

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

**Řízení kolaborativních
multi-agentních dynamických
systémů**

Plzeň 2019

Bc. Jindřich Wolf

**Místo této strany bude
zadání práce.**

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 27. srpna 2019

Bc. Jindřich Wolf

Poděkování

Tímto bych rád poděkoval vedoucí této diplomové práce, slečně RNDr. Janě Königsmarkové, za odbornou pomoc, nadměrnou dávku trpělivosti a velmi cenné rady, bez nichž by tato práce nespátřila světlo světa. Velké díky patří také mé rodině a přátelům, kteří mi během tvůrčího procesu poskytovali velkou psychickou podporu.

Abstract

This master's thesis studies the problem of multi-agent collaborative control. The aim is to show how to ensure the stability through a suitable connection of individual agents, followed by the control design that put the requirements on the whole system. For the higher-order linear systems, it is shown how to stabilize the individual agents by the state feedback and how to control the whole system with respect to the connection of its agents. Furthermore, it is shown how the designed state feedback does affect the movement of the whole system. Further, it is discussed the comparison of the design methods (linear quadratic regulator and Jordan form assignment).

Key words

multi-agent collaborative control, consensus protocol, state space, state feedback, Jordan form assignment, linear quadratic regulator (LQR), graph theory, collaborative control of formation, collaborative control with a virtual leader

Abstrakt

Tato diplomová práce se zabývá problémem řízení multi-agentních systémů. Cílem je ukázat, jak u takových systémů zajistit stabilitu pomocí správného propojení dílčích agentů a následně navrhnout řízení, díky němuž je možné klást na celý systém další nároky jako uspořádání do formace či následování trajektorie celé formace. Pro lineární systémy vyšších řádů je dále ukázáno, jak je možné je současně stabilizovat pomocí stavové zpětné vazby a celý systém následně řídit při respektování propojení agentů. Na tyto multi-agentní systémy jsou dále aplikovány metody návrhu stavové zpětné vazby pomocí lineárního kvadratického regulátoru LQR a přiřazením Jordanovy formy, u kterých jsou diskutovány důsledky jejich použití na řízení multi-agentních systémů.

Klíčová slova

multi-agentní řízení, consensus protokol, stavový popis, stavová zpětná vazba, přiřazení Jordanovy formy, lineární kvadratický regulátor (LQR), grafová teorie, řízení do formace, řízení s virtuálním leaderem

Obsah

1 Úvod	8
1.1 Řízení multi-agentních systémů	9
1.2 Synchronizace v přírodě	10
1.3 Implementace Reynoldsových pravidel na dynamické systémy .	11
2 Přípravná kapitola	14
2.1 Základy grafové teorie	14
2.2 Algebraická grafová teorie	16
2.3 Stavový popis lineárního t-invariantního systému	22
2.3.1 Stavová zpětná vazba	22
3 Řízení multi-agentních systémů	25
3.1 Systém s dynamikou prvního řádu	25
3.1.1 Řízení agentů popsaných vektorem stavů	28
3.1.2 Consensus protokol a řízení do požadované formace . .	30
3.2 Řízení s leadrem	31
3.2.1 Řízení s virtuálním leadrem	35
3.2.2 Řízení s externím leadrem	38
3.3 Lineární systémy vyšších řádů	45
3.3.1 Consensus protokol a řízení do požadované formace . .	50
3.3.2 Consensus protokol a řízení do požadované formace s virtuálním leadrem	51
3.3.3 Consensus protokol a řízení do požadované formace s ex- terním leadrem	57
3.3.4 Návrh stavové zpětné vazby	61
4 Závěr	70
Literatura	71

1 Úvod

Multi-agentní kolaborativní řízení představuje jeden ze způsobů, jak řídit více samostatných strojů, aby vzájemnými interakcemi byly schopny plnit společné cíle. Pojmem agent označujeme systém, později subsystém, který reprezentuje jeden autonomní celek ze skupiny (např. kvadrokoptéra, vozidlo či termostat). Provázání těchto agentů vazbami (např. informační, mechanické) vytvoří společnou skupinu, kterou dále budeme nazývat celkový systém. Způsob provázání agentů tvoří síť, která bude označována pojmem topologie a popisována pomocí grafu.

Kolaborativní řízení se zabývá především vzájemnými vazbami mezi těmito agenty, pomocí kterých jsou jako skupina schopny řešit úkoly, které by jako jednotlivci nezvládly nebo by pro ně byly obtížnější. Zatímco pojmy stabilita a optimalita jsou v klasické teorii řízení dobře prozkoumány, v teorii multi-agentního řízení je pro stabilitu celkového systému důležitá nejen stabilita a dynamika dílčích agentů, ale i vzájemná topologie propojení.

Hlavní výhodou tohoto přístupu je decentralizovanost řízení, která přináší výhody i se vzrůstajícím počtem agentů. V centralizovaném řízení, kde veškerý komunikační tok prochází jedním uzlem, nastává problém s velkou výpočetní a komunikační náročností. Se vzrůstajícím počtem agentů se zvyšuje náročnost na tohoto agenta, která v mnoha případech není splnitelná. Nevýhodou centralizace je i přílišná závislost na jednotlivci, kde pokud se poškodí centrální agent, upadá funkčnost celé skupiny. Těmto problémům se dá u decentralizovaného řízení předejít vhodným návrhem topologie systému.

Multi-agentní řízení zaznamenalo velký rozmach v posledním desetiletí, kde díky rozvoji výpočetní techniky a komunikace, vznikla široká škála možností, jak jej uplatnit. Aplikace zasahují od řízení energetických sítí, optimalizace výpočetního procesu až po autonomní vozidla. Mezi hlavní problémy řešené tímto řízením patří *flocking* (shlukování), synchronizace, *rendezvous* (shoda na poloze i čase) a consensus protokol.

Pojem *flocking* se využívá při řízení pohyblivých agentů a představuje sdružení skupiny. Synchronizace představuje problém sjednocení především z časového pohledu, kde se jedná zejména o synchronizaci frekvence. *Rendezvous* reprezentuje problém shody v poloze a zároveň v čase, např. společný útok.

Consensus, tedy problém obecné shody, představuje sjednocení celého systému na společném cíli, hodnotě či stavu. U pohyblivých agentů (např. kvadrokoptér) to může být shoda na stejné výškové hladině. V energetice se tento způsob řízení používá pro sjednocení elektráren zapojených do sítě, kde je zapotřebí

sesynchronizovat frekvenci všech dílčích elektráren (agentů) pro maximalizaci výkonu. Ve vesmírném průmyslu je snaha využít toto řízení pro systémy tvořené soustavou teleskopů, jejichž vzájemné provázání a správné natočení povede ke kvalitnějšímu snímání oblohy.

Pojmem consensus protokol dále popisujeme způsob řízení, který řeší obecný problém shody. Základem consensu protokolu je distribuovaný zákon řízení založený na informaci pouze lokálního okolí, který však vede na kolektivní synchronizované řízení celé skupiny. Inspirace pro tento způsob řízení vznikla z pozorování přírody, kde probíhá kolektivní přirozená synchronizace v oblastech biologie, sociologie, fyziky i chemie.

Tato práce cílí na využití consensu v oblasti řízení dynamických systémů popsaných druhým Newtonovým pohybovým zákonem [4], kterými lze zjednodušeně popsat autonomní pohyblivé objekty (kvadrokoptéry, různé druhy pozemních vozidel apod.). Při řešení problémů v této aplikaci ukážeme, jak docílit consensu i pro nestabilní systémy, regulovat skupinu na požadovanou formaci či docílit pohybu skupiny po zadané trajektorii. Na závěr této práce je pak nastíněno, jak volit stavovou zpětnou vazbu při respektování dynamiky systému tak, abychom dosáhli námi zvolené optimality (například nejkratší trajektorie).

1.1 Řízení multi-agentních systémů

Vývoj multi-agentních systémových technologií a jejich způsob řízení se datuje do 80. let 19. století, kde se vývoj soustředil na využití v oblasti mobilních robotů. Do té doby se teorie řízení soustředila především na řízení jednoho robota nebo na distribuované řešení problémů, které ale neobsahovalo robotické komponenty. Jednou z prvních aplikací byl projekt *California's Partners for Advanced Transit (PATH)* [7], který se soustředil na skupinové řízení automobilů. S pokrokem v komunikačních technologiích a nárůstem výpočetních kapacit se počátkem 21. století rozrostla snaha využít tuto teorii v letectví. Především v řízení *unmanned aerial vehicles (UAVs)*, což je skupina létajících objektů, do které spadají například drony. V posledních deseti letech si tato teorie řízení našla široké uplatnění v armádních aplikacích a transportních systémech [6].

Součástí dnešní armádní techniky jsou jak pilotované tak nepilotované stroje. Zatímco u pilotovaných strojů řeší všechna rozhodnutí člověk, u autonomních strojů je potřeba určit nějaký zákon řízení. Ve válečném prostředí čelí autonomní stroje komplexnímu nasazení a jsou na ně kladeny velké nároky. V takových podmínkách je důležité využívat decentralizovaného řízení, aby při poruše jednoho stroje nedošlo k vyřazení celé jednotky. Je však důležité sjednotit všechny stroje k jednomu cíli, proto se většinou u těchto aplikací využívá centralizované loka-

lizace (domluvě na společném cíli či množině cílů) následované decentralizovaným provedením (každý stroj je zodpovědný za svůj cíl či množinu cílů). Multi-agentní řízení se v této oblasti používá převážně pro řízení na formaci (bezpilotní letouny) a pro řešení *rendezvous*, které hraje významnou roli v řízení naváděných střel.

Velké využití multi-agentního řízení zaznamenala v posledních desetiletích i oblast transportních systémů, ať se jedná o pozemní či vzdušné zásobování, automatizace tohoto odvětví by byla hodně využívána. Překážkou v plné automatizaci je komplexnost celého problému, kde je potřeba zaručit nejen vhodně navržený systém ve smyslu topologie a stability, ale i adaptivnost chování na vnější nečekané podněty.

V případě pozemního transportu je jedním z největších projektů tzv. *intelligentní dálnice*. Jedná se o automatický transportní systém využívající distribuované zákony řízení k řešení problémů nastávajících s přepravou nákladu. Jedním ze způsobů, jak tyto problémy řešit, je zvýšení interakcí mezi jednotlivými vozidly. To by vhodnou aplikací vedlo na snížení zamezení kolizí mezi vozidly, vyhnutí se překážce a zvýšení výkonnosti dálnice (ukázka projektu *PATH*, který využitím automatických přepravních prostředků snížil vzdálenost mezi vozidly a tím zvýšil hustotu provozu [7]). Hlavní překážkou v realizaci tohoto projektu jsou komunikační systémy, u kterých je obtížné zaručit spolehlivost a tím i stabilitu celého systému.

V leteckém průmyslu se automatizace potýká s podobnými problémy jako na zemi. S přibývajícím hustotou vzdušného prostoru roste snaha o zautomatizování velkých letišť a leteckého prostoru obecně. V praxi už jsou nasazovány automatické systémy, které varují před možnou kolizí létajících objektů. Letové řídicí systémy příští generace by pravděpodobně mohly přejít z lidmi řízené centralizované struktury na více distributivní automatické řízení, umožňující létajícím objektům létat po libovolné dráze spíše než v předem daném vzdušném koridoru.

1.2 Synchronizace v přírodě

Velkou inspirací pro multi-agentní řízení bylo pozorování přirozené synchronizace v přírodě. Jedním z nejjasnějších příkladů je kolektivní chování shluků zvířat. Ať se jedná o hejno ptáků, ryb či stádo koní, kde ačkoliv se každý živočich chová a hýbe sám za sebe, navenek se skupina jeví jako jeden celek. Toto společné chování využívají skupiny zvířat ke zvýšení šancí při lovu, zvýšení povědomí o nebezpečí nebo k úspoře pohybové energie. V takové skupině reaguje každý člen sám za sebe a jeho chování je možné zachytit pomocí několika pravidel. Tato pravidla jsou určována nejbližším okolím jedince, neboť má smyslový kontakt (na-

příklad vizuální) pouze se svými sousedy a o zbytku stáda nemusí mít přehled. V souladu s těmito omezeními jsou pravidla určující chování jedinců chápána jako vyvážení dvou opačných požadavků, vyhnutí se překážce a zachování blízkosti ke zbytku stáda.

Pozorováním těchto úkazů v přírodě zachytil Reynolds [9] zákony chování jednotlivců ve skupině pomocí tří pravidel:

1. Vyhnutí se kolizi
2. Shoda rychlosti a směru pohybu s okolím
3. Udržení blízké vzdálenosti s okolím

Okolím se v tomto případě myslí zbytek stáda či hejna, o kterém má daný jedinec přehled. Tato pravidla zachycují velmi dobře rozhodování jednotlivých členů (budeme značit též jako agentů) stáda v kolektivním pohybu a proto se dají dobře aplikovat i na inženýrské problémy, kde se najde jejich využití například v řízení vozidel ve formaci.

Pohyb výsledné skupiny, která se řídí těmito pravidly, je významně ovlivněn počtem sousedů, na který každý agent vidí a způsobem jakým je informace předávána. Tyto vlastnosti určují rychlost celého shluku a později i stabilitu celého systému. Například přenos informace ve stádu koní je především vizuální, kde má každý jedinec přehled jen o pár nejbližších sousedech, zatímco ve shluku ryb je významná část informace přenášena pomocí vodního tlaku, který vzniká pohybem všech členů hejna. Díky této vlastnosti mají ryby přehled o větším počtu sousedů a jsou schopny rychlejších reakcí. Tento jednoduchý příklad demonstruje, jak počet okolních sousedů působí na dynamiku celé skupiny. Jak je dokázáno například v literatuře [2], s větším počtem sousedů (s větší provázaností) roste rychlost celé skupiny.

1.3 Implementace Reynoldsových pravidel na dynamické systémy

Uvedená Reynoldsova pravidla [9] udávají způsob rozhodování jednotlivců v pohybující se skupině. Celkové chování vychází z provázanosti celé skupiny. Jednotliví členové skupiny budou dále značeni jako agenti (uzly) a jejich bezprostřední komunikační okolí (ostatní agenti, s kterými si vyměňují nějakou informaci) budeme nazývat sousedy.

Dále budeme uvažovat pohyb v rovině (x, y) , kde poloha každého agenta bude dána jako $x_i = [p_i \ q_i]^T \in \mathbb{R}^2$. Informační tok pro všechny agenty bude vždy obousměrný. Pro implementaci prvního pravidla, vyhnutí se kolizi, zavedeme

okolí každého agenta δ_C . Cílem každého agenta je zachovat toto okolí prázdné $N_i^c = \{j : r_{ij} \geq \delta_C\}$, kde vzdálenost mezi agenty i a j je dána následovně:

$$r_{i,j} = |x_j - x_i| = \sqrt{(p_j - p_i)^2 + (q_j - q_i)^2}.$$

Dále zavedeme komunikační okolí $\delta_V > \delta_C$, kde je interakce každého agenta dána jako $N_i = \{j : r_{ij} \geq \delta_V\}$. Tyto množiny okolí jsou různé pro různé agenty (zvířata, stroje). Zároveň ani nemusí být toto okolí vždy definované jako kruhové.

Dynamika chování je též odlišná pro různé skupiny zvířat či strojů, nicméně se předpokládá stejná či podobná dynamika pro všechny členy skupiny. Použijeme tedy pro ukázkou velice jednoduchý dynamický popis, který však stačí pro dosažení realistických výsledků. Dynamika jednoho agenta v dvourozměrném prostoru bude dána následovně:

$$\dot{x}_i = u_i, \quad x_i = [p_i \quad q_i]^T, u_i = [u_{pi} \quad u_{qi}]^T \in \mathbb{R}^2.$$

Odpovídající zákon řízení pro vyhnutí se kolizi je dán jako:

$$u_i = - \sum_{j \in N_i^c} c_{ij}(x_j - x_i). \quad (1.1)$$

Toto řízení způsobí, že pokud se nachází nějaký agent v okolí N_i^c *i-tého* agenta, agent i se odvrátí. Konstanta c_{ij} je váhové ohodnocení informační vazby mezi danými agenty.

Zákon řízení popisující druhé pravidlo, udržení shluku (centrování), lze zapsat jako:

$$u_i = \sum_{j \in N_i \setminus N_i^c} a_{ij}(x_j - x_i), \quad (1.2)$$

který způsobí, že se agent i přikloní k ostatním agentům, kteří jsou v jeho komunikačním okolí N_i , ale zároveň nejsou v kolizním okolí N_i^c . Shoda rychlosti a pohybu s ostatními agenty je dána podobným principem řízení, kde:

$$\begin{aligned} \dot{p}_i = u_{pi} &= \sum_{j \in N_i \setminus N_i^c} a_{ij}(p_j - p_i), \\ \dot{q}_i = u_{qi} &= \sum_{j \in N_i \setminus N_i^c} a_{ij}(q_j - q_i). \end{aligned}$$

Výsledné řízení působící na *i-tého* agenta je dáno jako součet zákonů řízení (1.1) a (1.2).

$$u_i = \sum_{j \in N_i \setminus N_i^C} a_{ij}(x_j - x_i) - \sum_{j \in N_i^C} c_{ij}(x_j - x_i).$$

Tento zákon řízení je znám jako *lokální*, protože rozhodování každého agenta závisí na rozdílu jeho stavu a stavů jeho sousedů, kde se snaží docílit nulového rozdílu, čímž zajistí shodu (consensus) se svými sousedy. Skupinu pohybujících se živočichů lze pomocí tohoto řízení poměrně přesně popsat, čímž je zajištěno kolektivní chování. Nedostatkem tohoto popisu je absence vazby na okolí, neboť každý agent je definován pouze lokálním řízením.

Stejně jako v přírodě, je potřeba i při popisu skupiny zavést nějakého velitele, neboli *leadra*. Tímto termínem je označován agent, který má vazbu na globální okolí, ve kterém se skupina pohybuje. Jeho úkolem je vést celou skupinu k nějakému dalšímu společnému cíli (např. směr letu během migrace). Dále je možné definovat nějakou formaci, ve které se má skupina vyskytovat. Například při letání migrujících ptáků se hejno snaží držet formaci písmene V, což jim umožňuje šetřit energii. K šetření energie dochází díky turbulentnímu víření, které vzniká při průchodu křídla vzduchem a následně je použito jako vztlačková síla pro dalšího souseda ve formaci.

2 Přípravná kapitola

Tato kapitola připomíná důležité pojmy, které jsou v práci dále využity. Nejdříve jsou uvedeny některé pojmy z grafové teorie [5], jejíž aparát je hojně využíván při řešení consensu. Následuje provázání grafové a algebraické teorie [5], která je využívána pro maticový popis grafových vlastností. Další kapitola se věnuje stavovému popisu lineárního t -invariantního systému a vybraným metodám, pomocí kterých lze měnit jeho dynamické chování [13].

2.1 Základy grafové teorie

Definice 2.1. Graf je uspořádaná dvojice $G = (U, H)$, kde U , $|U| = N$, je neprázdná konečná množina uzlů (někdy vrcholů) a H , $|H| = M$, je konečná množina hran splňující vlastnost $H \subset \binom{U}{2} \cup U^2$. Graf se nazývá neorientovaný, pokud $H \subset \binom{U}{2}$ a orientovaný, jestliže $H \subset U^2$.

V této práci se dále budeme zabývat výhradně neorientovanými grafy. Pro hrany v neorientovaném grafu platí vlastnost $H \subseteq \{\{a, b\} | a, b \in U, a \neq b\}$. Vlastnost $a \neq b$ říká, že graf neobsahuje smyčky čili neexistuje hrana, která začíná i končí ve stejném vrcholu. Grafům obsahující tuto množinu se někdy říká pseudografy, kterými se však zabývat nebudeme.

Definice 2.2. Stupeň $d_G(u)$ uzlu u v grafu G je počet hran grafu G , které obsahují uzel u neboli $d_G(u) = |\{v \in U(G), \{v, u\} \in H(G)\}|$.

Definice 2.3. Sled v grafu G je posloupnost vrcholů taková, že mezi každými dvěma po sobě jdoucími vrcholy je hrana.

Definice 2.4. Tah v grafu G je sled, ve kterém se neopakují hrany.

