

Alati za simulaciju umreženih sistema upravljanja: TrueTime pristup

Nataša Popović

Odsjek za automatiku i elektroniku
Elektrotehnički fakultet
Istočno Sarajevo, Bosna i Hercegovina
natasa.popovic@etf.unssa.rs.ba

Milica Naumović

Katedra za automatiku
Elektronski fakultet
Niš, Srbija
milica.naumovic@elfak.ni.ac.rs

Sadržaj — U radu je dat pregled alata za simulaciju umreženih sistema upravljanja. Razmotreni su softverski paketi koji simuliraju kako upravljački dio tako i komunikacionu mrežu, te su navedeni osnovni problemi koji se javljaju pri tome. Detaljnije je opisan alat TrueTime koji je razvijen kao proširenje Matlab-a. Njegove mogućnosti ilustrovane su na konkretnom primjeru.

Ključne riječi - upravljanje, simulacija, umreženi sistemi upravljanja, TrueTime

I. UVOD

Umreženi sistemi upravljanja su upravljački sistemi koji se sastoje od objekta upravljanja, aktuatora, senzora i regulatora čiji se rad koordiniše preko komunikacione mreže [1]-[5]. Pojednostavljeni blok dijagram umreženog upravljačkog sistema je prikazan na Slici 1.

Korišćenje višenamjenske djeljive komunikacione mreže za povezivanje prostorno distribuisanih elemenata rezultovalo je fleksibilnom arhitekturom umreženih sistema upravljanja i dovelo je do smanjenja troškova prilikom instalacije i održavanja [5]. Kao posljedica toga, umreženi sistemi upravljanja su pronašli primjenu u velikom broju različitih oblasti, kao što su industrijski sistemi upravljanja, transportni, elektroenergetski, vodovodni, bespilotne letjelice, itd.

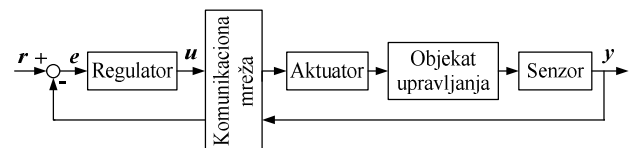
Umreženi sistemi upravljanja objedinjuju teoriju upravljanja i teoriju komunikacija. Komunikaciona mreža se u ovim sistemima razmatra eksplicitno jer značajno utiče na dinamičko ponašanje upravljačkog sistema. Rad umreženog upravljačkog sistema uglavnom zavisi od dvije stvari koje unosi mreža: kašnjenje u komunikaciji i gubitak paketa. Neki autori su pokušali da kašnjenje matematički modeluju, a drugi su pokušali da predvide gubitak paketa simulacijom brzine kojom se oni gube, pri čemu su koristili Bernulijeve procese [6]-[9]. Međutim, ovakvi matematički opisi nisu dali dobre rezultate jer nisu omogućavali analitičko razmatranje sistema, već su se kao takvi mogli koristiti samo za numeričku analizu i poređenje numerički datih sistema.

Sa ciljem da se kompenzuju različiti negativni uticaji koje u upravljački sistem unosi mreža, osim već navedenih,

predloženi su i drugi različiti pristupi [10]-[13]. Ovi pristupi imaju određene prednosti sa stanovišta analize jer, na primjer, omogućavaju formalno navođenje uslova i dokazivanje stabilnosti sistema. Sa druge strane, pretpostavke koje se uvode u ovakvim pristupima mogu biti nerealne i ponekad ne mogu na adekvatan način opisati pravu dinamiku umreženog sistema. Kada se u fazi projektovanja sistema u obzir uzmu takve pretpostavke, one mogu dovesti do ozbiljnih posljedica kasnije, kada se realizuju stvarni fizički sistemi. Takođe, pitanja vezana za protokole koji se koriste u upravljačkim komunikacionim mrežama se vrlo teško mogu iskazati analitički.

Kako umreženi sistemi upravljanja u današnje vrijeme postaju sve kompleksniji, istraživačima je i dalje izazov kako formalno analizirati performanse i stabilnost sistema, te kako obezbijediti njihovu sigurnost i bezbjednost. Zbog toga je potrebno vršiti evaluaciju kako upravljačkih tako i mrežnih komponenti umreženog sistema istovremeno, za šta je pogodno koristiti simulacione alate. Rezultati simulacije bi trebalo da procijene validnost i praktične benefite razvijenih teorija i algoritama.

Rad je organizovan na sljedeći način. U drugom poglavlju su dati najčešće korišćeni softverski paketi za simulaciju upravljačkih sistema i komunikacionih mreža posebno, dok se u trećem poglavlju razmatraju proširenja postojećih paketa i kosimulatori. U četvrtom poglavlju je dat primjer simulacije umreženog upravljačkog sistema korišćenjem TrueTime softvera. Na kraju su dati zaključci rada.



