

# Industrial Automation Base

Cours AutB

Author: [Cédric Lenoir](#)

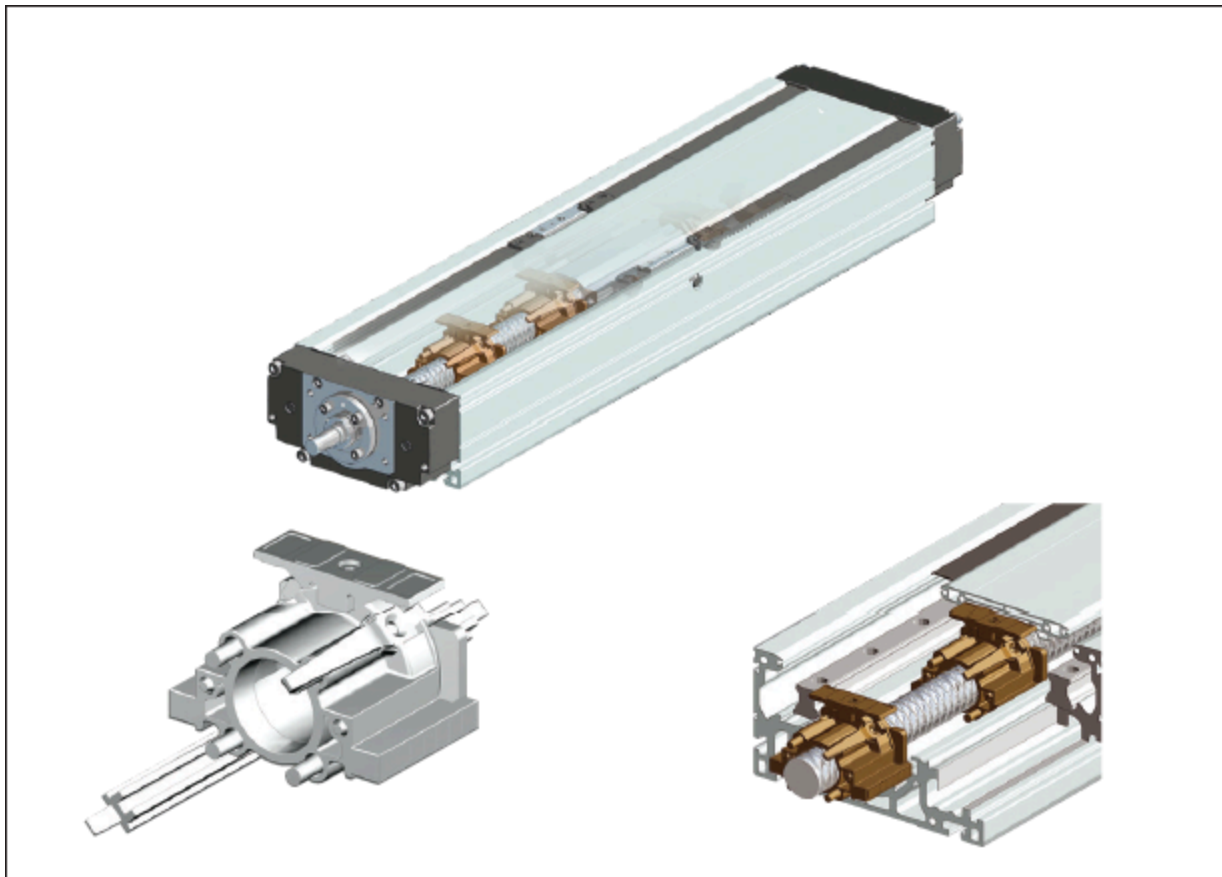
## Module 07 le drive, software.

