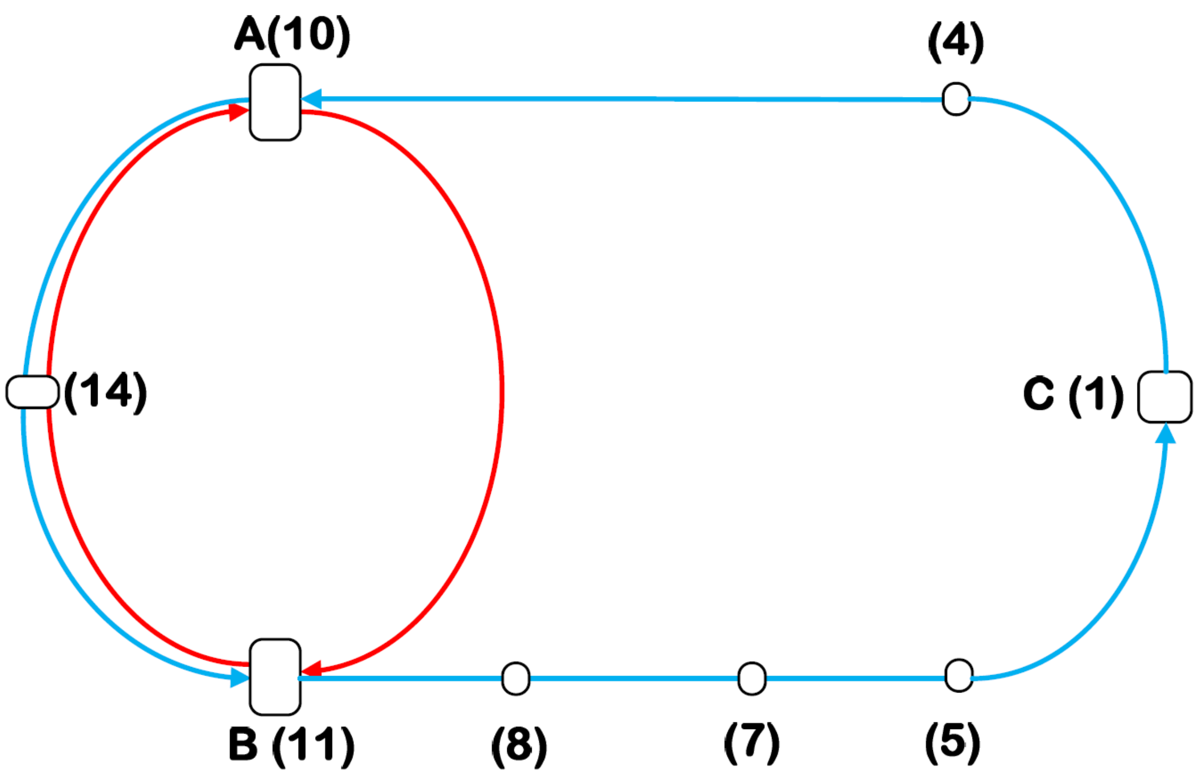
BARTULOVIĆ MIHOVIL,0036453342 ERDELIĆ TOMISLAV, 0036450291 PAVLOVIĆ ALEN, 0036452163 1.D_AUT AUTOMATIKA	FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA ZAGREB ZAVOD ZA AUTOMATIKU I RAČUNALNO INŽENJERSTVO **Sustavi s diskretnim događajima** 1. seminarski zadatak	5.6.2013

Opis zadatka

Neka se na pruzi prikazanoj na Slici 2 nalaze tri grada: A, B i C koji odgovaraju redom čvorovima 10, 11 i 1. Pojednostavljeni prikaz željezničke pruge koja povezuje ta tri grada prikazan je na Slici 1. Promatraju se dvije linije: crvena i plava. Crvena linija je kraća i povezuje gradove A i B te prometuje u smjeru kazaljke na satu. Plava linija povezuje sva tri grada te prometuje u smjeru suprotnom od kazaljke na satu. Vremena trajanja puta na pojedinim segmentima prikazana su u Tablici 1.



Slika 1. Isječak željezničke pruge

(10,14),(14,10)	(14,11),(11,14)	(10,11)	(11,8)	(8,7)	(7,5)	(5,1)	(1,4)	(4,10)
7	7	14	5	3	5	7	7	12

Tablica 1. Trajanje vožnje na bridovima

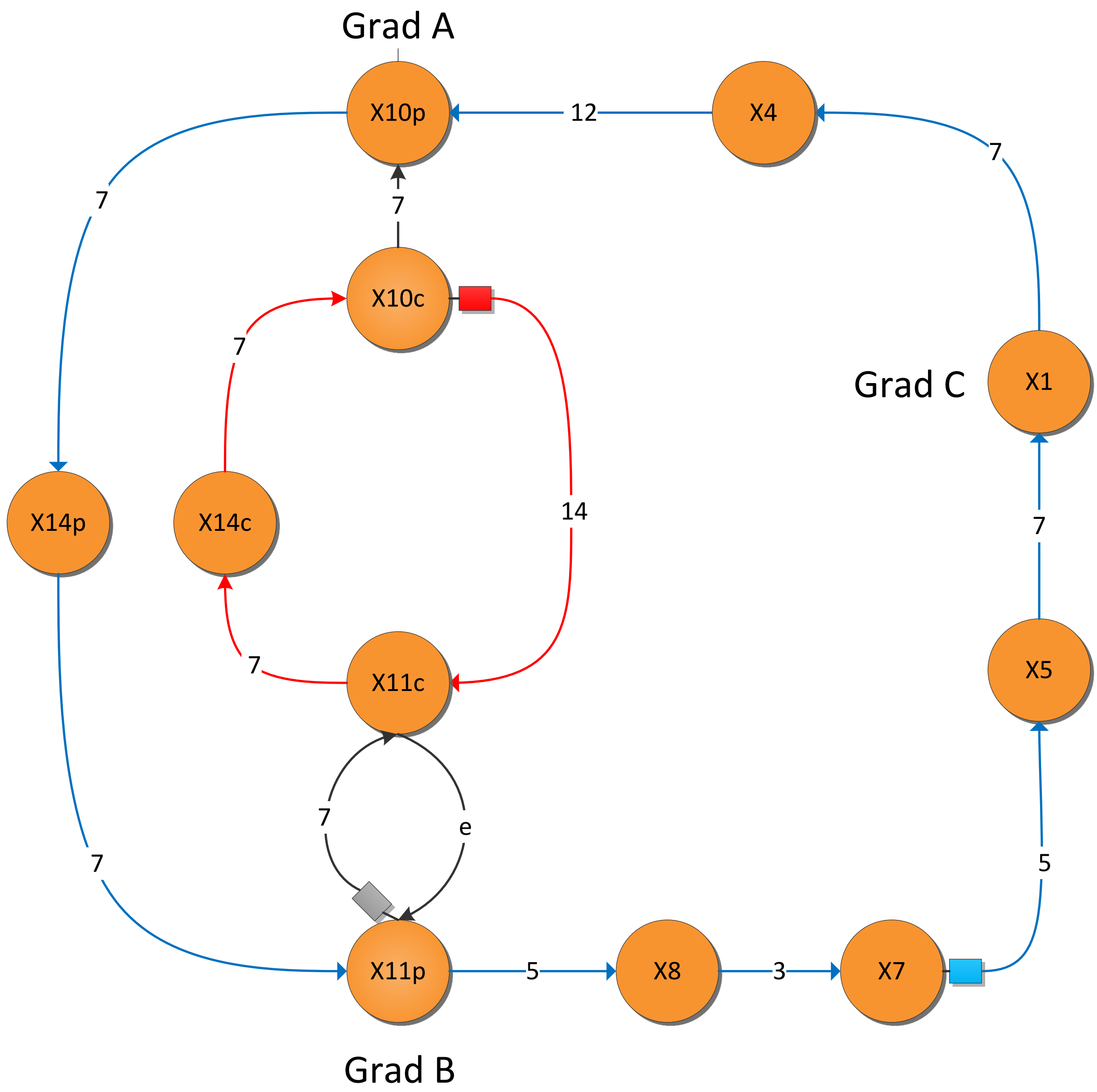
1. Neka su na raspolaganju dva vlaka (po jedan vlak na svakoj liniji). Korištenjem max-plus algebre potrebno je projektirati sustav upravljanja kojim se postiže što manji prosječni ciklus. Odrediti max-plus model za taj slučaj, početno stanje te prosječni ciklus.

