

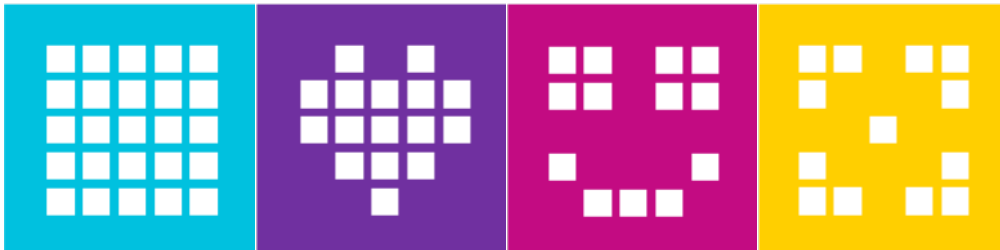
SPIKE PRIME LESSONS

By the Creators of EV3Lessons



SEGUIDOR DE LINHA PID

POR SANJAY E ARVIND SESHAN



OBJETIVOS

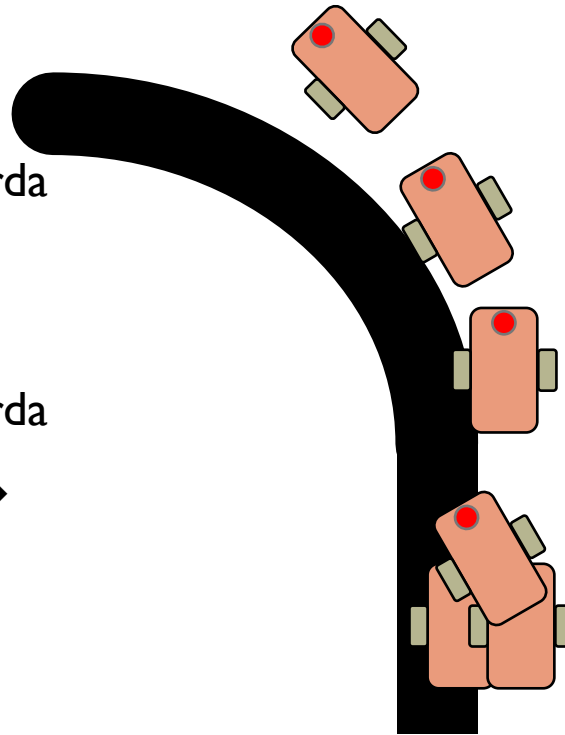
- Aprender as limitações do controle proporcional.
- Aprender o que significa PID
- Aprender a programar e ajustar o PID

QUANDO O SEGUIDOR POROPORCIONAL TEM PROBLEMAS?

Note: the following few slides are animated. Use PowerPoint presentation mode to view them

O que um humano faria?

- Na linha → seguir reto
- No branco → virar a esquerda
- Cruzando a linha → virar a direita
- No branco → virar a esquerda
- Se afastando mais da linha → virar ainda mais



O que o controle proporcional faz?

- Na linha → seguir reto
- No branco → virar a esquerda
- Cruzando a linha → Ir reto!**
- No branco → virar a esquerda
- Se afastando mais da linha → vira a esquerda a mesma quantidade!**

Leitura de luz = 50%

COMO PODEMOS ARRUMAR O CONTROLE PROPORCIONAL?

○ que um humano faria?

Desviando a esquerda/na linha
→ virar a direita.

Se afastando mais da linha →
virar ainda mais

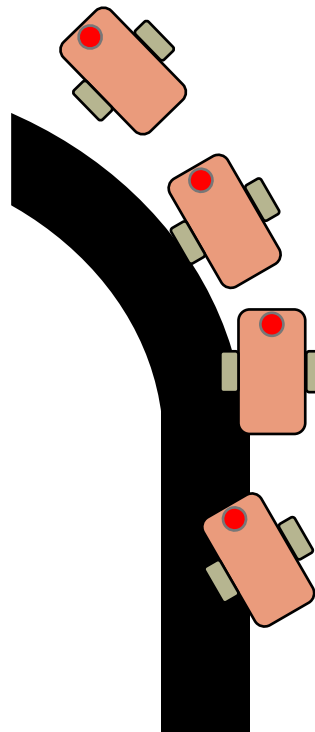
1. Prever qual será a
próxima leitura do sensor

○ que o controle proporcional faz?

