



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Návrh a realizace řídicího algoritmu pro kreslicí lustr Pendulum

Bakalářská práce

Studijní program:

Autor práce:

Vedoucí práce:

B0714A270001 Mechatronika

Martin Světlák

Ing. Lukáš Hubka, Ph.D.

Ústav mechatroniky a technické informatiky





Zadání bakalářské práce

Návrh a realizace řídicího algoritmu pro kreslicí lustr Pendulum

Jméno a příjmení: **Martin Světlák**
Osobní číslo: M19000103
Studijní program: B0714A270001 Mechatronika
Zadávací katedra: Ústav mechatroniky a technické informatiky
Akademický rok: 2021/2022

Zásady pro vypracování:

1. Seznamte se s mechanickou konstrukcí kyvadlového mechanismu, jeho pohony a možnostmi řízení.
2. Realizujte komunikaci s obslužným HMI, popište komunikační interface a formát předávaných dat.
3. Na základě známých délek vodicích lan navrhnete algoritmus pro řízení pohybu koncového kreslicího bodu.
4. Realizujte řídicí algoritmus pohonů a zajistěte jejich synchronizaci.
5. Testujte stroj nakreslením základních zkušebních obrazců (kružnice, čtverec, ...).
6. Navrhnete a implementujete bezpečnostní funkce pro zamezení přepětí v lanech.

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:
Jazyk práce:

dle potřeby dokumentace
30–40 stran
tištěná/elektronická
Čeština



Seznam odborné literatury:

- [1] SOUČEK, Pavel. Servomechanismy ve výrobních strojích. Praha: Vydavatelství ČVUT, 2004. ISBN 80-01-02902-6.
- [2] BALÁTĚ, Jaroslav. *Automatické řízení*. 2., přeprac. vyd. Praha: BEN, 2004. ISBN 978-80-7300-148-3.
- [3] BUŠEK, Martin, Jaroslav ANTOŠ a Miroslav VÁCLAVÍK. Anwendung der Mechatronik zur Umsetzung der kinetischen Skulptur Pendulum. Fachtagung Mechatronik [online]. 2019, 2019, 137-142 [cit. 2021-10-5]. Dostupné z: doi:10.17619/UNIPB/1-769.
- [4] Sysmac Studio [online]. OMRON Corporation, 2021 [cit. 2021-10-5]. Dostupné z: <https://industrial.omron.cz/cs/products/sysmac-studio>.
- [5] YAO, Yan-an, Ce ZHANG a Hong-Sen YAN. Motion control of cam mechanisms. Mechanism and Machine Theory [online]. 2000, 35(4), 593-607 [cit. 2021-10-5]. ISSN 0094114X. Dostupné z: doi:10.1016/S0094-114X(99)00025-7.

Vedoucí práce:

Ing. Lukáš Hubka, Ph.D.
Ústav mechatroniky a technické informatiky

Datum zadání práce:

12. října 2021

Předpokládaný termín odevzdání:

16. května 2022

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

doc. Ing. Josef Černožorský, Ph.D.
vedoucí ústavu

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

15. května 2022

Martin Světlák

Poděkování

Chtěl bych poděkovat svému vedoucímu Ing. Lukáši Hubkovi, Ph.D. za vstřícnost a pomoc při tvorbě této práce. Dále Ing. Ondřeji Tesařovi, který mě naučil všem potřebným základům elektrotechniky, a Mgr. Evě Elízové za výrazné zlepšení mých češtinářských dovedností. Velké poděkování patří Ing. Pavlu Dostrašilovi, Ph.D. z firmy VÚTS, za ochotu a asistenci při programování. V neposlední řadě bych chtěl poděkovat své rodině a přátelům za neskonalou podporu.

Návrh a realizace řídicího algoritmu pro kreslicí lustr Pendulum

Abstrakt

Tato práce se zabývá tvorbou algoritmu pro PLC, které řídí model kinetické plastiky Pendulum. Kinetická plastika Pendulum funguje na principu závěsného kyvadla, jehož dráha je řízena pomocí tří bočních tažných lan navíjených servopohony s převodovkou. Závěs kyvadla se pohybuje po kulové ploše a pomocí výsuvného hrotu vykresluje obrazce do pískové plochy pod ním umístěné. Celý mechanismus je možné ovládat z obslužného HMI.

Řídicí algoritmus je navržen ve vývojovém prostředí Sysmac Studio pro PLC a servopohony značky Omron. Zajišťuje načtení a zpracování dat z FTP severu, na který jsou poslána z externí návrhové aplikace umístěné na obslužném notebooku. Data jsou přijímána ve formátu pozice hrotu kyvadla v čase a jsou interpolována buďto lineárně, nebo kubickým splinem. Synchronizace pohybu servopohonů je zajištěna pomocí virtuálních os a vektorového přepočtu zpracovaných dat na délky jednotlivých lan.

Klíčová slova: kyvadlo, PLC, řídicí algoritmus, Sysmac Studio, servopohony, interpolace, FTP server, vektory

Design and Implementation of a Control Algorithm for the Pendulum Drawing Chandelier

Abstract

This thesis deals with the creation of an algorithm for PLC, which controls the model of kinetic sculpture Pendulum. The Pendulum kinetic sculpture works on the principle of a pendulous construction, the path of which is controlled by three ropes on its sides. Ropes are wound with servo drives with a gearbox. The pendulum moves along a spherical surface and uses an extendable drawing tip to draw shapes into a sand surface below it. The whole mechanism can be controlled from the operating HMI panel.

The control algorithm is designed in the Sysmac Studio development environment for Omron PLCs and servo systems. The algorithm performs reading and processing of data from an FTP server, which is sent from an external design application located on the operating laptop. The data is received in a format of pendulum tip position as a function of time and is interpolated either linearly or by a cubic spline. Synchronization of the servomotors is ensured by means of virtual axes and vector conversion of processed data into the lengths of individual ropes.

Keywords: Pendulum, PLC, control algorithm, Sysmac Studio, servo systems, interpolation, FTP server, vectors

Obsah

Seznam zkratk	10
Seznam obrázků	12
Úvod	13
1 Popis kinetické plastiky pendulum	14
1.1 Základní myšlenka kinetické plastiky Pendulum a její reálná aplikace	14
1.2 Popis konstrukčního a elektrotechnického řešení kyvadla	16
1.3 Porovnání kinetické plastiky s testovacím modelem	17
1.4 Cíle pro vylepšení modelu kyvadla	18
2 Teoretický rozbor jednotlivých komponentů a použitého softwaru	20
2.1 Mechanická konstrukce modelu kyvadla	20
2.2 Popis použitého PLC a jednotlivých modulů	22
2.3 Software použitý pro programování PLC	23
2.4 Stejnoseměrný zdroj pro napájení komponentů	24
2.5 Průmyslový Ethernet switch	25
2.6 Servopohony a měniče	26
2.7 Použitý HMI panel	27
2.8 Rozbor jednotlivých úloh pro řešení	28
3 Postupy pro vyřešení jednotlivých úloh	30
3.1 Počáteční nastavení parametrů nového projektu v Sysmac Studiu	30
3.2 Nastavení pohybových os	30
3.3 Nastavení servopohonů a jejich regulace	33
3.4 Struktura řídicího programu	35
3.5 Počáteční průchod programem po zapnutí systému	39
3.6 Způsoby řízení a příkazy pro jednotlivé pohybové osy	41
3.7 Vektorové přepočty dat na délku lan	43
3.8 Kalibrace, Homing a doplňkové programy	46
3.9 Příjem, čtení a zpracování dat	47
3.10 Algoritmus pro vykreslení obrázků	50
3.11 Ochrana proti poruchám a chybová hlášení	52
3.12 Vizualizace pro obslužné HMI	52
4 Testování a vyhodnocení výsledků	56

Závěr	61
Použitá literatura	62

Seznam zkratek

PLC	Programable Logic Controller, programovatelný logický automat
FTP	File Transfer Protocol, protokol pro přenos souborů pomocí počítačových sítí
IP	Internet Protocol, základní protokol v počítačových sítích
HMI	Human Machine Interface, rozhraní pro komunikaci mezi člověkem a strojem
SD	Secure Digital, paměťová karta
VÚTS	Výzkumný ústav textilních strojů

Seznam obrázků

1.1	Pohled na kyvadlo [2]	15
1.2	Funkčnost kyvadla [2]	15
1.3	Tripodová konstrukce kyvadla [1][2]	16
1.4	Model kyvadla [1]	18
1.5	Obrazec s obličejem šejka [1]	19
2.1	Rozvaděčová deska s komponenty (1. PLC NX102-1220, 2. DC zdroj, 3. Ethernet switch, 4. Měníče pro servopohony řídicích lan, 5. Měníč pro servopohon hrotu, 6. HMI panel)	20
2.2	Modely uchycení [1]	21
2.3	Konstrukce navíjecích jednotek [1]	22
2.4	Použité PLC NX102-1220 [3]	23
2.5	Vývojové prostředí Sysmac Studio [3]	24
2.6	Stejnoseměrný spínaný zdroj S8VK-G12024 [3]	25
2.7	Průmyslový switch Wago 852-101 [5]	25
2.8	Servopohon a frekvenční měnič [3]	26
2.9	Servopohon a frekvenční měnič [3]	27
2.10	HMI bez displaye mTV-100 [6]	28
3.1	Přiřazení pohybových os servopohonům	31
3.2	Příklad nastavení pohybové osy s převodovkou	32
3.3	Regulační struktura servopohonu	35
3.4	Ilustrace půdorysu pískové plochy a rámu kyvadla s body ABC, které reprezentují průchodky lan	39
3.5	Trojúhelník vzniklý ze známých hodnot pro lano L1	40
3.6	Funkce bloku MC_SyncMoveAbsolute [4]	42
3.7	Funkce bloku MC_MoveLinearAbsolute [4]	43
3.8	Ilustrace vychýlení kyvadla a změny délky lan při pohybu (pohled ze strany navíjecích jednotek A a B, třetí byla vynechána pro přehlednost)	44
3.9	Ilustrace vychýlení kyvadla s body definovanými ze známých délek (pohled shora)	44
3.10	Obrazec Šťastný lenochod nakreslený v návrháři	47
3.11	Vrchní lišta vizualizace	52
3.12	Chybové kódy	53
3.13	Obrazovka Home pro ovládání vykreslování	53
3.14	Obrazovka Manual pro ruční ovládání pohybových os	54
3.15	Obrazovka Settings	55
4.1	První jedno-tahové testovací obrazce	56
4.2	Více-tahové obrazce	56
4.3	Srovnání interpolací na obrazci Šťastný lenochod	57
4.4	Graf pozic virtuálních os X a Y pro lineární interpolaci	58
4.5	Graf rychlosti virtuálních os X a Y pro lineární interpolaci	58

4.6	Graf pozic virtuálních os X a Y pro kubickou interpolaci	59
4.7	Graf rychlosti virtuálních os X a Y pro kubickou interpolaci	59
4.8	Graf rychlosti lana Rope1, porovnání kubické a lineární interpolace .	60

Úvod

Bakalářská práce se zabývá návrhem řídicího algoritmu pro kinetickou plastiku Pendulum, která svým pohybem vykresluje obrazce do pískové plochy. Kinetická plastika funguje jako závěsné kyvadlo, jehož vychýlení je provedeno pomocí tří lan. Kreslení do pískové plochy je provedeno výsuvným kreslícím hrotem, který kompenzuje vzdálenost od pískové plochy při vychýlení kyvadla.

Cílem práce je navrhnout řídicí algoritmus pro model plastiku tak, aby dokázal zpracovat trajektorii z dat zaslaných do PLC a následně vykreslil na základě příkazů z obslužného HMI požadovaný obrazec do pískové plochy. Data potřebná pro vykreslení jsou zpracována a poslána z interaktivní návrhové aplikace od pana Davida Dostrašila. Dosavadní řídicí algoritmus byl navržen pouze pro vykreslování předem definovaných obrazců, jejichž návrh je časově velmi náročný. Důvodem pro tvorbu nového algoritmu je výrazné zvýšení interaktivnosti plastiku i pro běžného uživatele.

V první kapitole je popsána kinetická plastika Pendulum a fyzický testovací model v poměru 1:10. Zároveň jsou v ní blíže specifikovány důvody pro vylepšení funkcí kinetické plastiku. Druhá kapitola popisuje mechanickou konstrukci testovacího modelu, všechny použité elektronické komponenty, použitý software a jednotlivé úlohy, které budou řešeny v rámci práce. Ve třetí kapitole jsou podrobně rozebrány matematické a programovací prostředky, které byly použity pro tvorbu nového řídicího algoritmu, a vizualizace vytvořená pro obslužné HMI. Čtvrtá kapitola popisuje postup vývoje a testování a porovnává na průbězích vykreslování použití lineární a kubické interpolace.

1 Popis kinetické plastiky pendulum

1.1 Základní myšlenka kinetické plastiky Pendulum a její reálná aplikace

Plastikou je myšleno umělecké dílo, které mimo svou základní funkci může plnit i další doplňkové estetické funkce, např. osvětlující, nebo ozvučující. V případě popisované plastiky se jedná o model závěsného lustru, který svou konstrukcí a pohybem připomíná kyvadlo. Hlavní funkcí plastiky Pendulum je vykreslování obrazců do vrstvy písku pomocí čtyř synchronně řízených servomotorů. Model plastiky je umístěn ve firmě VÚTS a slouží k testování fyzikálního chování kyvadla a principů řízení, aby je bylo následně možné použít na reálné aplikaci.[1] Tvůrcem aplikace je česká sklářská a designová společnost Lasvit z Nového Boru a jejich designová návrhářka Wanda Valihrachová. Řízení pohybu plastiky zřizovala liberecká firma VÚTS, a.s., která je zároveň zadavatelem této bakalářské práce. Umělecké dílo nese název Art of Healing a nachází se ve Spojených arabských emirátech v nemocnici Burjeel Medical City ve městě Abu Dhabi. Jedná se o 16 metrů vysoké kyvadlo, složené z 300 křišťálových trubiček, přičemž každá má vlastní světelný zdroj.[2]



Obrázek 1.1: Pohled na kyvadlo [2]

Celá konstrukce váží přibližně 620 kilogramů. Kyvadlo má umělecký význam jednak díky svému designu a osvětlení, které poskytuje, ale také díky obrazcům, které kreslí svým hrotem do pískové plochy o velikosti 10 metrů čtverečních, umístěné pod konstrukcí kyvadla. Dílo má zároveň disponovat terapeutickými a uklidňujícími účinky pro zdejší klienty.[2]



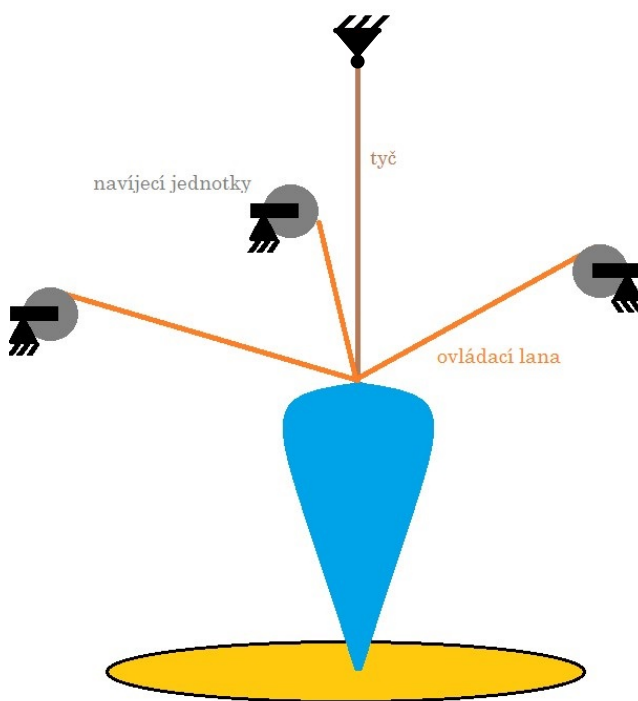
(a) Křišťálové trubičky se světelnými zdroji (b) Obrazec vykreslený do pískové plochy

Obrázek 1.2: Funkčnost kyvadla [2]

1.2 Popis konstrukčního a elektrotechnického řešení kyvadla

Z možných druhů konstrukcí byla zvolena konstrukce na principu lanového tripodu s centrální nosnou trubkou, která je uchycena k základnímu nosnému rámu konstrukce pomocí kulového kloubu se třemi stupni volnosti (viz obrázek 1.3a).

Řízení a pohon zajišťují komponenty od japonského výrobce Omron. Na stranách kyvadlové konstrukce jsou uchycena tři tažná lana (viz obrázek 1.3b). Pohyb lan zajišťují navíjecí jednotky se servomotory řady R88M-1L1K530C-BS2 s frekvenčními měniči R88D-1SN15F-ECT. Moment je z motorů převáděn na navíjecí cívky pomocí planetových převodovek, které zajišťují jeho dostatečné zvýšení tak, aby servomotor bezproblémově utáhl konstrukci kyvadla.