2. Prema uputama u prethodnom poglavlju potrebno je definirati staze za svaki vlak koje će se zapisati u .cfg datoteku. Potom je potrebno u programskom jeziku Python (verzija $\geq$ 2.7.3) napisati program za upravljanje vlakovima. Varijable **vc** i **vs** su tipa rječnika s ključevima 'green' i 'blue'. Prije testiranja programa vlakovi će se ručno dovesti u poziciju koja je određena početnim uvjetom.

3. Ponoviti prethodna dva zadatka ako su na raspolaganju tri vlaka: jedan vlak na crvenoj liniji i dva vlaka na plavoj liniji. Budući da vlakovi nemaju implementiran sustav za sprečavanje sudara, potrebno je dodatno osigurati da između dva susjedna grada uvijek bude najviše jedan vlak. Varijable **vc** i **vs** su tipa rječnika s ključevima '*red*', '*green*' i '*blue*'.

1. Zadatak – rješenje

Slikom 2. prikazan je graf sustava dobiven iz opisa sustava danog zadatkom. Graf se sastoji od čvorova koji predstavljaju događaje i linija koje predstavljaju puteve s označenom težinom iznad njih. Sustav se sastoji od 11 događaja koji su dani skupom $X = \{x_1, x_4, x_5, x_7, x_8 x_{10p}, x_{11p}, x_{14p}, x_{10c}, x_{11c}, x_{14c}\}$.



Slika 2. Graf sustava

Na Slici 2. pomoću kvadratića označen je početni položaj vlakova u sustavu. Crveni kvadratić je početni položaj vlaka koji ide crvenom linijom, plavi kvadratić je početni položaj vlaka koji ide plavom linijom te sivi kvadratić je tu zbog sinkronizacije. Iz grafa možemo jednostavno zapisati jednadžbe sustava. Jednadžbe sustava u max-plus zapisu dane su izrazima (1-1).

$$
\begin{aligned}
x_1(k) &= 7 \otimes \mathrm{x}_5(\mathrm{k}) \\
x_4(k) &= 7 \otimes \mathrm{x}_1(\mathrm{k}) \\
x_5(k) &= 5 \otimes \mathrm{x}_7(\mathrm{k}-1) \\
x_7(k) &= 3 \otimes \mathrm{x}_8(\mathrm{k}) \\
x_8(k) &= 5 \otimes \mathrm{x}_{11\mathrm{p}}(\mathrm{k}) \\
x_{10p}(k) &= 12 \otimes \mathrm{x}_4(\mathrm{k}) \oplus 7 \otimes \mathrm{x}_{10\mathrm{c}}(\mathrm{k}) \\
x_{11p}(k) &= 7 \otimes \mathrm{x}_{14\mathrm{p}}(\mathrm{k}) \oplus \mathrm{e} \otimes \mathrm{x}_{11\mathrm{c}}(\mathrm{k}) \\
x_{14p}(k) &= 7 \otimes \mathrm{x}_{10\mathrm{p}}(\mathrm{k}) \\
x_{10c}(k) &= 7 \otimes \mathrm{x}_{14\mathrm{c}}(\mathrm{k}) \\
x_{11c}(k) &= 14 \otimes \mathrm{x}_{10\mathrm{c}}(\mathrm{k}) \oplus 7 \otimes \mathrm{x}_{11\mathrm{p}}(\mathrm{k}) \\
x_{14c}(k) &= 7 \otimes \mathrm{x}_{11\mathrm{c}}(\mathrm{k})
\end{aligned} \tag{1-1}
$$

Iz sustava jednadžbi (1-1) možemo pisati matrice $\boldsymbol{A_0}$ i $\boldsymbol{A_1}$.

$$
A_0 = \begin{bmatrix}
\varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
7 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & 12 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & e & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon
\end{bmatrix} \tag{1-2}
$$

$$
A_1 = \begin{bmatrix}
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & 14 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon
\end{bmatrix} \tag{1-3}
$$

gdje su elementi u matricama $\boldsymbol{A_0}$ i $\boldsymbol{A_1}$ zapisani idućim redom:

Stanje	Stupac/redak matrice
x_1	1
x_4	2
x_5	3
x_7	4
x_8	5
x_{10p}	6
x_{11p}	7
x_{14p}	8
x_{10c}	9
x_{11c}	10
x_{14c}	11

Tablica 2. Elementi u matricama

Za izračun matrice sustava $\boldsymbol{A}$ koristimo jednadžbu (1-4) :

$$\boldsymbol{A} = (\boldsymbol{E} \oplus \boldsymbol{A}_0^1 \oplus \boldsymbol{A}_0^2 \oplus \boldsymbol{A}_0^3 \oplus \ldots \oplus \boldsymbol{A}_0^{11}\,) \otimes \mathbf{A}_1, \tag{1-4}$$

$$A = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & 12 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 19 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 53 & \varepsilon & \varepsilon & 50 & \varepsilon & 57 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 50 & \varepsilon & \varepsilon & 47 & \varepsilon & 45 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 31 & \varepsilon & \varepsilon & 28 & \varepsilon & 35 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 45 & \varepsilon & \varepsilon & 42 & \varepsilon & 49 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 38 & \varepsilon & \varepsilon & 35 & \varepsilon & 42 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 21 & \varepsilon & 28 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & 14 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 14 & \varepsilon & 21 & \varepsilon & \varepsilon \end{bmatrix} \tag{1-5}$$

