

SPIKE PRIME LESSONS

By the Creators of EV3Lessons



SUIVI DE LA LIGNE PID

BY SANJAY AND ARVIND SESHAN



OBJECTIFS DE LA LEÇON

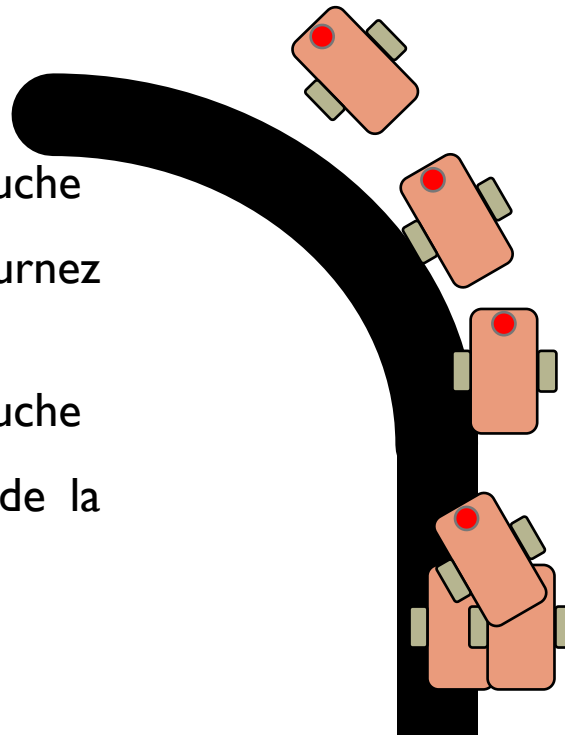
- Apprenez les limites du contrôle proportionnel
- Apprenez ce que signifie le PID
- Apprenez à programmer le PID et à le régler

QUAND LE CONTRÔLE PROPORTIONNEL POSE-T-IL PROBLÈME ?

Note : les quelques diapositives suivantes sont animées. Utilisez le mode de présentation PowerPoint pour les visualiser

Que ferait un humain ?

- En ligne → allez tout droit
- Sur le blanc → tournez à gauche
- En traversant la ligne → tournez à droite
- Sur le blanc → tournez à gauche
- Éloignez-vous encore plus de la ligne !



Que ferait un contrôle proportionnel ?

- En ligne → allez tout droit
- Sur le blanc → tournez à gauche
- En traversant la ligne → allez tout droit !
- Sur la ligne blanche → tournez à gauche
- En s'éloignant de la ligne → tournez à gauche du même nombre !

LECTURE DE LA LUMIÈRE = 50%

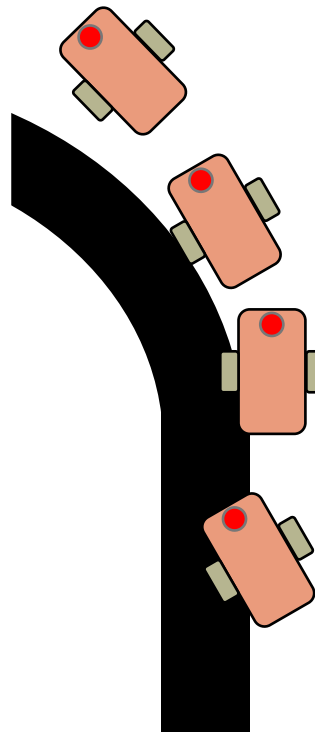
COMMENT POUVONS-NOUS RÉGLER LE PROBLÈME DU CONTRÔLE PROPORTIONNEL ?

Que ferait un humain ?

Tournez à gauche/en ligne →
tournez à droite

S'éloigner encore plus de la
ligne → tourner encore plus !

1. Prévoir le prochain
relevé du capteur



Que ferait un contrôle proportionnel ?

