

## ALGORITHM PID DE URMĂRIRE A LINIEI

BY SANJAY AND ARVIND SESHAN

This lesson uses SPIKE 3 software

# OBIECTIVELE LECȚIEI

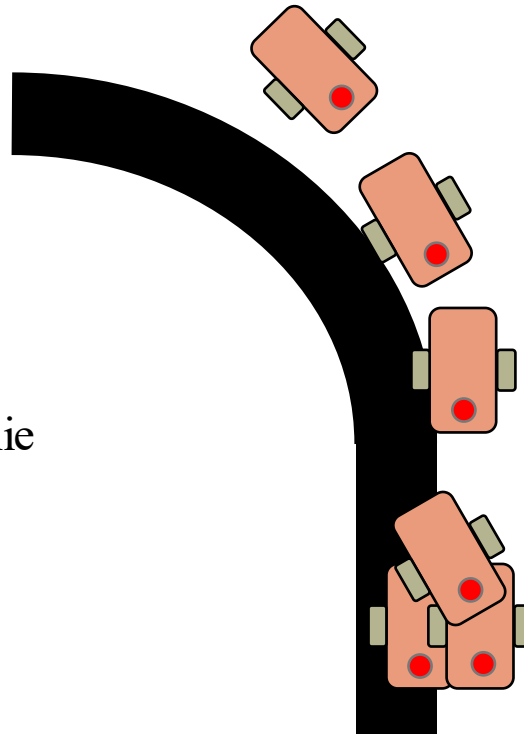
- Învățăm limitările controlului proporțional
- Învățăm ce înseamnă PID
- Învățăm cum să programăm PID și cum să-l ajustăm

# CÂND CONTROLUL PROPORȚIONAL ARE PROBLEME?

Notă: următoarele slide-uri sunt animate. Utilizează prezentarea PowerPoint pentru a le vedea

## Ce face omul?

Pe linie → mergi înainte  
Pe alb → întoarce stânga  
Mergi pe linie → întoarce dreapta  
Pe alb → întoarce stânga  
Ne îndepărtăm mai mult de linie  
→ întoarcem chiar mai mult!



## Ce face controlul proporțional?

Pe linie → mergi înainte  
Pe alb → întoarce stânga  
**Merge înafara liniei → mergi înainte!**  
Pe alb → întoarce stânga  
**Merge mai departe de linie → întoarce stânga cu același număr de grade!**

Citirea reflexivității = ~~100%~~

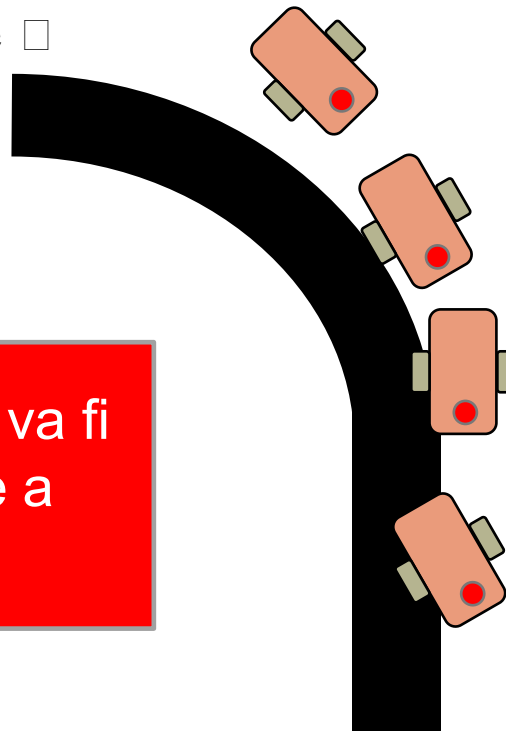
# CUM PUTEM REZOLVA PROBLEMELE CONTROLULUI PROPORȚIONAL?

Ce face omul?

Întoarce stânga/pe linie ☐  
întoarce dreapta

Merge mai departe de linie ☐  
întoarce mai mult!

