

Načrtovanje programske kode s končnim avtomatom za avtomatska drsna vrata

Franc HANŽIČ, Karel JEZERNIK, Slavko CEHNER

Izveček: Z rekonstrukcijo krmilnika za avtomatska drsna vrata, ki vključuje mikrokrmilnik ARM Cortex M3 [2], se je odprla rešitev za izvedbo zmogljivejšega programskega algoritma. S tem bi izboljšali dinamiko in vzdržljivost vrat. Z novim generatorjem giba bi izpopolnili dinamiko vrat, vendar se pojavi problem izvedbe zanesljive programske kode, ki se mora izdelati za mikrokrmilnik. Rešitev s končnimi avtomati daje možnost, da je izvedba programske kode izvedena v koračnem načinu in s tem je vsaka programska funkcija ločena v posameznem stanju. V kolikor so programske funkcije ločene med seboj, obstaja boljša možnost izločevanja programskih napak. V nadaljevanju so prikazana raziskovalna dela na področju generatorja giba z S-obliko hitrosti, uporaba končnih avtomatov, razlog za uporabo generatorja pri vratih, izdelava v programskem okolju Matlab/Simulink/Stateflow ter rezultati.

Ključne besede: programsko načrtovanje, avtomatska drsna vrata, končni avtomati, mikrokrmilnik, oblikovalnik giba

1 Uvod

Želimo izdelati generator giba za avtomatska drsna vrata [1], ki bi zmanjšal segrevanje aktuatorja, mehanske sunke (vibracije) in porabo električne energije. Z raziskovanjem novega generatorja je potrebno izdelati tudi dobro dokumentacijo, ki prikazuje delovanje v končnem avtomatu. Trenutni generator giba izdeluje trapezno obliko hitrosti, ki ima sunke pri prehodih med hitrostmi. S sunki se tako stopnjujejo vibracije na konstrukciji in mehanizmu vrat, večja poraba električne energije itd. Z uporabo generatorja, ki zna generirati S-obliko hitrosti, se omenjeni problemi zmanjšajo. Generator z S-

obliko hitrosti je znan princip in ga uporabljajo v specifičnih industrijskih strojih. Različni avtorji so že izdelali matematične modele (enačbe) različnih vrst hitrostnih profilov (trapezni, S, \sin^2 itd.) [7], [8]. Cilj naloge je razdelati matematični model v obliko, ki bi bila razumljiva za delovanje in izdelavo programske kode v kakršni koli obliki programskega jezika. Prav tako je potrebno generator giba preurediti za avtomatska drsna vrata. Z metodo opisa delovanja mehanizma v končnem avtomatu

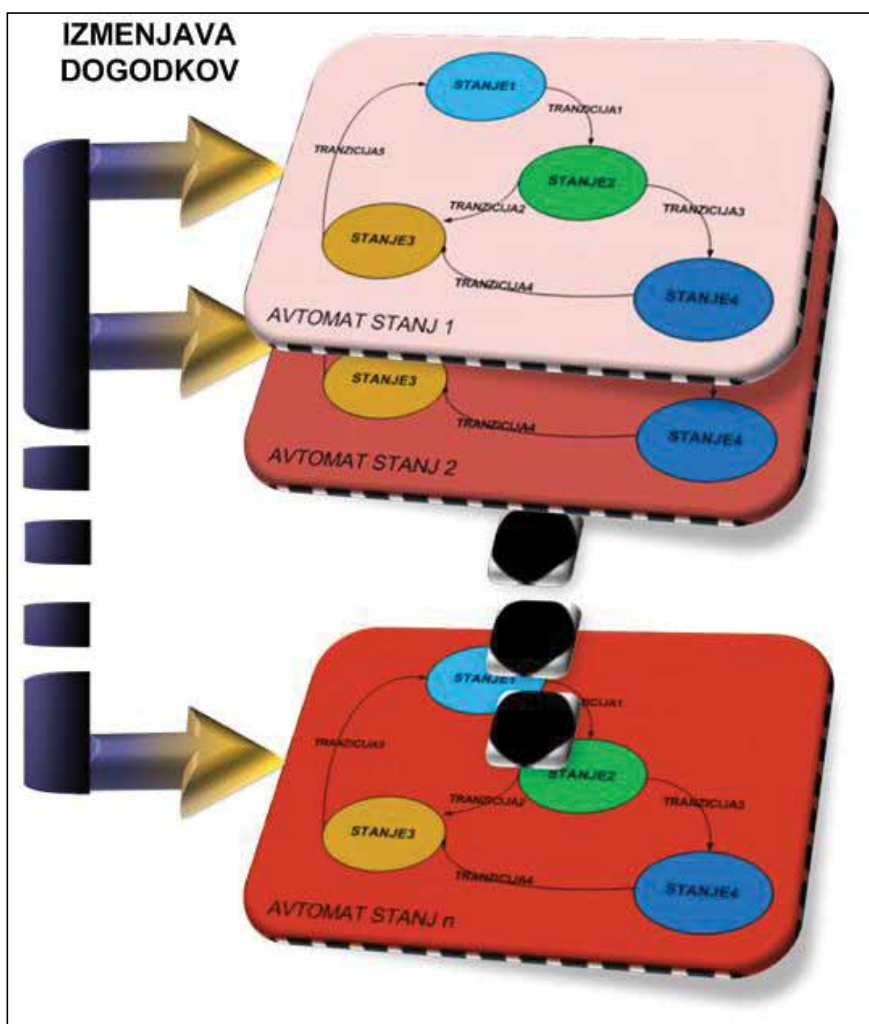
[4] rešimo problem nerazumnega delovanja in nadaljnje probleme pri izgradnji programske kode in dokumentacije.

V nadaljevanju so opisani osnovno delovanje avtomatskih drsni vrat, generator giba ter urejanje kode v končnem avtomatu. Končni avtomat zajema dve vrsti urejanja, to sta grafični način s tabelami in oblikovanje v programskem jeziku C. Grafični način je namenjen za lažje razumevanje delovanja kode za osebo, ki



Slika 1. Prototipna krmilna enota [6]

Franc Hanžič, univ. dipl. inž., Doorson, d. o. o., Maribor, prof. dr. Karel Jezernik, univ. dipl. inž., Univerza v Mariboru, FERi, Maribor, Slavko Cehner, univ. dipl. inž., Doorson, d. o. o., Maribor



Slika 2. Primer izvajanja končnih avtomatov v večopravilnem sistemu

se ne spozna na programske jezike. Tekstovno oblikovanje pa se nanaša na izdelovalce programske kode. Bistvo tega članka je prikazati povezljivost med grafičnim in tekstovnim načinom vodenja drsnih vrat in enostavnost izgradnje kompleksnega mehanizma, kot je generator giba. Programska koda v C jeziku postane urejena in berljiva z uporabo metode končnih avtomatov, s tem pa lažje prepoznavanje in izločevanje morebitnih programskih napak.

