_{2B}

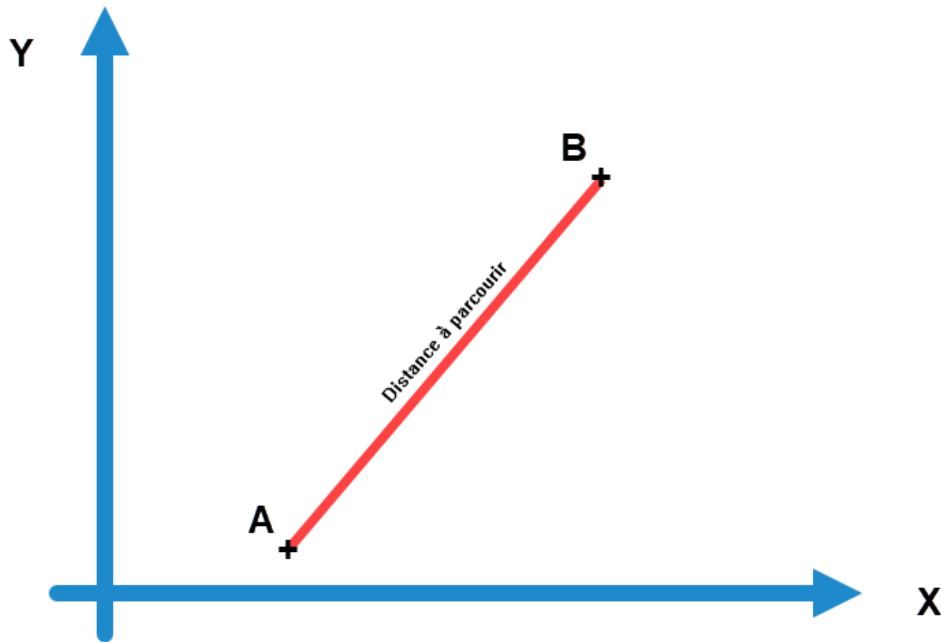


A = position actuelle.

B = consigne de position à atteindre.

4.2 Calcul de la distance à parcourir [X,Y], (Pythagore)

$$\text{Distance à parcourir} = \sqrt{\text{distance sur } X^2 + \text{distance sur } Y^2}$$



4.3 Calcul des temps d'accélération et décélération

$$\text{temps}_{acc} = \frac{\text{Vitesse}}{\text{accélération}}$$

$$\text{temps}_{dec} = \frac{\text{Vitesse}}{\text{décélération}}$$

4.4 Calcul des distances d'accélération et de décélération

$$\text{distance acc} = \frac{1}{2} * \text{accélération} * \text{temps}_{acc}^2$$

$$\text{distance dec} = \frac{1}{2} * \text{décélération} * \text{temps}_{dec}^2$$