Definice 2.5. Cesta v grafu $G = (U, H)$ je sled, ve kterém se neopakují vrcholy (a tedy ani hrany). Označuje tedy posloupnost $P = (u_0, h_0, u_1, h_1, \dots, u_n, h_n)$, kde navíc platí $u_i \neq u_j$ pro $i \neq j$.

Definice 2.6. Graf G je souvislý, pokud mezi každými dvěma vrcholy existuje cesta neboli pro každé $u, v \in U(G)$ existuje v grafu G sled z u do v . V takovém grafu též platí, že $M \geq N - 1$.

Definice 2.7. Graf G je úplný, pokud jsou v něm všechny dvojice vrcholů spojené hranou neboli $G = (U, H)$ je úplný pokud $|H| = \binom{|U|}{2}$.

Definice 2.8. Graf se nazývá regulární, pokud stupeň všech jeho vrcholů je stejný. Pro grafy s uzly stupně k se graf nazývá k -regulární.

Definice 2.9. Graf G_1 se nazývá podgrafem grafu G , jestliže $U(G_1) \subset U(G)$ a $H(G_1) \subset H(G)$.

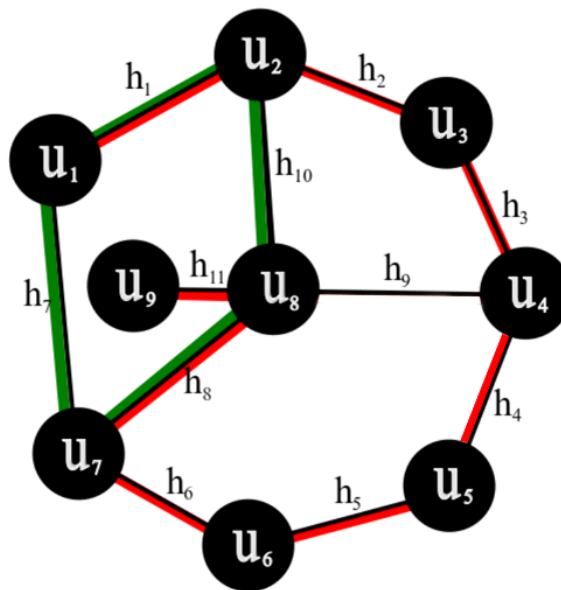
Definice 2.10. Podgraf G_1 se nazývá faktorem grafu G , jestliže $U(G_1) = U(G)$ a $H(G_1) \subset H(G)$.

Definice 2.11. Neorientovaná uzavřená cesta v grafu G se nazývá kružnice. Graf s vrcholy $U = \{u_1, u_2, \dots, u_n\}$ a hranami $H = \{\{u_1, u_n\}, \{u_1, u_2\}, \dots, \{u_{n-1}, u_n\}\}$, pro $n \geq 3$ se nazývá kružnice.

Definice 2.12. Graf G , který je souvislý a jako podgraf neobsahuje žádnou kružnici, se nazývá strom.

Definice 2.13. Podgraf grafu G obsahující všechny jeho vrcholy, který je zároveň stromem, se nazývá kostrou grafu.

Příklad 2.1. Na grafu G na obrázku 2.1 jsou demonstrovány některé uvedené pojmy.



Obrázek 2.1: Ukázka neohodnoceného, neorientovaného grafu G , s vyznačenými pojmy kostra (červeně) a kružnice (zeleně).

Graf G je souvislý, neboť z každého uzlu existuje cesta, posloupnost dvojic $\{h_i, u_i\}$, která vede do všech ostatních vrcholů grafu G . Posloupnost $P =$

$\{u_1, h_1, u_2, h_2, u_3, h_3, u_4, h_4, u_5, h_5, u_6, h_6, u_7, h_8, u_8, h_{11}, u_9\}$ představuje kostru grafu G , která je v obrázku 2.1 vyznačená červeně.

Posloupnost $C = \{u_1, h_1, u_2, h_{10}, u_8, h_8, u_7, h_7\}$, označená zeleně, reprezentuje kružnici.

2.2 Algebraická grafová teorie

Definice 2.14. Čtvercová matice $\mathbf{M} \in \mathbb{R}^{N \times N}$ je symetrická, pokud platí:

$$\mathbf{M} = \mathbf{M}^T.$$

Definice 2.15. Vlastní vektor čtvercové matice \mathbf{M} je takový nenulový vektor u , pro který existuje číslo λ takové, že platí:

$$\mathbf{M}u = \lambda u.$$

Číslo λ označuje vlastní číslo matice \mathbf{M} a u označuje její vlastní vektor. Vlastní čísla matice \mathbf{M} jsou kořeny polynomu, který lze vyjádřit následovně:

$$\det(\lambda \mathbf{I} - \mathbf{M}) = 0,$$

kde \mathbf{I} představuje identickou matici odpovídajících rozměrů a zápis $\det()$ označuje výpočet determinantu.

Definice 2.16. Matice $\mathbf{M} \in \mathbb{R}^{n \times n}$ se nazývá pozitivně semidefinitní, jestliže $x^T \mathbf{M}x \geq 0$ pro každé $x \in \mathbb{R}^n$, a pozitivně definitní, jestliže $x^T \mathbf{M}x > 0$ pro každé $x \in \mathbb{R}^n \setminus \{0\}$.

Definice 2.17. Nulový prostor matice \mathbf{M} , $N(\mathbf{M})$, někdy označován jako *jádro matice*, $Ker(\mathbf{M})$, je tvořen množinou řešení x^T homogenní soustavy lineárních rovnic:

$$\mathbf{M}x^T = o^T,$$

kde o^T představuje nulový vektor příslušné dimenze matice \mathbf{M} .

Lemma 2.1. Je-li \mathbf{M} reálná symetrická matice, pak každé vlastní číslo matice \mathbf{M} je reálné [10].

Důkaz. Je-li $\lambda \in \sigma(M)$ a vlastní vektor $x \neq 0$ příslušný vlastnímu číslu λ , pak platí:

$$\begin{aligned} \mathbf{M}x &= \lambda x, \\ x^* x &\neq 0, \end{aligned} \tag{2.1}$$

kde symbol x^* značí komplexní sdruženost daného vektoru. Pomocí přechodu ke komplexně konjugovaným maticím získáme z rovnice (2.1) rovnost:

$$\mathbf{M}^* x^* = \bar{\lambda} x^*,$$

kde $\bar{\lambda}$ značí komplexně sdružené vlastní číslo. Z toho plyne:

$$x^* x (\lambda - \bar{\lambda}) = x^* (\lambda - \bar{\lambda}) x = x^* \mathbf{M}x - x^* \mathbf{M}^* x = 0,$$

neboť $\mathbf{M} = \mathbf{M}^*$. Odtud $\lambda = \bar{\lambda}$.

Lemma 2.2. Pro každou matici $\mathbf{M} \in \mathbb{R}^{m \times n}$ je $\mathbf{M}^T \mathbf{M}$ pozitivně semidefinitní. Pokud jsou sloupce matice \mathbf{M} nezávislé, je $\mathbf{M}^T \mathbf{M}$ pozitivně definitní [10].

Důkaz. $\mathbf{M}^T \mathbf{M}$ je symetrická, neboť platí $(\mathbf{M}^T \mathbf{M})^T = \mathbf{M}^T (\mathbf{M}^T)^T = \mathbf{M}^T \mathbf{M}$. Pro každé $x \in \mathbb{R}^n$ platí:

$$x^T \mathbf{M}^T \mathbf{M} x = (\mathbf{M}x)^T (\mathbf{M}x) = \|\mathbf{M}x\|_2^2 \geq 0,$$

kde $\|\mathbf{M}x\|_2^2$ představuje 2-normu příslušného parametru, která z definice normy musí být nezáporná a tedy $\mathbf{M}^T \mathbf{M}$ je pozitivně semidefinitní [8]. Jsou-li sloupce \mathbf{M} lineárně nezávislé, potom platí:

$$(\mathbf{M}x)^T (\mathbf{M}x) = 0 \quad \Rightarrow \quad \|\mathbf{M}x\|_2^2 = 0 \quad \Rightarrow \quad \mathbf{M}x = 0,$$

což znamená, že $x = 0$, tj. $\mathbf{M}^T \mathbf{M}$ je obecně pozitivně definitní.

Nyní uveďme několik pojmů sloužících k maticovému vyjádření některých grafových vlastností.

Definice 2.18. Matice sousednosti grafu $G = (U, H)$ je čtvercová matice $\mathbf{A}_G = (a_{i,j})_{i,j=1}^N$ definovaná předpisem:

$$\begin{aligned} a_{i,j} &= 1 \quad \text{pro } \{u_i, u_j\} \in H, \\ a_{i,j} &= 0 \quad \text{jinak.} \end{aligned}$$

Pokud je graf G úplný, matice \mathbf{A}_G obsahuje na diagonále nuly a jinde samé jedničky. Pro neorientované grafy plyne, že jejich matice sousednosti je symetrická.

Definice 2.19. Uzlo hranová matice (někdy též matice incidence) grafu $G = (U, H)$ je matice velikosti $\mathbf{B}_G \in \mathbb{R}^{N \times M}$ definovaná předpisem $\mathbf{B}_G = (b_{i,j})_{i=1,\dots,N,j=1,\dots,M}$, kde:

$$\begin{aligned} b_{i,j} &= 1, & \text{pro } \{u_i, h_j\} \in (U, H), \\ b_{i,j} &= 0, & \text{jinak.} \end{aligned}$$

Definice 2.20. Matice stupňů (též stupňová matice) je čtvercová matice $\Delta \in \mathbb{R}^{N \times N}$, která má na diagonále stupně jednotlivých uzlů.

Definice 2.21. Pokud je matice \mathbf{A} rozměru $m \times n$ a matice \mathbf{B} je rozměru $p \times q$, pak Kroneckrův součin $\mathbf{A} \otimes \mathbf{B}$ je matice rozměru $mp \times nq$ daná vztahem:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}$$

Pokud jsou matice $\mathbf{A}, \mathbf{B}, \mathbf{C}$ a \mathbf{D} takových rozměrů, že lze utvořit maticové součiny \mathbf{AC} a \mathbf{BD} , pak platí:

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}).$$

Definice 2.22. Matice Laplaciánu (někdy též pouze Laplacián) grafu G je čtvercová matice $\mathbf{L} \in \mathbb{R}^{N \times N}$ definovaná předpisem:

$$\mathbf{L} = \Delta - \mathbf{A},$$

kde Δ je stupňová matice a \mathbf{A} je matice sousednosti téhož grafu G . Matice Laplaciánu grafu G je čtvercová matice $\mathbf{L} \in \mathbb{R}^{N \times N}$ a lze ji též definovat předpisem:

$$\mathbf{L} = \mathbf{EE}^T,$$

kde \mathbf{E} je uzlohranová matice grafu G . Tato definice se hojněji využívá při práci s orientovanými grafy.

Laplacián neorientovaných grafů je symetrický, pozitivně semidefinitní a jeho vlastní čísla jsou reálná a tedy i seřaditelná. Součet všech prvků každé řádky je roven nule a jedno vlastní číslo Laplaciánu je vždy nulové. Pokud je graf G souvislý, pak $\lambda_2 > 0$. Tyto vlastnosti vyplývají z jeho definice.

Symetrii lze dokázat z definice, kde $\mathbf{L} = \Delta - \mathbf{A}$, protože obě matice Δ i \mathbf{A} jsou pro neorientovaný graf symetrické. Pro reálnou symetrickou matici \mathbf{L} z lemmy 2.1 plyne, že každé její vlastní číslo bude reálné. Pozitivní semidefinitnost matice \mathbf{L} plyne z lemmy 2.2, neboť lze Laplacián vyjádřit jako $\mathbf{L} = \mathbf{E}\mathbf{E}^T$.

Pro Laplacián platí:

$$s_i = \sum_{j=0}^c L_{ij} = 0,$$

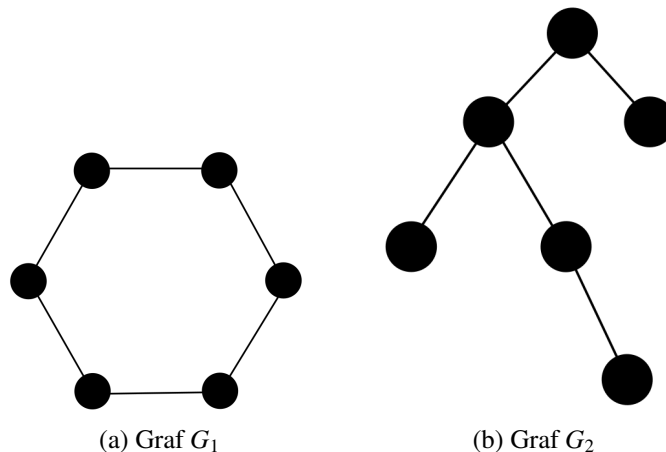
kde c je počet sloupců matice L . Tato vlastnost plyne z definice $\mathbf{L} = \Delta - \mathbf{A}$, neboť:

$$\sum_{j=0}^{c_A} \Delta_{ij} = - \sum_{j=0}^{c_A} \mathbf{A}_{ij}.$$

Tedy součet všech řádek matice \mathbf{L} je roven nule. Z toho plyne, že matice \mathbf{L} obsahuje alespoň jeden lineárně závislý řádek. Potom platí:

$$L\underline{1}c = 0, \quad \underline{1} = [1 \dots 1]^T \in \mathbb{R}^N,$$

kde $\underline{1}$ je vektor jedniček a c je libovolná konstanta. Pro vektor $\underline{1}c$ tak platí, že $\underline{1}c \in N(L)$, kde $N(L)$ je nulový prostor matice L . Z této vlastnosti vyplývá, že alespoň jedno vlastní číslo matice L bude vždy nulové. Pokud je navíc graf G souvislý, matice Laplaciánu \mathbf{L} příslušného grafu bude mít hodnost $\text{rank}(\mathbf{L}) = N - 1$, tudíž velikost báze nulového prostoru $N(L)$ je jedna. Z toho plyne, že pokud je graf G souvislý, bude matice \mathbf{L} obsahovat právě jedno nulové vlastní číslo.



Obrázek 2.2: Ukázka topologie neohodnocených, neorientovaných grafů.

Příklad 2.2. Uvažujme graf G_1 z obrázku 2.2, stupeň každého uzlu je $d_i = 2$, graf představuje kružnici a tedy získáme matice Δ a \mathbf{A} v následujícím tvaru:

$$\Delta = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & -1 \\ -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}.$$

Laplacián grafu G_1 lze vyjádřit jako:

$$\mathbf{L} = \Delta - \mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}.$$

Příklad 2.3. Uvažujme graf G_2 z obrázku 2.2, který představuje strom. Pokud budeme uvažovat orientaci hran od horních uzlů k dolním, získáme uzlohranovou matici \mathbf{E} a následně Laplacián \mathbf{L} grafu G_2 ve tvaru:

$$\mathbf{E} = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{L} = \mathbf{E}\mathbf{E}^T, \quad \mathbf{L} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 3 & 0 & -1 & -1 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

Pokud bychom postupovali podle definice (2.22) (resp. jako v předchozím příkladu 2.2), získáme matice Δ a \mathbf{A} v následujícím tvaru:

$$\Delta = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$\mathbf{L} = \Delta - \mathbf{A} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 3 & 0 & -1 & -1 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

2.3 Stavový popis lineárního t-invariantního systému

Stavový popis je jednou z možností, jak zachytit dynamiku lineárního t-invariantního systému (LTI systém) s respektováním vazeb na jeho okolí. Jedná se o soustavu lineárních diferenciálních rovnic:

$$\begin{aligned}\dot{x}(t) &= \mathbf{A}x(t) + \mathbf{B}u(t), \\ y(t) &= \mathbf{C}x(t) + \mathbf{D}u(t),\end{aligned}\tag{2.2}$$

kde $x(t) \in \mathbb{R}^N$ je stav systému, $\dot{x}(t)$ jeho časová derivace, $u(t) \in \mathbb{R}^M$ je vstup do systému a $y(t) \in \mathbb{R}^P$ je výstup systému. Matice $\mathbf{A} \in \mathbb{R}^{N \times N}$ je matice dynamiky systému, matice $\mathbf{B} \in \mathbb{R}^{N \times M}$ je matice vstupů, matice $\mathbf{C} \in \mathbb{R}^{P \times N}$ je výstupní matice, určující, jaké stavy se projeví na výstupu, a matice $\mathbf{D} \in \mathbb{R}^{P \times M}$ je matice vstupů působících přímo na výstup (někdy též chyby působící na systém).

Stabilita takto popsaného LTI systému je určována pomocí autonomní rovnice:

$$0 = \mathbf{A}x(t).$$

V závislosti na řešení této rovnice rozhodujeme, zda je systém (2.2) stabilní. To nastane tehdy, bude-li mít vlastní čísla matice dynamiky \mathbf{A} zápornou reálnou část.

2.3.1 Stavová zpětná vazba

Stavová zpětná vazba je jeden z možných předpisů řízení stavově popsaného systému (2.2) a je dána předpisem:

$$u(t) = \mathbf{F}x(t),\tag{2.3}$$

kde $\mathbf{F} \in \mathbb{R}^{M \times N}$ je hledaná zpětnovazební matice. Jak tuto matici zvolit, abychom docílili požadovaného chování výsledného systému, můžeme rozhodnout pomocí dále uváděných metod.

Výhodou stavové zpětné vazby je jednoduchost implementace a nulová dynamika regulátoru. Tudiž dynamika systému není zatížena dalšími přechodovými ději regulátoru. Nevýhodou této metody je potřebná znalost všech stavů daného systému, která nemusí být vždy známá. Tento problém se často řeší zavedením rekonstruktoru stavu, který však navyšuje počet stavů výsledného systému a tím se výrazně zvyšuje výpočetní náročnost. Pro naše účely řízení uváděného v kapitole 3 bude postačující stavová zpětná vazba.

Přiřazení Jordanovy formy stavovou zpětnou vazbou

Tato metoda pro návrh stavové zpětné vazby je založena na přiřazení Jordanovy formy výslednému uzavřenému systému. Přiřazení využívá podobnosti matic, kde si návrhář zvolí, jak by měla vypadat výsledná matice dynamiky systému. Následně je pro tento požadavek navržena stavová zpětná vazba. Pro systém (2.2) a stavovou zpětnou vazbu (2.3) dostaneme:

$$\begin{aligned}\dot{x}(t) &= \mathbf{A}x(t) + \mathbf{B}\mathbf{F}x(t), \\ \dot{x}(t) &= (\mathbf{A} + \mathbf{B}\mathbf{F})x(t).\end{aligned}$$

Pro přiřazení požadované matice je potřeba zavést podobnost matic. Matice \mathbf{A} a \mathbf{Z} jsou si podobné, $\mathbf{A} \sim \mathbf{Z}$, právě tehdy, jestliže existuje regulární matice \mathbf{T} , pro kterou platí $\mathbf{A} = \mathbf{T}\mathbf{Z}\mathbf{T}^{-1}$. Nyní je zapotřebí zvolit výslednou matici dynamiky \mathbf{Z} , kterou budeme přiřazovat uzavřenému systému $\mathbf{A} + \mathbf{B}\mathbf{F}$, neboli:

$$\begin{aligned}(\mathbf{A} + \mathbf{B}\mathbf{F}) &\sim \mathbf{Z}, \\ (\mathbf{A} + \mathbf{B}\mathbf{F}) &= \mathbf{T}\mathbf{Z}\mathbf{T}^{-1} \quad / \leftarrow \mathbf{T}, \\ \mathbf{A}\mathbf{T} + \mathbf{B}\mathbf{F}\mathbf{T} &= \mathbf{T}\mathbf{Z}, \\ \mathbf{A}\mathbf{T} - \mathbf{T}\mathbf{Z} + \mathbf{B}\mathbf{F}\mathbf{T} &= 0.\end{aligned}\tag{2.4}$$

Pro řešení této rovnice bylo v [11] ukázáno, že lze zavést substituci $\mathbf{H} = \mathbf{F}\mathbf{T}$ rovnice (2.4), která přejde na rovnici:

$$\mathbf{A}\mathbf{T} - \mathbf{T}\mathbf{Z} + \mathbf{B}\mathbf{H} = 0.\tag{2.5}$$

V [12] je ukázáno, že pro skoro každou matici \mathbf{H} existuje řešení rovnice (2.4).