Slika 1. Pojednostavljeni blok dijagram umreženog upravljačkog sistema

II. ALATI ZA SIMULACIJU UMREŽENIH SISTEMA UPRAVLJANJA

Softverski paketi za simulaciju umreženih upravljačkih sistema treba da obuhvate i predstave upravljačke i komunikacione aspekte za verifikaciju, validaciju i procjenu različitih upravljačkih i mrežnih algoritama. Međutim, malobrojni trenutno dostupni alati za simulaciju umreženih sistema upravljanja su uglavnom orijentisani ili na mrežni ili na upravljački dio.

Kao dobro rješenje za simulaciju umreženih sistema upravljanja pokazao se razvoj takozvanih kosimulatora. Kosimulatori su softverska proširenja postojećih alata za simulaciju koja omogućavaju da alati za simulaciju zajedno sa njima istovremeno simuliraju ponašanje svih dijelova cjelokupnog umreženog upravljačkog sistema. Strategija zasnovana na kosimulatorima može biti osnova za razvoj budućih alata koji će omogućiti simulaciju i upravljačkog i mrežnog dijela sistema u isto vrijeme [14]-[16].

U odnosu na pojedinačne alate za simulaciju, kosimulatori imaju sljedeće prednosti [17]:

- kada se koriste kosimulatori, obuhvaćeni su svi aspekti umreženog upravljačkog sistema i može se proučavati njegovo ponašanje u cjelini, što nije slučaj kada se simuliraju upravljački i komunikacioni dio sistema posebno,
- pomoću kosimulatora se može analizirati ponašanje složenih sistema koji u sebi sadrže veliki broj upravljačkih petlji, što nije moguće postići matematičkim putem.

A. Softverski paketi za simulaciju fizičkih sistema

Jedni od najčešće korišćenih softverskih paketa za simulaciju upravljačkih sistema su Matlab [18] i Modelica [19]. Oba paketa imaju mogućnost nadogradnje proširenjima namijenjenim za različite primjene, odnosno moguće im je dodati kosimulatore.

U Matlab-u se to postiže na dva načina: preko MX interfejsa koji omogućava programima napisanim u jeziku C da pozivaju funkcije napisane u Matlab-u, i preko generisanja koda za šta su razvijeni Matlab Coder i Simulink Coder koji omogućavaju da generisani kod napisan u jeziku C može da se importuje u druge softverske pakete za simulaciju.

Modelica je objektno orijentisani jezik za modelovanje kompleksnih fizičkih sistema koji u svojim komercijalnim i *open source* okruženjima, kao što su Dymola i SimulationX, ima mogućnost eksportovanja modela u druge simulacione alate. Tako, Dymola pomoću bloka DymolaBlock generiše S-funkcije iz Modelica modela i omogućava im da se simuliraju zajedno sa modelima razvijenim u drugim simulacionim paketima. SimulationX ima COM interfejs koji omogućava povezivanje modela generisanih u Modelica-i i drugih simulacionih paketa preko jezika C#.

Osim ovih softverskih paketa, upotrebljavaju se i VTB (za simulaciju multidisciplinarnih dinamičkih sistema), COMSOL (za različite inženjerske i fizičke aplikacije), SIMPLORER (za projektovanje, modelovanje, analizu i optimizaciju visokopformansnih sistema), itd [17].

B. Softverski paketi za simulaciju komunikacione mreže

Softverski paketi koji se najčešće koriste za simulaciju komunikacione mreže su: NS-2 [20], OPNET [21] i OMNet++ [22].

NS-2 (*Network Simulator*) je *open source* softver namijenjen za simulaciju dinamike paketa podataka prilikom njihovog prolaska kroz različite mrežne elemente, mrežnih protokola na različitim nivoima ISO/OSI referentnog modela, te konfiguracije različitih topologija mreže i mrežnih tehnologija. Baziran je na TCL (*Tool Command Language*) skriptu u kome se specificiraju mrežni parametri i protokoli koji se mogu proširiti dodavanjem novih klasa napisanih u jeziku C++. Iako nema grafički korisnički interfejs i mogućnost korisničkog prilagođavanja i ne podržava hijerarhijsku strukturu, on se veoma često koristi za simulaciju komunikacione mreže. Međutim, kada se koristi za simulaciju umreženih upravljačkih sistema, nije najpogodniji izbor jer je potrebno da algoritam koji se odnosi na upravljački dio sistema čitav bude napisan u jeziku C++, što je izuzetno teško kada se radi o složenijim sistemima upravljanja.