(a) Grafická reprezentace tripodové konstrukce



(b) Úchyty lan na kyvadle

Obrázek 1.3: Tripodová konstrukce kyvadla [1][2]

Pohyb kreslicího trnu je realizován servopohonem řady R88M-1M10030T-BS2 s ozubeným řemenem, který zajišťuje lineární pohyb trnu. Servopohon je řízen frekvenčním měničem řady R88D-1SN04H-ECT

Mazání pískové plochy pod kyvadlem je prováděno pomocí dvou servopohonů řady R88M-1M40030T-S2, určených pro posuv mazací lišty a jednoho servopohonu řady R88M-1M40030T-BS2 určeného pro rotaci. Servopohony jsou řízeny skrze frekvenční měniče typu R88D-1SN04H-ECT. Mazání pískové plochy probíhá tak, že lišta přejezdí lineárním pohybem pískoviště a poté dojde k její rotaci, což umožní

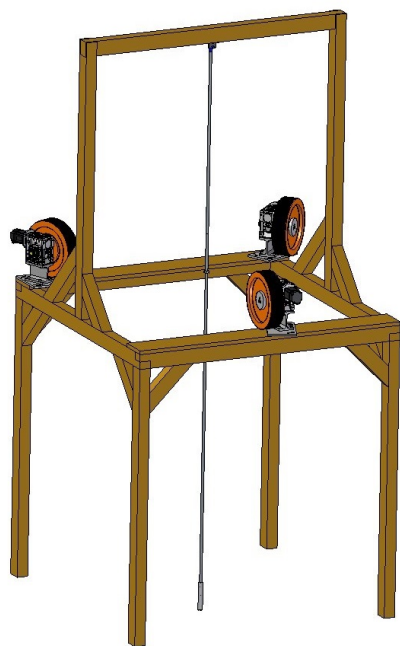
přejet nahrnutý val písku. Následně se lišta natočí zpět do své původní polohy a jede lineárním pohybem zpět.

Řízení aplikace je provedeno skrze PLC řady NJ501-1300. Pomocí obslužného HMI panelu je možné ovládat pohyb kyvadla, jeho parametry a volit jednotlivé scény pro vykreslení do písku. Scény jsou umístěny na SD kartě, která je zapojena přímo do kontroleru. PLC následně zajišťuje přepočít os a synchronní pohyb všech čtyř servomotorů. Vlastnosti konstrukce jsou podrobněji popsány ve vědeckém článku [1].

1.3 Porovnání kinetické plastiky s testovacím modelem

Testovací model kyvadla ve firmě VÚTS slouží k ověření funkčnosti a realizovatelnosti konstrukce a řídicích algoritmů. Jedná se o model v měřítku 1:10 oproti reálné konstrukci. Elektronické komponenty modelu plní stejnou funkci jako komponenty na reálném kyvadle. Není tedy problém po drobných úpravách použít program na obě aplikace. Konstrukce, její rozměry a použité komponenty jsou podrobněji popsány v kapitole 2.

Jedním z hlavních rozdílů ve funkci je absence automatického mazání písku. Mazání písku je zde prováděno ručně, pomocí speciální dřevěné lišty, která je vyrobena tak, aby tvarem odpovídala modelovému pískovišti a bylo možné dosáhnout rovné pískové plochy. Jelikož řízení pohonů pro mazání je již na reálném kyvadle funkční a na modelu není jak ho testovat, nebo zlepšovat, nebude se jím tato práce zabývat.



(a) Grafická reprezentace modelu



(b) Reálný model

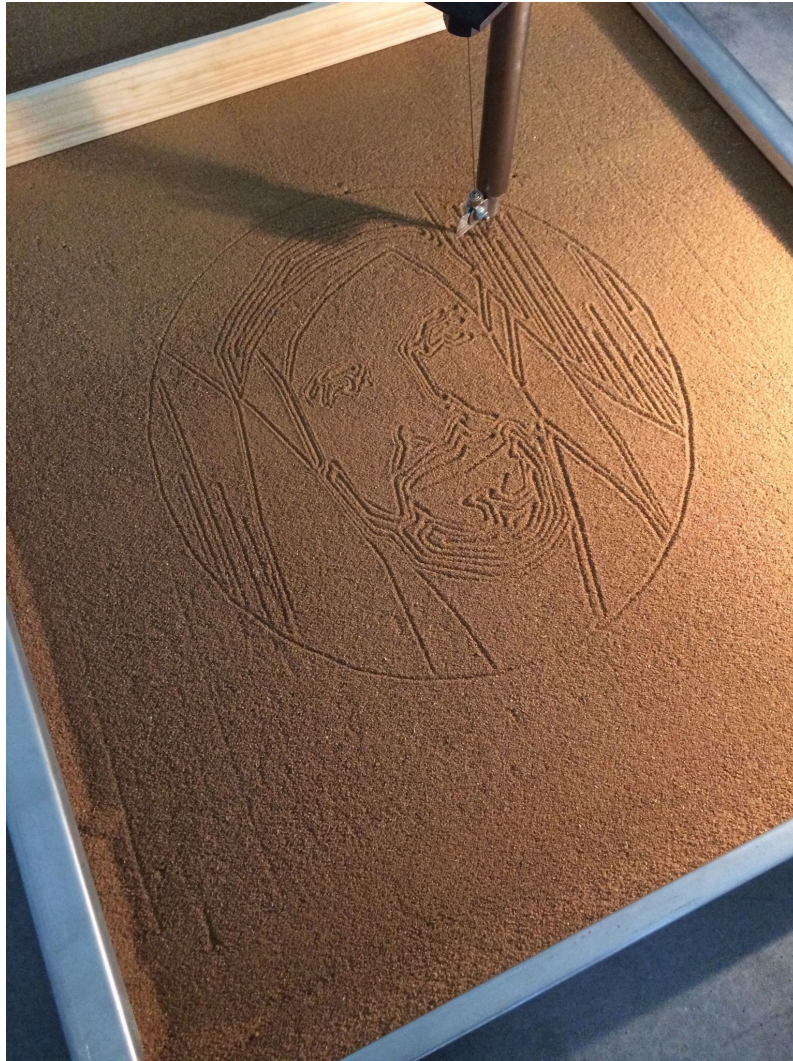
Obrázek 1.4: Model kyvadla [1]

Další rozdíl je v provedení pohybu hrotu. Zatímco na reálném kyvadle je pohon umístěn uvnitř plastiky a jeho rotační pohyb je převeden na lineární pomocí lineární jednotky s ozubeným řemenem, u modelu je hrot zavěšen na ocelovém lanku, které je pomocí servopohonu navíjeno na navíjecí jednotku.

V neposlední řadě jsou zde stop-tlačítka a světelné indikátory, které jsou na reálném kyvadle zapojeny do I/O modulů řídicího PLC. Protože model slouží pouze pro testování a potřeby firmy VÚTS, nejsou indikátory a tlačítka fyzicky přítomna, ale jsou provedena virtuálně na obslužném HMI.

1.4 Cíle pro vylepšení modelu kyvadla

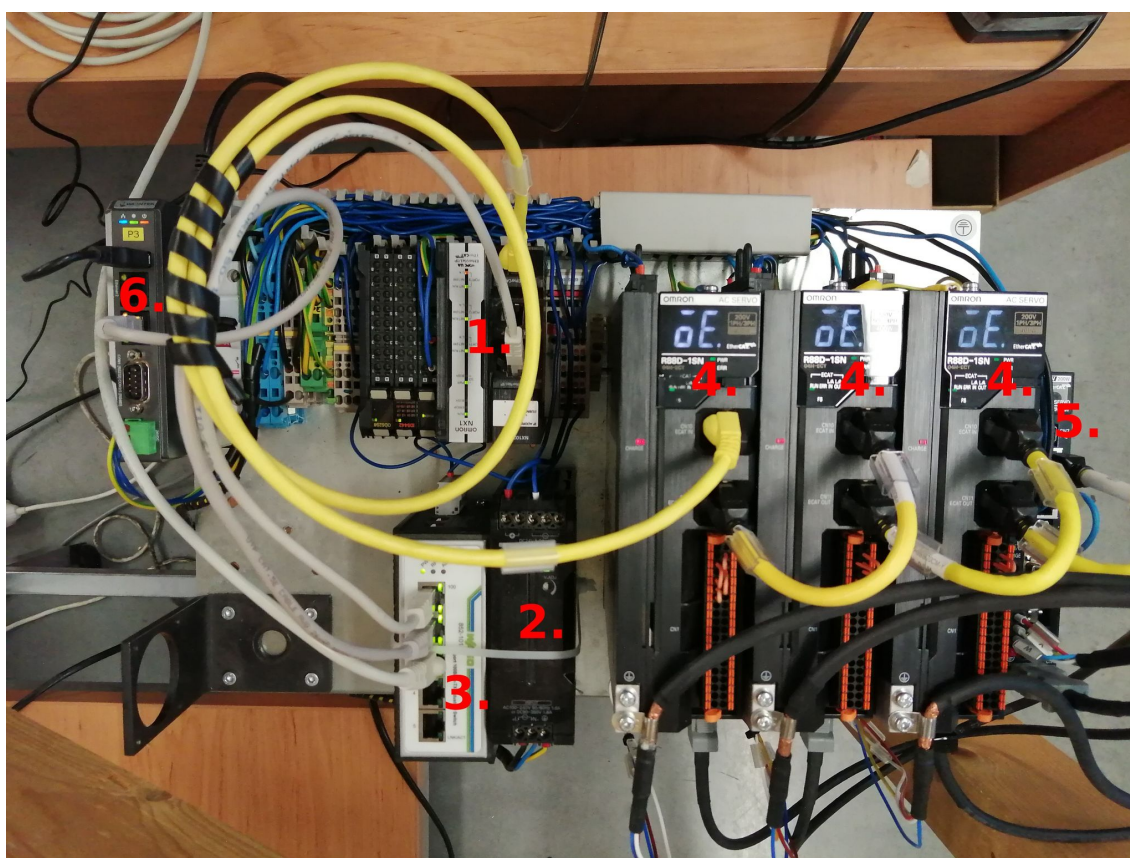
Dosavadní funkce kyvadla umožňuje pouze vykreslování obrázců, které jsou předem vytvořené v externím programu a uložené na SD kartu v PLC. Těchto obrázců je nyní omezené množství, především proto, že jejich návrh v dosavadním návrháři firmy VÚTS je velice časově náročný. Například návrh obrazce s obličejem šejka 1.5 zabral přibližně šest hodin. Zatím není možné, aby si uživatelé kyvadla mohli nakreslit libovolný obrazec, který by se poté vykreslil na pískovou plochu. Pokud by tato aplikace byla interaktivní, bylo by to nejen zajímavější z technického hlediska, ale v případě zájmu jistě i přínosnější pro klienty nemocnice z hlediska terapeutického.



Obrázek 1.5: Obrazec s obličejem šejka [1]

Hlavním cílem pro vylepšení je zajistit přenos dat z externího zařízení, např. notebooku nebo tabletu, jejich následné zpracování v PLC a navržení řídicího algoritmu pohonů při použití těchto dat. Jako datový vstup bude použit výstup bakalářské práce kolegy Davida Dostrašila, který zajišťuje softwarové řešení návrháře pro ruční kreslení obrazců a zpracování dat. Po realizaci komunikace mezi zařízeními, na kterém bude software pro kreslení, a PLC pro řízení pohybu kyvadla, bude možné z obslužného HMI panelu ovládat kyvadlo tak, aby vykreslilo do písku obrazec nakreslený uživatelem.

2 Teoretický rozbor jednotlivých komponentů a použitého softwaru



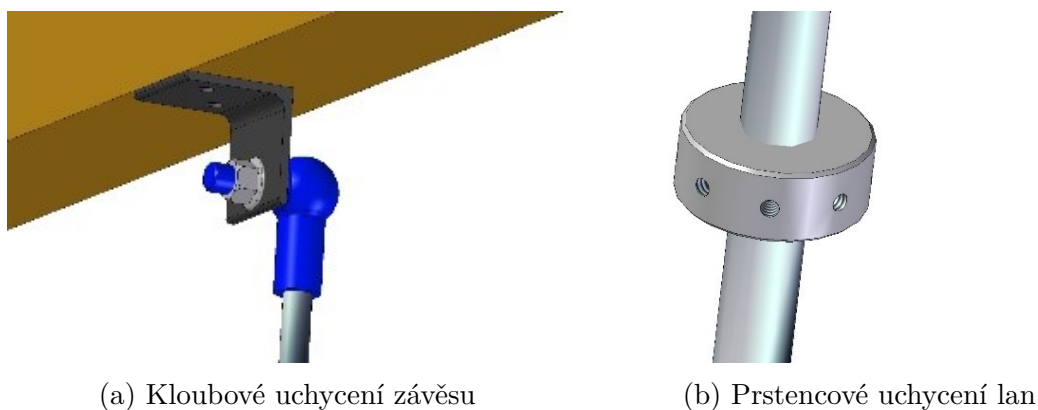
Obrázek 2.1: Rozvaděčová deska s komponenty (1. PLC NX102-1220, 2. DC zdroj, 3. Ethernet switch, 4. Měníče pro servopohony řídicích lan, 5. Měníč pro servopohon hrotu, 6. HMI panel)

2.1 Mechanická konstrukce modelu kyvadla

Model kyvadla (Na obrázku 1.4b) je realizován v poměru 1:10 vůči reálnému kyvadlu. Jeho konstrukce je tvořena dřevěnými hranoly a skládá se z podstavy, na které jsou připevněny navíjecí jednotky se servopohony, a z vrchního nosného rámu, na který je připevněn kyvadlový závěs. Pod kyvadlem je na stole umístěna železná

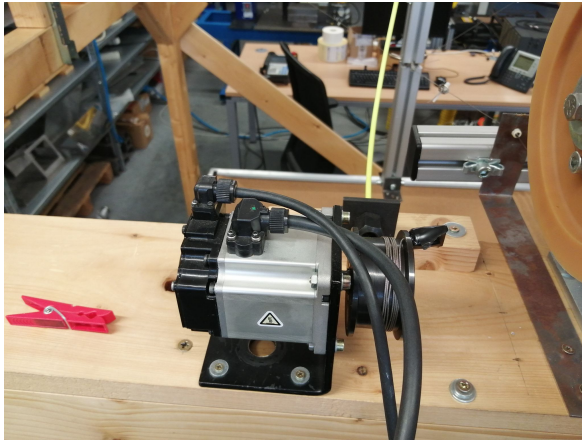
nádoba s pískem obdélníkového tvaru, která stojí na polohovatelných nohách, aby bylo možné vyrovnat případné nerovnosti. Na nádobě s pískem jsou umístěna LED svítidla, aby bylo možné alespoň z části simulovat vliv stínů na výsledné obrazy v písku (Obrázek 1.4).

Závěs modelu o délce 1,71 m je upevněn na vrchním nosném rámu pomocí železného úhelníku, na kterém je kulový kloub (Obrázek 2.2a). Montáž kyvadla na kulový kloub zajišťuje tři stupně volnosti a umožňuje vychylovat kyvadlo po kompletní kulové ploše. Jako závěs modelu kyvadla je použit tuhý hliníkový profil od značky Item. Ocelová lanka jsou uchycena k závěsu pomocí úchytek, které jsou připevněny ve spodní třetině k rotačnímu prstenci se závity (Obrázek 2.2a), který je umístěn mezi dva kusy hliníkového profilu. Na hliníkový profil je možné přidělat několik železných závaží o celkové váze 60 kg. Závaží slouží jako náhrada reálného skleněného komponentu, aby bylo možné otestovat chování kyvadla při zátěži v měřítku. Vychýlení kyvadla je při plné zátěži omezeno do kružnice s poloměrem 120 mm aby nedocházelo k přetěžování lan. Na konci profilu se nachází mechanismus kreslicího hrotu. Hrot je připevněný na laně, proti kterému působí silou pružina.

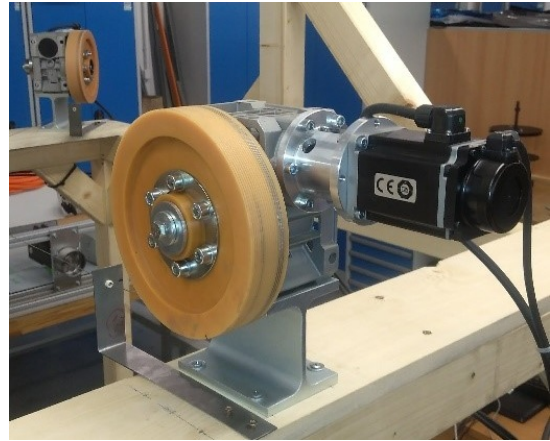


Obrázek 2.2: Modely uchycení [1]

Boční tažná lana jsou vedena skrze kruhovou keramickou průchodku a druhým koncem navinuta na navíjecí jednotky (Obrázek 2.3b), které jsou umístěny na rámu podstavy konstrukce. Navíjecí jednotka je tvořena navíjecím bubnem se spirálovou drážkou, šnekovou převodovkou a servopohonem s absolutním enkodérem. Kvůli vibracím a nežádoucím jevům, které způsobuje dřevěný materiál konstrukce, musel být rám vyztužen hliníkovými profily, připevněnými na vršky navíjecích jednotek. Navíjecí jednotku trnu (Obrázek 2.3a) tvoří pouze hladký navíjecí buben, ke kterému je ocelové lano přivedeno skrze bowden, a servopohon s inkrementálním enkodérem.



