SLOVENSKÁ TECHNICKÁ
UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

Ing. Erik Kučera

Autoreferát dizertačnej práce

MODELOVANIE A RIADENIE HYBRIDNÝCH SYSTÉMOV S VYUŽITÍM PETRIHO SIETÍ VYŠŠÍCH ÚROVNÍ

na získanie akademickej hodnosti doktor (philosophiae doctor, PhD.)

v doktorandskom študijnom programe: **Mechatronické systémy**
v študijnom odbore 5.2.16. mechatronika

Miesto a dátum: Bratislava, 30.5.2016

SLOVENSKÁ TECHNICKÁ UNIVERZITA
V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Ing. Erik Kučera

Autoreferát dizertačnej práce

MODELOVANIE A RIADENIE HYBRIDNÝCH SYSTÉMOV S VYUŽITÍM PETRIHO SIETÍ VYŠŠÍCH ÚROVNÍ

na získanie akademickej hodnosti doktor (philosophiae doctor, PhD.)

v doktorandskom študijnom programe:
Mechatronické systémy

Miesto a dátum: Bratislava, 30.5.2016

Dizertačná práca bola vypracovaná v dennej forme doktorandského štúdia

Na Ústave automobilovej mechatroniky
Fakulta elektrotechniky a informatiky STU v Bratislave (FEI STU BA)

Predkladateľ: Ing. Erik Kučera
Ústav automobilovej mechatroniky
FEI STU BA, Ilkovičova 3, 812 19 Bratislava

Školiteľ: Prof. Ing. Štefan Kozák, PhD.
Ústav automobilovej mechatroniky
FEI STU BA, Ilkovičova 3, 812 19 Bratislava

Oponenti: prof. RNDr. Gabriel Juhás, PhD.
Ústav informatiky a matematiky
FEI STU BA, Ilkovičova 3, 812 19 Bratislava

Ing. Ivana Budinská, PhD.
Ústav informatiky SAV
Slovenská akadémia vied, Dúbravská cesta 9, 845 07 Bratislava

Autoreferát bol rozoslaný: .....

Obhajoba dizertačnej práce sa koná: .....

Na na Fakulte elektrotechniky a informatiky STU v Bratislave, Ilkovičova 3, 812 19 Bratislava, v miestnosti D-124

prof. Dr. Ing. Miloš Oravec

SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program:	Mechatronické systémy
Autor:	Ing. Erik Kučera
Dizertačná práca:	Modelovanie a riadenie hybridných systémov s využitím Petriho sietí vyšších úrovní
Vedúci záverečnej práce:	prof. Ing. Štefan Kozák, PhD.
Miesto a rok predloženia práce:	Bratislava 2016

Hybridné systémy tvoria v súčasnosti význačnú skupinu rôznorodých systémov, ako sú výrobné systémy, systémy v automobilovom priemysle, zdravotníctve, službách a mnohé iné. Tieto systémy sú kombináciou diskrétneho udalostného a spojitého systému. Predložená dizertačná práca sa zaoberá modernou alternatívou modelovania a riadenia takýchto systémov, a to prostredníctvom Petriho sietí vyšších úrovní. Táto problematika je vysoko aktuálna a praxou žiadaná. Petriho siete vyšších úrovní obsahujú silný matematický formalizmus, ktorý znásobuje možnosti využitia modelu. Výsledkom práce je metóda a softvérový modul na podporu modelovania a riadenia udalostných a hybridných systémov pomocou časových Petriho sietí interpretovaných pre riadenie. Predložená metóda riadenia a podporný softvérový modul bol overený na riadení laboratórnych systémov prostredníctvom mikrokontrolérov.

Kľúčové slová: modelovanie, riadenie, automatizácia, Petriho sieť, hybridný systém

ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Programme:	Mechatronic Systems
Author:	Ing. Erik Kučera
Thesis:	Modelling and control of hybrid systems using High-level Petri nets
Supervisor:	prof. Ing. Štefan Kozák, PhD.
Place and year of submission:	Bratislava 2016

Presently, hybrid systems are an important group of heterogeneous systems encountered in manufacturing, automotive, healthcare, service industry, etc. These systems are combination of discrete-event and continuous-time systems. The dissertation thesis deals with a modern approach to modelling and control of such systems using High-level Petri nets. The thesis topic is highly relevant and required by practice. High-level Petri nets include a strong mathematical formalism, which multiplies possibilities of model implementation. The outcomes of the thesis are a methodology and a software module developed to support modelling and control of discrete-event and hybrid systems using timed Petri nets interpreted for control. The developed methodology and software module implemented on microcontrollers have been verified on laboratory plants.

Keywords: modelling, control, automation, Petri net, hybrid system

Obsah

Úvod **6**

1 Súčasný stav problematiky **7**

2 Ciele dizertačnej práce **13**

3 Výsledky riešenia dizertačnej práce **15**

3.1 Softvérový nástroj PN2ARDUINO 15

3.2 Modelovanie a riadenie udalostných systémov 21

3.3 Modelovanie a riadenie hybridných systémov 24

4 Diskusia **33**

Záver **34**

Publikačná činnosť autora **37**

Zoznam použitej literatúry **41**

Úvod

Hybridné systémy tvoria v súčasnosti význačnú skupinu rôznorodých systémov, ako sú výrobné systémy, systémy v automobilovom priemysle, zdravotníctve, službách a mnohé iné. Tieto systémy sú kombináciou diskrétneho udalostného a spojitého systému. Dizertačná práca sa zaoberá modernou alternatívou modelovania a riadenia takýchto systémov, a to prostredníctvom Petriho sietí vyšších úrovní. Výsledkom dizertačnej práce je vypracovanie a overenie metodiky na podporu modelovania a riadenia udalostných a hybridných systémov pomocou pokročilých foriem Petriho sietí. Na tento účel bol vytvorený originálny softvérový modul. Vyvinuté metódy a algoritmy boli implementované na laboratórne systémy, pričom riadenie systémov zabezpečujú mikrokontroléry.

Najdôležitejšia kapitola sa zaoberá rozšírením existujúceho editora Petriho sietí PNEditor. Na rozšírení tohto editora autor práce spolupracoval s autormi diplomových prác, ktoré viedol ako konzultant. Rozšírená softvérová aplikácia dostala názov PN2ARDUINO. Táto aplikácia zabezpečuje komunikáciu nadradenej Petriho siete v PNEditore s mikrokontrolérom cez protokol Firmata. Týmto bola dosiahnutá možnosť riadiť udalostné a hybridné systémy prostredníctvom mikrokontroléra za účasti nadradenej časovej Petriho siete interpretovanej pre riadenie bežiacej v softvéri na počítači. Softvérovo-hardvérová realizácia tejto metodiky je v danej oblasti originálna. Autor predloženej dizertačnej práce tiež navrhol a pomocou vyvinutej aplikácie overil originálnu metódu riadenia hybridných systémov, ktorá je najvýznamnejším vedeckým prínosom autora pre rozvoj vedného odboru.

V závere diskutujeme ďalšie možnosti rozvíjania danej problematiky a je uvedený význam výsledkov dizertačnej práce v príslušnom odbore.

1 Súčasný stav problematiky

Počas zisťovania súčasného stavu problematiky sa autor práce zameral na rešerš výskumných projektov, ktoré sa zaoberajú modelovaním a riadením diskrétnych udalostných a hybridných systémov [8] s využitím hybridných Petriho sietí [9] alebo iných Petriho sietí vyšších úrovní. Ako dôležitý fakt sa bral do úvahy, či sa daný výskum zaoberá len teoretickou stránkou predloženého problému alebo tento výskumný tím používa, prípadne priamo vyvinul, open-source softvérové nástroje na podporu modelovania a riadenia pomocou Petriho sietí vyšších úrovní. Pre praktický výstup rozpracovanej záverečnej práce bolo preto vhodné nájsť také výskumné projekty, ktoré túto podmienku spĺňajú.

Na talianskej univerzite v Bari sa riešia projekty pod vedením Mariegrazie Dotoli a Marie Pia Fanti, ktoré sa zaoberajú využitím hybridných a farebných Petriho sietí v oblasti modelovania a riadenia križovatiek a regulovania dopravy na diaľniciach [10], [14], [11], [26]. Ďalej sú to projekty, ktoré riešia problémy v oblasti výrobných systémov [12], [13], [15], [13], [7]. Uvedené publikácie sú vypracované veľmi kvalitne, nie je však v nich zmienka, či sú dané postupy len teoretickým modelom alebo boli aj odsimulované pomocou softvéru, prípadne nasadené do praxe.

Nemec Dr. Rainer Drath, ktorý v súčasnosti pracuje ako popredný vedecký pracovník v spoločnosti ABB, vyvinul začiatkom tisícročia veľmi dobrý nástroj na podporu hybridných Petriho sietí Visual Object Net++ [16]. Existuje viacero publikácií, ktoré veľmi názorne demonštrujú možnosti tohto nástroja, najmä od rumunského autora Drighiciu [19], [17], [18], [21], [22], [20]. Tieto články sú zároveň veľmi vhodné ako vstupná brána do štúdia výhod hybridných Petriho sietí pri modelovaní a riadení hybridných systémov. Zdrojový kód softvérovej aplikácie autor odmietol poskytnúť. Jej ďalšie rozvíjanie nie je teda možné.

Výborný teoretický podklad pre prediktívne riadenie systémov s vy-

užitím Petriho sietí ponúkajú publikácie od autorov Guia, Julvez a Mahulea [30] a [28].

Autormi SW nástroja Snoopy [40] sú nemeckí vedeckí pracovníci M. Heiner, M. Herajy, F. Liu, C. Rohr a M. Schwarick. Táto softvérová aplikácia ponúka prácu s veľkým množstvom tried Petriho sietí, ako sú stochastické, farebné, hybridné, ba dokonca "hudobné" (music) Petriho siete. Pomocou nástroja Snoopy sú v súčasnosti riešené najmä úlohy z oblasti biológie a chémie. Bohužiaľ, zdrojový kód aplikácie odmietli autori poskytnúť, nakoľko podľa ich názoru "je jeho kompilácia príliš zložitá".

Ako zaujímavá alternatíva sa javilo využitie jazyka Modelica a open-source prostredia OpenModelica. V tomto prostredí existuje knižnica, ktorá podporuje modelovanie pomocou Petriho sietí. Výhodou prostredia OpenModelica je, že by sa PN model dal prepájať s ďalšími súčasťami Modelicy. Prvý PN toolbox bol predstavený v [31]. Na tento toolbox existuje nadväzujúca práca [25]. Väčšiemu rozšíreniu tohto toolboxu sa venovala nemecká autorka Sabrina Pross, ktorá ho obohatila pre prácu s rozšírenými hybridnými Petriho sieťami pre potreby modelovania procesov v živých organizmoch [37], [36], [5], [33], [34], [38], [35]. Práca s touto knižnicou a jej ďalšie rozširovanie sa zdalo ako perspektívne. Napriek tomu, že štúdiu Modelicy a danému toolboxu venoval autor predloženej práce pomerne dlhý čas, k významnejším výsledkom nedospel. Toolbox bol primárne napísaný pre komerčný nástroj Dymola a nie pre open-source prostredie OpenModelica. V priebehu roka 2015 však vyššie uvedený výskumný tím (teraz pravdepodobne pod vedením Lennarta Ochela) uverejnil na serveri github.com [29] upravenú verziu knižnice PNlib, ktorá z veľkej časti fungovala aj v OpenModelice.