■ 2 Krmilna enota

Vodenje avtomatskih vrat poteka preko krmilne enote (slika 1), na katero so priključeni aktuatorji in senzorji, nameščeni v vratih. Sama krmilna enota ima vključene komunikacije (CAN – Controller Area Network in serijsko RS232). CAN-komunikacija zagotavlja komunikacijo z določenimi senzorji in univerzalnim modulom (vključuje požarno

in redundantno delovanje vrat). Redundantnost zagotavlja, da se vrata odprejo ob okvari krmilne enote. Povezljivost z osebnim računalnikom zagotavlja uporabniku ali serviserju enostavno nastavljanje parametrov, prenos programske kode, diagnostiko in nadzor. Jedro krmilne enote, 32-bitni mikrokrmilnik ARM Cortex M3 LPC1768 [2], upravlja delovanje avtomatskih vrat. Programska koda, ki poganja mikrokrmilnik, je izdelana v programskem jeziku C s pomočjo programskega urejevalnika [3].

■ 3 Programska koda

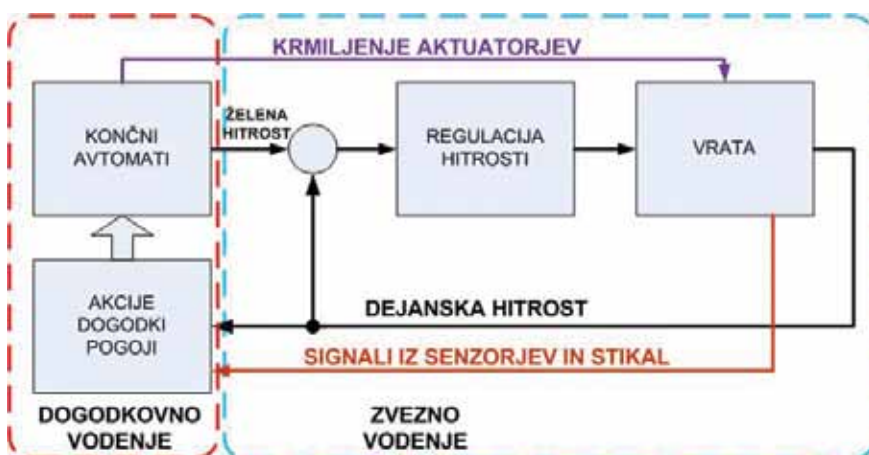


Slika 3. Primer končnega avtomata za semafor

Programska koda mikrokrmilnika je razdeljena na več opravil (komunikacije, varnost, generator giba, upravljanje z vrati, upravljanje z vhodi krmilnika, upravljanje s komandnim stikalom itd.) za ustrezno delovanje avtomatskih vrat. Vsako opravilo je oblikovano v končnem avtomatu, ki se mora izvajati vzporedno. Vzporedno izvajanje več opravil je nemogoče v mikrokrmilnikih, zato se mora vključiti operacijski sistem. Majhen operacijski sistem časovno preklaplja med opravili s časovnimi prekinitvami. Opravila predstavljajo končne avtomate, ki so medsebojno povezani s podatkovnim kanalom (slika 2). V primeru zahtevnih aplikacij pa se uporabljajo sistemi FPGA, ki vzporedno izvajajo več opravil brez operacijskega sistema.

■ 4 Končni avtomat

Tehnika programiranja z metodo končnega avtomata je že poznana. Teorija je preprosta, zajema stanja, tranzicije, dogodke in akcije. Prehodi med stanji se imenujejo tranzicije. Začetek tranzicije izvede dogodek, konec tranzicije pa predstavlja akcijo. Dogodek ima pogoje. Ob izpolnjenih pogojih se zgodi dogodek. Pogoji se lahko nanašajo na vhode krmilne enote ali spremenljivko v programu. Dogodek se lahko zgodi ob enem pogoju ali v kombinaciji z več pogoji z logičnimi funkcijami (AND, OR, NAND, >, <, = =; itd.). Akcija napoveduje, kaj se bo zgodilo ob določenem dogodku. Na semaforju imamo tri stanja, to so rdeča, rumena, zelena. Ta stanja so med seboj povezana s tranzicijami. Preprost semafor deluje s pomočjo štetja časa. Ko preteče čas, je prisoten dogodek »čas je potekel za rdečo luč« (izhod iz stanja), po dogodku pa nastopi akcija »preklopi na rume-



Slika 4. Algoritem vodenja vrat

no« (vstop v stanje). Dogodek in akcija predstavlja tranzicija, v primeru tranzicija1 (slika 3), povezavo med stanjem rdeča in rumena. Tranzicije 5, 6 in 7 pa predstavljajo števec, ki v aktivnem stanju narašča in šteje čas, v neaktivnem stanju pa se resetira.

5 Končni avtomat v avtomatskih vratih

Kot je že bilo povedano, imajo avtomatska vrata več končnih avto-

matov. V prispevku je predstavljeno delovanje vrat v končnem avtomatu, podrobneje pa generiranje giba (S-hitrostni profil). Programski del upravljanja z vrati daje zahtevo generatorju giba (generator giba v končnem avtomatu), ta pa mu sporoči, da je opravil nalogo. Generator giba v končnem avtomatu je bil zgrajen v programskem okolju Matlab/Simulink z orodjem »Stateflow«. Cilj naloge je izdelati gibanje

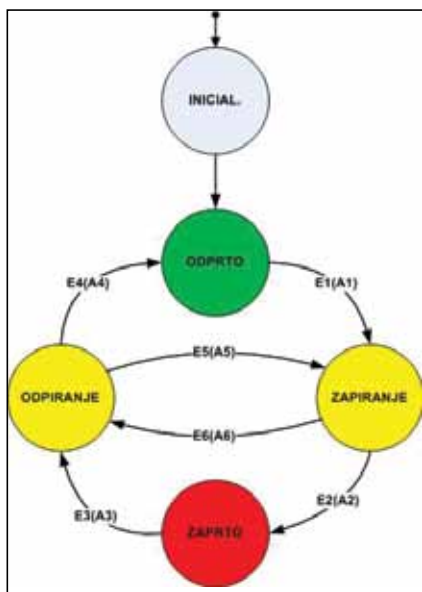
vrat z S-hitrostnim profilom, ki bi zamenjal obstoječi trapezni hitrostni profil. Splošni algoritem vodenja vrat (slika 4) je razdeljen na zvezno in nezvezno vodenje. Zvezno vodenje predstavlja regulacijo in upravljanje mikrokrmilniške periferije. Nezvezno vodenje pa predstavlja končne avtomate, ki vodijo in upravljajo vrata.

