

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

Diplomová práce

Řízení pohybu robotu UR3 pomocí protokolu Modbus

Místo této strany bude
zadání práce.

Prohlášení

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

V Plzni dne 18. května 2018

Bc. Štěpánka Blažková

Abstract

The primary goal of this thesis is to develop a universal system for the motion control of the UR3 robot using the Modbus communication protocol. The first part of the thesis summarizes the theoretical background related to the topic of this thesis and describes the basics of the Modbus protocol. Next, we describe the developed system consisting of multiple parts. We implement the motion control of the robot using the PolyScope interface which handles data provided by the REGYXEN. To design the trajectory, we have implemented the standalone application modeling the trajectory using the Bezier curves. To validate the designed system, we have designed several experiments which have proved that the implemented system is fully functional.

Abstrakt

Hlavním cílem práce je navrhnout univerzální systém ovládání pohybu robota UR3 prostřednictvím komunikačního protokolu Modbus. V práci je shrnuta teorie z oblasti robotiky a principu komunikačního protokolu Modbus. Pro řízení pohybu byl navržen systém obsahující několik částí, které se zabývají ovládáním robota a návrhem generátoru požadované trajektorie. Ovládání robota zajišťuje sekvence příkazů v uživatelském rozhraní PolyScope na základě dat poskytnutých řídicím programem implementovaném v programu REXYGEN. Pro návrh trajektorie je vytvořena samostatná aplikace, která umožňuje modelovat trajektorii robota pomocí Beziérových křivek a ověření realizovatelnosti navržené křivky. Na závěr jsou uvedeny experimenty výsledného systému na reálném robotu UR3. Na základě výsledků experimentů se ukázalo, že navržený systém je plně funkční.

Poděkování

Tímto bych ráda poděkovala panu Prof. Ing. Miloši Schlegelovi, CSc za vedení diplomové práce, cenné rady a odbornou pomoc, a také za poskytnutý čas kdykoliv bylo třeba. Dále bych ráda poděkovala panu Ing. Martinu Švejdovi, Ph.D. za pomoc v oblasti robotiky a za poskytnuté materiály pro simulační model robota. Poděkování patří také Ing. Arnoldu Jágerovi za pomoc s uživatelským rozhranním reálného robota a Ing. Milanu Štětinovi za pomoc s pochopením a nastavením komunikace protokolu Modbus. V neposlední řadě bych ráda poděkovala své rodině za podporu při studiu.

Obsah

1	Úvod	8
2	Uvedení do problematiky	9
2.1	Robotika jako vědní obor	9
2.2	Sériové manipulátory	9
2.2.1	Rozdělení dle kinematické konstrukce	9
2.2.2	Složení dle mechanické konstrukce	10
2.2.3	Vyjádření obecné polohy	11
2.3	Reprezentace obecného pohybu	11
3	Komunikační protokol Modbus	12
3.1	Modbus Organization	12
3.2	Protokol Modbus	12
3.3	Modbus TCP/IP	13
3.3.1	Protocol Data Unit	14
3.3.2	Datový model	14
4	Universal Robots	16
4.1	Robotické manipulátory UR	16
4.2	Technické parametry UR3	17
5	Návrh architektury systému řízení	19
6	Možnosti konstrukce křivek	20
6.1	Parametrizace křivky metodou nejmenších čtverců	20
6.1.1	Metoda nejmenších čtverců	21
6.1.2	Doplnění o Lagrangeovy multiplikátory	22
6.1.3	Ortogonální polynomy	23
6.1.4	Gramm-Schmidtův ortogonalizační proces	24
6.2	Parametrizace křivky pomocí Bézierových křivek	26
6.2.1	Algoritmus de Casteljaou	27
6.3	Zhodnocení a výběr vhodné metody	32
6.4	Znázornění křivek v prostoru	35
6.5	Parametry pohybu po trajektorii	37
6.5.1	Požadavky na pohybové vlastnosti	37
6.5.2	Výpočet délky křivky	37

6.5.3	Výpočet pohybových vlastností	39
7	Kinematický model manipulátoru	41
7.1	Denavit-Hartenbergova úmluva (D-H)	41
7.2	Inverzní kinematický popis manipulátoru	43
7.2.1	Kompenzace polohy základny a koncového efektoru .	43
7.2.2	Polohové závislosti	44
7.2.3	Závislosti rychlosti a zrychlení	45
8	Implementace systému řízení	47
8.1	PolyScope	47
8.1.1	Konfigurace robota	47
8.1.2	Program jako série instrukcí	49
8.2	REXYGEN	50
8.3	Generátor trajektorie	52
8.3.1	Založení projektu	52
8.3.2	Hlavní okno	53
8.3.3	Zobrazení pohybových vlastností	56
9	Demonstrace funkčnosti navrženého systému	58
9.1	Generování trajektorie	58
9.2	Experimenty s reálným robotem UR3	65
10	Závěr	69
	Literatura	71
A	Technické dokumenty	73

1 Úvod

Cílem práce je navrhnout univerzální systém ovládání pohybu robota UR3 prostřednictvím komunikačního protokolu Modbus.

V první části se práce zabývá úvodem do problematiky robotů, kde lze najít popis robotiky jako vědního oboru a užitečné informace týkající se sériových manipulátorů. Jsou zde také zavedeny a vysvětleny pojmy z robotiky, které jsou klíčové v dalším textu. Obdobným způsobem je popsána funkčnost a vlastnosti protokolu Modbus, kde je uveden stručný popis protokolu a způsobu komunikace, která bude využívána v navrženém systému řízení. V neposlední řadě jsou zde popsány funkční vlastnosti robotů Universal Robots, včetně technických parametrů.

Další částí práce je samotný návrh architektury systému řízení, kde jsou definovány jednotlivé moduly obsahující části systému řízení a jejich vzájemné vazby. Prvním modulem je aplikace řešení v programu PolyScope, která zajistí vlastní řízení robota UR3. Druhý modul je implementován v programu Rexygen, který zajišťuje dávkování dat a řízení sekvence funkcí pomocí protokolu Modbus v aplikaci řešení v PolyScope. Dalším modulem je generátor trajektorie, který je navržen jako grafický editor pro definování požadované trajektorie. K tomuto modulu je navržen poslední modul obsahující simulační model robota UR3, který zajistí ověření realizovatelnosti navržené trajektorie.

Pro způsob řešení návrhu požadované trajektorie je navrženo několik algoritmů aproximace a interpolace na základě zadaných bodů, ze kterých je dále vybrán nejvhodnější algoritmus pro požadovaný způsob generování trajektorie. Dále práce popisuje způsob vytvoření kinematického modelu manipulátoru, který je obsažen v modulu simulačního modelu.

V poslední části práce je popsána implementace řešení jednotlivých modulů a demonstrace funkčnosti navrženého systému, kde lze najít popis správného nastavení parametrů v jednotlivých modulech, návod pro pracování se systémem řízení a experimenty prováděné na reálném robotu UR3.

2 Uvedení do problematiky

Tato práce je zaměřená především na oblast robotiky a témat s ní souvisejících. Z důvodu velkého množství specifických pojmů náležících tomuto oboru, je zde uveden stručný úvod pro lepší pochopení dané problematiky. Následující text této kapitoly vychází z prací [21] a [22].

2.1 Robotika jako vědní obor

Robotika je jedním z významných oborů mechatroniky, která sjednocuje principy mechaniky, elektroniky, optiky, informatiky a automatického řízení za účelem konstruování jednodušších, účinnějších a spolehlivějších systémů [21]. V dnešní době se jedná o velmi rozšířený vědní obor, který se zabývá návrhem a konstrukcí robotů, návrhem vhodného algoritmu řízení, simulací chování robotů, vlastní realizací a uvedením do provozu. Obsahuje tak celý proces návrhu.

Pod pojmem robot si lze představit mechanické zařízení, které je schopné provádět definované operace na základě automatického řízení. Mezi základní skupiny robotů patří manipulátory a humanoidní roboti. Manipulátory byly navrženy především z důvodu zjednodušení práce člověka. Jedná se o programovatelné, plně automatické, přesné a rychlé zařízení, které neúnavně a bezchybně nahradí člověka [21]. Využití těchto manipulátorů se najde v různých odvětvích, například v průmyslu nebo medicíně, od výrobních linek po laparoskopické chirurgické operační roboty. Časté využití je také při práci v nebezpečném prostředí. Druhou skupinu tvoří humanoidní roboti, u kterých je snaha napodobit inteligenci a pohybovou dispozici člověka.

2.2 Sériové manipulátory

Manipulátory se rozdělují do dvou základních skupin podle jejich kinematické konstrukce na sériové a paralelní. Kinematickou konstrukcí je myšleno napojení jednotlivých ramen pomocí kloubů.

2.2.1 Rozdělení dle kinematické konstrukce

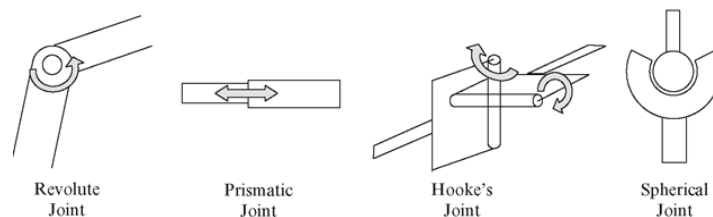
Sériové manipulátory lze jednoduše popsat jako manipulátory, kde je každé rameno spojeno klouby právě se dvěma dalšími rameny s výjimkou ramen

typu základna a koncový efektor. Ty obsahují pouze jedinou vazbu. Vazbami jsou nazvány jednotlivé klouby, které omezují vzájemný pohyb ramen [21]. Tento typ manipulátorů patří mezi nejrozšířenější díky svému rozsahu uplatnění. Další možná konstrukce jsou *paralelní manipulátory*, které mají spojený koncový efektor se dvěma a více částmi, které se samostatně dají chápat jako sériové manipulátory. Paralelní manipulátory nejsou využívány v takové míře jako sériové manipulátory. Důvodem je složitější mechanická konstrukce a tím také větší nároky na jejich návrh a řízení.

2.2.2 Složení dle mechanické konstrukce

Obecně se konstrukce manipulátorů skládá z ramen a kloubů. Rameno typu *základna* (base) je pevná část manipulátoru, na které je definován světový souřadný systém. *Koncový efektor* (end effector) je poslední rameno manipulátoru, ke kterému jsou obvykle připevňovány pracovní nástroje a jehož poloha je řízena na požadovanou hodnotu. Jednotlivá ramena jsou spojena pomocí čtyř různých typů kloubů (posuvný, rotační, univerzální a sférický) znázorněných na obrázku 2.1, které se od sebe liší počtem a typem stupňů volnosti [21]. Posuvný P-kloub (Prismatic) přidává jeden translační stupeň volnosti. Rotační R-kloub (Revolute) oproti tomu přidává jeden rotační stupeň volnosti. Dále se používá univerzální U-kloub (Universal, Hooke's), který přidává dva rotační stupně volnosti a poslední z nejčastěji používaných kloubů je sférický S-kloub (Spherical) přidává všechny tři rotační stupně volnosti [21].

Pohony, které zajišťují pohyb manipulátoru se nazývají aktuátory. Pro různé klouby byly vytvořeny různé typy aktuátorů, podle pohybu, který daný kloub vykonává. Typické pohony manipulátoru představují rotační a lineární (přímočaré) pohony [21].



Obrázek 2.1: Základní typy kloubů manipulátoru. Zdroj [2].

Počet stupňů volnosti (DoF - Degrees of Freedom) vyjadřuje minimální počet parametrů (rotace, translace), který jednoznačně popisuje polohu bodu nebo tělesa v rovině či prostoru [21].

2.2.3 Vyjádření obecné polohy

Obecná poloha tělesa v prostoru je určena jeho translací a rotací. Translace tělesa je zjednodušeně posun o určité hodnoty souřadnic x , y a z . Rotaci lze chápat jako otočení tělesa a jeho souřadného systému okolo jedné či více os. Realizace rotace je možná pomocí Eulerových úhlů, osy rotace a úhlu natočení kolem osy.

2.3 Re prezentace obecného pohybu

Re prezentace obecného pohybu vychází ze vzájemné transformace polohy souřadných systémů pevně umístěných v jednotlivých ramenech manipulátoru [22]. Re prezentaci polohy manipulátoru lze vyjádřit pomocí matice rotace, obecné osy rotace a jednotkového kvaternionu.

Problém v nastavení obecného pohybu manipulátoru nastává, pokud se manipulátor dostane do singulární polohy. Jedná se o polohu, kdy nastává nejednoznačnost v určení některého z kloubových úhlů. K této poloze může dojít při výpočtu zpětné (inverzní) transformace, kdy jsou získávány Eulerovy úhly α , β , γ z matice rotace nebo kdy je získávána rotace okolo obecné osy r o úhel θ z re prezentace pomocí obecné osy rotace. Tyto singularities lze zcela odstranit použitím re prezentace pomocí jednotkového kvaternionu [22]. Podrobný popis způsobů re prezentace orientace je popsán v [22].

3 Komunikační protokol Modbus

Protokol Modbus byl vytvořen v roce 1979 společností Modicon (dnes Schneider Electric), která dříve zajišťovala podporu a udržování protokolu. Dnes tuto roli zastává nezávislá organizace pro vývojáře a uživatele Modbus Organization. Protokol Modbus je velmi oblíbeným nástrojem pro komunikaci po síti nebo sběrnici. Otevřený protokol Modbus je nejpoužívanější síťový protokol v průmyslovém výrobním prostředí. Dodnes stále narůstá jeho využití, na což má vliv jednoduchá struktura protokolu a snadná komunikace ve všech typech sítí. Veškeré informace v této kapitole jsou uvedeny v dokumentacích protokolu [16], [17] a z oficiálních stránek Modbus Organization [15].

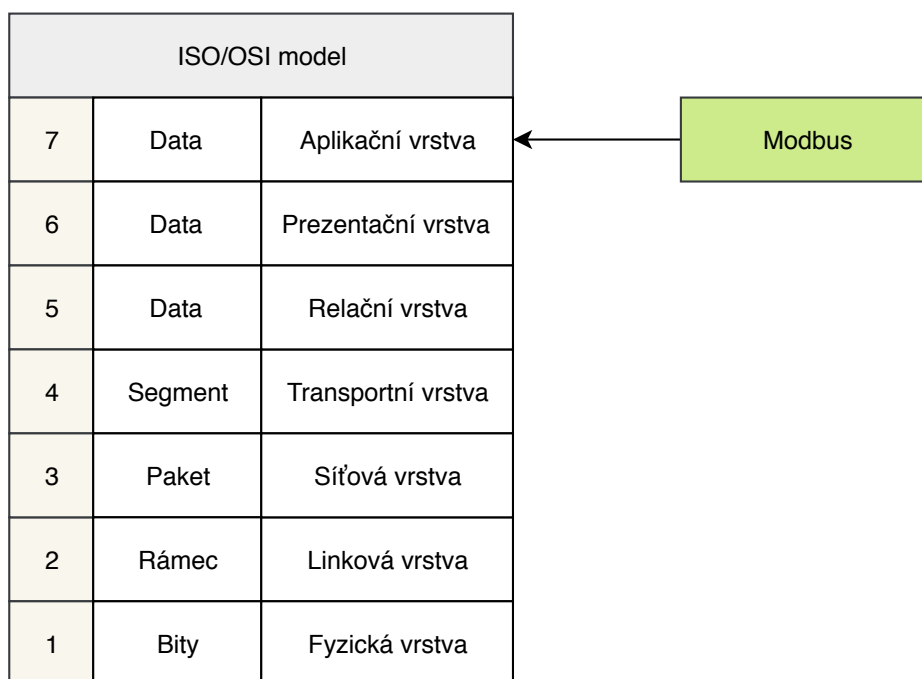
3.1 Modbus Organization

Modbus Organization je sdružení patřící k Modbus Organization, Inc. Jedná se o skupinu nezávislých uživatelů a vývojářů, která usiluje o další možné využití sady komunikačních protokolů Modbus napříč celou řadou průmyslových odvětví. Modbus Organization se zabývá mnoha činnostmi souvisejících s udržováním a šířením protokolu zahrnující vedení a vývoj samotného protokolu a jeho variant, uživatelskou podporu nebo zajištění co největší interoperability (schopnosti systémů vzájemně spolupracovat na technické úrovni, poskytovat si navzájem služby a dosáhnout vzájemné součinnosti).

3.2 Protokol Modbus

Jedná se o otevřený aplikační protokol pro zaslání zpráv aplikovaný na úrovni 7 OSI modelu. Umožňuje komunikaci typu klient-server mezi zařízeními připojenými na různých typech sítí a sběrnicí. V současné době se používá rozšíření protokolu Modbus, který využívá zásobník TCP/IP.

Používá se pro monitorování zařízení a pro zajištění komunikace mezi inteligentními zařízeními a senzory. Oblíbené využití je také pro RTU aplikace (Remote Terminal Unit) což je vzdálená terminálová jednotka řízená mikropočesorem, kdy je vyžadována bezdrátová komunikace.

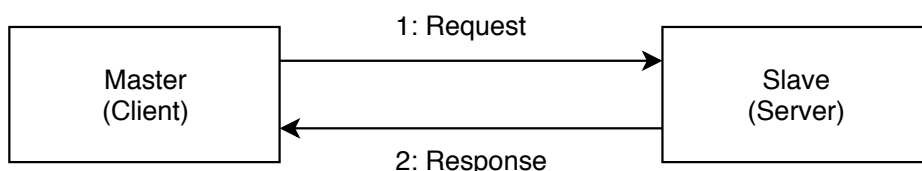


Obrázek 3.1: Grafické znázornění aplikace Modbus protokolu v ISO/OSI modelu.

3.3 Modbus TCP/IP

Zajištění komunikace v této práci bylo provedeno pomocí přenosového protokolu TCP/IP. Výhodou použití specifikace Modbus TCP/IP je možnost použití, protože TCP/IP je dnes rozšířený standard, dále zde nejsou žádné licenční poplatky jelikož se jedná o otevřený protokol a také kompatibilita mezi zařízeními, která Modbus podporují.

Modbus je protokolem postaveným na architektuře klient-server, který je založený na transakcích, které jsou složeny z požadavku (request) vydaného klientem a odpovědi (response) vydané serverem. V oblasti, kde se tento protokol obvykle používá, je jedním z možných schémat, které řídí komunikační chování nižší úrovně v síti pomocí sdíleného kabelu, koncept master-slave, kde master odpovídá klientovi a slave serveru.



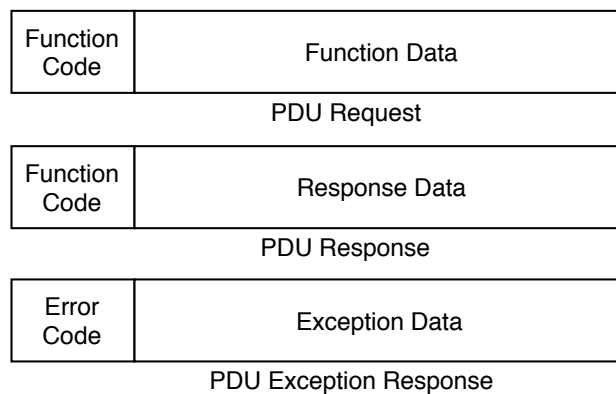
Obrázek 3.2: Grafické znázornění transakce Modbus.

Koncept architektury master-slave se běžně využívá v systémech pracujících v reálném čase, kde mohou být samostatné procesy přidruženy k pořizování dat z prostředí systému, zpracování dat a výpočtům a správě výkonných jednotek [4].

Pro komunikaci Modbus TCP/IP je na TCP vyhrazen port 502, avšak některé aplikace mohou vyžadovat poslech na jiném portu. Z toho důvodu je doporučeno, aby byl vždy definován, ale port 502 musí být nadále vždy dostupný.

3.3.1 Protocol Data Unit

Komunikace je založená na odesílaném balíčku PDU (Protocol Data Unit), tedy protokolové datové jednotce, pro kterou specifikace protokolu definuje tři typy, které jsou znázorněny na obrázku 3.3.



Obrázek 3.3: Modbus Protocol Data Unit.

PDU Request (požadavek) se skládá z kódu určující funkci (Function Code, 1 byte) a funkčních dat (Function Data, různý počet bytů). Response PDU (odpověď) se skládá z funkčního kódu odpovídajícího požadavku (Function Code, 1 byte) a dat odpovědi (Response Data, různý počet bytů). Poslední balíček je Exception Response PDU (výjimka), kde je součástí balíčku funkční kód odpovídající požadavku s přičtenou hodnotou 0x80 (Error Code, 1 byte) a kód upřesňující výjimku (Exception Code, 1 byte).

3.3.2 Datový model

Modbus specifikace definuje čtyři základní datové typy, znázorněné v tabulce 3.3.2, které byly vyvinuty pro výměnu dat.

Name	Type	Access
Discrete Input	single bit	read-only
Discrete Output (Coils)	single bit	read-write
Input Registers	16-bit word	read-only
Holding Registers (Registers)	16-bit word	read-write

Tabulka 3.1: Základní datové typy protokolu Modbus.

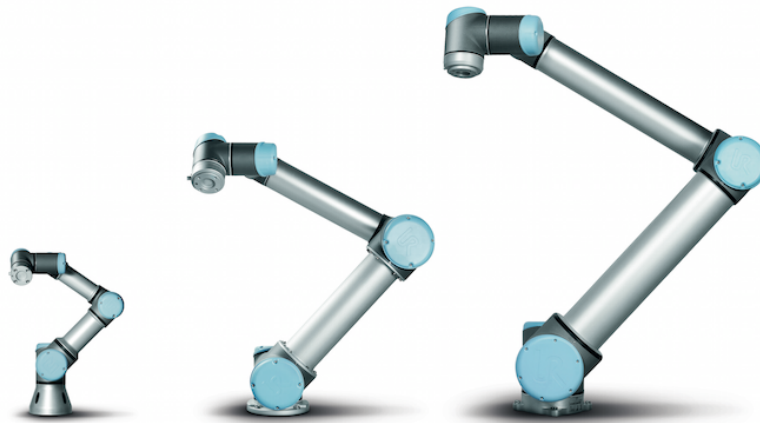
Pro výměnu dat je umožněno použít dvojici diginálních signálů diskrétní vstup (Discrete Input) a diskrétní výstup (Discrete output), které mají velikost jeden bit a vyhodnocují se pro 1 jako zapnuto (on) a pro 0 jako vypnuto (off). Dále je možno použít dvojici registrů (Input Registers, Holding Registers), které mají velikost 16 bitů (rozsah hodnot 0 - 65 535), kde je možné posílat libovolné hodnoty.

4 Universal Robots

Dánská společnost Universal Robots vyvíjí průmyslové roboty s cílem automatizace a zjednodušení opakujících se průmyslových procesů. Na trhu se společnost objevila v roce 2005 s cílem zpřístupnit technologii robotů malým a středním podnikům. Zabývá se robotickými technologiemi, které mohou být využity v širokém pásmu uplatnění. Vývoj byl zaměřen na robotická ramena několika velikostí, která mohou být využita pro různorodé procesy, které je možné automatizovat nebo jsou potenciálně nebezpečné. Uvedené informace jsou dostupné na [23].

