ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

BAKALÁŘSKÁ PRÁCE

Řízení rychlých servopohonů

Vypracoval: Lubomír Prudek

Vedoucí práce: Ing. Tomáš Fencl

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Lubomír Prudek**

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný
Obor: Kybernetika a měření

Název tématu: **Řízení rychlých servopohonů**

Pokyny pro vypracování:

1. Převeďte současný řídicí program pro model "žongler" do standardního jazyku s omezenou variabilitou dle IEC61131-3 a IEC61511-1 za využití standardních funkcí pro řízení servopohonů PLCopen. Nový řídicí program musí zachovat stávající funkce a umožnit zaměňování předpisů kinematiky bez nutnosti zásahů do programu.
2. Rozšiřte řídicí program o ovládání pátého, lineárního, pohonu a nadhazovacího mechanismu.
3. Navrhněte a realizujte detekci spadnutí přehazované koule pomocí analýzy průběhu krouticího momentu měřeného v reálném čase na jednotlivých servopohonech.
4. Připravte úlohu pro výuku řízení polohových aplikací.

Seznam odborné literatury:

Dodá vedoucí práce

Vedoucí: Ing. Tomáš Fencl

Platnost zadání: do konce zimního semestru 2010/2011

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

L.S.

doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 1. 4. 2010

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
Katedra řídicí techniky
K13135

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze, dne 21.5.2010

Lubomír Prudek
.....
Podpis

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
Katedra řídicí techniky
K13135

Abstrakt

Tato bakalářská práce se zabývá konverzí programu do standardního jazyka s využitím funkcí pro řízení servopohonů PLCopen a dále rozšiřuje program o část pro řízení pátého lineárního motoru pro nahazování koulí spolu s detekci nahozených kouli na přehazovací ramínka. Bakalářská práce je navíc doplněna o postup při vytvoření výukového materiálu pro seznámení studentů s polohovými zařízeními.

Abstract

This bachellor thesis deals with conversion of PLC program into standard language with usage of functions for motion control from PLCopen library. It also contains program for fifth linear motor for picking up balls, program detecting presence of balls in juggling arms and in the end of this document are described instructions to create educational materials for students working with positional servoing.

Poděkování

Na tomto místě bych rád poděkovat vedoucímu bakalářské práce Ing. Tomášovi Fenclovi za cenné rady a připomínky při psaní bakalářské práce.

Dále bych chtěl také poděkovat Ing. Pavlu Mezerovi za jeho rady při psaní programu, také svým týmovým kolegům Pavlu Jarošovi a Tomáši Kohoutovi.

Nakonec své rodině a přátelům za inspiraci a duševní podporu.

Obsah:

1. ÚVOD 1

2. ŘÍZENÍ SERVOPOHONŮ 3

2.1. ŘÍZENÍ STROJE 4
2.2. KOMUNIKACE 5
2.3. POPIS VYKONÁVÁNÍ PROGRAMU 6
2.3.1. ALGORITMUS VAČKOVÉHO PROFILU 7
2.3.2. JAZYK „LADDER DIAGRAM“ 8
2.4. VIZUALIZACE 12
2.5. KINEMATIKA KOULE 14
2.5.1. DEFINICE ZÁKLADNÍCH FYZIKÁLNÍCH KONSTANT 15
2.5.2. VRH SVISLÝ VZHŮRU 15
2.5.3. VRH ŠIKMÝ VZHŮRU 17
2.5.4. ROZMĚROVÁ IDENTIFIKACE MODELU ŽONGLÉRA 18
2.5.5. VYPOČTENÉ A NAVRŽENÉ HODNOTY PRO ZADANÉ FUNKCE 20
2.5.6. NÁVRH VAČKOVÝCH PROFILŮ 28

3. NAHAZOVACÍ MECHANISMUS 33

3.1. LINEÁRNÍ MOTOR 33
3.2. REALIZACE ZÁSOBNÍKU 35
3.3. VYPOČTENÉ A NAVRŽENÉ HODNOTY PRO ZADANÉ FUNKCE 38
3.3.1. FUNKCE NAHOĎ KOULI 38
3.3.2. FUNKCE ZAHOĎ KOULI 40

4. DETEKCE PŘÍTOMNOSTI KOULE 41

4.1. DETEKCE KOULE POMOCÍ KAMERY 41
4.1.1. KOMUNIKACE S KAMEROU 43
4.2. DETEKCE KOULE POMOCÍ KROUTICÍHO MOMENTU 45
4.2.1. MĚŘENÍ A ZPRACOVÁNÍ KROUTICÍHO MOMENTU 45

5. LABORATORNÍ ÚLOHA 50

5.1. ZADÁNÍ: 50
5.2. POSTUP PŘI VYTVOŘENÍ NOVÉHO PROJEKTU 50
5.2.1. VYTVOŘENÍ NOVÉHO PROJEKTU 51
5.2.2. HARDWAROVÁ KONFIGURACE 52
5.2.3. VYTVOŘENÍ PROGRAMU 56
5.2.4. NASTAVENÍ KOMUNIKACE 62

6. ZÁVĚR **65**

SEZNAM LITERATURY **67**

SEZNAM OBRÁZKŮ **68**

SEZNAM TABULEK **71**

SEZNAM PŘÍLOH **71**

1. ÚVOD

Výukový model žongléra (obrázek 1) vznikl na katedře řídící techniky jako diplomová práce Jana Pšeničky pod vedením Doc. Ing. Jiřího Bayera, CSc. Model žongléra se skládá z pěti synchronních motorů (čtyři třífázové rotační a jeden jednofázový lineární), z nichž dva největší vykonávají prostřednictvím lineárních modulů vertikální pohyb, dva menší motory horizontální kruhový pohyb, na kterých jsou umístěny přehazovací ramínka, a poslední lineární motor slouží pro nahazování koulí. Model žongléra bude sloužit jako vzdálený model, ke kterému se studenti mohou přihlásit a seznámit se s jednotlivými jeho částmi běžně používanými v průmyslu a navrhovat algoritmy pro jeho řízení.

Obrázek 1 – Model žongléra

Hlavním cílem této bakalářské práce je přepracování současného programu psaného v jazyce C do standardního jazyka s využitím funkcí pro řízení servopohonů PLCopen. Jako další část je rozšíření modelu o pátý lineární motor, který bude sloužit pro nahazování koulí na přehazovací ramínka, a nebude tedy nutná fyzická přítomnost člověka, který by obsluhoval stroj. Program bude nadále rozšířen o zpětnou vazbu detekující koule na přehazovacích ramínkách. Další funkce, kterými bude rozšířen model žongléra, jsou nahoď *kouli*, *zahoď kouli* a žonglování se *třemi koulemi*. Kinematickým popisem tří letících koulí se zabývá ve své bakalářské práci Dynamika rychlých servopohonů Pavel Jaroš. Poslední části této bakalářské práce je návrh vzorové úlohy, kterou budou studenti využívat při výuce.

Současně se na této bakalářské práci ještě podílel student Tomáš Kohout v rámci své bakalářské práce Vizuální kontrola chodu, ve které se zabývá detekci koulí pomocí rychloběžné kamery.

Tato bakalářská práce vznikla jako týmový projekt již jmenovaných studentů v rámci mezinárodní soutěže European Industrial Ethernet Award pořádanou společností B&R Automation v roce 2009.

2. ŘÍZENÍ SERVOPOHONŮ

Celý program byl naprogramován pomocí jazyku „Ladder Diagram“ (kapitola 2.3.2). Tento jazyk byl vybrán pro svoji názornost a snadné pochopení. Program je rozdělen do několika části, které obsahují inicializaci a definovaní základních pohybů všech pěti motoru, řídící část pro definování vačkových profilů pro jednotlivé motory, dále část, která se stará o komunikaci a zpracování zpráv z kamery obsahující informace o detekovaných koulí - detekce koulí je ještě prováděna měřením velikosti krouticího momentu, část, která slouží pro aktivaci alarmu, informativních a chybových hlášek pro vizualizaci na PLC a mnoho dalších.

Nejdůležitější část programu tvoří ovládání hlavní logiky, aby bylo možné vybrat správný typ vačky pro požadovaný pohyb, najetí do výchozích pozic pro detekci a přehazování koulí, zajištění ochrany programu proti nežádoucím stavům (zmáčknutí více tlačítek najednou a podobně). Z hlediska bezpečnosti bylo do programu implementováno hlídání aktuální teploty. Dojde-li k překročení kritické teploty, bude program automaticky zastaven.

Pro obsluhu stroje máme na výběr ze dvou základních režimů:

1. *Manuální řízeni*

   V tomto režimu je možné si každý servomotor posunout do požadované polohy a určovat, jakými rychlostmi se bude pohybovat. Dále je možné na jednotlivá přehazovací ramínka nahodit nebo zahodit kouli.

2. Automatické řízení

   Tento režim umožňuje téměř nulovou obsluhu celého stroje, poté co jsou připojeny jednotlivé servozesilovače a servomotory k elektrické síti, stačí vybrat zvolenou funkci a stisknout tlačítko Start. Po stisknutí tlačítka dochází k inicializaci všech servomotorů, pokud již tak ještě nebylo učiněno, a začne se vykonávat zvolená funkce.

Na výběr jsou tyto funkce:

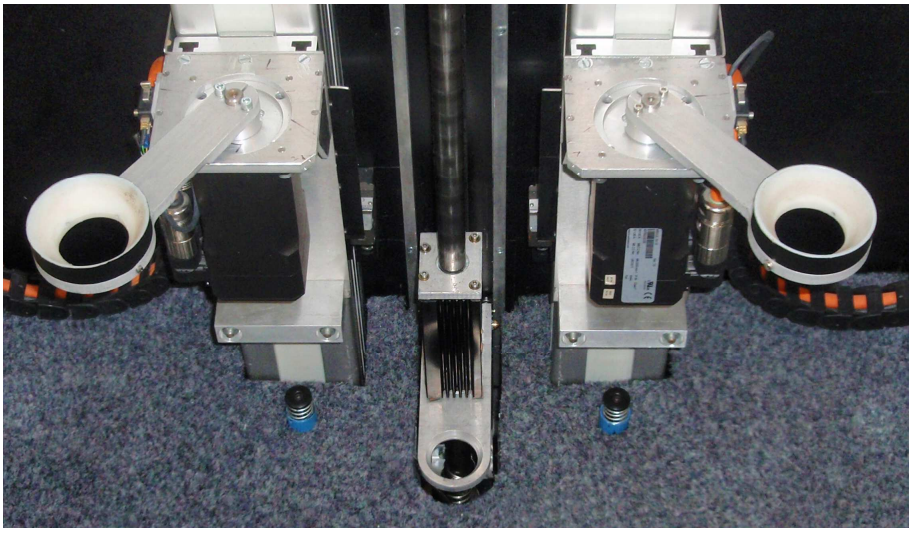
a) Katapult
b) Žonglování s 1 koulí
c) Žonglování s 2 koulemi
d) Žonglování s 3 koulemi

Podrobnější popis obsluhy stoje je obsažen v příloze č.1.

2.1. ŘÍZENÍ STROJE

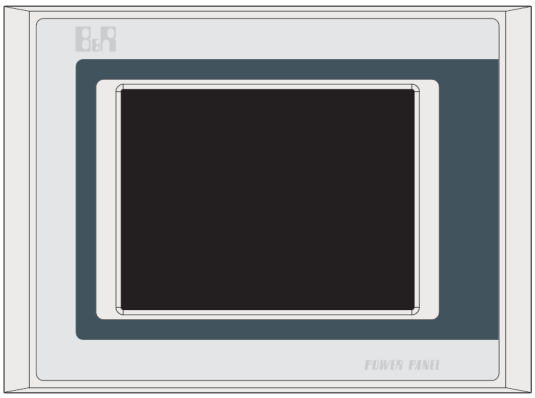
Polohování stroje je realizováno pomoci synchronních motorů (obrázek 2.1 a 2.2). Ovládání každého motoru zajišťuje servozesilovač (Acopos, obrázek 2.3), kterému jsou posílány řídící signály průmyslovým automatem (Power Panel 4PP420, obrázek 2.4) s využitím komunikace Ethernet Powerlink. Průmyslový automat ovládá servozesilovač pomocí definovaného vačkového profilu, který je volán na základě stavu vačkového automatů.

Obrázek 2.1 – Vertikální synchronní motor



Obrázek 2.2 – Horizontální synchronní motory + lineární motor

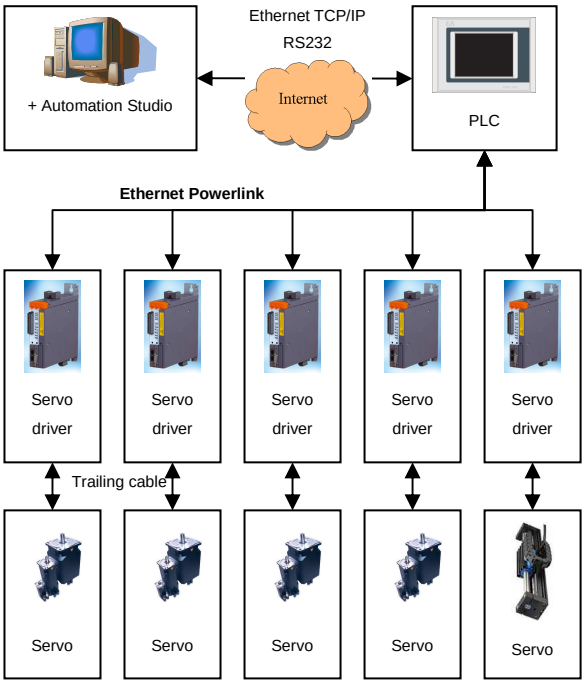
Obrázek 2.3 – Acopos



Obrázek 2.4 – Power panel 4PP420

2.2. KOMUNIKACE

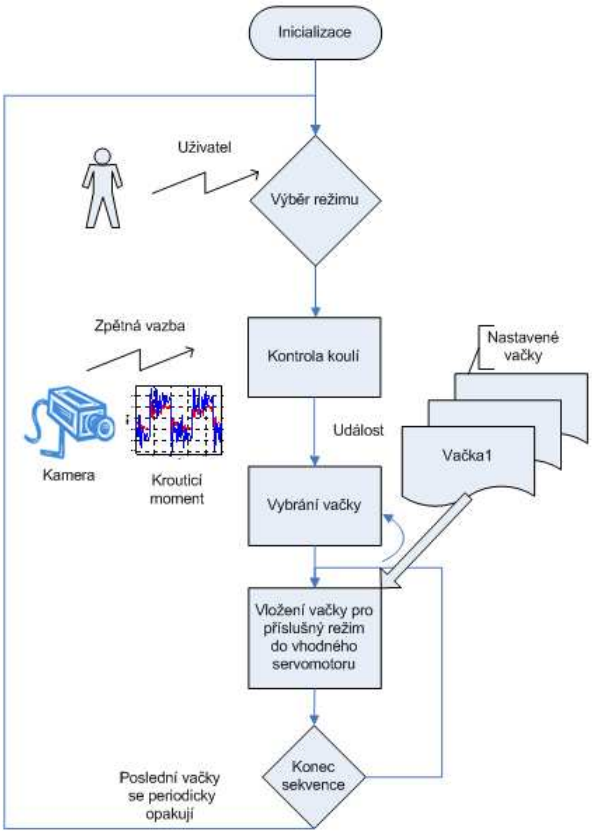
Komunikace mezi počítačem a PLC je realizována pomoci komunikací RS232 a Ethernet TCP/IP (obrázek 2.5). Pomocí těchto komunikací můžeme provést upload/download programu z/do PLC, dále se může pomocí Automation Studia monitorovat, vyhodnocovat a ladit příslušný program.



Obrázek 2.5 – Struktura řízení a komunikace modelu

2.3. POPIS VYKONÁVÁNÍ PROGRAMU

Poté co dojde k inicializaci stroje, následuje stav, kdy PLC čeká na událost od uživatele. Jakmile dojde ke spuštění této události, zasynchronizují se požadované vačky a dojde k postupné synchronizaci jednotlivých vaček za sebe. Výběr vačky závisí také na zpětné vazbě od kamery (kapitola 4.1) nebo na velikosti krouticího momentu (kapitola 4.2), je-li koule umístěna na přehazovacím ramínku či nikoliv. V okamžiku, kdy se začne vykonávat poslední vačka, už nedochází k synchronizaci další vačky, ale tato poslední vačka se bude vykonávat do doby, kdy znovu nedojde ke spuštění události od uživatele nebo od zpětné vazby. Obsluha a vykonávání této události je zobrazena na obrázku 2.6.



Obrázek 2.6 – Vykonávání programu

Podrobnější skládání vačkových profilů pro funkci *tří koulí* je obsaženo v příloze č.2.

2.3.1. ALGORITMUS VAČKOVÉHO PROFILU

Řízení vačkového profilu je realizováno dvěma algoritmy, které se začnou vykonávat po spuštění některé funkce pro žonglování. První algoritmus provede nahrání zvoleného vačkového profilu do příslušných servozesilovačů a druhý zajistí spuštění pěti reálných a jedné virtuální osy (kapitola 2.5.6).

Pro nahrání vačkových profilů platí:

1. Vyhodnocení zadávaných profilů z řídícího panelu PLC (power panel 4PP420)
2. Nahrání vačkových profilů z PLC do servozesilovače

Pro řízení os pomocí vačkového profilu platí:

1. Stisknutí tlačítka start pro požadovanou funkci
2. Najetí motorů do výchozí polohy pro žonglování
3. Nastavení počátečních parametrů pro pět reálných a jednu virtuální osu
4. Spuštění pohybu reálných a virtuální osy (vačkový automat)
5. Čekání na příkaz z tlačítka Stop pro zastavení spuštěných os
6. Začíná se znovu od kroku jedna

2.3.2. JAZYK „LADDER DIAGRAM“

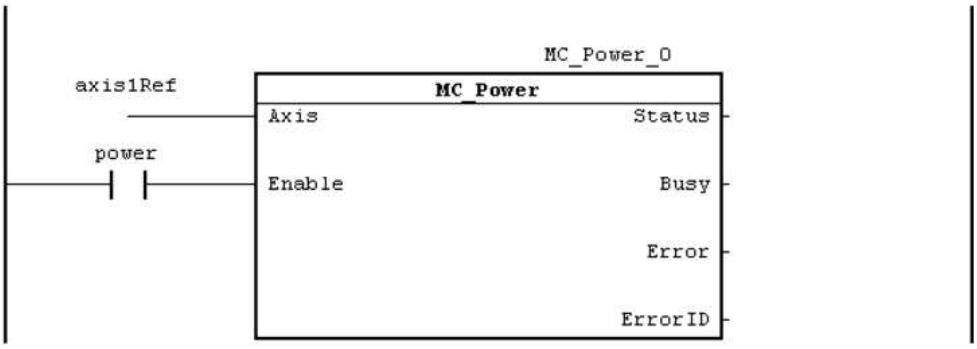
Ladder Diagram (*jazyk kontaktních schémat*) je grafický jazyk, který je určen k tvorbě úloh převážně logického typu. Tento jazyk je sestavován prakticky stejně jako liniové schéma reléových systému. Jednotlivé logické proměnné jsou tvořeny sérioparalelními kombinacemi kontaktů spolu se vstupními a výstupními funkcemi reléových cívek, čítačů a časovačů, paměťovými obvody a mnoha dalšími funkčními bloky. Ve vývojovém programu připomínají jednotlivé dílčí obvody příčky žebříku.

Původní program vytvořený Janem Pšeničkou byl psaný v jazyce C. Příklad tohoto kódu je na obrázku 2.7. Pro další rozšíření tohoto programu uživatelem případně obsluhou, která není srozuměna s jazykem C, byl tento jazyk předělán do jazyku Ladder Diagram (obrázek 2.8). Při porovnání obou typů jazyků je vytvoření a deklarace funkčního bloku v Ladder Diagramu jednodušší a kratší, a proto se hlavně užívá pro aplikace, kde je potřeba měnit program obsluhou zařízení, pro kterou je tento typ zápisu velmi jednoduchý.

```
#include <bur/plctypes.h>
#include <acp10man.h>
#include <acp10_mc.h>

#define STATE_READY  00
#define STATE_POWER  01
#define STATE_ERROR  02

_GLOBAL ACP10AXIS_typ  axis1;

_LOCAL UDINT  axis1Ref;
_LOCAL BOOL   power;
_LOCAL USINT  state;
_LOCAL UINT   axis1ErrorID;

_LOCAL MC_Power_typ  MC_Power_0;
```

```
_CYCLIC void basic_cyclic(void)
{
  switch (state)
  {
  ...
  case STATE_READY:
    if (power)
    {
      power=0;
      state=STATE_POWER;
    }
  break;
/*********************************************/
  case STATE_POWER:  /*power drive*/
    MC_Power_0.Enable=1;
    if (MC_Power_0.Status)
    {
      state=<NEXT_STATE>;
    }
    else if (MC_Power_0.Error)
    {  /*Store ErrorID in a variable,
         ErrorID is reset on falling edge of 'Enable'*/
      axis1ErrorID=MC_Power_0.ErrorID;
      MC_Power_0.Enable=0;
      state=STATE_ERROR;
    }  /* end Power */
  break;
  ...
  }  /*end switch*/
  MC_Power_0.Axis=axis1Ref;
  MC_Power(&MC_Power_0);
}  /* end cyclic */
```

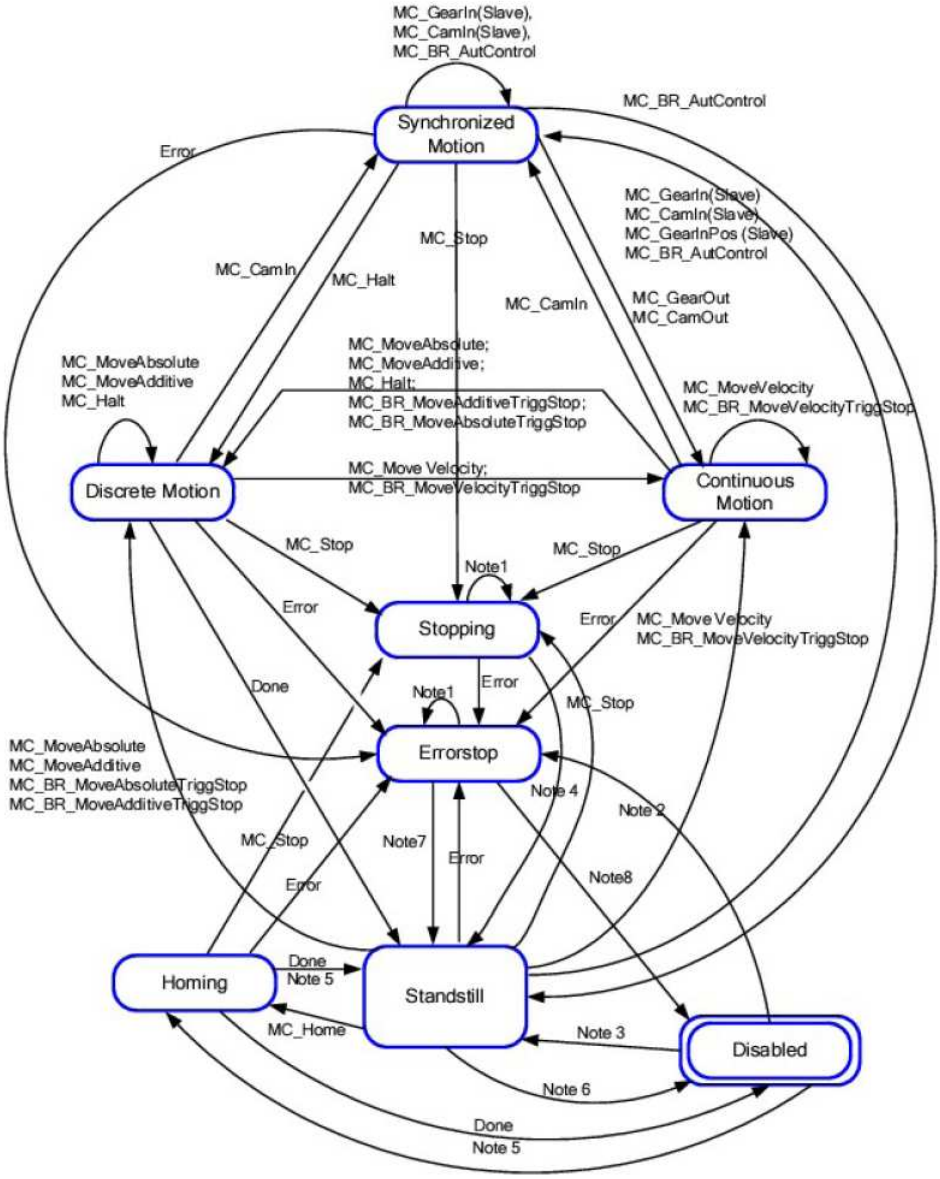
Obrázek 2.7 – Funkční blok „MC_Power“ v jazyce C



Obrázek 2.8 – Funkční blok „MC_Power“ v jazyce Ladder Diagram

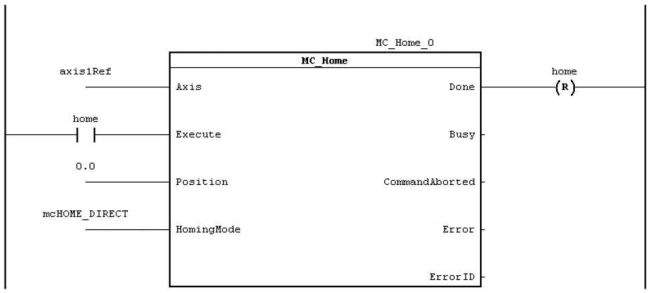
K tomu, aby mohl být celý program předělán jazykem Ladder Diagram, musel být použit mezinárodní standard PLCopen Motion Control Library. Celá specifikace standardu PLCopen Motion Control Library tvoří vlastní knihovnu funkčních bloků pro ovládání a řízení jedné a více os. Tuto knihovnu funkcí PLCopen (acp10_mc) je zapotřebí vložit do seznamu knihoven.

Při tvorbě celého funkčního programu se muselo postupovat podle stavového diagramu (obrázek 2.9), který definuje hierarchii a jednotlivé závislosti všech funkčních bloku.

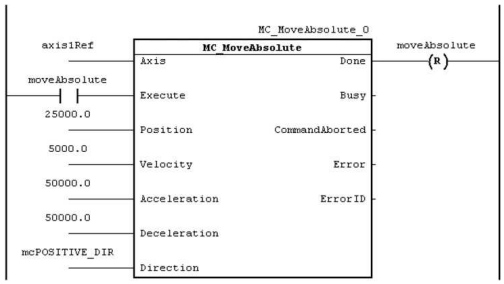


Obrázek 2.9 – Stavový diagram

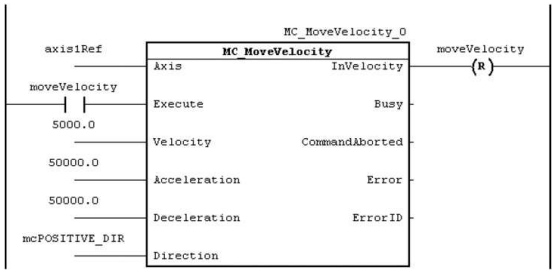
Nejdůležitější funkční bloky použité pro řízení modelu žongléra jsou uvedeny na následujících obrázcích. Například funkční blok „MC_Power“ (obrázek 2.8) slouží k zapnutí motorů na silové zařízení, „MC_Home“ (obrázek 2.10) k nalezení referenční pozice, od které se budou odvíjet všechny typu pohybu, „MC_MoveAbsolute“ (obrázek 2.11) pro najetí motoru do výchozí pozice, „MC_MoveVelocity“ (obrázek 2.12) pro spuštění virtuální osy, „MC_CamTableSelect“ (obrázek 2.14) pro načtení vačkového profilu, „MC_CamIn“ (obrázek 2.15) pro spuštění vačkového profilu a „MC_Stop“ (obrázek 2.13) pro zastavení motoru. Pro chod programu jsou tyto funkční bloky nejdůležitější, ale jedná se pouze o malou část funkční bloků, které byly při tvorbě programu použity.



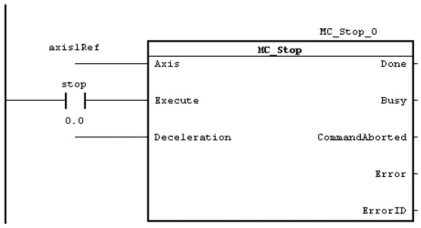
Obrázek 2.10 – Funkční blok „MC_Home“



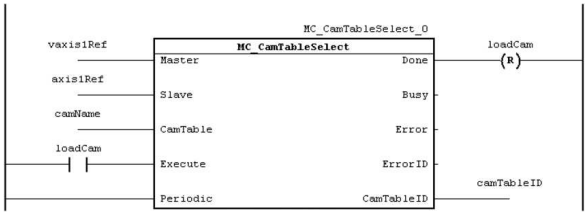
Obrázek 2.11 – Funkční blok „MC_MoveAbsolute“



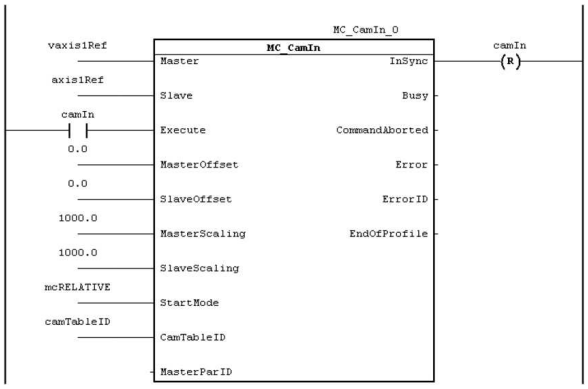
Obrázek 2.12 – Funkční blok „MC_MoveVelocity“



Obrázek 2.13 – Funkční blok „MC_Stop“



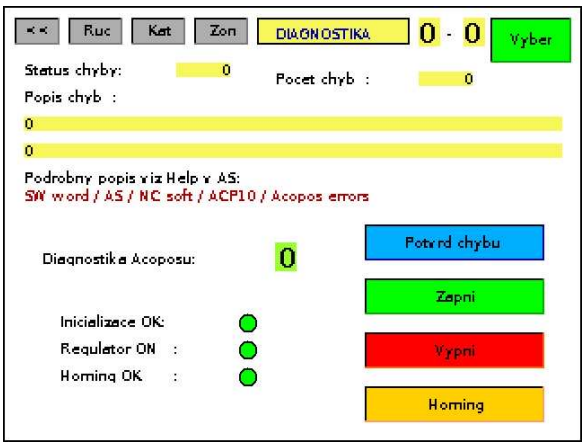
Obrázek 2.14 – Funkční blok „MC_CamTableSelect“



Obrázek 2.15 – Funkční blok „MC_CamIn“

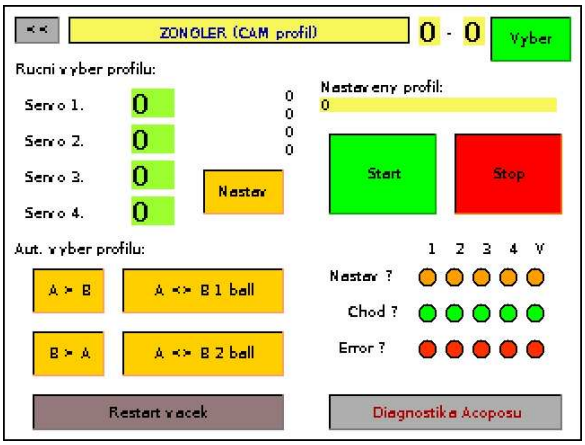
2.4. VIZUALIZACE

Jako další část bylo vytvoření nové vizualizace jako náhrada za stávající, která již byla nevhodná a pro uživatele příliš složitá, za vizualizaci novou. Příklad staré a nové vizualizace je na obrázcích 2.16, 2.17, 2.18 a 2.19. Výhodou nové vizualizace je, že pro spuštění jakékoliv funkce stačí pouze jedno tlačítko, kdy dojde k zapnutí motoru, zahoumování a posléze spuštění vybrané funkce. U staré vizualizace se každý z těchto kroku musel provést zvlášť. Tento způsob obsluhy byl pro uživatele nepohodlný a zdlouhavý.