5.1 Končni avtomat za osnovno delovanje vrat

V osnovi imajo vrata pet stanj: inicializacija, odprto, zapiranje, zaprto in odpiranje. Ob vklopu avtomatskih vrat na napajanje (230VAC) gredo ta v inicializacijo, kjer se ponastavi parametri vrat (hitrosti, čas odprtosti, zaklep itd.) in dolžine giba. Pred ponastavitvijo dolžine giba vrata izvedejo cikel zapiranja in odpiranja. Pri končnih legah se krila vrat naslonijo na odbijač in s tem krmilnik zazna končno lego (meritev toka ali pa hitrost nenadno pade), tako se v pomnilnik vpiše dejanski položaj. Iz podatka položaja zaprtosti in

Tabela 1. Tranzicijska tabela

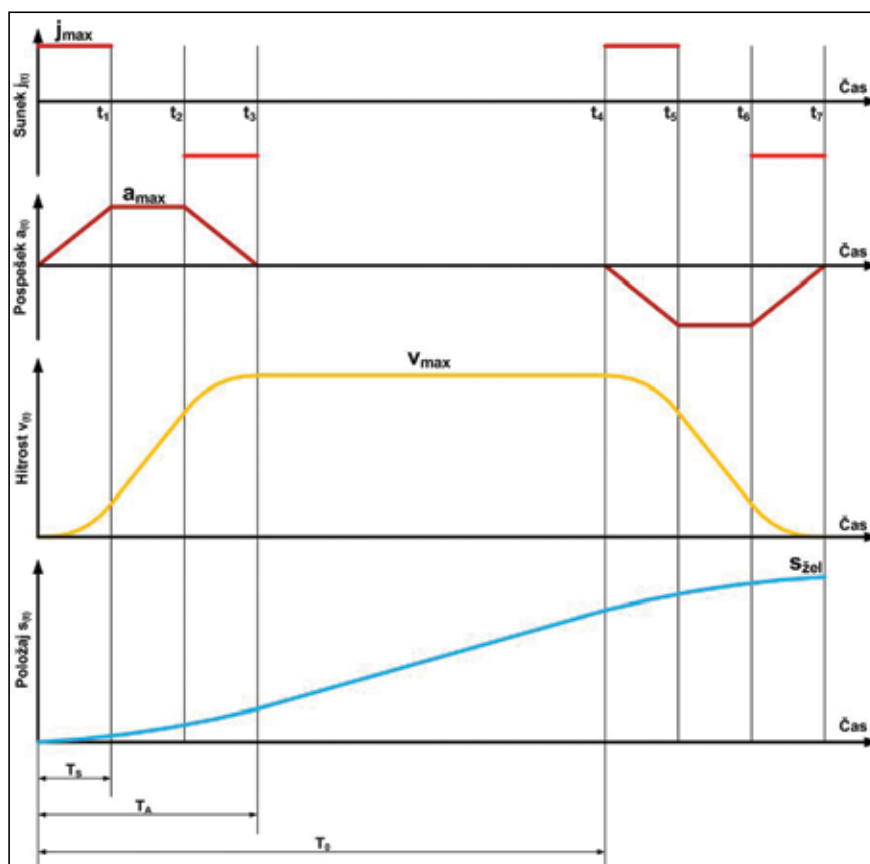
DOGODKI	POGOJI	OPIS DOGODKA				
E1	Meritev časa, požarni signal, signal zaklepa itd.	Pretekel je čas odprtosti, prisoten požar, zaklep vrat				
E2	Meritev položaja	Vrata dosegla končni položaj pri zapiranju				
E3	Signal senzorja prisotnosti, signal komandnega stikala itd.	Prisotna oseba, sprememba režima na odprto				
E4	Meritev položaja	Vrata dosegla končni položaj pri odpiranju				
E5	Tipanje toka na motorju, požarni signal itd.	Vrata blokirana, prisoten požar				
E6	Tipanje toka na motorju, signal senzorja prisotnosti itd.	Vrata blokirana, prisotna oseba				
AKCIJE	POGOJI	OPIS AKCIJE				
A1	Stanje ODPRTO in dogodek E1	Zapri vrata				
A2	Stanje ZAPIRANJE in dogodek E2	Ustavi vrata				
A3	Stanje ZAPRTO in dogodek E3	Odpri vrata				
A4	Stanje ODPIRANJE in dogodek E4	Ustavi vrata				
A5	Stanje ODIRANJE in dogodek E5	Zapri vrata				
A6	Stanje ZAPIRANJE in dogodek E6	Odpri vrata				
TRANZICIJSKA TABELA						
DOGODKI STANJA	E1	E2	E3	E4	E5	E6
ODPRTO	A1/ ZAPIRANJE	-	-	-	-	-
ZAPIRANJE	-	A2/ ZAPRTO	-	-	-	A6/ ODPIRANJE
ZAPRTO	-	-	A3/ ODPIRANJE	-	-	-
ODPIRANJE	-	-	-	A4/ ODPRTO	A5/ ZAPIRANJE	-



Slika 5. Končni avtomat pri avtomatskih drsnih vratih

odprtosti se izračuna dolžina giba kril. Po uspešni inicializaciji gredo vrata v stanje odprto. Ko poteče čas ali je izpolnjen kakšen drug pogoj (zaklep, požarni signal itd.), gredo v stanje zapiranje. Tranzicija iz stanja zapiranje v stanje zaprto se izvede ob dosegu končnega zaprtega položaja. Ob prisotnosti osebe (ali drug pogoj) se izvede tranzicija med stanjem zaprto in stanjem odpiranje. Ob dosegu končnega odprtega položaja se izvede tranzicija med stanjem odpiranje in stanjem odprto. Tako se cikel stanj ponavlja ob izpolnjenih pogojih. Opisano delovanje je prikazano v tabeli tranzicij (tabela 1). Princip je prikazan kot osnovni in ne predstavlja enakega delovanja v komercialnih avtomatskih drsnih vratih. Popolno delovanje vključuje še dodatna stanja in dodatne tranzicije med nekaterimi stanji.