4.1 Robotické manipulátory UR

Hlavními produkty Universal Robots je série robotických šestiramenných manipulátorů UR3, UR5 a UR10, které lze vidět na obrázku 4.1. Jednotlivé modely se liší velikostí, dosahem nebo také maximální vahou případného závaží.



Obrázek 4.1: Série robotických manipulátorů společnosti Universal Robots UR3, UR5, UR10. Zdroj [4].

Pro účely testování a návrhu univerzálního řídicího systému byl použit nejmenší model UR3, který je určený pro lehké montáže a automatizované procesy na pracovní stanici [23].

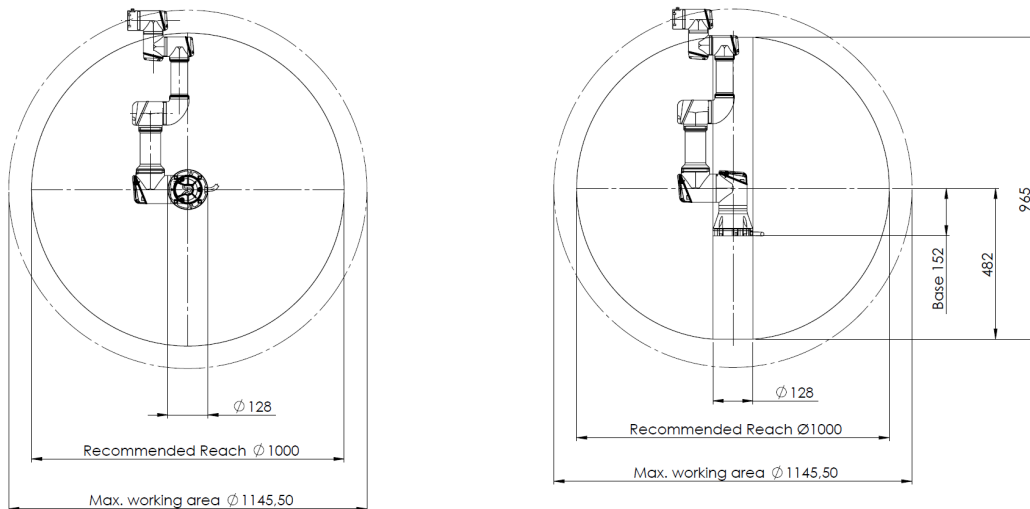
4.2 Technické parametry UR3

Robot UR3 je šestiramenný robotický manipulátor s šesti rotačními klouby, která má šest stupňů volnosti. UR3 váží 11kg a má maximální zatížení závažím 3kg. Na všech zápěstích jsou dostupné rotace 360° s výjimkou posledního kloubu, na který je napojen koncový efektor, který má dostupnou nekonečnou rotaci. Pracovní rozsah vzhledem k dostupným rotačním kloubům a maximální kloubové rychlosti je znázorněn v tabulce 4.1.

Pohyb os robotického ramena	Pracovní rozsah	Maximální rychlost
Podstavec	$\pm 360^\circ$	$\pm 180^\circ/s$
Rameno	$\pm 360^\circ$	$\pm 180^\circ/s$
Loket	$\pm 360^\circ$	$\pm 180^\circ/s$
Zápěstí 1	$\pm 360^\circ$	$\pm 360^\circ/s$
Zápěstí 2	$\pm 360^\circ$	$\pm 360^\circ/s$
Zápěstí 3	Nekonečné	$\pm 360^\circ/s$
Bežný nástroj		1m/s

Tabulka 4.1: Technické parametry jednotlivých kloubů manipulátoru UR3.

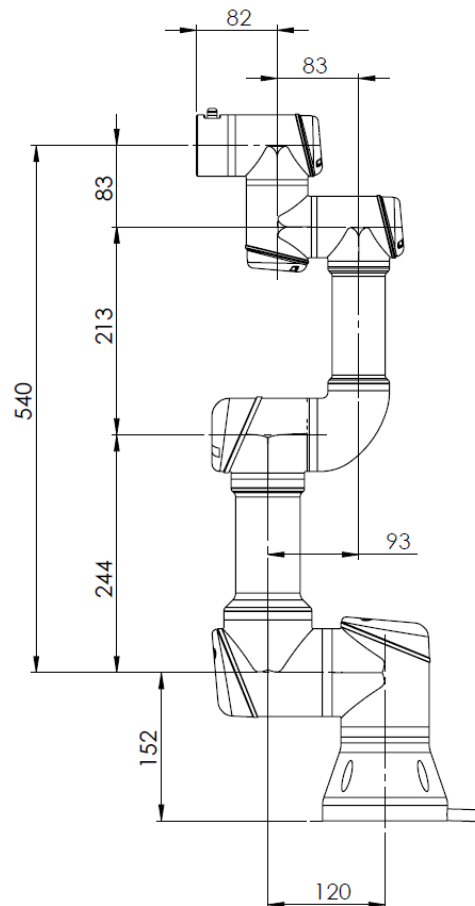
Celkový doporučený dosah manipulátoru výrobcem je 1m avšak pracovní plocha je vyměřena na 1.1455m, přičemž základna tvoří v průměru 0.128m.



Obrázek 4.2: Grafické znázornění dosahu a pracovní plochy manipulátoru UR3 (uvedeno v milimetrech). Zdroj [7].

Zmíněné hranice dosahu jsou znázorněny na obrázku 4.2, kde je grafické znázornění hranic dosahů z pohledu shora a zepředu.

Délky jednotlivých ramen manipulátoru jsou znázorněny na obrázku 4.2, kde je manipulátor v domovské poloze. Domovská poloha je taková poloha koncového efektoru, při které jsou jeho kloubové souřadnice nastaveny tak, že koncový efektor manipulátoru zaujímá domovskou polohu [21].

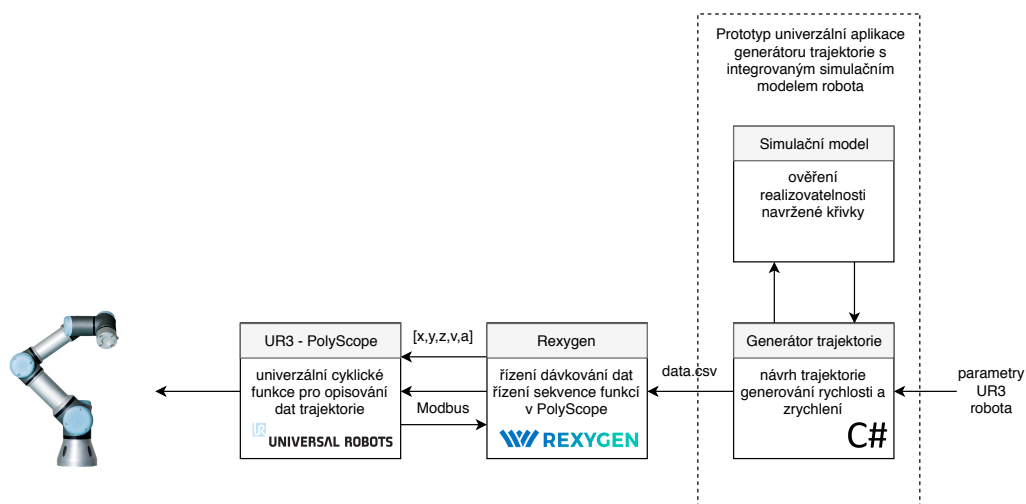


Obrázek 4.3: Grafické znázornění délek jednotlivých ramen manipulátoru UR3 (uvedeno v milimetrech). Zdroj [7].

Veškeré dokumenty obsahující technické parametry získané z oficiálních stránek společnosti Universal Robots [23] jsou dostupné v příloze.

5 Návrh architektury systému řízení

Návrh architektury systému řízení robotického manipulátoru na základě navržené trajektorie obsahuje čtyři moduly, které jsou vzájemně propojeny. Návrh architektury systému řízení je znázorněn na obrázku 5.1.



Obrázek 5.1: Grafické znázornění schématu zapojení jednotlivých modulů architektury navrženého systému. Loga programů a grafické znázornění UR3 byla získána z [5], [3].

Modul, který se stará o vlastní řízení robotického ramene a definuje univerzální cyklické funkce pro opisování trajektorie je navržen v grafickém uživatelském prostředí PolyScope, které je součástí robota UR3. Pro řízení chování prvního modulu byl zaveden řídicí cyklus pomocí REXYGEN aplikace, která zajišťuje správné dávkování vstupních dat a stará se o komunikaci mezi PolyScope a REXYGEN. Komunikace mezi těmito moduly byla navržena pomocí protokolu Modbus.

Dalším modulem je generátor trajektorie, kde bude implementováno grafické prostředí pro návrh křivky, a také bude využívat modul simulačního modelu, který zajistí ověření realizovatelnosti definovaného pohybu manipulátoru. Na vstupu do generátoru jsou požadovány parametry robota UR3.

6 Možnosti konstrukce křivek

Pro generování požadované křivky je bezpočet dostupných metod aproximace a interpolace pomocí polynomů či bázových funkcí [13], [12], [18]. Požadavkem na použité metody je vygenerování křivky na základně zadaných bodů, které budou rozděleny na dva typy. Pevné body jimiž musí bezpodmínečně křivka procházet a volné body určené k tvarování křivky na požadovaný tvar.

Pro návrh křivky byly použity dvě různé možnosti přístupu. Prvním přístupem byla idea, že bude daná křivka generována jako celek na základě zadaných dat. Pro tento přístup bylo použita metoda nejmenších čtverců pro parametrizaci pomocí polynomů. Druhým přístupem byl návrh křivky postavený na základě metody parametrizace pomocí Bézierových křivek, kdy byla uvažovaná křivka rozdělena na segmenty, které představovaly jednotlivé Bézierovy křivky. Tyto segmenty byly propojeny do výsledné křivky.

6.1 Parametrizace křivky metodou nejmenších čtverců

Metoda nejmenších čtverců je matematicko-statistická metoda určená pro aproximaci řešení přeuročených soustav rovnic, kde výsledné řešení má minimalizovat součet čtverců odchylek vůči každé rovnici [10]. Tato část kapitoly věnující se metodě nejmenších čtverců vychází z textů [12], [20], [14], [9], [10]. Jsou-li vstupem pevné (kontrolní) a volné (řídící) body, kdy je uvažována křivka jako jednotný celek, bude použita pro generování křivky metoda nejmenších čtverců, kde bude minimalizováno kritérium pro vzdálenost křivky od daných bodů. V případě, že jsou zadány kontrolní body, tudíž body, kterými musí křivka bezpodmínečně procházet, je nutné tuto metodu doplnit o metodu Lagrangeových multiplikátorů, která zajistí nulovou odchylku od kontrolních bodů.

6.1.1 Metoda nejmenších čtverců

Pro každou dimenzi budou formulovány parametrické polynomy ve tvaru

$$\begin{aligned}x(t) &= a_0 + a_1t + a_2t^2 + a_3t^3 + \dots a_nt^n, \\y(t) &= b_0 + b_1t + b_2t^2 + b_3t^3 + \dots b_nt^n, \\z(t) &= c_0 + c_1t + c_2t^2 + c_3t^3 + \dots c_nt^n,\end{aligned}\tag{6.1}$$

kde parametrem je uvažován čas t , pro který bude definována poloha v osách x , y případně z . Pro každou dimenzi, tedy i každý polynom budou uvažovány kritériální funkce

$$\sum_{i=1}^n (x(t_i) - x_i)^2, \sum_{i=1}^n (y(t_i) - y_i)^2, \sum_{i=1}^n (z(t_i) - z_i)^2 \rightarrow \min,\tag{6.2}$$

které budou minimalizovány. Souhrně je možno všechny tyto kritériální funkce do jedné ve tvaru

$$\sum_{i=1}^n [(x(t_i) - x_i)^2 + (y(t_i) - y_i)^2 + (z(t_i) - z_i)^2] \rightarrow \min.\tag{6.3}$$

Následující postup vyjádření metody nejmenších čtverců je analogický pro jednotlivé dimenze jelikož všechny polynomy jsou parametrizované časem t a liší se pouze v konstantách a_n , b_n či c_n . Z tohoto důvodu bude postup demonstrován na 1D případě.

Řešení metody nejmenších čtverců bude získán ze vztahu

$$HK_x = X,\tag{6.4}$$

který lze vyjádřit v maticovém tvaru

$$\begin{bmatrix} 1 & t_1 & t_1^2 & \dots & t_1^r \\ 1 & t_2 & t_2^2 & \dots & t_2^r \\ 1 & t_3 & t_3^2 & \dots & t_3^r \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & t_n^2 & \dots & t_n^r \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_r \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix},\tag{6.5}$$

kde n je počet zadaných kontrolních bodů a r je zvolený řád aproximovaného polynomu. Cílem je najít vektor parametrů K_x . V dalších krocích bude vyjádřeno řešení zadaného vztahu.

$$\begin{aligned}HK_x &= X \\H^T HK_x &= H^T X \\K_x &= (H^T H)^{-1} H^T X\end{aligned}\tag{6.6}$$

Přičemž K_x je vektor hledaných parametrů a_0, \dots, a_r . Tímto způsobem je možné vyjádřit aproximovanou křivku v polynomiálním tvaru, která se bude co nejvíce přibližovat zadaným bodům.

6.1.2 Doplnění o Lagrangeovy multiplikátory

Z důvodu zavadení dvou typů vstupních bodů je nutné metodu nejmenších čtverců rozšířit o Lagrangeovy multiplikátory, které zajistí splnění podmínky pro kontrolní body, a to nulovou odchylku křivky od pevných bodů. Problém metody nejmenších čtverců může být formulován vztahem

$$\|Hk_x - X\|_2^2, \quad (6.7)$$

takový, že podléhá pevně definovaným bodům

$$Ck_x = X_p, \quad (6.8)$$

kde X_p je vektor kontrolních bodů. Dále bude zadefinován Lagrangian

$$\mathcal{L}(k_x, \lambda) = \frac{1}{2}\|Hk_x - X\|_2^2 + \lambda^T(Ck_x - X), \quad (6.9)$$

kde C je definováno obdobným způsobem jako H s rozdílem, že zde budou uvažovány pouze parametry t pro kontrolní body nikoliv pro všechny zadané body.

$$C = \begin{bmatrix} 1 & t_1 & t_1^2 & \dots & t_1^r \\ 1 & t_2 & t_2^2 & \dots & t_2^r \\ 1 & t_3 & t_3^2 & \dots & t_3^r \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_m & t_m^2 & \dots & t_m^r \end{bmatrix} \quad (6.10)$$

kde m je počet kontrolních bodů a r je zvolený řád aproximovaného polynomu. Dále budou vyjádřeny parciální derivace Lagrangianu, které budou položeny nule. Tímto způsobem lze získat maticový zápis pro výpočet parametrů polynomu s rozšířením o Lagrangeovy multiplikátory.

$$\begin{aligned} \frac{\partial}{\partial k_x} \mathcal{L}(k_x) &= H^T(X - Hk_x) + C^T \lambda \\ \frac{\partial}{\partial \lambda} \mathcal{L}(k_x) &= Ck_x - X \end{aligned} \quad (6.11)$$

Výsledný maticový zápis řešení, kde budou hledány hodnoty parametrů k_x a λ bude ve tvaru

$$\begin{bmatrix} H^T H & C^T \\ C & 0 \end{bmatrix} \cdot \begin{bmatrix} k_x \\ \lambda \end{bmatrix} = \begin{bmatrix} H^T X \\ X_p \end{bmatrix}, \quad (6.12)$$

Výsledné hodnoty parametrů budou vyjádřeny obdobně jako v předchozím případě bez rozšíření o Lagrangeovy multiplikátory.

$$\begin{bmatrix} k_x \\ \lambda \end{bmatrix} = \begin{bmatrix} (H^T H)(H^T H) + C^T C & (H^T H)C^T \\ C(H^T H) & CC^T \end{bmatrix} \cdot \begin{bmatrix} H^T X \\ X_p \end{bmatrix} \quad (6.13)$$

6.1.3 Ortogonální polynomy

Při generování křivky pomocí polynomů může nastat problém, pokud bude na vstupu použito větší množství pevných bodů. Pro n pevných bodů, musí být polynom minimálně řádu n nebo více, aby byla splněna podmínka průchodu křivky všemi pevnými body. S narůstajícím řádem n však dochází ke špatnému podmínění řešení výrazu ze vztahu 6.6 z důvodu výpočtu inverzní matice. Možným řešením tohoto problému je použití ortogonálních polynomů.

Mezi tyto polynomy patří například Legendrovy polynomy, které jsou ortogonální polynomy s váhovou funkcí $v(x) = 1$ na intervalu $[-1, 1]$ a pro které platí předpis

$$\begin{aligned} P_0(x) &= 1, \\ P_1(x) &= x, \\ &\vdots \\ P_{n+1}(x) &= \frac{2n+1}{n+1}xP_n(x) - \frac{n}{n+1}P_{n-1}(x). \end{aligned} \tag{6.14}$$

Další volbou mohou být Čebyševovy polynomy 1. druhu, které jsou ortogonální s váhovou funkcí $v(x) = \frac{1}{\sqrt{1-x^2}}$ na intervalu $[-1, 1]$ a pro něž platí předpis

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ &\vdots \\ T_{n+1} &= 2xT_n(x) - T_{n-1}(x). \end{aligned} \tag{6.15}$$

Dále také Čebyševovy polynomy 2. druhu, které jsou ortogonální s váhovou funkcí $v(x) = \sqrt{1-x^2}$ na intervalu $[-1, 1]$ a pro něž platí předpis

$$\begin{aligned} U_0(x) &= 1, \\ U_1(x) &= 2x, \\ &\vdots \\ U_{n+1} &= 2xU_n(x) - U_{n-1}(x). \end{aligned} \tag{6.16}$$

Použitím ortogonálních polynomů bude zajištěna vyšší přesnost výpočtu pro špatně podmíněnou úlohu kvůli nutnosti počítat inverzní matici ve vztahu 6.6. Díky vlastnostem ortogonálních polynomů nebude již úloha špatně podmíněna, a nebudou vznikat pro aproximaci většího počtu bodů velké nepřesnosti.

6.1.4 Gramm-Schmidtův ortogonalizační proces

Jiné řešení zmírnit špatnou podmíněnost úlohy je použití ortogonalizace pro rozklad matice H použitím Gramm-Schmidtova ortogonalizačního procesu pro dekompozici matice H

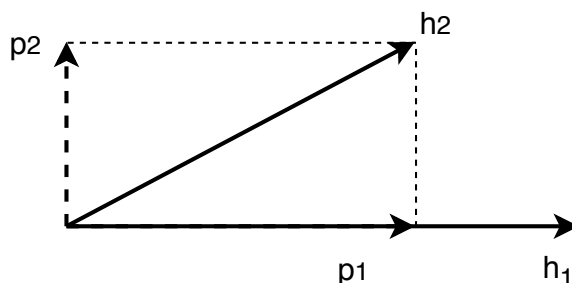
$$H = P \cdot S, \quad (6.17)$$

tak aby matice P splňovala podmínky ortonomální báze, kde vektory p_1, \dots, p_n obsažené v matici P jsou navzájem ortogonální $p_i \cdot p_j = 0$ pro $i \neq j$. Do původního algoritmu bude zpět dosazena matice S namísto H , čímž budou splněny podmínky ortogonality.

Matice H je uvažována jako matice tvořená vektory h_1, \dots, h_n , které nesplňují podmínky ortogonality. Dále je uvažována matice P jako ortogonální báze vektorů p_1, \dots, p_n , které budou navrženy tak, aby generovaly stejný bazový podprostor jako vektory matice H , s rozdílem, že na sebe budou navzájem kolmé. Cílem je najít horní trojúhelníkovou matici S tvořenou koeficienty, které zajistí, že vektory matice H a matice P budou tvořit stejný bazový podprostor.

$$\begin{bmatrix} h_1 & h_2 & \dots & h_n \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & \dots & p_n \end{bmatrix} \cdot \begin{bmatrix} 1 & s_{12} & \dots & s_{1n} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & s_{n-1,n} \\ 0 & \dots & 0 & 1 \end{bmatrix}. \quad (6.18)$$

Gramm-Schmidtův ortogonalizační proces je definován následujícími kroky. Postupně budou vyjadřovány jednotlivé vektory matice P . První vektor p_1 je roven původnímu vektoru h_1 . Druhý vektor p_2 je vyjádřen na základě prvního vektoru p_1 a vektoru h_2 , takovým způsobem, aby byl vůči vektoru p_1 kolmý a zároveň dodržel stejný bazový podprostor jako původní vektory jak je znázorněno na obrázku 6.1.



Obrázek 6.1: Vyjádření druhého bazového vektoru p_2 .

Třetí vektor bude vyjádřen obdobným způsobem, přičemž budou uvažovány již vyjádřené vektory p_1 a p_2 a vektor h_3 , který pro splnění podmínek ortogonalitativy bázového podprostoru, bude kolmý na oba vektory p_1 , p_2 zároveň.

Výpočet Gramm-Schmidtova procesu ortogonalizace lze popsat následujícími vztahy. Vektory tvořící ortogonální bázový systém p_1 , p_2 , p_3 jsou vyjádřeny jako

$$\begin{aligned} p_1 &= h_1, \\ p_2 &= h_2 - s_{12}p_1, \\ p_3 &= h_3 - s_{13}p_1 - s_{23}p_2, \end{aligned} \quad (6.19)$$

kde koeficient s_{12} je vyjádřen tak, aby byla splněna podmínka $p_1 \cdot p_2 = 0$, tedy, že navržené vektory na sebe budou kolmé.

$$p_1 \cdot h_2 - s_{12}p_1 \cdot p_1 = 0 \rightarrow s_{12} = \frac{p_1 \cdot h_2}{p_1 \cdot p_1} \quad (6.20)$$

Analogicky jsou vyjádřeny koeficienty s_{13} a s_{23}

$$s_{13} = \frac{p_1 \cdot h_3}{p_1 \cdot p_1}, \quad s_{23} = \frac{p_2 \cdot h_3}{p_2 \cdot p_2}. \quad (6.21)$$

Obecné vyjádření algoritmu Gramm-Schmidtova ortogonalizačního procesu lze zapsat jako

$$\begin{aligned} p_n &= h_n - s_{1n}p_1 - \dots - s_{n-1,n}p_{n-1}, \\ s_{jn} &= \frac{p_j \cdot p_n}{p_j \cdot p_j} \quad \text{pro } j = 1, \dots, n-1. \end{aligned} \quad (6.22)$$

Pro dosažení do odvozeného postupu pro metodu nejmenších čtverců rozšířenou o Lagrangeovy multiplikátory je nutné přepočítat vektor požadovaných hodnot X na vektor D dle vztahu

$$d_n = \frac{p_n \cdot X_n}{p_n \cdot p_n}. \quad (6.23)$$

Výsledné vyjádření metody nejmenších čtverců s ohledem na ortogonalitu rošířené o Lagrangeovy multiplikátory bude ve tvaru

$$\begin{bmatrix} S^T S & C^T \\ C & 0 \end{bmatrix} \cdot \begin{bmatrix} k_x \\ \lambda \end{bmatrix} = \begin{bmatrix} S^T D \\ X_p \end{bmatrix}. \quad (6.24)$$

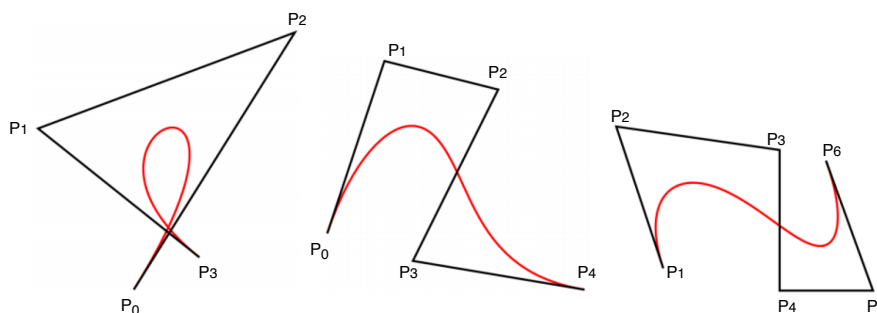
Ortogonalizací je dosaženo výrazného zlepšení podmíněnosti úlohy díky vlastnosti čtvercových ortogonálních matic, jejichž inverzní matice je zároveň transponovaná a tedy je možné namísto výpočtu inverze matice použít její transpozici.