Pro zajištění regularity výsledné Sylvestrovovy rovnice je potřeba splnění podmínky $\sigma(\mathbf{A}) \cap \sigma(\mathbf{Z}) = \emptyset$. Řešením je pak matice \mathbf{T} , kterou je-li regulární, lze následně využít pro vyjádření výsledné stavové zpětné vazby.

$$\mathbf{F} = \mathbf{H}\mathbf{T}^{-1}.$$

Lineární kvadratický regulátor

Jednou z možností, jak navrhnout stavovou zpětnou vazbu je metoda lineárního kvadratického regulátoru (LQR) [4]. Jedná se o přístup v teorii řízení, kde namísto toho, aby návrhář ladil regulátor na základě požadavků (bezpečnost v zesílení, ve fázi), jsou požadavky nahrazené vhodným kritériem. Tvar tohoto kritéria a jeho řešení následně určuje tvar regulátoru. V našem případě se jedná o lineární kvadratický regulátor, kde volíme váhové matice \mathbf{Q} a \mathbf{R} , jejichž vzájemný poměr určuje váhu konkrétní složky stavu či náročnost využití daného řízení. Toto

kritérium má dva tvary, případ s konečným horizontem a případ s nekonečným horizontem. V této práci jsme využívali případ s nekonečným horizontem, jehož kritérium vypadá následovně:

$$J(u) = \int_0^{\infty} (x^T \mathbf{Q}x + u^T \mathbf{R}u) dt \rightarrow \min, \quad (2.6)$$

kde matice \mathbf{Q} je pozitivně semidefinitní a matice \mathbf{R} pozitivně definitní. Tyto požadavky potřebujeme pro nalezení globálního minima.

Řešení úlohy (2.6) vede na tvar:

$$J = x^T \mathbf{P}x,$$

kde $\mathbf{P} > 0$ je pozitivně definitní matice získaná řešením Riccatiové maticové rovnice:

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} + \mathbf{Q} = 0.$$

Výsledná zpětnovazební matice je ve tvaru:

$$\mathbf{F} = -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}.$$

3 Řízení multi-agentních systémů

Tato práce se zaměřuje na pohybující se dynamické systémy, které je možné popsat Newtonovým druhým pohybovým zákonem a jeho variacemi, tedy $F = ma$, kde F je síla působící na těleso, m je hmotnost tělesa a a je příslušné zrychlení. Cílem této práce je návrh řízení, který pro takto popsané systémy zajistí stabilitu a shodu všech agentů na společné hodnotě (polohy, rychlosti, stavu obecně). Tento požadavek je možné modifikovat a navrhnout řízení, které zaručí shodu na požadované formaci či sledování referenční trajektorie.

Vývoj multi-agentního řízení však nezačínal na takto popsaných systémech, ale postupně se rozvíjel. V této práci je tedy snaha postupovat podobně. Kapitola začíná jednoduchými systémy, které jsou popsány dynamikou prvního řádu. Na těchto systémech jsou ukázány principy consensu protokolu popsané v přípravné kapitole, na které postupně navazují další požadavky řízení (např. řízení do formace). Poté jsou tyto principy odvozeny na systémech druhého řádu, které hrají klíčovou roli v našich aplikacích. Následně je tato forma přepsána do maticového tvaru, což umožňuje názornější aplikaci této teorie na systémy vyšších řádů. Vztahy udávající chování jednotlivých agentů vycházejí z implementace Reynoldsových pravidel, kde se dále soustředíme na druhý a třetí zákon chování.

3.1 Systém s dynamikou prvního řádu

Nejprve uvažujme kooperativní řízení, kde je stavový popis všech jednotlivých agentů (odpovídajících uzlům nějakého grafu G) reprezentován systémem prvního řádu:

$$\dot{x}_i(t) = u_i(t), \quad (3.1)$$

kde $x_i(t)$ a $u_i(t)$ jsou reálné funkce, dále označované jako x_i a u_i .

Cílem je navrhnout řízení u_i pro každého agenta, které usměrní jejich stavy na nějakou společnou hodnotu- tedy dojde ke consensu (shodě). Lokální protokol řízení pro každého agenta je dán vztahem:

$$u_i = \sum_{j=1}^N a_{ij}(x_j - x_i), \quad (3.2)$$

kde a_{ij} je váha hrany v grafu G spojující uzly n_i a n_j . Tento zákon řízení určuje řízení i -tého uzlu pouze na základně znalosti stavu sousedů při respektování

topologie grafu. Tomuto protokolu se říká "*local voting protocol*", jelikož řízení pro každý uzel je dáno rozdílem stavu daného uzlu a všech jeho sousedů. Pokud jsou stavy všech uzlů stejné, pak $\dot{x}_i = u_i = 0$.

Dynamika jednoho uzlu za použití tohoto zákona řízení vypadá následovně:

$$\begin{aligned}\dot{x}_i &= \sum_{j=1}^N a_{ij}(x_j - x_i), \\ \dot{x}_i &= \sum_{j=1}^N a_{ij}x_j - x_i \sum_{j=1}^N a_{ij}, \\ \dot{x}_i &= [a_{i1} \cdots a_{iN}] \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} - x_i d_i.\end{aligned}\tag{3.3}$$

Pro popsání dynamiky celého systému zavedeme vektor stavu všech uzlů $x = [x_1 \cdots x_N]^T \in \mathbb{R}^N$, stupňovou matici $\mathbf{D} = \text{diag}(d_i)$ a matici sousednosti \mathbf{A} definovanou odpovídajícími (informačními) vazbami. Výsledná dynamika celého systému je dána následovně:

$$\begin{aligned}\dot{x} &= \mathbf{A}x - \mathbf{D}x, \\ \dot{x} &= (\mathbf{A} - \mathbf{D})x, \\ \dot{x} &= -(\mathbf{D} - \mathbf{A})x, \\ \dot{x} &= -\mathbf{L}x.\end{aligned}$$

Zákon řízení pro celý systém $u = [u_1 \cdots u_N]^T \in \mathbb{R}^N$ je tedy dán jako:

$$u = -\mathbf{L}x.\tag{3.4}$$

kde \mathbf{L} představuje Laplacián příslušných vazeb popsaných grafem G .

Z tohoto vztahu vyplývá, že výsledné řízení systému závisí na topologii grafu. Klíčové jsou tedy vlastní čísla Laplaciánu, neboť reprezentují dynamiku výsledného systému. Jak bylo uvedeno v předchozí kapitole, vlastní čísla Laplaciánu leží v pravé uzavřené polorovině komplexní roviny a pro neorientovaný graf jsou všechna vlastní čísla reálná. Následující tvrzení jsou uváděna s respektováním těchto předpokladů.

Pokud má matice Laplaciánu hodnost $N - 1$, tj. $\text{rank}(L) = N - 1$, pak $\underline{1}c$, $c \in \mathbb{R}$ je jediný vektor v nulovém prostoru matice L a ustálený stav je dán jako:

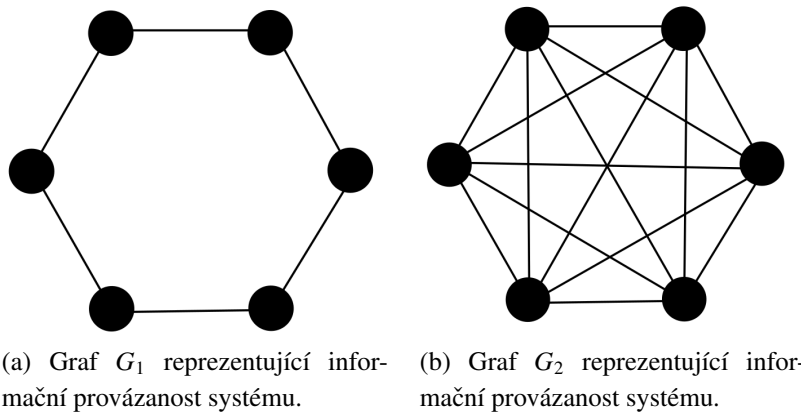
$$\begin{aligned}0 &= -\mathbf{L}x, \\ x &= \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \underline{1}c = \begin{bmatrix} c \\ \vdots \\ c \end{bmatrix}.\end{aligned}$$

Z tohoto vyplývá, že v ustáleném stavu je $x_i = x_j = c \in \mathbb{R}, \forall i, j$. Tudíž je dosaženo konsensu na hodnotě c , kterou je možné pro systémy prvního řádu vyjádřit [4] jako:

$$c = \sum_{i=1 \in N} p_i x_i(0),$$

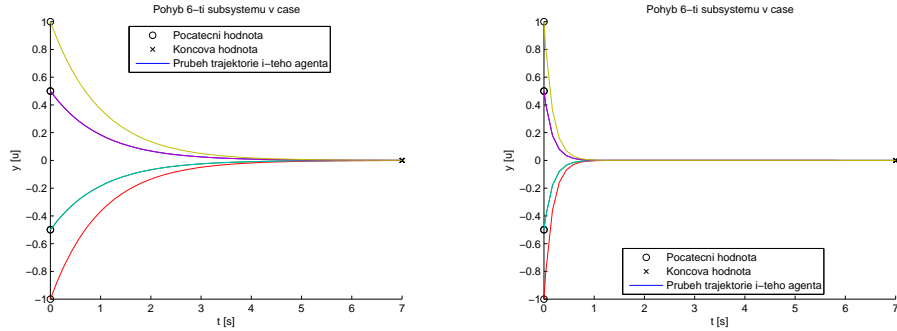
kde $w_1 = [p_1 \cdots p_N]^T$ je normalizovaný levý vektor matice Laplaciánu \mathbf{L} vlastního čísla $\lambda_1 = 0$. Druhé nejmenší vlastní číslo λ_2 matice \mathbf{L} ovlivňuje rychlost (dobu), za níž může dojít ke konsensu. Pro grafové topologie s větší hodnotou vlastního čísla λ_2 je dosaženo konsensu dříve [4]. Pokud matice grafu G neobsahuje kostru, pak matice Laplaciánu \mathbf{L} popisující tento graf má hodnotu $\text{rank}(\mathbf{L}) < N-1$, nulový prostor matice má dimenzi větší než jedna a konsensu není dosaženo.

Dříve bylo ukázáno, že hodnoty vlastních čísel matice \mathbf{L} pro neorientovaný graf G jsou nezáporné a seřaditelné. Jejich hodnota je ovlivněna strukturou grafu G . Hodnota vlastního čísla λ_2 udává rychlost kvergence systému k ustálenému stavu. Čím provázanější je graf G , tím větší je informační tok mezi jednotlivými uzly a tím rychlejší je jeho konvergence k ustálenému stavu. S větší provázaností však také narůstá výpočetní náročnost jednotlivých úzlů a jak je ukázáno v literatuře [2], klesá i robustnost celkového systému. Tento jev vzniká v důsledku příliš mnoha požadavků na jednotlivé uzly, což má za následek oscilace a případnou nestabilitu celého systému.



Obrázek 3.1: Ukázka grafových topologií systému.

Příklad 3.1. Uvažujme systém, jehož agenti jsou popsáni dynamikou z rovnice (3.1) a informační provázanost skupiny je popsána grafy z obrázku 3.1. Průběh polohy pro agenti na počátečních pozicích $x_i(0)$ vypadá následovně:



(a) Časový průběh výchylek jednotlivých agentů, kteří jsou posáni dynamikou prvního řádu a jejich topologie je popsána grafem G_1 z obrázku 3.1.

(b) Časový průběh výchylek jednotlivých agentů, kteří jsou posáni dynamikou prvního řádu a jejich topologie je popsána grafem G_2 z obrázku 3.1.

Obrázek 3.2: Průběh simulací z programového prostředí Matlab, Simulink.

Z průběhu simulací je patrné předchozí tvrzení, že provázanost systému ovlivňuje rychlost kovergence. Uvažované grafy G_1 a G_2 patří do třídy k -regulárních cyklických grafů, což ovlivňuje rychlost konvergence. Vlastní čísla Laplaciánů odpovídajících grafům G_1 jsou:

$$\Lambda(G_1) = \{0, 6, 6, 6, 6, 6\}.$$

Consensu pro graf G_1 je dosaženo přibližně v čase:

$$t_{G_1} \approx 5.4366s.$$

Vlastní čísla Laplaciánu grafu G_2 jsou:

$$\Lambda(G_2) = \{0, 1, 1, 3, 3, 4\}.$$

Consensu je dosaženo přibližně v čase:

$$t_{G_2} \approx 0.9845s.$$

3.1.1 Řízení agentů popsaných vektorem stavů

Do teď bylo v dynamice (3.1) uvažováno, že stav agenta x_i je popsán skalárně, tedy $x_i, u_i \in \mathbb{R}$. Pokud budou stavy jednotlivých agentů popsány vektorově, $x_i, u_i \in \mathbb{R}^n$, bude celkový stav systému, a jeho řízení, definované jako $x = [x_1^T \cdots x_N^T] \in \mathbb{R}^{nN}$, $u = [u_1^T \cdots u_N^T] \in \mathbb{R}^{nN}$. Prvky a_{ij} a d_i z rovnice jsou pomocí Kroneckorva součinu násobeny jednotkovou maticí \mathbf{I}_n velikosti $n \times n$ a dynamika výsledného

systemu lze zapsat následovně:

$$u = -(\mathbf{L} \otimes \mathbf{I}_n)x, \quad (3.5)$$

$$\dot{x} = -(\mathbf{L} \otimes \mathbf{I}_n)x, \quad (3.6)$$

kde \otimes reprezentuje Kroneckerův součin [1]. Systém si uchovává stejné vlastnosti a chování jako ve skalárním případě. Tato modifikace je vhodná při modelování chování systému ve více dimenzích.

Příklad 3.2. Simulace pohybu systému v rovině (x, y) , kde jsou jednotliví agenti popsáni dynamikou (3.1) a informační topologie systému je reprezentována grafem G_1 z obrázku 3.1. Laplaceán popisující topologii systému je identický pro obě osy a jelikož se pohybujeme v rovině, velikost matice \mathbf{I}_n bude 2×2 , tedy $n = 2$.

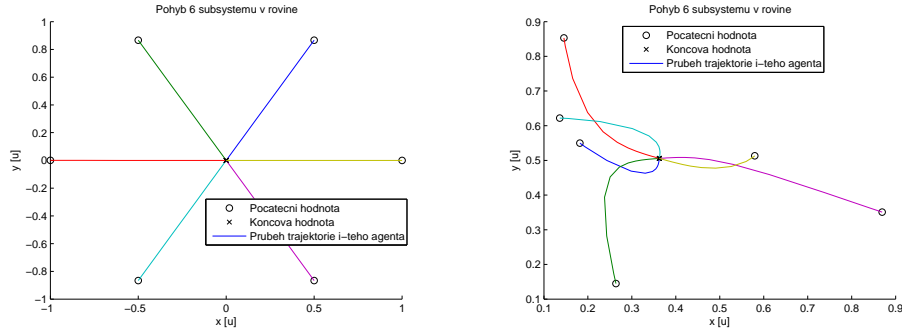
$$\mathbf{L}_x = \mathbf{L}_y = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}, \quad \mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

Dosažením do rovnice (3.5) získáme popis výsledného systému jako:

$$\dot{s} = -(\mathbf{L} \otimes \mathbf{I}_n)s,$$

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_N \\ \dot{y}_1 \\ \vdots \\ \dot{y}_N \end{bmatrix} = - \begin{bmatrix} \mathbf{L}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_y \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \\ y_1 \\ \vdots \\ y_N \end{bmatrix}.$$

Matice Laplaceánu popisující chování celého systému má rozměry 12×12 a stavy systému jsou ve výše uvedeném pořadí.



(a) Počáteční stav uzlů generován po kružnici. (b) Počáteční stav uzlů generován náhodně.

Obrázek 3.3: Simulace průběhu stavů systému pomocí programového prostředí Matlab- Simulink popisující průběh výše popsaného systému.

3.1.2 Consensus protokol a řízení do požadované formace

V předchozí části bylo ukázáno, že pro systémy popsané prvním řádem, které jsou propojeny informačním tokem dle daného grafu G , je celkový systém na mezi stability a pro neorientované silně souvislé grafy konverguje k váženému průměru počátečních hodnot stavu. V případě, že není cílem stabilizovat všechny stavy do jednoho bodu, je potřeba upravit zákon řízení.

Nyní tak přejdeme k úloze, kdy hledáme řízení, které agenti shodně na společném cíli a zároveň si od sebe budou udržovat danou relativní vzdálenost. Existuje více způsobů, jak zavést požadavek formace do řešení konsensu. Uvažujme systém, jehož agenti jsou popsáni dynamikou (3.1), k systémům popsaných rovnicemi (3.4) můžeme přejít obdobným způsobem. Zavedení požadavku relativní vzdáleností mezi agenty, neboli upravením vztahu (3.2), získáme:

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

kde Δ_j představuje požadovanou vzdálenost j -tého agenta od agenta x_i , který je od své pozice vychýlen o Δ_i . Respektive, jejich relativní vzdálenost lze zapsat jako Δ_{ij} , kde:

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

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

Tento zákon řízení lze přepsat do maticového tvaru obdobně jako v rovnici (3.4) a získáme uzavřený systém:

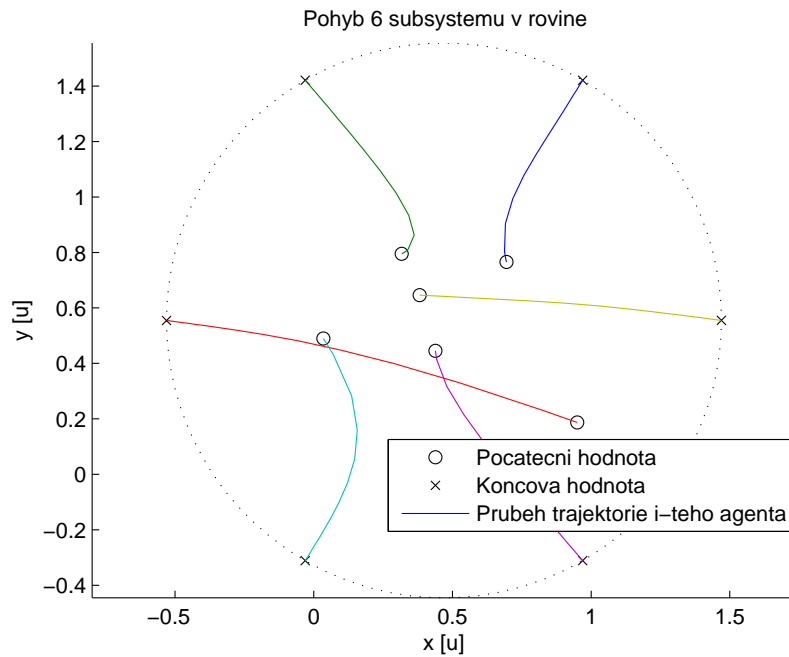
$$\dot{x} = -\mathbf{L}(x - h),$$

kde vektor $h = [\Delta_1 \cdots \Delta_N]^T$ udává vzájemnou relativní polohu všech uzlů. Pro vektor $h = [c \cdots c]^T$ je výsledkem řešení uvedené v předchozí kapitole, neboť relativní vzdálenost jednotlivých uzlů je nulová.

Příklad 3.3. Simulace systému popsaného v příkladu 3.2 s přidáním požadavkem rozmístění do formace ve tvaru kružnice, která byla definovaná vektory:

$$h_x = \left[\frac{1}{2} \quad -\frac{1}{2} \quad -1 \quad -\frac{1}{2} \quad \frac{1}{2} \quad 1 \right],$$

$$h_y = \left[\frac{\sqrt{3}}{2} \quad \frac{\sqrt{3}}{2} \quad 0 \quad -\frac{\sqrt{3}}{2} \quad -\frac{\sqrt{3}}{2} \quad 0 \right].$$



Obrázek 3.4: Simulace s náhodným rozmístěním agentů uvnitř kružnice s požadavkem řízení do kruhové formace.

