

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA APLIKOVANÝCH VĚD
KATEDRA KYBERNETIKY

Diplomová práce

Návrh manipulátorů pro řízení autonomního pohybu
hráčů stolního fotbalu

Plzeň, 2022

Bc. Martin Jandík

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2021/2022

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Martin JANDÍK**
Osobní číslo: **A20N0024P**
Studijní program: **N3918 Aplikované vědy a informatika**
Studijní obor: **Kybernetika a řídicí technika**
Téma práce: **Návrh manipulátorů pro řízení autonomního pohybu hráčů stolního fotbalu**
Zadávající katedra: **Katedra kybernetiky**

Zásady pro vypracování

1. Navrhněte a sestavte mechatronický model stolního fotbalu s kamerovým systémem pro detekci pohybu míčku a robotickým manipulátorem ovládajícím pohyb gólmána a bránících hráčů.
2. Vytipujte návrh provedení sestavy pohonů a mechaniky dvouosého manipulátoru pro řízení pohybu soustavy hráčů.
3. Navrhněte algoritmus pro plánování a řízení pohybu bránících hráčů na základě odhadované trajektorie míčku. Otestujte na virtuálním modelu.
4. Realizujte navrženou strategii řízení na mechatronickém modelu s využitím vhodných HW a SW prostředků.

Rozsah diplomové práce: **40 – 50 stránek A4**
Rozsah grafických prací:
Forma zpracování diplomové práce: **tištěná**

Seznam doporučené literatury:


Melichar, Goubej, Lineární systémy 1,2, ZČU 2018
Vukosavic, Digital Control of Electrical Drives, Springer 2011
Duník, Identifikace systému a filtrace, ZČU 2018

Vedoucí diplomové práce: **Ing. Martin Goubej, Ph.D.**
Katedra kybernetiky

Datum zadání diplomové práce: **1. října 2021**
Termín odevzdání diplomové práce: **23. května 2022**



Doc. Ing. Miloš Železný, Ph.D.
děkan



Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

V Plzni dne 1. října 2021

PROHLÁŠENÍ

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

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

V Plzni dne

.....

Bc. Martin Jandík

PODĚKOVÁNÍ

Tímto bych rád poděkoval vedoucímu diplomové práce Ing. Martinu Goubejovi Ph.D. za cenné profesionální rady a připomínky. Dále bych rád poděkoval spolužákům a kolegům, hlavně z katedry kybernetiky, za vzájemnou pomoc při studiu a v neposlední řadě nesmím opomenout přátele a rodinu, bez kterých by studium i tato práce nebyly možné realizovat.

Abstrakt

Cílem této diplomové práce je návrh a následná realizace manipulátorů pro řízení autonomního pohybu hráčů stolního fotbalu. V úvodu práce se zaměříme na obeznámení mechatronických systémů a propojení role řízení systémů s rolí umělé inteligence. Následně si představíme problémy, které vznikají při návrhu manipulátorů a přesuneme se k vlastnímu řešení práce, které se skládá z návrhu obecné architektury systému, návrhu mechanické části, tvorbě virtuálního modelu v prostředí Simscape, algoritmů pro řízení pohonů a sledování míčku a nakonec implementací všech zmíněných částí a jejich napojení na umělou inteligenci. V závěru práce si ukážeme výsledky provedených experimentů a to jak v simulaci s virtuálním modelem, tak na reálném fyzikálním systému. Tyto výsledky z reálného systému a simulací na modelu si nakonec porovnáme.

Klíčová slova

BLDC motor, simulace v prostředí Simscape, návrh řídicího systému, Kalmanův filtr, autonomní pohyb, stolní fotbal

Abstract

The main goal of this diploma thesis is the design and subsequent implementation of manipulators for controlling the autonomous movement of table football players. In the introduction we will focus on the familiarization of mechatronic systems and the connection of the role of system control with the role of artificial intelligence. We will then describe the problems that arise in the design of manipulators and move on to our own solution, which consists of designing the general system architecture, designing the mechanical part, creating a virtual model in Simscape, engine control algorithms and ball tracking, and finally implementing all the above parts and their connection with artificial intelligence. At the end of the work we will show the results of the performed experiments both in the simulation with the virtual model and on the real physical system. We will finally compare these results from the real system and simulations on the model.

Keywords

BLDC motor, simulation in add-on Simscape , design of control system, Kalman filter, autonomous movement, table football

Obsah

1 Úvod	8
1.1 Mechatronické systémy, manipulátory a roboty	9
1.2 Role řízení a umělé inteligence a jejich propojování	11
1.3 Formulace problémů při návrhu manipulátorů	14
2 Vlastní řešení	18
2.1 Návrh obecné architektury systému	18
2.2 Návrh mechanické části	24
2.3 Virtuální model v prostředí Simscape	28
2.4 Řízení pohonů	31
2.5 Algoritmy pro sledování míčku	41
2.6 Implementace všech částí a napojení na AI část	48
3 Výsledky experimentů	51
3.1 Simulace s virtuálním modelem	51
3.2 Fyzikální systém	52
3.3 Porovnání simulace a reálného systému	53
4 Závěr	56
Literatura	58
Seznam obrázků	59

Kapitola 1

Úvod

Tématem této diplomové práce je návrh manipulátorů pro řízení autonomního pohybu hráčů stolního fotbalu, který se následně stane dostatečně kvalitním soupeřem pro hru stolního fotbalu.

Celá práce je rozdělena do tří pomyslných celků. V úvodu se podrobněji seznámíme s mechatronickými systémy, manipulátory a roboty. Také se dozvíme jakou roli plní řízení a umělá inteligence ve všelijakých projektech a formulujeme si problémy, se kterými se potkáme, pokud se rozhodneme navrhovat manipulátor. V druhé, tedy prostřední kapitole se budeme věnovat vlastnímu řešení projektu. Na začátku si přiblížíme obecnou architekturu systému, abychom získali elementární přehled o všech částech. Tyto části se skládají z návrhu mechanické části, kde si projdeme jednotlivé celky, virtuálního modelu v prostředí Simscape, který umožňuje simulovat pohyb reálných objektů, části věnované řízení pohonů a přiblížení řešení jednotlivých problematik s reálnými pohony. Dále si představíme algoritmy, které můžeme použít pro sledování míčku včetně algoritmu Kalmanova filtru, který poté v úloze použijeme. Na závěr druhé kapitoly si přiblížíme implementaci všech zmíněných částí a jejich napojení na část umělé inteligence. V poslední kapitole si představíme výsledky experimentů, které probíhaly nejprve při simulaci s virtuálním modelem a následně i na reálném fyzikálním systému. Tyto výsledky porovnáme a náležitě okomentujeme jejich vlastnosti.

Tato práce je zároveň rozšířením předchozí bakalářské práce [2], kde jsme začali tuto problematiku. Cílem této práce je tedy vylepšení původního manipulátoru o druhou osu a odstranění nedostatků, na které jsme u původního řešení narazili. Zároveň je třeba zmínit, že původní cíl o dosažení dostatečně přesného a agresivního řídicího systému zůstává zachován a motivace přesunout tento projekt až na nasazení kompletního řešení stolního fotbalu taky.

1.1 Mechatronické systémy, manipulátory a roboty

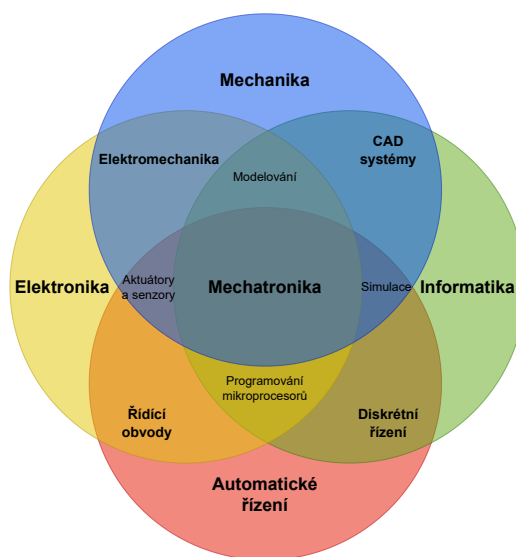
Historie

Lidská vynalézavost a důvtip nám od nepaměti pomáhá zlepšovat své schopnosti a ulehčovat veškerou práci. V počátcích jsme si naše aktivity usnadňovali vytvářením různých ručních nástrojů, které nám pomáhaly při fyzických činnostech. Tyto nástroje sloužily především k zajištění základních lidských potřeb. Jak šel čas, lidstvo postupně zdokonalovalo tyto nástroje až do bodu, kdy s nimi bylo možné vytvářet stále více sofistikovanější stroje. Ze začátku se jednalo hlavně o parní stroje, které spustily průmyslovou revoluci a umožnily lidem cestovat a přemísťovat těžký náklad po železnici na dlouhé vzdálenosti. O pár stovek let později, s příchodem elektřiny a vynechání několika historických událostí, byl člověk schopen postupně přejít až k prvnímu počítači. Lidský pokrok se od tohoto okamžiku začal několikanásobně zrychlovat, a to zejména v technické oblasti či v rozvoji lidského poznání. Vznikaly nové vědní obory, které, jak se dozvíme dále, jsou mimo jiné také základním stavebním kamenem mechatronických systémů.

Mechatronické systémy

Mechatronické systémy nebo mechatronika spojují obecné principy mechaniky, elektroniky, automatického řízení a informatiky. Výsledkem tohoto spojení je konstrukce systémů, které jsou jednodušší a účinnější než systémy, které jsou navrženy v jednotlivých vědních oborech samostatně. Jako příklad mechatronických systémů si můžeme představit protiblokovací systém brzd známý pod zkratkou ABS, aktivní tlumení vibrací, různé počítačem řízené obráběcí stroje nebo v dnešní době velice rozšířený 3D tisk. Toto je jen malý výčet aplikací, které se skrývají pod pojmem mechatronické systémy a rok co rok vznikají nové a už transformací nedokonalých mechanických systémů nebo nově vzniklé systémy, které dříve nebylo možné vytvořit. Na jednoduchém příkladu dálkově ovládaných vjezdových vrat si ukážeme, kde se schovávají všechny vyjmenované vědní obory. Mechanika zde tvoří samotná brána, kterou se snažíme navrhnout tak, aby byla lehká, odolná proti povětrnostním podmínkám a zároveň zabránila nedovolenému vniknutí, například zničením. Dále ji tvoří převodníky rotačních pohybů a různé aretace v koncových polohách. Elektronika představuje veškeré senzory, mikrokontroléry a zbývající elektronické součástky včetně dálkového ovládání. Díky informatice

jsme schopni namodelovat si mechanickou část a dále simulovat její vlastnosti jako například mechanické zatížení. Dále jsme schopni naprogramovat zašifrovanou dálkovou komunikaci, simulovat a rychleji navrhovat řídicí systémy, které spadají do vědní disciplíny automatického řízení. Navržené řízení přenešeme do zdrojového kódu a nahrajeme do mikrokontroléru. Tímto uzavřeme spojení všech zmíněných oborů a prozaicky řečeno, vdechneme život celé aplikaci. Pro lepší představu je toto propojení ještě zobrazené na následujícím obrázku 1.1.



Obrázek 1.1: Vennův diagram znázorňující kompozici mechatroniky

Manipulátory

Jedná se o nástroj, který pomáhá či nahrazuje periodickou lidskou práci. Pro jeho přesnost, bezchybnost a nepřetržitou práci je ideálním služebníkem nejen na výrobních linkách. S manipulátory se můžeme setkat také v lékařství, ve skladech, v průmyslu nebo v místech, které jsou pro člověka jakýmkoliv způsobem nehostinné. Jak již bylo psáno výše, v podkapitole historie, manipulátory byly dříve pouze mechanické zařízení, které se postupem času staly zařízením elektromechanickým. Nástroj, na kterém lze tento pokrok dobře pozorovat, je paletový vozík, kde bylo dříve nutné s vozíkem ručně přijet, zvednout závaží a přemístit na určenou pozici. Dnes už máme k dispozici elektrický paletový vozík, který práci zlehčuje, ale k jeho ovládání je nezbytná lidská přítomnost. V pokročilých řízených skladech či docích už

odpadla i tato povinnost. Manipulátory zde práci nejen ulehčují, ale také zpřesňují a plně automatizují.

Roboty

Když se řekne robot, většina lidí si představí nějaký stroj podobný člověku, který se lidem snaží pomoci nebo ublížit. Za tyto představy jsou zodpovědní tvůrci sci-fi filmů, například série filmů Hvězdných válek nebo Terminátor, kde se tyto postavy objevují. Tato představa však není chybná. Jedná se totiž o humanoidní roboty, které se vyznačují podobností vzhledu a myšlením lidem. Tyto roboty jsou postavené na mechatronickém základu, stejně jako dříve zmíněné manipulátory, ale jejich záměr spočívá spíše ve vysoké úrovni autonomie a dokonalé interakci s reálným prostředím. Jejich hlavní myšlenka tedy není v odvádění těžké práce, nýbrž důmyslném využití úloh počítačového vidění, rozpoznávání řeči a porozumění či rozhodování v daných situacích.

Aby však nedošlo k mylné představě, je třeba zmínit, že humanoidní roboty jsou pouze podkategorií robotů. Slovo robot pochází z knihy R.U.R. od českého spisovatele Karla Čapka a pod tento pojem lze zařadit také pomocníky v domácnosti, jako například robotický vysavač nebo multifunkční kuchyňské roboty, v blízké budoucnosti autonomní vozidla nebo již dříve diskutované manipulátory.

1.2 Role řízení a umělé inteligence a jejich propojování

Role řízení

Jednoduše můžeme říct, že role řízení dokáže lokálně měnit fyzikální zákony a pohled na svět, jak ho známe. Pro příklad si uvedeme inverzní kyvadlo na vozíku. Pokud přivedeme kyvadlo do svislé nestabilní polohy a necháme ho být, tak pokud zde není veliké tření, kyvadlo po chvíli spadne a bude se kývat do té doby, dokud se vlivem tření nezastaví ve svislé stabilní poloze. Pokud se však do tohoto problému vloží člověk, který disponuje znalostmi automatického řízení, je schopný dosáhnout stavu, kdy kyvadlo setrvá v nestabilní poloze. Abychom ale získali ještě detailnější obraz, o tom, co představuje role automatického řízení v nejrůznějších aplikacích, můžeme se vrátit zpět k mechatronickým systémům a k obrázku 1.1. Pokud to vezmeme popořadě, tak vždy začínáme nějakým modelem systému, který chceme řídit. V případě

inverzního kyvadla existuje přímo matematický model. Tento matematický model linearizujeme v nestabilní poloze a získáme lineární systém, který jsme schopni řídit. Na zmíněný lineární systém použijeme jednu z mnoha strategií řízení a tím získáme řídicí algoritmus ve spojitém čase, který udrží kyvadlo v nestabilní poloze. Abychom se vrátili k obrázku 1.1, tak tento algoritmus jsme museli simulovat, abychom si jej ověřili. Nyní je třeba tento algoritmus simulovaný ve spojitém čase upravit pro výpočetní techniku, tedy je potřeba ho diskretizovat, protože procesor, který následně v aplikaci použijeme, pracuje s diskrétním časem, s konkrétní periodou. Co jsem nezmínil, tak k pohybu vozíku je potřeba nějaká externí síla. Tuto sílu dodává aktuátor, který ve spoustě aplikací představuje elektromotor. Poslední věc, která dosud zůstala bez povšimnutí, jsou senzory, které jsou spolu s aktuátory důležitou součástí každé uzavřené smyčky. V případě inverzního kyvadla na vozíku snímáme polohu vozíku a natočení kyvadla.

Role umělé inteligence

Napodobovat lidskou inteligenci ve smyslu iterativního učení z předchozích znalostí je technologie, kterou nazýváme umělá inteligence. Můžeme si ji představit jako formu inteligence, kterou lze použít k řešení problémů, k získání odpovědí na otázky, k vymýšlení řešení úloh nebo k předpovídání a navrhování strategií.

Funkce umělé inteligence spočívá v kombinaci velkého množství dat s inteligentními, iterativními algoritmy, které dokáží zpracovat data, které analyzují a následně se naučí jejich vzorce a souvislosti. V průběhu každé iterace otestuje a ohodnotí aktuální výsledek a na jeho základě pak rozvíjí další znalosti. Čím víc má umělá inteligence dat a čím více jsou její algoritmy sofistikovanější, tím lepší výsledky bude vykazovat.