Tabelo tranzicij predstavimo v grafični obliki (slika 5), ki jo lahko primerjamo z delovanjem semaforja (slika 3). Programski del upravljanja z vrati komunicira z generatorjem giba. Upravljalni del daje zahteve drugim končnim avtomatom, ti pa mu oddajajo odgovore o njegovih stanjih.



Slika 6. Potek posameznih referenčnih veličin v odvisnosti od časa – S-profil hitrosti (legenda k sliki - spodaj)

\hat{j}_{\max} – maksimalni sunek	T_s – čas sunka
\hat{a}_{\max} – maksimalni pospešek	T_A – čas pospeševanja
v_{\max} – maksimalna hitrost	T_0 – čas do zaustavljanja
s_{zel} – zeleni položaj	

■ 5.2 Končni avtomat za generacijo giba

Generator giba daje referenčne vrednosti (pospešek, hitrost in položaj) za izvedbo odpiranja ali zapiranja vrat. Trenutna tehnika uporablja referenčno vrednost hitrosti, ki s pomočjo regulatorja regulira trenutno hitrost z dejansko (slika 4). Končni avtomat generatorja prejema opravila (akcije), kot so odpri, zapri, ustavi in oddajniku vrne sporočilo končal z odprtjem, zaprtjem ali z zaustavitvijo (dogodki). Generator je lahko izveden v več oblikah, to so trapezna, zvončasta, S, \sin^2 itd. Vsaka izmed metod ima svoje prednosti in slabosti. Trenutni končni avtomat upora-

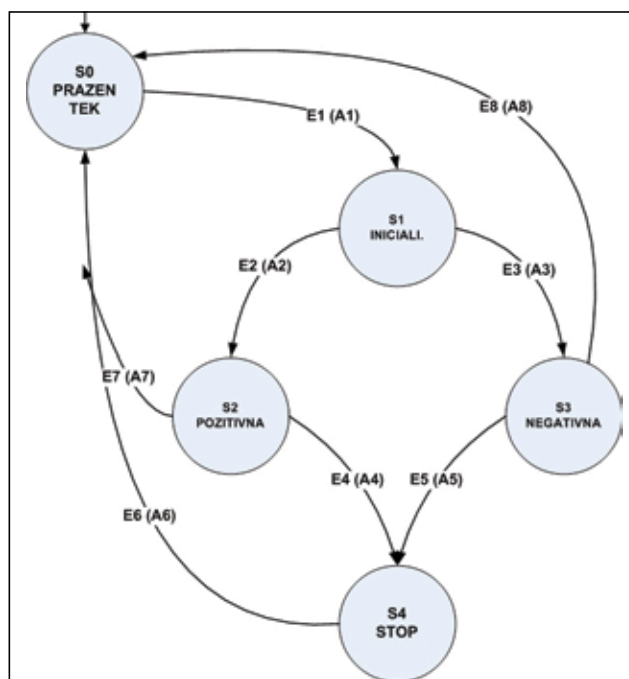
blja generator trapezne hitrosti in je enostaven za izdelavo. Vendar ima trapezna oblika hitrosti nenadne sunke (nezveznost med prehodi hitrosti), povzročata dodatna nihanja in večjo potrošnjo električne energije pri vodenju vrat. Za rešitev nezveznosti in drugih negativnih lastnosti bo uporabljen generator S-hitrostnega profila, ki ima zveznost med prehodi. Za primerjavo s trapeznim profilom ima S-profil zaokrožitve pri spremembi hitrosti in s tem zveznost med prehodi. Profil delimo na 7 stanj (slika 6), ki se bodo uporabila v končnem avtomatu. Generator izdelala profil po podanih zahtevanih vhodnih parametrih (maksimalni sunek, pospešek, hitrost in razdaljo giba). S sunkom se določa intenzivnost prehodov pri hitrosti. Večji je sunek, bolj so zaostreni prehodi. Profil ima sedem prehodov, ki so odvisni od časa