**Desviando a esquerda/na
linha → vai reto!**

**Se afastando mais da linha
→ virar a mesma
quantidade**

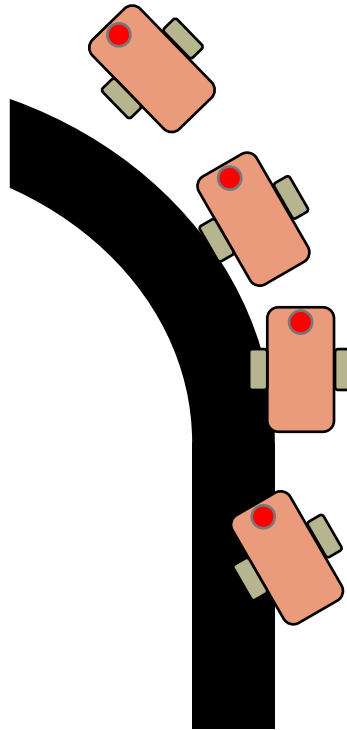
2. Movimentos anteriores
na direção ajudaram a
reduzir o erro?



INTEGRADAS E DERIVADAS

I. Prever qual será a próxima leitura do sensor?

- Se as leituras são: 75, 65, 55
→ qual você acha que será a próxima leitura?
 - E se as leituras fossem 57, 56, 55...
- Que informações você usou para dar seu palpite?
- Derivada → a taxa com que um valor esta variando



2. Movimentos anteriores na direção ajudaram a reduzir o erro?

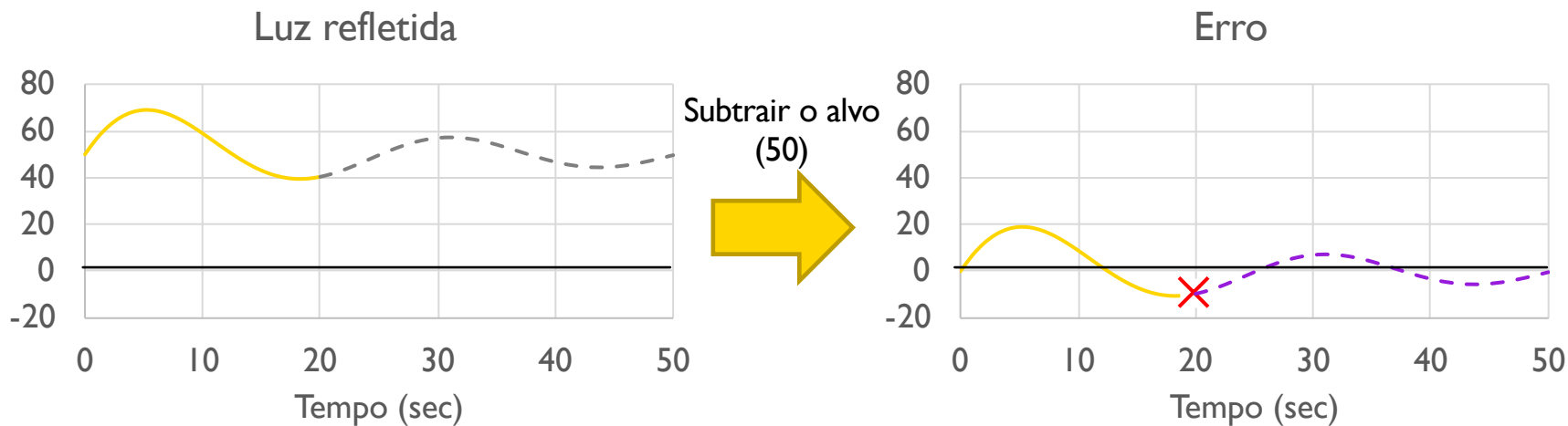
- Quando a correção está funcionando bem, como ficam as leituras de erro?
 - +5, -6, +4 -3... Por exemplo, sempre próximas a zero e com valores negativos E positivos
- Quando a correção não está funcionando bem, como ficam as leituras de erro?
 - +5, +5, +6, +5... , por exemplo, sempre positivas OU negativas.
- Como podemos detectar isso com facilidade?
 - Dica: olhe a soma dos erros anteriores.
- Qual é o valor ideal para a soma? O que representa uma soma onde os números são grandes?
- Integral → a “soma” dos valores

O QUE É PID?

