

# ADVANCED EV3 PROGRAMMING LESSON



## EV3 Classroom: Urmăritorul de linie de tip PID

---



EV3 CLASSROOM LESSON  
BY EV3LESSONS.COM

# Obiectivele lecției

- Învățăm despre limitările controlului proporțional
- Învățăm ce înseamnă PID
- Învățăm să programăm un PID și cum îl ajustăm

## Cerințe:

Block-uri de Matematică, Calibrarea senzorului de culoare, Data Wires, Variabile, Control Proporțional

Vă recomandăm să aveți cunoștințe minime de algebră. PID este un concept de calcul și elevii ar trebui să înțeleagă de ce este folosit și matematica din spate înainte de utilizarea lui.

Utilizează Modul de Prezentare pentru că lecția conține multe animații.

# Cand are Controlul Proporțional probleme?

## Ce ar face un om normal?

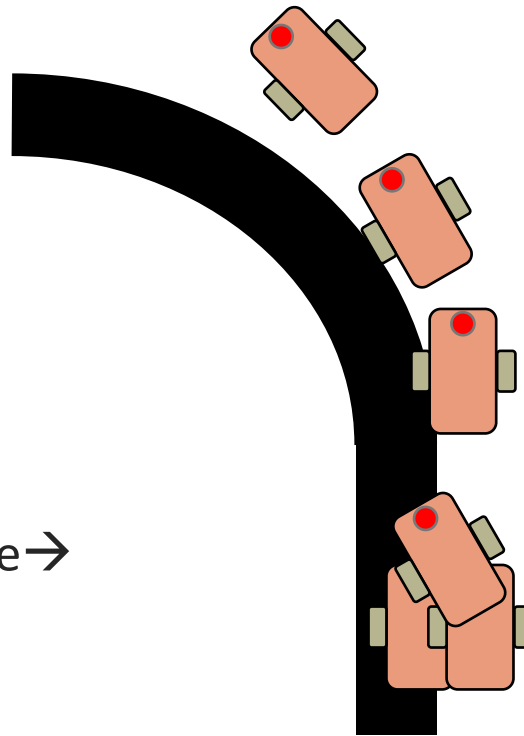
Pe linie → mergi înainte

Pe alb → întoarce stânga

Mergi pe linie → întoarce dreapta

Pe alb → întoarce stânga

Dacă ne îndepărtăm de linie →  
întoarce și mai mult!



## Ce ar face controlul proporțional?

Pe linie → mergi înainte

Pe alb → întoarce stânga

**Mers pe linie → mergi înainte!**

Pe alb → întoarce stânga

**Îndepărtarea de linie →  
întoarce spre stânga în  
aceeași măsură!**

# Cum reparăm Controlul Proportional?

Ce ar face un om  
normal?

Întoarce stânga/pe linie →  
întoarce dreapta

Dacă ne îndepărtăm de  
linie → întoarce și mai  
mult

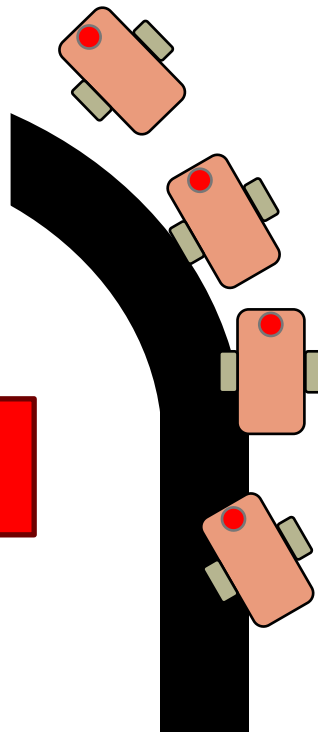
1. Previzionează care va fi  
eroarea viitoare

Ce ar face controlul  
propotional?

Întoarce stânga/pe  
linie → mergi înainte!

Dacă ne îndepărtăm mai  
mult de linie → întoarce  
stânga în aceeași  
măsură!

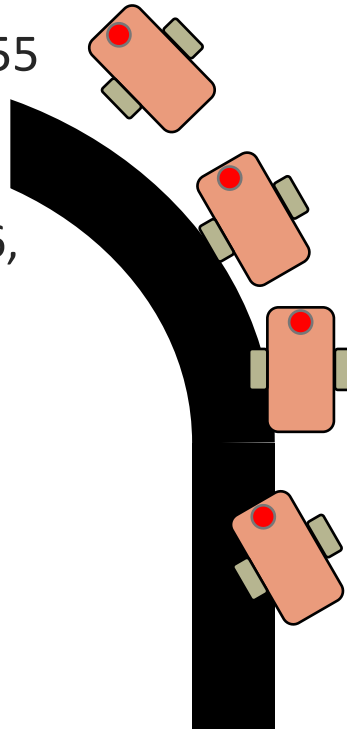
2. Corecțiile anterioare ne-au  
ajutat să reducem eroarea?