6.2 Parametrizace křivky pomocí Bézierových křivek

Dalším přístupem jak aproximovat křivku pomocí zadaných bodů je metoda parametrizace Bézierovou křivkou. Jedná se o odlišný způsob aproximace oproti metodě nejmenších čtverců jelikož se zde neuvažuje křivka jako celek vyjádřený jedním polynomem stupně n pro každou dimenzi, ale uvažuje se křivka jako soustava Bézierových křivek, které jsou na sebe napojeny kontrolními body. Konstrukce Bézierových křivek byla navržena na základě postupů z prací [11], [18].

Z definice je Bézierova křivka určena řídicím polygonem, čímž je myšlena lomená čára určená polohovými vektory bodů P_0, \dots, P_n . Řídicí polygony pro křivky definované různými polygony je možno vidět na obrázku (6.2).

Pro řídicí polygon platí, že prvním bodem P_0 a posledním bodem P_n daná křivka prochází. Ostatní body P_1, \dots, P_{n-1} dále pouze řídí požadovaný tvar křivky.



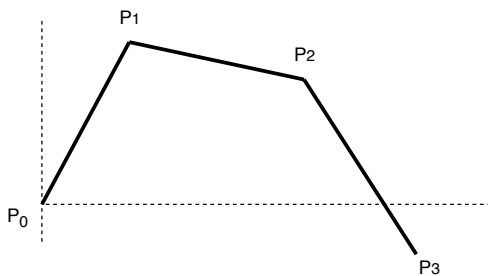
Obrázek 6.2: Příklad Bézierových křivek a jejich řídicích polygonů. Zdroj [1].

Pro aplikaci této metody na problém aproximace zadaných bodů je nutné uvažovat výslednou křivku, která bude tvořena sérií Bézierových křivek, které budou propojovat vždy dva za sebou jdoucí kontrolní body. Řídicí body, které se nachází mezi těmito kontrolními body budou dále uvažovány jako body tvořící řídicí polygon.

Tento přístup má výhodu oproti předchozím přístupům. Pro změnu tvaru jakékoliv části výsledné křivky, je nutné upravovat pouze jedinou Bézierovu křivku, která dále neovlivní tvar ostatních Bézierových křivek. Také je výhodou fakt, že nezáleží na počtu zadaných bodů a celkové složitosti křivky, jelikož s přibývajícím počtem dat přibudou pouze další Bézierovy křivky, a tedy kvalita řešení není limitována řádem polynomu jako v předchozích případech. Tímto způsobem je také možné navrhnout část křivky, která bude představovat přímkou tím, že řídicí polygon bude tvořen pouze dvěma body.

6.2.1 Algoritmus de Casteljaeu

Před začátkem algoritmu je nutné nadefinovat řídicí polygon, který je znázorněn na obrázku (6.3). Na obrázku je nadefinován polygon čtyřmi body $[P_0, P_1, P_2, P_3]$, jedná se tedy o kubickou Bézierovu křivku, přičemž krajními body bude křivka procházet.

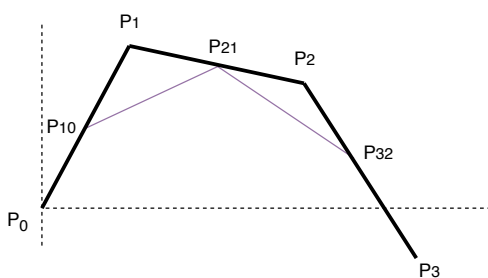


Obrázek 6.3: Nadefinování řídicího polygonu.

Prvním krokem algoritmu je provedena lineární interpolace pro všechny dvojice po sobě jdoucích bodů dle vztahů

$$\begin{aligned} P_0^1(t) &= (1-t) \cdot P_0 + t \cdot P_1, \\ P_1^2(t) &= (1-t) \cdot P_1 + t \cdot P_2, \\ P_2^3(t) &= (1-t) \cdot P_2 + t \cdot P_3. \end{aligned} \tag{6.25}$$

Interpolace pro všechny dvojice bodů je znázorněna na obrázku (6.4).

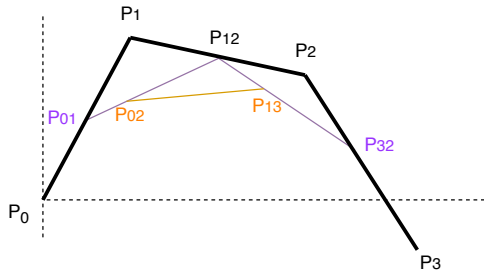


Obrázek 6.4: První krok algoritmu.

Pro druhý krok algoritmu dosadíme do vztahů získané body v prvním kroku. Nové vztahy budou ve tvaru

$$\begin{aligned} P_0^2(t) &= (1-t) \cdot P_0^1 + t \cdot P_1^2, \\ P_1^3(t) &= (1-t) \cdot P_1^2 + t \cdot P_2^3. \end{aligned} \tag{6.26}$$

Interpolace ve druhém kroku je znázorněna na obrázku (6.5).

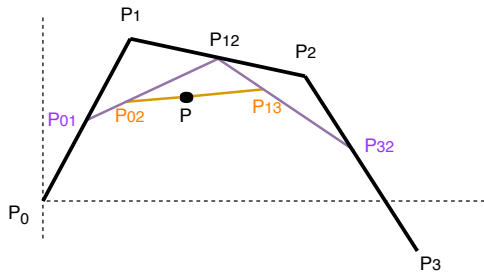


Obrázek 6.5: Druhý krok algoritmu.

V případě, kdy jsou definovány čtyři body v řídicím polygonu, zbývá poslední krok pro výpočet daného bodu na křivce, jehož vztah je vyjádřen jako

$$P(t) = P_0^3(t) = (1-t) \cdot P_0^2 + t \cdot P_1^3. \quad (6.27)$$

Znázornění vypočteného bodu je možné vidět na obrázku (6.6).



Obrázek 6.6: Třetí krok algoritmu.

Počet kroků algoritmu se odvíjí od počtu bodů v řídicím polygonu. Čím více bodů je zadáno, tím vícekrát je nutné provést interpolaci mezi danými body pro výpočet jednoho bodu na křivce. Pro výpočet všech těchto bodů je nutné tento algoritmus provést pro každé $t \in [0, 1]$.

Vzhledem ke skutečnosti, že není dopředu znám přesný počet definovaných bodů pro jednotlivé křivky, bude zdefinován obecný vztah pro výpočet Bézierovy křivky stupně n

$$P(t) = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i \cdot P_i. \quad (6.28)$$

Pro jednodušší implementaci bude zaveden postup výpočtu v maticovém tvaru. Je-li uvažována kubická Bézierova křivka z předchozího odvození bude předpis pro daný bod

$$P(t) = P_1 \cdot (1-t^3) + P_2 \cdot 3 \cdot (1-t)^2 \cdot t + P_3 \cdot 3 \cdot (1-t) \cdot t^2 + P_4 \cdot t^3. \quad (6.29)$$

Nebudou-li uvažovány aktuální souřadnice získáme následující tvar

$$P(t) = (1 - t^3) + 3 \cdot (1 - t)^2 \cdot t + 3 \cdot (1 - t) \cdot t^2 + t^3. \quad (6.30)$$

Vztah (6.30) lze vyjádřit jako součet čtyř výrazů

$$P(t) = [(1 - t^3)] + [3 \cdot (1 - t)^2 \cdot t] + [3 \cdot (1 - t) \cdot t^2] + [t^3], \quad (6.31)$$

který po roznásobení bude ve tvaru

$$P(t) = [-t^3 + 3 \cdot t^2 - 3 \cdot t + 1] + [3 \cdot t^3 - 6 \cdot t^2 + 3 \cdot t] + [-3 \cdot t^3 + 3 \cdot t^2] + [t^3]. \quad (6.32)$$

Vyjádření (6.32) bude dále převedeno na součet vektorových operací. Z každého sčítance lze vytknout vektor $[t^3, t^2, t, 1]$ čímž lze získat součet čtyř vektorových součinů, které lze vyjádřit v maticovém tvaru

$$P(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \quad (6.33)$$

Pro konečné vyjádření maticového tvaru je nutné převést původní vektor $[t^3, t^2, t, 1]$ na vektor $[1, t, t^2, t^3]$. Z tohoto důvodu bude převedena i odvozená číselná matice. Konečné vyjádření maticového zápisu Bézierovy křivky bude ve tvaru

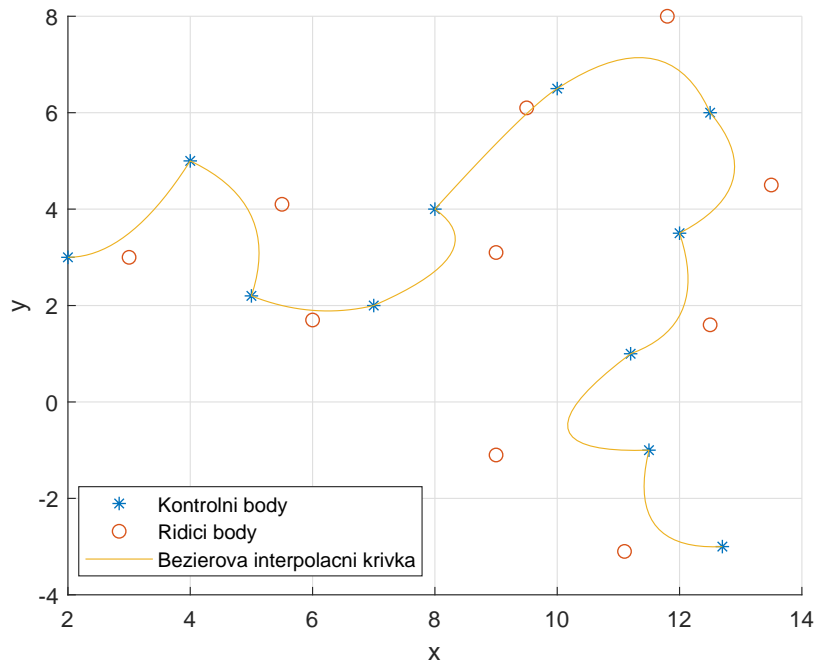
$$P(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}. \quad (6.34)$$

V obecném tvaru pak

$$P(t) = T \cdot M \cdot P_{points}, \quad (6.35)$$

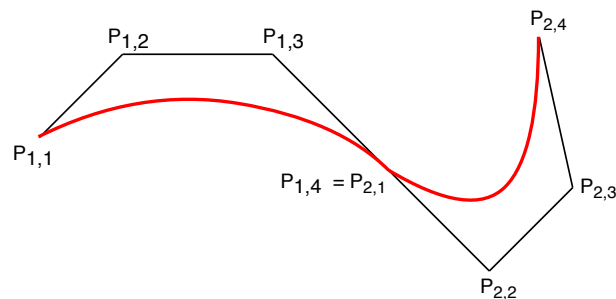
přičemž velikosti vektorů T a P_{points} budou rovny n a velikost matice M bude $n \times n$, kde n je počet zadaných bodů tvořící řídicí polygon.

Napojení jednotlivých křivek může být problematické z hlediska požadavku na hladkost výsledné křivky. Řídicí polygony jednotlivých křivek na sebe nemusí hladce navazovat. Tento případ je znázorněn v grafu 6.7, kde jednotlivé řídicí polygony na sebe hladce nenavazují a v pevných bodech tak můžou vznikat hrany.



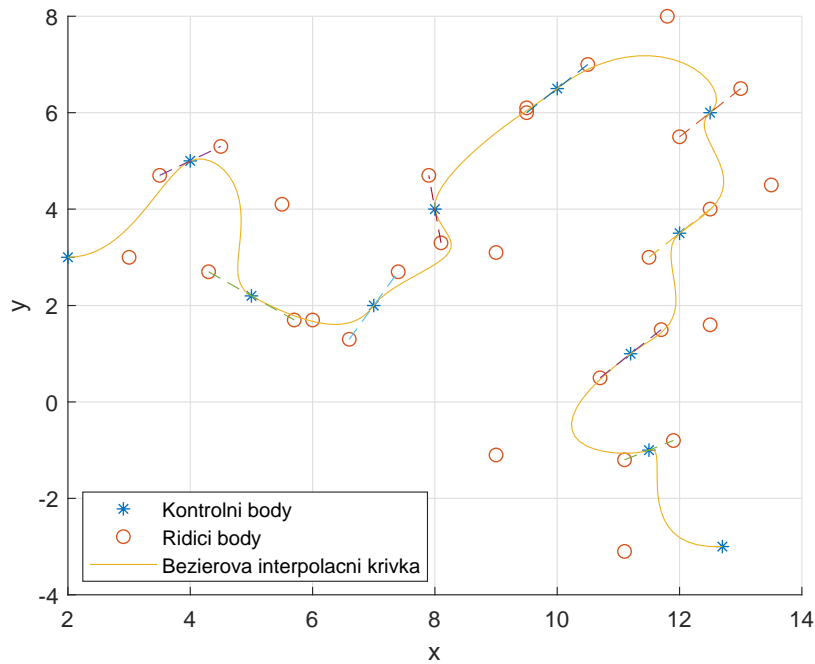
Obrázek 6.7: Interpolace 2D křivky Béziovými křivkami jejichž řídicí polygony na sebe hlace nenavazují (počet pevných bodů 11).

Navrhovaným řešením je ke každému kontrolnímu bodu přidat dva řídicí body, přičemž každý kontrolní bod je zároveň prvním nebo posledním bodem v řídicím polygonu nejbližších generovaných Béziových křivek. Přidané řídicí body budou přiřazeny po jednom ke každé nejbližší křivce a bude pro ně platit podmínka, že při proložení těchto dvou bodů přímkou, bude tato přímka procházet daným kontrolním bodem, kterému jsou tyto řídicí body přiřazeny. Zmíněné řešení bude vysvětlen na příkladu propojení dvou po sobě jdoucích křivek.



Obrázek 6.8: Napojení Béziových křivek s ohledem na zachování hladkosti výsledné křivky.

Na obrázku 6.8 je znázorněn způsob napojení dvou po sobě jdoucích řídicích polygonů, které jsou propojeny pevným bodem $P_{1,4}$ který je zároveň bodem $P_{2,1}$. Pro hladké propojení řídicích polygonů musí být volné body $P_{1,3}$ z prvního polygonu a $P_{2,2}$ z druhého v takové pozici, aby přímka, která vznikne jejich vzájemným propojením procházela pevným bodem $P_{1,4} = P_{2,1}$.



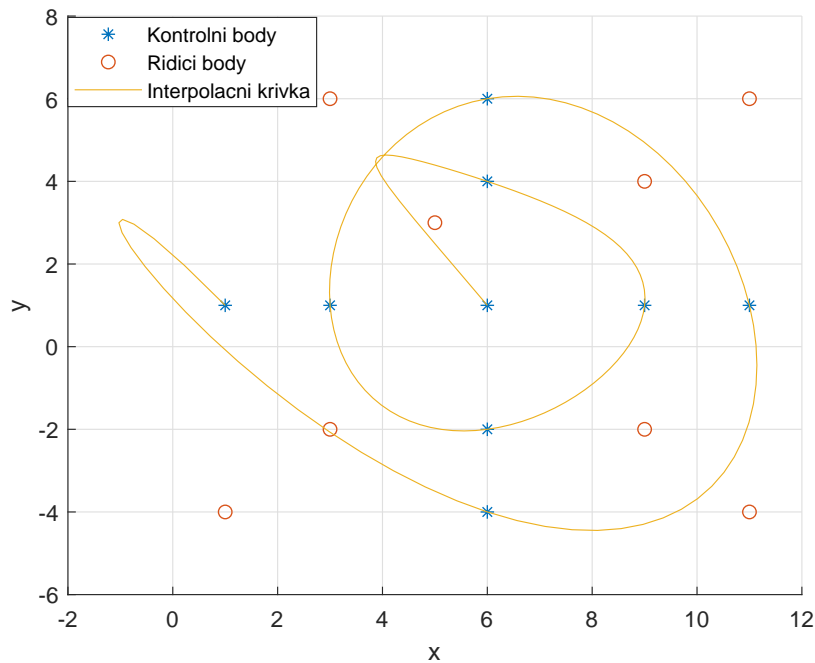
Obrázek 6.9: Interpolace 2D křivky Béziovými křivkami po přidání kontrolních bodů, které jsou propojeny přímkou (počet pevných bodů 11).

Na obrázku 6.9 je graf křivky z obrázku 6.7, která je doplněna o kontrolní body zajišťující hladkost vzájemných napojení. Při zachování vzájemného postavení těchto bodů zajišťující hladkost, lze pro libovolné natočení přímky spojující navržené body a k nim přiřazený pevný bod zachovat hladkost daného napojení křivek.

6.3 Zhodnocení a výběr vhodné metody

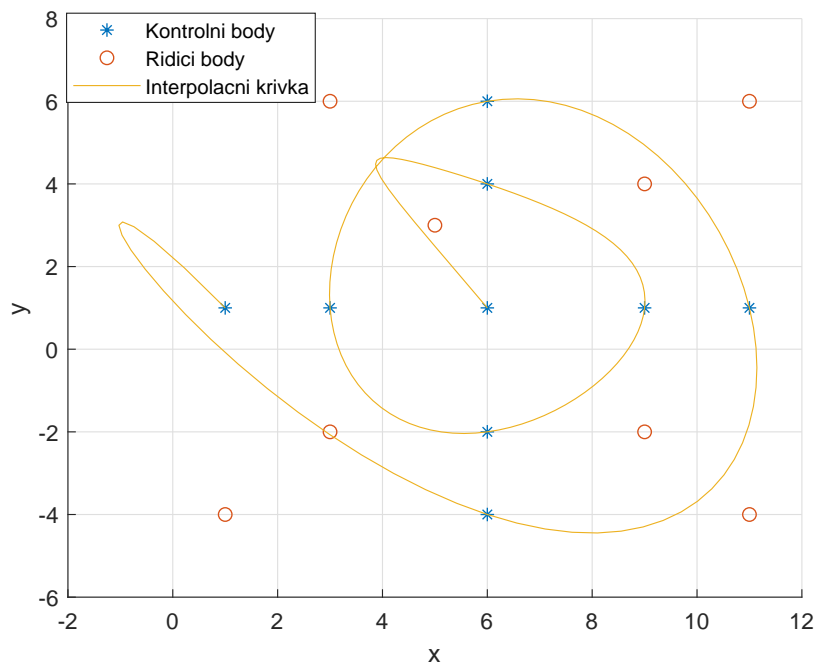
Výběr použité metody v návrhu systému byl proveden na základě požadavků na vhodný způsob tvarování křivky. Je požadováno, aby vysoký počet zadaných bodů neměl vliv na kvalitu vygenerované trajektorie. Dále je také cílem navrhnout křivku způsobem, aby se případně mohla jednoduše tvarovat, dle požadavků na trajektorii. Znázornění vygenerovaných křivek v prostoru pomocí jednotlivých metod jsou pro ilustraci dostupná v sekci 6.4.

První přístupem k tvorbě křivky, kdy byla uvažována křivka jako celek za použití klasického polynomu nastává problém v případě, kdy je použit polynom vysokého stupně. Ve výpočtu je použita inverze matice, jejíž výpočet pro polynom vysokého řádu je velmi špatně podmíněný a dochází tak k nepřesnostem ve výsledném tvaru polynomu. V grafu 6.10 je znázorněna výsledná aproximace metodou nejmenších čtverců, kde byl minimální stupeň polynomu $n = 9$ roven počtu pevných bodů.

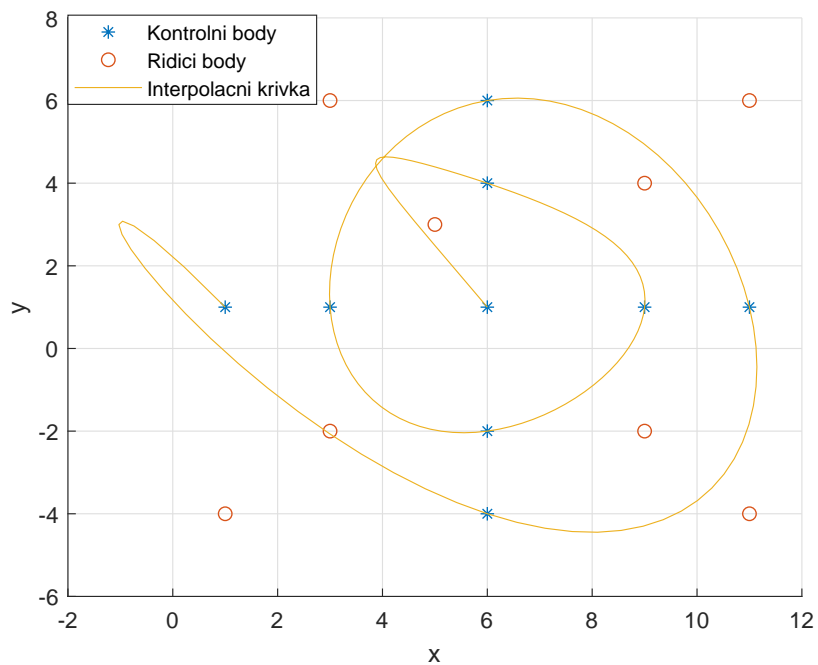


Obrázek 6.10: Interpolace 2D křivky rozšířenou metodou nejmenších čtverců pro řád polynomu 9 (počet pevných bodů 9).

Pro demonstraci řešení použití ortogonálních polynomů byla vybrána aproximace Čebyševovým polynomem 2. druhu. Z grafu (6.11) lze vidět výstupní křivka při řádu polynomu rovnému počtu zadaných kontrolních bodů $n = 9$.