Obrázek 2.16 – Diagnostika - stará

Obrázek 2.17 – Diagnostika - nová



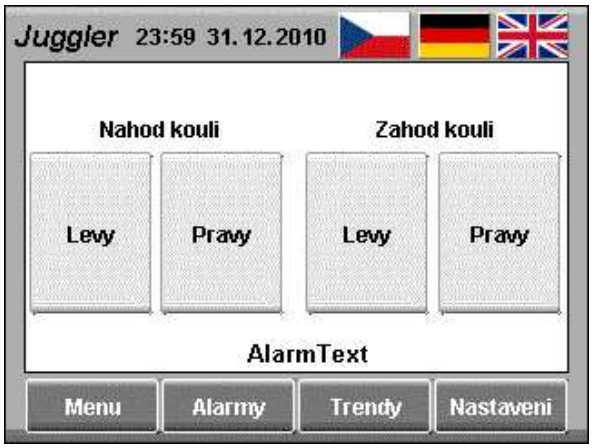
Obrázek 2.18 – Funkce *1-2 koulí* - stará

Obrázek 2.19 – Funkce *1-2 koulí* - nová

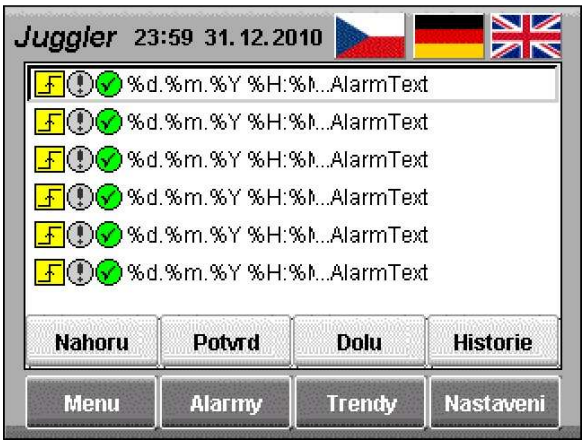
Nová vizualizace byla nadále rozšířena o funkci *3 koulí* (obrázek 2.20), *nahoď kouli*, *zahoď kouli* (obrázek 2.21), možnost výpisu informačních, varovných a chybových hlášek prostřednictvím implementované funkce „alarmu“ (obrázek 2.22), vykreslení teplotních charakteristik motorů (tzv. trendy - obrázek 2.23) a mnoho jiných aplikací.

Funkce alarmy informují uživatelé o současném stavu programu. Současně probíhá komunikace se Safety PLC, které spíná silovou část, servozesilovače (Acoposy), zámek pro ochranné plexisklo a jiné elektronické zařízení. V případě nějaké poruchy je pomoci funkce „alarmu“ informován uživatel o této závadě.

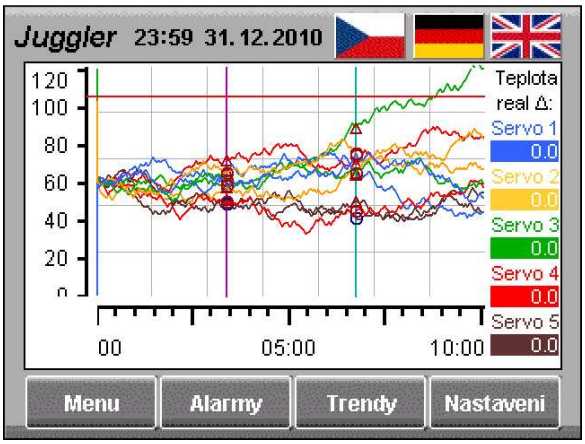
Obrázek 2.20 – Funkce *3 koulí*



Obrázek 2.21 – Funkce *nahoď* a *zahoď kouli*



Obrázek 2.22 – Alarmy



Obrázek 2.23 – Trendy

Kompletní vizualizace je obsažena v příloze č.3, kde jsou zobrazeny její jednotlivé části.

2.5. KINEMATIKA KOULE

K tomu, aby mohly být navrženy vačkové profily, bylo nutné nejprve nadefinovat základní rovnice z kinematiky pro vrh svislý a šikmý. Pohyb míčku při žonglování lze popsat rovnicemi šikmého vrhu. Rovnice popisující pohyb míčku odpovídají pohybu ideálního hmotného bodu, který není brzděn odporem vzduchu a na který nepůsobí ani jiné rušivé vlivy. Toto zjednodušení lze provést, protože žonglujeme kulečníkovými koulemi, jejichž setrvačnost je výrazně větší než vliv rušivých sil. Obecné vztahy popisu vrhu svislého vzhůru a vrhu šikmého vzhůru jsou popsány v kapitole 2.5.2 a 2.5.3. Vypočítané a částečně upravené výsledky těchto vztahů jsou popsány v kapitole 2.5.5. Tvorba a vyhotovení vačkových profilů jsou uvedeny v kapitole 2.5.6.

2.5.1. DEFINICE ZÁKLADNÍCH FYZIKÁLNÍCH KONSTANT

Dříve, než se začnou definovat základní rovnice pro vrh svislý a šikmý, je nutné uvést tabulku 2.1 základních fyzikálních konstant, které se objeví v těchto rovnicích.

Označení	Název	Velikost	Jednotka
y	Délka ve vertikálním směru	-	[m]
x	Délka v horizontálním směru	-	[m]
x_0	Počáteční délka v horizontálním směru	-	[m]
y_0	Počáteční délka ve vertikálním směru	-	[m]
v_y	Rychlost ve vertikálním směru	-	[m·s^{-1}]
v_x	Rychlost v horizontálním směru	-	[m·s^{-1}]
v_0	Počáteční rychlost	-	[m·s^{-1}]
v_{0x}	Počáteční rychlost v horizontálním směru	-	[m·s^{-1}]
v_{0y}	Počáteční rychlost ve vertikálním směru	-	[m·s^{-1}]
t	Čas	-	[s]
t_h	Čas nejvyššího výhozu	-	[s]
t_s	Čas dopadu	-	[s]
h	Výška nejvyššího výhozu koule	-	[m]
d	Délka vrhu - dostřel	-	[m]
g	Gravitační zrychlení země	9,81	[m·s^{-2}]
α	Elevační úhel	-	[°]

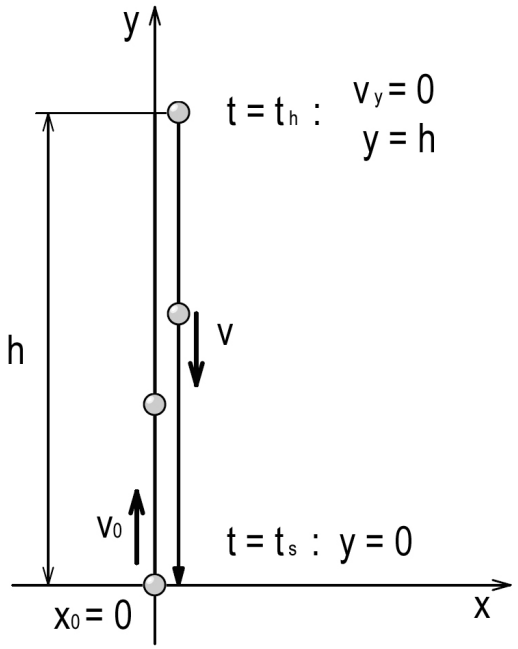
Tabulka 2.1 – Konstanty použité v rovnicích

2.5.2. VRH SVISLÝ VZHŮRU

Vrh svislý vzhůru je zobrazen na obrázku 2.24, kde lze velmi dobře pozorovat trajektorii koule. Koule nejprve zpomaluje při pohybu vzhůru, po dosažení maximální výšky a nulové rychlosti koule začne padat se stoupajícím zrychlením. Pro popsání tohoto jevu je důležité stanovit rovnice pro dráhu a rychlost koule.

Tento pohyb popisují následující rovnice (1):

$$y = v_0 \cdot t - \frac{1}{2} \cdot g \cdot t^2, \qquad v_y = v_0 - g \cdot t \tag{1}$$



Obrázek 2.24 – Kinematika vrhu svislého vzhůru

Z výše uvedené rovnice (1) je důležité odvodit dobu letu t_h do výšky h a dobu dopadu t_s.

Pro dobu výhozu t_h platí vztah (2):

$$t = t_h, \qquad v_y = v_0 - g \cdot t_h \quad \Rightarrow \quad t_h = \frac{v_0}{g} \tag{2}$$

Pro dobu dopadu t_s platí vztah (3):

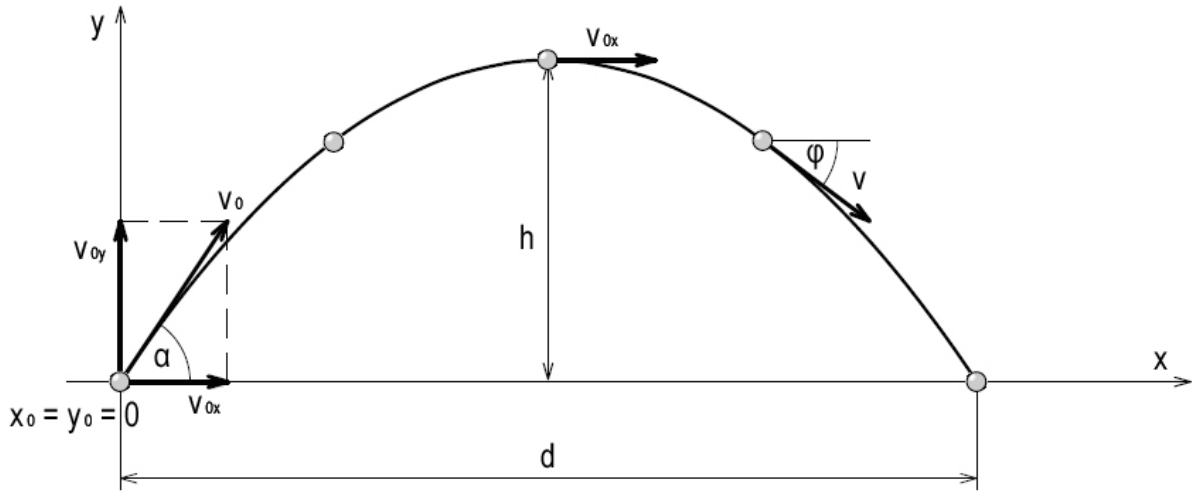
$$\begin{aligned} & t = t_s, \quad y = v_0 \cdot t_s - \frac{1}{2} \cdot g \cdot t_s^2 = \left(v_0 - \frac{1}{2} \cdot g \cdot t_s \right) t_s = 0, \\ & \Rightarrow t_s = \frac{2 \cdot v_0}{g} = 2 \cdot t_h \end{aligned} \tag{3}$$

Pro výšku výhozu h platí vztah (4):

$$y = h = v_0 \cdot t_h - \frac{1}{2} \cdot g \cdot t_h^2 = \frac{v_0^2}{g} - \frac{v_0^2}{2 \cdot g} = \frac{v_0^2}{2 \cdot g} \tag{4}$$

2.5.3. VRH ŠIKMÝ VZHŮRU

Vrh šikmý vzhůru lze nejlépe demonstrovat na obrázku 2.25. Z obrázku lze velmi dobře pozorovat, že se pohyb skládá ze dvou složek, složky horizontální-x a vertikální-y. Pohyb koule je tedy realizován ve dvou rovinách. Proto je potřeba odvodit základní rovnice pro rychlost a dráhu pro obě složky x a y.



Obrázek 2.25 – Kinematika vrhu šikmého vzhůru

Dráhu a rychlost, kterou opíše koule při svém pohybu, vyjadřují rovnice (5) a (6):

$$x = x_0 + v_0 \cdot t \cdot \cos\alpha$$
$$y = y_0 + v_0 \cdot t \cdot \sin\alpha - \frac{1}{2} \cdot g \cdot t^2 \qquad \textbf{(5)}$$

$$v_X = v_{0X} = v_0 \cdot \cos\alpha$$
$$v_Y = v_{0Y} - g \cdot t = v_0 \cdot \sin\alpha - g \cdot t \qquad \textbf{(6)}$$

Pro výpočet vektoru rychlosti v_0 můžeme použít Pythagorovu větu, pro kterou platí rovnice (7):

$$v_0 = \sqrt{{v_{0X}}^2 + {v_{0Y}}^2} \qquad \textbf{(7)}$$

Z těchto uvedených vztahů se může velmi jednoduše odvodit čas, za který bude koule ve své nejvyšší poloze a za jaký dopadne zpět do své výchozí polohy pro složku y posunutou o složku x. Pro tyto odvozené vztahy platí rovnice (8) po dosazení za $y_0 = 0$:

$$v_Y = v_0 \cdot \sin\alpha - g \cdot t_h \quad \Rightarrow \quad t_h = \frac{v_0 \cdot \sin\alpha}{g}$$
$$t_s = 2 \cdot t_h \tag{8}$$

Maximální výška výhozu, kterou koule při svém letu může dosáhnout, je dána rovnicí (9):

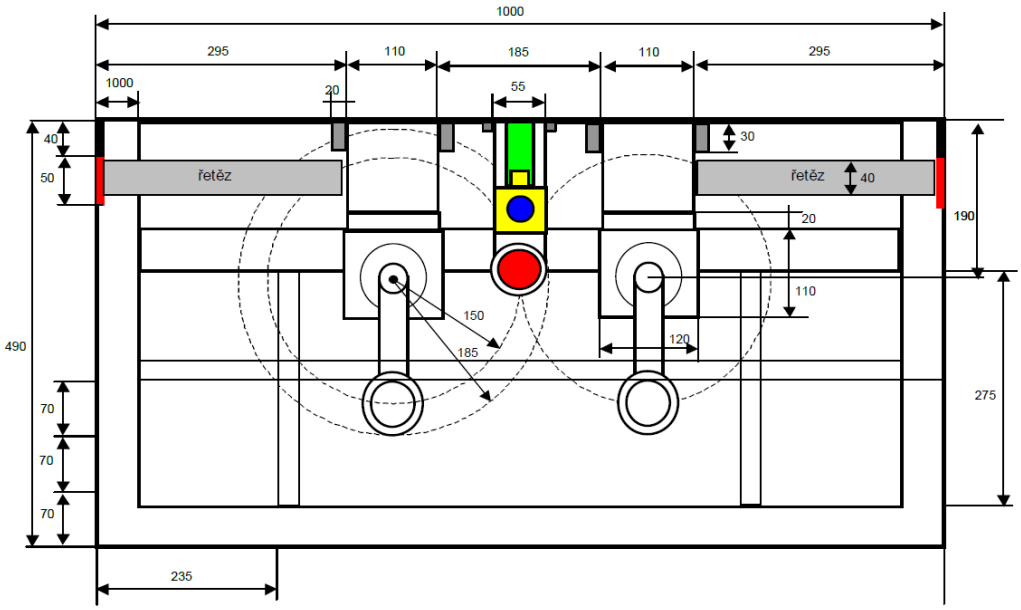
$$h = \frac{v_0^2 \cdot \sin^2\alpha}{2 \cdot g} \tag{9}$$

2.5.4. ROZMĚROVÁ IDENTIFIKACE MODELU ŽONGLÉRA

K tomu, abychom mohli vypočítat parametry pro zadané funkce (*katapult,* žonglování s *jednou koulí*, *dvěma koulemi* a posléze *nahazování koule* pomocí lineárního motoru, čímž se zabývá kapitola 3.3.1) z výše uvedených rovnic pro svislý vrh vzhůru a šikmý vrh vzhůru popsané v kapitole 2.5.2 a 2.5.3, je nutné provést rozměrovou identifikaci přehazovacích ramínek a prostředí, ve kterém dochází k přehozu koulí.

2.5.4.1. PŘEHAZOVACÍ RAMÍNKA

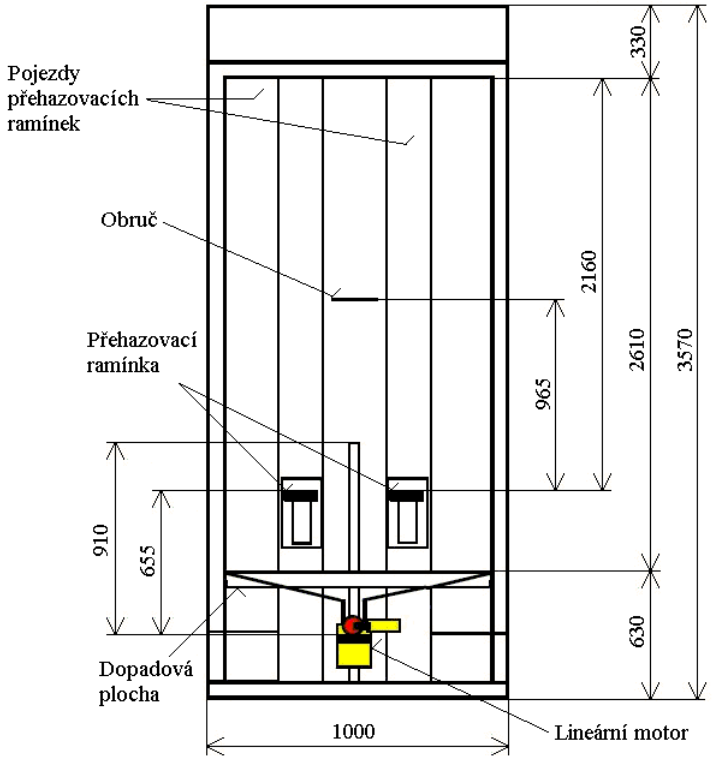
Na obrázku 2.26 lze vyčíst rozměrové hodnoty pro přehazovací ramínka. Délka ramen je 150 mm a jejich základny jsou od sebe vzdáleny **300 mm**. Tento údaj je velmi důležitý, protože v tomto rozmezí bude docházet k přehozu mezi jednotlivými ramínky.



Obrázek 2.26 – Rozměrová identifikace přehazovacích ramínek

2.5.4.2. ROZMĚRY MODELU ŽONGLÉRA

Konstrukční rozměry modelu žongléra lze vyčíst z obrázku 2.27, které jsou: výška je 3570 mm (výška, ve které se dá žonglovat je 2610 mm) a šířka je 1000 mm. Další důležitý rozměr je vzdálenost mezi přehazovacím ramínkem a nahazovacím lineárním motorem, který činí 655 mm. Tento rozměr je důležitý pro výpočet rychlosti a dráhy lineárního motoru při nahazování koulí umístěných v podavači na dopadové ploše žongléra. Jako poslední hodnota je vzdálenost mezi přehazovacím ramínkem a obručí, která slouží pro funkce katapult. Tato vzdálenost je 965 mm.



Obrázek 2.27 – Rozměry modelu žongléra

2.5.5. VYPOČTENÉ A NAVRŽENÉ HODNOTY PRO ZADANÉ FUNKCE

K tomu, aby mohly být vypočítány jednotlivé funkce žongléra, byl proveden kinematický popis vrhu svislého a šikmého vzhůru (kapitola 2.5.2 a 2.5.3), taktéž byla provedena rozměrová identifikace (kapitola 2.5.4). Nyní jsou dostupné všechny parametry a vzorce pro výpočet pohybu motorů. Těmito výpočty se zabývá kapitola 2.5.5.1, 2.5.5.2, 2.5.5.3, 3.3.1 a 3.3.2.

2.5.5.1. FUNKCE KATAPULT

Jak sám napovídá název, tato funkce má zajišťovat svislé vyhození koule, která posléze proletí pevně umístěnou obručí. Přehazovací ramínko, které vyhodilo kouli, se musí této obruči vyhnout a chytnou kouli nad obručí. Následně pustí kouli svisle dolů, aby došlo k opětovnému prohození obruče a znovu opětovnému zachycení koule pod obručí. Důležitý údaj, který je potřeba pro výpočet trajektorie

ramínka, je vzdálenost obruče od základní polohy přehazovací ramínka (obrázek 2.27). Obruč je vzdálena od přehazovacího ramínka 965 mm. Nyní je potřeba zvolit vzdálenost, ve které bude koule vyhazována z ramínka, aniž by došlo ke kolizi s obručí a byla zajištěna podmínka, že motor bude mít dostatečnou brzdnou dráhu (nesmí dojít k překročení povoleného zatěžovacího momentu). Jako startovací pozice vertikálních motorů byla stanovena výška 100 mm od inicializační polohy, proto se vyhazovací výška zkrátí na 865 mm. Výška, ve které koule bude opouštět vyhazovací ramínko, byla stanovena na 650 mm. Tato vzdálenost je dostatečná pro vyhození koule a zároveň k dobrzdění motoru, aniž by došlo ke kolizi s obručí či k překročení momentových hodnot motoru.

Pro rovnice z kinematiky pro vrh svislý vzhůru (kapitola 2.5.2) platí rovnice:

Pro výšku výhozu h platí vztahy (10) :

$$t = t_h, \qquad v_y = v_0 - g \cdot t_h \quad \Rightarrow \quad t_h = \frac{v_0}{g}$$
$$y = h = v_0 \cdot t_h - \frac{1}{2} \cdot g \cdot t_h^2 = \frac{{v_0}^2}{g} - \frac{{v_0}^2}{2 \cdot g} = \frac{{v_0}^2}{2 \cdot g} \tag{10}$$

Po dosazení za výšku h = 0,65 m se dostane počáteční vektor rychlosti v_0 (11):

$$0{,}65 = \frac{{v_0}^2}{2 \cdot g} \Rightarrow {v_0}^2 = 0{,}65 \cdot 2 \cdot g \tag{11}$$
$$v_0 = \sqrt{0{,}65 \cdot 2 \cdot g} = \sqrt{0{,}65 \cdot 2 \cdot 9{,}81} = \pm 3{,}571\ m \cdot s^{-1} \doteq 3{,}57\ m \cdot s^{-1}$$

Po vypočet času, za který bude mít koule nulovou rychlost platí (12):

$$t_h = \frac{v_0}{g} = \frac{3{,}57}{9{,}81} = 0{,}364\ s \tag{12}$$

V tento čas je potřeba nejpozději chytnout kouli, aby nedošlo k jejímu klesání směrem k zemi.

Ve chvíli, kdy bude koule chycena v horní části nad obručí, může být puštěna směrem k zemi, aby došlo k jejímu opětovnému prohození obručí a posléze k chycení v počáteční poloze. Vyhazovací ramínko musí mít větší rychlost a zrychlení, aby bylo možné kouli pustit směrem k zemi a zároveň ji chytit pod obručí.

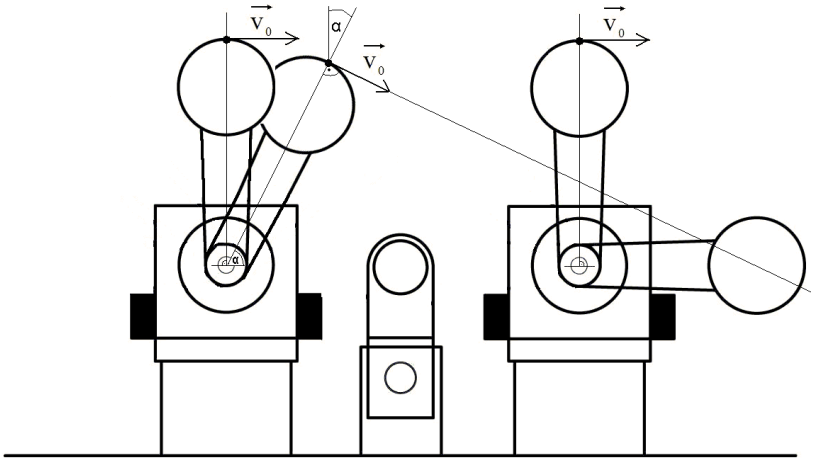
Čas, za který musí být koule chycena, udává rovnice (13):

$$t = t_s, \quad y = v_0 \cdot t_s - \frac{1}{2} \cdot g \cdot t_s^2 = \left(v_0 - \frac{1}{2} \cdot g \cdot t_s \right) t_s = 0\ m,$$
$$\Rightarrow t_s = \frac{2 \cdot v_0}{g} = 2 \cdot t_h = 2 \cdot 0{,}364 = 0{,}728\ s \tag{13}$$

Nyní jsou již vypočítané všechny parametry pro tvorbu vačkového profilu *katapult*. Návrh je obsažen v příloze č.4, kde jsou změněné rychlosti výhozu a čas chytání. Po vložení vypočítaných parametrů, funkce *katapult* fungovala s kulečníkovou koulí bez problémů, ale pokud byl použit tenisový míček, docházelo k chybám při chytání. Tento problém byl způsoben menší hmotností a větším objemem tenisového míčku oproti kulečníkové kouli. Odpor vzduchu na tenisový míček nemá zanedbatelné účinky na rozdíl od kulečníkové koule, která se chová daleko robustněji k odporu vzduchu. Tenisový míček dále výrazně více odskakuje od ramínka než kulečníkové koule. Proto byly parametry upraveny, aby bylo možné prohazovat i tenisový míček.

2.5.5.2. FUNKCE JEDNÉ KOULE

Tato funkce má demonstrovat člověka, který by si přehazoval kouli z jedné ruky do druhé. Jako ruce člověka slouží dvě přehazovací ramínka, která jsou od sebe vzdálena 300 mm (obrázek 2.26). K přehozu mezi těmito ramínky bude docházet pouze v rovinách x a y kolmé k přední stěně žongléra (obrázek 2.28). K tomu byly popsány rovnice v kapitole 2.5.3. Maximální úhel, při kterém je možné vyhodit prvním ramínkem kouli tak, aby byla ještě chycena druhým ramínkem, je 18°.



Obrázek 2.28 – Poloha ramínka při přehozu koule

Pro výpočet počáteční vektorové rychlosti v_0 a úhlu α se musí stanovit doba letu koule, která má hodnotu t = 0,8 s.

Po dosazení do rovnic (14) pro vrh šikmý vzhůru se dostane soustava dvou rovnic o dvou neznámých:

$$\begin{aligned} &x = x_0 + v_0 \cdot t \cdot \cos\alpha \\ &y = y_0 + v_0 \cdot t \cdot \sin\alpha - \frac{1}{2} \cdot g \cdot t^2 \\ &---------- \\ &0{,}3 = v_0 \cdot 0{,}8 \cdot \cos\alpha \\ &0 = v_0 \cdot 0{,}8 \cdot \sin\alpha - \frac{1}{2} \cdot 9{,}81 \cdot 0{,}8^2 \end{aligned} \qquad \textbf{(14)}$$

Nyní stačí z uvedených rovnic (14) vyjádřit v_0 a velikost úhlu α, platí (15) a (16):

$$\begin{aligned} &v_0 = \frac{0{,}3}{0{,}8 \cdot \cos\alpha} \\ &v_0 = \frac{0{,}5 \cdot 9{,}81 \cdot 0{,}8^2}{0{,}8 \cdot \sin\alpha} \\ &---------- \end{aligned} \qquad \textbf{(15)}$$

$$\frac{0{,}3}{0{,}8\cdot\cos\alpha}=\frac{0{,}5\cdot 9{,}81\cdot 0{,}8^2}{0{,}8\cdot\sin\alpha}\quad /\cdot 0{,}8\cdot\cos\alpha\cdot\sin\alpha$$

$$0{,}3\cdot\sin\alpha=0{,}5\cdot 9{,}81\cdot 0{,}8^2\cdot\cos\alpha$$

$$\frac{\sin\alpha}{\cos\alpha}=tg\alpha=\frac{0{,}5\cdot 9{,}81\cdot 0{,}8^2}{0{,}3}=10{,}464\Rightarrow\alpha=\underline{\underline{84°32'}} \qquad \textbf{(16)}$$

$$v_0=\frac{0{,}3}{0{,}8\cdot\cos\alpha}=\frac{0{,}3}{0{,}8\cdot\cos 84°32'}=\underline{\underline{3{,}936\,m\cdot s^{-1}}}$$

Z výsledných rovnic vyšel počáteční vektor rychlosti $v_0=3{,}936\,m\cdot s^{-1}$ a úhel $\alpha=84°32'$

Pro výpočet vektorů rychlosti v_{0x} a v_{0y} platí vztah (17):

$$v_{0X}=v_0\cdot\cos\alpha=3{,}936\cdot\cos 84°32'=0{,}375\,m\cdot s^{-1}$$
$$v_{0Y}-g\cdot t=v_0\cdot\sin\alpha=3{,}936\cdot\sin 84°32'=3{,}918\,m\cdot s^{-1} \qquad \textbf{(17)}$$

Nyní je potřeba přepočítat velikost rychlosti v_{0x} z $[m\cdot s^{-1}]$ na $[°\cdot s^{-1}]$, protože se údaje zadávají do vačkového profilu horizontálního motoru ve stupních za sekundu.

K tomu, aby mohl být proveden tento přepočet, musí být známa délka přehazovacího ramínka a úhel otočení, za který bude dosažena rychlost v_{0x}. Délka přehazovacího ramínka je 150 mm a úhel otočení je 36,5°.

Dráhu, kterou opíše přehazovacího ramínko při svém pohybu, udává vztah (18):

$$l_X=\frac{2\cdot\pi\cdot r\cdot 36{,}5}{360}=\frac{2\cdot\pi\cdot 0{,}15\cdot 36{,}5}{360}=0{,}09556\,m \qquad \textbf{(18)}$$

Nyní stačí dosadit do vzorce (19) pro zjištění rychlosti v_{0x} $\left[\circ \cdot s^{-1}\right]$:

$$\begin{aligned} &0{,}09556m \ldots\ldots\ldots\ldots 0{,}375m \cdot s^{-1} \\ &36{,}5° \ldots\ldots\ldots\ldots x° \cdot s^{-1} \\ &\text{----------} \\ &x\left[\circ \cdot s^{-1}\right] = v_{0x}\left[\circ \cdot s^{-1}\right] = \frac{36{,}5 \cdot 0{,}375}{0{,}09556} = 143{,}23° \cdot s^{-1} \end{aligned} \qquad \textbf{(19)}$$

Pro zajištění podmínky, že koule nenarazí při svém maximálním výhozu horní část žongléra, je potřeba vypočítat maximální výšku výhozu, pro kterou platí rovnice (20):

$$h = \frac{v_0^2 \cdot \sin^2 \alpha}{2 \cdot g} = \frac{3{,}936^2 \cdot \sin^2 84°32'}{2 \cdot 9{,}81} = 0{,}782m \qquad \textbf{(20)}$$

Z identifikace žongléra (kapitola 2.5.4.2) vyšla maximální výška pro žonglování $h_{\max} = 2160mm$. Podmínka je tedy splněna.

Nyní jsou již vypočítané všechny parametry pro tvorbu vačkového profilu *jedné koule*. Návrh je obsažen v příloze č.5, kde je také částečně změněná rychlost výhozu a chytání pro obě složky x a y. Zde byl stejný problém s tenisovým míčkem jako u funkce *katapult*, proto byly parametry změněny, aby bylo možné přehazovat i tenisový míček.

2.5.5.3. FUNKCE DVOU KOULÍ

Stejně jako funkce s *jednou koulí* má také funkce se dvě*ma koulemi* demonstrovat člověka přehazující dvě koule z jedné ruky do druhé. I tady místo rukou člověka slouží přehazovací ramínka. Pro výpočet jednotlivých parametrů této funkce se použijí již vypočítané výsledky z funkce *jedné koule*, protože se jedná o stejný princip. Aby mohly být přehozeny dvě koule najednou, musí být rozdílná rychlost výhozu nebo výška, ve které budou vyhazovány. Jednodušší je změnit výšku výhozu, kdy bude mít jedna koule o 10 cm větší počáteční výšku výhozu než ta druhá, a tím nedojde k jejich vzájemné srážce.

Po převzetí výsledků z funkce *jedné koule* dostáváme vztahy (21):

$$\begin{aligned} v_{0X} &= v_0 \cdot \cos\alpha = 3{,}936 \cdot \cos 84°32' = 0{,}375 m \cdot s^{-1} \rightarrow 143{,}23° \cdot s^{-1} \\ v_{0Y} &= v_0 \cdot \sin\alpha = 3{,}936 \cdot \sin 84°32' = 3{,}918 m \cdot s^{-1} \end{aligned} \qquad \textbf{(21)}$$

Ještě je nutné ověřit podmínku změny počáteční výšky výhozu, zda nedojde ke srážce s horní částí žongléra při maximálním výhozu koule. Pro výpočet stačí převzít maximální výšku pro jednu kouli a přičíst k ní 10 cm. Maximální výhoz koule je $h = 0{,}882 m$ a maximální výška pro žonglování je $h_{\max} = 2160 mm$. Podmínka je splněna.

Nyní jsou již vypočítané všechny parametry pro tvorbu vačkového profilu *dvou koulí*. Návrh je obsažen v kapitole 2.5.6, kde je také částečně změněná rychlost výhozu a chytání pro obě složky x a y, stejný problém jako pro funkce *katapult* a *jedné koule*.

2.5.5.4. FUNKCE TŘÍ KOULÍ

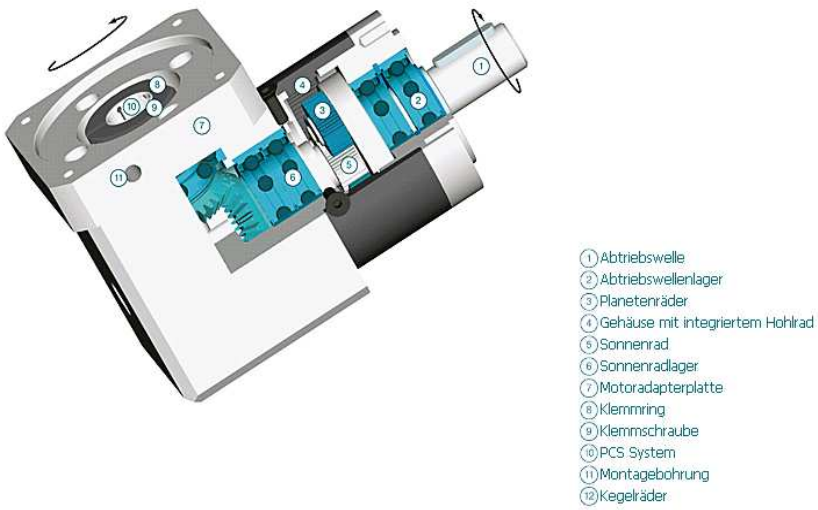
V rámci soutěže European Industrial Ethernet byl doplněn model žongléra o funkci *3 koulí*. Návrh vačkových profilů pro tuto funkci byla jedna z částí bakalářské práce *Dynamika rychlých servopohonů*, kterou vypracoval Pavel Jaroš.

