

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

BAKALÁŘSKÁ PRÁCE



BAKALÁŘSKÁ PRÁCE

Kooperativní řízení multiagentního systému

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejíž úplný seznam je její součástí.

V Plzni, 2023

.....

Poděkování

Tímto bych chtěl vyjádřit své upřímné poděkování všem, kteří mě při psaní bakalářské práce podporovali a pomohli mi ji dokončit. Nejprve bych rád poděkoval svému vedoucímu práce Karlu Kubíčkoví. Vaše odborné vedení, trpělivost, připomínky a rady byly velmi cenné a přínosné pro úspěšné dokončení této práce. Díky Vašemu vedení jsem získal mnoho cenných znalostí a zkušeností, které mi budou sloužit i nadále v životě. Také bych chtěl upřímně poděkovat své rodině, přátelům a přítelkyni za projevenou podporu, trpělivost a pochopení v průběhu realizace této práce, ale i v průběhu celého studia.

Abstrakt

Tato bakalářská práce se zaměřuje na kooperativní řízení multiagentních systémů, zvláště na úlohy rozdělení agentů do formací a jejich následné udržování v čase. Dále práce obsahuje seznámení s koncepty grafové teorie a jejich vliv na topologii a řízení multiagentních systémů. Je zde také vysvětlen a použit konsensus protokol pro různé topologie komunikační sítě. V práci jsou popsány různé způsoby řízení multiagentních systémů, kdy je využit Laplacián a dále jsou zavedeny a ukázány způsoby řízení s virtuálním nebo externím leadrem. Následně je zpracováno rozdělení agentů do předem definovaných formací. V těchto formacích jsou zahrnuty útvary jako je trojúhelník, čtverec, hexagon a finálně oktagon, přičemž nejprve systémy dosáhnou konsensu a teprve poté jsou rozděleny do formací, ve kterých jsou následně v čase udržovány. Jako rozšíření těchto úkonů je v práci zahrnuta změna formace v čase z oktagonu do čtyřcípé hvězdy. U všech příkladů jsou diskutovány hlavní vlivy, které ovlivňují chování jednotlivých systémů.

Klíčová slova: cooperative control, distribuované řízení, externí leader, formace agentů, konsensus, Laplacián, centralizované řízení, rekonfigurace formace, stavová zpětná vazba, virtuální leader.

Abstract

This bachelor's thesis focuses on cooperative control of multi-agent systems, particularly on the tasks of agent allocation into formations and their subsequent maintenance over time. Furthermore, the thesis includes an introduction to graph theory concepts and their impact on the topology and control of multi-agent systems. Additionally, a consensus protocol for different communication network topologies is explained and applied. The thesis describes various methods of controlling multi-agent systems, utilizing the Laplacian, and introduces ways of controlling with a virtual or external leader. Subsequently, the allocation of agents into predefined formations is addressed. These formations include shapes such as triangles, squares, hexagons, and ultimately octagons, where the systems first achieve consensus before being divided into formations and subsequently maintained over time. As an extension of these tasks, the thesis incorporates a time-varying formation change from an octagon to a four-pointed star. The main influences affecting the behavior of individual systems are discussed for all examples.

Klíčová slova: cooperative control, distributed control, external leader, agent formations, consensus, Laplacian, centralized control, formation reconfiguration, state feedback, virtual leader.

Obsah

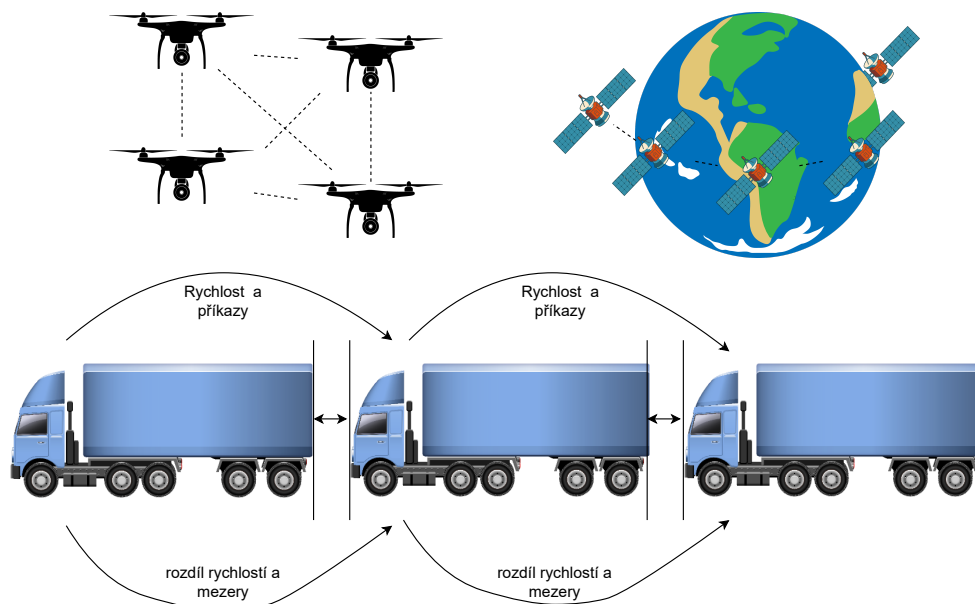
1	Úvod	1
1.1	Motivace	3
2	Multiagentní systémy	5
2.1	Stavový popis lineárního systému	5
2.1.1	Stavová zpětná vazba	6
2.2	Úvod do grafové teorie	7
2.2.1	Notace - pojmy z grafové teorie	7
2.3	Definice grafů	8
2.4	Grafy a matice	11
2.4.1	Notace - pojmy z algebraické grafové teorie	11
3	Algoritmus konsensu (shody)	17
3.1	Systém prvního řádu	17
3.2	Řízení s leadrem	18
3.3	Vlivy topologie na chování konsensu	22
3.4	Simulace konsensu pro různé topologie grafů	30
3.5	Konsensus protokol pro formace	32
4	Formace agentů	36
4.1	Udržování formace v čase	39
4.2	Rekonfigurace formace agentů	41
5	Diskuze získaných výsledků	44
5.1	Future works	46
6	Závěr	48
	Seznam obrázků	49
	Seznam tabulek	50

1 Úvod

S aktuálně přibývajícím zájmem o multiagentní systémy, způsobem jejich návrhu a řízení, je toto téma stále častěji řešeno v širokém spektru projektů a aplikací. Z tohoto důvodu si tato práce dává za cíl shrnout a sjednotit informace o problematice multiagentních systémů a ověřit tyto znalosti simulačně na specifických příkladech, kdy bude zejména uvažováno řízení a udržování formace agentů.

V práci budou popsány všechny potřebné matematické pojmy, které jsou využity pro popis a řízení multiagentních systémů. Jedná se zejména o grafovou teorii, maticové popisy, ale i o problematiku nalezení konsensu nebo Laplacián. Chování systému značně ovlivňuje použitý způsob řízení, kdy se můžeme v literatuře setkat s pojmy jako řízení s leadrem, řízení s virtuálním leadrem nebo zcela distribuovaný způsob řešení [21]. Je podstatné zmínit, že rozdíl mezi leadrem a aditivním leadrem je ten, že aditivní leader se fyzicky nenachází v systému, a je tak řešen pouze programově.

V práci bude diskutováno, jak tyto přístupy ovlivňují kvalitu výsledného řízení, rozdělení a udržování formace z různých počátečních podmínek. Dále bude popsán vliv počtu komunikačních hran v dané topologii sítě na rychlost předávání informace mezi jednotlivými agenty a tím pádem jejich vliv na rychlost nalezení konsensu.



Obrázek 1: Ukázka možné aplikace multiagentních systémů

Pojem formace neodmyslitelně patří do problematiky multiagentních systémů. Příkladem z reálného nasazení může být "Truck platooning", tedy udržování nákladních automobilů v konvoji. Znázornění tohoto konvoje je možno vidět na Obrázku 1. Benefitem tohoto přístupu by měla být plynulejší doprava, rychlejší přeprava většího množství nákladů a lepší spotřeba pohonných hmot v důsledku optimalizace rozestupů mezi jednotlivými kamiony na silnici. První vozidlo může být vybaveno speciálními doplňky karoserie, které přispějí

ke snížení odporu vzduchu, kdy zbytek konvoje pojede za ním, a nemusí už tak vzduch rozrážet znovu [6]. Zde navíc kombinace s moderní automatickou převodovkou může vést k dalšímu ušetření paliva (může se tak jednat až o jednotky procent na dané trase) [9]. Rozvoj dalších polo-autonomních a autonomních dopravních systémů může významným způsobem přispět ke snížení emisí, a díky tomu i k celkové udržitelnosti lidské činnosti.

Způsob řízení se dá rozdělit na řízení s leadrem, nebo plně distribuované řešení. V případě platooningu se způsobu řízení uvažuje aditivním přidáním nového agenta - leadra, který reprezentuje referenční chování celé skupiny, a je tak fyzicky přítomný v daném systému. Na základě komunikační topologie předává informace o požadovaném chování celku ostatním agentům, kteří jsou tak schopni udržovat optimální vzdálenost mezi jednotlivými vozidly formace [21].

Další významným projektem multiagentních systémů představuje Starlink společnosti SpaceX, jehož ilustrace je znázorněna na Obrázku 1. Tento projekt představuje konstelaci satelitů poskytující vysokorychlostní a nízkolatenční internet, mimo jiné díky své relativní blízkosti k Zemi. Tato komunikační síť totiž zaujímá, na rozdíl od většiny už existující satelitních sítí, nízkou orbitu Země.

V této aplikaci představují satelity jednotlivé agenty, kteří tvoří komplikovanou komunikační síť pokrývající značnou část zemského povrchu. Tento systém se skládá z většího počtu menších satelitů, které jsou schopny pokrýt větší plochu oblohy, a díky tomu poskytnout kvalitnější připojení k internetu. Na tom lze vidět myšlenku multiagentního přístupu, kdy je jeden velký, drahý satelit nahrazen množinou menších, levnějších satelitů, které poskytují stejné, nebo lepší služby. Starlink ve své finální formě bude poskytovat prakticky celosvětové pokrytí internetem, bez kterého se moderní společnost neobejde [8, 4].

Hlavním cílem této práce je zprostředkovat ucelený vhled a pochopení problematiky multiagentního přístupu, kdy si pomáhá řadou příkladů. Informace, jež budou v této práci popsány lze využít k úvodnímu vyzkoušení multiagentních algoritmů a jejich dalšímu rozvoji nebo vylepšení. Všechny simulace a algoritmy uvedené v této práci jsou vytvořeny v prostředí programu Matlab. Ten poskytuje účinné matematické nástroje pro tvorbu, analýzu a simulaci, které budou využity pro problematiku multiagentních systémů. Tento nástroj je hojně využíván i v průmyslové sféře, například pro možnosti generování kódu v automobilovém průmyslu [11].

Tato práce je členěna do pěti kapitol, z nichž první dvě jsou zaměřeny na vysvětlení rozebírané problematiky a uvedení klíčových pojmů pro pochopení nadcházejících třech kapitol, které obsahují hlavní témata celé práce spolu se získanými výsledky simulací vytvořených v Matlabu. První kapitola má za úkol naznačit důvody a motivace, proč jsou multiagentní systémy a jejich výzkum v dnešní době žádané a zároveň představit multiagentní systémy jako celek, nastínit jejich historii a uvést příklady různých aplikací. V druhé kapitole budou uvedeny významné pojmy a principy z grafové a algebraické grafové teorie, které umožní definovat vlastnosti multiagentních systémů. Ve třetí kapitole bude rozebírán pojem konsensus a jeho realizace ve formě lokálního protokolu řízení s Laplaciánem pro distribuované řešení spolu s řízením virtuálním a externím leadrem pro centralizované řešení. Budou zde analyzovány vlivy topologie grafů na systémy pro všechny výše zmíněné řídicí strategie spolu se simulacemi dosažení konsensu pro různé topologie, kde tyto vlivy

je možné pozorovat. Nově v této kapitole budou představeny formace agentů v rovině. Poslední kapitola se zabývá právě formacemi a zaměřuje se na udržování formací v čase pomocí stavového zpětnovazebního řízení, pro které je cílem nalézt korektní parametry zpětnovazební matice. Bude zaveden nový popis systému, který zahrnuje pozice a rychlosti agentů v prostoru. Finálně budou provedeny simulace udržování formací v čase pro různé formace spolu se simulací rekonfigurace původní formace v čase. V závěru práce jsou zmíněny případné možnosti budoucího rozšíření této práce, či další možný vývoj v oblasti multiagentních systémů.

1.1 Motivace

Multiagentní systémy zažívají v posledních letech významný rozvoj a to zejména v oborech jako je robotika, řídicí systémy, komunikační sítě, energetika a mnoho dalších [19]. Za zmínku stojí určitě populární projekt RoboCup Soccer, který podporuje mezinárodní výzkum multiagentních systémů v robotice pomocí fotbalového benchmarku [10]. V komunikačních sítích se nyní velmi často diskutuje systém Starlink, který byl popsán již v úvodu. Zajímavým projektem na poli energetiky je například decentralizovaný způsob řízení energetických sítí, zejména decentralizované řešení "Economic Dispatch Problem" (EDP), kdy se jedná o optimalizační úlohu. Úkolem EDP je rozdělit požadované zatížení mezi N aktivních generátorů v síti tak, aby výsledná cena za jednotku energie byla minimální [12].

Využití multiagentních systému kopíruje aktuální trend odklonu od centralizovaného k decentralizovanému řízení. Decentralizovaný přístup řízení přináší benefity v podobě levnějšího provozu, lepší potlačení vnějších poruch a odolnosti vůči chybám. Důležitou vlastností decentralizovaného řízení je, že pokud selže jeden agent, zbytek systému by stále mohl fungovat bez závažných problémů. To samozřejmě platí jen v případě, že zbylí agenti jsou stále schopni splnit požadovaný cíl. To se může lišit v závislosti na dané aplikaci. Decentralizovaný způsob řešení ale přináší větší nároky na komunikaci a programové řízení dané aplikace [22]. Naproti tomu centralizovaný způsob řešení se při zničení nebo nedostupnosti leadra, který řídí celý systém, stává neřízeným, takový systém není schopen dále plnit svůj původní cíl [19].

Vlastnosti decentralizovaných systémů jsou obzvláště výhodné v aplikacích jako monitorování velkých oblastí, mapování neznámých či těžko přístupných míst nebo v zónách, které byly zasaženy přírodní katastrofou. Aby takovéto úlohy byly spolehlivě a přesně vykonány, je nutné sestavit takové řízení, které účinně umožní agentům spolu efektivně kooperovat, komunikovat a zároveň omezovat množství dat, které jsou k tomu potřeba. Někdy je vhodné data limitovat kvůli rychlosti zpracování a jejich přenosu, lze se ale setkat i s aplikacemi, kde hraje roli bezpečnost nebo soukromí daných agentů - uživatelů, kdy agenti nechtějí sdílet s ostatními některá svá interní data nebo měření [12]. Dále pokud agent předá příliš mnoho informací dalším agentům, může nastat situace, že se celý systém přetíží. To by mělo za následek celkově snížený výkon u systému s pomalejšími odezvami na požadavky, které jsou kladeny celkově na celý systém. Z tohoto plyne, že správně navržený řídicí algoritmus, který limituje používaná data na minimum, může vést k zefektivnění funkce celého systému. S tím má také souvislost návrh komunikační topologie

daného systému [3].

Myšlenka multiagentních systémů se v odborné literatuře objevuje už několik desítek let, nicméně jednu z nejzazších zmínek o pojmu "agenta" můžeme najít v práci Johna Von Neumanna a Oskara Morgensterna, kteří vymysleli koncept herní teorie v padesátých letech minulého století [16]. Herní teorie v jejich podání je matematickým rámcem, který zkoumá chování racionálních hráčů - agentů ve hře, kde jsou zapotřebí strategická rozhodnutí. Pomocí této teorie jsou tak schopni analyzovat chování a interakce jednotlivých hráčů individuálně a mezi sebou a následně pak matematicky vyjádřit postup, jak mohou dosáhnout cíle, tedy vyhrát hru [16]. Na základě těchto myšlenek byl definován a rozšiřován pojem "agent". K rozvoji spolupráce a zkoumání nalezení konsensu mezi agenty došlo až po zavedení pojmů jako jsou "Cooperative control" nebo "Multi-agent system" [19]. Mezi hlavní zakladatele oblasti "Cooperative control" a "Multi-agent system" lze bez pochyby zařadit J. A. Faxe a Richarda M. Murrayho, kteří v této oblasti dosáhli významných výsledků. Jejich práce se tak stala základem jakéhokoliv dalšího vývoje této problematiky a je hojně citována i v nových vědeckých pracích [17, 5]. Zde by se dalo samozřejmě vyzdvihnout obrovské množství všech možných dalších autorů a prací, což ale není cílem této práce a zde se tak spokojíme pouze s krátkou motivací.

2 Multiagentní systémy

Multiagentní systémy představují soubor dvou a více autonomních agentů, kteří spolu komunikují a spolupracují k dosažení sledovaného společného cíle. Abychom mohli správně definovat, jak modelovat a analyzovat takovéto systémy, tak potřebujeme využít nástrojů z grafové teorie. Grafová teorie umožňuje zachytit topologické struktury multiagentních systémů ve formě grafu, kde jsou agenti zachyceni jako vrcholy a komunikační, nebo interakční vazby mezi nimi jsou zobrazeny jako hrany. Pro další využití při práci s multiagentními systémy se zavádí pojem Laplacián, který obsahuje všechny atributy systému daného grafem jako celku, včetně komunikace jednotlivých agentů mezi sebou, to vše v rámci jedné matice.

Laplacián má v oblasti multiagentních systémů velmi významné využití. Laplacián se využívá například v algoritmech určených pro řízení, dále se může jednat o nalezení konsensu nebo rozdělení do dané formace. Laplacián v takovýchto příkladech určuje chování, trajektorie a rychlost agentů, dokud se nedostanou do ustáleného stavu. Jsou i další možnosti dosažení konsensu jako je: "volení", či "průměrování" [18]. Nicméně tato práce se dále zaměří zejména na algoritmy, které využívají Laplacián pro svoji funkci.

