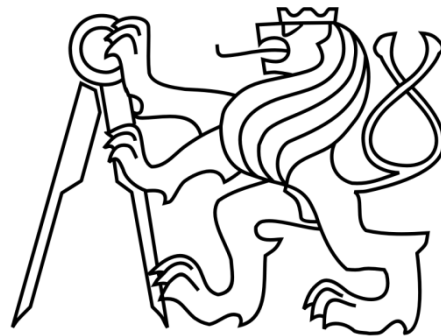


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ

ÚSTAV MECHANIKY, BIOMECHANIKY A MECHATRONIKY

Odbor mechaniky a mechatroniky



Diplomová práce

Redundantní lanový manipulátor

Praha, 2016

Karel Kraus

Zadání diplomové práce

- 1) Seznamte se s principy řízení lanových manipulátorů.
- 2) Vytvořte simulační model redundantního lanového manipulátoru v prostředí Matlab/Simulink.
- 3) Sestavte experimentální lanový manipulátor.
- 4) Do vytvořeného manipulátoru implementujte systém pro řízení redundantních lanových struktur.

Anotační list

| | |
|---------------------------------|--|
| Jméno autora: | Bc. Karel Kraus |
| Název diplomová práce: | Redundantní lanový manipulátor |
| Anglický název: | Redundantly actuated cable-driven manipulator |
| Akademický rok: | 2015/2016 |
| Obor studia: | Mechatronika |
| Ústav/odbor: | Ústav mechaniky, biomechaniky a mechatroniky Odbor Mechaniky a mechatroniky |
| Vedoucí diplomové práce: | Ing. Petr Beneš, Ph.D. |
| Bibliografické údaje: | Počet stran: 58 Počet obrázků: 38 Počet příloh: 1xCD |
| Klíčová slova: | paralelní lanový manipulátor, redundantní řízení, distribuce sil, Matlab, Arduino, RoboClaw |
| Keywords: | cable-driven parallel robot, redundant actuators, tension distribution, Matlab, Arduino, RoboClaw |
| Abstrakt: | Práce se zabývá redundantním řízením paralelního lanového manipulátoru s distribucí sil do všech lan najednou a netečnou parametrickou kontinuální změnou sil v lanech. Dále pak obsahuje model řízení, stavbu reálného experimentálního lanového manipulátoru a implementaci modelu řízení. |
| Abstract: | This paper deals with redundantly actuated cable-driven parallel robot with tension distribution through all cables at once and inert parametric continual change of cable tensions. Besides it also contains control model, real experimental cable driven robot and implementation of control model. |

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

V Praze dne

.....

Podpis

Poděkování

Rád bych poděkoval vedoucímu diplomové práce Ing. Petru Benešovi, Ph.D. za ochotu, veškerý věnovaný čas, připomínky a rady, které vedly k dokončení mé práce. Za čas, úsilí a snahu při výrobě potřebných mechanických součástí děkuji panu Marcelu Boháčkovi. Nakonec bych rád poděkoval všem lidem, kteří mě při tvorbě této práce podporovali.

Obsah

| | |
|---|--------|
| Zadání diplomové práce | - 1 - |
| Anotační list | - 2 - |
| Prohlášení | - 3 - |
| Poděkování | - 4 - |
| Seznam obrázků | - 6 - |
| 1. Úvod..... | - 7 - |
| 1.1. Rešerše..... | - 8 - |
| 1.2. Cíle práce..... | - 8 - |
| 2. Model paralelního lanového manipulátoru | - 9 - |
| 2.1. Kinematika systému | - 9 - |
| 2.1.1. Popis systému | - 9 - |
| 2.1.2. Eulerovy úhly s virtuální precesí..... | - 11 - |
| 2.1.3. Vazbové rovnice..... | - 14 - |
| 2.1.4. Jacobiho matice | - 15 - |
| 2.2. Dynamika systému | - 17 - |
| 3. Řídicí model..... | - 19 - |
| 3.1. Struktura řídicího modelu | - 19 - |
| 3.2. Dopředná kinematika..... | - 22 - |
| 3.3. Distribuce sil..... | - 23 - |
| 3.3.1. Redundance lanových systémů | - 23 - |
| 3.3.2. Efektivní využití redundantních pohonů | - 28 - |
| 3.3.3. Výpočet polygonu dosažitelnosti | - 31 - |
| 3.4. Optimalizace umístění lan | - 32 - |
| 4. Experimentální PLM..... | - 37 - |
| 4.1. Topologie fyzického manipulátoru..... | - 37 - |
| 4.2. Dimenzování..... | - 39 - |
| 4.3. Hardware | - 40 - |
| 4.3.1. Výkonný a měřicí hardware | - 40 - |
| 4.3.2. Výpočetní a řídicí hardware | - 42 - |
| 4.3.3. Zapojení a schémata | - 44 - |
| 4.4. Testy a výsledky | - 52 - |
| 4.4.1. Silové řízení navijáku..... | - 52 - |
| 4.4.2. Oživení lanového manipulátoru | - 55 - |
| 5. Závěr | - 57 - |
| 6. Citovaná literatura..... | - 58 - |

Seznam obrázků

| | |
|---|--------|
| obr. 1 - Schéma paralelního lanového manipulátoru..... | - 9 - |
| obr. 2 - Vektorový popis kinematiky systému | - 10 - |
| obr. 3 - Eulerovy úhly 3-1-3..... | - 11 - |
| obr. 4 - Decentralizovaný (lokální) model řízení [2] (FK = Forward Kinematics)..... | - 19 - |
| obr. 5 - Centralizovaný (globální) model řízení [2] (FK = Forward Kinematics) | - 21 - |
| obr. 6 - Schéma centralizovaného modelu řízení | - 22 - |
| obr. 7 - Parametrická plocha Σ | - 26 - |
| obr. 8 - Dosažitelný polygon A | - 26 - |
| obr. 9 - Soustava nerovnic..... | - 29 - |
| obr. 10 - Dosažitelný polygon A | - 29 - |
| obr. 11 - Distribuce sil skrze polygon | - 30 - |
| obr. 12 - Algoritmus stopování polygonu | - 32 - |
| obr. 13 - Optimalizace - číslování vrcholů | - 33 - |
| obr. 14 - Manipulovatelnost a lanový manipulátor | - 34 - |
| obr. 15 - Zavěšení platformy | - 35 - |
| obr. 16 - Parametr posuvu | - 35 - |
| obr. 17 - Kolize lan..... | - 35 - |
| obr. 18 - Optimalizace lanových konfigurací..... | - 36 - |
| obr. 19 - Optimální konfigurace lan | - 36 - |
| obr. 20 - Manipulátor - foto..... | - 37 - |
| obr. 21 - Model navijáku | - 38 - |
| obr. 22 - Foto navijáku | - 38 - |
| obr. 23 - Naviják - vedení lana | - 38 - |
| obr. 24 - Motor Pololu 131,25:1 37Dx73L | - 41 - |
| obr. 25 - Tenzometr 10kg C3 | - 42 - |
| obr. 26 - Tenzometr - způsob namáhání..... | - 42 - |
| obr. 27 - RoboClaw 2x15A | - 43 - |
| obr. 28 - RoboClaw - terminály | - 43 - |
| obr. 29 - Arduino Mega 2560..... | - 44 - |
| obr. 30 - RoboClaw – zapojení [7] | - 45 - |
| obr. 31 - RoboClaw - multi-unit zapojení [7]..... | - 46 - |
| obr. 32 - RoboClaw - multi-unit & multi-mcu zapojení | - 48 - |
| obr. 33 - Zesilovač - základní schéma [8] | - 50 - |
| obr. 34 - Zesilovač - celé schéma | - 51 - |
| obr. 35 - Schéma propojení HW..... | - 52 - |
| obr. 36 - PID regulátor - schéma | - 53 - |
| obr. 37 - PI regulace silově řízeného pohonu | - 54 - |
| obr. 38 - Test pohyblivosti - Průběh síly | - 55 - |

1. Úvod

Diplomová práce se zabývá silovým řízením redundantního lanového manipulátoru a sestavením reálného experimentálního manipulátoru. Ten je tvořen krychlovou konstrukcí o délce hrany 1 m s pracovním prostorem uvnitř takové krychle a s lany navíjenými na buben převodovaného motoru (dále jen naviják), která pohybují s platformou v části pracovního prostoru. Všechny navijáky jsou navrženy modulově – lze je tedy umístit na různé pozice konstrukce a měnit tak charakter pracovního prostoru. Stejným způsobem je možno variovat také body připevnění lan k samotné pohyblivé platformě a výrazně tak změnit charakter celé úlohy.