**Tournez à gauche/en ligne
→ allez tout droit !**

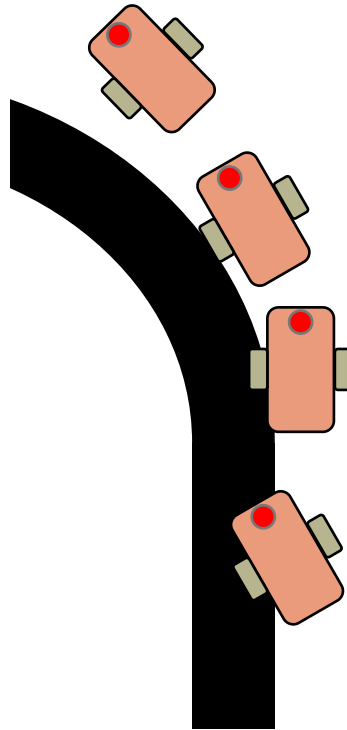
**Éloignez-vous de la ligne →
tournez à gauche du
même nombre !**

2. Les corrections de
pilotage passées ont-elles
contribué à réduire les
erreurs ?

INTÉGRALES ET DÉRIVÉS

1. Prévoir le prochain relevé du capteur

- Si les lectures sont : 75, 65, 55, quelle sera la prochaine lecture, selon vous ?
 - Et si les lectures étaient 57, 56, 55...
- Quelles informations avez-vous utilisées pour deviner ?
- Dérivée --> Le taux auquel une valeur change



2. Les corrections de direction passées ont-elles contribué à réduire les erreurs ?

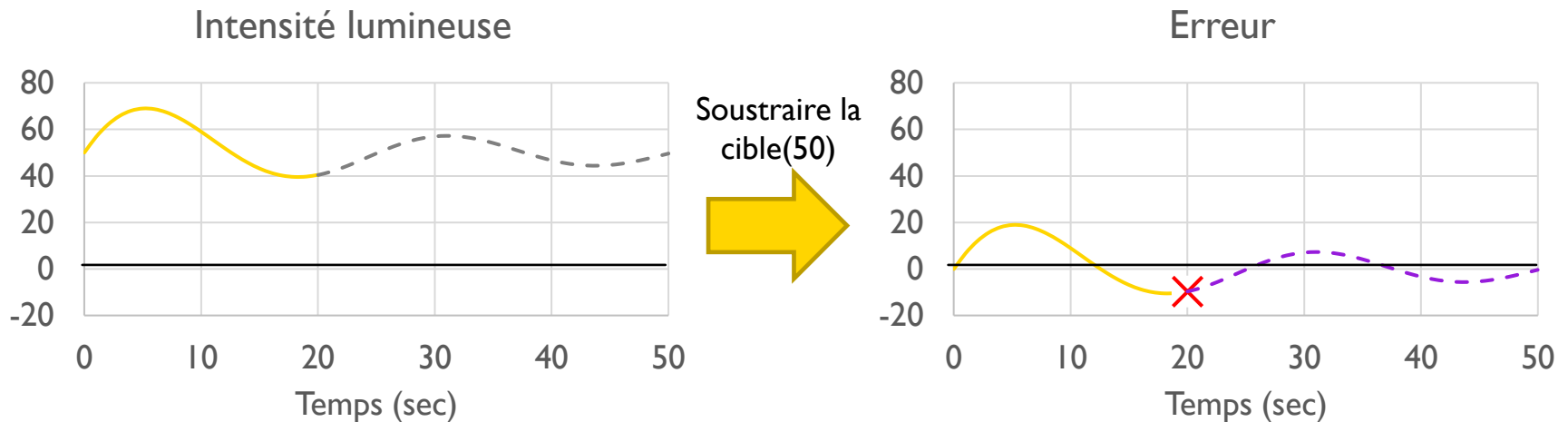
- Lorsque la correction fonctionne bien, à quoi ressemblent les erreurs de lecture ?
 - +5, -6, +4 -3... c'est-à-dire rebondissant autour de 0
- Lorsque le pilotage ne fonctionne pas, à quoi ressemble l'erreur ?
 - +5, +5, +6, +5... c'est-à-dire toujours d'un côté de 0
- Comment pouvons-nous le détecter facilement ?
 - Conseil : examinez la somme de toutes les erreurs passées
- Quelle est la valeur idéale de cette somme ? Qu'est-ce que cela signifie si la somme est importante ?
- Intégrale --> La "somme" des valeurs

QU'EST-CE QUE LE PID ?

