

RAČUNARSKI SISTEMI U REALNOM VREMENU I NJIHOVA PRIMJENA UZ UPOTREBU Real-Time Toolbox-a

¹Amel Toroman, ²Edin Mujčić

¹Tehnički fakultet Bihać, ulica dr. Irfana Ljubijankića bb, amel.toroman@gmail.com

²Tehnički fakultet Bihać, ulica dr. Irfana Ljubijankića bb, edin.mujcic@gmail.com

Ključne riječi: Sistemi u realnom vremenu, Regulacija nivoa vode, Rezervoar, MATLAB, UDP, Real-Time Toolbox

SAŽETAK:

Računarski sistemi u realnom vremenu zahtijevaju da se određeni zadaci izračunaju unutar strogih vremenskih ograničenja, koje definiraju krajnji rokovi, odnosno vrijeme odgovora u kojem računanje moraju biti završena. Postoje tri različite vrste rokova, hard (teški), soft (meki) i firm (čvrsti). Hard krajnji rok znači da ako računanje ne završi prije krajnjeg roka, može uzrokovati neuspjeh ukupnog sistema. Soft krajnji rok znači da se nakon krajnjeg roka smanjuje vrijednost izračunavanja, ali ne na nulu ili negativan broj. Ako je firm krajnji rok propušten, vrijednost zadatka je izgubljena, ali nije došlo do negativnih posljedica. Područja primjene sistema u realnom vremenu su veoma velika, polazeći od upravljanja kućanskih aparata, motorima automobila, telekomunikacijskim komutacijskim sistemima, industrijskim postrojenjima, upravljanja rezervoarom vode i mnoga druga područja. Sve ovo uključuje prikupljanje podataka iz okoline, obradu prikupljenih podataka, te pružanje pravovremene reakcije.

1. UVOD

Svaki računarski sistem sastoji se od skupa tehničkih komponenti – hardvera, programskog koda – softvera i podataka koji se obrađuju. Kao što je poznato, hardver predstavlja samu mašinu, dok softver predstavlja programe.

Operativni sistem je program koji predstavlja interfejs između hardvera računara i korisnika računarskog sistema. Postoji više definicija računarskih sistema u realnom vremenu, ali bitno je izdvojiti sljedeće:

- Računarski sistem u realnom vremenu je onaj sistem u kojem tačnost računanja ne ovisi samo o logičnoj tačnosti računanja, nego i od vremena u kojem se proizvode rezultati. Ako vremenska ograničenja sistema ne budu ispunjena, dogodit će se pad sistema [1].
- Bilo koja aktivnost obrade informacija (ili sistem) koji na spolja generisane ulaze treba da da odgovor u konačnom, unaprijed specificiranom periodu [2].
- Sistem u realnom vremenu je sistem koji zahtjeva reakciju na pobudu iz okruženja (uzimajući u obzir i tok fizičkog vremena) u vremenskim intervalima određenim od strane okruženja [2].

Riječ *real* pokazuje da se reakcija sistema na vanjske događaje mora desiti tokom njihove evolucije. Riječ *time* znači da ispravnost sistema ne ovisi samo od logičnog rezultata računanja, već i od vremena u kojem se proizvode rezultati. Izraz "real-time" znači sve što je dovoljno "brzo", "stvarno" ili "on-line". Primjena računarskih sistema u realnom vremenu pokriva spektre od onih vrlo jednostavnih do vrlo složenih, te postoje mnoga područja primjene.

2. RAČUNARSKI SISTEMI U REALNOM VREMENU

Operativni sistem je glavna arhitektonska komponenta odgovorna za osiguravanje pravovremenog izvršavanja svih zadataka imajući neke zahtjeve vremena [3].

Za operativne sisteme u realnom vremenu je tipično to da su, zajedno sa računarom, ugrađeni (*embedded*) u sistem, čije ponašanje se ili samo prati, ili čijim ponašanjem se upravlja. Zadatak operativnih sistema u realnom vremenu je da samo stvore okruženje za korisničke programe, jer komunikaciju sa krajnjim korisnikom obavljaju korisnički programi [4].

Sistem u realnom vremenu je računarski sistem koji zahtijeva ne samo da izračunati rezultat mora biti "tačan", ali isto tako taj rezultat mora biti proizveden unutar određenog vremenskog roka. Rezultati proizvedeni nakon isteklog roka čak i ako su ispravni ne mogu biti od stvarne vrijednosti.