**Cette partie est actuellement traitée  
directement dans le cadre du laboratoire:  
Voir: [LAB 05 Mise en service d'un axe  
électrique avec une vis à bille](#)**

## Régulateur PI

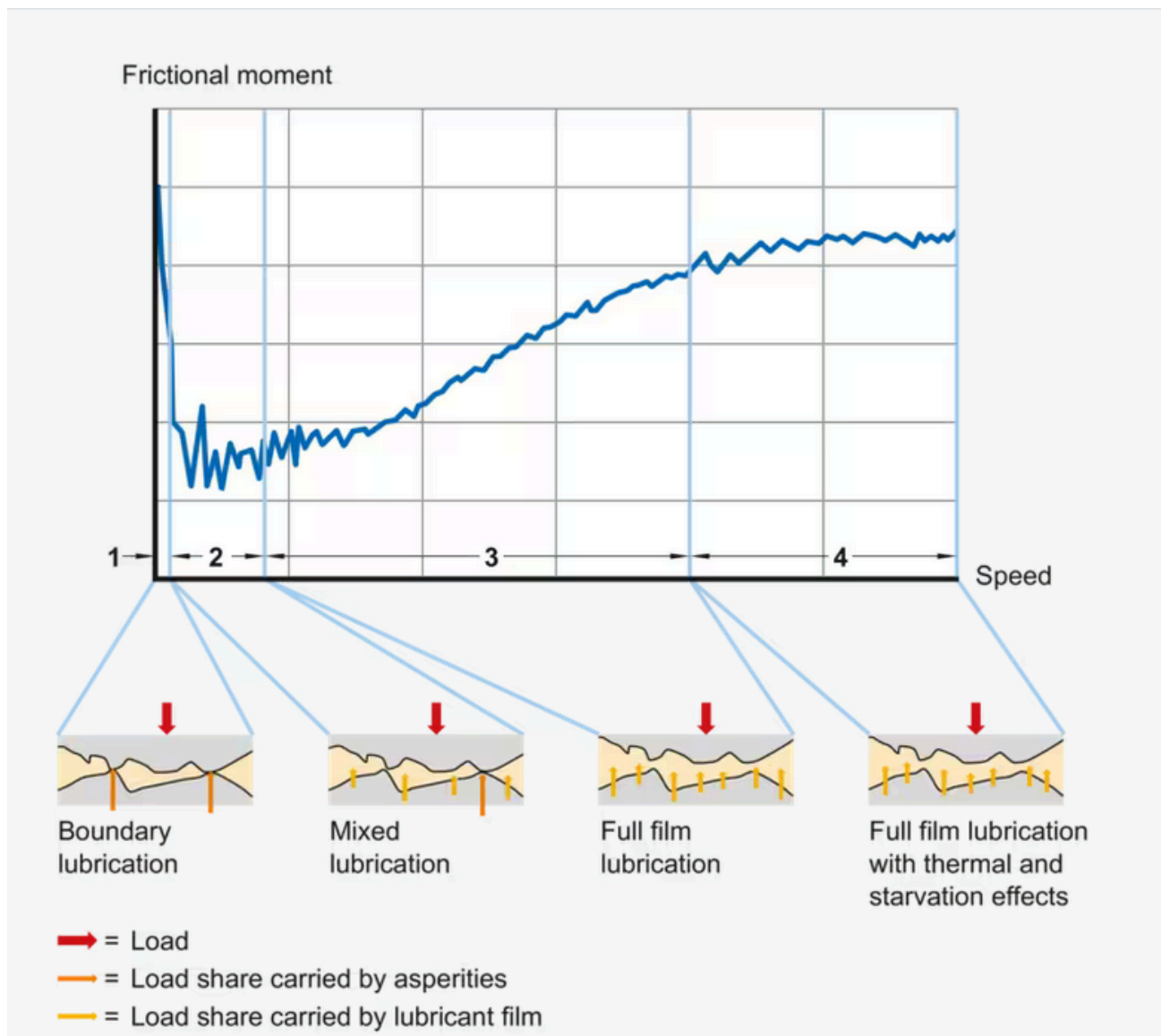
La théorie sur les régulateurs PI repose sur certains principes de linéarité des signaux. Dans la réalité, comme nous pouvons le voir dans le cadre du laboratoire, de nombreuses composantes ne sont absolument pas linéaires.

Ci-dessous, un aperçu des roulements de la vis à bille dans un axe CKK tel que dans le laboratoire d'automation.