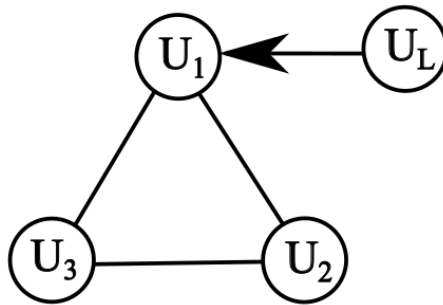
3.2 Řízení s leadrem

Consensus protokol řízení, který byl uvažován doted', nezahrnuje žádné vazby systému s okolím. Chování jednotlivých agentů je dáno lokálním řízením, které

závisí pouze na rozdílu stavů daného agenta a jeho okolních sousedů. V aplikacích, kde je žádoucí interakce skupiny s okolím, je často využíván způsob řízení s leadrem. Pojem leader označuje agenta, jehož dynamika představuje referenční chování, které má skupina kopírovat. Časový průběh jeho stavů tedy udává referenční hodnotu všem agentům skupiny, přičemž řízení leadra není ovlivněno ostatními agenty. Tím v grafu, který popisuje topologii systému, vznikají orientované hrany, čímž se rozšiřuje požadavek pro dosažení consensu celého systému. Aby bylo dosaženo consensu, shody všech agentů, na společné hodnotě (stavu, poloze), musí v grafu G reprezentujícího topologii systému existovat kostra, která obsahuje uzel leadra jako svůj kořen. Tedy musí existovat cesta z leadra ke všem ostatním agentům skupiny.

V této práci se soustředíme pouze na řízení s jedním leadrem, jehož dynamika je totožná s ostatními agenty. Zapojení leadra do skupiny, na které se zaměřujeme, jsou označeny jako *virtuální leader* a *externí (jmenovaný) leader*.

Příklad 3.4. Uvažujme systém, jehož agenti jsou popsáni dynamikou prvního řádu (3.1). Graf G_1 popisující topologii systému vypadá následovně:



Obrázek 3.5: Graf G_1 popisující informáční provázanost agentů s leadrem.

Dynamika agenta U_L plnícího roli leadra je dána předpisem:

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

kde $u_{ref}(t)$ je časový průběh požadované trajektorie, dále značen u_{ref} . Uvažujme lokální zákon řízení i -tého agenta daný předpisem:

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

Oproti předchozím návrhům řízení pro dosažení consensu a řízení na polohu, přibyl návrhový parametr c a k_p . Cílem úlohy je nalézt hodnotu těchto parametrů

takovou, aby systém dosáhl consensu $x_i \rightarrow x_L$ neboli $(x_i - x_L) \rightarrow 0$. Uvažujme chybu i -tého agenta definovanou jako $\tilde{x}_i = x_i - x_L$, potom platí:

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

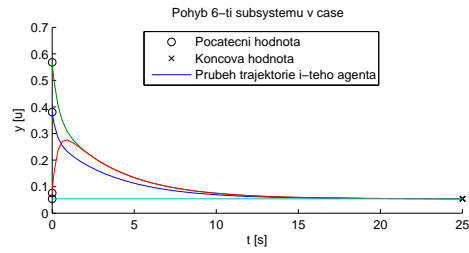
Člen $\sum_{j=1}^N a_{ij}(\tilde{x}_j - \tilde{x}_i)$ lze opět vyjádřit pomocí Laplaciánu \mathbf{L} , jehož tvar vychází z topologie systému. Dynamiku chyby celého systému se stavem $\tilde{x} = [\tilde{x}_1, \dots, \tilde{x}_N, x_L]^T$ lze maticově zapsat:

$$\begin{aligned}\dot{\tilde{x}} &= [\mathbf{I}_n \otimes (-k_p) - c\mathbf{L}]\tilde{x}, \\ \dot{\tilde{x}} &= -[\mathbf{I}_n \otimes k_p + c\mathbf{L}]\tilde{x}.\end{aligned}\tag{3.7}$$

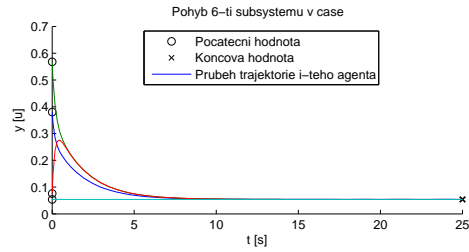
Pro zaručení konvergence chyby k nule musí být výraz (3.7) stabilní, tudíž návrhové parametry musí být voleny kladné, tj. $k_p > 0$, $c > 0$ a graf G , reprezentovaný maticí Laplaciánu \mathbf{L} , musí být souvislý. Parametr k_p obvykle označuje existenci hrany mezi daným agentem a leadrem, proto je volen z množiny $k_p \in \{0, 1\}$. Parametr c se využívá pro ovlivnění rychlosti konvergence, čímž je přímo ovlivněna přesnost sledování referenčního signálu. Dynamika výsledného systému, jehož topologie je popsána grafem G_1 z obrázku 3.5 vypadá následovně:

$$\begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \\ \dot{\tilde{x}}_3 \\ \dot{x}_L \end{bmatrix} = -c \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ x_L \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ u_{ref.} \end{bmatrix}.$$

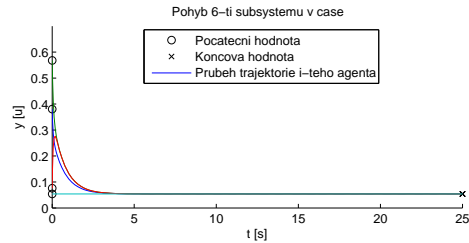
Pokud zvolíme $u_{ref.} = 0$, systém dosáhne consensu na hodnotě $x_L(0)$.



(a) $c = 1$



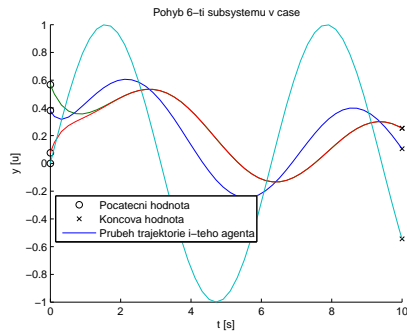
(b) $c = 2$



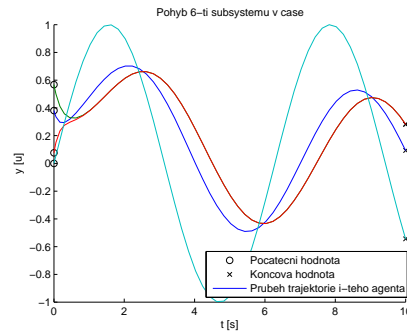
(c) $c = 5$

Obrázek 3.6: Průběh stavů systému popsaného vztahem (3.7) s náhodnými počátečními hodnotami, kde byly voleny různé hodnoty parametru c . Světle modrá čára představuje časový průběh polohy leadra (referenční trajektorie).

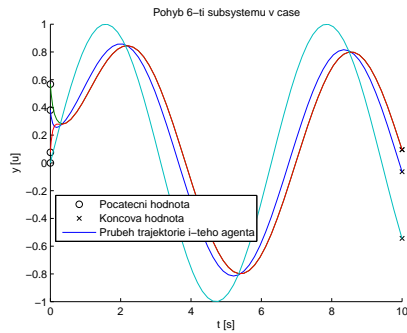
Pokud budeme uvažovat harmonický referenční signál $u_{ref.} = \sin(t)$, volba návrhového parametru c nabyde více na důležitosti.



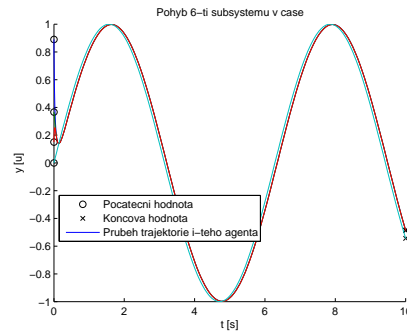
(a) $c = 1$



(b) $c = 2$



(c) $c = 5$



(d) $c = 15$

Obrázek 3.7: Průběh stavů systému popsaného vztahem (3.7) s náhodnými počátečními hodnotami, kde byly voleny různé hodnoty parametru c . Pro sledování nenulového referenčního signálu, který je v grafech reprezentován světle modrou čarou, je patrná důležitost správné volby parametru c .

3.2.1 Řízení s virtuálním leadrem

Tento způsob řízení označuje implementaci leadra, který je chápán jako reprezentace celé skupiny. Vektor stavů popisující virtuálního leadra je dán váženým průměrem vektorů stavů všech ostatních agentů skupiny. Tím vzniká požadavek na existenci vazby (orientované hrany) z každého uzlu k leadrovi. Řízení agenta popisujícího roli virtuálního leadra je dáno požadovanou referenční trajektorií. Pro systémy s dynamikou prvního řádu (3.1) je tedy chování leadra posáno vztahem:

$$\dot{x}_0(t) = u_{ref}(t), \quad (3.8)$$

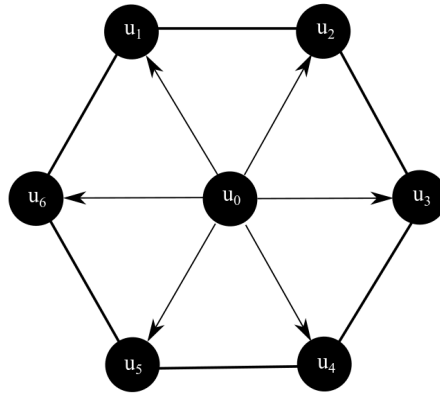
kde $u_{ref}(t)$ je časový průběh požadované trajektorie, dále značen u_{ref} .

Lokální protokol řízení osattních agentů, kteří jsou popsáni dynamikou (3.1), vypadá následovně:

$$u_i = \dot{x}_0 + (x_0 - x_i) + c \sum_{j=1}^N a_{ij}(x_j - x_i), \quad (3.9)$$

kde c je jediný návrhový parametr. Oproti příkladu 3.4 je předpoklad $k_p = 1$, $\forall i$ členu $(x_0 - x_i)$, což reprezentuje požadavek na existenci orientované hrany od virtuálního leadra ke všem vagentům. Jak bylo ukázáno v příkladu 3.4, systém dosáhne shody pokud graf G , který je reprezentován maticí Laplaciánu \mathbf{L} , obsahuje orientovanou hranu z leadra ke všem agentům. To je pro variantu virtuálního leadra vždy splněno, neboť existuje virtuální přímá orientovaná hrana od leadra ke každému agentovi.

Příklad 3.5. Mějme následující modifikaci grafu G_1 z obrázku 2.2, kde na pozici agenta u_0 uvažujme virtuálního leadra. Informační topologie této skupiny poté bude popsána následujícím grafem:

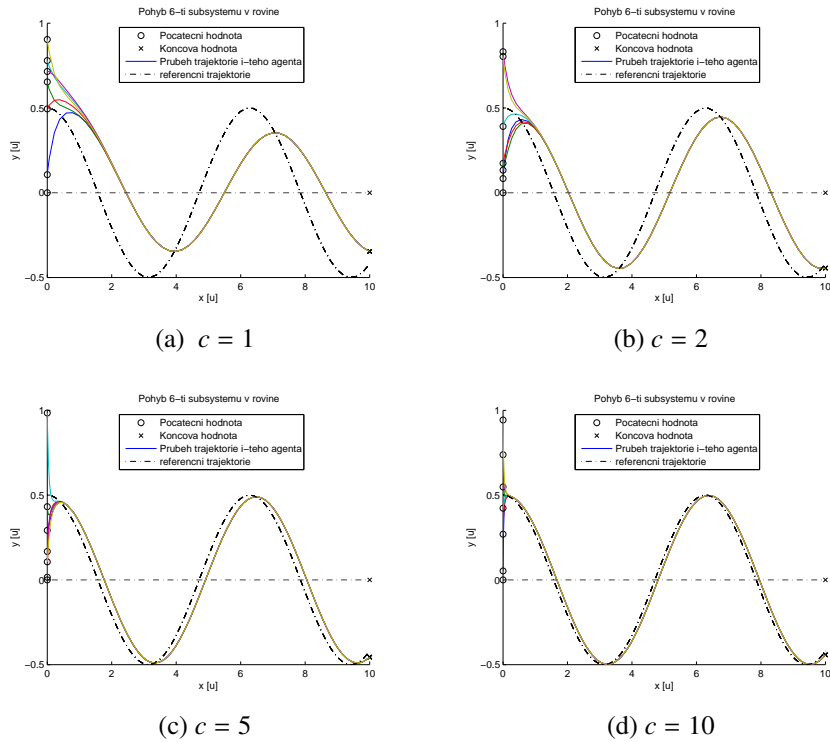


Obrázek 3.8: Graf G_3 reprezentující informační topologii systému, kde šipky reprezentují orientovanou hranu ve směru ukazatele.

Systém, jehož agenti $x_1 - x_6$ jsou popsáni dynamikou prvního řádu (3.1) a zákon řízení celého systému je popsán rovnicí (3.4), je popsán soustavou diferenciálních rovnic:

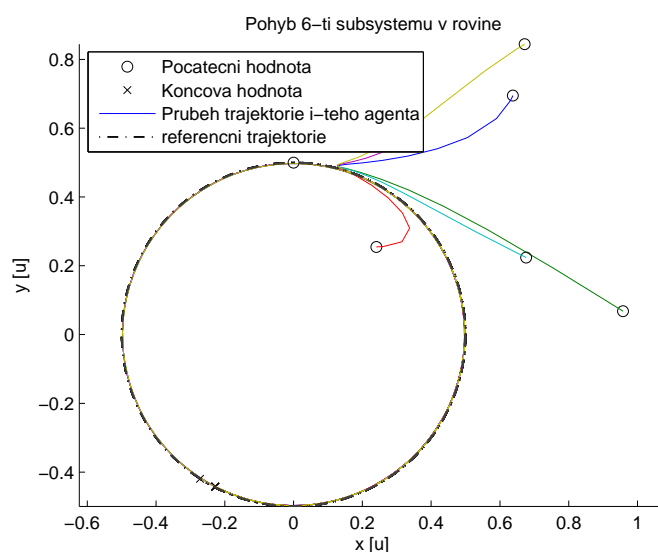
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_6 \\ \dot{x}_0 \end{bmatrix} = -c \underbrace{\begin{bmatrix} 3 & -1 & 0 & 0 & 0 & -1 & -1 \\ -1 & 3 & -1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 3 & -1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 3 & -1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & 0 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{L}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_6 \\ x_0 \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ u_{ref.} \end{bmatrix}. \quad (3.10)$$

Poslední sloupec matice \mathbf{L} , reprezentující vazby všech agentů k leadrovi, má identické prvky pro všechny agenty kromě samotného leadra. Tím je zaručeno, že všichni agenti budou následovat leadra se stejnou odchylkou. V porovnání s příkladem 3.4, kde vznikl rozpor mezi agenty, protože se k některým agentům dostávala informace o leadrovi zprostředkovaně. Simulace takto popsaného systému v 1D s množinou parametrů $c = \{1, 2, 5, 10\}$ vypadá následovně:



Obrázek 3.9: Průběh stavů systému, jehož agenti mají předpis řízení definovaný vztahem (3.9), s náhodnými počátečními hodnotami a pro různé hodnoty parametru c .

Simulace stejného systému (3.10) ve 2D prostoru, tedy rovině (x, y) , pro níž uvažujeme Kroneckerův součin dle (3.5), dopadne následovně:



Obrázek 3.10: Průběh stavů systému v rovině (x, y) , jehož agenti mají předpis řízení definovaný vztahem (3.9), s náhodnými počátečními hodnotami.

3.2.2 Řízení s externím leadrem

V případech, kdy není vhodné nebo možné aplikovat řízení s virtuálním leadrem, existuje možnost tzv. externího (jmenovaného) leadra. Tento přístup určí jednoho nebo množinu agentů za leadra, čímž tito agenti určují referenční trajektorii celé skupiny. Jmenování do pozice leadra je tedy chápáno jako získání externí informace o průběhu požadované trajektorie, která je často reprezentována pohybem přidaného agenta. Znalost této trajektorie je pak orientovaná hrana od přidaného agenta, který však není součástí původní skupiny. Podobně jako reprezentujeme provázanost sousedů grafu maticí Laplaciánu \mathbf{L} , jsou vazby na přidaného agenta popisovány pomocí matice \mathbf{G} .

Uvažujme systém, jehož agenti jsou popsáni dynamikou (3.1). Referenční trajektorie (časový průběh přidaného agenta) bude popsána vztahem:

$$\dot{x}_0(t) = u_{ref}(t)$$

Předpis řízení pro i -tého agenta vypadá následovně:

$$u_i = \dot{x}_0 + c \sum_{j=1}^N a_{ij}(x_j - x_i) + g_i(x_0 - x_i),$$

Pro množinu agentů, která má vazbu na přidaného leadra platí $g_i = 1$, jinak $g_i = 0$, tj. $g_i \in \{0, 1\}, \forall i$. Matice \mathbf{G} je diagonální maticí, která je tvořena diagonálními prvky g_i . Chování výsledného systému lze zapsat jako:

$$\begin{aligned}\dot{x} &= -c\mathbf{L}x - c\mathbf{G}x + c\mathbf{G}x_0, \\ \dot{x} &= -c(\mathbf{L} + \mathbf{G})x + c\mathbf{G}x_0,\end{aligned}$$

Pokud bychom uvažovali odchylku systému, podobně jako v příkladu 3.4, získáme:

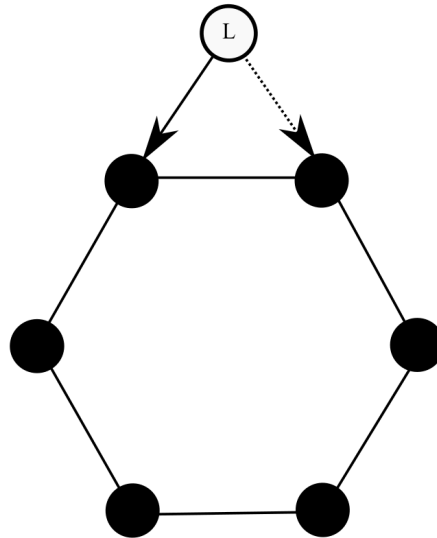
$$\begin{aligned}\tilde{x}_i &= x_i - x_0, \\ \dot{\tilde{x}}_i &= \dot{x}_i - \dot{x}_0, \\ \dot{\tilde{x}}_i &= \dot{x}_0 + c \sum_{j=1}^N a_{ij}(x_j - x_i) + g_i(x_0 - x_i) - \dot{x}_0, \\ \dot{\tilde{x}}_i &= c \sum_{j=1}^N a_{ij}(\tilde{x}_j - \tilde{x}_i) - g_i\tilde{x}_i.\end{aligned}$$

Chybu celého systému lze zapsat následovně:

$$\dot{\tilde{x}} = -c(\mathbf{L} + \mathbf{G})\tilde{x}.$$

Pro zaručení consensusu na stavu leadra, resp. referenční hodnotě, musí být zaručena stabilita matice $-c(\mathbf{L} + \mathbf{G})$. Tento předpoklad je splněn pro $c > 0$ a pozitivní semidefinitnost matice $(\mathbf{L} + \mathbf{G})$.

Příklad 3.6. Uvažujme systém, jehož agenti jsou popsáni dynamikou (3.1) a jeho grafová topologie je reprezentována následujícím grafem.

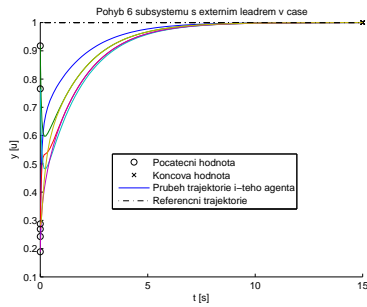


Obrázek 3.11: Graf G_1 popisující topologii systému s externím leadrem. Tečkovaná spojnice leadra a agenta reprezentuje možnou aditivní vazbu.

