České vysoké učení technické v Praze

Fakulta elektrotechnická

Diplomová práce

Modelování dopravních systémů s využitím Petriho sítí

Transport systems modelling using Petri nets

Vypracoval: Bc. Vojtěch Navrátil

Vedoucí práce: Ing. Michal Kutil

České vysoké učení technické v Praze
Fakulta elektrotechniky

Katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Vojtěch Navrátil**

Studijní program: Elektrotechnika a informatika (magisterský), strukturovaný
Obor: Kybernetika a měření, blok KM1 – Řídicí technika

Název tématu: **Modelování dopravních systémů s využitím Petriho sítí**

Pokyny pro vypracování:

1. Seznamte se s formalismem Petriho sítí a jeho využitím pro řešení problémů transportních sítí.
2. Implementujte nástroj pro podporu formalismu Petriho sítí.
3. Proveďte a porovnejte příkladové simulace transportních sítí modelovaných různými metodami (časované Petriho sítě, hybridní Petriho sítě, spojité Petriho sítě s konstantní maximální rychlostí).
4. Proveďte příkladovou simulaci transportních sítí většího rozsahu, minimálně dvou křižovatek.

Seznam odborné literatury

Dodá vedoucí práce

Vedoucí: Ing. Michal Kutil

Platnost zadání: do konce letního semestru 2010/2011

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 30. 11. 2009

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne …… …….

podpis

Poděkování

Na tomto místě bych rád poděkoval vedoucímu své bakalářské práce panu Ing. Michalu Kutilovi za užitečné rady a pomoc, bez které by tato práce nevznikla. Můj vděk patří také všem, kdo mě v mé práci vytrvale podporovali.

Téma

Modelování dopravních systémů s využitím Petriho sítí

Abstrakt

Náplní tohoto projektu je modelování dopravních sítí pomocí Petriho sítí různých typů. Za tím účelem je implementován simulátor hybridních Petriho sítí ve formě snadno rozšiřitelného programového balíku pro prostředí MATLAB®. Jeho jednotlivé moduly dovolují transformovat PNML definici Petriho sítě do matematické formy, generovat vývoj značení takové sítě a výsledky zobrazit v přehledné formě.

Dále jsou na základě formalismu Petriho sítí modelovány různé části dopravních sítí diskrétním, spojitým i hybridním přístupem. Tyto tři druhy modelů jsou porovnávány z několika hledisek na výsledcích simulace křižovatky a ulice. Práce je zakončena příkladovou studií oblasti několika vzájemně propojených křižovatek modelované spojitou a hybridní Petriho sítí. Model zahrnuje také dopravní zpoždění vzniklé nezanedbatelnou vzdáleností mezi křižovatkami.

Theme

Transport systems modeling using Petri nets

Abstract

Content of the project is a transport system modeling using Petri nets of various kinds. For the purpose a hybrid Petri nets simulator was implemented in a form of easily extensible software package for the MATLAB® environment. Its components allow transforming PNML definitions of a Petri net into the mathematical form, generating evolution of the net and displaying the results in an attractive and transparent form.

Further, using the Petri nets formalism of discrete, continuous and also hybrid approach, parts of a traffic system were modeled. The three types of models have been compared from several points of view using simulations of an intersection and a road. The final part of the work is a case study of a traffic area containing several intersections connected each to another. For this, the continuous and hybrid type of Petri net is used. The model also includes a transport delay caused by distance from each intersection to another.

Obsah

1. Úvod ... 1

2. Petriho sítě ... 2

2.1. Zobecněné Petriho sítě (GPN) ... 2

2.2. Časované Petriho sítě (TPN) ... 4

2.3. Spojité Petriho sítě (CCPN) ... 5

2.4. Hybridní Petriho sítě (HPN) ... 9

2.5. Rozšíření Petriho sítí o inhibiční hranu ... 11

3. Editace a reprezentace Petriho sítí ... 12

3.1. PN Editor ... 12

3.2. PNML ... 12

4. Simulace dopravních sítí pomocí Petriho sítí ... 14

5. Simulátor Petriho sítí ... 16

5.1. Složení simulátoru ... 16

5.2. Načtení dat ... 17

5.2.1. pnml2pn ... 17

5.2.2. joinnets ... 18

5.3. Generování vývoje Petriho sítí ... 21

5.3.1. playhpn ... 21

5.4. Prezentace vypočtených dat ... 27

5.4.1. dispevolution ... 29

5.4.2. dispstates ... 29

6. Simulace modelové dopravní křižovatky ... 30

6.1. Dané a vypočítané parametry ... 31

6.2. TPN model ... 32

6.3. CCPN model ... 33

6.4. HPN model ... 34

6.5. Simulace založená na reálných datech ... 35

6.6. Zhodnocení a porovnání modelů ... 37

6.6.1. Kvalita simulace ... 37

6.6.2. Výpočetní náročnost ... 39

7. Návrh modelu ulice ... 41

7.1. Modelování ulice pomocí HPN ... 42

8. Simulace reálné oblasti ... 47

8.1. Zvolená oblast **47**

8.2. Ukázka simulace křižovatky 3 ..... **48**

8.3. Studie oblasti šesti křižovatek **49**

8.3.1. Vstupní signál 49

8.3.2. Výsledek simulace 50

9. Závěr ***54***

10. Seznam použitých zdrojů ***55***

A. Příloha 1 – Přiložené CD ***58***

B. Příloha 2 – Seznam použitých zkratek ***59***

C. Příloha 3 – Údaje o simulované oblasti ***60***

Oblast Praha Smíchov **60**

Křižovatka 1: Matoušova – Preslova 61

Křižovatka 2: V Botanice – Elišky Peškové – Preslova 62

Křižovatka 3: V Botanice – Zborovská 63

Křižovatka 4: Matoušova – Zborovská 64

Křižovatka 5: Matoušova – Jiráskův most – Hořejší nábřeží 65

Křižovatka 6: V Botanice – Jiráskův most – Janáčkovo náb. – Hořejší náb. 66

Ulice 67

D. Příloha 4 – Vstupní naměřené signály ***68***

Seznam tabulek

Tabulka 1 – Parametry křižovatky *31*

Tabulka 2 – Srovnání výsledků simulace *40*

Tabulka 3 – Parametry ulic (zdrojem a cílem je míněna zdrojová a cílová křižovatka) *48*

Seznam kódů

Kód 1 – Definice propojení Petriho sítí a přiřazení proměnného signálu *18*

Kód 2 – Posloupnost příkazů pro provedení simulace a vykreslení vývoje Petriho sítě *28*

Seznam vývojových diagramů

Vývojový diagram 1 – Algoritmus funkce joinnets *20*

Vývojový diagram 2 – Základní algoritmus funkce playhpn *24*

Vývojový diagram 3 – Spojitá část funkce playhpn *25*

Vývojový diagram 4 – Hledání uvolněných přechodů – ctenabled *26*

Vývojový diagram 5 – Diskrétní část funkce playhpn *27*

Seznam obrázků

Obrázek 1 – Petriho síť 2
Obrázek 2 – Konflikt v zobecněné Petriho síti 3
Obrázek 3 – Časovaná Petriho síť – semafor 4
Obrázek 4 – Vztah diskrétní a spojité Petriho sítě 5
Obrázek 5 – Spojitá Petriho síť a vývoj jejího značení 5
Obrázek 6 – Spojitá Petriho síť s konfliktem, hledání skutečných rychlostí slabě uvolněných přechodů 8
Obrázek 7 – Hybridní Petriho sítě: a) spojitá řízená diskrétní, b) diskrétní řízená spojitou, c) konverze spojitá → diskrétní, d) konverze diskrétní → spojitá 10
Obrázek 8 – Hybridní konflikt 10
Obrázek 9 – Použití inhibiční hrany 11
Obrázek 10 – PN Editor: pravá část umožňuje kreslení, v levé části je umístěn editovatelný seznam prvků Petriho sítě 12
Obrázek 11 – Znázornění hierarchie elementů ve struktuře PNML od vnějších prvků (zleva) po vnitřní 13
Obrázek 12 – Schematické znázornění modulů simulátoru 16
Obrázek 13 – Petriho sítě pro simulaci ulice: zdroj (vlevo) a dopravní zpoždění (vpravo) 19
Obrázek 14 – Výsledek spojení podle kódu 1 19
Obrázek 15 – Hybridní Petriho síť pro ukázku vykreslení výstupů simulace 28
Obrázek 16 – Zobrazení výsledků simulace hybridní Petriho sítě z obrázku 15 28
Obrázek 17 – Modelová křižovatka 30
Obrázek 18 – Diagram časového rozdělení fází křižovatky s naznačením dob uvolnění jednotlivých ulic. 30
Obrázek 19 – Model křižovatky tvořený časovanou Petriho sítí 32
Obrázek 20 – Model křižovatky tvořený spojitou Petriho sítí s konstantní rychlostí 33
Obrázek 21 – Model křižovatky tvořený hybridní Petriho sítí 34
Obrázek 22 – Výstupy detektorů projíždějících vozidel 35
Obrázek 23 – Výsledky simulace – rychlost, jíž vozidla opouštějí křižovatku do výstupních ulic 36
Obrázek 24 – Výsledky simulace – celkový počet vozidel upouštějících křižovatku 36
Obrázek 25 – Výsledky simulace – neaproximovaný vstupní signál pro simulaci TPN modelem 37
Obrázek 26 – Výsledky simulace – celkový počet vozidel na výstupu křižovatky – ukázka jednoho řídícího cyklu 38
Obrázek 27 – Ulice – dopravní zpoždění při modelování 41
Obrázek 28 – Modelování ulice: a) TPN model, b) HPN model, c) Výsledek simulace 42
Obrázek 29 – HPN model jednoduché ulice 43
Obrázek 30 – Modelování ulice zakončené světelnou signalizací 44
Obrázek 31 – Vývoj rozjíždějící se fronty vozidel 44
Obrázek 32 – Modelování ulice zakončené semaforem modelující frontu čekajících vozidel 46
Obrázek 33 – Letecký snímek ukazující vybranou oblast. Jeho zdrojem je ©GEODIS Brno, s.r.o. (cit. 4. 12. 2009). 47
Obrázek 34 – Náhled na seskupení pražských křižovatek vybraných pro simulaci 47
Obrázek 35 – Křižovatka 3: V Botanice – Zborovská 50°4'31.5"N, 14°24'27.4"E 48
Obrázek 36 – Porovnání výstupů CCPN modelu i reálné křižovatky 3 v závislosti na čase 49
Obrázek 37 – Průjezd vozidel z ulice Elišky Peškové jednotlivými křižovatkami s odbočovacími poměry. 50
Obrázek 38 – Originální a modifikovaný signál. Modifikací je pulz délky 10 s. 50
Obrázek 39 – Výsledky simulace oblasti šesti křižovatek modelované Petriho sítí. 51
Obrázek 40 – Výsledky simulace oblasti šesti křižovatek modelované Petriho sítí – detail. 52

1. Úvod

V dnešním rychle se rozvíjejícím světě hraje doprava nezastupitelnou úlohu. Přeprava potravin, materiálů, výrobků, hromadná a osobní doprava - bez toho všeho si neumíme život představit. Rychle a bezpečně se dopravit kamkoliv je prioritou dnešní doby. Každý zná příklady ukazující nutnost budování kvalitní a spolehlivé dopravní sítě opatřené řídicím systémem, který zajistí plynulost a bezpečnost provozu na silničních komunikacích. Především v oblastech velkých aglomerací, kde se setkáváme s hustou dopravou směřující až k dopravním zácpám, by naše snahy měly směřovat k vytvoření fungujícího řízení.

Nezbytným předpokladem jeho realizace je znalost chování řízeného systému. V mnohých případech však není možné toto chování studovat ve skutečném světě. Vhodně vytvořeným obrazem dané problematiky získáme model, jejž můžeme analyzovat. Díky moderní výpočetní technice jsme schopni v krátkém čase a s poměrně nízkými náklady simulovat odezvu modelovaného systému i na takové vstupní podmínky, které bychom v realitě uměle navodili jen s obtížemi.

Formalizmus Petriho sítí nám dává do rukou nástroj, který je pro takové využití velice vhodný. Existuje již řada přístupů a pohledů na problematiku simulování dopravních či transportních sítí. Cílem této práce je srovnání způsobů modelování s využitím různých typů Petriho sítí a předvedení jejich aplikace na reálnou dopravní síť.

Aby mohl být tento cíl naplněn, musíme mít k dispozici kromě znalostí zvoleného formalizmu také nástroje, které nám dovolí modely vytvářet a zkoumat. Máme porovnávat modely využívající různé přístupy. Základním kamenem tohoto projektu tedy bude realizace univerzálního simulátoru schopného operovat s diskrétními, spojitými i hybridními Petriho sítěmi.

Srovnání, které je dalším z úkolů této práce, je třeba provést na situaci představující netriviální případ řešeného problému. Musíme tedy zvolit vhodnou křižovatku, provést návrh modelů různých typů a jejich odezvy na stejné vstupní podmínky srovnat jak z hlediska přesnosti výstupů, tak z hlediska náročnosti výpočtů simulace.

Pro splnění poslední části zadání musíme být schopni vytvořit model značného rozsahu. V takovém případě je výhodné vytvořit modely parciální, představující zlomky celého systému a ty před provedením simulace nechat spojit automaticky, což současně sníží pracnost a eliminuje případné chyby v návrhu. V rámci této práce tedy implementujeme i nástroj schopný spojovat Petriho sítě na základě jednoduchého intuitivního předpisu.

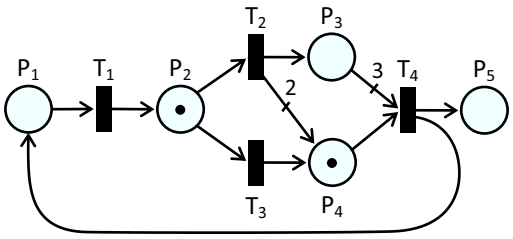
2. Petriho sítě

Carl Adam Petri, německý matematik, ve své disertační práci (1962) definoval obecný matematický model popisující vztahy podmínek a událostí, nyní nazývaný Petriho síť (PN – Petri Net). Inženýři jej začali používat pro modelování systémů diskrétních událostí a to především v aplikacích automatického řízení a výpočetní techniky. Od té doby bylo navrženo mnoho různých rozšíření základního formalizmu obecných Petriho sítí o nové vlastnosti. Vznikly prioritní, časové a časované (deterministické a stochastické) Petriho sítě. V 80. letech je prezentována spojitá Petriho síť, což je limitní případ diskrétní Petriho sítě. Spojení spojitého a diskrétního pohledu v jednom modelu vytváří hybridní Petriho síť. Dalšími typy Petriho sítí se specializovanými vlastnostmi jsou zkrácené (kapacitní, barvené) a rozšířené (s inhibiční hranou).

Tato práce je věnována hybridním Petriho sítím (HPN – Hybrid Petri Net). Týká se jak časovaných (TPN – Timed Petri Net), tak spojitých sítí s konstantní maximální rychlostí (CCPN – Constant speed Continuous Petri Net) a kombinuje je do hybridní formy. Rozšířením základních vlastností aparátu spojitých i časovaných Petriho sítí je inhibiční hrana.

Petriho síť je reprezentována orientovaným ohodnoceným bipartitním grafem. Jedná se tedy o graf se dvěma druhy uzlů, pojmenovaných místa a přechody (**Places**, **Transitions**), propojených orientovanými ohodnocenými hranami (**Arcs**). Hrana v bipartitním grafu vždy spojuje dva uzly různého typu. Spojení dvou uzlů je provedeno nejvýše jednou hranou. Petriho síť lze jednoznačně a úplně popsat grafickým znázorněním.

2.1. Zobecněné Petriho sítě (GPN)



$$Pre = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Post = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Obrázek 1 – Petriho síť

Obrázek 1 ukazuje Petriho síť tvořenou množinou míst $P = \{P_1, \dots, P_5\}$ a množinou přechodů $T = \{T_1, \dots, T_4\}$. Podle [Hanzálek II] zaveďme:

$^\circ P_i$	množina všech vstupních přechodů místa P_i,
$P_i{}^\circ$	množina všech výstupních přechodů místa P_i,
$^\circ T_j$	množina všech vstupních míst přechodu T_j,
$T_j{}^\circ$	množina všech výstupních míst přechodu T_j.

Místa a přechody jsou propojeny hranami, které lze zapsat ve formě dvou matic:

Pre Precondition – reprezentuje spojení místo → přechod – $P \times T \rightarrow Z_0^+$,

$Post$ Postcondition – reprezentuje spojení přechod → místo – $P \times T \rightarrow Z_0^+$.

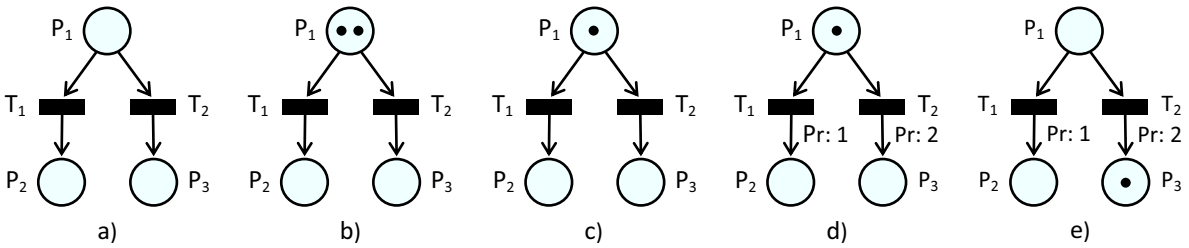
Pro zobecněnou Petriho síť prvky těchto matic nabývají hodnot (vah, **weights**) jednotlivých hran. V případě neexistující hrany je hodnota 0. Každé místo je značeno celočíselným počtem tokenů $\mathrm{m}(P_i)$. Značení (**marking**) Petriho sítě je určeno vektorem $m = \big(m(P_1), \dots, m(P_5)\big)$, který je zobrazením $P \rightarrow Z_0^+$. Značení definuje stav modelovaného systému. Označená Petriho síť je definována pěticí $(P, T, Pre, Post, m)$. Vývoj stavu modelu je způsoben přeskoky přechodů.

Přeskok přechodu (**firing**) je umožněn, jedině obsahuje-li každé jeho vstupní místo nejméně tolik tokenů, kolik je váha příslušné hrany. Říkáme, že přechod je uvolněn (**enabled**). Přechod bez vstupního místa je uvolněn vždy. Přeskok přechodu se skládá ze dvou kroků: vyjmutí tokenů ze vstupního místa a vložení tokenů do výstupního místa. Jejich počet je pro každý krok dán váhou příslušné hrany spojující místo a přechod. Matematicky vyjádříme značení nového stavu m' po přeskoku přechodu T_j jako

$$m'(P) = m(P) + Post\big(P, T_j\big) - Pre\big(P, T_j\big). \tag{1}$$

Zobecněná Petriho síť je plně autonomní, časově invariantní. To znamená, že přeskok přechodu je podmíněn jedině jeho uvolněním, a žádný vnější faktor (čas či synchronizace) na něj nemá vliv.

Nicméně uvolnění přechodu nemusí znamenat jeho okamžitý nebo nevyhnutelný přeskok. Je-li například v zobecněné Petriho síti uvolněno více přechodů současně (obrázek 2b), nelze určit pořadí přeskoků. Může nastat situace, kdy jedno místo je místem vstupním pro více přechodů zároveň. Na obrázku 2a je ukázána Petriho síť se strukturálním konfliktem ($m(P_1) = 0$). Pokud hodnota značení vstupního místa strukturálního konfliktu není dostatečně vysoká, aby mohli být přeskočeny všechny uvolněné přechody najednou, dochází k efektivnímu konfliktu – obrázek 2c. Model se v tomto případě chová nedeterministicky. Tato vlastnost může být nežádoucí, proto lze přijmout opatření, která zajistí deterministický výsledek přeskoku. Nejjednodušším způsobem řešení je přiřazení priorit jednotlivým přechodům. K přeskokům pak dochází v jejich nerostoucím pořadí – obrázek 2e.



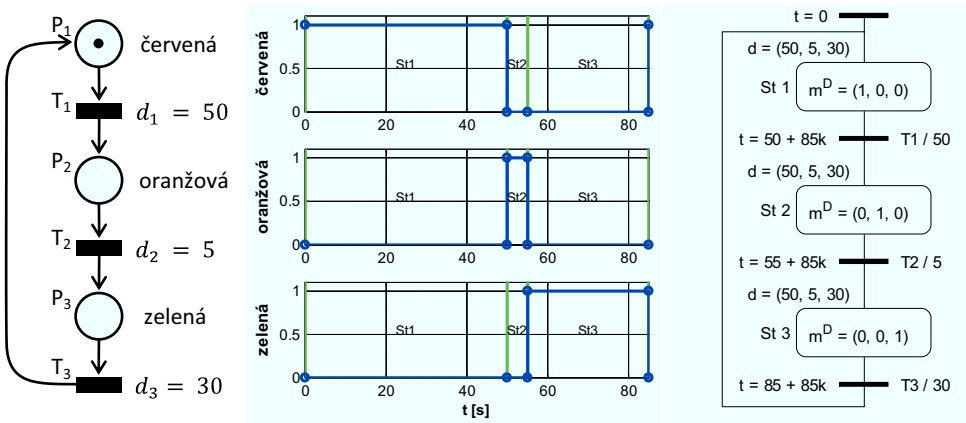
Obrázek 2 – Konflikt v zobecněné Petriho síti

Jako matematický model lze Petriho síť analyzovat v některých případech i bez generování grafu všech dosažitelných značení, čehož se využívá při verifikaci modelovaného reálného systému. Protože se analýzy tato práce netýká, nebudou zde vlastnosti ani metody verifikace vysvětlovány. Jejich rozbor lze nalézt v [Hanzálek I]. Za zmínku stojí vlastnost uváznutí (**deadlock**), která je u dopravních systémů důležitá. Uváznutí nastane, pokud žádný z přechodů není uvolněn.

2.2. Časované Petriho sítě (TPN)

Čas lze do modelu systému tvořeného Petriho sítí zakomponovat přiřazením časových úseků místům nebo přechodům. Oba způsoby jsou vzájemně převoditelné. Můžeme se na čas dívat dvěma základními způsoby:

- **Časové Petriho sítě**: k přeskoku přechodu může dojít kdykoliv během příslušného časového intervalu po jeho uvolnění.
- **Časované Petriho sítě**: k přeskoku může dojít, pokud je přechod uvolněn, po dobu trvání časového intervalu – vzniká tak zpoždění přeskoku po uvolnění.



Obrázek 3 – Časovaná Petriho síť – semafor

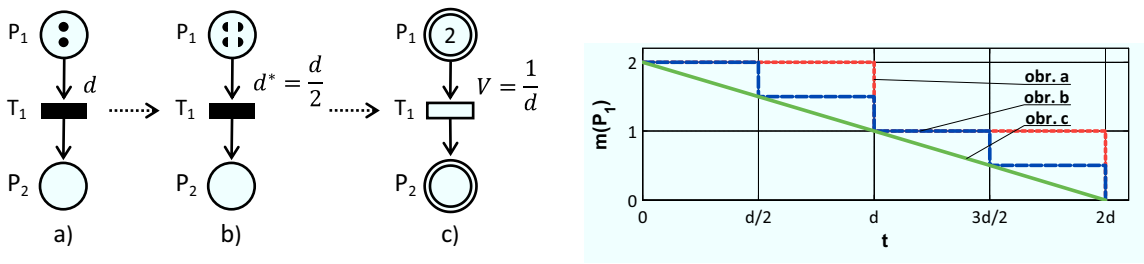
T-časované Petriho sítě, v nichž je čas přiřazen přechodům, lze dělit na sítě bez rezervace a s rezervací tokenů pro konkrétní uvolněný přechod. Rezervace zde znamená, že přechodu se příslušný počet tokenů přiřadí (rezervuje) hned po jeho uvolnění a po uplynutí doby dojde k přeskoku. Během té doby však rezervované tokeny nemohou uvolnit žádný jiný přechod.

Pro modelování dopravních systémů a jejich částí jsou vhodné T-časované Petriho sítě bez rezervace tokenů a proto jsou v této práci využity. Definujeme je jako $(P, T, Pre, Post, m, d)$, kde $d = \big(d(T_1), \dots, d(T_5)\big)$ je vektor časových zpoždění přiřazených jednotlivým přechodům. Jedná se o vektor zobrazení $T \rightarrow R_0^+$. Obdobně jako zobecněná Petriho síť může v časované Petriho síti nastat efektivní konflikt. A to tak, že zdrojové místo strukturálního konfliktu neobsahuje dostatečný počet tokenů pro přeskok všech připravených přechodů, tzn. přechodů, jež byly uvolněny minimálně po do-

bu svého zpoždění d_i. Řešení konfliktu je obdobné jako pro zobecněnou Petriho síť – diferenciace priorit přechodů.

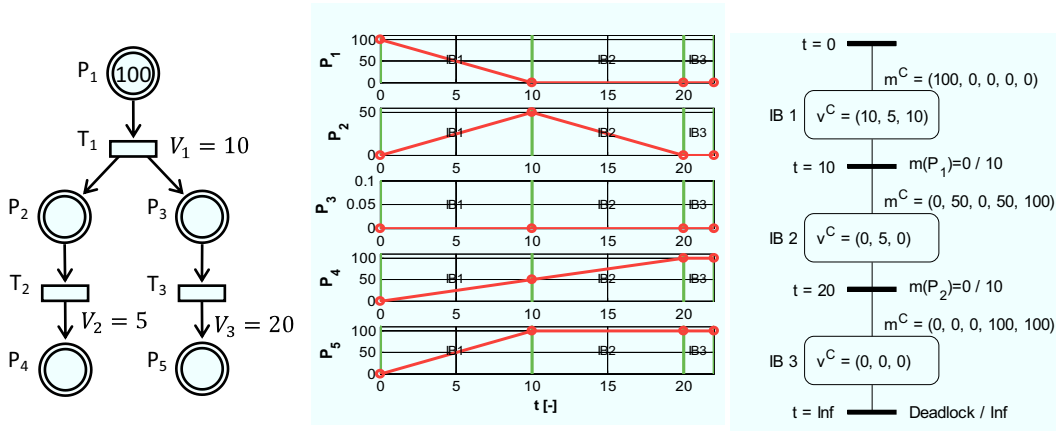
2.3. Spojité Petriho sítě (CCPN)

Značení diskrétních Petriho sítí (zobecněných, časovaných,…) je tvořeno celočíselným počtem tokenů. Omezený počet kombinací jejich rozmístění určuje stavy modelu. Rozdělením tokenů na k dílů získáme k-krát více dosažitelných stavů. Limitně se z tokenů stává spojitá veličina, která jednotkovou rychlostí „protéká“ přes přechody. Jak již bylo řečeno, vývoj Petriho sítí je rozdělen do stavů a stav diskrétní Petriho sítě je určen aktuálním značením. Oproti tomu stav ve spojité Petriho síti (označován jako IB-stav – Invariant Behaviour state) je určen rychlostí přechodů. Tím je umožněno provádět zhuštěnou aproximaci vývoje diskrétních Petriho sítí.



Obrázek 4 – Vztah diskrétní a spojité Petriho sítě

Vezmeme-li v úvahu časovanou Petriho síť se zpožděním přechodu d (obrázek 4) a provedeme-li s ní operaci popsanou v předchozím odstavci, získáme spojitou Petriho síť, jejíž přechod má rychlost $1/d$. Značení jejích míst nabývá libovolné nezáporné reálné hodnoty závislé na čase. Tato skutečnost umožňuje přesně modelovat spojité systémy, ale také aproximovat systémy diskrétních událostí a simulovat tak vývoj diskrétních Petriho sítí zhuštěnou formou. Původně dva stavy vývoje Petriho sítě z obrázku 4a tak nahradíme jedním IB-stavem spojité Petriho sítě.



Obrázek 5 – Spojitá Petriho síť a vývoj jejího značení

Matematicky je spojitá Petriho síť jednoznačně definována jako šestice $(P, T, Pre, Post, m, V)$. Prvky matic Pre a $Post$ mohou nabývat reálných hodnot: $P \times T \rightarrow R_0^+$. Vektor m je zde zobrazením $P \rightarrow R_0^+$. V je vektor zobrazení $T \rightarrow R_0^+$ a vyjadřuje maximální rychlosti přechodů. Skutečná rychlost přechodu T_j v čase je $v_j(t)$. Platí $v_j(t) \in \langle 0, V_j \rangle$. Podle [Hanzálek III] nyní definujme základní pojmy popisující spojitou Petriho síť a vývoj jejího značení. Na obrázku 5 je znázorněná jednoduchá spojitá Petriho síť, na které je možné, si je ukázat:

- P_i je značené (**marked**) místo, pokud $\mathrm{m}(P_i, t) > 0$
- P_i je zásobené (**supplied**) místo, existuje-li alespoň jeden přechod $T_j \in {}^\circ P_i$, který je uvolněn
- Přechod T_j je silně uvolněn (**strongly enabled**), pokud všechna místa $P_i \in {}^\circ T_j$ jsou značená. V čase t pak tí $v_j(t) = V_j$.
- Přechod T_j je v čase t slabě uvolněn (**weakly enabled**), pokud existuje místo $P_i \in {}^\circ T_j$, které není značené, je zásobené a zbývající místa z množiny ${}^\circ T_j$ jsou značená nebo zásobená.
- Podle [Hanzálek III] odpovídá rovnováha (**balance**) místa P_i v čase t derivaci jeho značení. Udává přírůstek (úbytek) v místě za jednotku času – rovnice (2).

- P_1 je značené místo, v čase $> t$ se i místo P_2 stane značeným
- P_2, P_3, P_4, P_5 jsou zásobená místa.
- T_1 je silně uvolněný přechod.
- T_2, T_3 jsou slabě uvolněné přechody: $v_2(t) = 5$ a $v_3(t) = 10$

$$B_i(t) = \sum_{T_j \in {}^\circ P_i} Post(P_i, T_j) \cdot v_j(t) - \sum_{T_k \in P_i{}^\circ} Pre(P_i, T_k) \cdot v_k(t). \tag{2}$$

Spojité Petriho sítě, kterým se dále tato práce bude věnovat, jsou spojité Petriho sítě s maximální rychlostí. Uvolněný přechod T_j má nejvyšší možnou rychlost. Ta je daná $v_j(t) = V_j$, jedná-li se o přechod silně uvolněný. Pokud se jedná o slabě uvolněný přechod, bude jeho nejvyšší možná rychlost minimem z maximální rychlosti a minima rychlostí zásobujících místa ${}^\circ T_j$. Tedy

$$v_j(t) = \min\left(V_j, \min_i \left(B_i(t) + v_j(t)\right)\right) \tag{3}$$

pro taková i, pro něž $\mathrm{m}(P_i, t) = 0$ a $P_i \in {}^\circ T_j$.