Oni su obično sastavni dio nekog većeg sistema ili uređaja i iz tog razloga se nazivaju ugrađeni (ili *embedded*) mikroprocesorski sistemi. Ključni dio specifikacije računarskih sistema u realnom vremenu se odnosi na *vrijeme odziva*, koje je određeno prirodom fizičkog procesa. Korektnost rada računarskih sistema u realnom vremenu zavisi ne samo od logičke ispravnosti rezultata izračunavanja, već i od vremena za koje se ti rezultati generišu. Drugim riječima, da bi korektno obavio svoj zadatak, računarski sistem u realnom vremenu mora svoju funkciju da izvrši u unaprijed zadatom vremenu. Ukoliko se zadato vrijeme odziva prekorači može doći do otkaza sistema što može imati katastrofalne posljedice [5].

Računarski sistem u realnom vremenu se sastoji od *upravljačkog* i *upravljanog* sistema. Upravljački sistem (ili *embedded system*) predstavlja elektronski podsistem fizičkog sistema. Upravljeni sistem se može posmatrati kao *okruženje* sa kojim upravljački sistem interaguje. Na primjer, u slučaju automatizovane fabrike, *upravljeni sistem* je fabrički pogon koji se sastoji od: robota, manipulatora, pokretnih traka itd, dok *upravljački sistem* čini računar zajedno sa interfejsom prema operateru, koji upravlja i koordinira svim aktivnostima u fabričkom pogonu.

Osnovne funkcije računarskih sistema u realnom vremenu su sljedeće [5]:

1. Prikuplja informacije o tekućem stanju fizičkog sistema, npr. nadgleda fizičke veličine kao što su: temperatura, pritisak, hemijski sastav.
2. Obraduje prikupljene informacije na bazi matematičkog modela fizičkog sistema.
3. Generiše izlazne signale koji utiču na ponašanje fizičkog sistema, a u cilju upravljanja ili ostvarivanja neke zadate performansne mjere.

Upravljački sistem interaguje sa svojim okruženjem na bazi informacija o okruženju prikupljenih posredstvom *senzora*. Od ključne važnosti za ispravan rad računarskih sistema u realnom vremenu je da stanje okruženja, kako ga upravljački sistem "vidi", bude usklađeno (konzistentno) sa stvarnim stanjem okruženja. U suprotnom, efekti akcija, koje upravljački sistem preduzima na bazi pogrešne predstave o tekućem stanju okruženja, mogu imati katastrofalne posljedice. Drugi aspekt vremenske korektnosti sistema u realnom vremenu odnosi se na fizičke posljedice koje mogu u okruženju prouzrokovati nepravovremene akcije upravljačkog sistema. Na primjer, ako računar koji upravlja robotom ne izda na vrijeme komandu za zaustavljanje ili promjenu smjera kretanja, robot se može sudariti sa nekim drugim objektom koji se nalazi u fabričkom pogonu [5].

2.1. Hard, Soft i Firm računarski sistemi u realnom vremenu, odnosno vremenska ograničenja

Pravovremenost je jedini najvažniji aspekt računarskih sistema u realnom vremenu. Ti sistemi odgovaraju na niz vanjskih inputa, koji dolaze u nepredvidljivim stilovima. Kao što je definirao Donald Gillies " *računarski sistem u realnom vremenu je onaj u kojem tačnost računanja ne ovisi samo od logične tačnosti računanja nego i od vremena u kojem se daju rezultati. Ako vremenski zahtjevi nisu ispunjeni, dogodit će se neuspjeh sistema*" [6].

Bitno je da su vremenska ograničenja sistema zasigurno ispunjena. Garancije vremenskog ponašanja zahtijevaju da će sistem biti predvidljiv.