V příkladu uvedeném výše, tedy úloze s inverzním kyvadlem na vozíku umělá inteligence nefiguruje. Bylo by možné ji použít místo senzorů a získávat informace o poloze vozíku a úhel natočení kyvadla pomocí počítačového vidění. Toto použití by bylo zbytečně nákladné a nepřineslo žádné významné zlepšení. Kde se však můžeme setkat s umělou inteligencí a její použití zde dává smysl, jsou aplikace, kde neexistuje senzor pro danou veličinu nebo by jeho použití bylo velice drahé. Umělou inteligenci ale nemusíme nutně využívat pouze jako náhradu za senzor. Jako příklady si můžeme uvést metodu podpurných vektorů, se kterou jsme se jako jednou z prvních úloh setkali při studiu, pokročilé překladače vět a internetové vyhledávače, úlohy pro rozpoznávání obrazu a rozpoznávání řeči, v dnešní době hodně diskutované

použití nejen v marketingu na sociálních sítích, programy sloužící k hraní her a mnoho dalších. S postupem času narůstá výpočetní výkon a paměťové možnosti počítačů což způsobuje posouvání hranic využití umělé inteligence k sofistikovanějším a rozsáhlejším projektům.

Propojení rolí

Když už máme nějakou představu o tom, co představují jednotlivé role a příklady jejich použití, můžeme si přiblížit aplikace, kde se v jednom projektu využijí obě role najednou. Takových projektů je spousta, ale v posledních letech jsou hodně diskutovaným tématem autonomní vozidla. Právě autonomní vozidla představují ukázkový příklad toho, co se stane když se role řízení a umělé inteligence propojí. Role řízení zde představuje v první řadě bezchybný a plynulý chod vozidla, tedy zvládnutí práce s aktuátory. Co se ale autonomie týče, tak řízení se zde stará o regulaci poruch, které způsobuje například nerovnost vozovky, regulaci řízení automobilu na požadované hodnoty získanou z výstupu umělé inteligence či ovládání teploty v interiéru.

Senzorovou část, potřebnou pro autonomní jízdu, zde z velké části zastupuje umělá inteligence, která zároveň celou cestu do cílové destinace poskytuje požadované hodnoty regulátorům. Z každého snímku senzoru či obrazu kamery vyhodnocuje rizika blížící se překážky, rozpoznává dopravní značení, predikuje pohyb nejen protijedoucích vozidel, ale i všech ostatních účastníků silničního provozu. Na základě všech těchto informací je schopna sestavit správnou požadovanou hodnotu, kterou následně odešle do řídicího algoritmu, který z uživatelského hlediska provede setrvání jízdy v pruzích na požadované rychlosti, zvládnutí dopravní situace v křižovatce nebo dokonce vyhnutí kolize způsobené neočekávanými situacemi.

Druhé populární téma propojení rolí řízení a umělé inteligence jsou humanoidní roboti. Zde je hlavní myšlenka v podstatě stejná jako u autonomních vozidel. Podstata je tedy zvládnout orientaci v prostoru, správně reagovat v tomto reálném prostředí a plnit roli, pro který tento robot vznikl. U autonomních vozidel tato role představuje odvést pasažéry z bodu A do bodu B. U humanoidního robota se jeho role může měnit například na základě povelu, který mu zadá jeho majitel či nadřízený

1.3 Formulace problémů při návrhu manipulátorů

V této podkapitole se posuneme od obecného pojetí mechatronických systémů a definice rolí řízení a umělé inteligence k bližšímu zaměření na problém návrhu manipulátoru. Postupně si nastíníme problémy, které vznikají a je nutné se s nimi vypořádat v oblasti návrhu manipulátorů, řízení autonomního pohybu a zpracování obrazu v úloze návrhu manipulátorů pro řízení autonomního pohybu hráčů stolního fotbalu.

Návrh manipulátorů

Návrh manipulátoru se dělí na dvě části. Návrh mechanické části manipulátoru a část návrhu řízení. Vždy je lepší, když obě části navrhujeme současně, protože se pak nemůže stát, že vlivem nějaké ustálené mechanické vlastnosti nelze manipulátor řídit tak, jak požadujeme. Jedná se také o správné umístění aktuátorů a senzorů, které je z pohledu řízení dost podstatné. Špatná konstrukce nebo chybné umístění senzorů pak může způsobit omezenou pohyblivost, nízkou rychlost, poruchovost a dále. Naštěstí touto možností návrhu obou částí současně disponujeme, a jsme schopni ji plně využít.

Z konstrukčního hlediska se snažíme vytvořit co nejjednodušší převod rotačního pohybu na posuvný, který rozpohybuje tyč s hráči v jednom směru. V předchozí práci[2] jsme zjistili, která možnost tohoto převodu vychází nejlépe a tu jsme zde znovu použili. Jedná se o převod pomocí řemene, který pohybuje vozíkem. Co však od předchozí verze přibylo, je rotační pohyb hráčů. Kvůli co nejnižší hmotnosti vozíku a potlačení nelinearity, které by zde vznikly, se v počátku objevily myšlenky opracovat konec tyče, který už nezasahuje do hry, ozubením. Výhoda tohoto řešení by byla, že by oba motory byly staticky upevněné ke konstrukci arény. Tento mechanismus by sice minimalizoval zmíněné nelinearity, ale způsobil by vysoké nároky na nízké tření v místě styku onoho ozubení s případným řemenem převádějícím moment motoru na tyč s hráči. Tyto nároky na vysokou přesnost tyče společně s vyššími náklady nás od této myšlenky nakonec odradily a došli jsme k názoru, že první varianta s druhým motorem připevněným na vozíku je výhodnější. Motor má tedy v jedné ose spojenou hřídel s tyčí a hráči konajícími rotační pohyb.

Toto umístění aktuátorů způsobí navíc to, že se úloha stane jednodušěji říditelnou. Je to dáno tím, že mezi jednotlivými pohyby není žádná vzájemná vazba, a tak můžeme řídit oba pohyby samostatně bez nutnosti použít komplexnějších řídicích technik robustního návrhu řízení. Jednotlivé pohyby tedy

vyřeší samostatné uzavřené regulační smyčky a o komplexní pohyby hráčů se postará až umělá inteligence.

Problém umístění sensorů je zde vyřešen jen z části, protože senzory potřebné pro správné řízení aktuátorů jsou obsaženy přímo v nich. Protože se, už jen z podstaty věci, nejedná o absolutní senzory, je nutné vyřešit umístění koncových sensorů. Tyto senzory jsou důležité, protože zajistí, že při zapnutí celého zařízení se následně provede kalibrace poloh. Umístění koncových sensorů pro horizontální pohyb je realizované pomocí fyzického spínacího kontaktu v ose pohybu tak, aby se spínač sepnul, když do něj vozík s motorem narazí. V momentě sepnutí si manipulátor v tomto směru vynuluje pozici. Provedení této kalibrace je důležité hlavně kvůli ohraničení pracovního prostoru a tím pádem nedocházelo k neúmyslnému poškozování. Druhý sensor, pro rotační pohyb hráčů, by bylo možné realizovat například pomocí indukčních sensorů a nenáročné úpravě tyče s hráči. Kvůli plnému využití potenciálu umělé inteligence a nulovým dodatečným finančním výdajům je však kalibrace úhlu natočení hráčů zajištěna pomocí techniky prahování, která v každém snímku při postupném otáčení hráči vypočítává obsah plochy hráče. Pokud je tato plocha největší či nejmenší, jedná se o předem daný úhel natočení, který se následně použije jako referenční.

Řízení autonomního pohybu

Při snaze cokoliv řídit vzniká spousta otázek a problémů, se kterými je nutné se vypořádat. Nejprve je potřeba identifikovat systém, na který chceme aplikovat řídicí smyčku, kvůli možnosti počítačové simulace a podrobnému zkoumání jeho chování. Po provedení identifikace je potřeba zvolit jednu ze spousty strategií řízení. Následně je nutné zjistit, jestli je možné, za námi stanovených požadavků, takový systém řídit. Pokud jsou naše požadavky proveditelné, tak s vybranou strategií opakovaně provádíme simulace a upravujeme parametry do té doby, dokud nebudeme s výsledky simulace spokojeni. V dalším kroku, když už máme naladěné parametry, nasadíme řídicí algoritmus na reálný systém. V případě, že jsme doposud postupovali správně, navržený řídicí systém bude na reálném systému odpovídat výsledkům ze simulací. Pokud se tak nestane, protože je například model systému z počítačové simulace hodně rozdílný od reálného, je možné znovu doladit parametry, tak abychom vyhověli zadání. Návrh řídicího systému je tak u konce. Takhle lze v jednoduchosti shrnout všechny potíže, se kterými je nutné se při návrhu řídicí smyčky vypořádat. Všem těmto překážkám se budeme v průběhu této práce podrobněji věnovat.

Součástí řídicího algoritmu je však také reagovat na vstupní požadované hodnoty. Tyto hodnoty jsou generovány nějakou trajektorií pohybů, které má náš manipulátor provést. Jsou zde v zásadě dvě možnosti, které se nabízejí jako správné řešení, jak se vypořádat s úlohou generování trajektorií. První možnost je vytvořit nějaký soubor předem nadefinovaných pohybů a ty poté posílat do řídicích algoritmů. Tyto, například bránící sekvence pohybů, lze náhodně střídat. Tento přístup by ale nebyl příliš úspěšný při hře proti zkušeným hráčům, kteří, věřím, by tuto strategii brzy odhalili a získali tak převahu. To by se stalo i kdyby se tyto sekvence náhodně nestřídaly, ale užívaly se za nějakých pevně stanovených podmínek. Zde by poté stačilo protihráči zdokonalit daný pohyb tak, aby pevně danou obrannou sekvenci překonal. Tím pádem by mohl používat pořád dokola jeden útočný manévr, se kterým by se náš systém nedokázal vypořádat. Druhá možnost, která se nabízí je použití umělé inteligence. Tato umělá inteligence se natrénuje na softwarovém modelu, který by co nejpřesněji odpovídal realitě, a poté na základě toho, co se naučí, by se měla adaptovat na jakýkoliv manévr a následně se s ním vypořádat. V tomto případě bychom neměli pevně uložené sekvence pohybů, které by se posílali jako požadované hodnoty do řídicích smyček. Dynamicky měnící se požadované hodnoty by generovala umělá inteligence na základě aktuální situace. Jedinou překážkou by tak měla zůstat rychlost a síla aktuátorů, rychlost a kvalita zpracování obrazu a kvalita učícího algoritmu s dostatečně bohatými a přesnými vstupními daty.

Zpracování obrazu

Značnou část kompletního řešení stolního fotbalu tvoří zpracování obrazu. Jak bylo řečeno výše, je na něm závislá přesnost kalibrace referenčního úhlu natočení a jak bude řečeno dále, ovlivňuje správnost získaných informací, konkrétně polohy míčku a protihráčů. Od polohy míčku se následně odvíjí i predikce jeho polohy a pokud je tato informace nedůvěryhodná, bude nedůvěryhodná i predikce jeho polohy v budoucnu. Je tedy klíčové aby bylo se zpracováním obrazu nakládáno s dostatečnou péčí. Protože se jedná o výpočetně náročnou operaci, je zde nutné najít nejlepší poměr mezi kvalitou zpracovávaného obrazu a rychlostí zpracování. Čím kvalitnější bude zpracovávaný obraz, tím bude vyšší přesnost výstupů po zpracování, ale zároveň se zvýší výpočetní náročnost celé operace. Protože je nutné z každého snímku získat několik informací, je nutné vzít v potaz rychlost výpočetní techniky, která bude pro úlohu použita.

Další bariéru, se kterou je nutné se vypořádat, představují dynamicky se měnící světelné podmínky. Pro jednoduchost řešení a konzistenci výsledků

je vhodné umístit celý projekt do, z hlediska světelných podmínek, homogenního prostředí. Je zřejmé, že se toho v praxi ne vždy podaří dosáhnout, a proto jsme se na tento problém zaměřili a přizpůsobili jsme různé prvky projektu pro podmínky, které jsou přibližně shodné s denním světlem bez přímého svitu slunce. Například výběr barev hráčů obou týmů je pro zjednodušení vybrán podle toho, jak na zvoleném zeleném pozadí lze hráče v obraze dobře segmentovat.

Poslední překážka již byla zmíněna výše a týká se zkreslení obrazu, které je způsobené vlivem vlastností kamery. Tento jev lze kompenzovat použitím techniky, které se říká narovnání obrazu. Zjednodušeně lze říct, že základem této operace je maticová rotace a translace oblasti zájmu, v našem případě hřiště, po jehož aplikaci získáme narovnaný obraz. V naší situaci je toto zkreslení zanedbatelné a tím pádem ho nebudeme opravovat. Tato skutečnost se hodí, protože klademe vysoké nároky na rychlost odezvy.

Výsledkem této práce však nemá být pečlivě provedené zpracování obrazu, ale jedná se o součást společného snažení ve vytvoření plně funkčního automatizovaného stolního fotbalu. Tímto zakončíme nastínění problémů při návrhu manipulátorů a přesuneme se k vlastnímu řešení této práce.

Kapitola 2

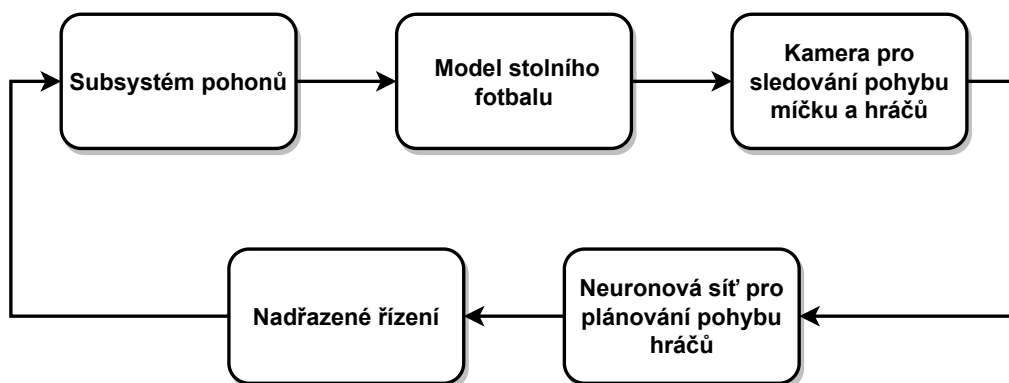
Vlastní řešení

V této kapitole se budeme zabývat vlastním řešením úlohy návrhu manipulátorů pro řízení autonomního pohybu hráčů stolního fotbalu. Detailně si probereme jednotlivé části architektury systému a návrh mechanické části celku. Ukážeme si také virtuální model v prostředí *Simscape*, který použijeme pro simulaci všech pohyblivých částí a navržení řízení pohonů, které se starají o pohyb hráčů a ovlivňují tak pohyb míčku. Dále se zaměříme na algoritmy pro sledování míčku, díky kterým budeme schopni predikovat jeho polohu a nakonec si probereme implementaci všech použitých částí a jejich napojení na část umělé inteligence.

2.1 Návrh obecné architektury systému

Návrh manipulátorů pro řízení autonomního pohybu hráčů stolního fotbalu je obecně složitější problém, který lze rozdělit na jednotlivé podproblémy. Toto rozdělení můžeme vidět na obrázku 2.1, kde uprostřed vidíme samotný model stolního fotbalu, který představuje fyzickou část tvořenou arénou, míčkem, hráči a tyčemi které hráče drží. Tedy klasický stolní fotbal, který můžeme potkat na veřejnosti například v tréninkových centrech pro ligové hráče nebo v některých restauračních zařízeních.

Součástí kompletního mechatronického modelu stolního fotbalu je celý vrchní řádek, který reprezentuje hardwarovou část systému, jmenovitě ještě subsystém pohonů, zajišťující pohyb hráčů a kamera pro sledování pohybu míčku a útočníků. Spodní část schéma je naopak zaměřena na softwarovou část, která obsahuje nadřazené řízení a neuronovou síť pro plánování pohybu hráčů.