(a) Jednotka pro trn



(b) Jednotka pro boční lana

Obrázek 2.3: Konstrukce navíjecích jednotek [1]

2.2 Popis použitého PLC a jednotlivých modulů

Pro řízení modelu kyvadla je použito modulární PLC řady NX102-1220 od firmy Omron. Napájení 24 V je přivedeno z externího zdroje na odnímatelný napájecí modul PLC. Pro zachování času, uložených proměnných a programů po delší dobu, než je schopen zajistit vnitřní kondenzátor, může být do jednotky vložena záložní baterie. Součástí řídicí jednotky je modul se třemi vstupy pro konektor RJ45, který umožňuje připojení jednoho datového kabelu pro EtherCAT komunikaci a dvou kabelů pro EtherNet/IP komunikaci. Do PLC může být vložena SD karta, která umožňuje ukládání a načítání dat mimo paměť jednotky a práci se soubory. Bližší informace lze nalézt na stránkách výrobce [3].

Základní parametry PLC:

- Programová paměť: 5 MB
- Paměť vyhrazená pro proměnné: 33,5 MB
- Doba vykonání cyklu: od 1 ms
- Maximální počet os: 15 (8 synchronizovaných, 4 PTP, 3 virtuální)
- Maximální počet lokálních modulů: 32

K řídicí jednotce jsou připojeny dva moduly, ID5442 pro digitální vstupy a OD5256 pro digitální výstupy. Moduly jsou v dosavadní verzi modelu nevyužity, avšak do budoucna mohou sloužit pro ovládání nasvícení plochy, případně pro vstupy tlačítek.

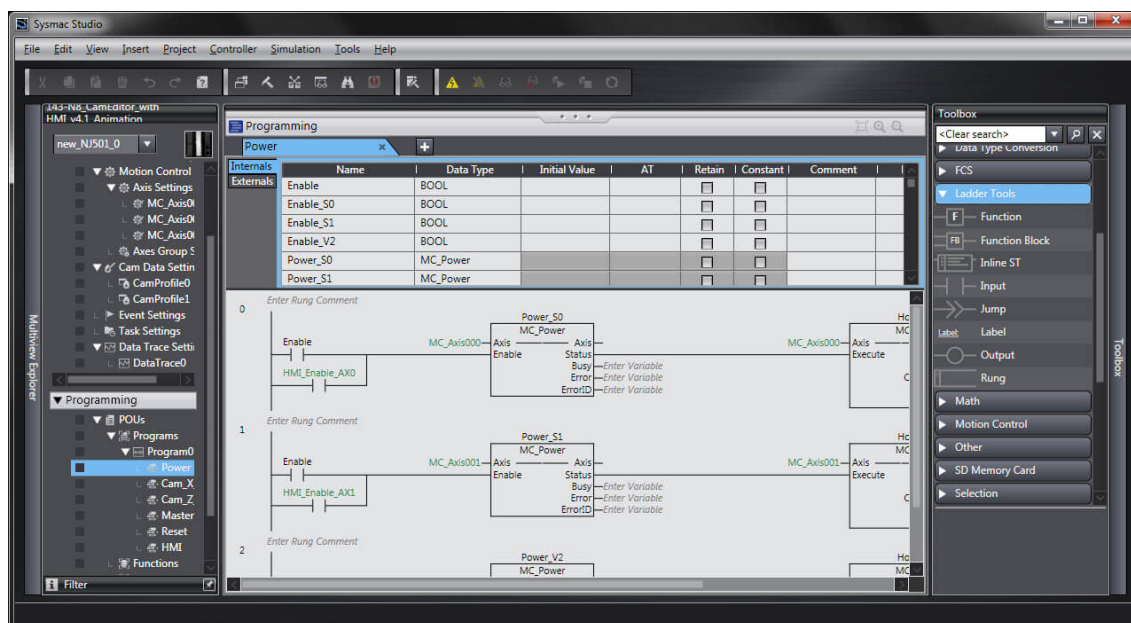


Obrázek 2.4: Použité PLC NX102-1220 [3]

2.3 Software použitý pro programování PLC

Při tvorbě programu bylo vzhledem k typu a značce PLC použito integrované vývojové prostředí Sysmac Studio verze 1.45. Sysmac Studio umožňuje programování logiky, ovládání pohybu a regulaci pohonů, využití bezpečnostních funkcí při přidání safety modulu, práci s kamerovými systémy, ale také programování a vizualizaci HMI.[3]

Sysmac Studio nabízí možnosti programování v jazyce kontaktních schémat a ve strukturovaném textu. Jelikož řízení trajektorie kyvadla vyžaduje velké množství numerických výpočtů, instrukcí pro řízení pohonů a je poměrně programově rozsáhlé, byl z hlediska přehlednosti a osobních preferencí zvolen pro programování strukturovaný text.



Obrázek 2.5: Vývojové prostředí Sysmac Studio [3]

Sysmac Studio obsahuje rozsáhlou knihovnu s funkcemi a funkčními bloky, která obsahuje nejen standardní bloky, jako například časovače a čítače, ale pro tuto aplikaci velmi důležité bloky pro řízení servopohonů a práci s daty. Jednotlivé funkce a funkční bloky jsou do podrobnosti popsány v manuálu pro Sysmac Studio a jsou vždy doplněny o příklad použití. Pro synchronní řízení servopohonů jako elektronických vaček je možnost editování profilu pohybu pomocí grafického CAM editoru. Tato funkce je efektivní pro použití u jednoduchých, opakujících se výrobních procesů. Jelikož vykreslování kyvadlem vyžaduje proměnný pohybový profil na každý obrazec, je nutné pohyb os řešit výpočetně v reálném čase. Pro ověřování správnosti programu má Sysmac možnost offline simulace. Při testování správnosti průběhu je velmi užitečný nástroj Data Trace, který umožňuje zaznamenávání proměnných do grafů, ze kterých se pak dá zjistit například ve kterém čase vznikla chyba.[4]

2.4 Stejnoseměrný zdroj pro napájení komponentů

Pro napájení jednotlivých komponentů je využit jednofázový stejnosměrný zdroj S8VK-G12024 od značky Omron. Výstupní napětí zdroje je 24 V a maximální proudový odběr 5 A. Jmenovitý odebíraný výkon je 120 W s účinností až 89 %. Zdroj disponuje ochranou proti přetížení a přepětí a funkcí zesílení výkonu až na 120 %. Zajišťuje napájení PLC, HMI, ethernet switche a frekvenčních měničů. Spotřeba stejnosměrné energie celého systému dosahuje přibližně 60 W. Více informací o zdroji lze zjistit na stránkách výrobce [3].



Obrázek 2.6: Stejnsměrný spínaný zdroj S8VK-G12024 [3]

2.5 Průmyslový Ethernet switch

V rozvaděči modelu je umístěn průmyslový switch typu 852-101 od značky Wago, který propojuje PLC, HMI panelu a obslužného PC do hvězdicové topologie. Switch je vybaven pěti porty 10/100Base-TX, umožňuje redundantní napájení a sledování funkčnosti pomocí relé. Více informací v datasheetu na stránkách výrobce [5].



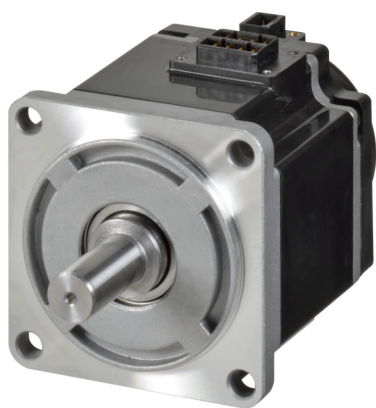
Obrázek 2.7: Průmyslový switch Wago 852-101 [5]

2.6 Servopohony a měniče

Navíjení bočních ovládacích lan zajišťují servopohony typu R88M-1M40030T-S2 od značky OMRON (viz obrázek 2.8a). Jedná se o synchronní servopohony s permanentními magnety na rotoru. Řízení otáček motorů zajišťují jednofázově napájené frekvenční měniče řady R88D-1SN04H-ECT (viz obrázek 2.8b). Protože motor není schopen vyvinout dostatečný moment k pohybu kyvadla a není zde požadavek na vysoké otáčky, je jeho krouticí moment zvýšen v poměru 1:60 pomocí šnekové převodovky. Více informací o servopohonu lze nalézt na stránkách výrobce [3].

Parametry motorů pro tažná lana:

- Jmenovité napětí: 230 V
- Výkon: 0,4 kW
- Jmenovité otáčky: 3000 ot/min
- Jmenovitý moment: 1,27 Nm
- Enkodér: Absolutní
- Rozlišení enkodéru: 23 bitů



(a) R88M-1M40030T-S2



(b) R88D-1SN04H-ECT

Obrázek 2.8: Servopohon a frekvenční měnič [3]

Navíjení lana pro kreslící trn zajišťuje servopohon typu R88M-K20030H-S2 od značky OMRON (viz obrázek 2.9a). Jedná se o synchronní servopohon s permanentními magnety na rotoru. Řízení otáček motoru zajišťuje jednofázově napájený frekvenční měnič řady R88D-KN02H-ECT (viz obrázek 2.9b). Více informací o servopohonu

lze nalézt na stránkách výrobce [3].

Parametry motoru pro zdvih kreslicího trnu:

- Jmenovité napětí: 230 V
- Výkon: 0,2 kW
- Jmenovité otáčky: 3000 ot/min
- Jmenovitý moment: 0,64 Nm
- Enkodér: Inkrementální
- Rozlišení enkodéru: 20 bitů



(a) R88M-K20030H-S2



(b) R88D-KN02H-ECT

Obrázek 2.9: Servopohon a frekvenční měnič [3]

Frekvenční měniče a PLC mezi sebou komunikují skrze průmyslovou sběrnici s protokolem EtherCAT. Tento protokol umožňuje díky rychlému komunikačnímu cyklu velmi přesné řízení servopohonů v reálném čase a zároveň je vysoce odolný vůči chybám. Sběrnice je zapojena do sériové topologie. Z hlediska správného řízení je nutné dbát dodržení správného pořadí vstupů a výstupů při zapojování sběrnice.

2.7 Použitý HMI panel

Pro obsluhu modelu kyvadla byla možnost výběru ze dvou HMI panelů. Malého kompaktního dotykového panelu OMRON s přímou podporou programování v Sysmac Studiu, nebo panelu mTV-100 bez displeje od firmy Weintek s možností připojení

monitoru a myši. Kvůli rozměrům a větší přehlednosti vizualizace byl zvolen panel mTV-100. Pro panel byla vytvořena vizualizace v programu EasyBuilder Pro. Podrobné informace lze najít na stránkách výrobce [6].



Obrázek 2.10: HMI bez displaye mTV-100 [6]

2.8 Rozbor jednotlivých úloh pro řešení

Pro snadnější definování jednotlivých úloh a požadavků pro jejich vyřešení bylo zvoleno členění do následujících dílčích podkategorií.

Řídicí algoritmus

Pro správné řízení pohybu kyvadla musí být zajištěn synchronní pohyb všech čtyř servomotorů v závislosti na zpracovaných datech. Před vykonáním pohybu musí být tažná lana dopnutá a řídicí jednotka musí mít přehled o délce lan a o referenčních nulových bodech. Toho se dá dosáhnout dopnutím lan pomocí pohybových instrukcí, nebo kalibrační sekvencí a následným nastavením pozice Home.

Data obrazce jsou generována ve 2D rovině (x, y) , která odpovídá kreslicí ploše. Pohyb hrotu kyvadla a úchyty tažných lan však probíhá po kulové ploše ve 3D rovině (x, y, z) . Z tohoto důvodu je nutné přepočíst požadovanou polohu hrotu kyvadla na kreslicí ploše na požadovanou délku všech lan.

Nová aplikace pro návrh obrazců umožňuje kreslit ostré hrany i více-tahové obrazce. Řídicí algoritmus musí tyto události rozpoznat a adekvátně na ně zareagovat. Pro interpolaci dat je nutné zvolit vhodnou metodu proložení.

Práce s daty

Při práci s daty je nutné zvážit, jaký datový formát je nejvhodnější, jaké informace mají data obsahovat, a jaký způsob přenosu z externího zařízení s kreslícím softwarem do řídicího PLC bude použit. Stěžejní je co nejkvalitnější vykreslení obrazu, ale zároveň co nejmenší velikost a množství dat, jednak aby nebyla příliš velká časová prodleva mezi odesláním dat a vykreslením, ale také, aby nedošlo k přeplnění omezené paměti PLC. Data je v PLC nutné uložit a vhodně zpracovat pro následné použití v řídicím algoritmu. Kvalita vykreslování může být ovlivněna volbou metody interpolace.

Obsluha modelu kyvadla

Obsluha bude prováděna pomocí obslužného HMI panelu a notebooku s návrhovou aplikací. Vizualizace v HMI by měla být co nejpřehlednější, zároveň by však měla umožnit manipulaci a nastavení parametrů kyvadla v plném rozsahu. Pokud by někdy došlo k ovládání kyvadla laickou osobou, změna kritických parametrů bude chráněna přístupovým heslem. K vizualizaci je vhodné vytvořit manuál na obsluhu.

Bezpečnostní funkce

Největším bezpečnostním rizikem je rychlý neočekávaný pohyb kyvadla, nebo přetržení tažných lan, ať už při chybě ve zpracovaných datech, nebo při selhání řídicího algoritmu. Je nutné navrhnout a implementovat programové funkce, které tomu zabrání.

Doplňkové funkce

Samotný program pro řízení může být doplněn o další funkce, které usnadňují ovládání kyvadla. Mezi doplňkové funkce může spadat například jogování jednotlivých pohybových os, nebo pohyb kyvadla na určité souřadnice, ať už kvůli usnadnění mazání, nebo kvůli výhledu po dokončení kreslení.

3 Postupy pro vyřešení jednotlivých úloh

3.1 Počáteční nastavení parametrů nového projektu v Sysmac Studiu

Po založení projektu v Sysmac Studiu je nutné nejprve definovat hardwarovou konfiguraci a několik síťových a softwarových parametrů.

Projekt je možné založit pomocí tlačítka New Project a manuálně definovat parametry PLC, jako jsou typ a verze. Zde se jedná o kontrolér NX102-1220 ve verzi 1.33. Manuální založení je vhodné pokud není okamžitý přístup k PLC a uživatel si chce předpřipravit program. Pro následnou komunikaci s PLC v režimu Online je nutné definovat parametry připojení.

Pohodlnější volbou je připojení skrze Ethernet, nebo USB pomocí tlačítka Connect to Device, jehož funkce po připojení automaticky založí nový projekt se správnými parametry kontroleru. Sysmac Studio nabízí možnost přímého i vzdáleného připojení pomocí Ethernetu a USB. Protože Ethernet je veden skrze switch, byla zvolena možnost vzdáleného ethernetového připojení.

V novém projektu je nutné definovat připojené moduly a zařízení v síti EtherCAT. Obojí lze opět provést manuálně vyhledáním modulů a zařízení v katalogu, nebo po připojení ke kontroleru v režimu Online pomocí funkce rozpoznání a následného přepsání.

V nastavení kontroléru je možné nastavovat chybová hlášení, provádět změnu IP adres jednotlivých portů, měnit nastavení funkcí SD karty a definovat parametry FTP serveru, který je využit pro přenos dat.

3.2 Nastavení pohybových os

Pro řízení pohonů v programu je nutné je přiřadit k pohybovým osám. Sysmac Studio umožňuje vytvářet jak osy reálné s přiřazeným servopohonem, tak osy virtuální, které v této úloze slouží pro přepočítání dat na délku lan a synchronizaci pohybu. V programu jsou definovány čtyři reálné pohybové osy: Rope0, Rope1, Rope2 a Thorn. Dále jsou zde definovány tři virtuální osy: VM, X a Y.

Virtuální pohybové osy

Virtuální pohybové osy pracují jen v programu a není nutné jim přiřazovat servo-

kontroler. Osa VM, neboli virtuální master, slouží při vykreslování jako časová osa, na kterou jsou výpočetně vázány ostatní osy.

Virtuální osy X a Y figurují v programu jako kartézský souřadný systém pro kreslicí plochu a jsou přidány do jedné skupiny os GroupXY, která může být následně v programu svázána či rozvázána dle potřeby. Koncové body os jsou nastaveny na kladnou i zápornou hodnotu 150 mm, aby nedošlo k příliš velkému vychýlení kyvadla.

Reálné pohybové osy typu Rope

Pohybové osy Rope0, Rope1 a Rope2 jsou přiřazeny servopohonům pro tažná boční lana. Jejich rozmístění je vidět na obrázku 3.1. Každé ose je přiřazen odpovídající servokontroler.



