

**МАШИНСКИ ФАКУЛТЕТ  
УНИВЕРЗИТЕТА У КРАГУЈЕВЦУ**



**Идентификација и управљање мотором  
једносмерне струје путем интернета**

Ментор:  
*проф.др Милан Матијевић*

Студент:  
*Ненад Бабајић 1045/2008*

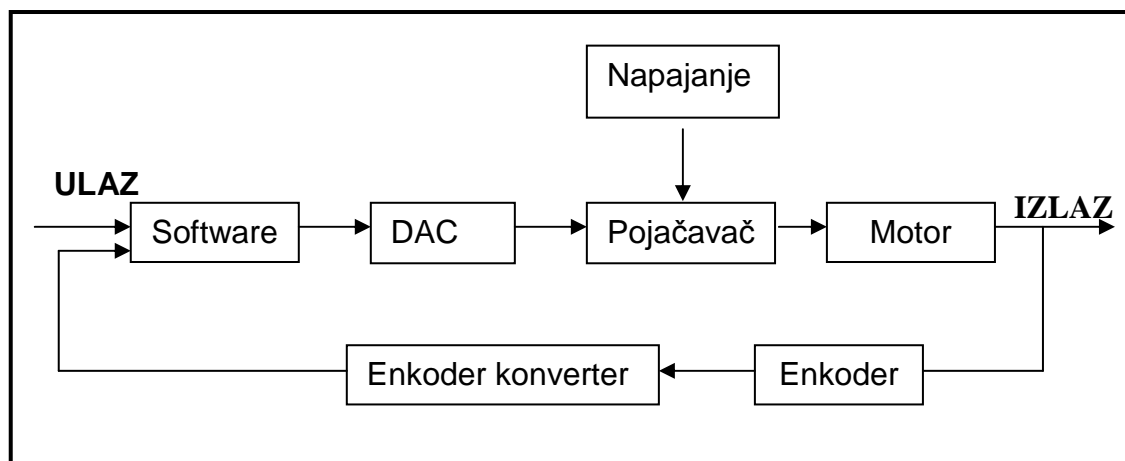
Крагујевац, 2009.

## Садржај

1. Математички модел мотора једносмерне струје (MJC) управљаног струјом ротора .....	3
2. Идентификација процеса .....	8
2.1 Дефинисање експеримента .....	8
2.2 Практична идентификација система .....	9
2.2.1 Одскачна ( <i>step</i> ) функција .....	10
2.2.2 Синусни сигнал са променљивом фреквенцијом .....	10
2.2.3 Низ правоугаоних сигнала константне амплитуде .....	11
2.2.4 Низ правоугаоних сигнала случајне амплитуде .....	11
2.2.5 Комбиновани сигнал који се састоји од одскачне, синусне и правоугаоне функције .....	12
2.3 Мерење и обрада података .....	12
2.4 Упоредна анализа резултата добијених у simulink-у .....	14
2.4.1 Одскачна функција .....	14
2.4.2 Синусни сигнал са променљивом фреквенцијом .....	14
2.4.3 Низ правоугаоних сигнала константе амплитуде .....	15
2.4.4. Закључак .....	15
3. Дискретни PID регулатор .....	16
Пропорционални члан .....	16
Интегрални члан .....	16
Деривациони члан .....	16
Дискретни PID контролер .....	17
4. Управљање системом преко интернета –WebLAB .....	19
MatLAB као део weblab-а .....	20
5. Практична реализација експерименталног модела .....	23
5.1 Управљање једносмерним моторима .....	23
5.2 Појачавачи .....	25
5.3 Енкодер .....	26
5.4 Реализација енкодера на систему Серво 300 .....	28
6. Серво 300 .....	29
Закључак .....	36
Додатак А - Подршка за NI6008 у MATLAB –у .....	37
Додатак Б - Карактеристике мотора ESCAP 28D2R .....	38

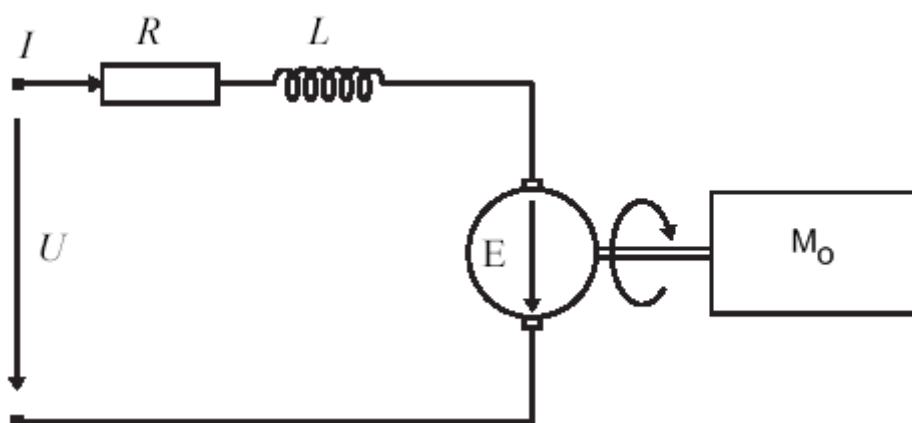
## 1. Математички модел мотора једносмерне струје (MJC) управљаног струјом ротора

Типични систем за контролу серво мотора је приказан на слици 1., где излази, односно посматране величине могу бити угаона брзина  $\omega$  или угао  $\theta$  (позиција).



Слика 1 Блок дијаграм система за управљање DC серво моторима.

Принципијелна шема MJC се може приказати следећом сликом:



Очигледно је да се она састоји из две компоненте – електричне и механичке.

Електрична компонента је описана једначином

$$u(t) = Ri(t) + L \frac{di(t)}{dt} + E \quad (1.1)$$

која представља Кирхово правило за круг кроз који тече струја  $i$ .

Где је :

$u$  – напон напајања мотора [V]

$R$  – омски отпор намотаја ротора [ $\Omega$ ]

$i$  – струја која тече кроз намотаје ротора [A]

$L$  – индуктивитет намотаја ротора [H]

$E$  – индукована (против)електромоторна сила [V]

Индукована електромоторна сила  $E$  је сразмерна магнетском току (флукс) намотаја  $\Phi$  и угадној брзини  $\omega$ :

$$E = K_E \Phi \omega \quad (1.2)$$

Како је магнетски ток константан код мотора са сталним магнетом може се писати да је  $E = K_E \omega$ , односно

$$E = K_E \omega(t) = K_E \frac{d\Theta}{dt} \quad (1.3)$$

где је:

$\omega$  - угадна брзина ротора  $\frac{d\Theta}{dt}$  [rad/s]

$K_E$  - напонска константа [Vs] и зависи од конструкције мотора и обично се узима да је:

$$K_E = \frac{pz}{2\pi a} \quad (1.3a)$$

$p$  – број пари полова

$z$  – број проводника арматурног намотаја

$a$  - број паралелних грана арматурног намотаја

Ради једноставности понекад се усваја да је

$$K_E \approx p \quad (1.3b)$$

Механичка компонента, односно закретни момент, је сразмеран струји која протиче кроз ротор:

$$M_m(t) = K_m I \quad (1.4)$$

Користећи принцип да је сума свих сила које делују на тело једнака сили инерције, такозвана динамичка равнотежа момената (Д'Алемберов принцип, односно други Њутнов закон) добијамо:

$$M_m(t) = M_t(t) + M_j(t) + B\omega \quad (1.5)$$

При томе је:

$M_m$  – момент сведен на осовину ротора [ $\text{kgms}^{-2}$ ]

$M_t$  - статички момент (момент отпора, момент терета) [ $\text{kgms}^{-2}$ ]

$M_j$  - динамички момент одређен укупном моментом инерције  $J$  на осовини мотора [ $\text{kgms}^{-2}$ ]

$B$  - коефицијент вискозног трења [ $\text{Nms/rad}$ ]

Коефицијент вискозног трења је изузетно мали па се може занемарити, тако да можемо писати:

$$M_m(t) = K_m i(t) + J \frac{d\omega(t)}{dt} \quad (1.6)$$

$J$  - момент инерције ротора [ $\text{kgm}^2$ ]

$K_m$  – механичка константа мотора

Једначине (1.1) и (1.6) описују **простор стања** (*state space*) математичког модела једносмерног мотора са сталним магнетима.

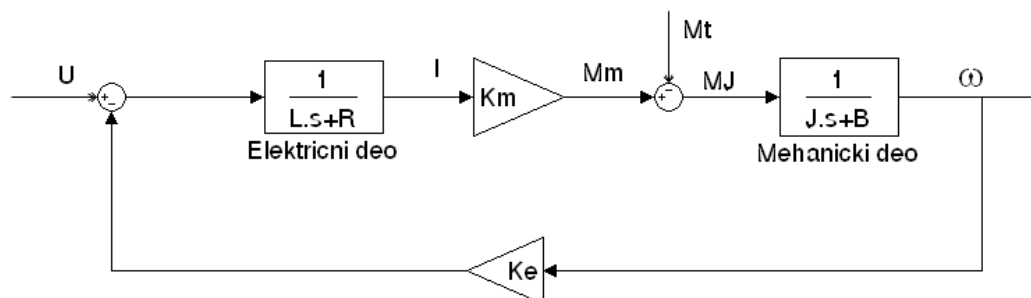
Пошто се посматра SISO (*single input – single output*) систем узимамо да је напон  $u(t)$  улазна величина а угаона брзина  $\omega(t)$  излазна величина.

Преласком на Лапласову варијаблу  $s$  једначине 1.1 и 1.6 постају

$$U(s) = RI(s) + LsI(s) + K_e \Omega(s) \quad (1.6)$$

$$M_m(s) = M_t(s) + Js(s) = K_m I(s) \quad (1.7)$$

Према томе, блок шема једносмерног мотора има следећи изглед:



На основу претходних једначина, промена брзине ротације  $\Omega(s)$ , када се мења напон арматуре  $U(s)$  и момент мотора  $M_t$ , описана је једначином:

$$\Omega(s) = \frac{K_M}{T_a T_m s^2 + T_m s + 1} U(s) - \frac{R}{K_E K_m} \frac{1 + T_a s}{T_a T_m s^2 + T_m s + 1} M_t(s) \quad (1.8)$$

$T_a = \frac{L}{R}$  - електрична (арматурна) временска константа

$T_m = \frac{JL}{K_e K_m}$  - електрична (арматурна) временска константа

$K_M = \frac{1}{K_e}$  - статички коефицијент појачања мотора

Уз почетни услов  $M_t=0$ , добија се преносна функција по улазној величини:

$$G(s) = \frac{\Omega(s)}{U(s)} = \frac{K_M}{T_a T_m s^2 + T_m s + 1} \quad (1.9)$$

Решења карактеристичне једначине преносне функције су:

$$s_{1,2} = -\frac{1}{2T_a} \left( 1 \pm \sqrt{1 - 4 \frac{T_a}{T_m}} \right) \quad (1.10)$$

Очигледно је ће решења једначине бити реална уколико је  $T_m \geq 4T_a$ . У том случају преносна функција има апериодичан карактер. У супротном, преносна функција има осцилаторни карактер, што је свакако непожељно.