# Integrale si Derivate

## 1. Previzionează care va fi eroarea viitoare?

- Dacă citirile sunt: 75, 65, 55  
→ care credeți că va fi următoarea citire?
  - Dacă citirile ar fi 57, 56, 55...
- Ce informație ai utiliza pentru a „ghici”?
- Derivată → rata la care valoarea este schimbată



## 2. Corecțiile anterioare ne-au ajutat să reducem eroarea?

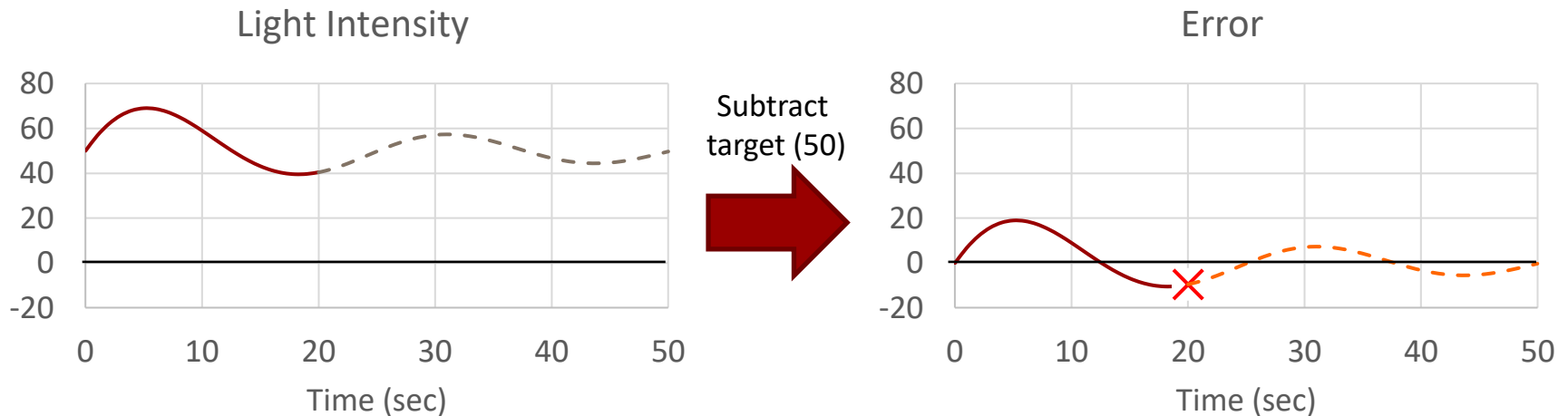
- Când corecția funcționează bine, cum arată citirile erorilor?
  - +5, -6, +4 -3.... i.e. variază în jurul lui 0
- Când corecția nu funcționează, cum arată corecțiile erorilor?
  - +5, +5, +6, +5... i.e. mereu doar pe o parte a 0
- Cum vom detecta asta ușor?
  - Indiciu: uită-te la suma erorilor anterioare
- Care este valoarea ideală a acestei sume? Ce înseamnă că suma este mare?
- Integrală → “suma” acestor valori

# Ce este un PID?

- Proportional [Eroarea] → Cât de rea este situația acum?
- Integral → Corecțiile anterioare au rezolvat problema?
- Derivata → Cum s-a schimbat situația?
- PID control → combină valorile erorii, a integralei și derivatei pentru ca robotul să decidă cum trebuie să întoarcă

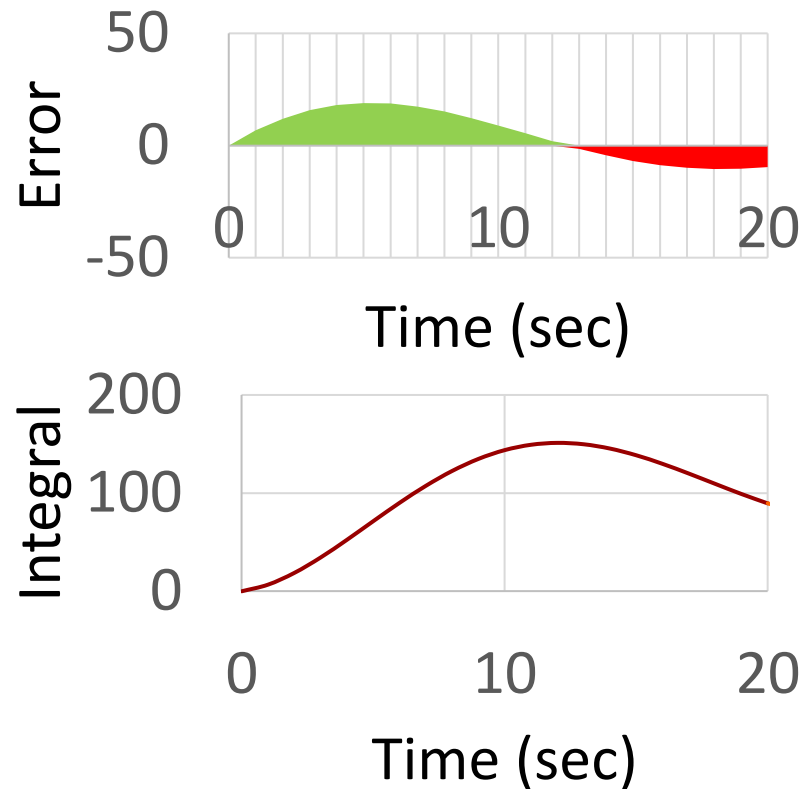
# Eroarea

- Linia continuă reprezintă ceea ce ai văzut ceva, linia punctată reprezintă viitorul
- La momentul 20, citirea senzorului de culoare = 40 și eroarea = -10 (red X)