Obrázek 3.1: Přirazení pohybových os servopohonům

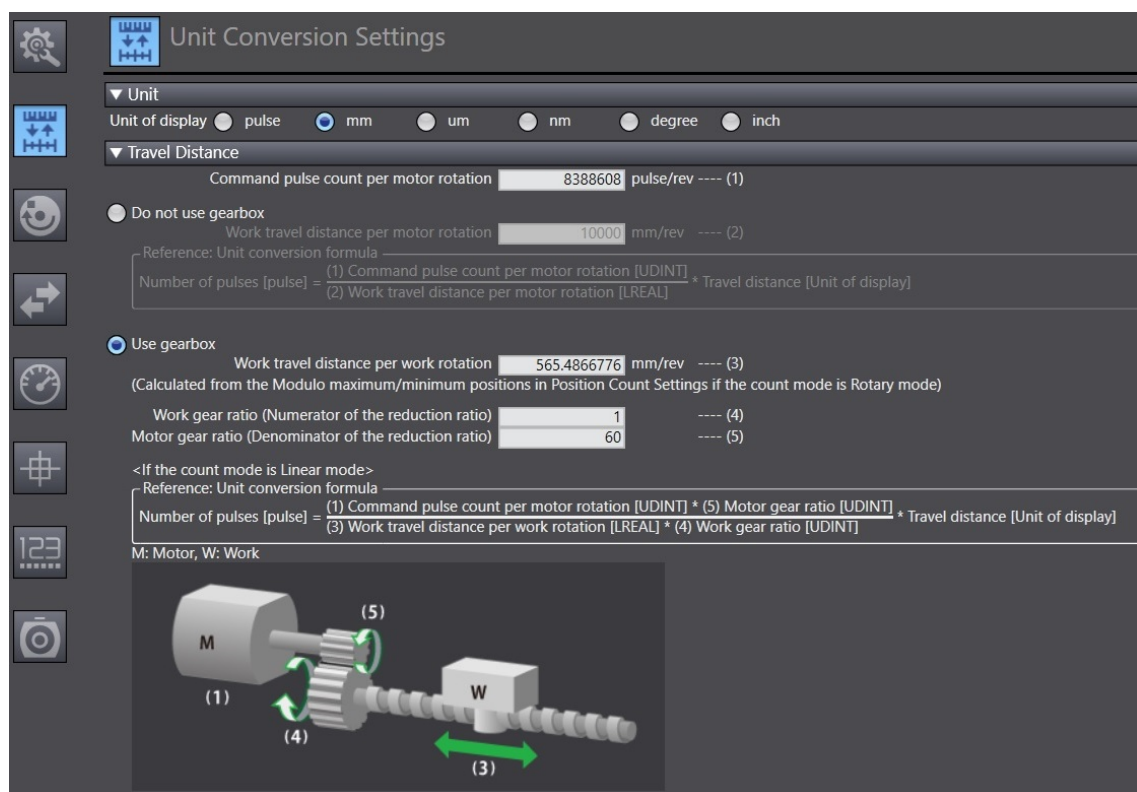
Parametr počtu pulzů na otáčku je dán enkodérem, který má u všech tří pohonů rozlišení 23 bitů, což odpovídá 8388608 pulzům za otáčku. Enkodér je nutné definovat jako absolutní. Protože navíjení lan je provedeno přes převodovku, je nutné zvolit použití převodovky. Pro použitou převodovku je definován převodový poměr

1:60 a pracovní vzdálenost motoru, která odpovídá délce lana navinutého do drážky cívký v rámci jedné otáčky. Pro zjištění této vzdálenosti je využita znalost poloměru naměřeného od středu cívký ke středu navinutého lana, která odpovídá vzdálenosti 90 mm. Z poloměru je za využití známého vzorce pro obvod vypočtena hodnota 565,4866776 mm. Pro co nejvyšší přesnost pohybu není vhodné příliš velké zao-krouhlování, avšak přesnost bude ovlivněna hlavně mechanickými vlastnostmi lan a převodovky. Nastavení hodnot a jejich přepočty z pulzů enkodéru jsou vidět na obrázku 3.2.

Fyzikální omezení rychlosti, zrychlení a momentu osy jsou spíše pro případ, že operátor kyvadla chybně zadá požadovanou rychlost, protože při vykreslování obrazce se rychlost řídí výpočetně z optimalizovaných dat a jakákoliv chyba, která by mohla vést k poruše, je detekována bezpečnostními funkcemi. Mimo automatické vykreslování obrazce jsou hodnoty nastaveny na konstantní hodnotu. Maximální rychlost je nastavena na hodnotu 100 mm/s, zrychlení na hodnotu 100 mm/s² a moment je omezen na 40 % jmenovité hodnoty.

Softwarové omezení koncových bodů je dáno délkou lana. Záporný limit je nastaven na 0 mm a kladný limit na 1500 mm.

Jako homovací metoda bylo zvoleno nastavení nulové pozice.



Obrázek 3.2: Příklad nastavení pohybové osy s převodovkou

Reálná pohybová osa typu Thorn

Pohybová osa typu thorn je přiřazena servopohonu pro zdvih kreslicího trnu.

Parametr počtu pulzů na otáčku je dán enkodérem s rozlišením 20 bitů, což odpovídá 1048576 pulzům za otáčku. Enkodér je nutné definovat jako inkrementální. Pro zjištění pracovní vzdálenosti motoru za otáčku je využita znalost poloměru naměřeného od středu cívky ke středu navinutého lana, která odpovídá 25 mm. Z poloměru, za využití známého vzorce pro obvod, je vypočtena hodnota 157,1 mm.

Maximální rychlost je nastavena na hodnotu 100 mm/s, zrychlení na hodnotu 100 mm/s² a moment je omezen na 150 % jmenovité hodnoty. Momentové omezení může přesáhnout jmenovitou hodnotu, protože servopohon je možné dočasně přetížít. Toho je využito například při kalibraci kreslicího trnu, protože motor musí vyvinout dostatečný moment, aby překonal sílu, kterou proti němu působí pružina.

Softwarové omezení koncových bodů hrotu je nastaveno na -40 mm a 40 mm.

Jako homovací metoda bylo zvoleno nastavení nulové pozice.

3.3 Nastavení servopohonů a jejich regulace

Pro zajištění kvalitního řízení pohybu kyvadla je nutné nastavit regulaci jednotlivých servopohonů. Špatné seřízení regulace může vést k nežádoucím jevům, jako jsou například vibrace, problémy s přesností pohybu, nebo nežádoucí zvukové projevy mechanických částí vlivem rezonance. Pro regulaci servopohonů je využíváno kaskádní regulace. Kaskádní struktura obsahuje vnitřní rychlostní regulační smyčku s PI regulátorem a poziční vnější smyčku s P regulátorem. Regulační struktura je doplněna o dopředné vazby a frekvenční filtry. Podrobný popis jednotlivých částí regulace lze nalézt v [7].

Sysmac Studio nabízí pro nastavení regulace několik možností odvíjejících se od znalosti dynamických a mechanických vlastností systému a typu použitého servosystému.

Nastavení regulace pro servopohon kreslicího trnu

Servosystém pro pohyb kreslicího trnu je řady G5-series. Pro nastavení regulace pro tuto řadu servosystémů umožňuje Sysmac Studio použít funkci autotuning v reálném čase, nebo manuální nastavení regulace.

Funkce autotuning umožňuje automatické nastavení parametrů regulace na základě odhadu setrvačnosti zátěže a automatického nastavení parametru Machine Rigidity. Machine Rigidity definuje, jakou reakci musí servopohon vyvolat při působení zátěže na hřídel. Čím větší je hodnota parametru, tím větší jsou zesílení regulátoru a regulace reaguje tvrději na poruchové veličiny. Pro autotuning je možné nastavit rozsah zkoušených hodnot, který je předem definován pro určité typy poháněných soustrojí. Autotuning následně při pohybu servopohonu zvyšuje hodnotu parametru Machine Rigidity, čímž se zvyšují i zesílení regulátoru, až dokud nedosáhne žádaných parametrů překmitu, doby ustálení a hladiny momentových vibrací. Pro potlačení mechanických vibrací soustrojí je možné zapnout adaptivní filtr, který pracuje jako pásmová zádrž pro frekvence způsobující vibrace. Pokud není funkce autotuning schopná zajištění požadované funkce regulovaného systému,

je pak nutné parametry nastavit ručně. Podmínky při kterých autotuning neprovede korektně a další informace lze nalézt v datasheetu servosystému G5-series na stránkách výrobce [3].

Další možností nastavení regulace servopohonu je manuální nastavení. Zde je možné sledovat chování motoru a u toho měnit parametry regulace. Pro nastavení regulace je postupně zvyšován parametr Machine Rigidity, se kterým se mění i regulační konstanty. Pokud chování pohonu není žádoucí, parametry se mění ručně dle návodu od výrobce [3].

Protože autotuning nebyl schopen nastavit parametry vhodně, byly pro servopohon manuálně zvoleny parametry odpovídající hodnotě Machine Rigidity 15. To odpovídá parametrům regulátoru v tabulce 3.1. Poměr setrvačnosti zátěže ku setrvačnosti rotoru tzv. Inertia ratio je 900 %. Zároveň byl zapnut adaptivní filtr pro odstranění nežádoucích vibrací. Ostatní volby byly ponechány ve výchozím nastavení. Nastavení regulace není optimální především kvůli konstrukci a nelinearitě pružiny.

Parametry regulátorů	
Zesílení P regulátoru polohové smyčky	72
Zesílení P regulátoru rychlostní smyčky	40
Integrační složka I regulátoru rychlostní smyčky	14

Tabulka 3.1: Tabulka parametrů regulátoru

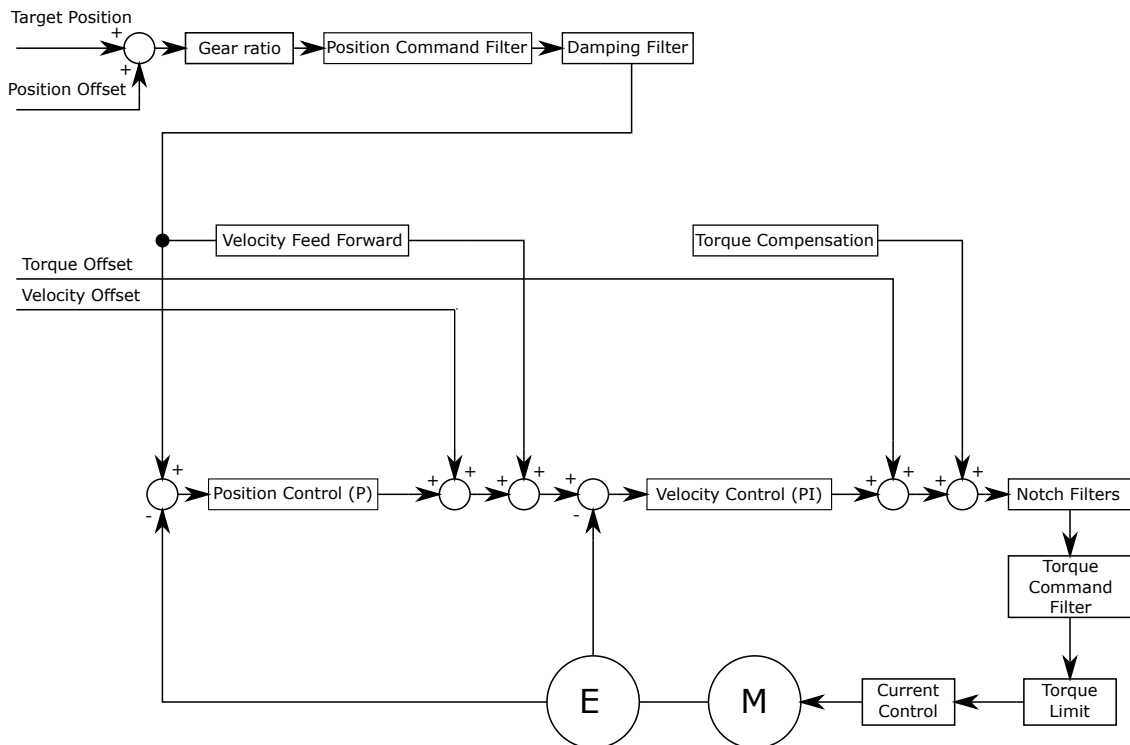
Nastavení regulace pro servopohony tažných lan

Servosystémy pro pohyb tažných lan jsou řady 1S-series. Pro nastavení regulace umožňuje Sysmac Studio využít dvě funkce autotuningu, Easy Tuning a Advanced Tuning, a jedné funkce manuálního nastavení. Easy Tuning hýbe motorem v cyklech a podle požadavků na dynamiku systému nastaví parametry regulátorů. Advanced Tuning funguje obdobně, akorát na základě pohybu pohonu, nebo zadaných hodnot uživatelem, vytvoří simulaci a regulační strukturu nastaví podle simulace. Jako poslední je zde možnost parametry opět definovat ručně, nebo podle předem definovaných profilů hodnot odvíjejících se od parametru Machine Rigidity. Všechny metody nastavení mohou být provedeny buď pro jednotlivé servopohony, nebo pro skupinu. Více informací o nastavení regulace pro servosystémy 1S-series lze nalézt v jejich datasheetu na stránkách výrobce [3].

Pro servopohony lan bylo použito manuální nastavení regulátorů u kterého byla vyhovující hodnota Machine Rigidity 13. Regulační struktura je vidět na obrázku 3.3. Parametry regulátoru jsou vidět v tabulce 3.2 Zbytek možností byl ponechán ve výchozím nastavení.

Parametry regulátorů	
Zesílení P regulátoru polohové smyčky	4.4
Zesílení P regulátoru rychlostní smyčky	21.9
Integrační složka I regulátoru rychlostní smyčky	5.5

Tabulka 3.2: Tabulka parametrů regulátoru



Obrázek 3.3: Regulační struktura servopohonu

Ostatní parametry servopohonu

Mimo parametrů pro regulaci je nutné nastavit směr otáčení pohonu při kladném směrovém příkazu. Pohony pro tažná lana jsou nastavena na směr otáčení proti směru hodinových ručiček. Pohon pro zdvih kreslicího trnu je nastaven na směr pohybu po směru hodinových ručiček. Ostatní možnosti byly ponechány ve výchozím stavu.

3.4 Struktura řídicího programu

Nový program pro řízení modelu kyvadla má obdobnou strukturu jako původní řídicí program, který je nyní využíván k řízení kyvadla v Abu Dhabi. To v případě zájmu

umožňuje jednoduchou implementaci nových řídicích algoritmů a funkcí z nového programu do původního.

Pro řízení kyvadla byly vytvořeny dvě verze programu. Jedna verze pracuje s lineární interpolací a druhá s interpolací kubickým splinem. Rozdíly jsou především ve způsobu zpracování dat. Vlivem pozdějšího vývoje programu s kubickou interpolací došlo k drobným změnám a vylepšením i v řízení vykreslování obrazců.

Tasky a programy

Kyvadlo je řízeno pomocí několika programů, z nichž každý je přiřazen do určité úlohy, neboli tasku, na základě jeho použití. PrimaryTask je periodicky vykonávaný task s periodou vykonání 2 ms. Použité PLC sice umožňuje využití nejrychlejší periody vykonání 1 ms, avšak kvůli výpočetní náročnosti při vykreslování obrazců byl čas periody překračován. Tento task má největší prioritu a jsou do něj přiřazeny všechny programy vykonávající řízení servopohonů a programy jim nadřazené. Dalším taskem je PeriodicTaskComms s periodou vykonání 10 ms a sloužící pro komunikaci s obslužnými zařízeními. V programu jsou využity také dva tasky spouštěné událostí, EventTask0 a EventTask1, které jsou spouštěny pouze při jejich zavolání ve verzi programu pracující s interpolací dat kubickým splinem.

Programy přiřazené do tasku PrimaryTask:

- MotionControl
- StateControl
- StInit
- StPowerOFF
- StStop
- StHome
- StMoveXY
- StMoveRopes
- StMoveThorn
- StJogXY
- StJogRopes
- StJogThorn
- StParking
- StDataRead

- StRefr
- StCam
- StCalibrationRopes
- StCalibrationThorn
- AlarmControl

Tento task využívá cyklického vykonávání programů od shora dolů a programy jsou tedy seřazeny v požadovaném pořadí. Program MotionControl vykonává všechny požadavky pro ovládání os a servopohonů a je tedy na prvním místě. Program StateControl řídí přechody mezi stavovými programy s předponou St na základě požadavků od ostatních programů. Stavové programy fungují na principu stavového automatu, tedy dojde k jejich kompletnímu vykonání pouze při splnění dané podmínky. Každý stavový program řídí určitou činnost kyvadla. Program AlarmControl zajišťuje detekci chyb, jejich výpis do proměnných a následné zastavení vykonávání programů.

Program přiřazený do tasku PeriodicTaskComms:

- HmiControl

Program HmiControl slouží výhradně ke zpracování proměnných pro komunikaci mezi obslužným HMI a PLC. Perioda tasku pro komunikaci nijak neovlivní kvalitu řízení a vykreslování a není tedy nutné, aby byl program přiřazen do nejrychlejšího tasku.

Program přiřazený do tasku EventTask0:

- DataToPoly

Task je zavolán pouze pokud dochází ke zpracování a přepočtu dat v programu StDataRead. Důvodem je, že přepočet je časově náročný, a při provedení v periodickém PrimaryTasku se nestihne vykonat a způsobí Timeout celého tasku. Program DataToPoly obsahuje funkce pro výpočet koeficientů kubických polynomů.

Program přiřazený do tasku EventTask1:

- ClearArrays

Task je volán vždy po dokončení vykreslování obrazce. V programu ClearArrays dochází ke smazání hodnot z polí pracujících s daty. Funkce smazání jsou vzhledem k velikosti polí časově náročné na provedení a při jejich zařazení do PrimaryTasku dojde k Timeoutu.

Struktura stavových programů

U všech programů s příponou St se opakuje stejná, nebo podobná struktura. Celý obsah programu je vždy vnořen do jedné podmínky příkazu IF. To zajistí, že se program vykoná pouze pokud stavová proměnná, řízená programem StateControl, odpovídá příslušnému stavu. Program StateControl zároveň řídí jestli se jedná o první průchod daným programem. Toho je možné využít při nastavení prvotních hodnot proměnných použitých v programu. Pro programy využívající stavových automatů s příkazem CASE je implementováno sledování prvního průchodu stavem, což opět může být použito pro nastavení počátečních podmínek, nebo pokud je požadavek na provedení příkazu jen jednou.