Уколико знамо конструктивне карактеристике мотора, преносна функција се може добити и помоћу једначине 1.8:

$$G(s) = \frac{\Omega(s)}{U(s)} = \frac{\frac{k}{LJ}}{s^2 + \frac{R}{L}s + \frac{k^2}{LJ}} \quad (1.11)$$

Обзиром да је  $T_m$  најчешће много веће од  $T_a$  (обично један реда величине, тј. бар 10 пута),  $T_a$  се може занемарити па се преносна функција може описати као функција 1. реда.

$$G(s) = \frac{\Omega(s)}{U(s)} = \frac{K_M}{T_m s + 1} \quad (1.12)$$

## **2. Идентификација процеса**

Идентификација процеса представља дефинисање математичког модела на основу података о понашању система који су добијени експерименталним путем. Да би се идентификација успешно обавила, потребно је извршити низ мерења користећи одговарајуће улазне податке. Анализом улазних података, као и одзивом система на њих, може се добити преносна функција која математички описује понашање система.

### **2.1 Дефинисање експеримента**

Да би идентификација била успешна потребно је испоштовати основне критеријуме:

1. Експеримент мора показати сва стања система – Процес идентификације није успешан уколико измерени подаци не показују све особине система.
2. Морају се обрадити и прелазне промене, као и стационарна стања. Прелазне промене показују динамичко понашање система а стационарна стања омогућавају одређивање општих карактеристика система.
3. Експеримент не сме оштетити опрему нити прећи у нелинеаран режим рада – Идентификација се, најчешће, мора одрадити у „радном“ окружењу система. Код мањих, наменски урађених поставки (setup), као што су лабораторијске вежбе, ради се о релативно малим силама и контролисаним условима, тако да је могућност оштећења опреме практично искључена. Међутим, код комерцијалних система свако веће одступање од нормалног режима рада може изазвати квар опреме. То се нарочито односи на скоковите и импулсне промене режима рада.

Важно је напоменути да се идентификација система врши за развој линеарних модела и због тога треба бирати такав улазни сигнал који омогућава да понашање система остане линеарно. То значи да амплитуда и фреквенција улазног сигнала морају бити такви да систем може дати линеаран одзив. Међутим, треба обратити пажњу на то да амплитуда мора бити и довољно велика да се утицаји шума и квантизације сведу на најмању могућу меру.

У зависности од врсте модела, односно система, који се посматра бирају се и испитни сигнали, то јест функције. То су најчешће одскочна (*step*) функција, хармонијска, трапезоидна, квадратна.



## **2.2 Практична идентификација система**

Обзиром да је поставка рађена са намером да се прикажу сви аспекти управљања једносмерним мотором, тенденција је била да се прикаже одзив на што више испитних функција да би се направио што веродостојнији модел.

Да би проверили веродостојност идентификације систем је побуђиван са 5 испитних сигнала

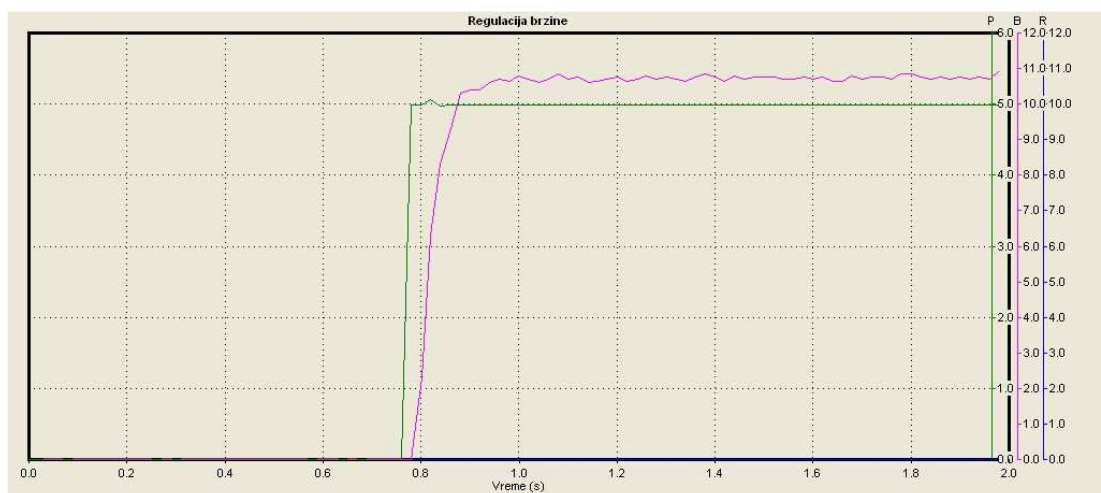
1. Одскачна (*step*) функција
2. Низ правоугаоних сигнала са константном амплитудом
3. Низ правоугаоних сигнала са случајном амплитудом
4. Синусни сигнал са променљивом фреквенцијом
5. Комбиновани сигнал који се састоји од одскачне, синусне и правоугаоне функције

Три (1,2 и 4) од ових 5 функција имају своје еквиваленте у MatLAB-у, тј. Simulink-у, тако да је врло лако извршити симулацију понашања система и упоредити са реалним одзивом.

На дијаграмима зелени крива означава напон напајања мотора, црвена угаону брзину, а плава жељену (референтну) вредност угаоне брзине.

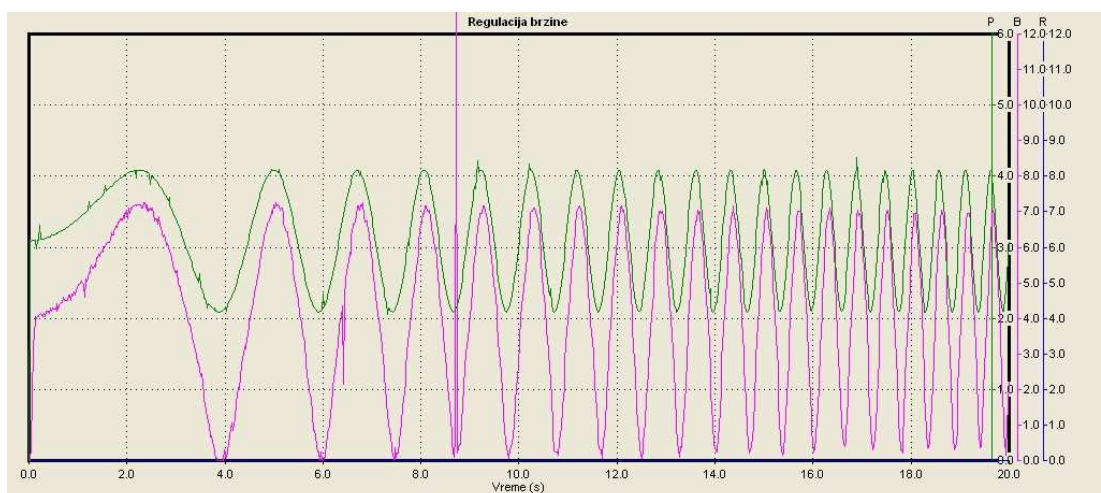
### 2.2.1 Одскочна (step) функција

Одскочна функција је најједноставнији начин побуде. Највећи недостатак ове функције је што добро описује прелазна стања, али може знатно одступати код динамичког начина рада.



### 2.2.2 Синусни сигнал са променљивом фреквенцијом

Овакав испитни сигнал се сматра најсвеобухватнијим јер може приказати одзив на читав спектар фреквенција. Најједноставнији начин да се систем побуди низом различитих фреквенција је да се користи синусни сигнал са временски помереном фреквенцијом (*swept*, *chirp* или *multitone*)



Једначина (2.1) описује промену фреквенције од  $t=0$  до  $t=t_f$ . Ова једначина генерише сигнал који мења фреквенцију линерано од  $f_0$  до  $f_f$  циклуса у секунди (Hz) и амплитуде  $A$ .

$$\omega(t) = 2\pi * \left[ f_0 + (f_f - f_0) \frac{t}{t_f} \right]$$

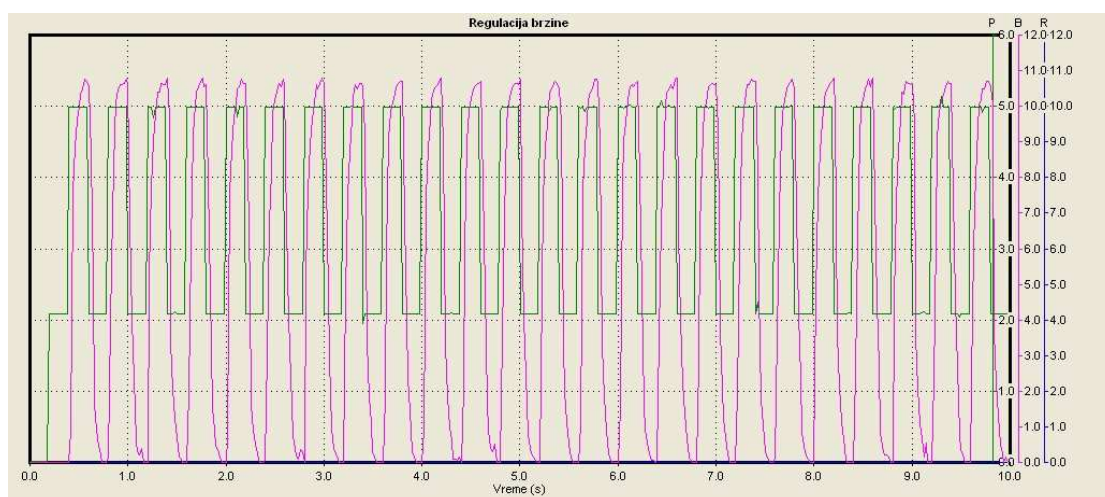
$$u(t) = A \sin\left(\omega(t) \frac{t}{2}\right) \quad (2.1)$$

Избором фреквенција  $f_0$ ,  $f_f$  и амплитуде  $A$  постављају се услови који ће најбоље описати понашање система.

Овакав начин омогућава добро моделирање динамичког понашања система, али није најприкладнији за одређивање статичког понашања система. Тада је много погодније одредити одзив на одскачну (*step*) функцију.

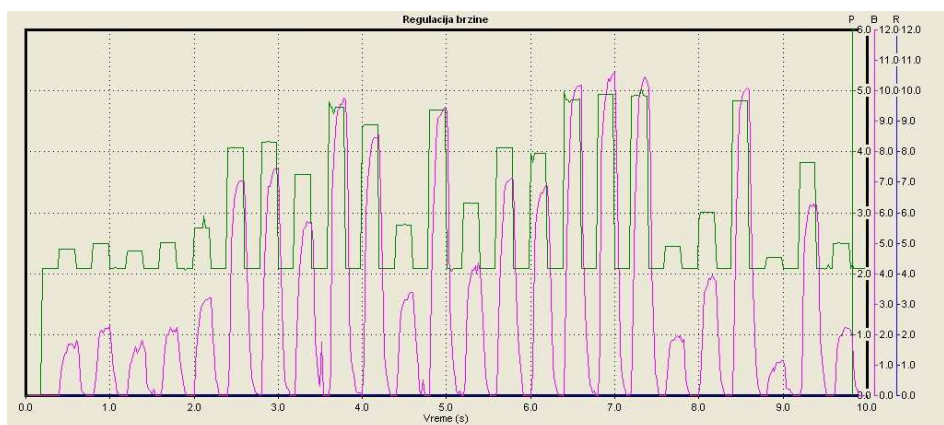
### 2.2.3 Низ правоугаоних сигнала константне амплитуде

Ширина импулса је износила 10 интервала узорковања, односно 200 ms. Амплитуда је износила максималних 5V, односно мотор је добијао 12V.