# Integrala

- Uită-te la istoricul următorului de linie
- Suma erorilor anterioare
- Uită-te la zona de sub curbele din grafic (integrala)
  - Verde = zona pozitivă
  - Roșie = zona negativă





# Derivata

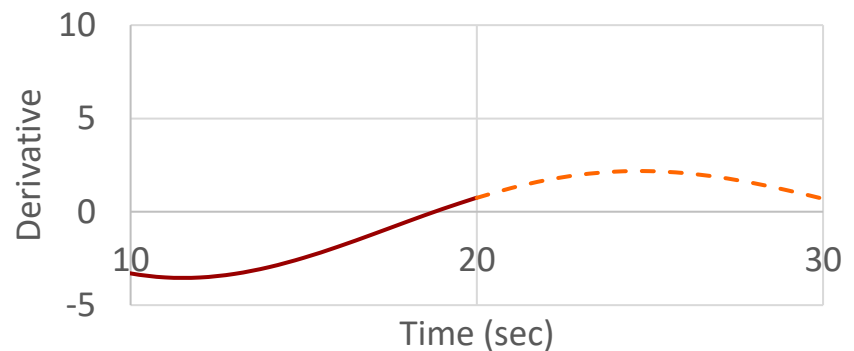
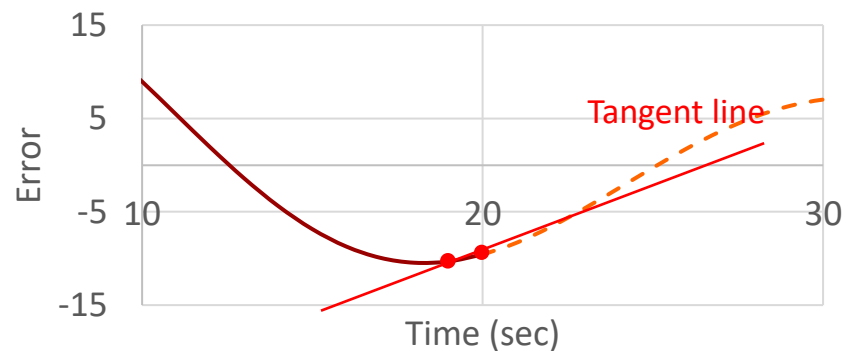
➤ Cât de rapid se schimbă poziția?

➤ Previzionează unde va fi robotul în viitorul imediat

➤ La fel trebuie să se schimbe și eroarea

➤ Poate fi măsurat, utilizând linia tangentă a 2 măsurători → derivata

➤ Utilizează două puncte aproximativ alăturate de pe grafic



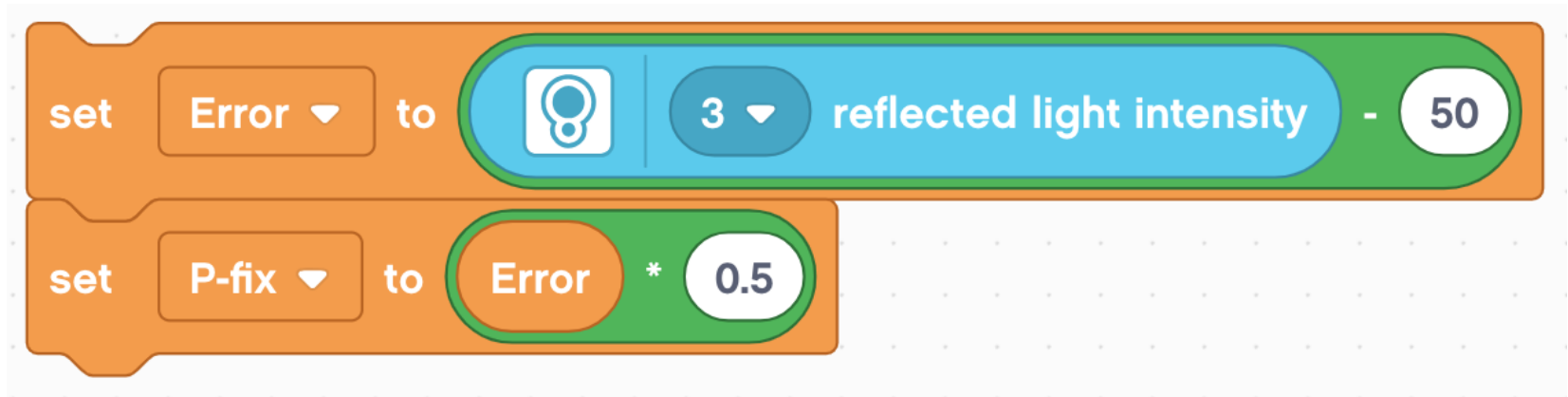
# Pseudocode

1. Ia o nouă citire a senzorului de culoare
2. Calculează “eroarea”
3. Scalează eroarea pentru a determina contribuția actualizării întoarcerii (proportional control)
4. Utilizează eroarea pentru a actualiza integrala (suma tuturor erorilor)
5. Scalează integrala pentru a determina contribuția la actualizarea întoarcerii (control pe baza integralei)
6. Utilizează eroarea pentru a actualiza derivata (diferența de la ultima eroare)
7. Scalează derivata pentru a determina contribuția la actualizarea întoarcerii (controlul derivatei)
8. Combină feedback-ul P, I, și D și întoarce robotul.