Jak už bylo zmíněno výše, často řešenou problematikou je rozdělení agentů do předem dané formace a například její udržování v čase. Formace mohou mít různý tvar: trojúhelník, čtverec, hexagon a mnoho dalších.

První část úlohy tak lze popsat jako reformaci všech agentů z počátečních podmínek, které mohou být i náhodné, do předem dané formace. To samozřejmě zabere nějaký čas, jehož doba se odvíjí od komunikační topologie v daném grafu reprezentující předem danou formaci. Dále může být žádoucí udržování této formace v čase, případně může vzniknout potřeba po nějakém čase formaci změnit. Toho můžeme dosáhnout mimo jiné pomocí řízení multiagentního systému stavovou zpětnou vazbou, která je široce používána v mnoha oblastech teorie řízení [1]. Tento přístup zahrnuje měření stavů jednotlivých agentů, které obsahují informace například o pozici, rychlosti nebo i zrychlení jednotlivých agentů v každém časovém okamžiku. Tyto informace pak celý systém pomocí zpětné vazby využívá pro své řízení.

2.1 Stavový popis lineárního systému

Při návrhu řídicích algoritmů pro multiagentní systémy je uvažováno, že tyto systémy jsou lineární t-invariantní (LTI) systémy, tedy časově nezávislé dynamické systémy. Mezi jejichž vlastnosti patří:

1. **Aditivita** — Výstup součtu dvou vstupních signálů se rovná součtu výstupů jednotlivých vstupních signálů zvlášť. $f(x_1 + x_2) = f(x_1) + f(x_2)$, kde $f()$ je tedy lineární funkce.
2. **Homogenita** — Výstup násobku vstupu je rovný násobku výstupu stejného vstupu. $f(\alpha x) = \alpha f(x)$, kde α je libovolná konstanta a $f()$ lineární funkce.

3. **Princip superpozice** — Pokud platí obě výše zmíněné vlastnosti, je řečeno, že platí princip superpozice.
4. **Časová invariance** — Systém nemění své chování v čase. To znamená, že pokud bude do systému puštěn vstup nyní, nebo za T sekund, tak výstup bude v obou případech identický, bude pouze posunut právě o T sekund. Tvar výstupu je dán pouze vstupním signálem, či stavem systému.

Jednou z možností popsání výše zmíněného systému je stavový popis, někdy také označovaný jako stavová reprezentace daného systému. Stavový popis je definován jako soustava lineárních diferenciálních rovnic:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t),\end{aligned}$$

kde $\dot{x}(t)$ je derivace stavu systému v čase, $x(t)$ je stav systému, $u(t)$ je vstup systému a $y(t)$ je výstup systému. Dále jsou zde zastoupeny matice A, B, C, D , kde A představuje matici vnitřních vazeb systému (matice dynamiky), B značí matici vazeb na vstupy (matice řízení), C odpovídá matici výstupu na stav a D je matice, jež popisuje vazby vstupů, které přímo působí na výstup. Tvar matice C určuje, které stavy jsme schopni na výstupu měřit.

2.1.1 Stavová zpětná vazba

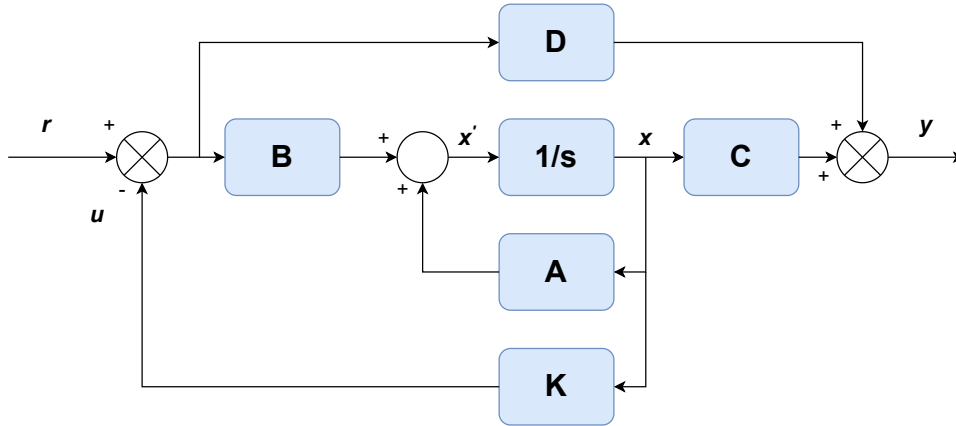
Jak už bylo zmíněno výše, stavová zpětná vazba je jedna z možností řízení multiagentních systémů. Všeobecně v tomto přístupu je uvažováno řízení ve tvaru:

$$u(t) = Kx(t),$$

kde matice K je konstantní zpětnovazební matice, jež představuje nedynamický regulátor, který upravuje dynamiku celého systému. Cílem řízení pomocí stavové zpětné vazby je najít matici K ve tvaru, který splní návrhové požadavky na regulaci.

Dále v této práci bude popsáno, jak navrhnout tuto matici vzhledem k daným vlastnostem multiagentních systémů. Na Obrázku 2 je zobrazeno blokové schéma stavové reprezentace spolu se stavovou zpětnou vazbou ve formě bloku, který obsahuje matici K . Na Obrázku 2 jsou znázorněny i jednotlivé proměnné. Konkrétně r značí referenci, tedy požadovanou hodnotu, nebo požadovaný referenční signál. Dále u je řízení definované vztahem 2.1.1. Proměnná x představuje stavový vektor a \dot{x} je derivace stavového vektoru a y představuje výstup systému.

Mezi další způsoby syntézy a návrhu regulátorů patří například metody založené na principu robustních regionů, které mohou být ještě doplněny analýzou integračních kritérií kvality získané regulační smyčky [12]. Nebo metody zabývající se "fractional-order" (FO), kdy stojí za zmínku typ "fractional-order proportional-integral-derivative" (FOPID). Tento regulátor má více nastavitelných parametrů [23].



Obrázek 2: Stavová zpětná vazba K v blokově reprezentovaném stavovém popisu

2.2 Úvod do grafové teorie

Grafová teorie se zabývá studiem grafů. Graf lze zapsat jako $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, který matematicky popisuje vztahy mezi objekty. V našem případě považujeme objekty jako agenty, které jsou v grafu zachyceny jako vrcholy $v_i \in \mathcal{V}$ a komunikační spoje mezi agenty jako hrany $e_{ij} = (v_i, v_j) \in \mathcal{E}$, které vrcholy spojují. Takže \mathcal{V} představuje sadu vrcholů a \mathcal{E} je sada hran. Tyto hrany mohou být neorientované (komunikace tedy probíhá obousměrně), nebo orientované, v nichž tok dat probíhá jedním směrem. Lze se setkat i s ohodnocenými hranami grafu. Formální rozdíl mezi těmito přístupy bude popsán níže. Grafy s ohodnocenými hranami jsou značeny $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, kde \mathcal{W} reprezentuje množinu všech vah spjatých s množinou hran \mathcal{E} .

V celé této kapitole jsou využívány informace, které jsou čerpány z knihy: "A Textbook of Graph Theory" od R. Balakrishnana a K. Ranganathana [2] a z knihy "Introduction to Graph Theory" od Robina J. Wilsona [20].

2.2.1 Notace - pojmy z grafové teorie

V této podkapitole budou uvedeny nejdůležitější potřebné definice, které budou popsány slovně místo toho, aby byly vyjádřeny přesnými matematickými definicemi. Tyto pojmy budou dále využity v nadcházejících podkapitolách pro pochopení a zasazení do kontextu níže uvedených komplexnějších pojmů.

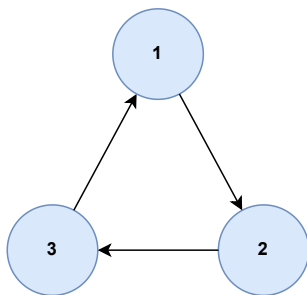
- **Smyčka** — Smyčka v grafu je hrana, která spojuje jeden a ten samý vrchol sám se sebou.
- **Přilehlost** — v_j a v_i jsou přilehlé pokud mezi nimi existuje hrana $e = (v_i, v_j)$ nazýváme je tedy sousedy.
- **Množina přilehlých vrcholů** — \mathcal{N}_i je soubor všech přilehlých sousedících vrcholů daného vrcholu v_i .

- **Stupeň vrcholu** — $deg(v_i)$ je stupeň vrcholu v_i a značí počet výstupních i vstupních hran vrcholu.
- **Vstupní stupeň vrcholu** — $deg_{in}(v_i)$ je stupeň daného vrcholu v_i a značí počet vstupních hran vcházejících do vrcholu.
- **Výstupní stupeň vrcholu** — $deg_{out}(v_i)$ je stupeň daného vrcholu v_i a značí počet výstupních hran vycházejících z tohoto vrcholu.
- **Řád grafu** — Řád grafu \mathcal{G} je počet uzlů grafu \mathcal{G} a označuje se $|\mathcal{G}|$ nebo n .
- **K-periodický graf** — K-periodický graf je takový graf, ve kterém množina délek všech cyklů má společný dělitel $k \leq 1$.

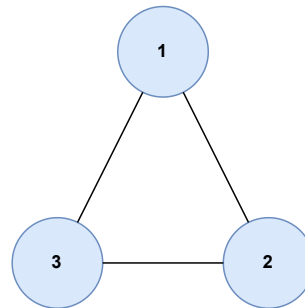
2.3 Definice grafů

Orientovaný graf

Orientovaný graf \mathcal{D} , jinak také digraf se skládá z množiny vrcholů ν a množiny hran $A \subset \nu^2$, kde $a = (\alpha, \beta)$. První element a je označen $tail(a)$ a druhý $head(a)$. Orientovaná hrana tedy vždy vychází právě z $tail(a)$ směrem do $head(a)$. α a β představují dva rozdílné vrcholy. Nadále předpokládáme, že $tail(a) \neq head(a)$ pro všechny elementy, což znamená, že graf neobsahuje žádné vnitřní smyčky ze stejného vrcholu. Příklad orientovaného grafu je vidět na Obrázku 3a.



(a) Příklad orientovaného grafu \mathcal{D}



(b) Příklad neorientovaného grafu \mathcal{G}

Obrázek 3: Příklad dvou typů grafů s rozdílnou orientací hran

Neorientovaný graf

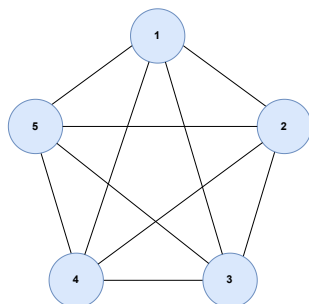
Neorientovaný graf \mathcal{G} je takový graf, který má vlastnost $e_{ij} \in \mathcal{E} \Rightarrow e_{ji} \in \mathcal{E}$. Tedy všechny hrany jsou obousměrné a nemají orientaci. Platí tedy: $deg(v_i) = |\mathcal{N}_i|$, pokud všechny vrcholy v mají stejný stupeň $k = deg(v_i) \forall i$, tak se neorientovaný graf s touto vlastností označuje jako regulární nebo k-regulární. Obrázek 3b zobrazuje ukázkou tohoto grafu.

Kompletní graf

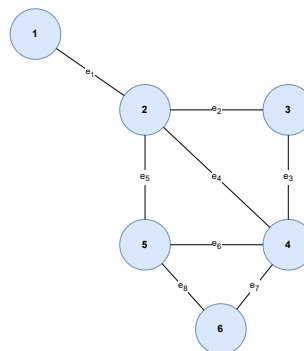
Kompletní grafy jsou grafy, ve kterých jsou všechny uzly spojeny mezi sebou a zároveň je mezi každými sousedícími uzly hrana. Jinými slovy, pokud existuje každá možná hrana mezi uzly, nazýváme graf kompletním a označuje se K_n , kde n značí počet uzlů v celém grafu. Tyto grafy mohou být orientované i neorientované. Příklad kompletního grafu je uveden na Obrázku 4a.

Cesta v grafu

Cesta v grafu, někdy označována také jako sled, je běžně označován jako alternující posloupnost $\mathcal{W} : v_0, e_1, v_1, e_2 \dots e_p, v_p$, kde v_0 značí začátek (origin) sledu a v_p jako konec (terminus) sledu \mathcal{W} . Za předpokladu, že uzly v_0, \dots, v_p jsou rozdílné, jedná se o cestu - sled. V grafové teorii se můžeme setkat i s pojmem nejkratší cesta, která limituje počet uzlů mezi začátkem a koncem na minimum. Na Obrázku 4b je graf obsahující cestu například z vrcholu 1 do vrcholu 6, která je popsána: v_1, v_2, v_4, v_6 .



(a) Příklad kompletního grafu C_5



(b) Příklad možných cest v grafu

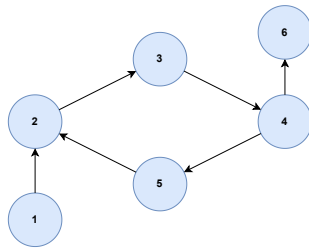
Obrázek 4: Příklad kompletních grafů

Cyklický graf

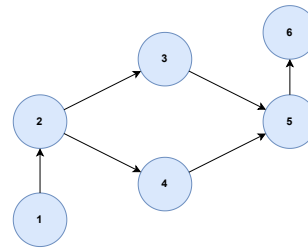
Cyklický graf je takový graf, ve kterém je alespoň jeden uzel, ze kterého vede cesta zpět do stejného začínajícího uzle. Je označen C_n , kde n představuje počet uzlů v grafu. Na Obrázku 5a je možno vidět cyklus C_4 , kterému odpovídá cesta $\mathcal{P} : v_2, v_3, v_4, v_5, v_2$.

Acyklický graf

Acyklický graf je takový graf, který neobsahuje cykly. Tedy neexistuje v grafu ani jediný uzel, do kterého by vedla cesta zpět do sebe sama. Tento graf je ilustrován na Obrázku 5b.



(a) Příklad grafu s cyklem C_4

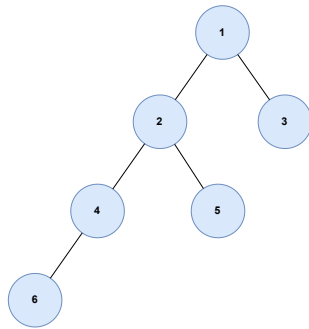


(b) Příklad acyklického grafu

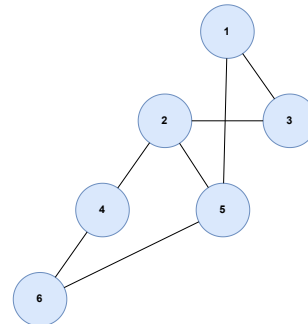
Obrázek 5: Příklad dvou grafů, které ukazují rozdíl mezi cyklickým a acyklickým grafem

Konexe neorientovaných grafů

Neorientovaný graf \mathcal{G} je souvislým grafem, pokud existuje cesta \mathcal{P} mezi každou rozdílnou dvojicí uzlů grafu \mathcal{G} . Délka nejdelší cesty v souvislém grafu \mathcal{G} je nazvána průměr. Příkladem souvislého acyklického grafu je strom, jak je vidět z Obrázku 6a. Ukázka všeobecného souvislého neorientovaného grafu \mathcal{G} je zobrazena na Obrázku 6b.



(a) Příklad souvislého neorientovaného grafu \mathcal{G} , který představuje strom

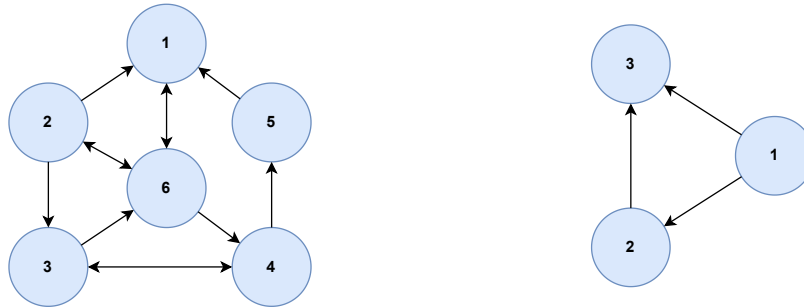


(b) Příklad souvislého neorientovaného grafu \mathcal{G}

Obrázek 6: Příklady souvislých neorientovaných grafů

Konexe orientovaných grafů

Orientovaný graf, tedy digraf \mathcal{D} je silně souvislý, pokud existuje orientovaná cesta \mathcal{P} mezi každou rozdílnou dvojicí uzlů grafu \mathcal{D} . Tento graf je ukázán na Obrázku 7a. Alternativně je digraf slabě souvislý pokud existuje neorientovaná cesta mezi každou rozdílnou dvojicí uzlů grafu \mathcal{D} . Jinak řečeno, je možné dosáhnout jakéhokoliv uzlu z libovolného startovacího uzlu změnou směru nějaké hrany, to znamená, že nemusíme nutně dodržovat orientace daných hran. Slabě souvislý orientovaný graf je demonstrován na Obrázku 7b.



(a) Příklad silně souvislého orientovaného grafu \mathcal{D} (b) Příklad slabě souvislého orientovaného grafu \mathcal{D}

Obrázek 7: Příklady souvislých orientovaných grafů

2.4 Grafy a matice

Nyní je na místě otázka, jak by bylo možné shrnout vlastnosti grafů do nějaké podoby, která by byla snadno použitelná. Vlastnosti grafů mohou být shrnuty do speciálních matic, které budou uvedeny níže.

Nadcházející sekce má za úkol sjednotit pojmy z grafové a maticové teorie k jejich vysvětlení a použití v dalších kapitolách, přičemž je čerpáno z knih "Graph Theoretic Methods in Multiagent Networks" [15] a "Matrix Analysis" od Rogera A. Horna a Charlese R. Johnsona [7]. Pro téměř všechny následující příklady matic budou použity dva příklady orientovaných a neorientovaných grafů pro názorné ukázky z Obrázku 8, kde na Obrázku 8a je znázorněn neorientovaný graf a naopak na Obrázku 8b je vykreslena orientovaná varianta.