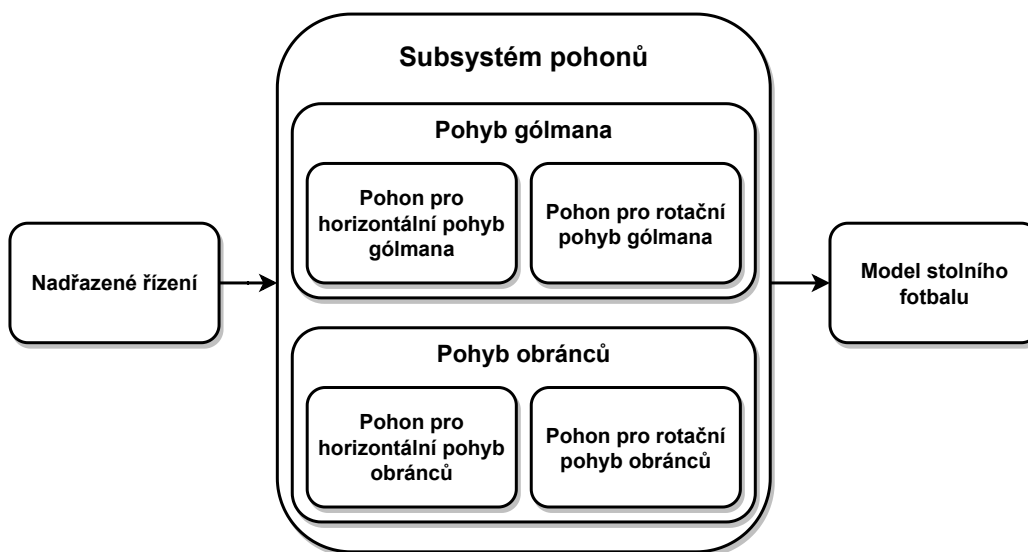
Obrázek 2.1: Základní schéma architektury systému

Subsystém pohonů

Pokud chceme dosáhnout plného rozsahu pohybu na jedné tyči na jakékoliv sadě hráčů od gólmána po útočníky, musíme získat dva stupně volnosti a toho docílíme použitím dvou motorů. Jeden motor se nachází v ose tyče a bude zajišťovat rotační pohyb tyče. Druhý motor umístěn kolmo k tyči a pomocí převodníku rotačního pohybu na lineární pohyb pokryje druhý a zároveň poslední stupeň volnosti. Na obrázku 2.2 vidíme hierarchický popis pohonů zasazený do základního schéma architektury systému. Bližší představu dále získáme v kapitole 2.2, kde se budeme zabývat touto problematikou dopodrobna.

Z předchozí práce[2], kde jsme si podrobně popsali rozdíly jednotlivých elektrických motorů, jsme zjistili, že pro tuto aplikaci je nejvhodnější stejnosměrný motor zapojený do zpětné vazby se snímači polohy motoru. Díky svým vysokým otáčkám a zpětné vazbě dosahuje vyšších rychlostí, rychlejších reakcí a vysoké přesnosti natočení hřídele. Tyto vlastnosti jsou klíčové pro správné fungování manipulátoru a jedině s nimi dokážeme navrhnout manipulátor pro řízení autonomního pohybu hráčů, který bude představovat silného soupeře i pro ty nejlepší hráče stolního fotbalu.

BLDC, tedy bezkartáčové stejnosměrné motory, které jsme vybrali jsou od firmy Trinamic. Typové označení motoru je $PD42-x-1670$, kde x se odvíjí od velikosti a síly motoru. My použijeme $PD42-1-1670$ pro řízení rotačního pohybu hráčů a $PD42-3-1670$ pro řízení lineárního pohybu. Výběr je odůvodněn tím, že na rotační pohyb potřebujeme hlavně vyšší rychlost a nižší hmotnost kvůli dynamice celého pohybového ústrojí a na lineární pohyb potřebujeme dostatečně rychlý a silný motor, aby byl schopen hýbat motorem pro rotační pohyb a překonal tření, které se v aplikaci nachází. Komunikace s



Obrázek 2.2: Schéma subsystému pohonů

motory je zajištěna přes protokol *CANopen* a porovnání těchto dvou motorů můžeme vidět v tabulce 2.1. Bližší informace o řízení pohonů pak najdeme v kapitole 2.4.

Výpočet síly pohonů

Začneme ověřením velikosti síly u motoru, který zajišťuje lineární pohyb hráčů. Průměr ozubeného kola připevněného na hřídel motoru je 37,69mm, tedy jeho poloměr je 18,845mm. Zaokrouhlíme číslo nahoru na 2cm. Sílu motoru na rameni dlouhém 2cm získáme vydělením točivého momentu dvěma, čímž dostaneme hodnotu 9,25N. Z Newtonova zákona zní, že hmota o váze 1 kg poblíž zemského povrchu má tíhu zhruba 9,81 N. Když k těmto skutečnostem připojíme, že váha motoru, který zajišťuje rotační pohyb a je nutné jím tímto motorem pohybovat, je 287g, vyplyne z toho závěr, že máme zhruba 700g rezervu na váhu tyče, hráčů na tyči a kompenzaci tření. Protože lze motor krátkodobě přetížít, je rezerva točivého momentu motoru pro tuto aplikaci více než dostatečná.

Zbývá ověřit motor použitý pro rotační pohyb, který má točivý moment o velikosti 6,25Ncm. Vzdálenost středu místa styku hráče s míčem od osy tyče je maximálně 8cm. Tím pádem je síla v místě styku hráče s míčem 0,78N. Tento motor musí překonat pouze váhu míčku, který váží okolo 20g. Rezerva je zde znovu dostatečná uvažíme-li navíc fakt, že tento motor se bude většinu

času snažit o jemné pohyby kvůli přesné ovladatelnosti pohybu míčku a při střele dojde nejdříve k roztočení motoru prakticky bez zátěže.

	PD42-1-1670	PD42-3-1670
Příruba	42 mm	
Napětí	10-28 V	
Točivý moment	6,25 Ncm	18,5 Ncm
Fázový proud	1,78 A	5,14 A
Jmenovitá rychlost	4000 ot./min.	
Váha	287 g	642 g

Tabulka 2.1: Parametry použitých pohonů [4]



Obrázek 2.3: Pohon PD42-3-1670 použitý v manipulátoru [4]

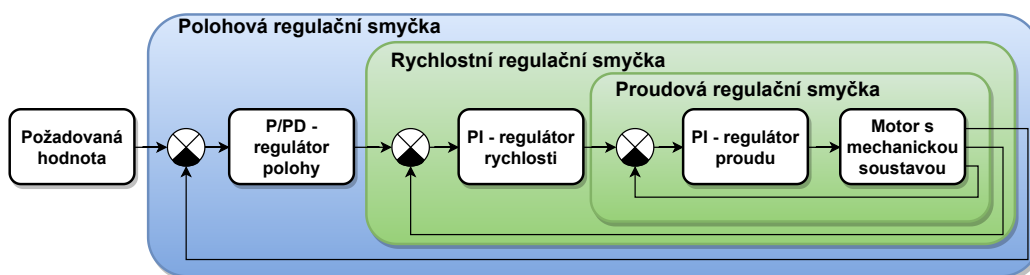
Nadřazené řízení

Obecně lze říci, že řízení bude dostávat pokyny ve formě požadovaných hodnot od neuronové sítě, která se stará o plánování pohybu hráčů a bude dávat rozkazy motorům aby vykonaly onen naplánovaný pohyb. Toto řízení bude realizováno v průmyslovém PC používaném pro automatizaci, na kterém bude spuštěn řídicí systém zvaný *REXYGEN*. Tento pokročilý řídicí systém umožňuje navrhovat a simulovat regulační smyčky, a tak je vhodnou volbou pro tuto práci. Mimo jiné je schopen pracovat se zdrojovými kódy v jazyce *Python*, ve kterém je naprogramovaná a připravená neuronová síť a zároveň Kalmanův filtr, sloužící k predikci pohybu míčku. Později v kapitole

2.6 se dozvíme, že jazyk *Python* sice použijeme, ale na jiném PC místo na řídicím systému *REXYGEN* s ostatními úlohami.

Co se regulační smyčky týče, jedná se o kaskádní zpětnovazebné řízení, které se skládá ze tří smyček. Z vnitřní proudové uzavřené smyčky, po ní následující rychlostní smyčky a obě je obalí vnější polohová smyčka. Jednoduché schéma lze vidět na obrázku 2.4, kde si můžeme všimnout, že polohový regulátor je realizován jako proporcionální, značen zkráceně písmenem P nebo proporcionálně derivační značen PD a rychlostní regulátor spolu s proudovým jako proporcionálně integrační, tedy PI. Z obrázku je také patrné, že vnitřní proudová a rychlostní smyčka je realizována v procesoru, který se nachází přímo na motoru, a v obrázku označena zelenou barvou. Parametry těchto dvou regulátorů nastavujeme komunikací pomocí protokolu *CANopen*, kde nastavujeme proporcionální a integrační zesílení. Poslední polohový regulátor nastavujeme již v *REXYGENu*. Polohová smyčka má nevýhodu v pomalé periodě řízení, která je dána vlivem pomalé komunikace.

Původně byla realizace řídicího algoritmu zamýšlena ještě s realizací všech tří smyček v *REXYGENu*, protože normálně dosahuje komunikace *CAN* vysokých rychlostí a kontrolu nad celým řízením bychom měli kompletně ve svých rukou. Výrobce pohonů sice uvádí možnost řízení momentu ve svých prospektech, nicméně pro tuto konfiguraci motoru, s komunikací pomocí protokolu *CANopen*, nelze posílat do měniče motoru požadované hodnoty momentu, a tak jsme byli nuceni přejít na záložní řešení, o kterém jsme se zmiňovali dříve.



Obrázek 2.4: Schéma kaskádní regulační smyčky

Dobrá zpráva je, že motor umožňuje alespoň číst hodnotu aktuálního momentu. Tato vlastnost je důležitá ve chvílích, kde by se míček zasekl mezi hráčem a hrací plochou. V tento moment by téměř jistě došlo k poškození některé z částí projektu, z největší pravděpodobnosti ke zničení hráče. K chybovým stavům tohoto typu se však dá přijít mnoha způsoby. Dalším takovým způsobem může být zjištění od aktuální odchylky polohové smyčky.

Informace, která by ještě měla zaznít ohledně návrhu nadřazeného řízení je, že bude kladen důraz na agresivní nastavení regulátoru a zároveň na provedení polohové regulace na požadovanou hodnotu bez překmitnutí. Tento přístup vede k rychlým a přesným odezvám, které jsou nutné pro zajištění dostatečně kvalitního protihráče.

Počítačové vidění pro sledování míčku a protihráčů

První zmínka o tomto tématu byla již v podkapitole 1.3, kde jsme si obecně přiblížili, jaké vznikají problémy při řešení počítačového vidění pro následné sledování míčku a protihráčů. Pokud přejdeme rovnou k prvnímu problému, který se týkal rychlosti zpracování obrazu a výběrem nejlepšího poměru mezi rychlostí a kvalitou obrazu, odpovědí bude, že jsme se zde podřídili rychlosti zpracování. Pro srovnání sil počítače a hráče stolního fotbalu, zvolili jsme rychlost snímků kamery 60fps. Tato volba je podmíněná skutečností, že rychlost snímkové frekvence očí u člověka nabývá hodnot právě okolo 60Hz. Tato hodnota je hojně používaná například u monitorů, které tuto frekvenci také používají.

Rozlišení obrazu jsme zvolili 800 na 600, protože dokáže zachytit celou plochu hracího pole a zároveň má přijatelnou kvalitu pro určení souřadnic sledovaných objektů a zachování výše zmíněné vzorkovací frekvence.

Další potíž, na kterou jsme předtím narazili se vztahovala ke světelným podmínkám. Jak bylo výše zmíněno, tak jsme se snažili připravit podmínky prostředí, které jsou shodné přibližně s denním světlem bez přímého svitu slunce. Kvůli snížení nároků na různé hodnoty zdrojů napětí jsme upustili od pomocného přísvitu přidaným zdrojem světla. Tento zdroj jsme vynechali také proto, že pozadí hřiště, které tvoří zelená barva, je z lesklého materiálu, a tak by přísvit vyžadoval dodatečné konstrukční úpravy a použití clonících prvků, které by nezpůsobovali nechtěné stíny vzniklé umístěním světelného zdroje.

Barva hráčů z jednoho týmu je růžová a druhého žlutá. Tato volba vzešla od testování možných barev, které jsme dostali na výběr. Zvolené barvy měly, v porovnání s ostatními barvami, nejlepší výsledky v oblasti segmentace obrazu od zeleného pozadí. Segmentace probíhala v různých světelných podmínkách pro zajištění funkčnosti celé aplikace i v případě, že tyto podmínky nebudou úplně ideální.

Neuronová síť pro plánování pohybu hráčů

Stejně jako počítačové vidění, tak i tato část byla nastíněna výše v podkapitole 1.3, tedy formulace problémů při návrhu manipulátorů. Neuronových sítí respektive strategií, které lze pomocí neuronových sítí aplikovat na daný problém je nepřeberné množství. Naše neuronová síť použitá pro plánování pohybu hráčů využívá posilované učení, známé spíše pod anglickým názvem *reinforcement learning*. Jedná se o jeden z mnoha druhů učení, které je specifické tím, že na rozdíl od učení s učitelem nemáme dostupný soubor statických dat, ale data získáváme sekvenčně ze zpracování obrazu. Tyto data můžeme pomocí daných akcí, v našem případě posunem a rotací, částečně ovlivňovat. Na základě odměn z provedených akcí pak posilované učení určí, jestli se mu daří dobře nebo ne a zda daná akce byla správná. Z této neuronové sítě poté získáváme požadované hodnoty do regulačních smyček, které se starají o pohyb hráčů.

Více informací o tomto problému se však dozvíme v kolegovo diplomové práci [3], která s touto diplomovou prací utváří společný projekt autonomního stolního fotbalu.

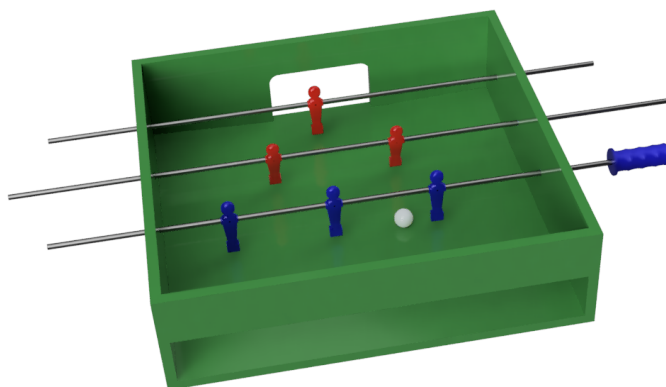
2.2 Návrh mechanické části

Konstrukce celku je složena z pomyslných tří částí. Jmenovitě to jsou herní stůl jádro celé konstrukce, kamera s držákem a manipulátor, který se stará o pohyb hráčů. Následně si jednotlivé části podrobněji rozebereme.

Herní stůl

První část tvoří herní stůl s hráči, který slouží ke hře stolního fotbalu. Stůl je modifikován tak, že se jedná přesně o jednu třetinu oficiálního stolu na ligové zápasy, tedy všechny rozměry jsou totožné s rozměry oficiálního herního stolu až na jeho délku. Tato třetina se skládá z obránců, gólmana a útočníků. Obránci a gólmani jsou ovládáni pomocí manipulátorů a útočníky ovládá člověk, který se bude snažit nad počítačem zvítězit. Co se materiálů týče, tak aréna je složená z lamino desek a herní plocha je potažená zelenou tapetou, aby stůl odpovídal i svými barevnými kombinacemi. Barva hráčů je zvolena tak, abychom si zjednodušili segmentaci obrazu. Tyče pro gólmana, obránce i útočníky jsou z hliníku pro jeho nízkou hmotnost, ale tyč pro útočníky by byla lepší z nerezové oceli, protože hliníková tyč se příliš prohýbá kvůli

nepřesnému pohybu zápěstí. Obrázek 2.5 ukazuje herní stůl, který je v práci nazýván také jako aréna.



Obrázek 2.5: Herní stůl

Kamera

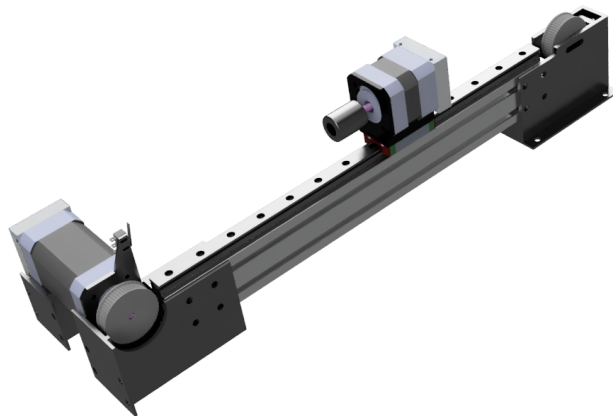
Druhou část tvoří stojan s kamerou, který je pevně přichycen k aréně. Díky zkušenostem z předchozí práce [2], kde jsme polohu kamery potřebovali ještě různě nastavovat, je nyní stojan s kamerou výškově a zároveň v jednom směru nastavitelný. Abychom zrychlili výpočetní čas potřebný pro zpracování obrazu, má stojan fixní pozici. Zrychlení spočívá v absenci hledání referenčních os hřiště, což vede ke následnému zkrácení doby zpracování jednotlivých snímků.