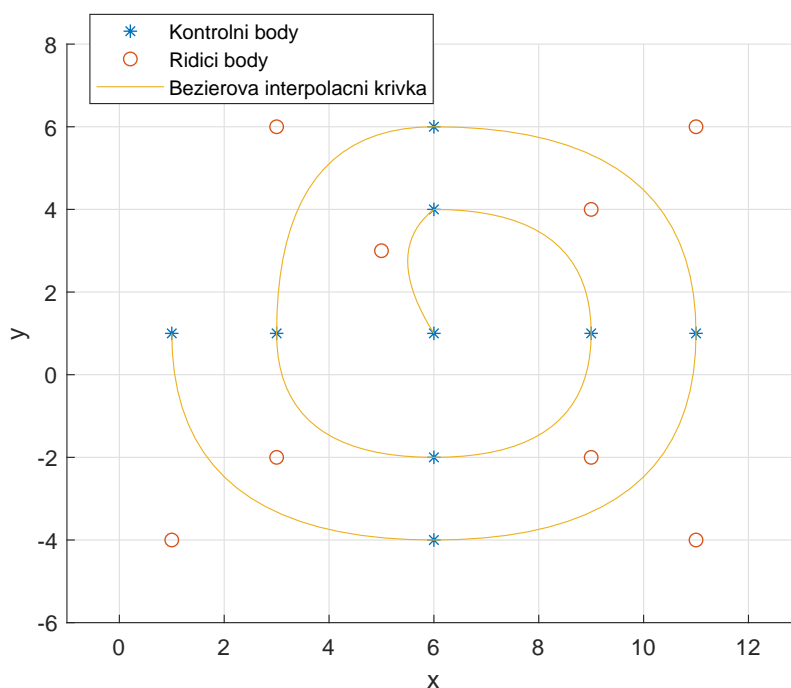
Obrázek 6.11: Interpolace 2D křivky rozšířenou metodou nejmenších čtverců pro řád Čebyševa polynomu 9 (počet pevných bodů 9).



Obrázek 6.12: Interpolace 2D křivky rozšířenou metodou nejmenších čtverců s ohledem na ortogonalitu pro řád polynomu 9 (počet pevných bodů 9).

V grafu 6.12 je znázorněna výsledná křivka při použití rozšířené metody nejmenších čtverců s ohledem na ortogonalitu použitých bazových vektorů. Z dosavadních grafů znázorňujících interpolaci bodů pomocí různě navržených polynomů lze vidět, že pro devět volných bodů všechny metody poskytují velmi dobré výsledné křivky pro daný počet bodů.

Druhým přístupem k tvorbě křivky je parametrizace pomocí Bézierových křivek. Z grafu (6.13) je vidět, že interpolace pomocí Bézierových křivek velmi dobře odpovídá požadovanému tvaru křivky.



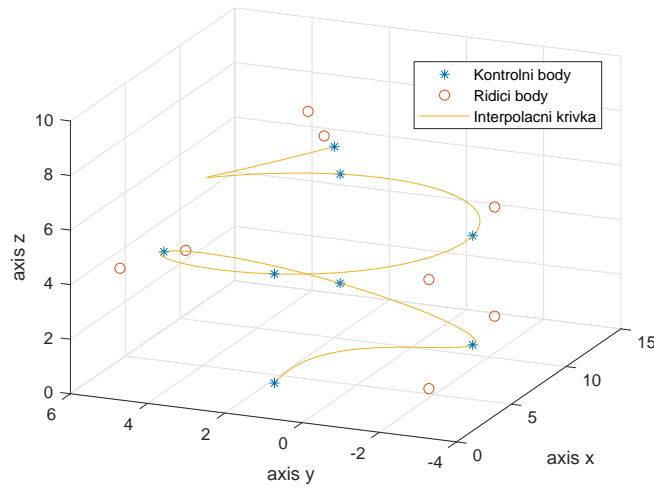
Obrázek 6.13: Interpolace 2D křivky Bézierovo křivkama (počet pevných bodů 9).

Z uvedených metod je pro návrh trajektorie nejvhodnější metoda aproximace ze zadaných bodů pomocí Bézierových křivek. Důvodem je nezávislost na množství zadaných bodů, jelikož se jedná o sérii propojených křivek, nikoliv o jeden polynom, kde je riziko, že úloha bude špatně podmíněná a bude snížena výsledná přesnost výpočtu. Toto riziko minimalizovalo použití ortogonálních vektorů tvořící ortogonální bazové podprostory.

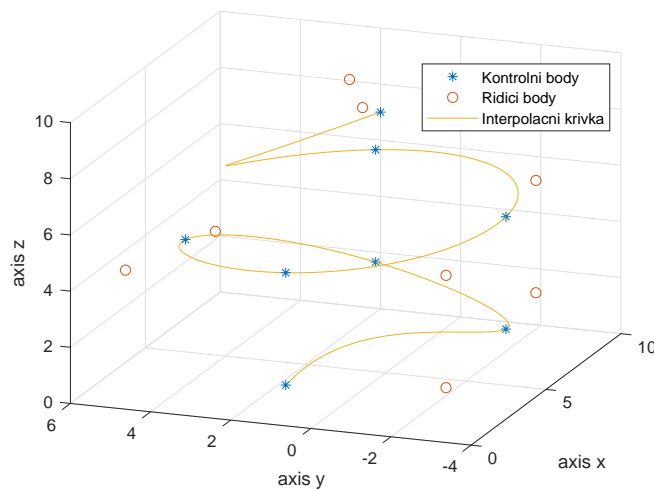
Výhodou použití Bézierových křivek je také vzájemná nezávislost jednotlivých křivek. Pokud se změní jeden řídicí bod pro daný segment (Bézierovu křivku), neovlivní se tím průběh ostatních segmentů. Také může být výhodou skutečnost, že pomocí Bézierových křivek lze navrhnout hladké křivky ale také křivky obsahující hrany a geometrické vzory tvořené přímkami.

6.4 Znázornění křivek v prostoru

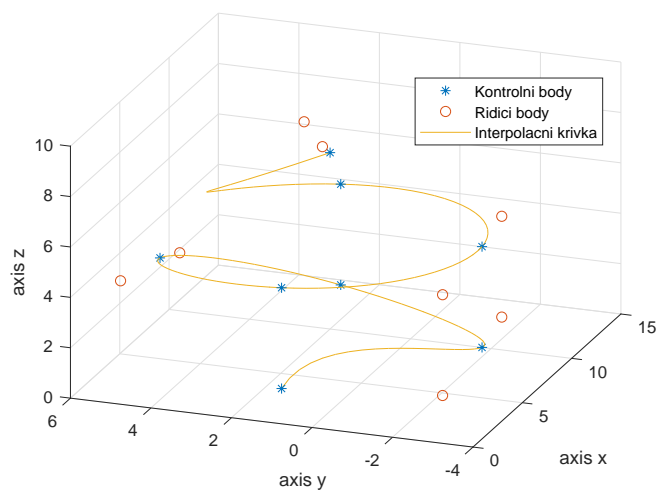
V následující sekci jsou pro ilustraci uvedeny znázornění vygenerovaných křivek v prostoru, kdy byla v řešení uvažována třetí dimenze a tedy se výpočet rozšířil o řešení pro souřadnice z .



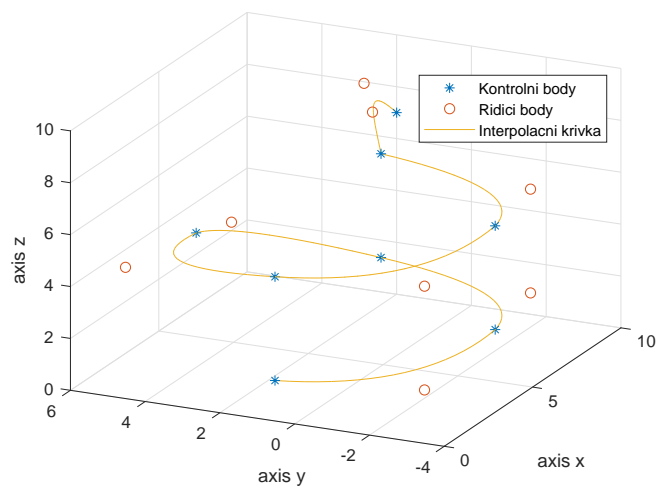
Obrázek 6.14: Interpolace 3D křivky rozšířenou metodou nejmenších čtverců pro řád polynomu 8 (počet pevných bodů 8).



Obrázek 6.15: Interpolace 3D křivky rozšířenou metodou nejmenších čtverců pro řád Čebyševa polynomu 8 (počet pevných bodů 8).



Obrázek 6.16: Interpolace 3D křivky rozšířenou metodou nejmenších čtverců s ohledem na ortogonalitu pro řád polynomu 8 (počet pevných bodů 8).



Obrázek 6.17: Interpolace 3D křivky Béziovými křivkami (počet pevných bodů 8).

6.5 Parametry pohybu po trajektorii

Pro definování pohybu po trajektorii je kromě polohy nutné získat také její vzdálenost, rychlost, kterou se koncový efektor bude po křivce pohybovat a také s jakým zrychlením a jerkem. Výpočtem a následným porovnáním s maximálními možnými hodnotami tak bude možné ověřit realizovatelnost výsledné trajektorie a plně definovat navržený pohyb. Dále je potřeba definovat návrhové požadavky na pohybové vlastnosti, které budou důležité pro možné provedení pohybu po navržené trajektorii.

6.5.1 Požadavky na pohybové vlastnosti

Pohybovými vlastnostmi je myšlena rychlost, zrychlení a jerk. Je-li uvažována poloha $s(t)$, pak rychlost je možné vyjádřit jako časovou derivaci polohy

$$v(t) = \frac{d}{dt}s(t). \quad (6.36)$$

Návrhovým požadavkem je spojitost $v(t)$ a omezení na maximální možnou rychlost $v(t) \leq v_{max}$. Další vlastností je zrychlení, které je časovou derivací rychlosti

$$a(t) = \frac{d}{dt}v(t). \quad (6.37)$$

Návrhovým požadavkem na zrychlení je spojitost a omezení na maximální a minimální zrychlení $|a(t)| \leq a_{max}$. Časovou derivací zrychlení je jerk

$$j(t) = \frac{d}{dt}a(t). \quad (6.38)$$

Návrhovým požadavkem na jerk již spojitost není, pouze omezení $|j(t)| \leq j_{max}$, přičemž j_{max} je návrhovým parametrem, který bude pokud možno minimální.

6.5.2 Výpočet délky křivky

Vzhledem ke způsobu generování výsledné křivky je možné vyjádřit výpočet délky analyticky pomocí integrálu vyjádřeného pro každou Bezierovu křivku obsaženou ve výsledné trajektorii. Vyjádření integrálu pro jednu Bezierovu křivku v obecném tvaru

$$\int_a^b f(t)dt = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{b-a}{2} \cdot t + \frac{a+b}{2}\right) dt, \quad (6.39)$$

který bude vyjádřen pomocí následující aproximace

$$\int_a^b f(t)dt \approx \frac{b-a}{2} \sum_{i=1}^n w_i \cdot f\left(\frac{b-a}{2} \cdot t_i + \frac{a+b}{2}\right), \quad (6.40)$$

kde n je počet bodů použitých k aproximaci, w_i, t_i jsou tabulkové hodnoty pro dané n . Bude-li uvažován počet bodů $n = 2$, tedy pouze krajní body, tabulkové hodnoty budou $t_i = \pm \frac{1}{\sqrt{3}}$ a $w_i = 1$. Po dosazení bude výsledný tvar aproximace

$$\int_a^b f(t)dt \approx \frac{b-a}{2} \left[f\left(\frac{b-a}{2} \cdot \frac{1}{\sqrt{3}} + \frac{a+b}{2}\right) + f\left(\frac{b-a}{2} \cdot \frac{-1}{\sqrt{3}} + \frac{a+b}{2}\right) \right]. \quad (6.41)$$

Dále bude definována funkce $f(t)$, která bude uvnitř integrálu. Pro dvoudimenzionální případ bude funkce ve tvaru

$$f(t) = \sqrt{\left(\frac{dx(t)}{dt}\right)^2 + \left(\frac{dy(t)}{dt}\right)^2}, \quad (6.42)$$

a pro třidimenzionální případ je nutné do funkce navíc zahrnout směr z

$$f(t) = \sqrt{\left(\frac{dx(t)}{dt}\right)^2 + \left(\frac{dy(t)}{dt}\right)^2 + \left(\frac{dz(t)}{dt}\right)^2}. \quad (6.43)$$

Jednotlivé derivace jsou vyjádřeny dle použité metody pro aproximaci křivky. Například pro polynom třetího stupně $x(t) = a_0 + a_1x + a_2x^2 + a_3x^3$ platí

$$\frac{dx(t)}{dt} = a_1 + a_2x + a_3x^2. \quad (6.44)$$

Pro Bézierovu křivku popsanou rovnicí (6.30) bude derivace pro směr x následovná

$$\begin{aligned} \frac{dx(t)}{dt} = & P_1 \cdot (-3 + 6 \cdot t - 3t^2) + P_2 \cdot (3 - 12 \cdot t + 9 \cdot t^2) + \\ & + P_3 \cdot (6 \cdot t - 9 \cdot t^2) + P_4 \cdot (3 \cdot t^3). \end{aligned} \quad (6.45)$$

Derivace v ostatních směrech budou počítány analogicky. Výpočet délky bude tedy vyjádřen aproximací (6.41), do které za každé $f(\dots)$ dosadíme funkce (6.42) nebo (6.43) dle požadované dimenze.

6.5.3 Výpočet pohybových vlastností

Výpočet rychlosti, zrychlení a jerku pro Bézierovy křivky lze provést analyticky. Při implementaci řešení byla pro zjednodušení použita dostatečně přesná aproximace výpočtu těchto veličin.

Analytický výpočet

Vyjádření analytického výpočtu pohybových vlastností bude uvedeno pro jeden segment trajektorie, tedy pro jednu Bézierovu křivku. Ostatní segmenty jsou počítány analogicky.

Odvození výpočtu rychlosti pohybu po Bézierovo křivce je vyjádřen jako derivace $P(t)$ ze vztahu 6.34. Bude tedy uvažována křivka definovaná čtyři body P_1, P_2, P_3, P_4 . Vztah 6.34 bude po roznásobení ve tvaru

$$P(t) = P_1 \cdot (1 - 3t + 3t^2 - t^3) + P_2 \cdot (3t - 6t^2 + 3t^3) + P_3 \cdot (3t^2 - 3t^3) + P_4 \cdot t^3. \quad (6.46)$$

Zderivováním $P(t)$ podle parametru t získáme vyjádření rychlosti v čase

$$\dot{P}(t) = P_1 \cdot (-3 + 6t - 3t^2) + P_2 \cdot (3 - 12t + 9t^2) + P_3 \cdot (6t - 9t^2) + P_4 \cdot 3t^2. \quad (6.47)$$

Výpočet zrychlení a jerku bude proveden analogicky jako derivace podle parametru t z předpisu pro rychlost $\dot{P}(t)$ pro zrychlení

$$\ddot{P}(t) = P_1 \cdot (6 - 6t) + P_2 \cdot (-12 + 18t) + P_3 \cdot (6 - 18t) + P_4 \cdot 6t, \quad (6.48)$$

a derivace podle parametru t z předpisu pro zrychlení $\ddot{P}(t)$ pro výpočet jerku

$$\ddot{\dot{P}}(t) = -6P_1 + 18P_2 - 18P_3 + 6P_4. \quad (6.49)$$

Z jednotlivých polynomů, které definují rychlost, zrychlení a jerk pohybu po křivce je vidět, že při nadefinování křivky pomocí čtyř bodů se pro rychlost jedná o polynom druhého stupně, a průběh rychlosti tedy bude spojitý a hladký. Polynom vyjadřující zrychlení je již prvního stupně. Spojitost je zde zachována, ale průběh již nebude hladký. S třetí derivací, jerkem, již průběh ztrácí i spojitost. Pro hladký průběh rychlosti je tedy potřeba, aby byla Bézierova křivka definovaná alespoň čtyřmi body. Pokud je požadavkem hladký průběh zrychlení je potřeba minimálně bodů pět.

Aproximace použitá pro implementaci řešení

Výpočet pohybových vlastností je závislý na pojetí parametru t , kterým je parametrizováno matematické vyjádření jednotlivých křivek. V případě, že parametrem t je uvažován čas, ve kterém se koncový efektor musí dostat do daného bodu, bude výpočet rychlosti založen na vztahu pro časovou derivaci v dvoudimenzionálním případě

$$v(t) = \frac{\sqrt{(x(t_i) - x(t_{i-1}))^2 + (y(t) - y(t_{i-1}))^2}}{t_i - t_{i-1}}, \quad (6.50)$$

pro třídimenzionální případ je nutné obdobně jako u výpočtu délky rozšířit vztah o směr z .

$$v(t) = \frac{\sqrt{(x(t_i) - x(t_{i-1}))^2 + (y(t) - y(t_{i-1}))^2 + (z(t) - z(t_{i-1}))^2}}{t_i - t_{i-1}}. \quad (6.51)$$

Takto vyjádřená rychlost může obsahovat skoky, které nabývají hodnot vyšších než je fyzikálně realizovatelné. Stejným způsobem je ovlivněno také zrychlení a jerk. Důvodem těchto skoků je obtížnost určení optimálního času, ve kterém se do daného bodu koncový efektor dostane.

Z tohoto důvodu byl navržen druhý přístup výpočtu rychlosti pohybu po dané trajektorii. Jako parametr t v tomto případě bude uvažováno pouze pořadí daných bodů, které je klíčové pro správné sestavení Bezierových křivek. V tomto případě je nutné vyjádřit nejen rychlost v každém bodě, ale také čas. Řešením byl navržen generátor požadované rychlosti, na základě které byl následně dopočten čas v každém bodě.

Při návrhu generátoru byl uvažován na začátku trajektorie plynulý nárůst rychlosti až na požadovanou konstantní hodnotu, která bude volitelná, a před koncem trajektorie naopak plynulý pokles rychlosti zpět na nulu. Plynulý nárůst a pokles rychlosti byl získán využitím Bezierových křivek, kdy byla jako počáteční bod uvažována rychlost nula a koncový bod zvolená konstantní hodnota rychlosti. Stejným způsobem byl vyjádřen také pokles.

Vyjádření vztahu pro výpočet zrychlení v daném bodě bude pro oba způsoby výpočtu rychlosti stejný. Jak již bylo poznamenáno ve výpočtu rychlosti, výpočet byl proveden numericky ze vztahu

$$a(t) = \frac{v(t) - v(t_{i-1})}{t_i - t_{i-1}}. \quad (6.52)$$

Numerický výpočet jerku, tedy derivace zrychlení, byl vypočten ze vztahu

$$j(t) = \frac{a(t) - a(t_{i-1})}{t_i - t_{i-1}}. \quad (6.53)$$

7 Kinematický model manipulátoru

Důležitou součástí generátoru požadované trajektorie je modul, který obsahuje kinematický simulační model, který definuje závislosti mezi obecnými polohami, rychlostmi a zrychleními bez ohledu na síly či silové momenty, které tyto závislosti ovlivňují [21].

Pro návrh trajektorie pohybu je důležité znát přesné hranice, kde je robotický manipulátor schopný pracovat. Hranice maximálního dosahu na základně technických parametrů je definována v v sekci 4.2. V mnoha případech však může nastat situace, že ani této hranice nebude možné dosáhnout, protože že se robot přiblíží k singulárním polohám, kde hrozí nebezpečí destrukce nevhodným řízením. Dále mají také vliv na maximální možnou rychlost a zrychlení v této oblasti.

Z tohoto důvodu je nutné definovat pracovní prostor, kam je možné trajektorii umístit, případně ověřit, zda je možné tento pohyb realizovat. Pro účely testování prototypu aplikace byla pro vyšetření pracovního prostoru uvažována konstantní orientace koncového efektoru (constant orientation workspace).

Následující text zabývající se modelováním kinematického modelu manipulátoru vychází z práce [21].

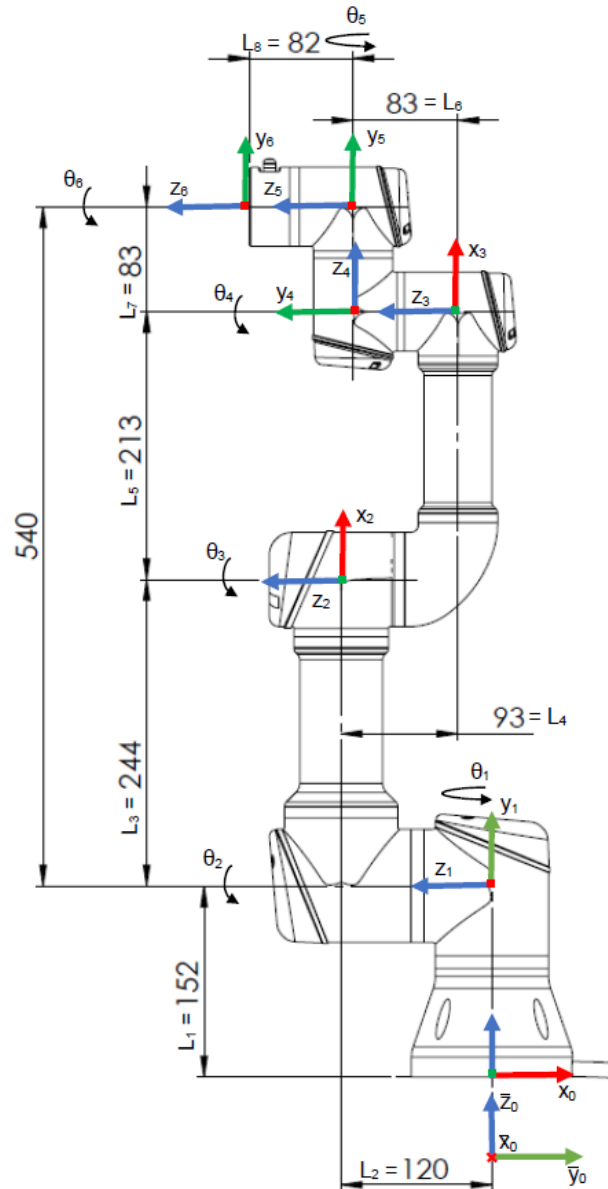
7.1 Denavit-Hartenbergova úmluva (D-H)

Denavit-Hartenbergova úmluva, neboli D-H úmluva, slouží k popisu geometrie sériových manipulátorů, kde lze rekurzivně definovat souřadné systémy reprezentující jednotlivá ramena manipulátoru a jejich vzájemnou polohovou transformaci [21]. Konstantní geometrické parametry a kloubové souřadnice Θ definují polohové transformace souřadných systémů jdoucích za sebou. Mezi geometrické parametry patří geometrický tvar ramen, kloubů a vzájemná konfigurace ramen a kloubů.