Pro funkci *3 koulí* dokázaly vertikální motory běžet pouze cca 5 minut, protože se velmi rychle zahřívaly, a proto po dosažení této teploty byly bezpečně odpojeny. V rámci veletrhu Ampér 2010 byl model žongléra modernizován a doplněn o převodovky, tak aby bylo možné mít funkci *3 koulí* nepřetržitě zapnutou a zamezit přehřívání motorů.

2.5.5.4.1. ROZŠÍŘENÍ MODELU O PŘEVODOVKY

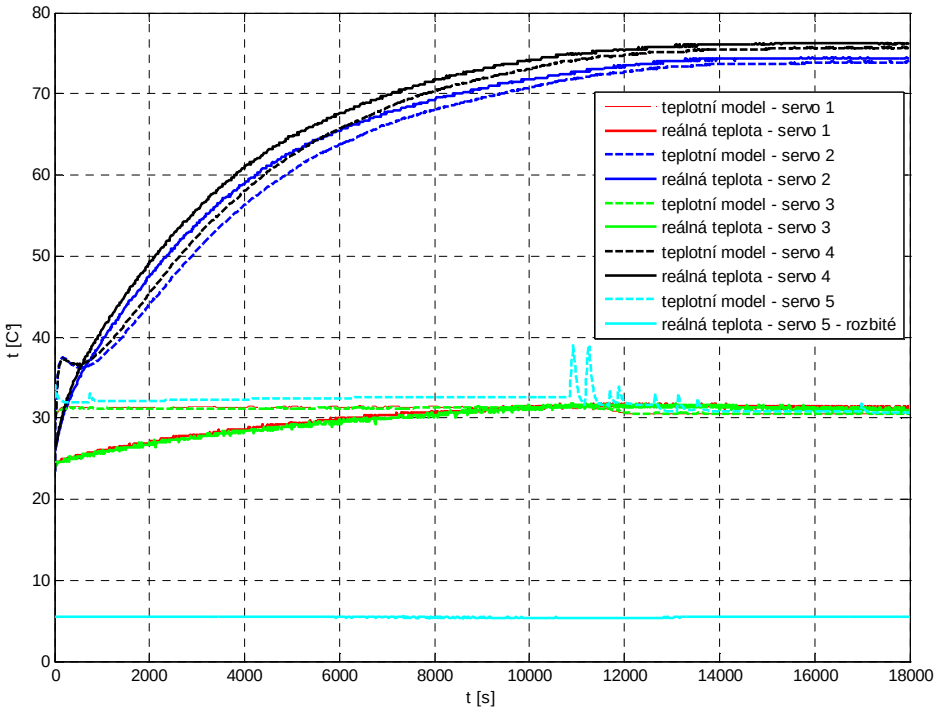
Rozšířením modelu o převodovky (obrázek 2.29) zajistíme menší namáhání vertikálních pohonů, které v původní verzi vydržely v kontinuálním režimu žonglování se třemi koulemi pouze po dobu cca 5 minut, poté byly z důvodu teplotního přetížení pomocí servozesilovačů bezpečně odpojeny od zdroje. Aplikace převodovek slibuje teplotní stabilizaci motoru pod maximální povolenou úroveň (110 ℃), p případně zajistí rezervu pro žonglování se čtyřmi koulemi.

Protože byly vertikální pohony přetěžovány momentově (dlouhodobě překročená hodnota nominálního momentu), ale měly rezervu v nominálních otáčkách, byly k tomuto účelu zakoupeny převodovky WPLE 120 - 3 od firmy fa Neugart s převodovým poměrem 3:1 (převod do pomala).



Obrázek 2.29 – Použitá převodovka s převodem 3:1

Po nasazení převodovek je možný nepřetržitý běh v plném provozu, aniž by došlo k přehřátí motorů. Teplotní model žongléru s použitím převodovek je na obrázku 2.30.

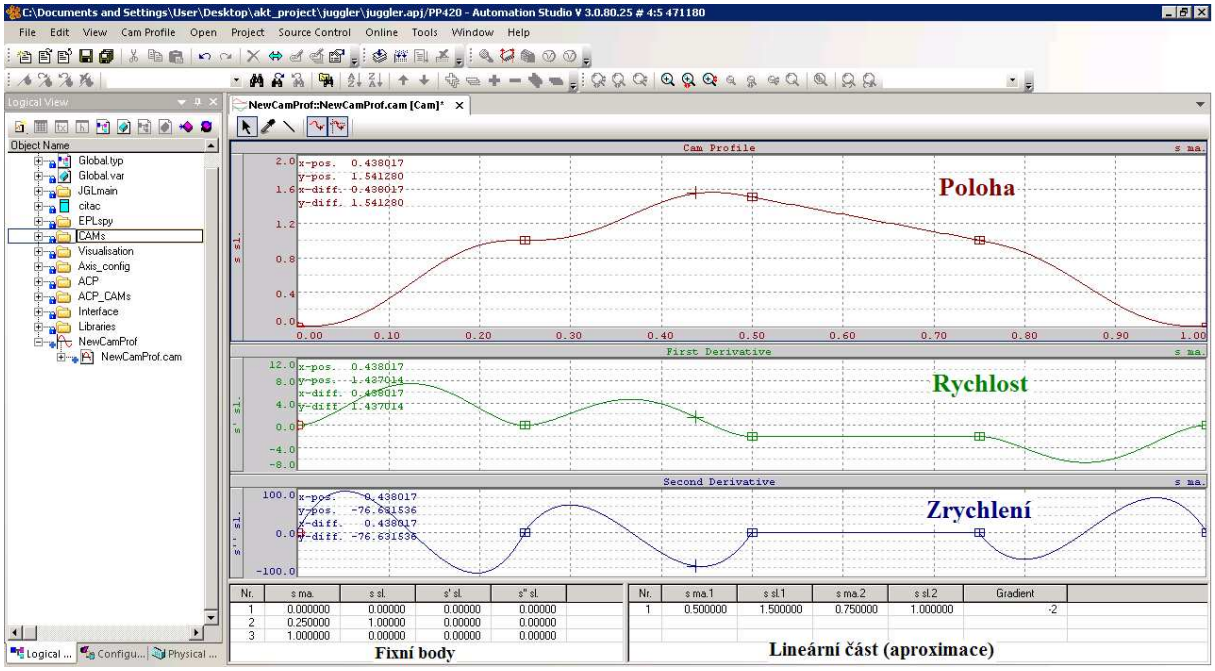


Obrázek 2.30 – Teplotní model žongléra s použitím převodovek

2.5.6. NÁVRH VAČKOVÝCH PROFILŮ

Vačkový profil je součástí softwaru Automation Studia, který umožňuje definování polohy, rychlosti a zrychlení v závislosti na čase. Vačkový editor se skládá z několika částí (obrázek 2.31). Pro tvorbu jsou nejdůležitější veličiny poloha, rychlost a zrychlení. Průběhy se tvoří pomocí fixních bodů případně jejich aproximací. Předností vačkového profilu je automatické dopočítávání rychlosti a zrychlení při zadávání polohy v čase.

Poloha, rychlost a zrychlení jsou definovány tzv. „unitech“. Motory, které vykonávají vertikální pohyb, mají na 3 otáčky 2900 unitů ($10000\,unitů = 1m$). Pro horizontální motory platí 1 otáčka 3600 unitů ($3600\,unitů = 360°$). Výhodou vačkového editoru je, že je možné nastavit poměr v unitech k určité vzdálenosti, která je pro uživatele nejlepší při tvorbě vačkových průběhů. Pro následující průběhy (kapitola 2.5.6.1) platí, že vertikální směr pohybu má poměr $1unit = 1cm$ a horizontální $1unit = 0{,}1°$.



Obrázek 2.31 – Vačkový profil v Automation studiu

Aby bylo možné ovládat současně několik motorů (model žongléra jich obsahuje pět) vačkovými profily, je nutné tyto motory synchronizovat. Synchronizace vačkových profilů v Automation studiu je realizována virtuální osou, která tvoří **master** osu. Vačky závislé se pak nazývají **slave** a sledují pohyb nadřazené osy - master.

Při tvorbě vačkových profilů lze nastavit v jakém časovém poměru má slave osa sledovat master osu. Je-li například rychlost master osy 1 unit/s, potom 1 dílek slave osy je 1 sekunda. Ale změní-li se rychlost master osy na 100 unit/s, potom 1 dílek slave osy je 0,01 sekundy. V našem případě je rychlost master osy 1 unit/s, proto jsou všechny uvedené časové diagramy vačkových profilů pro osu x (časová osa) v sekundách.

2.5.6.1. FUNKCE DVOU KOULÍ

Velikosti fixních bodů pro horizontální a vertikální průběhy *dvou koulí* byly vypočítány v kapitole 2.5.5.3. Nyní je stačí nadefinovat do vačkového editoru v závislosti na vypočítaném čase.

Vyhotovené vačkové průběhy polohy, rychlosti a zrychlení pro horizontální a vertikální směr jsou na obrázcích 2.32 a 2.33. Velikosti fixních bodů pro tyto průběhy jsou uvedeny v tabulkách 2.2 a 2.3, kde lze dobře vidět, jak se musela změnit vypočítaná rychlost, aby bylo možné žonglovat i s tenisovým míčkem. Kompletní návrh vačkového profilu je obsažen v příloze č.6.

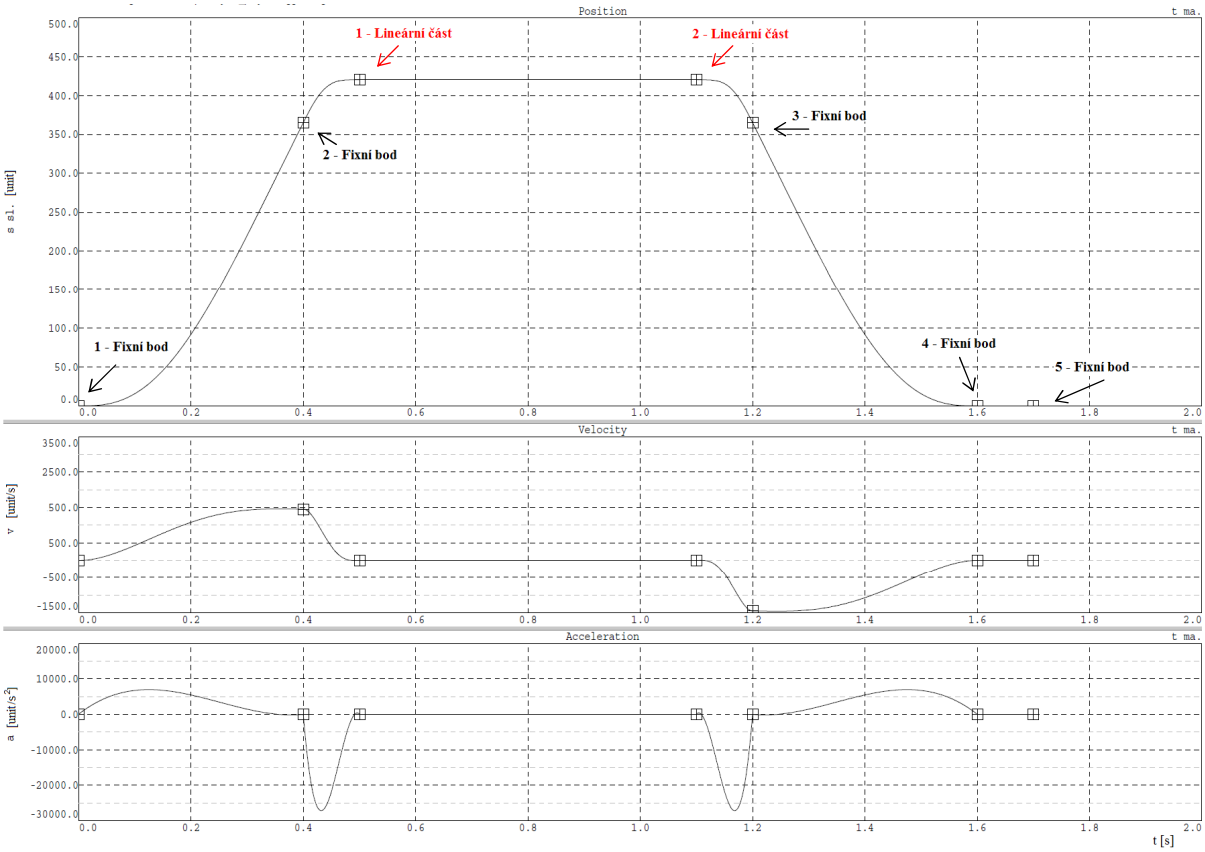
Příklad:

- Horizontální směr
  - vypočítaná rychlost $v_{0X} = 143{,}23° \cdot s^{-1}$ (kapitola 2.5.5.3 rovnice (21))
  - skutečná rychlost $v_{0X} = 1450{,}00 unit \cdot s^{-1}$ (tabulka 2.2 – číslo fixpointu 2 – Fixní bod, pozor na poměr $1 unit = 0{,}1°$)

- Vertikální směr
  - vypočítaná rychlost $v_{0Y} = 3{,}918 m \cdot s^{-1} = 391{,}8 cm \cdot s^{-1}$ (kapitola 2.5.5.3 rovnice (21))
  - skutečná rychlost $v_{0Y} = 393{,}6 unit \cdot s^{-1}$ (tabulka 2.3 – číslo fixpointu 1 – Fixní bod, pozor na poměr $1 unit = 1 cm$)

Na obrázku 2.33 lze také velmi dobře vidět, že polohy fixních bodů pro výhoz a chytání koule ve vertikálním směru pro druhý servomotor mají různou výšku (tabulka 2.3 – číslo fixpointu 1 a 2 – Fixní bod). Rozdíl těchto výšek je stanovených 10 cm.

Horizontální pohyb

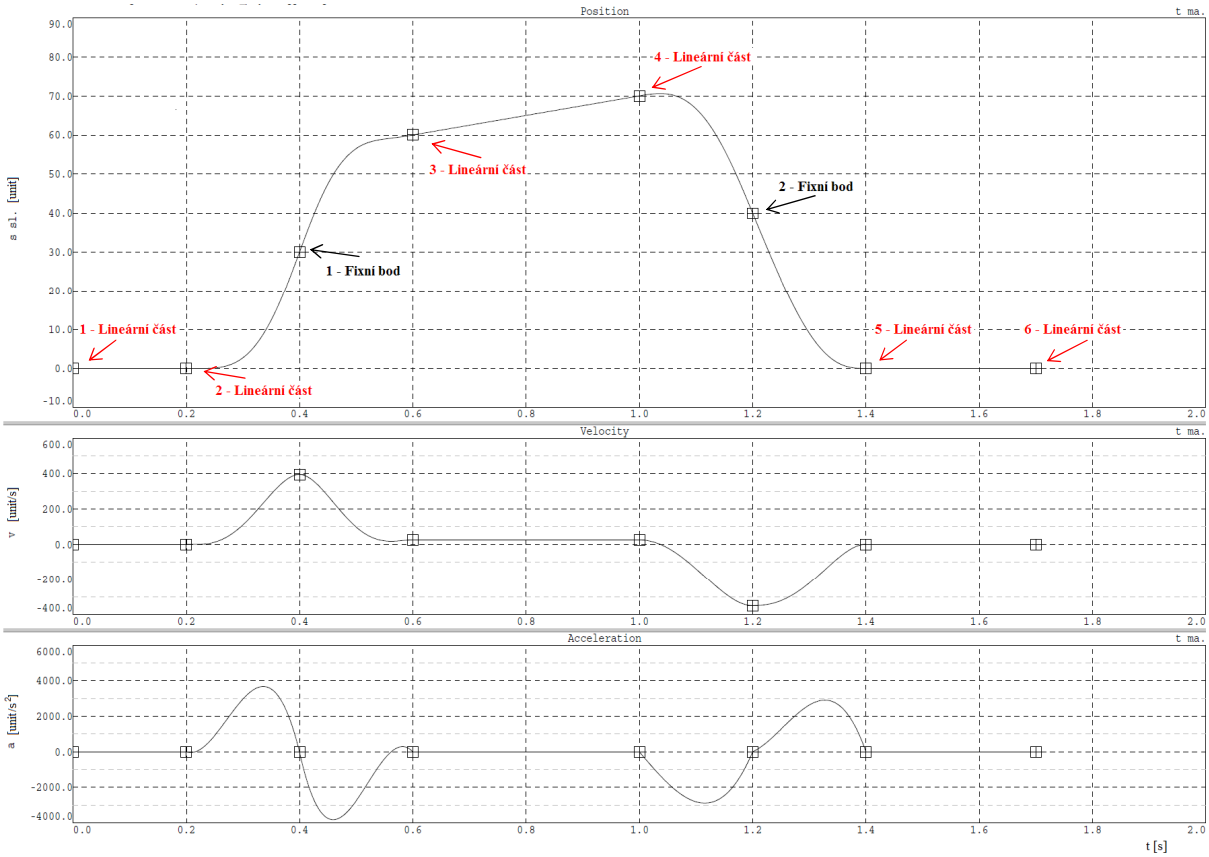


Obrázek 2.32 – Návrh vačky pro dvě koule – horizontální pohyb (motor 1 a 3)

Fixní bod					Lineární část (aproximace)	
č.f.	čas	poloha	rychlost	zrychlení	čas	poloha
	[s]	[unit]	[unit/s]	[unit/s^2]	[s]	[unit]
1	0,000	0,000	-	-	0,500	420,000
2	0,400	365,000	1450,000	-	1,100	420,000
3	1,200	365,000	-1450,000	-		
4	1,600	0,000	-	-		
5	1,700	0,000	-	-		

Tabulka 2.2 – Definice fixních bodů do vačko. editoru (2 kou. – H – s1 a s3)

Vertikální pohyb



Obrázek 2.33 – Návrh vačky pro dvě koule – vertikální pohyb (motor 2)

Fixní bod					Lineární část (aproximace)	
č.f.	čas [s]	poloha [unit]	rychlost [unit/s]	zrychlení [unit/s^2]	čas [s]	poloha [unit]
1	0,400	30,000	393,600	-	0,000	0,000
2	1,200	40,000	-350,000	-	0,200	0,000
3					0,600	60,000
4					1,000	70,000
5					1,400	0,000
6					1,700	0,000

Tabulka 2.3 – Definice fixních bodů do vačko. editoru (2 kou. – V – s2)

3. NAHAZOVACÍ MECHANISMUS

Aby bylo možné používat model žongléra jako vzdálený výukový model, který si student v určitý čas zaregistruje, vzdáleně ho ovládá a pomocí kamery vizuálně kontroluje činnost stroje, muselo se zajistit nahození koule na jednotlivá přehazovací ramínka.

Pro nahození stačí vyhazovat kouli pouze v jedné rovině, a to ve vertikálním směru, protože se využije rozsah pohybu přehazovacího ramínka, které se dokáže pohybovat ve všech třech osách a po nahození koule si pro ni může zajet a v letu ji zachytit. Jelikož se koule nahazuje pouze v jedné rovině, postačí pro tuto aplikaci lineární motor (kapitola 3.1).

Pro nahození se musí zajistit umístění koule na tento lineární motor, proto byl model žongléra rozšířen o zásobník na koule (kapitola 3.2), který obsahuje dopadovou plochu, a zásobník na koule s příslušným mechanickým a elektrickým ovládáním.

Aby bylo možné vyhodit kouli, musely být vypočteny a navrhnuty vačkové profily (kapitola 3.3) pro tento typ pohybu. Ty se skládají z funkce *nahoď kouli* a *zahoď kouli*. Funkce *zahoď kouli* slouží pro shození koule z přehazovacího ramínka, aby bylo možné testovat pomocí zpětné vazby (kapitola 4) je-li koule nahozena či ne. Jako další aplikace se používá pro funkci *jedné koule*, kdy je potřeba z jednoho přehazovacího ramínka vyhodit kouli, aby nedošlo ke kolizi s přehazovanou koulí.

3.1. LINEÁRNÍ MOTOR

Pro mechanizmus nahození koule byl vybrán lineární motor Power Rod fy Parker (obrázek 3.1), který obsahuje misku (obrázek 3.2) pro umístění koule při náhozu ve vertikálním směru. Celé zařízení se skládá z magnetické tyče, která tvoří stator, a pohyblivých cívek tvořící rotor. Aby nedocházelo k rotaci cívek kolem magnetické tyče, byl lineární motor doplněn o vodící konstrukci (obrázek 3.2), která není moc vhodná pro naše řešení. Po zahřátí dochází k teplotní roztažnosti kovů a motor po delší době provozu přestává správně nahazovat a není možné zachytit kouli v letu přehazovacím ramínkem pro funkci *3 koulí*.

Obrázek 3.1 – Lineární motor typ PRC 25

Požadavky kladené na lineární motor:

Délka pojezdu cca:	1 m
Minimální rychlost:	5 m/s
Minimální zrychlení:	50 m/s^2
Váha koule:	0,2 kg
Minimální nadhoz:	0,5 m

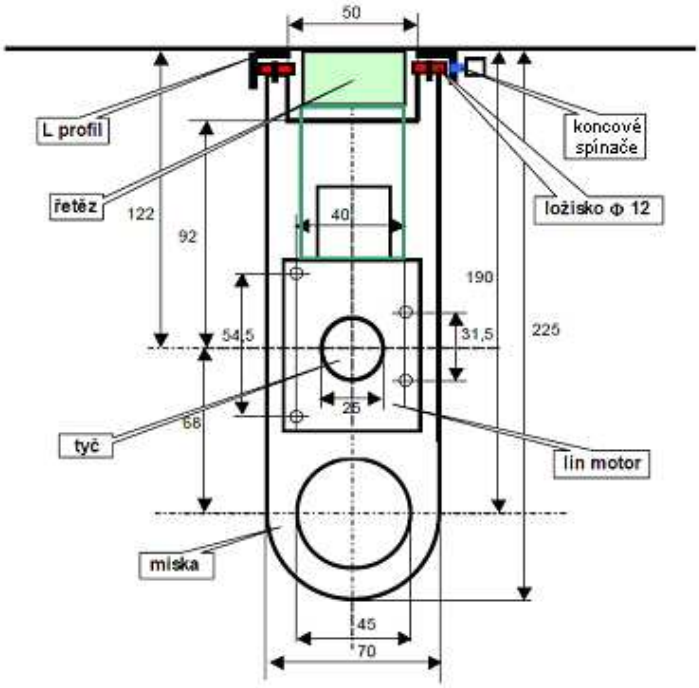
Parametry lineárního motoru:

Výrobce: Parker Hannifin GmbH

Motor: PRC2504 lineární motor Power Rod s pohyblivými cívkami a s pevnou zmagnetovanou tyčí

- Rozměry:
  - rotor PRC2504 – pohyblivé cívky 54x68x160 mm
  - stator TRC 25 – zmagnetovaná tyč Φ 25 mm, délka 1380mm
- Délka pojezdu: 1180 mm
- Max rychlost: 8,7 m/s – bez zátěže
- Max zrychlení: 223 m/s^2 – bez zátěže
- Špičková síla: 312 N (pro 1 sec) - 14,1 A
- Stálá síla: 42 N – 2,31 A

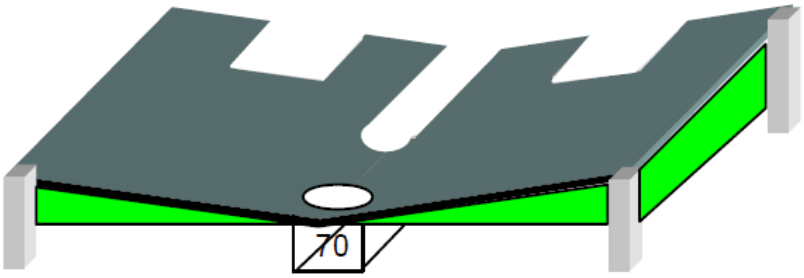
- Moment:
  - jmenovitý 51 N*m
  - maximální 312 N*m
- Hmotnost: 1,15 kg



Obrázek 3.2 – Řešení misky a vedení lineárního motoru

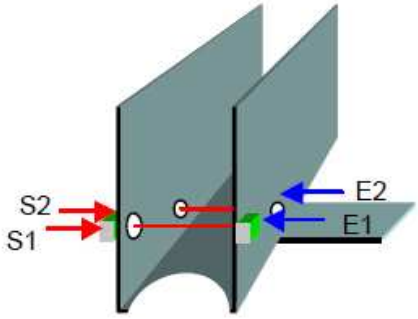
3.2. REALIZACE ZÁSOBNÍKU

Aby bylo možné do zásobníku doplňovat spadené či náhradní koule, byla vymyšlena a realizována dopadová plocha (obrázek 3.3). Jakmile dojde ke spadnutí či položení koule na dopadovou plochu, vlivem jejího tvaru, která připomíná roztáhnuté V, dojde k zajetí koule do otvoru v nejnižším místě plochy a zařazení do zásobníku. Velikost otvoru je 70 mm, proto je možné nahazovat kulečníkovou kouli i tenisový míček.

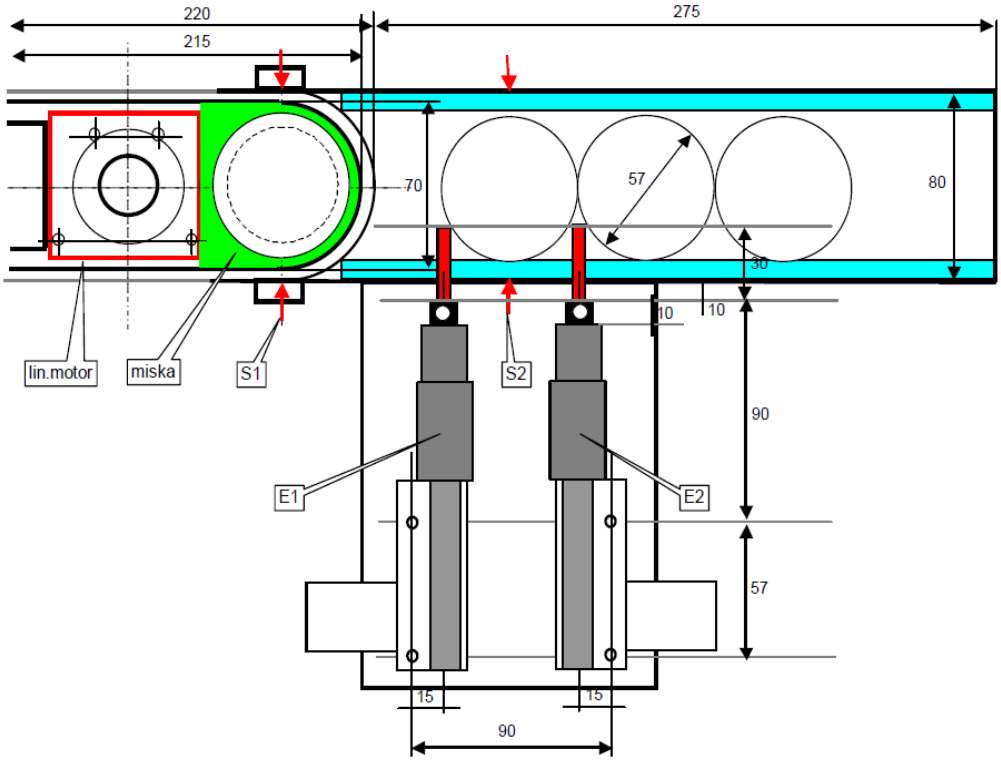


Obrázek 3.3 – Dopadová plocha zásobníku

Zásobník je tvořen dvojčinnou elektrickou zarážkou E1 a E2 (obrázek 3.4 a 3.5), která na základě řídícího povelu z PLC uvolní nachystanou kouli nebo míček na zásobník lineárního motoru. Pro detekování koule slouží dvě optické brány S1 a S2 (obrázek 3.4 a 3.5). První optická brána S1 slouží pro detekování koule na misce lineárního motoru a druhá optická brána S2 detekuji kouli mezi zarážkami S1 a S2, je-li koule nachystána pro najetí na misku lineárního motoru.



Obrázek 3.4 – Zásobníkový tunel



Obrázek 3.5 – Princip mechanické realizace zásobníku

Pro správnou funkci zásobníku fungují zarážka E1 a E2 v protifázi, aby nedošlo k najetí více míčků na misku lineárního motoru. Zarážky E1 a E2 jsou aktivovány na základě vyhodnocení informace z optických bran S1 a S2, zdali je koule přítomna či nikoliv.

Bude-li aktivován režim pro nahození koule a koule nebude umístěna na misku lineárního motoru za určitý čas, bude aktivována na ovládacím panelu zpráva pro uživatele, že došlo k zaseknutí koule v podavači případně, že koule úplně chybí. V nejhorším případě může dojít k poruše optické brány nebo motorku ovládající zarážku.

3.3. VYPOČTENÉ A NAVRŽENÉ HODNOTY PRO ZADANÉ FUNKCE

Pro funkce *nahoď* a *zahoď kouli* postačí kinematický popis pro vrh svislý vzhůru popsaný v kapitole 2.5.2. Pro tyto funkce bude koule vyhazována pouze v jedné rovině a podle toho, jaká funkce bude zapnuta, bude koule nahazována pomocí lineárního motoru nebo odhozena z přehazovacího ramínka.

3.3.1. FUNKCE NAHOĎ KOULI

K tomu, aby bylo možné žonglovat s kulečníkovými koulemi, je nutné tyto koule nejprve nahodit ze zásobníku na přehazovací ramínka pomocí lineárního motoru.

Ve chvíli, kdy koule dosáhne nejvyšší polohy a začne padat, zajede pod ni přehazovací ramínko, které při dotyku s koulí má stejnou rychlost a zrychlení, a poté koná pohyb rovnoměrně zpomalený, aby bylo možné s kouli zastavit, aniž by vypadla. Výška, ve které dochází k chycení koule přehazovacím ramínkem od výhozu koule lineárním motorem, je 655 mm. Pro počáteční rychlost výhozu koule platí podmínka, že musí být dostatečně velká, aby koule dosáhla minimální výšky výhozu 655 mm. Platí vztah (22):

$$y = v_0 \cdot t_h - \frac{1}{2} \cdot g \cdot t_h^2 = \frac{v_0^2}{g} - \frac{v_0^2}{2 \cdot g} = \frac{v_0^2}{2 \cdot g} \qquad \textbf{(22)}$$

Po vyjádření v_0 a dosazení do rovnice (23), dostaneme minimální rychlost výhozu koule, která je $v_{0\min} = 3{,}585\ m \cdot s^{-1}$

$$v_{0\min} = \sqrt{y \cdot 2 \cdot g} = \sqrt{0{,}655 \cdot 2 \cdot 9{,}81} = \pm 3{,}585\ m \cdot s^{-1} \qquad \textbf{(23)}$$

Aby nedošlo ke srážce koule s přehazovacím ramínkem ve chvíli, kdy bude pod kouli najíždět, byla stanovena rychlost $v_0 = 4\,m \cdot s^{-1}$. Zvětšením vyhazovací rychlosti v_0 dojde k větší časové rezervě pro najetí koule vyhazovacím ramínkem.

Čas, kdy dochází k chycení koule, je dán rovnicí (24):

$$\begin{aligned} & y = v_0 \cdot t - \frac{1}{2} \cdot g \cdot t^2 \\ & ---------- \\ & t^2 \cdot \frac{1}{2} \cdot g - t \cdot v_0 + y = 0 \\ & t^2 \cdot \frac{1}{2} \cdot 9{,}81 - t \cdot 4 + 0{,}655 = 0 \end{aligned} \qquad \textbf{(24)}$$

Z rovnice (24) pro čas vychází kvadratická rovnice, pro kterou platí vztahy (25):

$$\begin{aligned} & a = \frac{1}{2} \cdot 9{,}81 \quad b = -4 \quad c = 0{,}655 \\ & D = b^2 - 4 \cdot a \cdot c = (-4)^2 - 4 \cdot \frac{1}{2} \cdot 9{,}81 \cdot 0{,}655 = 3{,}1489 \\ & t_{1,2} = \frac{-b \pm \sqrt{D}}{2 \cdot a} = \frac{4 \pm \sqrt{3{,}1489}}{2 \cdot \frac{1}{2} \cdot 9{,}81} \Rightarrow t_1 = 0{,}5886\ s,\ t_2 = 0{,}2269\ s \end{aligned} \qquad \textbf{(25)}$$

Čas, ve kterém dochází k chycení koule je $t_1 = 0{,}5886\,s$ .V čase $t_2 = 0{,}2269\,s$ dosáhne koule také výšku 0,655, ale při stoupavém pohybu směrem vzhůru. Pro návrh vačkového profilu je potřebný čas, kdy koule padá směrem dolů.

Ještě je potřeba vypočítat rychlost koule v bodě chytání, aby bylo možné nastavit tuto rychlost vertikálnímu motoru, aby nedošlo k vypadnutí koule z přehazovacího ramínka.

Pro tuto rychlost platí rovnice (26):

$$v = v_0 - g \cdot t = 4 - 9{,}81 \cdot 0{,}5886 = -1{,}774 m \cdot s^{-1} \quad \textbf{(26)}$$

Tato rychlost vyšla záporná, protože motor koná pohyb směrem dolů, nikoliv nahoru.

Nyní jsou již vypočítané všechny parametry pro tvorbu vačkového profilu *nahoď kouli*. Návrh vačkového profilu je v příloze č.7, kde je také částečně změněná rychlost výhozu a chytání pro obě složky x a y, stejný problém jako pro funkce *katapult*, *jedné koule* a *dvou koulí*.