Pro výpočet skutečných rychlostí přechodů můžeme podle výše uvedených vztahů sestavit soustavu lineárních nerovnic. Tu lze s výhodou řešit pomocí lineárního programování (LP). Pro spojité Petriho sítě s maximální rychlostí maximalizujeme součet skutečných rychlostí slabě uvolněných přechodů.

Uvažujme množinu indexů silně uvolněných přechodů C_S, množinu indexů slabě uvolněných přechodů C_W a množinu indexů zásobených míst C_P, pak pro $\forall i \in C_P$, $\forall j \in C_W$ a $\forall n \in C_S$:

$$\max\left(\sum_j v_j(t)\right), \tag{4}$$

$$\begin{aligned} v_n(t) &= V_n, \\ v_j(t) &\leq V_j, \\ v_j(t) &\geq 0, \end{aligned} \tag{5}$$

$$\sum_{T_j \in P_i^\circ} v_j(t) \leq \sum_{T_j \in {}^\circ P_i} v_j(t) + \sum_{T_n \in {}^\circ P_i} v_n(t). \tag{6}$$

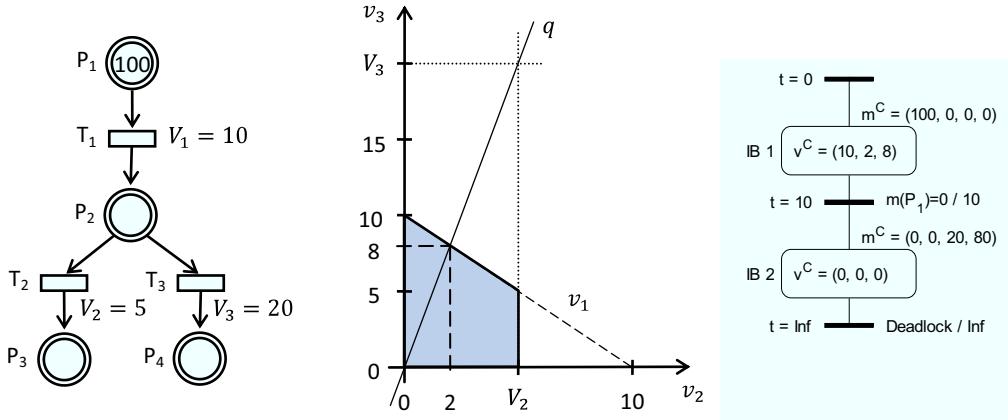
Pro výpočet skutečných rychlostí spojité Petriho sítě z obrázku 5 v jednotlivých IB-stavech řešíme LP úlohy (výsledek řešení je ukázán v obrázku):

IB-stav 1	IB-stav 2	IB-stav 3
$C_P = \{2,3,4,5\}$	$C_P = \{4\}$	$C_P = \emptyset$
$C_w = \{2,3\}$	$C_w = \emptyset$	$C_w = \emptyset$
$C_S = \{1\}$	$C_S = \{2\}$	$C_S = \emptyset$
$\max\big(v_2(t) + v_3(t)\big)$		
$v_1(t) = 10,$	$v_2(t) = 5$	
$0 \leq v_2(t) \leq 5,$		
$0 \leq v_3(t) \leq 20,$		
$v_2(t) \leq 10,$		
$v_3(t) \leq 10$		

Stejně jako v diskrétní Petriho síti, i zde může nastat konflikt. Spojitá Petriho síť má efektivní konflikt v místě strukturálního konfliktu P_i, je-li toto místo neoznačené, zásobené a rychlost, kterou je zásobeno, nestačí k uspokojení všech přechodů z množiny P_i°. Jedním z možných řešení je použití priorit stejně jako v předchozích typech Petriho sítí. Toto řešení však nemusí být vhodné.

Jak je popsáno v článku [Kutil I], můžeme s využitím lineárního programování vypočítat skutečné rychlosti přechodů v konfliktu proporcionálně k jejich maximální rychlosti. Způsob řešení konfliktu touto metodou ilustruje prostřední část obrázku 6. Máme-li pouze dva přechody v konfliktu, můžeme problém nakreslit do dvou dimenzí. Množina možných řešení je omezena maximálními rychlostmi přechodů v konfliktu V_2 a V_3 a skutečnou rychlostí zdrojového přechodu v_1. Maximální rychlosti svým

poměrem určují směrnici přímky q. Na obrázku 6 nacházíme v bodě jejího průsečíku s hranicí danou rychlostí v_1 rychlosti v_2 a v_3. Obecně může být množina řešení příliš omezena maximálními rychlostmi přechodů, průsečík pak je mimo tuto množinu a optimální bod neleží na přímce q.



Obrázek 6 – Spojitá Petriho síť s konfliktem, hledání skutečných rychlostí slabě uvolněných přechodů

Při sestavování problému LP vyjdeme ze vztahů (4), (5) a (6). Je potřeba vhodně rozšířit kritérium a doplnit nerovnice omezující řešení. Opět uvažujme množiny indexů a $\forall i \in C_P$, $\forall j \in C_W$ a $\forall n \in C_S$ a nově pro dvojice konfliktních přechodů $\forall k, \forall l \in C_w$, kde $k < l$:

$$\max\left(\sum_j v_j(t) - \epsilon \sum_{k,l} z_{kl}\right), \tag{7}$$

$$0 < \epsilon < \min_{k,l}\left(1, \frac{V_k}{V_l}\right). \tag{8}$$

Použijeme soustavy nerovnic (5) a (6) a přidáme nová omezení:

$$\begin{aligned} v_l(t) - v_k(t)\frac{V_l}{V_k} &\leq z_{kl}, \\ v_l(t) - v_k(t)\frac{V_l}{V_k} &\geq -z_{kl}. \end{aligned} \tag{9}$$

Nerovnice (9) – 2 nerovnice pro každý pár přechodů – jsou vyjádřením absolutní hodnoty $|v_l(t) - v_k(t)\, V_l/V_k|$, která představuje vzdálenost od přímky q. Jak záporné znaménko v kritériu ukazuje, hledáme minimální hodnotu absolutní hodnoty. Velikost konstanty ϵ podle vztahu (8) zaručuje, že význam minimalizace vzdálenosti nemá přednost před maximalizací hodnot skutečných rychlostí. Příklad sestavení LP problému pro Petriho síť z obrázku 6, v němž je uvedeno i řešení:

IB-stav 1	IB-stav 2
$C_P = \{2,3,4\}$	$C_P = \emptyset$
$C_w = \{2,3\}$	$C_w = \emptyset$
$C_S = \{1\}$	$C_S = \emptyset$
$\max(v_2(t) + v_3(t) - \epsilon \cdot z_{23})$	
$0 < \epsilon < 1/4$	
$v_1(t) = 10,$	
$0 \leq v_2(t) \leq 5,$	
$0 \leq v_3(t) \leq 20,$	
$v_2(t) \leq 10,$	
$v_3(t) \leq 10,$	
$-z_{23} \leq v_3(t) - 4v_2(t) \leq z_{23}$	

V článku [Kutil I] je uvedeno i vysvětlení speciálních případů, které při řešení konfliktu touto metodou mohou nastat – grafické znázornění by se pak od obrázku 6 lišilo. Kromě uvedeného způsobu existují i další metody výpočtu skutečných rychlostí slabě uvolněných přechodů. Ty lze nalézt například v knize [David-Alla]. V této práci však byla využita pouze proporcionální metoda využívající LP.

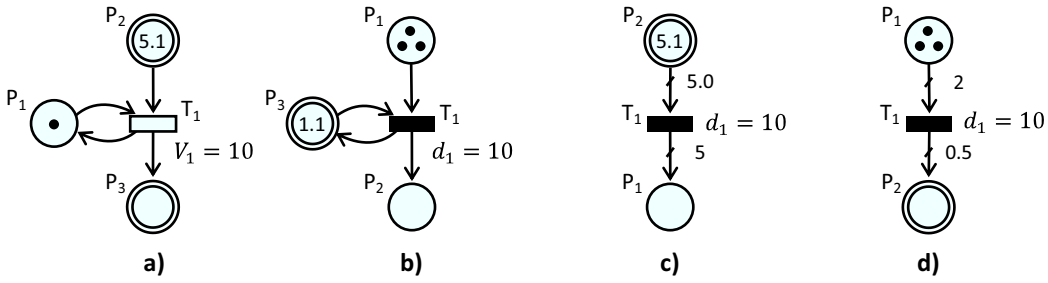
2.4. Hybridní Petriho sítě (HPN)

Při modelování dopravy jako toku vozidel je využití spojitých Petriho sítí velmi výhodné. Semafory na křižovatkách však tento tok přerušují. Obdobný je například vliv ventilu na průtok tekutiny. Tyto náhlé děje lze s úspěchem modelovat zavedením diskrétního prvku – vytvořením hybridní Petriho sítě. Jak je uvedeno v [David-Alla], existují 4 způsoby interakce diskrétních a spojitých Petriho sítí – obrázek 7. Značená hybridní Petriho síť je definována osmicí $(P, T, Pre, Post, m, d, V, h)$, kde h je „hybridní funkce“ $P \cup T \rightarrow \{D, C\}$ indikující, zda je uzel sítě diskrétní (patří do množiny P^D, resp. T^D) či spojitý (patří do množiny P^C, resp. T^C).

Funkce zobrazených Petriho sítí je intuitivní. Za zmínku však stojí, že diskrétní místo může být spojeno se spojitým přechodem jedině párem dvou hran stejné váhy tvořících smyčku. Musí být splněna podmínka:

$$Pre(P_i, T_j) = Post(P_i, T_j), \tag{10}$$

pro všechny dvojice P_i, T_j takové, že $P_i \in P^D$ a $T_j \in T^C$. Jedině tak bude totiž zaručeno $m(P_i) \in Z_0^+$. Konverze reálné hodnoty značení na celočíselný počet tokenů a opačně může být prováděna jedině přeskokem diskrétního přechodu.



Obrázek 7 – Hybridní Petriho sítě: a) spojitá řízená diskrétní, b) diskrétní řízená spojitou, c) konverze spojitá → diskrétní, d) konverze diskrétní → spojitá

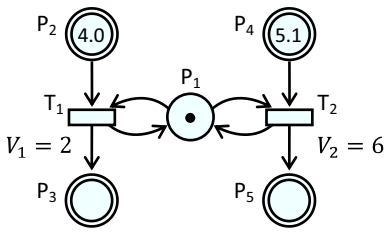
Vývoj hybridních Petriho sítí se stejně jako vývoj spojitých Petriho sítí dělí na IB-stavy. Ty jsou od sebe odděleny událostmi, které jsou v [David-Alla] definovány takto:

- **C1-událost** nastává, dojde-li k vyprázdnění spojitého místa, tedy $m(P_i) = 0$ pro $P_i \in P^C$.
- **D1-událost** je způsobena přeskočením diskrétního přechodu.
- **D2-událost** nastane při uvolnění diskrétního přechodu T_j naplněním spojitého místa P_i na potřebnou hodnotu, tzn. splněním $m(P_i) = Pre(P_i, T_j)$, kde $P_i \in P^C$ a $T_j \in T^D$.
- **C2-událost** je neprodlená změna chování spojité části sítě v důsledku jiné události, např. změna aktuální rychlosti v_j přechodu T_j přeskokem T_k, kde $T_j \in T^C$ a $T_k \in T^D$.

V hybridních Petriho sítích může nastat jak diskrétní, tak spojitý konflikt. Oba typy již byly probrány v předchozím textu. Novinkou je konflikt hybridní – obrázek 8. Stejně jako v ostatních případech, můžeme i zde přechodům přiřadit priority, jejichž diferenciací je konflikt vyřešen. Pokud jej nelze na základě priorit rozhodnout, je podle [David-Alla] potřeba zajistit „spravedlivé“ rozdělení sdíleného tokenu mezi konkurující si přechody. Představme si situaci z obrázku 8, jediný token může v jeden okamžik uvolňovat pouze jediný přechod. Ve výsledku tedy bude skutečná rychlost každého z přechodů poloviční. Obecně: máme-li zdrojové místo hybridního konfliktu P_i (diskrétní) a množinu indexů všech spojitých přechodů v konfliktu C_i, tak platí

$$v_j^* = \frac{m(P_i)}{\sum_k Pre(P_i, T_k)} \cdot v_j \tag{11}$$

pro $\forall k \in C_i$.



Obrázek 8 – Hybridní konflikt

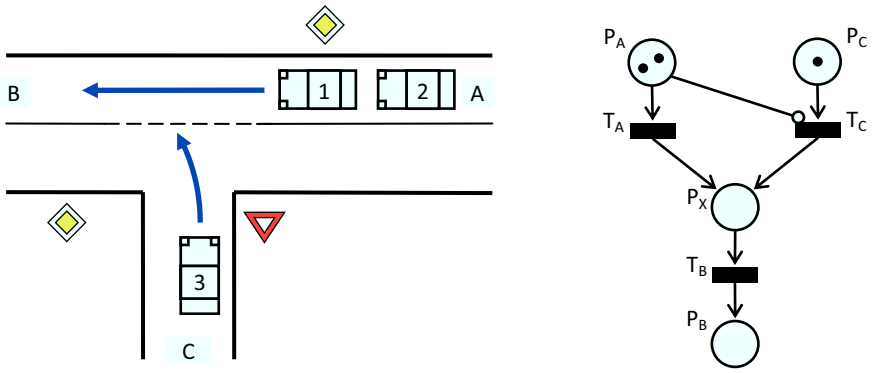
Známe-li skutečné rychlosti, jsme schopni vypočíst značení na konci aktuálního IB-stavu. Doba do jeho ukončení je dána tou událostí ze seznamu na předchozí stránce, která nastane nejdříve. Výpočet nového značení tedy provedeme výpočtem rovnice

$$m' = m + (Post - Pre) \cdot \begin{bmatrix} e \\ t \cdot v \end{bmatrix}, \tag{12}$$

kde e je sloupcový vektor s hodnotou 1 na indexu j pro všechny diskrétních přechody T_j připravené k přeskoku a s hodnotou 0 jinde, t je doba trvání IB-stavu a v sloupcový vektor skutečných rychlostí spojitých přechodů.

2.5. Rozšíření Petriho sítí o inhibiční hranu

Ve všech typech Petriho sítí popsaných v předcházejícím textu je přechod uvolněn po splnění podmínky $m(P_i) \geq Pre(P_i, T_j)$ nezávisle na typech P_i a T_j. Při modelování systémů reálného světa však může dojít k situaci, kdy chceme zakázat uvolnění přechodu, pokud značení místa dosáhne určité hodnoty. Obrázek 9 ukazuje příklad využití z oblasti simulování dopravy. Vzhledem k tomu, že inhibiční hrana slouží k testování hodnoty značení místa, má smysl jedině v orientaci místo → přechod.



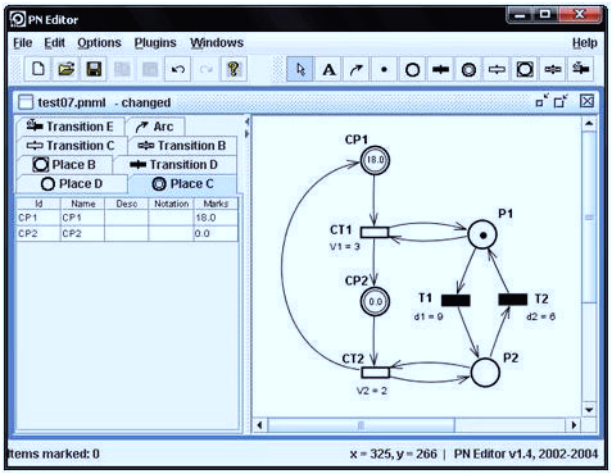
Obrázek 9 – Použití inhibiční hrany

Mějme inhibiční hranu $P_i \rightarrow T_j$ s váhou s_{ij}. Přechod T_j je uvolněn při splnění podmínky $m(P_i) < s_{ij}$. Platí, že pro $P_i \in P^D$ bude $s_{ij} \in Z^+$, pro $P_i \in P^C$ bude $s_{ij} \in R^+$. V případě jednotkové váhy hrany se zdrojovým místem $P_i \in P^D$ je přechod uvolněn jedině, pokud $m(P_i) = 0$. Takový případ se podle [David-Alla] nazývá „test nuly“ (zero test). Pro $P_i \in P^C$ mluvíme o testu nuly, pokud $s_{ij} = 0^+$. V obecném případě působí inhibiční hrana jako práh pro zakázání uvolnění přechodu libovolného typu.

3. Editace a reprezentace Petriho sítí

3.1. PN Editor

V této práci byl k vytváření modelů použit PN Editor. Jedná se o softwarový nástroj vyvinutý na katedře Řídicí techniky Fakulty elektrotechnické Českého vysokého učení technického. Je navržen pro vytváření a editaci diskrétních, spojitých i hybridních Petriho sítí. Našel uplatnění již v balíku PN Matlab Toolbox 2.0 popsaném v článku [Svádová]. Jedná se o jednoduchý editor s velmi intuitivním ovládáním. Umožňuje připojení programových modulů pro rozšíření funkčnosti. Jeho základní funkce byly pro tuto práci dostačující. Důležitou vlastností editoru je možnost ukládat editované Petriho sítě v PNML formátu.



Obrázek 10 – PN Editor: pravá část umožňuje kreslení, v levé části je umístěn editovatelný seznam prvků Petriho sítě

PN Editor nezná inhibiční hrany jako speciální podporovaný prvek. Protože je ale chceme používat, je nutné zajistit jejich jednoznačnou identifikaci a rozlišit je od hran klasických. Na počátku kapitoly 2 jsou matice Pre a $Post$ reprezentující hrany definovány tak, že jejich prvky musí být nezáporné. To nám poskytuje prostor pro uložení dat o hranách v celé škále množiny Z^- pro diskrétní a R^- pro spojité prvky Petriho sítě. Varianta reprezentace inhibiční hrany jako hrany se zápornou váhou byla vyhodnocena jako přijatelnější než například ukládání příznaků ve formě notace. Samozřejmě s maticemi Pre a $Post$ obsahujícími záporné hodnoty nelze provádět výpočty vývoje Petriho sítě. Informace o inhibičních hranách z nich tedy musí být před zpracováním extrahovány.

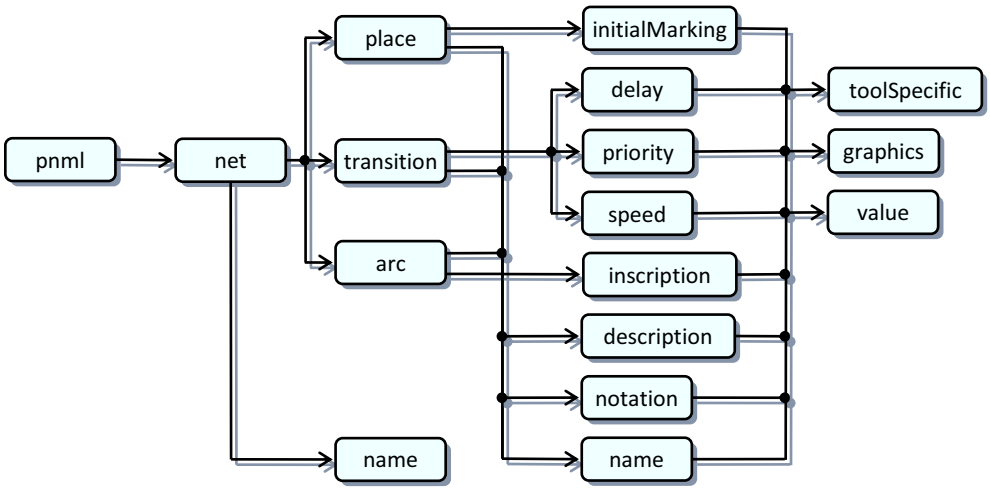
3.2. PNML

Petri Net Markup Language (PNML) je základní návrh standardu pro výměnu dat definujících Petriho sítě. Jedná se o jazyk založený na XML (eXtensible Markup Language) standardu. Názvy jeho elementů jsou definovány, což zajišťuje kompatibilitu s jinými nástroji zabývajícími se prací s Petriho sítěmi. Tato

kompatibilita je však omezená. Důvodem je neúplnost návrhu, který prozatím specifikuje pouze základní parametry. Reprezentaci ostatních parametrů musí zvolit uživatel PNML dodatečně, což klade překážky na sdílení dat v univerzálním, pevně definovaném formátu.

Na internetových stránkách autora PNML návrhu [PNML] lze nalézt přesnou a úplnou definici tvořenou jazykem Relax NG, který je hojně užíván pro definování a verifikaci XML dokumentů. Bližší informace o tomto nástroji můžeme nalézt ve zdroji [Relax NG]. Kompletní návrh je však ve své úplnosti pro náhled nepřehledný, proto zde uvádíme pouze výpis důležitých prvků popisujících Petriho síť. Jak je jistě známo, struktura XML a tedy i PNML je složena z elementů, které mohou obsahovat libovolný počet atributů. Jazyk PNML má definovány tyto elementy:

- **<pnml>** je základním elementem struktury, do něhož je definice Petriho sítě vložena,
- **<net>** element zapouzdřující členy Petriho sítě (uzly a hrany), má atributy **id** a **type**,
- **<place>** definuje místo Petriho sítě, jehož typ je určen pomocí atributu **id**,
- **<transition>** definuje přechod Petriho sítě, jeho typ je opět určen pomocí atributu **id**,
- **<arc>** reprezentuje hranu určenou atributem **id**. Rovněž musí obsahovat atributy **source** a **target** obsahující id svého zdroje a cíle,
- **<graphics>** určuje polohu nadřazeného prvku pro obecnou grafickou interpretaci,
- **<name>** definuje jméno prvku,
- **<description>** může obsahovat nepovinný vysvětlující komentář,
- **<notation>** pro uchování případné dodatečné informace prvku sítě,
- **<toolSpecific>** je element důležitý pro kompatibilitu a sdílení. Obsahuje totiž parametry prvku (nadřazeného elementu) potřebné pro softwarový nástroj. Ten je určen atributy **tool** a **version**. V případě PN Editoru definuje vzhled v grafické reprezentaci Petriho sítě.
- **<initialMarking>**, **<delay>**, **<speed>**, **<priority>** a **<inscription>** jsou elementy specifické pro svého vlastníka. Jejich význam je zřejmý.



Obrázek 11 – Znázornění hierarchie elementů ve struktuře PNML od vnějších prvků (zleva) po vnitřní

4. Simulace dopravních sítí pomocí Petriho sítí

Rozvoj služeb a průmyslu v naší přetechnizované době vede ke zvyšujícím se nárokům na dopravu. Rychle a bezpečně se dopravit kamkoliv je prioritou dní dnešní uspěchané doby. Kvalita i rozsah dopravních sítí jsou se vzrůstajícím provozem neustále zvyšovány. U velkých projektů, u nichž nelze v praxi vyzkoušet funkčnost, je simulace v podstatě jedinou možností ověření správnosti. Formalizmus Petriho sítí je vzhledem ke své variabilitě a spolehlivému matematickému aparátu často využíván pro testování i návrh řízení dopravy. Simulaci transportních sítí, na základě Petriho sítí se věnuje řada vědeckých pracovníků, inženýrů a matematiků po celém světě.

K vědcům hojně přispívajícím do této oblasti je Angela DiFebbraro, jež se společně se svými kolegy zabývá modelováním dopravních systému pomocí hybridních Petriho sítí. V [Di Febbraro III] autoři popisují modelování vozovky dálnice hybridní stochastickou Petriho sítí, kde je tok vozidel řízen diskrétními prvky v závislosti na aktuální situaci provozu. Ve článku [Di Febbraro II] jsou představeny jednoduché modely křižovatek tvořící modely dopravní sítě. V případě neřízené křižovatky je použita pouze pojitá Petriho síť, řízené křižovatky jsou simulovány hybridním modelem. Taktéž hybridní Petriho sítí je modelována křižovatka v [Di Febbraro I] a s ním souvisejícím [Di Febbraro IV]. První hovoří o důležitosti koordinace řízení křižovatek. Je zde ukázána případová studie průjezdu vozidla hromadné dopravy rozsáhlejší oblastí. Druhý je věnován podobné studii většího rozsahu založené na reálných datech.

Díky vzdálenostmi mezi křižovatkami existuje v silniční dopravě nezanedbatelné zpoždění. To popisuje [Mourni] na příkladu dopravníkového pásu. V článku je použita speciální spojitá Petriho síť obsahující přechody se zpožděním. Ačkoliv název článku [Júlvez] hovoří rovněž o spojité Petriho síti modelující křižovatku, ve skutečnosti se jedná hybridní model, v němž diskrétní část představuje světelnou signalizaci na obrázku modelu prezentovanou krabičkou s názvem „Traffic light“. Zajímavé je na tomto modelu zohlednění omezené kapacity ulic a sekcí křižovatky. Článek [Tolba] ukazuje princip převodu modelu křižovatky tvořeného časovanou Petriho sítí na Petriho síť spojitou. Spojitý model je i zde řízen diskrétně. Oba přístupy jsou porovnány na simulačních příkladech.

Pomocí stochastické časované Petriho sítě je modelována křižovatka (pohyb vozidel i řídicí systém) v článku [Wang]. Křižovatka je řízena dvěma fázemi. Model je zajímavý tím, že představuje dvě oddělené křižovatky (pro každou fázi jednu). Jejich interakce jsou aproximovány statistickým modelem. Tím je zde dosaženo výrazně nižší výpočetní složitosti.

Dalším zajímavým příspěvkem je studie [Gallego] ukazující model řídicího systému křižovatky umožňující dynamické přepínání řídicího plánu a tím zefektivnění provozu.

Model křižovatky využívající inhibiční hrany je prezentován v [Tzes]. Obsahuje rovněž pseudokód algoritmu pro simulaci dopravního systému pomocí mikroprocesorů a důkaz vyloučení uváznutí uvedeného modelu. Barvenou časovanou Petriho síť využívá pro modelování své řízené křižovatky [Doto-

li]. Svůj model podkládá rozsáhlou studií a porovnáním výsledků simulace s daty získanými měřením na reálné křižovatce.

Existuje i celá řada principů a nástrojů modelujících dopravu, které z formalizmu Petriho sítí nevycházejí. Přehled několika z nich lze nalézt v práci [Stibor]. Obecně strategiím řízení je věnován článek [Papageorgiou], v němž jsou představeny problémy vzrůstající hustoty městského provozu i dopravy po dálnicích a nabízí několik matematicky ověřených strategií efektivního řízení.

5. Simulátor Petriho sítí

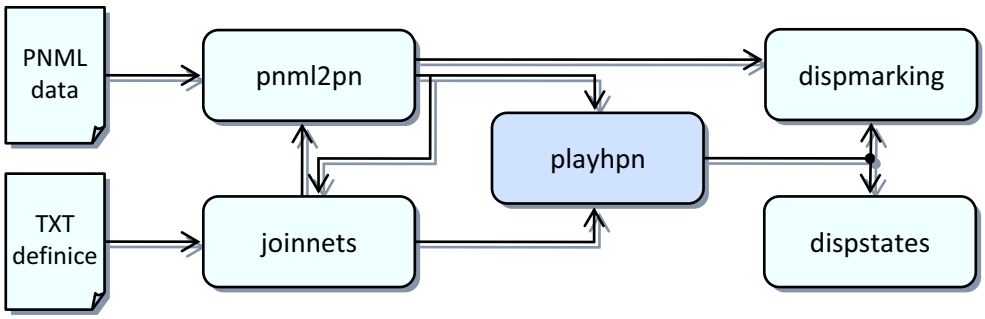
Úkolem této práce je simulovat dopravní sítě pomocí Petriho sítí. Aparát potřebný pro modelování je popsán v kapitole 2. Poznatky stručně sepsané v kapitole 4 přinášejí inspiraci a námět pro realizaci vhodných modelů různých typů. Pomocí nástrojů zmíněných v kapitole 3 jsme nyní schopni Petriho sítě snadno a pohodlně vytvářet a exportováním do definované podoby uchovávat. V této fázi jsme tedy schopni modelování provádět, avšak abychom byly schopni ověřit funkčnost a přesnost teoretických obrazů reálných křižovatek, musíme mít k dispozici prostředek, jímž „oživíme“ grafy, které jsou prozatím pouze obrázky.

Protože výpočty stavů vývoje Petriho sítě jsou převážně operacemi maticovými, byl pro realizaci simulátoru zvolen MATLAB® – prostředí a programovací jazyk pro technické výpočty od společnosti MathWorks. V rámci systému MATLAB je k dispozici nástroj XSLT, jenž slouží ke snadnému zpracování dat v XML formátu. Toho bylo již využito při realizaci PN Matlab Toolboxu 2.0 zmíněného v předchozí kapitole. Součástí celého balíku MATLAB je i Optimalizační toolbox nabízející prostředky k řešení problémů lineárního programování, pomocí nichž bude potřeba řešit výpočet skutečných rychlostí ve spojité Petriho síti. Poslední výhodou zvoleného způsobu řešení, kterou je zde uvedena, je skutečnost, že MATLAB je na katedře Řídicí techniky hojně využíván, a proto se jedná o prostředí dobře známé jak pro případné rozšíření simulátoru, tak především pro další zpracování dat generované samotným simulátorem nebo některým z jeho modulů. MATLAB je prostředí katedrou poskytované studentům pro studijní účely zdarma.

Přestože MATLAB umožňuje objektové programování, byla pro reprezentaci Petriho sítě zvolena forma maticová tak, jak je uvedena v kapitole 2. Důvodem je jednoduší implementace. Objektově orientovaný přístup by v tomto případě patrně nepřinesl žádnou výhodu.

5.1. Složení simulátoru

Celý implementovaný balík simulátoru se skládá z pěti primárních funkcí, které můžeme rozdělit do tří částí: načtení dat, generování vývoje Petriho sítě, prezentace vypočtených dat.