- **P**roportional [Erro] → O quão ruim esta a situação agora?
- **I**ntegral → As minhas correções anteriores ajudaram a melhorar as coisas?
- **D**erivada → Como a situação esta mudando?
- Controle PID → combina os valores do erro, integral e derivada para decidir como dirigir o robô.

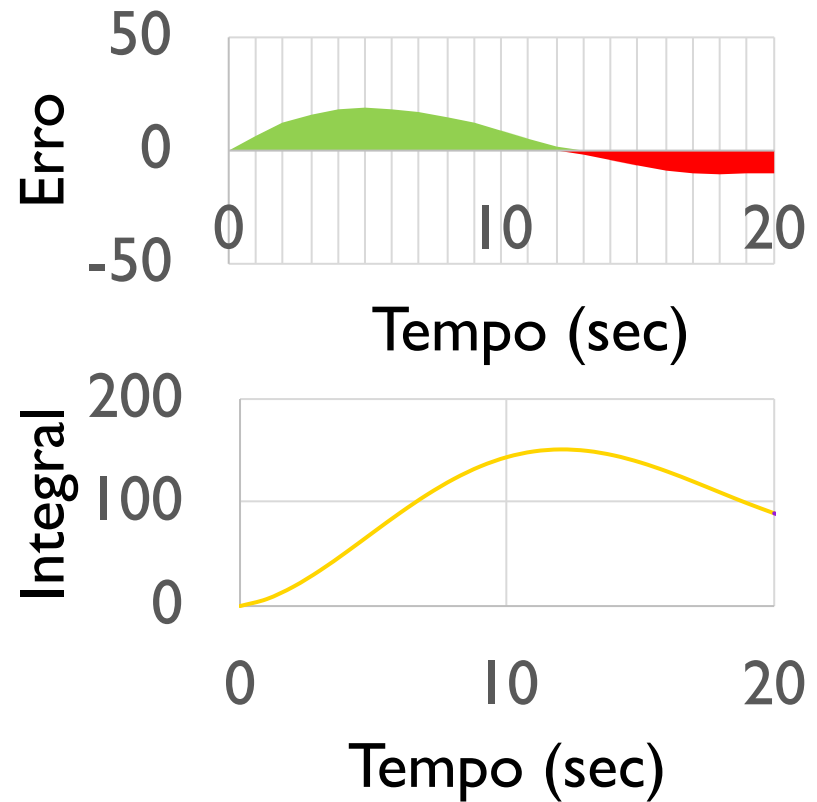
ERRO

- Linhas sólidas representam o que sabemos, pontilhadas o futuro (projeção).
- No tempo 20, você vê Leitura de Luz = 40 e erro = -10. (X vermelho)



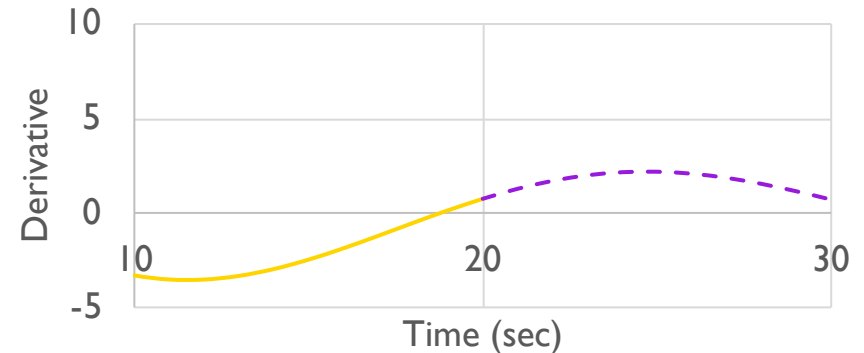
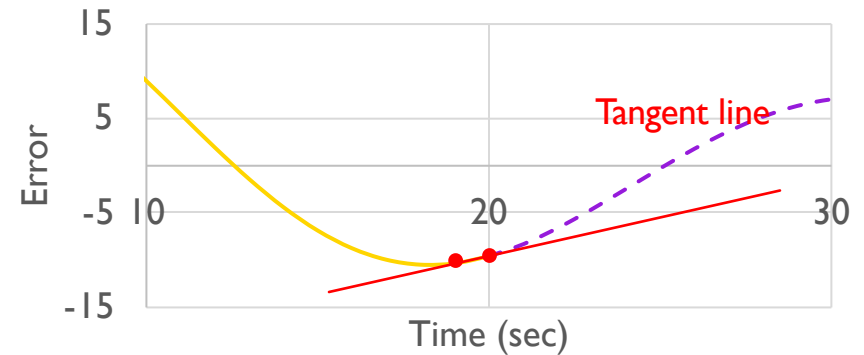
INTEGRAL