# Cod – Proportional

➤ Este la fel ca programul de control proporțional

Eroarea = distanța față de linie = citire - target

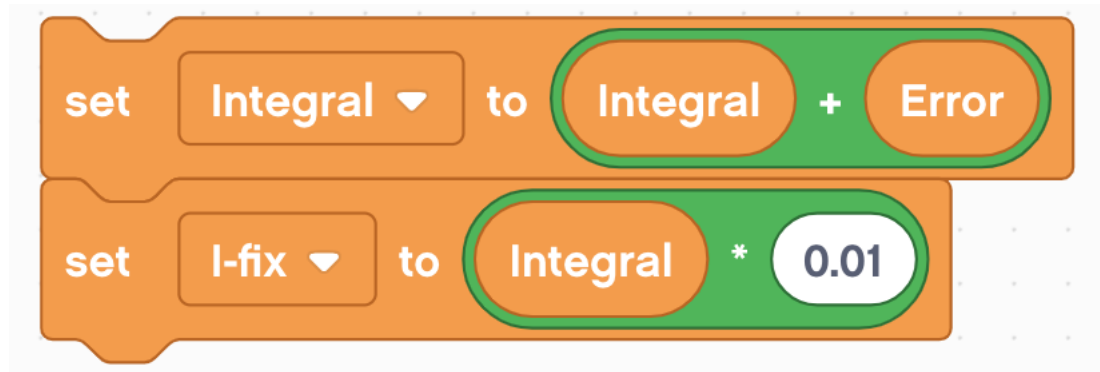


Corecția (P\_fix) = Eroarea scălată cu o constantă proporțională ( $K_p$ ) = 0.5

# Cod - Integrala

- Această secțiune calculează integrala. Adaugă eroarea curentă la o variabilă care este suma tuturor erorilor anterioare.
- Constanta de scalare este mică din moment ce integrala este mare.

Integrala = suma tuturor erorilor anterioare = erorile anterioare + eroarea cea mai nouă

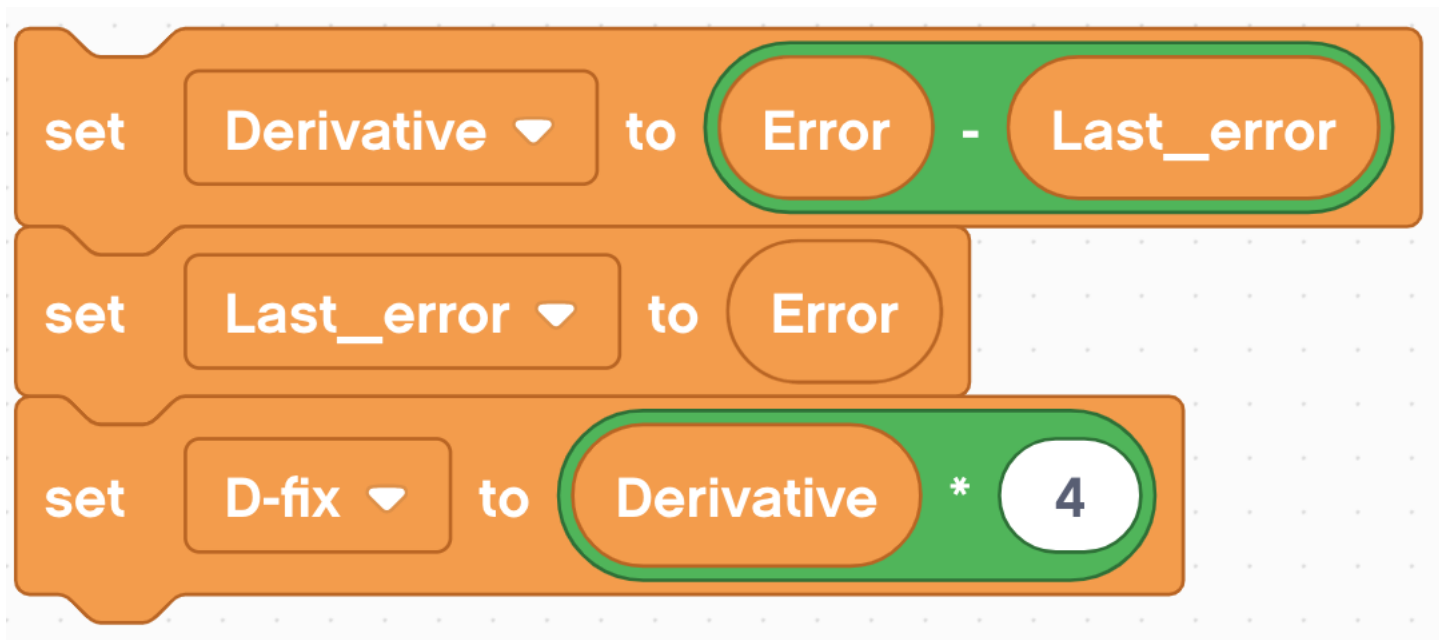


Corecție (I\_fix) = Integrala scalată cu o constantă proporțională ( $K_i$ ) = 0.01

# Cod - Derivata

- Această secțiune de cod calculează derivata. Scade eroarea curentă din ultima eroare pentru a vedea schimbarea în eroare.