Využitím PNlibu v OpenModelice na modelovanie hybridných systémov sa zaoberala Miriama Nižnanská vo svojej diplomovej práci [32], ktorej konzultantom bol autor predloženej dizertačnej práce. V rámci predmetnej diplomovej práce vznikla aj unikátna vývojárska príručka pri editáciu a pridávanie modulov do OpenModelica. Existujúce návody

sa totiž ukázali ako nekompletné.

Knižnica PNlib síce ponúka v softvéri OpenModelica výborné možnosti pre modelovanie hybridných systémov, avšak prepojiť tieto modely s reálnym hardvérom (mikrokontrolérom za pomoci protokolu Firmata) a na základe modelu ho riadiť sa ukázalo ako prakticky nemožné. OpenModelice totiž chýba podpora komunikácie so sériovým portom. Túto funkcionalitu však ponúka Wolfram SystemModeler.

Wolfram SystemModeler je jednoducho použiteľný modelovací a simulačný program pre kyber-fyzikálne systémy [4]. Je taktiež založený na jazyku Modelica. Oproti OpenModelica ide o proprietárny softvér. Ide o súčasť balíka Mathematica, avšak dá sa používať aj samostatne.

Rovnako ako OpenModelica, tak aj Wolfram SystemModeler ponúka možnosť modelovania hybridných systémov. Možno uviesť ako príklad systém z oficiálnej stránky produktu [1]. Ide o model katupultovacieho systému lietadla. Pre namodelovanie diskrétnej udalostnej logiky je použitá knižnica StateGraph. Ide v podstate o zjednodušenú Petriho sieť, ktorá obsahuje len jeden token, alebo o variant konečného automatu (stavový graf). Klasické alebo Petriho siete vyšších úrovní Wolfram SystemModeler neobsahuje.

Obrovskou výhodou prostredia Wolfram SystemModeler je podpora komunikácie so sériovým portom. Existuje voľne dostupné rozšírenie ModelPlug, ktoré zabezpečuje prepojenie simulácie s mikrokontrolérmi, ktoré podporujú protokol Firmata [41]. Firmata je protokol, ktorý je navrhnutý na zabezpečenie komunikácie medzi mikrokontrolérom a počítačom, prípadne mobilným zariadením [3] [27]. Samotný protokol je možné implementovať vo firmvéri mikrokontroléra akejkoľvek architektúry. Takisto môže byť súčasťou ľubovoľného softvérového balíka pre počítače. Firmata je najčastejšie využívaná práve pre mikrokontroléry platformy Arduino, kde ponúka komplexnú podporu vrátane mnohých knižníc, ktoré podporujú periférie typu servomotor, krokový motor alebo komunikáciu prostredníctvom I^2C. Podpora zo strany počítača je dnes veľmi bohatá, nakoľko sú vytvorené knižnice v rôznych jazykoch.

Z pohľadu konektivity sa teda Wolfram SystemModeler v spolupráci s knižnicou ModelPlug zdal ako ideálny kandidát pre modelovanie a riadenie hybridných systémov pomocou Petriho sietí vyšších úrovni. Problémom však je, že SystemModeler podporuje len spomínaný StateGraph, ktorý nemožno označiť za Petriho sieť vyššej úrovne.

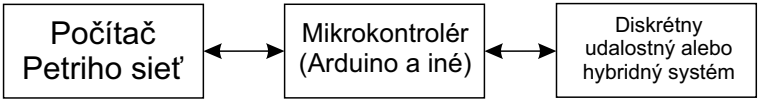
Riešením by teda mohla byť integrácia knižnice PNlib do Wolfram SystemModelera. Pokusy o integráciu prebiehali v rámci riešenia tejto dizertačnej práce a diplomovej práce [23], ktorej konzultantom bol autor predloženej dizertačnej práce. Prebiehala tiež spolupráca s tímom Lennarta Ochela, ktorý PNlib vyvinul, a konzultácie na fórach Wolfram SystemModelera.

Aj keď je PNlib napísaný v jazyku Modelica, táto knižnica nie je priamo kompatibilná so SystemModelerom. Dôvodov je viac. Interný prekladač používaný SystemModelerom pri prekladaní kódu z jazyka Modelica do C++ nepodporuje niektoré typy operátorov (z unárnych je to *not* a z binárnych operátorov *and* a *or*) na prácu s poľami. Táto chyba bola autorom diplomovej práce [23] opravená a následne autormi PNlibu zakomponovaná do oficiálnej verzie knižnice, ktorá je dostupná na GitHube [29].

Niektoré ďalšie chyby nebolo možné jednoduchým spôsobom opraviť zmenou kódu v PNlibe. Tieto chyby boli nahlásané priamo autorom softvéru Wolfram SystemModeler. Keďže tento softvér nie je typu opensource a nové verzie vychádzajú cca každého pol roka, z hľadiska času vyhradeného na riešenie dizertačnej práce nebolo možné čakať na nápravu v SystemModelere, ktorá aj tak nemusí byť dostačujúca pre úspešnú integráciu PNlibu do SystemModelera. Pre budúcich výskumníkov v tejto oblasti je však užitočná informácia, že časom by integrácia PNlibu mala byť možná.

Vyhodnotenie súčasného stavu problematiky

Hlavnou témou prezentovanej dizertačnej práce je riadenie udalostných a hybridných systémov prostredníctvom mikrokontrolérov za účasti Petriho siete vyššej úrovne bežiacej v softvéri na počítači. Jednoduchú schému možno vidieť na obrázku 1.



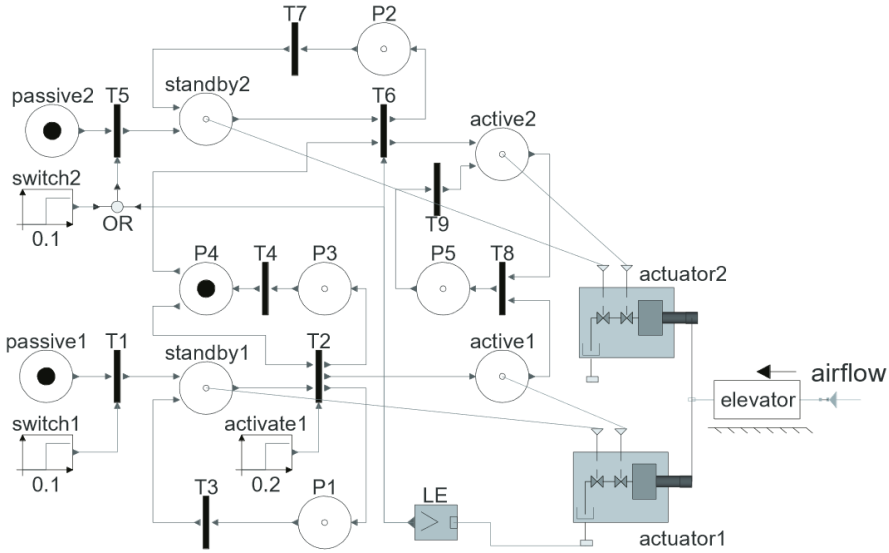
Obrázok 1: Jednoduchá schéma vízie riešenia

Túto predstavu veľmi dobre ilustruje obrázok 2. Na tomto obrázku možno vidieť model riadenia pohybu výťahu (technické detaily fungovania výťahového systému v tomto prípade nie sú podstatné). Táto sieť je doplnená ďalšími prvkami. Všimnime si napríklad prvok pre vstup signálu `SWITCH1`. Prechod T_1 je aktivovaný v tom momente, keď je aktivovaný `SWITCH1`. Toto môže byť reprezentované tlačidlom pripojeným na Arduino. Odpálením prechodu T_1 bude na mieste `STANDBY1` prítomný token. Je vidieť, že toto miesto je pripojené na aktuátor (napr. motor) `ACTUATOR1`. Prítomnosťou tokenu sa teda motor výťahu spustí. Docieliť porovnateľnú logiku bolo cieľom dizertačnej práce. Podľa umiestnenia tokenu (teda podľa stavu systému) by tiež bolo možné nastavovať rôzne parametre pre PID regulátor realizovaný Arduinom, čím získame jednoduchý switched / hybridný systém.

Obrázok 2 pochádza z článku z roku 1998, kedy vznikol prvý toolbox s Petriho sieťami pre jazyk Modelica. Pár rokov na to vznikli rôzne nadstavby, vrátane spomínaného PNlibu, a táto pôvodná knižnica už nebola ďalej podporovaná v novších verziách OpenModelicy.

Nakoľko sa v rámci skúmania súčasného stavu problematiky zistilo, že neexistuje softvérové riešenie, ktoré by zabezpečovalo podporu riadenia udalostných a hybridných systémov pomocou mikrokontroléra za účasti online bežiacej Petriho siete vyššej úrovne na počítači, bolo pot-

rebné takéto riešenie vyvinúť. Ako základ pre takýto softvér bol vybraný editor Petriho sietí s názvom PNEditor [39]. Tento softvér bol vyvinutý na Fakulte elektrotechniky a informatiky STU v rámci záverečnej práce Martina Riesza a je dobre rozšíriteľný.



Obrázok 2: Petriho sieť interpretovaná pre riadenie - koncept z roku 1998 [31]

2 Ciele dizertačnej práce

Rozvoj metód riadenia hybridných systémov patrí k moderným smerom v oblasti automatizácie a mechatroniky. Na základe analýzy dostupnej literatúry a najnovších výskumných projektov na Slovensku a vo svete bolo zistené, že metódy riadenia založené na využití Petriho sietí vyšších úrovní existujú len ako teoretický koncept. Doposiaľ teda nejestvuje softvérová aplikácia alebo prostredie, ktoré by umožňovalo modelovanie a riadenie udalostných a zároveň hybridných systémov za podpory Petriho sietí vyšších úrovní pomocou mikrokontrolérov. Dostupnosť a ich neustále sa zväčšujúci výkon ich predurčuje využiť vo výskume v predmetnej oblasti, ale aj v priemyselnej praxi.

Hybridný systém je chápaný ako kombinácia spojitého a diskrétneho udalostného systému. Riadenie hybridných systémov prináša nové výzvy, nakoľko spája metódy riadenia udalostných systémov, kde dokáže byť formalizmus Petriho siete nápomocný, s metódami riadenia spojitého systému, ktoré vychádzajú z klasickej teórie riadenia. Správnym návrhom metodiky a softvérového modulu je možné tieto prístupy synergicky kombinovať. Získame tak funkčný a originálny riadiaci systém, ktorý dovolí zosúladiť metódy riadenia udalostného systému s metódami riadenia spojitého systému (napríklad pomocou PID regulátora), pričom efektívna kooperácia týchto prístupov umožní riadiť hybridný systém.

