

Neizrazito, evolucijsko i neuro računarstvo

Projekt

1. dio – Sustav za neizrazito upravljanje

Uvod

U prvom djelu projekta iz NENR-a potrebno je ostvariti sustav za neizrazito upravljanje koji upravlja brodom koji plovi po kanalu. U svrhu lakše vizualizacije i vrednovanja ostvarenog sustava razvijen je simulator ponašanja broda. Cilj prvog djela projekta je napraviti sustav koji može uspješno (bez sudara s obalom) upravljati brodom.

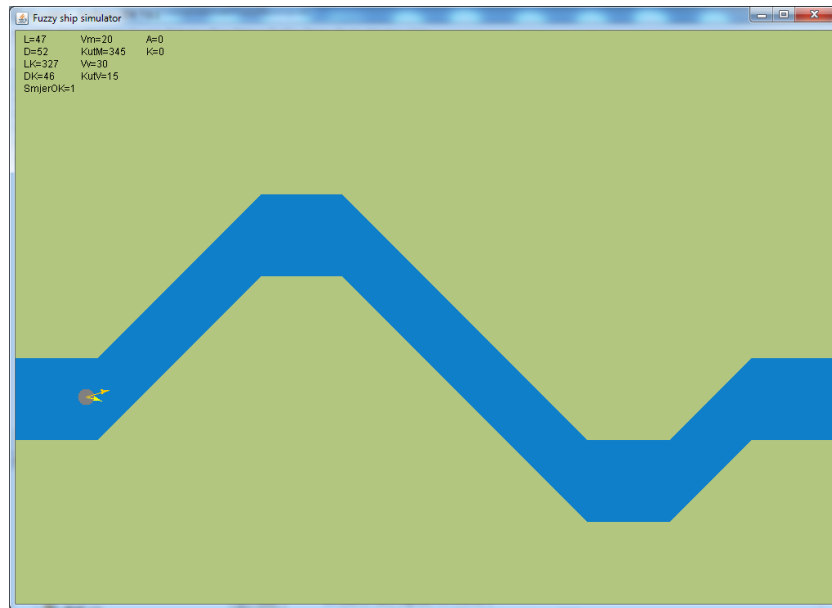
Simulator

Za potrebe simulacije i vizualizacije rada sustava za neizrazito upravljanje koristi se pojednostavljeni model broda koji plovi po kanalu. Brod je predstavljen kružićem polumjera 10 piksela. Dimenzije mape po kojoj brod plovi su 1000 piksela (širina) i 700 piksela (visina).

Na kretanje broda utječu sile koje stvaraju motor broda i vjetar. Vektor brzine broda može se rastaviti na komponentu motora (V_m , na slici 1 označena žuto) i komponentu vjetra (V_v , na slici 1 označena narančasto). One su određene svojim kutem (u odnosu na horizontalu) i iznosom. Iznosi ovih vektora mogu se interpretirati kao brzina u danom smjeru izražena u piksel/s. Ukupan vektor brzine broda dobiva se kao zbroj ove dvije komponente.

Iznos i smjer vjetra polako se mijenjaju kroz vrijeme na slučajan način (postupno dakle bez naglih promjena). Komponentu V_m brzine kontrolira sustav upravljanja brodom preko dva parametra:

- A – akceleracija
 - ovaj parametar modelira pritisak na papučicu gasa ili papučicu kočnice
 - pozitivna akceleracija iznosa X uzrokovati će povećanje iznosa V_m za X piksela/s svake 1s
 - smanjenje iznosa V_m moguće je postići na isti način negativnim iznosom akceleracije
 - apsolutni iznos akceleracije ne može biti veći od 35

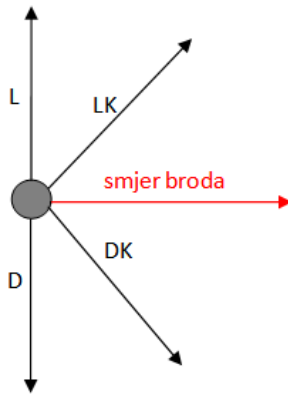


Slika. 1: Simulator broda koji plovi po kanalu.

- V_m nije moguće smanjiti ispod 0
- K – kut kormila
 - ovaj parametar modelira zakretanje broda pomoću kormila
 - kut kormila od X° uzrokuje zakretanje V_m komponente brzine za kut X svakih 1s.
 - pozitivni kutevi uzrokuju skretanje u lijevo (u smjeru suprotnom od kazaljke na satu)
 - negativni kutevi uzrokuju skretanje u desno
 - zbog modeliranja ograničenih mogućnosti zakretanja broda dopušteni raspon vrijednosti je $[-90, 90]$
 - ako se zada vrijednost izvan dopuštenog intervala ona se zaokružuje na najbližu dopuštenu vrijednost

Odluku o vrijednostima upravljačkih parametara sustav za upravljanje donosi na temelju sljedećih informacija:

- L – udaljenost broda od obale prema lijevo
- D – udaljenost broda od obale prema desno
- LK – udaljenost broda od obale naprijed lijevo pod kutem od 45°



Slika 2: Smjerovi u kojima se mjere udaljenosti L, D, LK i DK uz dani smjer broda.