$a_{(t)} = \begin{cases} j_{\max} \cdot t & 0 < t \leq t_1 = T_S \\ a_{\max} & t_1 < t \leq t_2 = T_A - T_S \\ -j_{\max} \cdot (t - t_2) + a_{\max} & t_2 < t \leq t_3 = T_A \\ 0 & t_3 < t \leq t_4 = T_0 \\ -j_{\max} \cdot (t - t_4) & t_4 < t \leq t_5 = T_0 + T_S \\ -a_{\max} & t_5 < t \leq t_6 = T_0 + T_A - T_S \\ j_{\max} \cdot (t - t_6) - a_{\max} & t_6 < t \leq t_7 = T_0 + T_A \end{cases}$ <p>1)</p>	$T_S = \frac{a_{\max}}{j_{\max}}$ $T_A = \frac{v_{\max}}{a_{\max}} + T_S$ $T_0 = \frac{s_{zel}}{v_{\max}}$ <p>4)</p>
$v_{(t)} = \begin{cases} \frac{1}{2} \cdot j_{\max} \cdot t^2 & 0 < t \leq t_1 = T_S \\ a_{\max} \cdot (t - t_1) + v_{(t_1)} & t_1 < t \leq t_2 = T_A - T_S \\ -\frac{1}{2} \cdot j_{\max} \cdot (t - t_2)^2 + a_{\max} \cdot (t - t_2) + v_{(t_2)} & t_2 < t \leq t_3 = T_A \\ v_{(t_3)} & t_3 < t \leq t_4 = T_0 \\ -\frac{1}{2} \cdot j_{\max} \cdot (t - t_4)^2 + v_{(t_4)} & t_4 < t \leq t_5 = T_0 + T_S \\ -a_{\max} \cdot (t - t_5) + v_{(t_5)} & t_5 < t \leq t_6 = T_0 + T_A - T_S \\ \frac{1}{2} \cdot j_{\max} \cdot (t - t_6)^2 - a_{\max} \cdot (t - t_6) + v_{(t_6)} & t_6 < t \leq t_7 = T_0 + T_A \end{cases}$ <p>2)</p>	$T_S = \frac{a_{\max}}{j_{\max}}$ $T_A = \frac{-a_{\max}^2 + \sqrt{a_{\max}^4 + 4 \cdot a_{\max} \cdot j_{\max}^2 \cdot s_{zel}}}{2 \cdot a_{\max} \cdot j_{\max}} + T_S$ $T_0 = T_A$ <p>5)</p>
$s_{(t)} = \begin{cases} \frac{1}{6} \cdot j_{\max} \cdot t^3 & 0 < t \leq t_1 = T_S \\ \frac{1}{2} \cdot a_{\max} \cdot (t - t_1)^2 + v_{(t_1)} \cdot (t - t_1) + s_{(t_1)} & t_1 < t \leq t_2 = T_A - T_S \\ -\frac{1}{6} \cdot j_{\max} \cdot (t - t_2)^3 + \frac{1}{2} \cdot a_{\max} \cdot (t - t_2)^2 + v_{(t_2)} \cdot (t - t_2) + s_{(t_2)} & t_2 < t \leq t_3 = T_A \\ v_{(t_3)} \cdot (t - t_3) + s_{(t_3)} & t_3 < t \leq t_4 = T_0 \\ -\frac{1}{6} \cdot j_{\max} \cdot (t - t_4)^3 + v_{(t_4)} \cdot (t - t_4) + s_{(t_4)} & t_4 < t \leq t_5 = T_0 + T_S \\ -\frac{1}{2} \cdot a_{\max} \cdot (t - t_5)^2 + v_{(t_5)} \cdot (t - t_5) + s_{(t_5)} & t_5 < t \leq t_6 = T_0 + T_A - T_S \\ \frac{1}{6} \cdot j_{\max} \cdot (t - t_6)^3 - \frac{1}{2} \cdot a_{\max} \cdot (t - t_6)^2 + v_{(t_6)} \cdot (t - t_6) + s_{(t_6)} & t_6 < t \leq t_7 = T_0 + T_A \end{cases}$ <p>3)</p>	$T_S = \frac{a_{\max}}{j_{\max}}$ $T_A = 2 \cdot T_S$ $T_0 = T_A$ <p>6)</p>

in so označeni od t1 do t7.

Maksimalni sunek, pospešek, hitrost in položaj so parametri, ki se določijo glede na lastnosti vrat. Maksimalni sunek \hat{j}_{\max} se navezuje na strmino pospeška. Pri S-hitrostnem profilu je pospešek trapezne oblike in posledično trapezna oblika hitrosti. Sunek pri drsnih vratih se uporablja za nivo zveznosti med različnimi hitrostmi. Želja je, da so vrata pri odpiranju zelo hitra in da pri prehodu iz mirovanja v premikanje ni velike razlike hitrosti, vzrok so velike mehanske obremenitve in vibracije. S stopnjo sunka lahko določamo, da na začetku vrata postopno zvišujejo hitrost. Maksimalni pospešek \hat{a}_{\max} se navezuje na

strmino hitrosti in se oceni na maso vrat. Večji je pospešek, hitreje vrata dosežejo zeleno hitrost. Pri odpiranju je zaželen večji pospešek kot pri zapiranju. Pri pospešku se morajo upoštevati meje glede na sposobnost mehanizma vrat (motor). Motor, ki poganja vrata, ne zmore izvesti giba s katerokoli vrednostjo pospeška. Maksimalni pospešek



Slika 7. Končni avtomat generatorja giba

Tabela 2. Vhodi in izhodi generatorja in opis tranzicij

Vhod	Tip podatka	Opis
MAXPOS	število	Vhod za določitev končnega položaja [m]
SETTIME	število	Vhod za določitev časovnega intervala T_i v katerem se izvaja avtomat stanj ($T_i < 1e-3$)
ACT_POS	število	Vhod za dejanski položaj
ACT_VEL	število	Vhod za dejansko hitrost
MAX_VELP	število	Vhod za določitev maksimalne hitrosti pozitivnega hitrostnega profila [m/s]
MAX_VELN	število	Vhod za določitev maksimalne hitrosti negativnega hitrostnega profila [m/s]
MAXACC_P	število	Vhod za določitev maksimalnega pospeška pozitivnega hitrostnega profila [m/s ²]
MAXACC_N	število	Vhod za določitev maksimalnega pospeška negativnega hitrostnega profila [m/s ²]
JERK_P	število	Vhod za določitev maksimalnega sunka pozitivnega hitrostnega profila [m/s ³]
JERK_N	število	Vhod za določitev maksimalnega sunka negativnega hitrostnega profila [m/s ³]
PROMACHINE_IN	število	Vhod za upravljanje avtomata stanj (generator profila), komunikacija med različnimi drugimi avtomati stanj »0« - ne naredi ničesar »1« - izvedi pozitivni hitrostni profil »2« - izvedi negativni hitrostni profil »3« - izvedi zaustavljanje
JERK_S	število	Vhod za določitev maksimalnega sunka pri zaustavljanju [m/s ³]
MAXACC_S	število	Vhod za določitev maksimalnega pospeška pri zaustavljanju [m/s ²]

Izhod	Tip podatka	Opis
ACC	število	Izhod iz generatorja profila, kateri zajema potek pospeška [m/s ²]
POS	število	Izhod iz generatorja profila, kateri zajema potek položaja [m]
VEL	število	Izhod iz generatorja profila, kateri zajema potek hitrosti [m/s]
PROMACHINE_OUT	število	Izhod iz generatorja profila, kateri zajema informacije o prehodih med stanji
STATUS	število	Izhod iz generatorja profila, kateri zajema informacije o izvedbi hitrostnega profila, komunikacija med različnimi drugimi avtomati stanj »0« - v izvajanju »1« - izvedel pozitivni hitrostni profil »2« - izvedel negativni hitrostni profil »3« - izvedel zaustavljanje

se navezuje na sposobnost motorja (na dovoljeni tok motorja). Maksimalna hitrost v_{max} je omejena na maso vrat, ki se navezuje na varnost pred trki ovir (osebe, predmeti). Pri težjih vratih mora biti zagotovljena manjša hitrost kot pri lažjih vratih. Težja vrata z višjo hitrostjo imajo večjo vztrajnost, ki je nevarna za ljudi ob morebitnih trkih. Hitrost je nastavljiva s strani uporabnika z zgornjo omejitvijo 0.8 m/s, oziroma ocenjeno omejeno hitrost odvisno od mase vrat.