Paralelní lanové manipulátory (dále jen PLM) mají několik výhod [1]. Například náročnost systému ve smyslu celkové hmotnosti, ceny, atd. prakticky nezávisí na velikosti pracovního prostoru. Dokud se jedná o stejnou platformu/zátěž, je už vcelku nedůležité, jestli se chystáme obsluhovat prostor malé místnosti nebo třeba fotbalového stadionu, vše totiž závisí pouze na pozici výstupních bodů kabelu v prostoru. S tím souvisí také výhoda umístění navijáků (např. vzhledem k jejich hmotnosti) kdekoli v okolí pracovního prostoru a vyvedení lan přes kladky do potřebných míst. Výhodou je také skutečnost, že skrze pracovní prostor se pohybují pouze lana a případná nesená zátěž, takže prakticky veškeré dynamické účinky a silové limity jsou generované pouze platformou, a nikoli např. dlouhým ramenem jeřábu, což jednak zlepšuje poměr nesené zátěže ku zrychlení a případně pracovnímu prostoru a navíc to značně zjednoduší dynamický model systému. Mezi nevýhody se potom řadí omezení pracovního prostoru v důsledku kolize lan mezi sebou.

V této práci se budu zabývat platformou o $n = 6$ stupních volnosti řízené $m = 8$ lany, přičemž redundance u lanových systémů je složitější a nastává až při $m > n + 1$ (viz kapitola 3.3.1).

Vyjma kapitoly 4.4 budu uvažovat všechna lana jako nepoddajná.

1.1. Rešerše

Jelikož lze aktuátory řídit vždy pouze silově nebo pouze polohově, nastává u systémů s redundantními pohony potíže. Ovládat všechny pohony vzhledem k požadované poloze výkonného bodu přepočítané na polohy každého jednotlivého pohonu nelze, jelikož žádný robot není nikdy ve všech polohách přesný či dokonale tuhý a při snaze aktuátorů dosáhnout své přesné polohy dochází k nestabilitě, případně vzájemnému přetahování [2] apod. U PLM je to navíc složitější díky skutečnosti, že lana mohou působit pouze tahovou silou a nikoli tlakovou, což má za následek potřebu $n + 1$ lan pro řízení (čímž se systém ještě nutně nestává redundantním – více v kapitole 3.3.1).

Dosud se již mnoho odborných článků zabývalo PLM (skládajícími se z $n+1$ či $n+2$ lan) a termínem „redundance“ takového manipulátoru. Jen několik z nich se ale zabývá výslednou distribucí sil v lanech. Například v [3] se autor Merlet, J.-P. domnívá, že u systému s n stupni volnosti řízeného $n+1$ či více lany se do výsledného silového účinku na platformu může vždy zapojit pouze n lan a zbytek zůstane bez zátěže (v průvěsu), přičemž takových n -tic je (závisle na poloze) několik a lze aplikovat libovolnou z nich. V jiné své práci [4] Merlet, J.-P. zase uvádí, že síly lze parametricky distribuovat do všech lan na jednou, ale mezi jednotlivými distribucemi nelze kontinuálně přecházet.

Zhruba ve stejném období (2015) přichází skupina autorů s článkem [5], ve kterém uvádí jak úspěšné rozložení sil, tak algoritmus distribuce sil pro kontinuální řízení a následné zasazení teorie do praxe.

1.2. Cíle práce

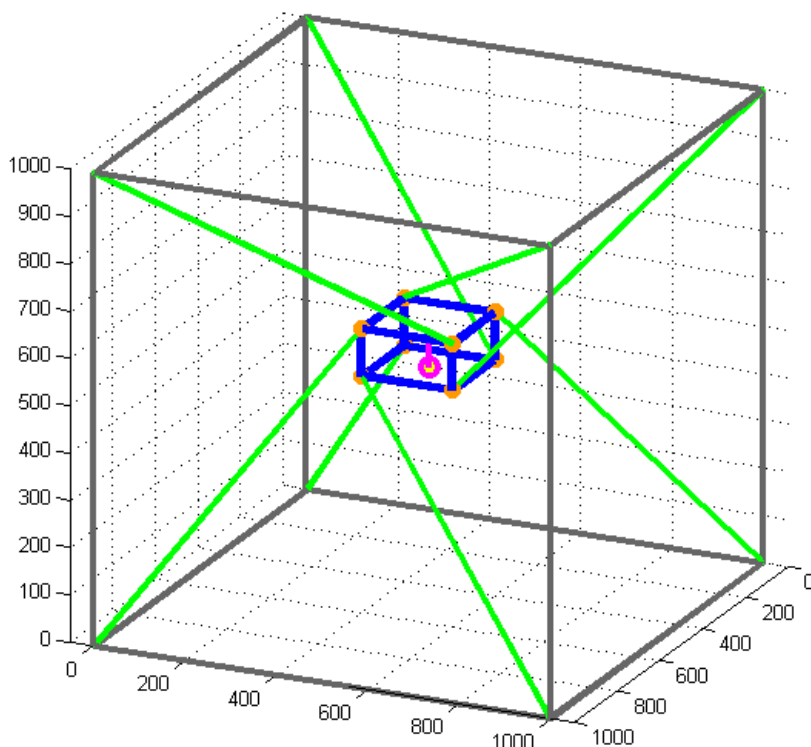
Cílem mé diplomové práce je vytvoření návrhu modelu s měřením sil v lanech určeného pro ověření možností distribuce napětí lan v různých konfiguracích, aniž by to mělo signifikantní efekt na výkonný člen manipulátoru. Druhým hlavním cílem je návrh a sestavení reálného experimentálního PLM s implementovaným řídicím modelem. Sekundárním cílem je také snaha využít nízkonákladový hardware, jako jsou výpočetní mikrokontroléry či měřicí prvky.

2. Model paralelního lanového manipulátoru

2.1. Kinematika systému

2.1.1. Popis systému

Manipulátor tvoří platforma (představující výkonný člen) zavěšená na 8 lanech uvnitř pracovního prostoru vymezeného hliníkovou konstrukcí (viz schéma na obr. 1), jejíž hrany jsou tvořeny hliníkovými profily o délce 1000 mm. Platforma má tak v prostoru 6 stupňů volnosti – tedy posuvy a rotace vzhledem ke všem třem osám. Takový systém je schopen kinematicky obsloužit prakticky celý 3D prostor uvnitř krychle. Možnost natáčení v celém tomto prostoru je ale omezena kvůli kolizím lan při velkých úhlových výchylkách a případně kvůli singularitám v Jacobiho matici (viz kapitola 2.1.4). Mimo kinematickou stránku je pak celý systém omezen ještě limity maximálních možných sil v lanech/motorech, což se výrazně projeví hlavně ve výše umístěných oblastech pracovního prostoru (viz kapitola 4.2), jelikož síly v lanech s vertikálním pohybem platformy směrem vzhůru neúměrně rostou.



obr. 1 - Schéma paralelního lanového manipulátoru

Vektorový popis kinematiky systému vyznačený na obr. 2 je následující:

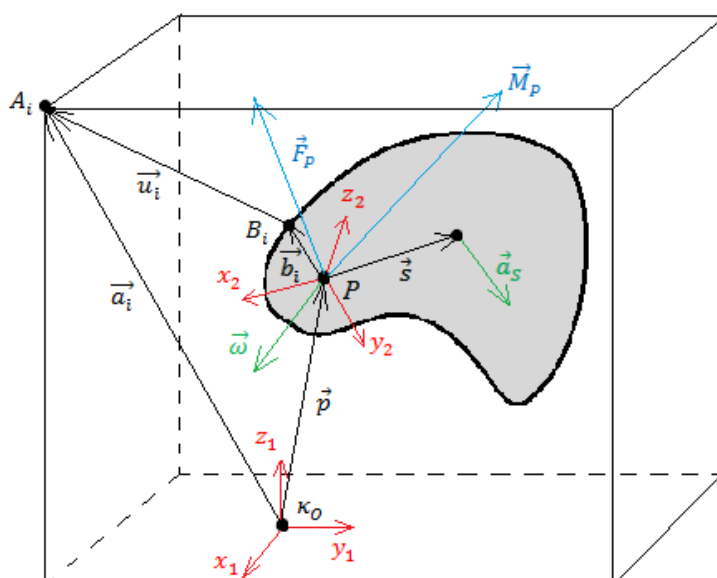
Absolutní počátek κ_0 globálního souřadnicového systému 1 popsany triviálním radiusvektorem ${}^1\vec{d} = [0,0,0]^T$ je pevně spojen s okolním světem.

Polohy bodů A_i , značících vrcholy pracovního prostoru ohraničeného hliníkovou krychlovou konstrukcí, ve kterých vyúsťují lana z kladek do pracovního prostoru, jsou v globálním systému vyjádřeny konstantními radiusvektory ${}^1\vec{a}_i$.