(a) Příklad neorientovaného grafu \mathcal{G} oktagonové topologie (b) Příklad orientovaného grafu \mathcal{D} oktagonové topologie

Obrázek 8: Příklady grafových topologií

2.4.1 Notace - pojmy z algebraické grafové teorie

V této podkapitole budou prezentovány klíčové definice z algebraické grafové teorie, jejichž vysvětlení je poměrně stručné na to, aby bylo uvedeno v samostatné sekci. Týkají

se výhradně matic a jejich vlastností. Tyto definice a pojmy jsou potřebné v nadcházejících podsekcích pro pochopení složitějších pojmů a myšlenek z oblasti multiagentních systémů.

- **Vlastní čísla matice** — Vlastní čísla matice A jsou kořeny charakteristického polynomu $\det(\lambda I - A) = 0$, kde číslo λ je vlastní číslo matice A , I je jednotková matice a $\det()$ značí výpočet determinantu.
- **Vlastní vektor** — Vlastní vektor u matice A je takový vektor, pro který existuje nějaké charakteristické číslo λ tak, že splňuje: $Au = \lambda u$.
- **Čtvercová matice** — Matice se nazývá čtvercová, pokud má stejný počet n řádků a sloupců.
- **Symetrická matice** — Čtvercová matice $A \in R^{n \times n}$ je symetrická pokud platí $A = A^T$.
- **Pozitivně semidefinitní matice** — Symetrická matice A je pozitivně semidefinitní, pokud platí pro A a libovolné x : $x^T Ax \geq 0$.
- **Pozitivně definitní matice** — Symetrická matice A je pozitivně definitní, pokud platí pro A a libovolné x : $x^T Ax > 0$.
- **Reálná symetrická matice** — Reálná symetrická matice je taková matice, která má všechny vlastní čísla reálná.
- **Nulový prostor matice** — Nulový prostor matice A , někdy také jádro matice A , je množina všech vektorů x , pro které platí $Ax^T = o^T$, kde o^T je transponovaný nulový vektor. Je značen jako $N(A)$ nebo též $\ker(A)$.

Matice sousednosti

Matice sousednosti grafu je čtvercová matice o rozměru $R^{n \times n}$, kde n je počet vrcholů grafu, neboli jeho řád. Řádky a sloupce jsou značeny uzly grafu a na odpovídajících pozicích (v_i, v_j) se bude nacházet hodnota jedna, nebo nula v závislosti na tom, zda mezi vrcholy v_i a v_j vede hrana. U orientovaných grafů je nutno brát v potaz, odkud a kam vede hrana, což určuje pozici hodnoty jedna v matici. Pro všechny jednoduché grafy, které neobsahují vlastní smyčky, bude mít matice sousednosti na diagonále nuly. Pro neorientované grafy bude matice sousednosti symetrická.

Matice sousednosti grafu značená $A(\mathcal{G})$ je definována následujícím předpisem:

$$A_{ij} = \begin{cases} 1, & (v_i, v_j) \in \mathcal{G}, \\ 0, & \text{jinak.} \end{cases}$$

Pro názornou ukázkou se budeme držet značení $A(\mathcal{G})$ pro neorientovaný graf z Obrázku 8a a $A(\mathcal{D})$ pro orientovaný graf z Obrázku 8b. Nyní je možno ukázat výsledný tvar matic sousednosti:

$$A(\mathcal{G}) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad A(\mathcal{D}) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Zde je vidět, že matice $A(\mathcal{G})$ je symetrická.

Normalizovaná matice sousednosti

Normalizovaná matice sousednosti je čtvercová matice o rozměru $R^{n \times n}$, kde n je počet vrcholů grafu. Stejně jako u matice sousednosti jsou řádky a sloupce značeny podle uzlů v grafu. Značí se $H(\mathcal{G})$. Hlavním rozdílem mezi maticí sousednosti a její normalizovanou variantou je její předpis:

$$H_{ij} = \begin{cases} \frac{1}{d_0(v_i)}, & (v_i, v_j) \in \mathcal{G}, \\ 0, & \text{jinak.} \end{cases}$$

Jedna z hlavních vlastností této normalizované matice sousednosti je, že součet každého řádku je jedna. $d_0(v_i)$ představuje stupeň uzlu v_i , tedy počet všech sousedících vrcholů s vrcholem v_i . Stejně jako u normalizované matice jsou uvedeny příklady normalizovaných matic $H(\mathcal{D})$ pro orientovaných graf na Obrázku 8b a $H(\mathcal{G})$ pro neorientovaný graf na Obrázku 8a:

$$H(\mathcal{D}) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad H(\mathcal{G}) = \begin{bmatrix} 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \end{bmatrix}.$$

Alternativně je možno vyjádřit normalizovanou maticí sousednosti jako: $H = \frac{A}{D}$, kde A je matice sousednosti a D je stupňová matice, která bude definována níže.

Stupňová matice

Stupňová matice, někdy též matice stupňů je značena $D(\mathcal{G})$. Je to diagonální matice o rozměrech $R^{n \times n}$, kde n je počet vrcholů grafu, a její diagonální prvky jsou rovny $d(v_i)$ stupňům odpovídajících uzlů:

$$D = \begin{bmatrix} d(v_1) & 0 & \cdots & 0 \\ 0 & d(v_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d(v_n) \end{bmatrix}.$$

Pro úplnost jsou dále uvedeny numerické příklady stupňových matic $D(\mathcal{D})$ pro orientovaných graf na Obrázku 8b a $D(\mathcal{G})$ pro neorientovaný graf na Obrázku 8a:

$$D(\mathcal{D}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad D(\mathcal{G}) = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}.$$

Laplacián grafu

Další důležitým pojmem v grafové teorii je reprezentace grafu pomocí Laplaciánu grafu. Laplacián je další matice, která zachycuje důležité vlastnosti grafu, které jsou obsaženy v matici sousednosti a stupňové matici. Matice může být definovaná několika způsoby, nicméně výsledek je vždy stejný. Jeden ze způsobů definice je následující:

$$L = D - A, \tag{1}$$

kde D je stupňová matice a A je matice sousednosti. Důležitou vlastností je, že součet všech prvků v řádku se rovná nule. Dále jsou uvedeny numerické příklady pro referenční topologie z Obrázku 8b $L(\mathcal{D})$ pro orientovaných graf a na Obrázku 8a $L(\mathcal{G})$ pro neorientovaný graf:

$$L(\mathcal{D}) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$L(\mathcal{G}) = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}.$$

Dalším způsobem výpočtu Laplaciánu je:

$$L = I - H,$$

kde I je jednotková matice, tedy matice, která má prvky na diagonále rovno jedné a H značí normalizovanou matici susednosti. V tomto způsobu se projeví vlastnosti normalizované matice susednosti, kde jsou použity převážené hodnoty, nicméně pořád platí vlastnost, že součet prvků v řádku se rovná nule.

$$L(\mathcal{D}) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$L(\mathcal{G}) = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 1 \end{bmatrix}$$

Vlastnosti Laplaciánu pro neorientované grafy

Laplacián je pro neorientované grafy symetrický a reálný. To vychází z jeho definice $L = D - A$, kde stupňová matice a matice susednosti jsou vždy u neorientovaného grafu, vlivem jeho konstrukce také symetrické a reálné. Tedy i Laplacián získá tyto vlastnosti.

Laplacián je pozitivně semidefinitní, pokud je matice A pozitivně definitní, tudíž jsou vlastní čísla L reálná a seřaditelná a zároveň jedno z jeho vlastních čísel je nulové. Toto

tvrzení lze zapsat matematicky: $0 = \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, kde n je počet všech vlastních čísel L .

Jak už bylo výše zmíněno, suma každého řádku Laplaciánu je rovna nule. Z toho vyplývá, že matice obsahuje minimálně jeden řádek, který je lineárně závislý a tedy: $L1c = 0$, kde je vektor $1 = [1, \dots, 1]^T$ a c je libovolná konstanta, která splňuje $1c \in N(L)$.

Kroneckerův součin matic

Kroneckerův součin, nebo též tenzorový součin je značen \otimes . Je to operace prováděná na dvou maticích nebo vektorech s libovolnými rozměry, jejichž výsledek vede na blokovou matici. Nechť A je matice o rozměru $n \times m$ a B je matice o rozměru $p \times q$ potom je výsledný tvar matice následující:

$$A \otimes B = (a_{ij} \cdot B) = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ a_{21}B & a_{22}B & \cdots & a_{2m}B \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1}B & a_{n2}B & \cdots & a_{nm}B \end{bmatrix}.$$

Výsledná matice má velikost $np \times mq$. Jako příklad je možno uvést matice $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ a matici $B = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$ jejich výsledná matice Kroneckerova součinu bude mít rozměr 4×4 .

$$A \otimes B = \begin{bmatrix} 1 & \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} \\ 3 & \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 4 & 3 & 8 & 6 \\ 2 & 1 & 4 & 2 \\ 12 & 9 & 16 & 12 \\ 6 & 3 & 8 & 4 \end{bmatrix}$$

Kroneckerův součin má spoustu aplikací v multiagentních systémech, specificky ve tvorbě a řízení formací agentů, které budou řešeny dále v této práci.

3 Algoritmus konsensu (shody)

V kontextu multiagentních systémů se pojem konsensus používá pro označení shody, mezi jednotlivými agenty, na určitém cíli nebo na společné proměnné hodnotě. Pro potřeby této práce se uvažují stavové proměnné, jinak také jako informační stavy x_i , každého uzlu v_i reprezentujícího daného agenta i . Je řečeno, že pouze pokud všechny hodnoty x_i dosáhnou stejné hodnoty, tak poté agenti dosáhnou konsensu [1]. Aby agenti mohli dosáhnout konsensu, je potřeba definovat algoritmus pro nalezení konsensu. Ten tak musí obsahovat množinu pravidel pro komunikaci sousedících agentů a možnost aktualizace informačních stavů agentů v závislosti na stavech přilehlých sousedů. Základní dva principy řízení využívající se v multiagentních systémech mohou být rozděleny do dvou hlavních skupin a to: konsensus s leadrem a bez leadra. Jednotlivé algoritmy budou rozebírány v práci dále, nicméně jako první bude uveden příklad konsensus algoritmus pomocí Laplaciánu, kde bude uvažován stavový popis jednotlivých agentů jako systém prvního řádu.

3.1 Systém prvního řádu

V této sekci uvažujeme, že všichni agenti mají dynamiku prvního řádu, tedy integrátoru ve tvaru:

$$\dot{x}_i(t) = u_i(t),$$

kde $\dot{x}_i(t)$ je derivace funkce informačního stavu, za který je považována právě pozice i -tého agenta a $u_i(t)$ funkce řízení, jež je nutné specifikovat. Nadále pro zjednodušení budou označovány jako \dot{x}_i a u_i . Řízení u_i je definováno pomocí lokálního protokolu pro řízení jednotlivých agentů následujícím způsobem:

$$u_i = \sum_{j=1}^N a_{ij}(x_j - x_i), \quad i = 1, \dots, n, \quad (2)$$

kde a_{ij} je prvek matice sousednosti A , který určuje váhu odpovídající hrany v grafu \mathcal{G} spojující vrcholy v_j a v_i , pro které odpovídají právě informační stavy x_j a x_i . Tento protokol umožní dosáhnout shody všech agentů v závislosti na topologii grafu. Z výše zmíněné rovnice $\dot{x}_i(t) = u_i(t)$ je možné tedy vyjádřit dynamiku každého i -tého uzlu následovně:

$$\begin{aligned} \dot{x}_i &= \sum_{j=1}^N a_{ij}(x_j - x_i), \\ \dot{x}_i &= -x_i \sum_{j=1}^N a_{ij} + \sum_{j=1}^N a_{ij}x_j, \\ \dot{x}_i &= -d_i x_i + [a_{i1} \quad \cdots \quad a_{in}] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \end{aligned}$$

kde d_i jsou diagonální prvky stupňové matice D a vektor $[x_1 \ \cdots \ x_n]^T$ je zaveden jako stavový vektor. Výslednou dynamiku celého systému je tedy možno zapsat v maticovém tvaru:

$$\begin{aligned}\dot{x} &= -Dx + Ax = -(D - A)x, \\ \dot{x} &= -(D - A)x, \\ \dot{x} &= -Lx.\end{aligned}\tag{3}$$

Tato rovnice popisuje dynamiku uzavřeného systému, pro který je definováno řízení $u = -Lx$. V těchto rovnicích hraje hlavní roli matice Laplaciánu L , obzvláště je zásadní vliv jeho vlastních čísel. Z Rovnice 3 vyplývá, že určují stabilitu a dynamiku celého systému. Jak už bylo výše zmíněno pro neorientované grafy, má matice L nezáporná vlastní čísla, tedy jsou všechny situovány v uzavřené pravé komplexní polorovině na reálné ose s jedním vlastním číslem v nule. Tedy pokud uvažujeme Laplacián přenásobený hodnotou mínus jedna: $-L$, potom se jeho vlastní čísla budou nacházet v levé uzavřené komplexní polorovině s jedním vlastním číslem v nule, čímž se stane systém stabilní, respektive na mezi stability.

Jelikož má matice L jeden lineárně závislý řádek, pak můžeme vyjádřit jeho ustálený stav $0 = -Lx_{\text{ustálený}}$ pomocí vektoru $1c$, který se nachází v nulovém prostoru matice L jako:

$$x_{\text{ustálený}} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = 1c = \begin{bmatrix} c \\ \vdots \\ c \end{bmatrix},\tag{4}$$

kde všechny proměnné jsou konstanty $v_j = v_i = c \ \forall i, j \in n$. V případě splnění této podmínky je možné tvrdit, že agenti dosáhli konsensu na hodnotě c . Tuto skalární veličinu c lze popsat rovnicí:

$$c = \sum_{j=1}^N p_j x_j(0),\tag{5}$$

kde $w_1 = [p_1 \ \cdots \ p_n]^T$ je normalizovaný levý vlastní vektor Laplaciánu L nulového vlastního čísla λ_1 , ve kterém jsou obsaženy p_i , $x_i(0)$ jsou počáteční stavy agentů. Dále je řečeno, že druhé vlastní číslo λ_2 ovlivňuje rychlost dosažení konsensu [14] s časovou konstantou:

$$\tau = 1/\lambda_2.\tag{6}$$

Jinými slovy, čím větší je druhé vlastní číslo matice L , tím rychleji dojde ke konsensu.

3.2 Řízení s leadrem

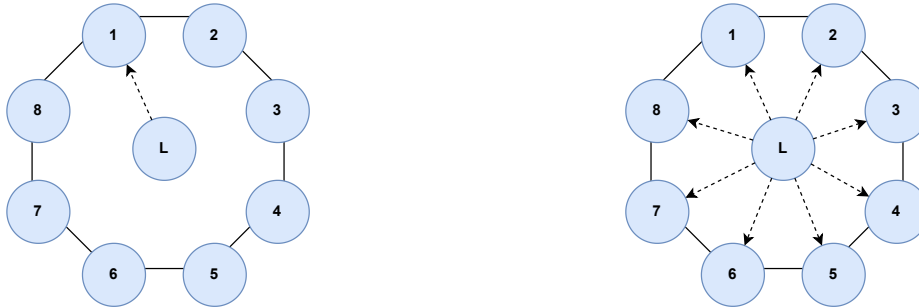
Pojem řízení s leadrem znamená, že referenční chování celého systému je určováno právě jedním agentem a ten se označuje jako leader. Tento leader musí být připojen k ostatním

agentům alespoň jednou orientovanou hranou tak, že existuje cesta od leadra ke každému agentu. Na základě takto definované komunikační topologie propojení leadra se zbytkem systému se informace o kýžené referenci předávají ke všem ostatním agentům, kdy agenti se této referenci přizpůsobují a reagují na ni. Zároveň platí, že ostatní agenti nemohou ovlivnit chování vedoucího agenta z důvodu jednostranné komunikace orientovanou hranou od leadra. V této práci budou rozlišovány dvě varianty řízení s leadrem, a to jsou: virtuální leader a externí leader.

Virtuální leader

Virtuální leader je takový agent, který není součástí systému, ale je pouze programový (virtuální) prvek, který vede ostatní agenty. Nicméně je ho možno vidět jako aditivní uzel v daných topologiích, které jsou jako příklad zobrazeny na Obrázku 9.

Nadále budou tedy uvažovány dvě varianty a to: připojení leadra jednou orientovanou hranou na Obrázku 9a a připojení leadra orientovanými hranami s každým agentem, zobrazené na Obrázku 9b. U tohoto typu bude lehce modifikováno řízení vzhledem k jeho komunikační topologii.



(a) Virtuální leader spojen jednou orientovanou hranou se zbytkem topologie (b) Virtuální leader spojen orientovanými hranami s každým uzlem

Obrázek 9: Příklady možných spojení virtuálního leadra se zbytkem topologie

V obou těchto variantách je uvažována dynamika všech agentů ve formě systému prvního řádu. A chování leadra je popsána rovnicí:

$$\dot{x}_L(t) = u_{ref}(t),$$

kde $u_{ref}(t)$ představuje požadované referenční řízení v čase. Pro uniformitu chování ve všech příkladech je uvažováno, že $u_{ref} = 0$.

Pro integraci řízení leadra do skupiny je zapotřebí modifikovat rovnici lokálního řízení následujícím způsobem:

$$u_i = \dot{x}_L + k_p(x_L - x_i) + c \sum_{j=1}^N a_{ij}(x_j - x_i),$$

kde k_p a c jsou návrhové parametry řízení s virtuálním leadrem. Parametr k_p představuje informaci o existenci orientované hrany od leadra k agentovi, nabývá tedy hodnot $k_p \in [0, 1]$. Hodnota c ovlivňuje rychlost dosažení konsensu a nabývá hodnot $c > 0$, aby si systém zachoval stabilitu, což bude očividné z tvaru níže uvedených rovnic. Tento parametr se někdy může označovat také jako konstanta učení.