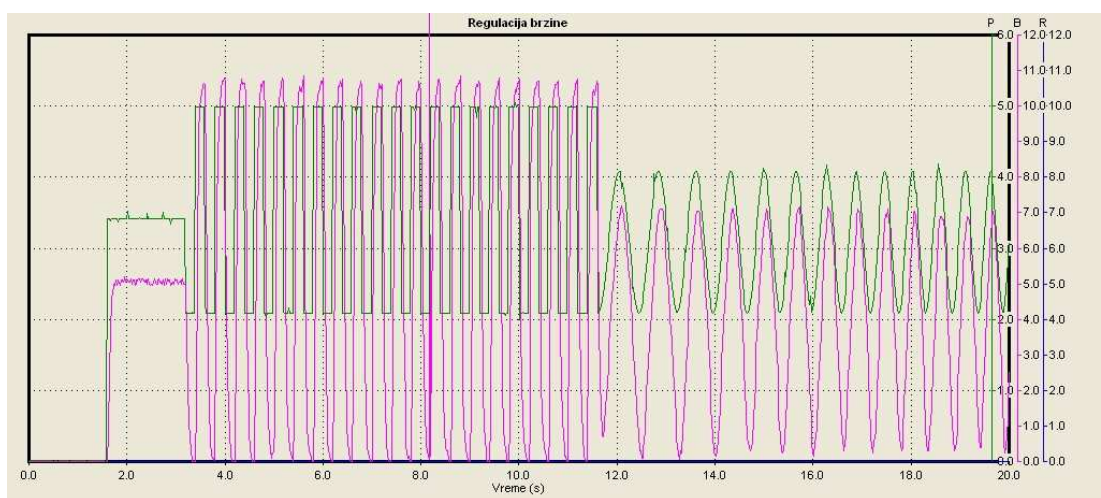
### 2.2.4 Низ правоугаоних сигнала случајне амплитуде

Обзиром да се ради о случајном сигналу, то значи да експеримент није могуће поновити, а самим тим ни верификовати идентификацију на истим подацима. Међутим, преносна функција која је добијена на овај начин може се користити при симулацији користећи друге испитних функција. На тај начин се индиректно може верификовати идентификација.



### 2.2.5 Комбиновани сигнал који се састоји од одскочне, синусне и правоугаоне функције

Идеја је била да се преносна функција добије користећи комбинацију свих претходних испитних сигнала.



### 2.3 Мерење и обрада података

Само генерисање испитних сигнала и аквизиција су вршени користећи програм писан у Delphi-ју. Након мерења подаци су снимани и учитавани у Matlab, где је помоћу Identification Toolbox-а вршена идентификација, то јест одређивање преносне функције. У свим случајевима је вршена идентификација користећи функције првог и другог реда, користећи једначине (1.6) и (1.7) да би се одредила оптимална преносна функција.

Општи облик функција које даје MatLAB је:

$$G(s) = \frac{K}{(1 + T_{p1}s)(1 + T_{p2}s)} \qquad G(s) = \frac{K}{(1 + Ts)}$$

Табеларни приказ добијених резултата на основу експеримената

Испитна функција	Систем 2. реда			Систем 1. реда	
	К	T <sub>p1</sub>	T <sub>p2</sub>	К	Т
Одскачна	0.97878	0.026765	0.016424	0.98187	0.044933
Синусни сигнал са променљивом фреквенцијом	1.0074	0.059439	0.001	290	0.048095
Низ правоугаоних сигнала константне амплитуде	347.42	0.012722	0.012721	357.87	0.025849
Низ правоугаоних сигнала случајне амплитуде	214.69	0.014559	0.014552	243.46	0.02351
Комбиновани сигнал који се састоји од одскачне, синусне и правоугаоне функције	60.113	0.034635	0.01	62.984	0.027175

Тако добијене преносне функције су затим коришћене за симулацију у *simulink* модулу. Коришћене су испитне функције које су генерисане и у Delphi-ју:

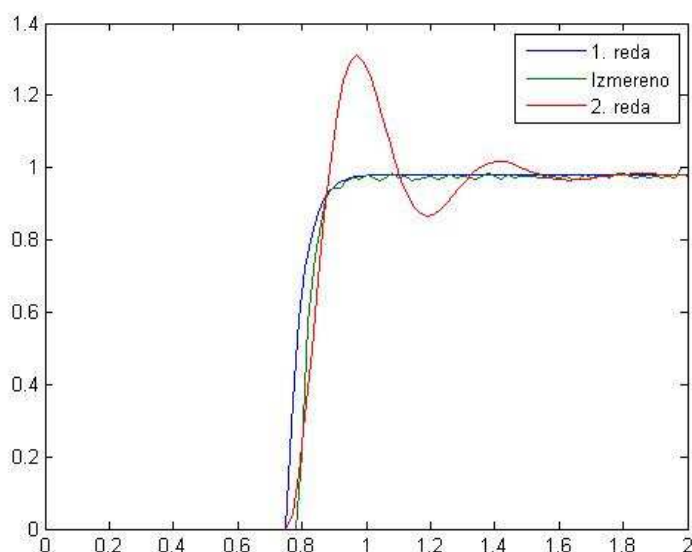
- одскачна (*step*)
- синусна (*chirp*)
- правоугаони сигнали (*pulse generator*)

да би се могли упоредити експериментални резултати и они добијени симулацијом.

## 2.4 Упоредна анализа резултата добијених у *simulink*-у

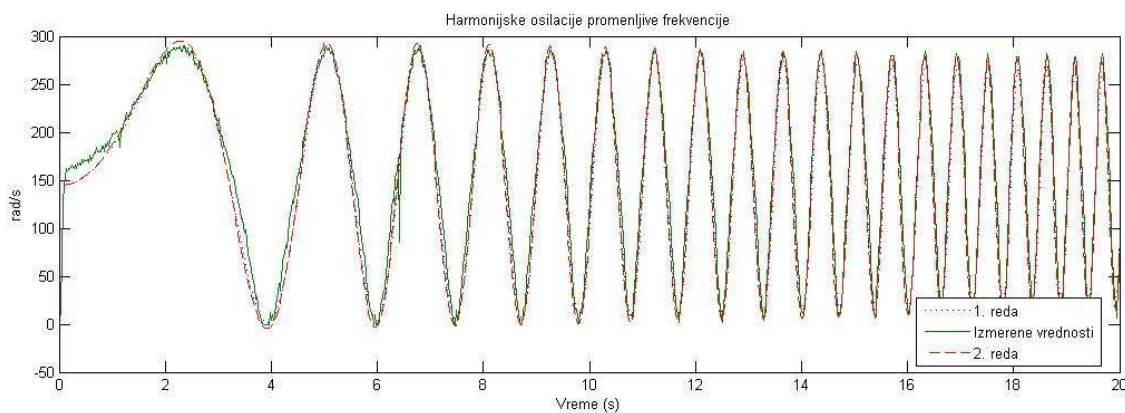
### 2.4.1 Одскочна функција

Као што се види функција првог реда сасвим добро апроксимира реални систем. Код функције другог реда постоји велики прескок па је она практично неприхватљива. Разлог лоше апроксимације лежи у релативно великом шуму који је добијен са енкодера и које је приликом моделирања узет као коректна вредност. Уз већи временски период посматрања процеса, функција би дала много прихватљивије вредности.



### 2.4.2 Синусни сигнал са променљивом фреквенцијом

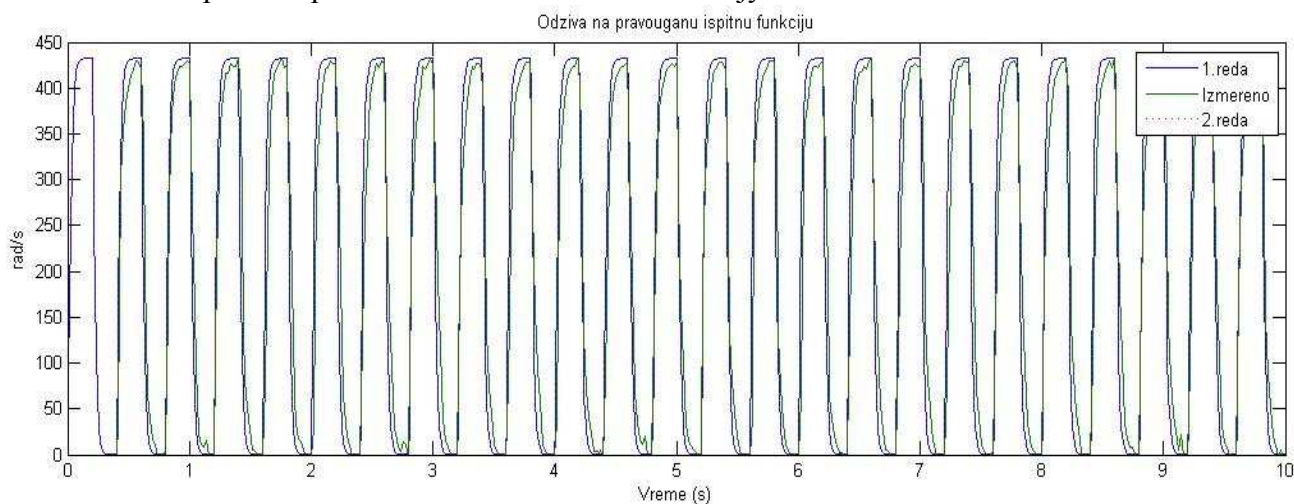
Са дијаграма се види да је идентификација успешно урађена и да обе функције дају прихватљиве резултате.



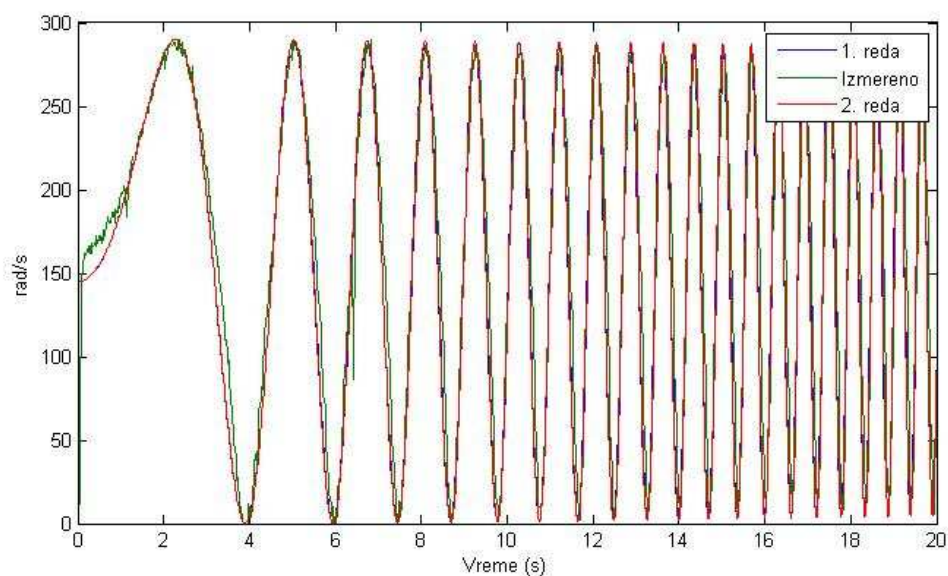


### 2.4.3 Низ правоугаоних сигнала константе амплитуде

Овде се види да се обе криве 1. и 2. реда практично поклапају и да је слагање са измереним вредностима сасвим задовољавајуће.



Као последња верификација идентификације урађена је симулација користећи преносну функцију добијену комбиновањем 3 сигнала. Као испитна функција коришћена је синусна функција описана у 2.2.2. И овде је такође постигнут висок степен тачности.



### 2.4.4. Закључак

На основу упоредних резултата, може се закључити да се систем може апроксимирати функцијом првог реда.