Spomínaná metóda je napríklad využiteľná pri systémoch, kde je potrebné na základe rozdielnych stavov istého subsystému využiť rozdielne algoritmy riadenia spojitého subsystému (napr. PID regulátory s rozdielnymi parametrami). Sila tejto metódy tkvie v tom, že koncept Petriho sietí dokáže takéto riadiace pravidlá súvisiace so stavom systému pokryť veľmi efektívnym, robustným a zároveň prehľadným (grafickým) spôsobom. Ako príklad využitia možno uviesť riadenie motora vrtuľníka, pričom v závislosti od výšky, v ktorej sa nachádza, je potrebné

prepínať medzi rôznymi režimami tohto riadenia.

Ciele práce možno zhrnúť do nasledovných bodov:

1. Výber hardvérových a komunikačných prostriedkov pre realizáciu a overenie navrhnutej metódy
   - Ako vhodný komunikačný prostriedok (protokol) pre účely práce sa ukázal protokol Firmata, ktorý je kompatibilný s mnohými mikrokontrolérmi.
   - Platforma Arduino ponúka mikrokontroléry postavené na procesoroch od spoločnosti Atmel, ktorá dodáva procesory aj pre priemyselné aplikácie napríklad v sfére inteligentných domácností a v niektorých jednoduchších systémoch v automobile. Tento fakt a kompatibilita s protokolom Firmata potvrdzuje, že výber platformy Arduino je z hľadiska riešenia perspektívny.
2. Implementácia softvérového modulu pre programovú realizáciu navrhnutej metódy do vybraného prostredia podporujúceho Petriho siete
   - V rámci analýzy sa ako vhodné prostredie ukázal editor Petriho sietí PNEditor, ktorý je rozšíriteľný o nové moduly.
3. Overenie metódy riadenia na laboratórnych modeloch udalostných a hybridných systémov pomocou mikrokontrolérov
   - Navrhnuté metódy a softvérový modul by mali byť otestované na laboratórnych udalostných a hybridných systémov.
   - Pre realizáciu spojitej časti laboratórneho hybridného systému bol vybraný jednosmerný motor s enkóderom pre zabezpečenie spätnej väzby pri spojitej zložke riadenia.
4. Diskusia o výsledkoch práce a ďalšom využití metód modelovania a riadenia hybridných systémov vo výskume a priemyselnej praxi

3 Výsledky riešenia dizertačnej práce

V tejto časti sú opísané praktické výstupy a výsledky dizertačnej práce.

Úvodom sa zaoberá opisom softvérového nástroja PN2ARDUINO a jeho využitím na riadenie udalostných a hybridných systémov. Softvér PN2ARDUINO bol vyvinutý v spolupráci s riešiteľkou diplomovej práce [24], ktorej konzultantom bol autor predloženej dizertačnej práce.

Ďalšie podkapitoly zaoberajúce sa riadením systémov predstavujú originálnu metódu modelovania a riadenia hybridných systémov, ktorá je najväčším vedeckým prínosom tejto práce pre rozvoj vedného odboru.

3.1 Softvérový nástroj PN2ARDUINO

Existuje viacero konceptov riadenia prostredníctvom Petriho sietí. Petriho sieť ako riadiacu logiku je treba prepojiť s riadeným systémom, a to napríklad prostredníctvom mikrokontroléra. Podstatným bodov návrhu riadiaceho systému je otázka, či umiestniť logiku Petriho siete priamo do mikrokontroléra [6] alebo do počítača, ktorý by následne s mikrokontrolérom komunikoval.

Obidva prístupy majú svoje výhody a nevýhody. Ak je logika Petriho sietí umiestnená do mikrokontroléra, hlavnou výhodou je nezávislosť riadiacej jednotky od softvérovej aplikácie, resp. programu na počítači. Výhodou je tiež schopnosť reagovať na podnety z vonkajšieho prostredia v reálnom čase. Medzi nevýhody možno zaradiť obmedzené výpočtové a pamäťové zdroje mikrokontroléra. Ďalšou markantnou nevýhodou (najmä počas ladenia návrhu riadiaceho systému) je nutnosť opakovaného kompilovania a nahrávania programu do mikrokontroléra.

Ak je logika Petriho siete umiestnená do špecializovanej softvérovej aplikácie na počítači, toto riešenie prináša možnosť riadiť systém priamo z neho. Toto odstraňuje nutnosť prekompilovávania a nahrávania programu do mikrokontroléra počas ladiacej fázy návrhu. Ďalšou z výhod

je odstránenie obmedzenia výpočtových a pamäťových zdrojov, keďže v porovnaní s mikrokontrolérom má bežný počítač tieto zdroje prakticky neobmedzené. Jednou z nevýhod konceptu však je, že nedokážeme zaručiť reakcie na podnety v reálnom čase.

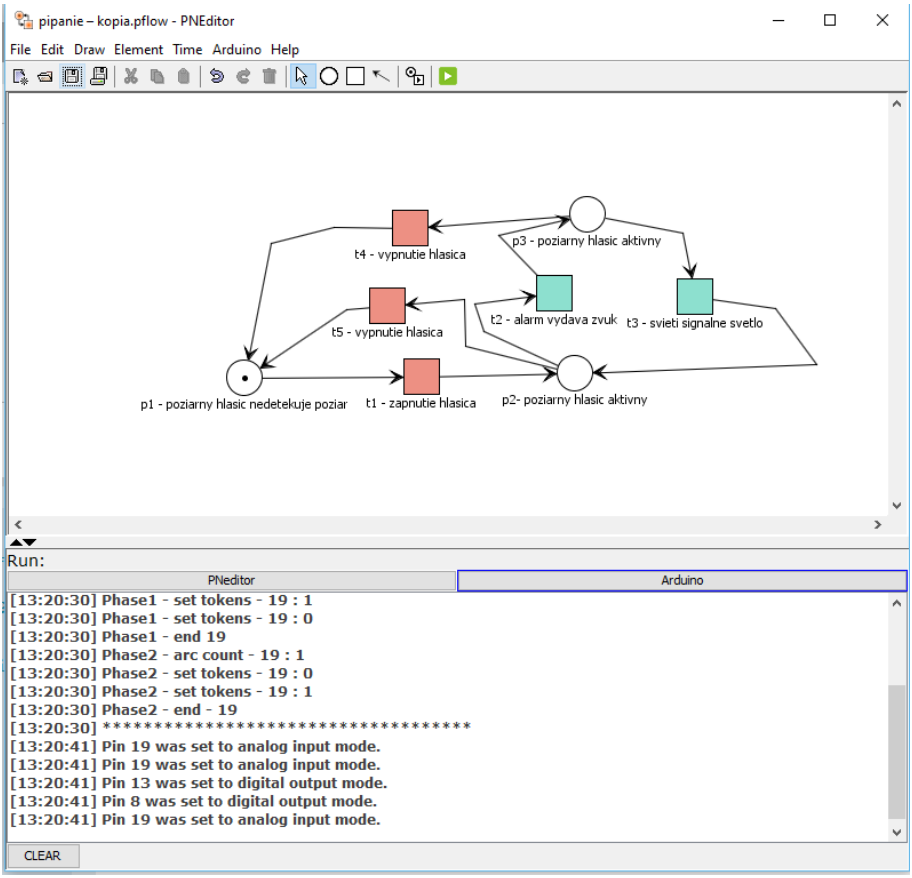
Nová softvérová aplikácia PN2ARDUINO bola navrhnutá na základe druhého konceptu. Nadradená riadiaca Petriho sieť teda beží na počítači. Ako základ tejto aplikácie bol použitý PNEditor [39]. Ide o grafický editor pre modelovanie P/T Petriho sietí. Je napísaný v programovacom jazyku Java a ide o open-source program. Jeho výhodou je prehľadný návrh a dobre štruktúrovaný kód.

Pre komunikáciu medzi softvérovou aplikáciou a mikrokontrolérom bol použitý protokol Firmata. Na strane Arduina ide o verziu 2.3.2 (Standard Firmata). Na strane Java aplikácie bola použitá klientská knižnica s názvom Firmata4j (verzia 2.3.3). Výhodou takéhoto návrhu je, že namiesto Arduina možno použiť akýkoľvek mikrokontrolér podporujúci protokol Firmata.

PN2ARDUINO dopĺňa PNEditor o viaceré funkcie. Na strane Petriho sietí je to hlavne pridanie možnosti oneskorenia na prechody a doplnenie vlastnosti kapacity pre miesta. Bol tiež implementovaný automatický režim spúšťania prechodov práve pre potreby riadenia, nakoľko pôvodný PNEditor obsahoval len manuálny režim.

Aplikácia do PNEditora prináša nový modul, ktorý zodpovedá za komunikáciu s kompatibilným mikrokontrolérom. Je zložený z dvoch základných časti. Prvá z nich zodpovedá za vytvorenie spojenia s mikrokontrolérom, čo zahŕňa nastavenie komunikačného COM portu, na ktorom sa mikrokontrolér nachádza. Druhá časť zabezpečuje implementáciu možnosti pridania Arduino komponentu na prechod alebo miesto Petriho siete. Podporované sú tieto typy Arduino komponentov: digitálny vstup a výstup, analógový vstup, ovládanie servomotora, PWM výstup, posielanie správ, posielanie vlastných správ typu SYSEX.

Uvádzame krátky opis SYSEX telegramov [27]. V prípade telegramu SYSEX sa zanedbáva 3-bytový tvar Firmata telegramu a vytvára sa tele-



Obrázok 3: Prostredie softvérovej aplikácie PN2ARDUINO

gram, ktorý ma variabilnú dĺžku, určený napríklad na prenos reťazcov. Potom takýto telegram začína riadiacim bytom SYSEX START (s tvarom 0xF0). Nasleduje subriadiaci byte, ktorý určí typ správy, a to napríklad 0x71 pre prenos znakov reťazca. Ďalšie byty budú vždy obsahovať 14-bitové znaky až pokiaľ nepríde byte, ktorý ma hodnotu 0xF7. Tento sa nazýva SYSEX END (koniec telegramu). Subriadiacich príkazov je veľké množstvo. Slúžia na bližšiu špecifikáciu správ SYSEX. Je tu tiež možnosť definície vlastných príkazov. Pre tieto účely sú rezervované subriadiace príkazy s označením 0x00 – 0x0F. V týchto prípadoch je možné na strane mikrokontroléra definovať vlastnú funkcionalitu, ktorá sa bude spúšťať práve pomocou týchto subriadiacich príkazov. Tento fakt bol počas riešenia dizertačnej práce hojne využívaný a je jeden z pilierov predloženej metodiky riadenia udalostných a hybridných systémov a súvisiacej softvérovej aplikácie.