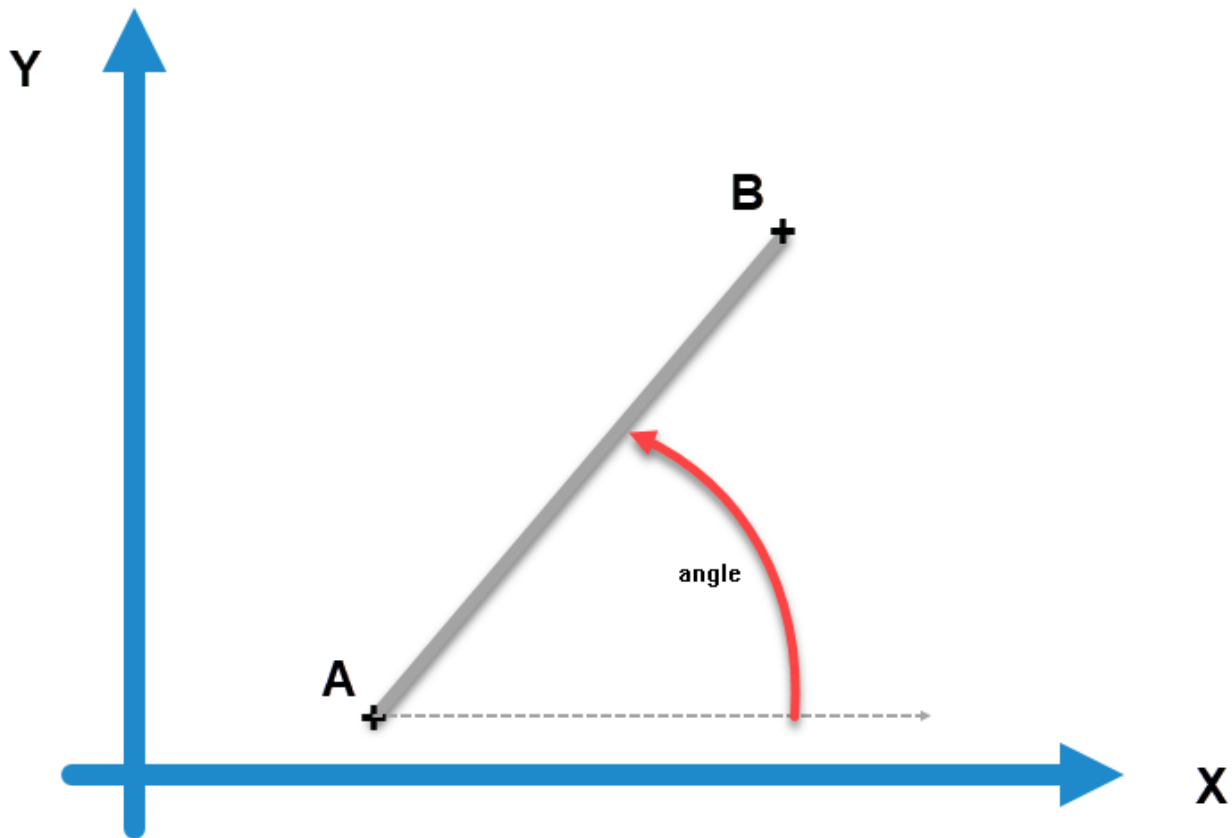
4.5 Calcul de la distance parcourue à Vmax

$$\text{distance } V_{max} = \text{distance à parcourir} - (\text{distance Acc} + \text{distance Dec})$$

Formule optionnelle, pour déterminer si on peut atteindre la vitesse maximum $\text{distance } V_{max} > 0$

4.6 Détermination de l'angle de direction de mouvement

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



4.7 Projection sur les axes X et Y

α = angle

distance Acc sur X = $\sin \alpha$ * *distance Acc*

distance Dec sur X = $\sin \alpha$ * *distance Dec*

distance Acc sur Y = $\cos \alpha$ * *distance Acc*

distance Dec sur Y = $\cos \alpha$ * *distance Dec*

4.8 Conversion des distances d'accélération et décélération rotation moteur

On considère que l'unité paramétré est : Rev [r, r/s, r/s² ...] et que l'avance par tours = 38mm

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

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

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

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

4.9 Détermination des accélération et décélération en rotation/s²

$$\text{Acc M1} = 2 * \frac{\text{Distance Acc M1}}{\text{Temps Acc}^2}$$

$$\text{Dec M1} = 2 * \frac{\text{Distance Dec M1}}{\text{Temps Dec}^2}$$

$$\text{Acc M2} = 2 * \frac{\text{Distance Acc M2}}{\text{Temps Acc}^2}$$

$$\text{Dec M2} = 2 * \frac{\text{Distance Dec M2}}{\text{Temps Dec}^2}$$

4.10 Détermination de la vitesse en rotation/s

$$\text{Vitesse M1} = \text{Acc M1} * \text{temps Acc}$$

$$\text{Vitesse M2} = \text{Acc M2} * \text{temps Acc}$$

5 Commande de mouvements

Dans cet exemple la bibliothèque Festo point à point pour Codesys 3.5 est utilisée, elle est disponible sur le portail d'assistance Festo.