Pro získání geometrického modelu je nutné si postupně vyjádřit geometrické parametry manipulátoru, nebo-li D-H parametry. Mezi tyto parametry patří d_i , θ_i , a_i a α_i , přičemž θ_i je proměnná definující pohyb kloubu a d_i , a_i , α_i jsou konstanty, které definují geometrické uspořádání ramene Link i .

Od světového souřadného systému byly postupně odvozeny souřadné sys-

témy v jednotlivých kloubech, dle pravidel D-H úmluvy. Pro každou skupinu D-H parametrů byla uvažována dvojice ramen manipulátorů, značí se Link $i-1$ a Link i , která jsou spojena kloubem Joint i . Na obrázku 7.1 jsou znázorněny výsledné souřadné systémy a parametry θ_i , které definují rotační pohyb kloubů.



Obrázek 7.1: Grafické znázornění odvozených souřadných systémů. Zdroj původního obrázku sloužící k zobrazení manipulátoru [8].

Výsledné geometrické parametry získané D-H úmluvou jsou zobrazeny v tabulce 7.1. Celkový postup odvození řešení s bližším popisem jednotlivých kroků úmluvy je dostupný v [21].

i	d_i	θ_i	a_i	α_i	θ_{home}
1	L_1	θ_1	0	$\frac{\pi}{2}$	$-\frac{\pi}{2}$
2	L_2	θ_2	L_3	0	$\frac{\pi}{2}$
3	$-L_4$	θ_3	L_5	0	0
4	L_6	θ_4	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$
5	L_7	θ_5	0	$\frac{\pi}{2}$	π
6	L_8	θ_6	0	0	0

Tabulka 7.1: Geometrické parametry manipulátoru s vyjádřenými kloubovými souřadnicemi základny θ_{home} .

Dále bude vyjádřena matice přechodu T_i^{i-1} , jejíž odvození je na základě postupu v [21], a která definuje transformaci přechodu z O_i do souřadné soustavy O_{i-1} .

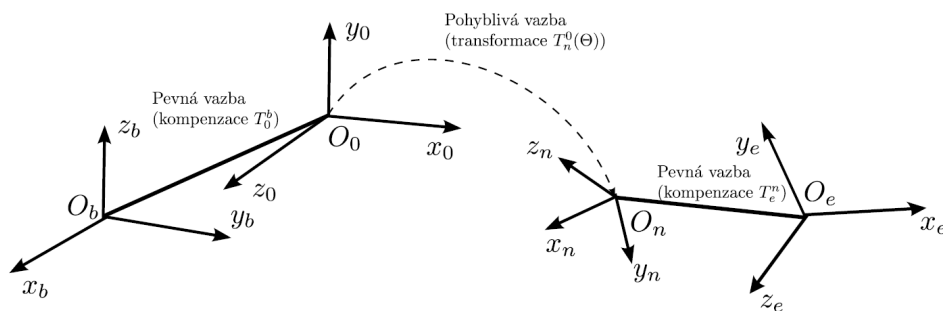
$$\begin{aligned}
 T_i^{i-1}(\Theta) &= Trans(z, d_i) \cdot Rot(z, \theta_i) \cdot Trans(x, a_i) \cdot Rot(x, \alpha_i) = \\
 &= \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \cos(\alpha_i) & \sin(\theta_i) \cos(\alpha_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \cos(\alpha_i) & -\cos(\theta_i) \sin(\alpha_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7.1)
 \end{aligned}$$

7.2 Inverzní kinematický popis manipulátoru

Pro účely ověřování realizovatelnosti navržené trajektorie je potřeba navrhnout inverzní kinematický problém, pro zjištění hodnot kloubových souřadnic θ_i , které dále budou vyjadřovány ve vektoru kloubových souřadnic Θ z navržených zobecněných souřadnic x, y, z obsaženy ve vektoru zobecněných souřadnic X .

7.2.1 Kompenzace polohy základny a koncového efektoru

Pro kompenzaci umístění manipulátoru a polohy nástroje umístěného na koncovém efektoru je třeba vyjádřit souřadné systémy základny (rámu) manipulátoru a koncového efektoru. Na obrázku 7.2 je znázorněn princip kompenzací manipulátoru, ze kterého budou výsledné kompenzace vycházet.



Obrázek 7.2: Princip kompenzací manipulátoru [21].

Výsledná matice přechodu $T_e^b(\Theta)$, která závisí na poloze kloubových souřadnic o rozšířená o kompenzace polohy základny a koncového efektoru bude ve tvaru

$$T_e^b(\Theta) = T_0^b \cdot T_6^0 \cdot T_e^6 = \left[\begin{array}{ccc|c} R_e^b & & & r_{b,e}^b \\ \hline 0 & 0 & 0 & 1 \end{array} \right], \quad (7.2)$$

kde $R_e^b = [x_e^b, y_e^b, z_e^b]$ je matice rotace a $r_{b,e}^b = O_e^b - O_b^b = O_e^b$ je translační vektor souřadné soustavy koncového efektoru vzhledem k souřadné soustavě vzhledem k základně [21].

7.2.2 Polohové závislosti

Polohu kloubových souřadnic Θ lze pro danou polohu koncového efektoru X stanovit dle vztahu 7.3 [21].

$$\Theta = G^{-1}(X), \quad (7.3)$$

kde $G()$ je funkcí geometrických omezení. Vyjádření vztahů popisujících 7.3 je obecně komplikovaný problém z důvodu nejednoznačnosti, nerozhodnutelnosti a také kvůli nutnosti řešit inverzi G^{-1} , kde se vykytují součty násobků a mocnin členů transformační matice $T_e^b(\Theta)$ ze vztahu 7.2. Nejednoznačnost je daná existencí více izolovaných řešení a také existencí nekonečně mnoha řešení v singulárních polohách. Nerozhodnutelnost vyplývá z nutnosti určit, které řešení odpovídá aktuální konfiguraci manipulátoru.

Výpočet polohových závislostí byl proveden přímým analytickým řešením, které je možno uplatnit na jednoduché architektury manipulátorů. Přímé analytické řešení je založeno na zkušenostech v řešení modelování pohybových vlastností robotů. Z tohoto důvodu byl algoritmus pro výpočet řešení převzat od [6] a je dostupný v programové části na CD.

7.2.3 Závislosti rychlosti a zrychlení

Pro definování úplného kinematického popisu manipulátoru je nutné definovat závislost rychlosti a zrychlení kloubových a zobecněných souřadnic pomocí inverzní okamžité kinematické úlohy. Jednotlivé vztahy lze vyjádřit následovně [21].

$$\begin{aligned}\dot{\Theta} &= \frac{\partial G(X)}{\partial X} \cdot \dot{X} = J_A^{-1}(X) \cdot \dot{X}, \\ \ddot{\Theta} &= \dot{J}_A^{-1}(X) \cdot \dot{X} + J_A^{-1}(X) \cdot \ddot{X},\end{aligned}\tag{7.4}$$

kde vztah

$$J_A^{-1}(X) = \frac{\partial G(X)}{\partial X},\tag{7.5}$$

představuje inverzní analytický jakobián, který závisí na aktuální poloze manipulátoru X . Časové derivace zobecněných souřadnic X lze pro obecný manipulátor s n klouby vyjádřit následujícím způsobem [21].

$$\begin{aligned}\dot{X} &= [\dot{R}_e^b \ \dot{O}_e^b]^T, \\ \ddot{X} &= [\ddot{R}_e^b \ \ddot{O}_e^b]^T,\end{aligned}\tag{7.6}$$

kde R_e^b je matice rotace a O_e^b je translační vektor souřadné soustavy koncového efektoru vzhledem k souřadné soustavě základny.

Analytický jakobián J_A je možné získat přímou derivací funkce $G()$ podle kloubových souřadnic Θ , avšak tímto způsobem by implementovaný algoritmus získal přílišnou složitost vzhledem k tomu, že její přímá derivace vede na složité a mnohačlenné vztahy.

Jinou alternativou výpočtu je předpoklad systematické geometrické metodologie výpočtu translační rychlosti \dot{O}_e a úhlové rychlosti ω_e koncového efektoru v závislosti na rychlosti kloubových souřadnic $\dot{\Theta}$ manipulátoru [21] (způsob výpočtu popsán v [19]).

Dle vztahů definovaných v [21] (str. 20), které definují vztah mezi derivací matice rotace \dot{R}_e a vektorem úhlové rychlosti ω_e a zrychlením $\dot{\omega}_e$, budou uvažovány derivace zobecněných souřadnic v následujícím tvaru.

$$\begin{aligned}\dot{X} &= \begin{bmatrix} \dot{O}_e^b \\ \dot{\omega}_e^b \end{bmatrix} \\ \ddot{X} &= \begin{bmatrix} \ddot{O}_e^b \\ \ddot{\omega}_e^b \end{bmatrix}\end{aligned}\tag{7.7}$$

Inverzní okamžitou úlohu pro obecný manipulátor s n klouby lze poté vyjádřit jako

$$\begin{aligned}\dot{\Theta} &= J^{-1}(X) \cdot \dot{X}, \\ \ddot{\Theta} &= \dot{J}^{-1}(X) \cdot \dot{X} + J^{-1}(X) \cdot \ddot{X},\end{aligned}\tag{7.8}$$

kde $J^{-1}(X)$ je inverzní kinematický jakobián.

Algoritmus výpočtu řešení inverzní okamžité úlohy byl převzat od [6] a je dostupný v programové části na DVD.

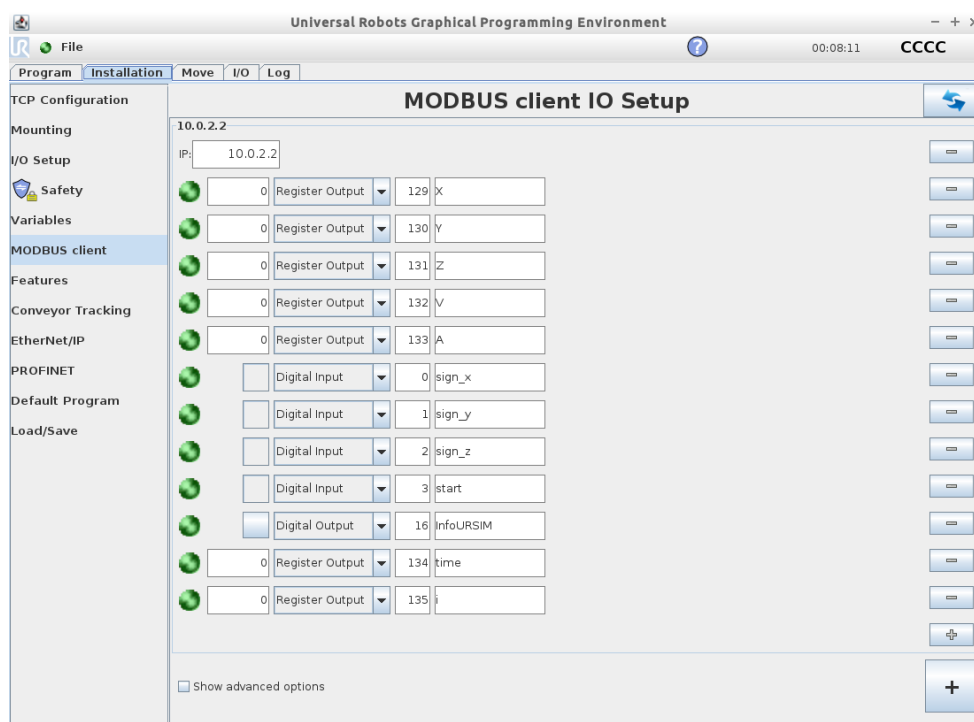
8 Implementace systému řízení

8.1 PolyScope

Uživatelské prostředí PolyScope je určeno pro vlastní řízení robota. Program v tomto prostředí byl navržen tak, aby bylo možné pomocí komunikačního protokolu Modbus získávat postupně souřadnice určující polohy koncového efektoru v aktuálním čase. Implementace byla provedena s pomocí [3].

8.1.1 Konfigurace robota

Před sestavením programu definující instrukce je nutné nejprve nastavit konfiguraci komunikace modbus a nastavit rovinu, po které se robot bude pohybovat.

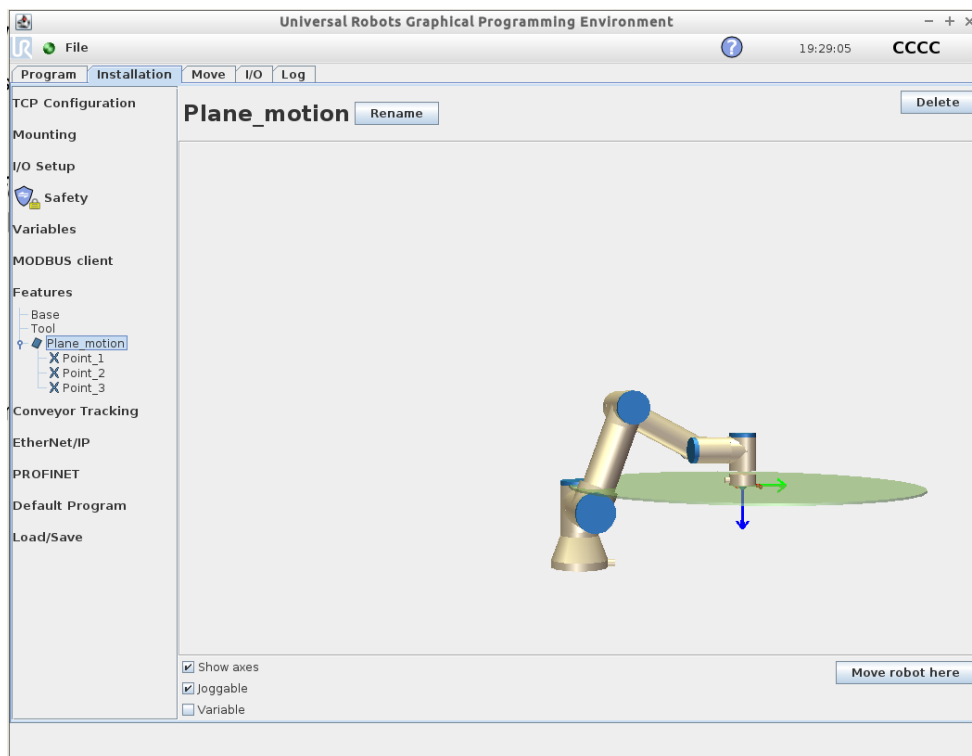


Obrázek 8.1: Nastavení komunikace pomocí protokolu Modbus.

Pro posílání souřadnic byly jednotlivé hodnoty rozděleny na registry, které obsahovaly absolutní hodnoty souřadnic a digitální vstupy, které defi-

nují znaménka příslušné ke každé hodnotě. Definované signály se přijímají nebo odesílají s frekvencí 10Hz.

Pro komunikaci mezi programy byl využíván digitální vstup start, který udává, zda je spuštěno dávkování dat z řídicího programu (v REXYGENu). Digitální výstup InfoURSIM poskytuje řídicímu programu informaci o tom, zda je robot připraven na řízení pohybu. Signál i označuje pořadí každého příchozího bodu pro ověření, zda již byl bod použit nebo zda se jedná o nový bod, do kterého robot zahájí pohyb.



Obrázek 8.2: Nastavení roviny.

Definování roviny v programu PolyScope je navrženo následujícím způsobem, který je zobrazen na obrázku 8.2. Pro každou rovinu je potřeba definovat tři body, přičemž první z těchto bodů udává střed souřadného systému roviny, druhý udává kladný směr osy y a třetí bod je při zachování nulového sklonu určen libovolně v rovině. V případě, že je požadován náklon roviny, je třeba nastavit souřadnici z buď ve druhém nebo ve třetím bodě.

Při sestavení série instrukcí je možné definovanou rovinu (nebo více rovin) použít pro řízení pohybu, od které se poté odvíjí souřadný systém požadované polohy.

8.1.2 Program jako série instrukcí

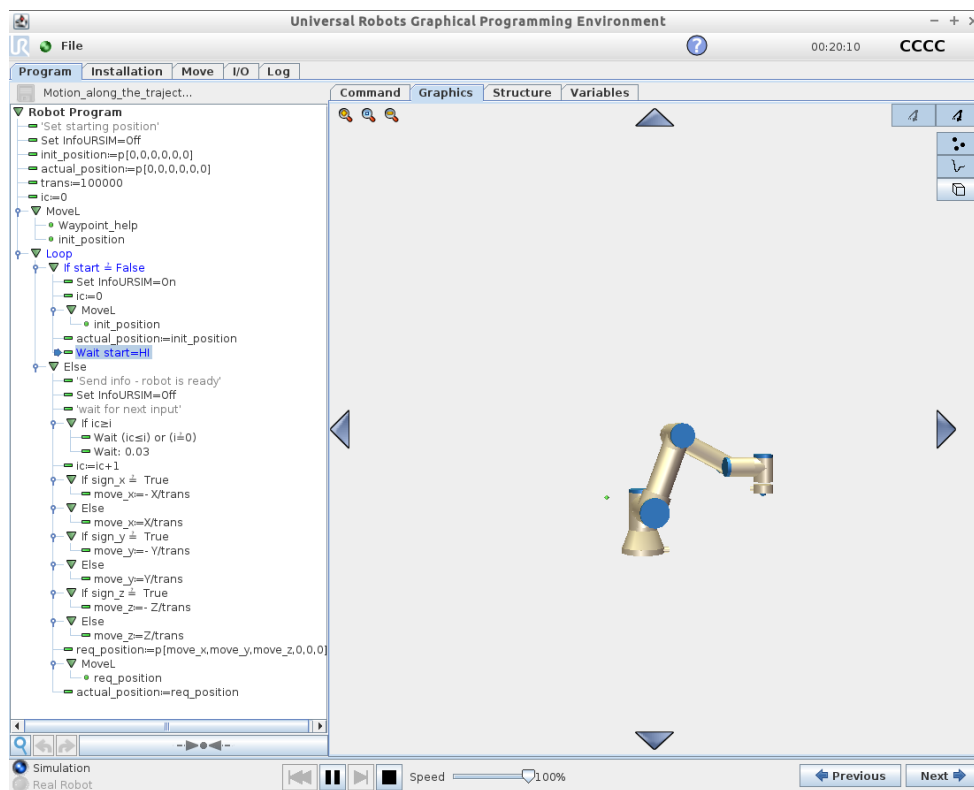
Při sestavení ovládacího programu byl nastaven přesun manipulátoru do počátku souřadného systému definované roviny, aby manipulátor pro každou navrženou trajektorii začínal pohyb ve stejném bodě.

V dalším kroce je definována nekonečná smyčka, která obsahuje celý proces řízení manipulátoru. Po vstupu do smyčky vyšle program signál (InfoURSIM), že je již robot připraven.

Na začátku smyčky se vždy ověřuje, zda řídicí program vysílá data o požadované trajektorii. Pokud ne, program se v tomto kroce zastaví a čeká na signál o spuštění (start).

Po spuštění dávkování dat začne robot vykovánat pohyb do určené polohy. Pokud se robot do určené polohy dostane dřív, než předpokládá řídicí program, smyčka se opět zastaví a robot čeká na další instrukce.

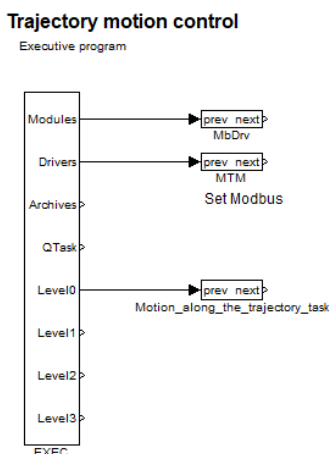
Po skončení dávkování dat řídicí program vyšle signál o ukončení vysílání dat. Robot po přijetí signálu ukončí pohyb po trajektorii a vrátí se do výchozí polohy, kde čeká na nová data.



Obrázek 8.3: Navržený program jako série definovaných instrukcí v programu PolyScope.

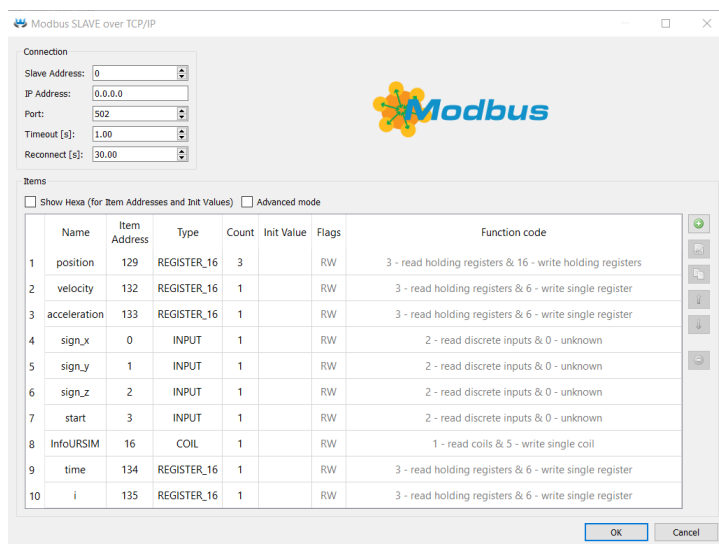
8.2 REXYGEN

Řídicí program byl implementován v programu REXYGEN. Zde bylo cílem navrhnout program, který bude pomocí protokolu Modbus komunikovat s ovládacím programem (PolyScope). Implementace v REXYGENU byla provedena pomocí [5].



Obrázek 8.4: Schéma zapojení v REXYGENu (exec).

Výkonná jednotka je znázorněna na obrázku 8.4. Zde je nastavena komunikace a napojení na task soubor s hlavním řídicím programem.

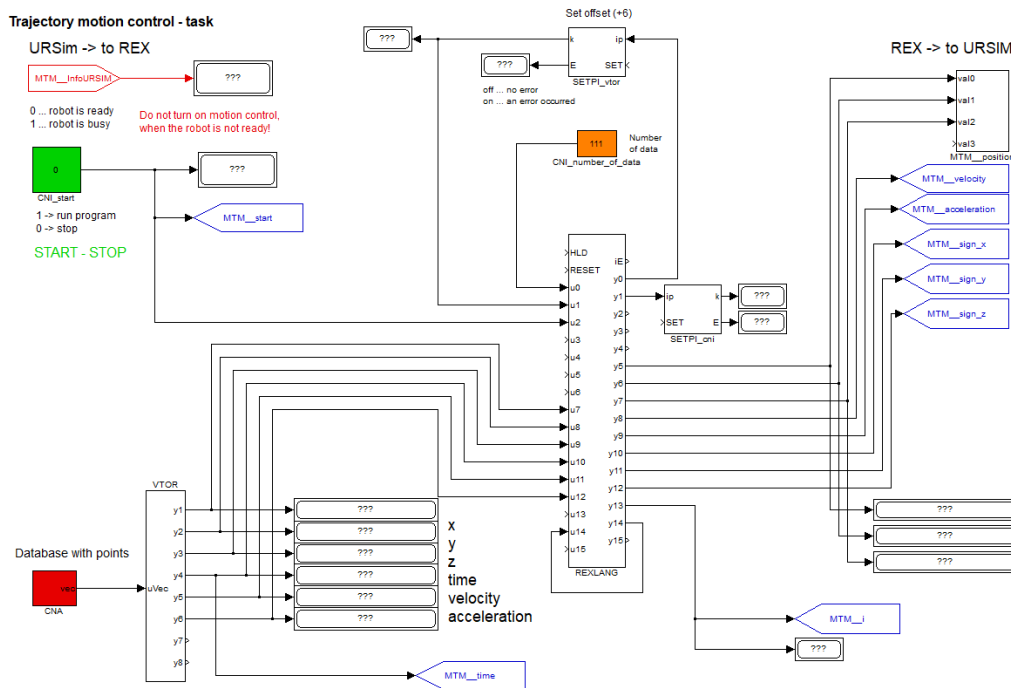


Obrázek 8.5: Nastavení signálů pro komunikaci pomocí protokolu Modbus.