Algoritmus 1 Struktura programů St

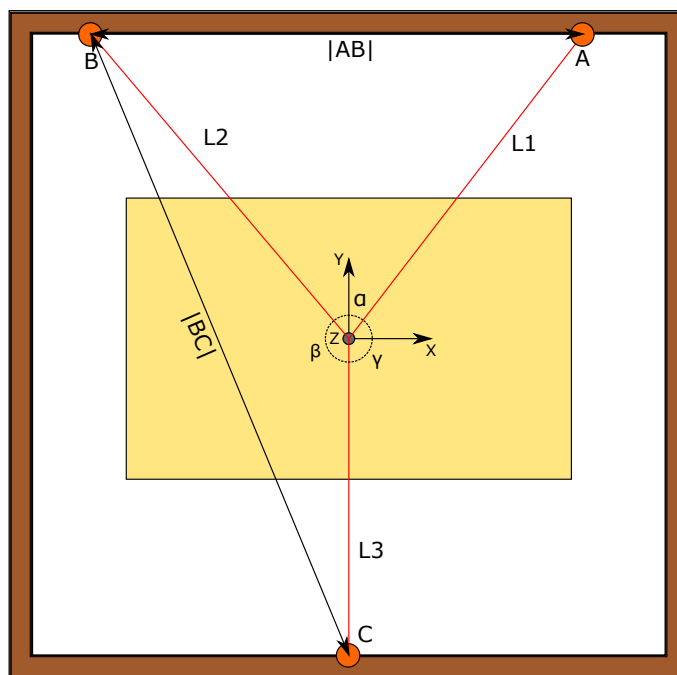
```
1: if Stavová proměnná = Stav pro daný program then
2:   if První průchod programem = TRUE then
3:     Nastavení počátečních podmínek programu
4:   end if
5:
6:   case Aktuální stav of
7:
8:     Stav1:
9:     if První průchod stavem = TRUE then
10:       Nastavení počátečních podmínek stavu;
11:       Zavolání funkce pouze jednou na začátku stavu;
12:     else
13:       Nový stav := Stav2;
14:     end if
15:
16:     Zbytek stavů
17:
18:   end case
19:   if Aktuální stav <> Nový stav then
20:     Aktuální stav := Nový stav;
21:     První průchod stavem := TRUE;
22:   else
23:     První průchod stavem := FALSE;
24:   end if
25: end if
26:
```

3.5 Počáteční průchod programem po zapnutí systému

Po zapnutí systému je nutné nejprve definovat všechny známé počáteční hodnoty pro výpočty a řízení. To je provedeno v prvním stavovém programu v pořadí StInit. Je zde definována délka lan v klidové poloze kyvadla, pozice úchyť lan v prostoru, výchozí hodnoty nastavitelné z HMI, programové koncové body os pro joggování a hodnoty pro bezpečnostní funkce.

Výpočet pozic úchyť lan v prostoru

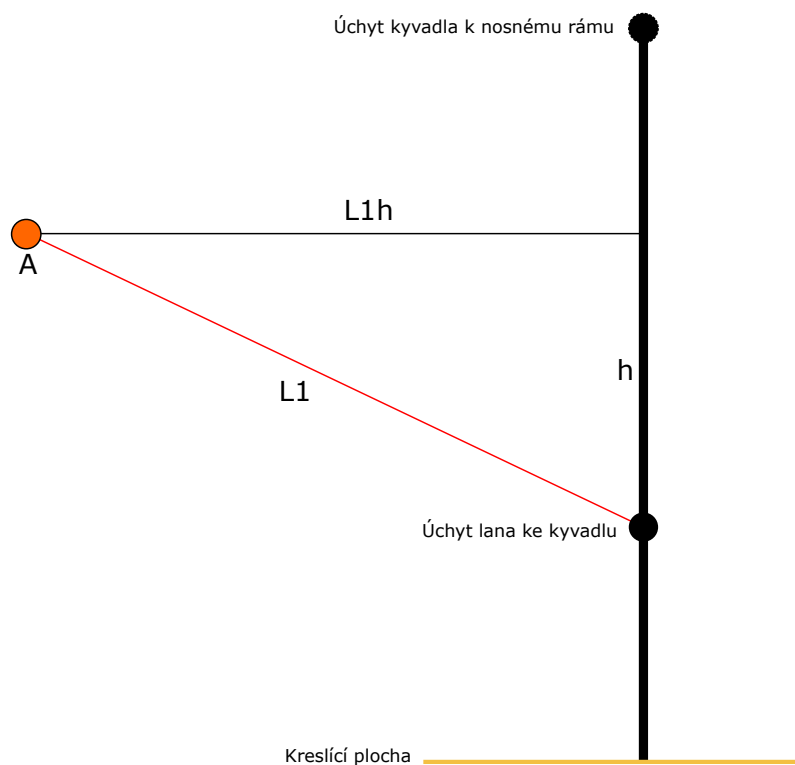
Protože data jsou přijímána ve formátu bodů v dvourozměrném souřadném systému (x, y) , je nutné přepočítat pozici hrotu kyvadla na délku lan (přepočet v 3.7). Pro výpočet délky lan v každém bodě trajektorie je potřeba nejprve znát pozici keramických průchodek v prostoru. Pro výpočet je uvažován trojrozměrný souřadný systém (x, y, z) s počátkem na pískové ploše pod kyvadlem v klidovém stavu.



Obrázek 3.4: Ilustrace půdorysu pískové plochy a rámu kyvadla s body ABC, které reprezentují průchočky lan

Pro výpočet pozic jednotlivých průchodek je využito znalosti následujících hodnot: délek jednotlivých lan v klidovém stavu ($L1$, $L2$, $L3$), vzdálenost mezi body $|AB|$ a $|BC|$ (viz obrázek 3.4), vzdálenost úchyty lan od pískové plochy, vzdálenost od závěsného kloubu k úchyty lan a vzdálenost průchodek od úchyty lan v ose z , označenou ve výpočtu jako h . Při úvaze, že je kyvadlo v klidovém stavu kolmé ke kreslicí ploše, je možné pomocí Pythagorovy věty vypočítat vzdálenosti L_h od

průchodek ke středu kyvadla ve vodorovné rovině (viz rovnice 3.1).



Obrázek 3.5: Trojúhelník vzniklý ze známých hodnot pro lano $L1$

$$L1h^2 = L1^2 - h^2 \quad (3.1)$$

Z vypočtených vzdáleností Lh je možné zjistit jednotlivé úhly mezi přímkami vzniklými z těchto vzdáleností. Protože navíc jednotky s průchodkami nejsou vzhledem k ose y symetrické, je nutné použít pro výpočet kosinovou větu (Viz. 3.2).

$$\alpha = \arccos \frac{-AB^2 + L2h^2 + L1h^2}{2 \cdot L2h \cdot L1h} \quad (3.2)$$

Z vypočtených úhlů je možné při použití jednoduchých goniometrických funkcí zjistit pozice průchodek v prostoru. Vzdálenost průchodek od kreslicí plochy v ose z je spočtena ze vzdálenosti úchytu lan od kreslicí plochy a vzdálenosti průchodek od úchytu v ose z .

Výpočet souřadnic bodu A může vypadat následovně:

$$A.x = \sin(180 - \gamma) * Lh1$$

$$A.y = \cos(180 - \gamma) * Lh1$$

Zapnutí servopohonů

Pro ovládání servopohonů je nejprve nutné zapnout jejich napájení. To je provedeno v programu StPower, který nastaví příkaz pro zapnutí všech servopohonů. Zapnutí je na základě příkazu vykonáno v programu MotionControl pomocí funkčního bloku MC_Power. Po úspěšném zapnutí se servokontrolery dostanou do stavu Ready a mohou přijímat instrukce k operacím se servopohony.

Klidový stav

Po počátečním průchodu a vždy po dokončení řízení kyvadla, přechází kyvadlo do klidového stavu, který je řízen v programu StStop. V něm dochází nejprve k vypnutí všech příkazů a ve verzi programu pracující s polynomy k zavolání úlohy Task1, která obsahuje program ClearArrays s funkcí Clear pro smazání zpracovaných dat. Na základě požadavku od uživatele je zde řízena stavová proměnná. Po průchodu programem StateControl dojde k jejímu přepsání a začne se vykonávat požadovaný stavový program.

3.6 Způsoby řízení a příkazy pro jednotlivé pohybové osy

Vykonávání všech pohybových instrukcí pro jednotlivé pohybové osy probíhá v programu MotionControl na základě příkazů z ostatních programů. Program je rozdělen do čtyř částí pro osy VM, Ropes, Thorn a skupinu os X a Y. Osy typu Ropes a jejich proměnné jsou přiřazeny do polí o třech prvcích, protože jejich řízení probíhá v jednom cyklu FOR.

Proměnné pro ovládání pohybových os a jejich datové struktury

Jednotlivé osy jsou řízeny pomocí funkčních bloků z knihovny Motion Control Instruction. Funkční bloky pohybových instrukcí využívají jako vstupy a výstupy globální proměnné, které jsou definovány pomocí uživatelských datových struktur pro přehlednější a snazší použití. Příkazy pro spuštění funkčních bloků jsou definovány proměnnou Cmd s datovou strukturou ST_Command. Každý člen struktury je přiřazen jako příkaz vždy jednomu vstupu funkčního bloku typu Execute. Názvy příkazů odpovídají vždy funkčnímu bloku a řízené ose.

Pro nastavení parametrů pohybové instrukce slouží proměnná Par s datovou strukturou ST_AxisParameter. Mezi nastavované parametry patří: HomeOffset, Position, Velocity, Acceleration, Deceleration a Jerk. Při každém požadavku na vykonání pohybové instrukce jsou tyto parametry buď definovány v programu, nebo nastavovány z obslužného HMI.

Informace o dokončení pohybových instrukcí z výstupu Done se ukládají do proměnných datové struktury St_AxisStatus.

Chybová hlášení z Error výstupů funkčních bloků se ukládají do proměnných

s datovou strukturou ST_Alarms, konkrétně do jejího členu McInstErr, který je definován jako datový typ Union. V datovém typu Union jsou definovány dva datové typy, jeden pro pole typu BOOL a jeden pro datový typ WORD. Protože proměnné v jednom Unionu mají stejnou paměť, je bitová reprezentace chyb převedena na číselnou hodnotu typu WORD, která definuje danou chybu. Stejného principu je využito i pro další chybová hlášení napříč všemi programy. Chybová hlášení jsou zpracovávána v programu AlarmControl.

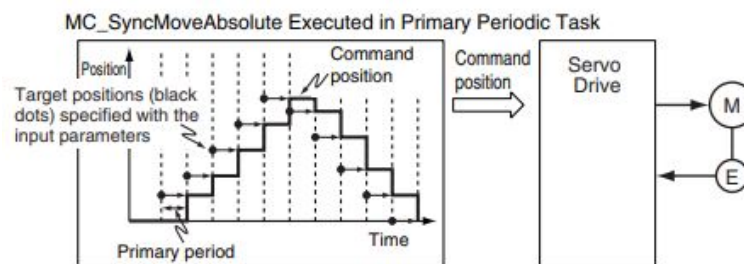
Použité funkční bloky

Všechny osy, ať už virtuální, nebo reálné, musí mít nastavenou výchozí pozici, ke které se vztahuje pozici při jejich řízení. Pro homování všech pohybových os je zde využito funkčního bloku MC_HomeWithParameter s parametrem HomingMode nastaveným na `_mcHomePreset`, který při vykonání nastaví v bodě, kde se osa momentálně nachází, nulovou pozici. Hodnotou parametru Offset je následně možné upravit aktuální pozici na nenulovou hodnotu.

Pro pohyb jednotlivých os do specifikované, absolutně zadané pozice je využit funkční blok MC_MoveAbsolute. Tomuto bloku jsou definovány parametry: Position, Velocity, Acceleration a Deceleration.

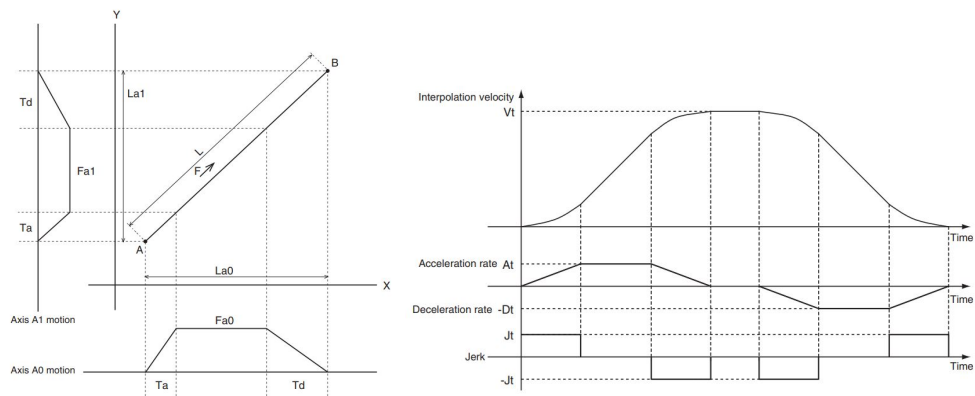
Přerušování pohybu během vykonávání instrukce může být nutné například při vzniku chyby. To je provedeno pomocí bloku MC_Stop, kterému je definován pouze parametr Deceleration.

Synchronizovaný pohyb os, nutný pro pohyb kyvadla, je proveden pomocí funkčního bloku MC_SyncMoveAbsolute, jehož vstupním parametrem je pouze Position. Blok pro synchronizovaný pohyb na rozdíl od MC_MoveAbsolute přijímá pohybové instrukce v každém cyklu a do servokontroléru je posílá vždy v následující periodě (viz obrázek 3.6). Maximální rychlost pohybu je dána omezením nastavení osy.



Obrázek 3.6: Funkce bloku MC_SyncMoveAbsolute [4]

Protože s virtuálními osami X a Y se pracuje jako se skupinou os GroupXY, musí pro ně být použity funkční bloky pro pohyb skupiny os, které jsou obdobné jako výše zmíněné bloky. Blok MC_MoveLinearAbsolute má parametry: Velocity, Acceleration, Deceleration, Jerk a Position. Protože funkční blok umožňuje interpolaci až pro 4 osy, je vstupem pro pozice pole o čtyřech prvcích, z nichž první dva jsou pozice os X a Y a zbytek zůstává prázdný. Trajektorie pohybu je lineárně interpolována z příkazových pozic všech os (viz obrázek 3.7a).



(a) Lineární interpolace dráhy (b) Vliv parametru Jerk na průběh rychlosti

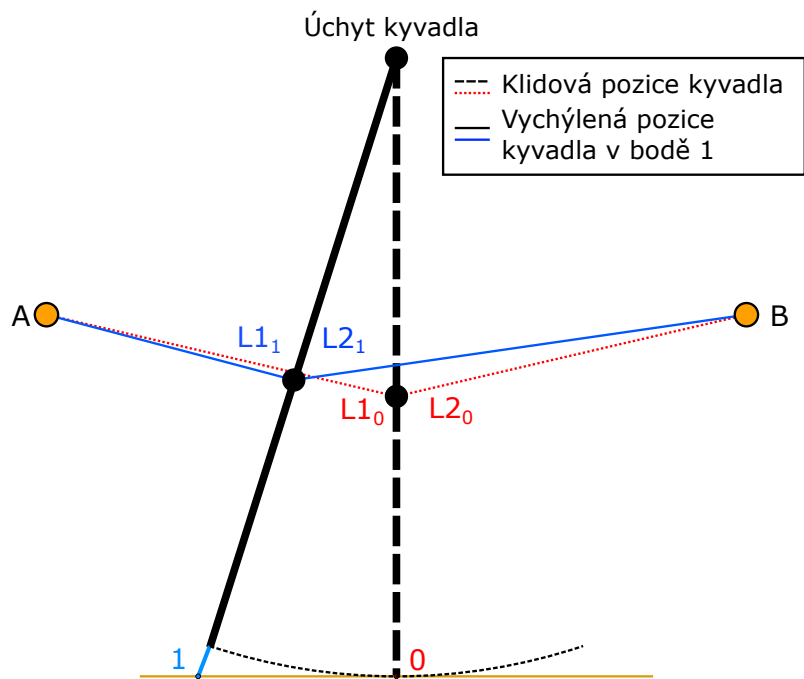
Obrázek 3.7: Funkce bloku MC_MoveLinearAbsolute [4]

Funkční bloky MC_GroupSyncMoveAbsolute a MC_GroupStop fungují stejně a jsou použity se stejnými parametry jako odpovídající bloky k ovládní samostatných os.