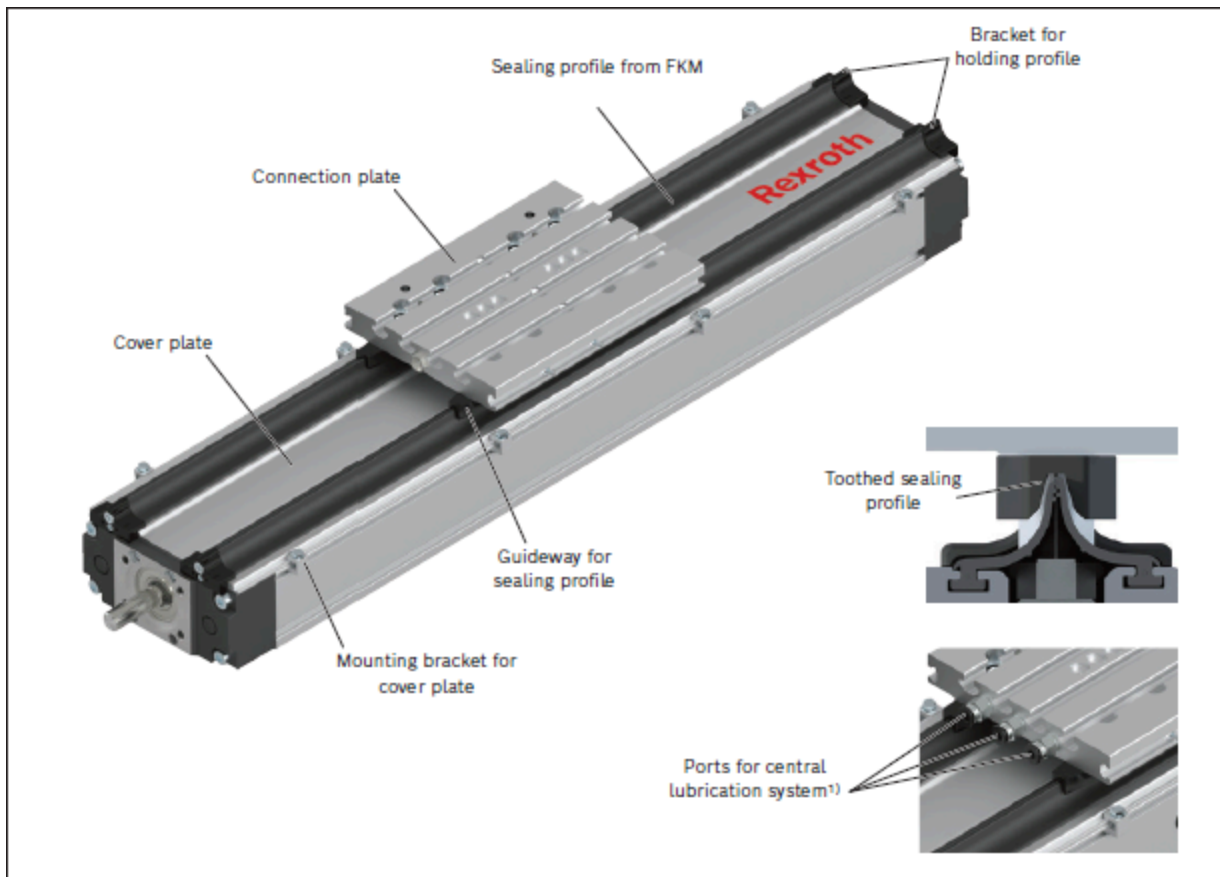
CKK Internal View

Selon un figure d'origine SFK, fournisseur de roulement à bille, on peut observer un exemple de variation du frottement des roulements à bille en fonction de la vitesse.