Manipulátor

Třetí a tedy poslední část celé konstrukce představují manipulátory, které se starají o plnohodnotný pohyb hráčů. Jedná se zejména o pohyb ruky, který představuje horizontální posun a pohyb zápěstí způsobující rotaci hráčů. V terminologii mechatroniky se jedná o robot s dvěma stupni volnosti, které zajišťují rotační a prizmatický kloub. Výsledný kinematický řetězec manipulátoru je tak PR. V předchozí práci [2] byl hráč schopen pouze horizontálního posunu, což z úlohy vytvořilo pouze pohyblivou překážku, která bránila proniknutí míče do brány. Nyní, když už hráči mohou vykonávat oba pohyby,

jsou tak schopni vzájemné kooperace v podobě nahrávky či střelby na bránu. Další věc, kterou jsme v porovnání s bakalářskou prací výrazně vylepšili, bylo vymodelování napínacího mechanismu pro řemen tak, abychom mohli pohodlně, rychle a relativně přesně napínat řemen. Tento mechanismus je také vhodný například pro výměnu řemenu za nový kus či dopínat řemen, když se vlivem používání opotřebí.

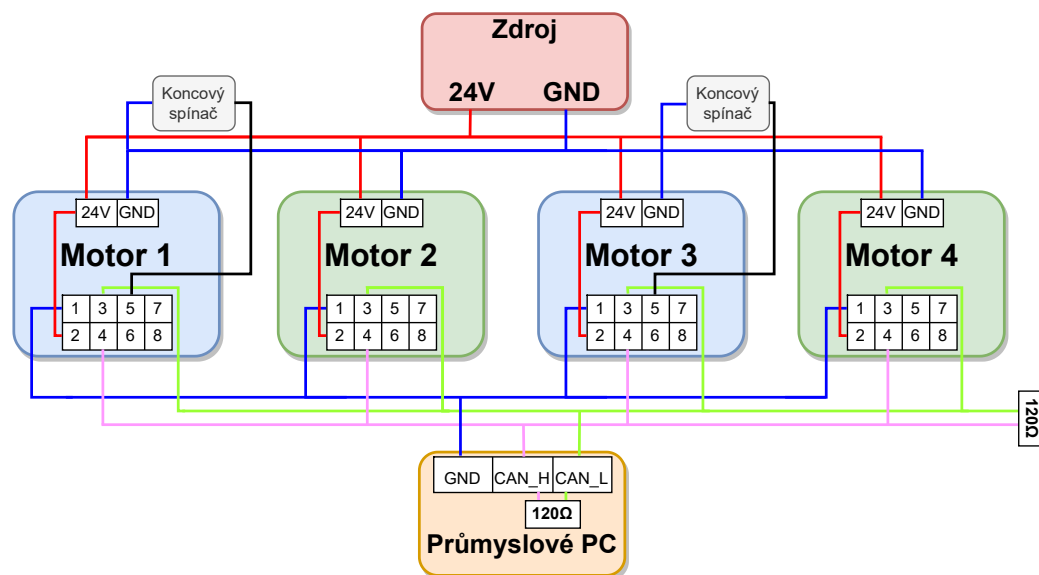
Manipulátor je navržen tak, aby byl modulární. To znamená že chceme, aby se tento návrh dal použít pro všechny typy hráčů od gólmána po útočníky bez nutnosti dalších návrhů. Vždy tak stačí použít rozdílnou délku řemenu, hliníkového profilu a příslušného lineárního ložiska, která vyhoví požadavkům jednotlivých hráčů na rozsah pohybu. Při použití stejných pohonů tak není třeba dál upravovat namodelované součástky a můžeme dále rozšířit stávající úlohu na kompletní řešení. To je myšleno tak, že místo stávající arény použijeme oficiální herní stůl při kterém budou použity další dva manipulátory, které se postarají o pohyb útočníků a záložníků. Při použití rozdílných motorů je nutné upravit upevnění motorů k navrženým součástkám. Stávající upevnění sice svými rozměry připomíná standard pro krokové motory NEMA17, ale bohužel jsou zde díry určené pro přišroubování blíže k ose motoru.



Obrázek 2.6: Modulární manipulátor

Autodesk Fusion 360

Kompletně celý návrh proběhl právě v CAD aplikaci jménem *Fusion 360*, což je produkt od společnosti *Autodesk*. Tento systém nabízí volnou licenci pro nadšence do modelování a 3D tisku, rozšířenou studenskou verzi, která pokročilým uživatelům dovolí provádět různé simulace a samozřejmě nabízí také plnou, komerční verzi, kde můžeme, ke všemu zmíněnému, navrhovat ještě třeba PCB desky. Tento nástroj je tedy možný použít k návrhu mechanické i elektrické části různých projektů. My, právě díky tomuto softwaru, vděčíme za to, že nebylo nutné vyrábět či kupovat žádnou součástku dvakrát, protože navržené díly jsme si mohli přesně nakreslit a předejít tak problémům při montáži a nasimulovat jejich pohyb, kde jsme zkoumali, jestli se nenachází nějaká překážka v trase pohybu. Všechny vymodelované části, které se nedají koupit byly vytištěny pomocí 3D tiskárny. Díly, zejména kvůli jejich složitosti, jsou vytištěné z materiálu PLA (kyselina polymléčná) pro jeho jednoduchost tisku. Protože žádná ze součástí není tepelně příliš namáhaná tak, aby došlo k deformaci a následné nefunkčnosti celku, nebylo nutné použít jiný materiál. Pokud by však nastalo, že nějaká část nezvládne mechanické nebo tepelné namáhání, je možné použít tepelně či mechanicky odolnější materiál.



Obrázek 2.7: Elektrické schéma zapojení

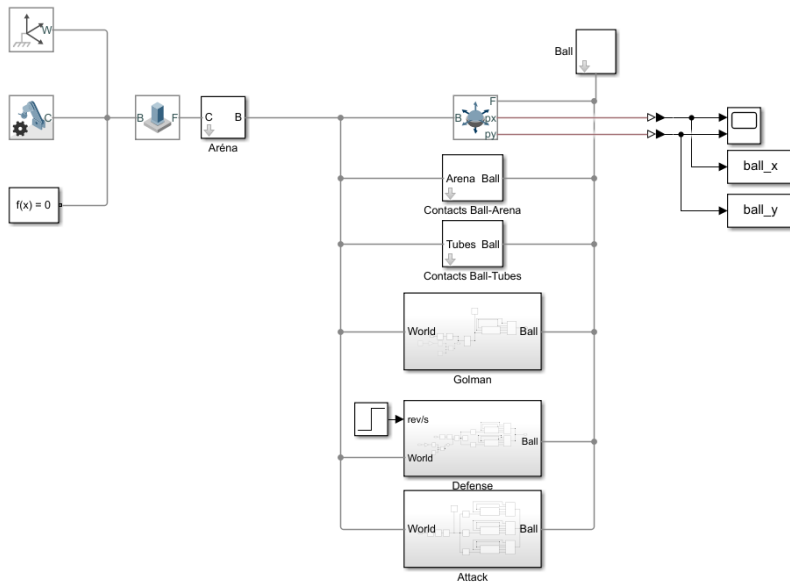
2.3 Virtuální model v prostředí Simscape

Nejen pro tvorbu regulačních smyček je potřeba mít model. Neuronová síť použitá v práci lze učit na reálném systému nebo na softwarovém modelu. Pokud bychom síť trénovali na reálné aplikaci, trvalo by to nepřiměřené množství času a výsledek by nemusel být uspokojující. Proto je třeba vytvořit jakési virtuální dvojče reálného stolního fotbalu se zahrnutím veškerých fyzikálních zákonů, na kterém se bude učit neuronová síť. Toto dvojče jsme vytvořili v platformě *MATLAB*, jehož součástí je prostředí, kterému se říká *Simulink*. *Simulink* se používá k návrhu a simulaci modelů před tím, než se přesunou do hardwarové části. Práce s ním probíhá pomocí spojování jednotlivých bloků, které plní určitou funkci. Mimo jiné v něm lze používat různé doplňky a ten, který použijeme my se nazývá *Simscape*. Tento doplněk umožňuje rychlou tvorbu modelů fyzikálních systémů.

Prostředí Simscape

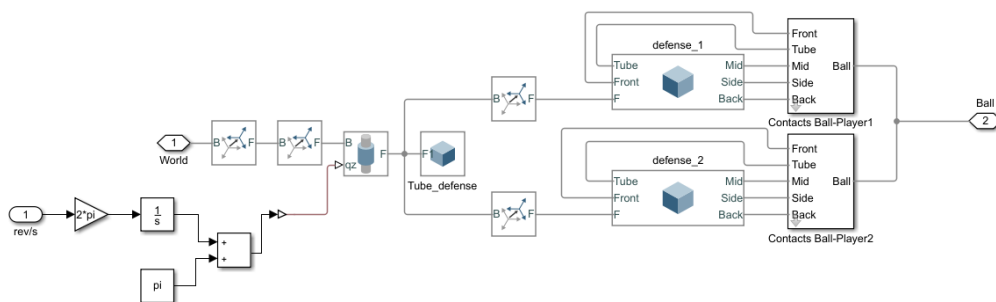
Jak už bylo řečeno výše, všechny modely představující fyzické součástky jsme navrhli a vytvořili v CAD systému. Prostředí *Simscape* umožňuje tyto modely použít jako pevné těleso a dále s ním pracovat. Tyto tělesa jsou v prostoru umístěna na základě pozice od tak zvaně světové souřadnice. U pohyblivých součástí je nutné zdefinovat počet stupňů volnosti, což provádíme pomocí kloubů. Hráče vůči aréně spojuje cylindrický kloub, který odpovídá pohybu hráčů ve stolním fotbale, konkrétně posuvnému a rotačnímu. Míči se přiřadí kloub představující všech šest stupňů volnosti.

Dále je nutné použít externí knihovnu s názvem *Simscape Multibody Contact Force Library*, která je volně dostupná ke stažení na internetu. Pokud bychom tuto knihovnu nepoužili, mezi tělesy by neprobíhaly žádné interakce, takže by například procházela skrz sebe, což je v reálném světě nemožné. Knihovna má jedno malé omezení, a tím je malé množství tvarů, pro které lze tyto vzájemné síly simulovat. Jedná se o rovinu, válec a kouli. Pro naše účely, je tato vlastnost v pořádku, protože při kontaktu dochází vždy k odrazu míčku od ostatních objektů, které lze poskládat z tvarů zmíněných výše.



Obrázek 2.8: Schéma z prostředí Simscape

Součástí této knihovny je nastavení parametrů vzájemného působení mezi dotčenými objekty. Pro ověření fungování knihovny stačí ponechat výchozí parametry, ale pro vytvoření virtuálního dvojčete, je potřeba aby tyto parametry odpovídali realitě. Zjištění těchto parametrů si podrobněji přiblížíme v další podkapitole.



Obrázek 2.9: Detailní pohled na subsystém obránce v prostředí Simscape

Identifikace reálných parametrů pro vzájemnou interakci

Při identifikaci reálných parametrů musíme vyřešit v zásadě tři situace. První z nich se týká kontaktu míče a hráče. Tuto situaci stačí nasimulovat s jedním hráčem a následně ji použít pro všechny ostatní, protože jsou všichni hráči ze stejného materiálu. Druhá situace je, že se míč kutálí po povrchu arény. Míček samozřejmě při tomto pohybu zpomaluje, což je dáno třením. Pro tření musíme získat dvě konstanty a to tu, která ovlivňuje statické tření a druhou, která ovlivňuje kinetické tření. Velikost statického tření určuje, jak velkou sílu musíme použít, aby se míček lehce pohnul z místa. Kinetické tření působí v momentě, kdy se míček pohybuje a s vyšší rychlostí je toto tření také vyšší. Poslední situace řeší odraz míčku od stěny. Jedná se o podobnou situaci jako byla první. Hráče zde nahrazuje stěna, která nevykazuje žádný pohyb a na rozdíl od hráče má tvrdší materiál, takže konstanty pro tuhost pružiny a tlumení v místě nárazu zde budou jiné.

Celý experiment jsme prováděli tak, že jsme míč postavili před hráče, vždy na stejnou pozici. Následně jsme hráče postavili vzhůru nohama a začali otáčet vždy konstantní rychlostí jedna, dva a tři otáčky za sekundu. Hráč vždy vystřelil míček proti zdi, takže jsme všechny tři situace, které jsme zmínili dříve, pokryli.

V kapitole 3 můžeme pozorovat, k jakým výsledkům jsme dospěli, při rychlosti otáčení hráčů tři otáčky za sekundu. Parametry, které jsou zde použity, jsme hledali expertní metodou, kde hlavní kritérium volby byl jejich fyzikální význam. Po pár iteracích jsme tyto parametry zvolili jako ty, které dostatečně odpovídají realitě. Pokud bychom chtěli dosáhnout co nejpřesnější shody, mohli bychom použít vhodnou numerickou gradientní metodu, která by pro každou výše zmíněnou situaci otestovala parametry a na základě nich by při dalších iteracích volila parametry, které by ji dovedli právě k co nejpřesnější shodě. Bohužel je tato simulace velice časově náročná, a proto nebylo možné ji použít.

Dále, protože se jedná pouze o model a použitá knihovna neumí spočítat interakci mezi námi namodelovanými objekty, dochází zde k různým krajním stavům, které absolutně neodpovídají realitě. Například pokud se míček střetne s hráčem přímo v rohu jeho spodní části, dojde k vyvinutí nepřiměřeně veliké síly na míček, který se následně odrazí nereálnou rychlostí. Toto je způsobeno tím, že model hráče je rozkouskovaný na základní tvary abychom mohli tuto knihovnu použít. V rohu hráče se potkají dvě plochy, které jsou na sebe kolmé a v momentě, kdy se zde míč narazí dojde k odrazu od obou rohů ploch a tudíž k nesprávnému výpočtu reakční síly.

2.4 Řízení pohonů

Vlastnosti motorů

Jak již bylo zmíněno výše, regulátor rychlosti a momentu probíhá v měniči, který je umístěn přímo na motoru. Oba dva regulátory jsou typu PI a předpokládáme, že vnitřní proudová smyčka je rychlejší než vnější rychlostní smyčka. Protože však v dokumentaci motorů chybí podrobný popis, je potřeba vložit důvěru do výrobce, že vše implementoval správně.

Všechny čtyři parametry, které je třeba nastavit, představovali zesílení, tedy proporcionální K_p a integrační K_i zesílení zvlášť pro proudovou a rychlostní smyčku. Všechny tyto parametry se dají nastavit v rozsahu 0 až 65535, což představuje velikost dvou bytů. Nejprve jsme si mysleli, že se jedná o datový typ *short* s plovoucí řádovou čárkou, ale toto tvrzení bylo rychle vyloučeno, protože při nastavování desetinných čísel následně převedených na zmíněný datový typ neodpovídalo zvýšení či snížení hodnoty zesílení. Tímto pokusem jsme si ověřili, že se ve měniči nachází procesor, který neobsahuje koprocesor, který by urychlil výpočty čísel s pohyblivou desetinnou čárkou. Znamená to tedy, že rozsah hodnot, které máme možnost měnit, je následně přeškálován na námi neznámý rozsah a celá regulace v procesoru probíhá v celých číslech.

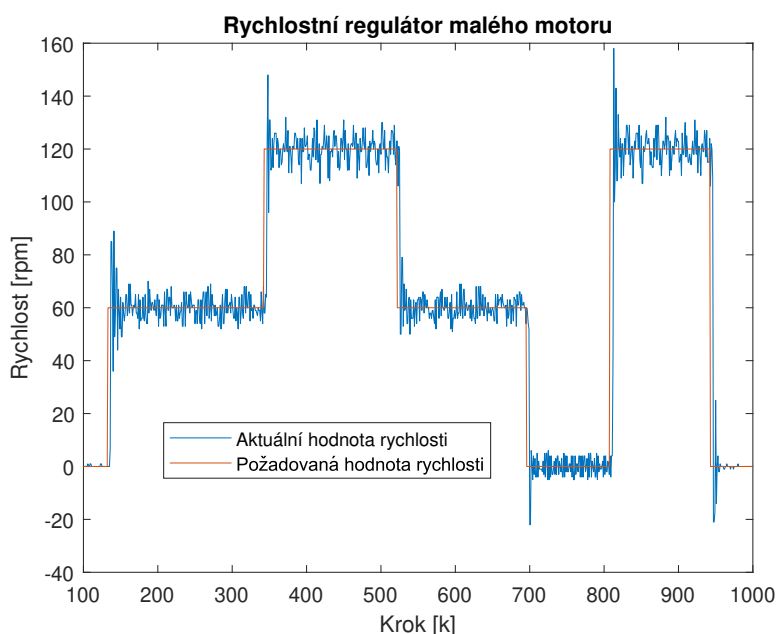
Další omezení, které motor poskytuje se týká nepříjemného hluku, který vzniká volbou vysokého proporcionálního zesílení. Při volbě hodnoty zesílení vyšší než 10 000 u větších motorů zajišťujících horizontální pohyb a 4 000 u menších motorů starajících se o rotaci, se zvýší vysokofrekvenční šum. Tento šum způsobí nepatrné vibrace motoru, který tyto vibrace přenáší na celou soustavu, která bohužel už při takto nízkých hodnotách rezonuje a vytváří zmíněný zvuk.