1. Fă predicții care va fi următoarea citire a senzorului



Ce face controlul proporțional?

Întoarce stânga / pe linie

☐ mergi înainte!

Mergi mai departe de linie

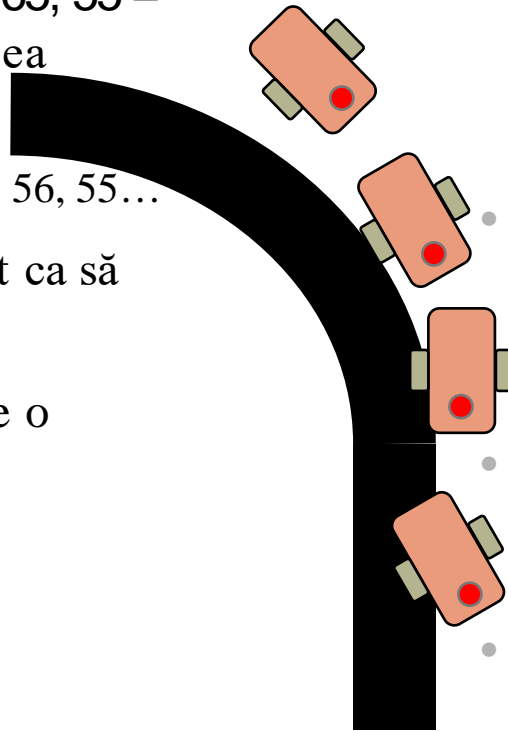
☐ întoarce stânga cu același număr de grade!

2. Virările anterioare ajută să reduci eroarea?

# INTEGRALE ȘI DERIVATE

## 1. Fă o predicție referitoare la ce va citi în continuare senzorul de culoare?

- Dacă citirile sunt: 75, 65, 55 – care crezi că va fi citirea următoare?
  - Dacă citirile sunt 57, 56, 55...
- Ce informații ai folosit ca să ghicești?
- Derivate – rata la care o valoare se schimbă



## 2. Virările anterioare ajută să reduci eroarea?

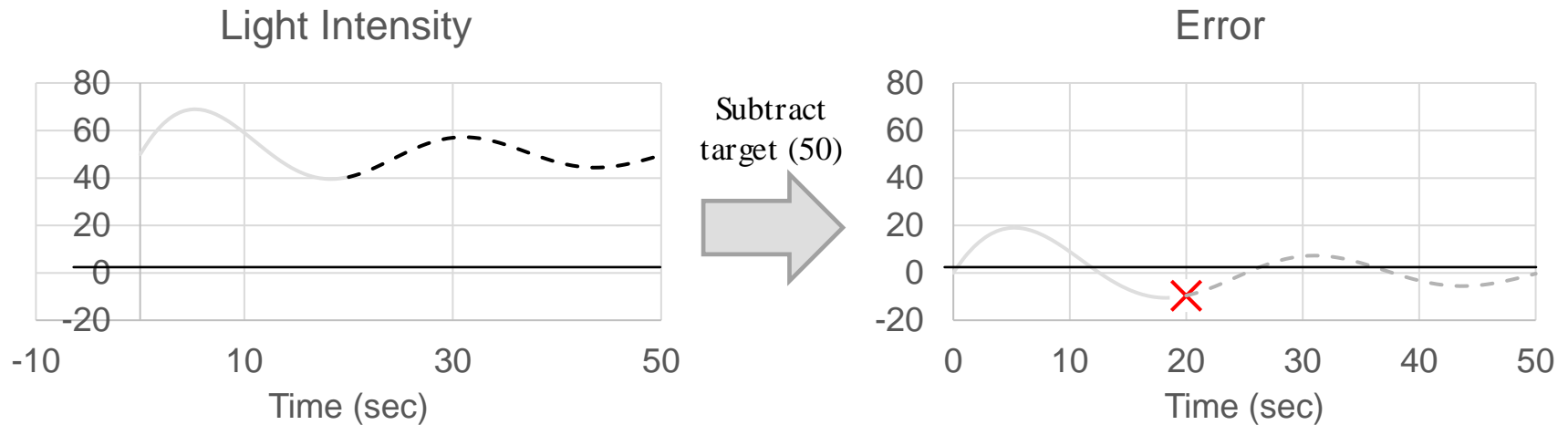
- Dacă corecția merge bine, cum vor arăta citirile?
  - +5, -6, +4 -3.... i.e. oscilează în jurul lui 0
- Când întoarcerea nu va merge, cum va arăta eroarea?
  - +5, +5, +6, +5... i.e. întodeauna pe o parte a lui 0
- Cum putem detecta asta ușor?
  - Sfat: privește suma tuturor erorilor anterioare
- Care va fi valoarea ideală pentru această sumă? Ce înseamnă că suma este mare?
- Integral → “suma” valorilor 5