- **P**roportionnelle [Erreur] → Quelle est la situation actuelle ?
- **I**ntégrale → Mes corrections passées ont-elles aidé à arranger les choses ?
- **D**érivée → Comment la situation évolue-t-elle ?
- Le contrôle **PID** → combine les valeurs d'erreur, d'intégrale et de dérivée pour décider de la manière de diriger le robot

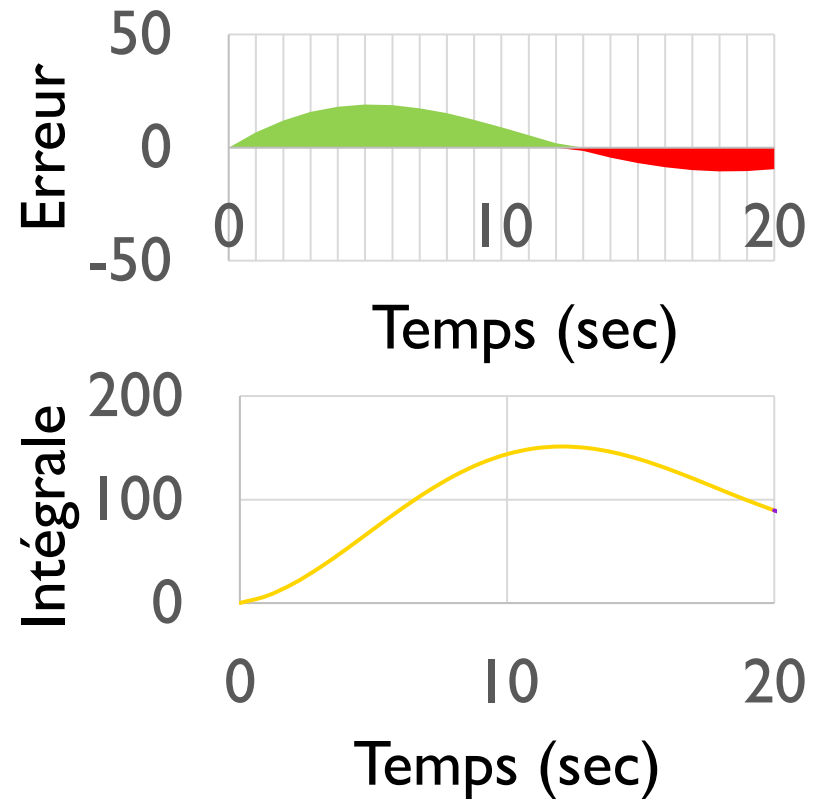
ERREUR

- La ligne continue représente ce que vous avez vu, la ligne pointillée est l'avenir
- Au temps 20 sec, vous voyez une lecture lumineuse = 40 et une erreur = -10 (X rouge)



INTÉGRALE

- Regard sur l'histoire passée du suiveur de ligne
- Somme des erreurs passées
- Surface similaire sous la courbe dans le graphique (intégrale)
 - Vert = zone positive
 - Rouge = zone négative