3.3.2. FUNKCE ZAHOĎ KOULI

Tato funkce se používá pro žonglování s jednou koulí, kdy na druhém ramínku nesmí být obsažena koule, aby bylo možné do tohoto ramínka přehodit kouli. Také se uplatňuje, když není dostatek koulí v podavači, případně pro testování zpětné vazby (kapitola 4), je-li koule na přehazovacím ramínku přítomna či nikoliv.

Princip funkce spočívá ve vyhození koule do určité výšky. Po vyhození koule přehazovací ramínko najede do jiné polohy v horizontálním směru, aby mohla koule spadnou na dopadovou plochu, a poté do zásobníků koulí.

Pro počáteční rychlost výhozu koule je potřeba zvolit výšku výhozu koule, která je 20 cm. Pro výpočet rychlosti platí rovnice (27):

$$y = v_0 \cdot t_h - \frac{1}{2} \cdot g \cdot t_h^2 = \frac{v_0^2}{g} - \frac{v_0^2}{2 \cdot g} = \frac{v_0^2}{2 \cdot g} \quad \textbf{(27)}$$

Po vyjádření v_0 a dosazení do rovnice (28), dostaneme rychlost výhozu koule, která je $v_0 = 2\, m \cdot s^{-1}$

$$v_0 = \sqrt{y \cdot 2 \cdot g} = \sqrt{0{,}2 \cdot 2 \cdot 9{,}81} = \pm 1{,}98\, m \cdot s^{-1} \doteq 2\, m \cdot s^{-1} \quad \textbf{(28)}$$

Nyní jsou již vypočítané všechny parametry pro tvorbu vačkového profilu *zahoď kouli*, který je obsažen v příloze č.8.

4. DETEKCE PŘÍTOMNOSTI KOULE

Pro jednotlivé režimy žonglování je program nastaven tak, aby předpokládal, že na přehazovacích ramínkách není umístěna žádná koule (nejhorší možnost), protože model žongléra neobsahuje žádné senzory na detekci koule. Proto byla koule vždy nahozena, i když se na přehazovacím ramínku vyskytovala. Tento způsob je velmi neefektivní a zbytečně dochází k časové prodlevě, proto je potřeba zavést zpětnou vazbu na detekci koule. Protože není možné jednoduchým způsobem umístit na přehazovací ramínka senzory, které by dokázaly snímat a posléze posílat zprávu řídícímu PLC, jsou-li koule přítomny na ramínkách, byly zvoleny tyto dostupnější metody:

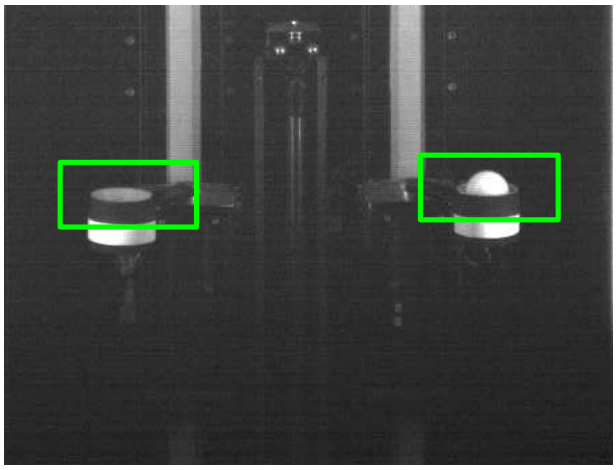
1. Detekce koule pomocí rychloběžné kamery (kapitola 4.1)
2. Detekce koule pomocí kroutícího momentu (kapitola 4.2)

4.1. DETEKCE KOULE POMOCÍ KAMERY

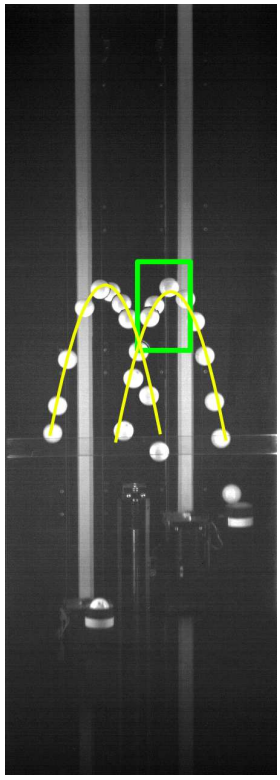
Pro detekci koule byla použita rychloběžná kamera (obrázek 4.1), protože již jednou byla použita pro řízení tohoto modelu v rámci diplomové práce s názvem: *Positional Visual Feedback Control* od Tomáše Němce.

Obrázek 4.1 – Detekce koule pomocí kamery

Detekce koule se provádí při inicializaci, jsou-li koule umístěny na ramínkách pro žonglování vybraného režimu (obrázek 4.2), a při žonglování se třemi koulemi (obrázek 4.3), kdy dochází k nahození spadené koule za chodu, aniž by došlo k zastavení žongléra.



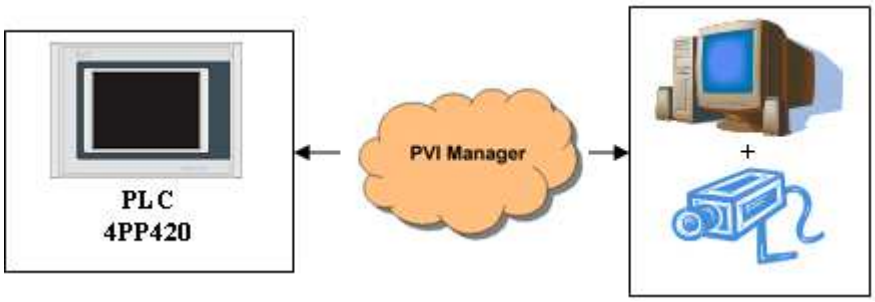
Obrázek 4.2 – Detekce koule při inicializaci pomocí kamery



Obrázek 4.3 – Detekce koule v režimu 3. koulí pomocí kamery

4.1.1. KOMUNIKACE S KAMEROU

Komunikace s PC a počítačem obsluhující kameru je prováděna pomocí PVI Manager (obrázek 4.4). PVI Manager navazuje spojení se zařízeními v B&R Automation Net.



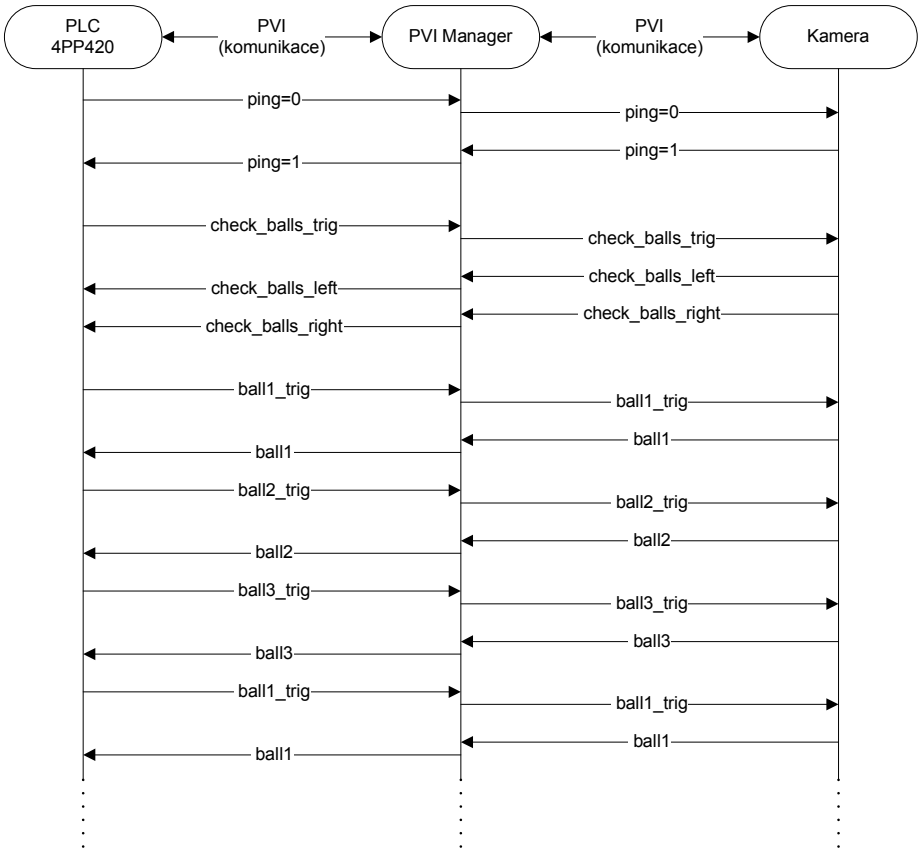
Obrázek 4.4 – Komunikace přes PVI Manager

Pro testování kamery, je-li zapnuta, se posílá signál `ping`, který je nastaven do logické jedničky. Pokud je kamera zapnuta, nastaví se tato proměnná do logické nuly. Tento signál je posílán každých 5 sekund, aby mohla být obsluha informována, došlo-li k vypnutí kamery.

Pro počáteční inicializaci koule na přehazovacích ramínkách je posílán signál `check_balls_trig`, který informuje kameru, že může pustit proces rozpoznávání pro detekci levé a pravé koule. Není-li koule detekována, nastaví logické proměnné `check_ball_left` a `check_ball_right` do logické nuly a tím dojde ke spuštění na PLC podprogramu pro nahození těchto koulí.

Pro detekování míčků v režimu tří koulí jsou posílány kameře v předem daných časových intervalech tři signály `ball1_trig`, `ball2_trig` a `ball3_trig`, které se periodicky opakují a posílají kameře spouštěcí signály pro detekci příslušné koule. Pokud dojde ke špatnému chycení a koule spadne, kamera detekuje, že došlo k chybě, a podle toho jaký ji přišel spouštěcí signál z PLC, nastaví logické proměnné `ball1`, `ball2` nebo `ball3` do logické jedničky, a tím dojde ke spuštění na PLC podprogramu pro nahození příslušné koule.

Celý proces komunikace a přenos jednotlivých proměnných je zobrazen na obrázku 4.5.



Obrázek 4.5 – Komunikace s kamerou – posílání proměnných

Nevýhoda této rychloběžné kamery je, že snímá pouze černobílý obraz. Protože model žongléra má černý podklad, byl algoritmus pro rozpoznání koulí určen pro bílé spektrum, proto je možné rozpoznat pouze koule, které jsou bílé. Návrh algoritmu a řídícího procesu jako zpětná vazba pro detekci koulí na modelu žongléra se zabýval v bakalářské práci *Vizuální kontrola chodu* stroje Tomáš Kohout.

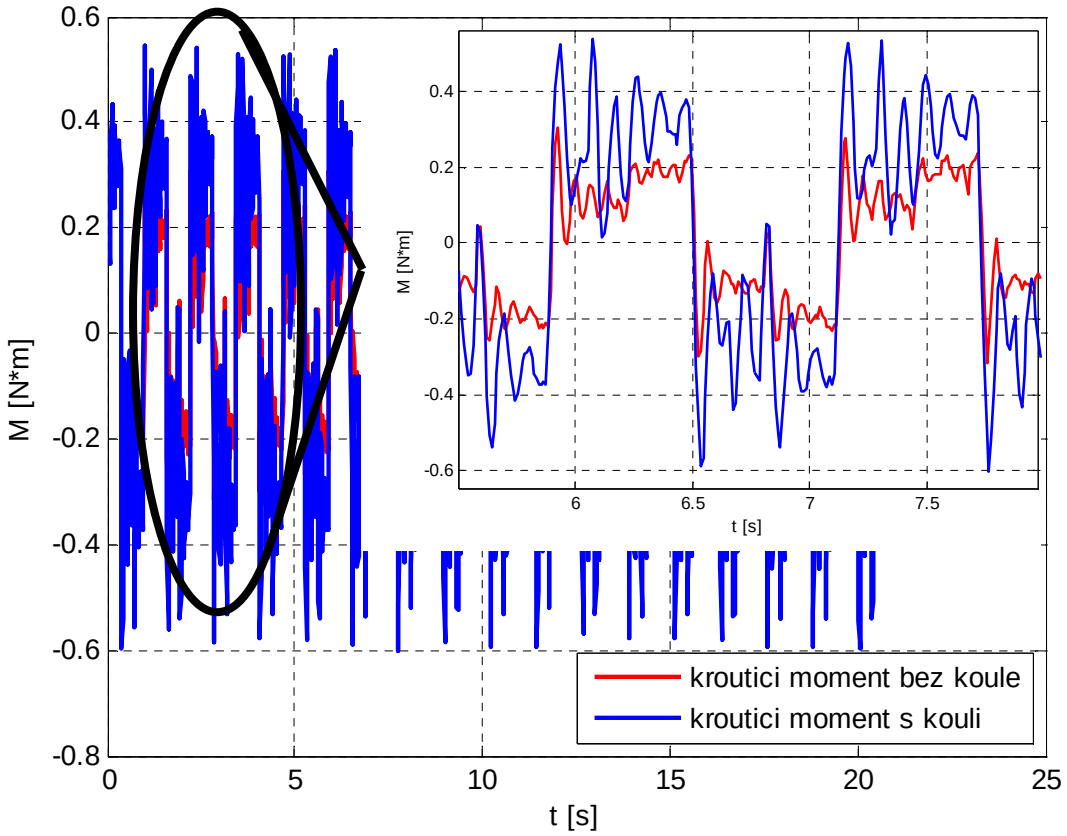
Detekce koule pomocí rychloběžné kamery je dosti drahá záležitost, která by šla nahradit levnějšími měřícími senzory. Ale pro další aplikace na modelu žongléra ji lze využít jako bezpečnostní prvek při kontaktu s obsluhou, pro kontrolu chodu stroje a mnoho dalších aplikací.

4.2. DETEKCE KOULE POMOCÍ KROUTICÍHO MOMENTU

Aby bylo možné model žongléra používat pro výstavy a různé prezentace, bez nutnosti převozu rychloběžné kamery a její následné instalace, kalibrace a jiných podpůrných prací, byla zavedena druhá zpětná vazba v závislosti na velikosti krouticího momentu, která je měřena na dvou servomotorech, které uskutečňují horizontální pohyb při žonglování. Při návrhu této zpětné vazby byla využita možnost Acoposu měřit různé veličiny na motoru v závislosti na čase. V tomto případě velikost krouticího momentu.

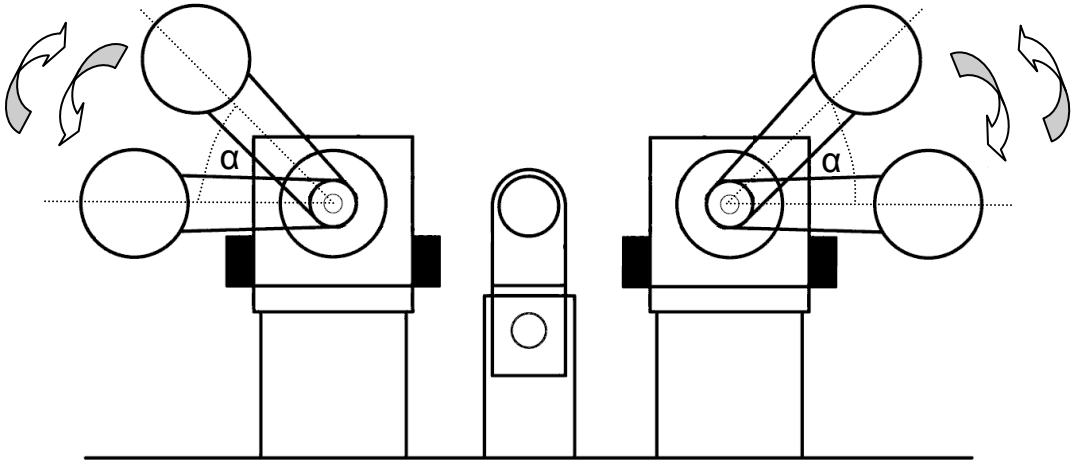
4.2.1. MĚŘENÍ A ZPRACOVÁNÍ KROUTICÍHO MOMENTU

Průběh krouticího momentu s kulečníkovou koulí a bez ní při inicializaci je zobrazen na obrázku 4.6.



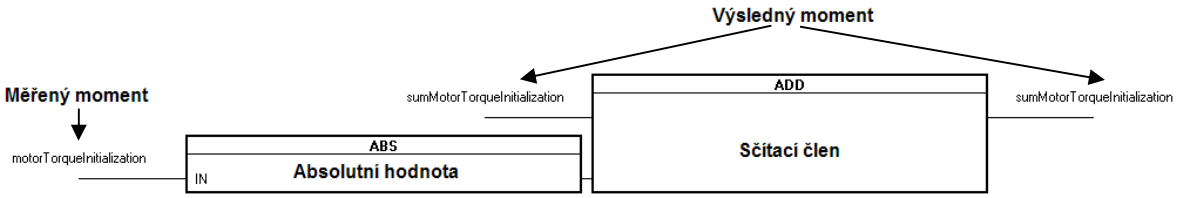
Obrázek 4.6 – Srovnáni krouticího momentu ve výchozí pozici

Pro detekci koule při počáteční inicializaci vykonají obě přehazovací ramínka v předem dané pozici rychlý pohyb tam a zpět o úhel $\alpha = 53{,}2°$ (obrázek 4.7).



Obrázek 4.7 – Zjištění krouticího momentu pro počáteční inicializaci

Při tomto pohybu je zároveň měřena okamžitá hodnota krouticího momentu, která je následně zpracována součtovým členem (obrázek 4.8). Na obrázku 4.6 je vidět, že krouticí moment dosahuje záporných hodnot. To je způsobeno pohybem přehazovacího ramínka při zpětném pohybu, proto je potřeba měřený moment před vstupem do součtového členu dát do absolutní hodnoty, aby nedocházelo k odečítání od kladně naměřeného momentu.



Obrázek 4.8 – Zpracování kroutícího momentu součtovým blokem

Podle naměřené velikosti krouticího momentu je hlídaná mez, pro kterou platí následující tabulka (4.1):

č.m.	S kulečníkovou koulí	S tenisovým míčkem	Na prázdno
	M	M	M
	[N*m]	[N*m]	[N*m]
1	19,36	11,92	8,70
2	19,40	11,20	10,00
3	19,20	11,50	8,90
4	19,00	11,40	9,90
aritmetický průměr	19,24	11,51	9,38

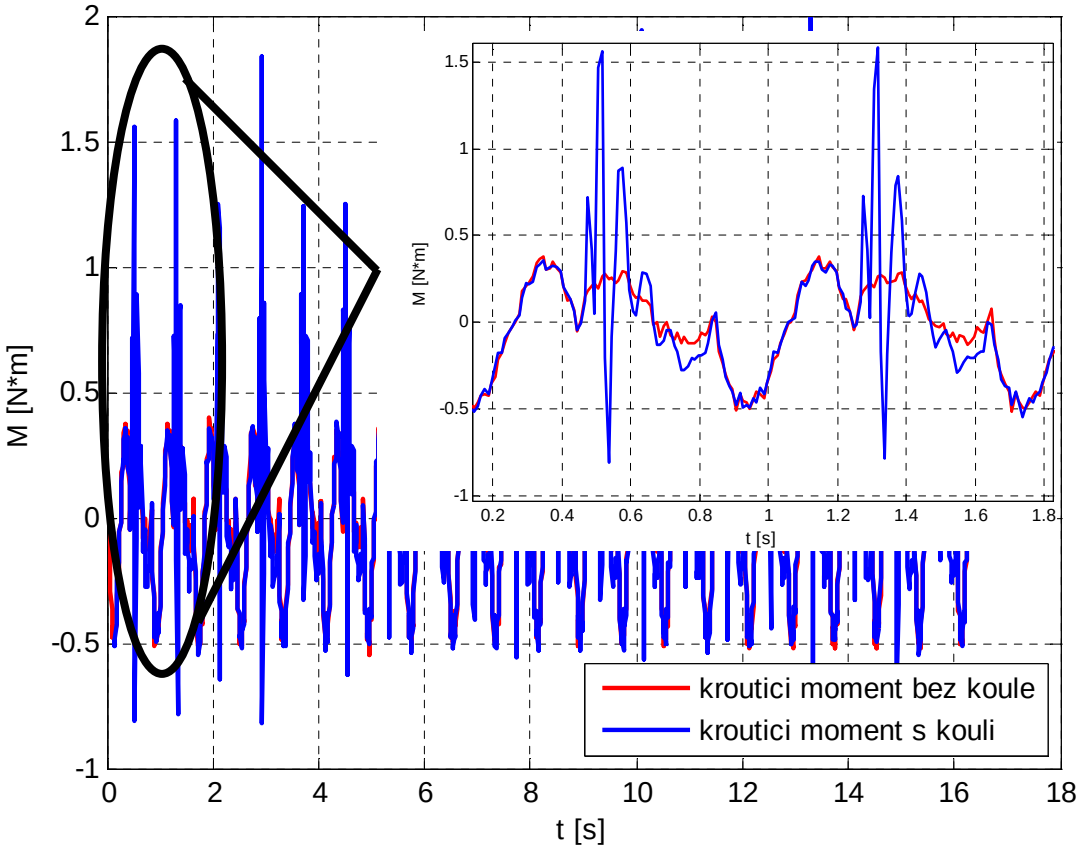
Tabulka 4.1 – Celková velikost krouticího momentu při inicializaci

Pro detekci chybějící koule byla stanovena hranice 11 N*m. Jakmile je krouticí moment při měření menší než 11 N*m dochází k nahození příslušné koule na přehazovací ramínko. I zde, stejně jako u kamery, není-li koule detekována, jsou nahozeny logické proměnné `check_ball_left` a `check_ball_right` do logické nuly, a tím dojde ke spuštění na PLC podprogramu pro nahození těchto koulí.

Pro začátek a konec měření krouticího momentu se používá, obdobně jako u kamery, logická proměnná `mbsg_startTorque_initialization`. Je-li v logické jedničce, probíhá měření, pokud je v logické nule, měření je ukončeno a pošle se signál pro nastavení logických proměnných `check_ball_left` a `check_ball_right`.

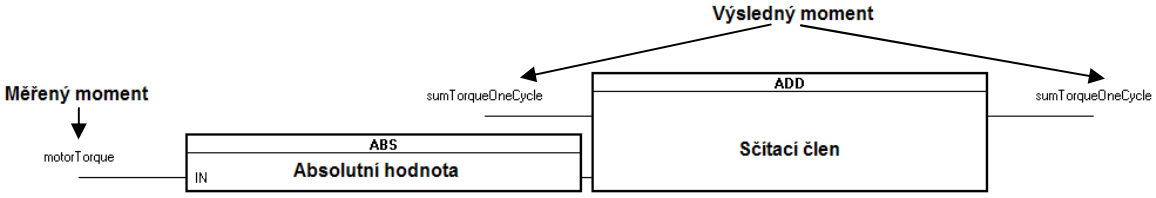
Citlivost krouticího momentu je tak velká, že dokáže detekovat i tenisový míček, který má na rozdíl od kulečníkové koule 3,7 krát menší hmotnost (kulečníková koule m_k=215 g, tenisový míček m_t=57,5 g).

Výsledný graf okamžité hodnoty krouticího momentu v závislosti na čase pro režim 3 koulí je zobrazen na obrázku 4.9. Pro zjištění velikosti kroutícího momentu se provádí měření pouze na pravém motoru vykonávající vodorovný pohyb.



Obrázek 4.9 – Srovnáni krouticího momentu v režimu *tří koulí*

Stejně jako při inicializaci je okamžitá hodnota krouticího momentu následně zpracována součtovým členem (obrázek 4.10). I zde krouticí moment dosahuje záporných hodnot, proto je na vstup součtového členu přivedena absolutní hodnota.



Obrázek 4.10 – Zpracování kroutícího momentu součtovým blokem

Časová doba oběhu jedné koule z jednoho ramínka na druhé a zpět je 2,4 sekundy. Protože se žongluje se třemi koulemi, probíhá měření krouticího momentu na jedné kouli po dobu 0,8 sekund. Stejně jako u kamery je spouštěcí impulz pro měření logická proměnná `ball1_trig`, `ball2_trig` a `ball3_trig`. Každý z těchto signálů přichází 0,8 sekund periodicky po sobě, dokud nedojde k ukončení režimu tří koulí. Pokud bude mít výsledný moment menší hodnotu než hlídaná mez, která byla stanovena na hodnotu 15,7 N*m, dojde k nahození logické proměnné `ball1`, `ball2` nebo `ball3` do logické jedničky, a tím dojde ke spuštění podprogramu pro nahození příslušné koule. Hlídaná mez byla určena z tabulky 4.2.

Bohužel v režimu tří koulí není možné vyhodnotit velikost krouticího momentu pro tenisový míček, protože se zde neprojevují takové setrvačné síly míčku na přehazovací ramínko jako při inicializaci. To je způsobeno menší rychlostí přehazovacího ramínka.

č.m.	S kulečníkovou koulí	Na prázdno
	M	M
	[N*m]	[N*m]
1	18,32	14,22
2	18,64	14,58
3	17,95	15,07
4	18,51	14,93
aritmetický průměr	18,36	14,70

Tabulka 4.2 – Celková velikost krouticího momentu při inicializaci

5. LABORATORNÍ ÚLOHA

5.1. ZADÁNÍ:

1. Seznamte se s výukovým modelem ve vaší laboratoři
2. Založte jednoduchý projekt s hardwarovou konfigurací vašeho modelu
3. Vytvořte jednoduchý program, který umožní zapnout a pohybovat s polohovým zařízením modelu
    a. Program navrhněte tak, aby bylo možné inicializovat a v případě potřeby kdykoliv zastavit polohové zařízeni
    b. Pokud nastane chyba, ošetřete ji bez nutnosti restartovat PLC
    c. Program bude obsahovat jednoduché komentáře v každém networku

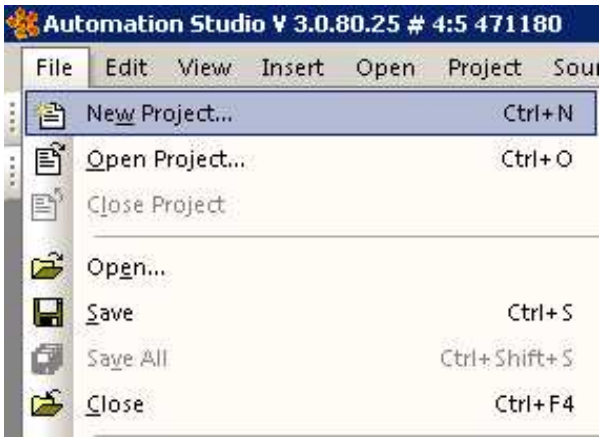
5.2. POSTUP PŘI VYTVOŘENÍ NOVÉHO PROJEKTU

Vytvoření řídicího programu lze shrnout do několika základních kroků:

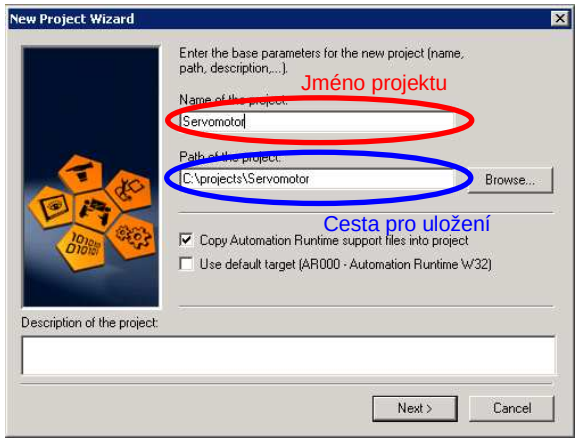
1. Vytvoření nového projektu
2. Hardwarová konfigurace
3. Vytvoření programu
    - Vložení knihoven
    - Deklarace proměnných
    - Vložení funkčních bloků
    - Vlastní naprogramování algoritmů
4. Nastavení komunikace
5. Přeložení programu
6. Nahrání programu do řídicího PLC
7. Ladění a testování

5.2.1. VYTVOŘENÍ NOVÉHO PROJEKTU

Otevřete si Automation studio na vašem počítači a v horní nabídce vyberte File a poté klikněte na New Project, jak je znázorněno na obrázku 5.1. Po kliknutí se vám zobrazí automatický průvodce. Do prvního okna (obrázek 5.2) se musí zadat jméno projektu (Servomotor) a cestu (automatická), kam se má váš projekt uložit.

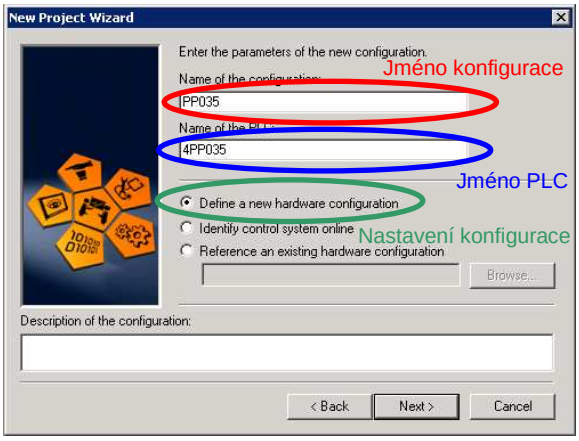


Obrázek 5.1 – Založení projektu

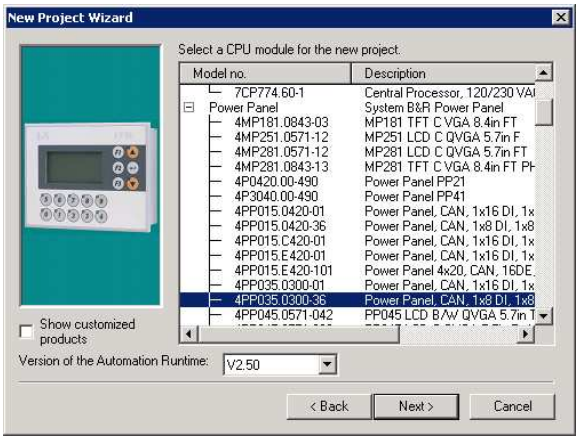


Obrázek 5.2 – Jméno projektu

V dalším okně (obrázek 5.3) je potřeba zadat název konfigurace a PLC. Jelikož pracujeme s PLC 4PP035.0300-36, název konfigurace pojmenujeme „PP035“ a PLC „4PP035“. Dále je třeba vybrat, jestli chceme definovat celou konfiguraci ručně, nebo ji načíst z PLC. Důležité je mít PLC zapnuté a musí být navázána komunikace. Pokud je nemáme zapojeno, vybereme první položku. Nyní vybereme typ PLC (obrázek 5.4), s kterým chceme pracovat.

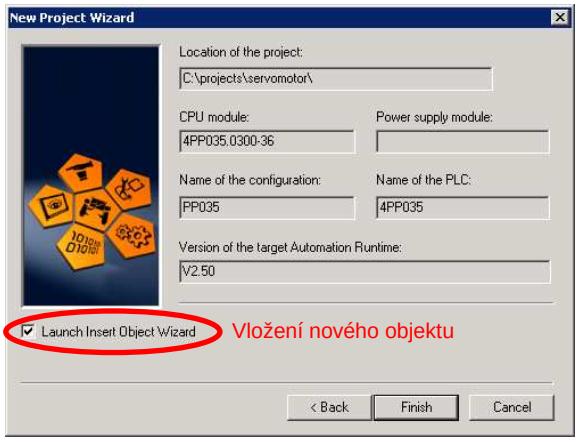


Obrázek 5.3 – Jméno konfigurace

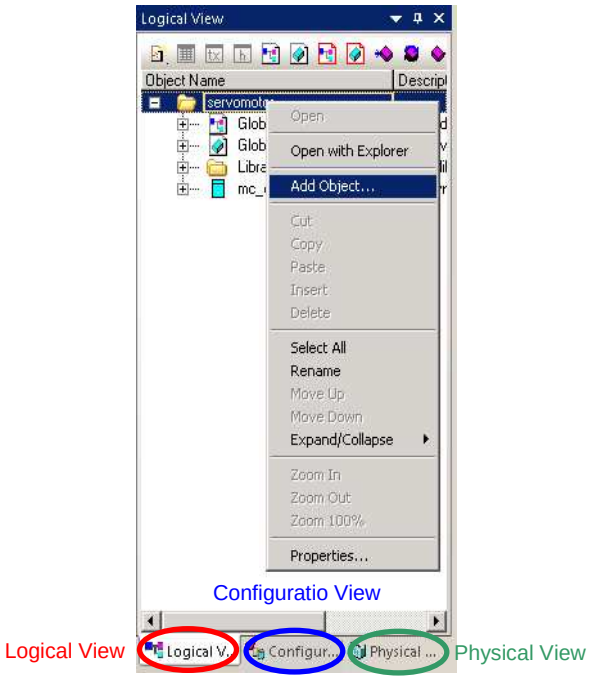


Obrázek 5.4 – Výběr PLC

V posledním okně (obrázek 5.5) se nám objeví rekapitulační tabulka s parametry, které jsme zadali. Pro další nastavení je potřeba zaškrtnout položku pro vložení nového objektu. Pokud tak neučiníme, lze ji vybrat ručně. Po úspěšném vytvoření projektu můžeme vybírat mezi třemi základními okny (obrázek 5.6): Logical View, Configuratio View a Physical View. Pokud vybereme okno Logical View, lze vložit nové objekty ručně, když klikneme pravým tlačítkem na položku Servomotor a zvolíme Add Object (obrázek 5.6).