# Ce este PID?

- Proportional [Eroare] → Cât de rea este situația acum?
- Integral → Ajută reglările anterioare să rezolve lucrurile?
- Derivative → Cum se schimbă situația?
- Control de tip PID → combină valorile erorii, integralei și derivatei pentru a decide cum să vireze robotul.

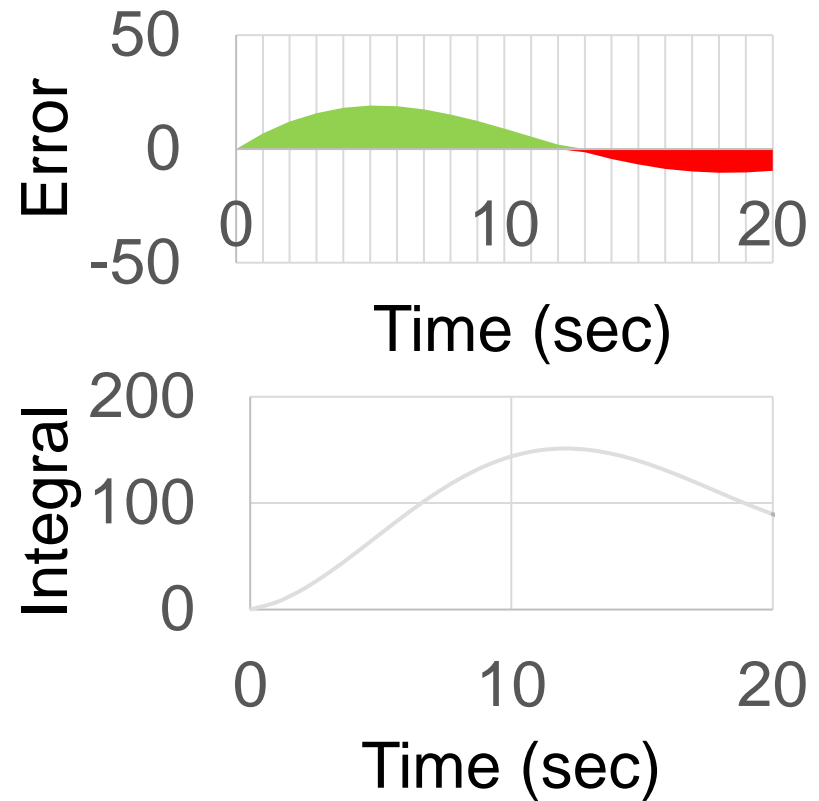
# ERO AREA

- Liniile solide reprezintă ceea ce ai văzut iar liniile punctate reprezintă viitorul.
- La momentul 20, vezi citirea intensității luminii reflectate = 40 iar eroarea = -10 (roșu X)



# INTEGRALA

- Privește istoria recentă a următorului de linie
- Însumează erorile anterioare
- Aceasta este reprezentată de suprafața de sub curba grafului (integrala)
  - Verde = ramura pozitivă
  - Roșu = ramura negativă





