



Control Award, Sponsored by Arm, Inc. Submission Form

Team # 19109

Team Name: RaSky

Obiective Autonomie (Scor maxim de 50 de puncte):

1. Punctarea a 6 conuri pe stâlpul înalt (1 preload + 5 din stack, 30 de puncte)
2. Detectarea cazului de autonomie și parcare în zona corespunzătoare (20 de puncte)

Senzori folosiți (6 în total):

1. Encoder REV Through Bore (x3): Folosirea a 3 roți de odometrie permit robotului măsurarea distanței parcurse în teren. În plus, roțile sunt independente de roțile folosite la mișcarea robotului. Acest lucru reduce foarte mult alunecările roților și le permite să își păstreze măsurătorile corecte chiar și la perturbări.
2. Encoder de motor (x1): Un encoder de motor este folosit pentru liftul robotului. Pentru a putea dicta liftului poziția la care trebuie să ajungă, acesta trebuie să știe în orice moment la ce poziție se află.
3. Giroscop (x1): Giroscopul încorporat în Control Hub este folosit atât în autonomie pentru a detecta unghiul robotului față de poziția de start, cât și în perioada de driving pentru a detecta și corecta înclinarea robotului.
4. Cameră Web (x1): Această cameră este folosită pentru detectarea cazului în perioada de autonomie.

Algoritmi principali:

1. Localizarea robotului în plan cartezian: Cu ajutorul roților de odometrie menționate mai sus putem măsura poziția robotului în plan cartezian (x , y , θ - rotație) față de poziția inițială a robotului. Cu ajutorul acestor coordonate putem să facem robotul să urmărească traiectorii complicate până la orice poziții din teren. În plus, robotul își poate să continue autonomia în mod corect și după un impact cu alt robot sau cu un obstacol.
2. Sistem de control PID: Pentru poziționarea liftului la înălțimea potrivită dar și pentru menținerea acesteia ne folosim de un algoritm de reglare PID scris de către noi. Acest algoritm are trei componente, termenul proporțional care este și cel principal, la care se adaugă termenul derivativ și cel integral, permițându-ne să controlăm liftul cu ușurință și cu precizie.
3. Sistem de control Feedforward: Cu ajutorul algoritmilor de tip feedforward putem combate forțele pe care nu le dorim să interacționeze cu sistemele de pe robot (ex: gravitația sau forța de frecare). Folosim acest algoritm pentru a anula forța de gravitație, motoarele în repaus nefiind îndeajuns de puternice pentru a menține poziția liftului.
4. Profilarea mișcării: Pentru a realiza mișcări mai precise realizăm un profil de mișcare care accelerează treptat și menține viteza dorită până aproape de final când începe să decelereze treptat. Acest lucru este folosit spre exemplu în traiectorii pentru a reduce alunecările roților.

Îmbunătățiri aduse perioadei de driving:

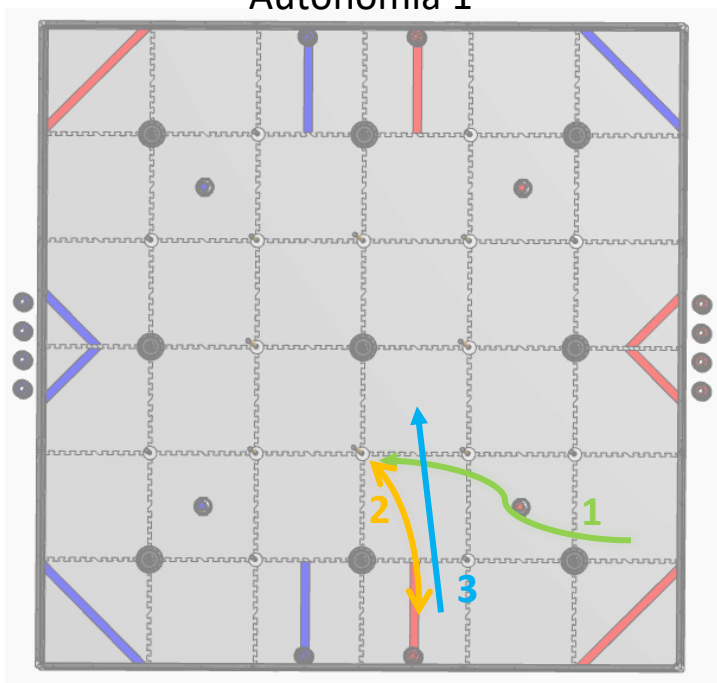
1. Mod de condus anti înclinare: Având un robot mic și rapid a fost nevoie să reducem riscul înclinării robotului sau chiar al prăbușirii acestuia. Am realizat acest lucru printr-un sistem anti înclinare care, cu ajutorul controlului PID și a giroscopului poate să își corecteze unghiul în timp real.
2. Mod de condus relativ terenului: Conducerea robotului după schimbarea orientării acestuia poate să inducă confuzie. Folosind giroscopul putem să ne folosim de unghiul de start al robotului pentru a urmări mișcarea joystick-ului nu după orientarea robotului, ci după planul cartezian inițial. Acest lucru este realizat prin rotirea vectorului de mișcare bazat pe diferența dintre unghiului robotului curent și unghiul inițial.
3. Control al vitezei inteligent: Pentru controlul vitezei nu am vrut să complicăm sarcina driverilor așa că ne-am gândit la un sistem prin care viteza robotului este micșorată doar atunci când trebuie. Acest lucru este realizat cu ajutorul poziției