Aby systém byl schopen dosáhnout konsensu, je potřeba upravit zákon řízení tak, aby $(x_L - x_i) = 0$. Toho je docíleno zavedením upraveného stavu i -tého agenta $\tilde{x}_i = x_i - x_L$, jenž je dále upraven:

$$\begin{aligned}\dot{\tilde{x}}_i &= \dot{x}_i - \dot{x}_L, \\ \dot{\tilde{x}}_i &= u_i - \dot{x}_L, \\ \dot{\tilde{x}}_i &= \dot{x}_L + k_p(x_L - x_i) + c \sum_{j=1}^N a_{ij}(x_j - x_i) - \dot{x}_L, \\ \dot{\tilde{x}}_i &= -k_p \tilde{x}_i + c \sum_{j=1}^N a_{ij}(\tilde{x}_j - \tilde{x}_i).\end{aligned}\tag{7}$$

Tuto rovnici je možno přepsat do maticového tvaru vzhledem k faktu, že člen $\sum_{j=1}^N a_{ij}(x_j - x_i)$ odpovídá matici Laplaciánu $-L$ a je možno tedy zapsat vztah:

$$\begin{aligned}\dot{\tilde{x}} &= [-P \otimes k_p - cL]\tilde{x}, \\ \dot{\tilde{x}} &= -[P \otimes k_p + cL]\tilde{x},\end{aligned}\tag{8}$$

kde nově figuruje diagonální matice P , která má na své diagonále hodnoty 0 a 1 v závislosti na spojení leadra s daným agentem. Tyto hodnoty odpovídají poslednímu řádku matice sousednosti A v topologii řízení s virtuálním leadrem, a definují tak propojenost leadra se zbytkem grafu v kontextu řízení. V našem druhém případě na Obrázku 9b je propojen leader se všemi ostatními agenty, tudíž je možno vynechat parametr k_p a matici P z rovnice, jelikož bude vždycky nabývat hodnot jedna. Je tedy na místě upravit Rovnici 7 do tvaru:

$$\dot{\tilde{x}}_i = -\tilde{x}_i + c \sum_{j=1}^N a_{ij}(\tilde{x}_j - \tilde{x}_i).$$

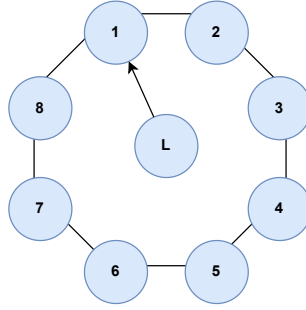
A stejně tak modifikujeme maticovou Rovnici 8 do tvaru:

$$\dot{\tilde{x}} = -[I + cL]\tilde{x},\tag{9}$$

kde I je diagonální jedničková matice.

Externí leader

Alternativou pro virtuálního leadra je externí leader, který je na rozdíl od minulého příkladu reálný hardwarový leader, jenž je součástí celého systému. V principu je tedy vybrán jeden nebo více agentů ze skupiny, který je jmenován leadrem, čímž je chápáno, že jeho chování reprezentuje požadované chování celého systému na základě externí referenční informace. Někdy se může jednat i o zcela nového agenta, který se v systému fyzicky nenacházel a byl tam přidán, aby reprezentoval leadra. Dále je uvažována topologie spojení leadra na Obrázku 10.



Obrázek 10: Externí leader spojen jednou orientovanou hranou se zbytkem topologie

Jako v předchozí sekci je uvažována dynamika všech agentů ve formě systému prvního řádu. A chování leadra je popsána rovnicí:

$$\dot{x}_L(t) = u_{ref}(t).$$

Předpis pro rovnici lokálního řízení je definován jako:

$$u_i = \dot{x}_L + c \sum_{j=1}^N a_{ij}(x_j - x_i) + g_i(x_L - x_i), \quad (10)$$

kde g_i je prvek, který nabývá hodnot $g_i \in [0, 1]$ podle toho, zda existuje spojení s agenta s leadrem. Po zavedení upraveného stavu $\tilde{x}_i = x_i - x_L$ získáváme:

$$\begin{aligned} \dot{\tilde{x}}_i &= \dot{x}_i - \dot{x}_L, \\ \dot{\tilde{x}}_i &= u_i - \dot{x}_L, \\ \dot{\tilde{x}}_i &= \dot{x}_L + c \sum_{j=1}^N a_{ij}(x_j - x_i) + g_i(x_L - x_i) - \dot{x}_L, \\ \dot{\tilde{x}}_i &= c \sum_{j=1}^N a_{ij}(\tilde{x}_j - \tilde{x}_i) - g_i \tilde{x}_i. \end{aligned} \quad (11)$$

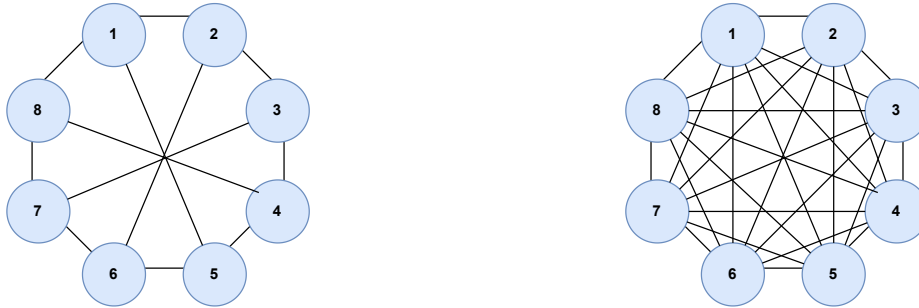
Parametr g_i je možné zapsat jako diagonální matici G s příslušnými hodnotami g_i právě na své diagonále. Jako v předchozí sekci můžeme přepsat Rovnici 11 do maticové podoby:

$$\dot{\tilde{x}} = -c[G + L]\tilde{x}. \quad (12)$$

3.3 Vlivy topologie na chování konsensu

Vliv provázanosti grafu na rychlost dosažení konsensu

Velikost vlastních čísel je přímo spojena s topologií grafu, specificky s jeho provázaností. Nyní bude ukázáno na třech příkladech z Obrázku 8a pro málo provázaný graf \mathcal{G}_1 a z Obrázku 11 pro středně a hodně provázané grafy \mathcal{G}_2 a \mathcal{G}_3 . Zejména se jedná o ukázkou vlivu jejich topologie na vlastní čísla Laplaciánu a v přeneseném pohledu i na rychlost dosažení konsensu.



(a) Příklad středně provázaného neorientovaného grafu \mathcal{G}_2 (b) Příklad hodně provázaného neorientovaného grafu \mathcal{G}_3

Obrázek 11: Příklady různě provázaných neorientovaných grafů

Matice Laplaciánu pro tyto typologie byly vypočítány jako:

$$L(\mathcal{G}_1) = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix},$$

$$L(\mathcal{G}_2) = \begin{bmatrix} 3 & -1 & 0 & 0 & -1 & 0 & 0 & -1 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 3 & -1 & 0 & 0 & -1 \\ -1 & 0 & 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & -1 & 0 & 0 & -1 & 3 \end{bmatrix},$$

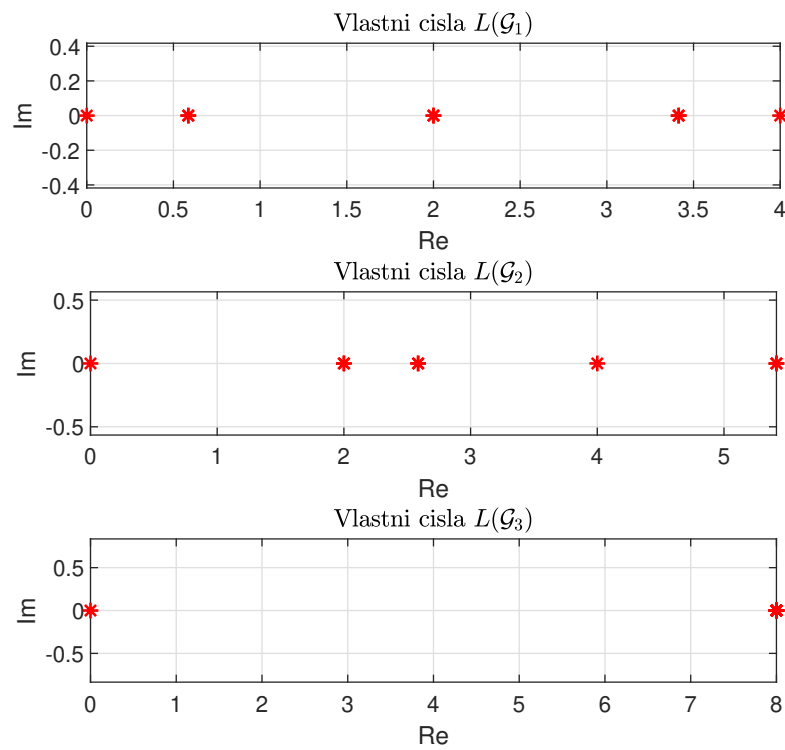
$$L(\mathcal{G}_3) = \begin{bmatrix} 7 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 7 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 7 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 7 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 7 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 7 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 7 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & 7 \end{bmatrix}.$$

Pro vypočtení vlastních čísel a následné jejich zobrazení na Obrázku 12 byl použit Matlab, ze kterého jsme získali následující hodnoty vlastních čísel:

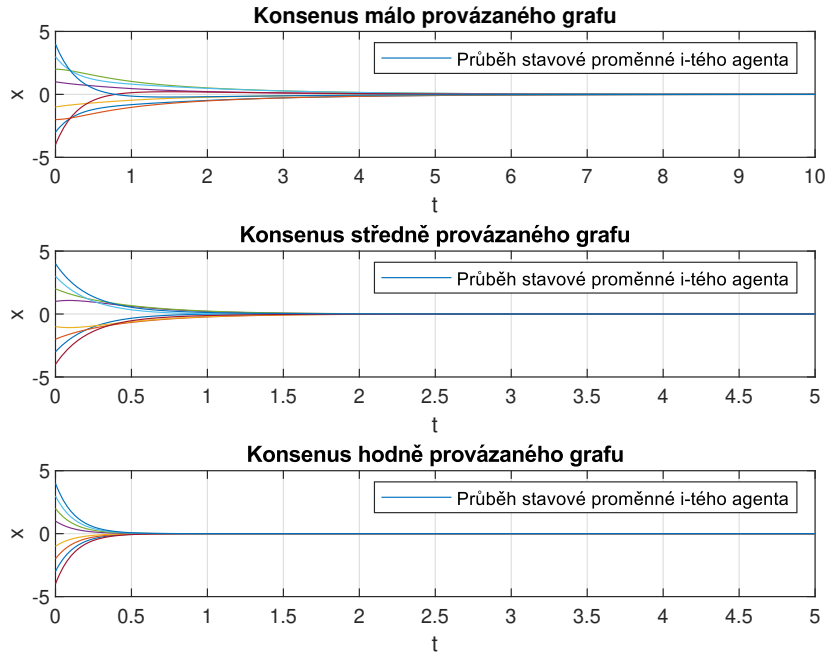
Tabulka 1: Vlastní čísla matic $L(\mathcal{G}_i)$

i	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8
1:	0	0.5858	0.5858	2	2	3.4142	3.4142	4
2:	0	2	2	2.5858	2.5858	4	5.4142	5.4142
3:	0	8	8	8	8	8	8	8

Z hodnot vlastních čísel je vidět, že čím více je provázaný graf, tím větší jsou vlastní čísla Laplaceanu daného grafu.



Obrázek 12: Vlastní čísla Laplaceanu $L(\mathcal{G}_i)$



Obrázek 13: Ukázka dosažení konsensu pro tři topologie \mathcal{G}_i

Dalším krokem je samotná simulace dosažení konsensu, kde ve všech případech uvažujeme počáteční stavy agentů vektorem $x_i(0) = [-3 \ -2 \ -1 \ 1 \ 2 \ 3 \ -4 \ 4]^T$. Dle Rovnice 5 je možno předem vypočítat hodnotu c konsensu, která vyšla s menší numerickou chybou danou výpočtem levého vektoru vlastních čísel jako $c \cong 0$.

Dle Rovnice 6 byly vypočítané časové konstanty τ_i , kterým vyšly výsledky $\tau_1 = 1.7071$, $\tau_2 = 0.5$ a $\tau_3 = 0.125$, přičemž časy dosažení konsensu z Obrázku 13 odpovídají hodnotám $t_1 \approx 7.951 \text{ s}$, $t_2 \approx 2.448 \text{ s}$ a $t_3 \approx 0.9385 \text{ s}$. Z toho tedy vyplývá fakt, že čím je časová konstanta menší, tím potřebuje systém méně času pro dosažení konsensu. Dále je vidět, že ve všech případech došlo k dosažení konsensu na hodnotě nula podle předpokladů.

Závěrem bylo simulačně ověřeno, že čím je víc neorientovaný graf provázaný, tím menší mají časovou konstantu τ agenti, kteří jsou více informačně provázáni, a díky tomu dosáhnou konsensu v kratším čase. Výsledky simulací jsou zobrazeny na Obrázku 13.

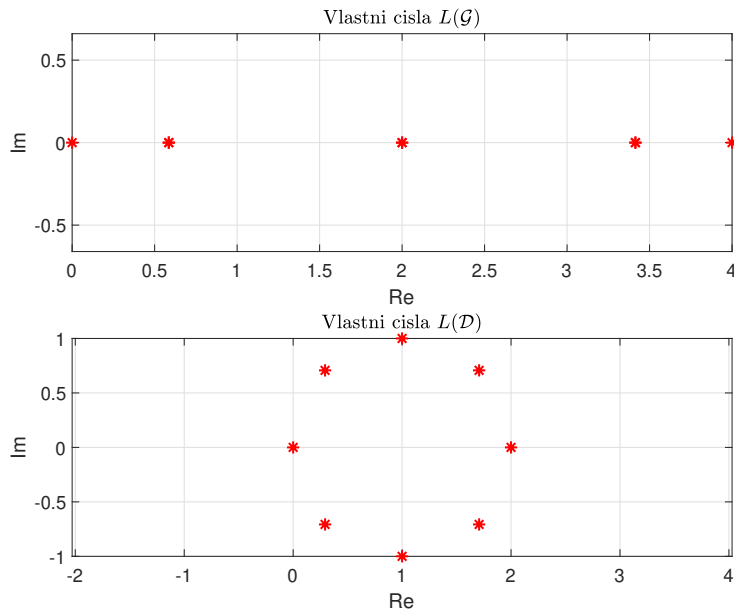
Rozdíl mezi orientovaným a neorientovaným grafem pro dosažení shody

V této sekci budou uvažovány referenční grafové topologie vyobrazené na Obrázku 8. Cílem je zjistit, jaký bude mít vliv topologie orientovaného grafu \mathcal{D} 3a na vlastní čísla Laplaciánu a chování systému při simulaci konsensu. Pro porovnání bude využit neorientovaný graf \mathcal{G} 3b. Matice Laplaciánu $L(\mathcal{G})$ a $L(\mathcal{D})$ jsou vypočítány výše v sekci 2.4.1. Vlastní čísla matic byly vypočítány a zobrazeny do Tabulky 4 a grafu 14.

Tabulka 2: Vlastní čísla matic $L(\mathcal{G})$ a $L(\mathcal{D})$

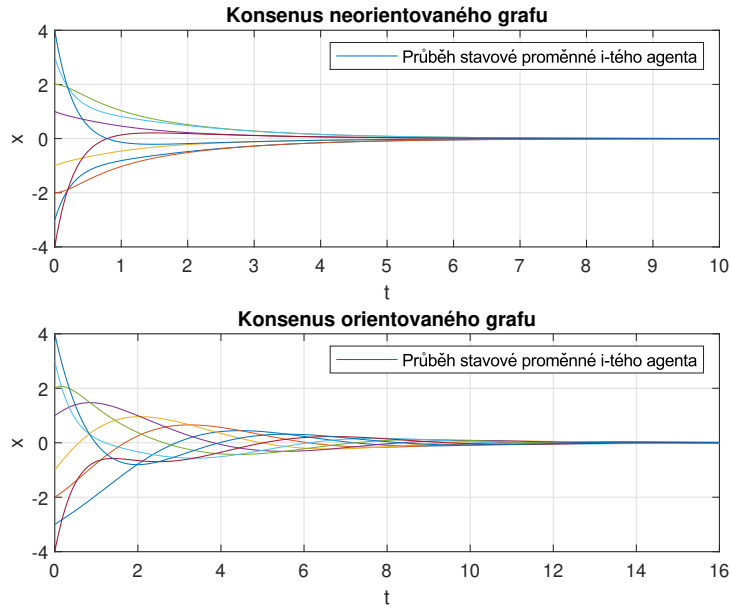
i	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8
1:	0	0.5858	0.5858	2	2	3.4142	3.4142	4
2:	0	$0.2929 + 0.7j$	$0.2929 - 0.7j$	$1 + 1j$	$1 - 1j$	2	$1.7071 + 0.7j$	$1.7071 - 0.7j$

Zde je očividné, že vlastní čísla Laplaciánu neorientovaného grafu mohou být i komplexní čísla. Tento fakt potom ovlivňuje chování systému, které je zobrazeno na základě simulace na Obrázku 15. Dále je možné si povšimnout, že vlastní čísla $L(\mathcal{D})$ se nachází na jedničkovém kruhu v pravé komplexní polorovině se středem v $[1, 0]$. Na níže vyobrazeném



Obrázek 14: Vlastní čísla Laplaciánu \mathcal{G} a \mathcal{D}

grafu 15 je vidět, že agenti propojení pouze orientovanými hranami dosáhnou konsensu přibližně v čase $t \approx 16$ s, oproti tomu systém daný neorientovaným grafem dosáhl konsensu v $t \approx 8$ s, což je přibližně dvakrát rychleji. Další zajímavé jevy, které lze pozorovat na Obrázku 15, jsou překmity a podkmity trajektorií jednotlivých agentů. Tento charakter chování je dán komplexními vlastními čísly Laplaciánu $L(\mathcal{D})$.



Obrázek 15: Ukázka dosažení konsensu pro neorientovaný a orientovaný graf

Vliv spojení externích a virtuálních leaderů a návrhového parametru c na dosažení konsensu

V předchozí sekci byl ukázán a simulačně ověřen vliv provázanosti grafu na tvar vlastních čísel Laplaciánu a jejich vztah pro rychlost dosažení konsensu. V této sekci je cílem ověřit, zda se stejné vlivy aplikují i na řízení s leadry a popřípadě jaké další okolnosti zde hrají roli.

