

RAZVOJ INTERAKTIVNIH SIMULACIJA POMOĆU EASY JAVA SIMULATIONS I SIMULINK MODELA KAO EKSTERNE APLIKACIJE

INTERACTIVE SIMULATIONS DEVELOPMENT USING EASY JAVA SIMULATIONS AND SIMULINK MODEL AS AN EXTERNAL APPLICATION

Nataša Popović, *Elektrotehnički fakultet, Istočno Sarajevo*
Milica Naumović, *Elektronski fakultet, Niš*

Sadržaj - U radu je opisan alat za kreiranje interaktivnih simulacija Easy Java Simulations. Simulacije se u oblasti upravljanja obično realizuju pomoću Matlab Simulink-a. Međutim, Simulink nema mogućnost interakcije korisnika sa simulacijom. Ovaj problem moguće je prevazići razvojem simulacija u Easy Java Simulations okruženju koje koristi Matlab Simulink kao eksternu aplikaciju. Razvoj simulacija je prikazan na konkretnom primjeru kreiranja simulacije kretanja matematičkog klatna sa trenjem.

Abstract - In this paper we described a software tool for creating interactive simulations - Easy Java Simulations. In control engineering, simulations are usually realized in Matlab Simulink modeling tool. However, Simulink doesn't offer interaction between users and simulation. This problem may be overcome by developing simulation in Easy Java Simulations environment, which uses Matlab Simulink as an external application. Simulation development is shown on the concrete example of creating the simulation of pendulum motion with friction.

1. UVOD

Interaktivne računarske simulacije igraju važnu ulogu u obrazovanju studenata, pogotovo u inženjerskim disciplinama. Računari se u savremenoj nastavi koriste u svrhu kreiranja i prezentovanja atraktivnih nastavnih sadržaja koji studentima pomažu da lakše shvate materiju koju uče. Međutim, računarske simulacije se, kada su nastavnici u pitanju, ne koriste često koliko bi mogle zato što nastavnici uglavnom nemaju interesovanja za njihovo kreiranje iz dva razloga. Prvi razlog je što za taj proces treba odvojiti poprilično vremena, a drugi razlog je što simulacije nekada u potpunosti ne zadovoljavaju njihove obrazovne ciljeve.

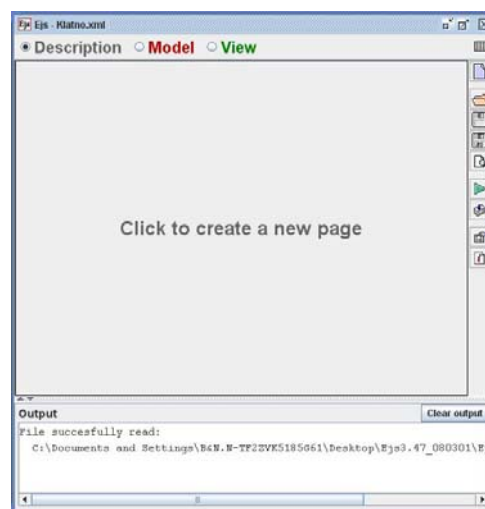
Jedno rješenje za prevazilaženje ovakvih problema jeste da se nastavnicima pomogne da kreiraju svoje sopstvene simulacije, zavisno od specifičnosti fenomena koje žele da izučavaju i prezentuju studentima, a nezavisno od postojećih softverskih paketa koji su namijenjeni za kreiranje simulacija. Drugo rješenje jeste da studenti sami kreiraju svoje simulacije, čime se uključuju u proces tzv. konstruktivnog modelovanja. Polazna tačka i za nastavnike i za studente pri kreiranju sopstvenih simulacija jeste potpuno razumijevanje fenomena koji će se simulirati kako bi se na što kvalitetniji način moglo prikazati njihovo ponašanje u računarskoj formi.

Sa stanovišta obrazovanja u oblasti automatskog upravljanja, softverski paket koji se najčešće koristi za kreiranje simulacija jeste Mathworks-ov Matlab, ali nedostatak interaktivnosti je danas često prepreka za kvalitetnije i brže izvođenje simulacija. Interaktivnost se u Matlab-ove simulacije može dodati ako se one kreiraju i koriste zajedno sa Easy Java Simulations, *open source* softverskim alatom baziranim na Java-i [1].