Derivata = rata de schimbare a erorii = eroarea curentă – ultima eroare

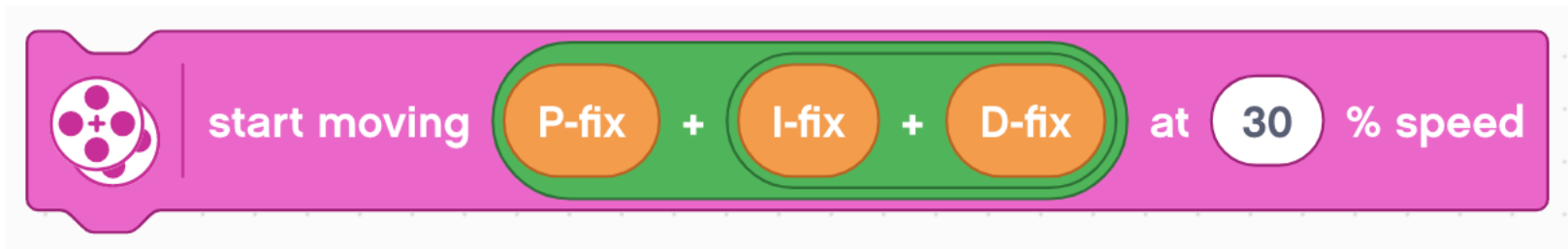


Corecție (D\_fix) = Derivata scalată cu o constantă proporțională ( $K_d$ ) = 4.0

# Cum le punem împreună

➤ Fiecare dintre aceste componente au fost scalate. La acest punct, le putem pune doar împreună.

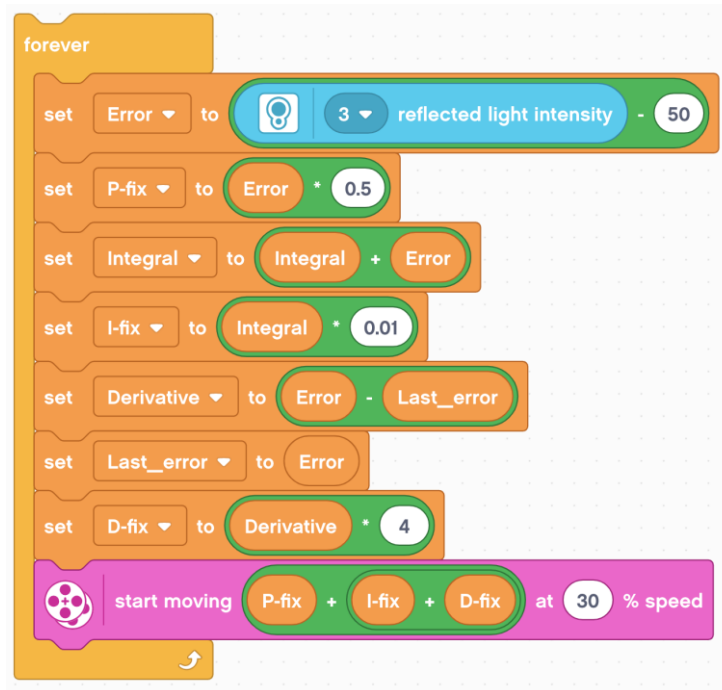
Aplică corecția la block-ul mișcare



Adaugă cele 3 corecții pentru P, I, și D împreună. Aceasta va fi corecția finală.

# Codul complet

- Asta e ce iese dacă le punem pe toate cele trei împreună.
- Sperăm că înțelegeți cum funcționează PID-ul un pic mai bine.