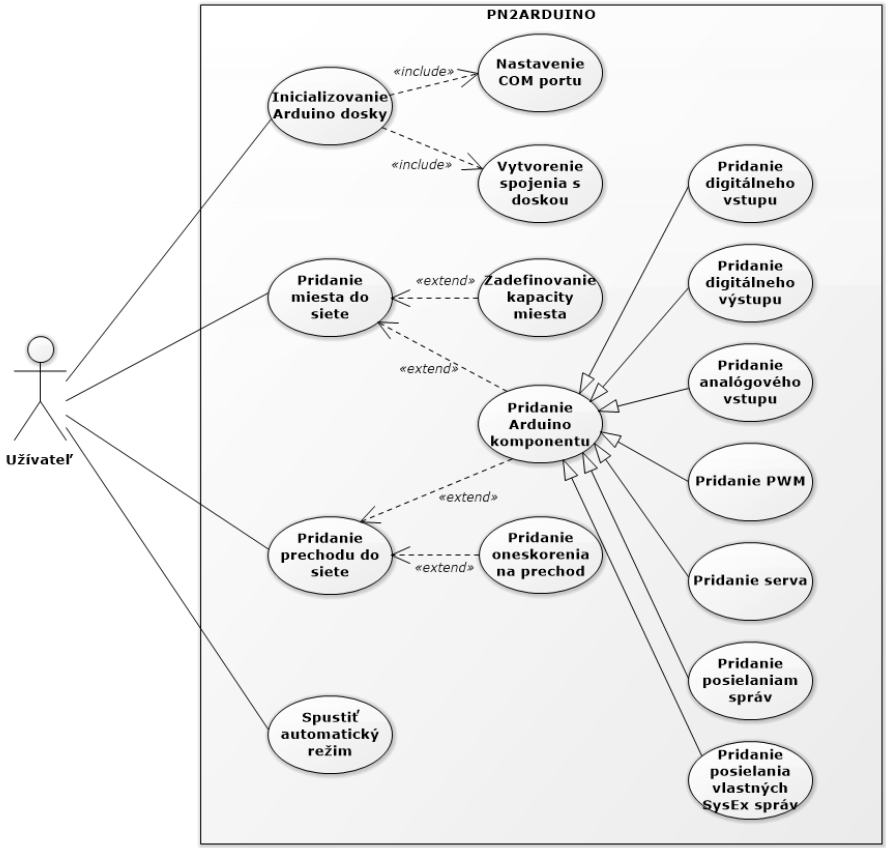
Ako už bolo uvedené, miesta a prechody v Petriho sieti je možné asociovať s Arduino komponentom. Digitálne a analógové vstupy plnia úlohu podmienky spustiteľnosti prechodu, pričom je samozrejmosťou, že musia byť splnené klasické podmienky spustenia prechodu v Petriho sieti. Digitálne a PWM výstupy a správy na sériový portu sú vykonávateľmi príslušných akcií.

Zaujímavou funkcionalitou je možnosť posielania vlastných SYSEX správ. Vyžaduje sa tak zadanie používateľského SYSEX príkazu (v rozsahu 0x00 – 0x0F) a voliteľne aj obsahu správy. Správa sa odošle vtedy, keď príde token do miesta, resp. keď sa spustí prechod.

Namodelovaná Petriho sieť sa ukladá do súboru s príponou `.pflow`. Detaily implementácie je možné nájsť v práci [24].

Hlavné okno softvérovej aplikácie PN2ARDUINO obsahuje menu rýchleho prístupu, hlavné menu, kresliace plátno pre Petriho sieť a logovaciu konzolu (obrázok 3).

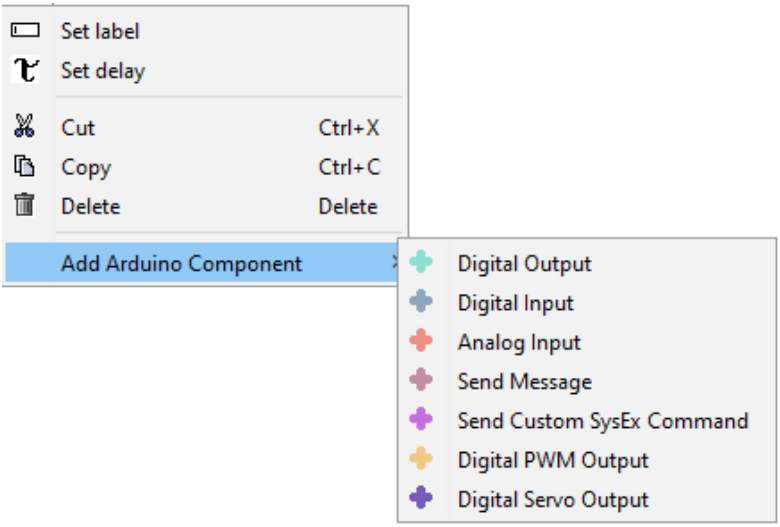
PN2ARDUINO podporuje dva režimy, a to režim návrhu a režim riadenia. Režim návrhu slúži na umiestňovanie jednotlivých elementov Petriho siete. Režim riadenia slúži na spúšťanie prechodov a sledovanie



Obrázok 4: Diagram prípadov použitia [24]

stavu siete. Pre spúšťanie prechodov tu existujú dva módy - ručný (implementovaný už v rámci základného PNEditora) a automatický. Automatický režim bol doplnený pre možnosti automatického riadenia systému.

Aby sa v aplikácií povolila možnosť pridávania Arduino komponentov k miestam alebo prechodom, je treba inicializovať komunikáciu s Arduinom (akcia *Setup board* v menu). Následne je možné po kliknutí na prechod (alebo miesto) vybrať položku *Add Arduino Component* - obrázok 5.



Obrázok 5: Pridávanie Arduino komponentu

Ďalšou pridanou funkcionalitou je časová politika. K prechodu je teda možné pridať časové oneskorenie, ktoré môže byť *deterministické* alebo *stochastické*.

3.2 Modelovanie a riadenie udalostných systémov

Demonštrácia a overenie metódy riadenia udalostných systémov pomocou nového softvérového modulu PN2ARDUINO bola realizovaná na navrhnutom laboratórnom systéme požiarneho hlásiča s mikrokontrolérom Arduino Uno. Schému zapojenia je možné vidieť na obrázku 6.

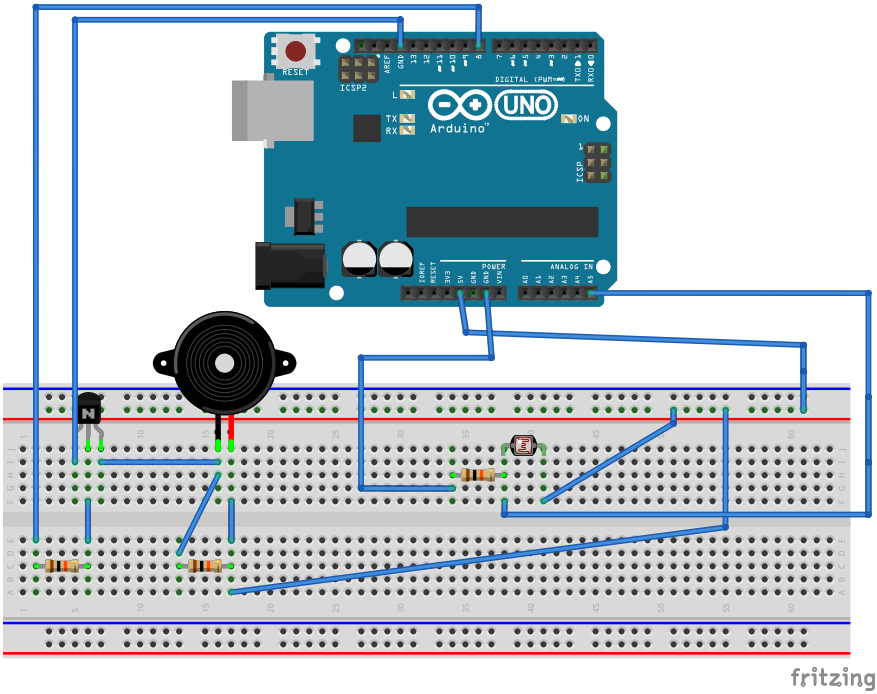
Tento systém sa skladá zo bzučiaka (*active buzzer*), fotorezistora, 3 rezistorov a NPN tranzistora. NPN tranzistor je potrebný pre zapojenie bzučiaka. Používa sa tiež vstavaná LED dióda na Arduino Uno, ktorá je umiestnená na pine 13. Fotorezistor bol namiesto dymového senzora použitý pre lepšiu realizovateľnosť experimentu.

Následne je treba zadefinovať želané správanie systému. Ak fotorezistor zaznamená nadmerné osvetlenie (bolo experimentálne určené ako vstupná hodnota väčšia ako 799 na analógovom pine Arduino Uno, pričom ten rozlišuje hodnoty od 0 po 1023), tak sa spustí prerušovaný tón bzučiaka, ktorý sa strieda so zapínaním LED diódy. Keď hodnota na vstupnom analógovom pine klesne pod 800, tak sa tieto zvukové a svetelné efekty zastavia. Toto sa cyklicky opakuje.

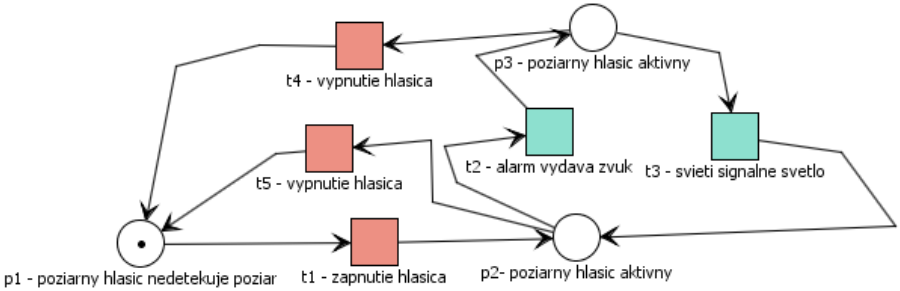
Počiatočné označkovanie časovej Petriho siete interpretovanej pre riadenie v PN2ARDUINO je na obrázku 7. Je zjavné, že token sa nachádza v mieste p_1, ktoré označuje stav, kedy nie je hlásič aktívny, keďže fotorezistor nezaznamenal potrebnú prahovú hodnotu svetelnej intenzity.

V momente, keď sa na analógovom pine Arduina nameria hodnota vyššia ako 799, spustí sa prechod t_1. Na tomto prechode je namapovaný Arduino komponent *Analog Input*, kde je nastavený rozsah vstupných hodnot, kedy je prechod spustiteľný (800-1023).

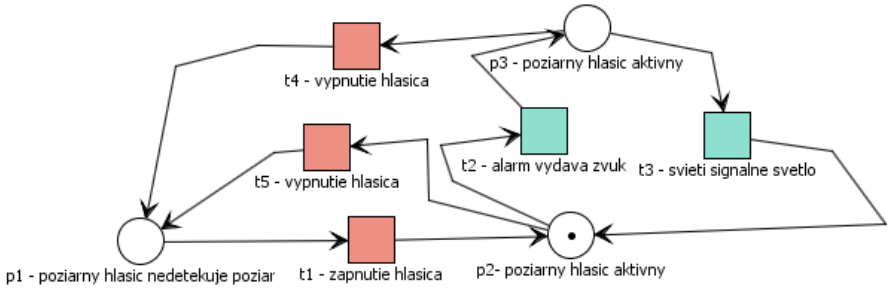
Token je teraz v mieste p_2. Prechod t_2 má namapovaný Arduino komponent *Digital Output*, v tomto prípade na pin 8, kde je pripojený bzučiak. Tento prechod má tiež priradenú funkcionalitu *time delay* - 2 sekundy. Znamená to, že spustenie prechodu a zvukový efekt bzučiaka bude trvať 2 sekundy.