Jako první příklady simulace dosažení konsensu byly uvažovány topologie z Obrázku 9. Pro Obrázek 9a jsou do Rovnice 8 numericky dosazeny konkrétní hodnoty pro danou topologii. Tímto dosazením získáme následující tvar dynamiky systému:

$$\begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \\ \dot{\tilde{x}}_3 \\ \dot{\tilde{x}}_4 \\ \dot{\tilde{x}}_5 \\ \dot{\tilde{x}}_6 \\ \dot{\tilde{x}}_7 \\ \dot{\tilde{x}}_8 \\ \dot{\tilde{x}}_L \end{bmatrix} = P - c \begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \\ \tilde{x}_5 \\ \tilde{x}_6 \\ \tilde{x}_7 \\ \tilde{x}_8 \\ \tilde{x}_L \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ u_{ref} \end{bmatrix}.$$

Obdobným způsobem je vyjádřena dynamika systému s leadrem propojeným se všemi

agenty, jejichž topologii lze vidět na Obrázku 9b.

$$\begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \\ \dot{\tilde{x}}_3 \\ \dot{\tilde{x}}_4 \\ \dot{\tilde{x}}_5 \\ \dot{\tilde{x}}_6 \\ \dot{\tilde{x}}_7 \\ \dot{\tilde{x}}_8 \\ \dot{\tilde{x}}_L \end{bmatrix} = I - c \begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 3 & -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 3 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 3 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 3 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \\ \tilde{x}_5 \\ \tilde{x}_6 \\ \tilde{x}_7 \\ \tilde{x}_8 \\ \tilde{x}_L \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ u_{ref} \end{bmatrix}.$$

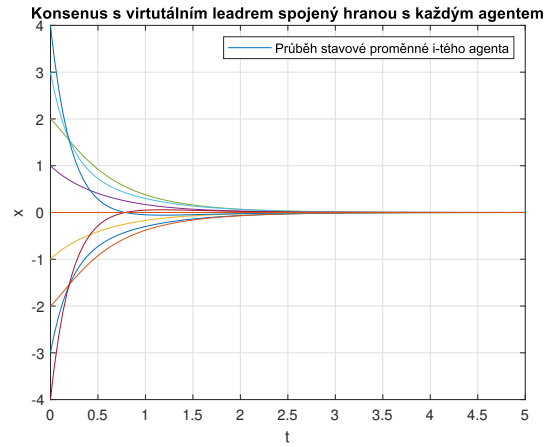
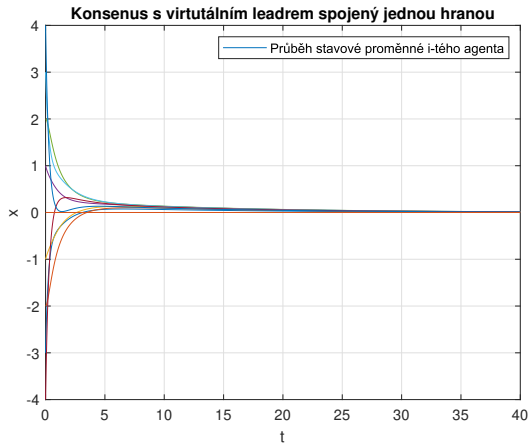
Po dopočítání celých matic s uvažovaným parametrem $c = 1$ a $c = 5$ jsou získány následující hodnoty vlastních čísel:

Tabulka 3: Vlastní čísla matic $(P + cL)$ a $(I + cL)$

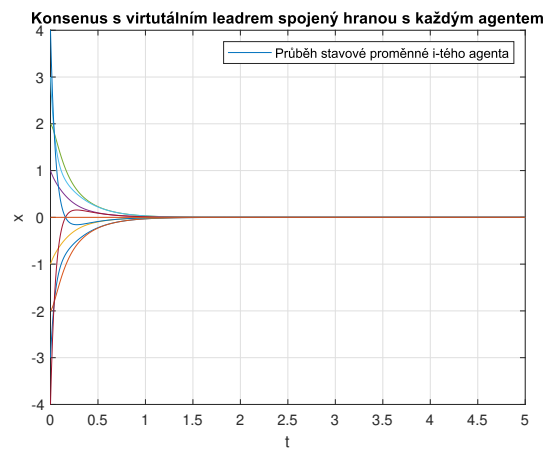
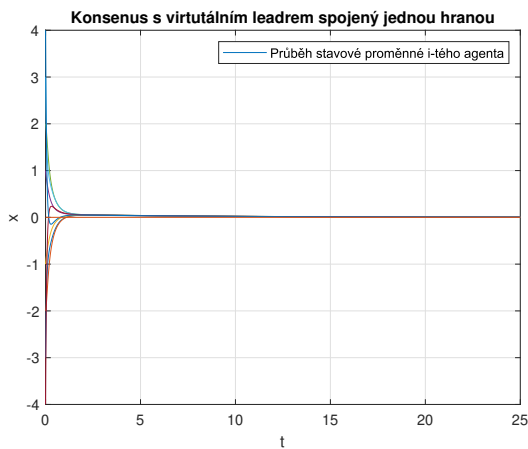
i	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9
$(P + 1 \cdot L)$:	0	0.0997	0.5858	0.9244	2	2.3951	3.4142	3.7499	4
$(P + 5 \cdot L)$:	0	0.3995	2.9289	4.1404	10	11.3571	17.0711	18.3799	21.7231
$(I + 1 \cdot L)$:	0	2	2.5858	2.5858	4	4	5.4142	5.4142	6
$(I + 5 \cdot L)$:	0	6	8.9289	8.9289	16	16	23.0711	23.0711	26

Z hodnot druhého vlastního čísla λ_2 je patrné, že virtuální leader spojený pouze jednou orientovanou hranou dosáhne konsensu značně pomaleji, než leader propojen se všemi agenty. Zároveň je vidět vliv návrhového parametru c , který při hodnotě $c = 5$ několikrát zvětší druhé vlastní číslo, a tak ještě více urychlí systém.

Vlastnosti vlastních čísel se projeví i v simulacích dosažení konsensu na Obrázku 16, kde je vidět ještě další zajímavé chování systému řízeného leadrem, který je spojen pouze jednou hranou. Oproti Obrázkům 16b a 16d, kde vlivem parametru c dojde k jasnému zrychlení, je na Obrázcích 16a a 16c možné pozorovat taktéž zrychlení ze začátku přechodového, ale s následným dlouhým ustalováním na požadovanou hodnotu. Toto chování je výsledkem funkce lokálního řízení, informace o kýžené referenci se k poslednímu agentovi v topologii dostane až za delší dobu, z důvodu průměrování okolních sousedů, kteří taktéž nemají přesnou referenci ihned.



(a) Virtuální leader spojen jednou orientovanou hranou, kde $c = 1$ (b) Virtuální leader spojen s každým uzlem, kde $c = 5$



(c) Virtuální leader spojen jednou orientovanou hranou, kde $c = 5$ (d) Virtuální leader spojen s každým uzlem, kde $c = 5$

Obrázek 16: Dosažení konsensu pro virtuální leadry s různými hodnotami parametru c

Dalším bodem v této kapitole je ukázat chování systému řízeného externím leadrem. Pro simulaci dosažení konsensu byla uvažována topologie z Obrázku 10, pro níž jsou do Rovnice 10 numericky dosazeny konkrétní hodnoty pro danou topologii.

Tímto dosazením získáme následující tvar dynamiky systému:

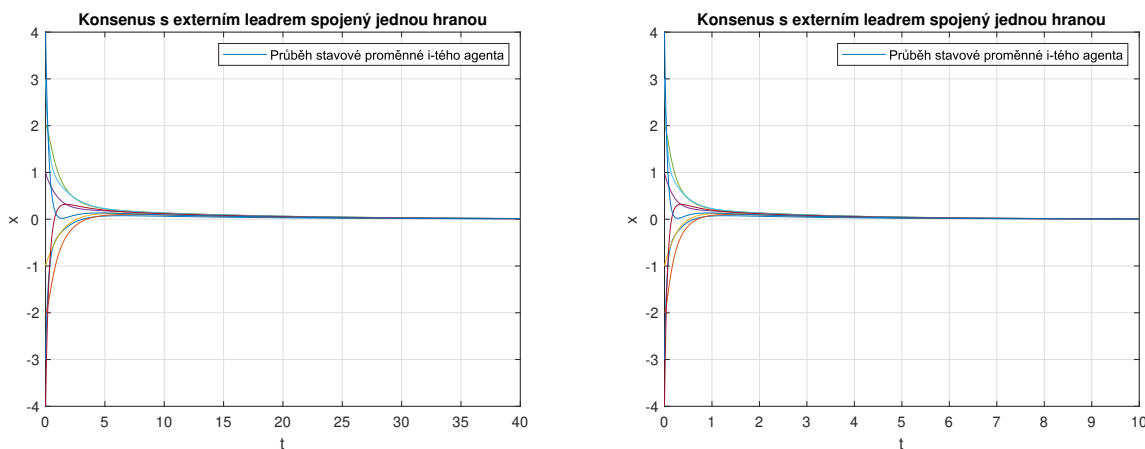
$$\begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \\ \dot{\tilde{x}}_3 \\ \dot{\tilde{x}}_4 \\ \dot{\tilde{x}}_5 \\ \dot{\tilde{x}}_6 \\ \dot{\tilde{x}}_7 \\ \dot{\tilde{x}}_8 \end{bmatrix} = -c \left[G + \begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix} \right] \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \\ \tilde{x}_5 \\ \tilde{x}_6 \\ \tilde{x}_7 \\ \tilde{x}_8 \end{bmatrix}.$$

Po dopočítání celé matice s uvažovaným parametrem $c = 1$ a $c = 5$ jsou získány následující hodnoty vlastních čísel:

Tabulka 4: Vlastní čísla matice $c(G + L)$

i	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8
$1(G + L)$:	0.0725	0.5858	0.7967	2	2.2327	3.4142	3.6449	4.2533
$5(G + L)$:	0.3626	2.9289	3.9834	10	11.1633	17.0711	18.2243	21.2664

Z hodnot druhého vlastního čísla λ_2 je vidět vliv návrhového parametru c , který při hodnotě $c = 5$ několikrát zvětší druhé vlastní číslo, a tak ještě více urychlí systém. Zajímavost tohoto přístupu řízení je, že výše zmíněná matice nemá nulové vlastní číslo.



(a) Externí leader spojen jednou orientovanou hranou, kde $c = 1$ (b) Externí leader spojen jednou orientovanou hranou, kde $c = 5$

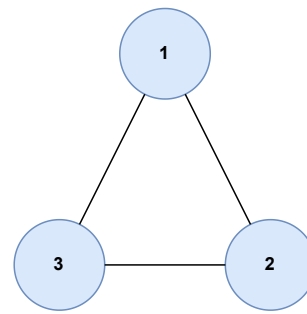
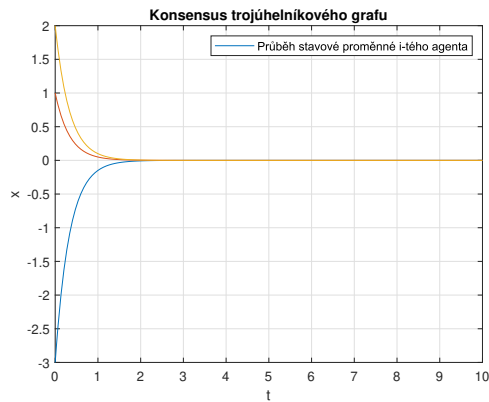
Obrázek 17: Dosažení konsensu pro externího leadra s různými hodnotami parametru c

Na Obrázcích 17a a 17b je možné si povšimnout podobného chování jako u virtuálního leadra spojeného jednou orientovanou hranou na Obrázcích 16a a 16c s tím rozdílem, že systém s externím leadrem dosáhne konsensu dříve vlivem parametru c , který přenásobuje až součet matice G a L . Tento vliv je patrný i z vlastních čísel z Tabulky 4, přestože jde o stejnou topologii spojení leadra.

3.4 Simulace konsensu pro různé topologie grafů

Nyní budou ukázány simulace dosažení konsensu pro trojúhelníkové, čtvercové a hexagonální topologie, pomocí konsensus protokolu. Systém s třemi agenty zobrazen na Obrázku 18b, bude značen \mathcal{G}_1 , stejně tak systém o čtyřech agentech vyobrazen na Obrázku 19b, bude pojmenován \mathcal{G}_2 a finálně systém o šesti agentech na Obrázku 20b, bude označen \mathcal{G}_3 .

Graf se třemi agenty

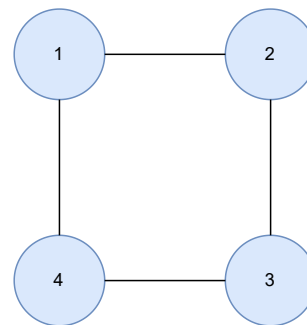
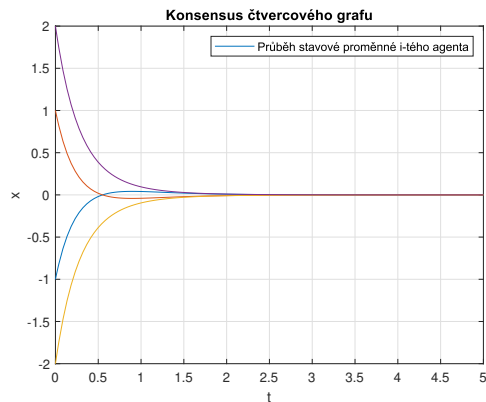


(b) Graf \mathcal{G}_1

(a) Konsensus systému definovaného grafem \mathcal{G}_1

Obrázek 18: Simulace konsensu pro tři agenty

Graf se čtyřmi agenty

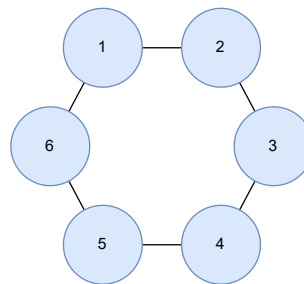
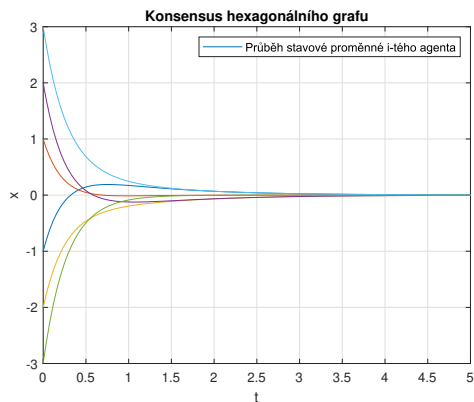


(b) Graf \mathcal{G}_2

(a) Konsensus systému definovaného grafem \mathcal{G}_2

Obrázek 19: Simulace konsensu pro čtyři agenty

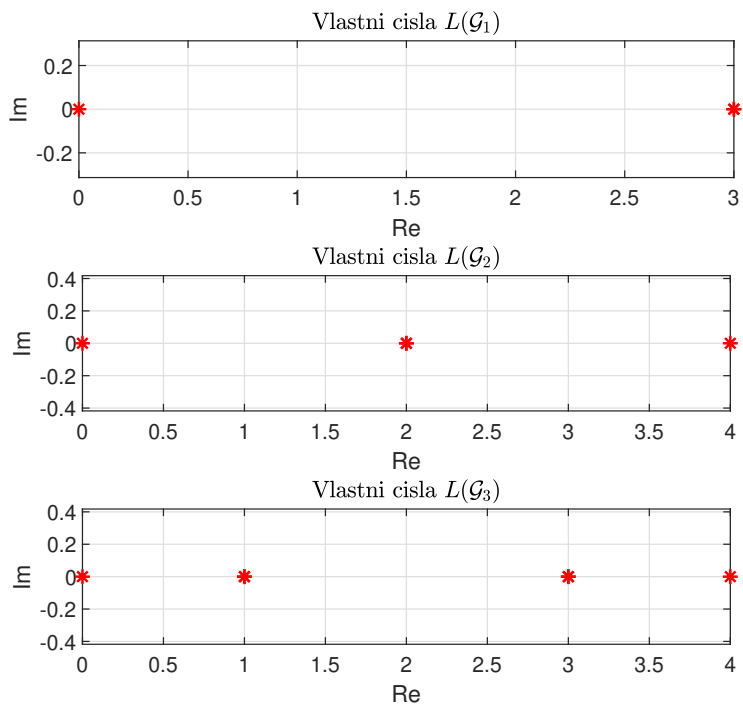
Graf se šesti agenty



(b) Graf \mathcal{G}_3

(a) Konsensus systému definovaného grafem \mathcal{G}_3

Obrázek 20: Simulace konsensu pro šest agentů



Obrázek 21: Vlastní čísla $L(\mathcal{G}_i)$

Ve všech těchto simulacích zobrazených na Obrázcích 18a, 19a a 20a úspěšně dojde k dosažení konsensu z pevně daných počátečních hodnot: $[-3 \ 1 \ 2]^T$ pro 18a, $[-1 \ 1 \ -2 \ 2]^T$ pro 19a a finálně $[-1 \ 1 \ -2 \ 2 \ -3 \ 3]^T$ pro 20a. Tyto počáteční stavy agentů byly voleny záměrně, aby jejich součet byl roven nule, a tak agenti dosáhnou konsensu na hodnotě nula. Na Obrázku 21 je možno pozorovat vlastní čísla výše uvedených topologií.