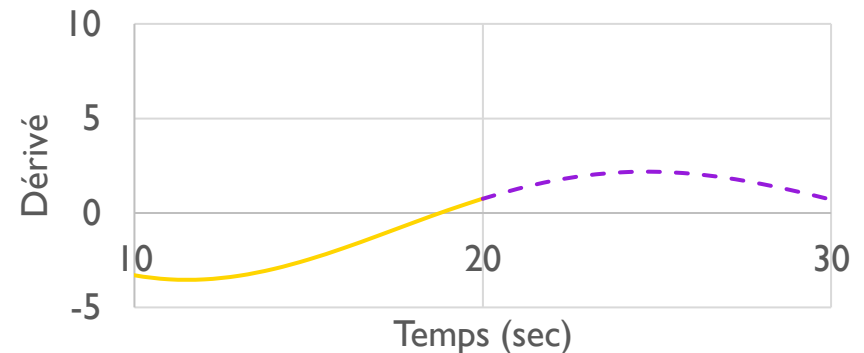
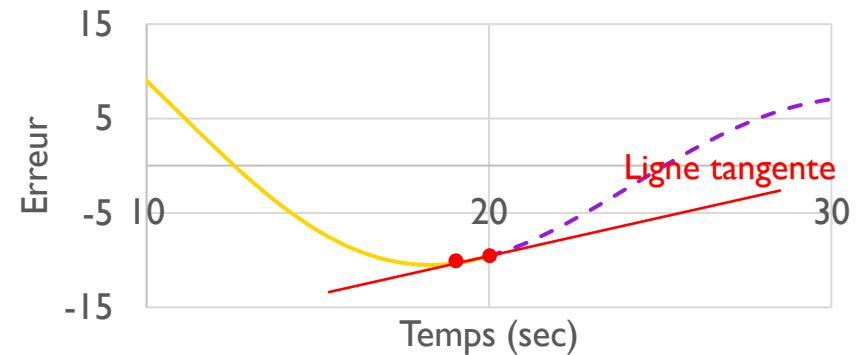
DÉRIVÉE

■ À quelle vitesse la position change-t-elle ?

- Prévoir où se trouvera le robot dans un avenir immédiat
- Même chose que la vitesse à laquelle l'erreur change

■ Peut être mesurée en utilisant la tangente aux mesures --> dérivée

- Approximatif en utilisant deux points proches sur le graphique



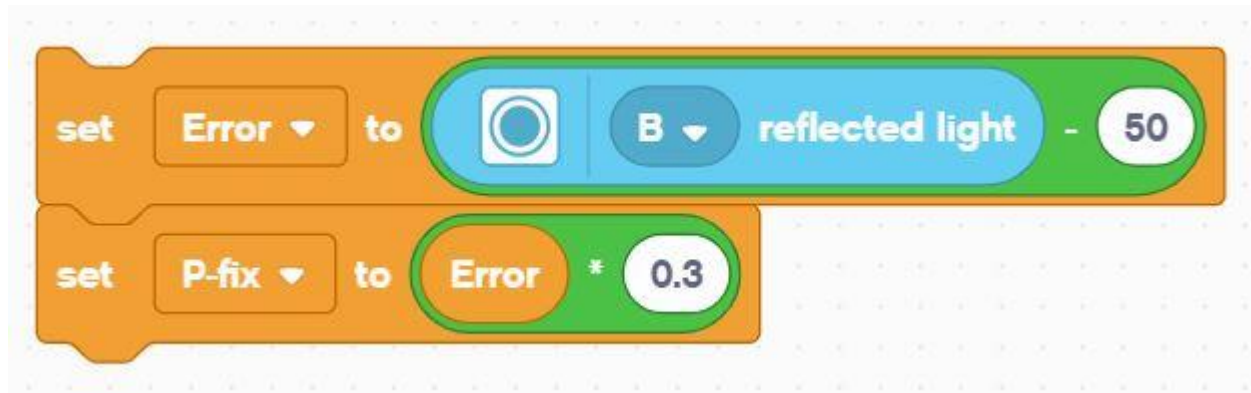
PSEUDO-CODE

1. Faites une nouvelle lecture du capteur de lumière
2. Calculez l'"erreur«
3. Mettez à l'échelle l'erreur pour déterminer la contribution à la mise à jour du pilotage (contrôle proportionnel)
4. Utilisez l'erreur pour mettre à jour l'intégrale (somme de toutes les erreurs passées)
5. Mettez à l'échelle l'intégrale pour déterminer la contribution à la mise à jour du pilotage (contrôle intégral)
6. Utilisez l'erreur pour mettre à jour le dérivé (différence par rapport à la dernière erreur)
7. Mettez à l'échelle le dérivé pour déterminer la contribution à la mise à jour du pilotage (contrôle du dérivé)
8. Combinez les réactions P, I et D et dirigez le robot