2. EASY JAVA SIMULATIONS

Easy Java Simulations (dalje u tekstu Ejs) je softverski paket koji omogućava kreiranje simulacija fizičkih i tehničkih fenomena i sistema u Java-i na brz i lak način, pa je pogodan za korišćenje u inženjerskom obrazovanju [2]. Namijenjen je

nastavnicima, istraživačima i studentima koji imaju osnovno znanje iz programiranja, a ne mogu sebi priuštiti veliki utrošak vremena za kreiranje kompletnog sofisticiranog interaktivnog grafičkog korisničkog interfejsa. Kako je Ejs baziran na Java-i, to je moguće simulacije kreirati na njemu koristiti kao nezavisne programe koji rade pod različitim operativnim sistemima, distribuisati ih preko Interneta i pokretati ih kao *html* stranice iz većine popularnih web čitača.



Sl. 1: Korisnički interfejs Easy Java Simulations

Aplikacije kreirane pomoću Ejs mogu biti samostalne Java aplikacije ili apleti. Izvorni fajlovi Ejs aplikacija se čuvaju u *.xml* formatu. Grafički korisnički interfejs koji nudi Ejs je strukturiran tako da se sastoji od tri dijela: opisa, modela i vizualizacije, slika 1. Opis je dio u kojem se definiše fenomen ili sistem koji se simulira, odnosno u njemu se nalazi teoretski dio neophodan za razumijevanje ponašanja fenomena ili sistema. Model je dio koji se odnosi na stranice sa Java kodom, diferencijalnim jednačinama ili na povezivanje sa eksternim aplikacijama (npr. Matlab, Sysquake). Vizualizacija se odnosi na vizuelni prikaz aplikacije kao i na elemente korisničkog interfejsa potrebnog

za interakciju. Model i vizualizacija su povezani tako da se bilo kakva promjena nastala u modelu automatski odražava na vizuelni prikaz i obratno.

3. MATLAB I EASY JAVA SIMULATIONS

Matlab je dobro poznat softverki paket koji se koristi u raznim oblastima inženjerstva. Mnoštvom svojih alata omogućava jednostavnu analizu, sintezu, modelovanje i simuliranje sistema [3]. Za kreiranje simulacija u Matlab-u koristi se Simulink, alatka namijenjena za modelovanje dinamičkih sistema u grafičkom okruženju. Njegovi simulacioni modeli su prikazani u obliku blok-dijagrama koji se sastoji od pojedinačnih blokova (integrator, diferencijator, sabirač, množač, pojačanje, osciloskop...). Parametri sistema se obično definišu u samim blokovima prije početka simulacije.

Nedostatak Simulink-ovih simulacija je nemogućnost podešavanja parametara tokom izvođenja simulacije. Naime, da bi se vrijednost nekog parametra ili promjenljive u Simulink modelu promijenila, neophodno je prvo zaustaviti simulaciju, promijeniti im vrijednost, sačuvati promjenu i ponovo pokrenuti simulaciju. Ovo ne predstavlja veliki problem u slučaju jednostavnih modela sa malim brojem parametara i promjenljivih, ali za složenije modele predstavlja svojevrsnu prepreku. Jasno je da Simulink nema visok stepen interaktivnosti sa korisnikom, pa je sa tog stanovišta neophodno dodati mu interaktivnost vještačkim putem.

Easy Java Simulations, verzija 3.4, je razvijena sa mogućnošću povezivanja sa eksternim aplikacijama i to Matlab-om i Sysquake-om [4], [5]. Prvobitno je razvijena za Matlab 6.0, ali bez problema radi i sa novim verzijama Matlab-a. Time je omogućeno da se Ejs koristi zajedno sa Simulink-om kako bi se kreirale potpuno nove simulacije bazirane na postojećim Simulink modelima ili onima koje autor tek treba da osmisli. To znači da autor može kreirati simulaciju u Ejs okruženju koja kontroliše Simulink model i prikazuje vrijednosti parametara i promjenljivih modela, a modifikaciju parametara i promjenljivih može obavljati direktno iz simulacije kreirane u Ejs. Osim toga, autor može pozvati bilo koju Matlab funkciju (bilo ugrađenu ili definisanu u .m fajlu) sa bilo kog mjesta u svom Ejs modelu. Jedini zahtjev koji korisnik treba zadovoljiti jeste da su i Ejs i Matlab instalirani na istom računaru. Za slučaj da su Matlab i Ejs aplikacija na različitim računarima, neophodno je koristiti Jim server koji omogućava da Ejs koristi udaljeni Matlab serever. Ovo se naročito koristi u laboratorijama sa daljinskim pristupom zasnovanim na Ejs i Matlab-u, što nije tema ovog rada.