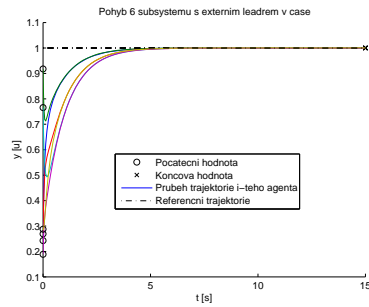
Maticе Laplaciánu \mathbf{L} a matice \mathbf{G} popisující vazbu agentů na leadra vypadají následovně:

$$\mathbf{L} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

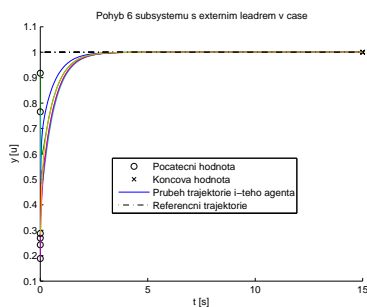
Zvolíme-li referenční signál $u_{ref} = 0$ při náhodných počátečních podmínkách stavů, systém by měl dosáhnout konsensu ve stavu prvního agenta. Ověříme tento předpoklad simulačně pro různé hodnoty parametru c , přičemž porovnáme i způsob zapojení externího leadra, kde porovnáme dosavadní zapojení a zapojení s aditivní vazbou z grafu G_1 .



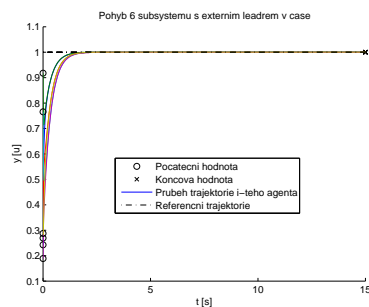
(a) Základní zapojení leadra s parametrem $c = 5$.



(b) Aditivní zapojení leadra s parametrem $c = 5$.



(c) Základní zapojení leadra s parametrem $c = 15$.

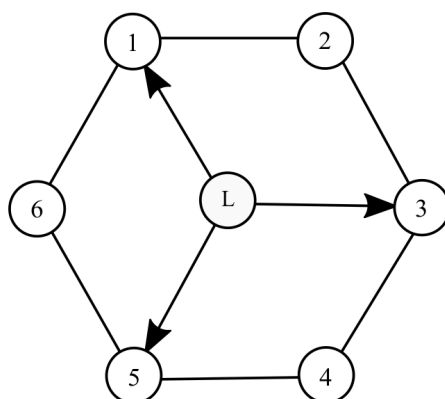


(d) Aditivní zapojení leadra s parametrem $c = 15$.

Obrázek 3.12: Průběh poloh agentů systému v časové závislosti s různými hodnotami návrhového parametru c . Obrázky (a) a (c) reprezentují časový průběh poloh systému, jehož topologie je popsána grafem G_1 bez aditivní hrany. Varianty (b) a (d) jsou průběhy pro systém s topologií popsanou grafem G_1 včetně aditivní hrany.

Z průběhů simulací poloh systémů je patrné, že aditivní zapojení má pozitivní vliv na rychlost konvergence, neboť informace o požadované trajektorii je prezentována přímo více agentům.

Příklad 3.7. Uvažujme systém v rovině (x, y) , jehož agenti jsou popsáni dynamikou (3.1). Požadavky na systém jsou, aby skupina agentů utvořila zadanou formaci a následně opisovala referenční trajektorii. Systém tedy bude obsahovat leadra, zvolíme implementaci externího leadra, u kterého uvidí polovina zbylých agentů na jeho stav. Topologie systému bude popsána grafem G :



Obrázek 3.13: Graf G .

Lokální zákon řízení bude dán vztahem:

$$u_i = \dot{x}_0 + c \sum_{j=1}^N a_{ij}((x_j - \Delta_j) - (x_i - \Delta_i)) + g_i(x_0 - x_i),$$

kde x_0 představuje stav leadra, a_{ij} jsou váhové parametry tvořící topologii systému a parametr c je váhový prvek určující rychlost konvergence.

Uvažujme chybu i -tého agenta, která je určena následovně:

$$\tilde{x}_i = x_i - \Delta_i - x_0,$$

$$\dot{\tilde{x}}_i = \dot{x}_i - \dot{x}_0,$$

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

Maticově lze pak celý systém vyjádřit jako:

$$\dot{\tilde{x}} = -c(\mathbf{L} + \mathbf{G})\tilde{x}.$$

Požadovaný průběh stavů x původního systému získáme zpětným přepočtem:

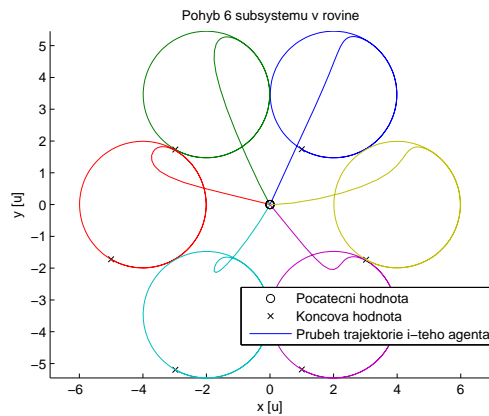
$$x = \tilde{x} + \Delta_i + x_0.$$

Parametry, pro které byl model simulován, jsou:

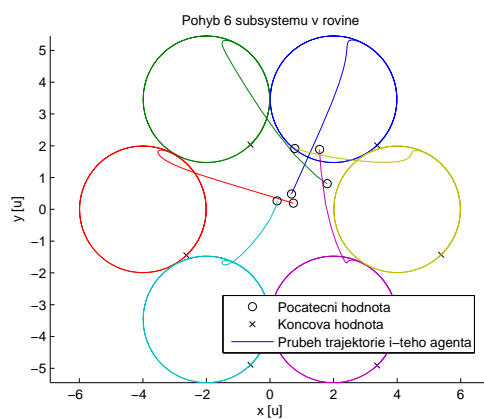
$$\mathbf{L} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad c = 20,$$

$$\mathbf{u}_{ref} = \begin{bmatrix} u_{refx} \\ u_{refy} \end{bmatrix} = \begin{bmatrix} 2\sin(t) \\ 2\cos(t) \end{bmatrix}, \quad \bar{\Delta} = \begin{bmatrix} \bar{\Delta}_x \\ \bar{\Delta}_y \end{bmatrix} = \begin{bmatrix} [1 \ -1 \ -2 \ -112] \\ [1.731.730 \ -1.73 \ -1.730] \end{bmatrix}.$$

Průběh trajektorií poloh jednotlivých agentů pro takto popsaný systém vypadá následovně:



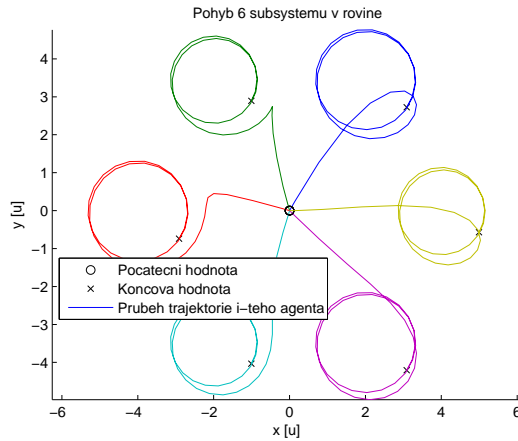
(a) Průběh dynamiky systému s nulovými počátečními hodnotami.



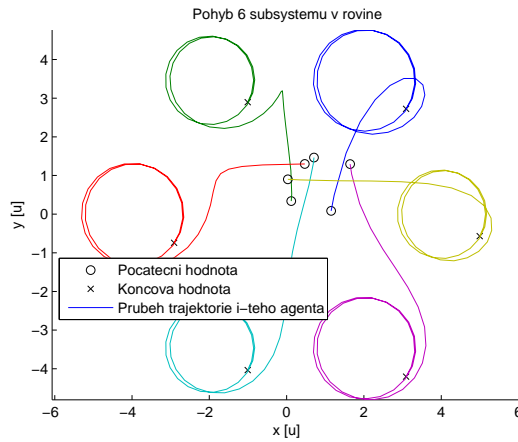
(b) Průběh dynamiky systému s náhodnými počátečními hodnotami.

Obrázek 3.14: Průběh dynamiky systému s volbou parametru $c = 20$.

Parametr c byl volen $c = 20$ pro redukci odchylky agentů od požadované trajektorie. Podobně jako v příkladech 3.4 a 3.5 dochází při nižších hodnotách parametru c k trvalému biasu všech agentů. Tento efekt je demonstrován v následující simulaci, kde je jediným rozdílem oproti předchozí simulaci hodnota parametru $c = 1$.



(a) Průběh dynamiky systému s nulovými počátečními hodnotami.



(b) Průběh dynamiky systému s náhodnými počátečními hodnotami.

Obrázek 3.15: Průběh dynamiky systému s volbou parametru $c = 1$.

3.3 Lineární systémy vyšších řádů

Postupy uvedené v předchozích sekcích můžeme s jistými úpravami zobecnit pro lineární systémy vyšších řádů. U řízení systémů vyšších řádů je cílem zajistit shodu na všech stavech systému. Například pro druhý řád diferenciální rovnice (2.2) je zapotřebí dosáhnout consensusu na poloze a rychlosti, pro systémy vyšších řádů bychom postupovali obdobně. Uvažujme nyní systém popsany diferenciálními rovnicemi ve tvaru:

$$\begin{aligned} \dot{x}_i &= v_i, \\ \dot{v}_i &= u_i, \\ \dot{z}_i &= \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\mathbf{A}_i} z_i + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{B}_i} u_i, \quad z_i = \begin{bmatrix} x_i \\ v_i \end{bmatrix}, \end{aligned} \quad (3.11)$$

Při implementaci consensusu protokolu je zapotřebí zajistit stabilitu výsledné uzavřené smyčky. Distribuovaný zákon řízení, pro *i-tého* agenta takto popsaného systému vypadá následovně:

$$\begin{aligned} u_i &= c \sum_{j=1}^N a_{ij}(x_j - x_i) + c\gamma \sum_{j=1}^N a_{ij}(v_j - v_i), \\ u_i &= c[1 \quad \gamma] \sum_{j=1}^N a_{ij} \begin{bmatrix} (x_j - x_i) \\ (v_j - v_i) \end{bmatrix}, \\ u_i &= c\mathbf{K} \sum_{j=1}^N a_{ij}(z_j - z_i), \end{aligned} \quad (3.12)$$

kde $c > 0$ reprezentuje váhu zpětné vazby daného uzlu a $\mathbf{K} = [1 \quad \gamma]$ reprezentuje zpětnovazební matici. Chování uzavřené smyčky celého systému lze vyjádřit jako:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \vdots \\ \dot{z}_N \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}_N \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} - \mathbf{L} \otimes \begin{bmatrix} \mathbf{B}_1 & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & \vdots \\ \vdots & & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{B}_N \end{bmatrix} \begin{bmatrix} c\mathbf{K} & \mathbf{0} \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & c\mathbf{K} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}. \quad (3.13)$$

Pokud jsou matice $\mathbf{A}_1, \dots, \mathbf{A}_N$ a $\mathbf{B}_1, \dots, \mathbf{B}_N$ stejné, lze chování systému popsat následujícím způsobem:

$$\begin{aligned}\dot{z} &= [(\mathbf{I}_N \otimes \mathbf{A}) - c(\mathbf{L} \otimes \mathbf{I}_D) \otimes (\mathbf{I}_N \otimes \mathbf{B})(\mathbf{I}_N \otimes \mathbf{K})]z, \\ \dot{z} &= [(\mathbf{I}_N \otimes \mathbf{A}) - c\mathbf{L} \otimes \mathbf{BK}]z = \mathbf{A}_c z,\end{aligned}\quad (3.14)$$

kde \mathbf{I}_N označuje jednotkovou diagonální matici velikosti $N \times N$ (počet agentů), \mathbf{I}_D označuje jednotkovou diagonální matici velikosti $D \times D$ (D reprezentuje dimenzi stavu jednoho agenta) a \otimes reprezentuje Kroneckerův součin [1]. Pokud je graf popisující topologii systému souvislý, je jeho matice dynamiky uzavřeného systému podobná blokově diagonální matici ve tvaru [4]:

$$\mathbf{A} \approx \mathbf{A}_c = \begin{bmatrix} \mathbf{A} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A} - c\lambda_2 \mathbf{BK} & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A} - c\lambda_N \mathbf{BK} \end{bmatrix}. \quad (3.15)$$

Z tohoto zápisu plyne, že o stabilitě celého systému rozhodují matice $\{\mathbf{A}, \mathbf{A} - c\lambda_2 \mathbf{BK}, \dots, \mathbf{A} - c\lambda_N \mathbf{BK}\}$. Je důležité si uvědomit, že stabilizující zpětná vazba \mathbf{K} musí individuálně stabilizovat všechny subsystémy, ale už sama nerozhoduje o stabilitě celého systému. Ta závisí na stabilitě všech dílčích agentů a topologii propojení subsystémů. Cílem je tedy najít hodnoty koeficientů c , \mathbf{K} takové, aby byla splněna následující lemma.

Lemma 3.1. [4] *Necht' λ_i , $i = 1, 2, \dots, N$ jsou vlastní čísla matice Laplaciánu \mathbf{L} . Systém (3.14) je asymptoticky stabilní právě tehdy, když jsou všechny matice*

$$\mathbf{A} - c\lambda_i \mathbf{BK}, \quad \forall i$$

asymptoticky stabilní.

Důkaz Systém (3.14) je asymptoticky stabilní právě tehdy, leží-li všechna vlastní čísla matice \mathbf{A}_c (3.15) v levé polorovině komplexní roviny. Pro matici \mathbf{L} existuje Jordanův rozklad pomocí regulární matice $\mathbf{S} \in \mathbb{R}^{N \times N}$ takový, že:

$$\mathbf{S}^{-1}(\mathbf{L})\mathbf{S} = \mathbf{J} = \begin{bmatrix} \mathbf{J}_{n_1}(\bar{\lambda}_1) & & & \\ & \mathbf{J}_{n_2}(\bar{\lambda}_2) & & \\ & & \ddots & \\ & & & \mathbf{J}_{n_k}(\bar{\lambda}_k) \end{bmatrix},$$

kde $\mathbf{J}_{n_1}, \mathbf{J}_{n_2}, \dots, \mathbf{J}_{n_k}$ jsou Jordanovy bloky velikostí n_1, n_2, \dots, n_k a $n_1 + n_2 + \dots + n_k = N$. Uvažujme následující aplikaci Jordanova rozkladu:

$$\begin{aligned} (\mathbf{S} \otimes \mathbf{I}_n)^{-1} \mathbf{A}_c (\mathbf{S} \otimes \mathbf{I}_n) &= (\mathbf{S} \otimes \mathbf{I}_n)^{-1} (\mathbf{I}_N \otimes \mathbf{A} - c\mathbf{L} \otimes \mathbf{BK}) (\mathbf{S} \otimes \mathbf{I}_n) = \\ &= (\mathbf{S} \otimes \mathbf{I}_n)^{-1} (\mathbf{I}_N \otimes \mathbf{A}) (\mathbf{S} \otimes \mathbf{I}_n) - (\mathbf{S} \otimes \mathbf{I}_n)^{-1} (c\mathbf{L} \otimes \mathbf{BK}) (\mathbf{S} \otimes \mathbf{I}_n) = \\ (\mathbf{S} \otimes \mathbf{I}_n)^{-1} (\mathbf{S} \otimes \mathbf{A}) - (\mathbf{S} \otimes \mathbf{I}_n)^{-1} (c\mathbf{L} \mathbf{S} \otimes \mathbf{BK}) &= \mathbf{I}_n \otimes \mathbf{A} - (\mathbf{S}^{-1} c\mathbf{L} \mathbf{S}) \otimes c\mathbf{BK} = \mathbf{I}_n \otimes \mathbf{A} - c\mathbf{J} \otimes \mathbf{BK}. \end{aligned}$$

Označme prvky diagonály matice \mathbf{J} jako $\{\lambda_1, \lambda_2, \dots, \lambda_N\}$, potom platí:

$$[\lambda_1, \lambda_2, \dots, \lambda_N] = [\bar{\lambda}_1, \dots, \bar{\lambda}_1, \dots, \bar{\lambda}_k, \dots, \bar{\lambda}_k].$$

Tedy platí, že množina vlastních čísel diagonálních bloků \mathbf{J}_{n_k} je totožná s vlastními čísly matice \mathbf{J} . Z toho plyne, že matice \mathbf{A}_c je stabilní pouze pokud jsou všechny submatice $\mathbf{A} - c\lambda_i \mathbf{BK}$, ($i = 1, 2, \dots, N$) stabilní.

Pokud je tato podmínka splněna, systém dosáhne shody pro:

$$\bar{x} = \frac{1}{N} \sum_{j=1}^N p_j x_j(0) + \frac{1}{N} \sum_{j=1}^N t p_j v_j(0), \quad (3.16)$$

$$\bar{v} = \frac{1}{N} \sum_{j=1}^N p_j v_j(0), \quad (3.17)$$

kde $x(0)$ a $v(0)$ jsou počáteční hodnoty polohy a rychlosti, $w_1 = [p_1 \dots p_N]^T$ je normalizovaný levý vektor matice Laplaciánu \mathbf{L} vlastního čísla $\lambda_1 = 0$. Výsledná shoda tedy závisí na počáteční poloze a rychlosti všech agentů. To způsobí, že pokud bude mít jeden agent nenulovou počáteční rychlost, rozpohybuje celou skupinu ve směru jeho pohybu.

Volba zpětnovazební matice \mathbf{K} a váhy c závisí na konkrétní aplikaci a požadovaných regulačních vlastnostech. V přípravné kapitole byly uvedeny dvě metody návrhu, přiřazení Jordanovy formy a lineární kvadratický regulátor (dále označován LQR). Při aplikaci těchto návrhových metod na multi-agentní systémy vznikají omezující požadavky na váhové parametry c a $c\gamma$. Pokud by subsystémy popsány maticí dynamiky \mathbf{A}_i byly stabilní, consensus bude dosaženo i při volbě váhového parametru $c = 0$, [4]. V případě nestability dílčích subsystémů musí být k zachování výsledné stability systému voleny váhové parametry $c > 0$, $c\gamma > 0$.

Příklad 3.8. Uvažujme systém, jehož agenti jsou popsáni dynamickým popisem (3.11) a topologie systému je reprezentována grafem G_1 z obrázku 3.1. Lokální protokol řízení každého agenta je dán vztahem (3.12). Výsledné chování celého

systemu je popsáno rovnicí (3.13). Zpětnovazební matice \mathbf{K} určíme jak přiřazením Jordanovy formy tak pomocí LQR, zatímco váhový parametr $c = 1$ zůstane stejný pro všechny případy. Matice \mathbf{K}_{JF} byla určena přiřazením Jordanovy formy \mathbf{L}_s ve tvaru:

$$\mathbf{L}_s = \begin{bmatrix} -3 & 1 & 0 & 0 \\ 0 & -3 & 0 & 0 \\ 0 & 0 & -3 & 1 \\ 0 & 0 & 0 & -3 \end{bmatrix}.$$

Výsledná matice \mathbf{K}_{JF} byla vypočtena dle vztahu (2.4) následovně:

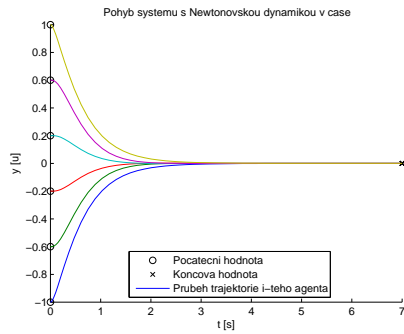
$$\mathbf{K}_{JF} = \begin{bmatrix} -9 & -6 \end{bmatrix}.$$

V případě návrhu zpětnovazební matice \mathbf{K}_{LQR} pomocí LQR, bylo řešeno kritérium (2.6) pro následující návrhové parametry \mathbf{Q} a \mathbf{R} :

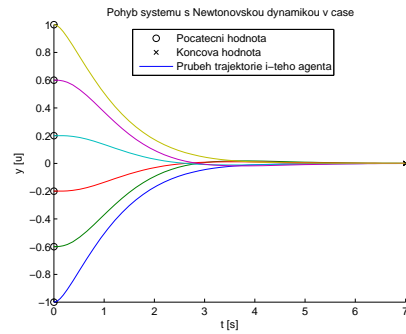
$$\mathbf{Q} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Výsledkem byla zpětnovazební matice:

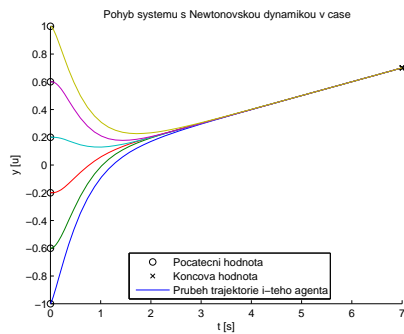
$$\mathbf{K}_{LQR} = \begin{bmatrix} -\sqrt{3} & -2.1128 \end{bmatrix}.$$



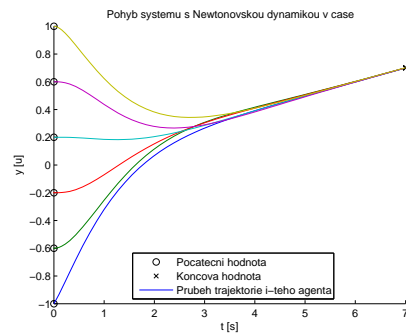
(a) Chování systému se zpětnovazební maticí \mathbf{K}_{JF} pro nulové počáteční rychlosti.