3.5 Konsensus protokol pro formace

V předchozích sekcích byla věnována pozornost různým metodám dosažení konsensu a samotným simulacím dosažení konsensu v jedné dimenzi v čase. V této kapitole budou využity principy a znalosti z předchozích sekcí pro úpravu konsensus protokolu tak, aby agenti mohli zaujmout formace ve 2D prostoru. Tedy cílem je upravit zákon řízení tak, aby agenti nekonvergovali do jediného bodu, nýbrž do předem daných bodů. Toho je možnost dosáhnout přidáním do Rovnice řízení 2 relativní vzdálenosti mezi jednotlivými agenty následujícím způsobem:

$$u_i = \sum_{j=1}^N a_{ij}((x_j - \Delta_j) - (x_i - \Delta_i)),$$

kde Δ_i značí kýženou vzdálenost od agenta x_j a stejně tak Δ_j představuje požadovanou vzdálenost od agenta x_j . Jejich relativní vzdálenost lze tedy definovat jako:

$$\Delta_{ij} = \Delta_i - \Delta_j.$$

Pro všechny relativní vzdálenosti mezi agenty lze najít vektor h , který se nachází v nulovém prostoru matice L , dle Rovnice 4. Nyní je možno zavést konsensus protokol pro formace ve tvaru:

$$\dot{x} = -L(x - h). \quad (13)$$

Jak už bylo výše zmíněno, pro formace je nutno uvažovat i druhá dimenze. V následujících simulacích jsou tedy provedeny výpočty zvlášť pro x a y hodnoty. Zde tedy vyvstává nutnost rozdělit vektor h na h_x a h_y . Jako první bude ukázána oktagonová formace systému daného topologií z Obrázku 8a, pro který byly vypočteny vektory:

$$\begin{aligned} h_x &= [0.7071 \ 0 \ -0.7071 \ -1 \ -0.7071 \ 0 \ 0.7071 \ 1], \\ h_y &= [0.7071 \ 1 \ 0.7071 \ 0 \ -0.7071 \ -1 \ -0.7071 \ 0]. \end{aligned}$$

V následujících simulacích mějme systém daný řízením:

$$\begin{aligned} \dot{x} &= -L(x - h_x), \\ \dot{y} &= -L(y - h_y). \end{aligned} \quad (14)$$

Alternativa pro systém daný Rovnicemi 14 je rozšíření systému jako celku pomocí kroneckerova součinu s jedničkovou diagonální maticí o rozměru n . V tomto případě $n = 2$ pro 2D tímto způsobem:

$$\dot{x} = -(L \otimes I)(x - h), \quad (15)$$

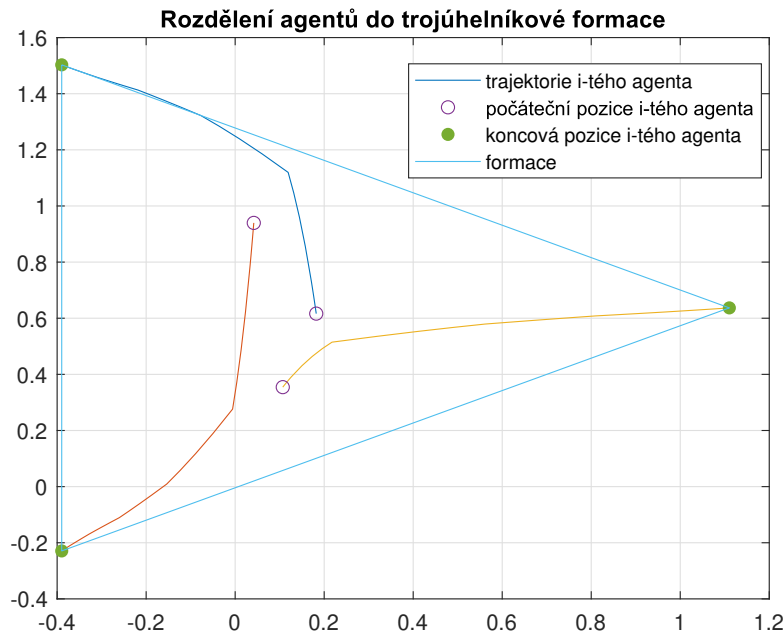
kde I je výše zmíněná diagonální rovnice a vektor h je definován jako $h = [h_x, h_y]^T$. Doteď byl uvažován stavový vektor x jako: $x = [x_1 \cdots x_n]^T$, nyní je rozšířen i o stavy, které představují y -ové hodnoty. Formálně dosadíme do Rovnice 15, čímž je získáno:

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \\ \dot{y}_1 \\ \vdots \\ \dot{y}_n \end{bmatrix} = \begin{bmatrix} L & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ y_1 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} h_{x_1} \\ \vdots \\ h_{x_n} \\ h_{y_1} \\ \vdots \\ h_{y_n} \end{bmatrix}.$$

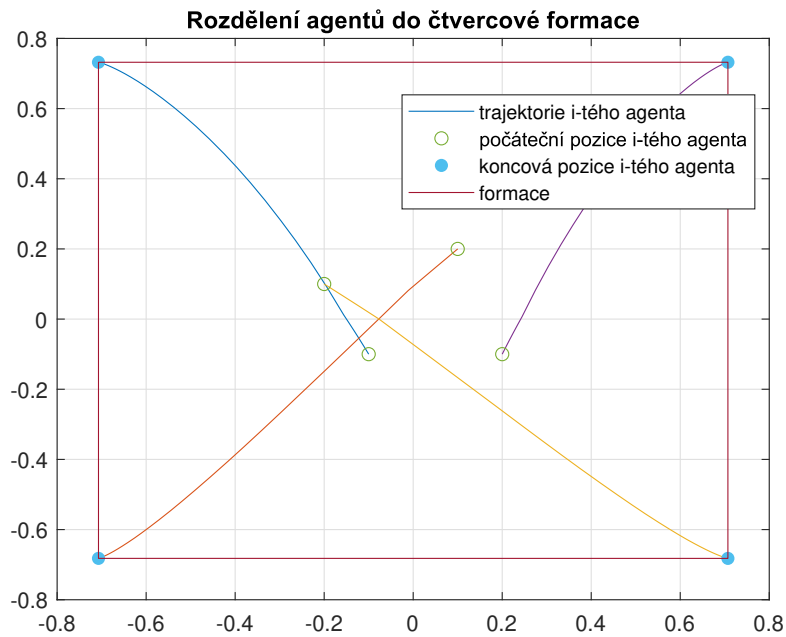
Takový systém se bude chovat ekvivalentně k systému 14.

Simulace rozdělení do různých formací pomocí konsensus protokolu

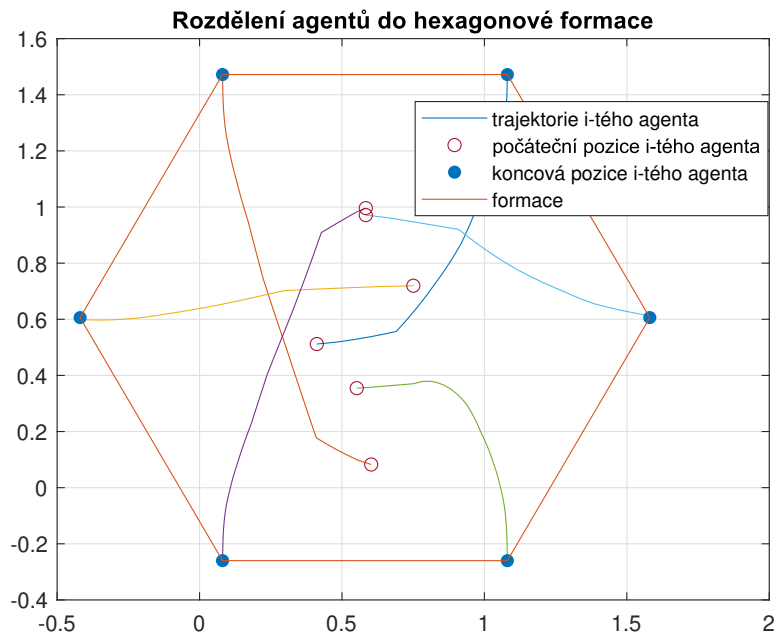
Nyní budou ukázány simulace rozdělení agentů do trojúhelníkové, čtvercové a hexagonální formace. Systém s třemi agenty zobrazen na Obrázku 18b, bude rozdělen do trojúhelníkové formace. Stejně tak systém o čtyřech agentech vyobrazen na Obrázku 19b zaujme čtvercovou formaci. Topologie o šesti agentech na Obrázku 20b bude konfigurován do formace ve tvaru hexagonu. Finálně systém s osmi agenty z Obrázku 8a bude rozdělen do hexagonální formace. Ve všech simulacích agenti vycházejí z náhodných počátečních souřadnic.



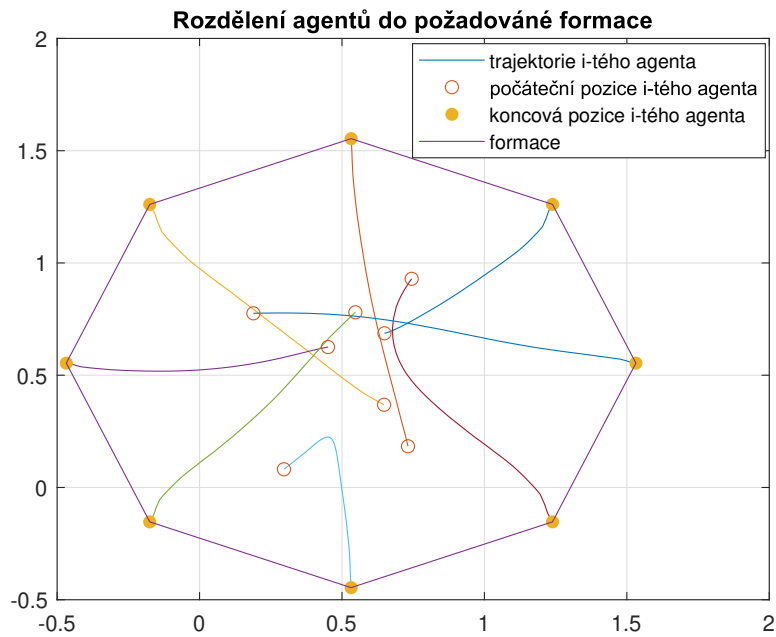
Obrázek 22: Ukázka rozdělení agentů do trojúhelníkové formace z náhodných počátečních pozic



Obrázek 23: Ukázka rozdělení agentů do čtvercové formace z náhodných počátečních pozic



Obrázek 24: Ukázka rozdělení agentů do hexagonové formace z náhodných počátečních pozic



Obrázek 25: Ukázka rozdělení agentů do oktagonové formace z náhodných počátečních pozic

Na všech výše uvedených Obrázcích 22, 23, 24 a 25 se podařilo rozdělit agenty do požadovaných formací příslušnými vektory h , které specifikují vzájemnou vzdálenost mezi agenty.

4 Formace agentů

V předchozí kapitole 3.5 bylo ukázáno, jak pomocí konsensus protokolu s Laplaciánem se systém za nějaký čas dostane do požadované formace dané vektorem h . V této kapitole bude tato disciplína rozšířena o schopnost udržování dané formace v čase pomocí stavového zpětnovazebního řízení. Dále zde bude uvažován komplexnější model systému, který kromě pozic agentů v prostoru, bude obsahovat i informace o rychlosti jednotlivých agentů.

Model agentů a vektor formací

V nadcházejících sekcích bude uvažován stavový model agentů ve tvaru:

$$\dot{x}_i = A_{veh}x_i(t) + B_{veh}u_i(t) \quad i = 1, \dots, n, \quad (16)$$

kde n je počet agentů a stavy $x_i \in R^2$ jsou definovány jako:

$$x_i = \begin{bmatrix} x_p \\ x_v \end{bmatrix}, x_p = \begin{bmatrix} (x_p)_1 \\ \vdots \\ (x_p)_n \end{bmatrix}, x_v = \begin{bmatrix} (x_v)_1 \\ \vdots \\ (x_v)_n \end{bmatrix},$$

kde x_p představují stavy polohy v rovině (x, y) tak, že $(x_p)_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$ a x_v představují odpovídající rychlosti, z čehož je dále možno usoudit tvar časové derivace stavu \dot{x}_i jako:

$$\dot{x}_i = \begin{bmatrix} x_v \\ x_a \end{bmatrix},$$

kde x_a jsou stavy zrychlení agentů. Matice A_{veh} a B_{veh} jsou matice dynamiky a řízení jednotlivých agentů, které jsou definovány jako:

$$A_{veh} = \begin{bmatrix} 0 & 1 \\ a_{21} & a_{22} \end{bmatrix}, B_{veh} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Je uvažováno $N = 2$ pro dvojrozměrný pohyb, tedy je možno zapsat referenční stavový popis agenta pro pohyb v jedné rovině přenosným A_{veh} a B_{veh} jedničkovou maticí I_2 do tvaru:

$$\begin{bmatrix} x_v \\ x_a \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_p \\ x_v \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u_i. \quad (17)$$

Pro nově zavedený model agentů je nutno definovat formační vektor h , který respektuje přidání stav rychlosti každého agenta. Toho je dosaženo kroneckerovým součinem vektoru h_p následujícím způsobem:

$$h = h_p \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

kde h_p je vektor pozic formace definovaný jako: $[(x_1, y_1)^1, (x_2, y_2)^2 \dots (x_n, y_n)^n]$ pro $1 \dots n$ agentů.

Stavový model systému se stavovou zpětnou vazbou

Nyní je možno začít definovat celkový stavový popis systému o n agentech a následně spojit tento popis s Rovnicí 13 pro dosažení formace pomocí Laplaciánu tak, že bude považována jako řízení u . Postup vytvoření celkového popisu systému je následující:

$$\begin{aligned}\dot{x} &= Ax(t) + Bu(t), \\ u(t) &= L(x(t) - h),\end{aligned}$$

kde:

$$A = I_n \otimes A_{veh}, \quad B = I_n \otimes B_{veh}, \quad L = L \otimes I_{2n}, \quad (18)$$

kde I_n je diagonální jedničková matice. Po spojení těchto rovnic je zavedena zpětnovazební matice K , která zajistí adekvátní řízení formace.

$$\dot{x}(t) = Ax(t) + BKL(x(t) - h) \quad (19)$$

Pokud bude do Rovnice 19 zpětně dosazeno z 18 a zároveň za K substituováno $K = I_n \otimes K_{veh}$, tak je získána finální rovnice:

$$\dot{x}(t) = I_n \otimes A_{veh}x(t) + L \otimes B_{veh}F_{veh}(x(t) - h). \quad (20)$$

Pro další práci s návrhy parametrů je nutné zavést vlastní čísla Laplaciánu přímo do Rovnice 20. Toho je docíleno definováním matice U , která transformuje matici L na horní trojúhelníkovou matici $\tilde{L} = U^{-1}LU$, čímž je docíleno, že vlastní čísla matice L jsou na diagonále matice \tilde{L} . Následným zavedením této transformace do Rovnice 20 se současným použitím 18 je získáno:

$$(U^{-1} \otimes I_{2n})(A + BKL)(U \otimes I_{2n}) = I_n \otimes A_{veh} + \tilde{L} \otimes B_{veh}K_{veh}, \quad (21)$$

kde pravá strana je blokově horně trojúhelníková, pro níž lze vyjádřit její diagonální bloky ve tvaru:

$$A_{veh} + \lambda B_{veh}K_{veh},$$

kde λ je vlastní číslo Laplaciánu pro daný blok a pro každý blok je své vlastní číslo. Tedy vlastní čísla $A + BKL$ jsou vlastní čísla $A_{veh} + \lambda B_{veh}K_{veh}$, tedy:

$$A + \lambda BK = I_{nN} \otimes (A_{veh} + \lambda B_{veh}K_{veh}). \quad (22)$$

Nadcházejícím cílem je tedy najít parametry matice K_{veh} , která zajistí stabilizující zpětnovazební řízení systému do požadované formace. Všeobecně stavové zpětnovazební řízení upravuje vlastní čísla systému, od tohoto faktu je též stavová zpětná vazba označovaná jako nedynamická. Nejprve je nutné definovat parametry a_{21} a a_{22} , které jsou obsaženy v matici A_{veh} , jenž ovlivňují dynamiku systému.

Je řečeno, že pokud pro každou formaci h existuje zpětnovazební stabilizační matice $K = I_n \otimes K_{veh}$, která zajistí, že rovnice $L(x - h)$ konverguje do nuly, potom $a_{21} = 0$ [13]. Už

z předchozích kapitol je ověřeno, že pokud $L(x - h) = 0$, potom systém dosáhne konsensu. Nově je tedy matice A_{veh} pro jednoho agenta definována jako:

$$A_{veh} = \begin{bmatrix} 0 & 1 \\ 0 & a_{22} \end{bmatrix}.$$

Jsou uvažovány dva parametry k_1 a k_2 pro zpětnovazební K_{veh} , která je psána jako:

$$K_{veh} = (k_1, k_2).$$

Za cíl je uvažováno hledání skalárních proměnných k_1 a k_2 . Toho je dosaženo vyjádřením charakteristického polynomu matice, která je získána dosazením výše zmíněných matic do Rovnice 22.

$$A + \lambda BK = I_n N \otimes \left(\begin{bmatrix} 0 & 1 \\ 0 & a_{22} \end{bmatrix} + \lambda \begin{bmatrix} 0 \\ 1 \end{bmatrix} (f_1, f_2) \right) = I_n N \otimes \begin{bmatrix} 0 & 1 \\ \lambda k_1 & a_{22} + \lambda k_2 \end{bmatrix} = I_n N \otimes M$$

Charakteristický polynom matice M je $p_m(x) = x^2 - (a_{22} + \lambda k_2)x - \lambda k_1$. Tudíž je matice stabilní pouze tehdy pokud:

$$a_{22} + \lambda k_2 < 0 \wedge \lambda k_1 < 0 \wedge |k_2| \gg 0. \quad (23)$$

Výpočet parametrů k_1 a k_2

Nyní je očividné, že je nutno volit záporné parametry matice K_{veh} . Pro jejich návrh bude využit experimentální přístup, kdy bude zjednodušen model agenta tak, že $a_{22} = 0$ a zároveň bude ručně zvolen parametr k_1 , pro který bude dopočítán druhý parametr. Vzhledem k faktu, že dynamiku systému též ovlivňují jeho vlastní čísla, bude použit výpočet determinantu. Předpis získáme výpočtem determinantu matice M :

$$\det(\lambda I - M) = (a_{22} + \lambda k_2)^2 + 4\lambda k_1,$$

a následným vyjádřením k_1 :

$$\frac{(\lambda k_2)^2}{4\lambda} < -k_1. \quad (24)$$

Tento výpočet lze aplikovat na systémy s topologií neorientovaného grafu, jež Laplacián má pouze reálná vlastní čísla. Pro neorientované grafy s komplexními vlastními čísly $\lambda_{komplexni} = \alpha + i\beta$ je uvažován výpočet ve tvaru:

$$\frac{k_2^2}{k_1} < -\frac{\beta^2}{\alpha(\alpha^2 + \beta^2)} = k_2^2 < \frac{\beta^2}{\alpha(\alpha^2 + \beta^2)} k_1, \quad (25)$$

kde α je reálná část a β imaginární část vlastního čísla.