U ovom trenutku kada poznajemo matricu sustava $\boldsymbol{A}$ možemo izračunati prvih nekoliko vrijednosti vektora stanja pomoću jednadžbe dane izrazom (1-6). Prvih nekoliko stanja dano je izrazom (1-7).

$$x(k) = \boldsymbol{A} \otimes \mathrm{x}(k-1) \tag{1-6}$$

Početno stanje sustava dobijemo iz grafa i zadanih početnih položaja vlakova u sustavu.

$$x(0) = [e \quad \varepsilon \quad \varepsilon \quad e \quad \varepsilon \quad \varepsilon \quad e \quad \varepsilon \quad e \quad \varepsilon \quad \varepsilon]^T.$$

$$\begin{aligned}
x(1) &= [\,12 \quad 19 \quad 5 \quad 57 \quad 54 \quad 35 \quad 49 \quad 42 \quad 28 \quad 14 \quad 21]^T \\
x(2) &= [69 \quad 76 \quad 62 \quad 110 \quad 107 \quad 88 \quad 102 \quad 95 \quad 70 \quad 56 \quad 63]^T \\
x(3) &= 122 \quad 129 \quad 115 \quad 163 \quad 160 \quad 141 \quad 155 \quad 148 \quad 123 \quad 109 \quad 116]^T \\
x(4) &= [175 \quad 182 \quad 168 \quad 216 \quad 213 \quad 194 \quad 208 \quad 201 \quad 176 \quad 162 \quad 169]^T \\
x(5) &= [228 \quad 235 \quad 221 \quad 269 \quad 266 \quad 247 \quad 261 \quad 254 \quad 229 \quad 215 \quad 222]^T
\end{aligned} \tag{1-7}$$

Oduzimanjem susjednih vektora stanja sustava dobivamo vektore razlika koje koristimo za proračun trajanja prosječnog ciklusa. Razlike vektora stanja dane su izrazom (1-8).

$$\begin{aligned}
x(2) - x(1) &= [\,12 \quad \varepsilon \quad \varepsilon \quad 57 \quad \varepsilon \; \varepsilon \quad 49 \quad \varepsilon \quad 28 \quad \varepsilon \quad \varepsilon]^T \\
x(3) - x(2) &= [\,57 \quad 57 \quad 57 \quad 53 \quad 53 \quad 53 \quad 53 \quad 53 \quad 42 \quad 42 \quad 42]^T \\
x(4) - x(3) &= 53 \quad 53 \quad 53 \quad 53 \quad 53 \quad 53 \quad 53 \quad 53 \quad 53 \quad 53 \quad 53]^T \\
x(5) - x(4) &= [\,53 \quad 53 \quad 53 \quad 53 \quad 53 \quad 53 \quad 53 \quad 53 \quad 53 \quad 53 \quad 53]^T
\end{aligned} \tag{1-8}$$

Prosječan ciklus sustava λ računamo pomoću izraza (1-9):

$$\lambda = \frac{trace(\boldsymbol{A})}{1} \oplus \frac{trace(\boldsymbol{A}^2)}{2} \oplus \ldots \oplus \frac{trace(\boldsymbol{A}^{11})}{11} = 53 \tag{1-9}$$

Iz izraza (1-8) vidimo da u stacionarnom stanju imamo jedan ciklus duljine 53. Iz toga slijedi da je prosječan ciklus $\lambda = 53$. Do istog rezultata smo došli pomoću jednadžbe (1-9).

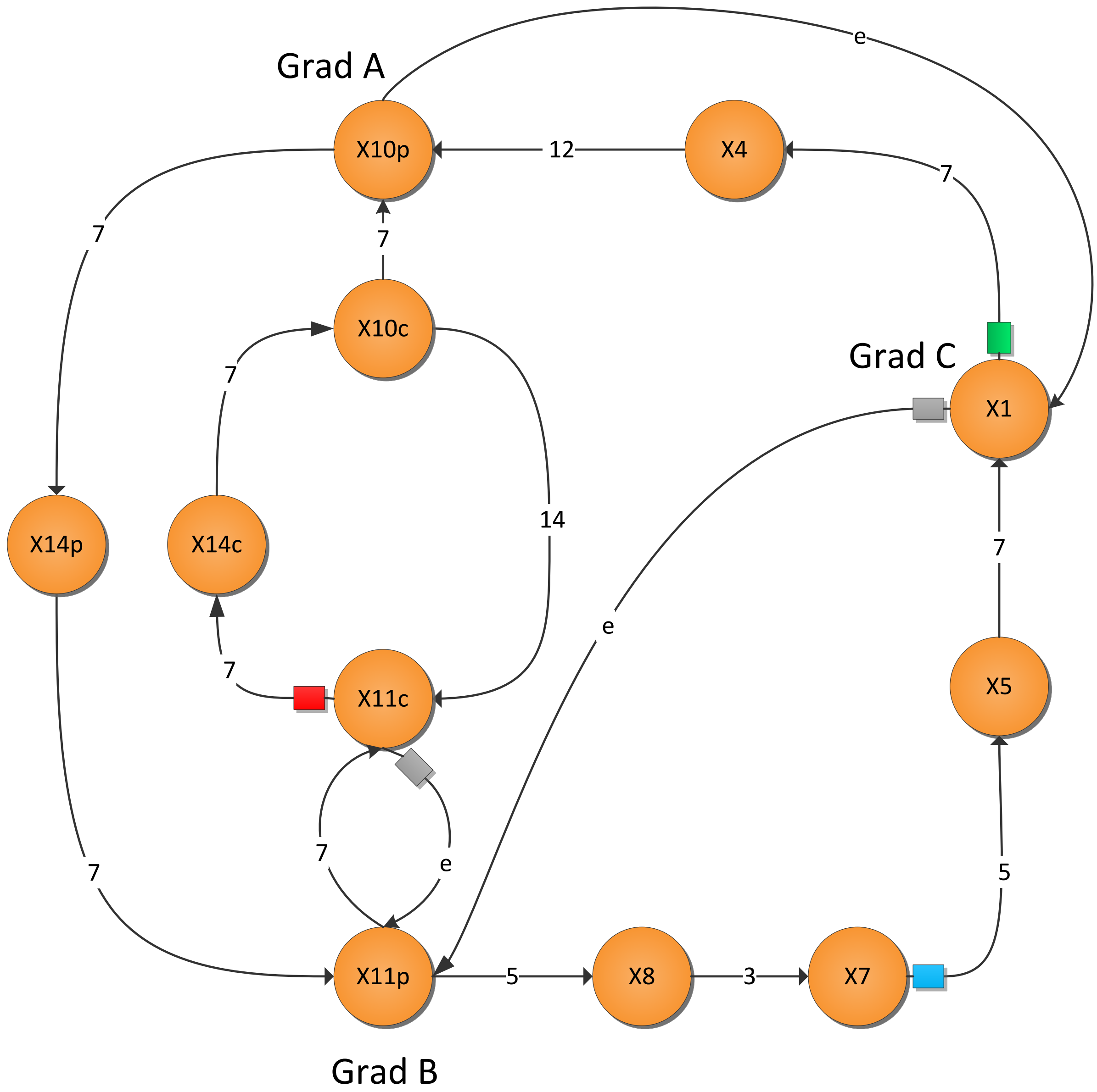
Po definiciji, kritični ciklus grafa je ona kružna staza na grafu koja ima najveću moguću srednju težinu, odnosno srednja težina kritičnog ciklusa grafa je upravo jednaka λ. U našem slučaju kritični ciklus grafa danog Slikom 1. je kružna staza koja odgovara plavoj liniji $\sigma_{krit} = \{x_1, x_4, x_{10p}, x_{14p}, x_{11p}, x_{8,} x_7, x_5\}$. Ako izračunamo srednju težinu te staze dobivamo da je $\bar{\sigma}_{kritw} = \frac{7+7+12+7+7+5+3+5}{1} = 53 = \lambda$ (Srednja težina staze $\bar{\sigma}$ definirana je kao omjer sume težina svih segmenata i broja paleta na stazi).