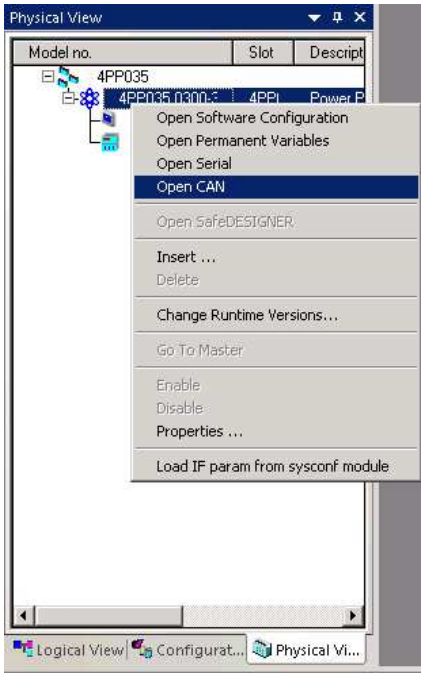
Obrázek 5.5 – Rekapitulační tabulka



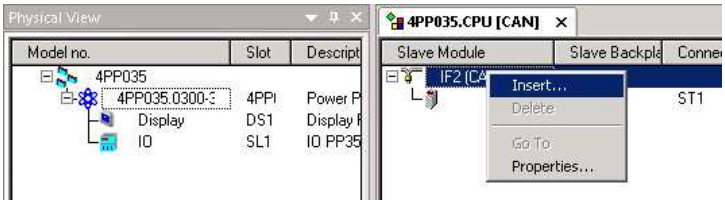
Obrázek 5.6 – Základní okna

5.2.2. HARDWAROVÁ KONFIGURACE

Nyní máme vytvořený projekt s PLC. Dále je potřeba nadefinovat konfiguraci servozesilovače (Acoposu) a k němu připojeného motoru. K tomu je nutné se přepnout do okna Physical View, kliknout pravým tlačítkem na vybraném PLC a dát Open Can (obrázek 5.7). Na pravé straně Automation studia se nám otevře nové okno, kam najedeme myší a klikneme pravým tlačítkem na IF2 (CAN) a vybereme Insert (obrázek 5.8). Tímto spustíme automatického průvodce, který nás provede celou konfigurací.

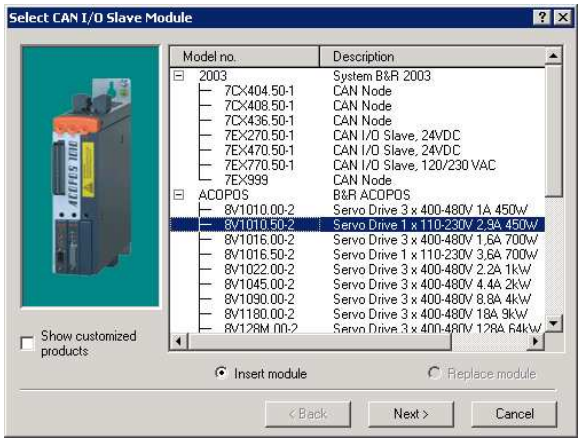


Obrázek 5.7 – Otevření CAN

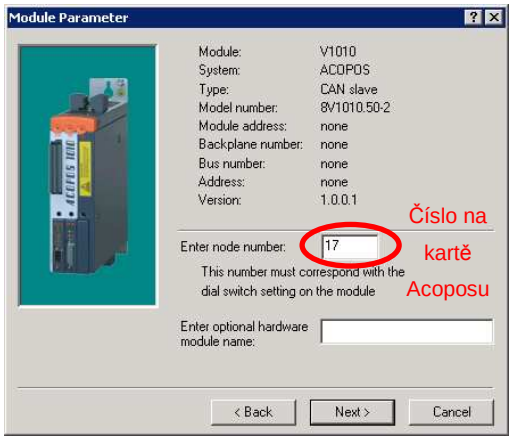


Obrázek 5.8 – Spuštěni průvodce konfigurací

Nejprve je potřeba zvolit typ Acoposu (8V1010.50-2), který je připojen k našemu PLC (obrázek 5.9). V dalším okně (obrázek 5.10) musíme zadat číslo, které je nastavené na přední straně zásuvné karty Acoposu (obrázek 5.11) pomocí přepínačů. Číslo na Acoposu je uvedeno v hexadecimální soustavě, proto je ho potřeba převést do desítkové soustavy. Příklad: $11_{16} = 17_{10}$.



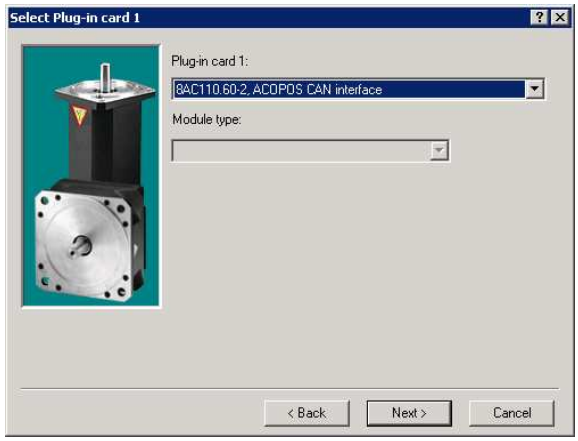
Obrázek 5.9 – Výběr Acoposu



Obrázek 5.10 – Nastavení čísla Acoposu

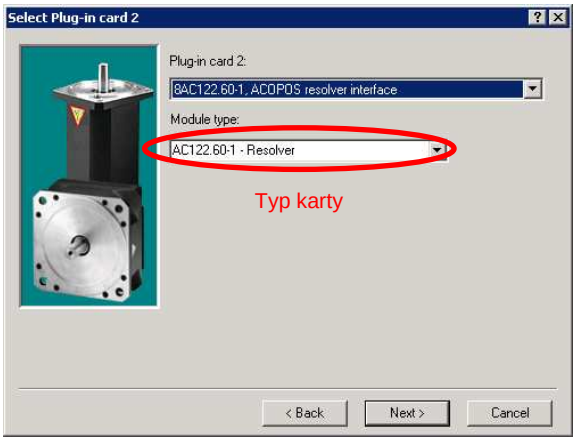
V dalších oknech vybereme typ zásuvných karet připojených v Acoposu. První karta je 8AC110.60-2, Acopos CAN interface (obrázek 5.12) a druhá 8AC122.60-1, Acopos resolver interface (obrázek 5.13), u této karty je ještě potřeba zvolit typ (AC122.60-1 - Resolver).

Obrázek 5.11 – Číslo na kartě Acoposu

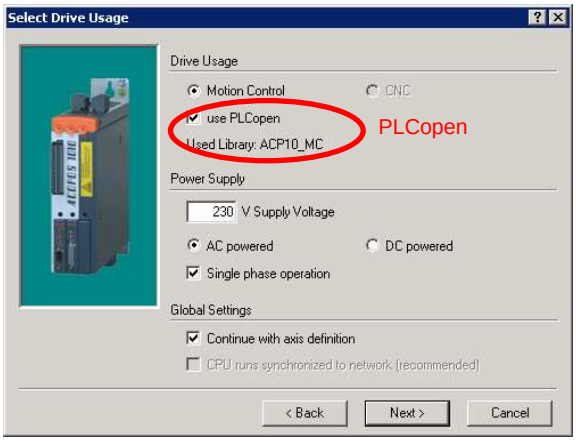


Obrázek 5.12 – Vložení první zásuvné karty

V dalším okně zaškrtneme všechny položky uvedené na obrázku 5.14, hlavně nezapomeneme zatrhnout použití PLCopen, protože se nám automaticky importuje knihovna ACP_MC, pomocí které budeme vytvářet náš program.

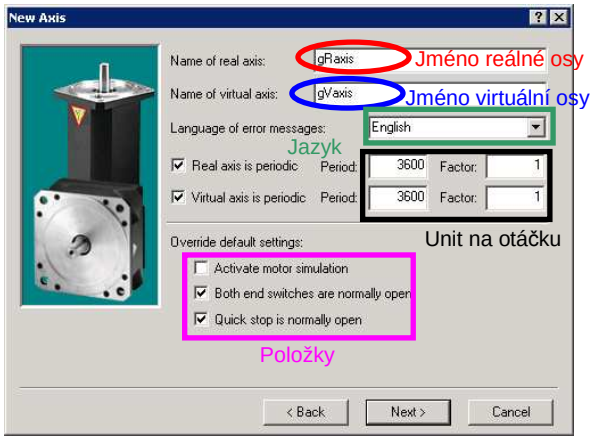


Obrázek 5.13 – Vložení druhé zásuvné karty



Obrázek 5.14 – Knihovna ACP10_MC

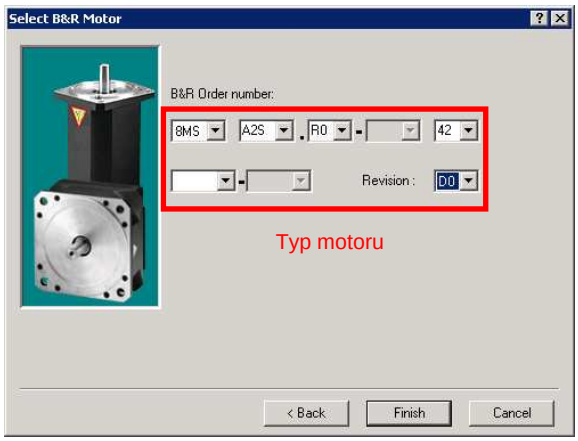
Nyní zadáme jméno reálné (gRaxis) a virtuální (gVaxis) osy, typ jazyka (English) pro výpis chybových hlášek, počet unitů (3600) na jednu otáčku a případně zaškrtnout položky, které chceme mít nastavené (obrázek 5.15). V dalším okně se musí označit použití motoru od B&R (obrázek 5.16).



Obrázek 5.15 – Definice os

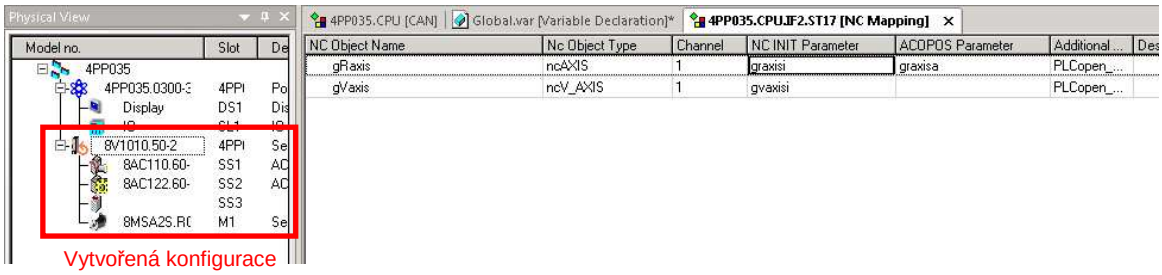
Obrázek 5.16 – Výběr motoru

V posledním okně je nutné zadat typ vašeho použitého motoru. Tyto údaje naleznete na štítku motoru, v našem případě se jedná o typ 8MSA2S.R042 Re D0.



Obrázek 5.17 – Nastavení typu motoru

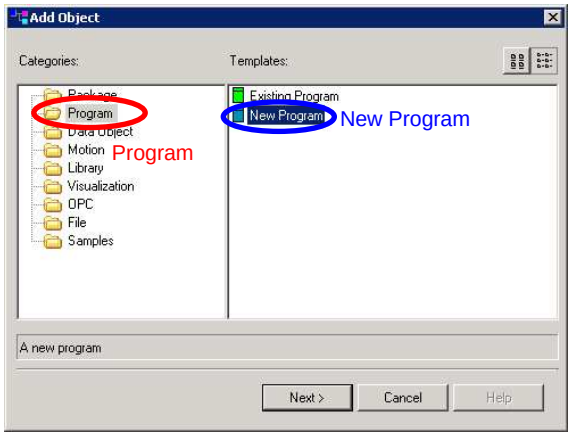
Po dokončení průvodce se v okně Physical View objeví námi vytvořená konfigurace (obrázek 5.18).



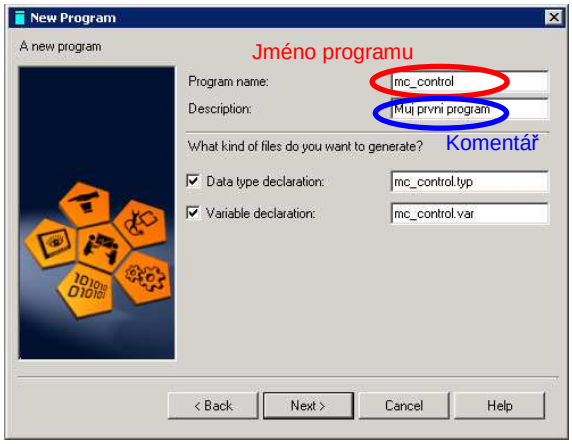
Obrázek 5.18 – Vytvořená konfigurace

5.2.3. VYTVOŘENÍ PROGRAMU

Pro vytvoření nového programu se musí vložit objekt. Vložení objektu je popsané v kapitole 5.2.1 a znázorněné na obrázku 5.6. Po vložení objektu se objeví okno (obrázek 5.19), kde je nutné vybrat kategorii Program a jako šablonu zvolit New Program. V dalším okně (obrázek 5.20) pojmenujeme náš program (př. mc_contol), případně doplníme o krátký komentář (Můj první program).

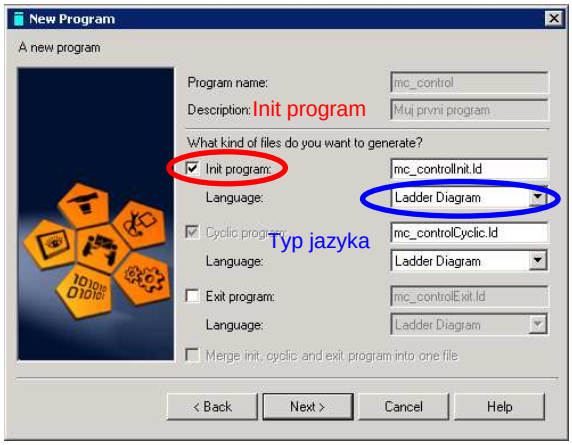


Obrázek 5.19 – Vytvoření programu

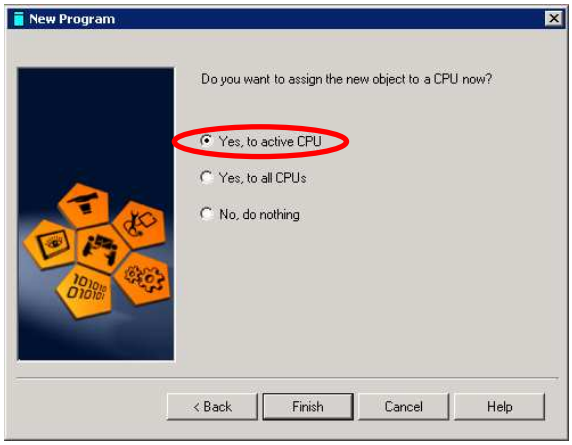


Obrázek 5.20 – Název programu

V dalším okně zaškrtneme položku Init program a vybereme typ jazyka Ladder Diagram (obrázek 5.21). Na konci průvodce označíme, že chceme zařadit náš program do aktivní části CPU (obrázek 5.22).

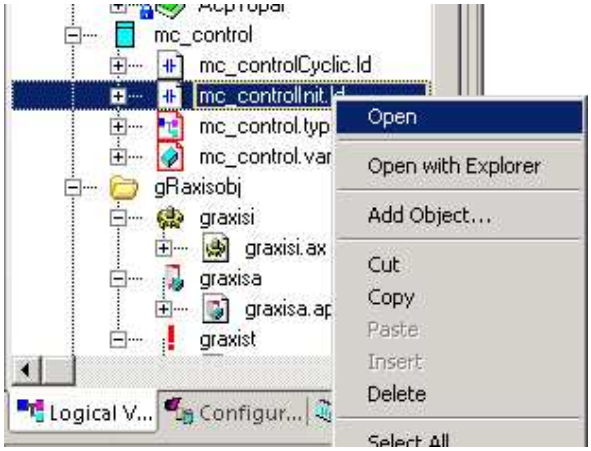


Obrázek 5.21 – Výběr prog. jazyka



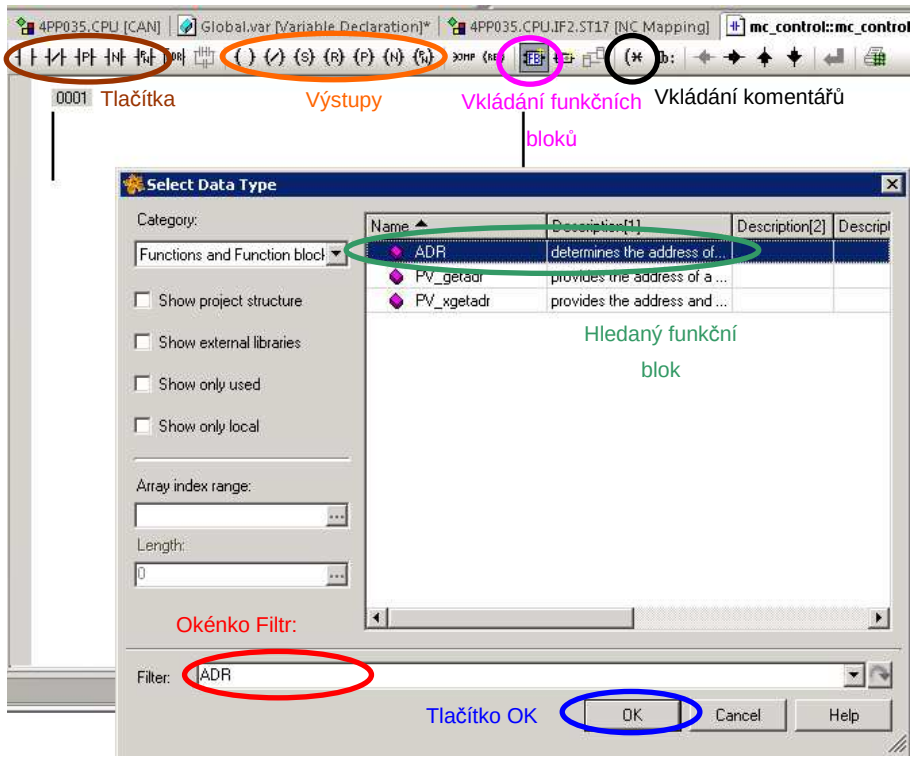
Obrázek 5.22 – Zařazení do CPU

Nyní se nám v okně Logical View objevil náš vytvořený program. Pro další část se musí rozbalit tento program a otevřít položka mc_contolInit.ld. Pro otevření na ni klikneme pravým tlačítkem a zvolíme Open (obrázek 5.23) nebo dvakrát klikneme levým tlačítkem.



Obrázek 5.23 – Otevření programu mc_contolInit.ld

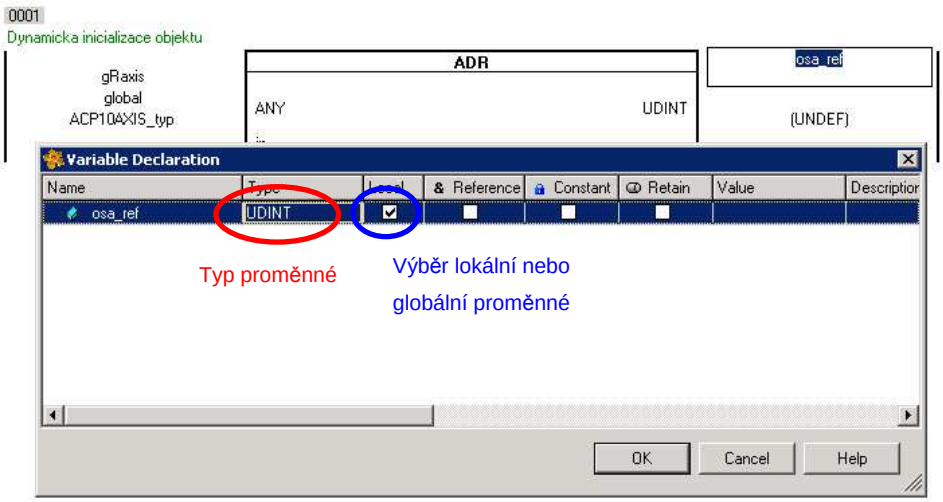
Na pravé straně Automation studia se nám otevře vývojové prostředí pro tvorbu programu. Pro vkládání funkčních bloků klikneme na ikonku FB (obrázek 5.24) a objeví se nabídka, ze které si vybereme náš funkční blok. Pokud chceme odfiltrovat všechny ostatní bloky a vybrat jenom náš hledaný, napíšeme jeho název (ADR) do okénka *Filter*: a po potvrzení se objeví všechny funkční bloky obsahující tento název. Nyní stačí vybrat náš blok a zvolit OK.



Obrázek 5.24 – Vkládání funkčních bloků

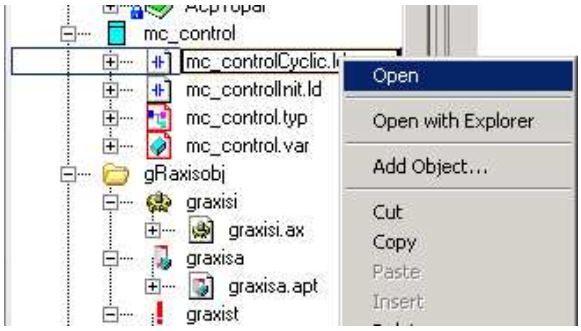
Při deklaraci nových proměnných se objeví tabulka (obrázek 5.25), ve které můžeme vybrat typ proměnné (UDINT), dále jestli chceme lokální nebo globální proměnnou (lokální) a podobně.

V této části programu říkáme, že lokální proměnná „osa_ref“ má adresu reálné osy gRaxis vytvořené při konfiguraci.

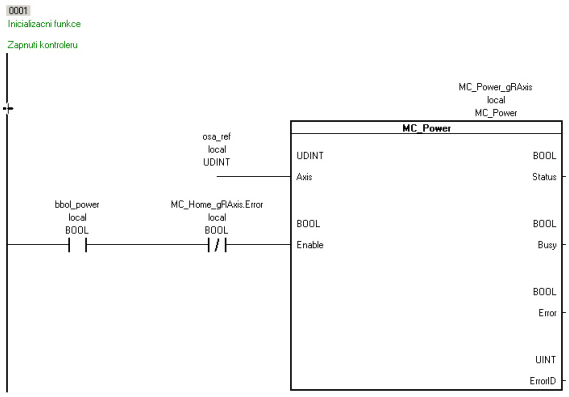


Obrázek 5.25 – Deklarace proměnných

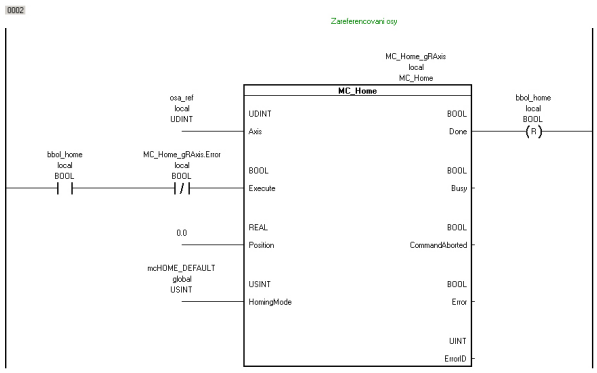
Nyní otevřeme stejným způsobem položku mc_controlCyclic.ld (obrázek 5.26), kde bude celý náš program. Do tohoto programu stejným způsobem jako předtím vložíme tyto funkční bloky: MC_Power, MC_Home, MC_MoveVelocity, MC_MoveAbsolute, MC_Halt, MC_Stop, MC_ReadStatus, MC_ReadActualPosition, MC_ReadAxisError a MC_Reset s příslušnými ovládacími tlačítky. Všechny tyto bloky spolu s tlačítky jsou na obrázcích 5.27 až 5.36.



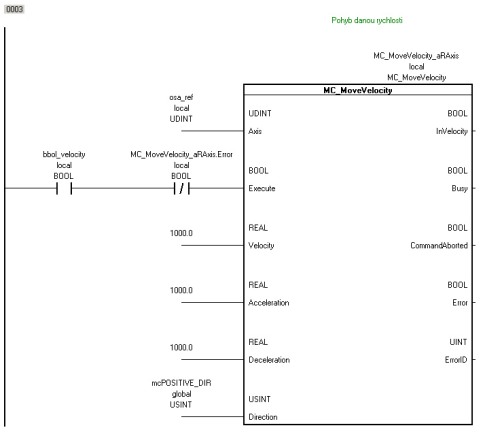
Obrázek 5.26 – Otevření programu mc_controlCyclic.ld



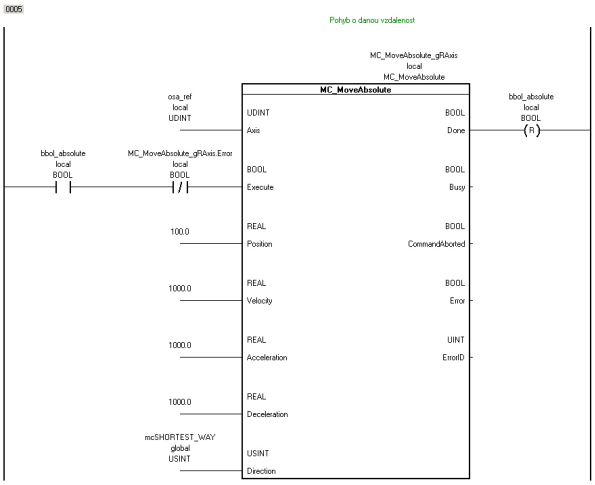
Obrázek 5.27 – Funkční blok MC_Power



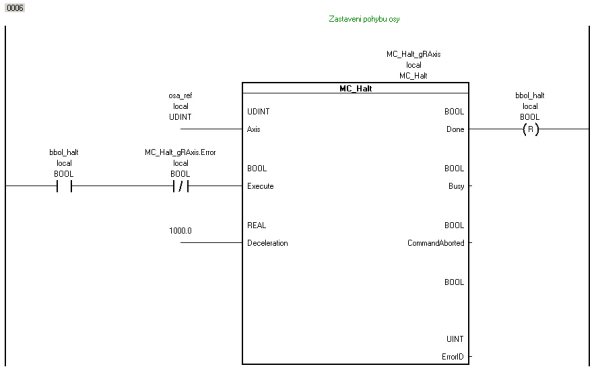
Obrázek 5.28 – Funkční blok MC_Home



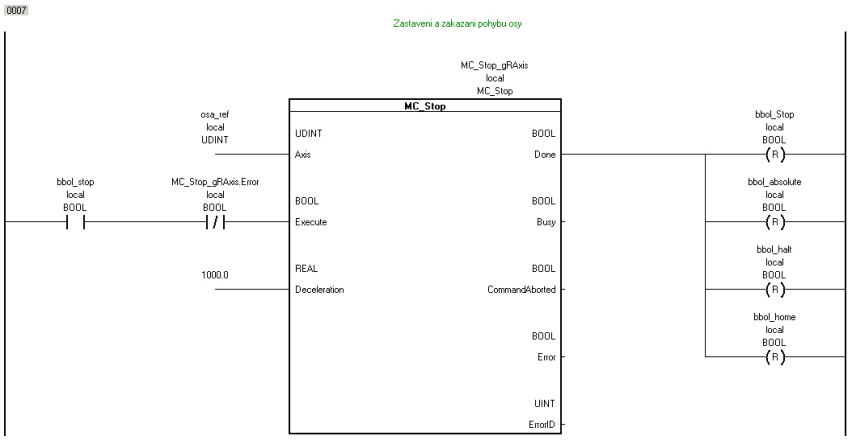
Obrázek 5.29 – Funkční blok MC_MoveVelocity



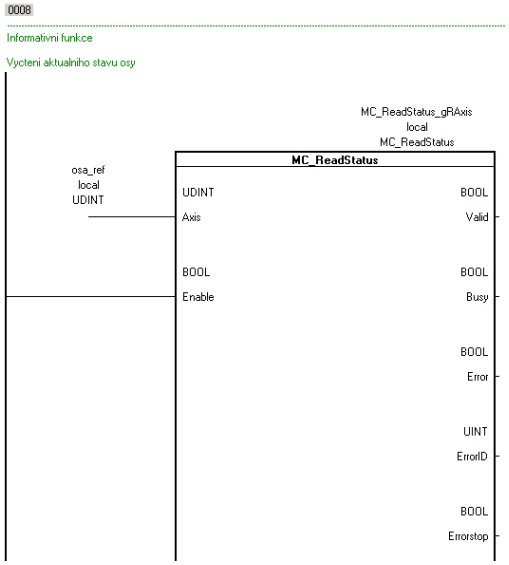
Obrázek 5.30 – Funkční blok MC_MoveAbsolute



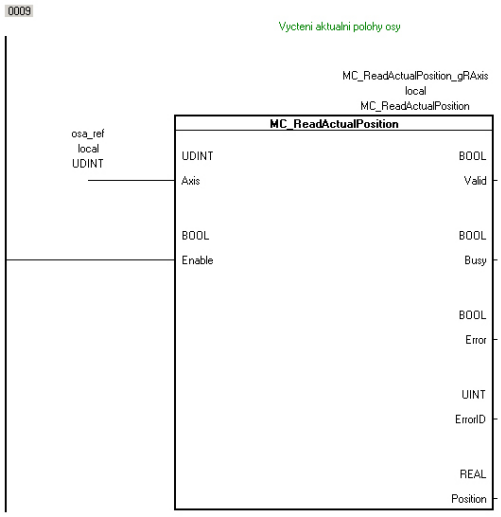
Obrázek 5.31 – Funkční blok MC_Halt



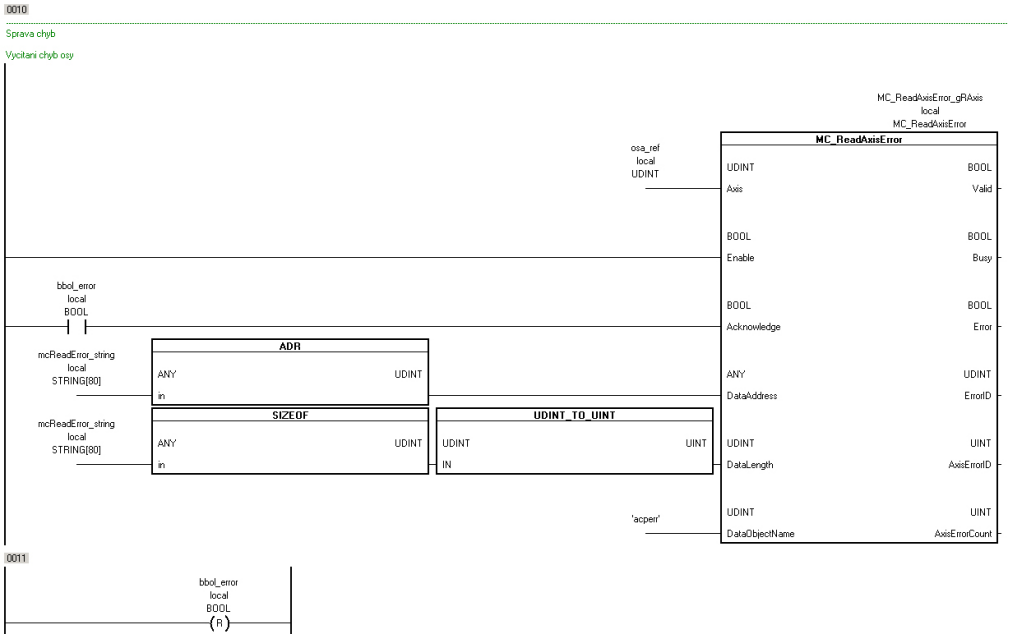
Obrázek 5.32 – Funkční blok MC_Stop



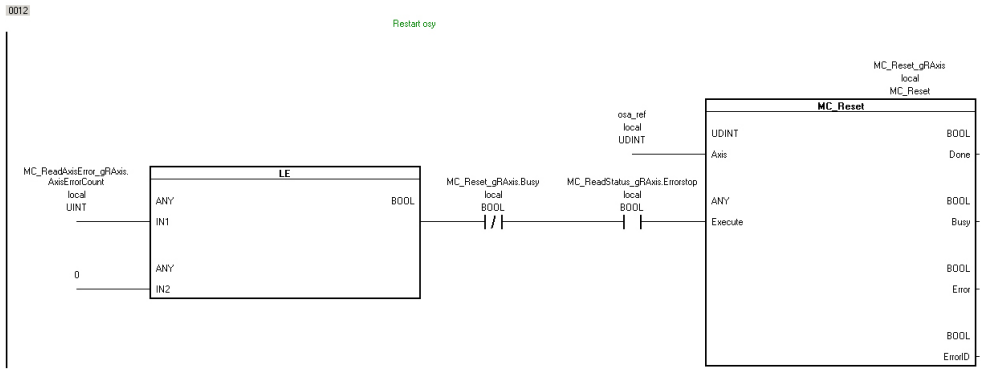
Obrázek 5.33 – Funkční blok MC_ReadStatus



Obrázek 5.34 – Funkční blok MC_ReadActualPosition



Obrázek 5.35 – Funkční blok MC_ReadAxisError

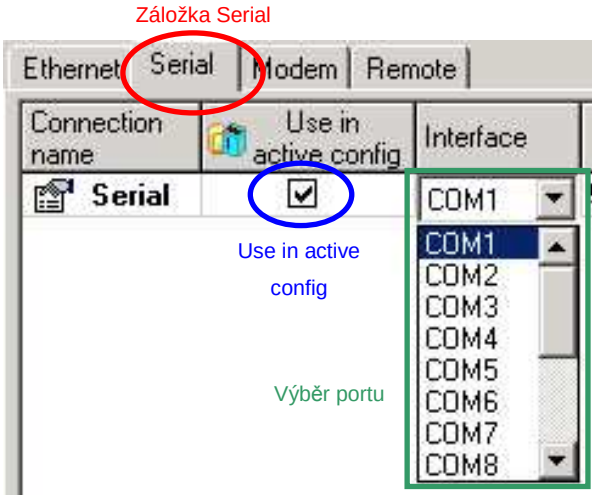


Obrázek 5.36 – Funkční blok MC_Reset

5.2.4. NASTAVENÍ KOMUNIKACE

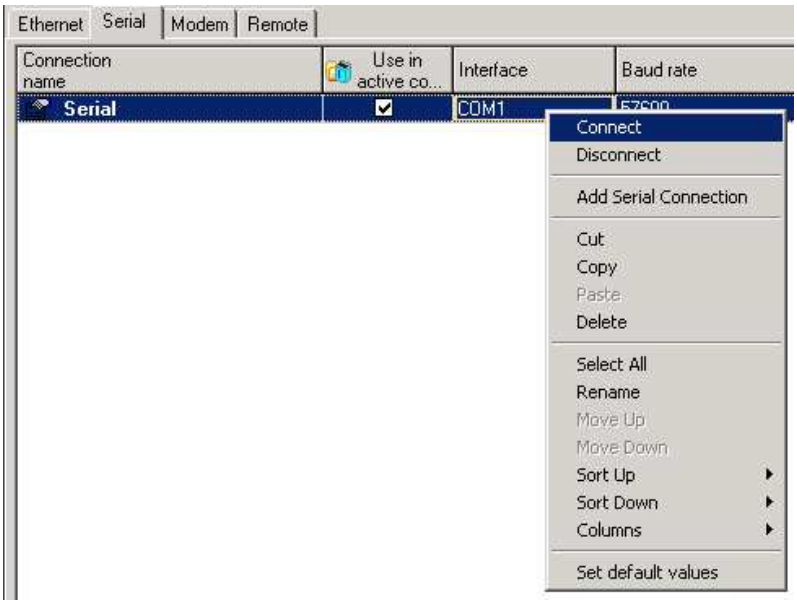
Pro navázání komunikace s PLC klikněte v horní nabídce Automation studia na záložku Online a vyberte Settings… (obrázek 5.37). V pravém okně vyberte záložku Serial, zatrhněte Use in active config a poté se vyberte sériový port RS232 na vašem počítači propojený s PLC (obrázek 5.38).