Obrázek 12 – Schematické znázornění modulů simulátoru

5.2. Načtení dat

5.2.1. pnml2pn

Pro načtení Petriho sítě z uloženého PNML slouží funkce ***pnml2pn***. Jejím jediným povinným vstupem je název zdrojového souboru včetně jeho cesty. Funkce pro transformaci XML struktury do užitečných dat využívá ***xslt*** modul, který je součástí MATLABu. K tomu je potřeba speciální předpis nazývaný styl uložený v souboru XSL. Zvláštností PN Editoru je, že rozlišuje typ Petriho sítě podle tvaru id jeho prvků. Tomu jsou přizpůsobeny i XSL soubory. Pro správné dekódování sítí vycházející z jiného zdroje stačí drobné přizpůsobení stylů odlišnostem v PNML. Základní myšlenka funkce *pnml2pn* je založena na balíku PN Matlab Toolbox 2.0. Informace o něm lze nalézt v [Svádová]. Forma XSL stylů použitá v této práci se od svého vzoru liší jen málo.

Funkce *pnml2pn* je samozřejmě připravena pro použití také s jinými typy PNML, než je ten produkovaný PN Editorem. Stačí pouze přidat XSL styly pro dekódování nového PNML formátu a při volání zadat druhý, nepovinný argument představující místo uložení XSL. Podle něj je pak rozhodnuto, pomocí jakých definic bude uložená Petriho síť do MATLABu načtena. Je tak zajištěna kompatibilita s mnoha grafickými editory Petriho sítí. Můžeme například využít také další nástroj vytvořeným na katedře Řídicí techniky – VSPNE ([VSPNE]). Který je původně určený pro práci se stochastickými Petriho sítěmi, ale v rámci [Stibor] byla jeho funkčnost rozšířena na podporu spojitých Petriho sítí.

Samotný algoritmus *pnml2pn* se kromě čtení zdrojových dat zabývá už jen transformací řetězce znaků do numerické podoby, přičemž je potřeba brát ohled i na speciální hodnoty typů 0^+ a ∞. Jednou možností je zadat hodnoty ***eps***[1] nebo ***inf*** přímo při vytváření Petriho sítě. V případě zadání znaků 0+, je tato hodnota převedena automaticky. Výstupem jsou matice a vektory definující Petriho síť:

- Pre matice vyjadřující spojení místo → přechod popsaná v kapitole 2
- $Post$ matice vyjadřující spojení přechod → místo
- m_0 počáteční značení Petriho sítě
- $Prio^T$ vektor priorit přechodů (diskrétních i spojitých)
- d vektor zpoždění časovaných přechodů
- V vektor maximálních rychlostí přechodů spojité Petriho sítě
- h^P, h^T hybridní funkce popsaná v podkapitole 2.4
- $name^P, name^T$ seznam názvů míst a přechodů
- not^P, not^T seznam notací míst a přechodů
- $desc^P, desc^T$ seznam komentářů k místům a přechodům

[1] Hodnota minimálního rozdílu mezi dvěma čísly v plovoucí řádové čárce: $2{,}2204 \cdot 10^{-16}$
[2] Simulace byla provedena na osobním počítači vybaveném Intel Core Duo CPU T2350 @1.86 GHz, 1 GB RAM.

Matice Pre a $Post$ jsou pro zvýšení efektivity zpracování, především z hlediska paměťové náročnosti, uloženy jako matice řídké (*sparse*).

5.2.2. joinnets

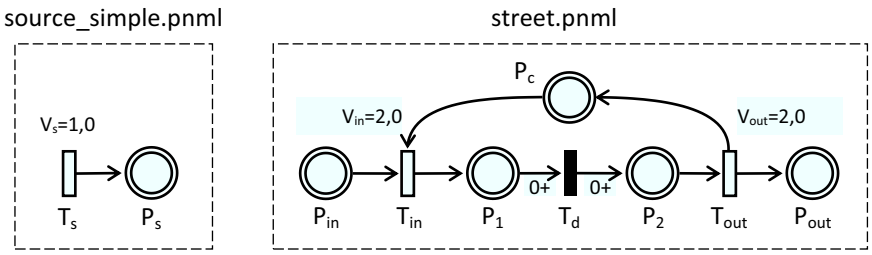
Chceme-li studovat systémy rozsáhlejšího rázu, je vhodné modelovat jejich části samostatně a pro finální simulaci je spojit. Příkladem může být právě model oblasti z kapitoly 8, který je složen z Petriho sítí reprezentující jednotlivé křižovatky, ulice různých délek a vstupní prvky, jimž jsou přiřazeny proměnné signály. Toto spojování provádí funkce ***joinnets*** na základě předpisu uloženém v textovém souboru, jehož název je zadán jako její jediný argument. Podobným způsobem je zadáváno spojování Petriho sítí v projektu [Stibor]. Právě touto prací byla syntaxe spojení inspirována.

Ukažme si nyní příklad způsobu spojování dvou Petriho sítí. Na obrázku 13 jsou zobrazeny dva modely uložené v adresáři „..\library“ v samostatných PNML souborech. Kód 1 ukazuje syntaxi předpisu, podle nějž funkce *joinnets* jednotlivé sítě zpracovává. Spojování se skládá ze tří fází. Tomu odpovídají tři operátory obsažené v předpise:

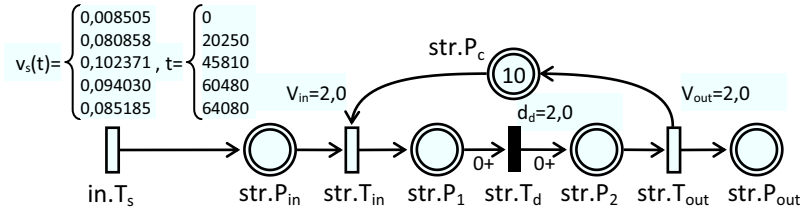
- **<** **přiřazení** – Petriho síť je přiřazena jménu. To je použito jako předpona názvů všech uzlů sítě.
- **=** **spojení** – uzly načtených sítí spojených tímto znakem jsou sjednoceny (platí i pro více než jen dva). Takto vzniklý uzel si zachová název a notaci uzlu prvního, komentáře jsou spojeny v jeden.
- **:** **nastavení** parametru přechodu – diskrétnímu přechodu je přiřazen čas, spojitému přechodu maximální rychlost. Tyto parametry mohou být proměnné, zadáme-li soubor hodnot. Souborem je míněn prostý textový soubor. V případě diskrétního přechodu, jemuž přiřazujeme proměnné zpoždění, obsahuje každý řádek souboru „čas“. Ke změně parametru pak dojde po každém přeskoku tohoto přechodu. Pro spojitý přechod každý řádek obsahuje nový parametr ve formě „čas, nová hodnota“. Změna parametru pak nastane ve stanoveném čase.

```
% Načtení částí PN
in < ..\library\source_simple.pnml
str < ..\library\street.pnml

% Spojení jednotlivých částí
str.P_in = in.P_s

% Nastavení konstantních parametrů
str.P_c : 10
str.T_d : 2

% Nastavení měnících se parametrů
in.T_s : ..\data\smichov\detektor_31.txt
```

Kód 1 – Definice propojení Petriho sítí a přiřazení proměnného signálu



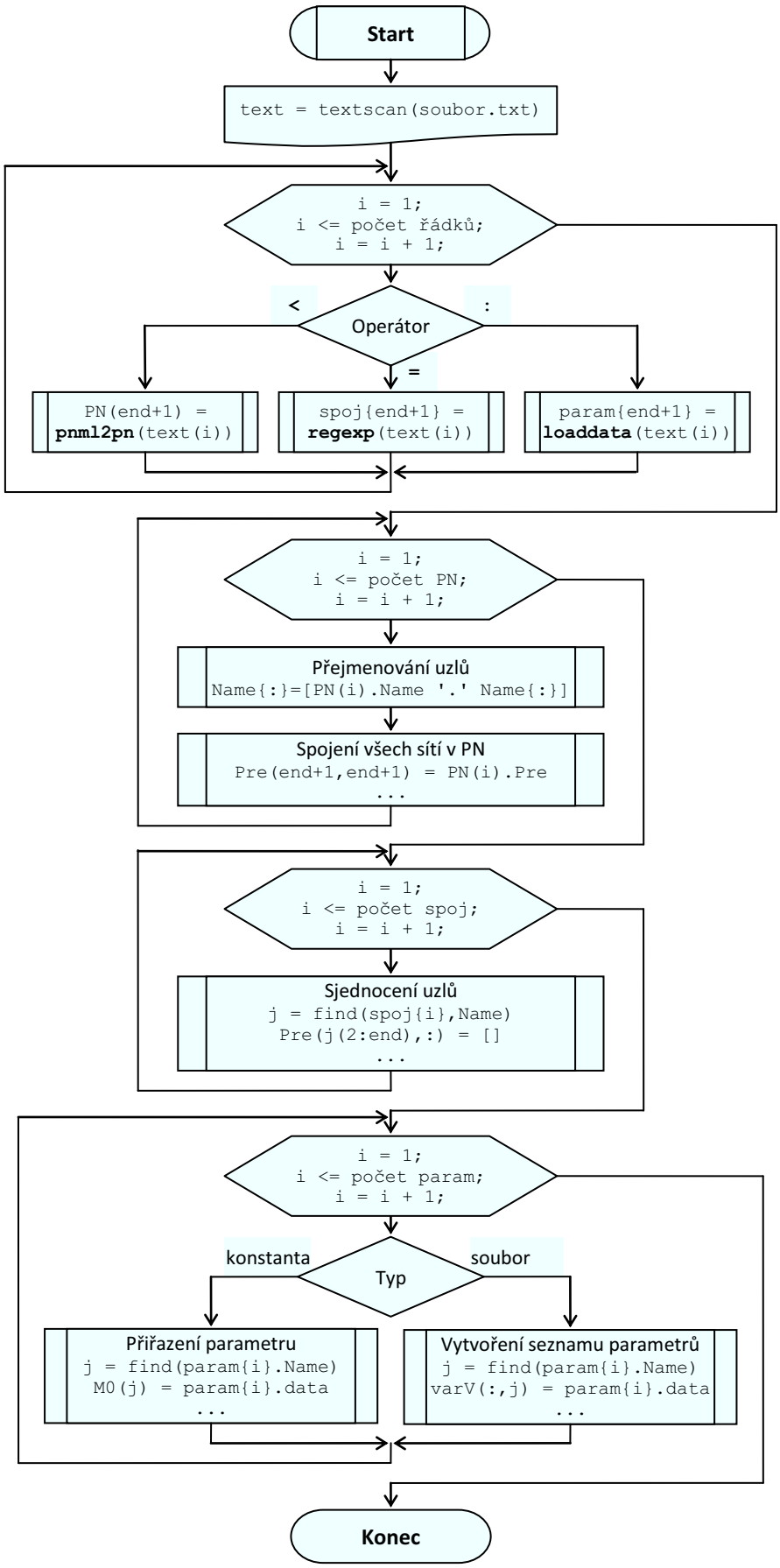
Obrázek 13 – Petriho sítě pro simulaci ulice: zdroj (vlevo) a dopravní zpoždění (vpravo)



Obrázek 14 – Výsledek spojení podle kódu 1

Algoritmus funkce ve formě vývojového diagramu je ukázán na stránce 20. Je z něj patrné, že po načtení konfiguračního předpisu (např. kód 1) proběhnou čtyři cykly zpracování:

1. načtení Petriho sítí, dat a parametrů definovaných v konfiguraci,
2. spojení všech Petriho sítí do jednoho celku,
3. sjednocení předepsaných uzlů – zde poznamenejme tyto fakta:
    a. může být naráz spojováno více uzlů stejného typu než dva,
    b. sjednocením uzlů zůstanou zachovány parametry prvního z nich,
    c. sjednocením hran, které může nastat v důsledku bodu b, zůstane zachována hrana jediná s váhou maximální ze všech sjednocovaných hran,
4. přiřazení nových parametrů – konstant či proměnných průběhů.



Vývojový diagram 1 – Algoritmus funkce *joinnets*

5.3. Generování vývoje Petriho sítí

Po načtení Petriho sítě reprezentované PNML souborem máme k dispozici její matematické vyjádření $(Pre, Post, m_0, d, V, h^P, h^T, Prio^T)$, kde $Prio^T$ je vektor priorit přechodů, význam ostatních vektorů a matic byl vysvětlen již v kapitole 2. Na základě těchto parametrů lze Petriho síť vyšetřovat. Naším cílem je generovat vývoj stavů dané Petriho sítě – simulovat modelovaný systém.

5.3.1. playhpn

Hlavním prvkem celého simulačního softwarového balíku, který v rámci této práce vznikl, je funkce ***playhpn***. Jedná se o komplexní algoritmus schopný zpracovávat diskrétní, časované, spojité i hybridní Petriho sítě s rozšířením zahrnujícím inhibiční hrany. Celý tento simulátor je obsažený v jediném souboru a kromě řešiče problémů lineárního programování z optimalizačního toolboxu využívá ke své činnosti pouze funkce obsažené v základní distribuci programovacího prostředí MATLAB. K dílčímu zpracování dat v průběhu generování vývoje jsou použity pouze funkce interní, uložené v souboru primární funkce *playhpn*.

Jako vstupy simulátor vyžaduje osmici matematické definice Petriho sítě uvedenou na počátku této kapitoly, která může být doplněna o další data ovlivňující průběh simulace:

- **Pre**, **Post** – matice Pre a $Post$ definující hrany v Petriho síti. V případě, že některý z prvků matice Pre obsahuje zápornou hodnotu, je mezi takto určenými uzly detekována inhibiční hrana s váhou odpovídající absolutně vzaté záporné hodnotě.
- **M0** – počáteční značení sítě m_0.
- **D**, **V** – vektory zpoždění diskrétních a maximálních rychlostí spojitých přechodů d a V.
- **PrioT** – vektor priorit jednotlivých přechodů $Prio^T$. Stejně jako v článku [Kutil I] dáváme přednost přechodům s vyšším číslem značícím prioritu.
- **TypeP** – vektor typů míst. Typ je určen na základě id místa tak, jak je definováno v PNML. Styl, který je použit pro čtení tohoto parametru z PNML rozeznává celkem 3 typy: diskrétní, spojité a dávkové místo. Jedná se o všechny typy, které pomocí PN Editoru můžeme kreslit. Funkce playhpn dokáže pracovat pouze s diskrétními a spojitými místy.
- **TypeT** – vektor typů přechodů. Typ přechodu je opět určen podle id. Také je brán zřetel na hodnotu času přiřazeného diskrétnímu přechodu. Pomocí *xslt* můžeme rozdělit přechody na 5 různých druhů: diskrétní s nulovým časem, diskrétní časovaný, rozšířený, spojitý a dávkový přechod. Funkce *playhpn* však pracuje pouze diskrétními přechody s nulovým i nenulovým časem a s přechody spojitými. Ostatní jsou ve stylech *xslt* zachovány a tedy *pnml2pn* je dokáže správně určit.

- → **NameP**, **NameT** – seznamy názvů uzlů potřebné pro označení událostí, které během vývoje v Petriho síti nastávají.
- → **varD**, **varV** – matice představující proměnné parametry přechodů – zpoždění a rychlosti. V obou případech první řádek obsahuje pořadová čísla přechodů, kterým jsou data přiřazena. Parametry nejsou povinné.
- → **Tmax** – simulátor musí nezbytně mít dispozici prostředek, kterým ukončí svou činnost v případě, že vývoj zadané Petriho sítě nekončí v přijatelné době uváznutím ani cyklem a může pokračovat až do nekonečna. Pokud se tedy vývoj modelu dostane až k nastavenému času, je simulace ukončena událostí „Tmax“. V případě, že se jedná o obecnou Petriho síť, která nezahrnuje čas, představuje hodnota *Tmax* maximální počet stavů. Tento parametr není povinný. V případě, že není zadán, je automaticky nastaveno *Tmax* = 1000.
- → **type** – pomocný parametr k určení simulace typu sítě. Může nabývat hodnot *'dpn'*, *'cpn'* nebo *'hpn'*. Tento parametr využijeme například, chceme-li simulovat čistě diskrétní Petriho síť jako síť hybridní. Rozdíl bude ve vyhodnocování událostí a tím i v lehce odlišném dělení vývoje do stavů. Argument není povinný. Není-li zadán, simulátor automaticky vyhodnotí typ Petriho sítě.
- → **conflict** – způsob řešení konfliktu v diskrétní Petriho síti v případě, že nelze rozhodnout na základě priorit. Může nabývat hodnot nebo *'order'* nebo *'random'*. V prvním případě je determinizmus přeskoků založen na pořadových číslech přechodů. V druhém případě je generováno náhodné číslo, podle jehož hodnoty je rozhodnuto, který z konfliktních přechodů bude přeskočen. Argument není povinný. Jeho výchozí hodnota je *'random'*.
- → **linprogtol** – tolerance řešení funkce *linprog* řešící problém lineárního programování, jímž je vypočítána skutečná rychlost slabě uvolněných přechodů. Argument není povinný. Výchozí hodnota je *linprogtol = 1e-12*.

Výstupem simulátoru je šest dále zpracovatelných proměnných popisujících vývoj modelu:

- ← **markings** – matice obsahující vývoj značení diskrétních i spojitých míst. Každý řádek odpovídá jednomu místu, sloupec konkrétnímu stavu (IB-stavu v případě spojité či hybridní Petriho sítě). První sloupec této matice odpovídá počátečnímu značení sítě m_0.
- ← **delays** – matice využitá pouze pro diskrétní a hybridní Petriho sítě. Její řádky korespondují s diskrétními přechody, sloupce stejně jako v předchozím případě odpovídají aktuálnímu času zpoždění jednotlivých přechodů pro počátek každého stavu vývoje.
- ← **speeds** – využito pouze pro spojité a hybridní Petriho sítě. Matice obsahuje skutečnou rychlost každého spojitého přechodu (řádky) během každého z IB-stavů (sloupce) představujících vývoj sítě.

- **events** – seznam událostí, které způsobují změnu stavu. Jedná se o datovou jednotku typu cell s délkou odpovídající počtu stavů. Každý její prvek obsahuje řetězec znaků označující děj, který aktuální stav ukončí.
- **eventsT** – seznam časů, po kterých došlo k uskutečnění událostí zaznamenaných v předchozí proměnné. Tento seznam neobsahuje absolutní čas, ale je v přírůstkové formě.
- **loopback** – byl-li během generování vývoje detekován cyklus, je zde uloženo pořadové číslo stavu, do kterého se cyklus uzavírá. Není-li cyklus, obsahuje tato proměnná hodnotu nekonečno (*inf*).

Vývojový diagram 2 ukazuje pohled na algoritmus simulační funkce. Jeho první část je věnována předzpracování zadaných dat. To je nezbytné v případě použití inhibičních hran a výhodné pro přehlednost, usnadnění implementace a také z hlediska efektivity cyklicky prováděných výpočtů.

Stručně zopakujme způsob reprezentace inhibičních hran podrobněji popsaný v podkapitole 3.1. Inhibiční hrana $P_i \rightarrow T_j$ je po převedení PNML struktury do maticových proměnných reprezentována zápornou hodnotou umístěnou na příslušné místo v matici $Pre(P_i, T_j)$. Záporné hodnoty jsou vyhledány, nahrazeny nulou a parametry inhibičních hran uloženy do jiné proměnné. Později mohou být v algoritmu využity při hledání uvolněných míst. Dále již pracujeme s maticí Pre s nezápornými prvky.

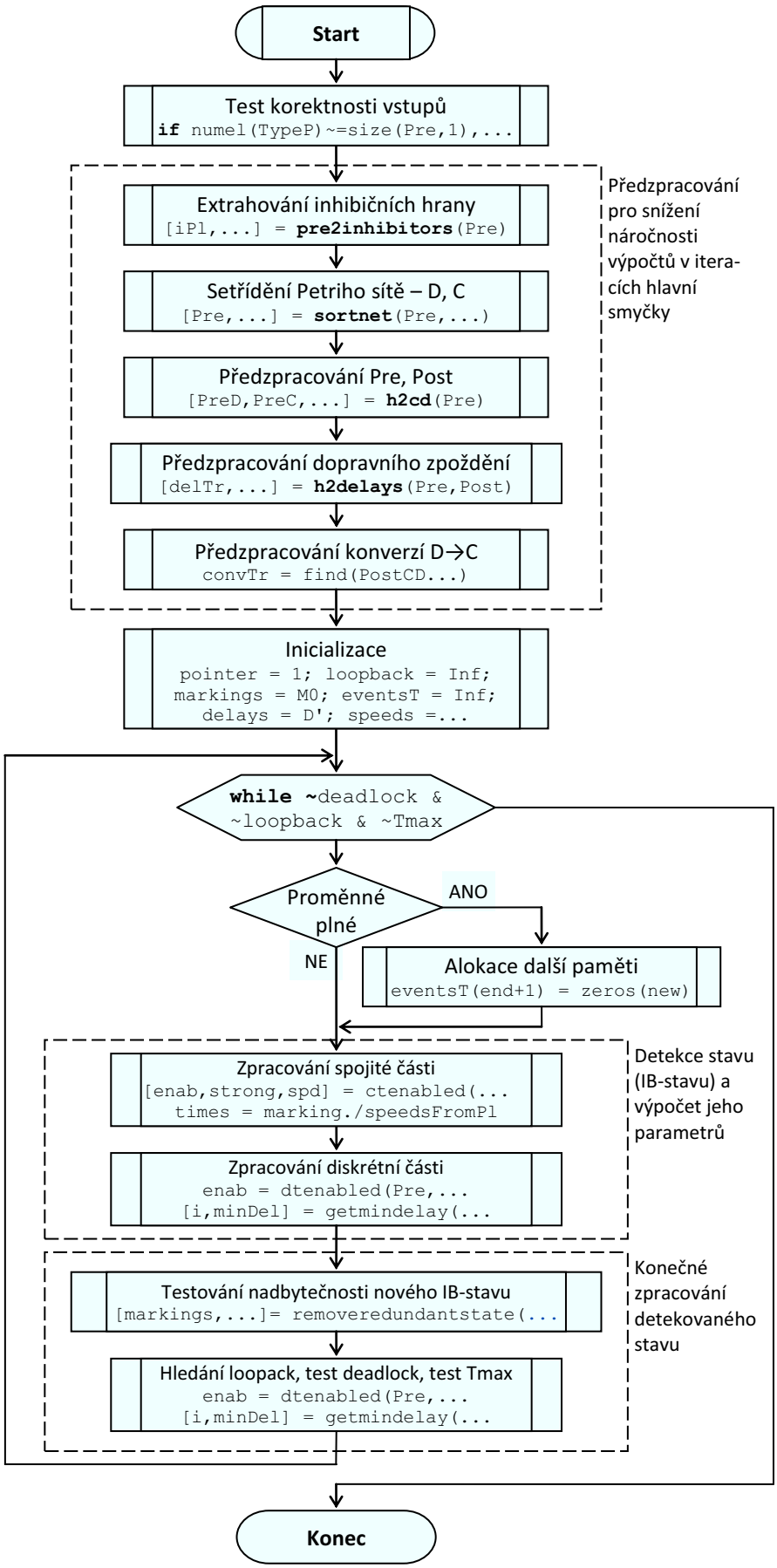
Jsou-li matice a vektory představující Petriho síť setříděny podle typů uzlů, můžeme rozdělit matice Pre, $Post$ a vektor značení m na části diskrétní, spojitou a v případě matic dvě hybridní:

$$Pre = \begin{bmatrix} Pre^D & Pre^{DC} \\ Pre^{CD} & Pre^C \end{bmatrix}, \qquad Post = \begin{bmatrix} Post^D & Post^{DC} \\ Post^{CD} & Post^C \end{bmatrix}, \qquad m = \begin{bmatrix} m^D \\ m^C \end{bmatrix}$$

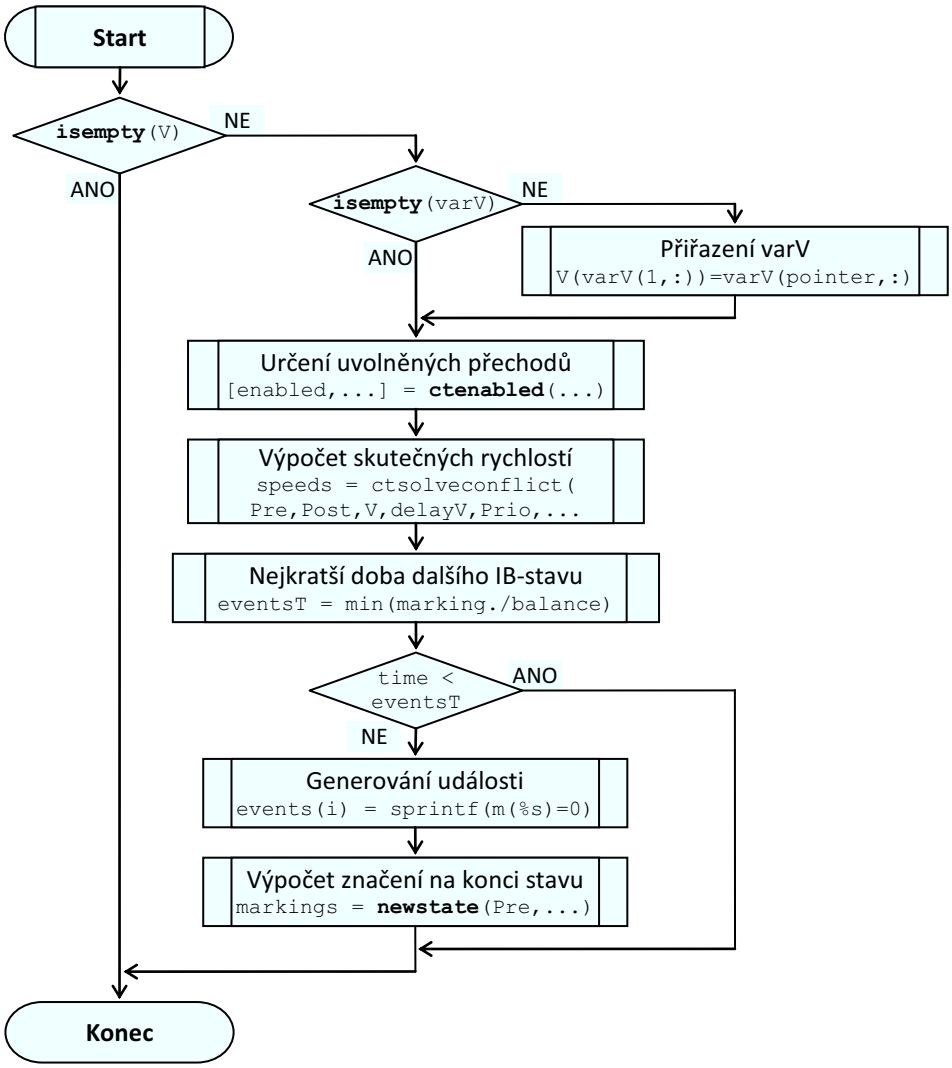
Velkou část operací pak provádíme pouze nad maticemi částečného rozměru.

Pro dopravní zpoždění jsou v úvodním předzpracování připraveny proměnné. Rovněž je výhodné připravit si proměnné i pro konverze $T_j \rightarrow P_k$, kde $P_k \in P^C$ a $T_j \in T^D$.

Stavy (v případě spojité či hybridní sítě IB-stavy) vývoje sítě jsou detekovány v hlavním cyklu algoritmu ve dvou blocích kódu. Každý z nich je podrobně ilustrován vlastním vývojovým diagramem. Spojitou část představuje diagram 3, diskrétní je znázorněna vývojovým diagramem 5.



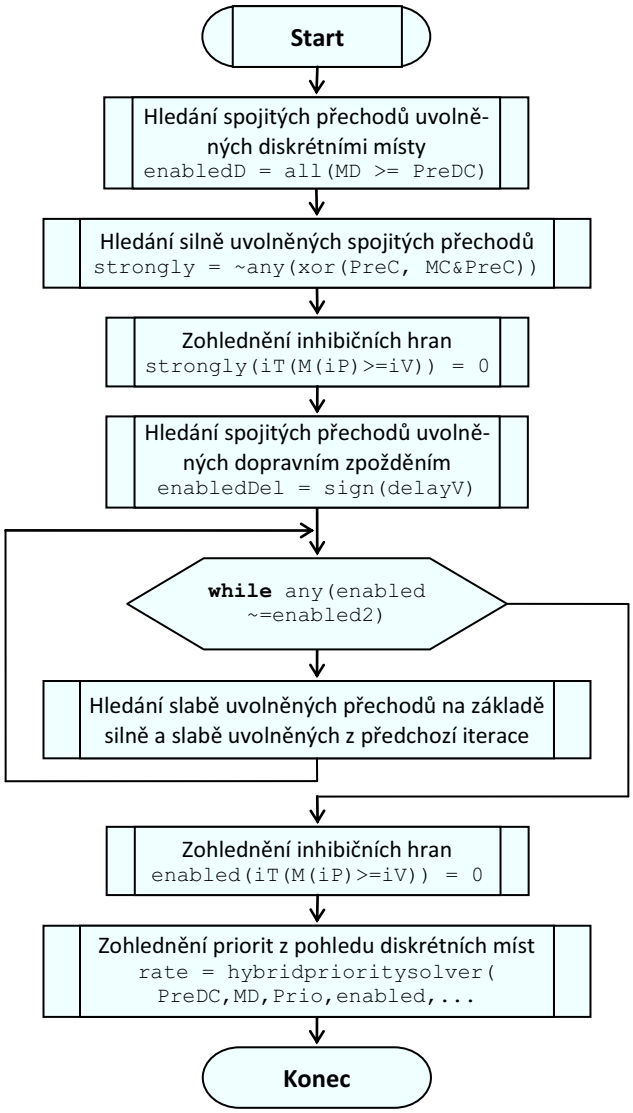
Vývojový diagram 2 – Základní algoritmus funkce *playhpn*



Vývojový diagram 3 – Spojitá část funkce *playhpn*

Za nejzajímavější a nejsložitější části algoritmu, jehož základní kostra je nastíněna ve vývojovém diagramu 3, lze považovat určení uvolněných přechodů a výpočet skutečných rychlostí aktuálního IB-stavu. Na diagramu 4, znázorňujícím funkci ***ctenabled***, je patrný postup detekce uvolněných spojitých přechodů. Algoritmus postupně zohledňuje všechny části Petriho sítě a logickou kombinací jejich vlivů detekuje silně a slabě uvolněné přechody. Způsob výpočtu skutečných rychlostí spojitých přechodů je vysvětlen i ukázán na příkladu v podkapitole 2.3. Implementovaný algoritmus generuje potřebné podmínky, které dosadí do řešiče lineárního programování. Stojí však za zmínku, že při sestavování LP problému je navíc potřeba zahrnout do omezujících rovnic také vliv prvků případného dopravního zpoždění a hybridního konfliktu podle obrázku 8.