4.1 Udržování formace v čase

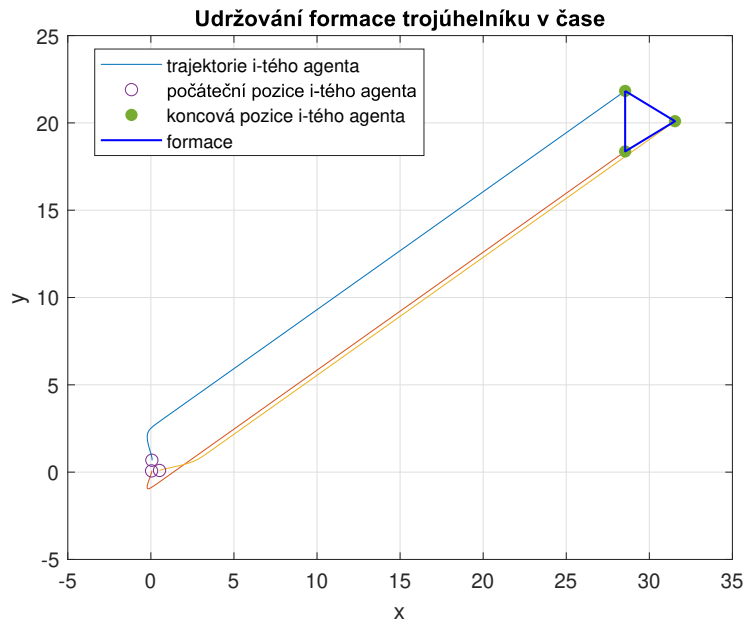
Z předchozích sekcí je připraven teoretický základ pro simulace udržování v čase pro dané formace. Pro sumarizaci a přípravu pro simulace ve 2D je uvažován systém pro n agentů ve tvaru:

$$\dot{x}(t) = Ax(t) + BKL(x(t) - h), \quad (26)$$

kde $A = I_{2n} \otimes \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B = I_{2n} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $K = I_{2n} \otimes [k_1, k_2]$ a $L = L_g \otimes I_{2n}$. Matice L_g je původní Laplacián příslušného grafu. Dále budou uvedeny simulace pro průvodní příklady topologií o třech, čtyřech, šesti a osmi agentech, jež se rozdělí příslušně do trojúhelníkové, čtvercové, hexagonální a oktagonové formace, přičemž bude sledováno jejich chování. U všech těchto simulací jsou uvažovány počáteční pozice jednotlivých agentů v intervalu $(0, 1)$.

Trojúhelníková formace

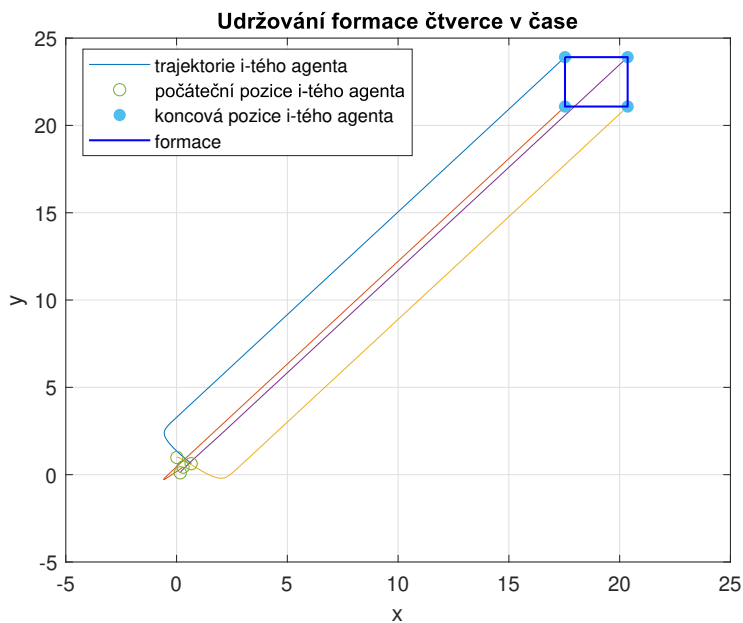
Pro trojúhelníkovou formaci a všechny následující byl zvolen parametr $k_1 = -10$, pro něhož byl vypočítán dle vztahu 24 parametr $k_2 = -3.6515$. Na Obrázku 26 je znázorněn průběh simulace pro udržování trojúhelníkové formace v čase.



Obrázek 26: Ukázka rozdělení agentů do trojúhelníkové formace a její udržování v čase

Čtvercová formace

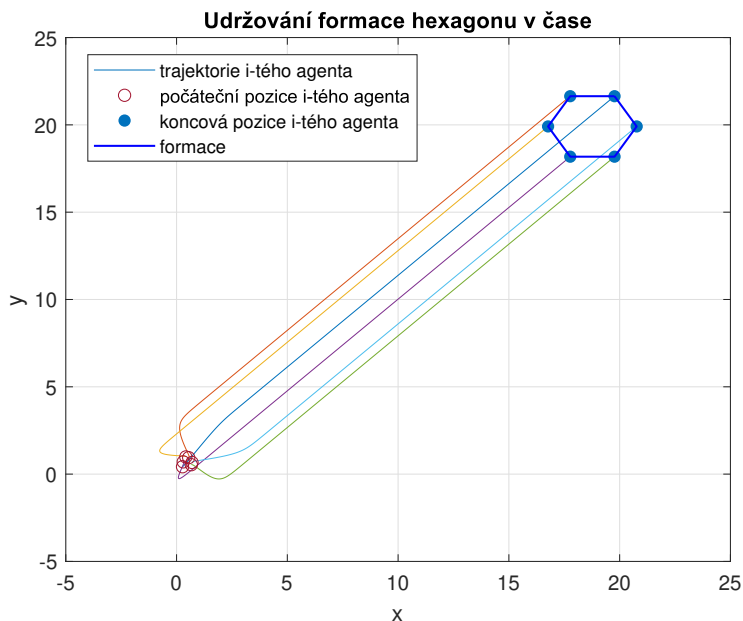
Pro čtvercovou formaci byl vypočítán dle vztahu 24 parametr $k_2 = -4.4721$. Na Obrázku 27 je znázorněn průběh simulace pro udržování formace v čase.



Obrázek 27: Ukázka rozdělení agentů do čtvercové formace a její udržování v čase

Hexagonová formace

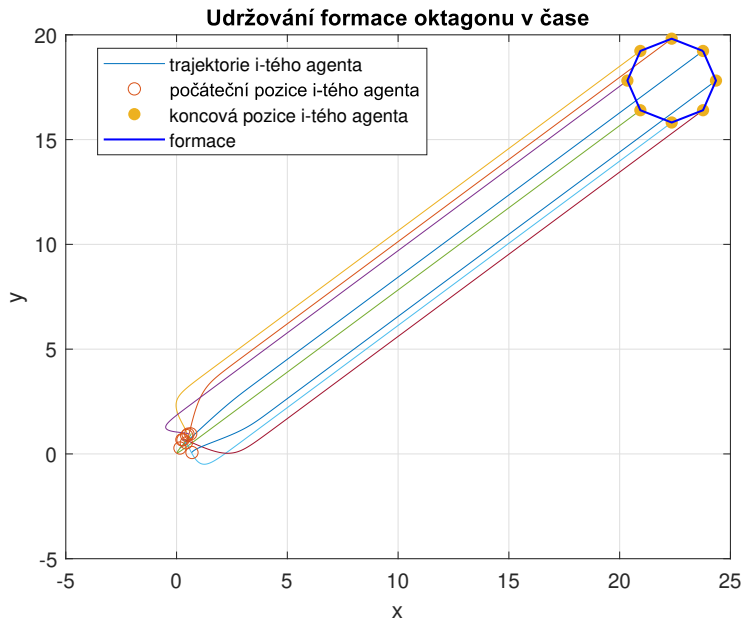
Pro hexagonovou formaci byl vypočítán dle vztahu 24 parametr $k_2 = -6.3264$. Na Obrázku 28 je znázorněn průběh simulace pro udržování formace v čase.



Obrázek 28: Ukázka rozdělení agentů do hexagonové formace a její udržování v čase

Oktagonová formace

Pro oktagonovou formaci byl vypočítán dle vztahu 24 parametr $k_2 = -8.2634$. Na Obrázku 29 je znázorněn průběh simulace pro udržování formace v čase.



Obrázek 29: Ukázka rozdělení agentů do oktagonové formace a její udržování v čase

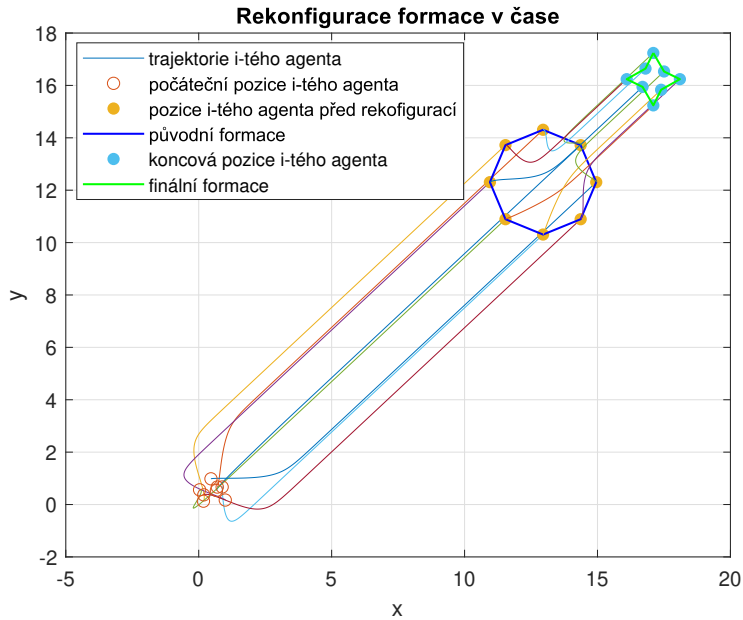
Na všech výše uvedených obrázcích (Obrázek 26, 27, 28 a 29) se podařilo rozdělit agenty do požadovaných formací příslušnými vektory h , které specifikují vzájemnou vzdálenost mezi agenty a nadále udržovat tuto formaci v čase. Trajektorie všech agentů jsou bez překmitu.

4.2 Rekonfigurace formace agentů

V předchozích kapitolách bylo ukázáno, jak se mohou agenti rozdělit do formací pomocí stavové zpětné vazby, která dále udržuje daný systém v čase. Jedním z možných požadavků na praktické využití tohoto řízení je rekonfigurace referenční formace na jinou. V této podkapitole bude ukázáno simulačně, jak se systém bude při dané rekonfiguraci chovat. Mějme systém daný osmi agenty daný topologií 8a s původní oktagonovou formací a náhodnými počátečními souřadnicemi. Cíl je rekonfigurovat tuto formaci do čtyřcípé hvězdy.

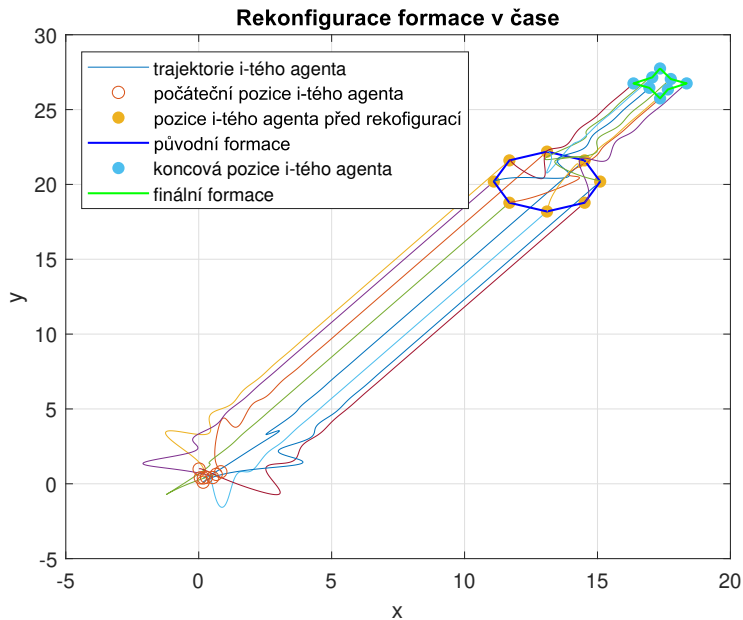
Toho je dosaženo změnou vektoru h v čase $t = 30$ s v průběhu simulace. Kde v rámci dalších deseti sekund se formace rekonfiguruje a je dále udržována v čase. Stejně jako v podkapitole 4.1 byl zvolen parametr $k_1 = -10$, pro něj byl vypočítán parametr $k_2 = -8.2634$. Jelikož tvar formace neovlivňuje vlastní čísla systému, toto řízení bude adekvátní i pro druhou formaci hvězdy.

V této konfiguraci s danými parametry je druhé vlastní číslo $\lambda_2 = -1.2580$ a dosáhne relativních vzdáleností mezi agenty pro první formaci v čase $t \approx 2.640$ s. Při rekonfiguraci dosáhne požadovaných relativních vzdáleností až v $t \approx 3.340$ s. Toto chování je dáno většími pozičními odchylkami při rekonfiguraci oproti počátečním pozicím.



Obrázek 30: Ukázka rekonfigurace z oktagonové do hvězdicové formace

Lze uvažovat nad tím, co se stane, pokud bychom zvolili špatně parametr k_2 například, pokud by byl parametr k_2 větší než z výpočtu 24 a zároveň tak, aby nebyl tak záporný jako ve vztahu 23. Mějme tedy stejný systém s rekonfigurací jako v předchozím příkladě s tím rozdílem, že $k_2 = -2$.



Obrázek 31: Ukázka rekonfigurace z oktagonové do hvězdicové formace s $k_2 = -2$

Na Obrázku 31 je možné si povšimnout silných překmitů společně s delším časem ustálení, oproti předchozímu případu. Nicméně systém je stále stabilní a dokáže udržovat

formaci v čase. Toto odpovídá předpokladům pro stabilitu ze vztahů 23. Důvod, proč takové chování nastává, je možno hledat ve vlastních číslech systému. Druhé vlastní číslo tohoto systému je $\lambda_2 = -0.5858 + 2.3483i$. Jeho menší reálná část vysvětluje pomalejší dosažení požadovaných relativních vzdáleností mezi agenty a jeho velká imaginární část má za následek velkou kmitavost systému.

Podobným způsobem jako bylo demonstrováno v této podkapitole by šlo realizovat rekonfiguraci libovolně velké formace na jiný tvar, pouze s podmínkou zachování počtu agentů.

5 Diskuze získaných výsledků

V celé této práci byl přítomen významný vliv matice Laplaciánu, který obsahuje důležité vlastnosti topologie systému. Analýzou Laplaciánu, zejména jeho vlastních čísel, bylo možné předpokládat chování multiagentního systému při uvažovaném distribuovaném i centralizovaném způsobu řízení.

Jako první bylo nahlédnuto na vlivy provázanosti grafu na dosažení konsensu. Jak už bylo výše zmíněno, Laplacián zachycuje provázanost grafu, která se prakticky projevuje v hodnotách jeho vlastních čísel. Pro grafy, jež jsou méně propojené, jsou vlastní čísla Laplaciánu menší, než pro grafy, které jsou propojené více. Zejména důležitý vliv má druhé vlastní číslo λ_2 , jehož velikost určuje rychlost dosažení konsensu. Čím menší je velikost λ_2 , tím pomaleji systém dosáhne konsensu a opačně. Tento efekt byl simulačně ověřen na několika grafech, kde se pro slabě provázaný graf, čas dosažení konsensu pohyboval okolo $t \approx 8$ s a naopak pro silně provázaný graf došlo až téměř k osminásobnému zrychlení na $t \approx 1$ s.

Spolu s provázaností grafů souvisí i orientované grafy, které ačkoliv mají hrany na stejných místech jako neorientovaný graf, vykazují v simulacích překmity a delší časy dosažení konsensu, než u neorientovaných topologií. Systém s topologií neorientovaného grafu dosáhl konsensu až v čase $t \approx 16$ s. Toto chování dává smysl vzhledem k faktu, že informace mohou téct pouze jedním směrem, a tak poslední agent dostane informaci od ostatních agentů o dost později. Zároveň vlastní čísla neorientovaného grafu mohou být komplexní, což vysvětluje výše zmíněnou kmitavost.

U centralizovaných způsobů řízení pomocí virtuálního a externího leadra byl pozorován stejný, ba i větší efekt. Virtuální leader, který je spojen pouze jednou orientovanou hranou, měl čas dosáhnout konsensu až 35 s s dlouhým ustalováním hodnotě konsensu. U externího leadra spojeného jednou orientovanou hranou bylo chování dost podobné, vzhledem k tomu, že jde o stejný styl řízení, pouze je zde rozdíl v reprezentaci leadra. Na rozdíl od distribuovaného řízení je zde možné dosažení konsensu urychlit pomocí návrhového parametru c , který značně zrychluje systém. Změnou parametru z $c = 1$ na $c = 5$ bylo dosaženo zkrácení doby dosažení konsensu o 15 s u virtuálního leadra s jednou hranou a až o 30 s u externího leadra s jednou hranou. Oproti tomu u virtuálního leadra, který je spojen se všemi agenty vliv parametru c nebyl tak signifikantní, došlo k urychlení o 2,5 s z původních 4 s pro parametr $c = 1$. Z toho je možné usoudit, že nejvýznamnější vliv na čas dosažení konsensu má topologie grafu a až v druhé řadě návrhový požadavek c , nicméně dalším navyšováním parametru by dále došlo ke zrychlování systému obou topologií.