Nejdůležitějším bodem je bod P představující výkonný bod platformy (na obr. 1 vyznačen fialově), ve kterém je lokální souřadnicový systém 2 pevně spojen s tělesem. Vektor ${}^1\vec{p}$ pak popisuje polohu výkonného bodu v globálním systému. Natočení lokálního systému 2 v prostoru je vyjádřeno vektorem úhlových výchylek ${}^1\vec{\gamma} = [\varphi, \psi, \vartheta]^T$. Jelikož bod P je v lokálním systému triviální a vektory ${}^1\vec{\gamma}$, ${}^1\vec{a}_i$ nemá smysl v lokálním systému vyjadřovat, budu v této práci zjednodušeně psát \vec{p} , $\vec{\gamma}$, \vec{a}_i ve prospěch systému globálního – tj. veškeré vektory bez levého horního indexu budou v této práci myšleny v globálním souř. systému a pro lokální systém budou značeny indexem 2.

Body B_i pak značí pozice úchyty lan k pohyblivé platformě. Jejich polohu od počátku lokálního systému vyjádřím vektorem ${}^2\vec{b}_i$, který je konstantní, protože pozice úchytlů lan se vzhledem k bodu P nemění. Užitečnější ovšem bude jeho transformovaná forma ${}^1\vec{b}_i$ v systému globálním, která se mění v závislosti na aktuální poloze platformy.

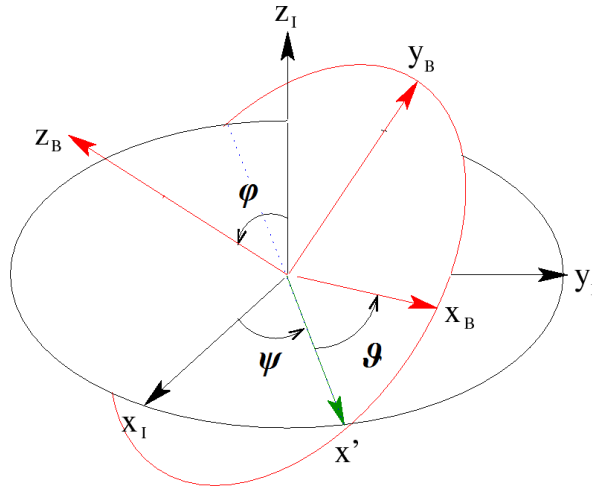
Vektory $\vec{u}_i = \vec{B}_i - \vec{A}_i$ pak představují samotná lana, resp. spojnice místa úchyty lana na platformě a příslušného navijáku, a je možno jej z ostatních vektorů snadno dopočítat.



obr. 2 - Vektorový popis kinematiky systému

2.1.2. Eulerovy úhly s virtuální precesí

Jelikož neexistuje žádné jednoznačné vyjádření natočení tělesa v prostoru, je třeba použít některého z Eulerových popisů. Ten používá sekvenci tří rotací okolo os systému 1, 2 či obou. Ve výsledku tak existuje 24 různých sekvencí [6]. Obvykle se používá např. sekvence 3-1-3 (viz obr. 3).



obr. 3 - Eulerovy úhly 3-1-3

V této práci používám pravotočivý systém s osami x, y v horizontální ploše a s vertikální osou z . Bez hlubšího úmyslu jsem se rozhodl místo sekvence 3-1-3 pro popis použít sekvenci 3-2-3, tedy

- precese φ okolo lok. osy z
- nutace ψ okolo lok. osy y
- rotace ϑ okolo lok. osy z

V takovém případě se pak dá spočítat rotační matice pro převod vektorů ze systému 2 do systému 1 jako

$$\mathbf{S}_{12} = \mathbf{S}_z(\varphi)\mathbf{S}_y(\psi)\mathbf{S}_z(\vartheta) \quad (2.1)$$

$$\mathbf{S}_{12} = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} \begin{bmatrix} \cos \vartheta & -\sin \vartheta & 0 \\ \sin \vartheta & \cos \vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.2)$$

kde S_i je rotační matice okolo osy i , a to tak, že libovolný obecný vektor ${}^2\vec{v}$ vyjádřený v lokálním systému lze do systému globálního transformovat způsobem

$${}^1\vec{v} = \mathbf{S}_{12} {}^2\vec{v}. \quad (2.3)$$

Dále budu ve své práci využívat matici \mathbf{B} pro převod úhlových rychlostí (tedy časových derivací Eulerových úhlů) $\dot{\gamma} = [\dot{\varphi}, \dot{\psi}, \dot{\vartheta}]^T$ na úhlové rychlosti okolo os lokálního systému vyjádřené vektorem ${}^2\vec{\omega} = [{}^2\omega_x, {}^2\omega_y, {}^2\omega_z]^T$ tak, že

$${}^2\vec{\omega} = \mathbf{B}\dot{\gamma}. \quad (2.4)$$

Postup získání matice \mathbf{B} popíšu na následujících řádcích [6].

Pozn.: pro jednoduchost zápisu budu u matic místo indexů os používat indexy značící rotace okolo daných os – tedy $\mathbf{S}_z(\varphi)\mathbf{S}_y(\psi)\mathbf{S}_z(\vartheta) \Rightarrow \mathbf{S}_\varphi\mathbf{S}_\psi\mathbf{S}_\vartheta$

Vycházím z předpokladu, že derivaci rotační matice lze napsat jako

$$\dot{\mathbf{S}}_{12} = \mathbf{S}_{12} {}^2\boldsymbol{\Omega}_{12}, \quad (2.5)$$

kde ${}^2\boldsymbol{\Omega}_{12}$ je antisymetrická matice úhlových rychlostí

$${}^2\boldsymbol{\Omega}_{12} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \quad (2.6)$$

Rovnici (2.5) lze díky ortogonalitě matice \mathbf{S}_{12} přepsat na

$${}^2\boldsymbol{\Omega}_{12} = \mathbf{S}_{12}^T \dot{\mathbf{S}}_{12}. \quad (2.7)$$

Časovou derivaci matice $\dot{\mathbf{S}}_{12}$ je možno pomocí (2.1) rozložit

$$\dot{\mathbf{S}}_{12} = \dot{\mathbf{S}}_\varphi \mathbf{S}_\psi \mathbf{S}_\vartheta + \mathbf{S}_\varphi \dot{\mathbf{S}}_\psi \mathbf{S}_\vartheta + \mathbf{S}_\varphi \mathbf{S}_\psi \dot{\mathbf{S}}_\vartheta \quad (2.8)$$

a za časové derivace rotačních matic dílčích pohybů lze dosadit analogicky z (2.5)

$$\dot{\mathbf{S}}_{12} = \mathbf{S}_\varphi {}^2\boldsymbol{\Omega}_\varphi \mathbf{S}_\psi \mathbf{S}_\vartheta + \mathbf{S}_\varphi \mathbf{S}_\psi {}^2\boldsymbol{\Omega}_\psi \mathbf{S}_\vartheta + \mathbf{S}_\varphi \mathbf{S}_\psi \mathbf{S}_\vartheta {}^2\boldsymbol{\Omega}_\vartheta, \quad (2.9)$$

kde ${}^2\boldsymbol{\Omega}_\varphi$, ${}^2\boldsymbol{\Omega}_\psi$ a ${}^2\boldsymbol{\Omega}_\vartheta$ jsou matice úhlových rychlostí dílčích pohybů. Po dosazení rovnice

(2.9) do (2.7) dostanu

$${}^2\boldsymbol{\Omega}_{12} = \mathbf{S}_\vartheta^T \mathbf{S}_\psi^T \mathbf{S}_\varphi^T \mathbf{S}_\varphi {}^2\boldsymbol{\Omega}_\varphi \mathbf{S}_\psi \mathbf{S}_\vartheta + \mathbf{S}_\vartheta^T \mathbf{S}_\psi^T \mathbf{S}_\varphi^T \mathbf{S}_\varphi \mathbf{S}_\psi {}^2\boldsymbol{\Omega}_\psi \mathbf{S}_\vartheta + \mathbf{S}_\vartheta^T \mathbf{S}_\psi^T \mathbf{S}_\varphi^T \mathbf{S}_\varphi \mathbf{S}_\psi \mathbf{S}_\vartheta {}^2\boldsymbol{\Omega}_\vartheta \quad (2.10)$$

a po úpravě dále

$${}^2\boldsymbol{\Omega}_{12} = \mathbf{S}_\vartheta^T \mathbf{S}_\psi^T {}^2\boldsymbol{\Omega}_\varphi \mathbf{S}_\psi \mathbf{S}_\vartheta + \mathbf{S}_\vartheta^T {}^2\boldsymbol{\Omega}_\psi \mathbf{S}_\vartheta + {}^2\boldsymbol{\Omega}_\vartheta. \quad (2.11)$$