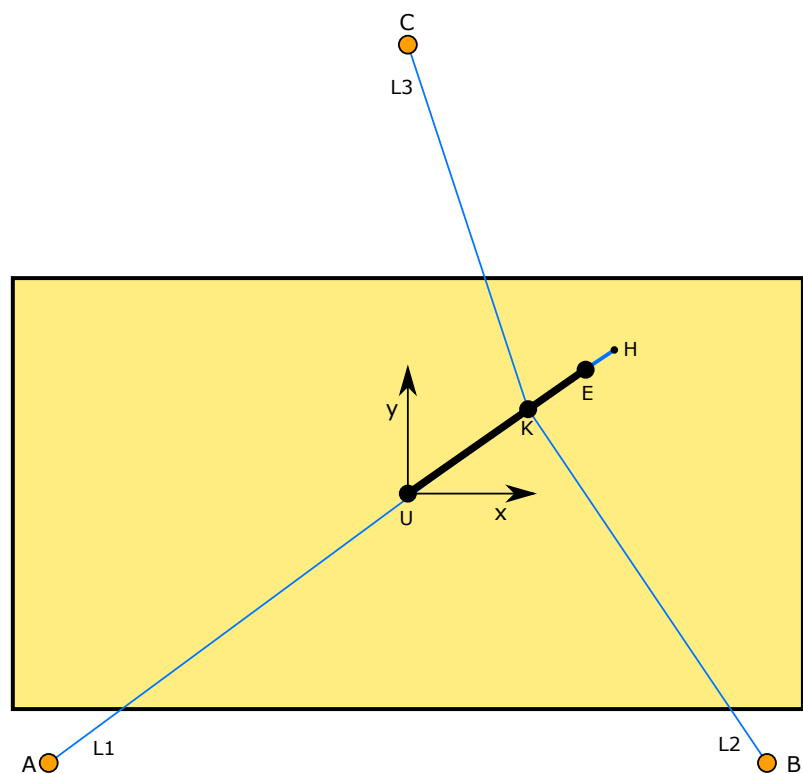
Pro použití os X a Y jako skupiny je nutné je nejprve svázat pomocí bloku MC_GroupEnable. Následné rozvázání při dokončení řízení je provedeno pomocí bloku MC_GroupDisable.

3.7 Vektorové přepočty dat na délku lan

Dodávaná data jsou ve formě bodů v kartézském souřadném systému, který odpovídá kreslicí pískové ploše pod kyvadlem. Protože pohyb kyvadla probíhá po kulové ploše (viz obrázek 3.8), je nutné přepočítat trajektorii z bodů na délku všech čtyř lan. To je provedeno vektorovými výpočty ve trojrozměrném souřadném systému. Vychýlené kyvadlo s body nutnými pro vektorový výpočet je vidět na obrázku 3.8. Výpočet je prováděn v každém cyklu v programu MotionControl pomocí uživatelských funkcí.



Obrázek 3.8: Ilustrace vychýlení kyvadla a změny délky lan při pohybu (pohled ze strany navíjecích jednotek A a B, třetí byla vynechána pro přehlednost)



Obrázek 3.9: Ilustrace vychýlení kyvadla s body definovanými ze známých délek (pohled shora)

Postup výpočtu vektorů

Prvním krokem výpočtu je vytvoření vektoru mířícího od úchyty kyvadla \mathbf{U} do bodu v písku, který představuje konec vysunutého kreslicího hrotu \mathbf{H} . Tento bod odpovídá souřadnicím ze zpracovaných dat (x, y) v rovině $z = 0$. Souřadnice vektoru v ose z je rovna vzdálenosti od úchyty kyvadla k písku. Ze vzniklého vektoru \mathbf{UH} je vypočtena jeho velikost a jednotkový vektor. Jednotkový vektor \mathbf{UH} udává směr vychýlení kyvadla v prostoru.

Vektor \mathbf{UH} je uložen ve formátu:

$$UH = (UH.x, UH.y, UH.z)$$

Délka vektoru (norma) je spočtena následovně:

$$UH.norma = \sqrt{UH.x^2 + UH.y^2 + UH.z^2}$$

Pomocí normy je vypočten jednotkový vektor $\mathbf{UH.e}$:

$$UH.ex = \frac{UH.x}{UH.norma}$$
$$UH.ey = \frac{UH.y}{UH.norma}$$
$$UH.ez = \frac{UH.z}{UH.norma}$$

Stejných výpočtů je využito i u následujících vektorů.

Pro výpočet vzdálenosti, o kterou musí být vysunut hrot při vychýlení kyvadla, je využita znalost proměnné velikosti vektoru \mathbf{UH} a konstantní velikosti vektoru \mathbf{UE} , který představuje vzdálenost od úchyty kyvadla ke konci hrotu v klidovém stavu. Při odečtení těchto hodnot je zjištěna vzdálenost \mathbf{EH} , o kterou musí být hrot vysunut.

$$EH = UH.norma - UE.norma$$

Z jednotkového vektoru \mathbf{UH} , při vynásobení vzdáleností \mathbf{UK} , je zjištěn vektor, který míří od úchyty kyvadla \mathbf{U} k úchyty lan \mathbf{K} . Protože pozice průchodek \mathbf{A}, \mathbf{B} a \mathbf{C} je definovaná vzdáleností od kreslicí plochy v ose z , je nutné k ní vztáhnout i pozici bodu \mathbf{K} přičtením délky \mathbf{UE} k souřadnici vektoru \mathbf{UK} v ose z . Z pozice průchodek \mathbf{A}, \mathbf{B} a \mathbf{C} a bodu \mathbf{K} v definovaném souřadném systému jsou následně spočteny velikosti vektorů \mathbf{AK}, \mathbf{BK} a \mathbf{CK} , které představují délku bočních lan.

Výpočet souřadnic vektoru \mathbf{AK} pro výpočet délky lana vypadá následovně:

$$AK.x = K.x - A.x$$

$$AK.y = K.y - A.y$$

$$AK.z = K.z - A.z$$

Přepočtení zajišťuje synchronizaci pohybu virtuálních os X a Y , jejichž poloha představuje požadovanou polohu v písku, a všech reálných pohybových os pro lana.

3.8 Kalibrace, Homing a doplňkové programy

Pro řízení kyvadla je nutné nastavit pro všechny osy nulovou polohu tak, aby vyhovovala řízení. Nulová pozice může být nastavena ručně skrze program StHome, nebo automaticky pomocí kalibračních sekvencí v programech StCalibrationRopes a StCalibrationThorn.

Ruční Homing

Program StHome funguje jako stavový automat, který reaguje na příkazy z obslužného HMI. Pokud přijde z HMI požadavek na nastavení nové homovací pozice, nastaví příkaz pro odpovídající funkční blok MC_Home v programu MotionControl s příslušným Offsetem nastaveným na HMI.

Pro homování trnu lze najet hrotem pomocí jogování tak, aby jeho konec byl těsně nad pískem a tuto pozici nastavit jako nulovou. Případné odchylky lze doladit po vizuální kontrole pomocí konstant v podobě proměnných uchovávaných v PLC a nastavitelných z HMI.

Protože nulová pozice os X a Y je definována jako klidový stav kyvadla s dopnutými lany, je možné ji manuálně nastavit až po tom, co jsou lana zkalibrována.

Tažná lana je možné dopnout jogováním na základě vizuální kontroly, nebo sledovat momenty servopohonů vypsané na obrazovce HMI. Po dopnutí lana se nastaví Offset jako změřená délka lana v klidovém režimu. To se opakuje pro všechna lana.

Automatické kalibrační sekvence

Kalibrační sekvence byly převzaty z původního programu kyvadla a byly upraveny a odladěny pro aktuální stav modelu.

Pro kalibraci trnu je využito principu maximální momentové hodnoty. V sekvenci je nejprve nastavena aktuální poloha jako nulová. Následně dostane trn příkaz k pohybu do kladné hodnoty 39 mm. Pokud kalibrační sekvence probíhá správně, nesmí se do této hodnoty dostat. Při dosažení hodnoty momentu 130 % se trn zastaví. Této pozici se nastaví hodnota Offset na 14,5 mm. Při provedení této sekvence odpovídá nula osy trnu přesně povrchu kreslicí plochy. Protože trn není nijak mechanicky zajištěn po vypnutí napájení a enkodér servopohonu je inkrementální, musí se sekvence provést při každém zapnutí modelu.

Před provedením kalibrace je vhodné, aby všechna lana byla stejně prověřena a pokud možno co nejméně. Protože kyvadlo nemusí být při zapnutí kalibrace v klidovém stavu, nejprve najede do nulové pozice skupiny os GroupXY. Následně se všechna lana prověsí o definovanou hodnotu. Po dokončení prověšení se lana začnou dopínat a měří se jejich moment. Ve chvíli, kdy první z lan dosáhne požadovaného maximálního momentu nastaveného v HMI, dojde k nepatrnému povolení lan, aby v nich nebyla příliš velká síla. Aktuální poloha se nastaví jako známé délky lan v klidovém stavu. Kalibrační sekvence lan slouží hlavně pro dosažení stejného dopnutí všech lan a před jejím provedením je nutné provést manuální nastavení nulových pozic.

Doplňkové programy

Pro usnadnění manipulace s kyvadlem byly z původního programu převzaty doplňkové programy. Doplňkové programy výrazně usnadňují práci při kalibraci, testování, nebo při řešení chyb, nejsou však pro funkčnost kyvadla nutné.

Programy typu StJog umožňují pomocí na základě definovaného inkrementu proměnné jogovat jednotlivé osy. Programy StMove přejíždí osami na zadanou pozici. Program StParking nejprve zvedne hrot a následně přejede skupinou os GroupXY do požadované pozice. To je využíváno například při ručním mazání, nebo aby kyvadlo nevadilo ve výhledu na obrazec.

Programy StMove

3.9 Příjem, čtení a zpracování dat

Návrhář obrazců, vytvořený Davidem Dostrašilem (viz obrázek 3.10), umožňuje kreslení libovolných, více-tahových obrazců, jejichž trajektorii optimalizuje pro vykreslení kyvadlem do písku. Následně do textového souboru s příponou .kyv ukládá data v podobě pořadí bodu v aktuálním tahu, času, souřadnice v ose x a souřadnice v ose y . Nový tah vždy začíná s pořadím 0 a stejným časem. Pokud je stejná i pozice, jedná se o ostrou hranu v obrazci. Pokud je pozice rozdílná, kyvadlo musí přejet do jiného bodu. Příklad dat s popisky je v tabulce 3.3. Zpracovaná data mohou být uživatelem odeslána na FTP server.



Obrázek 3.10: Obrazec Šťastný lenochod nakreslený v návrháři

Popis dat	Pořadí v tahu, Čas, Souřadnice X, Souřadnice Y
Začátek 1. tahu	0,0,-84.933129,19.638688
	1,0.77,-84.448058,21.578973
	2,1.152,-84.103657,22.956576
Konec 1. tahu	3,1.615243,-83.926017,23.667137
2. tah (tečka)	0,1.615243,-75.365563,21.652912
Začátek 3. tahu	0,1.615243,-66.805109,17.792316
	1,2.313993,-66.009431,18.928999
Konec 3. tahu	2,2.800133,-65.630145,19.470836
4. tah (ostrá hrana)	0,2.800133,-65.630145,19.470836
	1,3.520587,-64.957578,18.237797

Tabulka 3.3: Ilustrační tabulka výstupních dat z návrháře

Vytvoření FTP serveru pomocí PLC s SD kartou

Použité PLC umožňuje využití SD karty jako úložiště FTP serveru a může k němu následně v rámci sítě být přistupováno. V nastavení zabudovaných Ethernet/IP portů v Sysmac Studiu je nutné nastavit použití FTP serveru a jeho parametry: číslo portu FTP pro přenos příkazů (výchozí 21), přihlašovací jméno a heslo. Při použití těchto parametrů a IP adresy portu PLC je možné přistupovat k FTP serveru ze zařízení pomocí Ethernet sítě.

Zpracování dat

Zpracování dat probíhá v programu StDataRead, který je strukturovaný jako stavový automat. K zjištění názvu datového souboru na SD kartě je využito knihovny pro FTP server, která obsahuje funkční blok FTPGetFileList. Nejprve se nastaví parametry FTP serveru a následně se pomocí FTPGetFileList zjistí název nejnovějšího souboru ve složce pro ukládání dat.

Po zjištění názvu souboru se soubor otevře použitím FileOpen a FileSeek nastaví ukazatel na začátek souboru. Funkčním blokem FileGets jsou procházeny řádky a pokud jejich počet přesáhne velikost předdefinovaného pole, je na HMI zobrazena chyba velikosti dat.

Když je velikost dat v mezích, program využívá opět FileSeek pro nastavení ukazatele na začátek souboru a FileGets pro načtení řádku. Výstup funkčního bloku FileGets v podobě stringu načteného řádku je využit ve funkci SubDelimiter, která rozdělí string pomocí čárek a výsledek uloží do jednotlivých členů pole dataArray. Pole dataArray má definovanou datovou strukturu ST_Data, která obsahuje čtyři členy: Poradí, Cas, PoziceX, PoziceY.

Po úspěšném zpracování dat je soubor zavřen blokem FileClose. Protože datové soubory ukládá návrhový software do PC, není nutné je zachovávat na SD kartě a jsou smazány na konci programu blokem FileRemove. Pokud je řízení provedeno pomocí lineární interpolace dat, program zde končí a přechází se do programu StRefr. Ve verzi, kde jsou data interpolována kubickým splinem, je nejprve zavolán

Task0 s programem DataToPoly.

Interpolace dat kubickým splinem

Pro dosažení co nejvyšší kvality vykreslování byla vytvořena verze programu pracující s interpolací dat pomocí kubického Hermitova splinu C^2 se spojitou druhou derivací, se kterou pracuje i návrhový software obrazců. Výpočty probíhají v separátním tasku v programu DataToPoly, protože jsou časově náročné (přibližně 13 ms) a překračují periodu cyklu hlavního tasku.

Pro interpolaci se vychází z obecného tvaru pro dílčí kubické polynomy:

$$P_i(x) = C3_i x^3 + C2_i x^2 + C1_i x + C0_i$$

Pro zjištění jednotlivých koeficientů je využito znalosti dvou bodů a jejich derivace. Pro výsledný tvar rovnic koeficientů se vychází z Hermitova tvaru pro kubický polynom.

Rovnice koeficientů vycházející z Hermitova tvaru:

$$\begin{aligned} C0_i &= y_i \\ C1_i &= d_i \\ C2_i &= \frac{3(y_{i+1} - y_i) - h_i(d_{i+1} + 2d_i)}{h_i^2} \\ C3_i &= \frac{2(y_i - y_{i+1}) + h_i(d_{i+1} + d_i)}{h_i^3} \end{aligned} \quad (3.3)$$

kde

$$h_i = x_{i+1} - x_i$$

Z požadavku na spojitou druhou derivaci vznikne po úpravách soustava lineárních rovnic v maticovém tvaru:

$$Ad = r$$

Matice A je tridiagonální:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ h_2 & 2(h_1 + h_2) & h_1 & 0 & 0 & 0 \\ 0 & h_3 & 2(h_2 + h_3) & h_2 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & h_{n-1} & 2(h_{n-2} + h_{n-1}) & h_{n-2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Matice neznámých derivací d:

$$d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}$$

Matice r :

$$r = \begin{bmatrix} 0 \\ h_2\delta_1 + h_1\delta_2 \\ h_3\delta_2 + h_2\delta_3 \\ \vdots \\ h_{n-1}\delta_{n-2} + h_{n-2}\delta_{n-1} \\ 0 \end{bmatrix}$$

kde

$$\delta = \frac{y_{i+1} - y_i}{h_i}$$

Protože kyvadlo musí být vždy na začátku i na konci trajektorie v klidovém stavu, jsou matice upraveny tak, aby derivace d v prvním a posledním bodě trajektorie byly nulové.

Výpočty pro interpolaci byly převzaty z knihy o numerických výpočtech v Matlabu [8].

Pro výpočet koeficientů je nutné nejprve vyřešit matice a zjistit derivace d . Protože matice A je tridiagonální, je možné pro výpočet použít Thomasův algoritmus [9]. K výpočtu jsou vytvořeny uživatelské funkce FillMatrix, MatrixSolve a HermiteSorted, které jsou seskupeny ve funkci SolveFunctions. FillMatrix slouží k naplnění matice A a r , ze kterých jsou ve funkci MatrixSolve zjištěny derivace za pomoci Thomasova algoritmu. Derivace jsou následně využity pro výpočet koeficientů jednotlivých polynomů (dle 3.3) ve funkci HermiteSorted. Protože výsledná trajektorie je složena ze splinu v osách x a y , je nutné výpočet provést pro obě osy. Výstupem je pole Coef datové struktury ST_Hermite s časem koncového bodu a koeficienty kubických polynomů. Dále se do proměnné SizePoly uloží počet prvků v poli Coef a do pole Strokes s datovou strukturou ST_Strokes pozice v poli Coef a druh přechodů do jiných tahu.

3.10 Algoritmus pro vykreslení obrázků

Protože řízení s lineární interpolací dat probíhá podobně jako s polynomiální, bude zde popsána pouze verze s polynomiální interpolací. Výsledky obou verzí budou porovnány v kapitole 4.

Nájezd do referenčního bodu

Předtím než je spuštěna interpolace trajektorie na základě času, musí být kyvadlo v prvním bodě obrazce. Přejezd je řízen v programu StRefr. Nejprve se zvedne hrot do klidové pozice ve výšce 3 mm nad pískovou plochou. Následně jsou svázány osy X a Y . Skupině os se nastaví pozice pro lineárně interpolovaný pohyb odpovídající koeficientům $C0$ prvního polynomu v obou osách. Po dosažení tohoto bodu je kreslicí hrot spuštěn do odpovídající kreslicí pozice. Stejného principu je použito při vykreslování pro přejezd do jiného tahu.

Vykreslování obrazců

Vykreslování obrazců ze zpracovaných dat je řízeno v programu StCam. StCam je rozdělen na dvě části: interpolační část, ve které dochází k výpočtu trajektorie v každém cyklu, a stavový automat.