### 3. Дискретни PID регулатор

PID регулатор заузима посебно место међу континуираним регулаторима. Као и већина осталих регулатора и PID регулатор пореди измерену вредност ( $y$ ) са жељеном или референтном  $y$  (*Set-Point*). Разлика између те две вредности  $e$  је грешка (*error*). На основу те разлике регулатор врши потребне корекције параметара управљања (напон, струја) да би се елиминисала грешка, односно добиле жељене вредности.

За разлику од других, једноставнијих, алгоритама PID омогућава корекције засноване на анализама претходног понашања система као и на интезитету промене сигнала. На овај начин се омогућава прецизнија и стабилнија контрола управљаног система.

Подаци о понашању о понашању система се добијају са одговарајућег сензора. Разлика између жељене и добијене вредности се назива грешка и она се обрађује на три начина. Први је обрада тренутног стања (P – пропорционални), други је елиминисање грешке из прошлости (I – интегративно) и треће је предвиђање будућности (D – деривационо).

#### Пропоционални члан

Пропорционални члан обрађује само тренутно стање и то на основу грешке. Често се назива и појачање, пошто он одређује колико треба појачати (или смањити) управљачки сигнал. Уколико се регулација врши само помоћу P члана систем увек има стационарну грешку, осим ако је улазна вредност 0 или ако систем даје на изласку вредност жељеној. Уколико је P исувише велико систем постаје нестабилан и почиње да осцилује.

#### Интегрални члан

Интегрални члан омогућава бољу контролу, пошто он прати историју понашања система и на основу тога коригује управљачки сигнал. Грешке се сумирају све док се грешка не буде 0. И се ретко користи сам, пошто се добија спор одзив и систем је склон осцилацијама. Више се користи PI регулација која је стабилнија, пошто нема стационарне грешке и одзив је много бржи. Због тога већина индустријских контролера користи искључиво PI регулацију.

#### Деривациони члан

Деривациони члан **D** прати брзину промене грешке и на тај начин „предвиђа“ будуће понашање система. Ово нарочито побољшава одзив система код изненадних промена референтне вредности или стања самог система. **D** се практично никад не користи самостално већ у комбинацији као PD или PID. Превелико **D** доводи до нестабилности система. Битно је напоменути да се **D** понаша као високопропусни филтер за сигнал грешке и да због тога може унети нестабилност у систем и направити систем осетљивији на шум.



Основна једначина која описује деловање PID регулатора има облик:

$$u(t) = K \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (3.1)$$

при чему је

$K$  – константа појачања

$T_i$  - временска константа интеграцијског члана

$T_d$  - временска константа деривацијског члана

Након преласка у Лапласово подручје добија се

$$U(s) = K \left[ E(s) + \frac{1}{sT_i} E(s) + sT_d E(s) \right] \quad (3.2)$$

Овакав облик алгоритма назива се **паралелни алгоритам**, али се често назива и **идеални алгоритам** или **алгоритам без међуделовања**.

Ростоји доста варијација овог алгоритма, али се најчешће користи облик који се зове **идеално паралелни алгоритам**. Добијен је тако што су сва три члана помножена константом појачања, тако да се тада дефинишу константе појачања појединих чланова, а не њихове временске константе.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (3.3)$$

$K_p=K$  – константа појачања пропорционалног дела

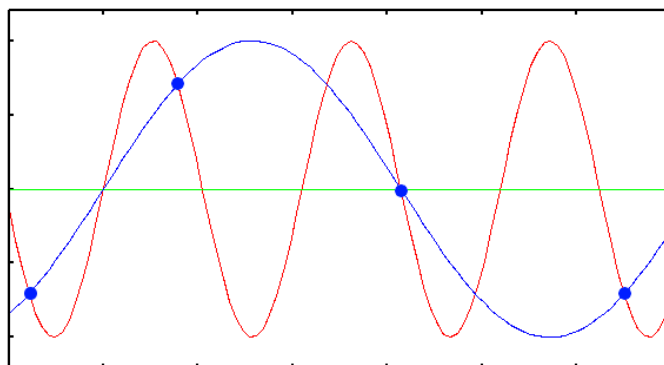
$K_i=K/T_i$  – константа појачања интегративног дела

$K_d=K/T_d$  – константа појачања деривационог дела

## Дискретни PID контролер

Аналогни PID контролери су уређени као самостални електронски склопови где функције појединих чланова обављају специјализиоване електронске компоненте. Подешавање се обично врши примерима а дијаграми се могу пратити помоћу осцилоскопа. Предност оваквог начина је што су сигнали континуалног типа и што се контрола одвија у реалном времену.

Дискретни контролери врше управљање користећи микропроцесоре. У одређеним временским интервалима врши се читавања сензора, рачуна грешку и шаље одговарајући сигнал. Највећи недостатак је тај што треба квантизовати континуалне вредности, односно одредити оптимални интервал узорковања. Мала вредност даје боље резултате, али захтева квалитетније уређаје (AD/DA конверторе) и више оптерећује процесор. Веће вредности су мање захтевне али се може догодити да читавања буду погрешна због такозваних „преклапања“ (*aliasing*).



Пример преклапања – црвени сигнал је прави, а плави је реконструисани .

Као што се види са слике, плаве тачке представљају очитане вредности и плава синусоида је сигнала реконструисна на основу тих тачака. Очигледно је да изгледа, рачуна грешку и шаље одговарајући сигнал. Највећи недостатак је тај што треба квантизовати континуалне вредности, односно одредити оптимални интервал узорковања.

Најједноставнији начин дискретизације PID регулатора је претварање једначине 3.3 у

$$u(kT) = K_p e(kT) + K_i \sum_{i=0}^k e(iT) + \frac{Kd}{T} (e(kT) - e[(k-1)T]) \quad (3.4)$$

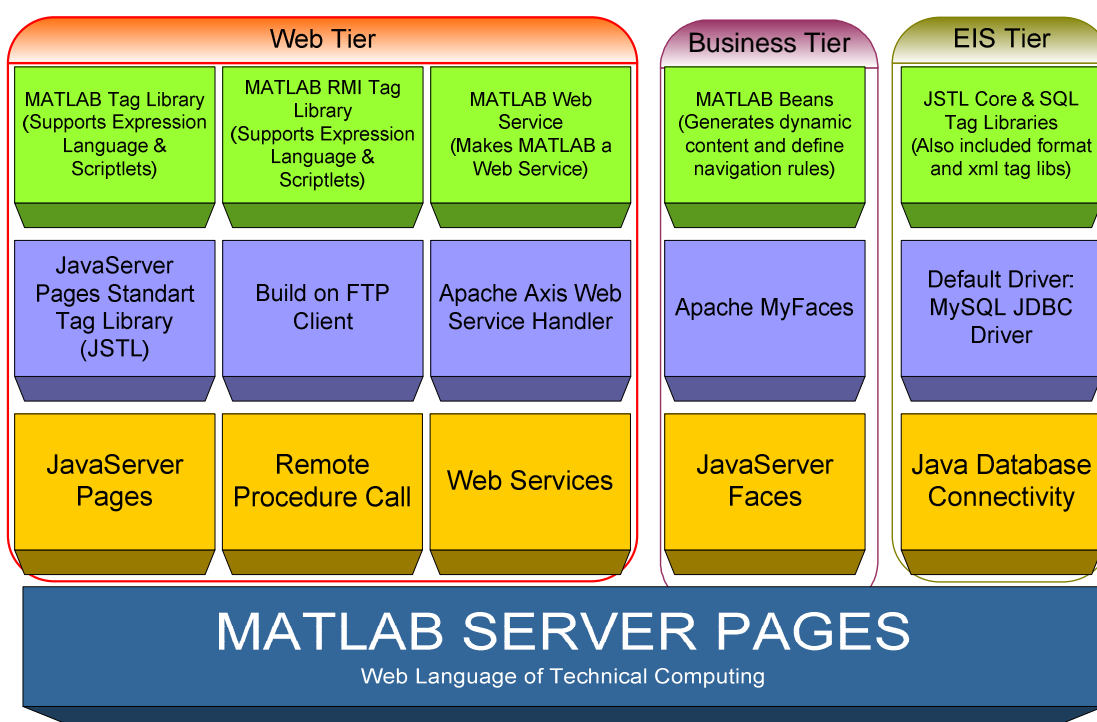
Једначина 3.4 назива се **позициони** или **апсолутни** алгоритам. Међутим, у пракси се показало да је много боље користити такозвани **брзински** или **инкрементални** алгоритам. Код њега се управљање врши на основу управљања у претходном тренутку и прираста управљања:

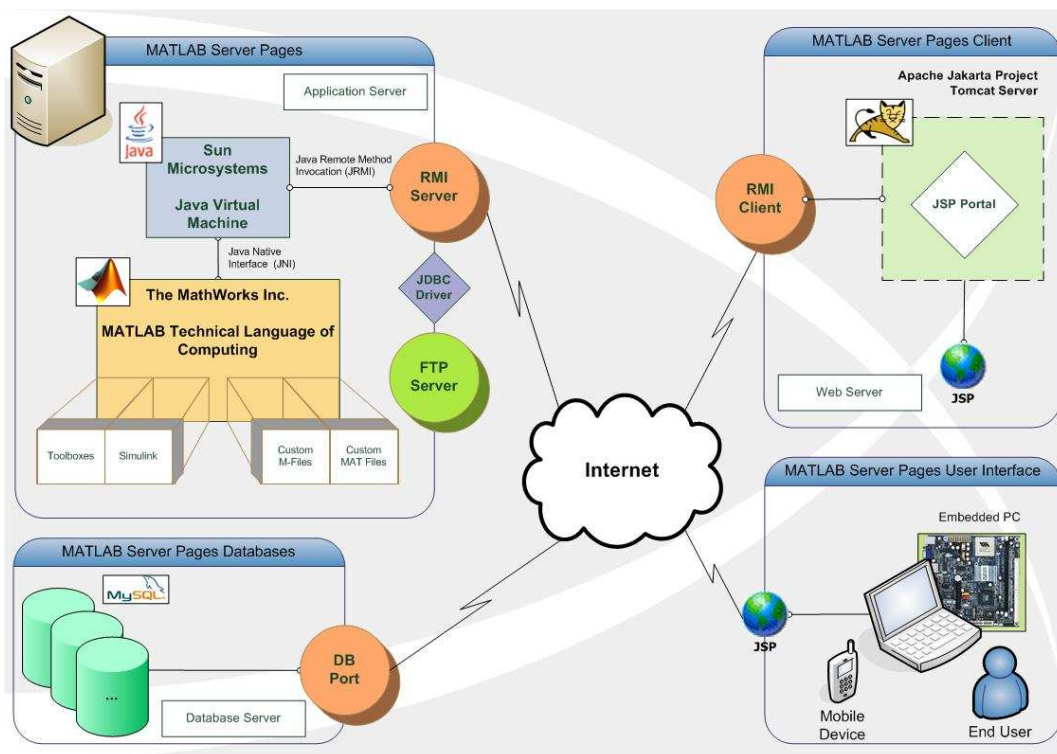
$$u(kT) = u[(k-1)T] + \Delta u(kT) \quad (3.5)$$

Због специфичности енкодера и програмирања у реалном времену под Delphi-јем, за PID регулацију је коришћена хибридна варијанта алгоритма која је подразумевала једначине (3.4) и (3.5)

## 4. Управљање системом преко интернета –WebLAB

Управљање експериментима преко WEB-а, такозвани weblab, све више постаје саставни део учења на даљину, *e-learning*-а. Обзиром да је MatLAB већим делом намењен студентима, таква компонента је неопходна за такав пакет и због тога MatLAB има подршку за web сервере. Међутим, ради боље демонстрације могућности такозваних *open-source* решења веће флексибилности информационог система изабран је MATLAB Server Pages (MSP). То је сложен систем који подржава трослојну архитектуру (Web – Пословно окружење- База података) користећи web сервисе.