(b) Chování systému se zpětnovazební maticí \mathbf{K}_{LQR} pro nulové počáteční rychlosti.



(c) Chování systému se zpětnovazební maticí \mathbf{K}_{JF} pro nenulovou počáteční rychlost jednoho agenta.



(d) Chování systému se zpětnovazební maticí \mathbf{K}_{LQR} pro nenulovou počáteční rychlost jednoho agenta.

Obrázek 3.16: Průběh trajektorií dílčích subsystémů se stavovou zpětnou vazbou. Agenti jsou rovnoměrně rozmístěni mezi body $[-1, 1]$. Pro případy (a) a (b) mají všichni nulovou počáteční rychlost. V simulacích (c) a (d) má pouze jeden agent nenulovou počáteční rychlost $v_y = 0.6 \text{ u/s}$ (units/seconds neboli jednotek za vteřinu).

Podle očekávání, volba zpětnovazební matice \mathbf{K} ovlivňuje rychlost konvergence. Pro obě uvažované možnosti \mathbf{K}_{JF} a \mathbf{K}_{LQR} je splněna podmínka stability 3.1, tudíž je výsledný systém stabilní. V případech (c) a (d) je dosaženo konsensu pro nenulovou rychlost všech agentů, přestože měl nenulovou počáteční rychlost pouze jeden agent.

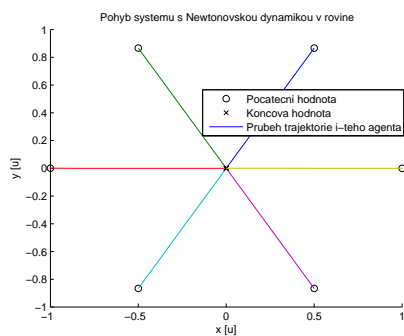
Příklad 3.9. Uvažujme systém z předchozího příkladu v rovině (x, y) . Stavový popis (3.11) přejde do tvaru:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

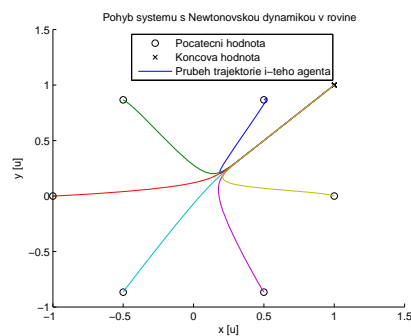
Uvažujme zpětnou vazbu \mathbf{K}_{JF} , určenou stejným postupem jako v předchozím příkladě. Pro takto popsany systém bude mít tvar:

$$\mathbf{K}_{JF} = \begin{bmatrix} -9 & 0 & -6 & 0 \\ 0 & -9 & 0 & -6 \end{bmatrix}.$$

Simulace pro výše zavedený systém v rovině (x, y) vypadá následovně:



(a) Simulace systému s počátečním rozmístěním agentů po kružnici, kde mají všichni agenti nulovou počáteční rychlost.



(b) Simulace systému s počátečním rozmístěním agentů po kružnici, kde má jeden agent nenulovou počáteční rychlost.

Obrázek 3.17: Průběh simulace systému v rovině s odlišnými počátečními podmínkami. V případě (b) má pouze jeden agent nenulovou počáteční rychlost $[v_x, v_y] = [0.6, 0.6] \text{ u/s}$.

3.3.1 Consensus protokol a řízení do požadované formace

V návaznosti na kapitulu 3.1.2 lze řídit do požadované formace i systémy, jejichž agenti jsou popsáni dynamikou druhého řádu (obdobně platí i pro dynamiku vyšších řádů). Uvažujme systém, jehož agenti jsou popsáni dynamikou (3.11). Zvolíme rozmístění agentů do formace pomocí vektoru $\bar{\Delta}$ a zavedeme lokální protokol řízení ve tvaru:

$$u_i = c \sum_{j=1}^N a_{ij} [(x_j - \Delta_j) - (x_i - \Delta_i)] + c\gamma \sum_{j=1}^N a_{ij} (v_j - v_i).$$

Požadavky na chování agentů v systému lze vyjádřit následovně:

$$\begin{aligned} x_i &\rightarrow \Delta_i, & \tilde{x}_i &= x_i - \Delta_i, \\ & & \Rightarrow & \\ v_i &\rightarrow v_i, & \tilde{v}_i &= v_i. \end{aligned}$$

Derivací odchylek systému získáme:

$$\begin{aligned} \dot{\tilde{x}}_i &= \dot{x}_i = v_i = \tilde{v}_i, \\ \dot{\tilde{v}}_i &= u_i = c \sum_{j=1}^N a_{ij}[(x_j - \Delta_j) - (x_i - \Delta_i)] + c\gamma \sum_{j=1}^N a_{ij}(v_j - v_i), \\ \dot{\tilde{v}}_i &= c \sum_{j=1}^N a_{ij}(\tilde{x}_j - \tilde{x}_i) + c\gamma \sum_{j=1}^N a_{ij}(\tilde{v}_j - \tilde{v}_i). \end{aligned}$$

Pro celý systém lze psát:

$$\dot{\tilde{z}} = \begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{v}}_1 \\ \vdots \\ \dot{\tilde{x}}_n \\ \dot{\tilde{v}}_n \end{bmatrix} = \left(\mathbf{I}_n \otimes \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\mathbf{A}_i} \right) \begin{bmatrix} \tilde{x}_1 \\ \tilde{v}_1 \\ \vdots \\ \tilde{x}_n \\ \tilde{v}_n \end{bmatrix} - c\mathbf{L} \otimes \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{B}_i} \underbrace{\begin{bmatrix} 1 & \gamma \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} \tilde{x}_1 \\ \tilde{v}_1 \\ \vdots \\ \tilde{x}_n \\ \tilde{v}_n \end{bmatrix},$$

$$\dot{\tilde{z}} = (\mathbf{I}_n \otimes \mathbf{A} - c\mathbf{L} \otimes \mathbf{BK})\tilde{z}. \quad (3.18)$$

Shody je dosaženo v zadané formaci, pokud je tento systém (3.18) stabilní, viz lemma 3.1.

3.3.2 Consensus protokol a řízení do požadované formace s virtuálním leadrem

V kapitole 3.2.1 bylo ukázáno, jak doplnit consensus protokol řízení o požadavek na formaci a následně jak zavést do skupiny virtuálního leadra. Dalším krokem je tyto požadavky zkombinovat. V této kapitole navážeme na tyto znalosti a budeme se věnovat řízení agentů do požadované formace s následným (resp. současným) sledováním leadra. Uvažujme dynamiku *i-tého* agenta danou vztahem:

$$\begin{aligned} \dot{x}_i &= \mathbf{A}x_i + \mathbf{B}u_i, \\ y_i &= \mathbf{C}x_i. \end{aligned}$$

kde $i = 1, \dots, N$, přičemž N označuje celkový počet agentů. Dynamika leadra je dána vztahem:

$$\begin{aligned}\dot{x}_0 &= \mathbf{A}x_0, \\ y_0 &= \mathbf{C}x_0,\end{aligned}$$

který je opět autonomním systémem, neboť jeho chování není ovlivněno žádným sledujícím agentem. Přidáním požadavků na formaci a sledování virtuálního leadra, přejde lokální protokol řízení (3.12) na tvar:

$$\begin{aligned}u_i &= \dot{v}_0 + \gamma k_v (v_0 - v_i) + k_p (x_0 + \Delta_i - x_i) + \\ &+ c \sum_{j=1}^N a_{ij} (x_j - \Delta_j) - (x_i - \Delta_i) + c\gamma \sum_{j=1}^N a_{ij} (v_j - v_i),\end{aligned}\quad (3.19)$$

kde k_v je váha vazby *i-tého* agenta na rychlost leadra, k_p je váha vazby *i-tého* agenta na polohu leadra a Δ představuje požadovanou odchylku daného agenta. Stav jednotlivých uzlů poté při zachování stability konvergují k následujícím hodnotám:

$$\begin{aligned}x_i &\rightarrow x_0 + \Delta_i, \\ v_i &\rightarrow v_0.\end{aligned}$$

Pokud budeme uvažovat dynamiku *i-tého* agenta \mathbf{A} , \mathbf{B} následovně:

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} \mathbf{0} & \mathbf{I}_n \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \\ \mathbf{B} &= \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_n \end{bmatrix},\end{aligned}$$

kde \mathbf{I}_n představuje jednotkovou matici potřebné dimenze a $\mathbf{0}$ nulovou matici odpovídajících rozměrů. Odchylku systému (3.14) lze vyjádřit následovně:

$$\begin{aligned}\tilde{x}_i &= x_i - \Delta_i - x_0, \\ \tilde{v}_i &= v_i - v_0, \\ \dot{\tilde{z}}_i &= \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I}_n \\ -k_p \mathbf{I}_n & -\gamma k_v \mathbf{I}_n \end{bmatrix}}_{\tilde{\mathbf{A}}_i} \tilde{z}_i + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{I}_n \end{bmatrix}}_{\tilde{\mathbf{B}}_i} \sum_{j=1}^N c a_{ij} [(x_j - \Delta_j) - (x_i - \Delta_i) + \gamma (v_j - v_i)].\end{aligned}\quad (3.20)$$

Pro celý systém se všemi agenty dohromady dostaneme:

$$\begin{bmatrix} \dot{\tilde{z}}_1 \\ \dot{\tilde{z}}_2 \\ \vdots \\ \dot{\tilde{z}}_N \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{A}}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{A}}_2 & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \tilde{\mathbf{A}}_N \end{bmatrix} \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \vdots \\ \tilde{z}_N \end{bmatrix} - \mathbf{L} \otimes \begin{bmatrix} \tilde{\mathbf{B}}_1 & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & \vdots \\ \vdots & & \mathbf{0} \\ \mathbf{0} & \cdots & \tilde{\mathbf{B}}_N \end{bmatrix} \begin{bmatrix} c\mathbf{K} & \mathbf{0} \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & c\mathbf{K} \end{bmatrix} \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \vdots \\ \tilde{z}_N \end{bmatrix},$$

$$\dot{\tilde{z}} = (\mathbf{I}_n \otimes \tilde{\mathbf{A}} - c\mathbf{L} \otimes \tilde{\mathbf{B}}\mathbf{K})\tilde{z} = \tilde{\mathbf{A}}_c\tilde{z}.$$

O stabilitě systému (3.14) lze opět rozhodnout na základě lemmy 3.1, ve které musí být splněné následující předpoklady:

$$c\gamma > 0, \quad c > 0, \quad k_p > 0, \quad k_v > 0.$$

Z předpokladů stability vyplývá, že všichni agenti musí znát polohu a rychlost leadra.

Příklad 3.10. Uvažujme systém se 6-ti agenty, kteří se pohybují v prostoru. Dynamika těchto agentů je dána předpisem:

$$\dot{p}_i = \begin{bmatrix} \mathbf{0} & \mathbf{I}_3 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} p_i + \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_3 \end{bmatrix} u_i, \quad p_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ v_{xi} \\ v_{yi} \\ v_{zi} \end{bmatrix}, \quad (3.21)$$

kde \mathbf{I}_3 představuje jednotkovou matici o rozměrech 3×3 a $\mathbf{0}$ udává nulovou matici stejných rozměrů.

Cílem úlohy je najít předpis distribuovaného řízení, které zaručí stabilitu systému obsahujícího 6 agentů v předepsané formaci, která opisuje zadanou trajektorii. Průběh trajektorie je definován virtuálním leadrem, jehož dynamika je popsána rovnicí:

$$\dot{p}_0 = \begin{bmatrix} \mathbf{0} & \mathbf{I}_3 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} p_0 + \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_3 \end{bmatrix} u_{ref}, \quad p_0 = \begin{bmatrix} x_0 \\ \vdots \\ v_0 \\ \vdots \end{bmatrix}.$$

Lokálního předpis řízení pro *i-tého* agenta je určen modifikací vztahu (3.19). Hlavní modifikací je rozšíření předpisu řízení na potřebnou dimenzi, tedy:

$$u_{xi} = \dot{v}_{x0} + \gamma k_v (v_{x0} - v_{xi}) + k_p (x_0 + \Delta_{xi} - x_i) + c \sum_{j=1}^N a_{ij} (x_j - \Delta_{xj}) - (x_i - \Delta_{xi}) + c\gamma \sum_{j=1}^N a_{ij} (v_{xj} - v_{xi}),$$

$$u_{yi} = \dot{v}_{y0} + \gamma k_v (v_{y0} - v_{yi}) + k_p (y_0 + \Delta_{yi} - y_i) + c \sum_{j=1}^N a_{ij} (y_j - \Delta_{yj}) - (y_i - \Delta_{yi}) + c\gamma \sum_{j=1}^N a_{ij} (v_{yj} - v_{yi}),$$

$$u_{zi} = \dot{v}_{z0} + \gamma k_v (v_{z0} - v_{zi}) + k_p (z_0 + \Delta_{zi} - z_i) + c \sum_{j=1}^N a_{ij} (z_j - \Delta_{zj}) - (z_i - \Delta_{zi}) + c\gamma \sum_{j=1}^N a_{ij} (v_{zj} - v_{zi}),$$

Mějme chybu *i-tého* agenta danou vztahem $\tilde{p}_i = p_i - p_0 - \Delta_i$, potom je dynamika chyby celého systému dána předpisem:

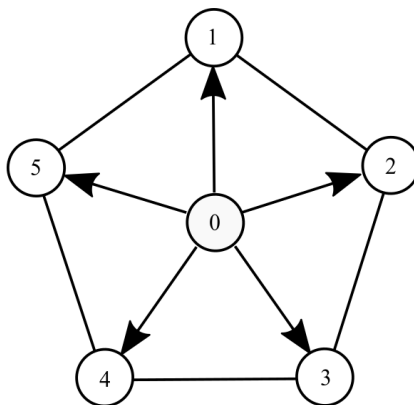
$$\dot{\tilde{p}} = (\mathbf{I}_6 \otimes \tilde{\mathbf{A}} - c\mathbf{L} \otimes \tilde{\mathbf{B}}\mathbf{K})\tilde{p},$$

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_3 \\ -k_p \mathbf{I}_3 & -\gamma k_v \mathbf{I}_3 \end{bmatrix}, \quad \tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_3 \end{bmatrix},$$

kde $\tilde{p} = [\tilde{p}_1, \dots, \tilde{p}_{12}]^T$. Matice \mathbf{L} je určena topologií systému a matice \mathbf{K} je navržená zpětnovazební matice stabilizující subsystém (3.23):

$$\dot{p}_i = (\mathbf{A} + \mathbf{B}\mathbf{K})p_i.$$

Uvažujme topologii systému danou následujícím grafem:



Obrázek 3.18: Graf G reprezentující informační provázanost systému. Na pozici agenta 0 uvažujeme virtuálního leadra.

Matice Laplaciánu \mathbf{L} reprezentující tuto topologii vypadá následovně:

$$\mathbf{L} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & -1 & 0 & 0 & -1 & -1 \\ -1 & 3 & -1 & 0 & 0 & -1 \\ 0 & -1 & 3 & -1 & 0 & -1 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & 0 & 0 & -1 & 3 & -1 \end{bmatrix}.$$

Navrhňeme nyní zpětnovazební matici \mathbf{K} pomocí návrhové metody LQR popsané v kapitole 2.3.1. Hodnoty váhových matic \mathbf{Q} a \mathbf{R} zvolme následovně:

$$\mathbf{Q} = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

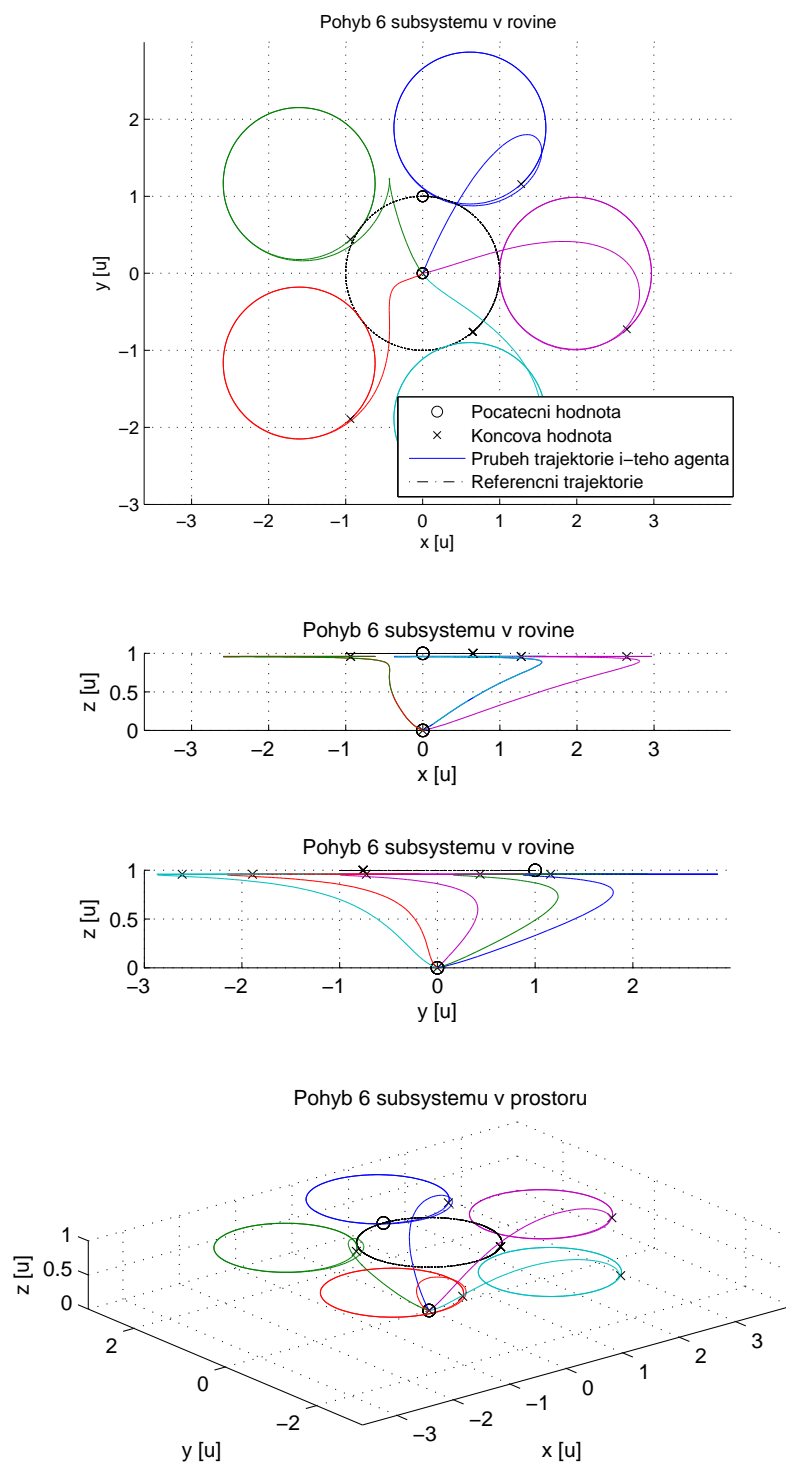
Zpětnovazební matici \mathbf{K} získáme řešením rovnice (2.6) ve tvaru:

$$\mathbf{K} = \begin{bmatrix} -2.3166 & 0 & 0 & -1.5755 & 0 & 0 \\ 0 & -2.3166 & 0 & 0 & -1.5755 & 0 \\ 0 & 0 & -2.3166 & 0 & 0 & -1.5755 \end{bmatrix}.$$

Zvolme referenční trajektorii leadra jako pohyb po kružnici ve zvolené výškové hladině, tj.:

$$\mathbf{u}_{ref} = \begin{bmatrix} \sin(t) \\ \cos(t) \\ 1 \end{bmatrix}.$$

Požadované uspořádání agentů zvolme podobně jako v grafu G na obrázku 3.18, tedy rovnoměrné rozmístění po kružnici s leadrem ve středu, a počáteční podmínky nulové. Simulace takto uvažovaného systému vypadá následovně:



Obrázek 3.19: Průběh trajektorií systému tvořeného 6-ti agenty, z nichž jeden představuje pouze virtuálního leadra.