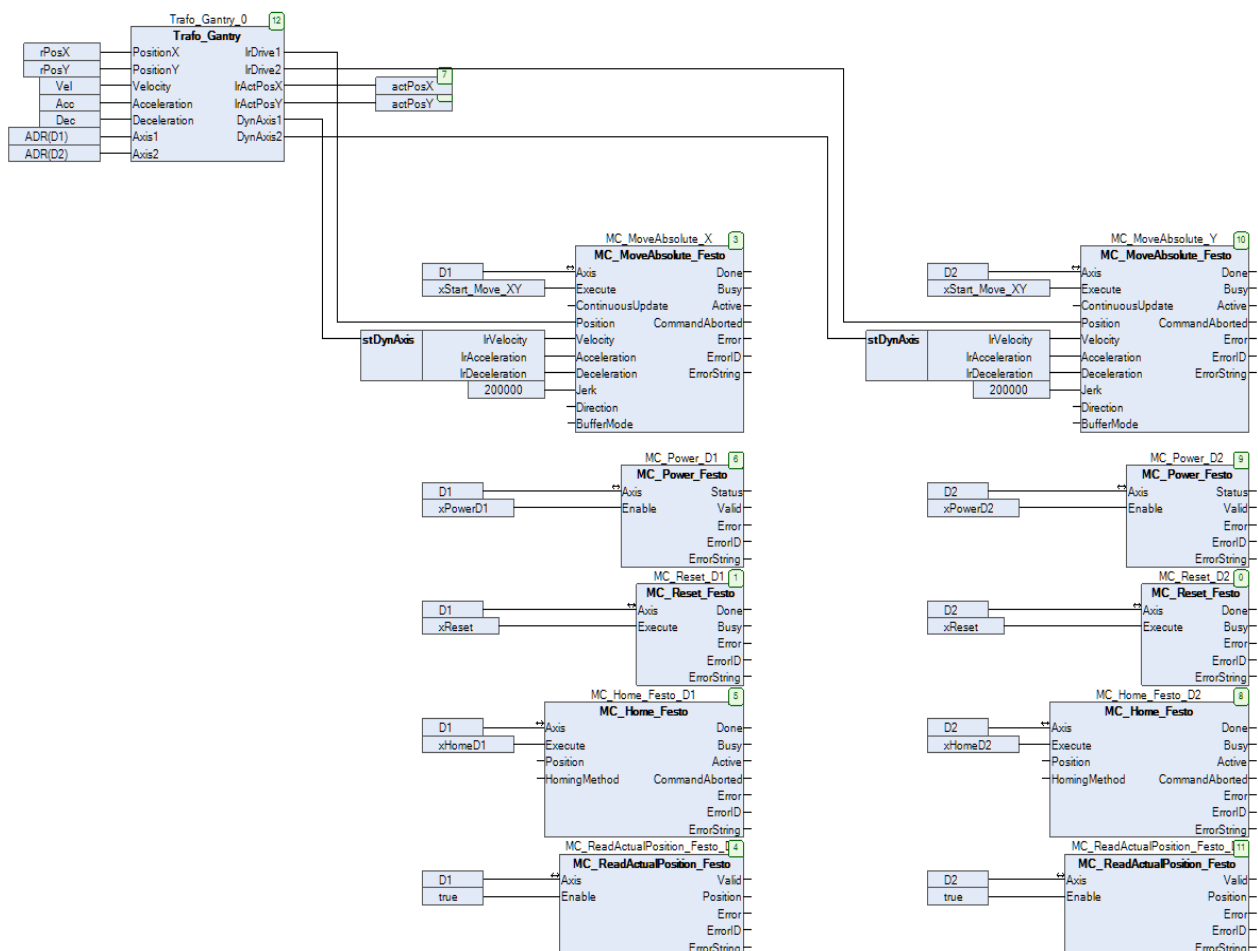
Function blocks CODESYS

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.