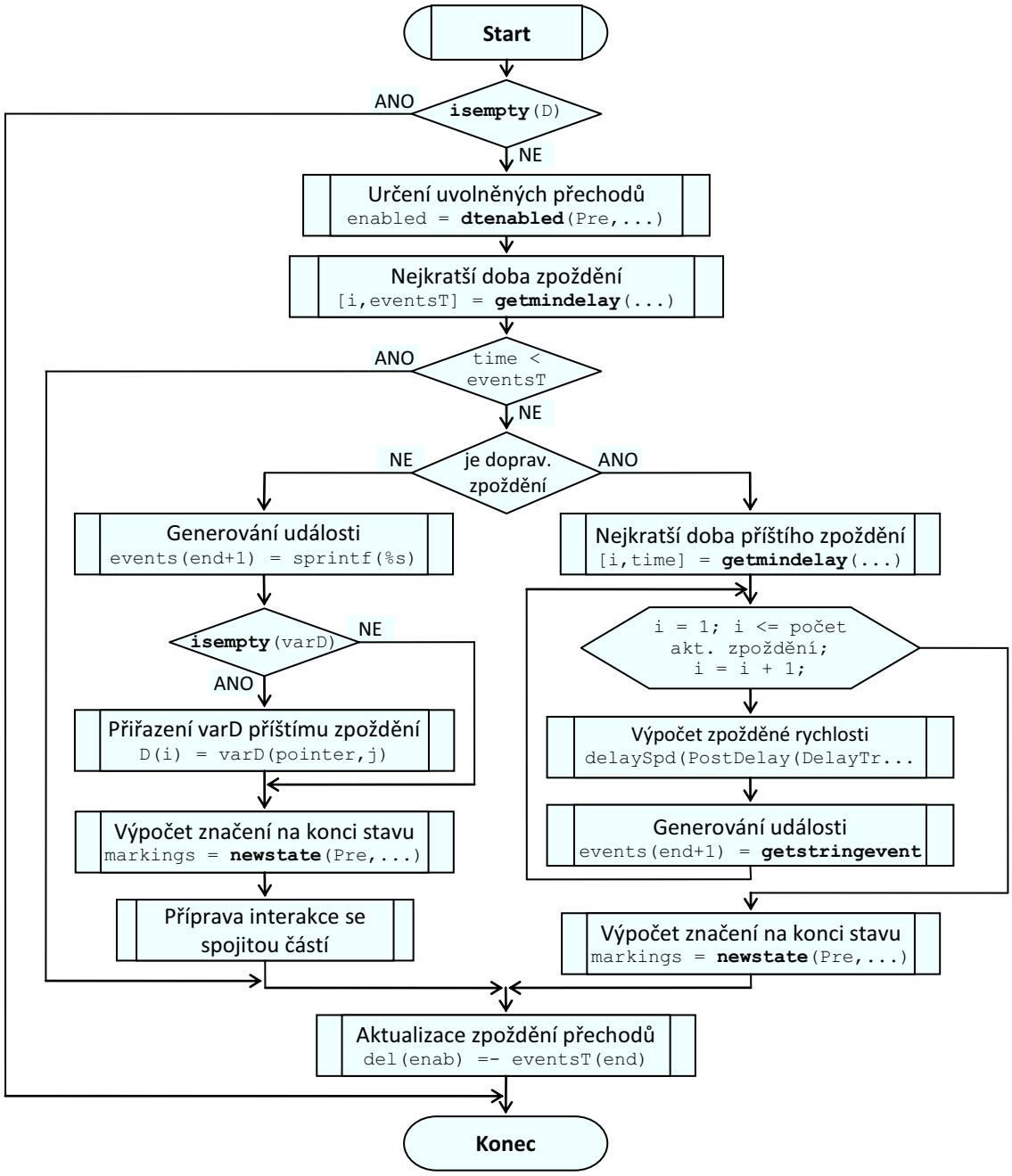
Vývojový diagram 5 ukazuje algoritmus testující diskrétní část simulované Petriho sítě. Stejně jako u části spojité jsou nejprve nalezeny všechny uvolněné přechody. To zajišťuje vnitřní funkce ***dtenabled***, která testuje dvě podmínky. První z nich je $m(P_i) \geq Pre(P_i, T_j)$. T_j je uvolněn, je-li splněna pro $\forall P_i \in {}^{\circ}T_j$. Druhá bere v potaz inhibiční hrany.



Vývojový diagram 4 – Hledání uvolněných přechodů – *ctenabled*

Další krok algoritmu z diagramu 5 hledá nejkratší dobu, po které dojde k přeskoku jednoho z uvolněných přechodů. Bylo-li na základě minimálního zpoždění vyhodnoceno, že stav bude ukončen přeskokem klasického diskrétního přechodu T_j, je vypočítáno značení začátku příštího stavu rovnicí (12) uvedené na straně 11.

Dopravní zpoždění v hybridní Petriho síti je v této práci simulováno podle [David-Alla]. V kapitole 7 nalezneme vysvětlení principu jeho funkce společně s názornými ilustracemi. Prozatím si je přiblížíme pouze slovně. Dopravní zpoždění je realizováno dvojitou konverzí $P_i \to T_j \to P_k$, kde $P_i, P_k \in P^C$ a $T_j \in T^D$. Hrany provádějící tuto konverzi mají váhu 0^+. Tok spojité veličiny je tedy zpožďován po limitně nízkých kvantech. Je zřejmé, že zde nelze provádět výpočty obdobné jako pro časovanou Petriho síť popsanou v podkapitole 2.2. V diagramu 5 jsou oddělené větve zpracování dobře patrné. Způsob zpracování dopravního zpoždění spočívá ve zpožďování rychlosti toku zásobující vstupní místo. Stejně jako u spojité části zde hledáme nejbližší změnu rychlostí – přechod mezi IB-stavy.



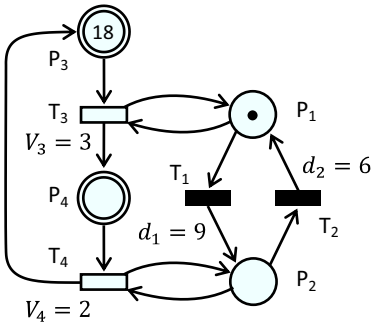
Vývojový diagram 5 – Diskrétní část funkce *playhpn*

5.4. Prezentace vypočtených dat

Abychom mohli výsledky simulací vizuálně vyhodnocovat, porovnávat a studovat chování modelů, potřebujeme nástroje, které bez dlouhého nastavování mnoha parametrů nabídnou vývoj Petriho sítě v přehledné a vzhledově příjemné formě. Jako součást simulačního balíku byly implementovány i funkce, které zobrazení výstupu simulátoru *playhpn* nabízejí. Existují dva způsoby, kterými můžeme zobrazení vývoje Petriho sítě zobrazit:

- vykreslení **průběhu parametru v závislosti na čase**
- zobrazení **vývojového diagramu** simulovaného modelu

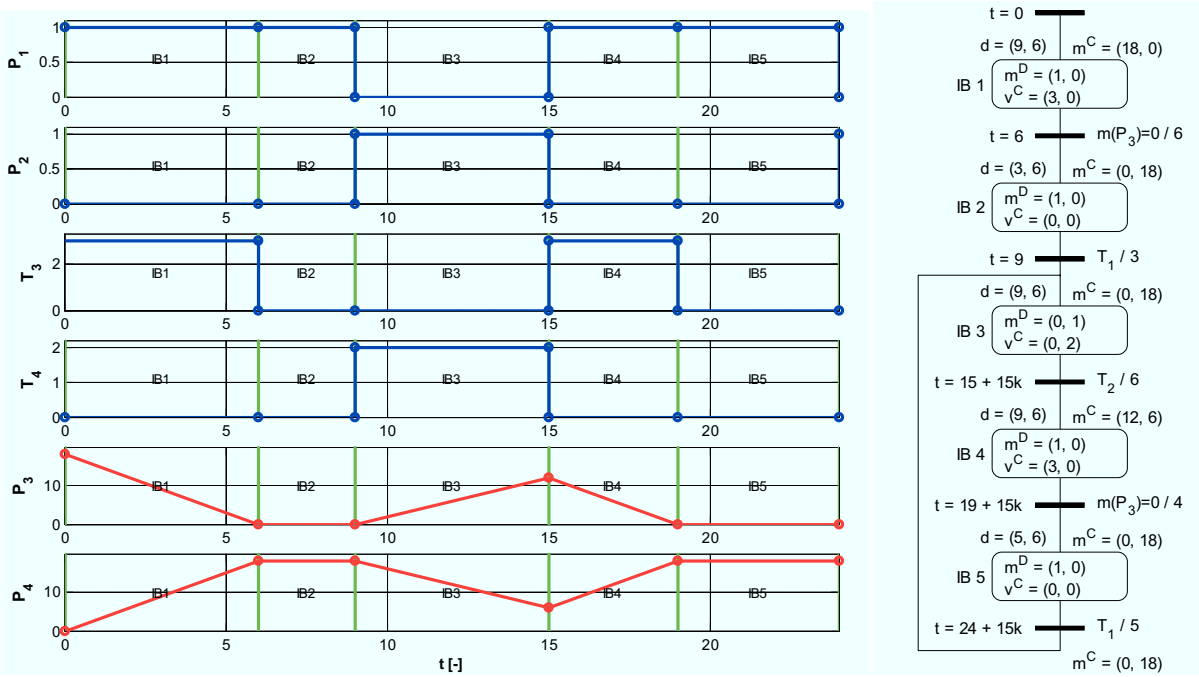
Nakresleme pomocí PN Editoru hybridní síť podle obrázku 15 (jedná se o příklad hybridní Petriho sítě převzatý z [David-Alla] a uložme ji do fomrátu PNML pod názvem „hnet.pnml“. Chceme-li provést simulaci a prohlédnout si výsledek, stačí v MATLABu zadat příkazy podle kódu 2. Výsledek je ukázán na obrázku 16.



Obrázek 15 – Hybridní Petriho síť pro ukázku vykreslení výstupů simulace

```
[Pre,Post,M0,Prio,D,V,TypeP,TypeT,NameP,NameT] = pnml2pn(...
                'hnet.pnml');

[markings,delays,speeds,events,eventsT,loopback] = playhpn(...
                Pre,Post,M0',D,V,Prio,TypeP,TypeT,NameP,NameT);

dispevolution([markings; [speeds, speeds(:,end)]],eventsT,...
                [NameP NameT(3:4)],[1 2 5 6],[3 4]);

dispstates(markings,delays,speeds,events,eventsT,loopback,...
                D,V,TypeP,Prio,[],[],110);
```

Kód 2 – Posloupnost příkazů pro provedení simulace a vykreslení vývoje Petriho sítě



a) **b)**

Obrázek 16 – Zobrazení výsledků simulace hybridní Petriho sítě z obrázku 15

5.4.1. dispevolution

Funkce ***dispevolution*** efektivně kombinuje vestavěné zobrazovací funkce MATLABu *subplot*, *plot* a *stairs*. Primárně je určena pro zobrazování vývoje značení v Petriho sítích, ale lze ji využít i pro vykreslování jiných údajů, například rychlosti přechodů, jak je ukázáno na obrázku 16a. Kromě povinných argumentů nutných pro vykreslení průběhu (např. hodnota, čas, informace o typu, …) lze nastavit několik dalších užitečných přepínačů. Jedním z nich je časový interval, který bude vykreslen. Další je volba, zda chceme zobrazit oddělovače stavů či IB-stavů. Vykreslujeme-li vývoj hybridní Petriho sítě obsahující konverzi mezi spojitými a diskrétními prvky, jsme schopni zadáním matice W^{CD} zajistit správné zobrazení skoků ve značení spojitých přechodů.

$$W^{CD} = Post(P_i, T_j) - Pre(P_i, T_j) \tag{13}$$

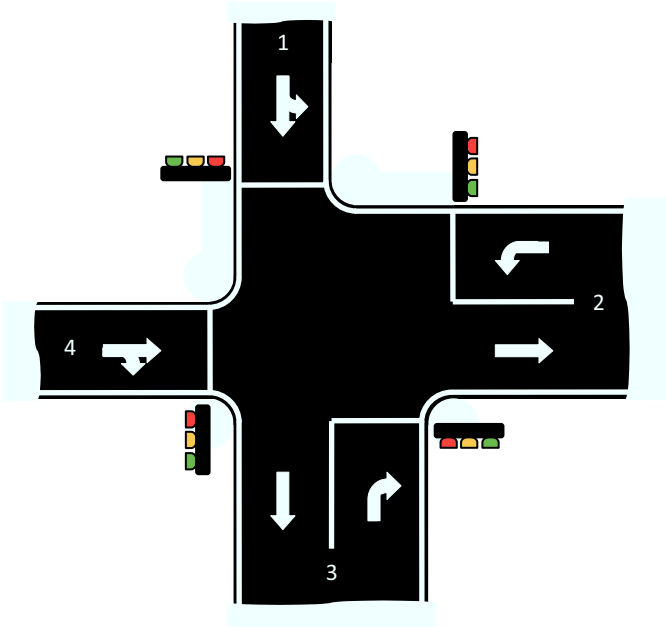
pro všechny $P_i \in P^C$ a $T_j \in T^D$. Bližší popis všech parametrů je uveden v nápovědě, která je součástí funkce.

5.4.2. dispstates

Na obrázku 16b je výsledek simulace interpretován formou vývojového diagramu. Jedná se o kompletní popis vývoje Petriho sítě velice přehlednou formou. Inspirací pro implementaci této funkce byly obdobné diagramy hojně užívané v knize [David-Alla]. Vykreslovací algoritmus je sestaven tak, aby bez jakýchkoliv změn nastavení bylo možné správně zobrazit diagram pro čistě diskrétní (obrázek 3), spojitou (obrázek 5) i kombinovanou hybridní Petriho síť. Kromě povinných argumentů, což jsou data popisující vývoj modelu vygenerovaná pomocí ***playhpn***, můžeme stejně jako u předchozí funkce zadat časový interval, který má být zobrazen. Dalšími parametry jsou volitelné rozměry obrazců představujících stavy.

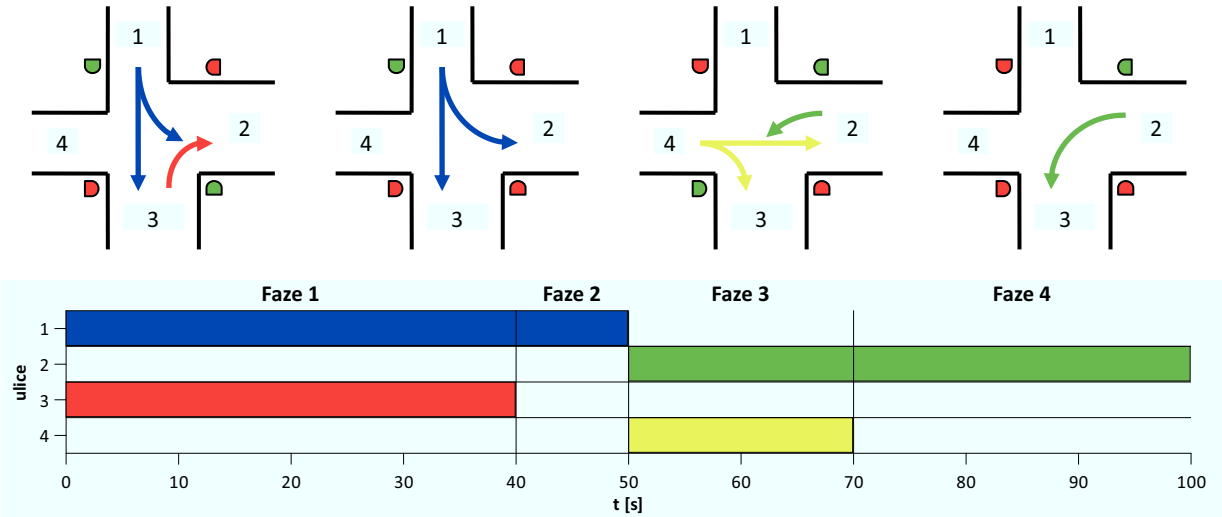
Užitečnou vlastností této funkce je, že všem vykreslovaným grafickým objektům, z nichž se diagram skládá, přiřazuje definované „tagy“. Jedná se o značky, pomocí nichž lze k objektům hromadně přistupovat a pomocí metod ***get*** a ***set*** měnit jejich parametry jako jsou typ čáry, font písma, barva a další. Tím je uživateli umožněno přizpůsobit si vzhled diagramu vlastním potřebám. Seznam těchto tagů je součástí nápovědy funkce.

6. Simulace modelové dopravní křižovatky



Obrázek 17 – Modelová křižovatka

Modely křižovatek musí řešit několik různých dopravních situací, do kterých se toky vozidel při jízdě jednotlivými směry dostávají. Obrázek 17 představuje křižovatku se čtyřmi vstupními a dvěma výstupními ulicemi. Kromě přímého průjezdu křižovatkou zde dochází k větvení toku vozidel z jedné vstupní do několika výstupních ulic, k změně směru jízdy vpravo i vlevo, přičemž musí být respektována přednost vozidel protijedoucích. Křižovatka je řízena světelnou signalizací. Její cyklus lze rozdělit do čtyř fází – obrázek 18. Tato křižovatka včetně jejích parametrů a většiny výpočtů byla převzata z článku [Kutil I], v němž je ukázán také její CCPN model využitý v této práci.



Obrázek 18 – Diagram časového rozdělení fází křižovatky s naznačením dob uvolnění jednotlivých ulic.

6.1. Dané a vypočítané parametry

Pro zjednodušení předpokládáme, že doba trvání fází i délka celého cyklu $\boldsymbol{T}$ zůstává konstantní – v tomto případě 100 s. Křižovatka má ještě další parametry, které je potřeba do modelu zavést. Ty jsou uvedeny v levé části tabulky 1. Prvním z nich je distribuční poměr větvení toku vozidel ze zdrojové ulice $\boldsymbol{z}$ do cílových $\boldsymbol{c}$ značený $\boldsymbol{\alpha_{z\to c}}$. Pro každé $\boldsymbol{z}$ musí platit $\sum_c \alpha_{z\to c} = 1$. Parametr $\boldsymbol{w_{z\to c}}$ značí reálnou rychlost průjezdu křižovatkou ve směru $z \to c$. Je zřejmé, že zatáčející vozidlo se nemůže pohybovat stejnou rychlostí jako vozidlo jedoucí přímo. Posledním ze základních parametrů je $\boldsymbol{t_{z\to c}}$. Jedná se o dobu trvání, po kterou mohou vozidla proudit při zachované platnosti předchozích údajů.

Tabulka 1 – Parametry křižovatky

$\boldsymbol{z}$ (zdroj)	$\boldsymbol{c}$ (cíl)	$\boldsymbol{\alpha_{z\to c}}$	$\boldsymbol{w_{z\to c}}$ [km·h^{-1}]	$\boldsymbol{t_{z\to c}}$ [s]	$\boldsymbol{d_{z\to c}}$ [s·uv^{-1}]	$\boldsymbol{U_{z\to c}}$ [uv·s^{-1}]	$\boldsymbol{U_{z\to}}$ [uv·s^{-1}]	$\boldsymbol{V_{z\to c}}$ [uv·s^{-1}]	$\boldsymbol{V_{z\to}}$ [uv·s^{-1}]
1	2	0,2	30	50	0,60	1,67	2,45	0,83	1,23
1	3	0,8	50	50	0,36	2,78		1,39	
1	2	0,2	30	10	-	-	-	0,17	0,25
1	3	0,8	50	10	-	-		0,28	
2	3	1,0	30	50	0,60	1,67	1,67	0,83	0,83
2	3	1,0	30	30	-	-	-	0,50	0,50
3	2	1,0	20	40	0,90	1,11	1,11	0,44	0,44
4	2	0,4	50	20	0,36	2,78	1,46	0,56	0,29
4	3	0,6	20	20	0,90	1,11		0,22	

V druhé části tabulky 1 se nacházejí parametry vypočítané podle vztahů (14) – (18), které jsou nezbytné pro vytvoření modelů pomocí Petriho sítí. Sloupec $\boldsymbol{d_{z\to c}}$ představuje dobu průjezdu jednoho vozidla (uv – unit vehicle) křižovatkou. Toto zpoždění bude v následujícím experimentu využito pro modelování křižovatky časovanou Petriho sítí.

$$d_{z\to c} = \frac{3{,}6 \cdot l_{uv}}{w_{z\to c}} \tag{14}$$

Inverzním parametrem ke zpoždění $d_{z\to c}$ je rychlost $\boldsymbol{U_{z\to c}}$. Jedná se o maximální počet vozidel, které projedou křižovatkou za dobu jedné sekundy. Tato rychlost je použita při modelování hybridní Petriho sítí.

$$U_{z\to c} = d_{z\to c}^{-1} = \frac{w_{z\to c}}{3{,}6 \cdot l_{uv}} \tag{15}$$

Pro spojitou Petriho síť je potřeba vypočítat maximální průměrnou rychlost za dobu periody $\boldsymbol{T}$ cyklu světelné signalizace.

$$V_{z\to c} = \frac{t_{z\to c}}{T} \cdot U_{z\to c} = \frac{w_{z\to c} \cdot t_{z\to c}}{3{,}6 \cdot l_{uv} \cdot T} \tag{16}$$

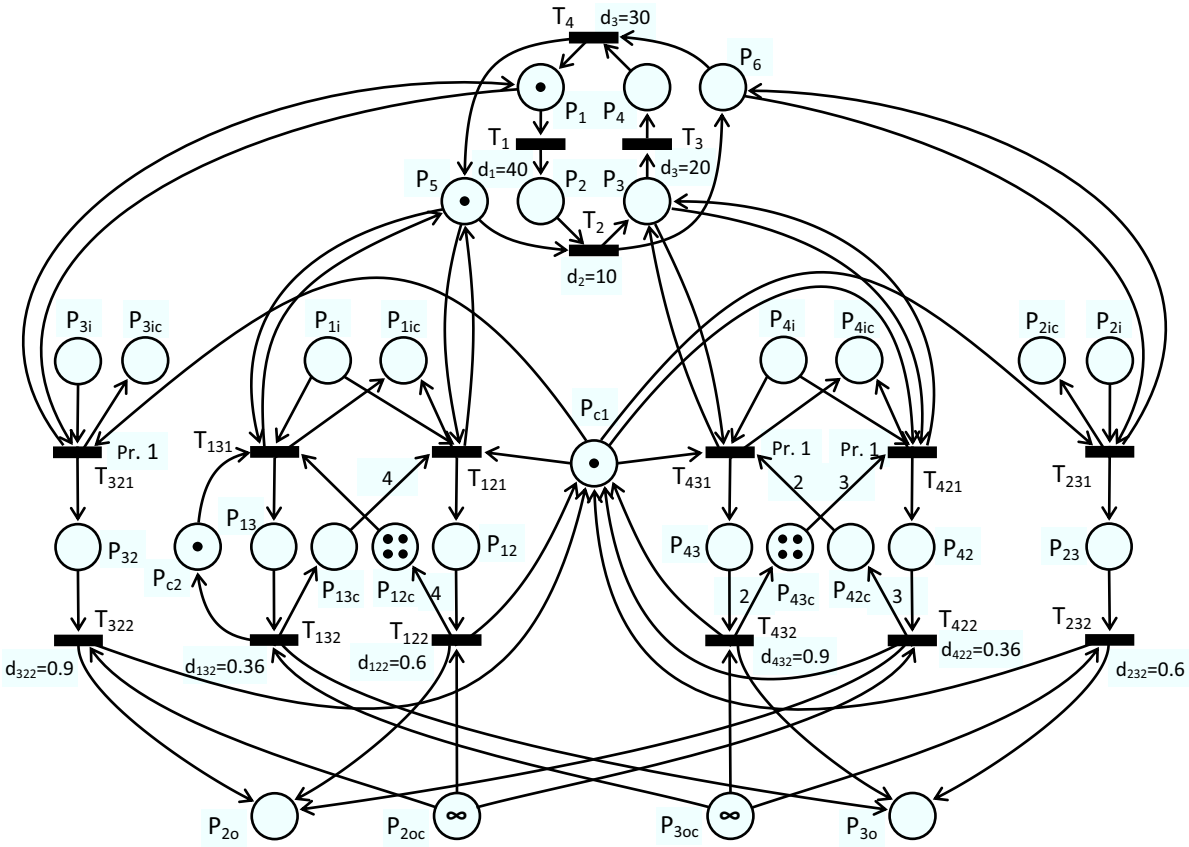
Proměnná $\boldsymbol{l_{uv}}$ figurující v předchozích vztazích představuje délku jednoho vozidla včetně rozestupů. V této práci je stejně jako v [Kutil I] i [Stibor] použita hodnota $l_{uv} = 5\ \mathrm{m}$. Při výpočtu zdrojové rychlostí $\boldsymbol{U_{z\rightarrow}}$ a $\boldsymbol{V_{z\rightarrow}}$ větvícího se toku vozidel je zohledněn dělicí poměr $\alpha_{z\rightarrow c}$. Výsledky uvedené v tabulce 1 jsou získány výpočty následujících vztahů:

$$U_{z\rightarrow} = \frac{\sum_c \alpha_{z\rightarrow c}}{\sum_c \frac{\alpha_{z\rightarrow c}}{U_{z\rightarrow c}}}, \tag{17}$$

$$V_{z\rightarrow} = \frac{t_{z\rightarrow c}}{T} \cdot U_{z\rightarrow} = \frac{\sum_c \alpha_{z\rightarrow c}}{\sum_c \frac{\alpha_{z\rightarrow c}}{V_{z\rightarrow c}}}. \tag{18}$$

6.2. TPN model

Obrázek 19 představuje model křižovatky z obrázku 17 tvořený časovanou Petriho sítí. Místa P_1, P_2, P_3 a P_4 představují čtyři fáze řízení provozu a zpoždění přechodů T_1, T_2, T_3 a T_4 doby jejich trvání. P_5 a P_6 spojují dvě fáze pro ulice 1 a 2.



Obrázek 19 – Model křižovatky tvořený časovanou Petriho sítí

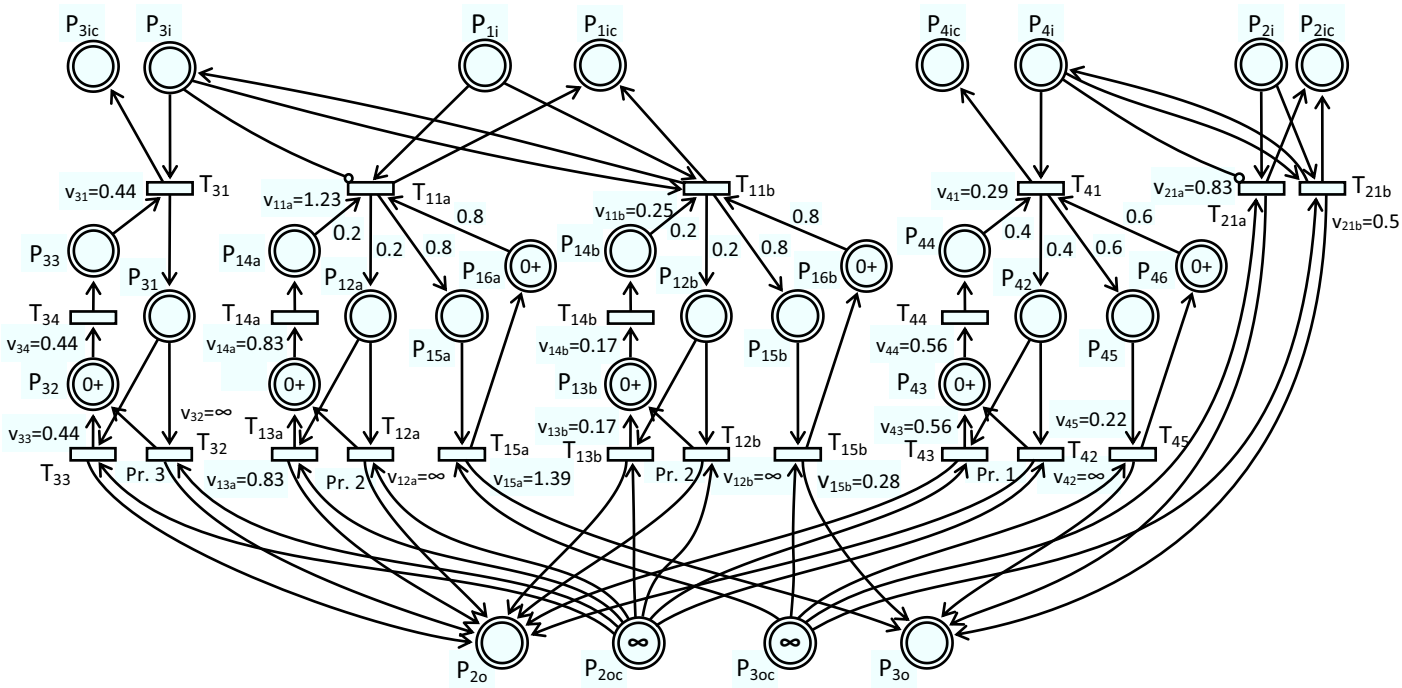
Místo P_{1i} reprezentuje ulici 1, v níž je formována fronta vozidel čekajících na průjezd křižovatkou. Na toto místo je při simulaci napojen generátor vozidel. P_{1ic} představuje kapacitu ulice 1. V případě simulace jediné křižovatky nemá toto místo skutečný význam, ale k jeho uplatnění dojde v případě

modelování celé oblasti, kdy mezi křižovatkami může docházet k dopravním zácpám – naplněním kapacity ulice. Přechody T_{131} a T_{132} představují průjezd křižovatkou do ulice 3, T_{121} a T_{122} průjezd do ulice 2. P_{13c} a P_{12c} společně s váhami hran zajišťují dodržení distribučního poměru $0{,}2/0{,}8$. Toto dělení není přesným obrazem reality, kdy rozdělení vozidel do ulic je náhodné, pro simulaci je však dostačující. P_{2o} je výstupem ulice 2. P_{2oc} je opět kapacita ulice 2. Protože v tomto případě není kapacita ulice důležitá, je hodnotou značení P_{2oc} nekonečno.