Zbog specifičnosti modela to je ujedno i minimalni kritični ciklus.

Programsko rješenje sustava s dva vlaka i ujedno **2. zadatka** dano je u **Dodatku A.**

3. Zadatak – rješenje

Slikom 3. prikazan je graf sustava dobiven iz opisa sustava danog zadatkom. Graf je identičan grafu sa Slike 2. jedino su dodane dodatne sinkronizacije kako bi se ispunio uvjet zadan zadatkom tj. imamo iste događaje kao i u prethodnom zadatku ovog seminara $X = \{x_1, x_4, x_5, x_7, x_8 x_{10p}, x_{11p}, x_{14p}, x_{10c}, x_{11c}, x_{14c}\}$.



Slika 3. Graf sustava s tri vlaka

Na Slici 3. pomoću kvadratića označen je početni položaj vlakova u sustavu. Crveni kvadratić je početni položaj vlaka koji ide crvenom linijom, plavi kvadratić je početni položaj vlaka koji ide plavom (zelenom) linijom, zeleni kvadratić je početni položaj vlaka koji ide zelenom (plavom) linijom te sivi kvadratić je tu zbog sinkronizacije. Iz grafa možemo direktno pisati jednadžbe sustava.

Sustav jednadžbi koji opisuje graf dan Slikom 3. dan je izrazima (1-10):

$$
\begin{gathered}
x_1(k) = 7 \otimes x_5(k) \oplus e \otimes x_{10p}(k) \\
x_4(k) = 7 \otimes x_1(k-1) \\
x_5(k) = 5 \otimes x_7(k-1) \\
x_7(k) = 3 \otimes x_8(k) \\
x_8(k) = 5 \otimes x_{11p}(k) \\
x_{10p}(k) = 12 \otimes x_4(k) \oplus 7 \otimes x_{10c}(k) \\
x_{11p}(k) = 7 \otimes x_{14p}(k) \oplus e \otimes x_{11c}(k-1) \oplus e \otimes x_1(k-1) \\
x_{14p}(k) = 7 \otimes x_{10p}(k) \\
x_{10c}(k) = 7 \otimes x_{14c}(k) \\
x_{11c}(k) = 14 \otimes x_{10c}(k) \oplus 7 \otimes x_{11p}(k) \\
x_{14c}(k) = 7 \otimes x_{11c}(k)
\end{gathered}
\tag{1-10}
$$

Iz sustava jednadžbi (1-10) možemo pisati matrice $\boldsymbol{A_0}$ i $\boldsymbol{A_1}$.

$$
A_0 = \begin{bmatrix}
\varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & 12 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & 14 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon
\end{bmatrix}
\tag{1-11}
$$

$$
A_1 = \begin{bmatrix}
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
7 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon
\end{bmatrix}
\tag{1-12}
$$

Elementi u matricama $\boldsymbol{A_0}$ i $\boldsymbol{A_1}$ posloženi su na isti način kao i u prvom zadatku tj. istim redom kao što je to navedeno u Tablici 2.

Također, za izračun matrice sustava $\boldsymbol{A}$ koristimo izraz dan jednadžbom (1-4). Od tuda slijedi:

$$A = \begin{bmatrix} 19 & \varepsilon & \varepsilon & 12 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 21 & \varepsilon \\ 7 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 41 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 43 & \varepsilon \\ 38 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 40 & \varepsilon \\ 19 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 21 & \varepsilon \\ 33 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 35 & \varepsilon \\ 26 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 28 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 14 & \varepsilon \\ 40 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon & 28 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon \end{bmatrix} \tag{1-13}$$

Kao i ranije pomoću izraza (1-6) i početnog stanja računom vektore stanja u idućim koracima. Dobiveni vektori stanja dani su skupo izraza (1-14).

$$x(0) = [e \quad \varepsilon \quad \varepsilon \quad e \quad \varepsilon \quad \varepsilon \quad \varepsilon \quad \varepsilon \quad \varepsilon \quad e \quad \varepsilon]^T.$$

$$\begin{aligned} x(1) &= [21 \quad 7 \quad 5 \quad 43 \quad 40 \quad 21 \quad 35 \quad 28 \quad 14 \quad 42 \quad 7]^T \\ x(2) &= [63 \quad 28 \quad 48 \quad 85 \quad 82 \quad 63 \quad 77 \quad 70 \quad 56 \quad 84 \quad 49]^T \\ x(3) &= [105 \quad 70 \quad 90 \quad 127 \quad 124 \quad 105 \quad 119 \quad 112 \quad 98 \quad 126 \quad 91]^T \\ x(4) &= [147 \quad 112 \quad 132 \quad 169 \quad 166 \quad 147 \quad 161 \quad 154 \quad 140 \quad 168 \quad 133]^T \\ x(5) &= [189 \quad 154 \quad 174 \quad 211 \quad 208 \quad 189 \quad 203 \quad 196 \quad 182 \quad 210 \quad 175]^T \end{aligned} \tag{1-14}$$

Oduzimanjem susjednih vektora stanja sustava dobivamo vektore razlika koje koristimo za proračun trajanja prosječnog ciklusa. Razlike vektora stanja dane su izrazom (1-15).