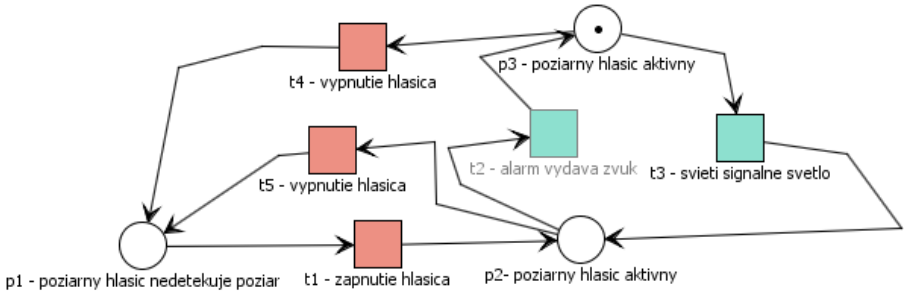
Obrázok 6: Schéma zapojenia laboratórneho modelu požiarneho hlásiča



Obrázok 7: PN pre požiarny hlásič (počiatočné označkovanie)



Obrázok 8: PN pre požiarny hlásič (spustenie prechodu t_1)



Obrázok 9: PN pre požiarny hlásič (spustenie prechodu t_2)

Následne je token na mieste p_3. Prechod t_3 má namapovaný Arduino komponent *Digital Output*, v tomto prípade na pin 13, na ktorý je pripojená vstavaná LED dióda Arduina. *Time delay* je nastavený na 1 sekundu. LED dióda teda zasvieti na 1 sekundu.

Toto sa cyklicky opakuje, kým sa vstupná hodnota na analógovom pine nezníži pod hodnotu 800. Vtedy sa spustí prechod t_4 alebo t_5 a token sa dostane do miesta p_1, kedy hlásič nedetekuje žiaden požiar.

3.3 Modelovanie a riadenie hybridných systémov

V tejto podkapitole je predstavený hlavný výsledok práce pre rozvoj vedného odboru.

Opis systému

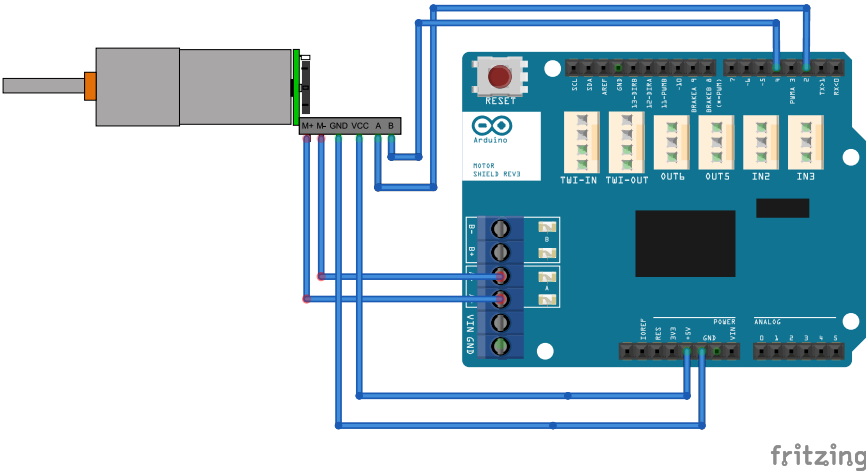
Pre overenie navrhovanej metódy riadenia hybridných systémov bolo potrebné navrhnúť laboratórny fyzikálny model takéhoto systému. Bol vybraný jednosmerný motorček s enkóderom. Enkóder je potrebný pre zabezpečenie spätnej väzby v systéme, nakoľko slúži na meranie otáčok motora, ktoré považujeme za sledovanú (referenčnú) veličinu. Parametre motorčeka dodané výrobcom sú uvedené v tabuľke 1.

Na zapojenie motorčeka k Arduino Uno bol použitý modul *motor shield*, ktorý umožňuje zapojiť motorček jednoduchým spôsobom bez použitia tranzistorov a ďalšej mierne pokročilej elektroniky. Arduino *motor shield* je založený na mostíkovom budiči L298, ktorý zabezpečuje zvládnutie indukčnej záťaže spôsobenej súčiastkami, ako je relé, solenoidy alebo motorčeky. Pomocou tohto *motor shieldu* možno nezávisle riadiť rýchlosť a smer otáčania dvoch jednosmerných motorov. Schému zapojenia motorčeka k Arduino Uno s *motor shieldom* možno vidieť na obrázku 10.

Enkóder v používanom motorčeku je inkrementálneho typu. Enkóder počas rotácie motora počíta počet *tikov*, z čoho sa dá vyvodiť rýchlosť

Pracovné podmienky	
Menovité napätie	6.0V (DC)
Rozsah teplôt	$-20°C$ ~$+60°C$
Rozsah vlhkosti	$0\% - 90\%$
Charakteristika v nezaťaženom stave	
Prúd v nezaťaženom stave	$\leq 200mA$
Rýchlosť v nezaťaženom stave	$185 \pm 10\%$ot./min
Charakteristika v zaťaženom stave	
Menovitá záťaž	$0.0883N.m$
Menovitý prúd	$\leq 550mA$
Menovitá rýchlosť	$135 \pm 10\%$ot./min
Počiatočný krútiaci moment	$0.4413N.m$
Prúd pri nulových otáčkach	$\geq 2.0A$

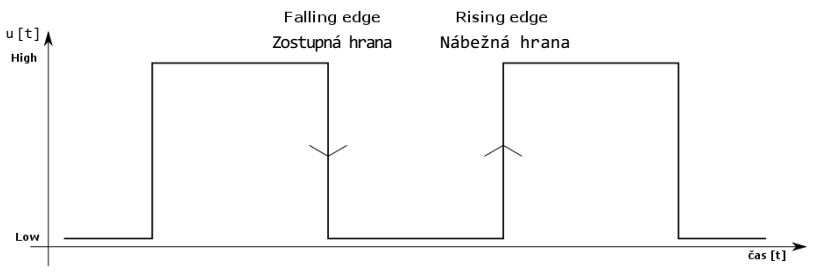
Tabuľka 1: Parametre motorčeka



Obrázok 10: Schéma zapojenia motora na Arduino Uno s motor shieldom

(otáčky) motora.

Pre meranie otáčok motorčeka pomocou enkódera je potrebné použiť funkcionalitu *hardvérového prerušenia* [42]. Pre realizáciu tejto vlastnosti, musí mať procesor možnosť hardvérového prerušenia zabudovanú.



Obrázok 11: Zostupná a nábežná hrana signálu [42]

Arduino Uno umožňuje dve externé prerušenia. Sú očíslované 0 a 1 a sú dostupné na digitálnych pinoch 2 a 3. Na týchto pinoch možno sledovať celkom 4 rôzne druhy udalostí:

- LOW - prerušenie nastane vtedy, keď je pin v logickej nule.
- CHANGE - prerušenie nastane pri zmene logickej hodnoty na danom pine.
- RISING - prerušenie s príchodom nábežnej hrany.
- FALLING - prerušenie s príchodom zostupnej hrany.

Pri využití enkóderu na meranie otáčok sa použilo prerušenie s príchodom nábežnej hrany. Ako je možné vidieť na obrázku 10, výstup enkódera A je zapojený na pin 2, ktorý podporuje prerušenia. Výstup enkódera B by bolo možné zapojiť na pin 3, ktorý tiež podporuje prerušenia, avšak na *motor shielde* je tento pin vyhradený na ovládanie napätia na motorčeku (`PWM A`). Výstup enkódera B je teda zapojený na pin 4. Toto

nie je prekážkou, nakoľko pri realizovanom experimente je treba merať otáčky motora, ktorý sa bude otáčať len jedným smerom. Stačí teda použiť prerušenie len pre jeden výstup enkódera.

Aktuálne otáčky motora sú v programe pre Arudino uložené v premennej `vel`. Hodnota premennej `vel` vyjadruje desaťnásobok aktuálnych otáčok motora za minútu. Pri vstupe `PWM = 255` dostávame hodnotu premennej približne 1860, čo korešponduje s údajmi výrobcu (185 otáčok za minútu).

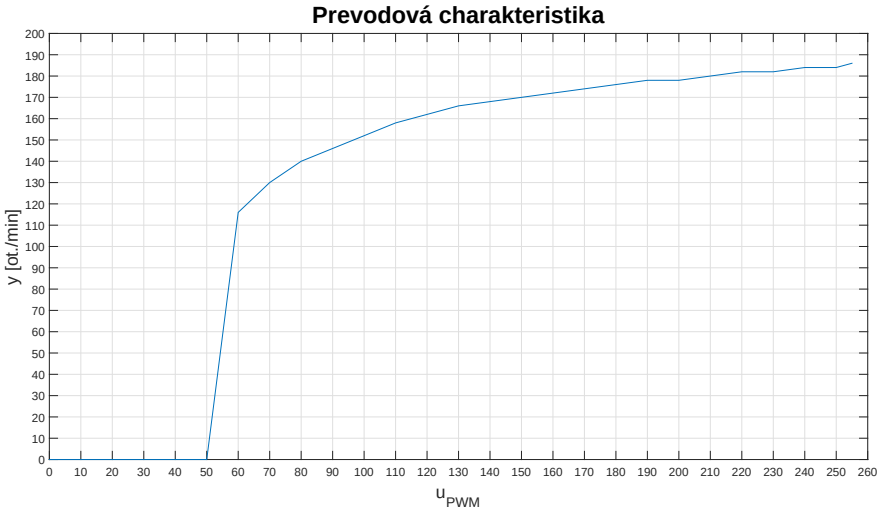
u_{PWM}	$y[ot./min]$	u_{PWM}	$y[ot./min]$
0	0	140	168
10	0	150	170
20	0	160	172
30	0	170	174
40	0	180	176
50	0	190	178
60	116	200	178
70	130	210	180
80	140	220	182
90	146	230	182
100	152	240	184
110	158	250	184
120	162	255	186
130	166	•	•

Tabuľka 2: Tabuľka odčítaných hodnôt v ustálených stavoch

Ďalším krokom je nameranie prevodovej charakteristiky pre konkrétne vstupy. Vstupom je napätie privádzané na motorček. Tieto vstupy sú o hodnote 0V až 5V, čo korešponduje s hodnotami modulovaného (PWM) signálu od 0 do 255 (rozlíšenie 8 bitov). Boli zvolené kroky o hodnote 10, pričom skok prebiehal každé 3 sekundy. Perióda vzorkova-

nia je 0.05 sekundy.

Následne bolo potrebné odčítať ustálené hodnoty výstupu $y[ot./min]$ pre príslušné hodnoty vstupu u_{PWM} (tabuľka 2). Hodnoty boli odčítavané s presnosťou na celé čísla. Prevodovú (statickú) charakteristiku možno vidieť na obrázku 12.