Proportional

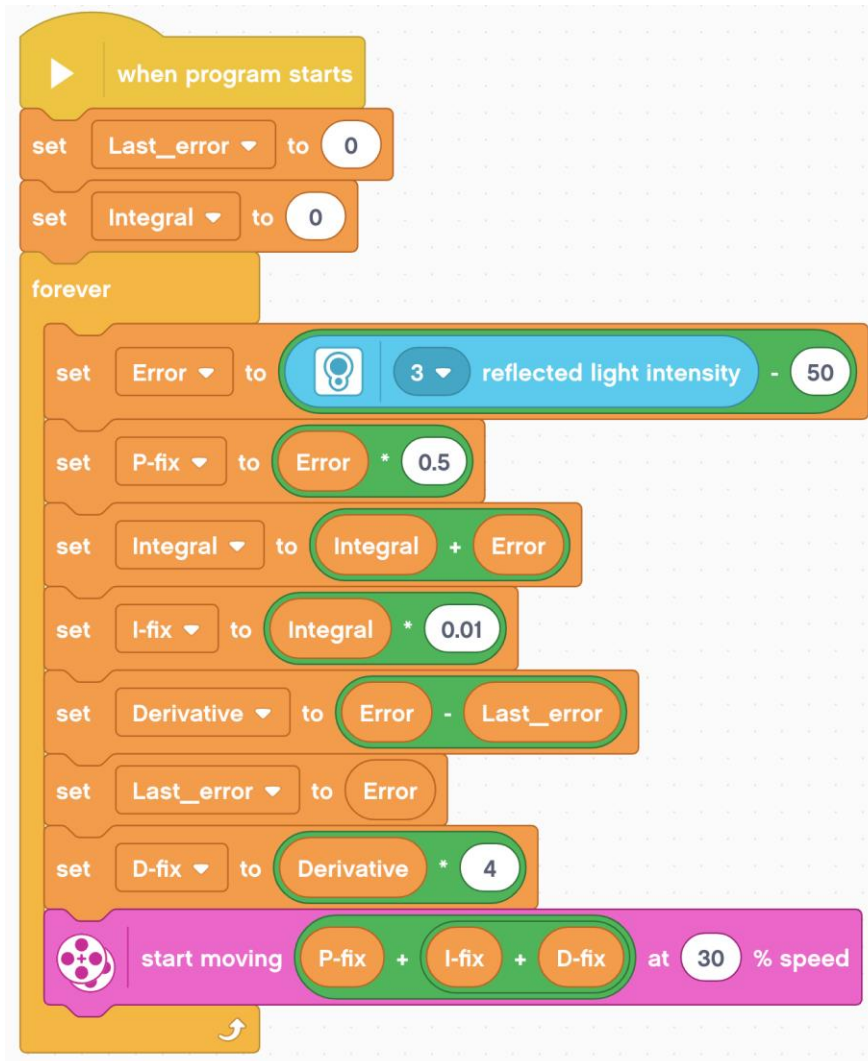
Integrala

Derivata

Puse împreună

# Codul complet

Setează variabilele pentru ultima eroare și integrala înainte de LOOP și inițializează-le la 0 pentru că ele sunt mai întâi citite și apoi scrise.



Proportional

Integrala

Derivata

Puse împreună



# Pas cheie: Ajustarea constantelor PID

- Cel mai comun mod de ajusta constantele PID este încercare și eroare.
- Asta poate cere timp. Aici sunt câteva sfaturi:
  - Dezactivați totul mai puțin partea proporțională (setează celelalte constante la zero). Ajustează constanta proporțională până când robotul urmărește linia bine.
  - Apoi, activează partea integrală și ajustează constanta până când robotul dovedește o bună performanță pe linie.
  - În sfârșit, activează partea derivată și ajustează constanta până când ești satisfăcut cu urmărirea liniei.
  - Când activezi fiecare segment, aici sunt câteva valori cu care poți să începi ajustarea constantelor:
    - P: 1.0 adjust by  $\pm 0.5$  initially and  $\pm 0.1$  for fine tuning
    - I: 0.05 adjust by  $\pm 0.01$  initial and  $\pm 0.005$  for fine tuning
    - D: 1.0 adjust by  $\pm 0.5$  initially and  $\pm 0.1$  for fine tuning