CODE - PROPORTIONNEL

- C'est la même chose que le code de contrôle proportionnel

Erreur = distance de la ligne = lecture - cible

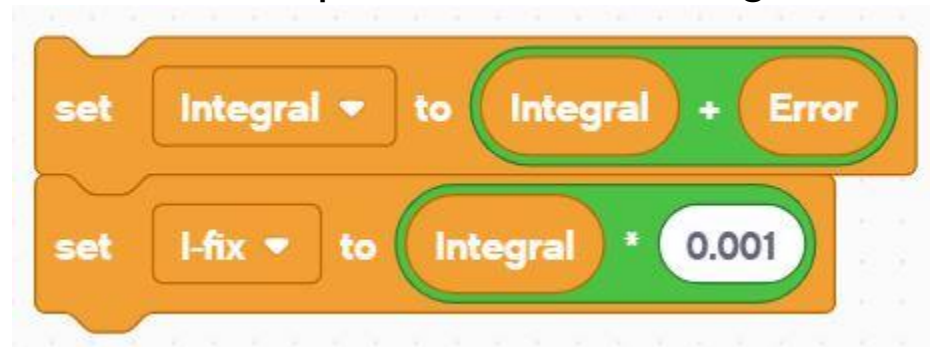


Correction (P_fix) = Erreur mise à l'échelle par la constante proportionnelle (Kp) = 0,3

CODE - INTÉGRAL

- Cette section calcule l'intégrale. Elle ajoute l'erreur actuelle à une variable qui a la somme de toutes les erreurs précédentes.
- La constante d'échelle est généralement petite puisque l'intégrale peut être grande

Intégrale=somme de toutes les erreurs passées=dernière intégrale + erreur la plus récente

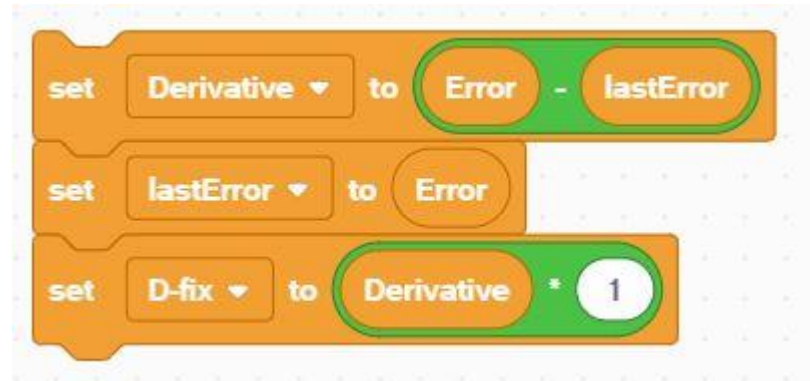


Correction (I_fix) = Intégrale mise à l'échelle par la constante proportionnelle (Ki) = 0,001

CODE - DÉRIVÉ

- Cette section du code calcule le dérivé. Elle soustrait l'erreur actuelle de l'erreur passée pour trouver la variation de l'erreur.

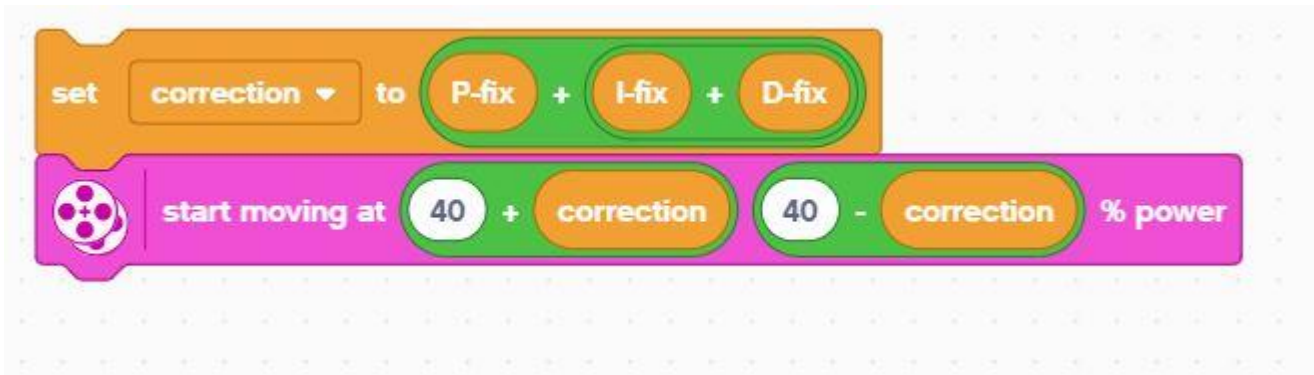
Dérivé = taux de variation de l'erreur = erreur actuelle - dernière erreur