Obrázek 5.37 – Nastavení komunikace



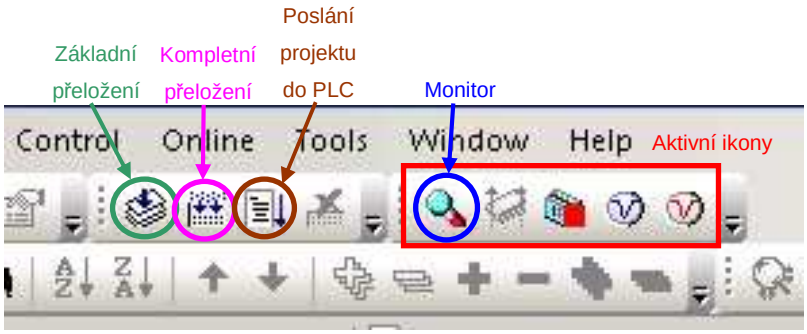
Obrázek 5.38 – Volba portu

Nyní stačí na tuto konfiguraci kliknout pravým tlačítkem a vybrat Connect (obrázek 5.39).



Obrázek 5.39 – Navázání komunikace s PLC

Pokud se připojení k PLC podařilo, budou aktivní ikony na obrázku 5.40 a v dolní části Automation studia bude RUN. Jestliže se vám komunikace nepodařila, zkontrolujte propojení sériového portu na vašem počítači s PLC, zapnutí PLC do elektrické sítě nebo zda máte správně vybraný typ portu.



Obrázek 5.40 – Komunikační ikony s PLC

Po navázání komunikace stačí projekt přeložit a poslat do PLC (obrázek 5.40). Pro spuštění vašeho programu klikněte na ikonku monitor, otevřete váš program a nastavte tlačítko, které máte u bloku MC_Power, do logické jedničky. Stejným postupem zapněte blok MC_Home a poté již můžete pohybovat s motorem pomocí bloků MC_MoveVelocity nebo MC_MoveAbsolute.

Pokud nastane nějaká chyba, lze její číslo vyčíst z příslušného bloku a najít si její specifikaci v nápovědě, která je součásti Automation studia. Tuto chybu potvrdíte tlačítkem u bloku MC_ReadAxisError. Pokud nastanou jiné chyby, obraťte se prosím na vašeho vyučujícího, případně použijte nápovědu Automation studia.

6. ZÁVĚR

Celý program a jeho rozšíření o zpětnou vazbu a funkce: *nahoď kouli*, *zahoď kouli* a žonglování se *třemi koulemi*, se podařilo realizovat v jazyce Ladder Diagram. Tento jazyk je pro pochopení a programování velmi jednoduchý a umožňuje zásah do programu od obsluhy, která nemusí být seznámena s vyšší formou programovacích jazyků, a je zároveň zcela dostačující pro tvorbu složitějších aplikací, jako je model žongléra. Program byl navržen tak, aby byla zajištěna nejjednodušší obsluha zařízení uživatelem, což u minulého programu bylo složité. Také byla kompletně předělána celá vizualizace a doplněna o nové funkce, kterými jsou trendy, alarmy a mnoho jiných. Při tvorbě nového programu byly znovu navrženy a vypočteny všechny kinematické pohyby při žonglování a vloženy do vačkových profilů, aby bylo možné nově žonglovat i s tenisovými míčky. Návrh všech vačkových profilů pro zadané funkce je obsažen v příloze.

Program byl nadále rozšířen o pátý lineární motor s připevněnou miskou pro nahazování spadlých koulí a dále o zásobník, který tvoří podavač se servomotorky a optickými branami pro umístění spadlých koulí na tuto misku s lineárním motorem. Pro tento účel byly také nově navrženy a vypočteny kinematické pohyby pro nahození a zahození koule. Tyto trajektorie byly úspěšně realizovány vačkovými profily i pro tenisový míček, který je několikrát lehčí než kulečníková koule. Velký problém, který se zde vyskytl, byl ten, že lineární motor po zahřátí vlivem teplotní roztažnosti kovů nebyl schopen opakovaně nahazovat kouli do stejného místa. Tento nedostatek je možné v budoucnu vyřešit novým vodícím vedením případně výměnou celého motoru za motor lepší.

Do programu byla také implementována zpětná vazba pro detekci koulí na přehazovacích ramínkách, a to dvojího typu. První zpětná vazba byla realizována pomocí rychloběžné kamery a druhá pomocí krouticího momentu měřeného na motorech vykonávající horizontální pohyb. Nevýhodou krouticího momentu bylo, že ho není možné použít pro lehké předměty, které zanedbatelně působí svoji setrvačnou silou na motory, zatímco kamera je schopna detekovat jakýkoliv předmět. V našem případě se jednalo o černobílou kameru, takže její snímaní bylo rovněž omezené. Detekování koule se provádí před procesem žonglování, kdy se zjistí, zda

je potřeba nahodit kouli na přehazovací ramínko, a při žonglování se třemi koulemi, zda nedošlo ke spadnutí jedné ze tří koulí.

Pro výuku na jednodušším modelu tvořeném pouze jedním servomorem byl vytvořen návod pro studenty, zahrnující založení nového projektu s vytvořením jednoduchého programu. Pro práci na modelu žongléra byly vytvořeny flashové animace, obsahující celý postup založení nového programu i s konfigurací. Tento materiál je součástí přiloženého CD.

V závěru práce jsou obsaženy přílohy, na které se odkazují jednotlivé kapitoly, a je zde také časové vyobrazení všech vačkových profilů pro jednotlivé funkce žonglování.

SEZNAM LITERATURY

[1] B&R AUTOMATION, *CAM Profile Automat ACOPOS Firmware*, 2004.

[2] HARDWAROVÁ DOKUMENTACE – ZÁSOBNÍK - KOULE ZP 2008.

[3] PŠENIČKA JAN, *Synchronizace servopohonů*, KATEDRA ŘÍDÍCÍ TECHNIKY ČVUT PRAHA, DIPLOMOVÁ PRÁCE, 2005.

[4] KOHOUT TOMÁŠ, *Vizuální kontrola chodu stroje*, KATEDRA ŘÍDÍCÍ TECHNIKY ČVUT PRAHA, BAKALÁŘSKÁ PRÁCE 2009.

[5] JAROŠ PAVEL, *Dynamika rychlých servopohonů*, KATEDRA ŘÍDÍCÍ TECHNIKY ČVUT PRAHA, BAKALÁŘSKÁ PRÁCE 2009.

[6] TECHNICAL DESCRIPTION, KATEDRA ŘÍDÍCÍ TECHNIKY – JUGGLER
⟨ http://dce.fel.cvut.cz/juggler/technical-description/technical-description.aspx ⟩

[7] OTEVŘENÁ ENCYKLODEDIE WIKIPEDIE,
⟨ http://CS.WIKIPEDIA.ORG ⟩, 2010.

[8] NÁPOVĚDA AUTOMATION STUDIA

SEZNAM OBRÁZKŮ

Obrázek 1 – Model žongléra 1
Obrázek 2.1 – Vertikální synchronní motor 4
Obrázek 2.2 – Horizontální synchronní motory + lineární motor 5
Obrázek 2.3 – Acopos 5
Obrázek 2.4 – Power panel 4PP420 5
Obrázek 2.5 – Struktura řízení a komunikace modelu 6
Obrázek 2.6 – Vykonávání programu 7
Obrázek 2.7 – Funkční blok „MC_Power“ v jazyce C 9
Obrázek 2.8 – Funkční blok „MC_Power“ v jazyce Ladder Diagram 9
Obrázek 2.9 – Stavový diagram 10
Obrázek 2.10 – Funkční blok „MC_Home“ 11
Obrázek 2.11 – Funkční blok „MC_MoveAbsolute“ 11
Obrázek 2.12 – Funkční blok „MC_MoveVelocity“ 11
Obrázek 2.13 – Funkční blok „MC_Stop“ 11
Obrázek 2.14 – Funkční blok „MC_CamTableSelect“ 12
Obrázek 2.15 – Funkční blok „MC_CamIn“ 12
Obrázek 2.16 – Diagnostika - stará 12
Obrázek 2.17 – Diagnostika - nová 12
Obrázek 2.18 – Funkce *1-2 koulí* - stará 13
Obrázek 2.19 – Funkce *1-2 koulí* - nová 13
Obrázek 2.20 – Funkce *3 koulí* 13
Obrázek 2.21 – Funkce *nahoď* a *zahoď kouli* 13
Obrázek 2.22 – Alarmy 14
Obrázek 2.23 – Trendy 14
Obrázek 2.24 – Kinematika vrhu svislého vzhůru 16
Obrázek 2.25 – Kinematika vrhu šikmého vzhůru 17
Obrázek 2.26 – Rozměrová identifikace přehazovacích ramínek 19
Obrázek 2.27 – Rozměry modelu žongléra 20
Obrázek 2.28 – Poloha ramínka při přehozu koule 23
Obrázek 2.29 – Použitá převodovka s převodem 3:1 27
Obrázek 2.30 – Teplotní model žongléra s použitím převodovek 28

Obrázek 2.31 – Vačkový profil v Automation studiu 29
Obrázek 2.32 – Návrh vačky pro dvě koule – horizontální pohyb (motor 1 a 3) 31
Obrázek 2.33 – Návrh vačky pro dvě koule – vertikální pohyb (motor 2) 32
Obrázek 3.1 – Lineární motor typ PRC 25 34
Obrázek 3.2 – Řešení misky a vedení lineárního motoru.....35
Obrázek 3.3 – Dopadová plocha zásobníku.....36
Obrázek 3.4 – Zásobníkový tunel.....36
Obrázek 3.5 – Princip mechanické realizace zásobníku 37
Obrázek 4.1 – Detekce koule pomocí kamery.....41
Obrázek 4.2 – Detekce koule při inicializaci pomocí kamery.....42
Obrázek 4.3 – Detekce koule v režimu 3. koulí pomocí kamery.....42
Obrázek 4.4 – Komunikace přes PVI Manager 43
Obrázek 4.5 – Komunikace s kamerou – posílání proměnných 44
Obrázek 4.6 – Srovnáni krouticího momentu ve výchozí pozici 45
Obrázek 4.7 – Zjištění krouticího momentu pro počáteční inicializaci 46
Obrázek 4.8 – Zpracování kroutícího momentu součtovým blokem.....46
Obrázek 4.9 – Srovnáni krouticího momentu v režimu *tří koulí* 48
Obrázek 4.10 – Zpracování kroutícího momentu součtovým blokem.....48
Obrázek 5.1 – Založení projektu 51
Obrázek 5.2 – Jméno projektu 51
Obrázek 5.3 – Jméno konfigurace.....51
Obrázek 5.4 – Výběr PLC 51
Obrázek 5.5 – Rekapitulační tabulka.....52
Obrázek 5.6 – Základní okna 52
Obrázek 5.7 – Otevření CAN.....53
Obrázek 5.8 – Spuštěni průvodce konfigurací.....53
Obrázek 5.9 – Výběr Acoposu.....53
Obrázek 5.10 – Nastavení čísla Acoposu 53
Obrázek 5.11 – Číslo na kartě Acoposu.....54
Obrázek 5.12 – Vložení první zásuvné karty.....54
Obrázek 5.13 – Vložení druhé zásuvné karty.....54
Obrázek 5.14 – Knihovna ACP10_MC 54

Obrázek 5.15 – Definice os 55
Obrázek 5.16 – Výběr motoru 55
Obrázek 5.17 – Nastavení typu motoru 55
Obrázek 5.18 – Vytvořená konfigurace 56
Obrázek 5.19 – Vytvoření programu 56
Obrázek 5.20 – Název programu 56
Obrázek 5.21 – Výběr prog. jazyka 57
Obrázek 5.22 – Zařazení do CPU 57
Obrázek 5.23 – Otevření programu mc_contolInit.Id 57
Obrázek 5.24 – Vkládání funkčních bloků 58
Obrázek 5.25 – Deklarace proměnných 58
Obrázek 5.26 – Otevření programu mc_controlCyclic.Id 59
Obrázek 5.27 – Funkční blok MC_Power 59
Obrázek 5.28 – Funkční blok MC_Home 59
Obrázek 5.29 – Funkční blok MC_MoveVelocity 59
Obrázek 5.30 – Funkční blok MC_MoveAbsolute 60
Obrázek 5.31 – Funkční blok MC_Halt 60
Obrázek 5.32 – Funkční blok MC_Stop 60
Obrázek 5.33 – Funkční blok MC_ReadStatus 61
Obrázek 5.34 – Funkční blok MC_ReadActualPosition 61
Obrázek 5.35 – Funkční blok MC_ReadAxisError 61
Obrázek 5.36 – Funkční blok MC_Reset 62
Obrázek 5.37 – Nastavení komunikace 62
Obrázek 5.38 – Volba portu 62
Obrázek 5.39 – Navázání komunikace s PLC 63
Obrázek 5.40 – Komunikační ikony s PLC 63

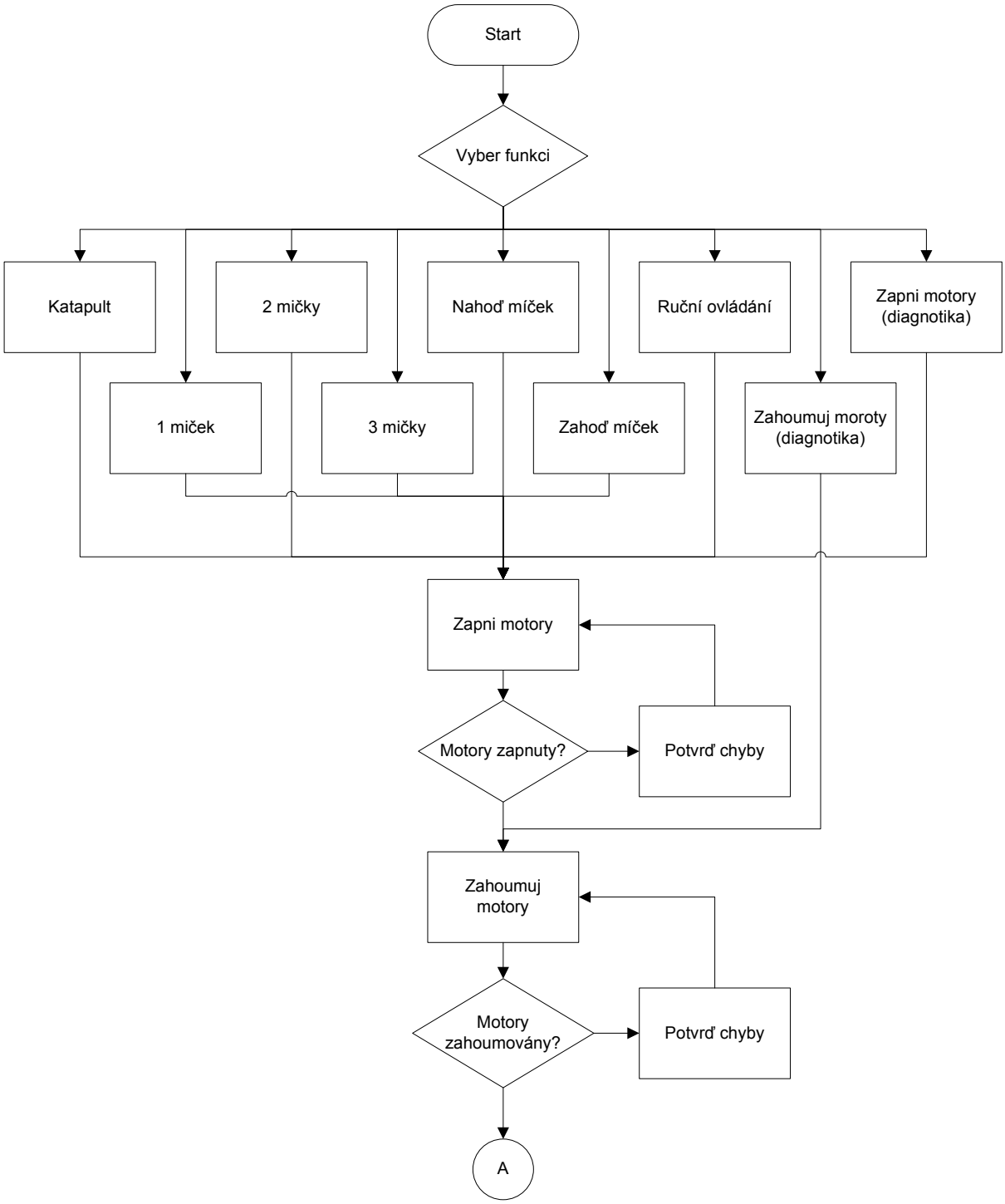
SEZNAM TABULEK

Tabulka 2.1 – Konstanty použité v rovnicích 15
Tabulka 2.2 – Definice fixních bodů do vačko. editoru (2 kou. – H – s1 a s3) 31
Tabulka 2.3 – Definice fixních bodů do vačko. editoru (2 kou. – V – s2) 32
Tabulka 4.1 – Celková velikost krouticího momentu při inicializaci 47
Tabulka 4.2 – Celková velikost krouticího momentu při inicializaci 49

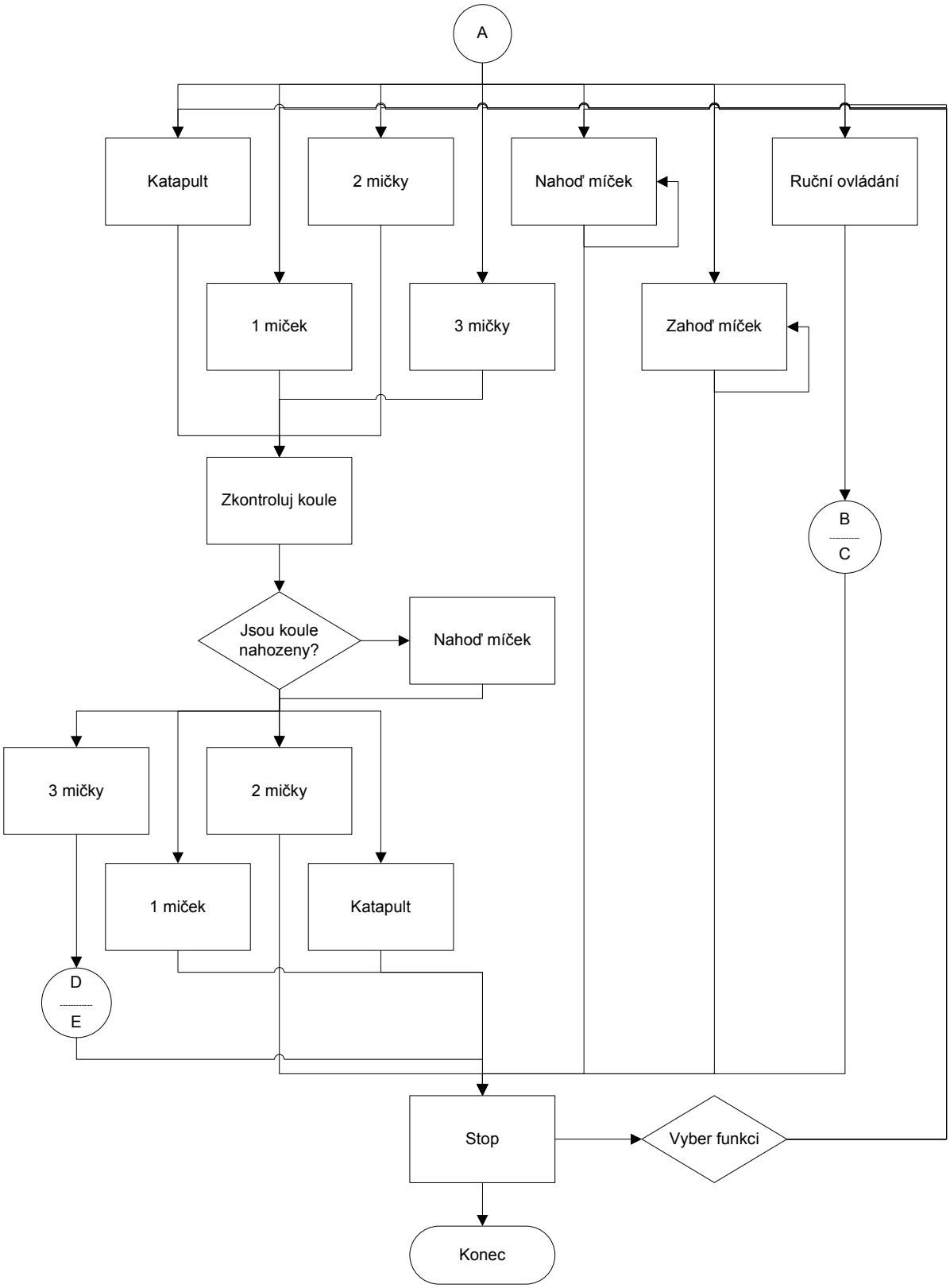
SEZNAM PŘÍLOH

Příloha č.1 – Provozní režimy.....72
Příloha č.2 – Skládání vaček.....75
Příloha č.3 – Vizualizace 77
Příloha č.4 – Vačkový profil - funkce katapult.....79
Příloha č.5 – Vačkový profil - funkce jedné koule 81
Příloha č.6 – Vačkový profil - funkce dvou koulí 85
Příloha č.7 – Vačkový profil - funkce nahoď kouli.....88
Příloha č.8 – Vačkový profil - funkce zahoď kouli 91
Příloha č.9 – Posloupnost vaček 93
Příloha č.10 – Přiložené CD 94