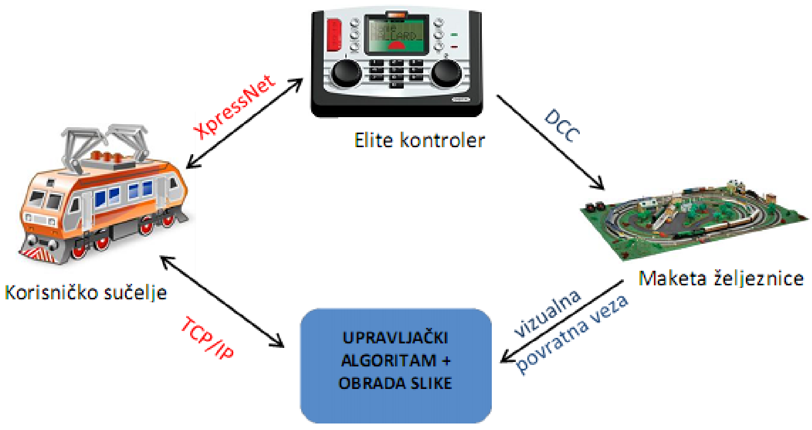
$$\begin{aligned} x(2) - x(1) &= [21 \quad \varepsilon \quad \varepsilon \quad 43 \quad \varepsilon \quad \varepsilon \quad 35 \quad \varepsilon \quad \varepsilon \quad 42 \quad \varepsilon]^T \\ x(3) - x(2) &= [42 \quad 21 \quad 43 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42]^T \\ x(4) - x(3) &= [42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42]^T \\ x(5) - x(4) &= [42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42 \quad 42]^T \end{aligned} \tag{1-15}$$

Prosječan ciklus sustava λ dodatno računamo i pomoću izraza (1-9) – $\lambda = 42$.

Iz izraza (1-15) vidimo da u stacionarnom stanju imamo jedan ciklus duljine 42. Iz toga slijedi da je prosječan ciklus $\lambda = 42$. Do istog rezultata smo došli pomoću jednadžbe (1-9) primijenjene na ovaj slučaj. Programsko rješenje sustava s tri vlaka dano je u **Dodatku B**.

STVARNI SUSTAV

Maketa željeznice sastoji se od sustava povezanih tračnica na kojima se gibaju vlakovi. Sustav tračnica sastoji se od pruga i skretnica koje se nalaze na svim račvanjima. Kako bi se ostvarilo autonomno ponašanje u sustav je uključena vizualna povratna veza i algoritam upravljanja.

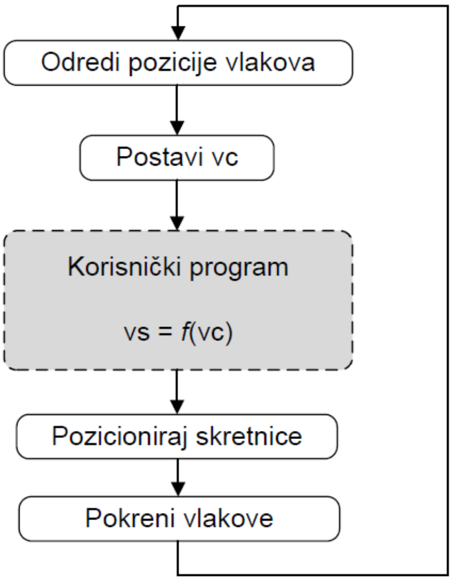


Slika 4. : Sustav upravljanja maketom željeznice

Maketa se sastoji se od proizvoda tvrtke Hornby Hobbies Ltd. koji podržavaju digitalni standard u upravljanju maketama željeznica – DCC (engl. Digital Command Control). Pomoću tog standarda omogućeno istovremeno upravljanje vlakovima i skretnicama s jednog mjesta. Elite DCC kontroler je povezan sa sustavom tračnica i generira DCC signale za upravljanje vlakovima i skretnicama. Elite DCC kontroler omogućava ručni mod rada, a može biti i povezan s računalom. U ovom radu upravljački algoritam izvodi se na računalu koje je povezano s Elite DCC kontrolerom preko serijske veze, a za komunikaciju se koristi XpressNet protokol. Implementirano je korisničko sučelje koje prevodi korisnički definirane naredbe u naredbe XpressNet protokola. Korisničko sučelje također može raditi u ručnom ili automatskom načinu. Vizualnu povratnu vezu čini web kamera koja se nalazi iznad makete i služi za određivanje pozicije svakog vlaka u sustavu. U tu svrhu na vlakovima se nalaze papirnati kvadrati određene boje koji algoritam za obradu slike jednostavno detektira. Vlak je određen svojom bojom.

U ovom sustavu može se upravljati vlakovima i skretnicama. Svaki vlak i skretnica imaju jedinstvenu adresu na DCC mreži. Skretnicama se može mijenjati položaj, a vlakovima se može mijenjati brzina. Da bi se moglo definirati upravljanje sustavom, svakoj skretnici (čvoru) u sustavu pridružuje se jedinstvena oznaka (Slika 1).

Staza za svaki vlak zadaje se kao niz čvorova koje vlak mora uzastopce posjećivati, pri čemu je potrebno zapisati sve čvorove na putu tj. nije dozvoljeno ’preskakati’ čvorove. Staza se zapisuje u zasebnoj *.cfg* datoteci. Svaka staza sastoji se od liste poslova - v, pri čemu poslovi odgovaraju usmjerenim bridovima po kojima vlakovi voze i određuje se na temelju staze zapisane u *.cfg* datoteci. Svaka lista poslova ima pripadnu listu započinjanja poslova - vs. Ako se želi započeti određena operacija, tada se pripadni element liste vs postavlja u 1. Nakon postavljanja elementa iz vs u 1, niža razina upravljanja postavlja skretnice u odgovarajući položaj te potom pokreće vlak. Posao se smatra završenim ako vlak dođe u blizinu željenog čvora. Tada se vlak automatski zaustavlja, što je ostvareno na nižoj razini upravljanja, te se u 1 postavlja pripadni element liste završenih poslova - vc. Vrijednosti u listama vc i vs postavljaju se u 0 kad vlak primi nalog za započinjanje novog posla. Primjerice vlak kad dođe ispred skretnice vc se postavlja u 1, vc poprima vrijednost 1 sve dok vlak ne krene, odnosno dok ne dođe naredba za započinjanje novog posla. Postavljanje vrijednosti na 0 je također ostvareno na nižoj razini upravljanja. Dijagram tijeka upraljačkog programa prikazan je na Slici 5.