Na obrázku 8.5 lze vidět nastavení komunikačního protokolu Modbus pro slave stanici, kde jsou znázorněny všechny signály zajišťující komunikaci s

ovládacím programem. První tři signály reprezentující souřadnice x , y a z jsou definovány v registru s názvem position. Souřadnice jsou posílány pomocí jednoho registru z důvodu nutnosti odesílání těchto zpráv najednou, jinak by ovládací program nemusel vždy aktualizovat všechny souřadnice správně a došlo by k nežádoucím chybám v definování polohy koncového efektoru.

Schéma zapojení v REXYGENu je znázorněno na obrázku 8.6. Jsou zde obsaženy bloky zajišťující komunikaci Modbus protokolu, blok pro importování dat CNA, také volně programovatelný blok REXLANG nebo bloky SETPI pro vzdálené nastavení parametrů uvnitř jiných bloků.



Obrázek 8.6: Schéma zapojení v REXYGENu (task).

Data k navržené trajektorii byla vyexportována z generátoru trajektorie ve formátu CSV, která byla do řídicího programu importována pomocí bloku CNA. První blok REXLANG řídí celkové dávkování dat na základě definovaného času pro každý bod.

Po spuštění programu je důležité počkat na signál, že je robot připravený na řízení (InfoURSIM předá hodnotu 0). Dávkování dat trajektorie se spustí blokem CNI_start. Po spuštění řízení je odeslán signál o startu dávkování dat a program začne automaticky posílat data na základě definovaného času. Po zaslání veškerých dat se vyšle signál o konci dávkování dat a komunikace se zastaví.

8.3 Generátor trajektorie

Prototyp aplikace generátoru trajektorie byl vytvořen s cílem umožnit jednoduchý grafický návrh požadované trajektorie, generování rychlosti, zrychlení a jerku pro navržený pohyb po křivce, ověření fyzikální realizovatelnosti navržené křivky a exportování dat do souboru pro další použití.

8.3.1 Založení projektu

Po spuštění aplikace se nejprve vytvoří formulář, do kterého budou specifikovány údaje pro vytvoření projektu a pro nastavení simulačního modelu či vykreslení dosahu robota. Formulář se vzorovým nastavením je znázorněn na obrázku 8.7.

Configuration settings

Project: UR3project Sets parameters for drawing and simulation. Load

Model: UR3model Units: mm Export Load

Range: X: 500 mm Max velocity: 500 mm/s
Y: 500 mm Max acceleration: 1000 mm/s²
Z: 200 mm

Set plane: Compensation:

Center: X: 35 mm End effector: RX: 3.1416 rad
Y: 375 mm RY: 0 rad
Z: 200 mm RZ: 0 rad

Rotation: RX: 0 rad Base: RX: 4.7124 rad
RY: 3.1416 rad RY: 0 rad
RZ: 0 rad RZ: 0 rad

Solution: 1:0:1 Transfer: 1.5708;3.1416;0;3.1416;-3

Creates new project or opens an existing one.
NOTE: If you want to save the configuration for latter use, export it as model. Create

Obrázek 8.7: Okno pro konfiguraci nastavení, založení nebo načtení nového projektu či modelu.

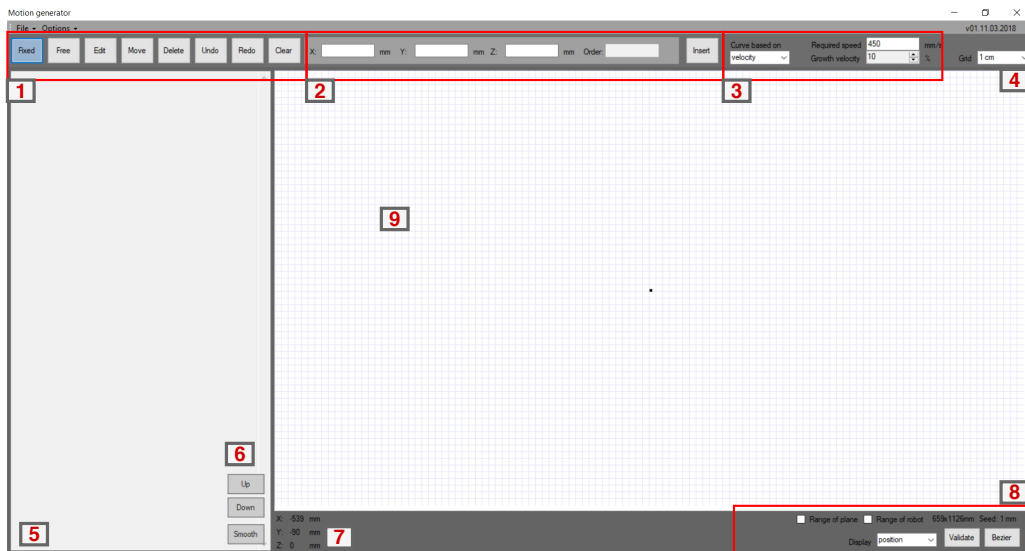
Hodnoty X , Y , Z pro rozsah (range) udávají poloměr doporučeného dosahu robota. Dále maximální rychlost (max velocity) a zrychlení (max acceleration) slouží ke znázornění limitů, kterých tyto veličiny nesmí přesáhnout. Nastavení roviny pro pohyb manipulátoru (set plane) udává jak bude souřadný systém základny transformován na souřadný systém koncového efektoru. Řešení (solution) udává postavení manipulátoru pro simulační model. Dále je třeba definovat kompenzaci koncového efektoru a kompen-

zaci základny taktěž pro definování simulačního modelu. Hodnoty udané v převodu (transfer) definují převod mezi souřadnicemi poskytnuté simulačním modelem a souřadnicemi generované pomocí PolyScope. Způsob získání těchto hodnot je uveden v pokynech přiložených u programů.

Po stisknutí tlačítka vytvořit (Create) se uvedené nastavení projektu uloží, a vytvoří se pracovní plocha. Pro usnadnění vytváření projektů lze exportovat do souboru nastavení modelu. Při vytváření nového projektu pak již stačí tento model načíst, případně po načtení upravit.

8.3.2 Hlavní okno

Po zadání požadovaných hodnot se spustí hlavní okno, znázorněno na obrázku 8.8, které je rozděleno na několik částí označených číslem.



Obrázek 8.8: Hlavní okno aplikace s vyznačenými částmi, které budou níže popsány.

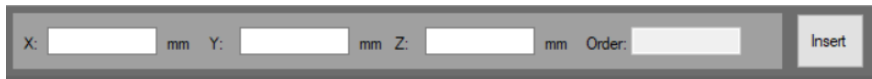
První část 8.9 znázorňuje tlačítka, která definují způsob jednotlivých činností na pracovní ploše 8.14. tlačítka pevné (Fixed) a volné (Free) umožňují nakreslit na pracovní plochu pevné a volné body.



Obrázek 8.9: Tlačítka pro nastavení způsobu činnosti na pracovní ploše (1 z obr. 8.8).

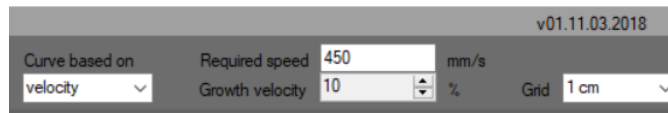
Tlačítko editovat (Edit) umožňuje po vybrání bodu na pracovní ploše editovat jejich vlastnosti. Podobným způsobem funguje tlačítko smazat (Delete), kde je možné po vybrání bodu tento bod smazat. V módu pohybu (Move) lze tlačítka přesouvat. Pokud nastane okamžik, kdy je potřeba vrátit zpět některé kroky, lze použít tlačítka zpět (Undo, Redo). Tlačítko vymazat (Clear) nenávratně vymaže celou historii bodů a pracovní plochu.

Obrázek 8.10 zobrazuje pole, kam je možné zadat požadované hodnoty pro vytvoření nového bodu nebo úpravy a smazání již definovaného bodu.



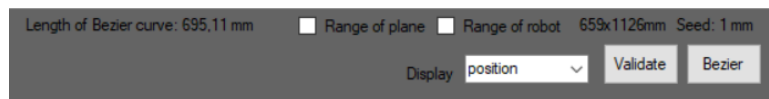
Obrázek 8.10: Panel pro vkládání, editaci a mazání bodů (2 z obr. 8.8).

Generování rychlosti, zrychlení a jerku je založeno na dvou různých přístupech. Prvním přístupem je generování trajektorie na základě předem definované rychlosti (curve based on velocity 8.11), kde je cílové pro body tvořící křivku zadat pouze správné pořadí a pro generování pohybových vlastností požadovanou rychlost (requested speed) a nárůst rychlosti (growth velocity), který definuje rychlost nárůstu vygenerované rychlosti na požadovanou hodnotu. Na základě vygenerované rychlosti je poté vypočten čas pro každý bod křivky.



Obrázek 8.11: Nastavení generování pohybových vlastností (3, 4 z obr. 8.8).

Druhým způsobem je generování trajektorie na základě předem definované času pro každý definovaný bod. Tento způsob návrhu však může být problematický z důvodu udržení spojitosti pohybových vlastností (rychlost, zrychlení, jerk).



Obrázek 8.12: Nastavení zobrazení a dosahu, validace a generování trajektorie (8 z obr. 8.8).

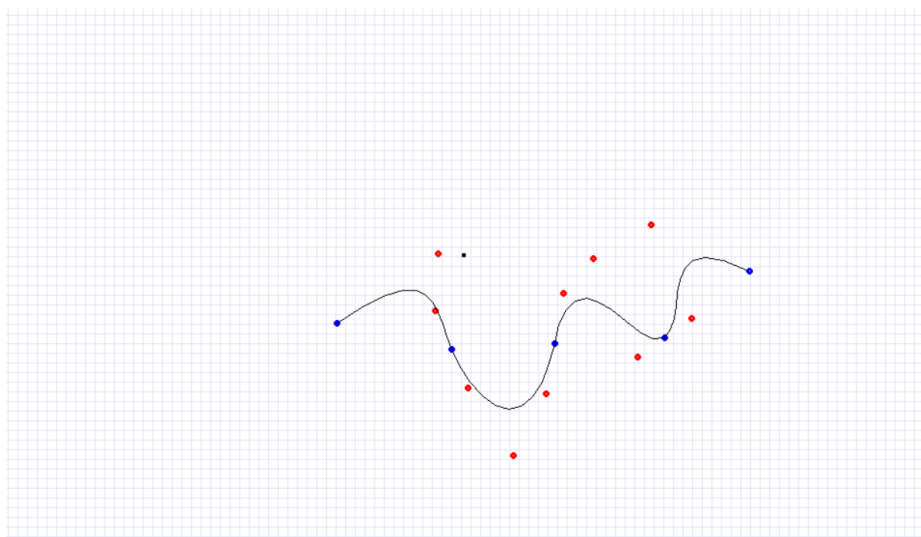
Na obrázku 8.12 lze vidět tlačítka pro zobrazení možného dosahu definovaného z technických vlastností jako bezpečný dosah robota (range of robot) nebo ověřený dosah pomocí simulačního modelu (range of plane).

Dále jsou zde znázorněna tlačítka pro zobrazení křivky vygenerované parametrizací pomocí Bézierových křivek (Bezier) a tlačítko pro validaci křivky, které po stisknutí ověří, zda se robot dokáže po této křivce pohybovat (validate). Nastavení zobrazení (display) definuje, zda se na pracovní ploše 8.14 zobrazí kreslicí plocha nebo zvolené pohybové vlastnosti.

1		Fixed		[x, y, z] = [-132,0 -69,5 0,0] mm		i = 0
2		Free		[x, y, z] = [-27,0 2,5 0,0] mm		i = 1
3		Free		[x, y, z] = [-30,0 -56,5 0,0] mm		i = 1,6 *
4		Fixed		[x, y, z] = [-13,0 -96,5 0,0] mm		i = 2 *
5		Free		[x, y, z] = [4,0 -136,5 0,0] mm		i = 2,4 *
6		Free		[x, y, z] = [51,0 -206,5 0,0] mm		i = 3
7		Free		[x, y, z] = [85,0 -142,5 0,0] mm		i = 3,6 *
8		Fixed		[x, y, z] = [94,0 -90,5 0,0] mm		i = 4 *
9		Free		[x, y, z] = [103,0 -38,5 0,0] mm		i = 4,4 *
10		Free		[x, y, z] = [134,0 -2,5 0,0] mm		i = 5
11		Free		[x, y, z] = [180,0 -104,5 0,0] mm		i = 5,6 *
12		Fixed		[x, y, z] = [208,0 -84,5 0,0] mm		i = 6 *
13		Free		[x, y, z] = [236,0 -64,5 0,0] mm		i = 6,4 *
14		Free		[x, y, z] = [194,0 32,5 0,0] mm		i = 7
15		Fixed		[x, y, z] = [296,0 -15,5 0,0] mm		i = 8

Obrázek 8.13: Textové pole obsahující navržené body (5, 6 z obr. 8.8).

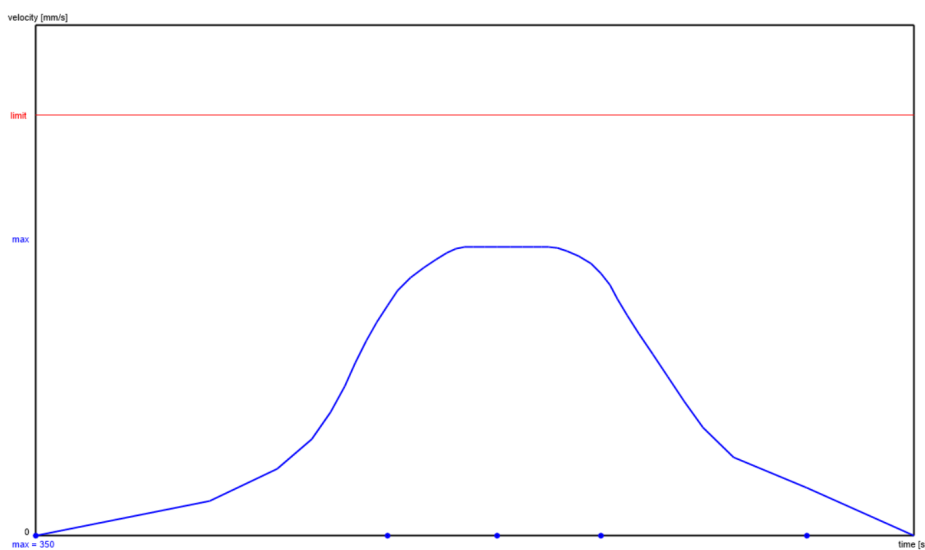
Levá část okna 8.8 znázorněna na 8.13 obsahuje textové pole, kde jsou vypsány jednotlivé nakreslené body s jejich vlastnostmi. Je zde uveden typ bodu (volný, pevný), pořadí jednotlivých bodů a zda-li jsou body párové (označeny znakem *). Párovými body jsou myšleny body, které jsou uměle vygenerované k vybranému bodu tlačítkem vyhladit (Smooth), které zajistí hladké napojení segmentů křivky. Tlačítka nahoru (Up) a dolů (Down) je možné v módu editace řadit vybrané body.



Obrázek 8.14: Pracovní plocha pro návrh křivky (9 z obr. 8.8).

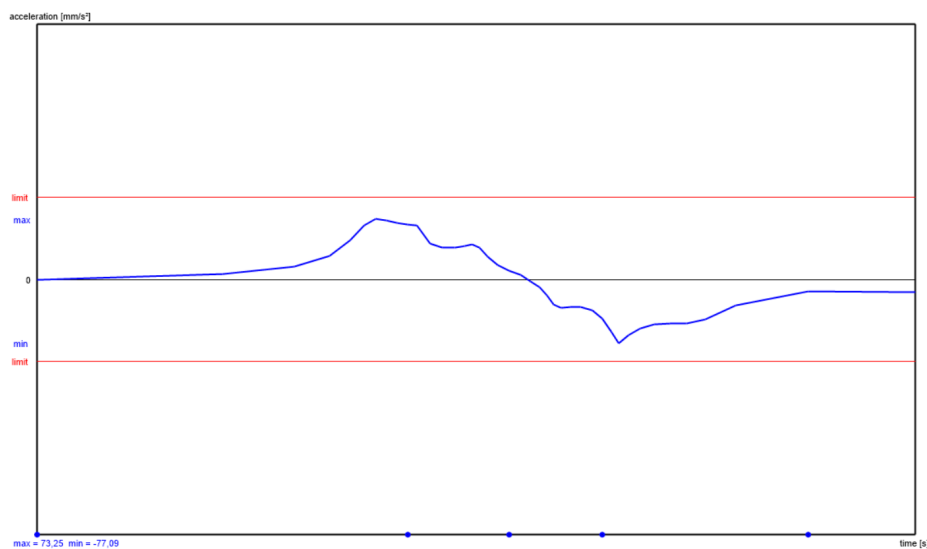
8.3.3 Zobrazení pohybových vlastností

Při zvolení možnosti zobrazení rychlosti se vygeneruje na pracovní ploše graf, znázorňující vypočtenou rychlost v závislosti na čase. V levém dolním rohu je vypsaná maximální rychlost, kterou lze dle návrhu dosáhnout. Také je zde zobrazen limit dostupné maximální rychlosti robota.



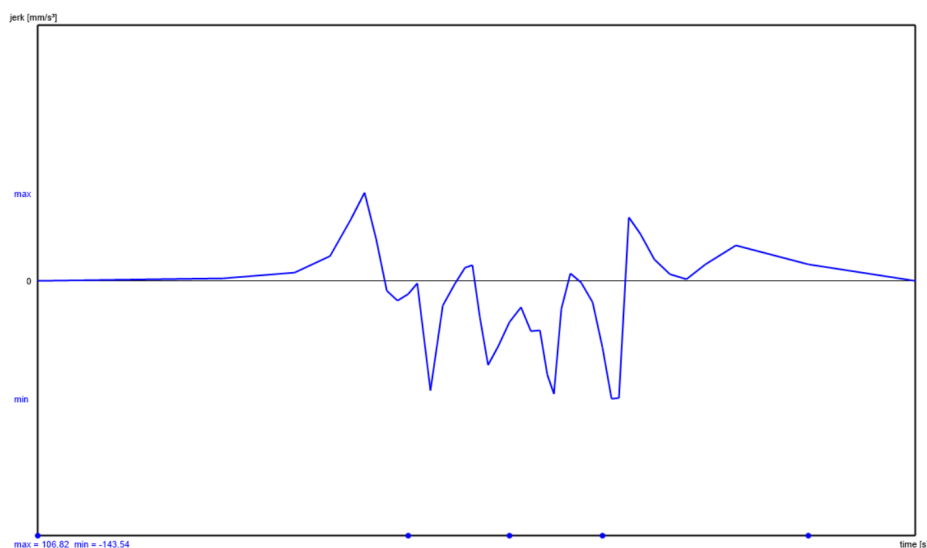
Obrázek 8.15: Zobrazení rychlosti v závislosti na čase.

Zobrazení rychlosti 8.15 ukazuje navržené rychlosti na základě požadované hodnoty. Pomocí Bézierových křivek byl vypočten nárůst a pokles rychlosti, pro hladký náběh na požadovanou rychlost. Rychlost nárůstu a poklesu je možné nastavit v procentech na obrázku 8.11. Počet procent udává na kolik vzorků z celkového počtu vzorků se má doba nárůstu či poklesu uplatnit.



Obrázek 8.16: Zobrazení zrychlení v závislosti na čase.

Na obrázcích 8.16 a 8.17 je stejným způsobem zobrazeno zrychlení, kde jsou zobrazeny limity pro maximální a minimální možné hodnoty, a jerk, který má pouze požadavek na minimalizaci maximální hodnoty.



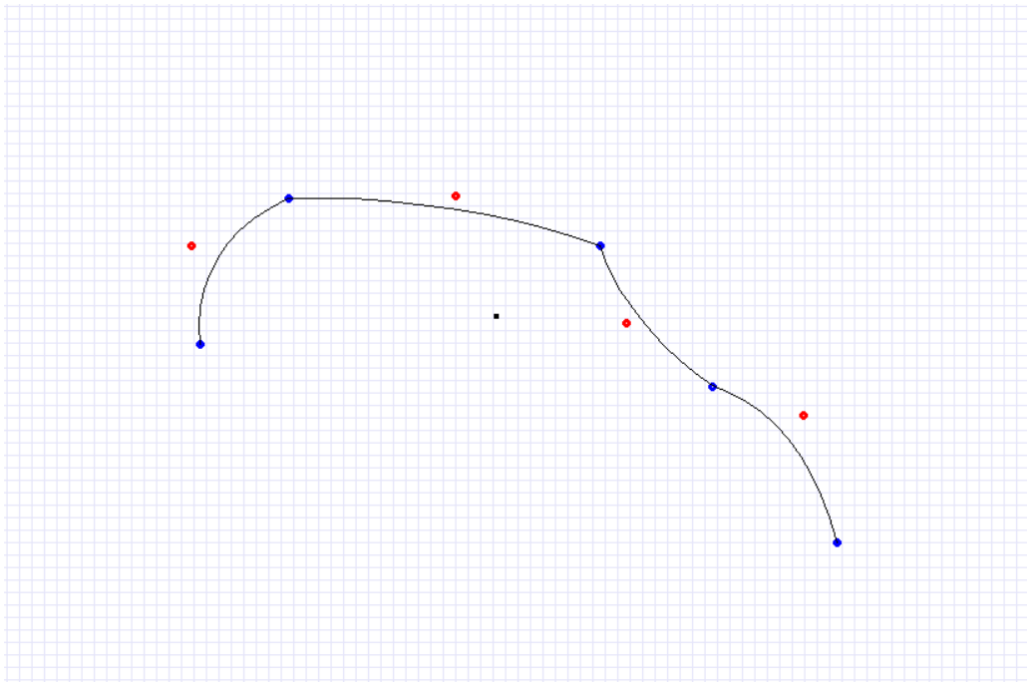
Obrázek 8.17: Zobrazení jerku v závislosti na čase.

9 Demontrace funkčnosti navrženého systému

V poslední části práce jsou uvedeny příklady generování trajektorie pomocí prototypu generátoru trajektorie a experimentální použití těchto trajektorií pro pohyb robota UR3, který je zaznamenán tužkou na papír, která byla připevněna na koncový efektor.