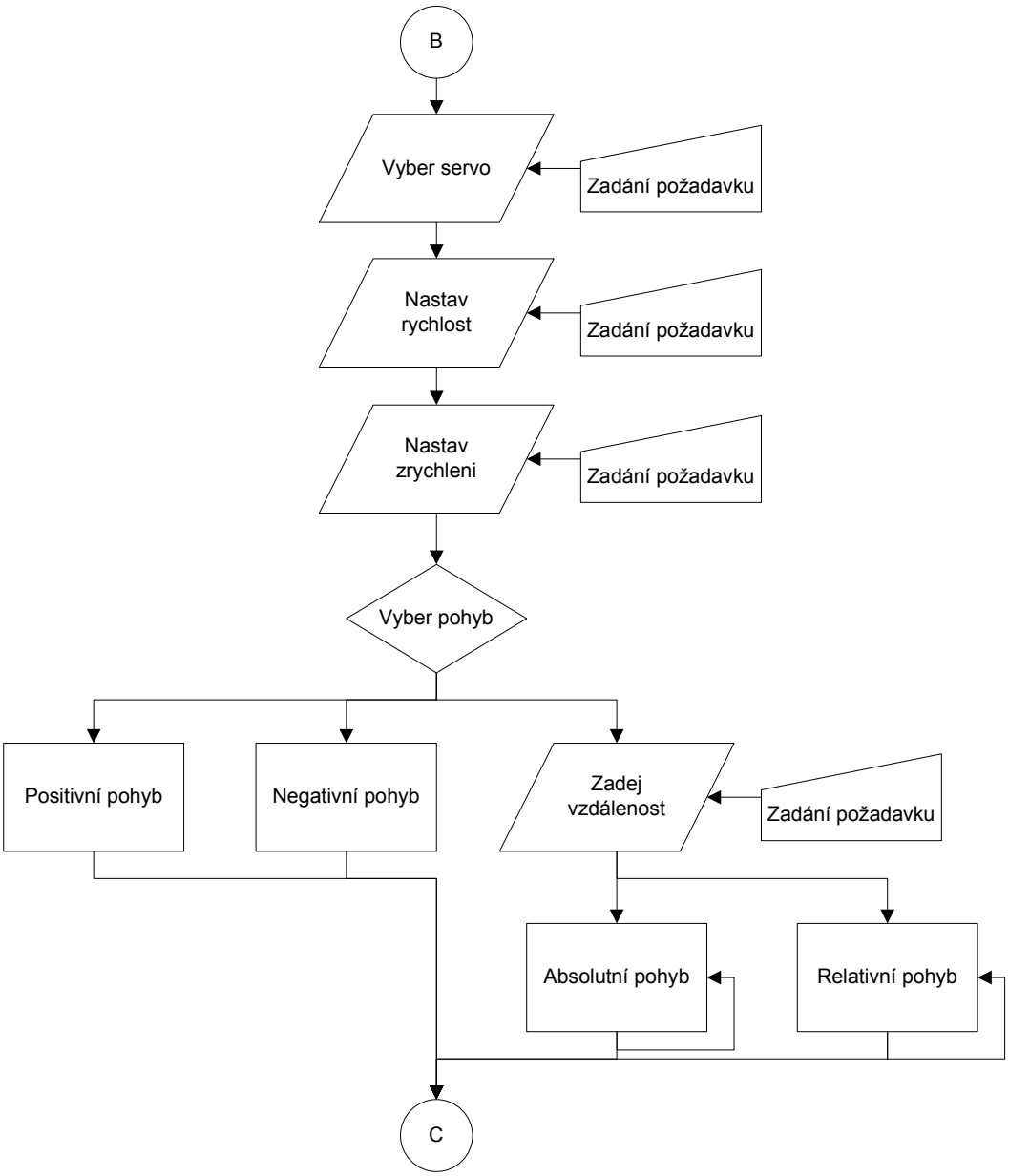
Příloha č.1 – Provozní režimy



Zapnutí a inicializace motorů

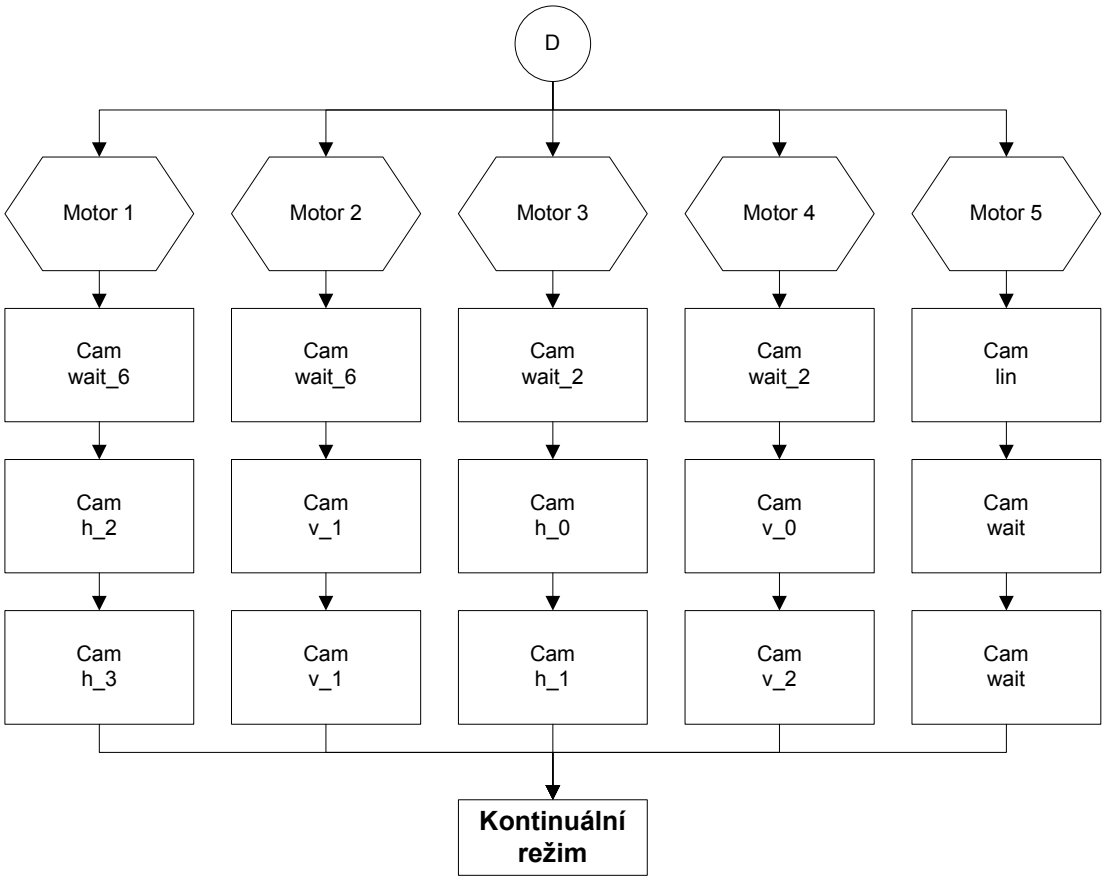


Jednotlivé režimy modelu žonglér

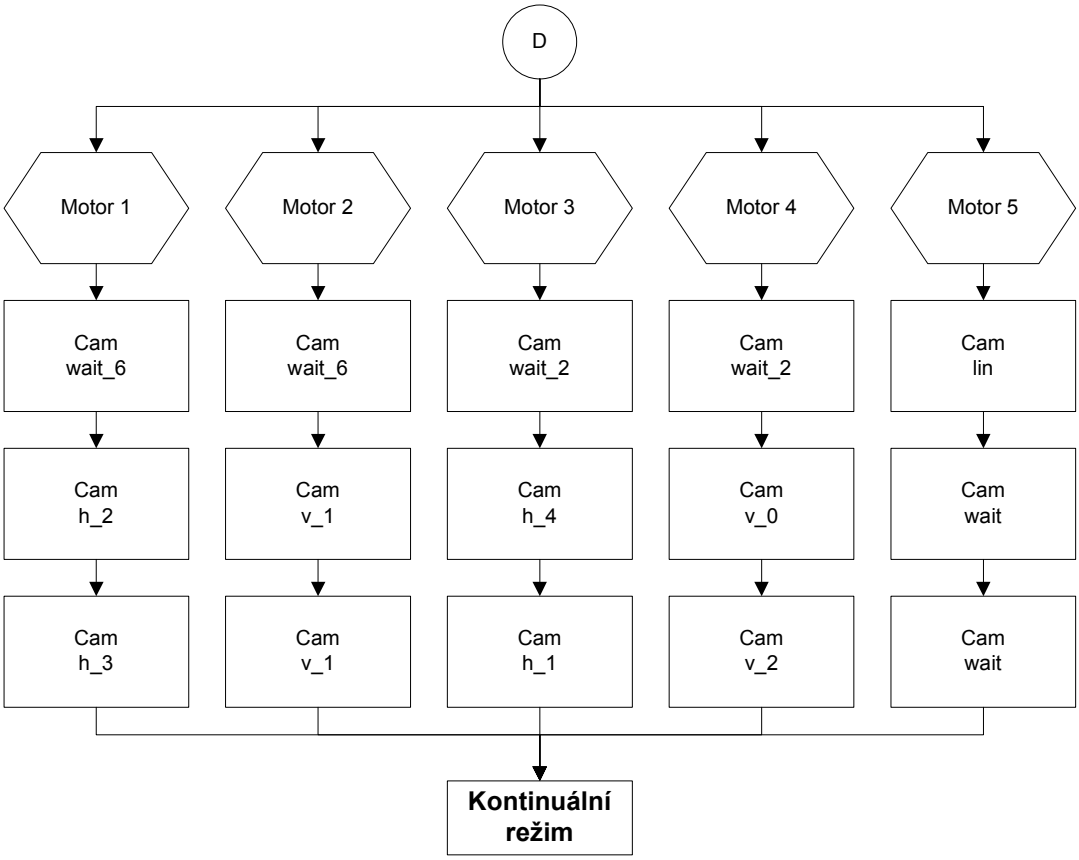


Ruční ovládání

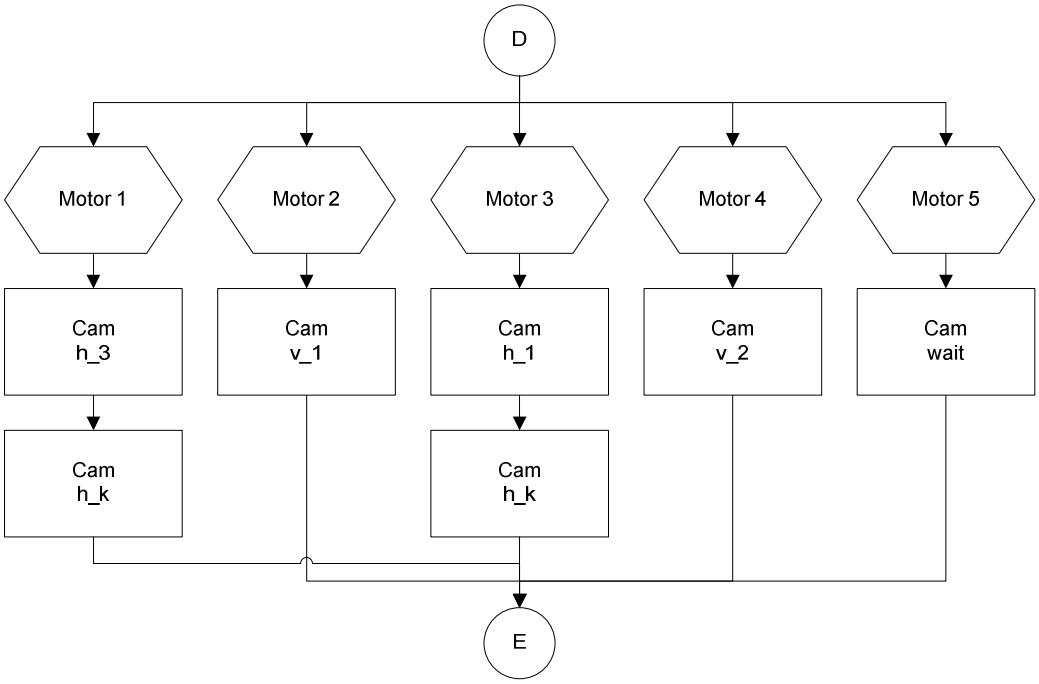
Příloha č.2 – Skládání vaček



Prvotní nahození třetí koule



Nahození spadlé koule za chodu

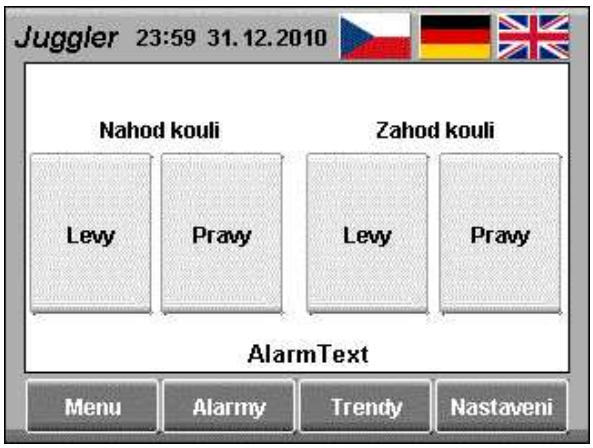


Bezpečné ukončení chodu všech motorů

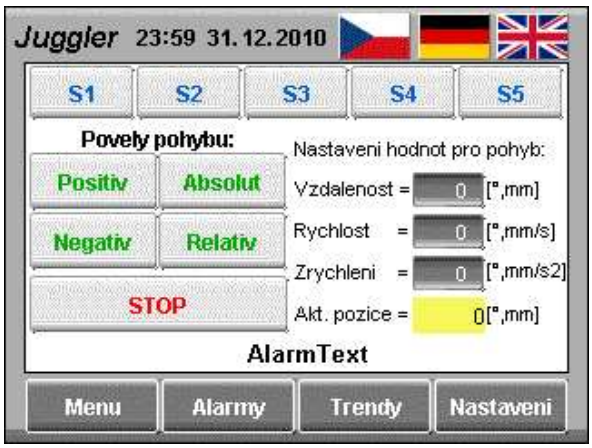
Příloha č.3 – Vizualizace

Startovací obrazovka

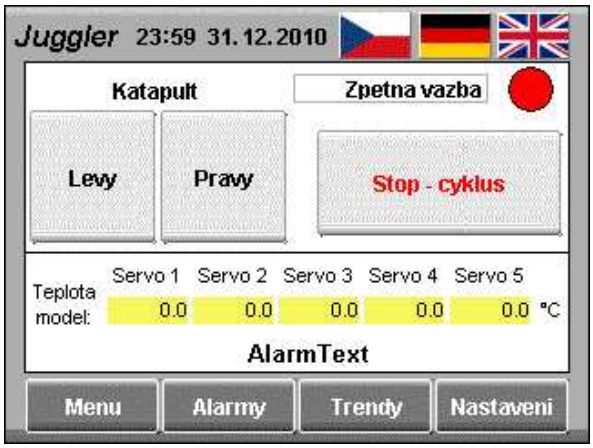
Diagnostika



Nastaveni míčků



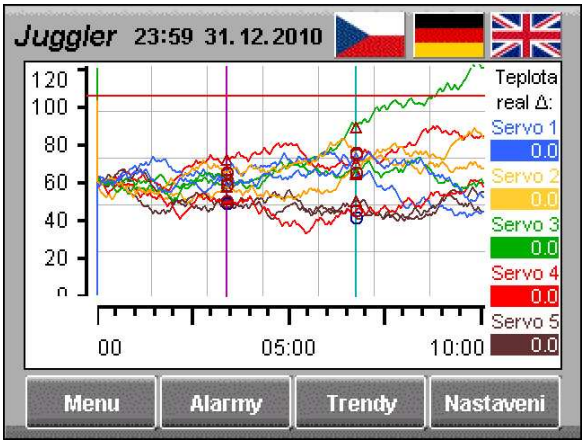
Ruční ovládání



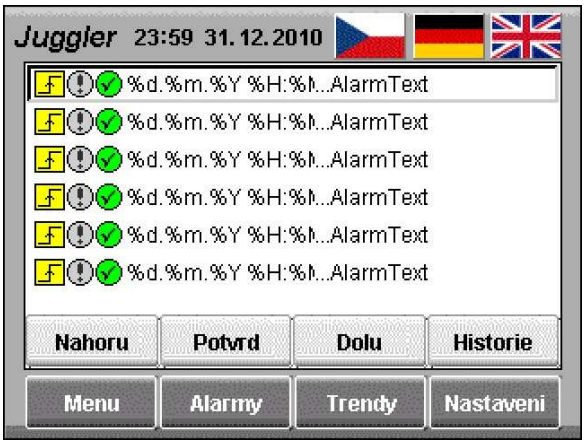
Funkce *katapult*

Funkce *1-2 koulí*

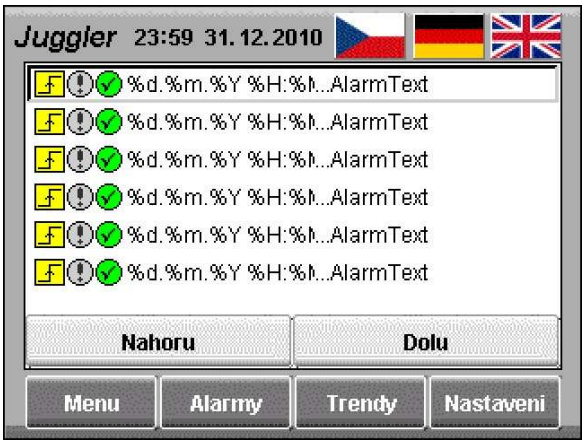
Funkce *3 koulí*



Trendy



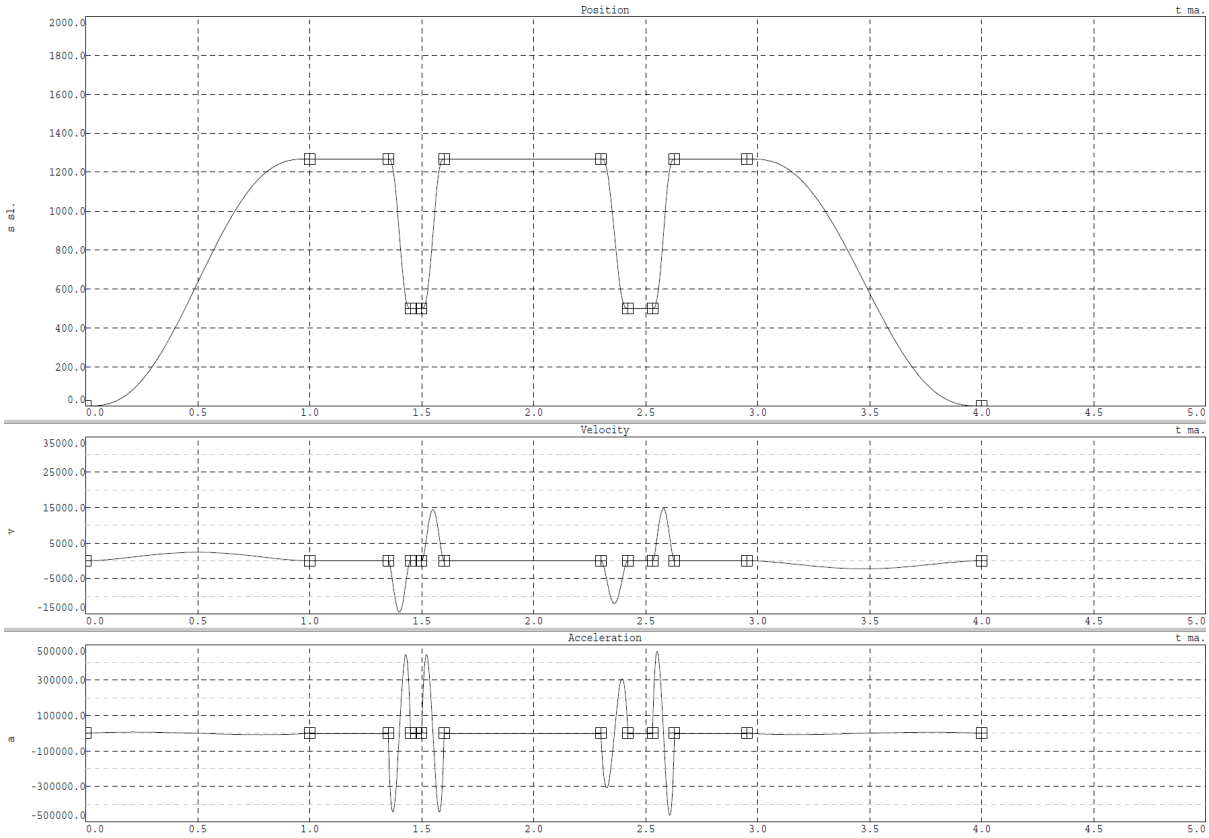
Alarmy



Alarmy - historie

Nastavení

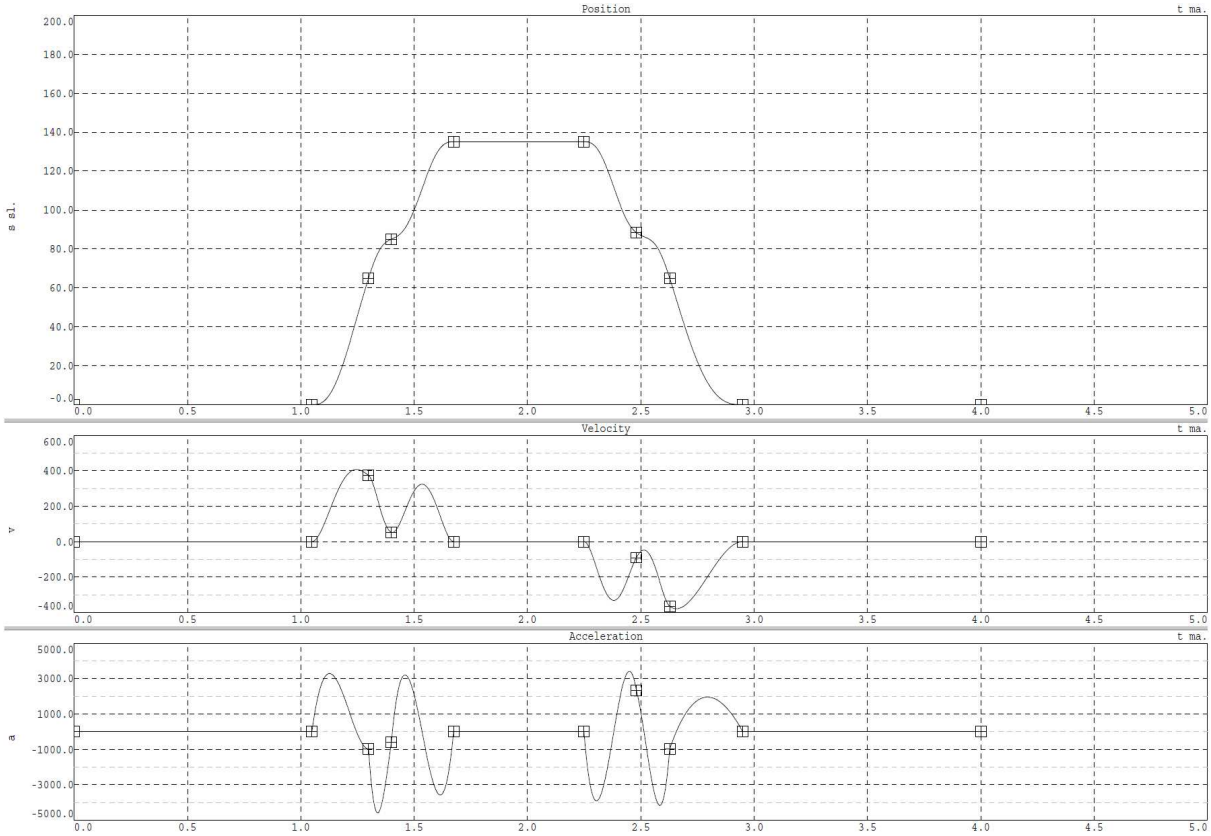
Příloha č.4 – Vačkový profil - funkce katapult



Návrh vačky pro katapult – horizontální pohyb (motor 1 a 3)

Fixní bod					Lineární část (aproximace)	
č.f.	čas	poloha	rychlost	zrychlení	čas	poloha
	[s]	[unit]	[unit/s]	[unit/s^2]	[s]	[unit]
1	0,000	0,000	-	-	1,600	1268,000
2	1,000	1268,000	-	-	2,300	1268,000
3	1,350	1268,000	-	-		
4	1,450	500,000	-	-		
5	1,500	500,000	-	-		
6	2,420	500,000	-	-		
7	2,530	500,000	-	-		
8	2,628	1268,000	-	-		
9	2,950	1268,000	-	-		
10	4,000	0,000	-	-		

Definice fixních bodů do vačko. editoru (kat. – H – s1 a s2)

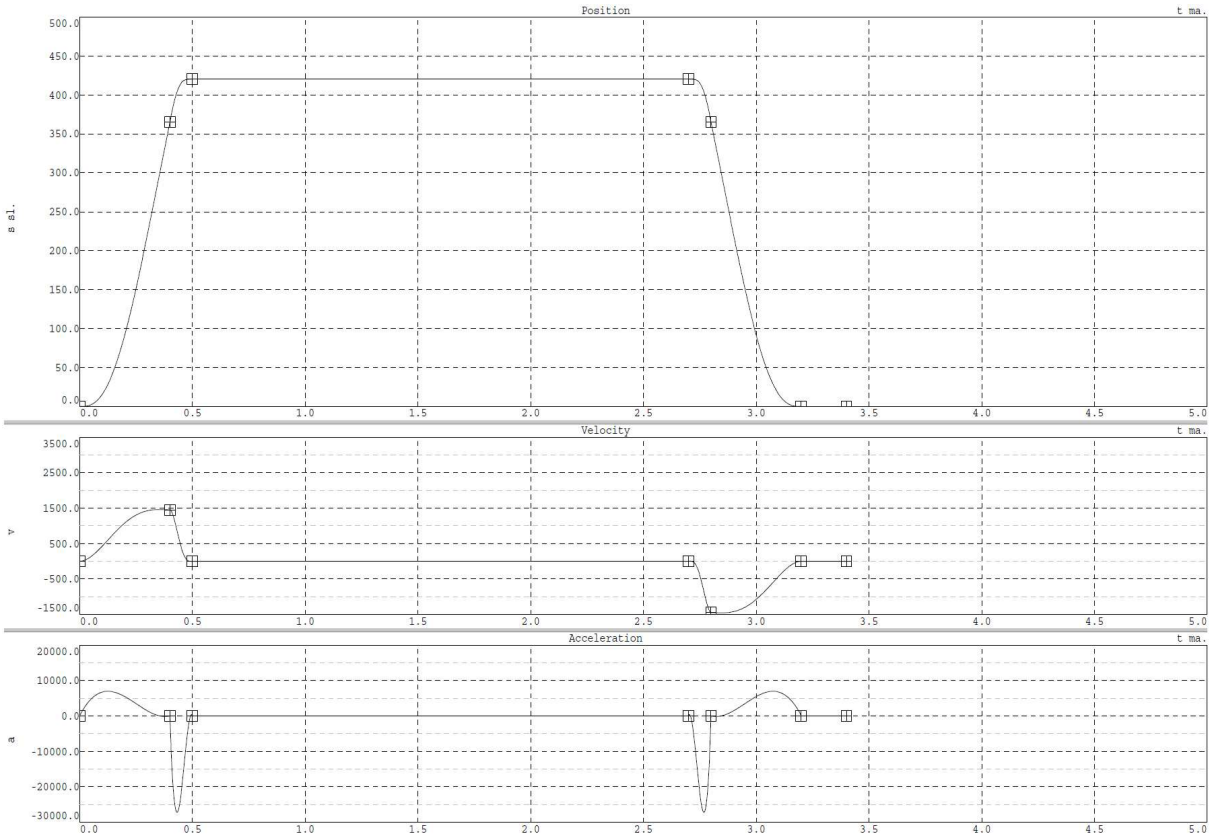


Návrh vačky pro katapult – vertikální pohyb (motor 2 a 4)

Fixní bod				
č.f.	čas	poloha	rychlost	zrychlení
	[s]	[unit]	[unit/s]	[unit/s^2]
1	0,000	0,000	-	-
2	1,050	0,000	-	-
3	1,300	65,000	373,000	-981,000
4	1,400	85,000	52,380	-590,551
5	1,675	135,000	-	-
6	2,250	135,000	-	-
7	2,480	88,372	-90,476	2322,835
8	2,628	65,000	-365,000	-981,000
9	2,950	0,000	-	-
10	4,000	0,000	-	-

Definice fixních bodů do vačko. editoru (kat. – V – s2 a s4)

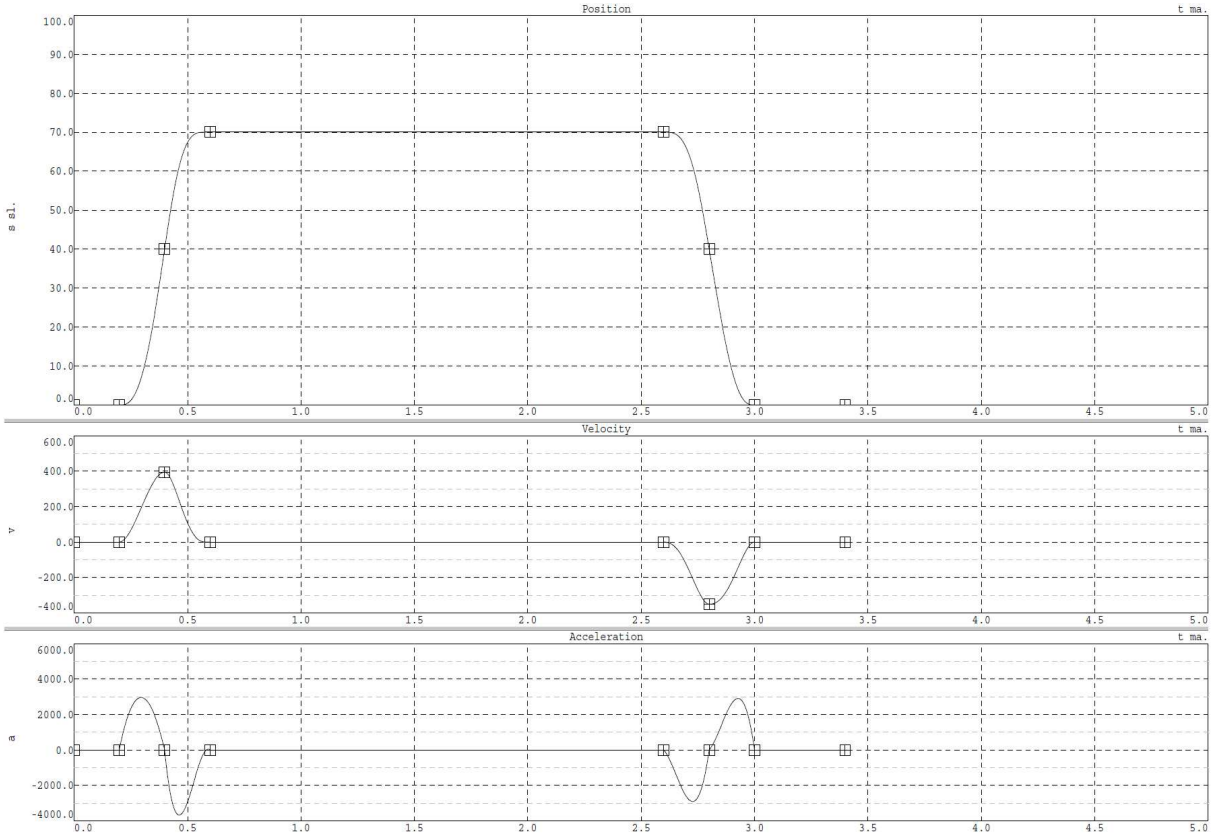
Příloha č.5 – Vačkový profil - funkce jedné koule



Návrh vačky pro jednu kouli – horizontální pohyb (motor 1)

Fixní bod					Lineární část (aproximace)	
č.f.	čas [s]	poloha [unit]	rychlost [unit/s]	zrychlení [unit/s^2]	čas [s]	poloha [unit]
1	0,000	0,000	-	-	0,500	420,000
2	0,400	365,000	1450,000	-	2,700	420,000
3	2,800	365,000	-1450,000	-		
4	3,200	0,000	-	-		
5	3,400	0,000	-	-		

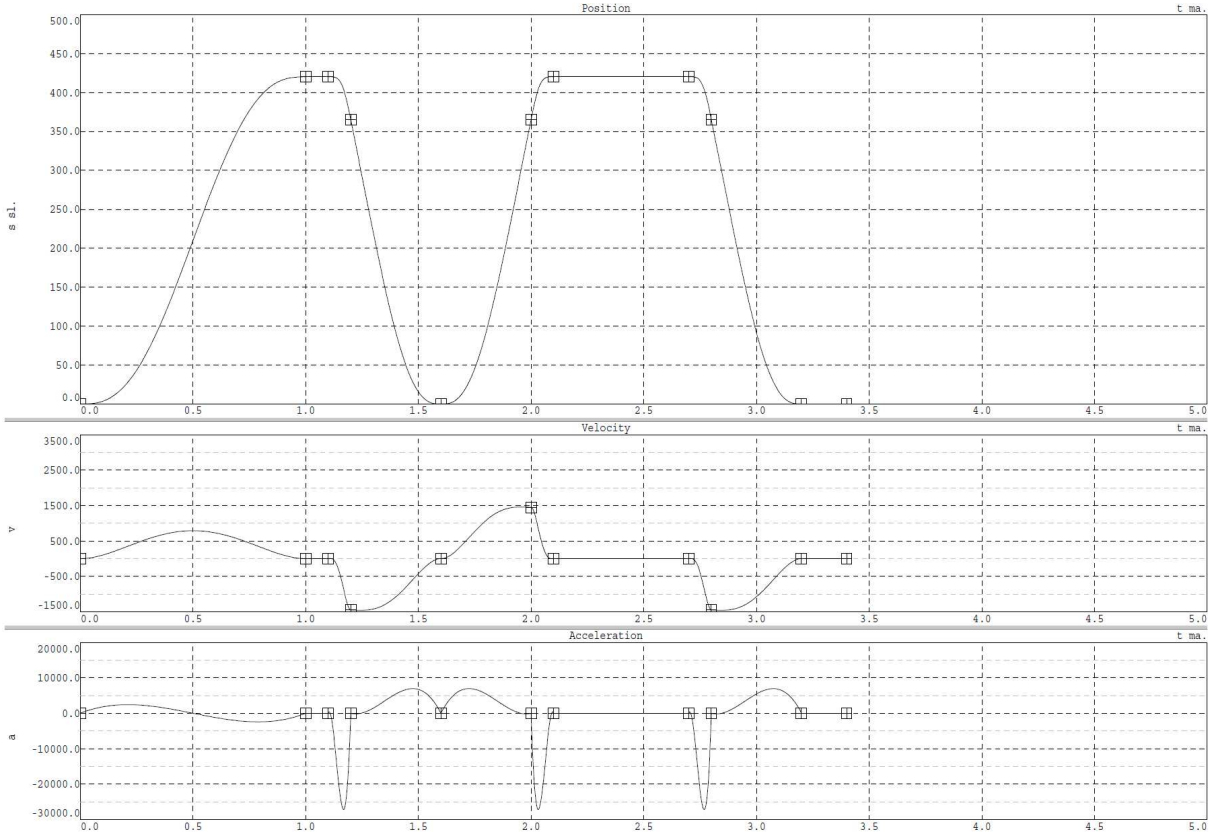
Definice fixních bodů do vačko. editoru (1 kou. – H – s1)



Návrh vačky pro jednu kouli – vertikální pohyb (motor 2)

Fixní bod					Lineární část (aproximace)	
č.f.	čas [s]	poloha [unit]	rychlost [unit/s]	zrychlení [unit/s^2]	čas [s]	poloha [unit]
1	0,400	40,000	393,600	-	0,000	0,000
2	2,800	40,000	-350,000	-	0,200	0,000
3					0,600	70,000
4					2,600	70,000
5					3,000	0,000
6					3,400	0,000

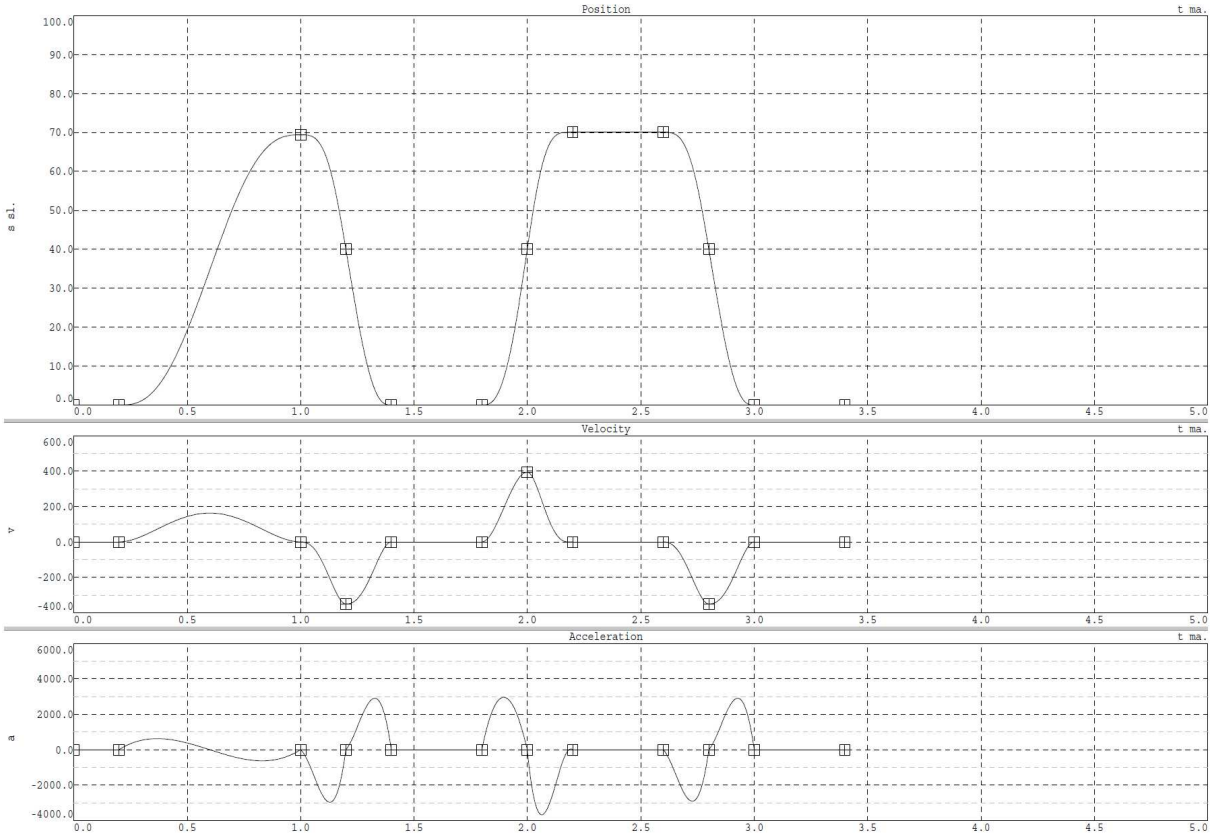
Definice fixních bodů do vačko. editoru (1 kou. – V – s2)



Návrh vačky pro jednu kouli – horizontální pohyb (motor 3)

Fixní bod					Lineární část (aproximace)	
č.f.	čas [s]	poloha [unit]	rychlost [unit/s]	zrychlení [unit/s^2]	čas [s]	poloha [unit]
1	0,000	0,000	-	-	1,000	420,000
2	1,200	365,000	-1450,000	-	1,100	420,000
3	1,600	0,000	-	-	2,100	420,000
4	2,000	365,000	1450,000	-	2,700	420,000
5	2,800	365,000	-1450,000	-		
6	3,200	0,000	-	-		
7	3,400	0,000	-	-		

Definice fixních bodů do vačko. editoru (1 kou. – H – s3)

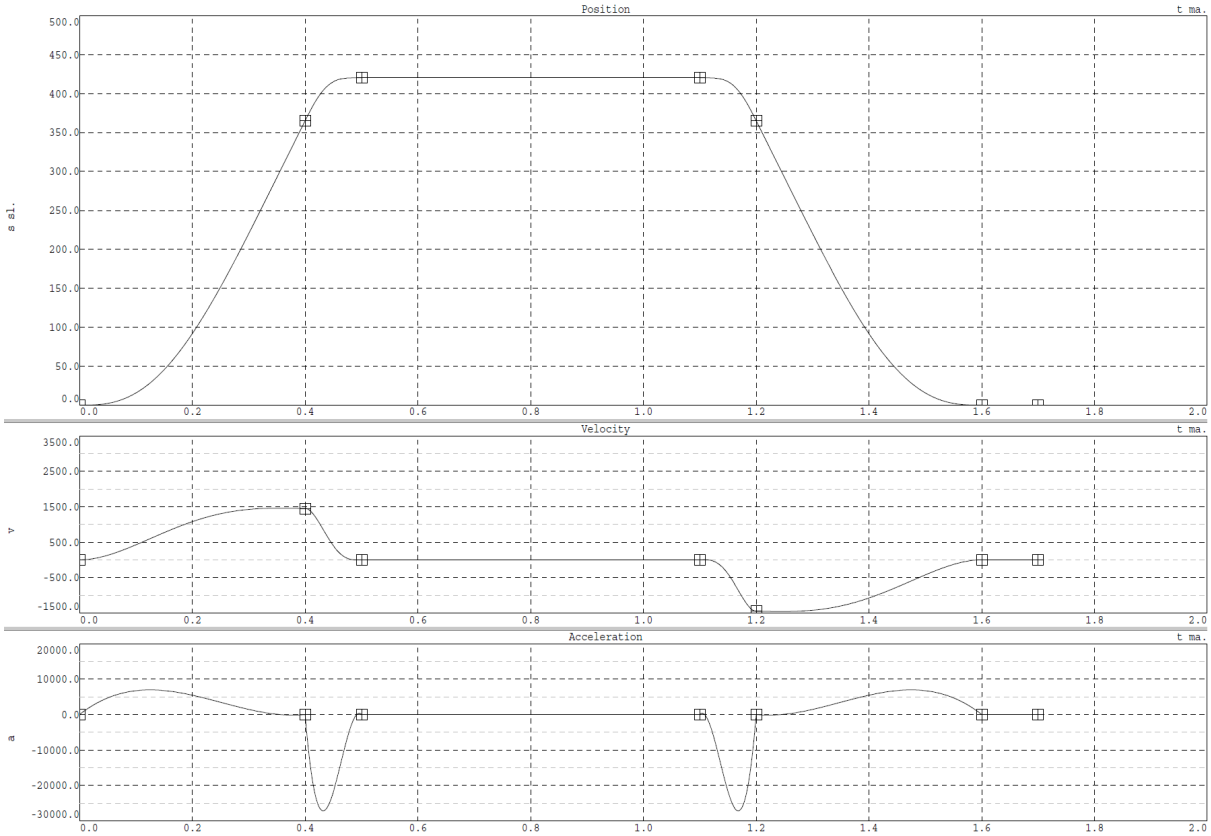


Návrh vačky jednu kouli – vertikální pohyb (motor 4)