Proudová smyčka

Protože máme možnost nastavovat v motoru pouze požadovanou rychlost a protože je proudová smyčka rychlejší než smyčka rychlostní, nemáme moc možností, jak správně naladit tuto proudovou smyčku. Navíc by nastavování této smyčky probíhalo pomocí řídicího systému *REXYGEN*, kde komunikace probíhá po sběrnici *CAN* s periodou 40ms, takže pokud bychom udělali jakoukoliv změnu v nastavení konstant, rozdíl bychom si ani nevšimli. Z tohoto důvodu jsme konstanty zesílení proudového regulátoru PI nechali beze změny.

Rychlostní smyčka

Rychlostní smyčku už bylo možné nastavovat a změny, při ladění konstant, byly vidět. Tuto smyčku jsme nastavovali zvláště pro velké a malé motory, protože vlivem omezených možností soustavy bylo nutné přistupovat rozdílně. Na obrázku 2.10 níže, je možné vidět dynamiku rychlostní smyčky po naladění. Jedná se o malé motory, které se starají o rotaci hráčů, kde není žádné omezení pohybu, tím pádem bylo snazší nalézt parametry regulátoru. Jediné omezení, které jsme zmínili dříve, zůstává maximální hodnota proporcionálního zesílení K_p , abychom nebudili zmíněný vysokofrekvenční šum.



Obrázek 2.10: Regulace rychlosti rotačního pohybu

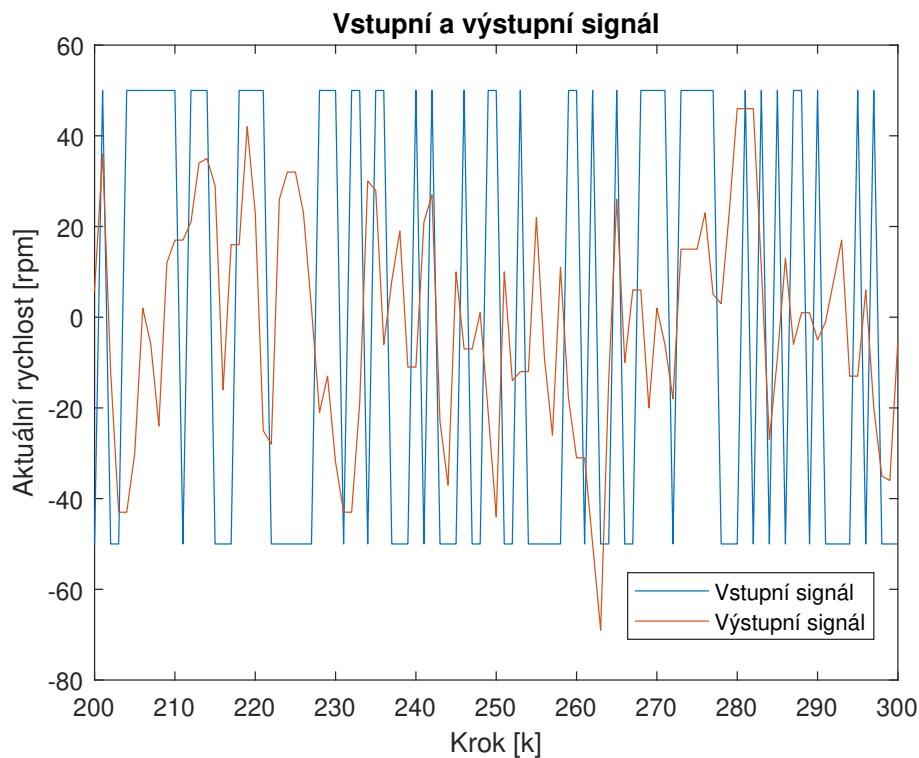
U větších motorů, které zřizují horizontální pohyb hráčů, vstupuje do ladění ještě omezení dané mechanikou soustavou. Protože je nutné regulátor nastavovat už na této soustavě, je nutné vytvořit signál, na kterém bude následně možné nalézt správné parametry regulátoru. Tento signál můžeme vidět na obrázku 2.11, kde je vidět i výsledná odezva motoru na tento signál.

Jsou zde ještě dvě drobnosti, které lze z grafů vyzorovat. První, možná na první pohled ne tak zřejmá drobnost, se týká dopravního zpoždění. Toto dopravní zpoždění trvá 120ms, což odpovídá třem periodám. Jednu periodu

Identifikace

Identifikaci jednotlivých motorů pro pozdější návrh regulátorů jsme prováděli změnou vstupní rychlosti a v závislosti k ní jsme odečítali aktuální rychlost motoru. Na vstup jsme přiváděli rozmítaný pseudonáhodný binární signál, který se po celý experiment snaží udržet nulovou střední hodnotou, což se dokonale hodí pro náš systém, kde u motoru ovládacího horizontální posun hráče je tato vlastnost dokonce nutná pro zamezení nesprávných výsledků experimentu nebo dokonce poškození konstrukce. Přiblížený vstupně výstupní signál získaný z experimentu můžeme vidět na obrázku 2.12.

Tyto data jsme následně vložili do sady nástrojů pro identifikaci systému v Matlabu. Při použití různých strategií pro identifikaci nakonec nejlépe vycházela shoda lehce přes 54% s třetím řádem stavové reprezentace systému. Tato shoda byla nalezená pomocí metody chyby predikce známé pod zkratkou PEM a její hodnota vycházela pro všechny motory stejně.



Obrázek 2.12: Data získaná experimentem a použita pro identifikaci

Parametry systému pro velký a malý motor vyšly rozdílně a jejich parametry jsou pro větší motor

$$A = \begin{bmatrix} 0.5518 & 0.5366 & -0.1079 \\ -0.6927 & -0.0530 & -0.4315 \\ -0.1901 & 0.4836 & -0.0837 \end{bmatrix}, B = 10^{-3} \cdot \begin{bmatrix} -0.3892 \\ -0.3546 \\ 0.1324 \end{bmatrix}, \quad (2.1)$$

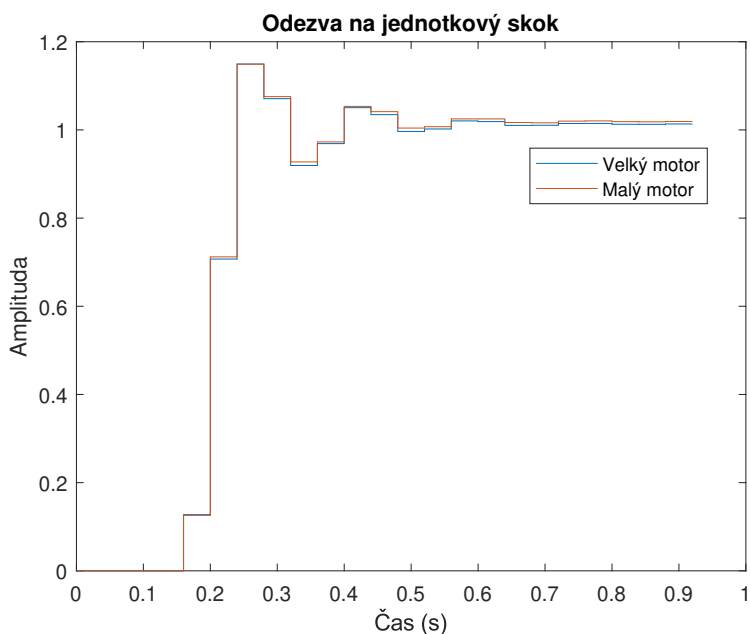
$$C = [-995.6374 \quad 859.7302 \quad 337.2928]$$

a parametry pro menší motor jsou

$$A = \begin{bmatrix} 0.5623 & 0.5350 & -0.1077 \\ -0.6914 & -0.0433 & -0.4297 \\ -0.1899 & 0.4820 & -0.0749 \end{bmatrix}, B = 10^{-3} \cdot \begin{bmatrix} -0.3900 \\ -0.3458 \\ 0.1030 \end{bmatrix}, \quad (2.2)$$

$$C = [-995.6374 \quad 859.7302 \quad 337.2928].$$

Na obrázku 2.13 můžeme vidět přechodový děj obou motorů, který je pro oba motory téměř shodný, což jsme schopni říci i z podobnosti matic popisující stavovou reprezentaci. Identifikované systémy mají taktéž dopravní zpoždění 120ms, o kterém jsme se již bavili a bylo zadáno jako parametr při identifikaci.

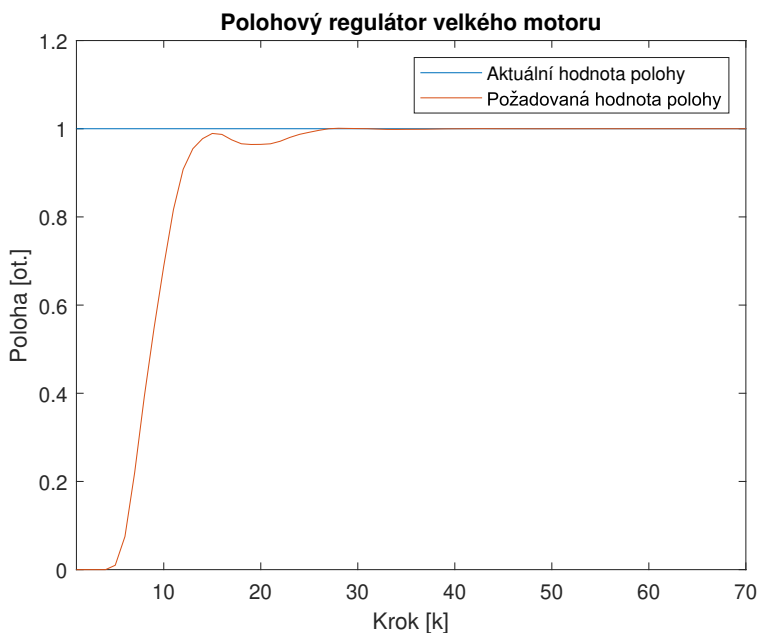


Obrázek 2.13: Odezva na jednotkový skok identifikovaných systémů

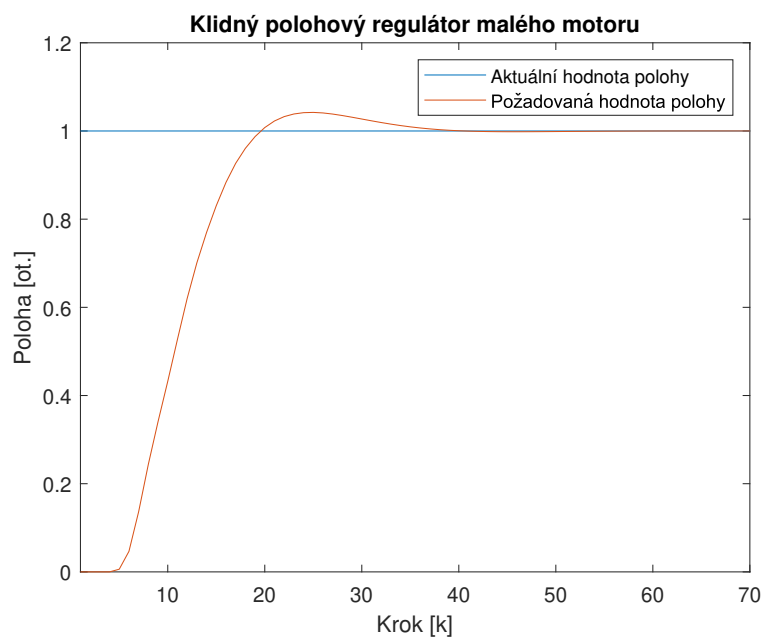
Návrh polohových regulátorů v simulaci

Protože se jedná o kaskádní regulátor, kde vnitřní smyčky obsahují integrátor zajišťující nulovou regulační odchylku, je zvykem navrhovat polohový regulátor čistě proporcionální. To jsme také udělali a nejprve navrhli P regulátor. K nastavení P regulátoru není zapotřebí příliš zkušeností, protože měníme jeden parametr a protože se nám doba náběhu, po návrhu proporcionálního regulátoru, zdála příliš dlouhá, přidali jsme derivační složku do záporné zpětné vazby od rychlosti, kterou nám poskytuje motor.

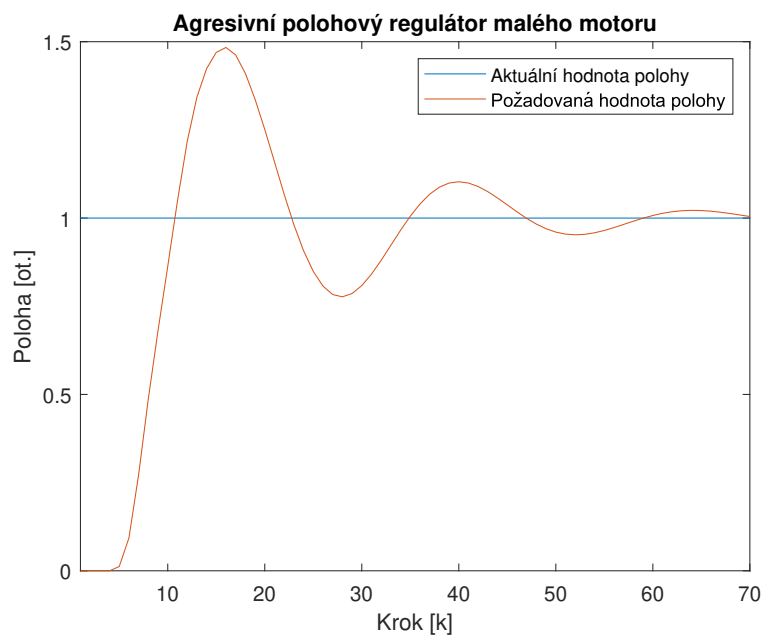
Parametr proporcionálního zesílení klidného polohového regulátoru u malého motoru vyšel roven $K_p = 145$. Ten samý parametr ale používaný pro střelbu vyšel $K_p = 290$. U polohového regulátoru velkého motoru, kde jsme použili dvě složky, proporcionální a derivační, vyšlo proporcionální zesílení $K_p = 220$ a derivační $K_d = 18$. Na obrázcích 2.14, 2.15 a 2.16 můžeme pozorovat přechodové děje při použití zmíněných regulátorů v simulaci.



Obrázek 2.14: Přechodový děj horizontálního pohybu



Obrázek 2.15: Přejchodový děj rotačního pohybu při použití klidného regulátoru



Obrázek 2.16: Přejchodový děj rotačního pohybu při použití agresivního regulátoru

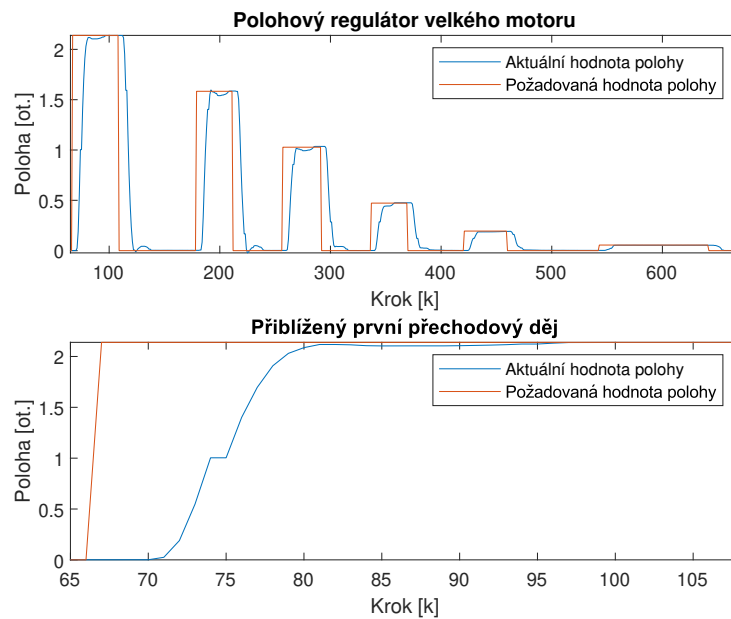
Návrh polohových regulátorů v reálném prostředí

Zde jsme postupovali stejně jako u simulace, kde jsme nejprve zkusili použít parametry regulátorů, které jsme při ní získali. Přechodové děje, s parametry použitými v simulacích, byly překvapivě dost podobné těm ze simulací. I když při identifikaci vyšla shoda systému lehce přes 50%, reálný systém se ve frekvenčním spektru, ve kterých jej používáme, shodoval s modelem. Proto i při nízké shodě vycházejí parametry podobně.