Za izvedbo profila je potrebno poznati enačbe za pospešek (1), hitrost (2) in položaj (3). Potrebno je tudi izračunati časovne intervale (T_S , T_A , T_0), ki so odvisni od dolžine giba, maksimalnega pospeška, sunka in hitrosti. Za izračun časovnih intervalov so vključene različne enačbe in pogoji, ki izbirajo način izračuna

časovnega intervala. Z upoštevanjem pogojev se časovni intervali izračunajo z naslednjimi enačbami (4, 5, 6).

Predstavljene enačbe zajemajo samo pozitivno generiranje hitrosti. Končni avtomat avtomatskih vrat vključuje še negativno hitrost in stop funkcijo. Izvedba S-hitrostnega profila je izdelana v programskem orodju Matlab/Simulink in ima naslednje vhode in izhode (tabela 2). Zajema ga stanje čakanja (PRAZNI TEK), stanje začetne izračune in nastavitve generatorja profila (INICIALIZACIJA), stanje pozitivnega hitrostnega profila (POZITIVNA), stanje negativnega hitrostnega profila (NEGATIVNA) in stanje zaustavljanja (STOP), skupaj 5 stanj. Stanja so med seboj povezana s prehodi (tranzicijami), ki imajo določen pogoj (dogodek) za dovoljenje prehoda iz eno v drugo stanje in s tem se izvrši določena akcija.

Končni avtomat ima naslednje tranzicije (pogoji, dogodki, akcije) in stanja (tabela 3).

Končni avtomat s pomočjo tranzicijske tabele predstavimo v grafični obliki (slika 7).

■ 6 Simulacijski eksperimenti

Generator hitrostnega profila je bil uspešno preizkušen v okolju Matlab/Simulink. Izvedlo se je odpiranje in zapiranje in zaustavljanje med gibanjem. Predstavljeni test je zajel odpiranje in zapiranje z naslednjimi parametri (slika 8):

■ 7 Oblika končnega avtomata v programskem jeziku C

Programska koda za krmilnik vrat je napisana v programskem jeziku C. Tako je potrebno grafično obliko končnih avtomatov pretvoriti v ustrezno obliko v tem jeziku [4]. V nadaljevanju je predstavljen C-jezik generatorja giba, oblikovanega v končnem avtomatu. Program prikazuje preklon stanj in tranzicije in ne zajema kode za izvedbo profila. Najprej se definirajo tranzicije (dogodki) in stanja (slika 9). Pogoji so izvedeni z IF-funkcijo in logičnimi operacijami. Pogoji so lahko oblikovani v matrični obliki ali pa se definirajo v IF-stavkih. Pri izpolnitvi pogoja se izvede dogodek, dogodku pa sledi akcija. Dogodek in akcijo predstavlja tranzicija, ki povezuje dve stanji. Tranzicija predstavlja funkcijo SWITCH v C-jeziku, ki preklaplja med stanji (slika 10).

Obstajajo programska orodja, ki podpirajo programiranje v grafičnem načinu z metodo končnih avtomatov »StateChart« in tako generira programsko kodo v C jeziku [5].

■ 8 Zaključek

Tehnologija ponuja vedno več modernjših metod krmiljenja, komunikacij in regulacij. Današnji mikrokrmilniki so zelo bogato opremljeni z zmogljivimi jedri in bogato periferijo. Vključitev teh metod v sistem izdelka pa potrebuje kompleksno pro-

Tabela 3. Tabela tranzicij z opisi

Pogoji	Tip podatka	Opis						
PROMACHINE_IN	število	Vhod v avtomat stanj »PROFILE MACHINE INPUT«, s katerim določimo operacijo: »0« - ne naredi ničesar »1« - izvedi pozitivni hitrostni profil »2« - izvedi negativni hitrostni profil »3« - izvedi zaustavljanje						
STATEINIT	logična spremenljivka	Spremenljivka, katera nosi informacijo o izvedbi začetnih nastavitev »0« - še niso nastavljeni »1« - so nastavljeni						
NEGATIVEM	logična spremenljivka	Spremenljivka, katera nosi informacijo o stanju negativnega hitrostnega profila »0« - še ni končal negativnega profila »1« - je končal negativni profil						
POSITIVEM	logična spremenljivka	Spremenljivka, katera nosi informacijo o stanju pozitivnega hitrostnega profila »0« - še ni končal pozitivnega profila »1« - je končal pozitivni profil						
STOPPED	logična spremenljivka	Spremenljivka, katera nosi informacijo o stanju zaustavljanja »0« - še ni končal zaustavljanja »1« - se je zaustavil						
Dogodki	Pogoji	Opis						
E1	POSITIVEM==0&& PROMACHINE_IN==1 NEGATIVEM==0&& PROMACHINE_IN==2	Dodeljena zahteva za gib vrat						
E2	STATEINIT==1&&PROMACHINE_IN==1	Opravljen inicializacija in dodeljena zahteva za odpiranje vrat						
E3	STATEINIT==1&& PROMACHINE_IN==2	Opravljen inicializacija in dodeljena zahteva za zapiranje vrat						
E4	PROMACHINE_IN==3	Zaznana ovira						
E5	PROMACHINE_IN==3	Zaznana ovira ali prisotna oseba						
E6	STOPPED==1	Vrata so se ustavila						
E7	POSITIVEM==1	Vrata so odprta						
E8	NEGATIVEM==1	Vrata so zaprta						
Akcije	Pogoji	Opis						
A1	S0&&E1	Opravi inicializacijo						
A2	S1&&E2	Opravi pozitivni profil hitrosti						
A3	S1&&E3	Opravi negativni profil hitrosti						
A4	S2&&E4	Opravi STOP funkcijo v pozitivnem profilu						
A5	S3&&E5	Opravi STOP funkcijo v negativnem profilu						
A6	S4&&E6	Vrni se v pripravljenost iz STOP funkcije						
A7	S2&&E7	Vrni se v pripravljenost iz pozitivnega profila hitrosti						
A8	S3&&E8	Vrni se v pripravljenost iz negativnega profila hitrosti						
Stanja	Opis							
S0	Stanje čakanja (IDLE)							
S1	Stanje začetnih izračunov in nastavitev osnovnih parametrov (INIT)							
S2	Stanje za izvedbo pozitivnega hitrostnega profila (POSITIVE)							
S3	Stanje za izvedbo negativnega hitrostnega profila (NEGATIVE)							
S4	Stanje za zaustavljanje (STOP)							
Tabela tranzicij								
Dogodki Stanja	E1	E2	E3	E4	E5	E6	E7	E8
S0	A1/S1	-	-	-	-	-	-	-
S1	-	A2/S2	A3/S3	-	-	-	-	-
S2	-	-	-	A4/S4	-	-	A7/S0	-
S3	-	-	-	-	A5/S4	-	-	A8/S0
S4	-	-	-	-	-	A6/S0	-	-