Postoje tri vrste vremenskih ograničenja [6] :

- **Hard** : Zakašnjeni odgovor je neispravan i podrazumijeva neuspjeh sistema. Primjer takvog sistema je praćenje vitalnih funkcija ljudskog tijela medicinskom opremom, gdje će se zakašnjeni odgovor smatrati kao neuspjeh.
- **Soft** : Pravovremenost zahtjeva se definira pomoću prosječnog vremena odgovora. Ako je jednom računanje zakasnilo, to obično nije značajno, ali iako je ponovljeno kašnjenje izračunavanja, može se rezultirati kvar u sistemu. Primjer takvog sistema uključuju zrakoplovni rezervacijski sistemi.
- **Firm** : Ovo je kombinacija *hard* i *soft* pravovremenosti zahtjeva. Izračunavanje se pokazalo kraće od *soft* zahtjeva i duže od *hard* zahtjeva. Na primjer, bolesnički ventilator mora mehanički ventilirati (provjetravati) pacijenta određeni iznos u određenom vremenskom periodu. Nekoliko sekundi zakašnjenja u pokretanju daha je dozvoljeno, ali ne više od toga.

Računarski sistemi u realnom vremenu zahtijevaju da se određeni zadaci izračunaju unutar strogih vremenskih ograničenja. Vremenska ograničenja definiraju krajnji rokovi, odnosno vrijeme odgovora u kojem računanja moraju biti završena. Postoje, kao što smo vidjeli tri različite vrste rokova, *hard* (teški), *soft* (meki) i *firm* (čvrsti). *Hard krajnji rok* znači da ako računanje ne završi prije krajnjeg roka, može uzrokovati neuspjeh ukupnog sistema. *Soft krajnji rok* znači da se nakon krajnjeg roka smanjuje vrijednost izračunavanja, ali ne na nulu ili negativan broj. Ako je *firm krajnji rok* propušten, vrijednost zadatka je izgubljena, ali nije došlo do negativnih posljedica [6].

Prednosti korištenja operativnih sistema u realnom vremenu su [5]:

1. smanjuju cijenu razvoja (skraćuje se vrijeme realizacije korisničke aplikacije);
2. povećava se pouzdanost sistema, jer se smanjuje vjerovatnoća programerske greške prilikom programiranja mehanizama niskog nivoa;
3. olakšana portabilnost (prenosivost) - prenos aplikacije na drugu hardversku platformu (npr. drugi CPU).

3. PRIMJENA RAČUNARSKIH SISTEMA U REALNOM VREMENU

Računarski sistemi u realnom vremenu igraju značajnu ulogu u našem društvu, a oni pokrivaju spektre od onih vrlo jednostavnih do vrlo složenih. Primjeri sadašnjih računarskih sistema u realnom vremenu uključuju upravljanje kućanskim aparatima kao što su perilice rublja i televizori, upravljanje motorima automobila, telekomunikacijskim komutacijskim sistemima, vojna zapovjedanja i upravljanja, upravljanje industrijskim procesima, upravljanje sistemima leta i svemirskim brodom i zračnom avioelektronikom [6].

Sve ovo uključuje prikupljanje podataka iz okoline, obradu prikupljenih podataka, te pružanje pravovremene reakcije. Neki računarski sistemi u realnom vremenu su označeni kao *sigurnosno-kritični sistemi*. U sigurnosno-kritičnim sistemima, neispravnost rada - obično zbog propuštenog krajnjeg roka- rezultira nekom vrstom "katastrofe". Primjeri sigurnosno-kritičnih sistema uključuju odbranbeni sistemi, protublokirajuće kočnice, sistemi upravljanja letom, i zdravstveni ugrađeni sistemi, kao što su pejsmejkri [4].

4. PRIMJER SISTEMA U REALNOM VREMENU ZA UPRAVLJANJE NIVOOM VODE U REZERVOARU

U ovom radu prikazan je sistem u realnom vremenu za upravljanje nivoom vode u rezervoaru u programskom paketu Matlab upotrebom Real-Time Toolbox-a.

Dati sistem upavljanja predstavlja sistem sa zatvorenom petljom, a upravlja se s ventilom koji treba mijenjati stanje u zavisnosti od trenutnog nivoa vode u rezervoaru i brzine dotoka vode kroz ventil. Ovaj sistem je primjer regulacije nivoa tečnosti u zatvorenoj petlji, koji je napravljen pomoću *Fuzzy*

Komunikacija između ova dva sistema će se ostvariti slanjem paketa između modela, odnosno ove dvije real-time aplikacije. Za slanje paketa će se iskoristiti *Packet Input* i *Packet Output*. Sami blokovi nemaju nikakvu funkciju bez uspostavljene konekcije s odgovarajućim uređajima.