Nyní již stačí dosadit a vynásobit všechny matice, čímž dostanu výraz

$${}^2\boldsymbol{\Omega}_{12} = \begin{bmatrix} 0 & -\dot{\vartheta} - \dot{\varphi} \cos \psi & \dot{\psi} \cos \vartheta + \dot{\varphi} \sin \psi \sin \vartheta \\ \dot{\vartheta} + \dot{\varphi} \cos \psi & 0 & -\dot{\psi} \sin \vartheta + \dot{\varphi} \sin \psi \cos \vartheta \\ -\dot{\psi} \cos \vartheta - \dot{\varphi} \sin \psi \sin \vartheta & \dot{\psi} \sin \vartheta - \dot{\varphi} \sin \psi \cos \vartheta & 0 \end{bmatrix} \quad (2.12)$$

Vidíme, že ${}^2\boldsymbol{\Omega}_{12}$ je skutečně antisymetrická matice úhlových rychlostí a tím představuje maticově zapsaný vektor ${}^2\vec{\omega}$, takže vzhledem k (2.6) můžeme psát

$${}^2\vec{\omega} = \begin{bmatrix} {}^2\omega_x \\ {}^2\omega_y \\ {}^2\omega_z \end{bmatrix} = \begin{bmatrix} \dot{\psi} \sin \vartheta - \dot{\varphi} \sin \psi \cos \vartheta \\ \dot{\psi} \cos \vartheta + \dot{\varphi} \sin \psi \sin \vartheta \\ \dot{\vartheta} + \dot{\varphi} \cos \psi \end{bmatrix} = \begin{bmatrix} -\sin \psi \cos \vartheta & \sin \vartheta & 0 \\ \sin \psi \sin \vartheta & \cos \vartheta & 0 \\ \cos \psi & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\psi} \\ \dot{\vartheta} \end{bmatrix} \quad (2.13)$$

a srovnáním s rovnicí (2.4) můžeme vyjádřit matici \mathbf{B} jako

$$\mathbf{B} = \begin{bmatrix} -\sin \psi \cos \vartheta & \sin \vartheta & 0 \\ \sin \psi \sin \vartheta & \cos \vartheta & 0 \\ \cos \psi & 0 & 1 \end{bmatrix}. \quad (2.14)$$

Nicméně vzhledem ke zvolenému souřadnicovému systému a k charakteru úlohy, kdy jde o platformu zavěšenou na lanech, budou úhly nutace a rotace dosahovat relativně malých výchylek, zatímco precese („směr podél horizontu“, do kterého se poté platforma nutací naklápí) může nabývat v podstatě libovolných hodnot. V praxi to znamená, že např. budu-li chtít platformu pouze naklopit do směru $\varphi = 90^\circ$ o nutační úhel $\psi = 20^\circ$, musím pak ještě i rotovat o úhel $\vartheta = -\varphi = -90^\circ$. Bez konečné rotace by totiž systém nebyl pouze naklopen do směru φ , ale zároveň i natočen kolem své lokální osy z právě o precesní úhel φ , což může snadno být kinematicky nemožné a případně vést i k zamotání lan navzájem.

Abych odstranil tento nedostatek, resp. zjednodušil ovládání platformy pomocí takových úhlů, zavedu tuto korekci již přímo do rotačních matic, takže

$$\mathbf{S}_{12} = \mathbf{S}_z(\varphi)\mathbf{S}_y(\psi)\mathbf{S}_z(\vartheta - \varphi) \quad (2.15)$$

$$\mathbf{S}_{12} = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} \begin{bmatrix} \cos(\vartheta - \varphi) & -\sin(\vartheta - \varphi) & 0 \\ \sin(\vartheta - \varphi) & \cos(\vartheta - \varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.16)$$

případně mohu matici \mathbf{S}_{12} definovat součinem rotační matice okolo jednotkového vektoru ležícího v rovině xy (čímž zahrnu precesi a nutaci) a rotační matice okolo osy z vyjadřující rotaci, nicméně výsledek bude analyticky ekvivalentní.

Podobnou korekci provedu také v definici matice pro převod úhlových rychlostí, takže

$$\mathbf{B} = \begin{bmatrix} -\sin \psi \cos(\vartheta - \varphi) & \sin(\vartheta - \varphi) & 0 \\ \sin \psi \sin(\vartheta - \varphi) & \cos(\vartheta - \varphi) & 0 \\ \cos \psi - 1 & 0 & 1 \end{bmatrix}. \quad (2.17)$$

kde je zároveň v rovnici (2.13) potřeba upravit i úhlovou rychlost $\dot{\vartheta}$ na $\dot{\vartheta} - \dot{\varphi}$, což se projeví změnou výrazu $\cos \psi$ v matici \mathbf{B} na výraz $\cos \psi - 1$.

Tím zavádím termín *virtuální precese*, kdy úhel φ pouze určuje směr, do kterého se platforma naklopí, a skutečná fyzická rotace platformy okolo lokální osy z tak bude pouze zásluhou rotačního úhlu ϑ .

V kapitole 2.2 bude krom úhlových rychlostí ${}^2\vec{\omega}$ zapotřebí také jejich derivace, tedy úhlová zrychlení ${}^2\dot{\vec{\omega}}$, jejichž vektor získám derivací rovnice (2.4) pomocí rozkladu

$${}^2\dot{\vec{\omega}} = \dot{\mathbf{B}}\dot{\vec{\gamma}} + \mathbf{B}\ddot{\vec{\gamma}}, \quad (2.18)$$

kde

$$\mathbf{B} = \begin{bmatrix} -\dot{\psi} \cos \psi \cos(\vartheta - \varphi) + (\dot{\vartheta} - \dot{\varphi}) \sin \psi \sin(\vartheta - \varphi) & (\dot{\vartheta} - \dot{\varphi}) \cos(\vartheta - \varphi) & 0 \\ \dot{\psi} \cos \psi \sin(\vartheta - \varphi) + (\dot{\vartheta} - \dot{\varphi}) \sin \psi \cos(\vartheta - \varphi) & -(\dot{\vartheta} - \dot{\varphi}) \sin(\vartheta - \varphi) & 0 \\ -\dot{\psi} \sin \psi & 0 & 0 \end{bmatrix}. \quad (2.19)$$

2.1.3. Vazbové rovnice

Pro popis polohy platformy v prostoru použiji popis skrze vektor \vec{p} a sadu tří Eulerových úhlů $\vec{\gamma}$. Pro jednoduchost budu pro celkový popis polohy používat souhrn těchto dvou vektorů do jednoho, tedy

$$\vec{X} = [p_x, p_y, p_z, \varphi, \psi, \vartheta]^T. \quad (2.20)$$

Dále bude v kinematice třeba pracovat se souřadnicemi délek 8 lan v úseku od úchyty na platformě ke sběrnému bodu na okraji pracovního prostoru (na hliníkové konstrukci). Na obr. 2 můžeme vidět, že takové souřadnice odpovídají velikosti vektoru $\|\vec{u}_i\|$. Takový zápis použiji, pokud se bude jednat o analyticky přesné vyjádření této souřadnice, a značení L_i zase v případě, že se bude jednat o souřadnici naměřenou na skutečném experimentálním manipulátoru, kde $i=1 \dots 8$ je index daného lana (v ideálním případě nastává rovnost $L_i = \|\vec{u}_i\|$ a takové rozdělení značení pak nemá smysl, ale v kapitole 3.2 využiji toto rozdělení pro část modelu řízení).

Tím je úloha popsána celkem 14 souřadnicemi, přičemž systém má 6 stupňů volnosti, takže bude potřeba vytvořit 8 vazbových rovnic. V systému tak je nutné vytvořit 8 kinematických smyček. Jak je vidět na obr. 2, vektory $\vec{a}_i, \vec{p}, \vec{b}_i, \vec{u}_i$ tvoří právě 8 uzavřených vektorových smyček (pro každé lano jednu). Samozřejmě je třeba dbát na to, aby všechny vektory byly vyjádřeny ve stejném, tedy v našem případě nejlépe globálním, systému. Vazbové rovnice ve formě vektorového součtu potom vypadají následovně

$$f_i(\vec{X}, L_i) = L_i - \|\vec{u}_i\| = 0. \quad (2.21)$$

kde \vec{u}_i mohu vyjádřit pomocí ostatních vektorů smyčky

$$f_i(\vec{X}, L_i) = L_i - \|\vec{a}_i - (\vec{p} + \vec{b}_i)\| = 0. \quad (2.22)$$

Z vektoru souřadnic \vec{X} se v rovnicích zdánlivě vyskytuje pouze část \vec{p} . Vektor délek lan \vec{L} známe z naměřených hodnot, vektor \vec{a}_i známý (a konstantní) a \vec{p} je aktuální poloha výkonného bodu v prostoru. A právě poslední člen \vec{b}_i skrývá druhou polovinu funkčních proměnných vektoru \vec{X} , protože \vec{p} je vektor proměnných, skrze které rotační matice přepočítává známý konstantní lokální vektor ${}^2\vec{b}_i$ do systému globálního, jako je tomu v rovnici (2.3).