Konekcija Ejs sa Matlab-om se ostvaruje tako što se promjenljive i parametri modela kreiranog u Ejs povežu sa odgovarajućim promjenljivim i parametrima u Simulink modelu.

4. SIMULACIJA KRETANJA MATEMATIČKOG KLATNA

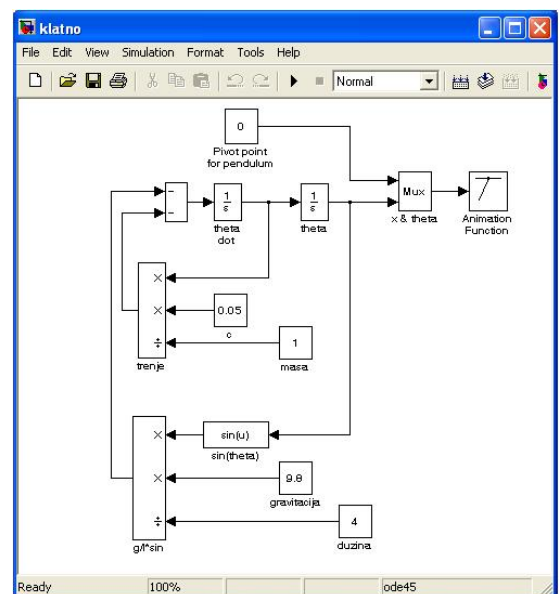
Na primjeru kretanja matematičkog klatna sa trenjem biće opisan postupak kreiranja simulacije u Ejs koja koristi model

klatna razvijen u Simulink-u. Prije nego se stupi u kreiranje modela bilo kog sistema ili fenomena, neophodno ga je u potpunosti razumjeti. To je važno zato što je jedino tada moguće definisati prave parametre i promjenljive pomoću kojih se kasnije upravlja izvršavanjem simulacije.

4.1 Razvijanje modela u Simulink-u

Na osnovu matematičkog modela izvedenog za kretanje klatna sa trenjem, realizovan je Simulink-ov blok-dijagram prikazan na slici 2. Kretanje klatna dužine l i mase m koje pod dejstvom gravitacione sile osciluje oko ravnotežnog položaja sa koeficijentom trenja c dato je diferencijalnom jednačinom oblika

$$\ddot{\theta} = -\frac{c}{m} \dot{\theta} - \frac{g}{l} \sin \theta. \quad (1)$$



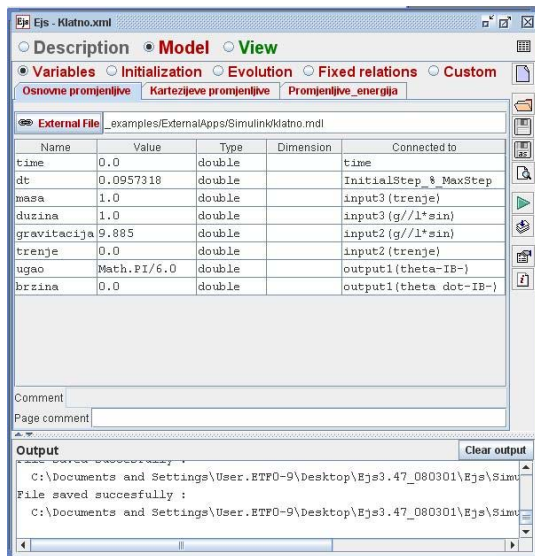
Sl. 2: Blok-dijagram kreiran u Simulink-u

Blok-dijagram je kreiran tako što se vodilo računa o tome da se promjenljive i parametri organizuju na način koji omogućava njihovo najjednostavnije povezivanje sa modelom u Ejs.