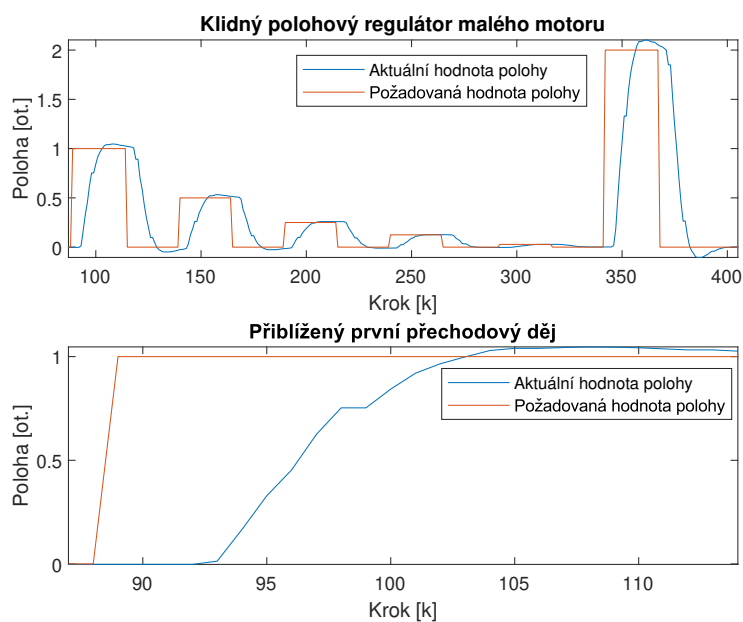
Původně jsme zamýšleli použití PD regulátoru také pro rotační motory, ale po naladění parametrů jsme zkusili vystřelit. Míček zde představuje vstupní poruchu na řízený systém, se kterým se nedokázal vypořádat, protože po střele se motor dostal na mez stability a začal netlumeně kmitat. Proto nakonec používáme tři druhy regulátorů, kde PD regulátor slouží pro horizontální posun velkým motorům, a oba P regulátory slouží pro rotační pohyb malým motorům, kde je podle aktuální situace přepínáme.

Kvůli detekci zaseknutí hráčů při kolizi s míčem jsme zvažovali použití bloku, který omezuje skokové změny systému známý jako *rate-limiter*. Pomocí tohoto bloku jsme schopni snížit velikost polohové odchylky při regulaci. Této vlastnosti jsme chtěli využít tak, že při kolizi vznikne polohová odchylka, která bude větší než ta, které dosáhneme při použití zmíněného bloku. Bohužel jsme polohovou odchylku nedokázali dostatečně snížit, aniž bychom příliš neovlivnili dobu náběhu, a tak jsme kvůli stanoveným požadavkům na prioritu co nejkratší doby regulace a jednoduchosti tento blok vůbec nepoužili. Detekci kolize stejně realizujeme pomocí polohové odchylky, avšak s pomocí statistiky jsme schopni z deseti po sobě jdoucích hodnot říct, zda se jedná o kolizi či nikoliv. Využili jsme při tom střední hodnoty a variance, kde kolize nastane v momentě, kdy je střední hodnota je vyšší, než námi stanovená mez, kvůli falešným alarmům při doregulování na požadovanou hodnotu a zároveň variance je nižší, než námi stanovená hodnota, kvůli skoro konstantní hodnotě polohové odchylky, která nastane při kolizi.

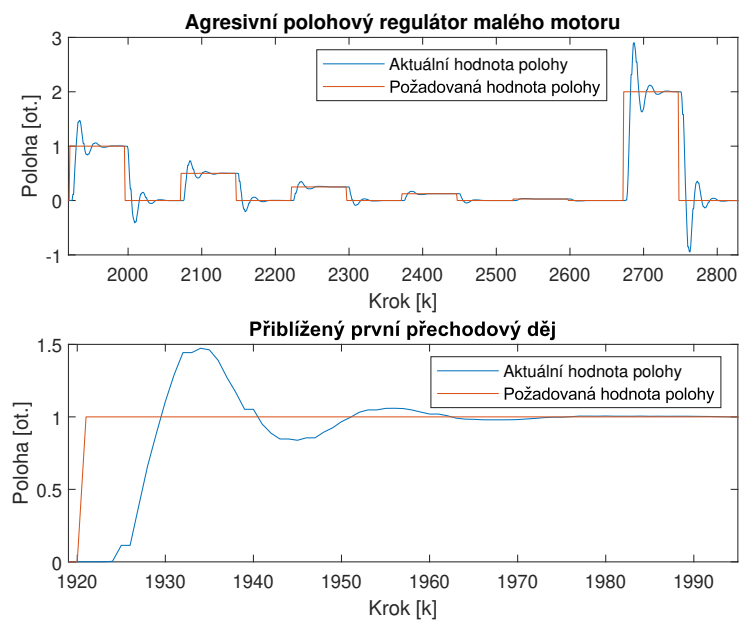
Parametr proporcionálního zesílení klidného polohového regulátoru u malého motoru vyšel roven $K_p = 185$ a pro střelbu vyšel $K_p = 320$. U polohového regulátoru velkého motoru, vyšlo proporcionální zesílení $K_p = 230$ a derivační $K_d = 15$. Na obrázcích 2.17, 2.18 a 2.19 můžeme pozorovat přechodové děje při použití zmíněných regulátorů na reálném systému.



Obrázek 2.17: Přechodový děj horizontálního pohybu



Obrázek 2.18: Přechodový děj rotačního pohybu - klidný regulátor



Obrázek 2.19: Přechodový děj rotačního pohybu - agresivní regulátor

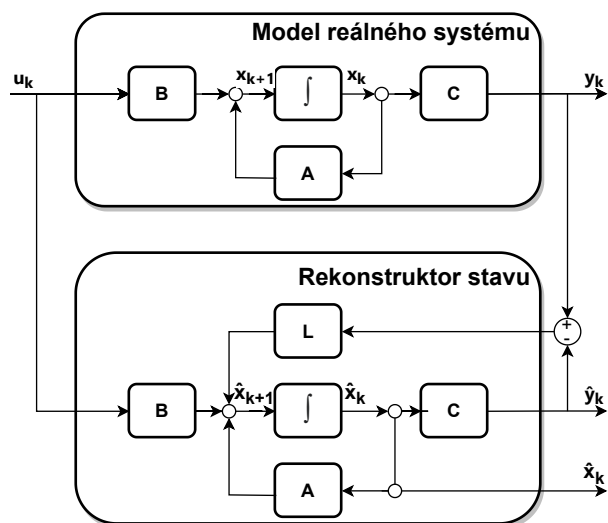
2.5 Algoritmy pro sledování míčku

Algoritmů pro sledování míčku je na světě více než dost. My si v této kapitole představíme všechny, se kterými jsme se setkali během studia. Pro upřesnění uvedeme hned v úvodu, že sledování míčku je úlohou sledování stavů míčku, konkrétně polohu a rychlost. Pokud je systém, který chceme sledovat deterministický, použijeme rekonstruktor stavu. Pokud však do našeho systému vstupuje určitý šum, musíme použít stochastické algoritmy. Za takzvaný svatý grál a jeden z lepších objevů dnešní doby se považuje Kalmanův filtr. Tento algoritmus uvažuje lineární systém do kterého vstupuje stochastický šum s Gaussovo rozdělením. Pokud je však systém, který chceme sledovat, nelineární, tak je třeba zvolit algoritmus, který si s takovým umí poradit. Jeden z nich se nazývá rozšířený Kalmanův filtr (angl. extended Kalman Filter) a další, který si představíme, se nazývá unscenovaný Kalmanův filtr (angl. unscented Kalman filter). První z jmenovaných se oproti druhému vyznačuje zejména nižšími výpočetními nároky, ale jeho správná funkčnost je závislá na nelinearitě systému. Rozšířený Kalmanův filtr je dobré použít pro odhad stavu u pomalu se měnících nelineárních systémů, tedy systémů, které lze dobře aproximovat linearizací.

Rekonstruktor stavu

Jak už název napovídá je rekonstruktor stavu systém, který odhaduje vnitřní stav reálného systému. Používá se v aplikacích, kde vnitřní stav reálného systému nelze změřit přímo. Konkrétní stav tedy odhadujeme na základě znalosti dynamiky reálného systému a jeho měřitelných vstupech a výstupech systému.

Nejlepší představa o tom, jak funguje rekonstruktor stavu, vychází z obrázku 2.20, kde jednoduše vytvoříme jakéhosi "dvojníka" modelu reálného systému. Vstup do rekonstruktora stavu bude stejný jako do reálného systému. Výstup z rekonstruktora stavu odečteme od výstupu z reálného systému a dále násobíme maticí L , přesně jako na obrázku. Pokud rekonstruktor dostatečně odpovídá reálnému systému a tento systém je zároveň pozorovatelný, budou při správně zvolené matici L vstupy a výstupy rekonstruktora konvergovat k vstupům a výstupům reálného systému. Díky shodnému chování je možné z rekonstruktora stavu získat odhadovaný stav reálného systému. Pro zajištění asymptotické stability rekonstruktora je nutná konvergence jeho chyby, tedy $e_k = \hat{x}_k - x_k$. Odtud přejdeme na vztah $e_{k+1} = (A - LC)e_k$, kde pro diskrétní systém je nutné zajistit aby matice $(A - LC)$ měla vlastní čísla



Obrázek 2.20: Luenbergerův rekonstruktor stavu

uvnitř jednotkového kruhu.

Uvažujeme následující diskrétní časově-invariantní lineární systém

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \\ y_k &= Cx_k + Du_k, \end{aligned} \quad (2.3)$$

kde k je krok, ve kterém se systém nachází, x_k je stav systému, u_k je vstup a y_k je výstup systému.

Tento popis systému je rozšířen právě o rekonstruktor stavu, kde jak je uvedeno výše, obsahuje stejný vstup jako reálný systém a používá výstup jak z reálného systému, tak z rekonstruktoru stavu pro zjištění odhadu stavu systému

$$\begin{aligned} \hat{x}_{k+1} &= A\hat{x}_k + L[y_k - \hat{y}_k] + Bu_k, \\ \hat{y}_k &= C\hat{x}_k + Du_k, \end{aligned} \quad (2.4)$$

kde "stříška" nad stavem systému x_k a výstupem systému y_k představuje právě odhad těchto veličin.

Kalmanův filtr

Kalmanův filtr poskytuje odhad stavu lineárních systémů, které jsou pod vlivem šumu. Pokud je tento šum bílý, jedná se o optimální stochastický

rekonstruktor stavu. Používá se v mnoha aplikacích, kde je potřeba navigovat či sledovat nějaký cíl, v úlohách počítačového vidění nebo při zpracování signálů. Například v situaci, kdy máme dostupné měření z více senzorů a chceme dosáhnout co možná nejpřesnějšího odhadu. Jedná se pořád o novinku na poli vědy, protože první aplikace, kde byl Kalmanův filtr použit, byla při vesmírném programu Apollo.

Uvažujeme stejný systém jako u rekonstruktoru stavu s menšími rozdíly ve značení. Navíc je zde vnesen aditivní šum a to jak do rovnice popisující vývoj stavu na prvním řádku, tak do rovnice měření na řádku druhém

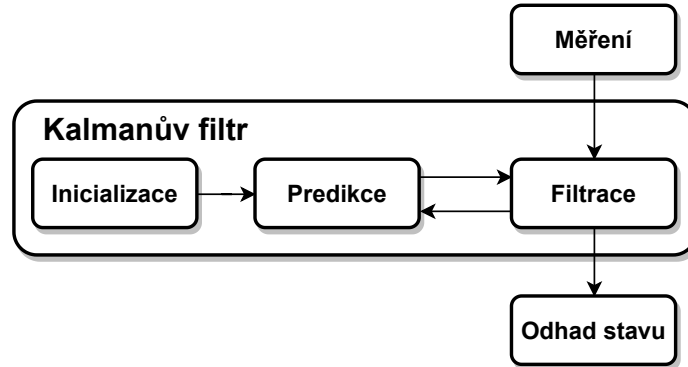
$$\begin{aligned}x_{k+1} &= Fx_k + Bu_k + w_k, \\z_k &= Hx_k + v_k,\end{aligned}\tag{2.5}$$

kde w_k je šum systému a v_k je šum měření. Další proměnné jsou matice dynamiky F , matice řízení B , která je v případě úlohy sledování neznámá a matice měření H .

Šum systému w_k může nahrazovat nepřesnosti v matematickém popisu, neznámé vstupy do systému nebo rozdílnost oproti reálnému systému způsobenou linearizací. Předpokládáme, že tento šum má normální rozdělení s nulovou střední hodnotou, tedy že $w_k \sim \mathcal{N}(0, Q_k)$. U šumu měření v_k předpokládáme stejné rozdělení ($v_k \sim \mathcal{N}(0, R_k)$), které zde bývá způsobeno fyzickým senzorem. Typickým příkladem zašuměného signálu měření je GPS signál.

Volba kovariančních matic šumů je závislá na návrháři a ve většině případů není přesně známá. Pro pochopení zde uvedeme alespoň základní myšlenku volby těchto matic. Pokud zvolíme kovarianční matici $Q = 0$, znamená to, že známe přesně model reálného systému a tudíž se pro přesný odhad obejdeme bez měření. Obráceně pro volbu kovarianční matice $R = 0$ plyne, že měření, které získáme pomocí senzoru je absolutně přesné, a tak není potřeba znát předchozí informaci.

Z obrázku 2.21 je patrné, že Kalmanův filtr pracuje ve dvou krocích. Nejprve určíme počáteční odhad stavu a jeho kovarianci, která představuje důvěru v onen počáteční odhad stavu. Následuje predikce, kde se posuneme o jeden časový krok dopředu viz. rovnice 2.5 popisující vývoj stavu. Kvůli predikci se nám typicky zvětší kovarianční matice odhadu stavu. Následně probíhá filtrace, která díky novému měření sníží kovarianční matici odhadu stavu a zpřesní tak celkový odhad, který právě z této fáze použijeme pro další aplikaci. Postup predikce a filtrace nadále opakujeme.



Obrázek 2.21: Funkční schéma Kalmanova filtru

Rovnice pro výpočet predikce nalezneme níže. Proměnné pro predikci budeme zobrazovat s apostrofem

$$\begin{aligned}\hat{x}'_{k+1} &= F\hat{x}_k + Bu_k, \\ P'_{k+1} &= FP_kF^T + Q_k.\end{aligned}\tag{2.6}$$

Následují rovnice pro výpočet filtrace

$$\begin{aligned}\hat{x}_k &= \hat{x}'_k + K_ke_k, \\ K_k &= P'_kH^T(HP_kH^T + R_k)^{-1}, \\ e_k &= z_k - Hx'_k, \\ P_k &= P'_k - P'_kH^TK_kHP'_k,\end{aligned}\tag{2.7}$$

kde e_k je inovace neboli chyba predikce a K_k je Kalmanův zisk.

Pokud je systém navíc časově invariantní, to znamená, že všechny matice z rovnice 2.5 a zároveň kovarianční matice Q_k a R_k musí být časově invariantní ($Q_k = Q$, $R_k = R$, $F_k = F$, $B_k = B$, $H_k = H$) a pokud navíc dvojice (H, F) je pozorovatelná a dvojice (F, S_Q) je dosažitelná, kde S_Q je rozklad kovarianční matice Q tak, že $Q = S_QS_Q^T$, kovarianční matice chyby odhadu P_k a P'_k spolu s Kalmanovým ziskem K_k se stanou též časově invariantní. Uvážením této skutečnosti můžeme tyto matice pomocí algebraické Riccatiho rovnice vypočítat dopředu a snížit tím výpočetní nároky [1].

Nakonec ještě zmíníme, že Kalmanův filtr lze také použít pro úlohy, kde nám měření ze sensorů nechodí tak často, jak bychom potřebovali. Stačí opakovat krok predikce a hned jak nám přijde nové měření, zpřesníme daný odhad.

Nelineární Kalmanovy filtry

V této podkapitole si popíšeme princip dvou zástupců nelineárních Kalmanových filtrů. Z původně lineární rovnice 2.5 přejde na obecně nelineární tvar znázorněný v rovnici 2.8, tedy

$$\begin{aligned}x_{k+1} &= f(x_k, u_k) + w_k, \\z_k &= h(x_k) + v_k.\end{aligned}\tag{2.8}$$

Každý z níže uvedených filtrů má k této nelinearitě jiný přístup. Hlavní myšlenka rozšířeného Kalmanova filtru spočívá v linearizaci nelineárního modelu v okolí námi zvoleného bodu. Pro linearizovaný model pak použijeme standardní Kalmanův filtr. K linearizaci se používá Taylorův rozvoj prvního řádu, ze kterého lze ve většině případů analyticky získat Jakobián. Pokud není nelineární funkce hladká a tudíž pro některé body neexistuje derivace nebo je výpočet derivace příliš komplikovaný, můžeme použít bez-derivační filtry.