OPNET (*Optimized Network Engineering Tools*) Modeler je komercijalni softver za projektovanje i analizu komunikacione mreže, mrežnih uređaja, protokola i aplikacija. Podržava hijerarhijsku strukturu generisanja modela mreže i ima mehanizme za interakciju sa ostalim softverskim paketima koji se očituju kroz programiranje izvršnog kontrolera koji je sastavni dio OPNET-a i upotrebu HLA (*High Level Architecture*) standarda. U odnosu na NS-2, daje bolje simulacione modele i konfiguraciju mreže, te omogućava vizuelizaciju preko grafičkog korisničkog interfejsa.

OMNet++ (*Objective Modular Network Testbed in C++*) je još jedan *open source* softverski paket koji služi za simulaciju komunikacione mreže. Ima integrisan dodatak Mobility Framework na osnovu koga može da podrži različite formate paketa, te grafičke interfejse, hijerarhijsku strukturu generisanja modela i različite mrežne topologije koje definiše korisnik. Kao i prethodna dva softverska paketa, posjeduje mehanizme za proširenje u obliku kosimulatora.

Od ostalih softverskih paketa za simulaciju komunikacione mreže mogu se pomenuti sljedeći: SystemC, NS-3, SymPy, RTNS, ali njihova upotreba nije toliko rasprostranjena [17].

III. SOFTVERSKA PROŠIRENJA I KOSIMULATORI

Postoje dva osnovna pristupa u korišćenju postojećih alata za simulaciju umreženih upravljačkih sistema koji uzimaju u obzir kako dinamiku sistema upravljanja tako i komunikacionu mrežu [23]. Prvi pristup podrazumijeva proširenje funkcija simulacionog softvera čime se omogućava simulacija dinamike upravljačkog sistema i komunikacione mreže istovremeno. Ovaj pristup se dalje može podijeliti na pristup koji uključuje proširenje softvera za simulaciju dinamičkog ponašanja sistema upravljanja i pristup koji uključuje proširenje softvera za simulaciju mreže. Drugi pristup podrazumijeva integraciju dva ili više nezavisnih alata namijenjenih za simulaciju određenog domena (upravljačkog

sistema ili komunikacione mreže) čime se formiraju kosimulatori koji mogu da se koriste za simuliranje umreženih sistema upravljanja.

A. Softverski alati za proširenje

Alati koji su namijenjeni za proširenje softvera za simulaciju dinamičkog ponašanja sistema upravljanja (fizičkog sistema) treba da omoguće precizno modelovanje mrežnog stek protokola da bi se pomoću njega mogao simulirati mrežni dio umreženog upravljačkog sistema. U ovu grupu alata ubrajaju se dodaci za Matlab [17]: TrueTime, Prowler i WSN simulator. TrueTime ima mogućnost simuliranja kako žičanih tako i bežičnih umreženih sistema, dok druga dva simuliraju samo bežične sisteme. Prowler može da simulira bežične sisteme od aplikacionog do fizičkog nivoa ISO/OSI referentnog modela, ali je njegov MAC nivo modelovan veoma jednostavno. WSN (*Wireless Sensor Network*) simulator je baziran na standardu IEEE 802.15.4 i ima relativno dobre mogućnosti modelovanja fizičkog i MAC nivoa, te grafički korisnički interfejs koji korisnicima omogućava jednostavno kreiranje različitih topologija mreže i izvođenja simulacije bežičnih sistema.

Primjeri proširenja softvera za simulaciju mreže u cilju simuliranja umreženih sistema upravljanja su malobrojni. Ovdje se mogu izdvojiti Agent/Plant i OPNET proširenje [17]. Agent/Plant, odnosno njegove klase NSCSPant i NSCSController, predstavljaju proširenja NS-2 simulatora koje omogućavaju simulaciju dinamičkog ponašanja fizičkog sistema i upravljanja istim. Ova proširenja definišu agente, čije se dinamičko ponašanje opisuje algebarskim linearnim diferencijalnim jednačinama, a imaju zadatak da simuliraju proces (sistem) i regulator. Osnovni problem koji se javlja u ovom pristupu je taj što se modelovanje dinamičkog ponašanja i upravljanja vrši eksplicitno pomoću algebarskih linearnih diferencijalnih jednačina za čije rješavanje se mora napisati poseban skript dokument ili pozvati neki drugi softver, na primjer Matlab. Što složenost sistema koji treba simulirati raste, to ovaj problem postaje izraženiji. OPNET proširenje je prvenstveno namijenjeno za simulaciju WAN (*Wide Area Network*) mreže u elektroenergetskim sistemima. Dinamika sistema se modeluje virtuelnim zahtjevom koji, kada treba da se pošalje podatak preko mreže, kreira paket u OPNET-u. OPNET zatim simulira ukupno kašnjenje paketa i predaje ga zahtjevu. Ovo proširenje je moguće koristiti u slučaju samo jednog agenta, što znatno ograničava njegovu upotrebu.