Систем је сложен и пружа велике могућности, али је и компактан, тако да је инсталација једноставна и релативно брзо се може започети са основним примерима.

## MatLAB као део weblab-a

Након инсталације MPLS-a и основних подешавања конфигурације може се направити једноставна страница која позива већ урађен модел у MatLAB-u:

```
<html>
<head>
<%@ include file="/Scripts/header.inc" %>
<title>MSP Simulink Test</title>
</head>

<body>
<matlab:Engine>
<matlab:Command cmd="Kp =0.2;Td = 0.3;Ti=0.028; save sim2in"/>
  <matlab:EvalMDL modelname="simul300" loadname="sim2in" savename=""/>

  <matlab:PlotSim filename="draw1" handle="h1" simresult="signalout">
    <matlab:Thumbnail imagename="draw1" width="500" height="500"/>
  </matlab:PlotSim>

  <matlab:PlotModel filename="draw2" simname="simul300"/>
  <br>

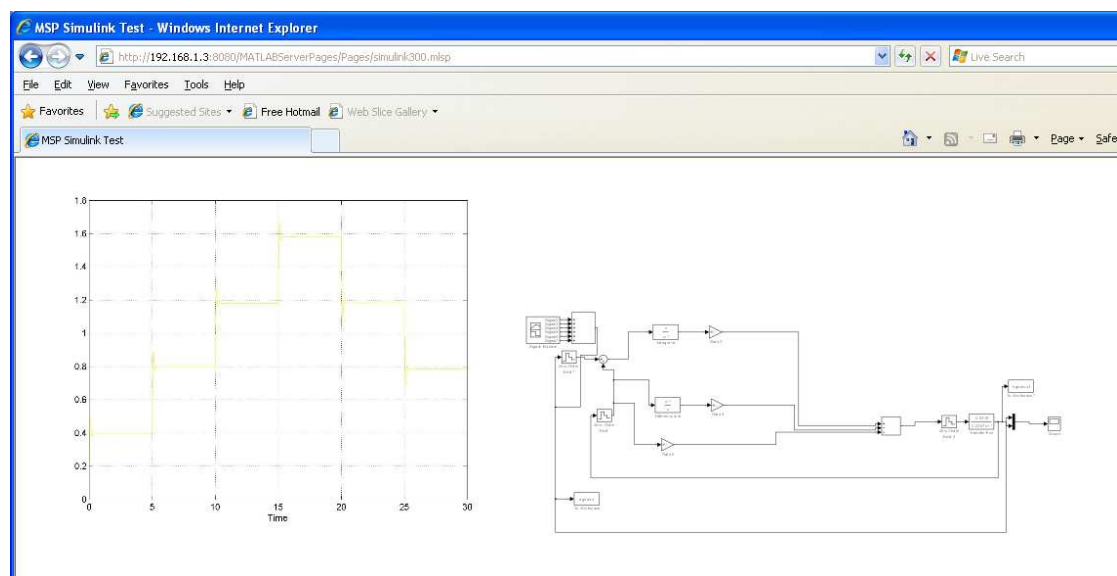
</matlab:Engine>
</body>
</html>
```

Као што се види, суштина је да MLSP додаје нове html тагове „<matlab:“. Коришћењем одговарајућих тагова позивају се и адекватне функције MatLAB-a. На пример, претходно наведени код, simulink300.mlsp, извршава се следећим редоследом:

1. Варијабле Kp, Td и Ti се постављају на одговарајуће вредности и чувају на web серверу под именом sim2in.  
<matlab:Command cmd="Kp=0.2;Td=0.3;Ti=0.028; save sim2in"/>
2. Затим се те вредности прослеђују моделу simul300.mdl. Тај модел враћа резултате преко signaluot структуре која се користи за приказ дијаграма.  
<matlab:EvalMDL modelname="simul300" loadname="sim2in" savename="/>
3. Дијаграми се цртају користећи signaluot структуру и опција Thumbnail се користи ако желимо да се слика не прикаже у пуној величини  
<matlab:PlotSim filename="draw1" handle="h1" simresult="signalout">  
    <matlab:Thumbnail imagename="draw1" width="500" height="500"/>  
</matlab:PlotSim>
4. На крају цртамо и сам модел simul300.mdl  
<matlab:PlotModel filename="draw2" simname="simul300"/>

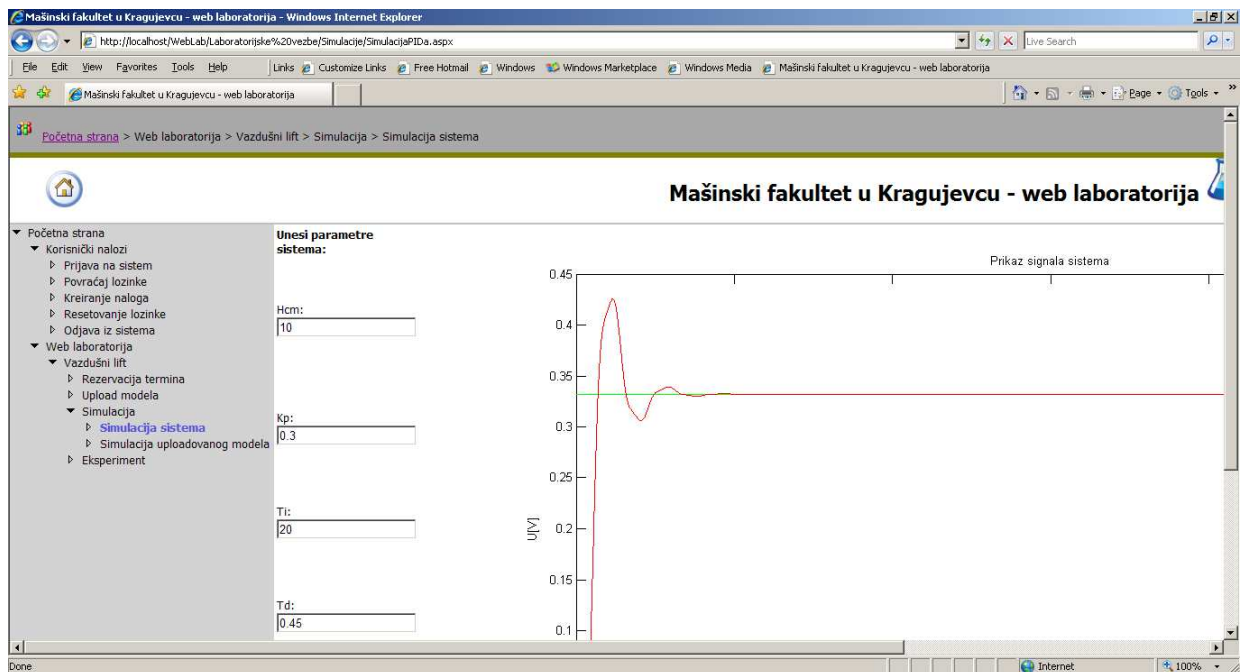
Позивањем

<http://147.91.203.32:8080/MATLABServerPages/Pages/simulink300.mlsp>  
добила се следећа страница:



Наравно, ово је доста оскудан пример, али је илустративан да би се показала могућа једноставност читавог система. Наравно, за озбиљну примену треба направити сложеније и сигурније окружење.

Једна од могућих варијанти је коришћење MatLAB-a, ASP.NET-a, SQL baze и IP камере



Ту се могу дефинисати кориснички налози, резервисати време када се вежбе изводе све што је неопходно за један стандардан weblab.

## 5. Практична реализација експерименталног модела

### 5.1 Управљање једносмерним моторима

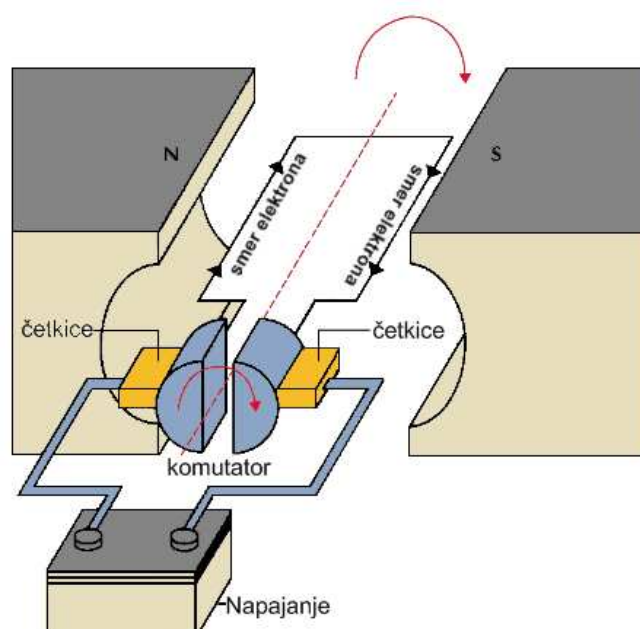
Једносмерни (DC – **D**irect **C**urrent) мотори су широко распрострањени у роботизи,

Основни разлози су:

- Релативно једноставна конструкција
- Једноставно управљање
- Велике брзине

Наравно, поред предности постоје и недостаци. Један од основних је потреба за одржавањем, односно повремено је потребно заменити графитне четкице.

Принципијелна шема једносмерног мотора би изгледала овако:



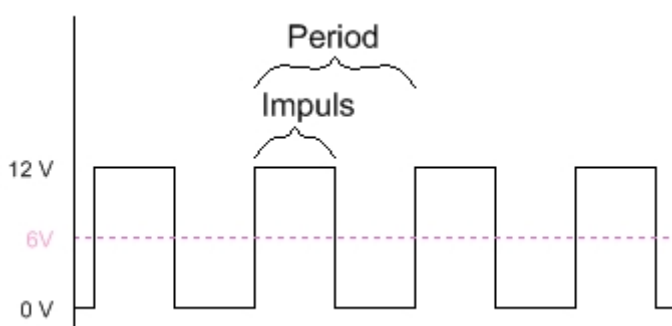
За рад електромотора је битан напон и струја која протиче кроз ротор. Променом напона се мења брзина ротације. Променом поларитета напајања мења се и смер окретања мотора. Струја која протиче кроз намотаје мења момент, односно снагу мотора. Уколико се осовина мотора оптерети, доћи ће и до повећања струје. Уколико напајање не може да пружи толико струје, долази до пада напона, односно смањења броја обртаја мотора. Уколико напајање може да одржава константан напон, онда и брзина остаје непромењена.

Једносмерно напајање може бити класично аналогно. На пример, то је батерија или исправљач који наизменичну струју претвара у једносмерну. Овакво напајање има континуалну промену амплитуде као и бесконачну резолуцију. Основна предност аналогних напајања је једностава конструкција. Са друге стране, недостатак је веома тешко одржавање константних вредности, односно излазног напона. И саме електронске компоненте, које се користе за изградњу оваквих напајања временом мењају карактеристе, што додатно

усложњава проблем. Проблем је и дисипација, па је потребно и одговарајуће хлађење. Такође су осетљива и на шум

Уместо аналогних кола, много је боље користити импулсно ширинску модулацију (PWM – **P**ulse **W**idth **M**oduation). У основи ИШМ омогућава да се дигитални сигнал користи за контролу аналогних уређаја. Многи микроконтролери (Microchip, Atmel) као и DSP имају већ уграђене ИШМ генераторе, што још више проширује подручје њихове примене и поједностављује употребу. Такође је и дисипација много мања.