Obrázok 12: Prevodová (statická) charakteristika

Pri určovaní pracovných bodov bolo potrebné vybrať také body, okolo ktorých je priebeh charakteristiky blízky lineárnemu. Z prevodovej charakteristiky boli vybrané dva pracovné body:

$$u_{P_1} = 80 \rightarrow y_{P_1} = 140 ot./min$$

$$u_{P_2} = 170 \rightarrow y_{P_2} = 174 ot./min$$

Metódy a algoritmus riadenia

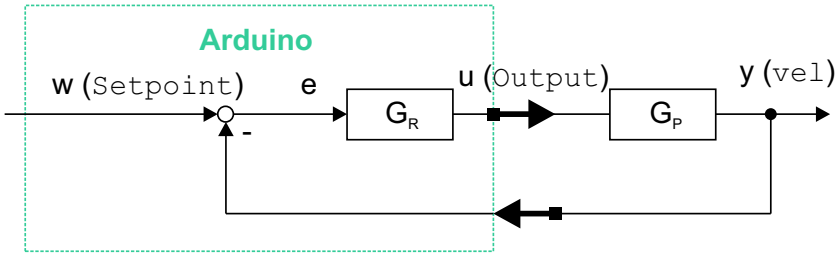
Nakoľko ide o návrh regulátora pre reálny systém s veľmi rýchlou dynamikou, šumami a nepresnosťami spôsobenými meraním otáčok enkóderom, boli použité inžinierske metódy návrhu regulátora s rešpektovaním známych vplyvov jednotlivých zložiek PID regulátora na priebeh

výstupu uzavretého regulačného obvodu (URO). PID regulátor bol programovo realizovaný prostredníctvom Arduina s využitím knižnice Arduino PID Library [2]. Vytvorenie objektu triedy PID má nasledovnú syntax:

- `PID (&Input, &Output, &Setpoint, Kp, Ki, Kd, Direction)`
  - `Input`: riadená veličina (`double`), v riešenom prípade otáčky motorčeka
  - `Output`: akčný zásah (`double`), v riešenom prípade efektívne napätie privádzané na motorček modulované PWM (0-255)
  - `Setpoint`: žiadaná (referenčná) hodnota (`double`), v riešenom prípade žiadané otáčky motorčeka
  - `Kp, Ki, Kd`: jednotlivé zložky regulátora (`double>=0`)
  - `Direction`: nastavuje sa na DIRECT alebo REVERSE; určuje, ktorým smerom (kladným alebo záporným) sa pohne akčný zásah pri regulačnej odchýlke; bežne sa používa DIRECT

Schému spätnoväzobného uzavretého regulačného obvodu možno vidieť na obrázku 13, kde strojovým typom písma (`monotype`) sú uvedené názvy premenných, ako sú používané v spomínanom programe. `Setpoint` značí žiadanú veličinu, v riešenom prípade teda žiadané otáčky motorčeka. Do programu uvádzame desaťnásobok žiadanej hodnoty z dôvodu použitého dátového typu pre túto veličinu, ktorá je *long*, čo sú celé čísla. Pridaním jedného rádu však získame možnosť práce s ekvivalentom čísla s jedným desatinným miestom. Premenná `Output` je akčným zásahom do riadeného systému a jej hodnota sa pohybuje medzi 0 a 255 (8 bitov). Ide o napätie 0-5 V modulované PWM v rozsahu 0-255 (8 bitov), ktoré je privádzané na motorček.

Experimentmi bolo zistené, že navrhnúť funkčný PID regulátor pre oblasť vyšších otáčok, a teda pre druhý pracovný bod (okolo 176 ot./min), je pomerne jednoduché, nakoľko je zabezpečená dobrá kvalita regulácie



Obrázok 13: Schéma spätnoväzobného uzavretého regulačného obvodu (URO)

s rôznymi parametrami regulátora. V nižších otáčkach (napr. v prvom pracovnom bode - okolo 140 ot./min) je návrh regulátora náročnejšia úloha a nie je možné použiť regulátor dobre fungujúci otáčkach vyšších, nakoľko získame systém na hranici stability, ba dokonca až systém nestabilný. Pre nižšie otáčky je potrebný regulátor s nižším zosilnením a integračnou konštantou, ktorý je síce použiteľný aj vo vyšších otáčkach, tu ale potom systém nedosahuje takú dynamiku a kvalitu riadenia (rýchlosť regulácie). V tomto prípade je teda vhodné použiť metódu riadenia hybridných systémov s využitím Petriho sietí vyšších úrovní predstavenú v rámci tejto práce, kedy je možné na základe stavu systému prepínať medzi rôznymi riadiacimi algoritmami v systéme.

Pre druhý pracovný bod bol na základe vyššie uvedených poznatkov navrhnutý PID regulátor s parametrami $P = 0.83$; $I = 5$; $D = 0.005$. S prihliadnutím na tolerančné pásmo $\pm 2 ot./min$ je doba regulácie 1.1 sekundy.

Tento regulátor bol odskúšaný aj pre skokovú zmenu žiadanej hodnoty v okolí prvého pracovného bodu (konkrétne z hodnoty 140 na 146). Na základe nameranej prechodovej charakteristiky URO, sa zistilo, že použitie tohto regulátora nezaručuje požadovanú kvalitu riadenia, nakoľko otáčky sa neustália na referenčnej hodnote s prihliadnutím na toleranciu $\pm 2 ot./min$.

Pre prvý pracovný bol nakoniec navrhnutý iný PID regulátor s nasle-

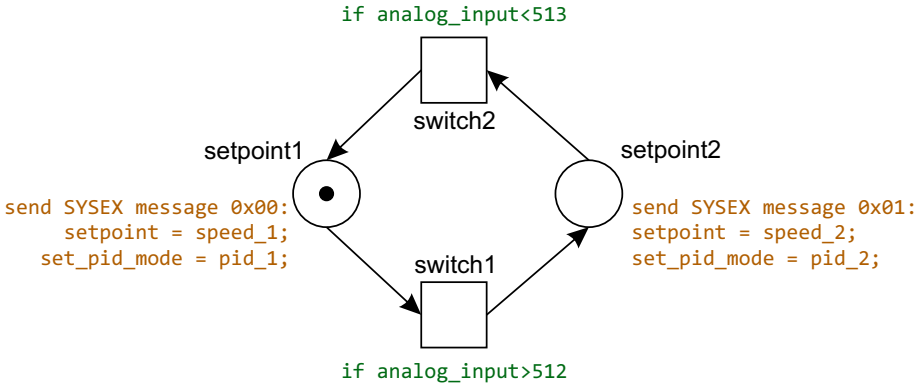
dovnými parametrami *P = 0.0001; I = 1; D = 0.01*. S prihliadnutím na toleranciu $\pm 2ot./min$ je doba regulácie 1.3 sekundy. Navrhovaný regulátor s týmito parametrami považujeme za vhodný a efektívny.

Regulátor s parametrami *P = 0.0001; I = 1; D = 0.01* bol odskúšaný aj v druhom pracovnom bode. Očakávaný predpoklad (menšia rýchlosť regulácie s regulátorom s nižším zosilnením a integračnou konštantou) sa splnil a teda kvalita regulácie je horšia ako pri prvom návrhu regulátora (*P = 0.83; I = 5; D = 0.005*). Doba regulácie je 2.75 sekundy, čo je viac ako pri prvom regulátore (1.1 sekundy).

Z analýzy riešenia je zrejmé, že pre každý pracovný bod je potrebné použiť rôzne regulátory. Jedným z vhodných riešení by bola možnosť prepínať medzi týmito regulátormi na základe toho, okolo akých otáčok (pracovného bodu) sa pohybujeme. Ďalšou možnosťou je využitie vyvinutého softvérového modulu PN2ARDUINO. Medzi jednotlivými regulátormi a žiadanými hodnotami je možné prepínať pomocou správ typu SYSEX. Ide o originálne riešenie autora tejto práce, nakoľko podľa analýzy súčasného stavu problematiky nebola takáto metóda riadenia hybridných systémov pomocou Petriho sietí vyšších úrovní softvérovo implementovaná a odskúšaná na riadení systému pomocou mikrokontroléra. Toto riešenie pokladáme za vedecký prínos práce k rozvoju vedného odboru.

Pre ilustráciu uvádzame schému na obrázku 14. Ide o jednoduchý príklad na demonštráciu tejto novej metódy riadenia. Uvažujme už spomínaný motorček a požadujme, aby efektívne fungoval v 2 módoch (pracovných bodoch). Pre efektívne ustálenie hodnoty otáčok na žiadanej hodnote potrebujeme regulátory s rôznymi parametrami pre každý z uvedených módov. Medzi módmi a regulátormi prepíname pomocou potenciometra napojeného na vstupný analógový pin Arduina. Prepínanie medzi módmi zabezpečujú prechody Petriho siete *switch1* a *switch2*, a to na základe vstupnej hodnoty súvisiacej s polohou potenciometra pripojenom na tomto pine. Vstupné hodnoty analógového pinu sa na Arduine pohybujú v rozmedzí 0-1023, preto ako prahová hodnota bola

použitá polovičná hodnotu (512). V momente, keď sa token v Petriho sieti dostane na miesto *setpoint1* alebo *setpoint2*, je do Arduina odoslaná správa typu SYSEX, ktorá zabezpečí spustenie užívateľom definovaného kódu na Arduine. V tomto prípade sa teda spustí konkrétny riadiaci algoritmus.



Obrázok 14: Schéma riadenia hybridného systému pomocou PN2ARDUINO

Z uvedených výsledkov je zrejmé, že využitie navrhovaných postupov, metódy a softvérového modulu je veľmi široké, nakoľko vďaka protokolu Firmata, SYSEX správam a možnosti pridania užívateľského kódu na stranu Arduina je možné použiť ľubovoľný riadiaci algoritmus, ktorý je vstavaný mikrokontrolér schopný realizovať. Logika Petriho sietí tiež vnáša možnosti riadenia udalostných systémov a riadenia na základe stavu systému. Charakter Petriho sietí tiež umožňuje v predstavenej softvérovej aplikácií vytvoriť efektívny expertný riadiaci systém.

4 Diskusia

Napriek tomu, že Petriho siete vyšších úrovní sú veľmi zaujímavým nástrojom na modelovanie a riadenie udalostných a hybridných systémov, dosiaľ neexistovalo ucelené riešenie, ktoré by umožňovalo riadenie takýchto systémov pomocou mikrokontrolérov. Z uvedených dôvodov dizertačná práca mala za jeden z hlavných cieľov navrhnúť metodiku, metódy a algoritmy riadenia diskrétnych udalostných a hybridných systémov s využitím Petriho sietí vyšších úrovni a overiť nich na riadení laboratórnych fyzikálnych systémov.

Uvedené nové prístupy a riešenia prezentované v dizertačnej práci sú zovšeobecniteľné a je ich možné modifikovať pre ďalší výskum a technickú prax. Vďaka formalizmu Petriho sietí, vyvinutej metóde riadenia hybridných systémov a príslušným SW nástrojom je možné navrhnúť a realizovať riadiaci systém aj ako expertný systém pre široké spektrum aplikácií (automobilový priemysel, vysoké pece, papierenský priemysel, čističky vôd a pod.).