B. Integracija nezavisnih softverskih alata

Integracija dva ili više nezavisnih alata namijenjenih za simulaciju određenog domena bazirana je uglavnom na standardu HLA ili DEVS (*Discrete Event System Specification*). Neki od primjera ove vrste su [17]:

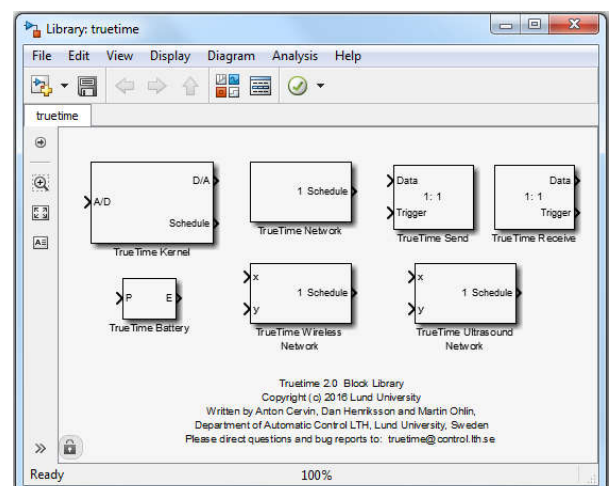
- PiccSIM (*Platform for integrated communications and control design, Simulation, Implementation and Modeling*) je baziran na korišćenju dva računara od kojih jedan koristi Matlab i PiccSSIM Toolchain za

simulaciju upravljanja a drugi NS-2 za simulaciju mreže. Namijenjen je za bežične sisteme, prvenstveno senzorske mreže i bežičnu automatizaciju.

- NMLab je kombinacija Matlab-a i NS-2 i namijenjen je specijalno za simulaciju umreženih sistema upravljanja. I pored toga što objedinjuje dva napredna softverska paketa, problematičan je u slučajevima simuliranja složenijih sistema.
- Matlab/Simulink+OPNET je razvijen za simulaciju bežičnih sistema preko *ad hoc* mreže pomoću OPNET-a i Matlab/Simulink-a. Pomoću njega je moguće ispitati uticaj kašnjenja i gubitka paketa, te promjene brzine prenosa podataka na stabilnost i performanse sistema.
- COSMO je baziran na Matlab-u i OMNeT++ i koristi se za simulaciju bežičnih sistema u zatvorenom prostoru. Loša strana mu je to što, u slučaju promjene modela u bilo kom od ova dva alata, potrebno je ispočetka napisati cjelokupan simulacioni kod.
- Modelica+NS-2 koristi Modelica-u za simulaciju dinamike i upravljanja sistema, a NS-2 za simulaciju mreže. Simulacijom upravlja NS-2, dok Modelica treba da obavlja svoje zadatke u unaprijed određenom vremenskom intervalu. Pošto na tok simulacije ne može uticati nikakav nepredviđeni događaj, ovaj simulator ne može na adekvatan način da simulira realne smetnje koje se mogu pojaviti u umreženom sistemu upravljanja.

IV. TRUETIME

TrueTime je simulacioni alat koji je namijenjen za simuliranje umreženih sistema upravljanja i embedid sistema, a baziran je na Matlab/Simulink-u [24], [25]. Intenzivno se razvija od 1999. godine i prati napredak u razvoju komunikacija i upravljačkih sistema, tako da njegove novije verzije pružaju mogućnost simulacije velikog broja različitih



Slika 2. Biblioteka TrueTime

sistema i tehnologija prenosa podataka, kao i mogućnost integracije sa softverskim paketima Modelica i PiccSIM. Za razliku od nekih drugih sličnih softverskih alata, TrueTime nije baziran na principu matematičkog modelovanja. Primarno je razvijen sa ciljem generisanja simulacije pisanjem koda u Matlab-u (m-fajl) ili jeziku C++.

TrueTime softver se sastoji od biblioteke sa Simulink blokovima i kolekcije MEX fajlova, Slika 2. Blok TrueTime Kernel je Simulink-ova S-funkcija koja simulira računar sa kernelom koji u realnom vremenu izvršava korisničke zadatke i obrađuje prekide. TrueTime Network blok omogućava čvorovima (kernel blokovima) u simuliranoj mreži da međusobno komuniciraju. Ovaj blok simulira pristup medijumu za prenos podataka i slanje paketa podataka (MAC i fizički nivo ISO/OSI referentnog modela) u lokalnoj mreži (LAN - *Local Area Network*). Podržava sljedeće tipove mreže: CSMA/CD (Ethernet), CSMA/AMP (CAN), Round Robin (Token Bus), FDMA, TDMA, Switched Ethernet, FlexRay, PROFINET i NCM. Samostalni TrueTime Send i TrueTime Receive blokovi su interfejsi za mrežni blok koji omogućavaju jednostavniju komunikaciju čvorova u mreži. Oni služe za slanje poruka u mreži bez korišćenja kernel blokova. Drugim riječima, kada se ovi blokovi koriste nije potrebno pisati m-fajlove niti C++ kod, dovoljno je koristiti Simulink blokove. Blokovi TrueTime Wireless Network i TrueTime Ultrasound Network služe za simulaciju bežičnih i ultrazvučnih sistema, dok se TrueTime Battery blok ponaša kao izvor napajanja za kernel blokove kojima je to potrebno. Tipovi mreže koje podržava blok TrueTime Wireless Network su: 802.11b (WLAN), 802.15.4 (ZigBee) i NCM_WIRELESS.