Stavový automat

V prvních dvou stavech se nejprve virtuálnímu masteru VM nastaví nulová pozice a osy X a Y se sváží do skupiny.

Dále je při prvním průchodu následujícím stavem dán ose VM příkaz k pohybu rychlostí 1 mm/s do pozice, která odpovídá času na konci současného tahu a povolí se interpolace. konce jednotlivých tahů jsou uloženy v poli Strokes. Pomocí příslušných příkazů je aktivováno synchronní vykonávání pohybu os pro všechna lana a skupiny os GroupXY. V tomto stavu program setrvává dokud osa VM nedojede do zadané pozice, kde se pohyb kyvadla zastaví. Po dokončení pohybu osy VM je na základě informace z pole Strokes zvoleno, zda se jedná o ostrou hranu, nebo přejezd do jiného bodu. Pokud se jedná o ostrou hranu, dochází k přechodu do stavu, který slouží pouze k vynulování indikátoru o prvním průchodu. Pokud je následující tah v jiném bodě, dojde k vypnutí všech pohybových příkazů a využívá se stejného principu jako při nájezdu do referenčního bodu. V obou případech se program vrací do původního stavu a celý proces se opakuje, dokud není dokončen poslední tah. Vykreslování je možné z HMI dočasně zastavit a následně znovu pustit nebo zcela přerušit. Po dokončení vykreslování je rozvázána skupina os GroupXY a přechází se do programu StStop.

Interpolace

Interpolační část programu se vykonává po celou dobu pohybu, mimo přejezdy do jiných tahů. Na základě aktuální pozice osy VM je vybírán polynom s odpovídajícím časem. Jeho pozice v poli je uložena do proměnné actPos, na základě které se ve výpočtu volí koeficienty polynomu. Následně se vypočítá vzdálenost virtuálního masteru od času začátku polynomu actVM, která slouží pro výpočet aktuální hodnoty polynomu vzhledem k času. Vypočtené hodnoty odpovídají souřadnicím pro skupinu os X a Y, které jsou v programu MotionControl přepočteny na délku lan.

Výpočet hodnoty polynomu na základě času vypadá následovně:

$$y = C0 + actVM \cdot (C1 + actVM \cdot (C2 + actVM \cdot C3))$$

Hrotu je předávána pozice z vektorového výpočtu, ke které jsou přičteny konstanty nastavené na HMI, případně může být kreslení hrotu z HMI vypnuto.

3.11 Ochrana proti poruchám a chybová hlášení

Tabulkový maximální moment převodovky je stanoven na 106 Nm. To po přepočtu na použitou navíjecí cívku odpovídá síle v laně 1178 N. Tabulková hodnota maximální síly pro lano je 530 N. Z toho vyplývá že nejslabším článkem soustavy je lano.

Pro ochranu lan proti přetržení byla pro model kyvadla navrhnutá ochranná funkce, která hlídá sílu v lanech. V případě překročení předem stanovené síly je kyvadlo nouzově zastaveno. Maximální přípustná síla byla stanovena na hodnotu 400 N. Dříve však vždy dojde k přetržení lanové pojistky s binderem u které se nedá přesně určit síla přetržení.

Síla v lanech je vypočtena pro všechna tři lana pomocí aktuálního momentu servopohonu v procentech $Act.Trq$, jmenovitého momentu servopohonu $RatedTorque$, poloměru navíjecí cívky r a převodového poměru i dle následujícího vztahu:

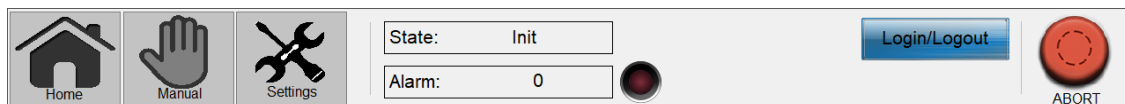
$$F = \frac{Act.Trq \cdot RatedTorque \cdot i}{100 \cdot r}$$

Protože při synchronním řízení os mohou osy na základě pozice dosáhnout až maximálního omezení rychlosti, byla převzata z původního programu diferenciální kontrola. Funkce diferenciální kontroly počítá z aktuální pozice os pro tažná lana zpětně pozici kyvadla v osách x a y . Pokud je rozdíl zpětně vypočtené pozice kyvadla a aktuální pozice skupiny os $GroupXY$ větší než nastavená hodnota, dojde k nouzovému zastavení.

Obě kontroly jsou provedeny v programu `AlarmControl`, kde je zároveň řešeno zpracovávání a vypisování chybových hlášení do HMI pomocí datových typů `Union` (princip byl již popsán na příkladě chyb funkčních bloků v 3.6).

3.12 Vizualizace pro obslužné HMI

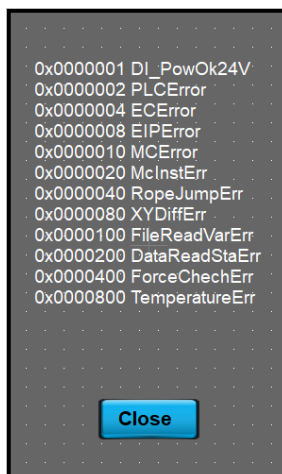
Pro obsluhu kyvadla byla vytvořena vizualizace v programu `EasyBuilder Pro` od firmy `Weintek`. Vizualizace je tvořena třemi přepínatelnými obrazovkami se sdílenou vrchní lištou.



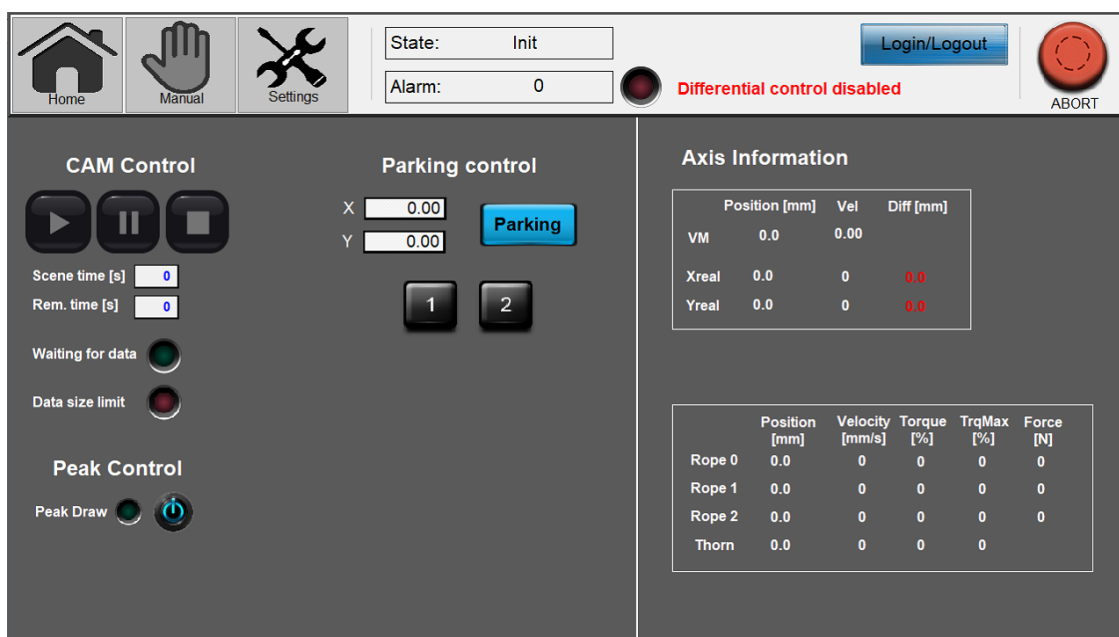
Obrázek 3.11: Vrchní lišta vizualizace

Tlačítka `Home`, `Manual` a `Settings` slouží k výběru obrazovky. Textové okénko `State` slouží k výpisu aktuálního stavu kyvadla v závislosti na vykonávaném stavovém programu. Do textového okénka `Alarm` jsou vypisovány chybové kódy (popisy kódů jsou dostupné na obrazovce `Settings` viz 3.12). Indikátor vedle textového

okénka signalizuje, zda je zapnutá diferenční kontrola. Protože přístup k obrazovkám Manual a Setting je omezen přihlašovacími údaji, je v rohu umístěno tlačítko Login/Logout pro přihlášení. Tlačítko ABORT slouží k nouzovému přerušení programu.



Obrázek 3.12: Chybové kódy

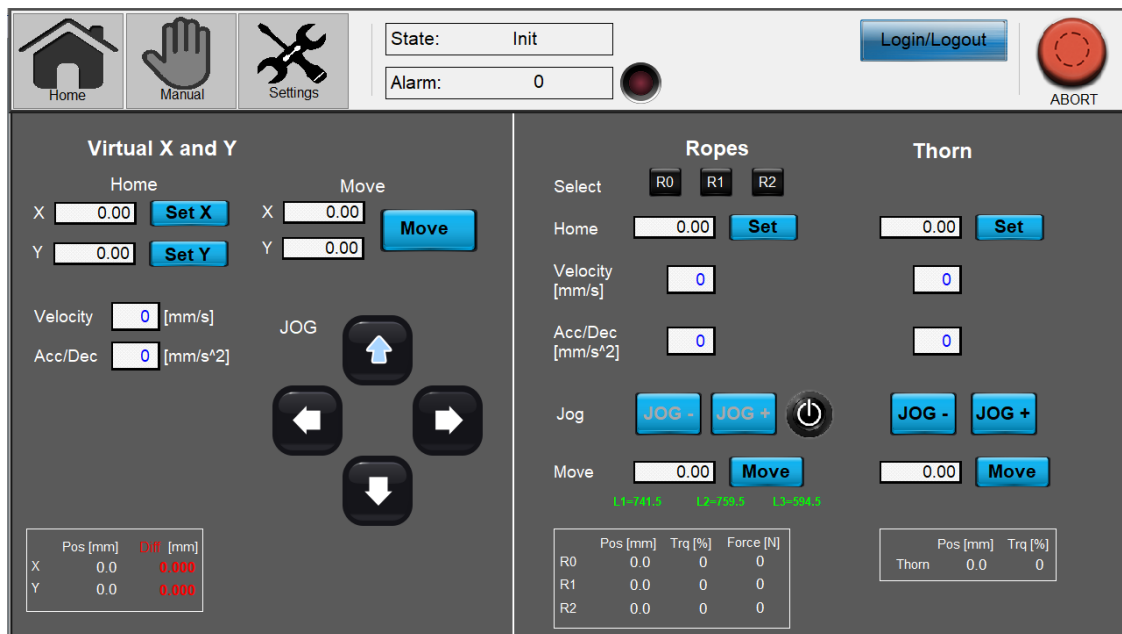


Obrázek 3.13: Obrazovka Home pro ovládání vykreslování

V oblasti CAM Control jsou umístěna tlačítka pro ovládání vykreslování obrazců. Tlačítko Play započne sekvenci načtení dat a následného vykreslení. Tlačítko Pause dočasně zastaví vykreslování. Tlačítko Stop vykreslování přeruší. V okénku Scene time je vypsán celkový čas vykreslování obrazce (nepočítají se přejezdy mezi tahy). Okénko Remaining time zobrazuje zbývající čas do konce vykreslování. Indikátor

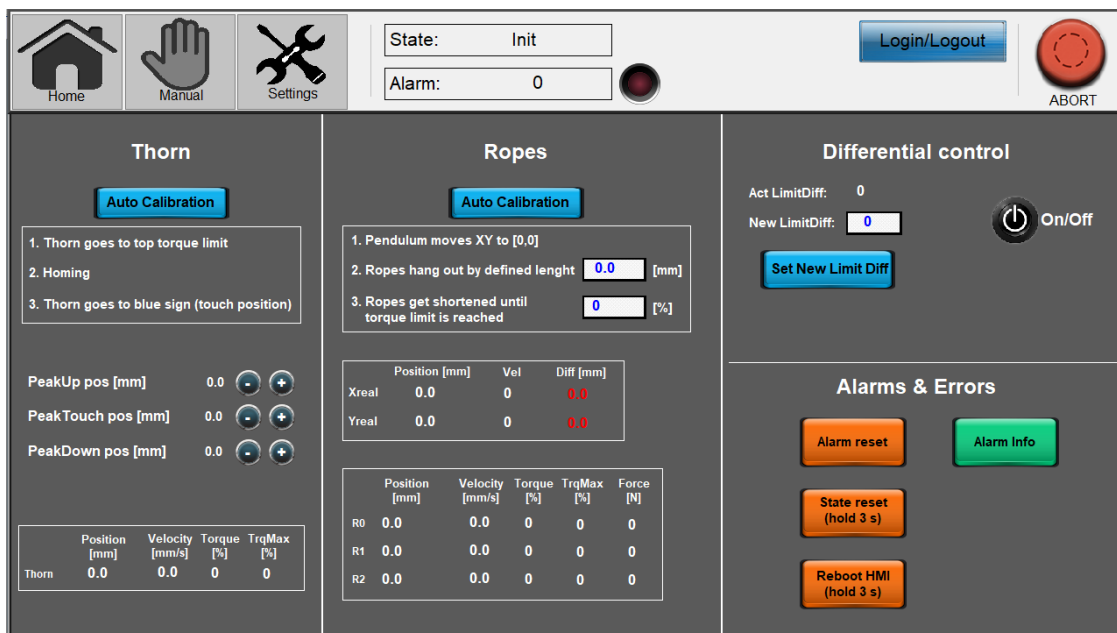
Waiting for data začne blikat pokud je spuštěna sekvence pro vykreslení, ale na SD kartě chybí datový soubor. Indikátor Data size limit začne blikat pokud počet bodů v datech přesáhl definovanou velikost polí a zhasne po opětovném zapnutí sekvence.

V sekci Peak Control je možné vypnout, nebo zapnout vykreslování hrotem. V oblasti Parking Control je možné nastavit pozici kyvadla pro parkování. Tlačítka 1 a 2 spustí parkování v předem definovaných pozicích, které slouží k usnadnění ručního mazání. V oblasti Axis Information jsou vypsané všechny dostupné informace o osách.



Obrázek 3.14: Obrazovka Manual pro ruční ovládání pohybových os

Na obrazovce Manual je možné nastavovat Home pro jednotlivé osy pomocí okénka s hodnotou a tlačítka Set. Pro ovládání os Ropes je nejdříve nutné vybrat příslušnou osu pomocí tlačítek R0, R1 a R2. Nastavení hodnoty pro Move a následné potvrzení tlačítkem Move přesune pohybovou osu do požadované pozice. Parametry Velocity a Acc/Dec slouží k nastavení rychlosti a zrychlení os. Pro virtuální osy X a Y je jogování provedeno pomocí směrových šipek. Modrá šipka směřující vzhůru odpovídá kladné souřadné ose y na pískové ploše. U os Ropes a Thorn jsou lana zkracována tlačítkem JOG- a prodlužována tlačítkem JOG+. Pro osy Ropes je jogování defaultně zamčeno vypínacím tlačítkem kvůli prevenci mylné manipulace.



Obrázek 3.15: Obrazovka Settings

V sekci Thorn je umístěno tlačítko Auto Calibration, které spustí automatické kalibrování kreslicího hrotu. Do proměnné PeakUp pos je uložena hodnota, ve které je kreslicí hrot v klidovém stavu. PeakTouch pos udává pozici hrotu těsně nad písčovou plochou. PeakDown pos je pozice, ve které hrot kreslí a může jí být ovlivněna šířka tahu. Všechny hodnoty mohou být upravovány pomocí inkrementů 0,5 mm a může tak být odladěna funkce hrotu.

V sekci Ropes je tlačítko Auto Calibration pro kalibraci lan. Zde je možné ovlivnit parametry momentu a prověšení při kalibraci pro její odladění.

V oblasti Differential control je možné spínat funkci diferenciální kontroly a nastavit limit difference.

V oblasti Alarms & Errors jsou umístěna tlačítka pro práci s chybami. Tlačítko Alarm reset vypne všechna chybová hlášení, což umožní dále operovat kyvadlem, pokud byla chyba odstraněna. Tlačítko State reset po třech sekundách držení přejde do programu StStop. Je používáno v případě chyby v programu PLC pokud dojde k zaseknutí v nějakém stavu. Tlačítko Reboot HMI restartuje obslužné HMI v případě chyby. Alarm Info vypíše všechny chybové kódy a k nim příslušící chyby (viz obrázek 3.12).

4 Testování a vyhodnocení výsledků

Kyvadlo bylo v prvotní fázi vývoje testováno na jednoduchých, jedno-tahových obrazcích, jejichž body byly interpolovány lineárně. Příkladem je kružnice vytvořená v programu Excel, nebo obrazec šneka vytvořený v návrhovém softwaru.



(a) Obrazec kružnice

(b) Obrazec šneka

Obrázek 4.1: První jedno-tahové testovací obrazce

Řídicí algoritmus byl postupně rozšiřován, nejprve o přejezdy do jiných tahů a následně i o zastavování v koncových bodech tahů pro zachování ostrých hran.



(a) Obrazec šneka s více tahy

(b) Obrazec vrány

Obrázek 4.2: Více-tahové obrazce

V poslední fázi vývoje byl algoritmus řízení, včetně zpracování dat, předělán pro práci s interpolací bodů kubickým splinem. Interpolace splinem se na kvalitě vykresleného obrazce vizuálně neprojevila. To může být způsobeno jednak hrubostí písku, ale také tím, že lineární interpolace pracuje s dostatečně velkým množstvím málo vzdálených bodů. Rozdíl by se výrazněji projevil, pokud by byl počet bodů snížen v rámci zrychlení zpracování dat. Na příkladu obrazce Šťastný lenochod (viz obrázek 4.3), není v písku vidět žádný rozdíl.