Correction (D_fix) = Dérivé mis à l'échelle par une constante proportionnelle (Kd) = 1,0

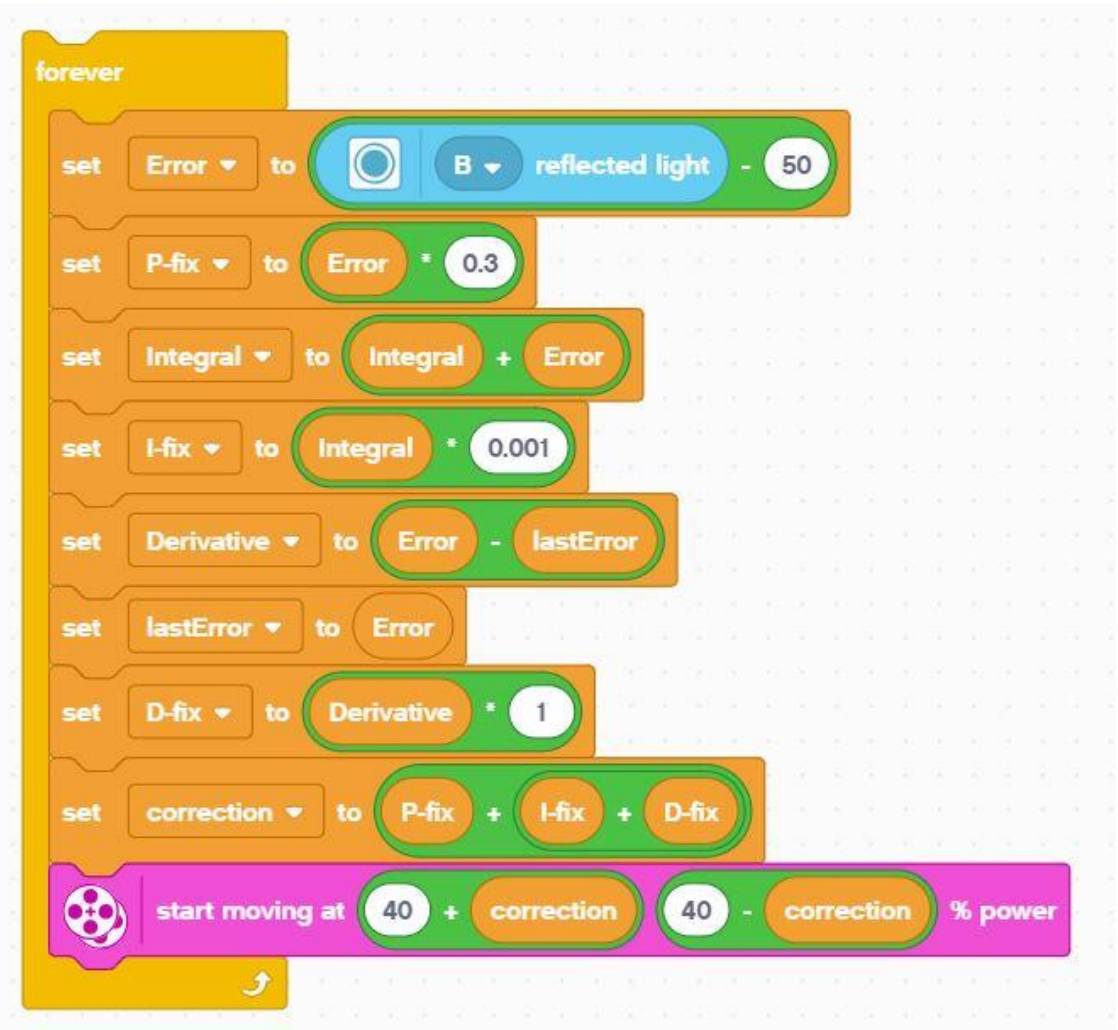
TOUT METTRE ENSEMBLE

- Chacun des composants a déjà été mis à l'échelle. À ce stade, nous pouvons simplement les additionner
- Ajoutez les trois corrections pour P, I et D. Cela permettra de calculer la correction finale
- Dans SPIKE Prime, nous utilisons le % de puissance pour que les moteurs soient non régulés



CODE COMPLET

- C'est ce que vous obtenez si vous mettez toutes ces parties ensemble