9.1 Generování trajektorie

Testování prototypu generátoru trajektorie zahrnuje ukázky vývoje rychlosti, zrychlení a jerku navržené křivky ve tvaru kruhu pro různě zvolené maximální rychlosti, pro oba přístupy návrhu generování rychlosti při volně napojených Bézierových křivkách a při napojení zajišťující hladkost křivky. Pro ukázkou chování pohybových veličin byla navržena křivka na obrázku 9.1, pro kterou platí, že její napojení jednotlivých segmentů není hladké.



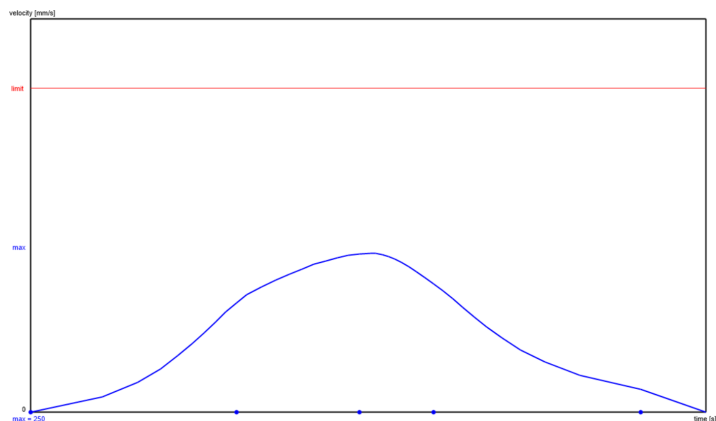
Obrázek 9.1: Zobrazení nakreslené křivky v prototypu aplikace generátoru trajektorie navržené bez hladkých propojení jednotlivých segmentů.

Grafy 9.2, 9.3 a 9.4 obsahují vykeslení pohybových veličin pro maximální navrženou rychlost $v_{max} = 250mm/s$ s postupným nárůstem na požadovanou hodnotu rychlosti do 50 procent navzorkovaných bodů křivky pro přístup generování trajektorie na základě předem definované rychlosti.

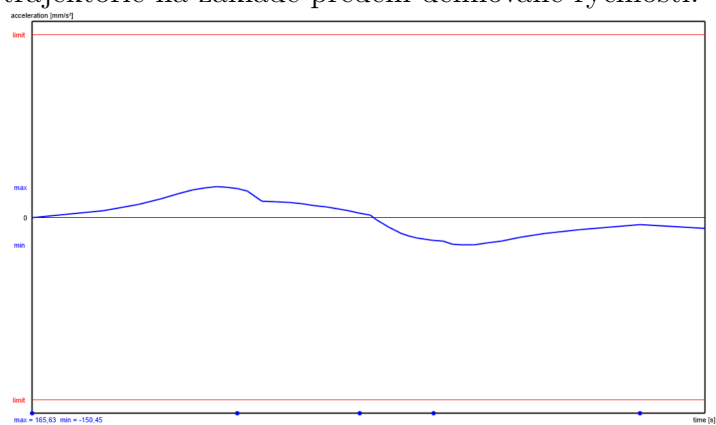
Následující grafy 9.5, 9.6 a 9.7 jsou získány s přístupem generování trajektorie na základě předem definovaného času.

Při porovnání těchto sérií grafů pro dva různé přístupy generování trajektorie lze vidět, že výhodou přístupu generování na základě předem definované rychlosti je cílený návrh průběhu rychlosti tak, aby postupně narůstala na požadovanou hodnotu a před koncem trajektorie postupně klesala na 0. Další výhodou oproti generování trajektorie na základě předem definovaného času je zřetelný z průběhů zrychlení, kde v grafu 9.3 je průběh pro většinu času hladký. Naopak v grafu 9.6 jsou zřetelné nežádoucí výkyvy zrychlení, které se dále projeví v časovém průběhu jerku.

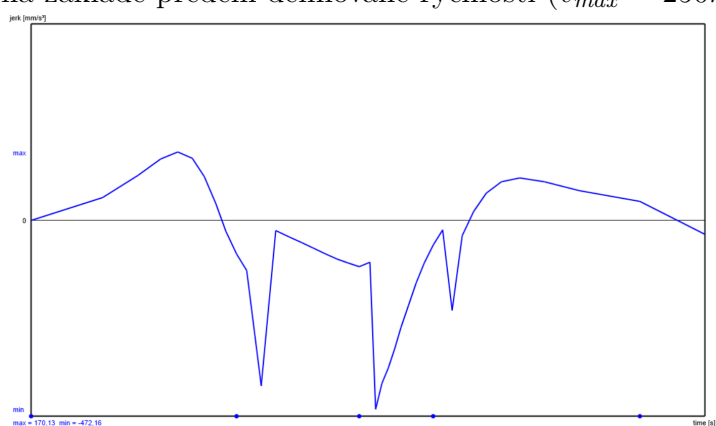
Pro způsob návrhu trajektorie na základě předem definované rychlosti jsou ve většině otestovaných případů splněny požadavky na spojitost rychlosti a zrychlení. Pro druhý způsob návrhu na základě předem definovaného času nejsou požadavky na spojitost splněny vždy, jelikož velmi záleží na navrženém čase, který je přiřazen jednotlivým bodům.



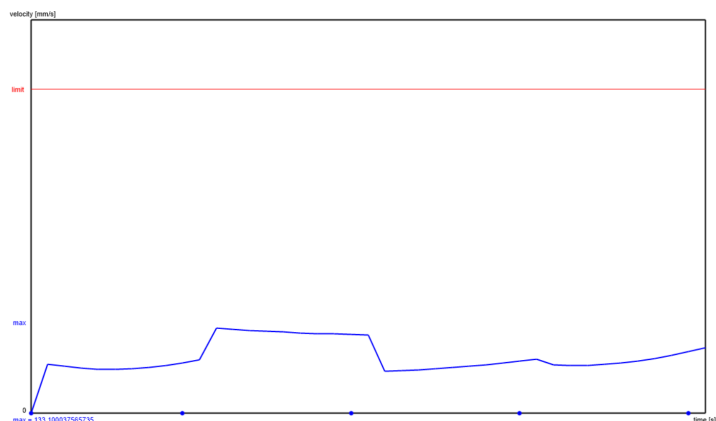
Obrázek 9.2: Zobrazení závislosti rychlosti na čase pro $v_{max} = 250\text{mm/s}$ při generování trajektorie na základě předem definované rychlosti.



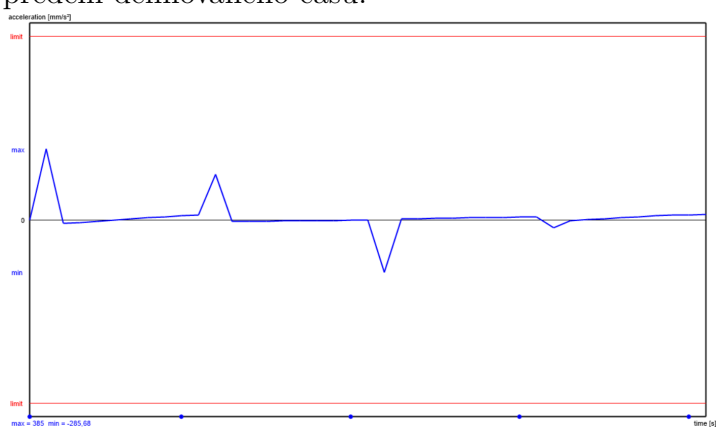
Obrázek 9.3: Zobrazení závislosti zrychlení na čase pro přístup generování trajektorie na základě předem definované rychlosti ($v_{max} = 250\text{mm/s}$).



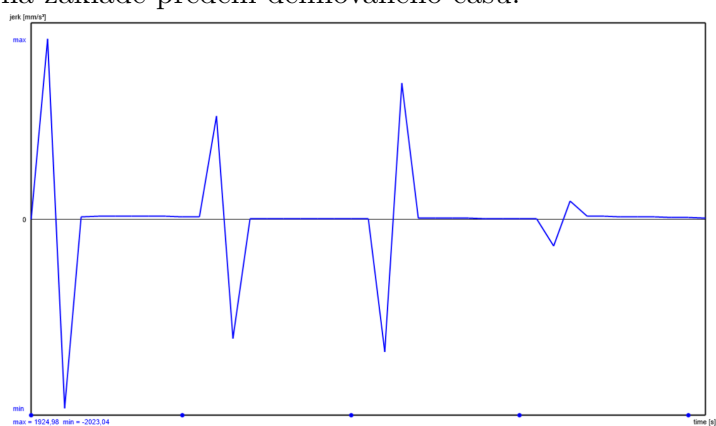
Obrázek 9.4: Zobrazení závislosti jerku na čase pro přístup generování trajektorie na základě předem definované rychlosti ($v_{max} = 250\text{mm/s}$).



Obrázek 9.5: Zobrazení závislosti rychlosti na čase při generování trajektorie na základě předem definovaného času.

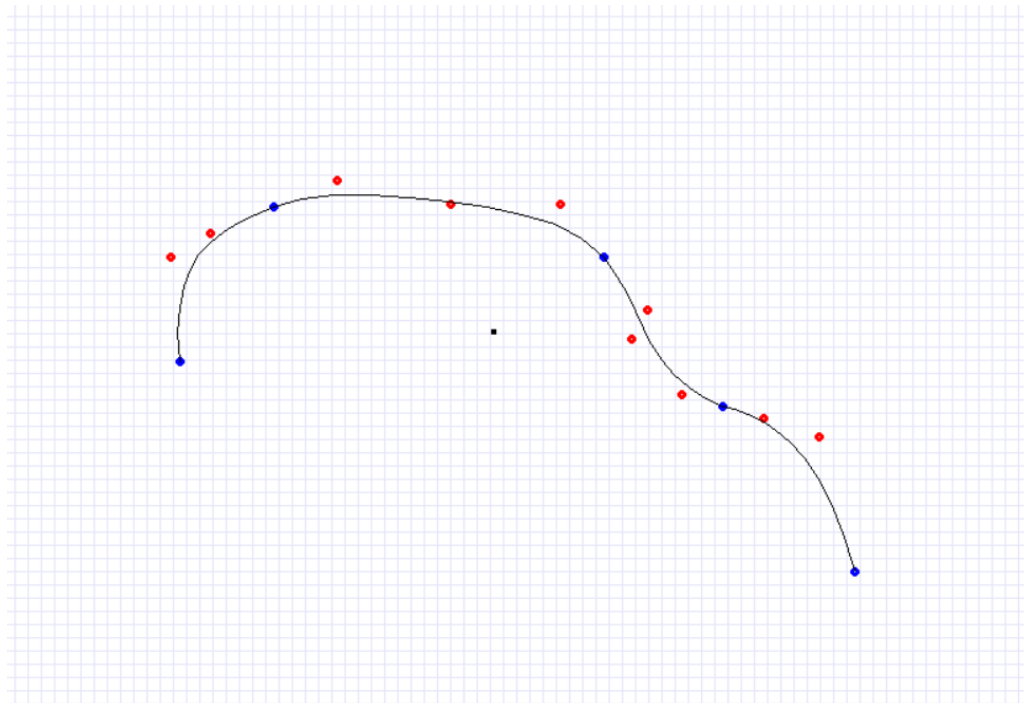


Obrázek 9.6: Zobrazení závislosti zrychlení na čase pro přístup generování trajektorie na základě předem definovaného času.



Obrázek 9.7: Zobrazení závislosti jerku na čase pro přístup generování trajektorie na základě předem definovaného času.

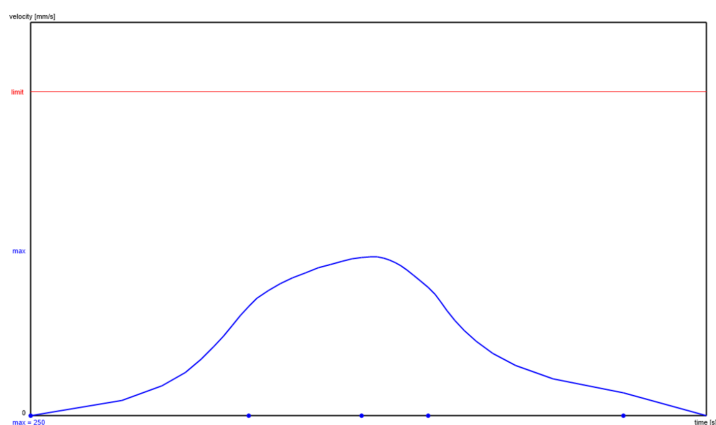
Úpravou navržené křivky 9.1 přidáním volných bodů, které propojí jednotlivé řídicí polygony bude dosaženo hladkosti křivky zobrazující trajektorii pohybu. Hladkost křivky se projeví i v jejích postupných derivacích.



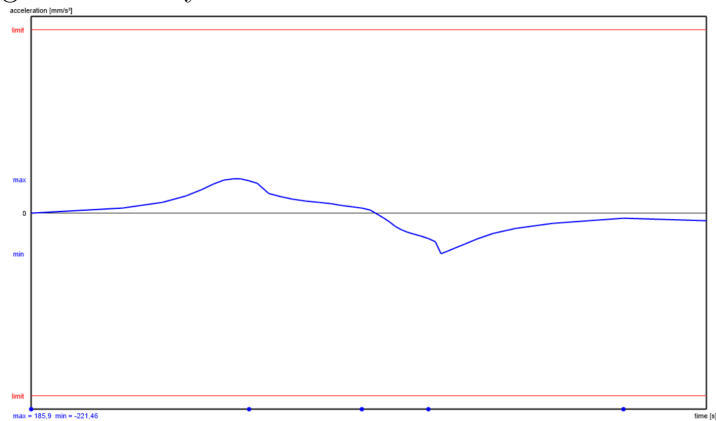
Obrázek 9.8: Zobrazení nakreslené křivky v prototypu aplikace generátoru trajektorie navržené s úpravou propojení jednotlivých segmentů, aby byla zajištěna hladkost křivky.

Pro přístup návrhu trajektorie na základě předem definované rychlosti je z grafů 9.9, 9.10 a 9.11 vidět vliv vyhlazení trajektorie nejvíce pro graf znázorňující časový průběh jerku 9.11.

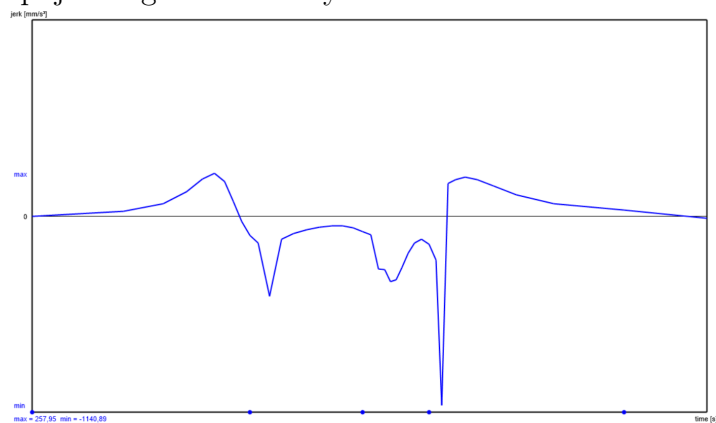
Pro druhý přístup návrhu trajektorie na základě předem definovaného času je změna průběhů pohybových veličin více zřejmá. Z grafů 9.12, 9.13 a 9.14 lze vidět, že výkyvy ve všech grafech z nevyhlazené trajektorie jsou menší a vyhlazenější. Přesto v některých případech průběhy rychlosti a zrychlení pro tento přístup nesplňovali požadavek na spojitost.



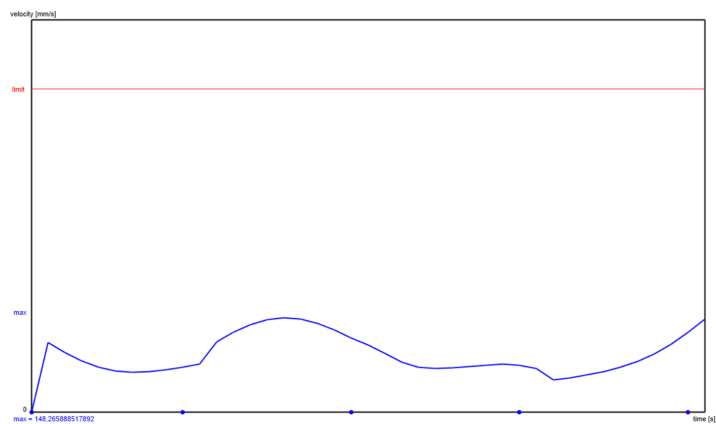
Obrázek 9.9: Zobrazení závislosti rychlosti na čase pro $v_{max} = 250\text{mm/s}$ při generování trajektorie na základě předem definované rychlosti po vyhlazení napojení segmentů křivky.



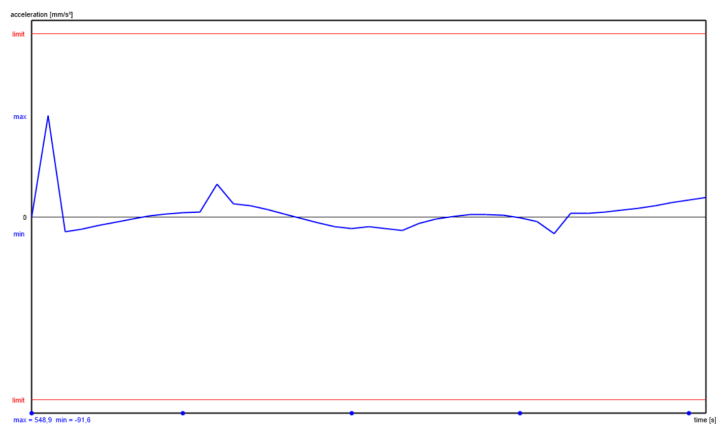
Obrázek 9.10: Zobrazení závislosti zrychlení na čase pro přístup generování trajektorie na základě předem definované rychlosti ($v_{max} = 250\text{mm/s}$) po vyhlazení napojení segmentů křivky.



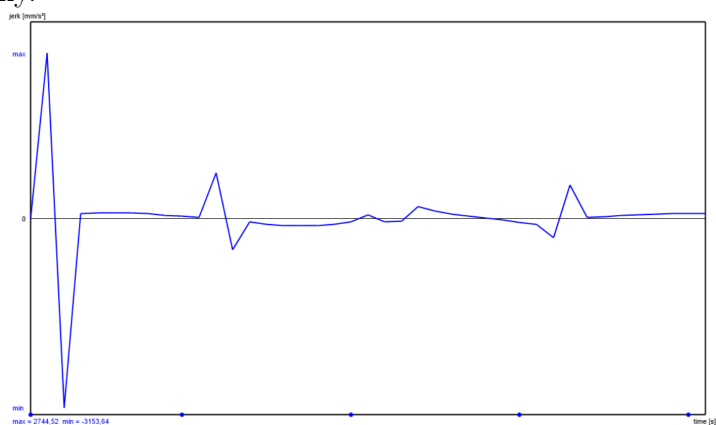
Obrázek 9.11: Zobrazení závislosti jerku na čase pro přístup generování trajektorie na základě předem definované rychlosti ($v_{max} = 250\text{mm/s}$) po vyhlazení napojení segmentů křivky.



Obrázek 9.12: Zobrazení závislosti rychlosti na čase při generování trajektorie na základě předem definovaného času po vyhlazení napojení segmentů křivky.



Obrázek 9.13: Zobrazení závislosti zrychlení na čase pro přístup generování trajektorie na základě předem definovaného času po vyhlazení napojení segmentů křivky.



Obrázek 9.14: Zobrazení závislosti jerku na čase pro přístup generování trajektorie na základě předem definovaného času po vyhlazení napojení segmentů křivky.

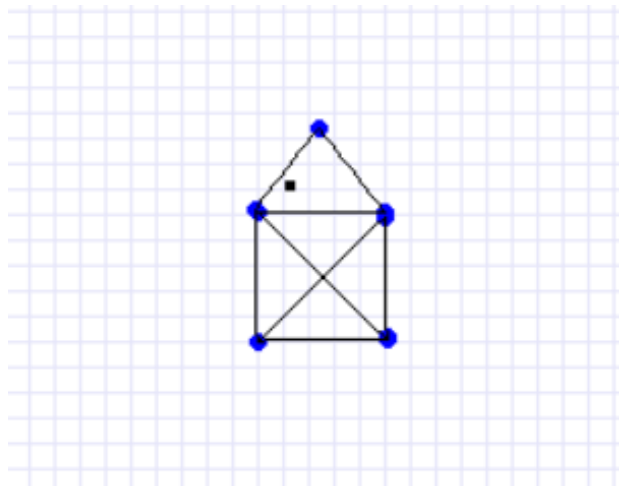
9.2 Experimenty s reálným robotem UR3

Ve druhé části demonstrace funkčnosti navrženého systému byl systém otestován s využitím všech vytvořených modulů pro návrh a ověření realizovatelnosti křivky, dodání vyexportovaných dat do modulu implementovaném v programu REXYGEN, který data příslušné navržené křivce postupně dávkoval do ovládacího modulu implementovaném v programu PolyScope. Systém byl nejprve otestován simulačním chováním robotického manipulátoru, které poskytuje program PolyScope a dále bylo chování systému testováno pomocí reálného robotického manipulátoru UR3, který navrženou trajektorii kreslil na papír.

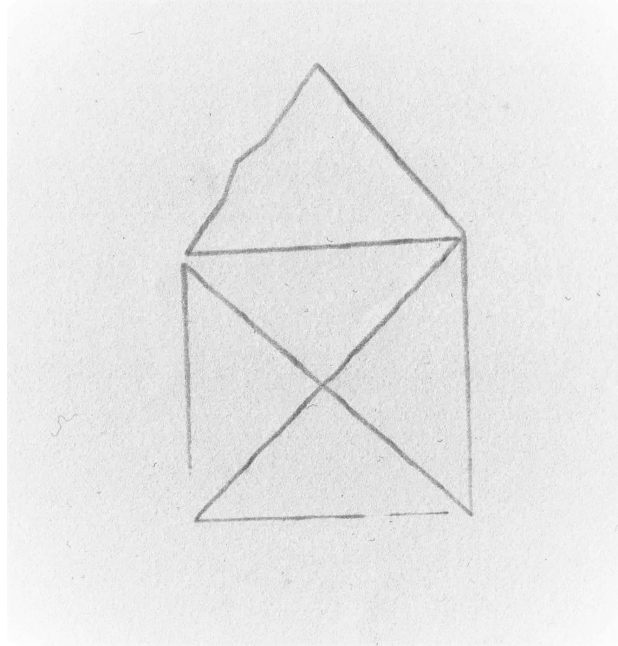
Pro navrženou rovinu, po které se robotický manipulátor pohybuje, je nutné navrhnout trajektorii převrácenou kolem osy y , protože při návrhu roviny je souřadná soustava otočená v ose x o 180 stupňů. V návrhu a implementaci nebyl tento případ uvažován a aby byl výstup zobrazen správně je potřeba návrh trajektorie nové souřadné soustavy přizpůsobit.