- Olha a trajetória anterior do seguidor de linha
- Soma o erro anterior
- Como a área em baixo da curva do gráfico (integral)
 - Verde = área positiva
 - Vermelho = área negativa



DERIVADA

- O quão rápido a posição esta mudando?
 - Preve onde o robô vai estar no futuro imediato
 - Mesma coisa que o quão rápido o erro esta variando.
- Pode ser medido usando a linha tangente a curva das medidas → derivada
 - Aproximada usando dois pontos próximos no gráfico.



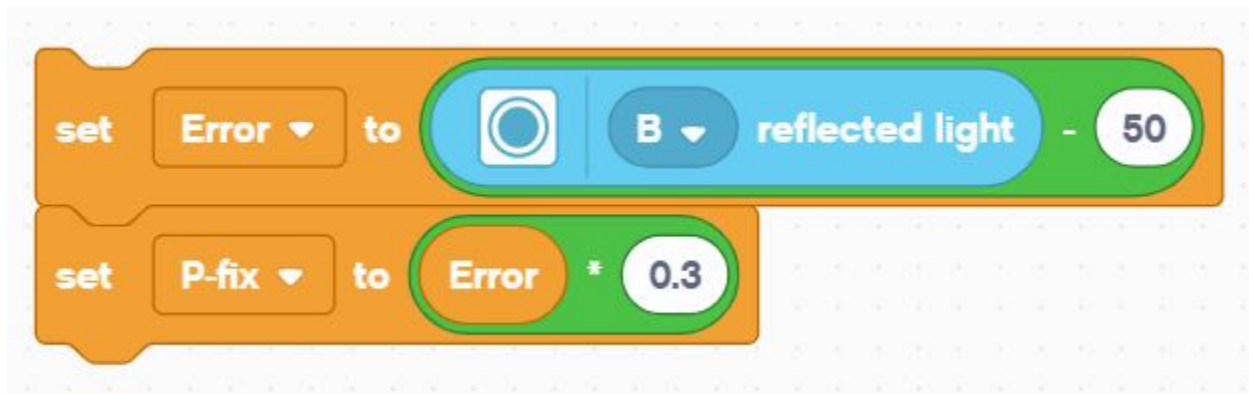
PSEUDOCÓDIGO

1. Obtenha uma nova leitura do sensor
2. Calcule o “erro”
3. Escale o erro para determinar a contribuição para a direção (Controle Proporcional)
4. Use o erro para atualizar a integral (soma dos erros anteriores)
5. Escale a integral para determinar a contribuição para a direção (Controle Integral)
6. Use o erro para atualizar a derivada diferença do erro anterior
7. Escale a derivada para determinar a contribuição para a direção (Controle Derivada)
8. Combine os resultados de P, I, D e guie o robô.

CÓDIGO - PROPORCIONAL

- Esse é o mesmo do controle proporcional.