Slika 5. Dijagram toga programa za upravljanje

Korisnički program se upisuje se u dio unutar Python skripte My_supervisor.py. Korisnik se pri izradi novog upravljačkog mora pridržavati unaprijed određene strukture funkcije koja implementira razvijeni algoritam upravljanja, ali unutar funkcije ima potpunu slobodu pisanja vlastitog koda. Korisnički algoritam upravljanja piše se u Python programskom jeziku. Funkcija tj. skripta koja sadrži upravljački algoritam poziva se pri svakom koraku diskretizacije upravljačkog dijela. Kao ulaz prima podatke o poziciji vlaka na pruzi, a kao izlaz daje vrijednosti upravljačkih signala za vlakove.

Pri završetku upravljanja željeznicom, sustav generira *.txt* datoteku u kojoj se nalaze *vs_time* i *vc_time*. U *vs_time* su zapisana vremena kada je postavljena jedinica u vs na nekom čvoru (skretnici) za bilo koji vlak, odnosno govori na vrijeme kada je pojedini vlak krenuo iz pojedinog čvora. Analogno *vc_time* govori vrijeme kada je pojedini vlak došao do pojedinog čvora. Vrijeme se broji u sekundama od početka 1970 godine.

1. Zadatak-rješenje na stvarnom sustavu

Korisnički upravljački algoritam napisan *Python*-u za sustav s dva vlaka (crveni i plavi, po jedan na svakoj liniji) dan je dodatku A. Početni smještaj vlakova napravljen je prema slici 2 Na kraju upravljanja generirana je text datoteka s vremenima kretanja i zaustavljanja vlakova ispred pojedinih čvorova (skretnica ili postaja). Iz datoteke očitavamo vremena trajanja pojedinih ciklusa za svaki vlak. Primjerice za plavi vlak koji prolazi kroz 8 čvorova prema slici 1., očitavamo vrijednosti *vs_time* svaki 8 mjerenja kako bi dobili koliko je trajanje puta nakon jednog, dva itd. napravljena kruga.

PLAVI		
Skretnica 7 k-ti put	Vrijeme polaska iz skretnice 7	Trajanje kruga
0. početak	1369119743.017	
1. krug	1369119793.701	50.684
2. krug	1369119844.045	50.344
CRVENI		
Skretnica 10 k-ti put	Vrijeme polaska iz skretnice 10	Trajanje kruga
0. početak	1369119743.25	
1. krug	1369119762.058	18.808
2. krug	1369119805.164	42.587
3. krug	1369119855.946	50.782

Tablica 3. Mjerenja na stvarnom sustavu na temelju *vs_time* za sustav s 2 vlaka

Kao što vidimo iz Tablice 3. prosječni ciklus iznosi oko 50 sekundi što je putanja plavog vlaka. Max-plus algebrom smo dobili da prosječni ciklus iznosi 53 sekunde. Vidimo da oba vlaka imaju ciklus trajanja oko 50-51 sekundi što se približno podudara s podacima dobivenih *Matlabom*. Tehnički na stvarnom sustavu trajanje putanje crvenog i plavog vlaka bi trebalo biti 53 sekunde, što je upravo prosječan ciklus. U stacionarnom stanju crveni čeka plavog, budući da ima kraću putanju te prolazi kroz stanicu A bez zaustavljanja, a plavi uvijek kruži putanjom trajanja 53 sekunde bez čekanja budući da mu je u stacionarnom stanju uvijek slobodan izlaz iz skretnice i stanice (u gradu B je prednost dana plavom vlaku tako da plavi vlak dođe u B i odmah krene). Prema tome trajanje putanje crvenog jest 53 sekunde. Ako gledamo kao početak neposredni izlazak plavog vlaka iz grada B onda crveni prvo čeka 7 sekundi kako bi izbjegao sudar. Zatim radi svoju putanju u trajanju 28 sekundi i dolazi opet do grada B, gdje čeka još dodatnih 18 sekundi koliko treba plavom vlaku da dođe do grada B, pa je to onda sveukupno 53 sekunde.

$$T_{plavi} = 5 + 3 + 5 + 7 + 7 + 12 + 7 + 7 = 53$$

$$T_{crveni} = 28 + (53 - 35) + 7 = 53$$

Iz tablice 3 je vidljivo još i to da sustav već nakon drugog kruga uđe u ustaljeno stanje, što se i podudara s vrijednostima danim u (1-8). Vidimo da se trajanje drugog kruga u potpunosti podudara s onim dobivenim max-plusom i iznosi 43 sekunde. Trajanje prvog kruga trebalo bi biti 28 sekundi, ali očito zbog smještaja početnih uvjeta se ono drastično smanji u stvarnosti.

Računanja smo mogli provesti i na temelju vrijednosti *vc_time*. Vrijednosti koje bismo dobili bile bi:

PLAVI	
1. krug	51.973
2. krug	50.331
CRVENI	
1. krug	22.828
2. krug	43.097

Tablica 4. Mjerenja na stvarnom sustavu na temelju *vc_time* za sustav s 2 vlaka

Iz Tablice 4. vidljivo je da su mjerenja preko *vc_time* približno ista kao i ona mjerena preko *vs_time*. Jedino je za početni krug crvenog vlaka točniji pokazatelj trajanja kruga, ali to nije niti bitno previše jer tu još nismo došli u stacionarno stanje. Ta razlika proizlazi iz toga što vlak na početku nije bio smješten odmah nakon skretnice nego nešto dalje, a budući da je u *vc_time* zapisano kad je vlak došao u skretnicu, prvo mjerenje se dogodi tek kada dođe u stanicu B, tako da nije ovisio previše o smještaju vlaka.

Razlika u vrijednostima može se pripisati tome što trajanja pojedinih dijelova putanje nisu u sekundu točno definirana. Dodatni uzrok mogućim netočnim podacima jest kamera koja u nekim slučajevima prekasno detektira dolazak vlaka ispred skretnice, pa se vlak prekasno zaustavi što može rezultirati sudarim dvaju vlakova. Još jedan uzrok je što pri brzine pri pokretanju vlakova nisu baš iste, jedan krene brže drugi sporije. Rezultati s boljom preciznošću bi se dobili da smo uzeli još nekoliko krugova, ali smo imali probleme s ispadanjem vlaka na jednom dijelu putanje.