Algoritmus pracující s interpolací splinem se však díky předchozím zkušenostem povedlo zjednodušit a výrazně zlepšit dobu vykreslování obrazců, které obsahují mnoho tahů. Obrazec Šťastný lenochod má pouze 8 tahů a trvá při vykreslování řídicím algoritmem pro lineární interpolaci 122 s, zatímco nový algoritmus pro kubický spline jej vykreslí za 116 s.

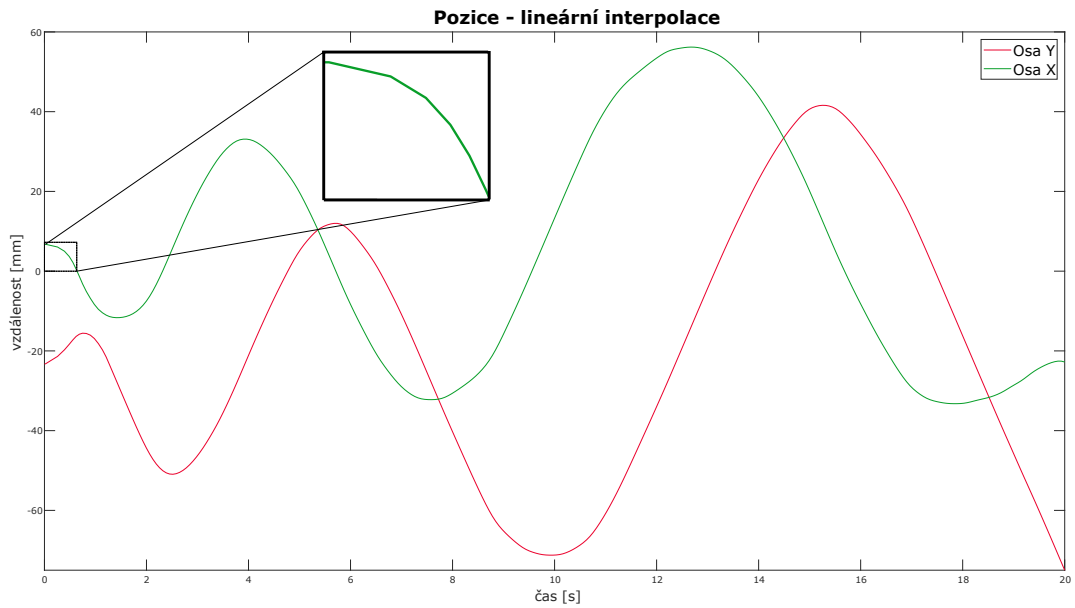


(a) Lineární interpolace

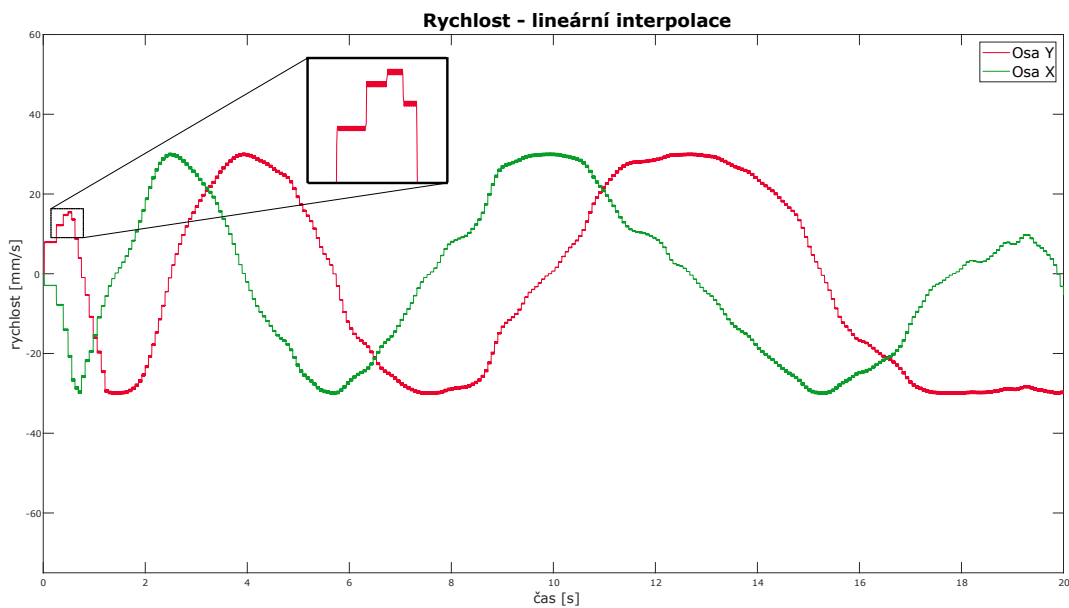
(b) Polynomiální interpolace

Obrázek 4.3: Srovnání interpolací na obrazci Šťastný lenochod

Pro analýzu průběhů vykreslování bylo využito funkce Data Trace v Sysmac Studiu a následného exportu a vykreslení grafů pomocí Matlabu. Trasována byla rychlost a pozice virtuálních pohybových os X a Y při vykreslování obrazce šneka. Pro trasování byla zvolena vzorkovací perioda 2 ms a zpracováno bylo prvních 20 s vykreslování.

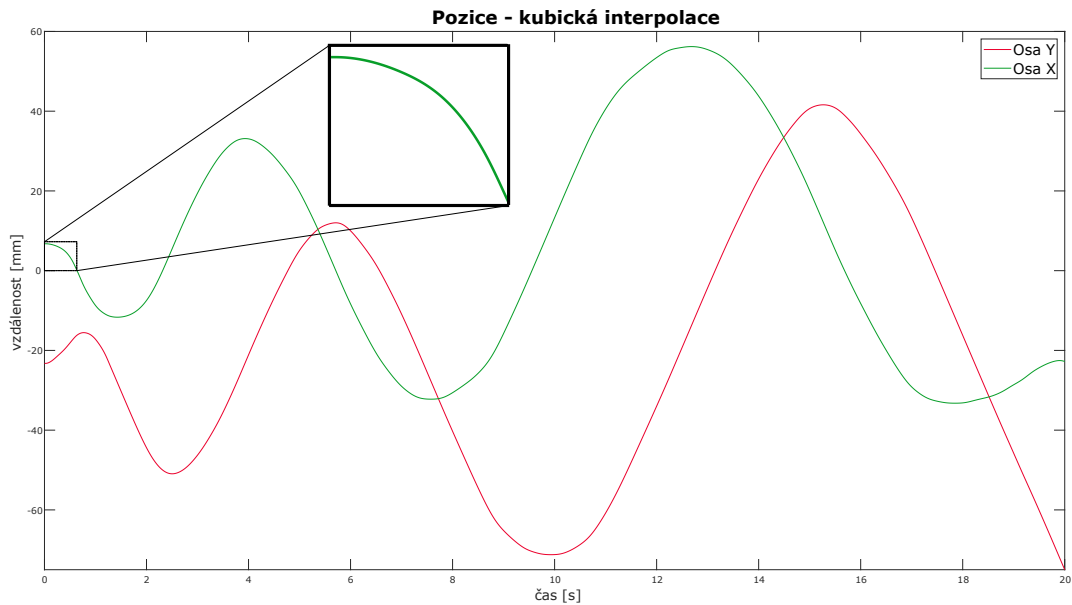


Obrázek 4.4: Graf pozic virtuálních os X a Y pro lineární interpolaci

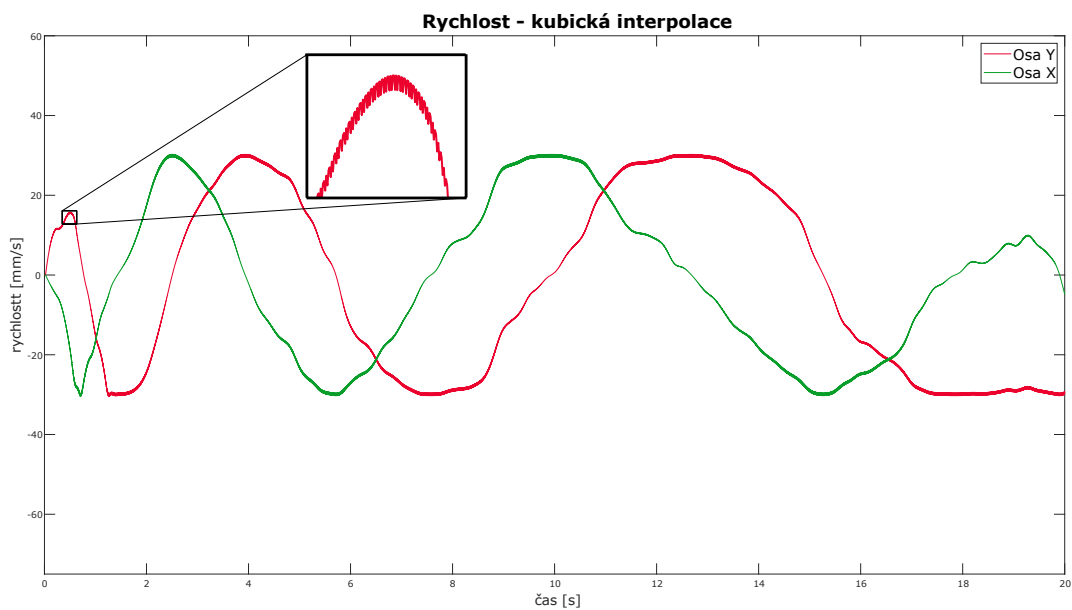


Obrázek 4.5: Graf rychlosti virtuálních os X a Y pro lineární interpolaci

Na grafu 4.4 je vidět, že přechody mezi body nejsou hladké a dochází v nich k mírnému zalomení. Z grafu 4.5 je patrné, že se rychlost v bodech mění skokově. To je způsobeno nespojitostí první a druhé derivace při lineární interpolaci.

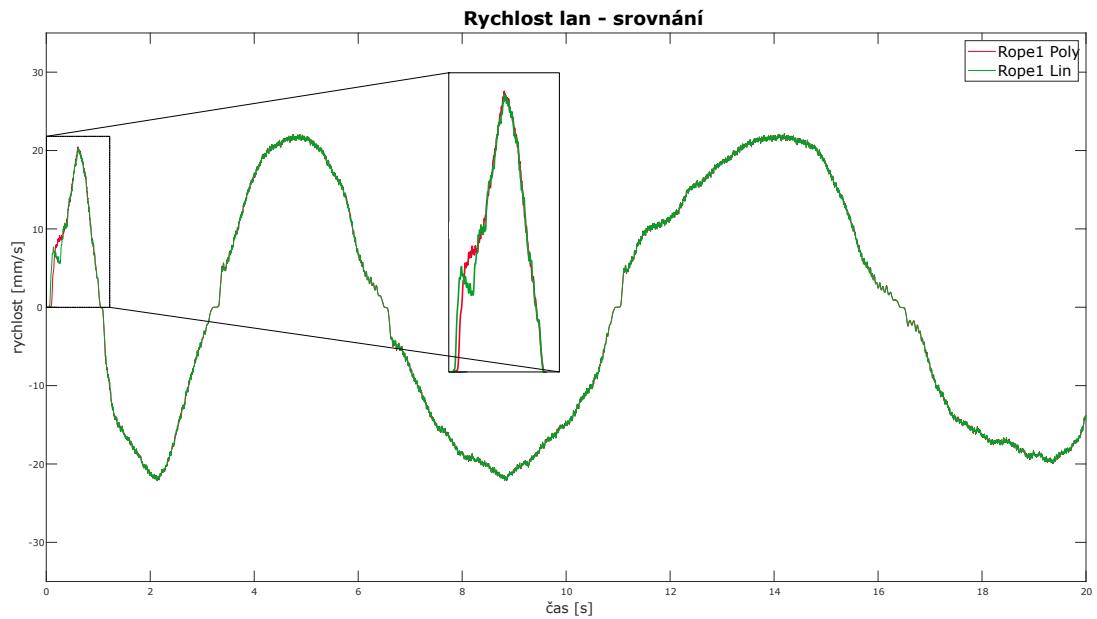


Obrázek 4.6: Graf pozic virtuálních os X a Y pro kubickou interpolaci



Obrázek 4.7: Graf rychlosti virtuálních os X a Y pro kubickou interpolaci

Z grafu 4.6 je vidět, že interpolace kubickým splinem zajistí plynulý přechod mezi body. Protože byla při výpočtu zajištěna spojitost druhé derivace, nedochází ke skokovým změnám v rychlosti (viz graf 4.7). Rychlost pouze kmitá okolo žádané rychlosti, která je dána časem v datech. Nejpatrnější rozdíl v kvalitě průběhu je vždy na počátku obrazce, tedy při rozjezdu kyvadla.



Obrázek 4.8: Graf rychlosti lana Rope1, porovnání kubické a lineární interpolace

Při vykreslení průběhu rychlosti lan (viz graf 4.8) je rozdíl patrný hlavně při rozjezdech vlivem větší časové vzdálenosti mezi body. U průběhu lana pro lineární interpolaci dochází vlivem nespojitosti derivací ke skoku v rychlosti. Z toho se dá usoudit, že polynomiální interpolace bude mít vhodnější průběh pro dynamicky náročné obrazce a bude mít plynulejší průběh i při menším počtu řídicích bodů. Rozdíl je však poměrně malý a pro řízení kyvadla díky velkému počtu řídicích bodů je postačující lineární interpolace.

Závěr

Bakalářská práce se věnovala návrhu, realizaci a optimalizaci řídicího algoritmu pro kinetickou plastiku Pendulum.

Řídicí algoritmus plastiky byl úspěšně vytvořen a testován na modelu kyvadla 1:10. Podařilo se zajistit komunikaci mezi notebookem s aplikací pro návrh obrazců a PLC. Řídicí algoritmus dokáže vykreslovat více-tahové obrazce a je schopen rozeznávat a zachovávat ostré hrany v obrazci. Pro interpolaci trajektorie byly vytvořeny dvě verze programu pracující s lineární interpolací, nebo s interpolací kubickým splinem. Obě verze byly testovány a porovnány na základě průběhu vykreslování. Z obslužného HMI s vytvořenou vizualizací je možné ovládat pohyb, průběh vykreslování a nastavovat jednotlivé parametry kyvadla. Boční tažná lana jsou proti přetržení chráněna funkcí, která hlídá sílu v lanech na základě momentu motoru.

Problémem algoritmu je časově náročné načítání dat z textového souboru. Práce s textovým souborem je uživatelsky jednodušší, ale čtení dat po jednotlivých řádcích způsobuje nárůst časové náročnosti v závislosti na počtu řídicích bodů v datech. Řešením by mohlo být použití jiného formátu pro přenos dat.

Řídicí algoritmus je poměrně limitován hardwarovými komponenty použitými na modelu kyvadla. Vlivem šnekových převodovek není možné přesně snímat moment přenášený na lana, což omezuje možnosti bezpečnostních a kalibračních funkcí. Konstrukce kreslicího trnu a jeho servopohon neumožňují optimální nastavení regulace, což se může projevat zhoršenou plynulostí pohybu.

Na kinetickou plastiku je možné umístit velkou škálu uměleckých i funkčních komponentů a jejich variací, což rozšiřuje oblast jejího použití. Zároveň se nabízí možnost využití plastiky pro určitou formu jednoduchého a interaktivního 3D tisku. Kinetická plastika by pak mohla plnit svůj umělecký záměr a zároveň by sloužila pro tvorbu hmatatelných výrobků.

Použitá literatura

- [1] BUŠEK, M., ANTOŠ, J., VÁCLAVÍK, M. *Anwendung der Mechatronik zur Umsetzung der kinetischen Skulptur Pendulum*. Fachtagung MechFaachttragunig 2019, Paderborn, 2019. Dostupné z: doi:10.17619/UNIPB/1-769
- [2] Pendulum. *Lasvit* [online]. 2019 [cit. 2022-02-27]. Dostupné z: <https://www.lasvit.com>
- [3] OMRON Industrial Automation. *OMRON* [online]. [cit. 2022-02-27]. Dostupné z: <https://industrial.omron.eu/>
- [4] NJ/NX-series Instructions Reference Manual. *Sysmac Studio* [online]. [cit. 2022-03-06]. Dostupné z: https://assets.omron.eu/downloads/manual/en/v4/w502_nx_nj-series_instructions_reference_manual_en.pdf
- [5] Wago - Průmyslové switche. *Wago* [online]. [cit. 2022-03-13]. Dostupné z: <https://www.wago.com/cz/průmyslový-switch/p/852-101#details>
- [6] Weintek. *Weintek* [online]. [cit. 2022-03-13]. Dostupné z: www.weintek.com
- [7] SOUČEK, Pavel. *Servomechanismy ve výrobních strojích*. Praha: Vydavatelství ČVUT, 2004. ISBN 80-01-02902-6.
- [8] MOLER, Cleve B. *Numerical computing with MATLAB*. Philadelphia: SIAM, 2004. ISBN 0-89871-560-1.
- [9] Tridiagonal matrix algorithm. *Wikipedia: the free encyclopedia* [online]. [cit. 2022-04-17]. Dostupné z: https://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm