

Sistem inteligent de locomoție pentru un robot hexapod

Rezumat

Adrian Bulat [bulat.adrian@gmail.com] Universitatea Tehnică „Gheorghe Asachi” din Iași

Supervizor: Ș.l.Dr.Ing. Andrei Stan [andreis@tuiasi.ro] Universitatea Tehnică „Gh. Asachi” din Iași

Abstract: Deși deplasarea cu ajutorul roților este simplă și permite acoperirea unor distanțe mari cu un cost redus, există situații în care prezența obstacolelor sau a altor impedimente fac necesară implementarea unui robot care să se deplaseze cu ajutorul picioarelor. Cea mai frecventă abordare este realizată cu ajutorul unui robot hexapod. Deși există studii care tratează deplasarea unui hexapod, puține dintre ele analizează locomoția în prezența erorilor mecanice severe sau a erorilor hardware de poziționare a motoarelor. În această lucrare se prezintă un algoritm inteligent de deplasare efectuându-se o analiză comparativă între diferiți algoritmi de locomoție.

Deplasarea eficientă a unui robot hexapod este o problemă dificilă în sine. Majoritatea algoritmilor existenți se bazează pe imitarea mișcărilor de locomoție a insectelor, mișcări ce s-au demonstrat a fi adaptabile în cele mai variate situații. În acest proiect se implementează și testează un robot hexapod ce se deplasează în prezența unor erori mecanice ridicate, necalculate în prealabil, efectuându-se o analiză comparativă între diferite moduri de deplasare. Robotul are 18 grade de libertate implementate cu ajutorul a 18 motoare servo, MG996R.

Motoarele sunt controlate cu ajutorul unui driver cu 32 de output-uri comandat serial pe UART de către o placa de dezvoltare bazată pe un ARM Cortex M4, STM32F407 Discovery. Comunicarea între platformă și dispozitivul extern de control – un PC cu o aplicație Desktop scrisă în C# va fi realizată wireless prin intermediul unui modul *lowcost (eng.)* și *lowpower (eng.)*, wi-fi către serial, ESP8266. Protocolul de comunicație va fi TCP. Avantajul oferit de acest model de comunicare este data de simplitatea implementării și ușurința conectării multiplelor dispozitive la el, fiind posibilă controlarea robotului prin intermediul internetului. Alimentarea este realizată cu ajutorul unei baterii LiPo cu factor ridicat de descărcare, de 7.4V. Este esențial ca sursa de alimentare să suporte un curent de intensitate ridicată deoarece curentul maxim teoretic atinge valori de peste 30A. În urma testelor efectuate anterior, în timpul deplasării s-a înregistrat un curent electric cu o intensitatea de 10-15A.

Programarea microcontrolerului este realizată în C ANSI cu ajutorul unui IDE freeware, CooCox IDE, bazat pe eclipse. Codul este structurat sub forma unei librării facilitând astfel reutilizarea sa în proiecte similare. În timpul implementării sa considerat drept un *overhead (eng.)* suplimentar utilizarea unui sistem de operare real-time, implementându-se un sistem simplist de management cu ajutorul unor timere. Sistemul decizional are la bază un *state machine (eng.)* care decide acțiunea ce trebuie executată, având ca intrare mesajul recepționat wireless de la aplicația Desktop.

Motoarele folosite permit rotirea la cel mult 180 de grade oferind posibilitatea executării unor mișcări diverse. Din păcate, în urma testelor a fost identificat un *backlash (eng.)* (distanța maximă sau unghiul prin care orice parte a unui sistem mecanic pot fi deplasate într-o singură direcție, fără a aplica o forță apreciabilă sau mișcare la următoarea parte din succesiune mecanică) semnificativ și o eroare de poziționare mare ce era dependentă de poziția de start. Sensibilitatea la deplasări pe distanțe scurte fiind redusă. Tentativele inițiale de calibrare nu au adus îmbunătățiri semnificative astfel încât să impus realizarea unui algoritm adaptiv care să poată ignora/trata erorile de poziționare ce ar putea afecta capacitatea de deplasare a robotului, mai ales în situații la limita. O altă dificultate întâlnită a

fost cauzată de masa mare a exoscheletului robotului, care împreună cu masa motoarelor ajunge să cântărească peste 2.5kg. Acest fapt a necesitat stabilirea unor constrângeri în privința brațului de forță cât și o balansare mai atentă a distribuției forței pe motoare pentru a evita suprasolicitarea lor și implicit uzura accelerată.

Inițial s-a optat pe o strategie de deplasare ce implica calcularea poziției membrelor prin metoda cinematicii inverse ce presupune determinarea unghiurilor fiecărui motor în funcție de dimensiunea brațelor și poziția finală în coordonate x, y, z. Deși determinarea modelului matematic a decurs cu succes, testele efectuate au fost afectate de erori mecanice de poziționare majore făcând într-un final nefezabilă utilizarea unui astfel de algoritm într-o situație de acest gen, cel puțin nu într-o formă brută.

Pe parcursul dezvoltării robotului a fost realizat și un simplu interpretor care parsează și interpretează comenzile către robot permițând controlul direct, prin comandarea serială driverului motoarelor. Soluția în cauză a facilitat testarea algoritmilor inițiali de deplasare a robotului și a permis o estimare a ordinului de mărime a erorilor de poziționare a motoarelor. De asemenea s-a implementat și un sistem de log-uri pentru a salva un istoric al acțiunilor efectuate și a facilita astfel depanarea aplicației relativ la mișcările și deciziile luate.

Lucrarea efectuează și o analiză comparativă a algoritmilor de deplasare pentru roboții hexapodi, fiind definite o serie de parametri ce cuantifică performanța fiecărui stil de deplasare. Accentul sa pus pe toleranța la erori, randamentul energetic al algoritmului și stabilitate.

Dezvoltarea ulterioară va fi axată în vederea optimizării gate-urilor de deplasare pentru a minimiza consumul de energie și spori viteza de deplasare prin implementarea unui algoritm ”liber” de deplasare. De asemenea se încearcă generalizarea adaptabilității algoritmului pe un număr cât mai ridicat de suprafețe, oferindu-se o soluție fiabilă, rezistentă la factori de eroare externi și interni.