3. Zadatak – rješenje na stvarnom sustavu

Korisnički upravljački algoritam napisan *Python*-u za sustav s tri vlaka (crveni na jednoj liniji, plavi i zeleni na drugoj liniji s uvjetom da između dva grada uvijek može biti samo jedan vlak) dan je dodatku B. Početni smještaj vlakova napravljen je prema Slici 3. Na kraju upravljanja generirana je text datoteka s vremenima kretanja i zaustavljanja vlakova ispred pojedinih čvorova (skretnica ili postaja). Iz datoteke očitavamo vremena trajanja pojedinih ciklusa za svaki vlak kao i u zadatku 1.

PLAVI		
Skretnica 7 k-ti put	Vrijeme polaska iz skretnice 7	Trajanje kruga [s]
0. početak	1369124767.306	
1. krug	1369124852.302	84.996
2. krug	1369124937.455	85.153
3. krug	1369125024.398	86.943
CRVENI		
Skretnica 11 k-ti put	Vrijeme polaska iz skretnice 10	Trajanje kruga [s]
0. početak	1369124767.776	
1. krug	1369124807.577	39.801
2. krug	1369124850.066	42.489
3. krug	1369124893.084	43.018
4. krug	1369124935.724	42.640

5. krug	1369124979.042	43.318
6. krug	1369125023.662	44.62
ZELENI		
Skretnica 1 k-ti put	Vrijeme polaska iz skretnice 1	Trajanje kruga [s]
0. početak	1369124767.54	
1. krug	1369124829.295	61.755
2. krug	1369124914.049	84.754
3. krug	1369125001.224	87.175

Tablica 5. Mjerenja na stvarnom sustavu na temelju *vs_time* za sustav s 3 vlaka

Kao što vidimo iz tablice 5, trajanje prosječnog kruga za plavi i zeleni iznosi oko 85 sekundi, što se ne podudara s prosječnim ciklusom dobivenim iz Max-plus algebre. Ali to je zato jer u max-plus algebri mi nismo razlikovali čvorove za plavi i čvorove za zeleni vlak. Stoga da bi se provjerili rezultati dobiveni Max-plus metodom potrebno je gledat kada plavi, a kada zeleni vlak kreće iz istog čvora.

PLAVI I ZELENI		
Skretnica 7 k-ti put	Vrijeme polaska iz skretnice 7	Trajanje [s]
0. početak	1369124767.306 PLAVI	
1. put	1369124809.327 ZELENI	42.021
2. put	1369124852.302 PLAVI	42.975
3. put	1369124894.818 ZELENI	42.516
4. put	1369124937.455 PLAVI	42.637
5. put	1369124981.303 ZELENI	43.848
6. put	1369125024.398 PLAVI	43.095

Tablica 6. Mjerenja plavog i zelenog na stvarnom sustavu na temelju *vs_time* za sustav s 3 vlaka

Ovakvim razmatranjem dobivamo isti prosječni ciklus kao i iz Max-plus metode. Iz Tablica 5. i 6. vidljivo je da već nakon prvog kruga sustav uđe u stacionarno stanje što potvrđuju i jednadžbe (1-15). Prosječni krug crvenog vlaka također iznosi 42 sekunde. Za njega nismo morali gledati kad je koji čvor aktivan budući da su za njega napravljena odvojena stanja *x10c* i *x11c*. Proračun se mogao provoditi i na temelju *vc_time*, dobili bi se vrlo slični rezultati.

Za kraj je potrebno napomenuti da sustav ima brojne probleme, tako da je bilo potrebno konstantno resetiranje svih uređaja, velika pažnja pri postavljanju i pratnji vlakova kako bi se dobili približno točni rezultati.

Dodatak A

Upravljački algoritam za upravljanje maketom željeznice na kojoj se nalaze dva vlaka.

```
"""
Upravljanje maketom željenice. Seminar 1 Sustavi s diskretnim događajima

Bartulović Mihovil
Erdelić Tomislav
Pavlović Alen

@FER, 21.05.2013

---> Upravljački algoritam za upravljanje maketom željenice, na kojoj se nalaze dva vlaka.
      >crveni
      >plavi
U .cfg datoteku je potrebno definirati staze vlaka prema max - plus modelu.

'blue': 7;5;1;4;10;14;11;8
'red':10;11;14

Algoritam upravljanje moguće je testirati na realnom sustavu.



"Liste poslova za pojedini vlak , spremljene u tip podataka DICTIONARY"
Za primjer s dva vlaka na temelju staze zapisane u .cfg datoteci dobiva se:
v = {'blue':[(7 , 5) ,( 5 , 1) ,( 1 , 4 ) ,( 4 , 10 ) ,( 10 ,14) , ( 14 , 11),\
             ( 11, 8 ) , ( 8 , 7) ],'red':[( 10 , 11),( 11 ,14 ) ,(14 ,10)]}
"""

"Učitavanje modula za mjerenje vremena"

from time import time

if firstScan :
        """Inicjalizacija pojedinog vlaka
        Plavi vlak vozi od 7 prema 5
        Crveni vlak vozi od 10 prema 11
        """
        vs['blue'][0]   = 1
        vs['red'][0]    = 1
        """Pomoćne varijable za mjerenje vremena i zauzetosti gradova,
        zbog sigurnosnih razloga čekanje od 7 sekundi.
        Ovaj dio nije potreban kad se piše u skriptu My_supervisor jer sa na početku klase
        kad se pozove inicijalizacija se inicijaliziraju ove vrijednosti.Ali su potrebne za
        prilikom testiranja i pisanja koda prije dolaska na vježbu.
        startBlue=0
        startRed =0
```

```
        "cityBred kazuje dali je putanje 11->14->10 zauzeta / slobodna"
        cityBred=False
        "pom omoguće / zabranjuje polazak iz grada B , uvjet sinkronizacije "
        pom=True
        """
else :
        "Provjera dali postoji posao koji je završen u vektoru vc "
        if vc['blue'].count(1) != 0 :

            " Upravljački algoritam za PLAVI vlak "

            if   vc['blue'][0]  == 1 :
                vs['blue'][1]  =  1

            elif vc['blue'][1] ==  1 :
                vs['blue'][2]  = 1

            elif vc['blue'][2]  == 1 :
                vs['blue'][3]   = 1

            elif vc['blue'][3] ==  1 and ((time()-startRed )>7) and not(cityBred):
                vs['blue'][4] =   1

            elif vc['blue'][4] ==  1 :
                vs['blue'][5]  =  1

            elif vc['blue'][5] ==  1 and vc['red'][0] == 1 :
                "Brojač vremena za sinkronizaciju"
                startBlue     = time()
                pom           = True
                vs['blue'][6] =   1

            elif vc['blue'][6] ==  1 :
                vs['blue'][7] =   1

            elif vc['blue'][7] ==  1 :
                vs['blue'][0] =   1

        if vc['red'].count(1) !=  0 :
            "Upravljački algoritam za CRVENI vlak"

            if  vc['red'][0]  == 1 and ((time() - startBlue) > 7 )  and pom :
                vs['red'][1]  =   1
                pom = False
                cityBred=True

            elif vc['red'][1] == 1 :
                vs['red'][2]   = 1

            elif vc['red'][2] == 1 :
                startRed= time()
```

```
            vs['red'][0] = 1
            cityBred=False
```