4.2 Razvijanje modela u Ejs

Prvi korak u kreiranju modela u Ejs je opis sistema u obliku promjenljivih. Dio Model Ejs interfejsa se sastoji od pet stranica: *Variables*, *Initialization*, *Evolution*, *Fixed relations* i *Custom*. Definisane promjenljivih se vrši njihovim navođenjem u tabelama promjenljivih koje se kreiraju u okviru stranice *Variables*, slika 3. Ova stranica za naš primjer ima tri stranice: osnovne promjenljive (one koje opisuju matematički model), Kartezijeve promjenljive (definisane Kartezijeve koordinate za položaj klatna i vektor brzine) i promjenljive za računanje energija (potencijalne, kinetičke i ukupne). Pošto se radi o aplikaciji koja treba da koristi Simulink model, to je tabela osnovnih promjenljivih smještena na posebnoj stranici koja sadrži polje u kojem je naznačeno gdje se nalazi tzv. eksterni fajl (*klatno.mdl*) koji će

se koristiti pri simulaciji. Odabiranjem ovakvog fajla automatski se naređuje Ejs da počne raditi sa Simulinkom.



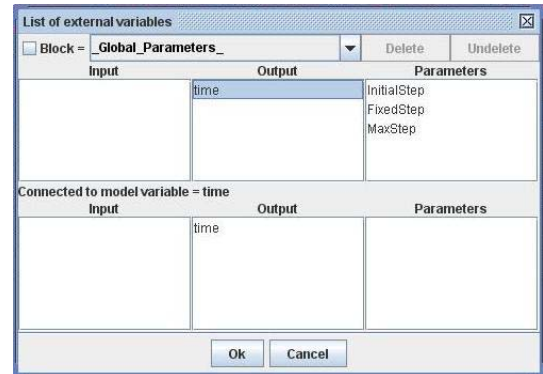
Sl. 3: Definisane promjenljivih u Ejs

Nakon definisanja promjenljivih, neophodno je izvršiti njihovo povezivanje sa promjenljivima iz Simulink modela. Za to služi kolona *Connected to* u tabeli promjenljivih. Desnim klikom miša na promjenljivu otvara se lista eksternih promjenljivih u kojoj odabiramo onu koja u Simulink modelu odgovara onoj u Ejs, slika 4. Kategorija iz koje biramo promjenljive zavisi od toga šta želimo sa promjenljivom da radimo u toku simulacije. Tako se ulazne promjenljive Simulink modela (*Input*) mogu slobodno mijenjati iz Ejs, izlazne promjenljive (*Output*) se mogu čitati ali ne i mijenjati iz Ejs, dok se parametri (*Parameters*) mogu mijenjati iz Ejs ali će Simulink uzeti u obzir nove vrijednosti samo ako model nije pokrenut.

Promjenljiva *dt* je povezana sa dva parametra u Simulink modelu, *InitialStep* i *MaxStep*, dok su promjenljive *ugao* i *brzina* (ugaona) povezane sa promjenljivim u blokovima koji imaju sufiks *-IB-*. Ovaj sufiks označava da se radi o posebnim promjenljivim. Naime, to su izlazne promjenljive Simulink blokova koji se zovu integrator i koji imaju određene početne uslove. Iako se radi o izlaznim promjenljivim, što znači da se ne bi smjele mijenjati iz Ejs, nekada je to potrebno učiniti da bi se promijenili početni uslovi sistema. Na primjer, za naš model to bi značilo postavljanje novih vrijednosti za ugaon ili ugaonu brzinu klatna. Mijenjanje početnih uslova diferencijalne jednačine u Simulink-u je moguće na način da se resetuju integratori a zatim ponovo pokrenu. Ejs ima poseban Java metod `_external.resetIC()` za mijenjanje početnih uslova, a poziva se svaki put kada želimo da Simulink prihvati bilo koju promjenu vezanu za promjenljive ugaon i brzinu u Ejs.

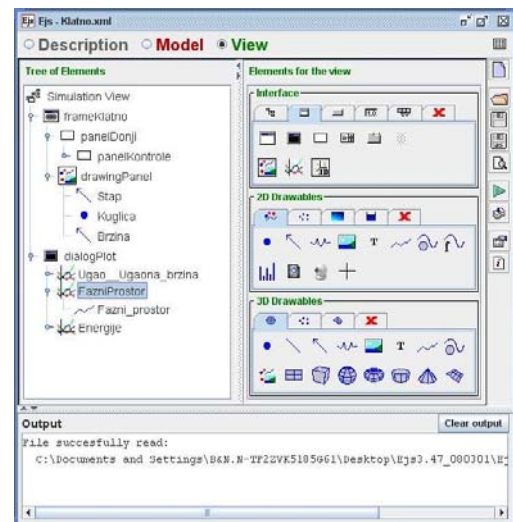
Nakon povezivanja, Ejs može preuzeti kontrolu nad pokretanjem Simulink modela koristeći *utility* metod `_external.step(int times)`. Pozivanje ovog metoda ima sljedeći efekat. Prvo, uzima vrijednosti promjenljivih u Ejs modelu koje su povezane sa ulaznim promjenljivim Simulink modela. Takođe uzima vrijednosti promjenljivih u Ejs modelu koje su povezane sa izlaznim promjenljivim koje se odnose samo na