# Evaluarea Urmăritorilor de linie

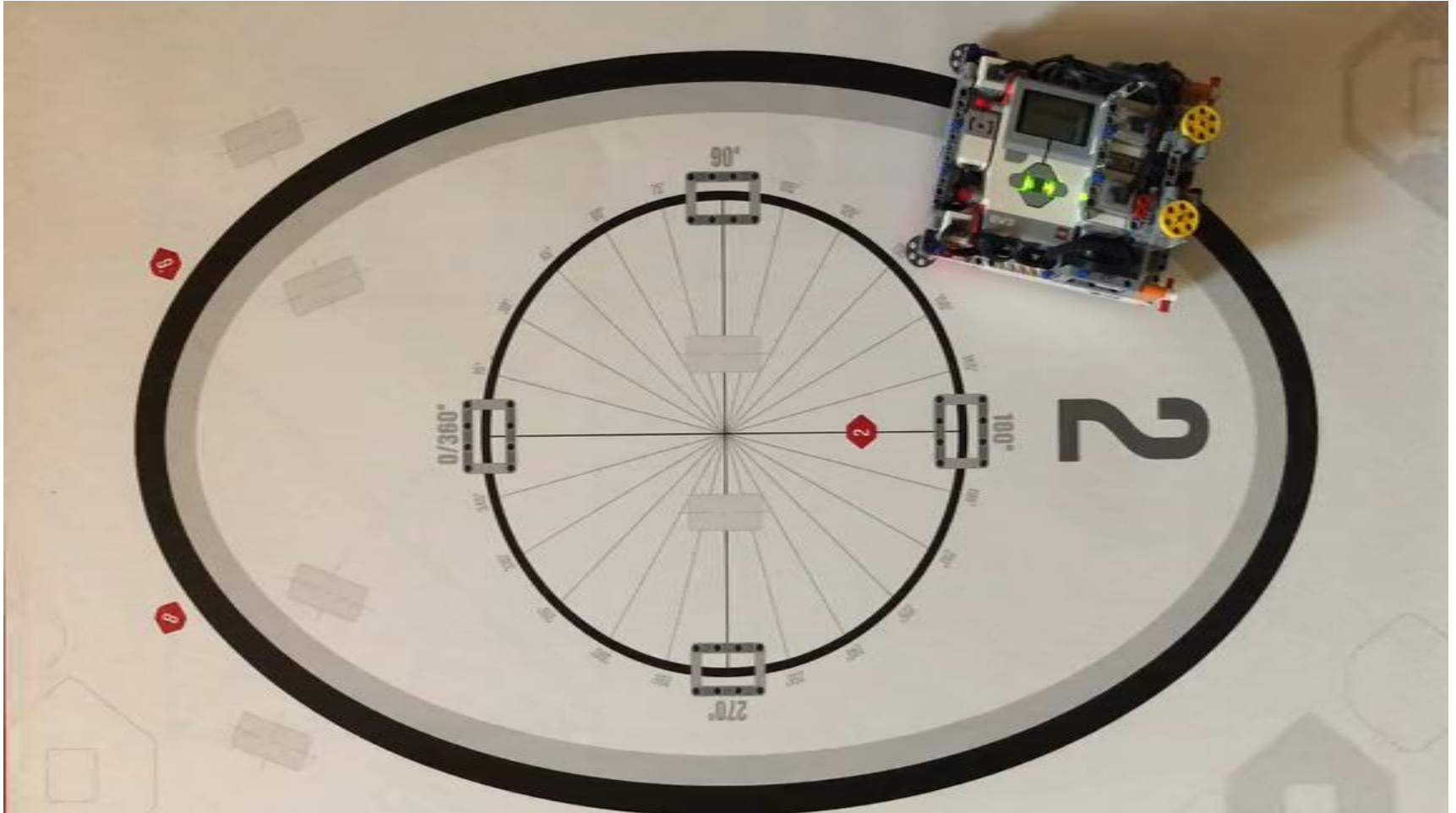
## Proportional

- Este de fapt “P” din PID
- Realizează întoarceri proporționale
- Merge bine atât pentru liniile drepte cât și pentru liniile curbe
- Bun pentru echipele cu experiență și avansate → trebuie să știți să folosiți block-uri de matematică și liniile de date.