2.1.4. Jacobiho matice

Jacobiho matice představuje transformaci (převodový poměr) mezi jednotlivými souřadnicemi. V našem případě jde o převod mezi derivacemi (rychlostmi) vektorů souřadnic $\dot{\vec{L}}$ a $\dot{\vec{X}}$, takže můžu vytvořit matici parciálních derivací každé ze souřadnic vektoru \vec{L} podle každé souřadnice vektoru \vec{X}

$${}^v\mathbf{J} = \begin{bmatrix} \frac{\partial L_1}{\partial X_1} & \dots & \frac{\partial L_1}{\partial X_6} \\ \vdots & \ddots & \vdots \\ \frac{\partial L_8}{\partial X_1} & \dots & \frac{\partial L_8}{\partial X_6} \end{bmatrix}, \quad (2.23)$$

kteřou pak lze použít k převodu rychlostí souřadnic polohy platformy na rychlost změny délky lana (tedy jejich derivací podle času)

$$\dot{\vec{L}} = {}^v\mathbf{J}\dot{\vec{X}}, \quad (2.24)$$

a nebo, jelikož rychlostní převod je inverzním silovým převodem, k převodu vektoru sil v lanech \vec{t} na vektor celkových silových účinků ${}^v\vec{Q}$ působících na platformu ve směru souřadnic \vec{X}

$${}^v\vec{Q} = {}^v\mathbf{J}^T\vec{t}. \quad (2.25)$$

Jacobiho matici lze dle (2.23) definovat tehdy, můžeme-li souřadnice \vec{L} napsat jako funkci souřadnic \vec{X} , tedy

$$\vec{L} = \vec{L}(\vec{X}), \quad (2.26)$$

což v našem případě lze pro i-tý prvek vektoru \vec{L} napsat jako

$$L_i = L_i(\vec{X}) = \|\vec{u}_i\| = \|\vec{a}_i - (\vec{p} + \vec{b}_i)\| = \|\vec{a}_i - (\vec{p} + \mathbf{S}_{12} {}^2\vec{b}_i)\|. \quad (2.27)$$

Dle rovnice (2.27) ale můžeme soudit, že analytické parciální derivace podle prvků vektoru \vec{X} jsou značně obtížné. Prvky vektoru \vec{X} jsou totiž zahrnuty v matici \mathbf{S}_{12} přenásobené vektorem ${}^2\vec{b}_i$. Po výpočtu Euklidovy normy pak vznikne složitý vektorový výraz, z něhož by po parciálních derivacích vznikla ještě složitější matice. Druhou nevýhodou tímto způsobem sestavené Jacobiho matice je její singularita (resp. existence závislých sloupců) v případě, že se platforma nachází v poloze s nulovou nutací – při nulovém náklonu do některého směru od osy z se totiž precese díky své virtualitě stává neurčitou – může nabývat libovolné hodnoty a sloupec Jacobiho matice odpovídající inkriminované ose generuje nulový vektor (pozn.: V případě klasické precese by tomu nebylo příliš jinak – problém by byl pouze převeden na neurčitost rozdělení natočení okolo lokální osy z do precese a rotace.). V takovém případě je vhodné využít analogie rovnosti (2.25). Jelikož známe směr \vec{u}_i síly v každém laně a také vektor \vec{b}_i ramene, na které síla působí vzhledem k bodu P , můžeme Jacobiho matici snadno sestavit [7] pomocí právě těchto vektorů

$$\begin{bmatrix} \frac{\vec{u}_1}{\|\vec{u}_1\|} & \dots & \frac{\vec{u}_8}{\|\vec{u}_8\|} \\ \frac{\vec{b}_1 \times \vec{u}_1}{\|\vec{u}_1\|} & \dots & \frac{\vec{b}_8 \times \vec{u}_8}{\|\vec{u}_8\|} \end{bmatrix} \begin{bmatrix} t_1 \\ \vdots \\ t_8 \end{bmatrix} = \begin{bmatrix} \vec{F} \\ \vec{M} \end{bmatrix}, \quad (2.28)$$

přičemž všechny vektory jsou opět vyjádřeny v globálním systému a vektory \vec{u}_i je zřejmě v tomto případě potřeba užívat jako jednotkové vektory, aby jejich velikost představovala jednotkovou sílu. Vektory \vec{F} a \vec{M} tak značí výslednici všech externích sil a momentů působících na platformu vzhledem k bodu P (včetně sil potenciálních). Tím vznikla matice

$$\mathbf{W} = \begin{bmatrix} \frac{\vec{u}_1}{\|\vec{u}_1\|} & \dots & \frac{\vec{u}_8}{\|\vec{u}_8\|} \\ \frac{\vec{b}_1 \times \vec{u}_1}{\|\vec{u}_1\|} & \dots & \frac{\vec{b}_8 \times \vec{u}_8}{\|\vec{u}_8\|} \end{bmatrix} \quad (2.29)$$

reprezentující prostor vazeb, přičemž analogicky dle rovnic (2.25) a (2.28) mohu psát

$$\mathbf{W} = \mathbf{J}^T. \quad (2.30)$$

Je ale důležité podotknout, že tímto způsobem spočítaná transponovaná Jacobiho matice \mathbf{J}^T se nevztahuje k vektoru $\dot{\vec{X}}$ (tedy k rychlosti pohybu bodu P a úhlovým rychlostem $\dot{\vec{\gamma}}$) jako je tomu u matice ${}^V\mathbf{J}$, ale k rychlosti bodu P a úhlovým rychlostem $\vec{\omega}$ vztaženým k osám globálního systému 1.

2.2. Dynamika systému

Stejně jako kinematiku také dynamiku budu vztahovat k výkonnému bodu P , který se může lišit od těžiště S . Pro popis užiji Newton-Eulerovy pohybové rovnice a nejprve budu vycházet ze situace, kdy bod P s těžištěm splývá, což pro rovnováhu sil znamená

$$\mathbf{m}\vec{a}_S = \vec{F}_S, \quad (2.31)$$

kde \vec{a}_S je zrychlení těžiště (nijak se netýká vektorů \vec{a}_i s podobným značením) a \mathbf{m} je diagonální matice hmotnosti. Pro rovnováhu momentů platí

$$\mathbf{I}_S \dot{\vec{\omega}} + \vec{\omega} \times \mathbf{I}_S \vec{\omega} = \vec{M}_S, \quad (2.32)$$

kde \mathbf{I}_S je matice setrvačnosti. Rovnice (2.31) a (2.32) lze vyjádřit jak v systému globálním, tak v lokálním. Nejdříve ale provedu jejich úpravu [6] do tvaru zohledňujícího obecnou polohu těžiště vzhledem ke vztažnému bodu P , přičemž radiusvektor těžiště v lokálním systému je \vec{s} (viz obr. 2).

Rychlost pohybu těžiště lze vyjádřit jako časovou derivaci jeho polohy

$$\vec{v}_S = \frac{d}{dt}(\vec{p} + \vec{s}) = \vec{v}_P + \vec{\omega} \times \vec{s} \quad (2.33)$$

a opětovnou derivací dle času získám zrychlení

$$\vec{a}_S = \frac{d}{dt}(\vec{v}_P + \vec{\omega} \times \vec{s}) = \vec{a}_P + \dot{\vec{\omega}} \times \vec{s} + \vec{\omega} \times (\vec{\omega} \times \vec{s}), \quad (2.34)$$

kde se již vyskytuje zrychlení \vec{a}_P bodu P . Dosazením do rovnice (2.31) dostanu rovnice silové rovnováhy vztahující se k bodu P

$$\mathbf{m}\vec{a}_P + \dot{\vec{\omega}} \times \mathbf{m}\vec{s} + \mathbf{m}(\vec{\omega} \times (\vec{\omega} \times \vec{s})) = \vec{F}_P. \quad (2.35)$$

Dle [6] je možno rovnici momentové rovnováhy (2.32) přepsat do tvaru pro referenční bod P způsobem

$$\vec{s} \times \mathbf{m}\vec{a}_S + \mathbf{I}_S \dot{\vec{\omega}} + \vec{\omega} \times \mathbf{I}_S \vec{\omega} = \vec{M}_P, \quad (2.36)$$

do které mohu opět dosadit rovnici (2.34) a po úpravách získat tvar

$$\vec{s} \times \mathbf{m}\vec{a}_P + \mathbf{I}_P \dot{\vec{\omega}} + \vec{\omega} \times \mathbf{I}_P \vec{\omega} = \vec{M}_P, \quad (2.37)$$

kde \mathbf{I}_P je matice setrvačnosti vzhledem k bodu P vypočtená z \mathbf{I}_S podle Steinerovy věty

$$\mathbf{I}_P = \mathbf{I}_S - \mathbf{m}\vec{s}^2, \quad (2.38)$$

a podobně jako (2.6) je \vec{s} antisymetrickou maticí složek vektoru \vec{s} .