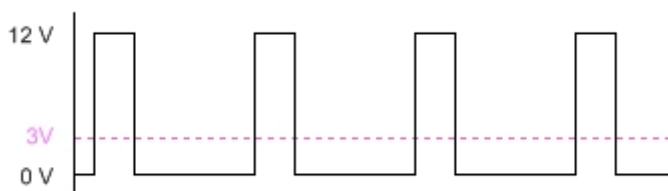
Суштина рада оваквих кола је да се у одређеном временском интервалу (фреквенцијом) шаљу и прекидају сигнали исте вршне вредности. Индуктивност мотора не дозвољава да се струја тренутно промени, односно да дође до пада напона на 0, тако да намотаји ротора добијају ефективни радни напон који је пропорционалан односу импулса и паузе.



$$\text{Испуна} = \frac{\text{импулс}}{\text{период}} * 100 (\%)$$
$$\text{фреквенција} = \frac{1}{\text{период}} / \text{сек}$$

Да би се разумео начин рада ИШМ-а треба објаснити три основна појма:

1. Импулс (*pulse*) – период у ком постоји напон
2. Пауза (*pause*) – период када нема напона
3. Период - чине га импулс и пауза
4. Испуна (*duty cycle*) – однос импулса/пауза. Испуна, односно фактор испуне дефинише колики је ефективни напон који “види” уређај. На пример на слици 2 (50%) напон ће бити 6V, док ће на слици 3, са фактором испуне 0,25 (25 %) бити 3V.



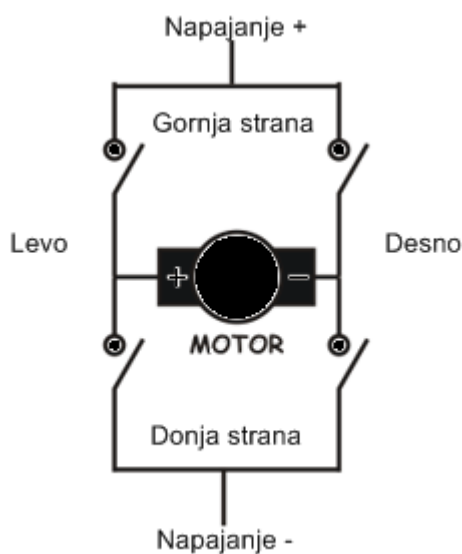
Ови параметри зависе од уређаја који се погоне. На пример, за моторе је препоручљиво да фреквенција буде 30-40KHz, пошто тада мотор не стиже да “региструје” паузе, односно струја и даље протиче кроз намотаје. Такође треба обратити пажњу да људско уво чује фреквенције испод 20 KHz, тако да се може чути зујање на уређајима.

Недостатак ИШМ може бити сложенија конструкција.



## 5.2 Појачавачи

Управљачке јединице (контролери, *drivers*) имају улогу да приме повратну информацију од система и пошаљу управљачки сигнал извршним органима. Обзиром да су извршни органи (актуатори) често велики потрошачи, управљачки сигнал није довољан за њихово покретање и тада је потребно додати појачавачке елементе. Најчешће се користи такозвани Н –мост (*H-bridge*). Тако је назван због распореда прекидачких елемената (транзистора или MOSFET-a) који својим радом дефинишу смер мотора.



Као што се види, четири прекидача, дефинисана као горња и доња страна, подељени на леву у десну половину, дефинишу смер кретања струје односно смер окретања мотора, у зависности од комбинације прекидача.

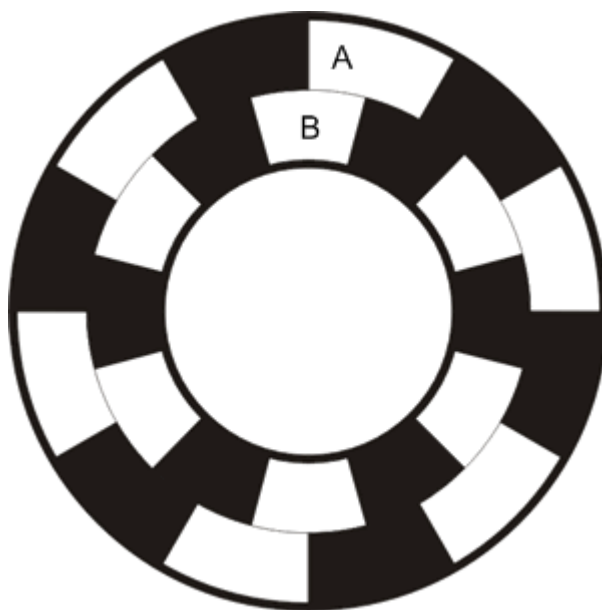
### 5.3 Енкодер

Технички гледано, енкодери су посебна врста ротационих прекидача. Могу бити оптички и механички. Оптички енкодери се састоје од диска са маркерима и транзистора и диода помоћу којих се врши читавање.



Данашњи енкодери типично имају два канала А и Б, као и посебан маркер који се означава као почетни импулс Z или индекс. Индекс служи као референтна тачка и може се користити за одређивање почетне тачке или проверу да ли су сви импулси прочитани. Канали А и Б су правоугаони импулси и они служе за одређивање брзине и смера ротације. Брзина би се могла одређивати и помоћу једног канала, пошто би било довољно бројати само број импулса у јединици времена. Међутим, тада не би било могуће одредити и смер ротације. Међутим, додавањем још једног канала, који је померен за 90°, могуће је тачно одредити и смер ротације. Одређивање смера се врши на основу такозваног Грејевог\* или Греј кода. То је бинарни систем у којем се два суседна стања разликују само по једном биту:

\* Назван по Френку Греју (Frank Gray), који је 1953. патентирао овај систем.



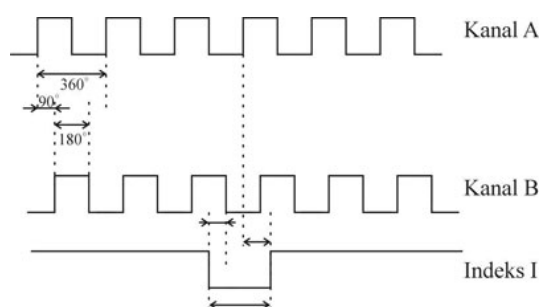
Фаза	A	Б
1	0	0
2	0	1
3	1	1
4	1	0

У смеру казаљке на сату

Фаза	A	Б
1	1	0
2	1	1
3	0	1
4	0	0

Супротно смеру казаљке на сату

### Изглед канала



Поред **инкременталног** енкодера, постоје и **апсолутни**. Разлика је у томе што апсолутни, као што им само име каже, памте тренутну позицију, тако да се и без почетне ротације може одредити тачан положај. Код инкременталних енкодера, мора се очитати почетни импулс и од њега започети мерење. Основни недостаци апсолутног енкодера су сложенија конструкција и компликованији алгоритми за обраду информација, а последица тога је већа цена.

## **5.4 Реализација енкодера на систему Серво 300**

Серво 300 користи мотор Escape 28D2R који већ има уграђен инкрементални енкодер. Да би се импулси од енкодера претворили у бројчани, односно бинарни облик, направљена је посебна картица. Поред обраде сигнала, она уједно служи за напајање енкодера. Основу картице чини Атмелов микропроцесор. Обзиром да се за аквизицију користи NI6008 који има 12 дигиталних улаза а ATME1 има 8 излаза, компромисно решење је да се могу читавати само 8 битни подаци. Преосталих 4 канала се користи за управљачке команде ка микропроцесору. Обзиром да је 8 битна резолуција може бити сувише мала (дозвољава само 255 вредности) прибегло се некој врсти мултиплексирања. Наиме, картица непрекидно прикупља податке. Када стигне команда за слање података, микропроцесор “замрзне” стање (**Latch**) и интерно чува 16-битну вредност, али као 2 суплементарне 8-битне вредности.. Програм који комуницира са картицом задаје команду да учита горњи (**LSB - Least Significant Bit**) а затим и доњи (**Most Significant Bit**) бајт. Након тога се та два бајта спајају у један 16-битни број. За следеће читавање се читав циклус понавља. Поред читавања вредности могуће је задати команду за тип података који се читава (положај или угао). Обзиром да је енкодер инкрементални позиција је у ствари само број импулса.

Псеудо-код за комуникацију :

```
// citanje brzine
P3 := $04; //Merimo brzinu -1, hi - 0 latch - 0
WriteByte(Порт, P3);

// citamo Lobyte
LoByte:=ReadByte(Порт);

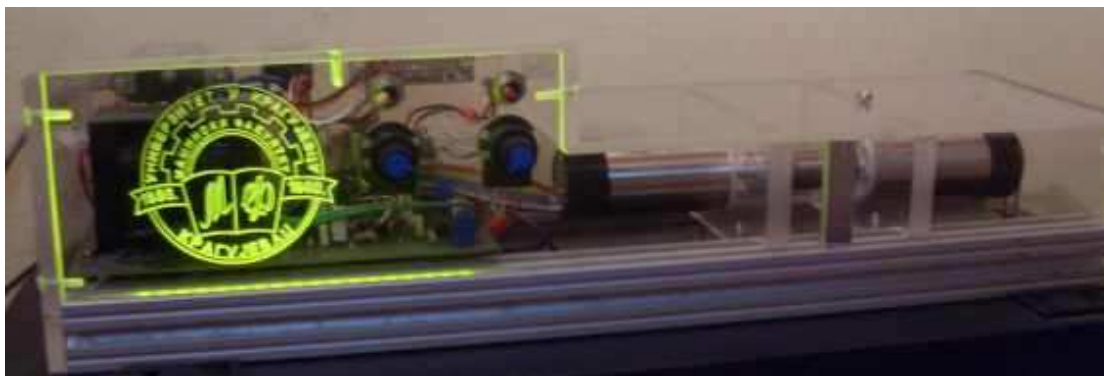
P3 := $05; // fi/omega msb Trazimo brzinu -1, hi - 0 latch -1
WriteByte(Порт1, P3);

// citamo Hibyte
Hibyte:=ReadByte(Порт0);

// Spajamo 8-bitne brojeve u 16-bitni
Brzina := BytesToward(HiByte,Lobyte);

// unlatch omega
P3 := $00; // fi/omega -0, hi - 0 latch -0
WriteByte(Порт1, P3);
```

## 6. Серво 300



### Опис

Серво 300 је систем намењен за анализу рада серво система. Систем се састоји од 4 елемента:

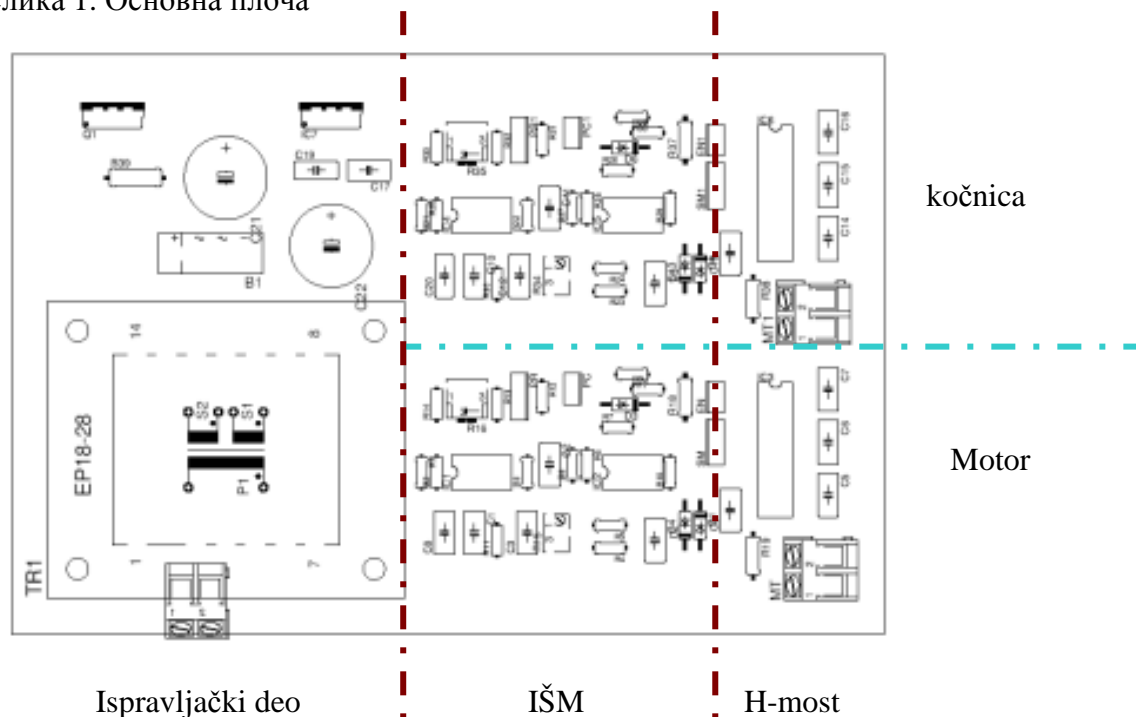
1. AD/DA конвертора – модел NI USB-6008, National Instruments
2. Електронике, која се састоји од:
  - а) исправљачког дела – извор струје са напонима +5V, +12V.
  - б) дела за обраду сигнала са енкодера – избор врсте сигнала: положај или угаона брзина
  - ц) ИШМ (импулсно ширинска модулација) регулатора – погон мотора
  - д) излазног појачала, односно H моста – омогућава до 1.5A, промену смера и enable/disable мотора.
3. Рогонског мотора, ESCAPE 28D2R и електромагнетне кочнице.
4. Програм за контролу и управљање, Тренутно се користи наменски написани програм, али је могуће коришћење и других програмских пакета, као што су MatLAB или Scilab.

### Опис електронике

Електроника је рађена модуларно и састоји се од две плочице.

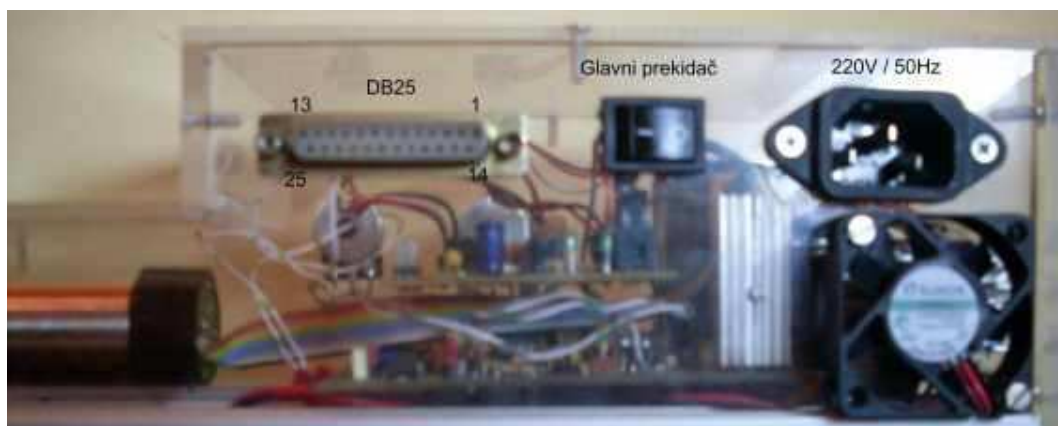
На основној плочи се налази исправљачки део (грец и пратећи елементи), ИШМ(PWM) регулатор и излазни појачавачки степен – Н мост.

Слика 1. Основна плоча



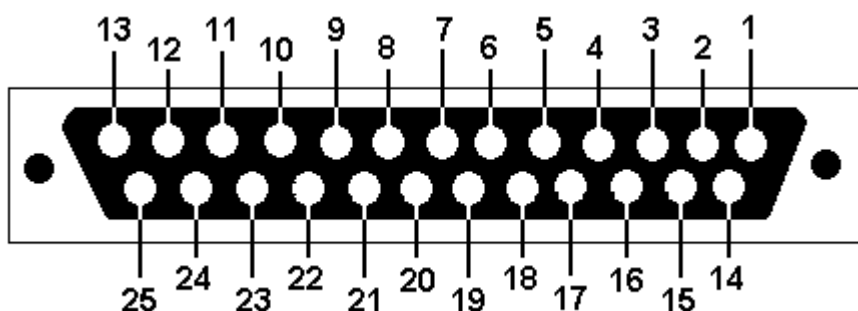
Изнад основне плоче је стављен енкодер, односно модул који напаја енкодер на мотору и обрађује сигнале примљене са њега.

## Опис конструкције

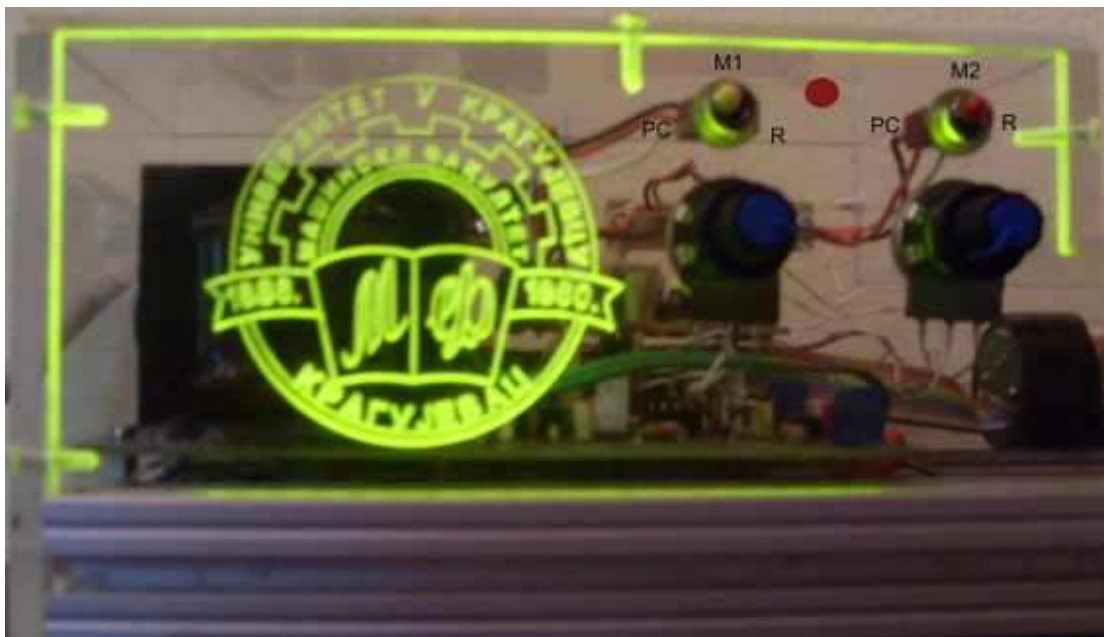


Слика 3.1. Изглед задње стране

Са задње стране се налази утичница за наизменичну струју 220V, главни прекидач и DB25 конектор. Конектор DB25 се користи за повезивање са AD/DA конвертом. Распоред пинова је приказан на следећој слици:



DB25	Намена	Боја жице	Сигнал	IN/OUT	NI6008
1	1. бит са енкодера	љубичасто-црна	Дигитални	IN	P0.0
2	2. бит са енкодера	браон	Дигитални	IN	P0.1
3	3. бит са енкодера	црвена	Дигитални	IN	P0.2
4	4. бит са енкодера	розе	Дигитални	IN	P0.3
5	5. бит са енкодера	наранџаста	Дигитални	IN	P0.4
6	6. бит са енкодера	жута	Дигитални	IN	P0.5
7	7. бит са енкодера	тамноплава	Дигитални	IN	P0.6
8	8. бит са енкодера	светлоплава	Дигитални	IN	P0.7
9	Избор $\alpha$ или $\omega$	зелена	Дигитални	OUT	P1.0
10	Li/Hi бајт	љубичаста	Дигитални	OUT	P1.1
11	Latch	светло зелена	Дигитални	OUT	P1.2
12		сива			
13		бела			
14	Излаз за M1	бело-црна	Аналогни	OUT	A01
15	Излаз за M1	црвено-црна	Аналогни	OUT	GND
16	Излаз за кочницу	жуто-црна	Аналогни	OUT	A00
17	Излаз за кочницу	зелено-црна	Аналогни	OUT	GND
18-25	0V (GND)	црна	Аналогни	-	GND



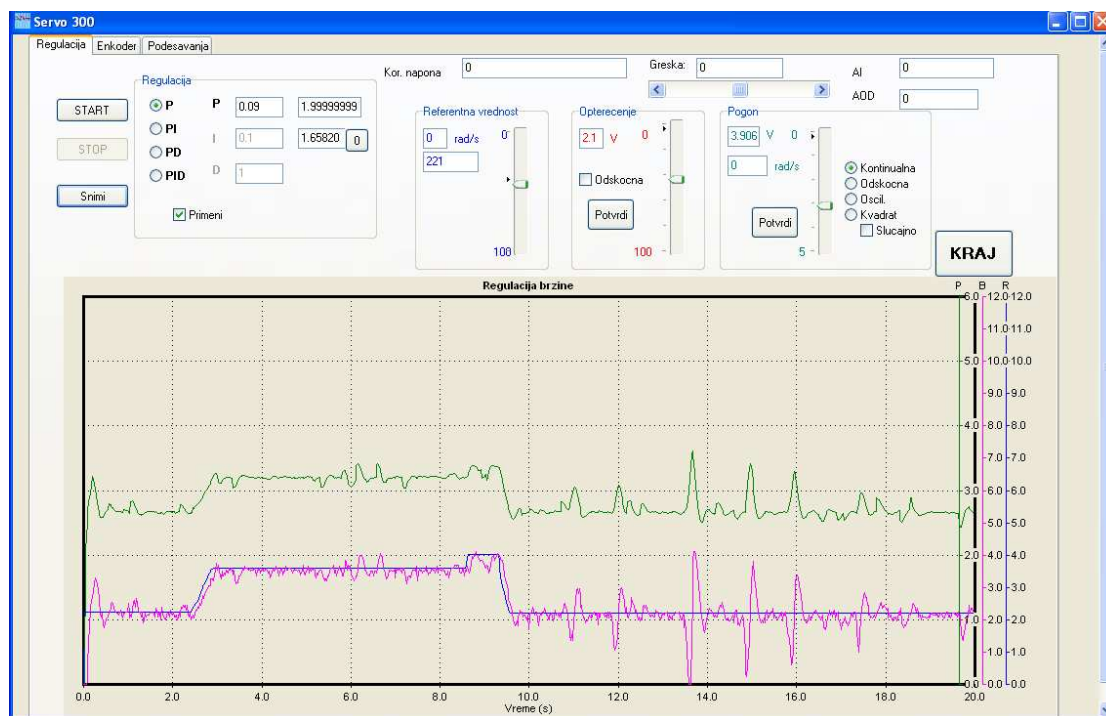
Слика 3.2. Изглед предње стране

На предњој страни се налазе два потенциометра и два прекидача, којима се одређује начин рада мотора. Група **M1** је за мотор 1, а **M2** за кочницу. Прекидач има два положаја – “**PC**” и “**R**”. “**PC**” значи да ће број обртаја бити регулисан преко програма који се налази на рачунару, а “**R**” означава ручни начин, односно преко потенциометра који се налази испод прекидача.