Erro = distância da linha = leitura - alvo

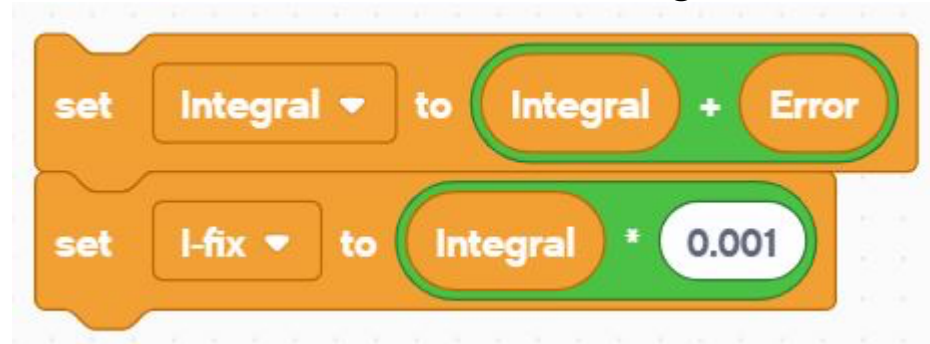


Correção (P_fix) = Erro Integral multiplicado pela constante de proporcionalidade (K_p) = 0.3

CÓDIGO - INTEGRAL

- Essa secção calcula a integral. Ela soma o erro atual a uma variável que contenha a soma dos erros anteriores.
- A constante de escalonamento é geralmente pequena já que o valor da Integrada pode ser grande.

Integral = soma dos erros anteriores = ultima integral + erro mais recente

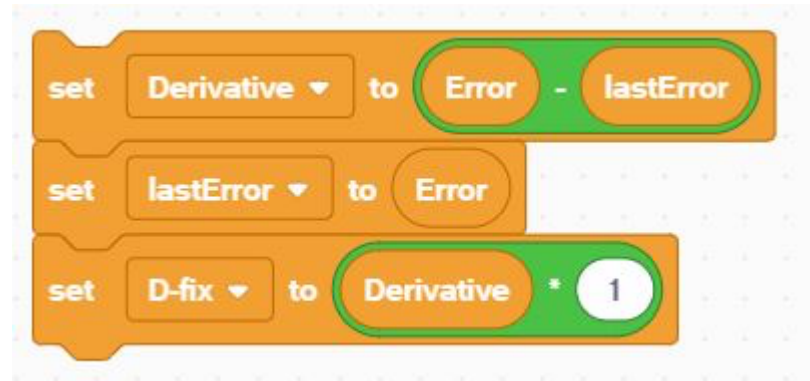


Correção (I_{fix}) = Integral multiplicada pela constante de proporcionalidade (K_i) = 0.001

CÓDIGO - DERIVADA

- Essa secção calcula a derivada. Ela subtrai o erro atual do erro anterior para achar a variação de erro.

Derivada = taxa de mudança do erro = erro atual – erro anterior



Correção (D_fix) = Derivada multiplicada pela constante de proporcionalidade (K_d) = 1.0