integratore. Zatim pokreće Simulink model sa onoliko koraka koliki je broj naveden u zagradi. Konačno, uzima vrijednosti svih izlaznih promjenljivih Simulink modela koje su povezane sa odgovarajućim promjenljivim u Ejs.



Sl. 4: Lista eksternih promjenljivih

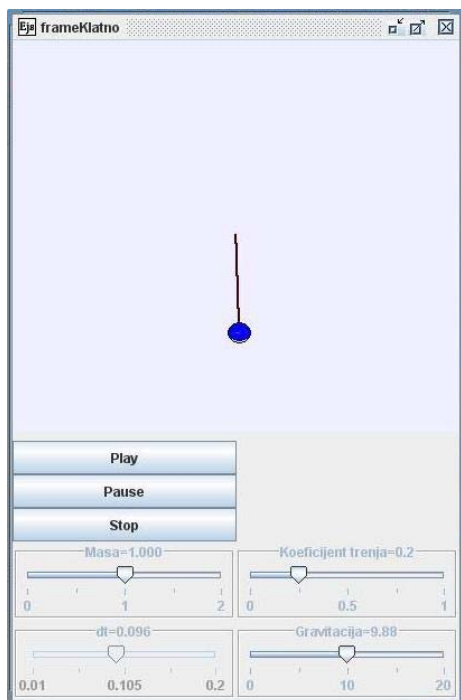
U stranicu *Evolution* Ejs modela se upisuju diferencijalne jednačine koje opisuju model, dok se u stranicu *Fixed relations* unose određena ograničenja. Za primjer klatna ograničenja su vezana za Kartezijeve promjenljive i energiju, odnosno tu su upisani obrasci kako se računaju položaj klatna, vektor brzine i energije. Stranica *Custom* služi za opis Java metoda koje autor simulacije može da definiše u slučaju da su mu neophodni a ne nalaze se u listi ugrađenih metoda.



Sl. 5: Stablo elemenata za vizuelizaciju

Posljednji korak u razvoju Ejs simulacije je kreiranje vizuelnog prikaza. Na slici 5 dat je pregled elemenata za vizuelizaciju u obliku stabla. Elementi se mogu odabrati iz skupa ponuđenih, ugrađenih elemenata, koji su svrstani u određene tipove. Svaki tip se odnosi na specifične zahtjeve vizuelnog prikaza i interakcije koji su različiti od simulacije do simulacije, a mogu se prilagođavati mijenjanjem internih vrijednosti koje modifikuju aspekt i ponašanje elementa na ekranu. Slika 6 prikazuje konačan izgled okvira u kojem se kontroliše simulacija, a koji je podijeljen na dva dijela. U gornjem dijelu se nalazi animacija kretanja klatna, a u donjem su kontrole pomoću kojih se kontroliše simulacija (*Play, Pause, Stop*) i parametri sistema koji se mogu mijenjati

tokom simulacije (masa, koeficijent trenja, gravitacija). Treba naglasiti da parametar dt nije moguće mijenjati u toku izvođenja simulacije, već prije njenog početka, tako da je klizač dt neaktivan dok je simulacija u toku.