Komunikacija se ostvaruje slanjem UDP datagrama kroz mrežu. Zbog toga ove blokove moramo povezati s mrežnim uređajem računara, te moramo instalirati driver koji će *Packet Input* i *Packet Output* blokovima omogućiti da šalju UDP datagrame u mrežu, te da primaju odgovore od drugog sistema u vidu UDP paketa. UDP protokol se nalazi u meniju *Standard Devices*. *Packet Input* blokovi pakuju podatke na aplikacionom nivou mreže, te ih predaju transportnom sloju u vidu UDP protokola. Na drugoj strani, *Packet Output* blokovi preuzimaju podatke od transportnog sloja i odvajaju podatke za dalje korištenje u aplikaciji.

UDP protokol je jednostavan protokol transportnog sloja, koji prihvata podatke od aplikacionog sloja koji se nalazi iznad njega, te ih ubacuje u svoj datagram, dodajući podacima svoje zaglavlje.

UDP protokol omogućava komunikaciju koja ne garantuje pouzdan prijenos podataka preko Interneta. Nije potrebna uspostava komunikacije između pošiljaoca i primaoca i podaci se šalju bez provjere da li je podatak stigao na određenu adresu ili ne. Ovaj način komunikacije zadržava redoslijed slanja podataka, tako da sačuva oblik izlaznog signala, ali podaci koji se izgube ili oštete prilikom prijena su zauvijek izgubljeni. Zbog sačuvanog oblika signala UDP protokol je pogodan za aplikacije koje zahtijevaju sačuvan oblik signala (za real-time upravljanje) s tim da je gubitak podataka veoma mali, tako da izgubljeni podaci ne predstavljaju problem jer se oni nadomještaju sa susjednim podacima.

Za slanje UDP datagrama jedine informacije koje su potrebne su mrežna adresa primaoca ili IP adresa i broj porta na koje željena aplikacija sluša za nove podatke. U (tabeli 1) se nalazi prikaz podataka potrebnih za postavke *Packet Input* i *Packet Output* blokova na strani klijenta i servera, odnosno prikazane su postavke komunikacijskog sistema. Potrebni podaci su adresa primaoca, port na koji se šalje, ali i port na kojem će slušati za povratne podatke od uređaja. Nakon unošenja parametara, potrebno je testirati konekciju. Kao prvo testira se da li je ikako uspostavljena konekcija s uređajem, a nakon toga da li je uređaj pravilno instaliran i da li ikako može operirati na željeni način.

Tabela 1: Postavke komunikacijskog sistema

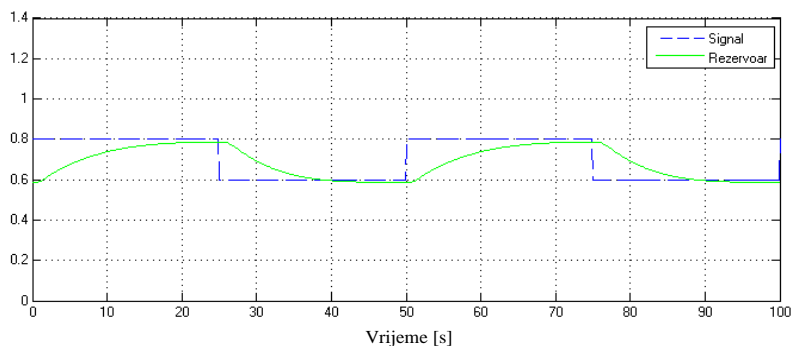
Klijent: 192.168.3.121		Server: 192.168.3.120	
<i>Packet Input</i>	<i>Packet Output</i>	<i>Packet Input</i>	<i>Packet Output</i>
Adresa servera: 192.168.3.120	Adresa servera: 192.168.3.120	Adresa klijenta: 192.168.3.121	Adresa klijenta: 192.168.3.121
Lokalni port: 9003	Lokalni port: 9001	Lokalni port: 9000	Lokalni port: 9002
Port pošiljaoca: 9000	Port pošiljaoca: 9002	Port pošiljaoca: 9003	Port pošiljaoca: 9001

Ova dva modela, koja će pomoću RTW biti pretvorena u real-time aplikacije, imaju po 2 bloka za mrežnu komunikaciju. Klijent ima *Packet Output* blok koji šalje podatke od senzora ka regulatoru, te jedan *Packet Input* blok koji prima podatke od regulatora te ih predaje aktuatorima sistema. Server posjeduje također po jedan od ova dva tipa blokova: *Packet Input* prima podatke od senzora te ih predaje regulatoru koji po nekoj logici izračunava pravila koja se pomoću *Packet Output* blokova vraćaju do realnog sistema i aktuatora.