- Nous espérons que vous comprenez maintenant un peu mieux comment fonctionne le PID



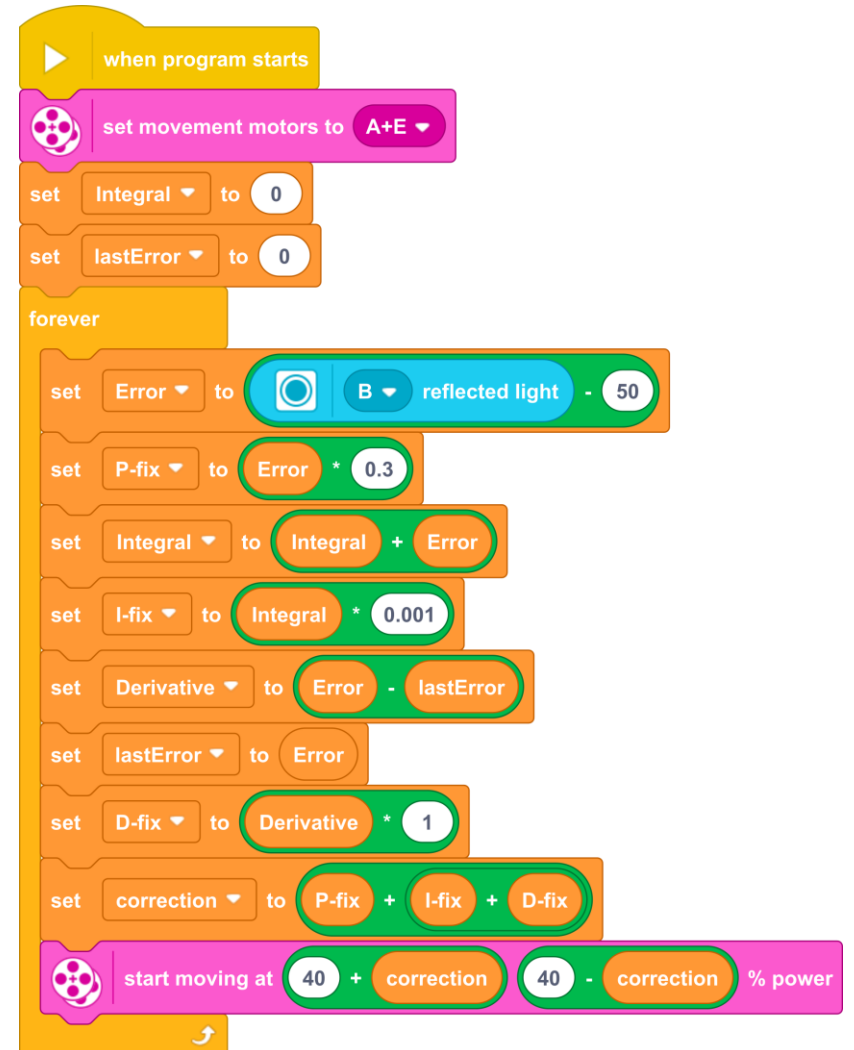
```
forever
  set Error to B reflected light - 50
  set P-fix to Error * 0.3
  set Integral to Integral + Error
  set I-fix to Integral * 0.001
  set Derivative to Error - lastError
  set lastError to Error
  set D-fix to Derivative * 1
  set correction to P-fix + I-fix + D-fix
  start moving at 40 + correction 40 - correction % power
```