gramsko kodo, ki pa jo mora izdelati proizvajalec oz. projektant. Čeprav današnja programska orodja ponujajo različne rešitve za enostavnejšo izdelavo kode, se problem stopnjuje zaradi nasičenega trga (konkurenca, vedno večje zahteve kupcev itd.) pri hitrem razvoju in izdelavi. Za izdelavo programske kode je potrebno uporabiti metodo, ki je enostavna za razumevanje, dograjevanje funkcij in vzdrževanja. Metoda s končnimi avtomati je ena izmed rešitev za preglednost, razumevanje, vzdrževanje itd. Grafična oblika končnih avtomatov je pomembna za izdelavo dobre dokumentacije za vzdrževanje, odpravljanje morebitnih napak in izpolnitve programske kode.

Z raziskovalno nalogo smo izbrali in preuredili matematični model generatorja giba z S-obliko hitrosti za avtomatska drsna vrata. Matematični model tako vključuje odpiranje in zapiranje vrat, nastavitve obeh (zapiranje/odpiranje) sunkov, pospeškov, hitrosti. Prav tako se lahko vpiše želena vrednost dolžine giba, ki ga bo ustvaril generator. V generator se je prav tako vključila funkcija STOP, ki hitro zaustavi gibanje vrat. Z ustrezno obliko matematičnega modela se je ta uporabil za izvedbo generatorja giba v končnem avtomatu. S tem je generator giba postal pregleden in razumljiv. Generator je izdelan vključno z izhodi za generacijo referenčnih vrednosti (pospešek, hitrost, položaj) in vhodi za nastavitve oblike giba še z ukaznim vhodom (odpri, zapri, ustavi), izhodom stanja (odprl, zaprl, zaustavil) in izhodom o internem stanju samega generatorja, ki pomaga pri analizi njegovega delovanja. Še ena prednost končnega avtomata je, da lahko s pomočjo spremenljivke spremljamo potek stanj in tako analiziramo pravilno delovanje programa. Generator je bil uspešno izveden v simulacijskem okolju Matlab/Simulink/Stateflow in tudi preizkušen na DSP-sistemu s sinhronskim motorjem. Prav tako se je izdelal osnovni okvir (z obliko končnega avtomata) generatorja giba v C-jeziku, s katerim želimo prikazati povezanost med grafičnim in tekstovnim načinom. S simulacijskimi rezultati je potrjeno pravilno delovanje genera-

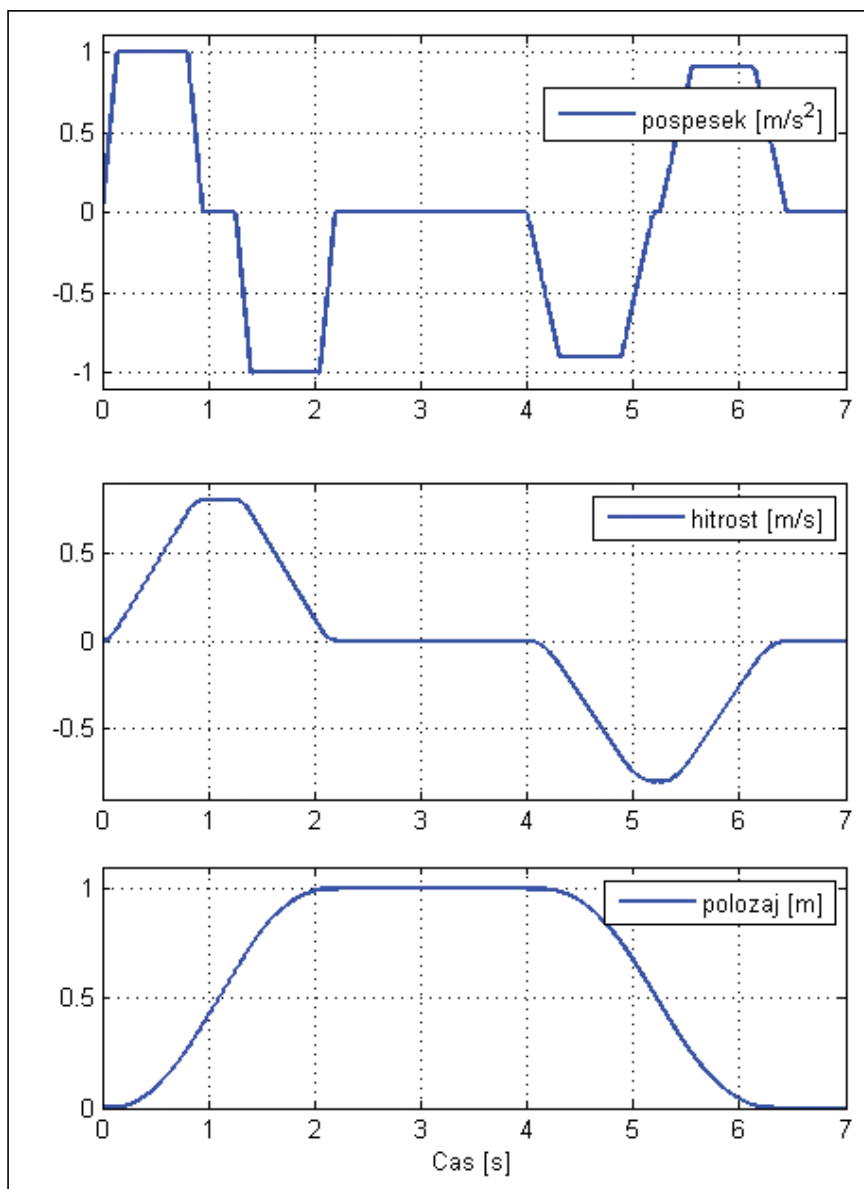
torja giba z S-obliko hitrosti in tako uspešno zaključen zadani cilj tudi z eksperimentom na DSP-sistemu.