Ostatní části modelu – ostatní směry průjezdu křižovatkou se skládají z míst a přechodů obdobných významů jako ty, jež jsou popsány výše.

6.3. CCPN model

Tak jako v obou předchozích, nalezneme i v tomto modelu místa pro vstupní frontu, výstupní místa a místa kapacit vstupních i výstupních ulic. Protože rychlosti přechodů jsou vyjádřeny jako průměr za celou dobu cyklu řízení křižovatky tak, jak je uvedeno v tabulce 1, není zde potřebná přesná definice jednotlivých fází cyklu. Přednost při odbočování vlevo je zde namísto priorit simulována povolením či zakázáním přechodu inhibiční hranou a smyčkou.



Obrázek 20 – Model křižovatky tvořený spojitou Petriho sítí s konstantní rychlostí

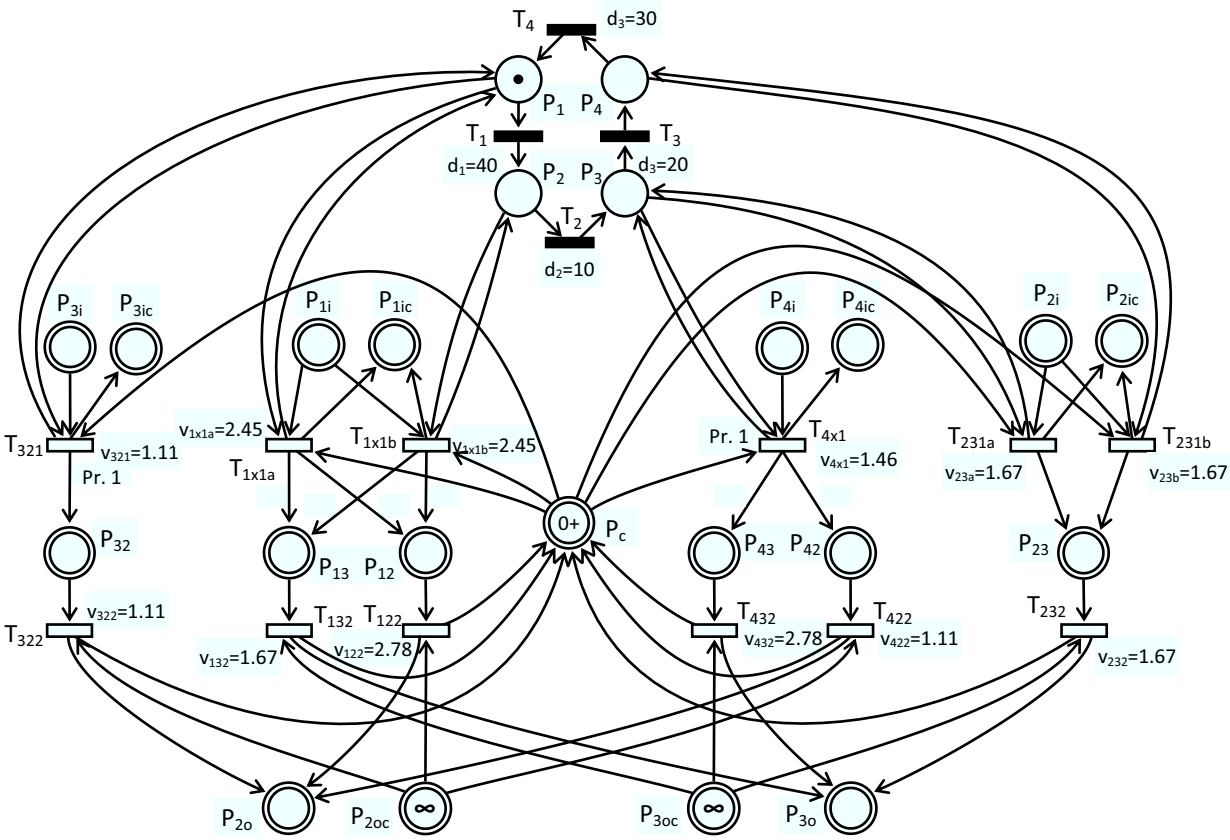
Není-li žádné vozidlo, které by do křižovatky vjíždělo z ulice 3, může tok vozidel čekajících ve frontě ulice 1 vjíždět do křižovatky přechodem T_{11a}, který má vyšší rychlost. Projíždí-li však nějaká vozidla ulicí 3 – místem P_{3i}, je aktivní přechod T_{11b} s rychlostí nízkou – vozidla z ulice 1 musejí dát přednost vozidlům z ulice 3. Obdobný je i vztah ulic 4 a 2. Vozidla přijíždějící do křižovatky z ulice 1 jsou po prů-

chodu přes vstupní přechod rozdělena do výstupních proudů ulic 2 a 3 v poměru daného parametry křižovatky obsaženými v tabulce 1.

V článku [Kutil I] je model z obrázku 20 ukázán s dalšími několika prvky, které zde chybí. Ty však při simulaci jediné křižovatky, kdy uvažujeme nekonečnou kapacitu výstupních ulic, ztrácí význam, a proto jsou zanedbány.

6.4. HPN model

Stejně jako diskrétní, obsahuje i hybridní model místa P_1, P_2, P_3 a P_4 a přechody T_1, T_2, T_3 a T_4 a tedy cyklické střídání jednotlivých fází křižovatky. Diskrétní místa a přechody jsou zde nahrazeny spojitými. Rovněž obsahuje místa vstupních a výstupních ulic stejně jako jejich kapacity. K zajištění distribučního poměru ve spojité síti stačí odpovídající nastavení vah hran a není nutné použití dodatečných míst, jako je tomu v případě časované Petriho sítě. Dodržení pravidla přednosti protijedoucím vozidlům při odbočování vlevo zde simulují priority přiřazené přechodům (není-li priorita u přechodu uvedena, je její hodnota 0).



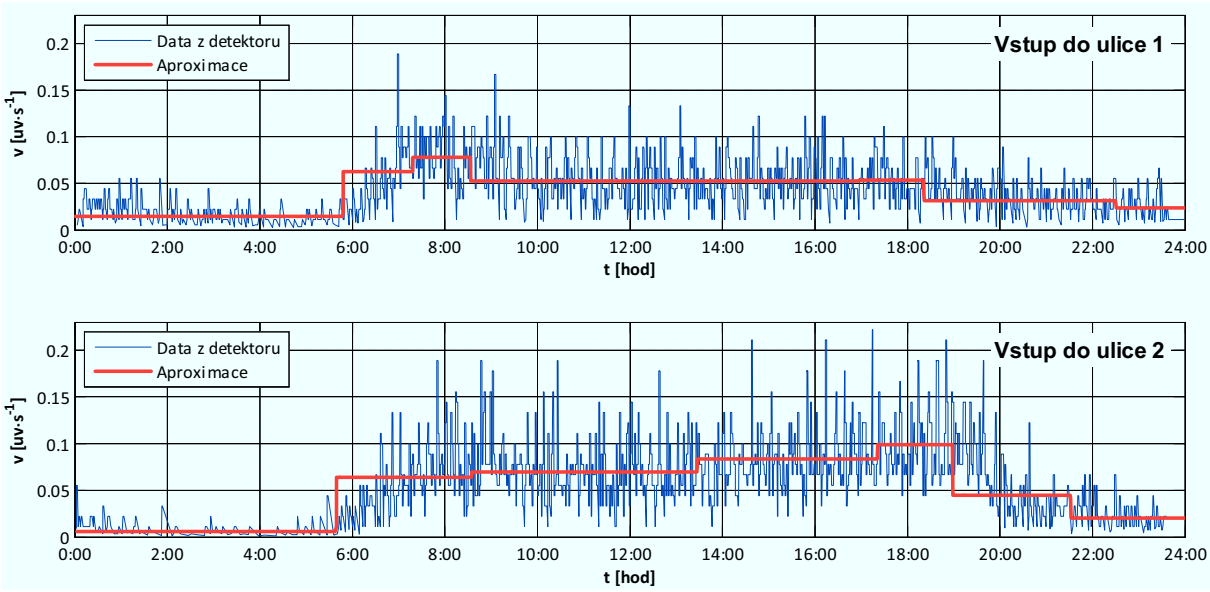
Obrázek 21 – Model křižovatky tvořený hybridní Petriho sítí

6.5. Simulace založená na reálných datech

Pro simulaci křižovatky, jejíž tři různé modely byly ukázány v předchozích částech této kapitoly, jsou použita reálná vstupní data. Ta jsou představovaná časy průjezdů jednotlivých vozidel naměřenými v průběhu jednoho celého pracovního dne v pražských ulicích. Úkolem této kapitoly je srovnání tří modelů tvořených různými typy Petriho sítí. Pomocí zmíněných dat lze srovnání velmi dobře provést.

Obrázek 22 ukazuje dva z naměřených průběhů signálů získaných induktivními senzory. Data z detektorů jsou průměrována přes 90 s dlouhou periodu. Pro zjednodušení simulace jsou navíc naměřená data aproximována vyhlazujícím filtrem. Tento vyhlazený signál je představován červeným schodovitým průběhem a slouží jako vstup pro všechny tři modely. Ukázané průběhy jsou použity jako vstupy ulic 1 a 2. Pro zbylé dvě ulice jsou použity průběhy obdobné.

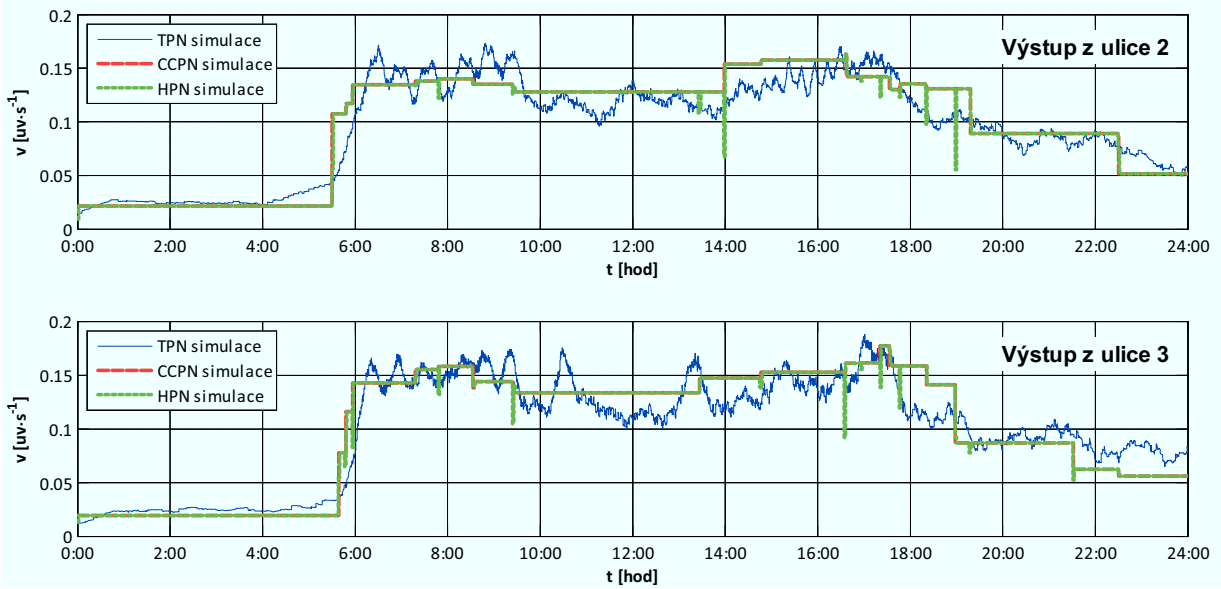
Vstup je modelům přiřazen dvěma způsoby. Vstupní signál pro časovanou Petriho síť byl transformován do formy měnícího se zpoždění přechodů generátorů tokenů = zpoždění vozidel přijíždějících za sebou. Signály pro hybridní a spojitou Petriho síť jsou měnící se rychlosti přechodů generátorů toku = rychlosti přijíždějících vozidel, které se s časem mění podle průběhů na obrázku 22. Samotná simulace probíhá tak, že celkový čas je rozdělen do několika intervalů, v nichž jsou parametry Petriho sítí konstantní, přičemž počáteční stav modelu je dán koncovým stavem simulace v předchozím intervalu.



Obrázek 22 – Výstupy detektorů projíždějících vozidel

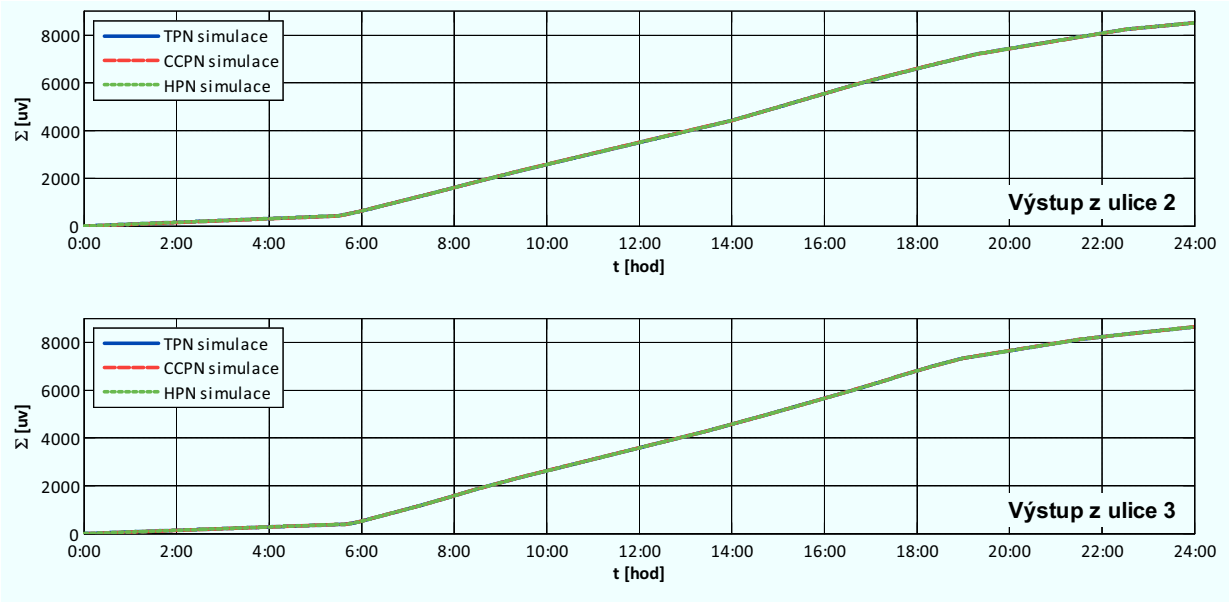
Obrázek 23 ukazuje rychlost vozidel opouštějících křižovatku pro všechny tři použité modely v průběhu simulované doby jednoho dne. Je patrné, že si výstupy modelů při stejném vstupním signálu přibližně odpovídají. Zobrazený výstupní signál pro PN model je filtrovaný, signál HPN modelu musí být

pro zobrazení průměrovaný na dobu periody řídicího cyklu světelné signalizace. V jiném případě by oba zobrazené průběhy byly kvůli mnoha fázím řídicího cyklu zcela nečitelné.

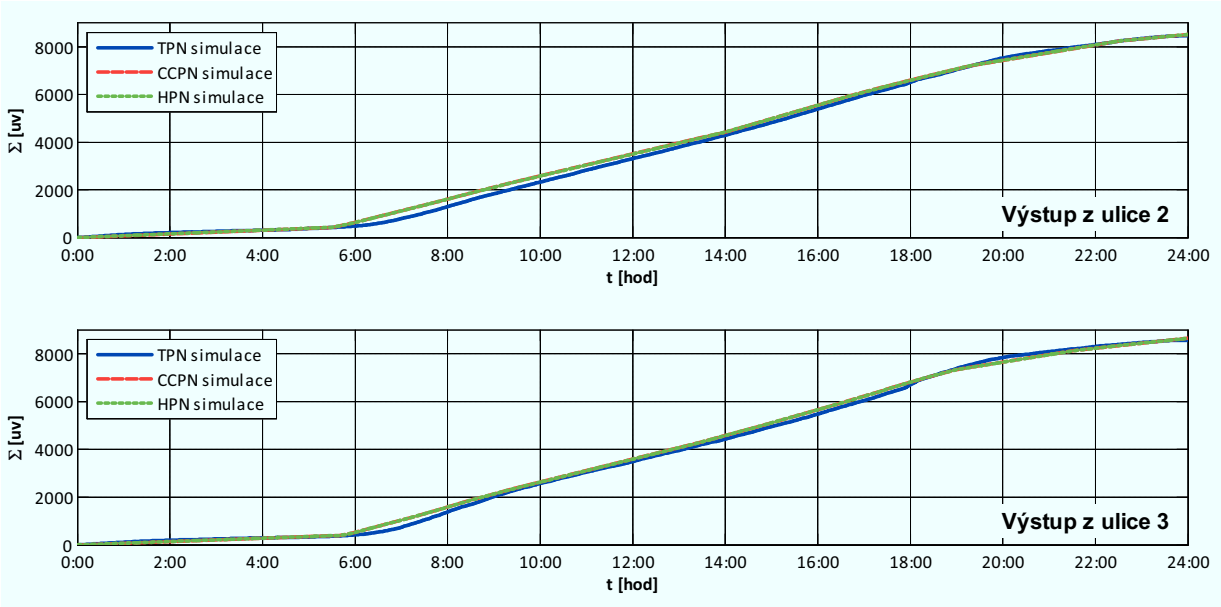


Obrázek 23 – Výsledky simulace – rychlost, jíž vozidla opouštějí křižovatku do výstupních ulic

Vzájemná korespondence všech tří modelů je ještě více patrná z obrázku 24. Na něm je jasně vidět, že celkový počet vozidel, která projela během dne křižovatkou je v rámci rozlišení nabízeném zobrazenými grafy shodný pro všechny tři modely. Ve skutečnosti se však křivky vzájemně pouze blíží, jak je zobrazeno na obrázku 26 a okomentováno v kapitole 6.6.1. Na základě tohoto experimentu lze tedy prohlásit, že navržené modely řízené dopravní křižovatky jsou z hlediska výsledků simulace a také své funkce ekvivalentní.



Obrázek 24 – Výsledky simulace – celkový počet vozidel upouštějících křižovatku



Obrázek 25 – Výsledky simulace – neaproximovaný vstupní signál pro simulaci TPN modelem

Obrázek 25 ukazuje srovnání CCPN a HPN modelu, jejichž vstupy jsou aproximacemi naměřených signálů, a TPN modelu tentokrát se signály neaproximovanými. To znamená, že simulaci křižovatky provádíme s maximální přesností. Na zobrazených průbězích je patrná odchylka způsobená aproximací. Při tomto experimentu je jak počet stavů řešení TPN modelu, tak čas jeho výpočtu téměř stejný jako v předchozím případě.

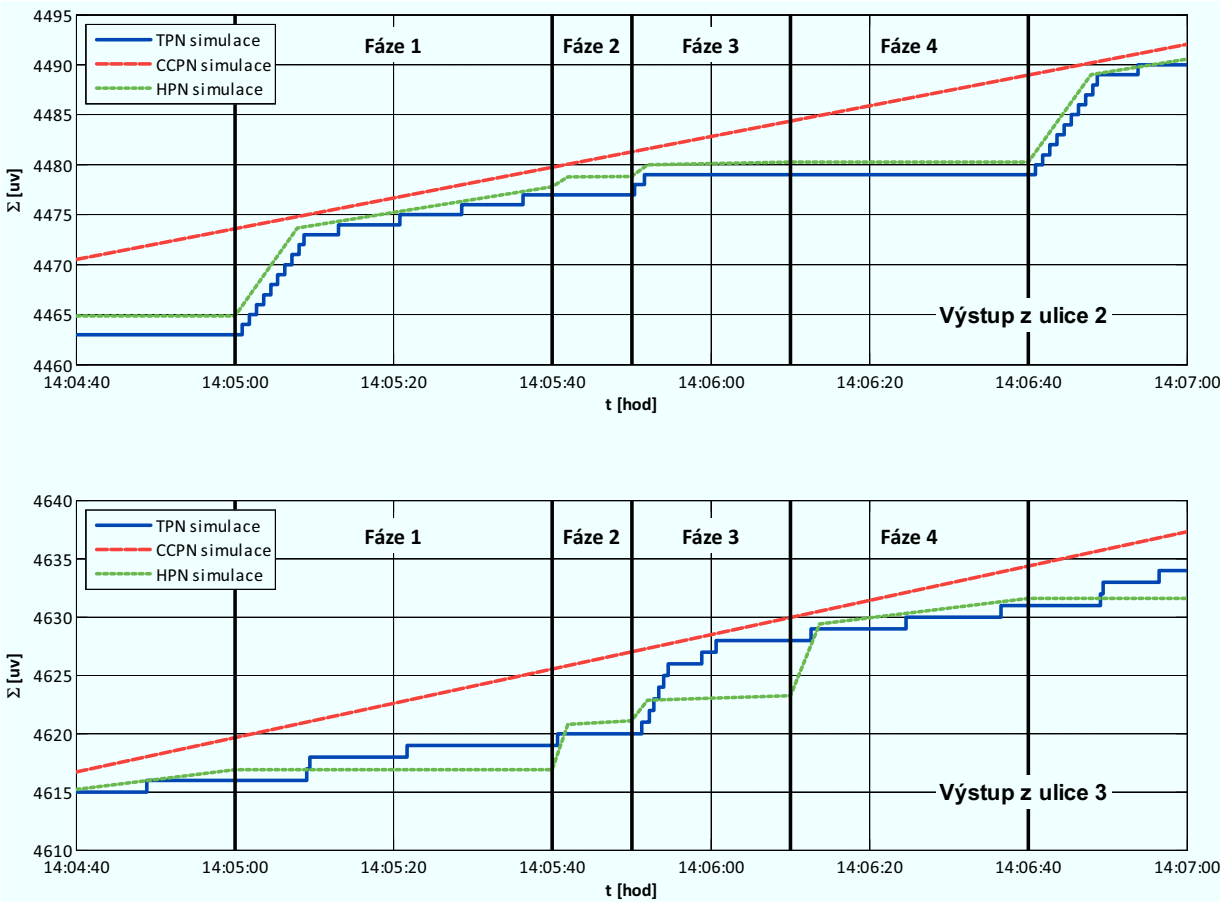
6.6. Zhodnocení a porovnání modelů

Simulované modely lze porovnávat ze dvou hledisek. Prvním z nich je kvalita simulace a přesnost dosažených výsledků v závislosti na čase. Druhým hlediskem je výpočetní náročnost vývoje Petriho sítě modelující křižovatku. Při praktickém využití tohoto způsobu modelování dopravních sítí je tedy nezbytné pro konkrétní aplikaci zvolit důležitost přesnosti a výpočetní doby. V následujících podkapitolách je uvedeno srovnání výsledků simulace tří modelů.

6.6.1. Kvalita simulace

Obrázek 26 představuje detailní zobrazení výsledku simulace z obrázku 24 v průběhu jednoho cyklu světelné signalizace během poměrně hustého provozu v odpoledních hodinách. Je zde vidět, že křivky představující výsledky simulací se nepřekrývají, ale mírně se liší.

Vzhledem k tomu, že u CCPN modelu je poměrná délka každé z fází vůči celému cyklu započítána přímo do parametrů spojitého modelu, nelze vývoj dopravní situace sledovat blíže. Výsledky CCPN jsou vhodné pro sledování modelovaného systému v delším časovém měřítku.



Obrázek 26 – Výsledky simulace – celkový počet vozidel na výstupu křižovatky – ukázka jednoho řídícího cyklu

Na schodovitém průběhu přesného řešení modelu tvořeného časovanou Petriho sítí lze snadno sledovat vývoj dopravní situace:

Fáze 1: Do ulice 2 vjíždějí vozidla nahromaděná u vstupu do křižovatky z ulice 3 během posledních fází předchozího cyklu. Po vyprázdnění fronty se frekvence nově přijíždějících vozidel sníží a nemusí-li dát přednost, mohou do ulice 2 vjíždět i vozy z ulice 1. Pokud jsou v ulici 1 na řadě vozy směřující do ulice 3, mohou volně projet.

Fáze 2: Do křižovatky z ulice 1 vjíždí jedno vozidlo. Směřuje do ulice 3, kam má průjezd volný.

Fáze 3: Vozidla čekající ve frontě v ulici 4 mají přednost před vozidly v ulici 2 a projíždějí křižovatkou do ulic 2 a 3 v příslušném distribučním poměru. Je-li v protisměru volno, mohou vozidla z ulice 2 projíždět křižovatkou do ulice 3.

Fáze 4: Jedině vozidla přijíždějící z ulice 2 mohou projíždět křižovatkou do ulice 3.

Na výstupu do ulice 2 na obrázku 26 je patrné, že křivka simulace HPN téměř kopíruje přesný průběh PN. Projíždějící jednotky vozidel jsou nahrazeny spojitou veličinou. To však způsobuje nepřesnost simulace. Přestože by totiž vozidla z ulice 1 mohla během první fáze křižovatkou projíždět tak, jak je popsáno v předchozím odstavci, kvůli pomalu proudícímu toku vozidel z ulice 3 jim je dán prostor až ve fázi 2. Na srovnání průběhů pro výstup do ulice 3 je tato simulační chyba, daná principem spojité Pet-

riho sítě, patrná ještě více. I když se průběhy jednotlivých fází mírně liší, po ukončení cyklu je výsledek HPN i TPN obdobný.

6.6.2. Výpočetní náročnost

Každé vozidlo je v diskrétní časované Petriho síti simulováno samostatně. To má za následek značnou přesnost, ale také veliký počet stavů vývoje značení Petriho sítě. Pro každé vozidlo jsou zde 3 stavy dané třemi událostmi: jeho zařazení do fronty čekající na vjezd do křižovatky, vstup vozidla do křižovatky a jeho výstup z ní. Další události ohraničující stavy představuje řízení dopravy světelnou signalizací – čtyři stavy na jeden cyklus délky 100 s. Celkový počet stavů lze tedy vypočíst jako

$$p_{pn} = p_f \cdot \frac{t_c}{T} + 3 \cdot \sum_z p_z = 4 \cdot \frac{24 \cdot 60 \cdot 60}{100} + 3 \cdot (6817 + 3454 + 2525 + 4365) = 54939\,, \quad (19)$$

kde p_f [-] je počet fází křížovatky, t_c [s] je celková simulovaná doba, T [s] je délka řídicí periody a p_z [uv] počet vozidel, který během dne do křižovatky vjíždí ze zdrojové ulice z.

Hybridní model křižovatky respektuje cyklus řízení včetně jeho fází. Průjezdy jednotlivých vozidel jsou však zhuštěny do spojité podoby. Místo stavů představujících průjezd jednotlivých vozidel máme pro každou fázi 2 stavy definované rychlostí v jednotkách $uv \cdot s^{-1}$: jeden stav pro vyprázdnění fronty čekajících vozidel, druhý pro průjezdy vozidel bez fronty. V případě hybridní Petriho sítě platí vztah

$$p_{hpn} = 2 \cdot p_f \cdot \frac{t_c}{T} + p_i - p_s = 8 \cdot \frac{24 \cdot 60 \cdot 60}{100} + 24 - 6 = 6930\,, \quad (20)$$

kde p_i [-] je celkový počet intervalů, v nichž je rychlost počtu přijíždějících vozidel neměnná (blíže popsáno v kapitole 6.5) a p_s [-] je počet intervalů, jejichž počátek nastane ve stejný čas jako změna fáze křižovatky – událost změny rychlosti v takovém případě nevytvoří oddělený stav. Tento počet lze snadno zjistit pomocí

$$p_s = \sum \mathrm{ismember}\big(\mathrm{mod}(t_i, T), t_f\big)\,, \quad (21)$$

kde funkce *mod* (modulo) je výpočet zbytku po dělení prvního argumentu argumentem druhým a *ismember* vrací hodnotu 1, pokud je prvek prvního argumentu obsažen ve vektoru argumentu druhého, t_i je vektor absolutních časů, kdy dojde ke změně vstupní rychlosti a t_f je vektor relativních časů počátků fází podle obrázku 18 v úvodu kapitoly.

Počet stavů vývoje modelu křižovatky simulované spojitou Petriho sítí s konstantní rychlostí je roven pouze počtu intervalů, do kterých je celá doba simulace rozdělena.

$$p_{ccpn} = p_i = 24\,. \quad (22)$$

Tabulka 2 ukazuje srovnání počtu stavů výsledků simulace všech tří typů modelů. Jak je z prvního sloupce tabulky patrné, je výsledek simulace pomocí spojité Petriho sítě mnohonásobně jednodušší a tomu také odpovídá doba zpracování modelu. Tato výpočetní efektivita je však získána na úkor přesnosti simulace. Hybridní model nabízí poměrně přesný pohled na situaci na křižovatce, avšak za cenu značné výpočetní doby dané nutností řešit problém lineárního programování minimálně dvakrát při každé změně fáze řídicího cyklu. Právě tato činnost zabírá simulátoru přibližně 85% celkového času. V závislosti na vstupních datech získáme řešením diskrétního modelu přesný obraz situace na křižovatce v průběhu celého dne. To však znamená velký počet stavů v porovnání s oběma předchozími modely. Vzhledem k tomu, že řešení časovaných Petriho sítí nevyužívá lineární programování, je čas řešení modelu kratší než u modelu hybridního. Přesto se jedná téměř o 100 násobek času simulace spojitého modelu.