## Програм

Систем се контролише помоћу програма Servo 300.



Програм има имплементиране основне функције, као што су:

- PID регулација,
- слање напона мотору и кочници
- облик сигнала може бити континуални, одскочни, правоугаони и синусни,
- подешавање временских параметара програма,
- графички приказ података добијених са AD/DA конвертора, као и њихово снимање у табеларном облику.

У оквиру програма постоје 3 картице, означене као

- **Регулација**
- **Енкодер**
- **Подешавања**

## Подешавање

Картица за подешавање има три групе података. Прва је **Калибрација** и практично се односи на подешавање параметара електронских компоненти:

- **Скала енкодера** – Картица за обраду сигнала са енкодера даје скалиране вредности за брзину. Овај однос зависи од верзије firmware-а и тренутно износи 2.132
- **Мин. излаз** – ова вредност дефинише доњи праг напона, односно то је вредност испод које мотор добија побуду од 0V,
- **Мах. излаз** – ова вредност дефинише доњи праг напона, односно то је вредност испод које мотор добија побуду од 0V,
- **Аутоматско** – Одређивање Мин. и Мах излаза се врши визелно или мерењем напона на мотору. Аутоматско одређивање омогућава да програм сам одреди те граничне вредности. Потребно је само активирати ову опцију и када се зачује звучни сигнал калибрација је завршена.

Друга група, **Узорковање**, се односи на параметре рада програма:

- **Трајање** – време, у секундама, колико ће трајати прикупљање узорака
- **Интервал** – интервал, у секундама, којим се дефинише колико често ће програм узимати узорке.
- **Тимер** – време, у милисекундама, којим се дефинише колико често ће програм правити прекиде да би одрадио захтеве за узорковањем.

Комбинација ови параметара одређује прецизност графика, као и колико су добијене криве “глатке”.

Трећа група, функција, дефинише параметре синусне функције.

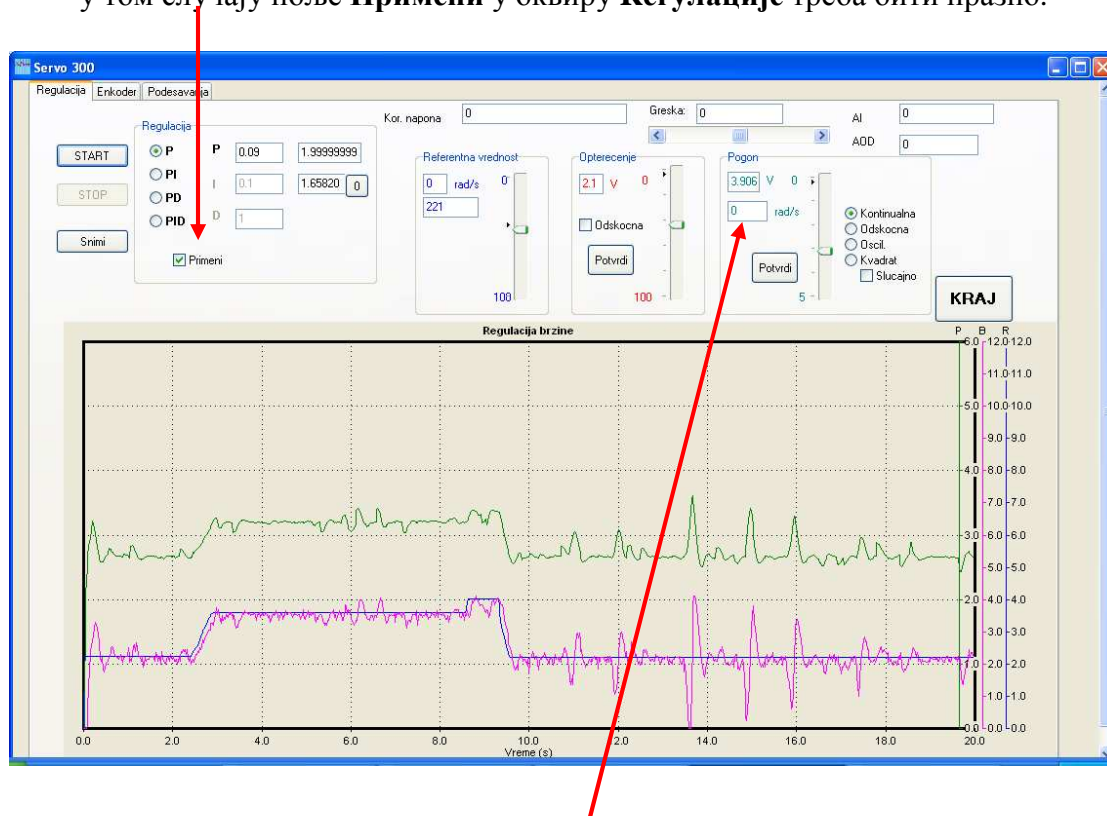
Четврта група **AD/DA конвертор**, дефинише параметре конвертора NI6008:

## Регулација

Картица за регулацију је окосница програма.

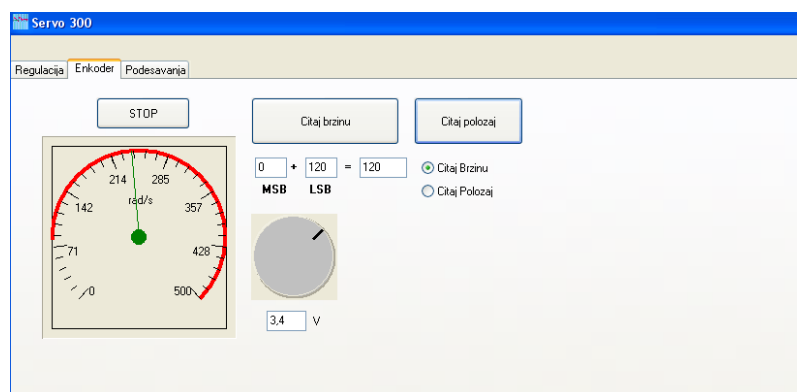
- Група података дефинисана као **“Регулација”** дефинише тип регулације који се користи, као и њихове вредности.

- **“Референтна”** вредност је вредност коју желимо да добијемо, односно да контролишемо. Тренутно је дата процентуално
- **“Оптерећење”** представља напон којим се побуђује електромагнетна кочница.
- **Погон** је напон који побуђује погонски мотор. Са померањем клизача та вредност се тренутно мења, али ако желимо да добијемо одзив на неку другу функцију, треба само обележити поље поред те функције. Наравно, у том случају поље **Примени** у оквиру **Регулације** треба бити празно.



Испод вредности напона, дата је и угаона брзина.

Картица означена као **Енкодер** омогућава проверу рада енкодера, односно лакше разумевање тумачење добијених података. Са леве стране се налази сат који визуелно приказује вредности (брзину или позицију). Поред њега се налази дугме чијом ротацијом се повећава или смањује напон на погонском мотору. Поља MSB и LSB означавају бајтове који се sukcesивно учитавају са енкодера и чијим се спајањем добија коначна вредност.



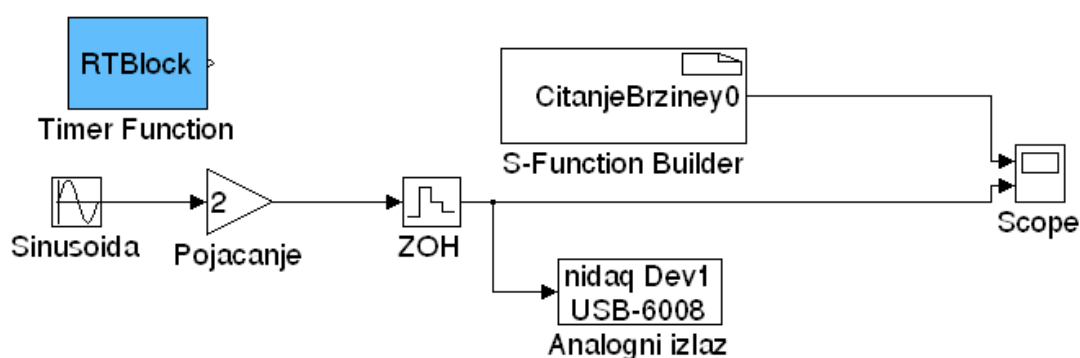
## **Закључак**

Експериментални модел је показао задовољавајуће резултате. Идентификација и верификација, које у обављене у MaLAB-у, такође су показале да су измерене вредности коректне.

Међутим, уочена су и два велика недостатка. Први је што Delphi није најоптималнији избор када се ради о програмирању у реалном времену, тако да интервали узорковања морају бити релативно велики, већи од 20ms и пажљиво усклађени са испитном функцијом. Тиме модел губи на општости. Други недостака је изведба енкодера. Наиме, мотор нема фабрички уграђен енкодер, већ само фототранзисторе и диоде. Генерисање А и Б канала се врши помоћу микроконтролера и добијена вредност се добија у дигиталном облику, преко два блока бајтова. Читав поступак добијања измерене вредности је релативно сложен и захтева петљу која понавља одређене секвенце читања и писања са AD картице. То уноси додатно кашњење, тако да подаци нису веродостојни и врло често изгледају као шум. Због тога је D компонента, код PID регулације, практично неупотребљива.

## Додатак А - Подршка за NI6008 у MATLAB –у

MATLAB R2008a, односно simulink, подржава AD/DA картицу NI6008 и то преко Data Acquisition Toolbox-a. У поглављу 5.4 је објашњен начин на који је извршена обрада сигнала са енкодера. Обзиром да је за један податак требало извршити више читања и уписивања тачно одређеним редом, било је потребно написати функцију која ће то радити, пошто се у *simulink* не може увек тачно задати приоритет функција, односно блокова. Та функција је направљена користећи *S-Function Builder*. Друга специфичност је рад у реалном времену (Real Time) . Због тога је додат и RTBlock, који се може бесплатно преузети. Слање аналогног сигнала (напона) ка мотору и читавање брзине има следећу блок-шему:



Наравно, уместо *S-Function Builder* блока, може се користити *S-Function* који је логичнији и визуелно лепши. Међутим, из едукативних разлога остављен је први блок. На тај начин се детаљно може проучити структура функција.

## Додатак Б - Карактеристике мотора ESCAP 28D2R



- Напон: 12 V
- Номинална брзина неоптерећеног мотора: 5700 о/мин
- Стартни момент: 96 mNm
- Струја неоптерећеног мотора: 44 mA
- Максимална струја: 1.5A
- Максимални трајни момент: 28.4 mNm
- Електрична константа: 0.0195 Vs/rad
- Механичка константа: 20 mNm/A
- Отпор арматуре: 2.5  $\Omega$
- Индуктивност rotora: 0.3 mH
- Константа вискозног трења: 1e-6 Nms/rad
- Број импулса по обртају: 144 (2 импулса померена за 90°)
- Напон напајања енкодера - 5V
- Распоред конектора енкодера  
Pin +/- =12VDC Motor  
Pin 1 = Emitter 1  
Pin 6 = Emitter 2  
Pin 2 = Kolektor 1  
Pin 5 = Kolektor 2  
Pin 3 = Katoda  
Pin 4 = Anoda
- Пречник мотора: 28 mm
- Дужина мотора: 75 mm
- Пречник осовинеротора: 4 mm
- Дужина осовине ротора: 20 mm
- Фиксирање мотора – 6 вијака М3.