SW moduly a systémy pre podporu modelovania a riadenia nazvané PN2ARDUINO a PetriNet editor + PetriNet engine sú postavené na rozšíriteľnom editore Petriho sietí PNEditor a je možné v budúcnosti ich dopĺňanie a zdokonaľovanie. Osobitnou výzvou bude vývoj a doplnenie SW podpory pre hybridné Petriho siete.

Záver

Predložená dizertačná práca sa zaoberá modernou metodikou modelovania a riadenia udalostných a hybridných systémov pomocou Petriho sietí vyšších úrovní prostredníctvom mikrokontrolérov.

Prvá kapitola rozoberá súčasný stav problematiky doma i v zahraničí. Diskutujú sa niektoré perspektívne vedecké výskumy a projekty v predmetnej oblasti.

V druhej kapitole je formulovaný problém riešený v rámci dizertačnej práce a sú vytýčené jej konkrétne ciele.

Štvrtá kapitola je ťažiskom autoreferátu dizertačnej práce a sú v nej uvedené originálne postupy a výsledky práce. Zaoberá sa opisom a využitím softvérového modulu PN2ARDUINO. Ide o rozšírenie existujúceho editora Petriho sietí PNEditor. Na jeho rozšírení autor predloženej dizertačnej práce spolupracoval s autormi diplomových prác, ktoré viedol ako konzultant. Softvérový modul PN2ARDUINO bol využitý na overenie originálnej metódy riadenia hybridných systémov, ktorá bola vyvinutá autorom dizertačnej práce. Toto riešenie pokladáme za vedecký prínos práce k rozvoju vedného odboru.

Záverečná kapitola diskutuje ďalšie možnosti rozvíjania danej problematiky a jej využitie v praxi.

Prínosy dizertačnej práce je možné zhrnúť do nasledovných bodov:

- **Vyvinutie softvérového modulu pre podporu modelovania a riadenia udalostných a hybridných systémov**
  V rámci riešenia dizertačnej práce bol vyvinutý softvérový modul pod názvom PN2ARDUINO založený na open-source editore Petriho sietí PNEditor. PN2ARDUINO umožňuje modelovanie systémov pomocou časových Petriho sietí interpretovaných pre riadenie a následnú realizáciu algoritmov riadenia prostredníctvom mikrokontrolérov rodiny Arduino a ďalších kompatibilných mik-

rokontrolérov, ktoré podporujú protokol Firmata. Tento protokol je použitý pre komunikáciu počítača s mikrokontrolérom.

- **Overenie navrhnutého programového systému pre riadenie udalostných systémov na laboratórnom fyzikálnom modeli**
  Pre overenie a demonštráciu možností riadenia udalostných systémov pomocou softvérového modulu PN2ARDUINO bol navrhnutý laboratórny model požiarneho hlásiča, ktorý sa skladá zo bzučiaka, fotorezistora a LED diódy. Na riadenie bol využitý mikrokontrolér Arduino Uno. V podpornom softvéri PN2ARDUINO bola namodelovaná príslušná riadiaca Petriho sieť a bolo demonštrované riadenie navrhnutého laboratórneho udalostného systému.

- **Návrh a overenie metódy riadenia hybridných systémov pomocou Petriho sietí vyšších úrovní**
  Pre overenie novej navrhovanej metódy riadenia hybridných systémov bolo potrebné vyvinúť laboratórny fyzikálny model takéhoto systému. Bol vybraný jednosmerný motorček s enkóderom pre zabezpečenie spätnej väzby. Z analýzy prevodovej charakteristiky systému boli vybrané dva pracovné body, pre ktoré boli navrhnuté dva algoritmy riadenia (regulátory). Optimálnym riešením je možnosť prepínania medzi týmito regulátormi na základe toho, okolo akých otáčok (pracovného bodu) sa pohybujeme. Jednou z možností je využitie softvéru PN2ARDUINO. Medzi jednotlivými regulátormi a žiadanými hodnotami regulovanej veličiny je možné prepínať pomocou SYSEX správ a protokolu Firmata. Ide o originálne riešenie autora práce. Toto riešenie pokladáme za najvýznamnejší prínos práce pre rozvoj vedného odboru. Vďaka protokolu Firmata, SYSEX správam a možnosti pridania užívateľského kódu na stranu Arduina je možné použiť ľubovoľný riadiaci algoritmus, ktorý je vstavaný mikrokontrolér schopný realizovať.

- **Návrh a simulačné overenie metódy riadenia triedy automatizovaných skladových systémov**
  Softvérová aplikácia PNetLab je vhodným nástrojom pre modelovanie systému pomocou farebných Petriho sietí. Výhodou tohto nástroja je to, že umožňuje riadenie modelu pomocou logických predikátov v externom programe (interpretovaný supervízor). Prostredníctvom takéhoto programu bolo navrhnuté riadenie triedy automatizovaných skladových systémov. Riadenie bolo simulačne overené na modeli skladovaného systému realizovaného pomocou farebnej Petriho siete.

Prínosy dizertačnej práce deklarované v štyroch bodoch predstavujú opis nových postupov a riešení v oblasti modelovania a riadenia diskrétnych udalostných a hybridných systémov, sú zovšeobecniteľné a je možné ich ďalej modifikovať pre ďalší výskum a technickú prax.

Publikačná činnosť autora

PRÍSPEVKY NA KONFERENCIÁCH

- Medzinárodné konferencie:

  1. BALOGH, Štefan, Fedor LEHOCKI, Daniel IVANIŠ, Erik KUČERA, Miloš LAJTMAN a Igor MIŇO. Data processing from mHealth patient data acquisition related to extracting structured data from EH records. In: GODARA, Balwant. *Wireless mobile communication and healthcare: Third International Conference, Mobihealth 2012*, Paris, France, November 21-23, 2012, revised selected papers. 1st ed. New York, NY: Springer-Verlag Berlin Heidelberg, 2013, s. 8. ISBN 3642378927. V databáze: SCOPUS.
  2. KUČERA, Erik, Branislav HRÚZ a Leo MRAFKO. Modelling of automated storage systems using coloured petri nets. In CIGÁNEK, Ján, Štefan KOZÁK, Alena KOZÁKOVÁ a Danica ROSINOVÁ. *Kybernetika a informatika 2014 : Medzinárodná konferencia SSKI*. Oščadnica, Slovenská republika, 5.-8. 2. 2014. Bratislava: STU v Bratislave FEI, 2014, s. 6. ISBN 978-80-227-4122-4.
  3. KUČERA, Erik a Branislav HRÚZ. Modelling of automated storage systems using hierarchical coloured petri nets. In *Proceedings of the 15th International Carpathian Control Conference [elektronický zdroj] : ICCC 2014*; Velké Karlovice, Czech Republic, May 28-30, 2014. [s.l.]: IEEE- Czechoslovakia Section of IEEE, 2014, s. 285–290. ISBN 978-1-4799-3527-7. V databázach: IEEE Xplore a SCOPUS.
  4. KUČERA, Erik, Branislav HRÚZ, Iveta DIRGOVÁ a Marek ŠIMON. Modelling and control of the manufacturing/storage systems using Petri nets. *In INFOCOM 6 : Infokommunika-*

cionnyje technologii v nauke, proizvodstve i obrazovanii. Časť 1. Stavropoľ, 2014, S. 8-15. ISSN 2219-293X.

5. KUČERA, Erik a Branislav HRÚZ. Modelling of AS/RS Using Hierarchical and Timed Coloured Petri Nets. In *RAAD 2014 [elektronický zdroj] : 23rd International Conference on Robotics in Alpe-Adria-Danube Region*. Conference Proceedings. September 3-5, 2014, Smolenice Castle, Slovakia. 1. vyd. Bratislava: Publishing House of Slovak University of Technology, 2014, s. 8. ISBN 978-80-227-4219-1. V databázach: IEEE Xplore a SCOPUS.
6. KUČERA, Erik, Miriama NIŽNANSKÁ a Štefan KOZÁK. Advanced techniques for modelling of AS/RS systems in automotive industry using high-level petri nets. In ICCC 2015 : 16th International Carpathian Control Conference. Szilvásvárad, Hungary. May 27-30, 2015. Danvers : IEEE, 2015, S. 261-266. ISBN 978-1-4799-7369-9. V databázach: IEEE Xplore a SCOPUS.
7. KUČERA, Erik, Oto HAFFNER a Štefan KOZÁK. Modelling and control of AS/RS using coloured petri nets. In 2016 Cybernetics & informatics (K& I) [elektronický zdroj] : 28th International conference. Levoča, Slovak Republic. February 2-5, 2016. 1. ed. [s.l.] : IEEE, 2016, CD-ROM, [6] s. ISBN 978-1-5090-1834-5. V databáze: IEEE Xplore.
8. NIŽNANSKÁ, Miriama, Erik KUČERA, Oto HAFFNER a Alena KOZÁKOVÁ. Modeling of hybrid systems by hybrid petri nets using open modelica and PNlib. In 2016 Cybernetics & informatics (K& I) [elektronický zdroj] : 28th International conference. Levoča, Slovak Republic. February 2-5, 2016. 1. ed. [s.l.] : IEEE, 2016, CD-ROM, [6] s. ISBN 978-1-5090-1834-5. V databáze: IEEE Xplore.
9. HAFFNER, Oto, Erik KUČERA a Mária BACHÚRIKOVÁ. Pro-

posal of weld inspection system with single-board computer and Android smartphone. In 2016 Cybernetics & informatics (K& I) [elektronický zdroj] : 28th International conference. Levoča, Slovak Republic. February 2-5, 2016. 1. ed. [s.l.] : IEEE, 2016, CD-ROM, [5] s. ISBN 978-1-5090-1834-5. V databáze: IEEE Xplore.

10. HAFFNER, Oto, Erik KUČERA, Erik a Štefan KOZÁK. Weld segmentation for diagnostic and evaluation method. In 2016 Cybernetics & informatics (K& I) [elektronický zdroj] : 28th International conference. Levoča, Slovak Republic. February 2-5, 2016. 1. ed. [s.l.] : IEEE, 2016, CD-ROM, [6] s. ISBN 978-1-5090-1834-5. V databáze: IEEE Xplore.

- Domáce konferencie:

1. KUČERA, Erik, Miloš LAJTMAN a Štefan BALOGH. An automated method for data mining in Slovak health records. In *ELITECH´14 [elektronický zdroj] : 16th Conference of Doctoral Students*; Bratislava, Slovakia, 4 June 2014. Bratislava: Nakladateľstvo STU, 2014, s. 6. ISBN 978-80-227-4171-2.
2. KUČERA, Erik. Modelling of storage / manufacturing systems using coloured petri nets. In ELITECH´15 [elektronický zdroj] : 17th Conference of doctoral students. Bratislava, Slovak Republic, May 25, 2015. 1. vyd. Bratislava : Nakladateľstvo STU, 2015, CD-ROM, 7 s. ISBN 978-80-227-4358-7.
3. KUČERA, Erik, Oto HAFFNER and Štefan KOZÁK. Modelling and Control of AS/RS Using Coloured Petri Nets with PNetLab Tool. In ELITECH´16 : 18th Conference of doctoral students. Bratislava, Slovak Republic, 2016.
4. NIŽNANSKÁ, Miriama, Erik KUČERA, Oto HAFFNER a Alena KOZÁKOVÁ. Modeling of Hybrid Systems using Hybrid Petri Nets with Open Modelica and PNlib Library. In ELITECH´16

: 18th Conference of doctoral students. Bratislava, Slovak Republic, 2016.