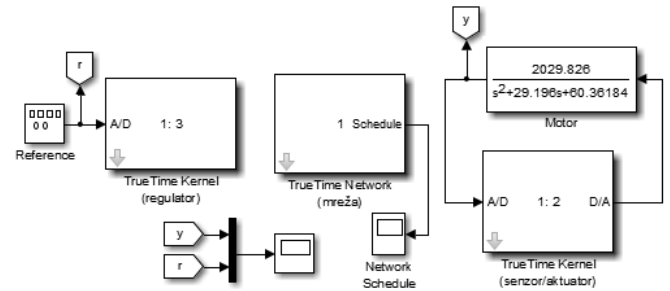
Svaki od navedenih blokova je konfigurisan u obliku maske, u koju korisnik upisuje odgovarajuće vrijednosti ponuđenih parametara. Moguće je u jednoj simulaciji kombinovati kernel i samostalne blokove, pri čemu za kernel blokove treba napisati odgovarajuće kodove. Prije pokretanja simulacije potrebno je izvršiti inicijalizaciju kernel blokova i bloka *Network*, što se postiže pokretanjem inicijalizacionog koda.

A. Simulacija upravljanja istosmjernim motorom

Na Slici 3. prikazan je blok dijagram umreženog sistema upravljanja čiji je objekat upravljanja istosmjerni motor opisan funkcijom prenosa [26]

$$G(s) = \frac{2029.826}{(s+26.29)(s+2.296)}. \quad (1)$$

U ovom simulacionom modelu korišćena su tri kernel bloka koji simuliraju regulator, komunikacionu mrežu i senzor sa aktuatorom. Senzor i aktuator su objedinjeni u jedan kernel blok radi pisanja jednog koda koji istovremeno upravlja razmjennom poruka i senzora i aktuatora, umjesto pisanja dva koda za senzor i aktuator posebno. Blok regulator preko bloka mreža prima poruke od senzorskog čvora, generiše upravljački signal i šalje ga aktuatorskom čvoru ponovo preko bloka mreža. Regulator je digitalni PID regulator [27] oblika



Slika 3. Blok dijagram umreženog sistema upravljanja sa kernel blokovima

$$P(k) = K(\beta r(k) - y(k))$$

$$I(k+1) = I(k) + \frac{Kh}{T_i}(r(k) - y(k)) \quad (2)$$

$$D(k) = \frac{T_d}{Nh + T_d} D(k-1) + \frac{NKT_d}{Nh + T_d}(y(k-1) - y(k))$$

gdje je K proporcionalno pojačanje, T_i vremenska konstanta integralnog dejstva, T_d vremenska konstanta diferencijalnog dejstva, h je perioda odabiranja, a N je parametar cjelobrojne vrijednosti.

Kod koji služi za generisanje upravljačkog signala u bloku regulator dat je u Listingu 1. Podešavanje parametara regulatora vrši se u fajlu za inicijalizaciju.

LISTING 1. KOD ZA GENERISANJE UPRAVLJAČKOG SIGNALA

```
function [exectime,data] = controller_code(seg,data)

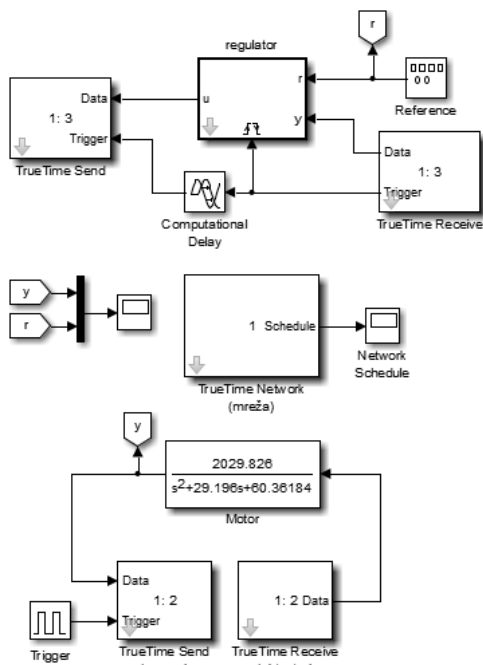
switch seg
case 1
    y = ttGetMsg; % prima vrijednost koju šalje
    senzor
    if isempty(y)
        disp('Greška: nijedna poruka nije primljena!');
        y = 0.0;
    end
    r = ttAnalogIn(1); % čita referentnu vrijednost
    P = data.K*(r-y);
    D = data.Td/(data.N*data.h+data.Td)*data.Dold+
    +data.N*data.K*data.Td/(data.N*data.h+data.Td)*
    *(data.yold-y);
    I = data.Iold;
    data.u = P + I + D;
    data.Iold = data.Iold+data.K*data.h/data.Ti*(r-y);
    data.yold = y;
    data.Dold = D;
    exectime = 0.0005;
case 2
    ttSendMsg(2,data.u,80); % šalje upravljačke
    podatke aktuatoru
    exectime = -1; % kraj
end
```

Blok mreža služi za razmjenu podataka između dva mrežna čvora koji su dati u obliku blokova regulator i senzor/aktuator. Ovaj blok prima upravljački signal od bloka regulator i očitavanja senzorskog čvora iz bloka senzor/aktuator, a šalje upravljački signal aktuatorskom čvoru bloka senzor/aktuator i očitavanja senzorskog čvora iz bloka senzor/aktuator bloku regulator. Dakle, blok mreža ima dva ulazna i dva izlazna

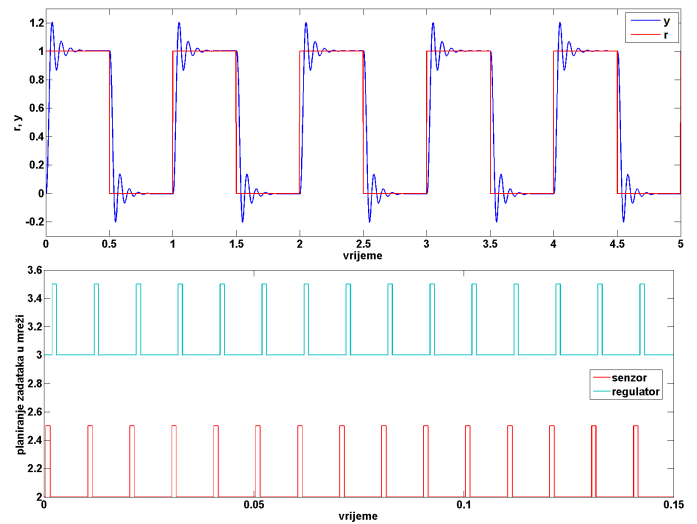
porta. Poruke koje dolaze u blok mreža čuvaju se u odgovarajućim ulaznim portovima u obliku redova čekanja koji funkcionišu na principu FIFO (*first-in-first-out*). One se dalje premještaju u unutrašnjost bloka mreža i konačno šalju na odgovarajuće izlazne portove shodno mrežnom protokolu koji se koristi. Od vrste protokola zavisi i upravljanje kolizijom. Vrijeme predaje poruke zavisi od njene dužine. Pomoću bloka mreža simulira se gubitak paketa što se postiže definisanjem stepena vjerovatnoće da će do toga doći. Blok mreža služi i za planiranje izvršavanja zadataka u mreži, drugim riječima na portu Network Schedule moguće je prikazati u kojim vremenskim trenucima blokovi regulator i senzor/aktuator izvršavaju svoje zadatke.

Blok senzor/aktuator simulira senzorski i aktuatorski čvor u umreženom sistemu upravljanja. Ovaj blok periodično očitava vrijednosti izlaza objekta upravljanja i šalje ih bloku regulator. Isto tako, prima upravljački signal iz bloka regulator i prosljeđuje ga objektu upravljanja.

Blok dijagram ekvivalentan onom sa Slike 3, u slučaju kada se umjesto kernel blokova koriste samostalni blokovi, prikazan je na Slici 4. Objekat upravljanja i regulator su isti, razlika je u tome što se sada koriste posebni blokovi za simuliranje senzora i aktuatora, te za simuliranje ulaznih i izlaznih signala regulatora. Ovi blokovi su dati svojim maskama u koje se upisuju odgovarajuće vrijednosti parametara, odnosno, ponašaju se kao standardni Simulink blokovi. Podešavanje parametara regulatora se obavlja u bloku regulator, a rezultati simulacije su identični onima u slučaju blok dijagrama sa Slike 3. Blok mreža simulira komunikacionu mrežu, i u ovom slučaju ima dva ulazna porta (za signale koji dolaze iz blokova TrueTime Send) i dva izlazna porta (za signale koji se šalju blokovima TrueTime Receive). Iako je broj simulacionih blokova ovdje veći u odnosu na prethodni