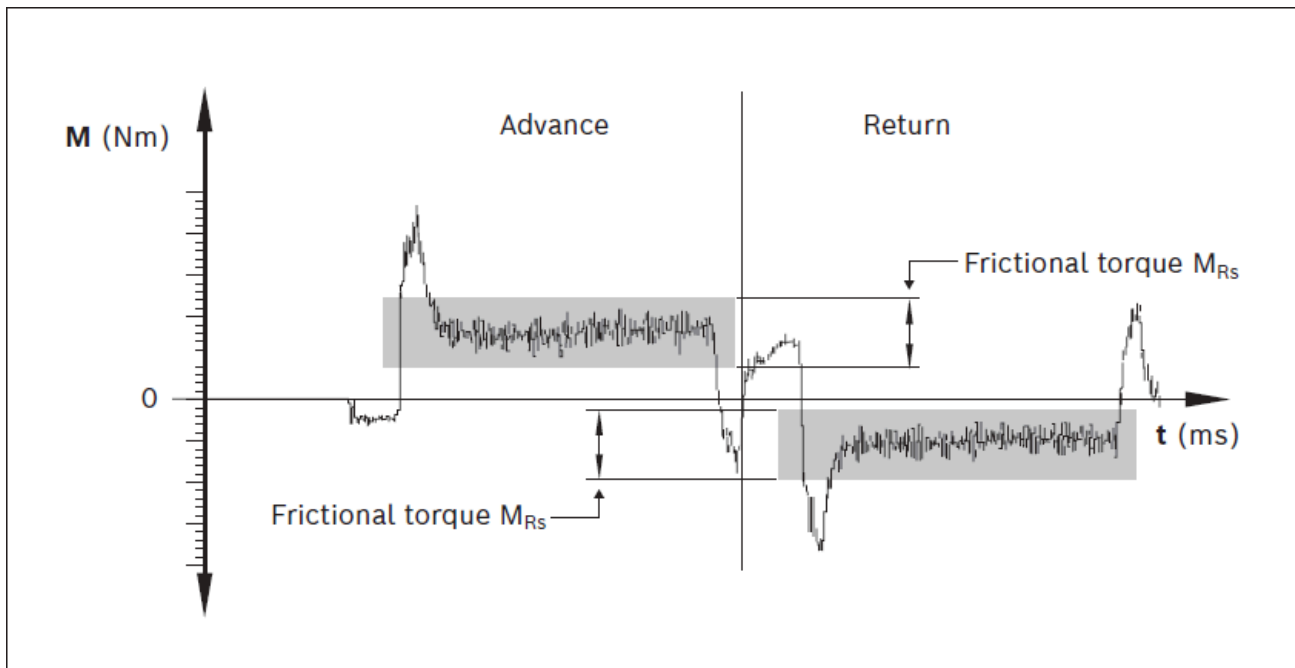
Source SKF, friction as a function of speed

Une vue externe d'un module SKF montre que nous avons en sus, des lamelles en caoutchouc protègent la mécanique interne de la poussière.



CKK External View

Nous obtenons, selon documentation Bosch Rexroth, un exemple de la variation du frottement en fonction du temps sur un mouvement aller et retour sur l'ensemble de la course d'un système.



Measurement of frictional torque of complete system

Finalement, il conviendra de tenir compte de l'ensemble de la structure en mouvement, qui inclu des câbles dont le comportement est compliqué à estimer ainsi que de la rigidité de l'ensemble qui en plus

des contraintes de masses, moments d'inertie et frottement ajoutera une composante élastique.



HEVS Lab, system overview

## Un pas plus loin

La régulation en position des moteurs synchrones utilise souvent le même schéma avec dans l'ordre:

- La commande de position,
- un régulateur de position,
- un régulateur de vitesse,
- un régulateur de courant qui via une constante de force ou couple convertit le courant en grandeur mécanique.

## Régulateur de courant

Le régulateur de courant dépend des caractéristiques électriques du moteur qui sont en principe bien connues et entrées comme paramètres dans le drive. Sauf exception, il n'est pas nécessaire de

modifier les paramètres du régulateur de courant qui sont calculés par le drive.

## Régulateur de vitesse

Il dépend des paramètres mécaniques du système et est donc particulièrement compliqué à mettre au point quand les paramètres mécaniques sont non linéaires. C'est pourquoi on utilise souvent une base de travail empirique ou un algorithme d'**autotuning** interne au drive.