Obě rovnice přepíšu maticově do tvaru

$$\begin{bmatrix} \mathbf{m} & \mathbf{m}\mathbf{s}^T \\ \mathbf{m}\mathbf{s} & \mathbf{I}_P \end{bmatrix} \begin{bmatrix} \vec{a}_P \\ \dot{\vec{\omega}} \end{bmatrix} + \begin{bmatrix} \mathbf{m}\vec{\omega} \times (\vec{\omega} \times \vec{s}) \\ \vec{\omega} \times \mathbf{I}_P \vec{\omega} \end{bmatrix} = \begin{bmatrix} \vec{F}_P \\ \vec{M}_P \end{bmatrix} = \vec{Q}, \quad (2.39)$$

což zapíšu symbolicky jako

$$\hat{\mathbf{I}}_P \vec{\alpha} + \vec{\beta} = \vec{Q}, \quad (2.40)$$

kde $\hat{\mathbf{I}}_P$ je matice kompozitní setrvačnosti tuhého tělesa [6].

Soustava rovnic (2.39) může být vyjádřena v libovolném systému, ale musí být zapsána konzistentně. Jelikož matice setrvačnosti \mathbf{I}_P je konstantní v lokálním systému spjatém s platformou a vektory $\vec{\omega}$, $\dot{\vec{\omega}}$ získáme pomocí rovnic (2.4), (2.18) také nejprve v lokálním systému, zůstává nám z levé části soustavy rovnic (2.39) v globálním systému pouze \vec{a}_P , které bude později i řídicím systémem generováno právě globálně. Vektor $\vec{\alpha}$ tedy v jeho horní části doplním transponovanou maticí rotace pro transformaci vektoru \vec{a}_P do lokálních souřadnic, ale ještě předtím od vektoru \vec{a}_P odečtu vektor gravitačního zrychlení \vec{g} (který je totiž v globálních souřadnicích zřejmý a konstantní). To z pravých stran soustavy rovnic učiní vektory sil \vec{F}_{PL} a momentů \vec{M}_{PL} jakožto výslednici pouze sil v lanech a nebudu se již dále muset zabývat potenciální gravitační silou, takže

$${}^2\vec{\alpha} = \begin{bmatrix} \mathbf{S}_{12}^T (\vec{a}_P - \vec{g}) \\ {}^2\dot{\vec{\omega}} \end{bmatrix}, \quad (2.41)$$

a nyní již celou levou stranu transponuji do souřadnic globálních, ve kterých se bude nacházet nyní i vektor \vec{Q} , což je příhodné pro jeho následnou distribuci do zatížení lan, takže

$$\begin{bmatrix} \mathbf{S}_{12} & \emptyset \\ \emptyset & \mathbf{S}_{12} \end{bmatrix} \left(\begin{bmatrix} \mathbf{m} & \mathbf{m} {}^2\mathbf{s}^T \\ \mathbf{m} {}^2\mathbf{s} & {}^2\mathbf{I}_P \end{bmatrix} \begin{bmatrix} \mathbf{S}_{12}^T (\vec{a}_P - \vec{g}) \\ {}^2\dot{\vec{\omega}} \end{bmatrix} + \begin{bmatrix} \mathbf{m} {}^2\vec{\omega} \times ({}^2\vec{\omega} \times {}^2\vec{s}) \\ {}^2\vec{\omega} \times {}^2\mathbf{I}_P {}^2\vec{\omega} \end{bmatrix} \right) = \begin{bmatrix} \vec{F}_{PL} \\ \vec{M}_{PL} \end{bmatrix} = \vec{Q}, \quad (2.42)$$

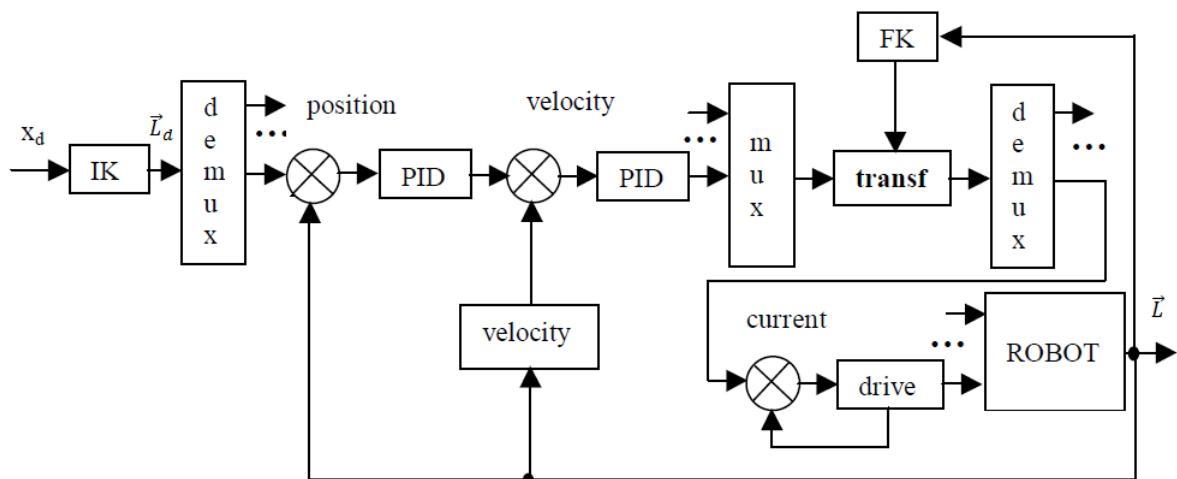
a tím dostávám výslednou soustavu pohybových rovnic ve tvaru vhodném pro výpočet inverzní dynamiky, což je situace, kdy nepočítám kinematické veličiny ze sil, ale kdy jako vstup slouží požadované a aktuální kinematické veličiny a z nich určím potřebné silové účinky (ty jsou zde vztaženy ke globálním osám a úhlovým rotacím okolo nich).

3. Řídicí model

3.1. Struktura řídicího modelu

K řízení lze využít klasických kaskádových PID regulátorů, které z požadované polohy, aktuální polohy a aktuální rychlosti určí požadované zrychlení, kterého se snaží dosáhnout skrze sílu aplikovanou pohony do robotického systému. U paralelních redundantních struktur ovšem nastává ta potíž, že při jakýchkoli nepřesnostech systému (jako jsou měření polohy, aplikovaná síla, rozměry mechanismu...) se v PID regulátorech tyto nepřesnosti kumulují a nastane tak případ, kdy se pohony začnou navzájem přetahovat, přičemž výsledný silový účinek na platformu zůstává stejný, ale pohony budou řídicím systémem zatíženy až za hranici svých limitů. Například článek [2] nabízí hned dva modely řízení (obr. 4 a obr. 5) pracující na principu pseudoinverze Jacobiho matice, resp. její transpozice.

První z nich (obr. 4) je tzv. decentralizovaný model řízení, který pomocí inverzní kinematiky přepočítá požadovanou polohu platformy $\vec{X}_d = [p_x, p_y, p_z, \varphi, \psi, \vartheta]_d^T$ na požadovanou polohu \vec{L}_d každého z motorů a za použití kaskádové PID regulace se zpětnou vazbou skrze aktuální polohy motorů spočítá požadované zrychlení pro každý motor.



obr. 4 - Decentralizovaný (lokální) model řízení [2] (FK = Forward Kinematics)