3.3.3 Consensus protokol a řízení do požadované formace s externím leadrem

V návaznosti na kapitolu 3.2.2 lze princip externího leadra aplikovat i na systémy s dynamikou druhého řádu (či obecně vyšších řádů). Implementace principu externího leadra probíhá opět pomocí přičítání matice \mathbf{G} k matici Laplaciánu \mathbf{L} . Uvažujme systém, jehož agenti jsou popsáni dynamikou (3.11) a požadované rozmístění agentů ve formaci je dáno vektorem $\bar{\Delta} = [\Delta_1, \dots, \Delta_N]^T$. Dynamika leadra je popsána vztahem:

$$\begin{aligned}\dot{x}_0 &= \mathbf{A}x_0, \\ y_0 &= \mathbf{C}x_0,\end{aligned}$$

Zavedeme řízení ve tvaru:

$$u_i = \dot{v}_0 + c \sum_{j=1}^N a_{ij}[(x_j - \Delta_j) - (x_i - \Delta_i)] + g_i(x_0 - x_i + \Delta_i).$$

Cílem je zajistit konvergenci *i-tého* agenta $x_i \rightarrow x_0 + \Delta_i$, $v_i \rightarrow v_0$. Úpravou odchylky získáme:

$$\begin{aligned}\tilde{v}_i &= v_i - x_0, \\ \tilde{x}_i &= x_i - \Delta_i - x_0, \\ \dot{\tilde{x}}_i &= \dot{x}_i - \dot{x}_0, \\ \dot{\tilde{x}}_i &= c \sum_{j=1}^N [a_{ij}(\tilde{x}_j - \tilde{x}_i) + g_i(-\tilde{x}_i)].\end{aligned}$$

Dynamiku odchylky celého systému lze zapsat jako:

$$\dot{\tilde{z}} = [\mathbf{I}_n \otimes \mathbf{A} - c(\mathbf{L} + \mathbf{G}) \otimes \mathbf{BK}] \tilde{z}.$$

O stabilitě systému (3.14) lze opět rozhodnout na základě lemmy 3.1. Při návrhu stavové zpětné vazby pomocí metody LQR je zapotřebí splnění následující podmínky k zaručení asymptotického sledování referenčního signálu, [4].

$$c \geq \frac{1}{2 \min_{\lambda_i} \Re(\lambda_i)}, \quad (3.22)$$

kde $\Re(\lambda_i)$ představuje reálnou část vlastního čísla λ_i a $\min()$ představuje minimum z dané množiny vlastních čísel. V této práci cílíme na cyklické, neorientované k -regulární grafy. Pro tento typ grafů jsou vlastní čísla Laplaciánu \mathbf{L} reálné a jejich hodnota klesá s přibývajícím počtem agentů. Podmínku (3.22) je tedy zapotřebí kontrolovat a zohledňovat převážně u aplikací s větším počtem agentů.

Příklad 3.11. Uvažujme systém se 6-ti agenty, kteří se pohybují v prostoru. Dynamika těchto agentů je dána předpisem:

$$\dot{z}_i = \begin{bmatrix} \mathbf{0} & \mathbf{I}_3 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} z_i + \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_3 \end{bmatrix} u_i, \quad z_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ v_{xi} \\ v_{yi} \\ v_{zi} \end{bmatrix}, \quad (3.23)$$

kde \mathbf{I}_3 představuje jednotkovou matici o rozměrech 3×3 a $\mathbf{0}$ udává nulovou matici stejných rozměrů.

Cílem úlohy je najít předpis distribuovaného řízení, které udrží stabilně systém obsahující 6 agentů v předepsané formaci, která opisuje zadanou trajektorii. Trajektorie je definovaná externím leadrem, jehož dynamika je popsána rovnicí:

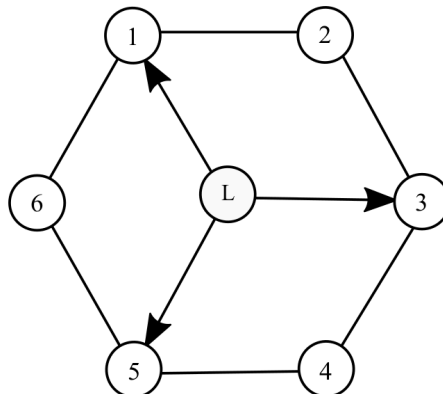
$$\dot{z}_0 = \begin{bmatrix} \mathbf{0} & \mathbf{I}_3 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} z_0 + \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_3 \end{bmatrix} u_{ref}, \quad z_0 = \begin{bmatrix} x_0 \\ \vdots \\ v_0 \\ \vdots \end{bmatrix}.$$

Předpis lokálního řízení pro *i-tého* agenta je zvolen jako:

$$u_{xi} = \dot{v}_0 + c \sum_{j=1}^N a_{ij} [(x_j - \Delta_j) - (x_i - \Delta_i)] + g_i(x_0 - x_i + \Delta_i).$$

Předpis řízení v osách y a z je obdobného charakteru, kde jsou změny pouze souřadnicové veličiny ($x \rightarrow y, x \rightarrow z$).

Topologie popisující informační provázanost jednotlivých agentů je dána grafem G :



Obrázek 3.20: Graf G reprezentující informační topologii systému, kde šipky reprezentují orientovanou hranu od leadra k daným agentům.

Matice Laplaciánu \mathbf{L} a matice \mathbf{G} reprezentující vazby agentů na leadra vypadají následovně:

$$\mathbf{L} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Zaved' me chybu i -tého agenta jako:

$$\tilde{z}_i = z - z_0 - \Delta_i,$$

kde z_0 představuje časový průběh požadované trajektorie a Δ_i udává požadovanou odchylku určující rozmístění formace. S využitím vztahu (3.18) lze formulovat chybu celého systému jako:

$$\dot{\tilde{z}} = (\mathbf{I}_n \otimes \mathbf{A} - c(\mathbf{L} + \mathbf{G}) \otimes \mathbf{BK})\tilde{z},$$

$$\tilde{z} = z - \bar{z}_0 - \bar{\Delta}, \quad \mathbf{A} = \begin{bmatrix} 0 & \mathbf{I}_3 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ \mathbf{I}_3 \end{bmatrix}, \quad \mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{L} + \mathbf{G} = \begin{bmatrix} 3 & -1 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}, \quad \mathbf{I}_n = \mathbf{I}_6 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Zpětnovazební matice \mathbf{K} rozměru 3×6 je určena tak, aby stabilizovala subsystém (3.23). K určení přesné hodnoty této matice využijeme jednu z uvedených metod návrhu, tedy přiřazení Jordanovy formy (2.4). Zvolíme požadované umístění pólů uzavřeného subsystému (3.23) jako:

$$\mathbf{Z} = \begin{bmatrix} -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 0 & 0 \\ 0 & 0 & -4 & 0 & 0 & 0 \\ 0 & 0 & 0 & -5 & 0 & 0 \\ 0 & 0 & 0 & 0 & -6 & 0 \\ 0 & 0 & 0 & 0 & 0 & -7 \end{bmatrix}.$$

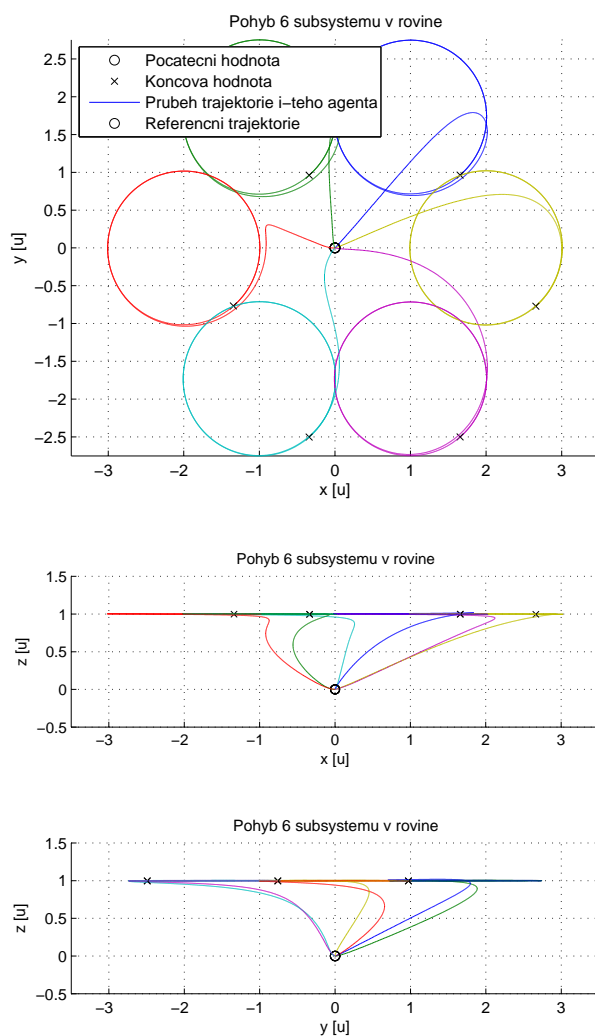
Řešením rovnice (2.5) získáme stavovou zpětnovazební matici \mathbf{K} ve tvaru:

$$\mathbf{K} = \begin{bmatrix} -13.0798 & -3.3306 & 6.8028 & -8.5190 & -1.7741 & 1.3014 \\ 7.5301 & -13.1429 & 9.9080 & 1.2645 & -7.7023 & 1.8586 \\ -6.3157 & 5.0223 & -28.3872 & -0.8198 & 1.8189 & -10.7786 \end{bmatrix}.$$

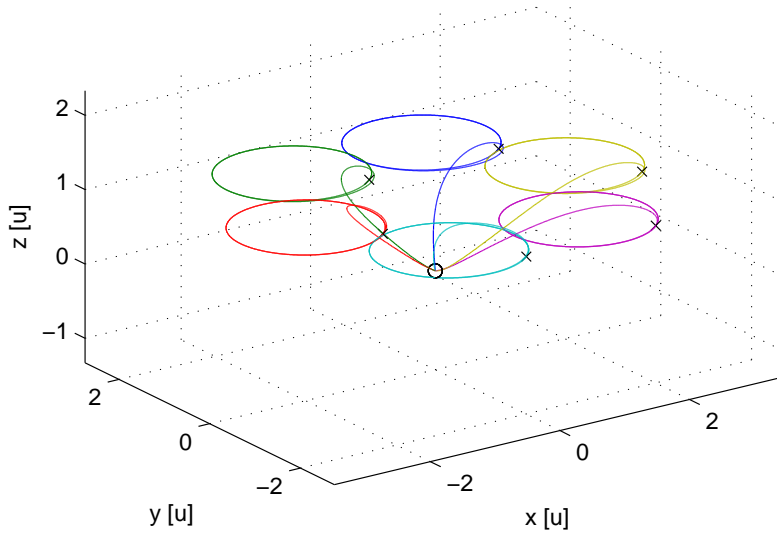
Zvolme referenční trajektorii leadra jako pohyb po kružnici ve zvolené výškové hladině, tj.:

$$u_{ref} = \begin{bmatrix} \sin(t) \\ \cos(t) \\ 1 \end{bmatrix}.$$

Požadované uspořádání agentů zvolme podobně jako v grafu G na obrázku 3.20, tedy rovnoměrné rozmístění po kružnici, a počáteční podmínky nulové. Simulace takto uvažovaného systému vypadá následovně:



Pohyb 6 subsystemu v prostoru



Obrázek 3.21: Průběh trajektorií systému tvořeného 6-ti agenty.

3.3.4 Návrh stavové zpětné vazby

V přípravné kapitole 2 byly představeny metody návrhu stavové zpětné vazby pomocí přiřazení Jordanovy formy a pomocí lineárního kvadratického regulátoru. Při aplikacích multiagentního řízení, kde jsou agenti vázani dynamikou vyšších řádů, je zapotřebí zaručit stabilitu dílších subsystémů, viz kapitola 3.3. Tvar výsledné zpětné vazby se odvíjí od požadavků na chování uzavřeného systému a každá z metod interpretuje tyto požadavky jiným způsobem.

Uvažujme systém, jehož agenti jsou popsáni dynamikou druhého řádu (3.11). Pohyb systému uvažujeme v rovině (x, y) , tedy:

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix}}_z = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}}_z + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_B \begin{bmatrix} u_x \\ u_y \end{bmatrix}. \quad (3.24)$$

Cílem je nalézt řízení $[u_x u_y]^T$ takové, aby systém dosáhl consensu v zadané

formaci, tedy:

$$u_x = c \sum_{j=1}^N a_{ij}[(x_j - \Delta_{xj}) - (x_i - \Delta_{xi})] + c\gamma \sum_{j=1}^N a_{ij}(v_{xj} - v_{xi}),$$

$$u_y = c \sum_{j=1}^N a_{ij}[(y_j - \Delta_{yj}) - (y_i - \Delta_{yi})] + c\gamma \sum_{j=1}^N a_{ij}(v_{yj} - v_{yi}),$$

Ukaŕme, Œe toto řízení zajistí shodu systému $x_i \rightarrow \Delta_{xi}, y_i \rightarrow \Delta_{yi}$, tedy:

$$\tilde{x}_i = x_i - \Delta_{xi},$$

$$\tilde{y}_i = y_i - \Delta_{yi},$$

$$\tilde{v}_{xi} = v_{xi},$$

$$\tilde{v}_{yi} = v_{yi},$$

$$\dot{\tilde{x}}_i = v_{xi},$$

$$\dot{\tilde{y}}_i = v_{yi},$$

$$\dot{\tilde{v}}_{xi} = u_x = c \sum_{j=1}^N a_{ij}(\tilde{x}_j - \tilde{x}_i) + c\gamma \sum_{j=1}^N a_{ij}(\tilde{v}_{xj} - \tilde{v}_{xi}),$$

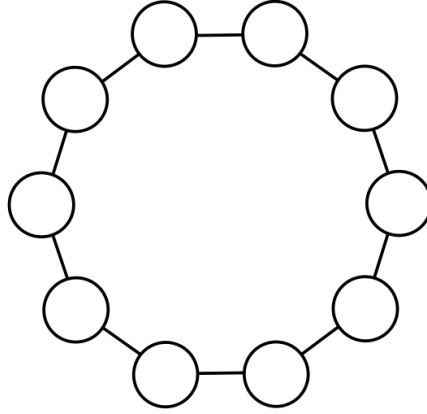
$$\dot{\tilde{v}}_{yi} = u_y = c \sum_{j=1}^N a_{ij}(\tilde{y}_j - \tilde{y}_i) + c\gamma \sum_{j=1}^N a_{ij}(\tilde{v}_{yj} - \tilde{v}_{yi}).$$

Pro celý systém lze psát (3.18):

$$\dot{\tilde{z}} = (\mathbf{I}_n \otimes \mathbf{A} - c\mathbf{L} \otimes \mathbf{BK})\tilde{z}.$$

Úkolem je najít hodnoty parametrů c a \mathbf{K} takové, aby stabilizovaly příslušný systém. Problematika stability celého systému byla rozebírána v lemmě 3.1. Zaměříme se na hledání zpětnovazební matice \mathbf{K} při pevné hodnotě parametru c . Cílem je nalézt takovou zpětnovazební matici \mathbf{K} , která bude optimální ve smyslu rychlosti konvergence.

Rychlost konvergence silně souvisí i s provázaností systému a počtem agentů, zafixujeme tedy tyto vlastnosti pevně následujícím grafem:



Obrázek 3.22: Graf G reprezentující topologii systému tvořeného deseti agenty.

Cyklický graf G , který je 2 -regulární, udává topologii systému popsáno rovnici:

$$\dot{\mathbf{z}} = \underbrace{(\mathbf{I}_{10} \otimes \mathbf{A} - c\mathbf{L} \otimes \mathbf{BK})}_{\mathbf{A}_c} \mathbf{z}, \quad (3.25)$$

kde \mathbf{I}_{10} představuje jednotkovou matici dimenze 10×10 . Necht' je váhový parametr $c = 1$. Vlastní čísla Laplaciánu \mathbf{L} jsou následující:

$$\Lambda(\mathbf{L}) = [0, 0.382, 0.382, 1.382, 1.382, 2.618, 2.618, 3.618, 3.618, 4]^T.$$

Vlastní čísla celého systému jsou určována dle lemma 3.1, tedy:

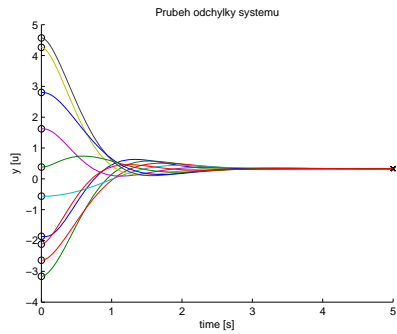
$$\Lambda(\mathbf{A}_c) = \mathbf{A} - c\lambda_i \mathbf{BK}, \quad i = 1, 2, \dots, 10.$$

Budeme-li určovat zpětnovazební matici \mathbf{K} metodou přiřazení Jordanovy formy vidíme, že vlastní čísla výsledného systému nebudou totožná s těma, která přiřadíme Jordanovou formou.

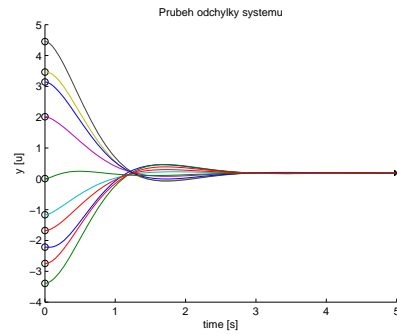
Zvolme požadované rozmístění agentů rovnoměrně po obvodu kružnice o poloměru $r = 1$ a náhodné počáteční hodnoty $\mathbf{z}(0)$. Ukažme chování systému pro následující množinu přiřazených Jordanových forem (2.4):

$$\mathbf{Z}_a = \begin{bmatrix} -3 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 \\ 0 & 0 & -5 & 0 \\ 0 & 0 & 0 & -6 \end{bmatrix}, \quad \mathbf{Z}_b = \begin{bmatrix} -3 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 \\ 0 & 0 & -5 & 0 \\ 0 & 0 & 0 & -5 \end{bmatrix},$$

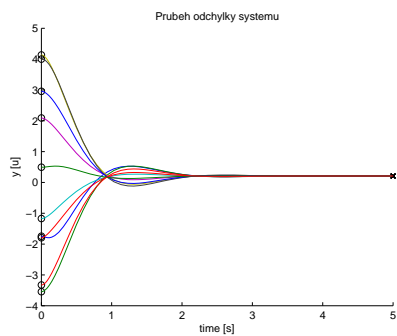
$$\mathbf{Z}_c = \begin{bmatrix} -5 & 1 & 0 & 0 \\ 0 & -5 & 0 & 0 \\ 0 & 0 & -5 & 1 \\ 0 & 0 & 0 & -5 \end{bmatrix}, \quad \mathbf{Z}_d = \begin{bmatrix} -3 & 1 & 0 & 0 \\ 0 & -3 & 0 & 0 \\ 0 & 0 & -5 & 1 \\ 0 & 0 & 0 & -5 \end{bmatrix}.$$



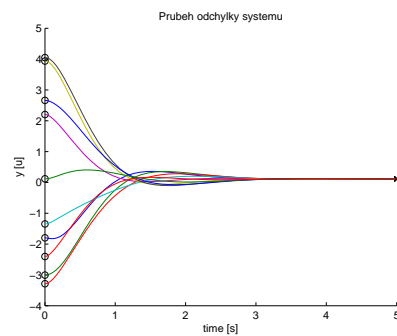
(a) Průběh chyby systému s přiřazením Jordanovy formy \mathbf{Z}_a všem jeho subsystémům.