Tabulka 2 – Srovnání výsledků simulace[2]

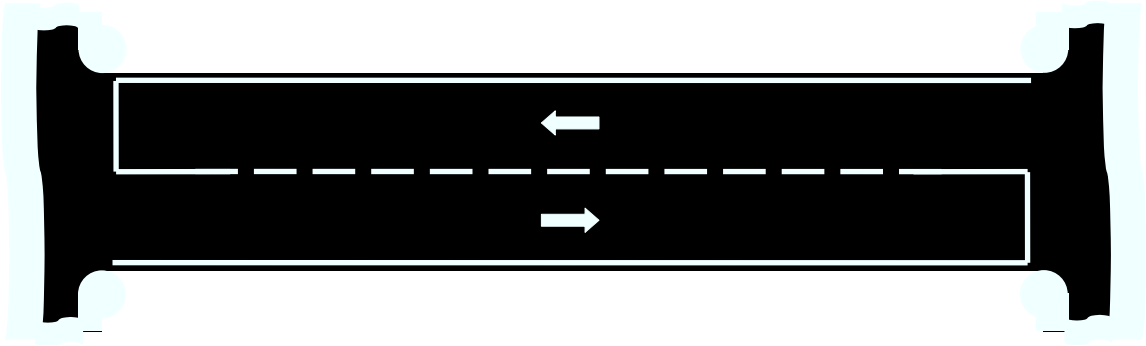
Model	Počet stavů	Doba simulace3
PN	54939	99,97 s
HPN	6930	194,23 s
CCPN	24	1,16 s

Za zmínku stojí, že pokud bychom chtěli použít detailnější aproximaci vstupních dat, museli bychom použít větší počet intervalů konstantní rychlosti. Složitost výsledku (počet stavů) CCPN modelu by tak rovněž narostla a s ní i výpočetní doba. Oproti tomu složitost řešení časovaného PN modelu je na přesnosti závislá jen málo a jak počet stavů výsledného řešení, tak výpočetní doba se prakticky nezmění.

[2] Simulace byla provedena na osobním počítači vybaveném Intel Core Duo CPU T2350 @1.86 GHz, 1 GB RAM.

[3] Do doby simulace není započtena doba načítání Petriho sítí z PNML souborů ani proměnných parametrů vstupního zpoždění a rychlostí toků vozidel z textových souborů.

7. Návrh modelu ulice



Obrázek 27 – Ulice – dopravní zpoždění při modelování

Ačkoliv ulice (jeden její dopravní pruh) tak, jak je pojatá na obrázku 27, představuje pouze přímý průjezd vozidel bez řízení dopravy, není její modelování bez obtíží. Představuje totiž dopravní zpoždění toku vozidel. To při vytváření modelu systému přináší nelinearitu, která simulaci komplikuje.

Každá ulice má určitou kapacitu vozidel, jež se do ní vejdou a dobu zpoždění, po kterou trvá průjezd vozidla. Tyto dva parametry jsou dány vztahy

$$C_u = n_u \cdot \frac{l_u}{l_{uv}}, \tag{23}$$

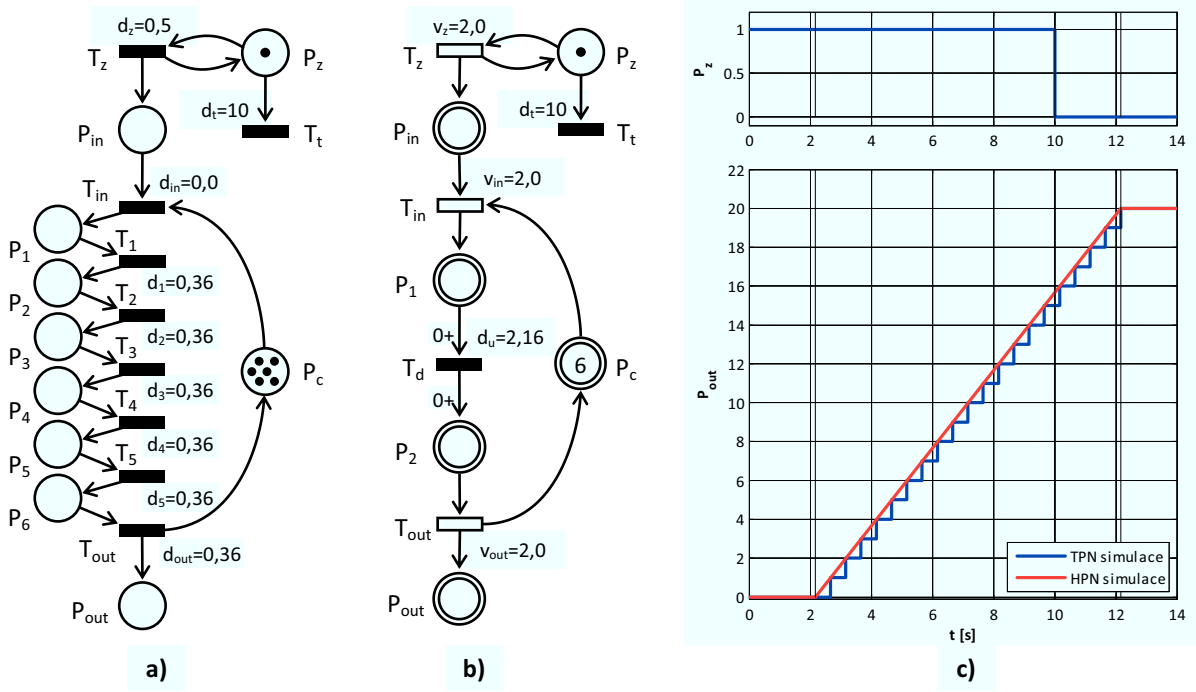
$$d_u = \frac{3{,}6 \cdot l_u}{w_u}, \tag{24}$$

kde C_u je kapacita ulice u, n_u počet jízdních pruhů (lze rovněž simulovat každý jízdní pruh zvlášť, pak $n_u = 1$), l_u [m] představuje délku ulice, d_u je trvání průjezdu ulicí při průměrné rychlosti w_u.

Veličiny, se kterými operuje časovaná Petriho síť (TPN), jsou množství v podobě značení míst a čas reprezentovaný zpožděním přechodů. Přestože zde čas jako takový existuje, vytvořit dopravní zpoždění toku vozidel není jednoduché. Pokud bychom chtěli použít jediný přechod s nastaveným zpožděním odpovídajícím době, po kterou vozidlo reprezentované tokenem ulicí projíždí, museli bychom zajistit, aby se čas odpočítával pro každý nově příchozí token zvlášť. Takové chování by však neodpovídalo časované Petriho síti tak, jak je definována v [David-Alla] či v [Hanzálek II] a tedy ani tomu, jak je její řešení implementováno v simulátoru, jenž je výsledkem této práce. Řešením je rozdělení ulice na několik úseků a simulování průjezdu vozidla po částech. Aby byly zachovány rozestupy mezi vozidly stejné jako na vstupu do ulice, nastavíme tyto úseky na délku jednoho vozidla a doba průjezdu každou z těchto částí bude dílčím časem průjezdu vozidla celou ulicí. Příklad modelu ulice tvořeného časovanou Petriho sítí je uveden na obrázku 28a.

Aparát spojitých Petriho sítí s konstantní maximální rychlostí (CCPN) pracuje se značením míst a rychlostí průtoku spojité veličiny přechody. Čas sám o sobě zde není obsažen, a proto zpoždění průjez-

du vozidla ulicí nelze namodelovat. Musíme použít hybridní Petriho síť kombinující spojitý průtok vozidel a časové zpoždění diskrétního přechodu. Příklad hybridního modelu ukazuje obrázek 28b.



Obrázek 28 – Modelování ulice: a) TPN model, b) HPN model, c) Výsledek simulace

Modely na obrázku 28 simulují ulici délky $l_u = 30\ \mathrm{m}$. Do ní jsou po dobu $t = 10\ \mathrm{s}$ vpouštěna s frekvencí $v_z = 2\ \mathrm{uv} \cdot \mathrm{s}^{-1}$ vozidla pohybující se rychlostí $w_u = 50\ \mathrm{km} \cdot \mathrm{h}^{-1} = 13{,}9\ \mathrm{m} \cdot \mathrm{s}^{-1}$. Doba průjezdu ulicí je tedy $d_u = 2{,}16\ \mathrm{s}$. Uvažujeme-li délku vozidla $l_{uv} = 5\ \mathrm{m}$, je kapacita ulice 6 vozidel. Průběh na obrázku 28c zobrazuje vývoj značení míst P_{out} v čase pro oba modely. Je na něm vidět, že vstupní tok vozidel je na výstupu skutečně zpožděn o $2{,}16\ \mathrm{s}$. Podíváme-li se však na výsledky simulace, zjistíme, že vývoj TPN modelu je uskutečněn ve 162 stavech oddělených přeskoky přechodů. Oproti tomu vývoj HPN je rozdělen pouze do 4 IB-stavů. Při simulování delší ulice by počet stavů vývoje TPN značně vzrostl, kdežto počet IB-stavů pro HPN model se s délkou nezmění.

Výsledkem tohoto srovnání je tedy poznatek, že HPN model ulice je z hlediska simulace výrazně výhodnější. V podkapitole 7.1 je na něj nabídnut detailnější pohled.

7.1. Modelování ulice pomocí HPN

Uvažujme ulici délky $l_u = 100\ \mathrm{m}$, ostatní parametry jsou podobné modelu na obrázku 28: po dobu $\mathrm{t} = 30\ \mathrm{s}$ jsou do ní opět s frekvencí $v_z = 2\ \mathrm{s}^{-1}$ vpouštěna vozidla pohybující se rychlostí $\mathrm{w_u} = 50\ \mathrm{km} \cdot \mathrm{h}^{-1}$. Uvažujeme-li délku vozidla $l_{uv} = 5\ \mathrm{m}$, je kapacita ulice $C_u = 20$ vozidel. Následující text popisuje děje, které probíhají při simulaci ulic modelovaných Petriho sítěmi na obrázcích 29, 30 a 32. Tyto tři modely se ve své konstrukci liší jen mírně. Jak vývoj jejich značení ukazuje, výsledky poskytují obraz různých situací silničního provozu.

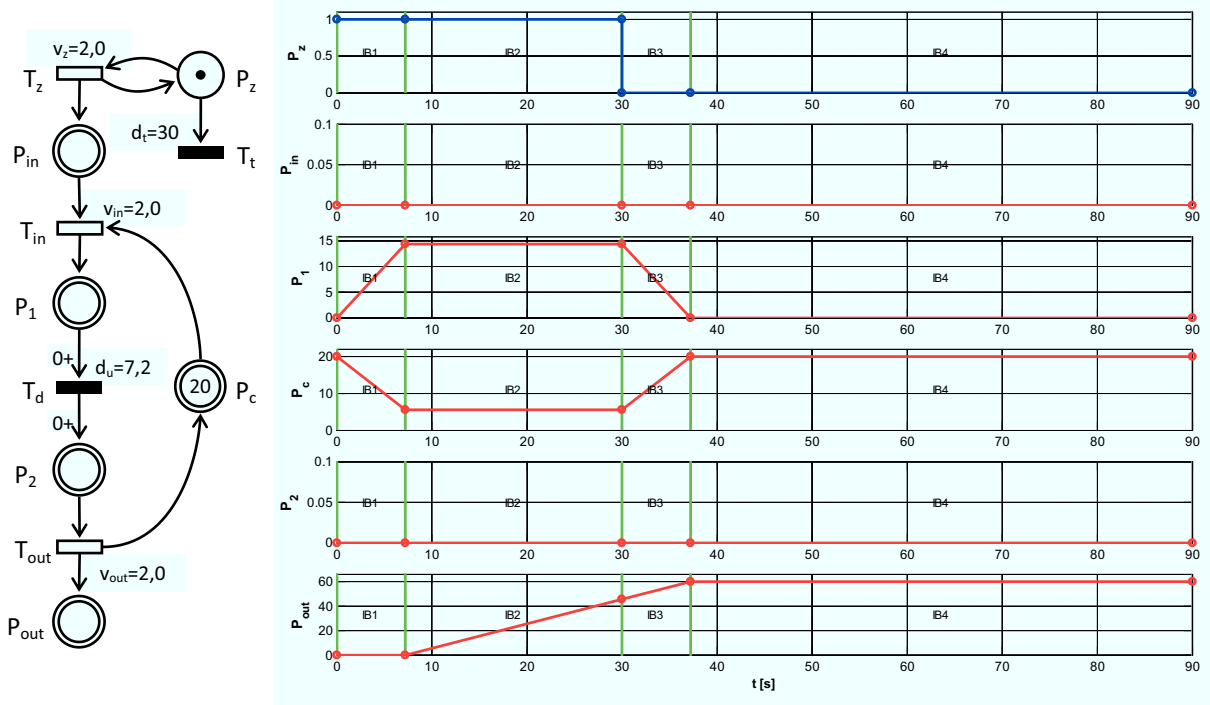
Obrázek 29 ukazuje jednoduchý model ulice a vývoj jeho značení:

IB 1 Vozidla vjíždí do ulice. Ta se plní, ale žádné z vozidel zatím nedojelo až na konec a výstupní místo ulice je stále prázdné.

IB 2 Do ulice vjíždí stejný počet vozidel, jako jí na jejím konci opouští.

IB 3 Již žádné nové vozy nepřijíždějí, ulice se vyprázdní do výstupního místa.

IB 4 V Petriho síti došlo k uváznutí, žádný další vývoj neprobíhá.



Obrázek 29 – HPN model jednoduché ulice

Model ukázaný na obrázku 30 obsahuje navíc semafor, který vytváří frontu vozidel čekajících v ulici. Provoz v takové ulici probíhá takto:

IB 1 Vozidla vjíždějí do ulice, ta se plní.

IB 2 První vozy dojedou až na konec, ale mají červenou. Zůstávají v ní stát a ulice je naplněna.

IB 3 Vozidla, jež se do ulice nevejdou, zůstávají stát mimo ní v místě P_{in}.

IB 4 Vozy v ulici stojí, čekají na signál povolující průjezd.

IB 5 Je zelená. Vozy ulici opouštějí, nové do ní bez prodlení vjíždějí (což není zcela korektní, jak je ukázáno dále). Je stále plně obsazena.

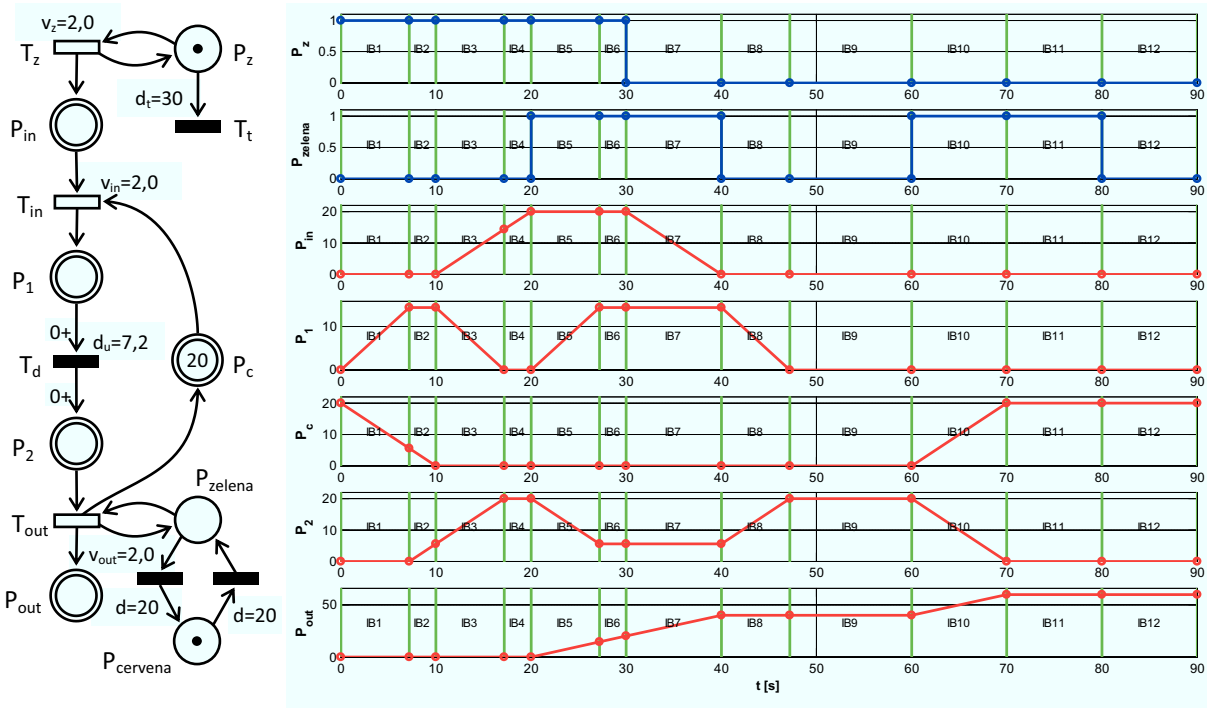
IB 6 Zelená trvá, odjíždějící vozy jsou nahrazovány novými.

IB 7 Zdrojový přechod T_z je zastaven. Vozidla čekající na vjezd do ulice se dostávají na řadu.

IB 8 Červená. Žádné vozy neodjíždějí a tím se tvoří fronta v místě P_2. Nová vozidla nepřijíždějí.

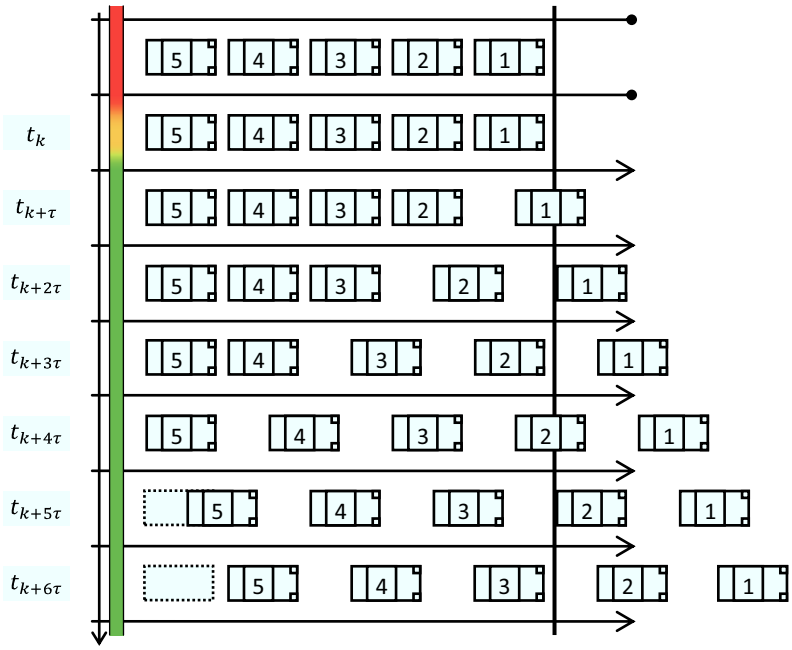
IB 9 Protože je provoz zastaven, nedochází k žádnému pohybu.

IB 10 Zelená. Zbývající vozidla čekající v P_2 odjedou, ulice je nyní prázdná.



Obrázek 30 – Modelování ulice zakončené světelnou signalizací

Petriho síť na obrázku 32 stejně jako předchozí typ popisuje ulici zakončenou semaforem. Kromě toho však obsahuje zpoždění i ve zpětné vazbě. Toto zpoždění má za úkol simulovat časovou prodlevu, se kterou se jednotlivé vozy čekající ve frontě rozjíždějí. U toku vozidel (na rozdíl od kapaliny), může zanedbání postupného rozjíždění znamenat značnou nepřesnost modelu. Znázornění celého jevu lze nalézt na obrázku 31.



Obrázek 31 – Vývoj rozjíždějící se fronty vozidel

Chceme-li stanovit hodnotu zpoždění ve zpětné kapacitní smyčce, musíme znát časový rozdíl rozjezdu jednotlivých vozidel. Označme tuto dobu τ. Její hodnota byla průměrováním několika měření

vypočtena přibližně na $\tau = 0{,}3\ s$. Dobu zpoždění zpětné smyčky pro ulice o kapacitě C_u lze vypočíst podle vztahu

$$d_{u2} = C_u \cdot \tau\ . \tag{25}$$

V případě uvažované ulice tedy

$$d_{u2} = 20 \cdot 0{,}3 = 6\ s\ .$$

Vývoj značení hybridní Petriho sítě modelující ulici včetně zpoždění při rozjíždění fronty vozidel čekajících na dopravní signalizaci je ukázán na obrázku 32. Zde je popis vývoje zobrazené situace:

IB 1 Vozidla vjíždějí do ulice. Ta se plní.

IB 2 První vozy dojedou až na konec, ale mají červenou. Zůstávají v ní stát a ulice je naplněna.

IB 3 Vozidla, jež se do ulice nevejdou, zůstávají stát mimo ní v místě P_{in}.

IB 4 Vozy v ulici stojí, čekají na signál povolující pokračování jízdy.

IB 5 Je zelená. Vozy ulici postupně opouštějí, ale nové do ní zatím vjet nemohou. Na začátku ulice není uvolněno místo, kam by nová vozidla mohla najet.

IB 6 Zelená trvá, fronta čekajících vozidel se již rozjela a nová vozidla vjíždějí do ulice.

IB 7 Přechod generující nová vozidla na vstupu je zastaven. Vozidla projíždějí ulicí a nějakou dobu trvá, než dojedou na její konec. Značení výstupního místa P_{out} se nemění, protože čekající vozidla již ulici opustila a nová ještě nedojela na její konec. Tuto mezeru způsobuje předpoklad, že rozjíždějící se auta okamžitě dosáhnou maximální rychlost.

IB 8-10 Vozidla projíždějí ulicí a opouštějí ji po dobu, kdy svítí zelené světlo.

IB 9-13 Na konci ulice se tvoří fronta čekající na zelenou. Vozy stále přijíždějí.

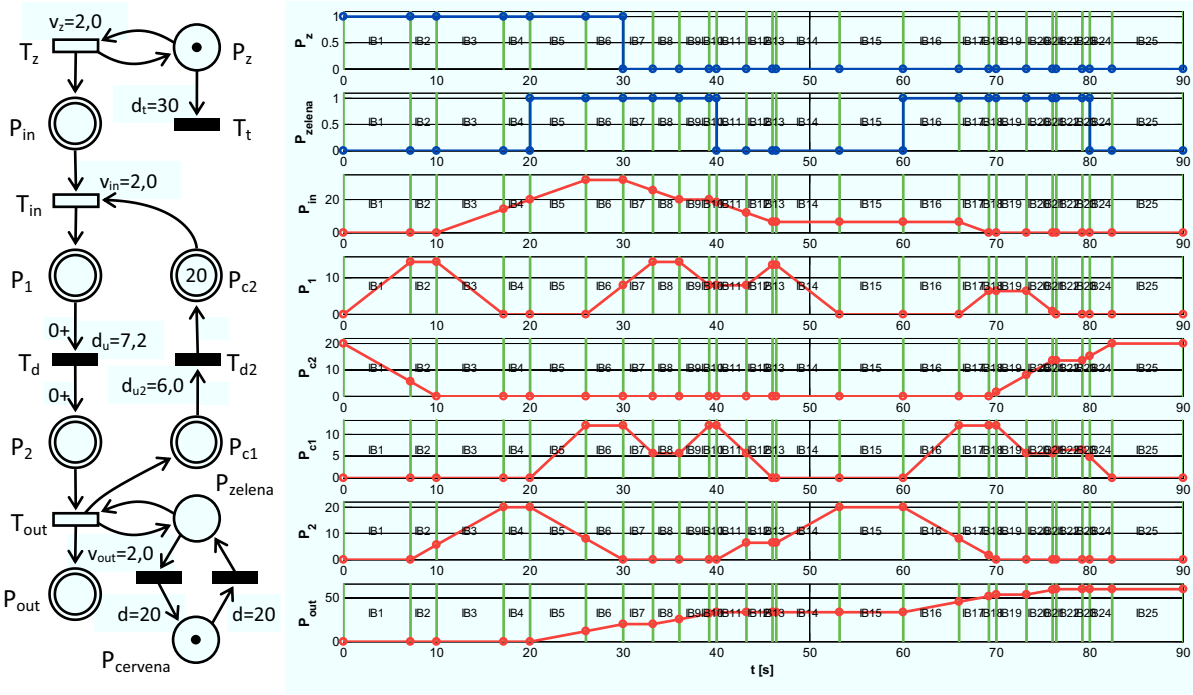
IB 10 Fronta nyní blokuje ulici v celé její délce. Před ulicí stále čeká několik málo vozů.

IB 11 Čeká se na skončení červené fáze řídícího cyklu.

IB 12 Fronta se postupně rozjíždí a vozy opouštějí ulici.

IB 13 Ulice je vyprázdněna natolik, že vozy čekající před ní mohou vjet.

IB 14 -24 Vozidla projíždějí ulicí, ta ještě v této fázi řídícího cyklu je zcela vyprázdněna.



Obrázek 32 – Modelování ulice zakončené semaforem modelující frontu čekajících vozidel

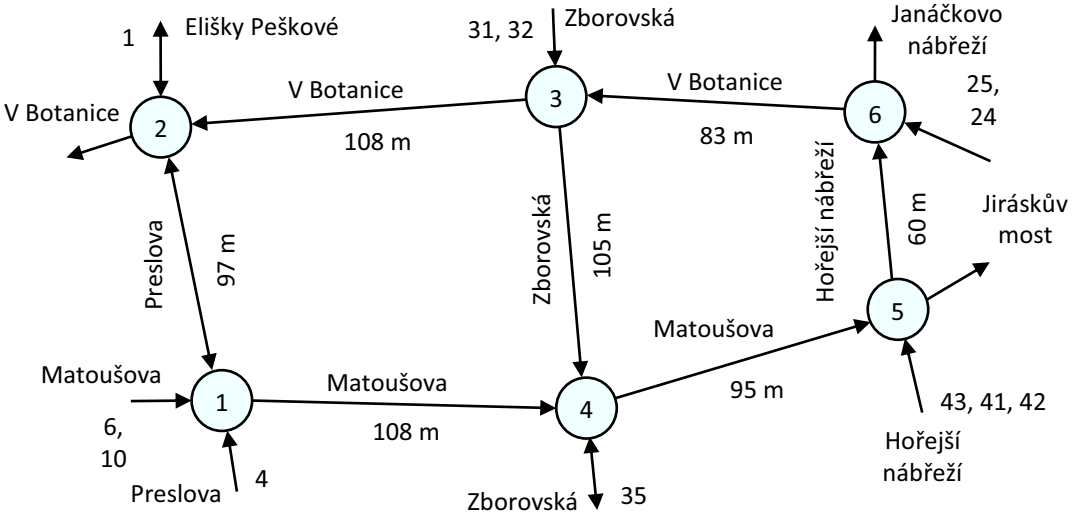
8. Simulace reálné oblasti

8.1. Zvolená oblast

Pro simulaci celé soustavy křižovatek byla vybrána malá oblast na pražském Smíchově, jejíž přibližná pozice je 50°4’29”N, 14°24’26”E. Fotografie na obrázku 33 oblast ukazuje. Jedná se o 6 vzájemně propojených řízených křižovatek. Úkolem této kapitoly je ukázat využití navrženého způsobu modelování dopravních sítí řešeného pomocí implementovaného simulátoru a jeho modulů.

Přehled všech křižovatek je uveden v příloze 3. Ta obsahuje obrázky a diagramy popisující fáze řízení jednotlivých křižovatek a tabulky s naměřenými a vypočtenými parametry křižovatek, jejichž význam stejně jako způsob jejich výpočtu byl popsán již v kapitole 1.

Obrázek 33 – Letecký snímek ukazující vybranou oblast. Jeho zdrojem je ©GEODIS Brno, s.r.o. (cit. 4. 12. 2009).



Obrázek 34 – Náhled na seskupení pražských křižovatek vybraných pro simulaci

Schematický graf na obrázku 34 obsahuje kromě čísel, jimiž budeme nadále jednotlivé křižovatky označovat, také jejich vzdálenosti nezbytné pro výpočet dopravních zpoždění. Kurzívou tištěná čísla u vstupů do oblasti reprezentují detektory zaznamenávající průjezdy vozidel a tedy vstupní signály obsažené v souhrnu dat naměřených ve smíchovských ulicích. Použití některých těchto průběhů lze nalézt již v kapitole 1.

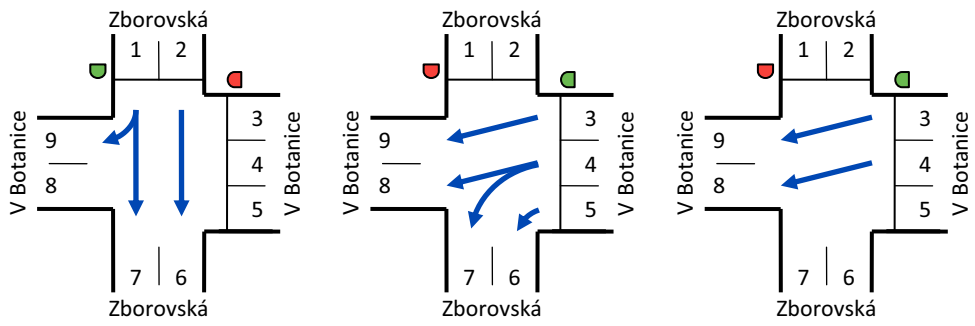
Celá simulace bude provedena na modelech křižovatek vytvořených pomocí spojitých Petriho sítí s konstantní maximální rychlostí přechodů (CCPN). Jednotlivé křižovatky jsou propojeny ulicemi modelovanými hybridní Petriho sítí (HPN) podle obrázku 29 na straně 43. Jejich parametry jsou uvedeny v tabulce 3. Druhý sloupec značí zdrojovou, třetí cílovou křižovatku. Můžeme vidět, že všechny ulice jsou jednosměrné s výjimkou ulice Preslova v úseku mezi křižovatkami 1 a 2. Parametry v posledních dvou sloupcích zpoždění $\boldsymbol{d_{z\to c}}$ a kapacita ulice $\boldsymbol{C_{z\to c}}$ byly vypočteny pomocí vztahů (23) a (24) na straně 41.