- DK – udaljenost broda od obale naprijed desno pod kutem od 45°
- V – iznos brzine broda (iznos vektora V_m)
- S – podatak o tome kreće li se brod u otprilike pravom smjeru (1 ako da, 0 inače)

Vrijednosti L, D, LK i DK izračunavaju se u odnosu na trenutni smjer u kojem je brod okrenut tj. smjer broda na slici 2 je smjer vektora V_m . Vrijednosti ovih veličina kreću se u intervalu $[0, 1300]$. Ako udaljenost broda od obale padne ispod 10 (polumjer broda) brod se sudario s obalom i simulacija prestaje. Ako se u nekom od smjerova L, D, LK ili DK nalazi ulaz ili izlaz iz kanala (npr. kada je brod blizu izlaza) smatra se da je udaljenost od obale u tom smjeru jako velika i pripadajuća vrijednost biti će 1300.

Za brod se smatra da ide u otprilike pravom smjeru ako njegov stvarni smjer ne odudara za više od 90° od predefiniranog ispravnog smjera.

Format rješenja

Simulator će komunicirati s vašim programom pomoću standardnog ulaza i izlaza. Sustav za neizrazito upravljanje može se ostvariti u proizvoljnom programskom jeziku tako da obavlja pseudokod prikazan u nastavku.

```
dok(true)
    lnIn = pročitaj liniju sa standardnog ulaza
    ako je lnIn = "KRAJ", prekini
    lnOut = izracunajIzlaze (lnIn)
    zapiši liniju lnOut na standardni izlaz
    napravi flush standardnog izlaza
```

Pri tome će linija *lnIn* koju ćete dobiti sadržavati 6 cijelih brojeva odvojenih razmacima i to redom *L*, *D*, *LK*, *DK*, *V* i *S*. Linija *lnOut* koju zapisujete mora sadržavati dva cijela broja odvojena razmakom i to redom *A* i *K*. Važno je napraviti **flush** standardnog izlaza nakon što ste na njega zapisali podatke jer inače komunikacija **neće raditi**. Na kraju simulacije linija *lnIn* će sadržavati riječ "KRAJ".

Kada ste implementirali svoj sustav neizrazitog upravljanja, potrebno je inicijalno podesiti simulator. U direktoriju simulatora nalazi se datoteka imena `config.txt`. U njen prvi red potrebno je upisati naredbu kojom se pokreće vaš program (npr. `"myFuzzyControlSys.exe"` ili `"python myFuzzyConSys.py"`). U njenom drugom redu potrebno je navesti broj staze koja će se koristiti (1, 2 ili 3). Postoje tri staze različitih težina. Konačno, treći i četvrti redak datoteke sadrže maksimalnu i minimalnu brzinu vjetra.

Primjerice ako se vaš program pokreće naredbom `"C:\fuzzy\fuzzCtrl.exe"`, želite koristiti drugu od ukupno tri definirane staze, maksimalnu brzinu vjetra 25 a minimalnu 10, vaša `config.txt` datoteka će izgledati ovako:

```
C:\fuzzy\fuzzCtrl.exe
2
25
10
```

Sada možete pokrenuti simulator i vidjeti kako vaš sustav upravlja brodom. To možete napraviti duplim klikom na datoteku `Simulator.jar` ili iz konzole koristeći sljedeću naredbu: `java -jar Simulator.jar`. Simulator će interno pozivati vaš program koristeći naredbu koju ste upisali u datoteku `config.txt`, ako pokretanje ne uspije na konzoli ćete dobiti ispis s porukom o grešci.

U arhivi sa Ferka osim Simulatora možete naći i primjere jednostavnih sustava upravljanja u jezicima C, C#, Java i Python koje možete iskoristiti kao predložak. Također, za lakše otklanjanje grešaka, simulator će prepisivati `stderr` vašeg sustava na svoj izlaz.

Ako nemate instaliran Java Runtime Environment potreban za pokretanje simulatora on se može preuzeti na <http://www.oracle.com/technetwork/java/javase/downloads/jre-7u9-downloads-1859586.html>.

Zadatak

Potrebno je napraviti sustav neizrazitog upravljanja koji će na temelju ulaznih vrijednosti *L*, *D*, *LK*, *DK*, *V* i *S* izračunati odgovarajuće vrijednosti za akceleraciju i kut kormila. Primarni cilj sustava je uspješno doći od početka do kraja kanala bez sudara s obalom. Sekundarni cilj je probati to ostvariti u što kraćem vremenu. Iako snagu vjetra možete mijenjati, dovoljno je da vaš sustav radi ispravno uz pretpostavljene vrijednosti (snaga između 10 i 20).

Prilikom izrade rješenja, nije potrebno pravila pisati u tekstualnom formatu pa pisati parser za njihovu obradu. Međutim, interno, program mora imati bazu pravila u koju se pravila dodaju na početku programa (programski). Rješenje

mora biti izvedeno tako da se nova pravila mogu jednostavno dodati u bazu pravila.

U rješenju se također moraju jasno razaznati modeli univerzalnog skupa te neizrazitog skupa. Implementacija operatora mora biti takva da se jednostavno podesi na jednom mjestu. Primjerice, odluka hoće li se za operator I koristiti *min* ili *Hamacherova parametrizirana t-norma* uz određenu vrijednost parametra mora se moći podesiti točno na jednom mjestu.

Za izgradnju sustava za neizrazito upravljanje možete koristiti bilo koju od metoda spomenutih na predavanjima. Pri tome ulazne vrijednosti možete tretirati kao točne vrijednosti (uređaji za njihovo mjerenje su iznimno precizni i nema nesigurnosti u izmjeranim veličinama).