9 Potrditve

Operacijo iz Evropskega socialnega sklada delno financira Evropska unija. Operacija se izvaja v okviru Operativnega programa razvoja človeških virov za obdobje 2007–2013, 1. razvojne prioritete: Spodbujanje podjetništva in prilagodljivosti, prednostne usmeritve 1.1: Strokovnjaki in raziskovalci za konkurenčnost podjetij.

Viri

- [1] Proizvajalec avtomatskih drsnih vrat www.doorson.si.
- [2] Proizvajalec mikrokrmilnikov www.nxp.com.
- [3] Programsko orodje za programiranje mikrokrmilnikov www.keil.com.
- [4] Ferdinand, W. et al. (2006): Modeling Software with Finite State Machines – A Practical Approach. Auerbach Publications.
- [5] Programsko orodje za grafično programiranje mikrokrmilnikov v končnem avtomatu <http://www.iar.com/website1/1.0.1.0/371/1/>.
- [6] Gačnik, V., Hanžič, F., Cehner, S., Majcen, G.: Prototipni krmilnik EC4 za avtomatska drsna vrata, interni dokument podjetja Doorson, d. o. o., marec 2011.
- [7] Macfarlane, S., Croft, E. A (2003): Jerk-Bounded Manipulator Trajectory Planning: Design for Real-Time Applications. IEEE Transactions on Robotics and Automation, Vol. 19, No. 1, 2003. PP. 42–52.
- [8] Li, H. Z. et al. (2007): A New Motion Control Approach for Jerk and Transient Vibration Suppression.



Slika 8. Odzivi pri odpiranju in zapiranju (legenda k sliki - spodaj)

začetek odpiranja pri času 0 s
 začetek zapiranja pri času 4 s
 dolžina odpiranja/zapiranja: 1 m
 maksimalna hitrost (pozitivni/negativni): 0,8 m/s
 maksimalni pospešek (pozitivni): 1 m/s²
 maksimalni pospešek (negativni): 0,9 m/s²
 maksimalni sunek (pozitivni): 7 m/s³
 maksimalni sunek (negativni): 3 m/s³


```

/**Stanja, Spremenljivke, Akcije in dogodki*****/
/*****STANJA*****/
#define IDLE_S0          0x00      //Stanje IDLE
#define INIT_S1          0x01      //Inicijalizacija hit. prof.
#define POSITIVE_S2     0x02      //Gen. poz. hit. profila
#define NEGATIVE_S3     0x03      //Gen. neg. hit. profila
#define STOP_S4         0x04      //Hitro ustavljanje

/*****Spremenljivke*****/
extern int PROMACHINE_IN;        //Vhod avtomata stanj
extern int POSITIVEM;           //Informacija o izvedbi poz. profila
extern int NEGATIVEM;          //Informacija o izvedbi neg. profila
extern int STOPPED;            //Informacija o izvedbi zaustavitve
extern int STATEINIT;          //Informacija o izvedbi iniciali.

/*****DOGODKI*****/
#define E1 (POSITIVEM==0&&PROMACHINE_IN==0|
           NEGATIVEM==0&&PROMACHINE_IN==2&&STATE==0) //Dogodek E1
#define E2 (STATEINIT==1&&PROMACHINE_IN==1&&STATE==1) //Dogodek E2
#define E3 (STATEINIT==1&&PROMACHINE_IN==2&&STATE==1) //Dogodek E3
#define E4 (PROMACHINE_IN==3&&STATE==2) //Dogodek E4
#define E5 (PROMACHINE_IN==3&&STATE==3) //Dogodek E5
#define E6 (STOPPED==1&&STATE==4) //Dogodek E6
#define E7 (POSITIVEM==1&&STATE==2) //Dogodek E7
#define E8 (NEGATIVEM==1&&STATE==3) //Dogodek E8

```

Slika 9. Definicije končnega avtomata v C-jeziku

```

#include "def.h"

/*****TRANZICIJE*****/
if (E1) STATE=INIT_S1; //Trancicija med S0 in S1
if (E2) STATE=POSITIVE_S2; //Trancicija med S1 in S2
if (E3) STATE=NEGATIVE_S3; //Trancicija med S1 in S3
if (E4) STATE=STOP_S4; //Trancicija med S2 in S4
if (E5) STATE=STOP_S4; //Trancicija med S3 in S4
if (E6) STATE=IDLE_S0; //Trancicija med S4 in S0
if (E7) STATE=IDLE_S0; //Trancicija med S2 in S0
if (E8) STATE=IDLE_S0; //Trancicija med S3 in S0

/*****STANJA*****/
switch (STATE)
{
    case IDLE_S0: //Stanje IDLE_S0
        //Ne naredi nicesar
        break;

    case INIT_S1: //Stanje INIT_S1
        //Izracun spremenljivk
        break;

    case POSITIVE_S2: //Stanje POSITIVE_S2
        //Izvedi poz. hit. profil
        break;

    case NEGATIVE_S3: //Stanje NEGATIVE_S3
        //Izvedi neg. hit. profil
        break;

    case NEGATIVE_S4: //Stanje STOP
        //Izvedi prisilno ustavljanje
        break;
}

```

Slika 10. Oblika generatorja giba v C-jeziku

Automatic sliding door control with state machine

Abstract: A reconstruction of the automatic sliding doors controller with an ARM Cortex M3 microcontroller [2] has opened a solution to implement better software algorithm. Implementation with the new algorithm will improve door dynamics and door durability. The new generator will improve door functionality but a program code must be implemented in the microcontroller with software bug probability. Solution of finite automata is an opportunity to implement program code without pain and code development with the low level of potential bugs. The research work is based on the motion generator with a S velocity form, the use of finite-state machines, the motion generator use at the automatic sliding door, a simulation experiments in Matlab / Simulink / Stateflow with the results.

Keywords: software design, automatic sliding doors, finite state machines, microcontroller, motion generator

Popravek

V reviji Ventil 17/2011/2, april, so bili v članku z naslovom End-quality assessment of electrical motors based on the concept of virtual sensors – str. 148, avtorjev Pavle Boškoski, Janko Petrovčič, Bojan Musizza, Đani Juričić in Andrej Biček navedeni napačni akademski nazivi avtojev Janko Petrovčič in Bojan Musizza. Pravilno: dr. Janko Petrovčič, univ. dipl. inž in dr. Bojan Musizza, univ. dipl. inž. Za neljubo napako se avtorjema in bralcem iskreno opravičujemo.

Uredništvo