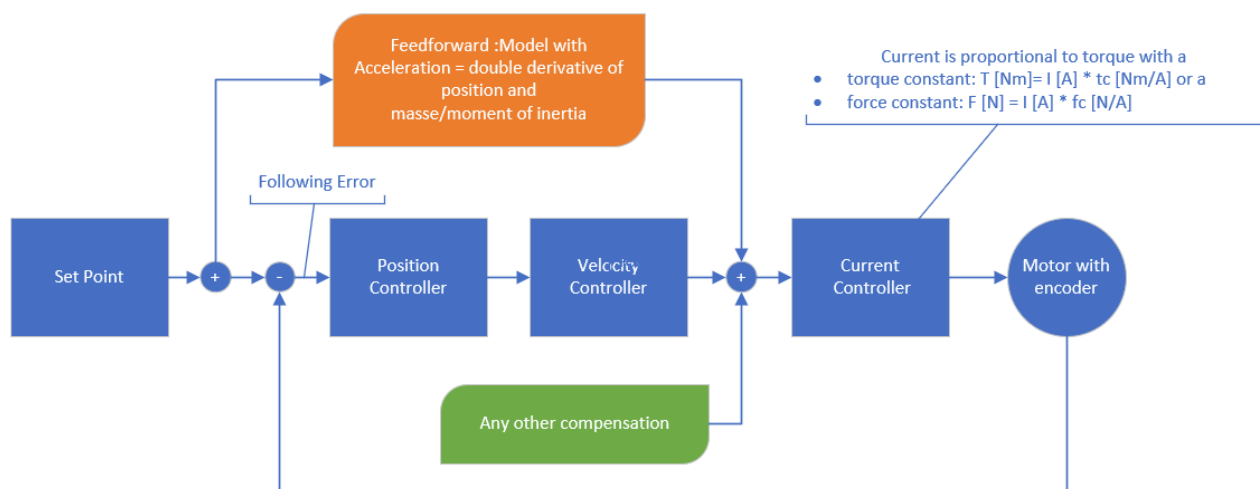
## Régulateur de position

Il dépend du type de processus. Par défaut un régulateur proportionnel avec un facteur 1 est souvent suffisant.

Dans l'exemple ci-dessous, on ajoute un feedforward.

Le principe est de convertir directement une variation de position en courant à condition de connaître la masse ou le moment d'inertie du système à piloter. Le feedforward convient bien au suivi de position et permet de limiter l'erreur de poursuite à savoir l'écart entre la position de commande et la position mesurée.

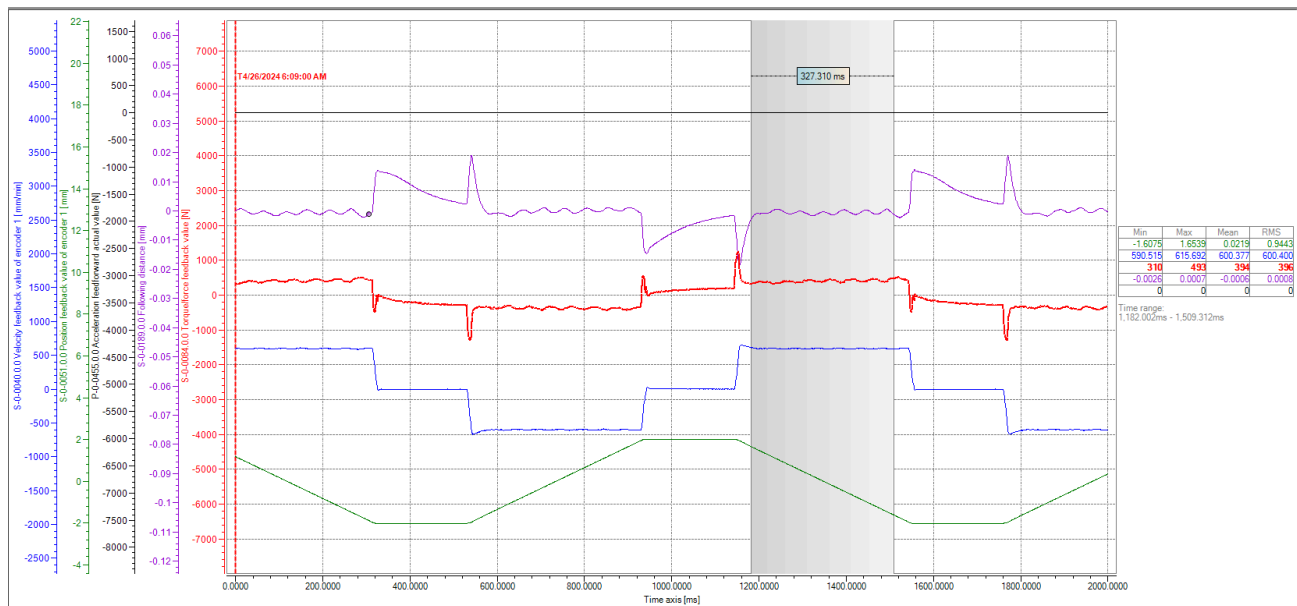
On peut garder en tête qu'un régulateur sert avant tout à réagir aux perturbations non mesurables mais que tout ce qui permettra d'anticiper, à mettre en relation avec **forward**, améliorera le système.