Tabulka 3 – Parametry ulic (zdrojem a cílem je míněna zdrojová a cílová křižovatka)

Ulice	$\boldsymbol{z}$ (zdroj)	$\boldsymbol{c}$ (cíl)	$\boldsymbol{n_{z\to c}}$	$\boldsymbol{l_{z\to c}}$ [m]	$\boldsymbol{w_{z\to c}}$ [km·h^{-1}]	$\boldsymbol{d_{z\to c}}$ [s]	$\boldsymbol{C_{z\to c}}$ [uv]
Preslova	1	2	1,5	97	50	7,0	29,1
	2	1	1,5	97	50	7,0	29,1
V Botanice	3	2	2,3	108	50	7,8	49,7
	6	3	2,6	83	50	6,0	43,2
Zborovská	3	4	2,0	105	50	7,6	42,0
Matoušova	1	4	2,0	108	50	7,8	43,2
	4	5	2,0	95	50	6,8	38,0
Hořejší nábřeží	5	6	2,0	60	50	4,3	18,0

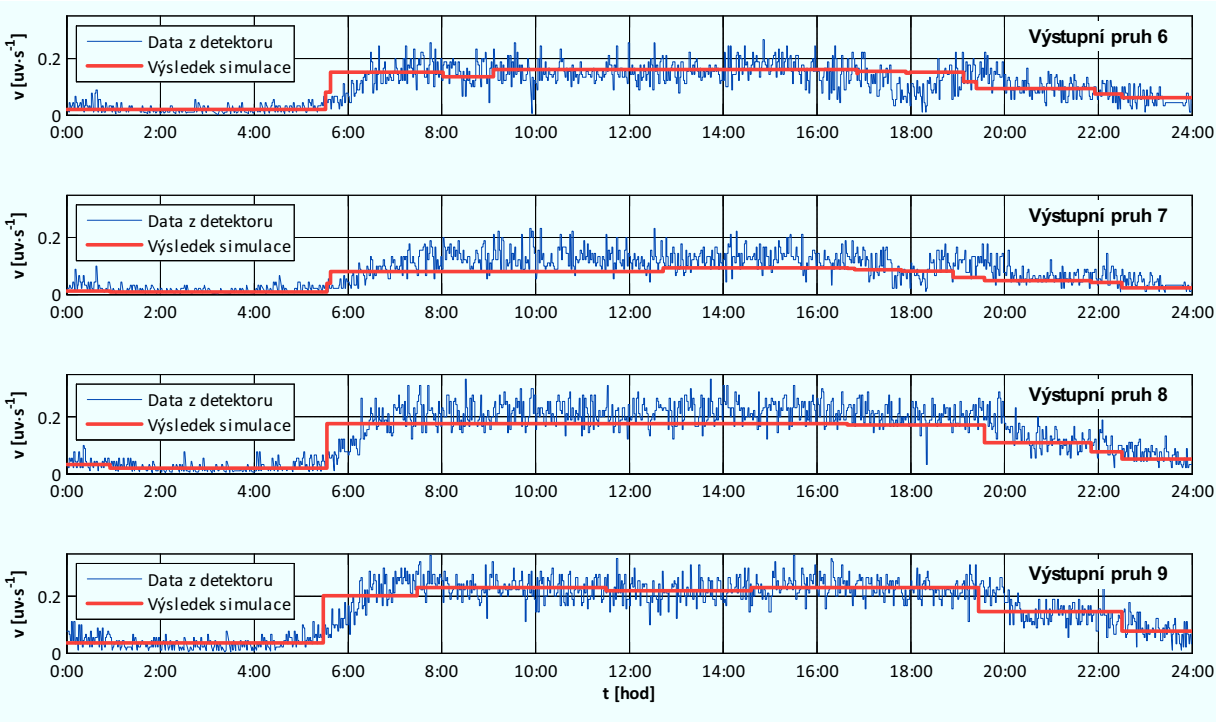
Protože doba průjezdu vozidla křižovatkou nemůže být v CCPN modelu křižovatky zachycena, zahrnuje ji zpoždění v příslušných ulicích.

8.2. Ukázka simulace křižovatky 3



Obrázek 35 – Křižovatka 3: V Botanice – Zborovská
50°4'31.5"N, 14°24'27.4"E

Křižovatka je reprezentovaná CCPN modelem, který lze nalézt v adresáři obsahujícím i ostatní modely křižovatek v pnml formě a lze jej tedy otevřít v editoru Petriho sítí. Tento model je stejně jako modely ostatní sestaven na základě údajů obsažených v příloze 3. V tomto případě máme 5 vstupních a 4 výstupní jízdní pruhy. Pro každý z nich jsou k dispozici naměřené průběhy během jednoho celého dne. Jsme tedy schopni porovnat simulátorem vygenerovanou odezvu na reálný vstupní signál s naměřeným výstupním signálem. Na obrázku 36 je toto porovnání ukázáno pro každý výstupní jízdní pruh. Zobrazené průběhy představují rychlosti přechodů v_c [uv·s^{-1}] v závislosti na denní době.



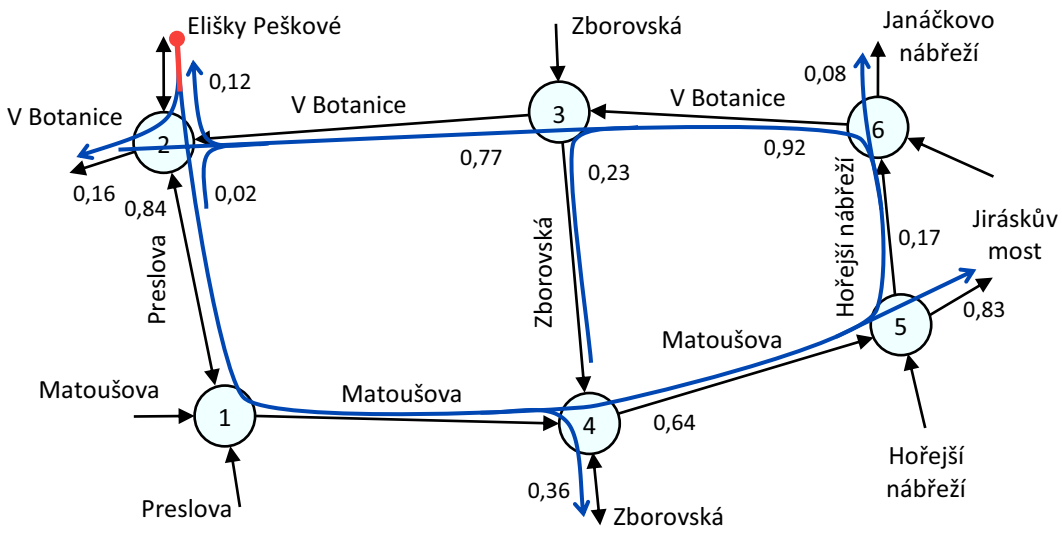
Obrázek 36 – Porovnání výstupů CCPN modelu i reálné křižovatky 3 v závislosti na čase

8.3. Studie oblasti šesti křižovatek

Simulujeme-li celý systém šesti křižovatek a jako vstupy použijeme naměřená data, získáme průběhy obdobné těm v předchozí podkapitole. Zajímavější proto je provést modifikaci jednoho ze vstupů a sledovat odezvu na tuto změnu.

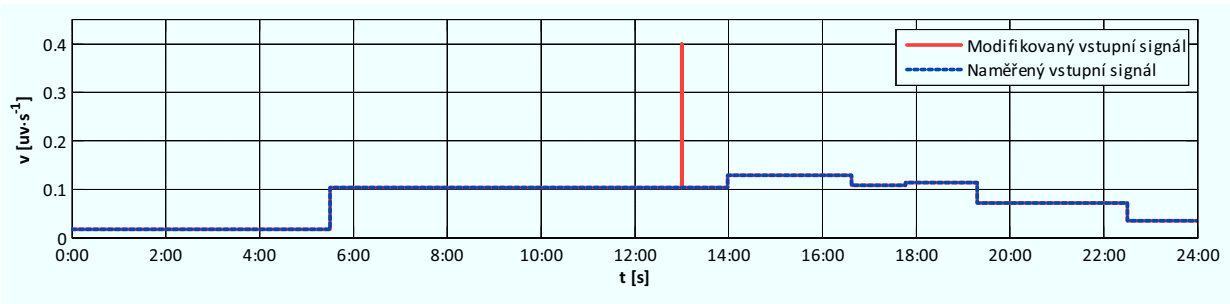
8.3.1. Vstupní signál

Pro tento účel využijeme vstupní signál číslo 1 – vstup do studované oblasti z ulice Elišky Peškové. Signál je doplněn o pulz délky 10 s umístěný do času (46800 s) 13:00 hod. V reálu si tuto změnu lze představit jako náhlý příjezd kolony vozidel. Tím se tok vozidel do křižovatky 2 z ulice Elišky Peškové dočasně znatelně zvýší. Na jednotlivých výstupech či v různých segmentech modelu budeme pozorovat zpožděnou „dopravní špičku“.



Obrázek 37 – Průjezd vozidel z ulice Elišky Peškové jednotlivými křižovatkami s odbočovacími poměry.

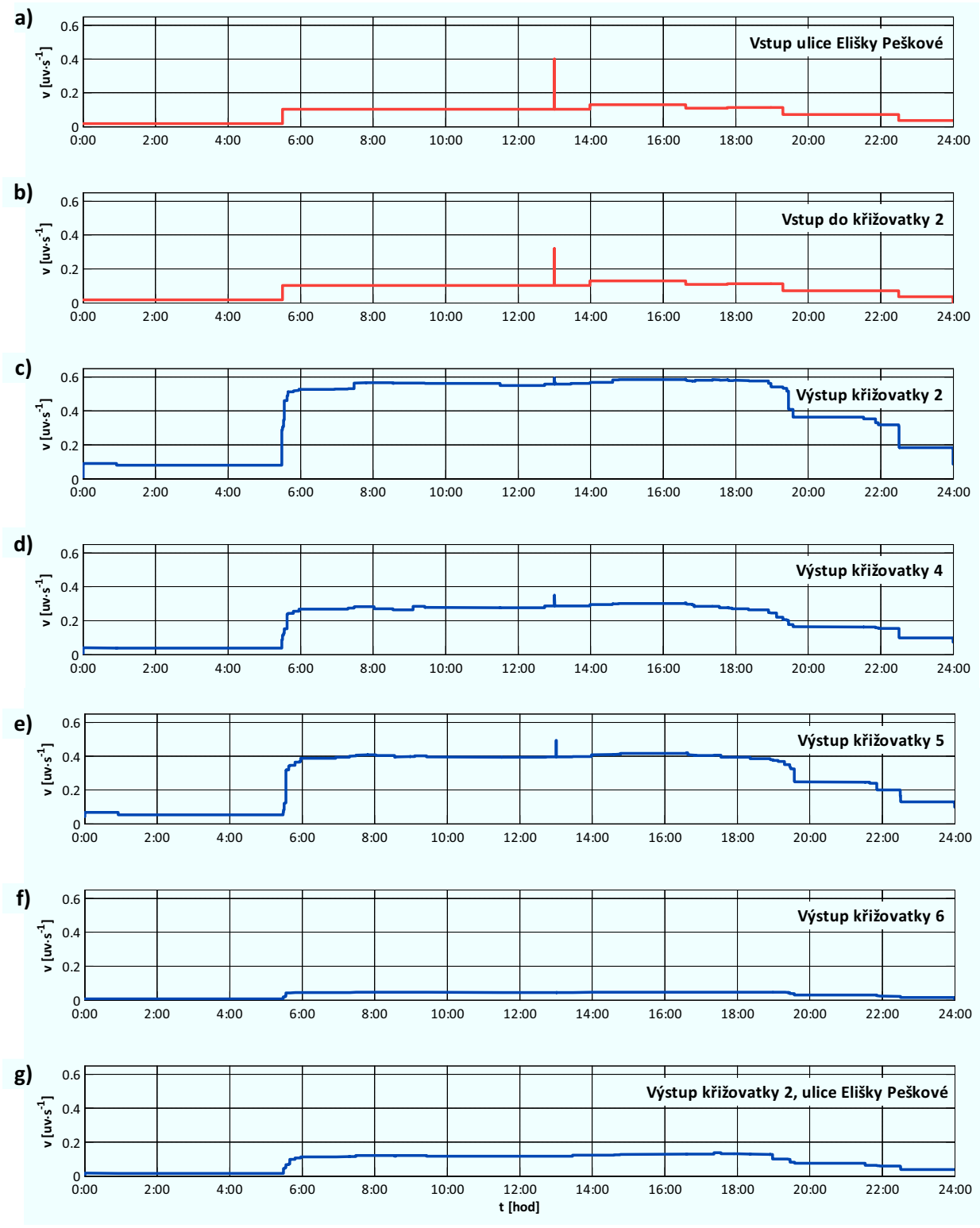
Obrázek 37 znázorňuje trasu pohybu vozidel, které do oblasti vjíždějí z ulice Elišky Peškové. Modrá křivka ukazuje průjezd jednotlivými křižovatkami a větvení toku vozidel v poměrech, které je možné nalézt v příloze 3. Zajímavá situace nastane, když se už jen zlomek původního pulzu dostane se zpožděním na křižovatku 3. Odsud se totiž část toku vydá ke křižovatce 4, čímž dojde k uzavření smyčky – několik vozidel jezdí v kruhu. Obdobný jev nastane i u křižovatky 2. Tento úkaz při simulaci vzniká, protože model je postaven na základě statisticky zjištěných odbočovacích poměrů a nelze do něj zapracovat směr jízdy a destinaci každého z vozidel. Sekundární pulzy, které takto vznikají, mají nízkou hodnotu: pro křižovatku 3 1%, pro křižovatku 2 0.04% předchozí hodnoty. Jejich vliv na pokračující simulaci je zanedbatelný.



Obrázek 38 – Originální a modifikovaný signál. Modifikací je pulz délky 10 s.

8.3.2. Výsledek simulace

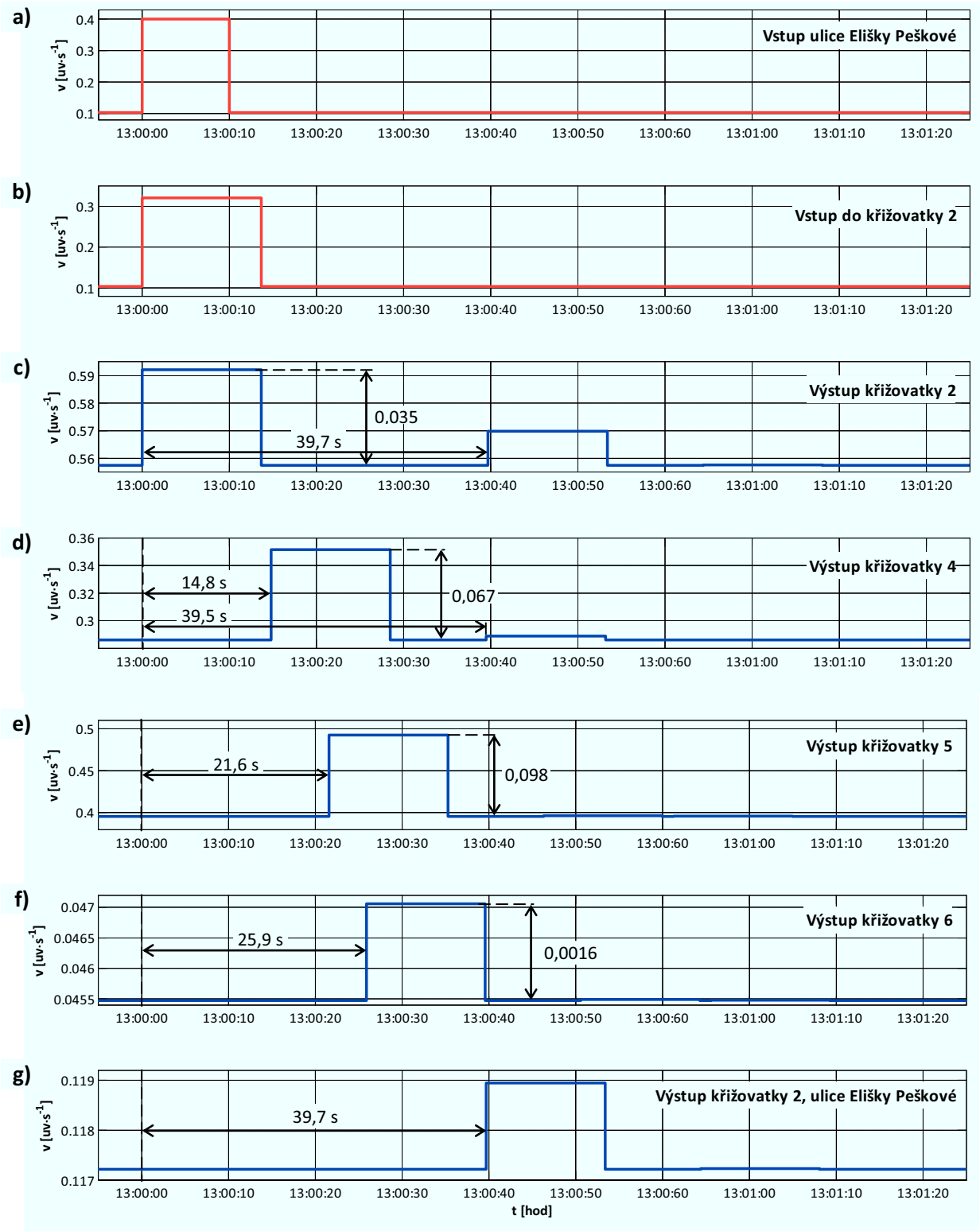
Modelovaný systém křižovatek má celkem 7 vstupních a 5 výstupních ulic, jak je patrné z obrázků 34 i 37. Na vstupy modelu byly přivedeny naměřené signály detektorů uložené na CD – příloha 1. Přiřazení jednotlivým vstupům je patrné z obrázku 34. Signál přivedený na vstupní ulici Elišky Peškové byl pozměněn tak, jak je popsáno v předchozím oddílu. Výsledky simulace v podobě rychlostí spojitých přechodů Petriho sítě jsou vykresleny na obrázku 39.



Obrázek 39 – Výsledky simulace oblasti šesti křižovatek modelované Petriho sítí.

Na obrázku 39a je pro porovnání zobrazen modifikovaný vstupní signál. Ten se mírně liší od hodnot, které skutečně na vstupním přechodu modelu křižovatky 2 pozorujeme – obrázek 39b. Dodatečně doplněný pulz má hodnotu 0,40 uv·s^{-1}, avšak maximální rychlost přechodu byla na základě řídícího cyklu křižovatky vypočtena na 0,32 uv·s^{-1}. Velikost původního impulzu délky 10 s je tedy zmenšena a

doba jeho trvání prodloužena na 13,8 s. To znamená, že při průjezdu křižovatkou 2 dochází k roztažení projíždějící kolony vozidel.



Obrázek 40 – Výsledky simulace oblasti šesti křižovatek modelované Petriho sítí – detail.

Průběhy zobrazené modrou křivkou jsou výstupy odpovídající modrým šipkám na obrázku 37. Jedná se o výsledky skládání vstupních signálů podle dělicích poměrů větvení toku vozidel při průjezdu křižovatkami. Na obrázku 39c, d a e vidíme odezvu na přidaný vstupní pulz. Ve zbývajících dvou přípa-

dech není patrná, ale při dostatečném zvětšení ji nalezneme i zde. O tom svědčí obrázek 40. Je zde zřetelně vidět změna na vstupu křižovatky popsanou v předchozím odstavci. Toto detailní zobrazení rychlostí výstupních přechodů ilustruje zpoždění „dopravní špičky“, které během jízdy oblastí narůstá. Zpoždění lze přesně dopočítat sečtením příslušných hodnot z tabulky 3. Pomocí dělicích poměrů vypsaných v obrázku 37 můžeme snadno dopočítat i amplitudu zobrazených pulzů. Ta rychle klesá úměrně tomu, kolik vozidel opouští oblast během průjezdu křižovatkami. V detailním zobrazení výstupních signálů můžeme také rozeznat zpožděné sekundární pulzy vzniklé jako důsledek smyček vyskytujících se v seskupení křižovatek.

Z výsledku experimentu je patrné, že Petriho síť vytvořená spojením modelů křižovatek a ulic představuje natolik věrný obraz skutečnosti, nakolik změřené a vypočtené parametry odpovídají reálnému stavu studované oblasti.

Kompletní model oblasti se skládá ze 129 míst a 106 přechodů, z čehož je 8 přechodů diskrétních. Ty představují zpoždění průjezdu ulicemi. Vývoj sestavené Petriho sítě je podmíněn vstupními daty. Na jejich základě bylo vygenerováno 1388 IB-stavů. Vysoký počet stavů je způsoben výskytem smyček v modelu, díky nimž se každá změna na vstupu promítne s příslušným zpožděním do všech dalších částí modelu a to hned několikrát. Celý výpočet[4] trval 2415 s.

[4] Simulace byla provedena na osobním počítači vybaveném Intel Core Duo CPU T2350 @1.86 GHz, 1 GB RAM.

9. Závěr

Tato práce je věnována jedné z mnoha oblastí využití modelovacího a verifikačního nástroje nazývaného Petriho sítě. V první části jsou popsány základní pravidla a možnosti tohoto formalizmu. V dalších části byl uvedený matematický i slovní popis převeden do programové podoby prostředí MATLAB. Výsledkem je implementace simulátoru Petriho sítí. Jedná se o modulární nástroj podporující diskrétní, spojitý i hybridní pohled na modelování světa, jehož každá část může být využita samostatně. Společně tento soubor nezávislých funkcí, původně zamýšlený pro využití simulování dopravních sítí, nabízí využití při simulaci různých systémů i mimo cílovou oblast.

Simulátor umožňuje načítání Petriho sítí definovaných PNML strukturou do matematické podoby. Pro vytvoření komplexní modelů byl implementován algoritmus spojující Petriho sítě podle předem nastaveného jednoduchého předpisu. Simulační funkce dokáže na základě zadaných parametrů generovat vývoj sítě, který nakonec můžeme studovat ve dvou podobách: jako závislost značení (příp. skutečných rychlostí přechodů) na čase nebo jako vývojový diagram. Volba realizace simulátoru v prostředí MATLAB zajišťuje snadnou rozšiřitelnost a kompatibilitu stejně jako intuitivnost použití pro uživatele, kteří s tímto systémem pracují.

Na základě publikací autorů zabývajících se touto problematikou byly sestaveny různé modely popisující řízenou křižovatku. Porovnáním jejich výstupů při totožných vstupních signálech byly ukázány silné i slabé stránky každého ze zvolených přístupů: modelování diskrétní Petriho sítí vyústí ve vysokou přesnost – popisuje chování každého vozidla samostatně, s tím je ale spojen vysoký počet stavů vývoje přinášející vysokou paměťová a časovou náročnost řešení. Spojitý přístup je z hlediska počtu stavů úsporný, pomocí něj však nelze sledovat děje probíhající během jednotlivých fází řídicího cyklu. Hybridní model nabízí kompromis v počtu stavů i přesnosti simulace, ale výpočetní doba je v závislosti na algoritmu řešení poměrně dlouhá. Při výběru vhodného modelu tedy záleží na konkrétní aplikaci.

Poslední část práce přináší pohled na celou oblast složenou ze separátně modelovaných křižovatek. Po zavedení dopravního zpoždění simulujícího průjezd vozidel ulicemi jsme schopni sledovat vývoj jednotlivých částí systému při změně na vstupu. Pozorujeme nejen distribuci částí toku vozidel mezi jednotlivé výstupy z oblasti, ale také zpoždění výstupních signálů.

Tato práce řešenou problematiku neuzavírá. Naopak nabízí možnosti rozšíření mnoha směry: implementovaný simulátor může být obohacen o funkce analýzy Petriho sítí, může být doplněn o podporu dalších typů používaných Petriho sítí či o vizualizační nástroje. Lze provést mnoho dalších experimentů zahrnujících například modelování křižovatek neřízených nebo zavést speciální typy vozidel jako jsou vozidla hromadné dopravy či vozidla s právem přednosti v jízdě. Dokonce lze celou oblast silniční dopravy opustit a využít obdobných principů v oblasti jiných druhů transportu. Možnosti využití Petriho sítí jsou veliké a jejich vývoj se jistě nezastaví.

10.Seznam použitých zdrojů

[David-Alla]
DAVID, René; ALLA, Hassane. *Discrete, Continuous, and Hybrid Petri Nets*. Springer, November 23, 2004. ISBN 3-540-22480-7

[David I]
DAVID, René; ALLA, Hassane. *On Hybrid Petri Nets*. Discrete Event Dynamic Systems: Theory and Applications, 11, 9-40. Kluwer Academic Publishers. Boston, 2001.

[Di Febbraro I]
DI FEBBRARO, Angela; GIGLIO, Davide; SACCO, Nicola. *Urban traffic control structure based on hybrid Petri nets*. (2004) ITS, 5, 224-237.

[Di Febbraro II]
DI FEBBRARO, Angela; GIGLIO, Davide; SACCO, Nicola. *Modular Representation of Urban Traffic Systems based on Hybrid Petri Nets*. IEEE Intelligent Systems Conference Proceedings. Oakland (CA) USA, 2001. 0-7803-7194-1/01.

[Di Febbraro III]
DI FEBBRARO, Angela; SACONE, Simona. *Hybrid Modelling of Transportation Systems by means of Petri Nets*. Department of Communications. Computer and System Sciences. University Of Genova. Genova, 1998. IEEE, 0-7803-4778-1/98.

[Di Febbraro IV]
DI FEBBRARO, Angela; SACCO, Nicola. *On modelling urban transportation networks via hybrid Petri nets*. Department of Computer and Control Engineering. Polytechnic of Turin, Corso Duca degli Abruzzi 24, 10129 Turin, Italy. (2003)

[Dotoli]
DOTOLI, Mariagrazia; FANTI, Maria P. *An urban traffic network model via coloured timed Petri nets*. Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari, Via Re David 200, 70125 Bari, Italy (2006)

[Gallego]
GALLEGO, Jean-Luis; FARGES, Jean-Loup; HENRY, Jean-Jacques H. *Design by Petri Nets of an intersection signal controller.* Automatic Research Department CERT/ONERA, 2, Av E. Belin BP 4025, 31055, Toulouse, France (1996).

[Hanzálek I]
HANZÁLEK, Zdeněk. *Petriho sít*ě [online]. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra řídicí techniky. Praha (2006). [cit. 2009-10-20]. <http://dce.felk.cvut.cz/hanzalek/prednasky/petri.pdf>.

[Hanzálek II]
HANZÁLEK, Zdeněk. *Petriho sít*ě *a čas*. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra řídicí techniky. Praha (2006). Studijní materiál předmětu X35DRS.

[Hanzálek III]
HANZÁLEK, Zdeněk. *Continuous Petri Nets and Polytopes*. Czech Technical University in Prague. Falculty of Electrical Engineering. Department of Control Engineering. Centre for Applied Cybernetics. Prague (2003). 0-7803-7952-7/03/2003 IEEE.

[Júlvez]
JÚVEZ, J. – BOEL, R. *Modelling and Controlling Traffic Behaviour with Continuous Petri Nets*. Departamento de Informática e Ingeniería de Sistemas. Universidad de Zaragoza. Zaragoza, 2005.

[Kutil I]
KUTIL, Michal; HANZÁLEK, Zdeněk. *Light controlled Intersection Model Based on the Continuous Petri Net.* Czech Technical University in Prague. Faculty of Electrical Engineering. Department of Control Engineering. Prague, 2008.

[Mourni]
MOURNI, Iyad; HENNEQUIN, Sophi; XIC, Xialon. *Continuous Petri nets with delays for performance evaluation of transfer lines.* International Conference on Robotics and Automation. Barcelona, Spain, April 2005

[Papageorgiou]
PAPAGEORGIOU, Markos; DIAKAKI, Christina; DINOPOULOU, Vaya; KOTSIALOS, Apostolos; WANG, Yibing. *Review of Road Traffic Control Strategies*. Technical University of Crete, Dynamic Systems and Simulation Laboratory, GR-73100 Chania, Greece (2003)

[PNML]
WEBER, M. *Petri Net Markup Languege* [online]. [cit. 2009-12-08]. <http://www2.informatik.hu-berlin.de/top/pnml/about.html>.

[Relax NG]
CLARK, J. *RELAX NG home page* [online]. [cit. 2009-12-08]. <http://relaxng.org/>.

[Stibor]

STIBOR, Miloslav. *Simulátor transportních sítí*. České vysoké učení technické v Praze, Fakulta elektrotechnická, Praha (2007). Diplomová práce.
<http://dce.felk.cvut.cz/dolezilkova/diplomky/2007/dp_2007_stibor_miloslav/dp_2006_ Stibor_Miloslav.pdf>.

[Svádová]

SVÁDOVÁ, Martina; HANZÁLEK, Zdeněk. *PN Matlab Toolbox 2.0*. Czech Technical University in Prague. Falculty of Electrical Engineering. Department of Control Engineering. Centre for Applied Cybernetics. Prague (2005).

[Tolba]

TOLBA, Chérif; THOMAS, Philippe; ELMOUDNI, Abdellah. Performances evaluation of the traffic control in a single crossroad by Petri nets. Université de Technologie de Belfort-Montbéleard, 90010 Belfort cedex, France (2003).

[Tzes]

TZES, Anthony; KIM, Seongho; MCSHANE, William R. *Applications of Petri Networks to Transportation Network Modeling.* IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 45, NO. **2,** MAY 1996

[VSPNE]

ŠREMER, M. *Editor pro Stochastické Petriho sítě na WWW pomocí Java technologií*. České vysoké učení technické v Praze. Fakulta elektrotechnická. Katedra řídicí techniky. Praha, 2006. Diplomová práce.
<http://dce.felk.cvut.cz/dolezilkova/diplomky/2006/dp_2006_sremer_martin/dp_2006_Sremer_Martin.pdf>.

[Wang]

WANG, Jiacun; JIN, Chun; DENG, Yi. *Performance Analysis of Traffic Networks Based on Stochastic Timed Petri Net Models*. School of Computer Science, Florida International University
Miami, FL33199

A. Příloha 1 – Přiložené CD

Součástí práce je přiložený CD-ROM obsahují zdrojové kódy implementovaného simulátoru a všech jeho součástí včetně definic stylů XSL, skripty pro vykreslení obrázků použitých v této práci pro umožnění zopakování provedených experimentů, knihovnu PNML definic Petriho sítí obsažených v této práci, podklady pro simulaci dopravy v regionu Praha – Smíchov (data zobrazená v příloze 4), softwarové aplikace použité při vývoji této práce a tento dokument v elektronické podobě.

1. Simulátor a jeho součásti

Simulátor s jeho programovými moduly je umístěný v adresáři *pnsimulator/*. Nápověda každé z funkcí je součástí zdrojového kódu a může být zobrazena tak, jak je zvykem u všech funkcí v prostředí MATLAB. Pro použití a správnou funkci je doporučeno, přidat si tento adresář mezi cesty MATLABu.

2. Skripty pro provedení experimentů a generování ilustrací

Skripty umožňující zopakování experimentů i generování obrázků jsou umístěny v adresáři *skripty/*. Tvarem jejich jména je identifikována kapitola této práce, které náleží. Ke složitějším experimentům je výsledek simulace – kompletní vývoj modelu přiložen ve formě binárního souboru, který je možné načíst MATLABem.

3. Soubor Petriho sítí ve formě PNML

V adresáři *trafficnets/* jsou umístěny modely popisující křižovatky, ulice a jejich spojení. Tyto modely jsou vytvořeny přiloženou aplikací PN Editor 1.4 a mohou v ní být opětovně editovány.

4. Podklady pro simulaci dopravy

Podklady pro simulaci jsou ve formě textových souborů uloženy v adresáři *data/smichov/*.

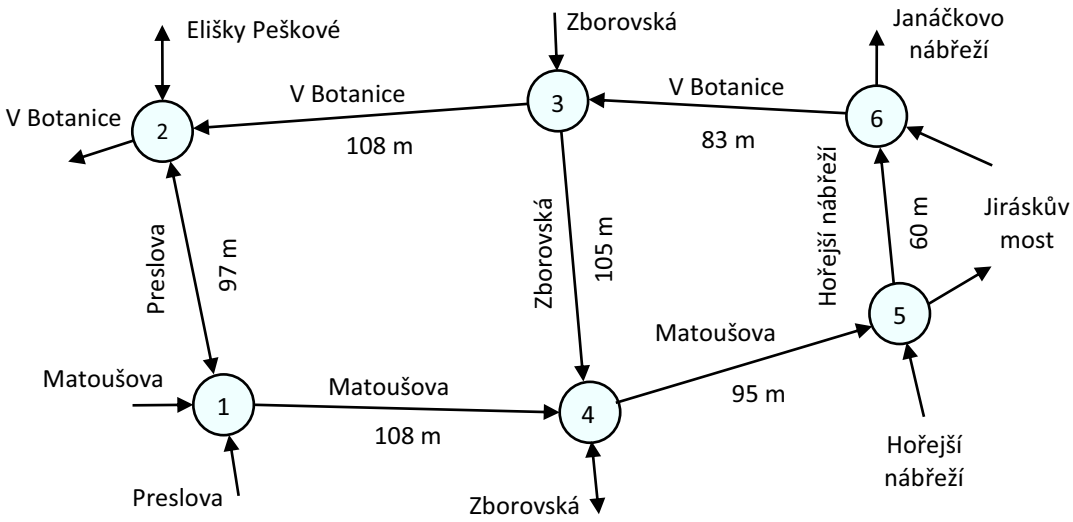
B. Příloha 2 – Seznam použitých zkratek

CCPN	Constant speed Continuous Petri Net
GPN	Generalized Petri Net
HPN	Hybrid Petri Net
IB	Invariant Behaviour (state)
LP	Linear programming
PC	Personal Computer
PN	Petri Net
PNML	Petri Net Markup Language
TPN	Timed Petri Net
TXT	Text
VSPNE	Variable Stochastic Petr Net Editor
XML	eXtensible Markup Language
XSL	XML Stylesheet Language
XSLT	XML Stylesheet Language Translator

C. Příloha 3 – Údaje o simulované oblasti

Oblast Praha Smíchov

Zeměpisná poloha: 50°4’29”N, 14°24’26”E

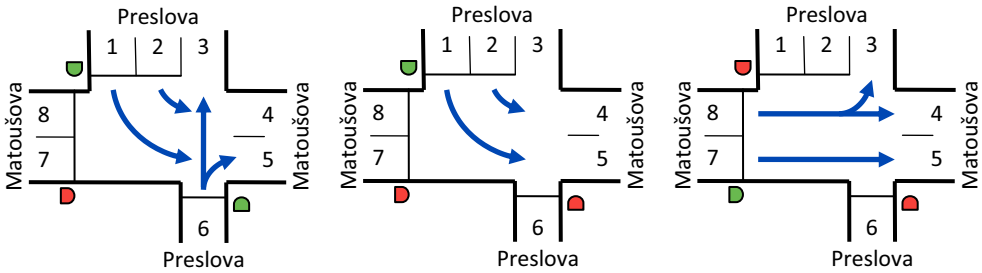


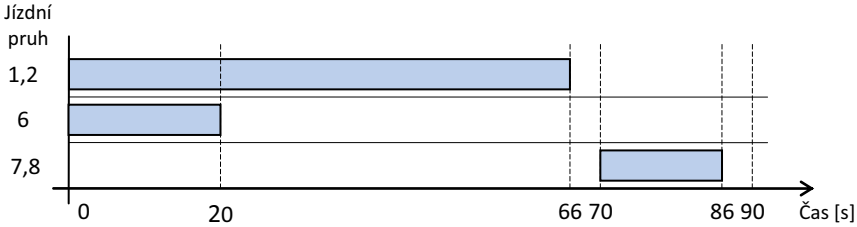
Tento dokument obsahuje soubor 6 křižovatek v oblasti Praha – Smíchov a seznam ulic se změřenými i vypočtenými parametry potřebnými pro modelování. Pro každou křižovatku jsou zde uvedeny podrobné údaje:

- náčrtek křižovatky
- diagram řídících fází včetně doby jejich trvání
- tabulka odbočovacích poměrů
- tabulka parametrů změřených i vypočítaných

Křižovatka 1: Matoušova – Preslova

Zeměpisná poloha: 50°4’27.7”N, 14°24’22.9”E



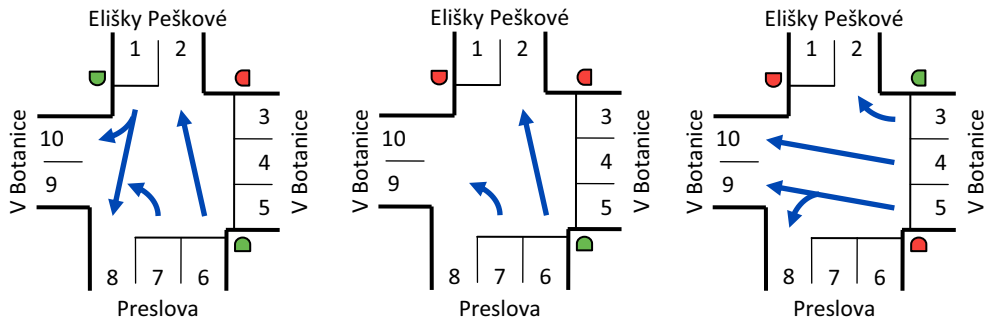


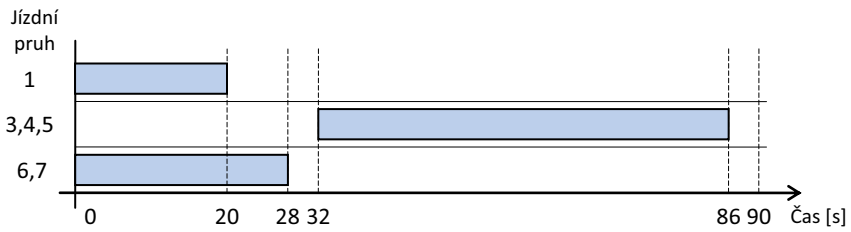
	Preslova	Matoušova
Preslova 6	0,85	0,15
Matoušova	0,25	0,75

Z (ulice)	z (zdroj)	c (cíl)	$\alpha_{Z\to z}$	$\alpha_{z\to c}$	$w_{z\to c}$ [km·h^{-1}]	$t_{z\to c}$ [s]	$d_{z\to c}$ [s·uv^{-1}]	$U_{z\to c}$ [uv·s^{-1}]	$U_{z\to}$ [uv·s^{-1}]	$V_{z\to c}$ [uv·s^{-1}]	$V_{z\to}$ [uv·s^{-1}]
Preslova	1	5	0,5	1	40	66	0,45	2,22	2,22	1,63	1,63
	1	5	0,5	1	30	46	0,60	1,67	1,67	0,85	0,85
	2	4	0,5	1	40	66	0,45	2,22	2,22	1,63	1,63
	2	4	0,5	1	30	46	0,60	1,67	1,67	0,85	0,85
	6	3	1	0,85	50	20	0,36	2,78	2.78	0,62	0,54
		5		0,15	25	20	0,72	1,39		0,31	
Matoušova	7	5	0,5	1	50	16	0,36	2,78	2,78	0,49	0,49
	8	3	0,5	0,50	30	16	0,60	1,67	2,09	0,30	0,37
		4		0,50	50	16	0,36	2,78		0,49	

Křižovatka 2: V Botanice – Elišky Peškové – Preslova

Zeměpisná poloha: 50°4’30.8”N, 14°24’21.9”E



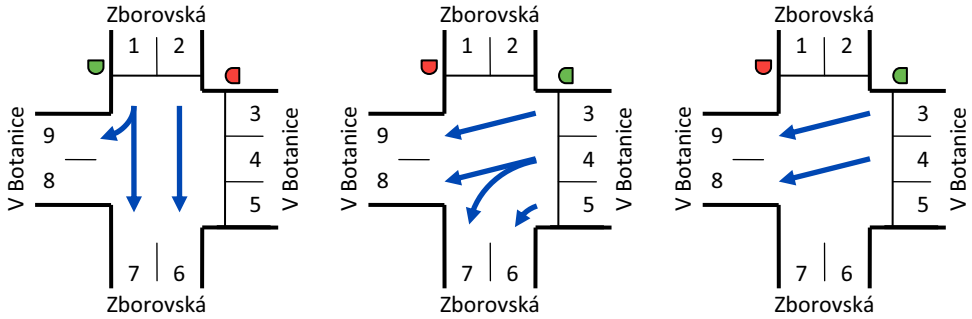


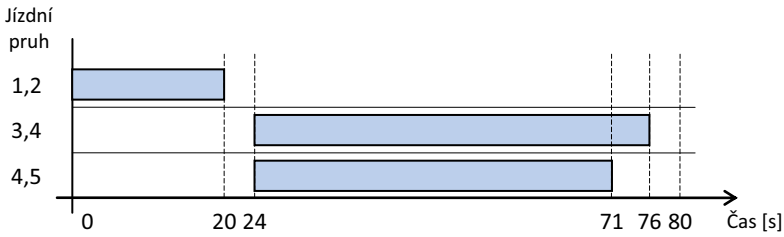
	E. Peškové	V Botanice	Preslova
E. Peškové	-	0,16	0,84
V Botanice	0,12	0,86	0,02
Preslova	0,57	0,43	-

Z (ulice)	z (zdroj)	c (cíl)	$\alpha_{Z\to z}$	$\alpha_{z\to c}$	$w_{z\to c}$ [km·h^{-1}]	$t_{z\to c}$ [s]	$d_{z\to c}$ [s·uv^{-1}]	$U_{z\to c}$ [uv·s^{-1}]	$U_{z\to}$ [uv·s^{-1}]	$V_{z\to c}$ [uv·s^{-1}]	$V_{z\to}$ [uv·s^{-1}]
Elišky Peškové	1	8	1	0,84	50	20	0,36	2,78	2,40	0,62	0,32
		10		0,16	25	20	0,72	1,39		0,31	
V Botanice	3	2	0,12	1	25	54	0,72	1,39	1,39	0,83	0,83
	4	10	0,44	1	50	54	0,36	2,78	2,78	1,67	1,67
	5	8	0,44	0,05	30	54	0,60	1,67	2,69	1,00	1,62
		9		0,95	50	54	0,36	2,78		1,67	
Preslova	6	2	0,57	1	50	28	0,36	2,78	2,78	0,86	0,86
	7	9	0,43	1	30	8	0,60	1,67	1,67	0,15	0,15
				1	30	28	0,60	1,67	1,67	0,52	0,52

Křižovatka 3: V Botanice – Zborovská

Zeměpisná poloha: 50°4’31.5”N, 14°24’27.4”E



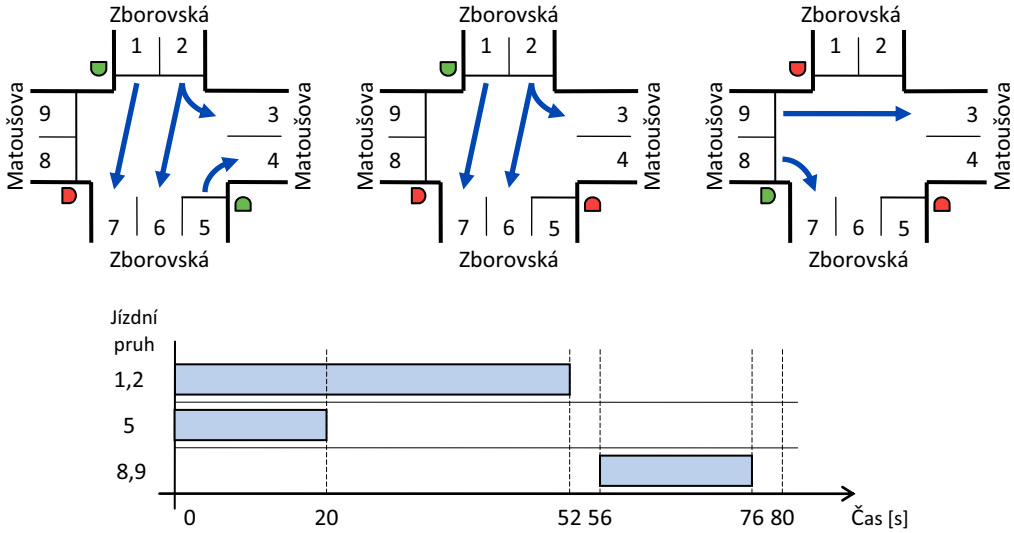


	Zborovská	V Botanice
Zborovská	0,79	0,21
V Botanice	0,23	0,77

Z (ulice)	z (zdroj)	c (cíl)	$\alpha_{Z\to z}$	$\alpha_{z\to c}$	$w_{z\to c}$ [km·h^{-1}]	$t_{z\to c}$ [s]	$d_{z\to c}$ [s·uv^{-1}]	$U_{z\to c}$ [uv·s^{-1}]	$U_{z\to}$ [uv·s^{-1}]	$V_{z\to c}$ [uv·s^{-1}]	$V_{z\to}$ [uv·s^{-1}]
Zborovská	1	7	0.50	0,58	50	20	0,36	2,78	2,01	0,70	0,50
		9		0,42	26	20	0,69	1,45		0,36	
	2	6	0.50	1	50	20	0,36	2,78	2,78	0,70	0,70
V Botanice	3	9	0,43	1	50	52	0,36	2,78	2,78	1,81	1,81
	4	7	0,43	0,16	28	47	0,64	1,56	2,62	0,92	1,50
		8		0,84	50	52	0,36	2,78		1,81	
	5	6	0,14	1	28	47	0,64	1,56	1,56	0,92	0,92

Křižovatka 4: Matoušova – Zborovská

Zeměpisná poloha: 50°4’28.2”N, 14°24’28.2”E

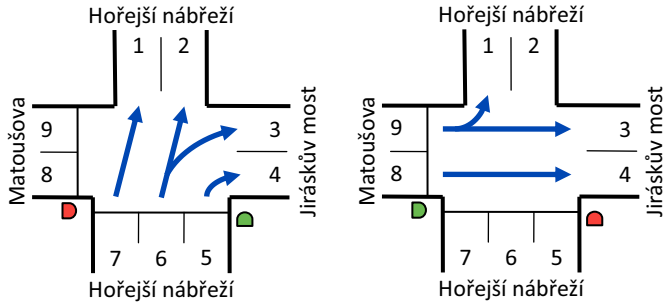


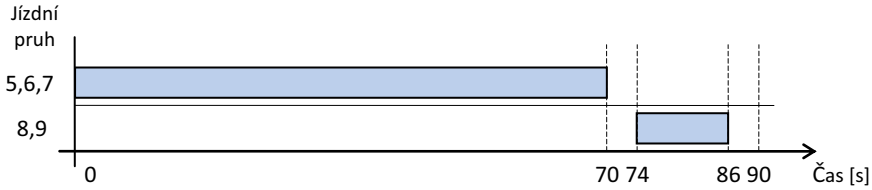
	Zborovská	Matoušova
Zborovská 1,2	0,76	0,24
Matoušova	0,36	0,64

Z (ulice)	z (zdroj)	c (cíl)	$\alpha_{Z\to z}$	$\alpha_{z\to c}$	$w_{z\to c}$ [km·h^{-1}]	$t_{z\to c}$ [s]	$d_{z\to c}$ [s·uv^{-1}]	$U_{z\to c}$ [uv·s^{-1}]	$U_{z\to}$ [uv·s^{-1}]	$V_{z\to c}$ [uv·s^{-1}]	$V_{z\to}$ [uv·s^{-1}]
Zborovská	1	7	0,50	1	50	52	0,36	2,78	2,78	1,81	1,81
	2	3	0,50	0,48	27	52	0,67	1,49	1,96	0,97	1,27
		6		0,52	50	52	0,36	2,78		1,81	
	5	4	1	1	40	20	0,45	2,22	2,22	0,56	0,56
Matoušova	8	7	0,36	1	22	20	0,82	1,22	1,22	0,31	0,31
	9	3	0,64	1	50	20	0,36	2,78	2,78	0,70	0,70

Křižovatka 5: Matoušova – Jiráskův most – Hořejší nábřeží

Zeměpisná poloha: 50°4’29.3”N, 14°24’32.5”E



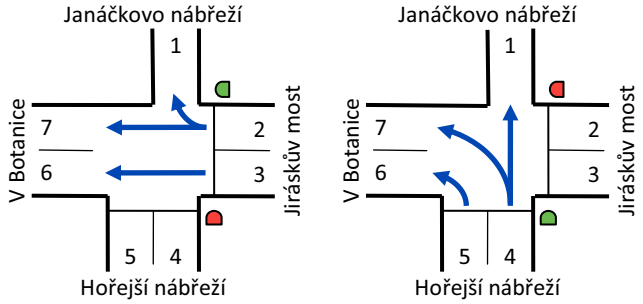


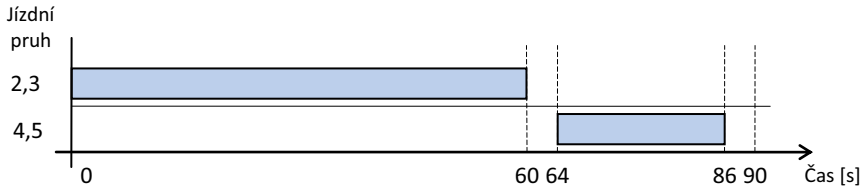
	Hořejší nábřeží	Jiráskův most
Hořejší nábřeží	0,16	0,84
Matoušova	0,17	0,83

Z (ulice)	z (zdroj)	c (cíl)	$\alpha_{Z\to z}$	$\alpha_{z\to c}$	$w_{z\to c}$ [km·h^{-1}]	$t_{z\to c}$ [s]	$d_{z\to c}$ [s·uv^{-1}]	$U_{z\to c}$ [uv·s^{-1}]	$U_{z\to}$ [uv·s^{-1}]	$V_{z\to c}$ [uv·s^{-1}]	$V_{z\to}$ [uv·s^{-1}]
Hořejší nábřeží	5	4	0,12	1	40	70	0,45	2,22	2,22	1,73	1,73
	6	3	0,44	0,09	45	70	0,40	2,50	2,75	1,94	2,14
		2		0,91	50	70	0,36	2,78		2,16	
	7	1	0,44	1	50	70	0,36	2,78	2,78	2,16	2,16
Matoušova	8	4	0,50	1	45	12	0,40	2,50	2,50	0,33	0,33
	9	1	0,50	0,34	35	12	0,51	1,94	2,28	0,26	0,30
		3		0,66	45	12	0,40	2,50		0,33	

Křižovatka 6: V Botanice – Jiráskův most – Janáčkovo náb. – Hořejší náb.

Zeměpisná poloha: 50°4’31.7”N, 14°24’31.4”E





	Janáčkovo nábřeží	V Botanice
Jiráskův most	0,05	0,95
Hořejší nábřeží	0,08	0,92

Z (ulice)	z (zdroj)	c (cíl)	$\alpha_{Z\to z}$	$\alpha_{z\to c}$	$w_{z\to c}$ [km·h^{-1}]	$t_{z\to c}$ [s]	$d_{z\to c}$ [s·uv^{-1}]	$U_{z\to c}$ [uv·s^{-1}]	$U_{z\to}$ [uv·s^{-1}]	$V_{z\to c}$ [uv·s^{-1}]	$V_{z\to}$ [uv·s^{-1}]
Jiráskův most	2	1	0,50	0,10	30	60	0,60	1,67	2,61	1,11	1,73
		7		0,90	50	60	0,36	2,78		1,85	
	3	6	0,50	1	50	60	0,36	2,78	2,78	1,85	1,85
Hořejší nábřeží	4	1	0,50	0,16	50	22	0,36	2,78	2,29	0,68	0,56
		7		0,84	40	22	0,45	2,22		0,54	
	5	6	0,50	1	35	22	0,51	1,94	1,94	0,48	0,48

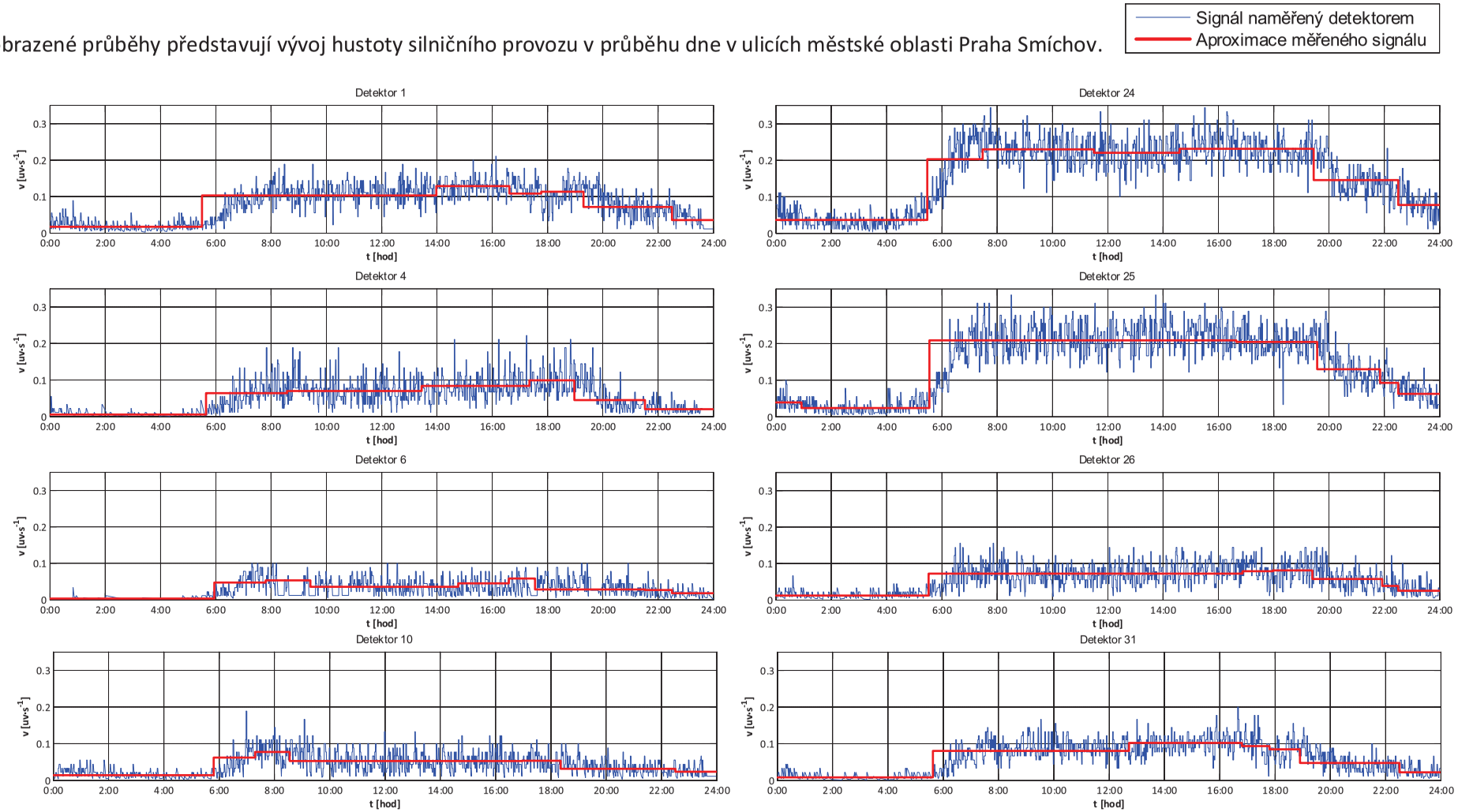
Ulice

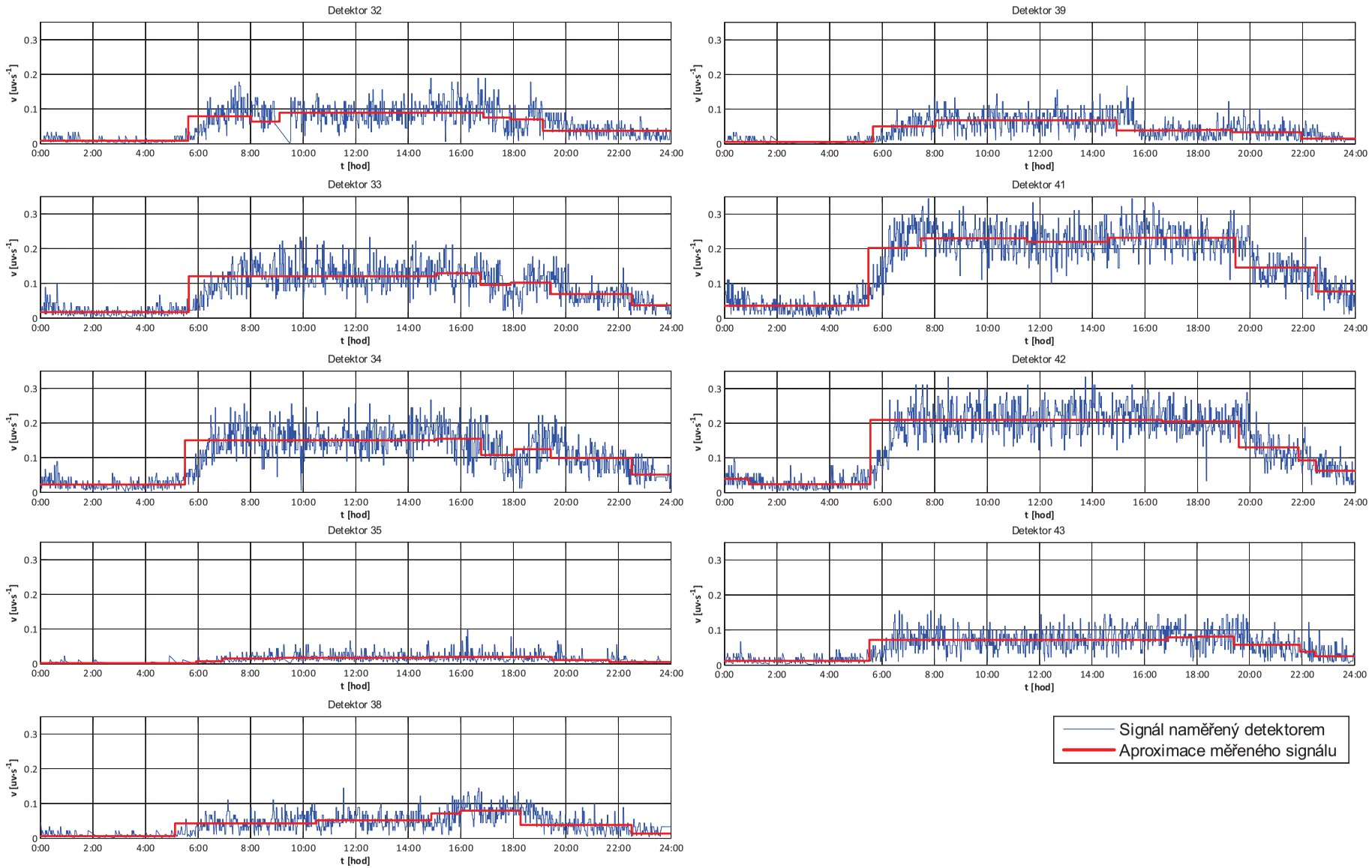
Ulice	z (zdroj)	c (cíl)	$n_{z\to c}$	$l_{z\to c}$ [m]	$w_{z\to c}$ [km·h^{-1}]	$d_{z\to c}$ [s·uv^{-1}]	$C_{z\to c}$ [uv]
Preslova	1	2	1,5	97	50	7,0	29,1
	2	1	1,5	97	50	7,0	29,1
V Botanice	3	2	2,3	108	50	7,8	49,7
	6	3	2,6	83	50	6,0	43,2
Zborovská	3	4	2,0	105	50	7,6	42,0
Matoušova	1	4	2,0	108	50	7,8	43,2
	4	5	2,0	95	50	6,8	38,0
Hořejší nábřeží	5	6	2,0	60	50	4,3	18,0

- z … zdrojová křižovatka
- c … cílová křižovatka
- $n_{z\to c}$ … počet jízdních pruhů

D. Příloha 4 – Vstupní naměřené signály

Zobrazené průběhy představují vývoj hustoty silničního provozu v průběhu dne v ulicích městské oblasti Praha Smíchov.




Detektor 32
Detektor 33
Detektor 34
Detektor 35
Detektor 38
Detektor 39
Detektor 41
Detektor 42
Detektor 43
v [uv·s-1]
t [hod]
Signál naměřený detektorem
Aproximace měřeného signálu