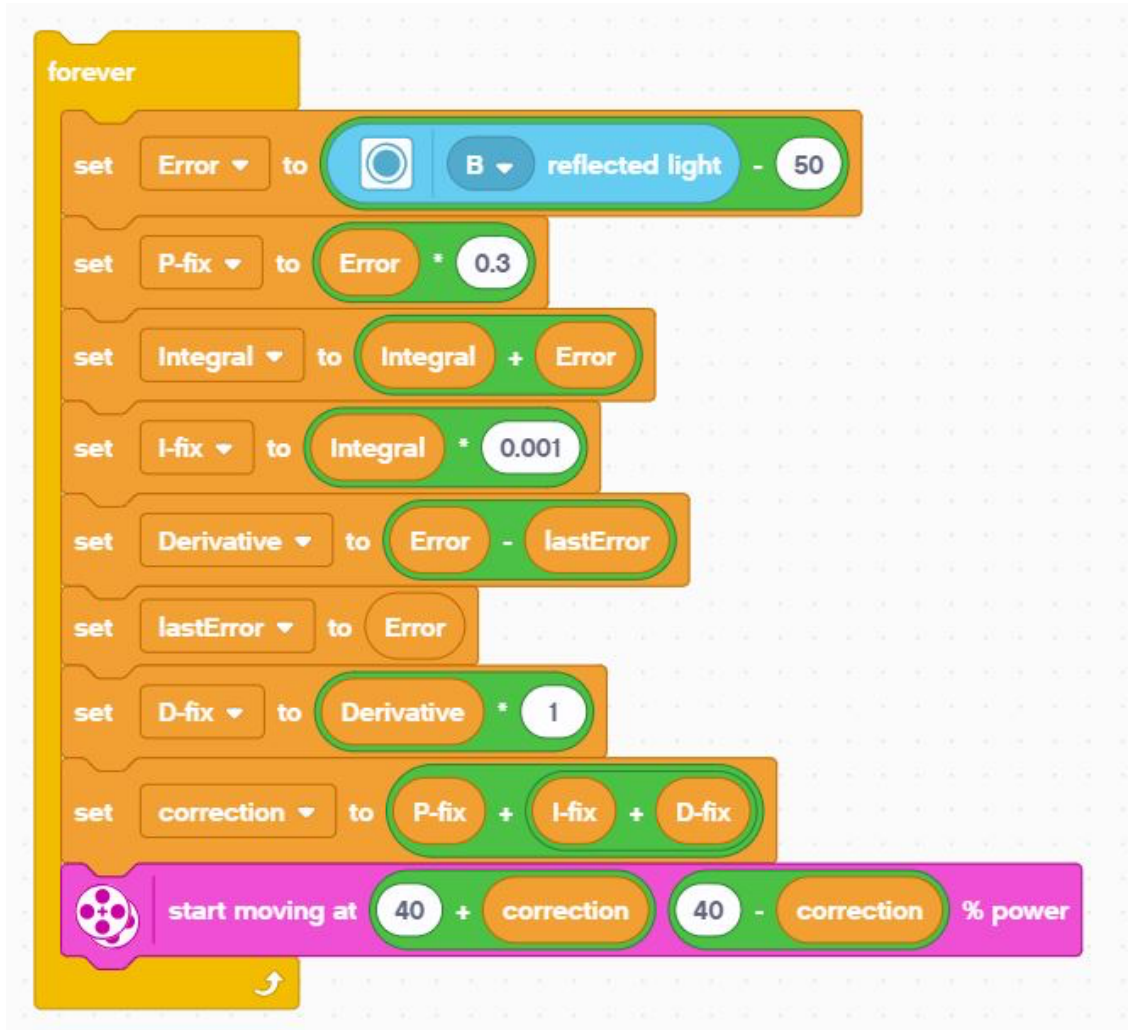
JUNTANDO TUDO

- Com cada componente já escalonado podemos simplesmente soma-los.
- Some os valores de “P-fix”, “I-fix” e “D-fix”. Isso calcula o valor final da correção.
- No SPIKE Prime, usamos % de potência para que os motores tenham maior autonomia de movimentação.



CÓDIGO COMPLETO

- Isso é o que você obtém ao juntar estas partes
- Esperamos que agora você entenda um pouco melhor como funciona o PID