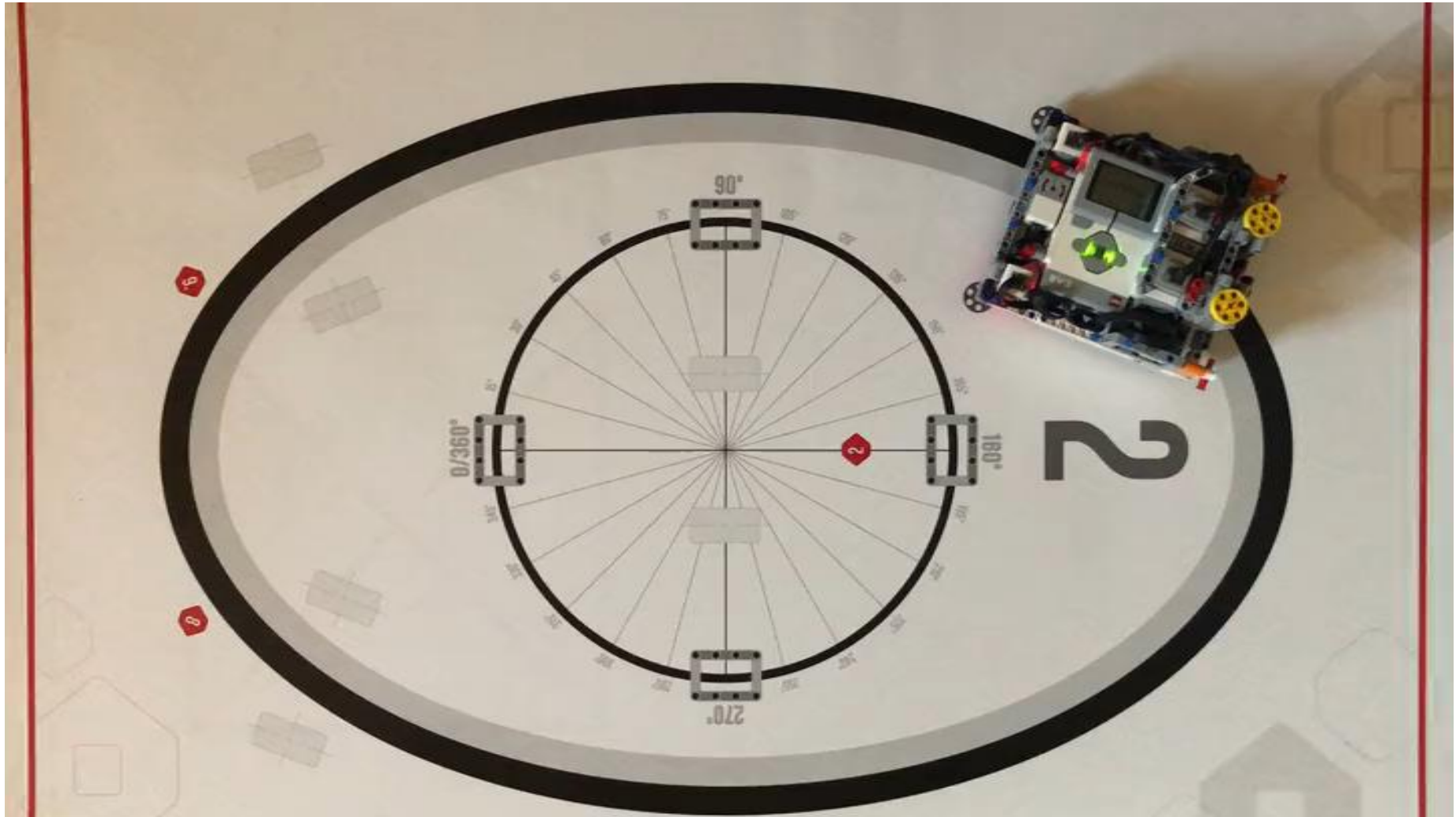
## PID

- Este mai bun ca și controlul proporțional pentru curbele strânse, pe măsură ce robotul se adaptează curburii.
- Cu toate acestea, pentru FIRST LEGO League, care are în mare măsură linii drepte, controlul proporțional este suficient.

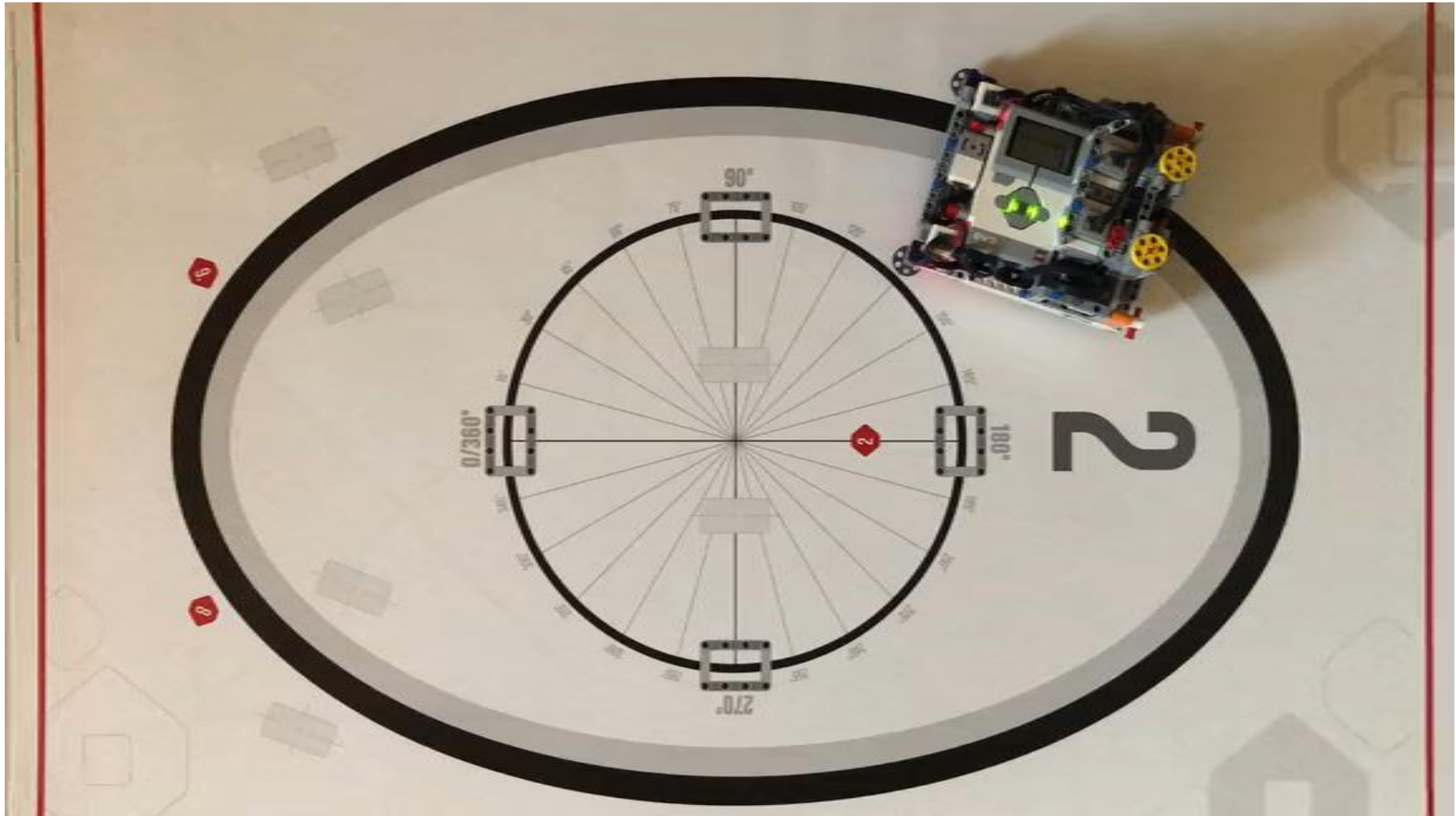
# Proportional Control (0.6 Constant)



# Proportional Control (0.8 Constant)



# PID Control



# CREDITS

- Această lecție de Mindstorms a fost realizată de Sanjay Seshan și Arvind Seshan.
- Mai multe lecții sunt disponibile pe [ev3lessons.com](http://ev3lessons.com)
- 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/).