Výsledné obrázky 9.16, 9.18 a 9.20 znázorňující naskenovaný výstup reálného manipulátoru UR3 odpovídají navržené trajektorii, avšak v některých částech naskenovaného výstupu trajektorie jsou drobné odchylky, které mohou být způsobeny nesrovnalostmi v poskytnutých datech pomocí komunikace Modbus do ovládacího programu v Polyscope.

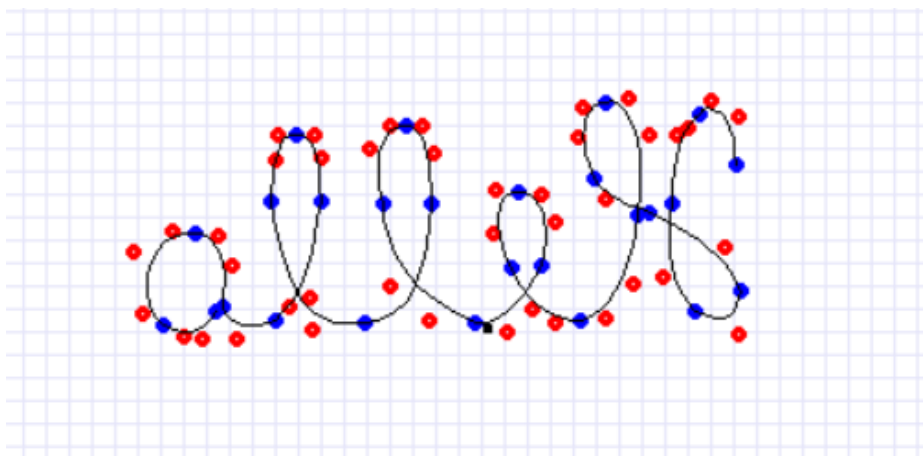
Z výstupů robotického manipulátoru UR3 je vidět, že vytvořený systém pro grafický návrh trajektorie funguje velmi dobře pro testové případy navržené trajektorie pohybu reálného robota.



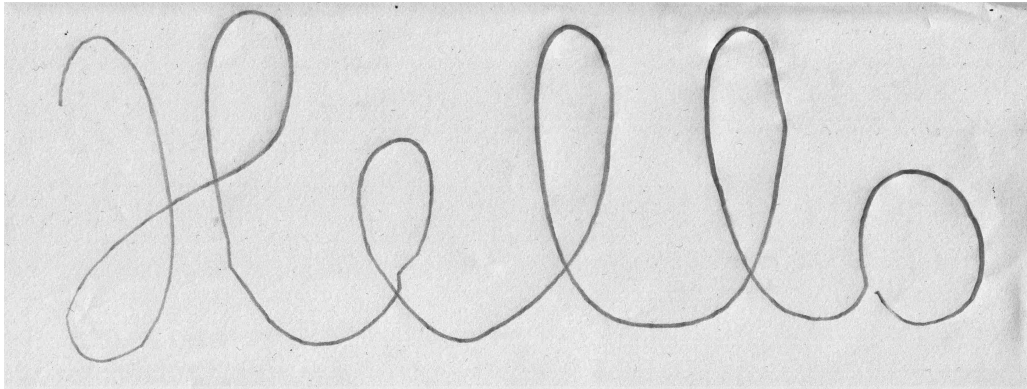
Obrázek 9.15: Zobrazení návrhu trajektorie v prostředí generátoru trajektorie pro domeček jedním tahem.



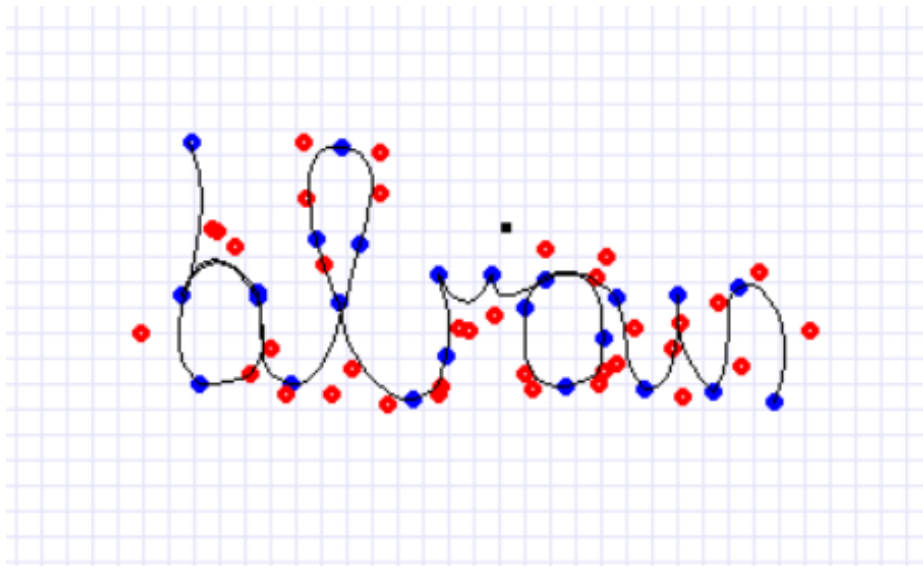
Obrázek 9.16: Naskenovaný výstup reálného robota UR3 v podobě domečku jedním tahem, který byl pomocí tužky napojené na koncový efektor znázorněn na papír.



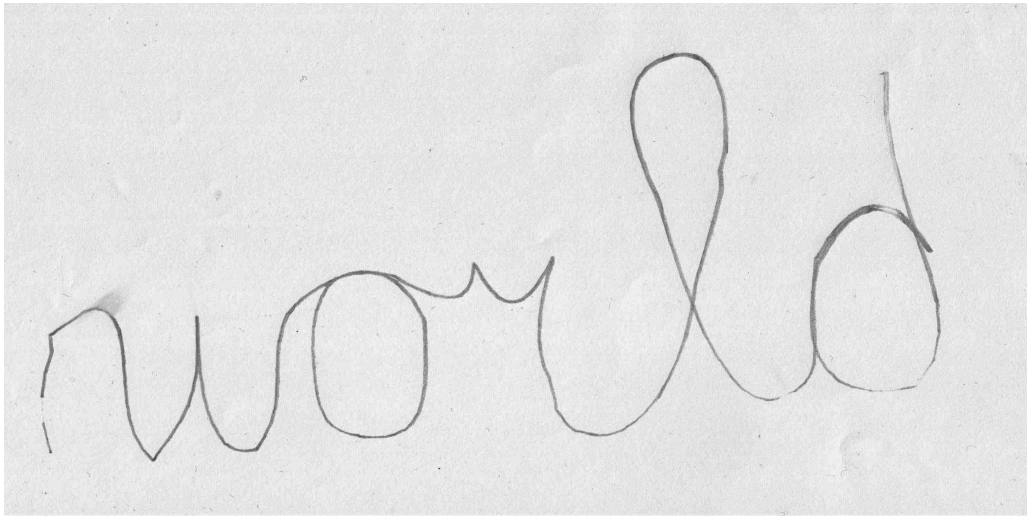
Obrázek 9.17: Zobrazení návrhu trajektorie v prostředí generátoru trajektorie pro nápis hello.



Obrázek 9.18: Naskenovaný výstup reálného robota UR3 v podobě nápisu hello, který byl pomocí tužky napojené na koncový efektor znázorněn na papír.



Obrázek 9.19: Zobrazení návrhu trajektorie v prostředí generátoru trajektorie pro nápis world.



Obrázek 9.20: Naskenovaný výstup reálného robota UR3 v podobě nápisu world, který byl pomocí tužky napojené na koncový efektor znázorněn na papír.

10 Závěr

Cílem práce je se seznámit s funkčními vlastnostmi robotů Universal Robots a jejich ovládáním pomocí komunikačního protokolu Modbus. Nejprve jsou v kapitole 2 zadefinované důležité pojmy a teoretické informace z oblasti robotiky pro snažší porozumění problematice. Dále se práce v kapitole 3 věnuje teoretickým základům vztahujících se ke komunikačnímu protokolu Modbus, kde je vysvětlen princip komunikace a způsob předávání dat. Funkční vlastnosti robotů Universal Robots jsou popsány v následující kapitole 4, kde je představena společnost Universal Robots, manipulátory UR a jejich technické parametry.

Dále je cílem navrhnout univerzální systém ovládání pohybu robota UR3 prostřednictvím komunikačního protokolu Modbus. Návrh systému je popsán v kapitole 5, kde je systém rozdělen do několika funkčních modulů. Funkčními moduly jsou ovládací program implementovaný v programu PolyScope, řídicí program implementovaný v programu REXYGEN a prototyp aplikace grafického editoru pro návrh požadované trajektorie využívající navíc modul poskytující simulační model reálného manipulátoru UR3.

Následující část práce se věnuje teoretickému návrhu jednotlivých modulů. Kapitola 6 se věnuje možným parametrizačním křivkám ze zadaných dat, které jsou testovány v programu Matlab, a výběru vhodné metody pro použití v prototypu aplikace grafického editoru. V další kapitole 7 je uveden inverzní kinematický model manipulátoru UR3 popisující jeho polohové závislosti a závislosti rychlosti a zrychlení.

Implementace navrženého systému je popsána v kapitole 8, kde je věnována pozornost především funkčním vlastnostem navržených modulů případně principům návrhu jejich logiky.

Kapitola 8 obsahuje zobrazení výsledků testování prototypu aplikace grafického editoru pro návrh požadované trajektorie a výsledky experimentálního využití vytvořeného systému pro ovládání reálného robotického manipulátoru UR3.

Pro pohyb manipulátoru po definované trajektorii vytvořený systém funguje velmi dobře při pohybu z klidu do klidu, z bodu do bodu, ale při interpretaci generované křivky není dodržována požadovaná rychlost a zrychlení poněvadž program PolyScope neumožňuje tyto hodnoty dynamicky měnit po komunikaci. Je třeba hledat jiný způsob předávání trajektorie do řídicího systému robota.

Jako příslušná zdokonalení navrženého systému by bylo vhodné použít jiný způsob komunikace mezi jednotlivými programy. Dále by bylo možné rozšířit prototyp aplikace generátoru požadované trajektorie o další funkce jako je například grafický návrh trajektorie v prostoru nebo simulace zobrazující průběh pohybu robota po navržené trajektorii.

Literatura

- [1] *Zdroj obrázku* [online]. [cit. 2018-05-17]. Dostupné z: http://www.karlin.mff.cuni.cz/~sir/modelovani/GPM_Bezier.pdf.
- [2] *Zdroj obrázku* [online]. [cit. 2018-05-17]. Dostupné z: http://what-when-how.com/wp-content/uploads/2012/07/tmpc2f9321_thumb_thumb.png.
- [3] *Uživatelská příručka robota UR3* [online]. [cit. 2018-05-17]. Dostupné z: https://www.universal-robots.com/media/207433/ur3_user_manual_cs_global.pdf.
- [4] *Zdroj obrázku* [online]. [cit. 2018-05-17]. Dostupné z: <https://spectrum.ieee.org/image/MjYxMTEwOQ.png>.
- [5] *Uživatelská příručka program REXYGEN* [online]. [cit. 2018-05-17]. Dostupné z: https://www.rexygen.com/doc/PDF/CZECH/BRef_CZ.pdf.
- [6] Švejda, Martin - Osobní konzultace.
- [7] *Technická dokumentace robota UR3* [online]. [cit. 2018-05-17]. Dostupné z: <https://s3-eu-west-1.amazonaws.com/ur-support-site/16336/100403.PDF>.
- [8] *Technická dokumentace robota UR3* [online]. [cit. 2018-05-17]. Dostupné z: https://www.pi4.de/fileadmin/material/Produktkatalog/Universal_Robots/pdf_components_partner/UR3_robot_working_area.PDF.
- [9] BJÖRCK, Å. Solving linear least squares problems by Gram-Schmidt orthogonalization. *BIT Numerical Mathematics*. 1967, 7, 1, s. 1–21.
- [10] BJÖRCK, A. *Numerical methods for least squares problems*. Siam, 1996.
- [11] FORREST, A. R. Interactive interpolation and approximation by Bézier polynomials. *The Computer Journal*. 1972, 15, 1, s. 71–79.
- [12] FRANK, A. Interpolation of 3D Surfaces for Contact Modeling. *University of Twente, The Netherlands*. 2005.
- [13] FRANKE, R. Scattered data interpolation: tests of some methods. *Mathematics of computation*. 1982, 38, 157, s. 181–200.
- [14] GAVIN, H. P. Constrained Linear Least Squares. *Department of Civil and Environmental Engineering, Duke University*. 2015. Dostupné z: <http://people.duke.edu/~hpgavin/cee201/constrained-least-squares.pdf>.

- [15] MODBUS ORGANIZATION. *Official website* [online]. [cit. 2018-05-3].
Dostupné z: <http://www.modbus.org/>.
- [16] MODBUS ORGANIZATION. *Official website, protocol documentation* [online].
[cit. 2018-05-3]. Dostupné z: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf.
- [17] MODBUS ORGANIZATION. *Official website, protocol documentation TCP/IP*
[online]. [cit. 2018-05-3]. Dostupné z: http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf.
- [18] PRAUTZSCH, H. – BOEHM, W. – PALUSZNY, M. *Bézier and B-spline techniques*. Springer Science & Business Media, 2013.
- [19] SCIAVICCO, L. – SICILIANO, B. *Modelling and control of robot manipulators*. Springer Science & Business Media, 2012.
- [20] SELESNICK, I. Least squares with examples in signal processing. *Connexions*. 2013.
- [21] ŠVEJDA, M. Kinematika robotických architektur. *Katedra Kybernetiky, ZČU v Plzni (online: http://home.zcu.cz/~msvejda/_publications/2011/rigo.pdf)*. 2011.
- [22] ŠVEJDA, M. Optimalizace robotických architektur. 2016.
- [23] UNIVERSAL ROBOTS. *Official website* [online]. [cit. 2018-05-2]. Dostupné z: <https://www.universal-robots.com>.

A Technické dokumenty

Dokumenty v příloze jsou převzaty z [7] a [8]. V příloze je uveden dokument obsahující technické detaily a dokument, ve kterém je zobrazen dostupný pracovní prostor manipulátoru UR3.

UR3

Výkon

Opakovatelnost	±0,1 mm / ±0,0039 in (4 mils)
Povolená okolní teplota	0–50°*
Příkon	Min. 90 W, typicky 125 W, max. 250 W
Kolaborativní fungování	15 vyspělých nastavitelných bezpečnostních funkcí. TüV NORD schválená bezpečnostní funkce Zkoušeno v souladu s: EN ISO 13849:2008 PL d

Specifikace

Užitečné zatížení	3 kg / 6,6 lbs
Dosah	500 mm / 19,7 in
Stupně volnosti	6 otočných kloubů
Programování	Grafické uživatelské rozhraní Polyscope na panelu s 12" dotykovým displejem

Pohyb

Pohyb os robotického ramena	Pracovní rozsah	Maximální rychlost
Podstavec	± 360°	± 180°/s
Rameno	± 360°	± 180°/s
Loket	± 360°	± 180°/s
Zápěstí 1	± 360°	± 360°/s
Zápěstí 2	± 360°	± 360°/s
Zápěstí 3	Nekonečné	± 360°/s
Běžný nástroj		1 m/s / 39,4 in/s

Vlastnosti

Klasifikace IP	IP64
ISO Class Cleanroom	5
Hluk	70dB(A)
Montáž robota	Libovolná
I/O porty	Digitální vstupy 2 Digitální výstupy 2 Analogové vstupy 2 Analogové výstupy 0

I/O elektrické napájení nástroje 12 V / 24 V 600 mA v nástroji

Fyzické

Rozměry	Ø 128 mm
Materiály	Hliník, plast PP
Typ konektoru nástroje	M8
Délka kabelu robotického ramena	6 m / 236 in
Hmotnost s kabelem	11 kg / 24,3 lbs

KONTROLÉR

Vlastnosti

Klasifikace IP	IP20
ISO Class Cleanroom	6
Hluk	<65 dB(A)
I/O porty	Digitální vstupy 16 Digitální výstupy 16 Analogové vstupy 2 Analogové výstupy 2

I/O elektrické napájení	24 V 2 A
Komunikace	TCP/IP 100Mbit, Modbus TCP, Profinet, EthernetIP

Napájecí zdroj	100-240 VAC, 50-60 Hz
Povolená okolní teplota	0–50°

Fyzické

Velikost kontroléru (ŠxVxH)	475 mm x 423 mm x 268 mm / 18,7 x 16,7 x 10,6 in
------------------------------------	---

Hmotnost	15 kg / 33,1 lbs
-----------------	------------------

Materiály	Ocel
------------------	------

Ovládací panel (TEACH PENDANT)

Vlastnosti

Klasifikace IP	IP20
-----------------------	------

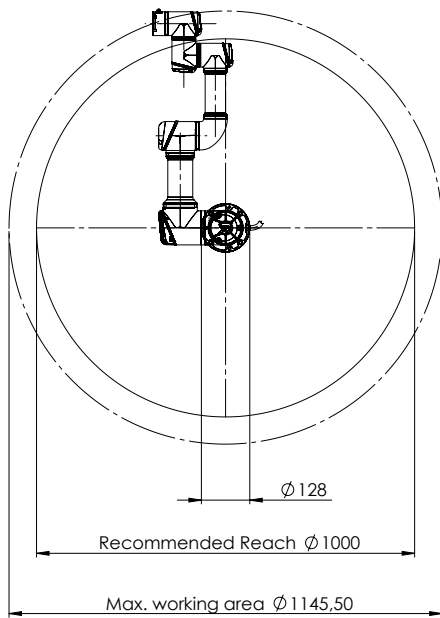
Fyzické

Materiály	Hliník, PP
Hmotnost	1,5 kg
Délka kabelu	4,5 m / 177 in

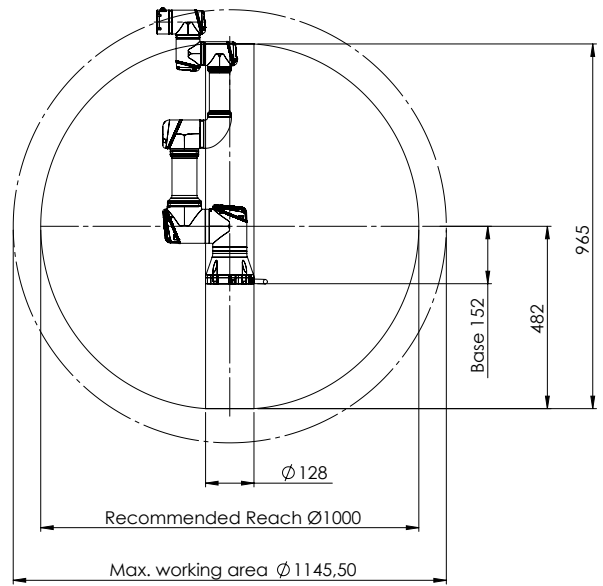


*Robot může pracovat v rozsahu teplot 0-50 °C. Při nepřetržité vysoké rychlosti kloubu se povolená okolní teplota snižuje.


UR3 working area, top view

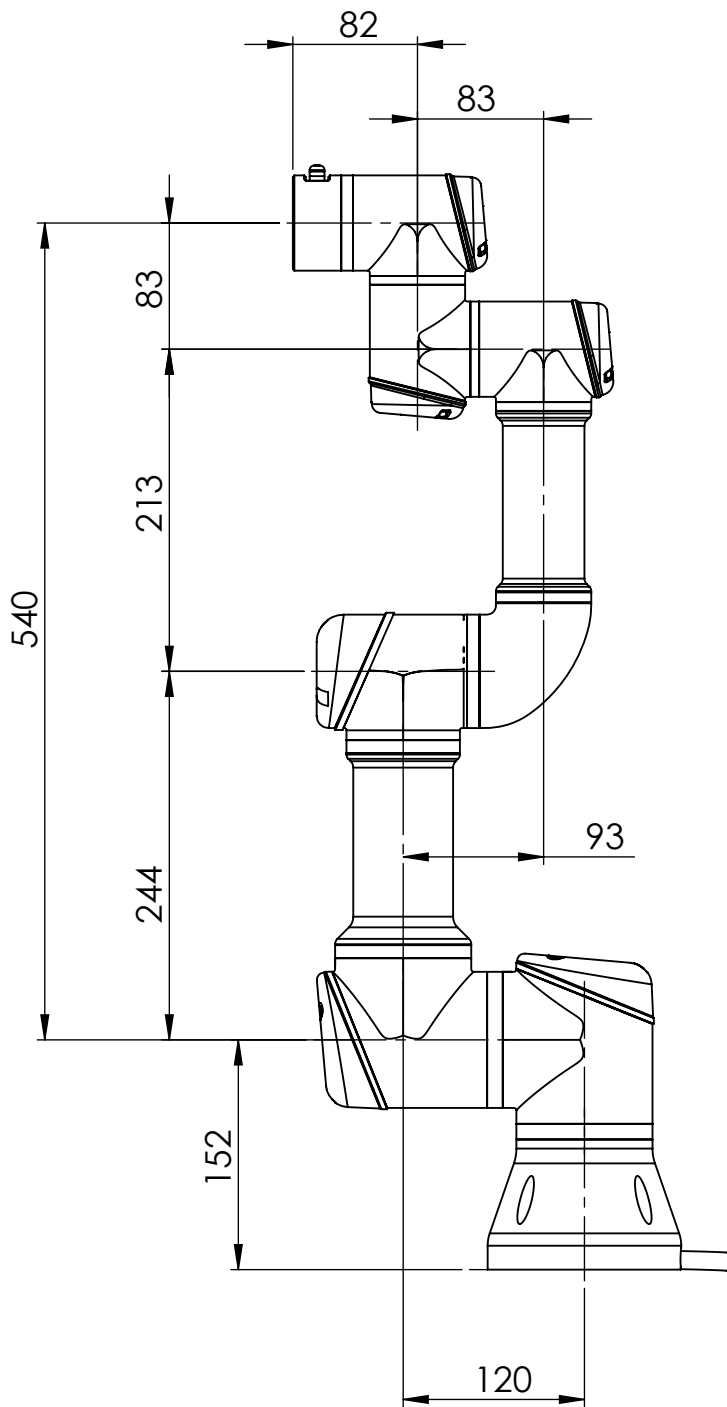


UR3 working area, side view



All dimension is in mm
For public use

 UNIVERSAL ROBOTS <small>TEL: +45 89 93 89 89 FAX: +45 38 77 89 89 WEB: universal-robots.com</small>	
<small>TITLE:</small> Working area UR3 CB3	
<small>DATE:</small>	
<small>Status change date:</small>	
<small>DWG NO.</small> 100403	<small>REV.</small> 0



All dimension is in mm
For public use



UNIVERSAL ROBOTS

TEL: +45 89 93 89 89 FAX: +45 38 79 89 89 WEB: universal-robots.com

TITLE:

Working area UR3 CB3

DATE

Status change date:

DWG NO.

100403

REV.

0