Fixní bod					Lineární část (aproximace)	
č.f.	čas [s]	poloha [unit]	rychlost [unit/s]	zrychlení [unit/s^2]	čas [s]	poloha [unit]
1	1,000	70,000	-	-	0,000	0,000
2	1,200	40,000	-350,000	-	0,200	0,000
3	2,000	40,000	369,600	-	1,400	0,000
4	2,800	40,000	-350,000	-	1,800	0,000
5					2,200	70,000
6					2,600	70,000
7					3,000	0,000
8					3,400	0,000

Definice fixních bodů do vačko. editoru (1 kou. – V – s4)

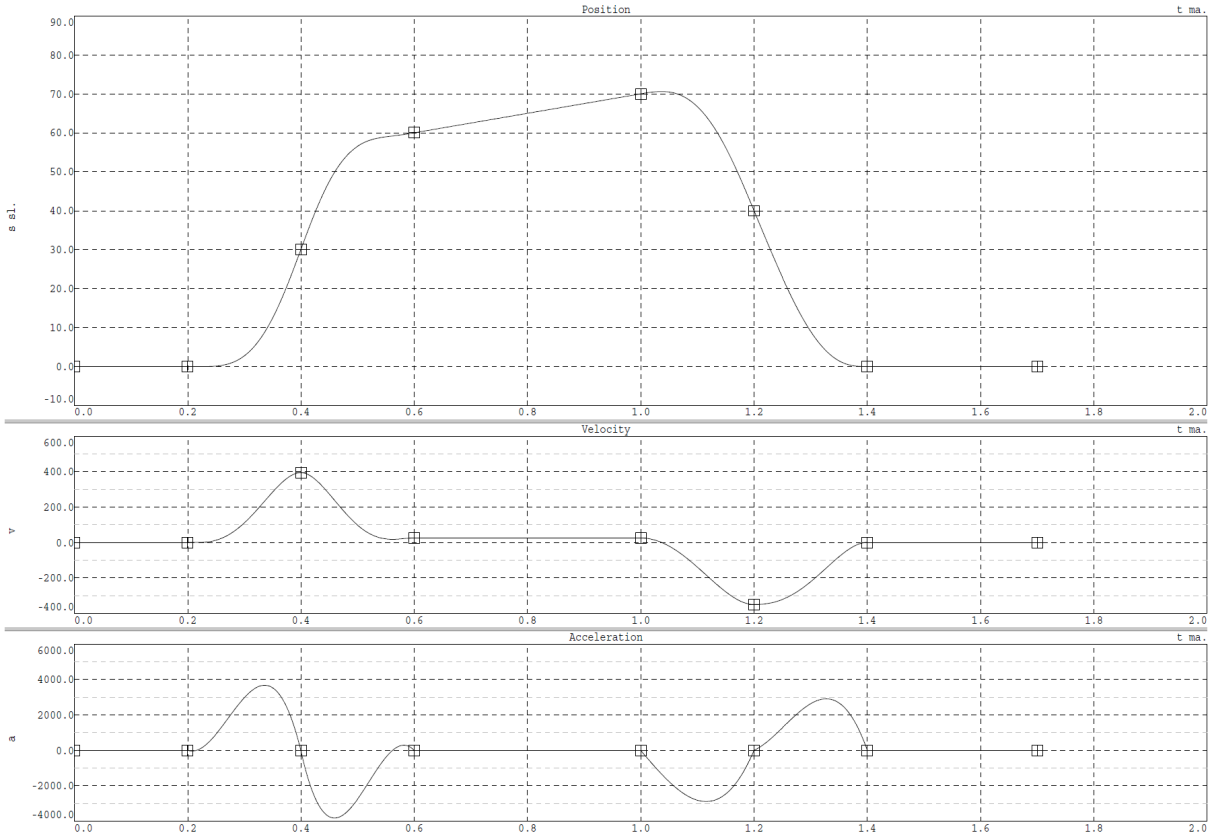
Příloha č.6 – Vačkový profil - funkce dvou koulí



Návrh vačky pro dvě koule – horizontální pohyb (motor 1 a 3)

Fixní bod					Lineární část (aproximace)	
č.f.	čas	poloha	rychlost	zrychlení	čas	poloha
	[s]	[unit]	[unit/s]	[unit/s^2]	[s]	[unit]
1	0,000	0,000	-	-	0,500	420,000
2	0,400	365,000	1450,000	-	1,100	420,000
3	1,200	365,000	-1450,000	-		
4	1,600	0,000	-	-		
5	1,700	0,000	-	-		

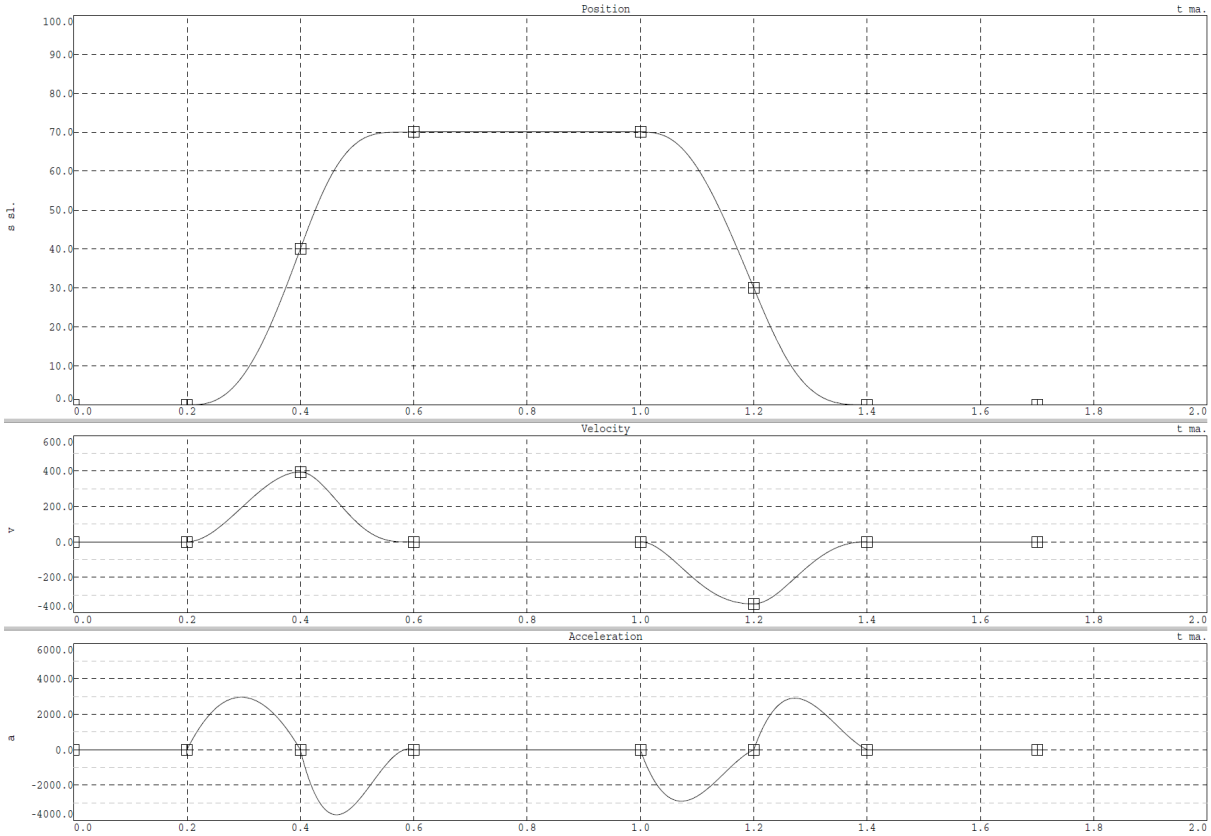
Definice fixních bodů do vačko. editoru (2 kou. – H – s1 a s3)



Návrh vačky pro dvě koule – vertikální pohyb (motor 2)

Fixní bod					Lineární část (aproximace)	
č.f.	čas [s]	poloha [unit]	rychlost [unit/s]	zrychlení [unit/s^2]	čas [s]	poloha [unit]
1	0,400	30,000	393,600	-	0,000	0,000
2	1,200	40,000	-350,000	-	0,200	0,000
3					0,600	60,000
4					1,000	70,000
5					1,400	0,000
6					1,700	0,000

Definice fixních bodů do vačko. editoru (2 kou. – V – s2)

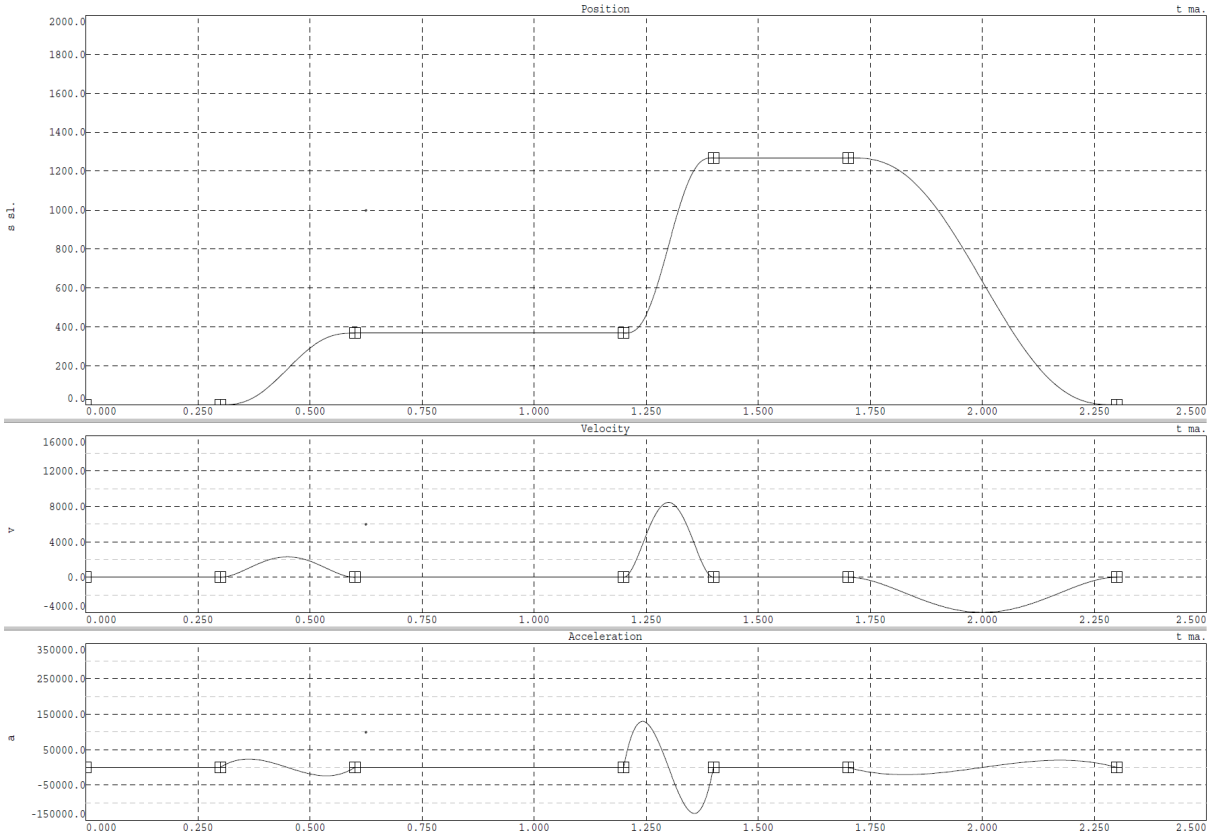


Návrh vačky pro dvě koule – vertikální pohyb (motor 4)

Fixní bod					Lineární část (aproximace)	
č.f.	čas [s]	poloha [unit]	rychlost [unit/s]	zrychlení [unit/s^2]	čas [s]	poloha [unit]
1	0,400	40,000	393,600	-	0,000	0,000
2	1,200	30,000	-350,000	-	0,200	0,000
3					0,600	70,000
4					1,000	70,000
5					1,400	0,000
6					1,700	0,000

Definice fixních bodů do vačko. editoru (2 kou. – V – s4)

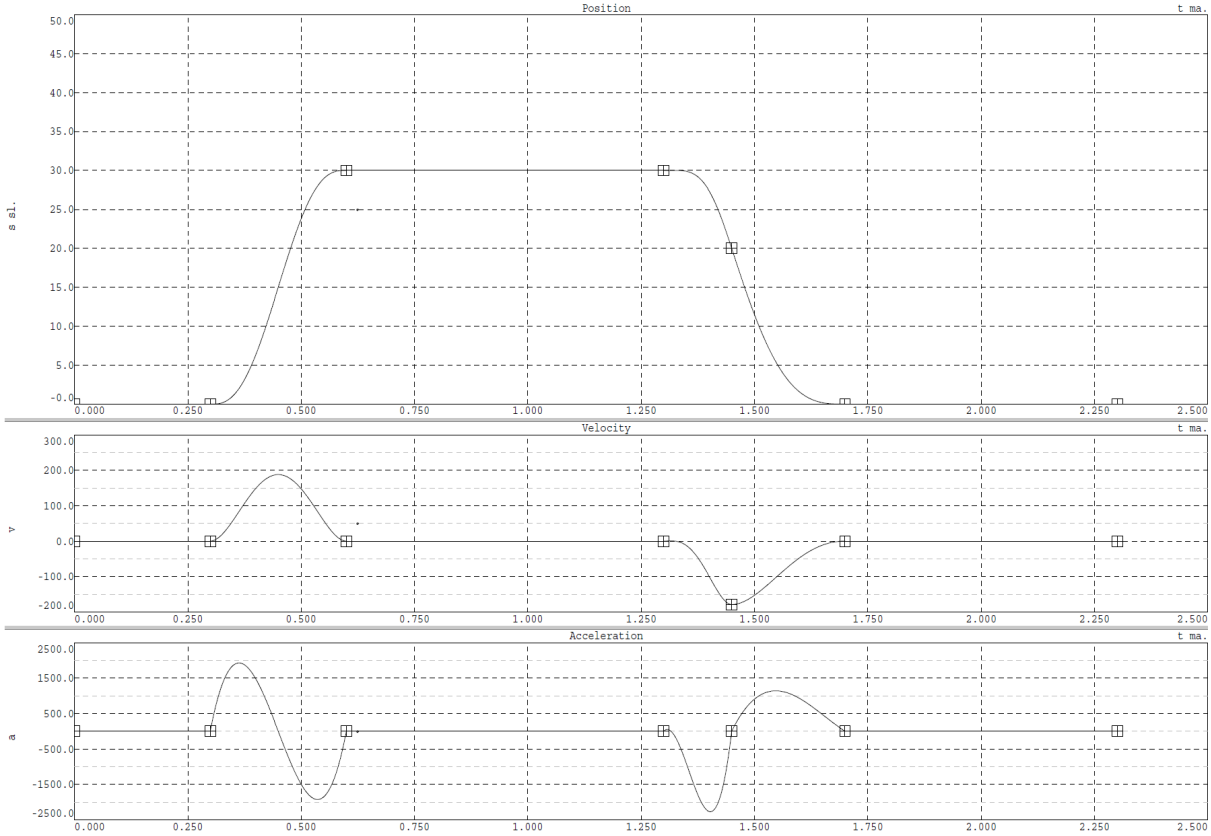
Příloha č.7 – Vačkový profil - funkce nahoď kouli



Návrh vačky pro nahození koule – horizontální pohyb (motor 1 a 3)

Fixní bod					Lineární část (aproximace)	
č.f.	čas [s]	poloha [unit]	rychlost [unit/s]	zrychlení [unit/s^2]	čas [s]	poloha [unit]
1	0,000	0,000	-	-	0,600	368,000
2	0,300	0,000	-	-	1,200	368,000
3	1,400	1268,000	-	-		
4	1,700	1268,000	-	-		
5	2,300	0,000	-	-		

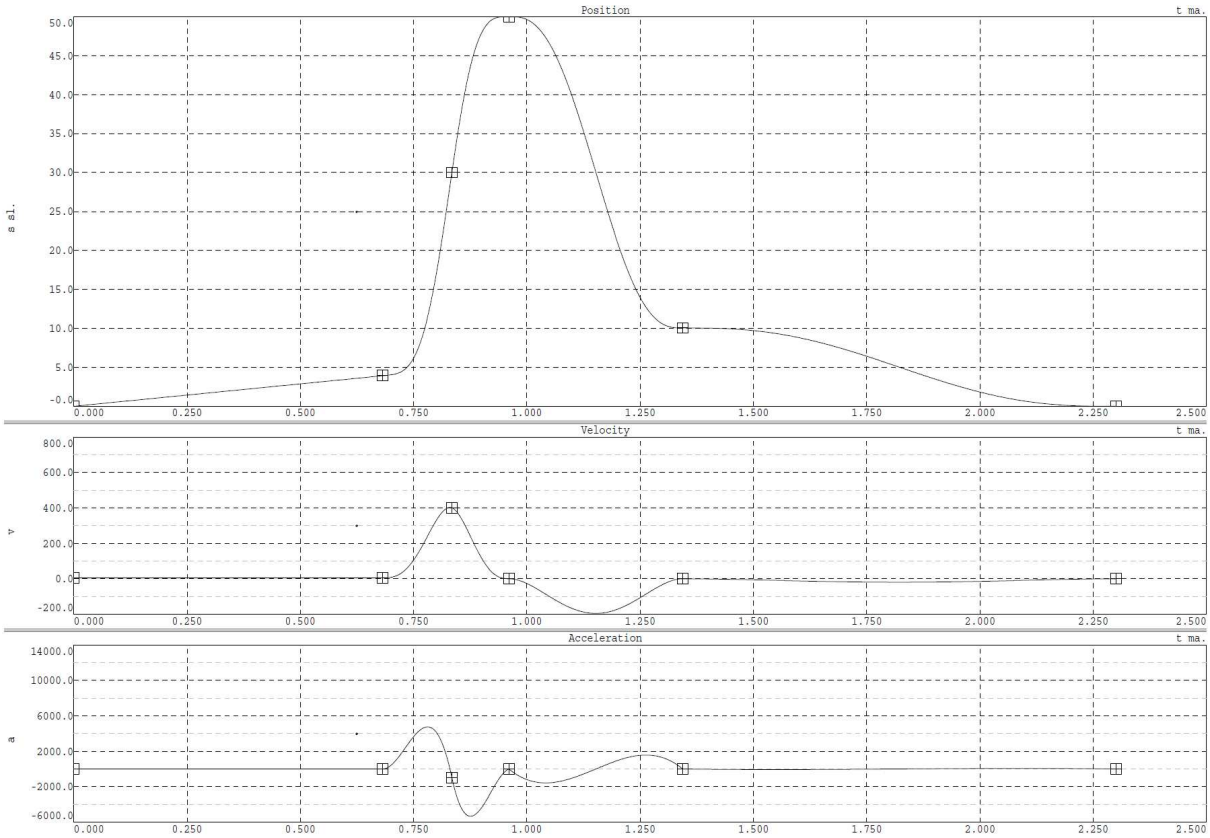
Definice fixních bodů do vačko. editoru (nah. kou. – H – s1 a s3)



Návrh vačky pro nahození koule – vertikální pohyb (motor 2 a 4)

Fixní bod					Lineární část (aproximace)	
č.f.	čas [s]	poloha [unit]	rychlost [unit/s]	zrychlení [unit/s^2]	čas [s]	poloha [unit]
1	0,000	0,000	-	-	0,600	30,000
2	0,300	0,000	-	-	1,300	30,000
3	1,450	20,000	-178,790	-		
4	1,700	0,000	-	-		
5	2,300	0,000	-	-		

Definice fixních bodů do vačko. editoru (nah. kou. – V – s2 a s4)

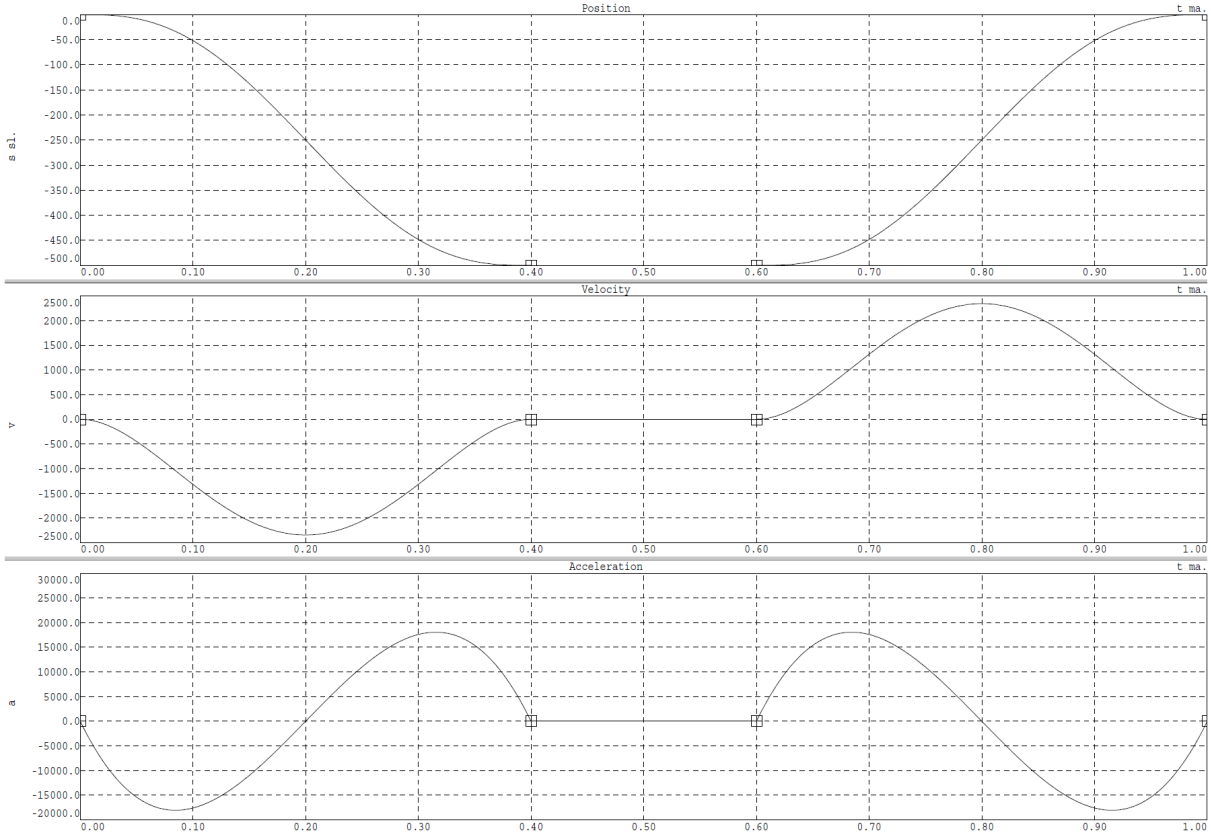


Návrh vačky pro nahození koule – vertikální pohyb (lineární motor)

Fixní bod					Lineární část (aproximace)	
č.f.	čas	poloha	rychlost	zrychlení	čas	poloha
	[s]	[unit]	[unit/s]	[unit/s^2]	[s]	[unit]
1	0,835	30,000	400,000	-981,000	0,000	0,000
2	0,960	50,000	-	0,000	0,683	3,922
3	1,350	10,000	-	0,000		
4	2,300	0,000	-	0,000		

Definice fixních bodů do vačko. editoru (nah. kou. – V – lin. motor)

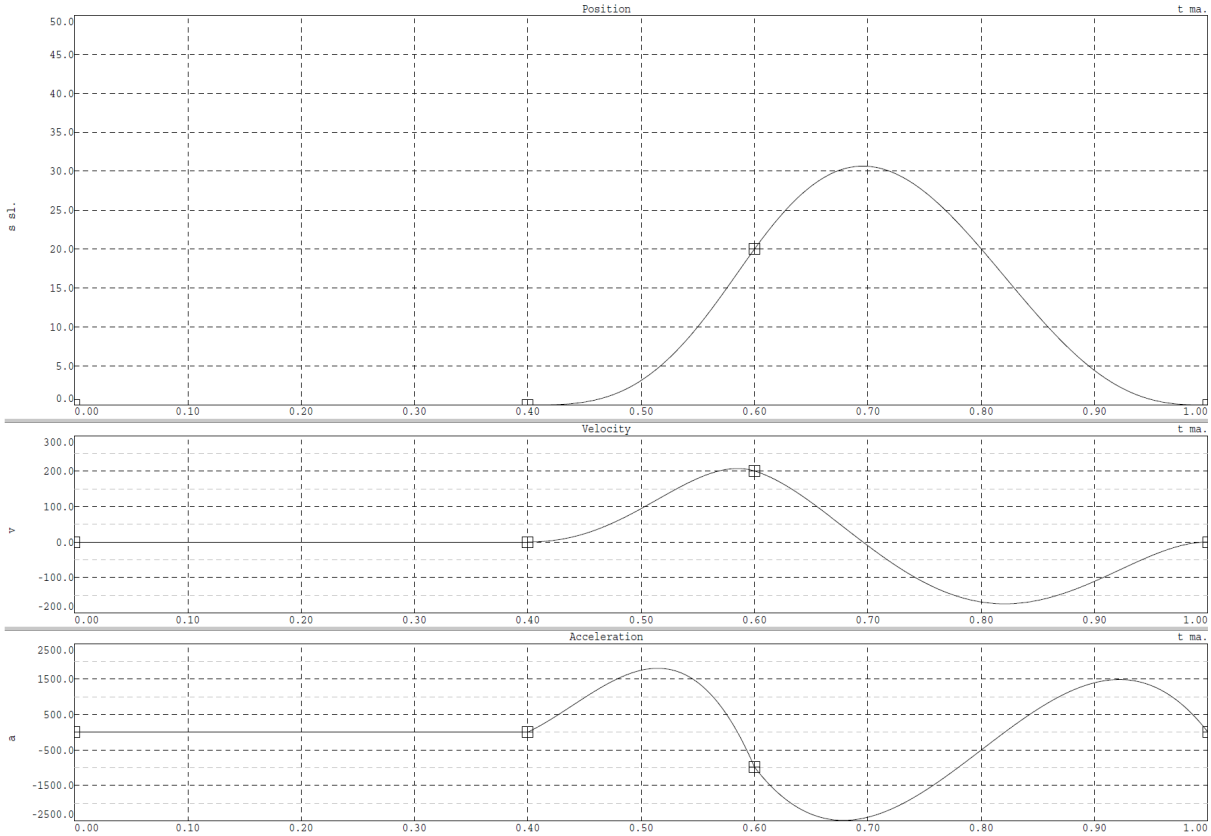
Příloha č.8 – Vačkový profil - funkce zahoď kouli



Návrh vačky pro zahození koule – horizontální pohyb (motor 1 a 3)

Fixní bod					Lineární část (aproximace)	
č.f.	čas	poloha	rychlost	zrychlení	čas	poloha
	[s]	[unit]	[unit/s]	[unit/s^2]	[s]	[unit]
1	0,000	0,000	-	-	0,400	-500,000
2	1,000	0,000	-	-	0,600	-500,000

Definice fixních bodů do vačko. editoru (zah. kou. – H – s1 a s3)

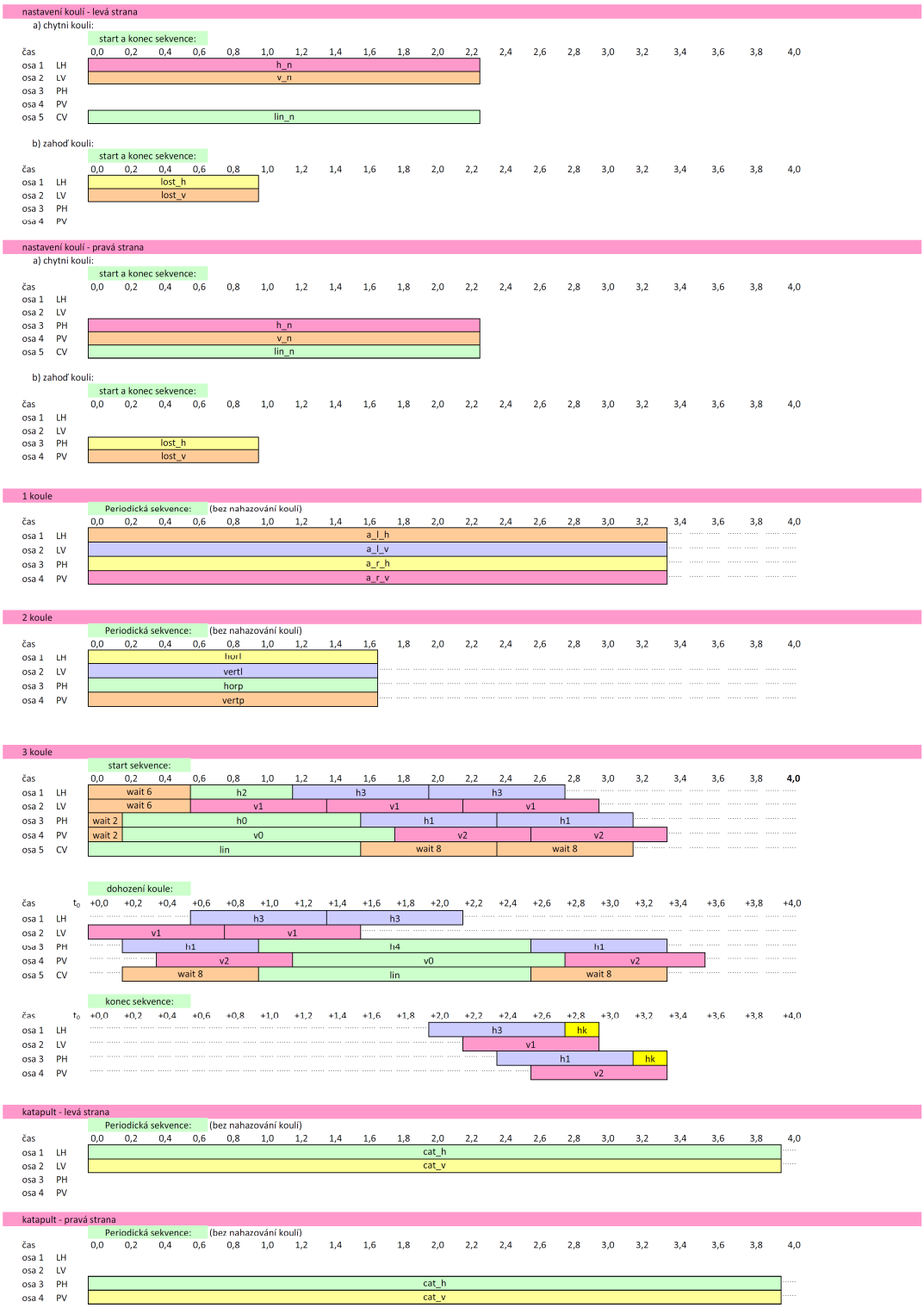


Návrh vačky pro zahození koule – vertikální pohyb (motor 2 a 4)

Fixní bod					Lineární část (aproximace)	
č.f.	[s]	[unit]	[unit/s]	[unit/s^2]	[s]	[unit]
	[s]	[cm]	[cm/s]	[cm/s^2]	[s]	[cm]
1	0,600	20,000	200,000	-981,000	0,000	0,000
2	1,000	0,000	-	0,000	0,400	0,000

Definice fixních bodů do vačko. editoru (zah. kou. – V – s2 a s4)

Příloha č.9 – Posloupnost vaček



Sekvence řazení vaček podle jednotlivých provozních módů

Příloha č.10 – Přiložené CD

Součástí bakalářské práce je CD obsahující:

- **Text** – bakalářská práce v digitální formě a podklady při její tvorbě
- **Dokumentace** – motory, PLC, Acoposy a mnoho jiných
- **Flashové animace** – návod pro studenty při založení nového programu na modelu žongléra
- **Videa** – jednotlivé funkce žonglování
- **Program** – navržený program pro ovládání modelu žongléra