Vrijeme uzorkovanja modela mora odgovarati vremenima uzorkovanja blokova RTW. *Packet Input* i *Packet Output* moraju imati vremena uzorkovanja koji su multiplicirane, odnosno iste vrijednosti. Za predstavljanje upravljačke varijable kao i nivoa vode u rezervoaru dovoljna je jedna *double* varijabla koja se nalazi u paketu veličine 8 bajtova.

Za razmjenu podataka, kao što je gore navedeno se koristi UDP protokol. Potrebno je podesiti adrese i portove sa kojih će se slati i primiti podaci, kako je i prikazano u tabeli 1. Za pokretanje modela i prebacivanje u RTW aplikaciju potreban je *fixed step solver*, dakle mora biti tačno definisan vremenski korak simulacije, zatim je potrebno odabrati *RT Windows Target* koji se definiše sa

rtwin.tlc fajlom u konfiguracijskom prozoru modela, odnosno kao TLC file, te klikom na *Build*, model će se pretvoriti u real-time C aplikaciju. Real-time aplikacija se pokreće pomoću MATLAB Simulinka, odnosno pomoću pripadajućeg modela. Nakon pokretanja aplikacije, možemo pratiti rad aplikacije u realnom vremenu, na nekim od izlaznih blokova sistema, gdje su najpogodniji *Scope* blokovi (slika 3).



Slika 3: Upravljački signal (stanje ventila) i nivo vode u rezervoaru

Željeni nivo vode u rezervoaru (referentna vrijednost) je podešeno na 0.7. Radi se o sistemu za upravljanje u zatvorenoj petlji. Sam klijent i server se nalaze na 2 različita računara te su povezani u lokalnu mrežu. Uz takve postavke, kašnjenje signala je praktički jednako nuli. Za upravljanje je iskorišten fuzzy regulator koji prima dvije ulazne varijable: nivo vode u rezervoaru te brzinu promjene nivoa vode (prvi izvod). Ulazne i izlazne varijable su povezani setom pravila za ponašanje fuzzy kontrolera. Iz rezultata simulacije se vidi da se sistem nakon određenog vremena stabilizira te radi u skladu s referentnim signalom, odnosno nivo vode u rezervoaru se stabilizira na vrijednost 0.7.

5. ZAKLJUČAK

Svakim danom računarski sistemi u realnom vremenu se sve više primjenjuju, tako da bi danas bilo teško zamisliti jednu djelatnost u kojoj se ne koriste računarski sistemi u realnom vremenu. Kako izgradnja, odnosno implementacija, tako i primjena računarskih sistema u realnom vremenu danas sve više napreduje, tako da područja primjene računarskih sistema u realnom vremenu svakim danom postaju sve veća, tako da ovi sistemi igraju veoma značajnu ulogu u našem društvu i pokrivaju područja od onih jednostavnih, do onih vrlo složenih. Primjena računarskih sistema u realnom vremenu uključuje prikupljanje podataka iz okoline, obradu prikupljenih podataka, te pružanje pravovremene reakcije. Sama primjena računarskih sistema u realnom vremenu obuhvata veoma široka područja.

Preporuka za daljnja istraživanja je realizacija sistema u realnom vremenu za upravljanje nivoom vode u rezervoaru sa upotrebom mikrokontrolera.

6. LITERATURA

- [1] John Ford, *A Real Time Operating System Survey*, NRAO, 1999.
- [2] <http://es.elfak.ni.ac.rs/rts/Materijal/RTS.pdf> (pregledano 06.05.2013. godine)
- [3] Dr. Božo Kristajić, *Operativni sistemi*, Podgorica, 2005.
- [4] Silberschatz, Galvin, Gagne, *Operating System Concepts*, Seventh Edition, John Wiley&Sons.Inc, 2005.
- [5] <http://es.elfak.ni.ac.rs/mps/materijal/3-RTS.pdf> (pregledano 12.05.2013 godine)
- [6] <http://sharegang.com/engineering-seminars-download/15607-rtos-real-time-operating-systems.html> (pregledano 11.06.2013. godine)