Unscentovaný Kalmanův filtr je zde prezentován jako zástupce bezderivačních filtrů. Oproti rozšířenému nechává unscentovaný Kalmanův filtr nelineární funkci neměnnou ale jeho funkce spočívá v aproximaci Gaussova rozdělení malou množinou bodů okolo střední hodnoty zvaných sigma body. Z těchto bodů a k nim příslušných nelineárních funkcí následně počítáme novou střední hodnotu a kovarianci odhadu stavu. Protože zde provádíme bodovou aproximaci, vyhneme se výpočtům Jakobiánů, které mohou být někdy složité na výpočet nebo dokonce nespočitatelné, jak jsme již zmínili výše.

Hlavní rozdíl rozšířeného i unscentovaného Kalmanova filtru je, že zde oproti základnímu Kalmanovu filtru nelze zaručit konvergenci odhadu stavu. Důležité je mít na paměti, že všechny vypočtené odhady stavu a jejich kovarianční matice jsou vždy pouze aproximativní.

Algoritmus použitý v úloze

Model pohybu míčku vychází z Newtonova pohybového zákona. Pohyb v jedné ose se řídí podle rovnice 2.9, tedy

$$m\ddot{x} = F \Rightarrow \ddot{x} = \frac{F}{m}.\tag{2.9}$$

Zdefinujeme si stav modelu, který je reprezentován rychlostí a polohou v dané ose

$$\begin{aligned}x_1 &= x, \\x_2 &= \dot{x}.\end{aligned}\tag{2.10}$$

Tyto dvě veličiny získáme pomocí dvou integrátorů zapojených v sérii. Přenos pohybu míčku v jedné ose tak představuje přenos popsany rovnicí 2.11, tedy

$$P(s) = \frac{1}{s^2}. \quad (2.11)$$

Jedná se o zjednodušený lineární model, který neobsahuje žádnou složitou dynamiku jako je například tření. Obecně by model popsany maticemi vypadal tak, jako na rovnici 2.5. Protože ale nemáme údaje o matici B , tedy informace o řízení pohybu míčku, necháme tuto matici nulovou. Když shrneme předchozí informace, získáme stavovou reprezentaci, která vypadá následovně

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + w. \quad (2.12)$$

Protože používáme Kalmanův filtr na počítači, je nutné převést tyto matice na diskretní tvar. To provedeme v Matlabu pomocí příkazu `c2d`, kde si jako parametr zvolíme Tustinovu diskretizaci. Navíc, po rozšíření modelu zahrnující obě osy, vypadají získané rovnice následovně

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \\ y_{1,k+1} \\ y_{2,k+1} \end{bmatrix} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \\ y_{1,k} \\ y_{2,k} \end{bmatrix} + w, \quad (2.13)$$

kde T je perioda vzorkování, $x_{1,k+1}$ je stav x_1 v časovém okamžiku $k+1$ a $x_{1,k}$ je předchozí stav. Proměnná w představuje Gaussův šum, který nahrazuje nepřesnosti modelu dynamiky míčku a zároveň supluje absenci matice B , o které jsme se bavili dříve. Matice dynamiky stavu je tedy rovna

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

a matice měření zůstává stejná, tedy

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (2.15)$$

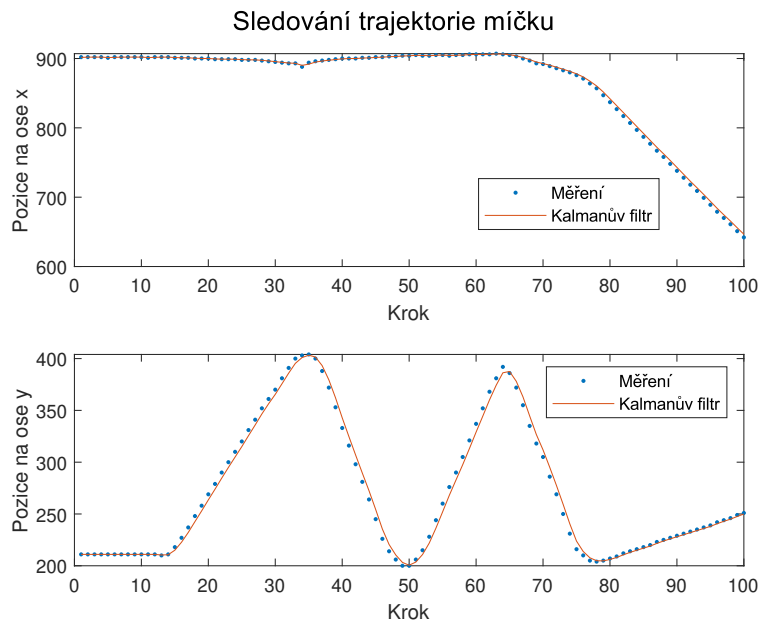
protože data získané z kamery se týkají pouze aktuální polohy. Tyto matice následně použijeme při programování. Chybí nám určit ještě počáteční odhad stavu \hat{x}_0 , počáteční kovarianční matice odhadu P_0 a kovarianční matice šumů Q a R . Obě dvě počáteční hodnoty určíme libovolně, protože předpoklad

úlohy je, že míček bude na startu chvíli na jednom místě. Za tuto chvíli, která je v rádech maximálně sekund, najde Kalmanův filtr aktuální polohu a zpřesní odhad. Matice Q a R volíme časově invariantní a vypadají následovně

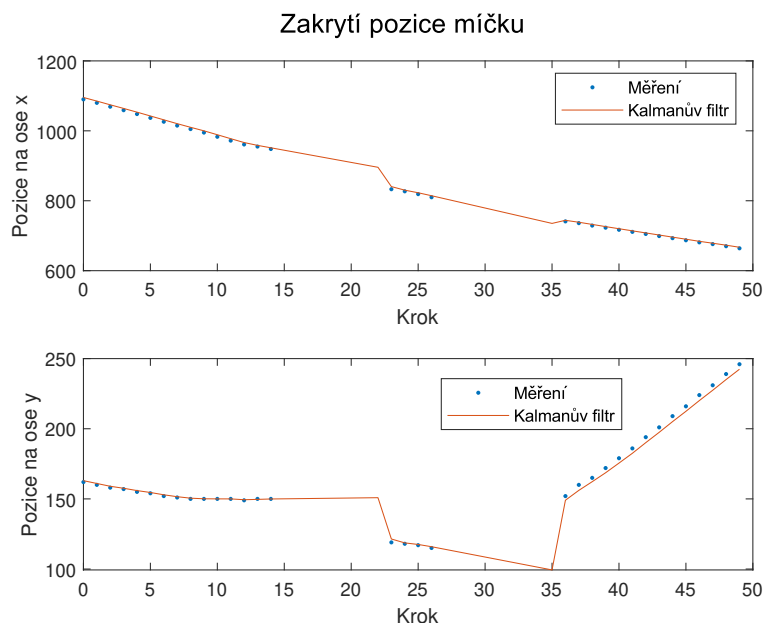
$$Q = \begin{bmatrix} 10^{-3} & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10^{-3} & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad (2.16)$$

$$R = 10^{-5} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \quad (2.17)$$

S touto volbou parametrů získáme filtr, který rychle reaguje na změny trajektorie míčku, který při hře nastává velice často. Tato skutečnost je vidět na obrázku 2.22, kde Kalmanův filtr dostatečně přesně a rychle filtruje sledované hodnoty. Na obrázku 2.23 pak vidíme, že pokud ztratíme informaci o poloze míčku, což může být způsobeno například zakrytím míčku hráčem, Kalmanův filtr uvažuje predikované hodnoty až do bodu, kdy znovu detekujeme pozici míčku. V tento moment začne znovu správně filtrovat aktuální polohu.



Obrázek 2.22: Graf zobrazující časový vývoj trajektorie míčku v osách x a y

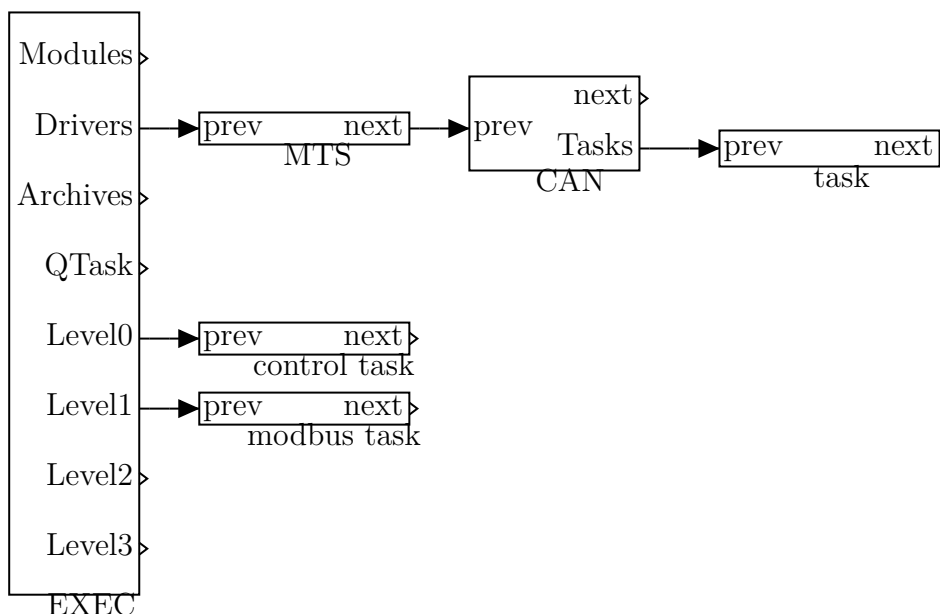


Obrázek 2.23: Graf zobrazující časový vývoj trajektorie míčku v momentě ztráty informace

2.6 Implementace všech částí a napojení na AI část

Implementace komunikace s motory, regulátorů i startovací sekvence je v programu *REXYGEN* na průmyslovém PC. Původní plán byl, že se řídicí systém *REXYGEN* postará o kompletní funkčnost aplikace stolního fotbalu včetně Kalmanova filtru, rozpoznávání obrazu i neuronová síť. Kvůli nekompatibilitě kamery s operačním systémem *Linux*, bylo nutné rozpoznání obrazu i neuronovou síť přesunout na externí počítač s operačním systémem *Windows*, se kterým již kamera kompatibilní je. Protože v tomto projektu slouží Kalmanův filtr zejména pro výpomoc neuronové síti v situacích, kdy při zpracování obrazu nejsme schopni najít míč, je spolu s nimi realizován na externím počítači. Tímto krokem jsme od sebe oddělili řídicí část a část, která se stará o zpracování a následné vyhodnocení situace, což bohužel způsobuje nutnost použití dalšího zařízení, ale časová úspora a zjednodušení je nesporné. Komunikace mezi oběma počítači je zajištěna přes protokol *Modbus TCP/IP*, který je hojně využíván v průmyslu a je dobře konstruován potřebám našeho projektu, tedy pro potřeby přenášet celočíselné a boolovské proměnné. Díky tomuto spojení, odpadá problém časování a udělování priorit při běhu více

úloh v průmyslovém počítači. Navíc je tato komunikace dostatečně rychlá, že bez problémů zvládá přenášet požadovaná data a neomezuje tak plynulý chod motorů.



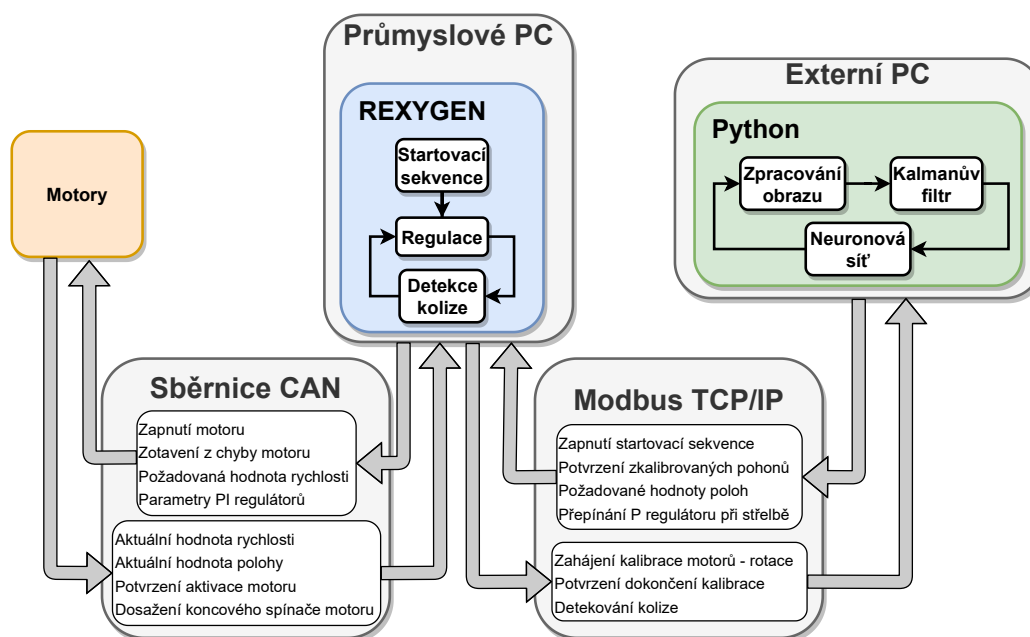
Obrázek 2.24: Schéma exekutivy řídicího systému

Na obrázku 2.24 můžeme vidět bloky, které se starají o jednotlivé úkoly. Bloky *MTS* a *CAN* představují drivery, které se starají o komunikaci *Modbus TCP/IP* a *CAN*. Blok *task* se stará o komunikaci mezi řídicím systémem a motory, blok *control task* obsahuje veškerou řídicí logiku, tedy regulátory, startovací sekvenci, detekci chybových stavů. Poslední blok *modbus task* pracuje s registry určenými pro komunikaci s druhým PC. Druhý PC je v komunikaci nadřazen, tedy zapisuje a čte hodnoty registrů v tomto bloku.

Startovací sekvence

Myšlenka startovací sekvence je v postupné aktivaci jednotlivých motorů a jejich následnou kalibraci. Ta je nutná, protože motory nedisponují absolutními enkodéry. Všechny motory se při nastavení první požadované polohy laicky řečeno "oklepou", kvůli počáteční inicializaci. Kvůli této vlastnosti motorů, je vždy po zapnutí do motoru poslán pulz na malou požadovanou rychlost, který spustí následnou inicializaci. Protože tato inicializace je nejspíš realizována v uzavřené smyčce, je nutné po dobu pulzu změnit parametry rychlostního a proudového regulátoru v měniči motoru na hodnoty, při kterých je motor

dostatečně neagresivní. Kvůli včasnému zásahu a zamezení poškození je tato počáteční inicializace časově oddělena a je viníkem zdlouhavého startu celého projektu. Po tomto počátečním rituálu můžeme přejít ke kalibraci. Začínáme vždy od obránců a větších motorů, tedy nejdříve kalibrujeme horizontální posun obránců, následně gólmánů a pak ve stejném pořadí probíhá kalibrace jejich rotace. Kalibrace horizontálního posunu je realizována pomalou jízdou směrem ke koncovému spínači, který při sepnutí pošle informaci do měniče motoru, který ji následně předá pomocí *CAN* do řídicího systému *REXYGEN*. V tento moment si zapamatujeme aktuální polohu jako odchylku od nulové pozice. Tuto odchylku odečítáme od aktuální polohy pro dosažení hodnot v rozmezí od nuly k maximálnímu posunu. Kalibrace rotačního pohybu probíhá za pomoci kamerového systému a zpracování obrazu, kde pomalu otáčíme hráči a kamerový systém hledá konzistentní údaj o aktuálním úhlu. Údaj, který vykazoval nejkonzistentnější výsledky, byl plocha hráče, přesněji její minimum. Po několika otáčkách je rotace hráčů zastavena. Úhel hráče v tuto chvíli však není nulový, a proto k aktuální poloze, podobně jako u horizontálního posunu, přičteme hodnotu, která úhel vynuluje. Po zkalibrování všech čtyř motorů nastavíme vlajku, která představuje zahájení hry, po které již získáváme požadované hodnoty poloh motorů z neuronové sítě.