# DERIVATA

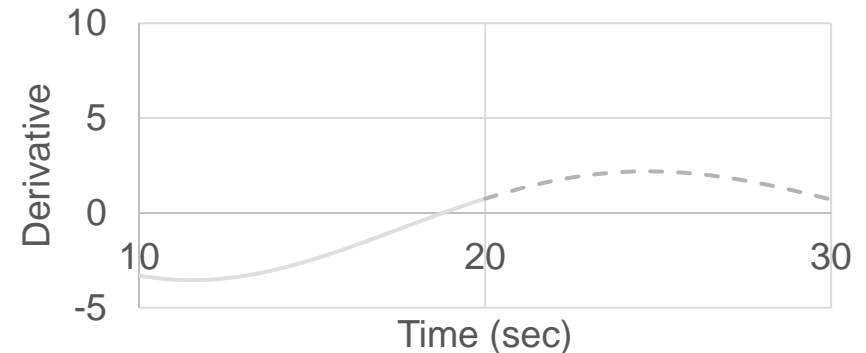
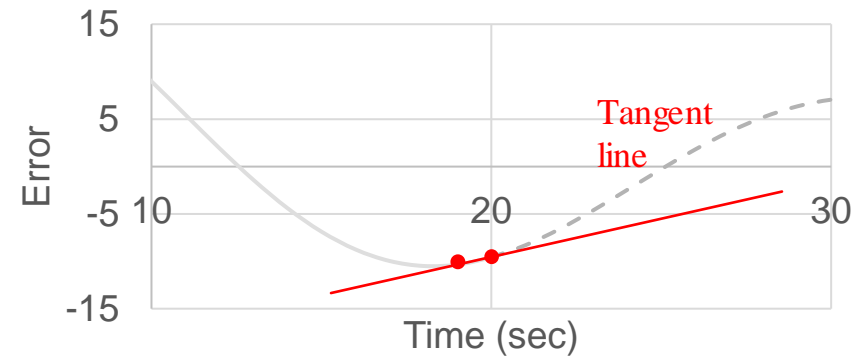
■ Cât de repede se schimbă poziția?

■ Previzionează unde va fi robotul în viitorul imediat

■ La fel cât de repede se schimbă eroarea

■ Poate fi măsurată utilizând linia tangentă la curba măsurărilor → derivata

■ Approximativ utilizând 2 puncte de pe graf



# PSEUDOCOD

1. Ia o nouă citire a senzorului de culaore
2. Calculează “eroarea”
3. Scalează eroarea pentru a determina contribuția la actualizarea corecției (control proportional)
4. Utilizează eroarea pentru a actualiza integrala (suma erorilor anterioare)
5. Scalează integrala pentru a determina contribuția la actualizarea corecției (controlul integralei)
6. Utilizarea erorii pentru a actualiza derivata (diferența de la ultima eroare)
7. Scalează derivata pentru a determina contribuția la actualizarea corecției (controlul derivatei)
8. Combină feedback-ul P, I, și D și corectarea robotului.

# CODE - PROPORTIONAL

Acesta este același cu codul de control proporțional, și urmărește marginea Negru-Alb (pe partea dreaptă a liniei negre)

Eroarea = distanța de la linie = citire - target

```
error = 50 - color_sensor.reflection(port.A)
P_fix = error * 0.5
```

Correction (P\_fix) = Error scaled by proportional constant ( $K_p$ ) = 0.5

# COD - INTEGRALA

- Această secțiune calculează integrala. Adună eroarea curentă într-o variabilă care însumează erorile anterioare.
- Această constantă de scalare este mică din moment ce valoarea Integralei poate fi mare.

Integral = sum of all past errors = last integral + newest error

```
integral = integral + error # or integral+=error  
I_fix = integral * 0.001
```

Correction ( $I_{fix}$ ) = Integral scaled by proportional constant ( $K_i$ ) = 0.001

# COD - DERIVATA

- Această secțiune calculează derivata. Aceasta sustrage eroarea curentă din eroarea anterioară pentru a vedea evoluția erorii.