Slika 4. Blok dijagram umreženog sistema upravljanja sa samostalnim blokovima



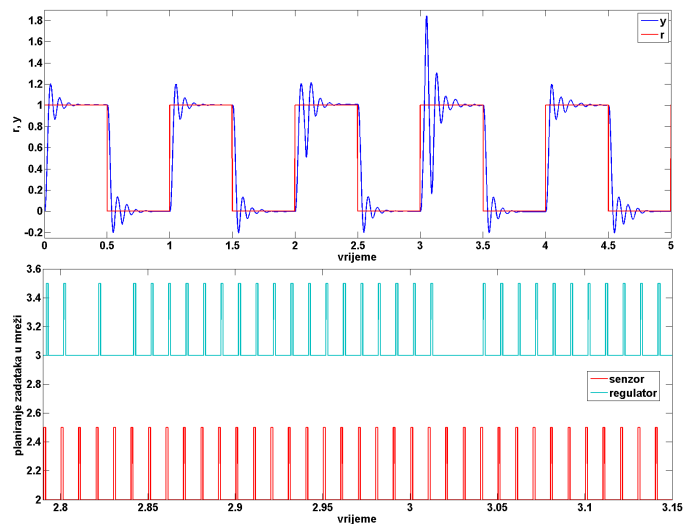
Slika 5. Signal na izlazu objekta upravljanja u odnosu na referentni signal i planiranje zadataka u mreži u slučaju kada nema gubitka podataka

primjer, formiranje ovakvog blok dijagrama je jednostavnije za korisnika jer ne iziskuje pisanje kodova za pojedine blokove.

B. Rezultati simulacije

Za simulaciju upravljanja istosmjernim motorom preko komunikacione mreže odabrana je CSMA/CD (Ethernet) tehnika, odnosno tehnika prenosa podataka koja omogućava detektovanje kolizije. Brzina prenosa podataka je 10 Mb/s. Inicijalno je usvojeno da u sistemu ne dolazi do gubitka paketa sa podacima, a kašnjenja su zanemarena. Za inicijalizaciju kernel blokova odabrana je funkcija *prioDM* (*deadline-monotonic scheduling*). Perioda kojom senzorski čvor očitava vrijednosti izlaza motora je 0.01s. Parametri regulatora su podešeni na sljedeće vrijednosti: $K = 1.6$, $T_d = 0.017s$, $T_i = 1.1s$, $h = 0.01s$ i $N = 100$.

Na Slici 5. prikazan je signal na izlazu motora i planiranje izvršavanja zadataka senzorskog čvora i čvora regulatora u mreži. Može se primijetiti da i senzor i regulator obavljaju



Slika 6. Signal na izlazu objekta upravljanja u odnosu na referentni signal i planiranje zadataka u mreži u slučaju kada postoji gubitak podataka

svoje zadatke u mreži na planirani način, što rezultuje odzivom motora u obliku prikazanom na slici. U slučaju da se u sistemu dogodi gubitak paketa sa podacima, to se odražava na rad sistema na način prikazan na Slici 6. Naime, na ovoj slici se jasno vidi da u trenucima kada se neki paketi izgube (paketi sa podacima koje je poslao senzor) dolazi do nemogućnosti slanja povratnih poruka (regulator ne može da obavi svoj zadatak i da pošalje upravljački signal aktuatoru). Ovo se primjećuje i u odzivu motora, tako da je, u trenucima kada je do gubitka paketa došlo, izražena znatna degradacija izlaznog signala.

V. ZAKLJUČAK

U radu je dat pregled alata za simulaciju umreženih sistema upravljanja. Navedeni su softverski paketi koji se koriste za simulaciju posebno upravljačkog i posebno mrežnog dijela umreženog sistema upravljanja, određeni broj njihovih softverskih proširenja kao i mogućnosti kombinovanja pojedinih alata. Kako je u simuliranju umreženih upravljačkih sistema neophodno da oba njegova aspekta (upravljački i komunikacioni) budu na adekvatan način obuhvaćena, iz ne velikog broja ponuđenih rješenja izdvojio se TrueTime, dodatak Matlab-u. Kroz primjer upravljanja istosmjernim motorom prikazane su njegove osnovne mogućnosti. TrueTime simulira upravljački dio umreženog sistema tako što koristi Simulink blokove ili pisanjem odgovarajućeg koda. Mrežni dio umreženog sistema simulira se pomoću posebnog kernel bloka koji sadrži parametre mreže. Posebno je značajno to što ovaj blok simulira kašnjenja u mreži i gubitak paketa koji se javljaju kao osnovni problemi u umreženim sistemima upravljanja. Buduća istraživanja autora biće bazirana na korišćenju TrueTime alata i kombinovanju različitih parametara sistema, uzimajući u obzir i različite smetnje koje djeluju u njemu, sa ciljem da se na relativno jednostavan način sagleda ponašanje umreženog sistema.