(b) Průběh chyby systému s přiřazením Jordanovy formy \mathbf{Z}_b všem jeho subsystémům.



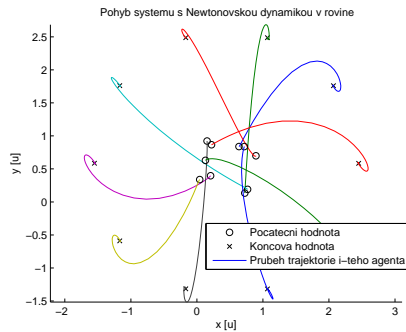
(c) Průběh chyby systému s přiřazením Jordanovy formy \mathbf{Z}_c všem jeho subsystémům.



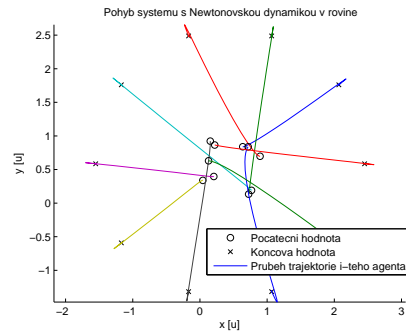
(d) Průběh chyby systému s přiřazením Jordanovy formy \mathbf{Z}_d všem jeho subsystémům.

Obrázek 3.23: Průběh odchylek agentů systému (3.25) v jedné ose.

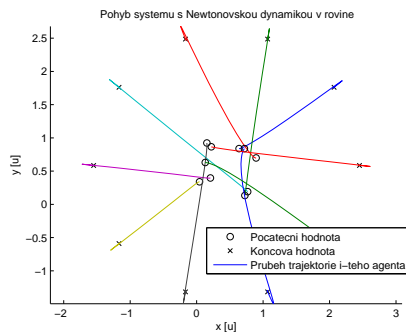
Zprůběhu odchylek se jeví nejlépe varianta \mathbf{Z}_c . Pokud vykreslíme tento průběh simulace v rovině a porovnáme ho například s variantou \mathbf{Z}_a , je patrné, že agenti konvergují k zadané formaci po přímějších trajektoriích.



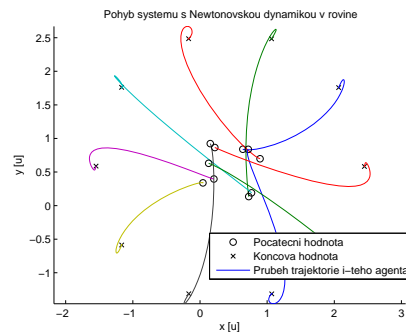
(a) Průběh polohy systému pohybujícího se v rovině s přiřazením Jordanovy formy \mathbf{Z}_a všem jeho subsystémům.



(b) Průběh polohy systému pohybujícího se v rovině s přiřazením Jordanovy formy \mathbf{Z}_b všem jeho subsystémům.



(c) Průběh polohy systému pohybujícího se v rovině s přiřazením Jordanovy formy \mathbf{Z}_c všem jeho subsystémům.



(d) Průběh polohy systému pohybujícího se v rovině s přiřazením Jordanovy formy \mathbf{Z}_d všem jeho subsystémům.

Obrázek 3.24: Průběh polohy agentů systému (3.25) v rovině konvergujících do požadované formace po obvodu kruhu.

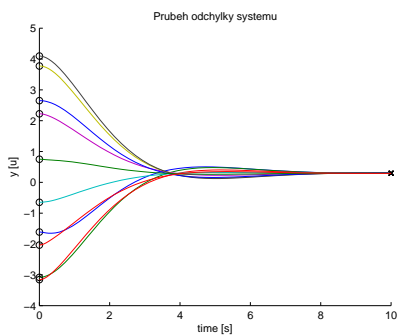
Pokud bychom přiřazovali póly s menší reálnou složkou, chyba systému by konvergovala rychleji. Tato rychlost by však byla za cenu větších energetických nákladů na řízení. K určení rovnováhy mezi těmito požadavky slouží návrhová metoda LQR představená v kapitole 2.3.1. Uvažujme tedy zpětnou vazbu subsystémů určenou pomocí metody LQR, kde zvolíme váhové parametry \mathbf{Q} a \mathbf{R} . Při hledání zpětnovazební matice \mathbf{K} řešením kritéria (2.6) je klíčový poměr váhových matic \mathbf{Q} , \mathbf{R} . Zvolme tedy pevnou matici $\mathbf{R} = \mathbf{I}_2$ a matici \mathbf{Q} volme z následující

množiny:

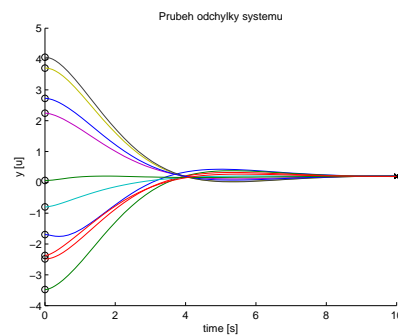
$$\mathbf{Q}_a = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 6 \end{bmatrix}, \quad \mathbf{Q}_b = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix},$$

$$\mathbf{Q}_c = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}, \quad \mathbf{Q}_d = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}.$$

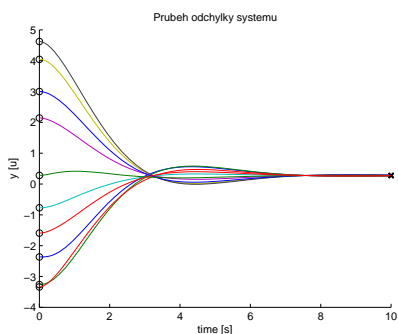
Povšimněme si, že varianty \mathbf{Q}_b a \mathbf{Q}_d nejsou stejné, neboť se v kritériu (2.6) přiřazuje váha konkrétnímu stavu. Jednou je tedy větší důraz kladen na polohu a jednou na rychlost. Průběh poloh subsystému, kde byla zpětná vazba subsystémů určena metodou LQR s příslušnými váhovými maticemi, vypadá následovně:



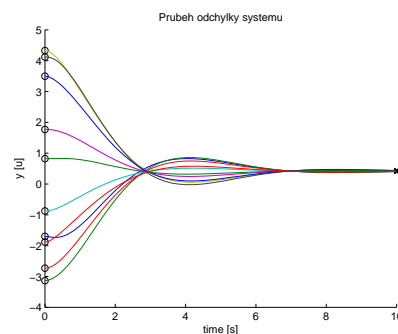
(a) Průběh chyby systému, jehož agenti mají přiřazenou stavovou zpětnou vazbu pomocí LQR s váhovou maticí \mathbf{Q}_a .



(b) Průběh chyby systému, jehož agenti mají přiřazenou stavovou zpětnou vazbu pomocí LQR s váhovou maticí \mathbf{Q}_b .



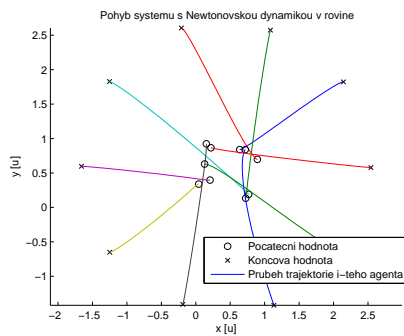
(c) Průběh chyby systému, jehož agenti mají přiřazenou stavovou zpětnou vazbu pomocí LQR s váhovou maticí \mathbf{Q}_c .



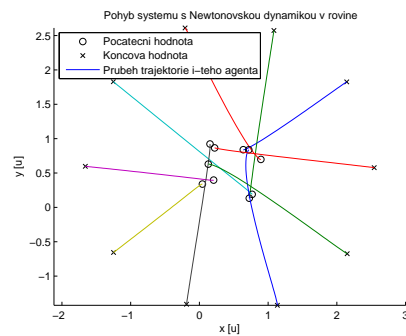
(d) Průběh chyby systému, jehož agenti mají přiřazenou stavovou zpětnou vazbu pomocí LQR s váhovou maticí \mathbf{Q}_d .

Obrázek 3.25: Průběh odchylek agentů systému (3.25) v jedné ose.

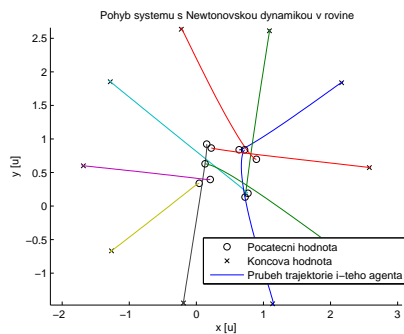
Povšimněme si téměř dvojnásobné doby ustálení všech variant simulací při použití stavové zpětné vazby pomocí LQR oproti přiřazení Jordanovy formy. Doba konvergence je delší pro všechny ukázané váhy, nicméně trajektorie polohy v prostoru daných agentů je přímější než v případě použití přiřazení Jordanovy formy.



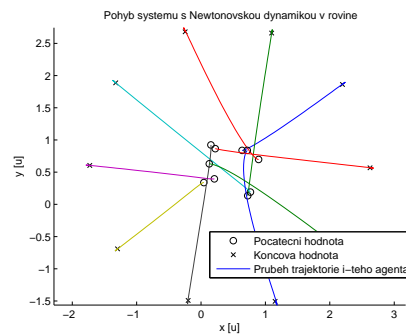
(a) Průběh trajektorie systému v rovině, jehož agenti mají přiřazenou stavovou zpětnou vazbu pomocí LQR s váhovou maticí Q_a .



(b) Průběh trajektorie systému v rovině, jehož agenti mají přiřazenou stavovou zpětnou vazbu pomocí LQR s váhovou maticí Q_b .



(c) Průběh trajektorie chyby systému v rovině, jehož agenti mají přiřazenou stavovou zpětnou vazbu pomocí LQR s váhovou maticí Q_c .



(d) Průběh trajektorie chyby systému v rovině, jehož agenti mají přiřazenou stavovou zpětnou vazbu pomocí LQR s váhovou maticí Q_d .

Obrázek 3.26: Průběh trajektorií agentů systému (3.25) v rovině konvergujících do požadované formace po obvodu kruhu.

Využitím poznatků z průběhu simulací variant zpětné vazby pomocí LQR a přiřazení Jordanovy formy můžeme navrhnout zpětnou vazbu, která využije vlastnosti obou přístupů. Navržením zpětné vazby pomocí návrhové metody LQR získáme uzavřenou smyčku pro daný subsystém. Výpočtem vlastních čísel dané smyčky získáme požadované rozmístění pólů, které vede na přímější trajektorie požadovaných odchylek. Posunutím daných pólů do zápornějších hodnot získáme

rychlejší konvergenci celého systému. Pro váhovou matici \mathbf{Q}_c získáme řešením LQR kritéria (2.6) zpětnovazební matici \mathbf{K} ve tvaru:

$$\mathbf{K} = \begin{bmatrix} -2.2361 & 0 & -3.0777 & 0 \\ 0 & -2.2361 & 0 & -3.0777 \end{bmatrix}.$$

Subsystém (3.24) je ve tvaru:

$$\dot{\mathbf{z}} = (\mathbf{A} + \mathbf{BK})\mathbf{z}.$$

Jordanova forma matice dynamiky $(\mathbf{A} + \mathbf{BK})$ má tvar:

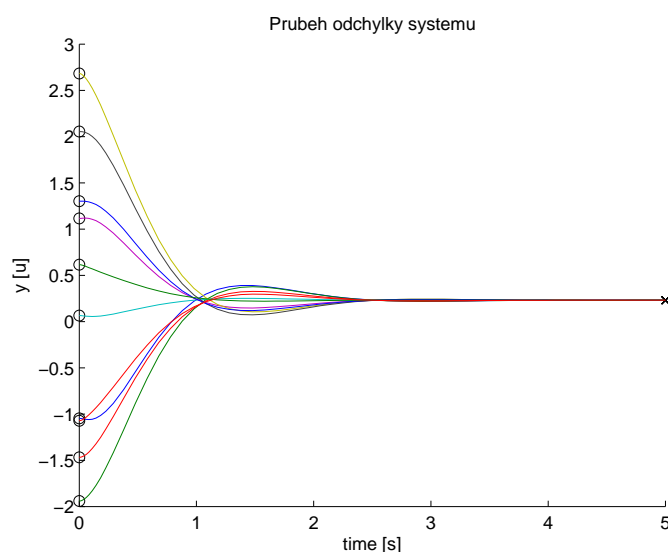
$$\mathbf{J}_F = \begin{bmatrix} -1.9021 & 0 & 0 & 0 \\ 0 & -1.9021 & 0 & 0 \\ 0 & 0 & -1.1756 & 0 \\ 0 & 0 & 0 & -1.1756 \end{bmatrix}.$$

Přiřadíme-li tuto formu přenásobenou vhodným koeficientem, v našem případě číslem 3, získáme požadované umístění pólů ve tvaru:

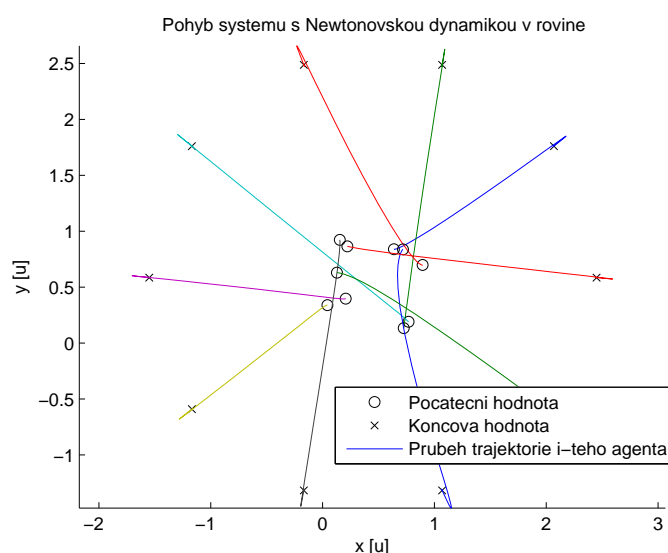
$$\mathbf{J}_F = \begin{bmatrix} -5.7063 & 0 & 0 & 0 \\ 0 & -5.7063 & 0 & 0 \\ 0 & 0 & -3.5268 & 0 \\ 0 & 0 & 0 & -3.5268 \end{bmatrix}.$$

Přiřazením zpětné vazby pomocí vztahu (2.4) získáme zpětnovazební matici \mathbf{K} ve tvaru:

$$\mathbf{K} = \begin{bmatrix} -20.1250 & 0 & -9.2331 & 0 \\ 0 & -20.1250 & 0 & -9.2331 \end{bmatrix}. \quad (3.26)$$



(a) Průběh chyby systému při použití návrhu zpětné vazby kombinací návrhových metod LQR a přiřazení Jordanovy formy.



(b) Průběh polohy systému při použití návrhu zpětné vazby kombinací návrhových metod LQR a přiřazení Jordanovy formy.

Obrázek 3.27: Průběh odchylek agentů systému (3.25) v rovině s využitím zpětnovazební matice (3.26).

Z průběhu simulace je patrné zlepšení rychlosti konvergence oproti návrhové metodě LQR. Přímost trajektorie je hůře měřitelná, nicméně by šla stále zlepšit. Zlepšení by bylo opět za cenu navýšení energetické náročnosti na řízení, což může být v některých aplikacích nežádoucí.

4 Závěr

V této práci jsme se zabývali problematikou distribuovaného kooperativního multi-agentního řízení, které je aplikovatelné na systémy tvořené množinou vzájemně propojených agentů (subsystémů). Diskutovali jsme problémy návrhu lokálního protokolu řízení jednotlivých agentů, kterým zajistíme stabilitu celé skupiny. Klíčovou vlastností k zaručení stability celého systému je kromě stability dílčích agentů i topologie jejich propojení, která je popisována pomocí neorientovaného grafu pro případ konsensu jednotlivých agentů a grafu s několika vybranými orientovanými hranami v případě uvažování leadra. Navržené řízení, které splňuje tyto požadavky, je demonstrováno pro systém tvořený agenty, jejichž dynamika je popsána diferenciálními rovnicemi prvního řádu. Pro toto řízení je následně ukázáno, jak zavést požadavek shody systému na zadané formaci a jak vhodným připojením leadra zavést požadavek na sledování referenční trajektorie. Tyto poznatky jsou poté s úpravami aplikovány pro lineární systémy vyšších řádů, u kterých jsme se soustředili zejména na použití multi-agentního řízení pro systémy popsanými diferenciálními rovnicemi druhého řádu, neboť zobecnění do vyšších dimenzí je již velice podobné. V případě nestability dílčích subsystémů je ukázáno, jak navrhnout stabilizující stavovou zpětnou vazbu pomocí metody lineárního kvadratického regulátoru a pomocí přiřazení Jordanovy formy. Výsledné řízení stabilizující celý systém s agenty popsanými dynamikou druhého řádu je pak tvořeno kombinací nalezené stavové zpětné vazby pro dílčí členy a lokálním protokolem řízení, který dává do souvislosti topologii propojení celého systému. Pro toto řízení jsou dále zobecněním předchozích poznatků zavedeny požadavky shody na zadané formaci systému a sledování referenční trajektorie. Výsledné řízení je demonstrováno na příkladech systému, jehož agenti se pohybují v trojrozměrném prostoru. V poslední části je diskutován návrh stavové zpětné vazby, u kterého jsou porovnány metody návrhu pomocí lineárního kvadratického regulátoru a přiřazením Jordanovy formy. Zaměřili jsme se na návrh stavové zpětné vazby, která vede na co nejvíce přímou trajektorii od počátečních hodnot ke koncovému stavu agentů.

Literatura

- [1] BREWER, J. Kronecker products and matrix calculus in system theory. *IEEE Transactions on circuits and systems*. 1978, 25, 9, s. 772–781.
- [2] FAX, J. A. – MURRAY, R. M. Information flow and cooperative control of vehicle formations. 2003.
- [3] FAX, J. A. *Optimal and cooperative control of vehicle formations*. PhD thesis, California Institute of Technology, 2002.
- [4] LEWIS, F. L. et al. *Cooperative control of multi-agent systems: optimal and adaptive design approaches*. Springer Science & Business Media, 2013.
- [5] MESBAHI, M. – EGERSTEDT, M. *Graph theoretic methods in multiagent networks*. 33. Princeton University Press, 2010.
- [6] MURRAY, R. M. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*. 2007, 129, 5, s. 571–583.
- [7] PALEN, J. et al. California Partners for Advanced Transit and Highways (PATH) enhanced loop-based traffic surveillance program. *ITS quarterly*. Vol. 8, no. 4. 2000.
- [8] PETERSEN, K. B. – PEDERSEN, M. S. – OTHERS. The matrix cookbook. *Technical University of Denmark*. 2008, 7, 15, s. 510.
- [9] REYNOLDS, C. W. *Flocks, herds and schools: A distributed behavioral model*. 21. ACM, 1987.
- [10] ROHN, J. Lineární algebra a optimalizace. *Karolinum, Praha*. 2004.
- [11] SCHLEGEL, M. Modální řízení konečněrozměrných lineárních systémů. *Disertační práce*. Vysoká škola strojní a elektrotechnická Plzeň, 1984.
- [12] SCHLEGEL, M. – KÖNIGSMARKOVÁ, J. Parametric Jordan Form Assignment Revisited. *Asian Journal of Control*. 2014, 16, 2, s. 409–420.
- [13] WILLEMS, J. C. *The analysis of feedback systems*. 197. MIT press Cambridge, MA, 1971.
- [14] YANAKIEV, D. – KANELAKOPOULOS, I. A simplified framework for string stability analysis in AHS. In *Proceedings of the 13th IFAC World Congress*, 182, s. 177–182, 1996.