Derivative = rate of change of error = current error – last error

```
derivative = error - lastError  
lastError = error  
D_fix = derivative * 1
```

Correction (D\_fix) = Derivative scaled by proportional constant ( $K_d$ ) = 1.0

# PUNEM TOTUL ÎMPREUNĂ

- Fiecare componentă este deja scalată. La acest punct, noi putem să le punem simplu împreună.
- Adaugă cele 3 valori pentru P, I, și D. Aceasta va calcula eroarea curentă finală.
- **NOTĂ:** Utilizează corecția la întoarcere după ponderarea de la -100 la 100 deoarece în SP3 nu apare intern., va accepta valori peste limite (e.g. -250) și face lucruri impredictibile.

```
correction = min(100, max(-100, int(P_fix + I_fix + D_fix)))  
motor_pair.move(motor_pair.PAIR_1, correction, velocity = 300)
```

# CODUL COMPLET

```
from hub import port
import motor, motor_pair, color_sensor, runloop, sys

# Constante pentru Drive Base 1
motor_pair.pair(motor_pair.PAIR_1, port.C, port.D)

# Urmărește partea dreaptă a liniei negre (marginea Negru-Alb).
# Pentru a urmări marginea Alb-Negru, schimbă condiția de eroare (reflecția - 50).
async def pid_line_follow_forever():
    integral = 0
    lastError = 0
    while (True):
        # Calculează eroarea.
        error = 50 - color_sensor.reflection(port.A)
        P_fix = error * 0.5
        integral = integral + error
        I_fix = integral * 0.001
        derivative = error - lastError
        lastError = error
        D_fix = derivative * 1
        # verifică valoarea pentru intervalul de la -100 la 100 deoarece SP3 nu o face intern.
        correction = min(100, max(-100, int(P_fix + I_fix + D_fix)))
        # utilizează corecția
        motor_pair.move(motor_pair.PAIR_1, correction, velocity = 300)

async def main():
    await pid_line_follow_forever()

runloop.run(main())
```

# PAȘI CHEIE: AJUSTAREA CONSTANTELOR PID

- Cel mai comun mod de a ajusta constantele PID este „trial and error”.
- Aceasta poate lua timp. Aici sunt câteva sfaturi:
  - Dezactivează totul mai puțin partea proporțională (setează celelalte constante la zero). Ajustează constanta proporțională până când robotul urmărește linia bine.
  - Apoi, activează integrala și ajustează până când robotul furnizează o performanță bună pe multe tipuri diferite de linii.
  - În final, activează derivata și ajustează până când ești mulțumit de următorul de linie.
- Când activezi fiecare segment, aici sunt câteva valori cu care poți începe pentru constante:
  - P: 1.0 ajustează cu  $\pm 0.5$  inițial și  $\pm 0.1$  pentru tuning fin
  - I: 0.05 ajustează cu  $\pm 0.01$  inițial și  $\pm 0.005$  pentru tuning fin
  - D: 1.0 ajustează cu  $\pm 0.5$  inițial și  $\pm 0.1$  pentru tuning fin



# EVALUAREA URMĂRITORILOR DE LINIE

## Proportional

- Utilizează “P” în PID
- Realizează întoarceri proporționale
- Merge bine pe ambele linii drepte și linii curbe
- Bun pentru echipele medii și avansate- trebuie să aibă cunoștințe de block-uri matematice.

## PID

- Este mai bun decât controlul proporțional pe o linie cu curba mai strânsă, pe măsură ce robotul se adaptează la curbura
- Cu toate acestea, pentru FIRST LEGO League, care în majoritate are linii drepte, controlul proporțional poate fi suficient.

# CREDITS

Această lecție a fost creată de Sanjay Seshan și Arvind Seshan for SPIKE Prime Lessons

La această lecție au contribuit membrii comunității FLL Share & Learn.

Mai multe lecții sunt disponibile pe [www.primelessons.org](http://www.primelessons.org)

Această lecție a fost tradusă în limba română de echipa de robotică FTC – ROSOPHIA #21455 RO20



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