Obrázek 2.25: Implementace celé aplikace

Kapitola 3

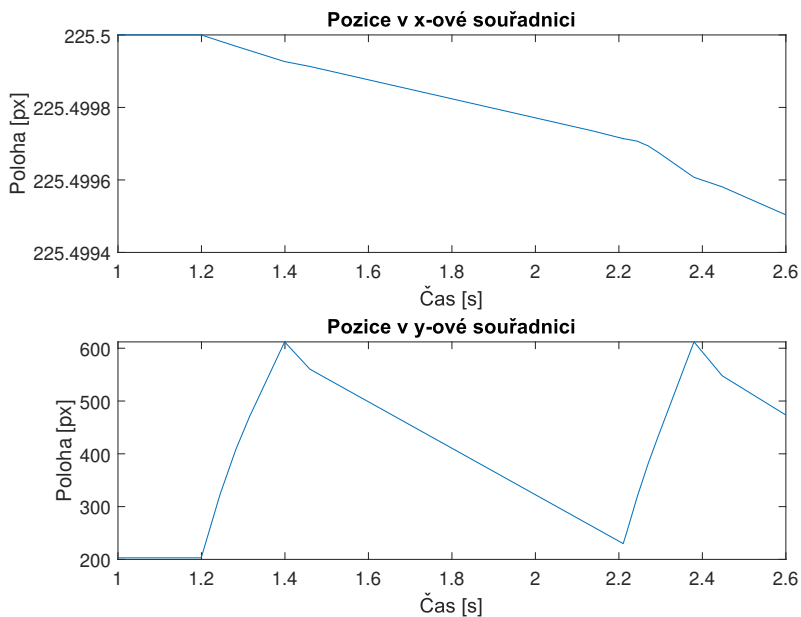
Výsledky experimentů

V této části práce se zaměříme na výsledky experimentů. Nejprve si ukážeme výsledky ze simulace s virtuálním modelem a poté z reálného systému. Na závěr si všechny výsledky porovnáme, abychom měli lepší představu o úspěšnosti experimentu.

3.1 Simulace s virtuálním modelem

Na obrázku 3.1 můžeme pozorovat pohyb míčku, který jsme simulovali ve virtuálním prostředí Simscape. Můžeme vidět, že hodnota v ose x je skoro konstantní a v ose y se hodně mění. Je to dáno popisem experimentu, o kterém jsme se bavili v kapitole 2.3.

Ve virtuálním modelu jsme také navrhovali a testovali regulátory, které byly poté použité na reálném fyzikálním systému. Výsledky těchto experimentů byly popsány v kapitole 2.4.

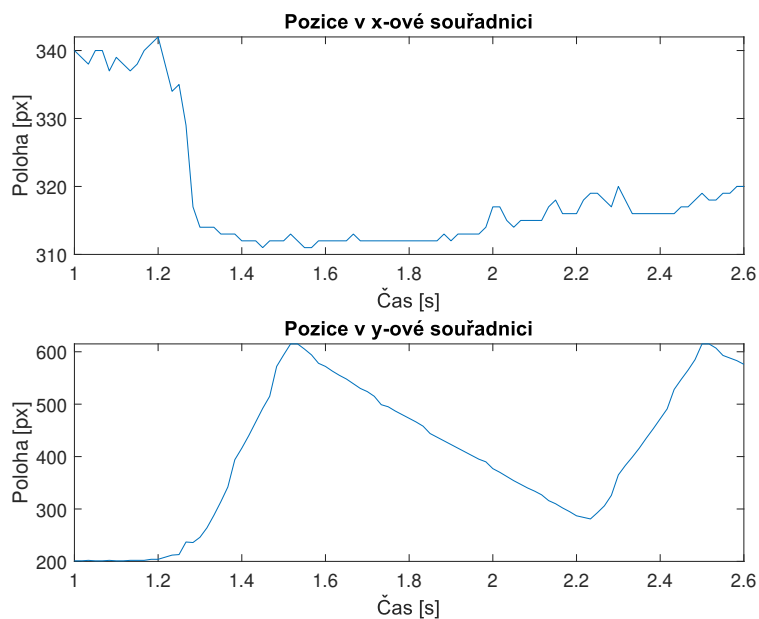


Obrázek 3.1: Simulace pohybu míčku v prostředí Simscape

3.2 Fyzikální systém

Velice podobný experiment jako v simulaci jsme se snažili provést také na reálném systému. Abychom získali data k tomuto obrázku, bylo nutné mít funkční program na detekci míčku ze zpracování obrazu. Tento experiment vystihuje obrázek 3.2, z něhož lze vyčíst, že chování míčku bylo stejné jako v simulaci, jen z počátku byl míč zakryt hráčem, a proto se zdá, že se míček pohyboval nejdříve v ose x, a poté v ose y.

Na fyzikálním systému jsme samozřejmě také testovali a použili regulátory, jejichž výsledky, stejně jako u předchozí podkapitoly, jsou popsány v kapitole 2.4.

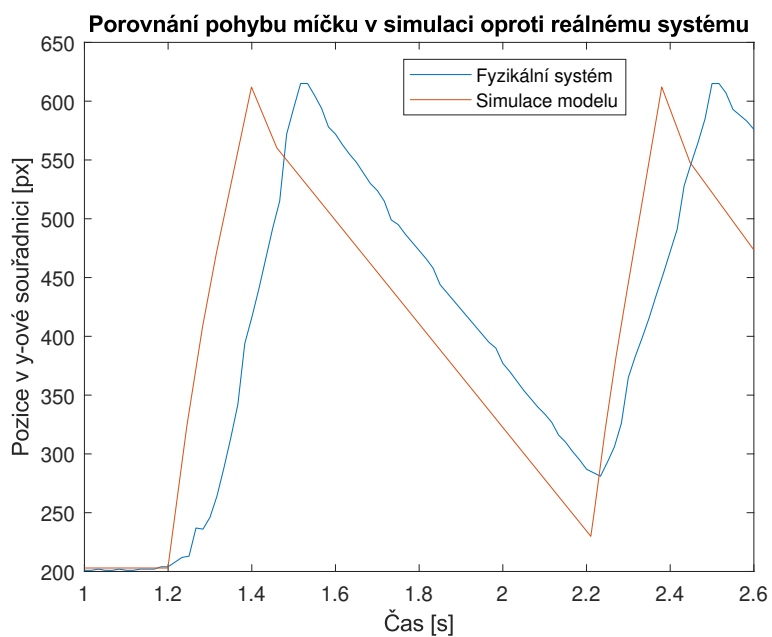


Obrázek 3.2: Pohyb míčku v reálném systému

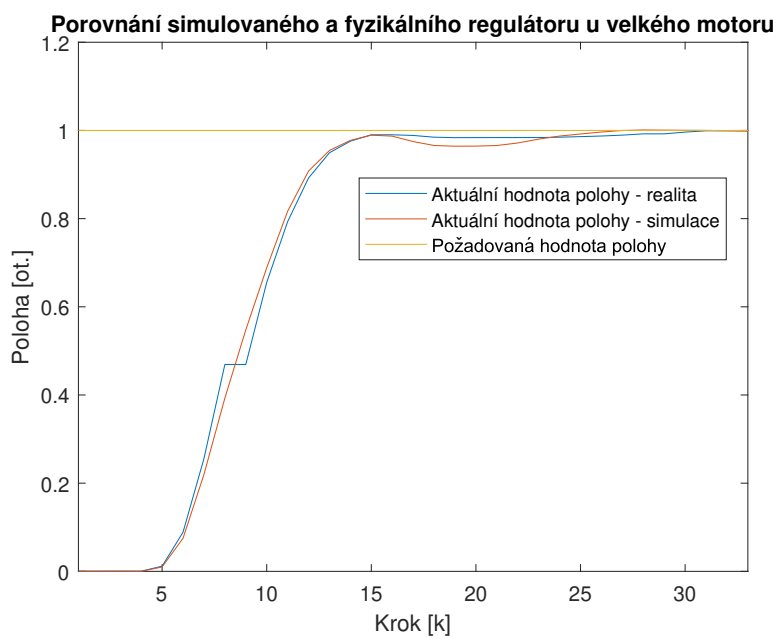
3.3 Porovnání simulace a reálného systému

V této podkapitole si porovnáme výsledky experimentů ze simulace a reálného systému. Na obrázku 3.3 si můžeme všimnout, že předchozí tvrzení o stejné trajektorii bylo správné. Porovnááme pohyb míčku pouze v ose y , protože ta pro nás byla stěžejní. Z důvodu, o kterém jsem se již v práci zmínil, neodpovídá naprosto přesně trajektorie míčku v simulaci a reálném modelu.

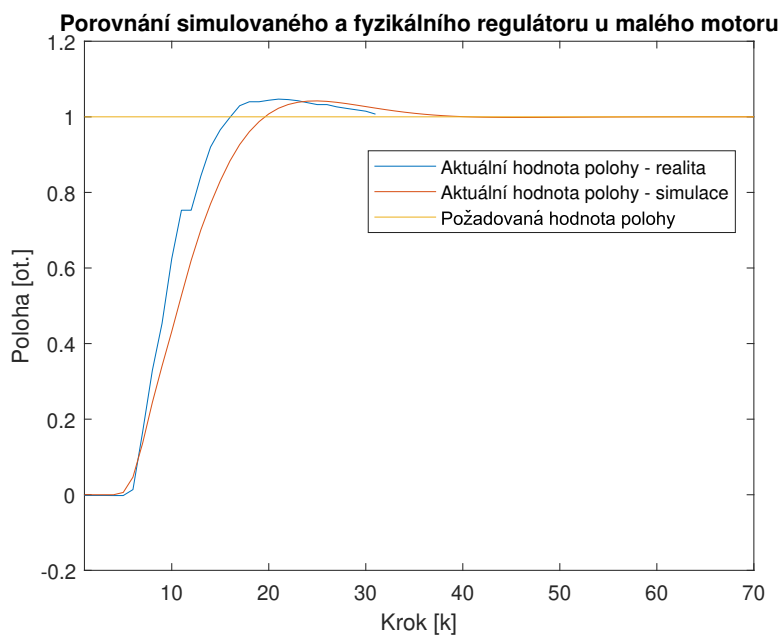
Na obrázcích 3.4, 3.5 a 3.6, můžeme postupně pozorovat srovnání polohového regulátoru velkého motoru, který zařizuje horizontální posun hráčů, polohového regulátoru malého motoru, který zařizuje rotaci hráčů při kontrole míče a nakonec polohového regulátoru malého motoru, který je použit při střelbě míče. Přechodové děje, které vznikly s použitím všech těchto regulátorů, se téměř shodují. Malé rozdíly jsou způsobeny zejména jinými parametry v nastavení regulátorů při simulaci a v reálném systému, ale také rozdílem mezi modelem použitým v simulaci a samotným reálným systémem. Jak jsem již zmiňoval dříve, tento rozdíl nebyl příliš veliký, a proto i parametry regulátorů se od simulačních velmi neliší.



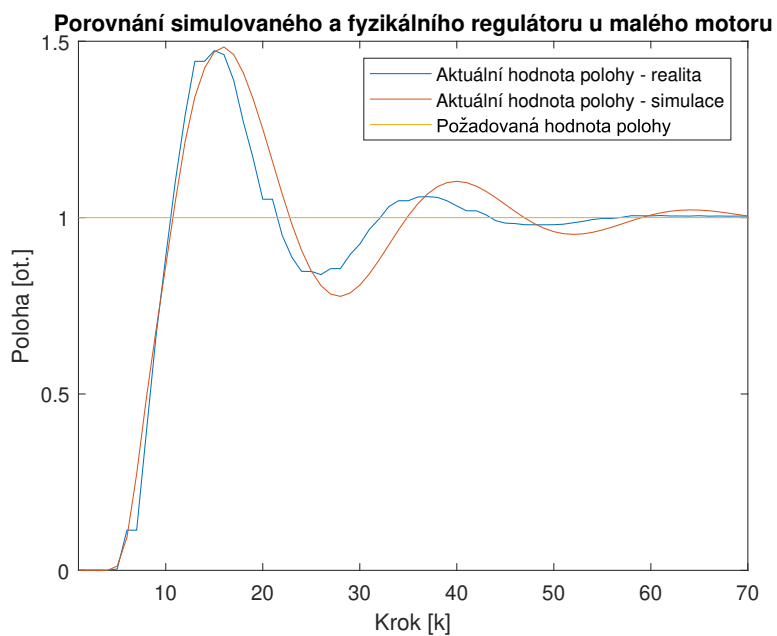
Obrázek 3.3: Porovnání simulace a reálného systému při pohybu míčku



Obrázek 3.4: Porovnání simulace a reálného systému při horizontálním pohybu



Obrázek 3.5: Porovnání simulace a reálného systému při rotačním pohybu za použití klidného regulátoru



Obrázek 3.6: Porovnání simulace a reálného systému při rotačním pohybu za použití agresivního regulátoru

Kapitola 4

Závěr

Cílem této práce bylo navrhnout a sestavit mechatronický model stolního fotbalu s kamerovým systémem pro detekci pohybu míčku a robotickým manipulátorem ovládajícím pohyb gólmana a bránících hráčů, zvolit sestavu pohonů a mechaniku dvouosého manipulátoru pro řízení pohybu hráčů, navrhnout algoritmus řízení pohybu bránících hráčů na základě odhadované trajektorie míčku, otestovat tyto algoritmy na virtuálním modelu a na závěr realizovat navrženou strategii řízení na reálném mechatronickém modelu.

V úvodu práce jsme se seznámili s pojmem mechatronických systémů, blíže představili roli řízení a roli umělé inteligence a nakonec jsme nastínilí problémy, které vznikají při návrhu manipulátorů. V druhé kapitole jsme detailně probrali celé řešení této práce, kde jsme si nejprve obecně popsali architekturu systému a následně jsme dílčí prvky podrobněji objasňovali. Tyto dílčí prvky představují návrh mechanické části, tvorbu virtuálního modelu v prostředí Simscape, návrh algoritmu pro obsluhu a řízení pohonů, návrh algoritmu pro sledování míčku a poslední implementace všech zmíněných prvků a jejich napojení na umělou inteligenci. Na závěr práce jsme si ukázali a vyhodnotili výsledky navržených algoritmů na reálném mechatronickém systému a ve virtuálním modelu.

Návrh mechanického modelu proběhl hladce a bez komplikací. Toto tvrzení se již nedá říct o návrhu řídicího systému, který byl zkomplikován použitými pohony, avšak přes všechny nečekané překážky se nám podařilo navrhnout agresivní a přesnou polohovou regulaci. I když je součástí těchto překážek také větší dopravní zpoždění, dosahují doby regulace hodnot, které jsou schopny konkurovat profesionálním hráčům stolního fotbalu, což bylo naše prvotní přání.

V příští iteraci tohoto projektu by bylo skvělé aplikovat celé toto řešení na profesionální herní stůl a implementovat strategie nejen pro obrannou hru, ale i pro útok včetně kontrolovaného postupu směrem k bráně. Tato implementace by vyžadovala obohatit model míčku o rotace, protože jsou hráči z materiálu, který je umožňuje.

Literatura

- [1] J. Duník. *Identifikace systémů a filtrace*. ZČU, 2018.
- [2] M. Jandík. *Návrh systému automatického řízení pohybu pro mechatronický model stolního fotbalu*. ZČU, 2020.
- [3] M. Sieber. *Analysis of the state of table football and prediction of its change based on image data*. ZČU, 2022.
- [4] TRINAMIC. Pandrive smart motors. <https://www.trinamic.com/products/pandrive-smart-motors>. Zpřístupněno: 19.3.2022.

Seznam obrázků

1.1	Vennův diagram znázorňující kompozici mechatroniky	10
2.1	Základní schéma architektury systému	19
2.2	Schéma subsystému pohonů	20
2.3	Pohon PD42-3-1670 použitý v manipulátoru [4]	21
2.4	Schéma kaskádní regulační smyčky	22
2.5	Herní stůl	25
2.6	Modulární manipulátor	26
2.7	Elektrické schéma zapojení	27
2.8	Schéma z prostředí Simscape	29
2.9	Detailní pohled na subsystém obránce v prostředí Simscape	29
2.10	Regulace rychlosti rotačního pohybu	32
2.11	Regulace rychlosti horizontálního posunu	33
2.12	Data získaná experimentem a použitá pro identifikaci	34
2.13	Odezva na jednotkový skok identifikovaných systémů	35
2.14	Přechodový děj horizontálního pohybu	36
2.15	Přechodový děj rotačního pohybu při použití klidného regulátoru	37
2.16	Přechodový děj rotačního pohybu při použití agresivního regulátoru	37
2.17	Přechodový děj horizontálního pohybu	39
2.18	Přechodový děj rotačního pohybu - klidný regulátor	40
2.19	Přechodový děj rotačního pohybu - agresivní regulátor	40

2.20	Luenbergerův rekonstruktor stavu	42
2.21	Funkční schéma Kalmanova filtru	44
2.22	Graf zobrazující časový vývoj trajektorie míčku v osách x a y .	47
2.23	Graf zobrazující časový vývoj trajektorie míčku v momentě ztráty informace	48
2.24	Schéma exekutivy řídicího systému	49
2.25	Implementace celé aplikace	50
3.1	Simulace pohybu míčku v prostředí Simscape	52
3.2	Pohyb míčku v reálném systému	53
3.3	Porovnání simulace a reálného systému při pohybu míčku . . .	54
3.4	Porovnání simulace a reálného systému při horizontálním po- hybu	54
3.5	Porovnání simulace a reálného systému při rotačním pohybu za použití klidného regulátoru	55
3.6	Porovnání simulace a reálného systému při rotačním pohybu za použití agresivního regulátoru	55