Feedforward or application of a model

Le principe est d'apprendre à comprendre un système en visualisant les différentes composantes du mouvement. L'avantage de comprendre le système et de maîtriser la programmation est qu'il sera ensuite possible de compléter soit même le model général de pilotage d'un système.

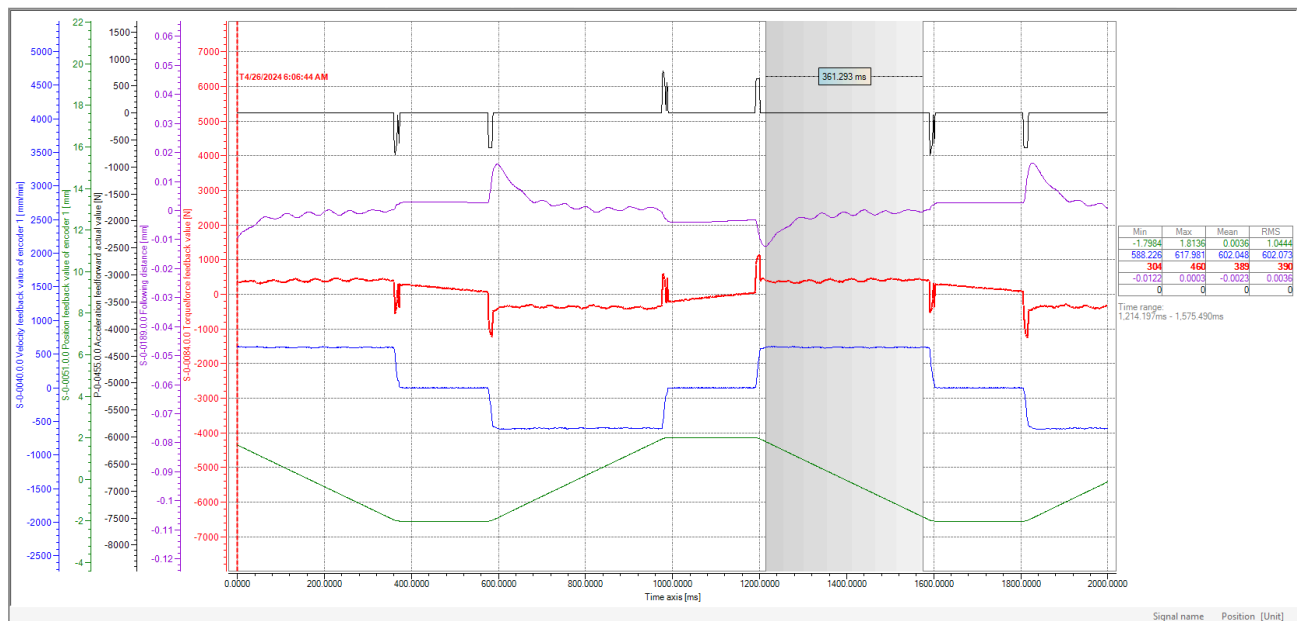
# Exemple sans feedforward



Motion without feedforward

En analysant ce mouvement à faible vitesse, +/- 2 [mm] et 10 mm/s, dans le graphe, 600 mm/min, on peut visualiser par exemple la force de frottement dynamique.