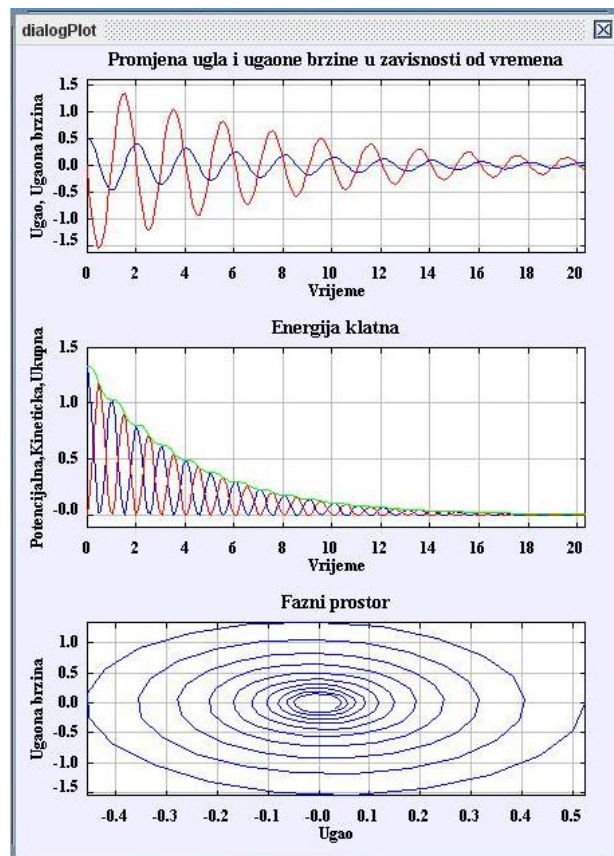
Sl. 6: Stablo elemenata za vizuelizaciju

4.3 Pokretanje i izvođenje simulacije

Pokretanje simulacije se vrši tako što se u Ejs interfejsu aktivira ikona *Run* (▶). Simulacija se generiše, kompajlira, pakuje u *.jar* fajl i izvršava kao nezavisan proces. Ejs generiše simulaciju u obliku apleta, tako da je kontrola toka simulacije moguća i iz web čitača. Pri pokretanju simulacije otvara se Matlab Simulink, učitava se originalni model i neznatno modifikuje kako bi se omogućile interne veze u Ejs neophodne za komunikaciju sa modelom preko Matlab-ovog *workspace*-a. Komunikacija je ostvarena preko *.dll* biblioteke kreirane u C i nekih *utility* Java klasa koje autor kreira pri razvoju svojih simulacija. Vizuelni prikaz simulacije se pojavljuje na ekranu čime je omogućeno interaktivno kontrolisanje toka simulacije. Kontrolno dugme *Play* služi za startovanje simulacije nakon njenog pokretanja, *Pause* služi za njeno pauziranje, a *Stop* za resetovanje.

Tokom simulacije generišu se određeni grafikoni, koji se definišu prilikom kreiranja vizuelnog prikaza, slika 7. Od značaja za naš primjer su sljedeći grafikoni:

- Promjena ugla i ugaone brzine u zavisnosti od vremena, pri čemu je promjena ugla prikazana plavom a promjena ugaone brzine crvenom bojom,
- Energija klatna koja se odnosi na potencijalnu (plava boja), kinetičku (crvena) i ukupnu (zeleno),
- Fazni prostor, odnosno zavisnost ugaone brzine od promjene ugla.



Sl. 7: Grafikoni generisani tokom simulacije

5. ZAKLJUČAK

Easy Java Simulations predstavlja moćno sredstvo za kreiranje interaktivnih simulacija naročito za osobe koje imaju osnovno znanje iz programiranja. Povezivanje Matlab Simulink-a sa Ejs omogućilo je korišćenje Ejs u automatskom upravljanju budući da je Simulink *de facto* alatka bez koje se u upravljanju ne može. Dodavanjem interaktivnosti u Simulink modele pomoću Ejs otvaraju se nove mogućnosti za lakše upravljanje simulacijom, a samim tim i studentima je omogućeno lakše sagledavanje fenomena ili sistema koji se simulira. Sljedeći korak je razvoj daljinskog upravljanja realnim, fizičkim modelom koji je potrebno realizovati pomoću Matlab Simulink-a i Ejs.

LITERATURA

- [1] Easy Java Simulations: <http://fem.um.es>, decembar 2009.
- [2] Francisco Esquembre, *Easy Java Simulations- Manual for version 3.4*, September 2005
- [3] www.mathworks.com, decembar 2009.
- [4] Francisco Esquembre, José Sánchez-Moreno, *Easy Java Simulations-Using Ejs to run Simulink models in an interactive way for version 3.4*, January 2005
- [5] Sebastian Dormido, Francisco Esquembre, Gonzalo Farias, Jose Sanchez, *Adding interactivity to existing Simulink models using Easy Java Simulations*, Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005, pp. 4163-4168, Seville, Spain, December 2005