Tous les calculs sont implémentés dans un bloc fonction nommé Trafo_Gantry, le code source est présenté à la page suivante. Chaque moteur peut ensuite être contrôlé individuellement et simultanément en appliquant les consignes calculées pour chacun : Position, Accélération, Vitesse et Décélération.

Position, Accélération, Vitesse et Décélération.



```

1 // needed for axis Position, could be replaced by MC_ReadActualPosition_Festo
2 D1:=Axis1^;
3 D2:=Axis2^;
4
5 //Hgantry Conversion Matrices
6 lrDrive1:=(1/lrRadius)*((-1*PositionX)+(-1*PositionY));
7 lrDrive2:=(1/lrRadius)*((-1*PositionX)+(1*PositionY));
8
9 lrActPosX:=(lrRadius/2)*((-1*D1.ActualPosition)+(-1*D2.ActualPosition));
10 lrActPosY:=(lrRadius/2)*((-1*D1.ActualPosition)+(1*D2.ActualPosition));
11
12 // Calculation of distances by Axis (mm) and motor (rev)
13 lrDistX:=lrActPosX-PositionX;
14 lrDistY:=lrActPosY-PositionY;
15
16 lrDistD1:=D1.ActualPosition-lrDrive1;
17 lrDistD2:=D2.ActualPosition-lrDrive2;
18
19 // Calculation of the distance to be covered [X,Y], (Pythagoras)
20 lrDistCart:=SQRT(EXPT(lrDistX,2)+EXPT(lrDistY,2));
21
22 // Calculation of acceleration and deceleration times
23 lrTpsAcc:=Velocity/Acceleration;
24 lrTpsDec:=Velocity/Deceleration;
25
26 // Calculation of acceleration and deceleration distances
27 lrDistAcc:=0.5*Acceleration*EXPT(lrTpsAcc,2);
28 lrDistDec:=0.5*Deceleration*EXPT(lrTpsDec,2);
29
30 // Determination of the Direction of Motion Angle
31 lrAngle:=ATAN(lrDistY/lrDistX);
32
33 // X & Y axis projection
34 lrDistAccX:=SIN(lrAngle)*lrDistAcc;
35 lrDistDecX:=SIN(lrAngle)*lrDistDec;
36 lrDistAccY:=COS(lrAngle)*lrDistAcc;
37 lrDistDecY:=COS(lrAngle)*lrDistDec;
38
39 // Conversion of Acceleration Distances and Engine Rotation Deceleration
40 lrDistAccD1:=ABS((1/lrRadius)*((1*lrDistAccX)+(1*lrDistAccY)));
41 lrDistDecD1:=ABS((1/lrRadius)*((1*lrDistDecX)+(1*lrDistDecY)));
42 lrDistAccD2:=ABS((1/lrRadius)*((1*lrDistAccX)+(-1*lrDistAccY)));
43 lrDistDecD2:=ABS((1/lrRadius)*((1*lrDistDecX)+(-1*lrDistDecY)));
44
45 // Determination of Rotational Acceleration and Deceleration/s²
46 DynAxis1.lAcceleration:=2*lrDistAccD1/(lrTpsAcc*lrTpsAcc);
47 DynAxis1.lDeceleration:=2*lrDistDecD1/(lrTpsDec*lrTpsDec);
48 DynAxis2.lAcceleration:=2*lrDistAccD2/(lrTpsAcc*lrTpsAcc);
49 DynAxis2.lDeceleration:=2*lrDistDecD2/(lrTpsDec*lrTpsDec);
50
51 // Determination of velocity, rev/s
52 DynAxis1.lVelocity:=DynAxis1.lAcceleration*lrTpsAcc;
53 DynAxis2.lVelocity:=DynAxis2.lAcceleration*lrTpsAcc;

```