The image shows a Scratch script for a PID controller. It is enclosed in a 'forever' loop. The script consists of the following steps:

- Set **Error** to **B reflected light** minus **50**.
- Set **P-fix** to **Error** multiplied by **0.3**.
- Set **Integral** to **Integral** plus **Error**.
- Set **I-fix** to **Integral** multiplied by **0.001**.
- Set **Derivative** to **Error** minus **lastError**.
- Set **lastError** to **Error**.
- Set **D-fix** to **Derivative** multiplied by **1**.
- Set **correction** to **P-fix** plus **I-fix** plus **D-fix**.
- Start moving at **40 + correction** and **40 - correction** % power.

CODE COMPLET

Définissez les variables pour la dernière erreur et l'intégrale avant la boucle et initialisez-les à 0 car elles sont lues avant d'être écrites. En outre, réglez les moteurs de mouvement.



```
when program starts
  set movement motors to A+E
  set Integral to 0
  set lastError to 0
  forever
    set Error to B reflected light - 50
    set P-fix to Error * 0.3
    set Integral to Integral + Error
    set I-fix to Integral * 0.001
    set Derivative to Error - lastError
    set lastError to Error
    set D-fix to Derivative * 1
    set correction to P-fix + I-fix + D-fix
    start moving at 40 + correction 40 - correction % power
```

The code block is a Scratch script for a PID controller. It starts with a 'when program starts' block, followed by three 'set' blocks: 'set movement motors to A+E', 'set Integral to 0', and 'set lastError to 0'. A 'forever' loop contains several 'set' blocks: 'set Error to B reflected light - 50', 'set P-fix to Error * 0.3', 'set Integral to Integral + Error', 'set I-fix to Integral * 0.001', 'set Derivative to Error - lastError', 'set lastError to Error', 'set D-fix to Derivative * 1', and 'set correction to P-fix + I-fix + D-fix'. The script ends with a 'start moving at' block: 'start moving at 40 + correction 40 - correction % power'.

ÉTAPE CLÉ : RÉGLER LES CONSTANTES DU PID

- La façon la plus courante de régler vos constantes PID est l'essai et l'erreur
- Cela peut prendre du temps. Voici quelques conseils :
 - Désactivez tout sauf la partie proportionnelle (mettez les autres constantes à zéro)
 - Ajustez seulement la constante proportionnelle jusqu'à ce que le robot suive bien la ligne
 - Ensuite, activez l'intégrale et ajustez jusqu'à ce qu'elle fournisse de bonnes performances sur une série de lignes
 - Enfin, activez la dérivée et ajustez jusqu'à ce que vous soyez satisfait du suivi de la ligne
 - Lorsque vous activez chaque segment, voici quelques bons chiffres pour commencer pour les constantes :
 - P : 1,0 ajusté par $\pm 0,5$ initialement et $\pm 0,1$ pour le réglage fin
 - I : 0,05 ajusté par $\pm 0,01$ initialement et $\pm 0,005$ pour le réglage fin
 - D : 1,0 ajusté par $\pm 0,5$ initialement et $\pm 0,1$ pour le réglage fin

EVALUER LES SUIVEURS DE LA LIGNE

Proportionnel

- utilise le "P" dans PID
- Effectue des tours proportionnels
- Fonctionne bien sur les lignes droites et courbées.
- Bon pour les équipes intermédiaires à avancées → Doivent connaître les blocs de mathématiques

PID

- C'est mieux que le contrôle proportionnel sur une ligne très courbée, car le robot s'adapte à la courbure
- Cependant, pour la FLL (FIRST LEGO League), qui a surtout des lignes droites, un contrôle proportionnel peut être suffisant

GÉNÉRIQUE

- Cette leçon a été créée par Sanjay Seshan et Arvind Seshan pour « SPIKE Prime Lessons »
- D'autres leçons sont disponibles à l'adresse suivante www.primelessons.org



Ce travail est autorisé dans le cadre d'une [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).