5. HAFFNER, Oto, Erik KUČERA, Erik a Štefan KOZÁK. Weld Segmentation for Diagnostic and Evaluation Method on Raspberry Pi 2. In ELITECH´16 : 18th Conference of doctoral students. Bratislava, Slovak Republic, 2016.

AUTORSKÉ OSVEDČENIA, PATENTY A OBJAVY

- Úžitkové vzory:

  1. HAFFNER, Oto a Erik KUČERA. Postup na vytvorenie povrchu s imitáciou dreva : Úžitkový vzor č. 7045. Dátum oznámenia o zápise ÚV: 16.1.2015. Vestník ÚPV SR č. 03/2015. Banská Bystrica : Úrad priemyselného vlastníctva SR, 2015. 2 s.

ČASOPISY

- Domáce nekarentované časopisy:

  1. KUČERA, Erik. *Študentské účty - podmienky poskytovania a porovnanie s bežnými účtami*. Posterus [elektronický zdroj] : Internetový časopis Vol. 6, Iss. 2. zv. 6, ISSN 1338-0087.

OSTATNÉ

1. KUČERA, Erik. Modelovanie automatizovaných skladových systémov pomocou farebných Petriho sietí. Bratislava, 2013. Príspevok na súťaži *Študentská vedecká a odborná činnosť 2013 (sekcia Kybernetika a robotika)*. Fakulta elektrotechniky a informatiky STU. Vedúci práce Leo Mrafko.

Zoznam použitej literatúry

[1] *Aircraft Catapult: Model Launch Mechanism. Wolfram SystemModeler*. [online]. [cit. 2016-04-23]. Dostupné z: `http://www.wolfram.com/system-modeler/examples/aerospace-defense/aircraft-catapult-launch.html`.

[2] *Arduino PID Library. GitHub*. [online]. [cit. 2016-05-16]. Dostupné z: `https://github.com/br3ttb/Arduino-PID-Library/`.

[3] *Firmata firmware for Arduino. GitHub*. 2016. [online]. [cit. 2016-04-17]. Dostupné z: `https://github.com/firmata/arduino`.

[4] *Wolfram SystemModeler. Wolfram - Computation Meets Knowledge*. [online]. [cit. 2016-04-23]. Dostupné z: `http://www.wolfram.com/system-modeler/`.

[5] BRAUN, W., BACHMANN, B., AND PROSS, S. Synchronous events in the openmodelica compiler with a petri net library application. In *EOOLT* (2010), Citeseer, pp. 63–70.

[6] BRIŠ, Ladislav. *Využitie formalizmov Petriho sietí v riadení laboratórnych systémov*. Diplomová práca. Bratislava: FEI STU, 2016. 58 s.

[7] COSTANTINO, N., DOTOLI, M., FALAGARIO, M., FANTI, M. P., AND MANGINI, A. M. A model for supply management of agile manufacturing supply chains. *International Journal of Production Economics 135*, 1 (2012), 451 – 457. Advances in Optimization and Design of Supply Chains.

[8] DAVID, R. Modeling of hybrid systems using continuous and hybrid petri nets. In *Petri Nets and Performance Models, 1997., Proceedings of the Seventh International Workshop on* (Jun 1997), pp. 47–58.

[9] DAVID, René a Hassane ALLA. *Discrete, continuous and hybrid petri nets*. Berlin: Springer, 2010, xxii, 550 s. ISBN: 978-3-642-10668-2.

[10] DOTOLI, M., FANTI, M., AND IACOBELLIS, G. Validation of an urban traffic network model using colored timed petri nets. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on* (Oct 2005), vol. 2, pp. 1347–1352.

[11] DOTOLI, M., FANTI, M., AND IACOBELLIS, G. A freeway traffic control model by first order hybrid petri nets. In *Automation Science and Engineering (CASE), 2011 IEEE Conference on* (Aug 2011), pp. 425–431.

[12] DOTOLI, M., FANTI, M., AND MANGINI, A. Comparing management policies for supply chains via a hybrid petri net model. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on* (Oct 2007), pp. 3469–3474.

[13] DOTOLI, M., FANTI, M., AND MANGINI, A. Fault monitoring of automated manufacturing systems by first order hybrid petri nets. In *Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on* (Aug 2008), pp. 181–186.

[14] DOTOLI, M., AND FANTI, M. P. An urban traffic network model via coloured timed petri nets. *Control Engineering Practice 14*, 10 (2006), 1213 – 1229. The Seventh Workshop On Discrete Event Systems (WODES2004) The Seventh Workshop On Discrete Event Systems (WODES2004).

[15] DOTOLI, M., FANTI, M. P., GIUA, A., AND SEATZU, C. Modelling systems by hybrid petri nets. an application to supply chains.

[16] DRATH, Rainer. *Visual Object Net++*. [online]. [cit. 2016-04-16]. Dostupné z: `http://www.r-drath.de/Home/Visual_Object_Net++.html`.

[17] DRIGHICIU, M., MANOLEA, G., CISMARU, D., AND PETRISOR, A. Hybrid petri nets as a new formalism for modeling electrical drives. In *Power Electronics, Electrical Drives, Automation and Motion, 2008. SPEEDAM 2008. International Symposium on* (June 2008), pp. 626–631.

[18] DRIGHICIU, M., PETRISOR, A., AND POPESCU, M. A petri nets approach for hybrid systems modelling. *International Journal of Circuits, Systems and Signal Processing 2*, 3 (2009).

[19] DRIGHICIU, M. A. Application du formalisme reseaux de pétri pour la modélisation de systèmes hybrides. *ICCPS, Moldova* (2007), 152–155.

[20] DRIGHICIU, M.-A., AND CISMARU, D. C. Modeling a water bottling line using petri nets. *Annals of the University of Craiova, Electrical Engineering series* (2011).

[21] DRIGHICIU, M. A., AND MANOLEA, G. Application des reseaux de petri hybrides a l'etude des systemes de production a haute cadence.

[22] DRIGHICIU, M. A., AND PETRIŞOR, A. A modified petri net framework for hybrid system modeling.

[23] ČEŠEK, Pavol. *Riadenie DEDS pomocou mikrokontrolérov s využitím Petriho sietí*. Diplomová práca. Bratislava: FEI STU, 2016.

[24] ČEŠEKOVÁ, Alžbeta. *Riadenie laboratórnych udalostných systémov*. Diplomová práca. Bratislava: FEI STU, 2016. 36 s.

[25] FABRICIUS, S., AND BADREDDIN, E. Modelica library for hybrid simulation of mass flow in process plants. In *Proceedings of the 2nd International Modelica Conference', Oberpfaffenhofen, Germany* (2002), Citeseer, pp. 225–234.

[26] FANTI, M., IACOBELLIS, G., MANGINI, A., AND UKOVICH, W. Freeway traffic modeling and control in a first-order hybrid petri net framework. *Automation Science and Engineering, IEEE Transactions on 11*, 1 (Jan 2014), 90–102.

[27] GALANDÁK, Ľuboš. *Počítačová podpora komunikácie a ovládania mechatronických systémov pomocou mikrokontrolérov*. Diplomová práca. Bratislava: FEI STU, 2016.

[28] JÚLVEZ, J., DI CAIRANO, S., BEMPORAD, A., AND MAHULEA, C. Event-driven model predictive control of timed hybrid petri nets. *International Journal of Robust and Nonlinear Control* (2013).

[29] LOCHEL, Lennart. *PNlib - Free Modelica library for modelling xHPN. GitHub*. 2016. [online]. [cit. 2016-04-23]. Dostupné z: `https://github.com/lochel/PNlib`.

[30] MAHULEA, C., GIUA, A., RECALDE, L., SEATZU, C., AND SILVA, M. Optimal model predictive control of timed continuous petri nets. *Automatic Control, IEEE Transactions on 53*, 7 (2008), 1731–1735.

[31] MOSTERMANY, P. J., OTTERY, M., AND ELMQVISTZ, H. Modeling petri nets as local constraint equations for hybrid systems using modelica. *retrieved online at http://citeseer. ist. psu. edu/359408. html* (1998).

[32] NIŽNANSKÁ, Miriama. *Aplikácia na editovanie a softvérovú podporu hybridných Petriho sietí*. Diplomová práca. Bratislava: FEI STU, 2015. 93 s.

[33] PROS, S., BACHMANN, B., JANOWSKI, S., AND HOFESTADT, R. A new object-oriented petri net simulation environment based on modelica. In *Simulation Conference (WSC), Proceedings of the 2012 Winter* (Dec 2012), pp. 1–13.

[34] PROSS, S., AND BACHMANN, B. Hybrid modelling and process optimization of biological systems. In *MATHMOD Conference, Wien, Austria* (2012).

[35] PROSS, S., AND BACHMANN, B. Pnlib-an advanced petri net library for hybrid process modeling. In *Modelica Conference* (2012).

[36] PROSS, S., BACHMANN, B., HOFESTÄDT, R., NIEHAUS, K., UECKERDT, R., VORHÖLTER, F.-J., LUTTER, P., AND STADTHOLZ, A. Modeling a bacterium's life: A petri-net library in modelica. In *Como, Italy, Modelica Conference proceedings* (2009).

[37] PROSS, S., BACHMANN, B., AND STADTHOLZ, A. A petri net library for modeling hybrid systems in openmodelica. In *submitted (Modelica Conference 2009)* (2009).

[38] PROSS, S., JANOWSKI, S. J., BACHMANN, B., KALTSCHMIDT, C., AND KALTSCHMIDT, B. Pnlib-a modelica library for simulation of biological systems based on extended hybrid petri nets. In *3rd International Workshop on Biological Processes & Petri Nets, Hamburg, Germany* (2012).

[39] RIESZ, Martin. *PNEditor - a Petri Net editor. PNEditor*. 2014. [online]. [cit. 2016-04-24]. Dostupné z: `http://www.pneditor.org/`.

[40] ROHR, C., MARWAN, W., AND HEINER, M. Snoopy—a unifying petri net framework to investigate biomolecular networks. *Bioinformatics 26*, 7 (2010), 974–975.

[41] RUIZ, Leonardo Laguna a Johan RHODIN. *Using Arduinos as SystemModeler Components. Wolfram Blog*. 2014. [online]. [cit. 2016-04-23]. Dostupné z: `http://blog.wolfram.com/2014/11/07/using-arduinos-as-systemmodeler-components/`.

[42] SLINTÁK, Vlastimil. *Arduino a přerušení. uArt.cz*. [online]. [cit. 2016-05-12]. Dostupné z: `http://uart.cz/271/arduino-a-preruseni/`.