LITERATURA

- [1] P. V. Zhivoglyadov, R. H. Middleton, *Networked Control Design for Linear Systems*, Automatica 39, pp. 743-750, Elsevier Science Ltd. 2003
- [2] Y. Tipsuwan, M. Chow, *Control Methodologies in Networked Control Systems*, Control Engineering Practice 11 pp. 1099-1111, Elsevier Ltd. 2003
- [3] J. Baillieul, P. J. Antsaklis, *Control and Communication Challenges in Networked Real-Time Systems*, Special Issue of the Proceedings of the IEEE, Vol. 95, No 1, January 2007
- [4] Y. Xia, Y. Gao, L. Yan, M. Fu, *Recent Progress in Networked Control Systems – A Survey*, International Journal of Automation and Computing 12 (4), pp. 343-367, August 2015
- [5] J.P. Hespanha, P. Naghshtabrizi, Y.Xu, *A Survey of Recent Results in Networked Control Systems*, Proceedings of the IEEE, 95 (1), 138-162, 2007
- [6] P.H. Bauer, M.L. Sichiitiu, K. Premaratne, *On the Nature of the Time-variant Communication Delays*, In Proceedings of the IASTED Conference on Modeling, Identification and Control (MIC'01), 2001
- [7] L. Huang, K. Sezaki, *End-to-end Internet Delay Dynamics*, In Proceedings of the 6th Asia-Pacific Conference on Communications (APCC), 2000

- [8] Y.B. Zhao, J. Kim, G.P. Liu, *Offline Model Predictive Control-based Gain Scheduling for Networked Control Systems* [brief paper], IET Control Theory & Applications, 6 (16), 2585-2591, 2012
- [9] Y.B. Zhao, J. Kim, G.P. Liu, *Error Bounded Sensing for Packet-based Networked Control Systems*, IEEE Transactions on Industrial Electronics, 58 (5), 1980-1989, 2011
- [10] B. Rahmani, A. H. D. Markazi, *Variable Selective Control Method for Networked Control Systems*, IEEE Transactions on Control Systems Technology 21 (3), 975-982, 2013
- [11] C. Benedetti, M. Franchini, P. Fiorini, *Stable Tracking in Variable Time-delay Teleoperation*, In Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001
- [12] R. Oboe, P. Fiorini, *A Design and Control Environment for Internet-based Telerobotics*, International Journal of Robotics Research, 17 (4), 433-449, 1998
- [13] J. Yao, X. Liu, G. Zhu, L. Sha, *NetSimplex: Controller Fault Tolerance Architecture in Networked Control Systems*, IEEE Transactions on Industrial Informatics, 9 (1), 346-356, 2013
- [14] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, S. Sastry, *Foundations of Control and Estimation Over Lossy Networks*, Proceedings of the IEEE 95 (1), 163-187, 2007
- [15] D. Dačić, D. Nešić, *Simultaneous Controller and Protocol Design for Networked Control Systems with Packet Based Communication*, In Proceedings of the 45th IEEE Conference on Decision and Control, pp. 508-513, 2006
- [16] Z. Lei, H. Dimitrios, *Communication and Control Co-design for Networked Control Systems*, Automatica, 42 (6), 953-958, 2006
- [17] W. Li, X. Zhang, H. Li, *Co-simulation Platforms for Co-design of Networked Control Systems: An Overview*, Control Engineering Practice 23, 44-56, Elsevier Ltd. 2014
- [18] <http://www.mathworks.com/>
- [19] <http://www.modelica.org/>
- [20] www.isi.edu/nsnam/ns/
- [21] <http://www.opnet.com/>
- [22] <http://www.omnetpp.org/>
- [23] E. Eyişi, J. Bai, et al. *NCSWT: An Integrated Modeling and Simulation Tool for Networked Control Systems*, Simulation Modelling Practice and Theory, 27, 90-111, 2012
- [24] A. Cervin, D. Henriksson, M. Ohlin, *TRUETIME 2.0 – Reference Manual*, Department of Automatic Control, Lund University, 2016
- [25] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, K.E. Årzén: "How Does Control Timing Affect Performance? Analysis and Simulation of Timing Using Jitterbug and TrueTime." *IEEE Control Systems Magazine*, 23:3, pp. 16-30, June 2003
- [26] Y. Tipsuwan, M. Y. Chow, *Control Methodologies in Networked Control Systems*, Control Engineering Practice 11, pp. 1099-1111, Elsevier Ltd. 2003
- [27] K. J. Åström, T. Hägglund, *PID Controllers: Theory, Design, and Tuning*, Instrument Society of America, Research Triangle Park, North Carolina. 1995

ABSTRACT

This paper presents an overview of networked control systems simulators. Control system simulators and network simulators are considered, as well as some problems that occur using them. Matlab extension TrueTime is described in more details, and its capabilities are demonstrated on concrete example.

NETWORKED CONTROL SYSTEMS SIMULATION TOOLS: TRUETIME APPROACH

Nataša Popović, Milica Naumović