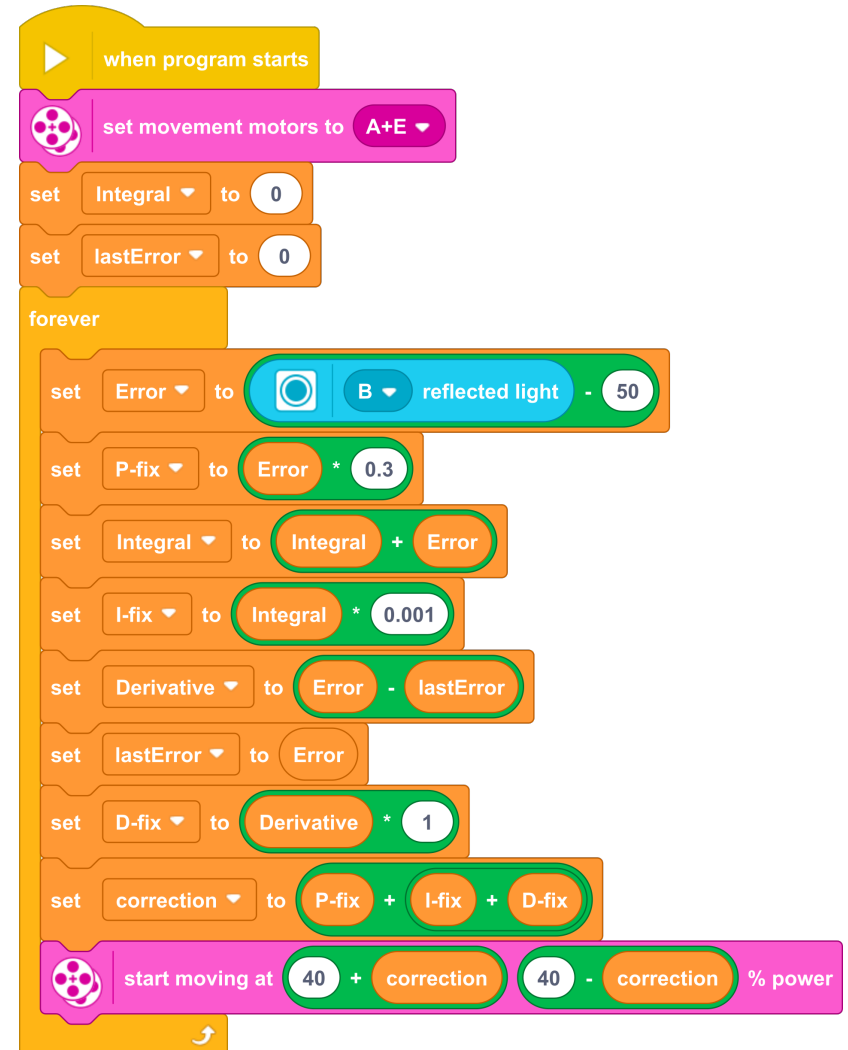
```
forever
  set Error to B reflected light - 50
  set P-fix to Error * 0.3
  set Integral to Integral + Error
  set I-fix to Integral * 0.001
  set Derivative to Error - lastError
  set lastError to Error
  set D-fix to Derivative * 1
  set correction to P-fix + I-fix + D-fix
  start moving at 40 +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

CÓDIGO COMPLETO

Defina as variáveis de “last error” e integral antes de iniciar o loop para 0 porque elas serão lidas antes de terem valores escritos. Também defina os motores de movimento.



```
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 is written in Scratch and implements 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 the following steps: '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', and 'start moving at 40 + correction 40 - correction % power'.

PASSO CHAVE: AJUSTANDO AS CONSTANTES DO PID

- A forma mais comum de ajustar as constantes de PID é através de tentativa e erro.
- Isso pode levar um tempo. Aqui estão algumas dicas:
 - Desabilite tudo, com exceção da parte Proporcional (defina as outras constantes em 0). Ajuste a constante proporcional até o robô seguir a linha razoavelmente bem.
 - Depois habilite a Integral e ajuste-a até ter uma boa performance em uma variedade de linhas.
 - Finalmente, habilite a Derivada e ajuste até estar satisfeito com a performance geral.
 - Quando habilitar cada segmento temos algumas sugestões de valores para começar com as constantes.
 - P: 1.0 ajuste em incrementos de ± 0.5 inicialmente e ± 0.1 para ajustes finos.
 - I: 0.05: ajuste em incrementos de ± 0.01 inicialmente e ± 0.005 para ajustes finos.
 - D: 1.0 ajuste em incrementos de ± 0.5 inicialmente e ± 0.1 para ajustes finos.

AVALIANDO SEGUIDORES DE LINHA

Proporcional

- Usa o “P” de PID
- Faz curvas proporcionais
- Funciona bem em trechos retos e em linhas curvas
- Bom para equipes intermediarias para avançadas → você precisa saber sobre blocos matemáticos.

PID

- É melhor do que controle proporcional em linhas muito curvadas, já que o robô se adapta a curvatura.
- Porém, para FLL, que tem em sua maioria linhas retas, controle proporcional pode já ser o suficiente.

CRÉDITOS

- Essa lição foi criada por Sanjay Seshan e Arvind Seshan para SPIKE Prime Lessons
- Mais lições em www.primelessons.org
- Traduzido para o português por Lucas Colonna



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).