liftului, care ne poate spune când robotul urmează să punteze. Pentru punteare este necesară o viteză mai mică în funcție de înălțimea stâlpului, iar această viteză este aplicată automat.

4. Stări predefinite ale liftului: Pentru a spori performanța driverilor am predefinit 5 stări ale liftului (start, ground, low, medium, high). Aceste stări sunt implementate într-un sistem de stări finite, starea actuală fiind dictată de driver.
5. Vibratul controllerului bazat pe timp: Controllerul driverilor este programat pentru a anunța diferite perioade de joc printr-un sistem de vibrație. Spre exemplu, cu 5 secunde înainte de perioada de endgame, controllerul va face câteva vibrații scurte pentru a atenționa driverii.
6. Relievarea mecanismelor forțate: Cu ajutorul senzorului de voltaj ce se află înăuntrul Control Hub-ului putem detecta scăderi de tensiune puternice ce înseamnă adesea forțarea unui mecanism de pe robot. În acest caz mecanismele sunt inversate temporar pentru a rezolva cauza forțării (ex: liftul este blocat).
7. Moduri multiple pe un singur buton: Sistemele de pe robot cu stări multiple sunt acționate de un singur buton pentru a reduce complexitatea. Spre exemplu cleștele are 2 moduri: închis și deschis, acționate de un singur buton.

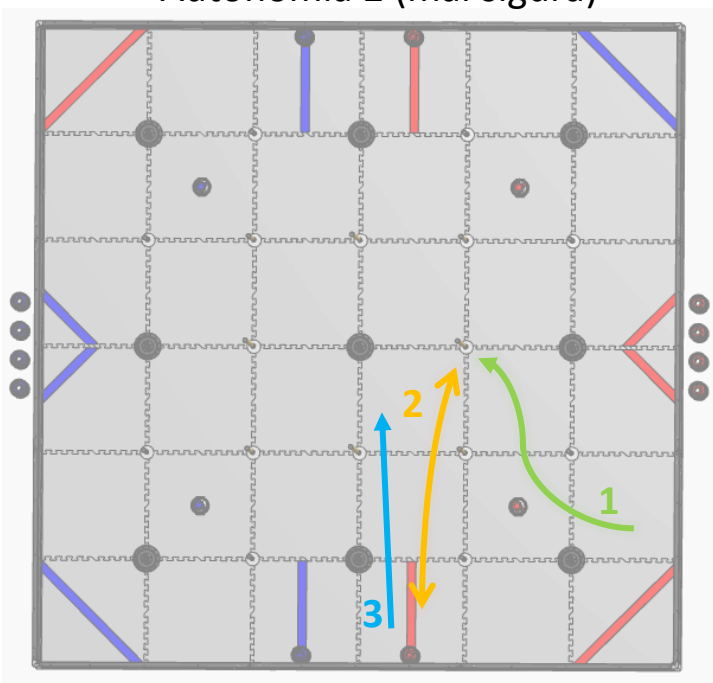
Diagrame cu autonomia:

Autonomia 1



1. Robotul execută o traiectorie șerpuită până în dreptul stâlpului înalt de pe partea corespunzătoare și puntează conul.
2. Robotul colectează și puntează alte 5 conuri din stack pe același stâlp, această traiectorie fiind executată de 5 ori.
3. Robotul se parchează în zona corespunzătoare cazului de început.

Autonomia 2 (mai sigură)



1. Robotul execută o traiectorie șerpuită până în dreptul stâlpului înalt din partea de mijloc a terenului și puntează conul.
2. Robotul colectează și puntează alte 5 conuri din stack pe același stâlp, această traiectorie fiind executată de 5 ori.
3. Robotul se parchează în zona corespunzătoare cazului de început.