Vektor zrychlení motorů pak již úzce souvisí se vektorem sil v lanech (resp. i s elektrickými proudy jednotlivých motorů). Pomocí rovnice (2.25) je pak možno přepočítat požadované síly \vec{t}_d v lanech do výsledného požadovaného silového účinku \vec{Q}_d na platformu

$$\vec{Q}_d = \mathbf{J}^T \vec{t}_d = \mathbf{W} \vec{t}_d. \quad (3.1)$$

Poté stačí provést inverzní operaci, a tedy z požadovaného silového účinku \vec{Q}_d na platformu spočítat již vektor potřebných sil do motoru \vec{t} . Podstata celé transformace je právě v oné inverzní operaci (viz níže), protože matice \mathbf{W} nabývá rozměrů 6×8 a nelze tak použít klasickou inverzi \mathbf{W}^{-1} . Je nutné použít Moore-Penroseovu pseudoinverzi

$$\mathbf{W}^+ = \mathbf{W}^T (\mathbf{W} \mathbf{W}^T)^{-1}, \quad (3.2)$$

takže vektor sil do motoru spočítám jako

$$\vec{t} = \mathbf{W}^+ \vec{Q}_d = \mathbf{W}^T (\mathbf{W} \mathbf{W}^T)^{-1} \vec{Q}_d \quad (3.3)$$

a po dosazení rovnice (3.1) do rovnice (3.3) dostanu výsledný tvar transformačního bloku

$$\vec{t} = \mathbf{W}^+ \vec{Q}_d = \mathbf{W}^T (\mathbf{W} \mathbf{W}^T)^{-1} \mathbf{W} \vec{t}_d. \quad (3.4)$$

Takový tvar transformačního bloku se sice zdá jako redundantní, ale pseudoinverze má při řešení soustavy lineárních rovnic následující vlastnosti:

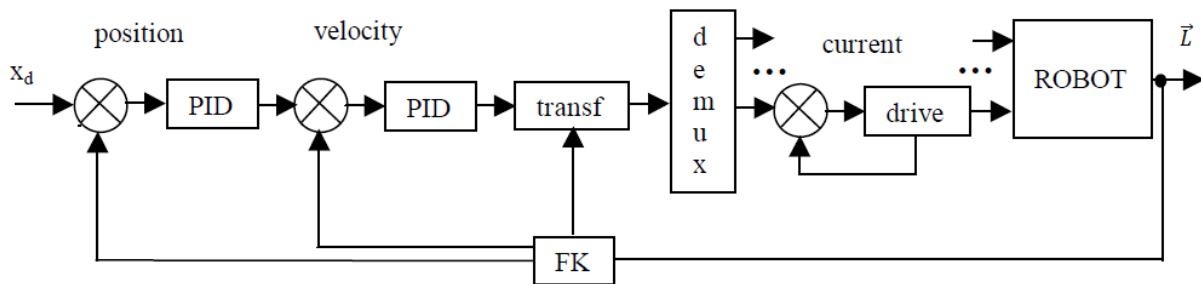
a) Pro libovolný rozměr vektoru neznámých \vec{t} spočítá řešení ve smyslu minimalizace Euklidovy normy $\|\mathbf{W} \vec{t} - \vec{Q}_d\|$.

b) Pokud má matice \mathbf{W} závislé sloupce (tedy více neznámých než rovnic, které vedou k nekonečně mnoho řešením, což je i náš případ), potom pseudoinverze určí řešení \vec{t} ve smyslu minimalizace Euklidovy normy $\|\vec{t}\|$.

Z toho vyplývá, že rovnice (3.4) transformačního bloku vždy nejdříve spočítá požadovaný výsledný silový účinek na platformu a z něj potom zpátky určí energeticky minimální potřebný silový účinek pro motory, takže nikdy nedojde k nadbytečnému vzájemnému přetahování jednotlivých pohonů.

Zároveň je potřeba pomocí dopředné kinematiky v reálném čase přepočítávat aktuální polohu motorů na aktuální polohu platformy, jelikož matice $\mathbf{W} = \mathbf{J}^T$ je funkcí právě polohového vektoru \vec{X} . Tím se více budu zabývat v kapitole 3.2.

Druhý z modelů (obr. 5) je tzv. centralizovaný model řízení, který také pracuje na principu pseudoinverze, ale liší se charakterem zpětné vazby.

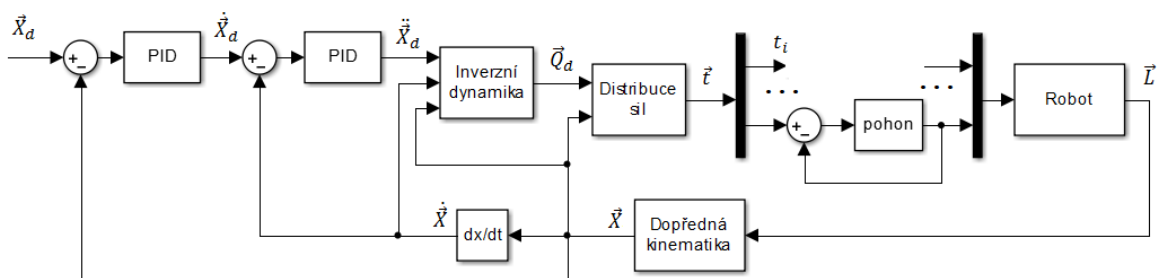


obr. 5 - Centralizovaný (globální) model řízení [2] (FK = Forward Kinematics)

Zatímco lokální model si požadovanou polohu platformy převádí na polohy jednotlivých motorů a na každý zvlášť aplikuje kaskádní PID regulaci, globální model pracuje v módu PID regulace přímo vektoru polohy platformy, k čemuž potřebuje kinematické veličiny z bloku dopředné kinematiky. U sériových struktur by to neznamenovalo příliš velkou změnu, ale u paralelních struktur (a navíc redundantně řízených) se pohony navzájem ovlivňují, a proto je lepší provádět regulaci centralizovaně v nezávislých souřadnicích \vec{X}_d .

U decentralizovaného řízení bylo snadné asociovat zrychlení pohonů se silami v lanech, protože směr změny délky lana a vektor síly v něm působící jsou kolineární. Nicméně zde u centralizovaného řízení je výstupem PID kaskádové regulace vektor $\ddot{\vec{X}}_d$, který obsahuje úhlové zrychlení ve směru Eulerových úhlů, zatímco vektor sil \vec{Q}_d se vztahuje k rotaci okolo globálních os, takže není možné poté již jednoduše použít transformaci pseudo inverzí, jako je tomu v rovnici (3.3). Jednou z možností je získat Jacobiho matici ${}^v\mathbf{J}$ (viz kapitola 3.2), nahradit jí původní matici \mathbf{J} v rovnici (3.1) a pomocí (3.3) spočítat výsledný silový účinek \vec{t} pro lana.

Místo toho ale v této práci modifikuji model globálního řízení (obr. 6), ve kterém využiji inverzní dynamiku, pomocí které přepočítám vektor $\ddot{\vec{X}}_d$ přímo na vektor sil \vec{Q}_d , což zároveň PID kaskádě ulehčí přechod od zrychlení k silovým účinkům (dodáním dynamických parametrů, které musím alespoň přibližně znát, a aktuální polohy se zrychlením). Blok inverzní dynamiky tak stejně jako PID kaskáda ke své funkci potřebuje aktuální kinematické veličiny platformy skrze dopřednou kinematiku. Také blok distribuce sil je vzhledem k lanovému charakteru manipulátoru odlišný a budu se jím zabývat v kapitole 3.3.



obr. 6 - Schéma centralizovaného modelu řízení

3.2. Dopředná kinematika

Zatímco inverzní kinematiku, která přepočítává polohu platformy na délku jednotlivých lan, lze snadno spočítat analyticky pomocí vazbových rovnic (2.22), dopředná kinematika je popsána více vazbovými rovnicemi než je neznámých. V reálné situaci, kdy je měření délek lan při experimentu ovlivněno chybami, by řešením analytického postupu prakticky vždy byla prázdná množina, takže je nutné použít numerickou metodu. V tomto případě je vhodným přístupem Newtonova iterační metoda, která je založena na Taylorově rozvoji vazbových rovnic (2.22) do prvního řádu v okolí bodu $\vec{X}^{(k)}$

$$\vec{f}(\vec{X}, \vec{L}) = \vec{f}(\vec{X}^{(k)}, \vec{L}) + \frac{\partial \vec{f}(\vec{X}, \vec{L})}{\partial \vec{X}^T} \Delta \vec{X}^{(k)} = 0. \quad (3.5)$$