Dodatak B

Upravljački algoritam za upravljanje maketom željeznice na kojoj se nalaze tri vlaka.

```
"""
Upravljanje maketom željenice. Seminar 1 Sustavi s diskretnim događajima

Bartulović Mihovil
Erdelić Tomislav
Pavlović Alen

@FER, 21.05.2013

---> Upravljački algoritam za upravljanje maketom željenice, na kojoj se nalaze tri vlaka.
      >crveni
      >plavi
      >zeleni
U .cfg datoteku je potrebno definirati staze vlaka prema max - plus modelu.

'blue': 7;5;1;4;10;14;11;8
'red':10;11;14
'green': 7;5;1;4;10;14;11;8

Algoritam upravljanje moguće je testirati na realnom sustavu.


"Liste poslova za pojedini vlak , spremljene u tip podataka DICTIONARY"
Za primjer tri vlaka na temelju staze zapisane u .cfg datoteci dobiva se:
v = {'blue':[(7 , 5) ,( 5 , 1) ,( 1 , 4 ) ,( 4 , 10 ) ,( 10 ,14) , ( 14 , 11), ( 11, 8 ) , ( 8
, 7) ],\
     'red':[( 10 , 11),( 11 ,14 ) ,(14 ,10)],\
     'green':[(7 , 5) ,( 5 , 1) ,( 1 , 4 ) ,( 4 , 10 ) ,( 10 ,14) , ( 14 , 11), ( 11, 8 ) , (
8 , 7) ]}
"""
"Učitavanje modula za mjerenje vremena"

from time import time

if firstScan :
    """ Pokretanje pojedinog vlaka s odgovarajuće pozicije,
    prema početnim uvjetima iz max-plus modela.
    """
    vs['blue'] [0]    = 1
    vs['red']  [1]    = 1
    vs['green'][2]    = 1
    """Ovaj dio nije potreban kad se piše u skriptu My_supervisor jer sa na početku klase
```

```
    kad se pozove inicijalizacija se inicijaliziraju ove vrijednosti.Ali su potrebne za
    prilikom testiranja i pisanja koda prije dolaska na vježbu.
    "Inicjalizacija vremena zbog sigurnosnih razloga "
    startBlue  = time()
    startRed   = time()
    startGreen = time()

    Pomoćne varijable kojima se osigurava TOČNO jedan vlak između dva grada.

    cityCgreen = True označava da je ZELENI vlak krenuo iz grada C , PLAVI vlak iz C
    ne može krenuti sve dok se varijabla ne vrati u FALSE. Implementaija u upravljačkom
algoritmu.



    cityCgreen = True
    cityBgreen = False
    cityAgreen = False

    cityCblue =  False
    cityBblue  = True
    cityAblue =  False

    cityBred=True


    "Pomoćne varijable za sinkronizaciju"
    pomBlue  = False
    pomGreen = False
    pomRed=False
    """

else :

# ------------------------- BLUE ---------------------------------

    if vc['blue'].count(1) != 0 :
        "Upravljački algoritam PLAVOG vlaka"

            if   vc['blue'][0]  == 1 :

                vs['blue'][1]  =  1

            elif vc['blue'][1] ==  1 and not(cityCgreen) and cityAgreen :

                vs['blue'][2]  =  1

                cityCblue      =  True
                cityBblue      =  False

            elif vc['blue'][2]  == 1 :
```

```
                vs['blue'][3]  =  1

            elif  vc['blue'][3]  ==    1  and  ((time()-startRed  )>7)  and  not(cityAgreen)  and
cityBgreen and not(cityBred) and pomRed:

                vs['blue'][4]  =   1

                cityAblue      =   True
                cityCblue      =   False
                pomRed         =   False
            elif vc['blue'][4] ==   1 :

                vs['blue'][5]  =   1

            elif vc['blue'][5] ==   1 and vc['red'][0] == 1 and not(cityBgreen) and cityCgreen
:

                startBlue      =   time()
                startGreen     =   time()
                pomBlue        =   True
                vs['blue'][6]  =   1

                cityAblue      =   False
                cityBblue      =   True

            elif vc['blue'][6] ==   1 :

                vs['blue'][7] =    1

            elif vc['blue'][7] ==   1 :

                vs['blue'][0]  =   1

# ------------------------- GREEN ---------------------------------

    if vc['green'].count(1) != 0 :
        "Upravljački algoritam ZELENOG vlaka"

            if   vc['green'][2]  == 1 :

                vs['green'][3]  =  1

            elif  vc['green'][3]  ==    1  and  ((time()-startRed  )>7)  and  not(cityAblue)  and
cityBblue and not(cityBred) and pomRed:

                vs['green'][4]  =  1

                cityAgreen      =  True
                cityCgreen      =  False
                pomRed         =   False
```

```
            elif vc['green'][4]  == 1 :

                vs['green'][5]   = 1

            elif vc['green'][5] ==  1 and vc['red'][0] == 1  and not(cityBblue) and cityCblue
:

                vs['green'][6] =   1

                cityAgreen     =   False
                cityBgreen     =   True

                pomGreen       =   True

                startGreen     = time()
                startBlue      =time()

            elif vc['green'][6] ==  1 :
                vs['green'][7]  =  1

            elif vc['green'][7] ==  1  :
                vs['green'][0] =   1

            elif vc['green'][0] ==  1 :
                vs['green'][1] =   1

            elif vc['green'][1] ==  1 and not(cityCblue) and cityAblue :
                vs['green'][2]  =   1

                cityBgreen      =   False
                cityCgreen      =   True

# ------------------------- RED ---------------------------------

    if vc['red'].count(1) !=  0 :
        "Upravljački algoritam CRVENOG vlaka "

            if vc['red'][0]  == 1 and ((((time() - startBlue) > 7 )  and pomBlue) or (((time()
- startGreen) > 7 )  and pomGreen) ):

                vs['red'][1]  =   1
                cityBred=True
                pomGreen = False
                pomBlue  = False

            elif vc['red'][1] == 1 :

                vs['red'][2]   = 1

            elif vc['red'][2] == 1 :
```

```
                startRed= time()
                pomRed  = True
                vs['red'][0] = 1
                cityBred=False
```