# Exemple avec feedforward



Motion with feedforward

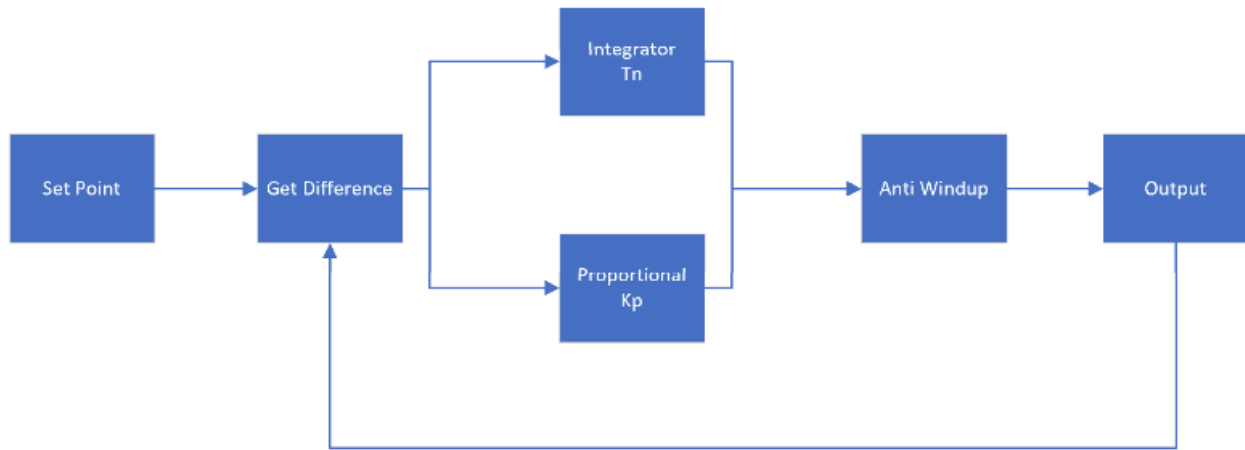
Ici, on peut visualiser le feedforward en noir et constater qu'il diminue efficacement l'écart de poursuite en violet pendant la phase d'accélération, augmentation de la vitesse en bleu.

On peut aussi constater que le feedforward permet de stabiliser plus rapidement la vitesse en bleu et évite un dépassement de la vitesse, la petite bosse en augmentation de vitesse sur le premier



schéma.

## Exemple, écrire un régulateur PI.



Simple PI regler



# Pseudo code:

```
Proportional Gain : Kp
Integration Time  := Tn
Integration Gain  := Ki

// Pseudo code PI Regler

Read(Setpoint)
Read(MeasureProcess)

error := Setpoint - MeasureProcess

// Integrator
// If Tn = 0, no integration
IF Tn > 0 THEN
    Ki := 1/Tn;
ELSE
    Ki := 0;
END_IF;
sumError := sumError + error;

// Before Limit
sumPI := Kp * error + Ki * sumError;

// Set Limit called sometimes
IF sumPI > LimitHigh then
    Output := LimitHigh;
ELSIF sumPI < LimitLow then
    Output := LimitLow;
ELSE
    Output := sumPI;
END_IF

Write(Output)
```

# Your job

- Utiliser le modèle d'un FB **Enable In Operation Base** pour réaliser un régulateur PI dans un FB\_PI.
- Utiliser l'état Init pour réinitialiser le sommateur de l'intégrateur: `sumError` .

Un exemple d'implémentation est donné en annexe

# Conclusion

Si on sait écrire un régulateur PI, on pourrait aussi théoriquement écrire son propre système d'optimisation pour compenser un problème particulier.