Jako poslední bylo nahlédnuto do problematiky udržování formací v čase pomocí stavové zpětné vazby, kde jsou uvažovány parametry k_1 a k_2 zpětnovazební matice K , která má za úkol stabilizovat systém. Z tvaru charakteristického polynomu matice uzavřeného systému bylo zjištěno, má-li být systém stabilní, musí oba parametry mít záporné hodnoty s tím, že k_2 musí být v absolutní hodnotě větší než nula. Volba těchto parametrů byla závislá na topologii systému, kde čím menší bylo druhé vlastní číslo Laplaciánu daného grafu, tím byl i vypočtený parametr k_2 zápornější. Pro korektně zvolené parametry k_2 došlo k rozdělení agentů do formací. Simulačně bylo ukázáno, jak se příslušné systémy rozdělí do formace

trojúhelníku, čtverce, hexagonu a oktagonu. Finálně bylo ukázáno na příkladu rekonfigurace varianta se špatně zvoleným parametrem k_2 . Pro $k_2 = -2$ se stal velice kmitavým s delším ustalováním do požadované formace vlivem komplexních vlastních čísel.

V následujících Tabulkách 5 a 6 budou ukázány vlastní čísla příslušných Laplaciánů, či rozšířených matic dynamiky s Laplaciánem všech výše zmíněných přístupů řízení pro dosažení konsensu.

Tabulka 5: Vlastní čísla matic Laplaciánů jeho rozšířených variant a)

i	λ_1	λ_2	λ_3	λ_4
1:	0	0.5858	0.5858	2
2:	0	2	2	2.5858
3:	0	8	8	8
4:	0	$0.2929 + 0.7j$	$0.2929 - 0.7j$	$1 + 1j$
5:	0	0.0997	0.5858	0.9244
6:	0	0.3995	2.9289	4.1404
7:	0	2	2.5858	2.5858
8:	0	6	8.9289	8.9289
9:	0.0725	0.5858	0.7967	2
10:	0.3626	2.9289	3.9834	10

Tabulka 6: Vlastní čísla matic Laplaciánů jeho rozšířených variant b)

i	λ_5	λ_6	λ_7	λ_8	λ_9
1:	2	3.4142	3.4142	4	
2:	2.5858	4	5.4142	5.4142	
3:	8	8	8	8	
4:	$1-1j$	2	$1.7071+0.7j$	$1.7071-0.7j$	
5:	2	2.3951	3.4142	3.7499	4
6:	10	11.3571	17.0711	18.3799	21.7231
7:	4	4	5.4142	5.4142	6
8:	16	16	23.0711	23.0711	26
9:	2	2.2327	3.4142	3.6449	4.2533
10:	10	11.1633	17.0711	18.2243	21.2664

V Tabulkách 5 a 6 jsou vypsány všechny vlastní čísla systémů příslušných příkladů v práci, se kterými byly prováděny simulace dosažení konsensu. Pro $i = 1, 2, 3$ se jedná o neorientované grafy o osmi agentech z Obrázků 8a, 11a, 11a, které jsou v každém dalším kroku provázanější. Na pozici $i = 4$ jsou vyobrazeny vlastní čísla Laplaciánu orientovaného grafu o osmi agentech 8b, který je málo provázaný. Řádky $i = 5, 6$ patří systémům řízeným

virtuálním leadrem, který je spojen jednou hranou 9a s parametry $c = 1, 5$. Stejně tak $i = 7, 8$ patří systémům řízeným virtuálním leadrem, který je spojen se všemi agenty 9b s parametry $c = 1, 5$. Finálně řádky $i = 9, 10$ odpovídají vlastním číslům systémů řízených externím leadrem, který je spojen jednou hranou 10 s parametry $c = 1, 5$.

Pro systém daný Rovnicí 26 jsou vlastní čísla odpovídající směru x, y a jejich zrychlení stejné. Proto bude pro porovnání vlastních čísel matic z výše uvedených příkladů $A + BKL$ použita pouze jedna sada odpovídající právě například pozici ve směru x v Tabulce 7

Tabulka 7: Vlastní čísla $A + BKL$

i	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8
1:	0	-5.477	-5.477					
2:	0	-2.929	-4.472	-15.27				
3:	0	-1.695	-1.741	-3.162	-17.23	-23.604		
4:	0	-1.258	-1.267	-1.315	-2.42	-15.212	-26.946	-32.796
5:	0	-0.586 + 2.3j	-0.586 - 2.3j	-2 + 4j	-2 - 4j	-3.414 + 4.7j	-3.514 - 4.7j	-4 + 4.9j

Když budou porovnány vlastní čísla z této tabulky s vlastními čísly Tabulek 5, 6, tak je patrný vliv stavové zpětné vazby na jejich tvar a pozici v komplexní rovině

5.1 Future works

Tato práce se může rozšířit o nový způsob reprezentace jednotlivých agentů. V tomto textu byl uvažován agent prvního řádu (integrátor), nabízí se tak jeho reprezentace vyšším řádem, například dvojnásobným integrátorem a podobně. Zde by bylo zajímavé sledovat vlivy topologie a přístupů řízení na dosažení konsensu a rozdělení agentů vyšších řádů do formací.

Další možností se nabízí rozšíření algoritmů o schopnost detekce a vzájemného předcházení kolizí mezi jednotlivými agenty pomocí Reynoldsových pravidel. Dosud tento problém v práci nebyl uveden. Bylo by proto zajímavé zohlednit tuto problematiku například pro dosažení konsensu, ve kterém je kladeno za cíl dosáhnout stejného stavu pro všechny agenty. S tím také souvisí reprezentace překážek ve stavovém prostoru tak, aby agenti mohli překážku detekovat, a zároveň se této překážce adekvátně vyhnout, také aby po jejím překonání dosáhli požadovaného cíle, například formace. To si lze v reálném světě představit na příkladu jedoucí formace vozidel v terénu, která narazí na překážku v podobě stromu nebo formace dronů, která se musí vyhnout výškovým objektům jako jsou budovy, či stožáry.

Tím, že se dostáváme do problematiky decentralizovaných systémů, je tedy jasné, že komunikace hraje důležitou úlohu. V této práci byla komunikace mezi agenty považována jako ideální, bez žádných nepřesností, dopravního zpoždění nebo nežádoucích šumů. Zde se tak nabízí rozšířit prezentované algoritmy právě o tyto nepřesnosti. Mohlo by se jednat například o časově variabilní zpoždění na každém agentu, který by předával různě zpožděné informace dalším agentům, nebo by se jednalo o začlenění šumu při vysílání, případně

přijímání informací pro každého agenta. V poslední řadě by se mohlo jednat o začlenění "drop-off" paketů (ztracených paketů) do reprezentace systému a jejich ošetření, které by vedlo k zamezení destabilizace systému vlivem ztráty informace.

V této práci byl zákon řízení realizován pomocí stavové zpětné vazby. Zde se tak nabízí možnost vyzkoušení jiného návrhu řízení a jeho následné porovnání se způsobem, který byl popsán v této práci. Případně by se mohlo jednat o diskuzi, jaký způsob je vhodnější v závislosti na situaci a jaké poskytuje benefity.

6 Závěr

Tato bakalářská práce se zabývá kooperativním řízením multiagentních systémů, jmenovitě lokálním protokolem a stavovým zpětnovazebním řízením pro distribuované řešení a řízení s externím a virtuálním leadrem pro centralizované řešení. Všechny popsané způsoby byly v každé kapitole demonstrovány na příslušných příkladech.

Začátek práce je věnován úvodu do problematiky multiagentních systémů. Chování a vlastnosti multiagentních systémů bývají často popsané pomocí komunikační topologie, pro jejichž matematický popis se využívá grafová teorie.

Následuje seznámení s grafovou teorií, kde je vysvětleno, co je myšleno pojmem graf. K tomu byly uvedeny příklady různých typů grafů a popsány všechny potřebné pojmy, které byly následně využity v dalších kapitolách.

V další části byly popsány matice, které mají úzkou souvislost s komunikační topologií. Jedná se například o matice sousednosti, Laplacián a podobně. Laplacián je jedna z nejdůležitějších matic, která provázela celou práci, protože vlastní čísla Laplaciánu určují chování celého systému.

Stěžejní kapitolou této práce byl algoritmus konsensu (shody), kde byly definovány algoritmy řízení jako je externí, virtuální leader nebo lokální protokol řízení s Laplaciánem. U všech těchto přístupů je uvažována dynamika každého agenta jako systému prvního řádu. Vliv podoby vlastních čísel Laplaciánu pro průběh všech zmíněných algoritmů pro nalezení konsensu bylo dokumentováno na příkladech. Dosažení konsensu bylo nasimulováno pro různé topologie pomocí programu Matlab. Mezi tyto topologie patří systémy o třech, čtyřech a šesti agentech. U řízení pomocí leaderů byl diskutován vliv parametru c na vlastní čísla složené matice dynamiky, a tím pádem i na kvalitu a rychlost dosažení konsensu. Zde byl také rozšířen lokální protokol řízení o schopnost rozdělení agentů do formací v jedné rovině pomocí nově definovaného formačního vektoru h (ten obsahuje relativní vzdálenosti mezi agenty tak, aby bylo dosaženo předem definované formace). Na příkladech bylo demonstrováno rozdělení agentů do předem daných formací, kdy se jednalo o trojúhelníkovou, čtvercovou, hexagonovou a oktagonovou formaci.

Posledním tématem této práce, na které byla kladena pozornost, bylo udržování dané formace v čase, které bylo dosaženo pomocí stavové zpětné vazby. V případě stavové zpětné vazby byl zaveden nový tvar systému, ve kterém agenti obsahují stavy pozice i rychlosti pro dvou-dimenzionální pohyb. Výzva tohoto typu řízení byla v nalezení správných parametrů zpětnovazební matice K tak, aby stabilizovala systém a zároveň umožnila agentům se rozdělit do požadovaných formací. Simulačně bylo ukázáno, jak v čase udržovat formace trojúhelníku, čtverce, hexagonu a také oktagonu. Jako další bod byl vybrán speciální případ, kdy je potřeba v čase změnit formaci na jinou. Tento fakt byl v práci nazván jako rekonfigurace formace v čase. Simulačně byl ukázán případ, kdy se z náhodných počátečních podmínek systém zformuje do hexagonové formace a následně se rekonfiguruje do čtyřcípé hvězdy. Jako poslední byl uvažován případ, kdy při příkladu rekonfigurace byly použity neoptimální parametry matice K , což mělo za následek delší ustalování do požadovaných pozic agentů se značnými překmity.

Seznam obrázků

1	Ukázka možné aplikace multiagentních systémů	1
2	Stavová zpětná vazba K v blokově reprezentovaném stavovém popisu . . .	7
3	Příklad dvou typů grafů s rozdílnou orientací hran	8
4	Příklady kompletních grafů	9
5	Příklad dvou grafů, které ukazují rozdíl mezi cyklickým a acyklickým grafem	10
6	Příklady souvislých neorientovaných grafů	10
7	Příklady souvislých orientovaných grafů	11
8	Příklady grafových topologií	11
9	Příklady možných spojení virtuálního leadra se zbytkem topologie	19
10	Externí leader spojen jednou orientovanou hranou se zbytkem topologie . .	21
11	Příklady různě provázaných neorientovaných grafů	22
12	Vlastní čísla Laplaciánu $L(\mathcal{G}_i)$	23
13	Ukázka dosažení konsensu pro tři topologie \mathcal{G}_i	24
14	Vlastní čísla Laplaciánu \mathcal{G} a \mathcal{D}	25
15	Ukázka dosažení konsensu pro neorientovaný a orientovaný graf	26
16	Dosažení konsensu pro virtuální leadry s různými hodnotami parametru c .	28
17	Dosažení konsensu pro externího leadra s různými hodnotami parametru c	29
18	Simulace konsensu pro tři agenty	30
19	Simulace konsensu pro čtyři agenty	30
20	Simulace konsensu pro šest agentů	31
21	Vlastní čísla $L(\mathcal{G}_i)$	31
22	Ukázka rozdělení agentů do trojúhelníkové formace z náhodných počátečních pozic	33
23	Ukázka rozdělení agentů do čtvercové formace z náhodných počátečních pozic	34
24	Ukázka rozdělení agentů do hexagonové formace z náhodných počátečních pozic	34
25	Ukázka rozdělení agentů do oktagonové formace z náhodných počátečních pozic	35
26	Ukázka rozdělení agentů do trojúhelníkové formace a její udržování v čase	39
27	Ukázka rozdělení agentů do čtvercové formace a její udržování v čase . . .	40
28	Ukázka rozdělení agentů do hexagonové formace a její udržování v čase . .	40
29	Ukázka rozdělení agentů do oktagonové formace a její udržování v čase . .	41
30	Ukázka rekonfigurace z oktagonové do hvězdicové formace	42
31	Ukázka rekonfigurace z oktagonové do hvězdicové formace s $k_2 = -2$	42

Seznam tabulek

1	Vlastní čísla matic $L(\mathcal{G}_i)$	23
2	Vlastní čísla matic $L(\mathcal{G})$ a $L(\mathcal{D})$	25
3	Vlastní čísla matic $(P + cL)$ a $(I + cL)$	27
4	Vlastní čísla matice $c(G + L)$	29
5	Vlastní čísla matic Laplaciánů jeho rozšířených variant a)	45
6	Vlastní čísla matic Laplaciánů jeho rozšířených variant b)	45
7	Vlastní čísla $A + BKL$	46

Odkazy

- [1] Shalimoon Anokina. “Cooperative Control of Multi-Agent Systems: Consensus, Flocking and Formation Control”. Master Thesis. California State University, Northridge, 2018.
- [2] R. Balakrishnan a K. Ranganathan. *A Textbook of Graph Theory*. Springer Berlin Heidelberg, 2013. ISBN: 9781461264224. URL: <https://books.google.cz/books?id=WvhfmwEACAAJ>.
- [3] Mary Berna-Koes, Illah Nourbakhsh a Katia Sycara. “Communication Efficiency in Multi-Agent Systems”. In: sv. 3. Led. 2004, 2129–2134 Vol.3. ISBN: 0-7803-8232-3. DOI: 10.1109/ROBOT.2004.1307377.
- [4] Space Exploration Technologies Corporation. *Starlink Project introduction*. 2023. URL: <https://www.starlink.com/technology> (cit. 21.04.2023).
- [5] J.A. Fax a R.M. Murray. “Information flow and cooperative control of vehicle formations”. In: *IEEE Transactions on Automatic Control* 49.9 (2004), s. 1465–1476. DOI: 10.1109/TAC.2004.834433.
- [6] Petrella Margaret Foreman Christina Keen Matthew. “FHWA Research and Technology Evaluation: Truck Platooning”. In: (2021). URL: <https://rosap.ntl.bts.gov/view/dot/57272>.
- [7] Roger A. Horn a Charles R. Johnson. *Matrix Analysis*. 2nd. USA: Cambridge University Press, 2012. ISBN: 0521548233.
- [8] Pawan Kalyani. “”Internet From Sky: Starlink”, An Empirical Study on The Introductory Offer from Starlink In Pandemic Situation -Its Competition, Opportunity and Future in one of the world’s biggest consumer Market -India”. In: 8 (břez. 2021), 2394–8124 JMEIT. DOI: 10.5281/zenodo.4733198.
- [9] Kubíček Karel. “Modelově orioentovaný vývoj softwaru: řízení spojky automatické převodovky kamionů”. Master Thesis. Západočeská univerzita v Plzni, 2019. URL: <http://hdl.handle.net/11025/39264>.
- [10] H. Kitano. “RoboCup Rescue: a grand challenge for multi-agent systems”. In: *Proceedings Fourth International Conference on MultiAgent Systems*. 2000, s. 5–12. DOI: 10.1109/ICMAS.2000.858425.
- [11] Karel Kubíček, Martin Čech a Jan Škach. “Continuous enhancement in model-based software development and recent trends”. In: *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2019, s. 71–78. DOI: 10.1109/ETFA.2019.8869237.
- [12] Karel Kubíček a Jindřich Wolf. “Distributed method for Economic Dispatch Problem in power network with multiple uncertainties”. In: *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2022, s. 1–8. DOI: 10.1109/ETFA52439.2022.9921437.

- [13] G. Lafferriere, J. Caughman a A. Williams. “Graph theoretic methods in the stability of vehicle formations”. In: *Proceedings of the 2004 American Control Conference*. Sv. 4. 2004, 3729–3734 vol.4. DOI: 10.23919/ACC.2004.1384492.
- [14] Frank Lewis et al. *Cooperative Control of Multi-Agent Systems: Optimal and Adaptive Design Approaches*. Led. 2014. ISBN: 978-1-4471-5573-7. DOI: 10.1007/978-1-4471-5574-4.
- [15] Mehran Mesbahi a Magnus Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. STU - Student edition. Princeton University Press, 2010. ISBN: 9780691140612. URL: <http://www.jstor.org/stable/j.ctt1287k9b> (cit. 13.04.2023).
- [16] J. von Neumann a O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1947.
- [17] Reza Olfati-Saber, J. Alex Fax a Richard M. Murray. “Consensus and Cooperation in Networked Multi-Agent Systems”. In: *Proceedings of the IEEE* 95.1 (2007), s. 215–233. DOI: 10.1109/JPROC.2006.887293.
- [18] Jeremy Pitt et al. “Voting in Multi-Agent Systems.” In: *Comput. J.* 49 (led. 2006), s. 156–170.
- [19] A. Sujil, Jatin Verma a Rajesh Kumar. “Multi Agent System: Concepts, Platforms and Applications in Power Systems”. In: *Artif. Intell. Rev.* 49.2 (ún. 2018), s. 153–182. ISSN: 0269-2821. DOI: 10.1007/s10462-016-9520-8. URL: <https://doi.org/10.1007/s10462-016-9520-8>.
- [20] Robin J. Wilson. *Introduction to Graph Theory*. New York: Prentice Hall/Pearson, 2010. ISBN: 027372889X 9780273728894.
- [21] Jindřich Wolf. “Řízení kolaborativních multi-agentních dynamických systémů”. diplomathesis. West Bohemia University, 2019.
- [22] Ping Xuan a Victor Lesser. “Multi-agent policies: From centralized ones to decentralized ones”. In: led. 2002, s. 1098–1105. DOI: 10.1145/545056.545078.
- [23] Vilém Žán, Karel Kubíček a Martin Čech. “Design of robust PI controller by combining robustness regions with time-domain criteria”. In: *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2022, s. 1–8. DOI: 10.1109/ETFA52439.2022.9921544.