Parciální derivace vazbových rovnic podle vektoru proměnných \vec{X}^T

$${}^v J = \frac{\partial \vec{f}(\vec{X}, \vec{L})}{\partial \vec{X}^T} \quad (3.6)$$

je Jacobiho matice ${}^v J$ ve tvaru (2.23), která se vztahuje k úhlovým rychlostem $\vec{\dot{\gamma}}$. Obecné určení takové matice je obtížné a v kapitole 2.1.4 jsem provedl výpočet pouze matice J vztahující se k úhlovým rychlostem $\vec{\omega}$ okolo globálních os. Nicméně díky matici B pro převod Eulerových úhlových rychlostí na úhlové rychlosti okolo lokálních os dle rovnice (2.4) a také pomocí rotační matice S_{12} pro převod vektorů z lokálního souř. systému do globálního dle rovnice (2.3) můžu matici J transformovat na ${}^v J$, přičemž výše zmíněné úpravy se musí týkat pouze úhlové části, takže

$${}^v J \vec{\dot{X}} = [{}^v J] \begin{bmatrix} \dot{P} \\ \dot{\gamma} \end{bmatrix} = [J] \begin{bmatrix} E & \emptyset \\ \emptyset & S_{12} \end{bmatrix} \begin{bmatrix} E & \emptyset \\ \emptyset & B \end{bmatrix} \begin{bmatrix} \dot{P} \\ \dot{\gamma} \end{bmatrix} = [J] \begin{bmatrix} E & \emptyset \\ \emptyset & S_{12} B \end{bmatrix} \begin{bmatrix} \dot{P} \\ \dot{\gamma} \end{bmatrix}, \quad (3.7)$$

a tedy můžu psát

$${}^{\gamma}\mathbf{J} = [\mathbf{J}] \begin{bmatrix} \mathbf{E} & \emptyset \\ \emptyset & \mathbf{s}_{12} \mathbf{B} \end{bmatrix}. \quad (3.8)$$

Nyní je možno z rovnice (3.5) vyjádřit výpočet pro k -tý přírůstek $\Delta\vec{X}^{(k)}$, ale jelikož

$${}^{\gamma}\mathbf{J}^{(k)} \Delta\vec{X}^{(k)} = -\vec{f}(\vec{X}^{(k)}, \vec{L}), \quad (3.9)$$

kde má matice ${}^{\gamma}\mathbf{J}^{(k)}$ rozměry 8×6 , není možno vyjádření učinit inverzí Jacobiho matice, nýbrž opět její pseudoinverzí. Pro výpočet přírůstku $\Delta\vec{X}^{(k)}$ v každém iteračním kroku tak dostanu tvar

$$\Delta\vec{X}^{(k)} = \left({}^{\gamma}\mathbf{J}^{(k)T} {}^{\gamma}\mathbf{J}^{(k)} \right)^{-1} {}^{\gamma}\mathbf{J}^{(k)T} \left(-\vec{f}(\vec{X}^{(k)}, \vec{L}) \right), \quad (3.10)$$

který zároveň minimalizuje Euklidovu normu rezidua $\|{}^{\gamma}\mathbf{J}^{(k)} \Delta\vec{X}^{(k)} + \vec{f}(\vec{X}^{(k)}, \vec{L})\|$. Vypočtený přírůstek je rozdílem mezi aktuální počáteční hodnotou $\vec{X}^{(k)}$ a tou pro následující iterační krok $\vec{X}^{(k+1)}$

$$\Delta\vec{X}^{(k)} = \vec{X}^{(k+1)} - \vec{X}^{(k)}, \quad (3.11)$$

takže výsledný tvar pro jednu iteraci výpočtu přibližného řešení dopředné kinematiky je

$$\vec{X}^{(k+1)} = \left({}^{\gamma}\mathbf{J}^{(k)T} {}^{\gamma}\mathbf{J}^{(k)} \right)^{-1} {}^{\gamma}\mathbf{J}^{(k)T} \left(-\vec{f}(\vec{X}^{(k)}, \vec{L}) \right) + \vec{X}^{(k)}. \quad (3.12)$$

Počet iterací k , po kterém ukončit výpočet, lze určit buď jako konstantní hodnotu nebo s ohledem na velikost přírůstku $\Delta\vec{X}^{(k)}$, který v blízkosti konečného řešení v průběhu výpočtu kvadraticky klesá, čímž lze přibližně odhadnout, jak moc se k -té řešení odchyluje od řešení pro $k = \infty$ a s rozumnou chybou výpočet předčasně ukončit. Pro Jacobiho matici s nezávislými sloupci pak Newtonova metoda vždy konverguje z kteréhokoli počátečního bodu $\vec{X}^{(k)}$.

3.3. Distribuce sil

3.3.1. Redundance lanových systémů

V kapitole 3.1 jsem se zabýval schématy řídicího modelu a pro PLM použiji model na obr. 6 vycházející z centralizovaného (globálního) modelu řízení. Řídicí model mimo jiné zahrnuje také transformační blok, který výsledný požadovaný silový účinek na platformu pomocí rovnice (3.3) distribuuje do jednotlivých pohonů tak, aby součet jejich účinků byl ekvivalentní a zároveň energeticky co nejméně náročný. Článek [2], ze kterého tato metoda

pramení, se sice zabývá paralelním manipulátorem, ale nikoli lanovým. To v praxi znamená, že výsledné rozdělení sil do pohonů může nabývat kladných i záporných hodnot, což s charakterem lan jakožto přenašečů silového účinku zřejmě nelze akceptovat. Stejně tak je také zřejmé, že pokud se platforma v ose z nebude nacházet pod úrovní všech navijáků, nýbrž nad některými z nich (což bude v našem případě celý pracovní prostor uvnitř krychlové konstrukce), potom tato metoda s principem nejmenších čtverců ve většině případů některým z těchto navijáků jistě přisoudí sílu tlakovou, v opačném případě by se totiž zbytečně přetahovaly s výše umístěnými pohony, a to se s minimem energetické náročnosti vylučuje.

Článek [5] nabízí analytické řešení tohoto problému. Vychází z faktu, že soustava rovnic

$$\mathbf{W}\vec{t} = \vec{Q} \quad (3.13)$$

vzhledem k rozměrům 6×8 matice \mathbf{W} obsahuje více neznámých než rovnic a tudíž existuje nekonečně mnoho řešení. Počet volitelných parametrů ovlivňujících výsledné řešení je potom právě rozdílem $r = m - n$. Budu uvažovat pouze situace (polohy platformy), kdy má matice \mathbf{W} plnou hodnotu $rank(\mathbf{W}) = 6$, tedy kdy Jacobiho matice \mathbf{J}^T nemá singulární sloupce a platformou je pomocí pohonů v dané konfiguraci stále možno pohybovat ve všech směrech a rotovat okolo všech os globálního systému 1. V takových případech budu vždy pracovat s $r = 2$ parametry $\lambda_{1,2}$. Výpočet parametrického řešení takového systému rovnic je potom

$$\vec{t} = \mathbf{W}^+\vec{Q} + \mathbf{N}\vec{\lambda} = \vec{t}_p + \mathbf{N}\vec{\lambda}, \quad (3.14)$$

kde \mathbf{W}^+ je Moore-Penroseova pseudoinverze dle rovnice (3.2) a $\vec{\lambda} = [\lambda_1 \quad \lambda_2]^T$ je vektor parametrů řešení. Matice \mathbf{N} nabývá rozměrů $m \times 2$ a představuje nulový prostor $\mathbf{N} = null(\mathbf{W})$, jehož dva sloupcové vektory tvoří ortonormální bázi, takže

$$\mathbf{N}^T\mathbf{N} = \mathbf{I}. \quad (3.15)$$

První sčítanec

$$\vec{t}_p = \mathbf{W}^+\vec{Q} \quad (3.16)$$

rovnice (3.14) tedy představuje řešení charakteru minima čtverců a druhý sčítanec $\mathbf{N}\vec{\lambda}$ představuje homogenní řešení rovnice (3.13), což znamená, že libovolná lineární kombinace vektorů matice \mathbf{N} s parametrickými násobnými koeficienty $\vec{\lambda}$ splňuje homogenní soustavu rovnic

$$\mathbf{W}\mathbf{N}\vec{\lambda} = \vec{0}. \quad (3.17)$$

Toto homogenní řešení je vůči vektoru pravých stran soustavy rovnic (3.13) netečné a lze jej přičíst k libovolnému řešení z nekonečné množiny možných řešení, takže

$$\mathbf{W}(\vec{t} + \mathbf{N}\vec{\lambda}) = \mathbf{W}\vec{t} + \mathbf{W}\mathbf{N}\vec{\lambda} = \mathbf{W}\vec{t} + \vec{0} = \mathbf{W}\vec{t} = \vec{Q}, \quad (3.18)$$

což se dá využít jako nástroj ke změně charakteru sil v pohonech.

V našem případě jde o nalezení všech vektorů \vec{t} , které spadají do množiny řešení soustavy rovnic (3.13)

$$\Sigma = \{\vec{t} \mid \mathbf{W}\vec{t} = \vec{Q}\}, \quad (3.19)$$

a jejichž prvky spadají do zvoleného intervalu možných zátěží pohonů

$$\Theta = \{\vec{t} \mid t_i \in [t_{min}, t_{max}], 1 \leq i \leq m\}, \quad (3.20)$$

takže hledáme průnik množin $\Lambda = \Sigma \cap \Theta$. Tento problém je možné převést na nalezení množiny všech možných vektorů parametrů $\vec{\lambda}$ splňujících soustavu dvojitéch nerovnic

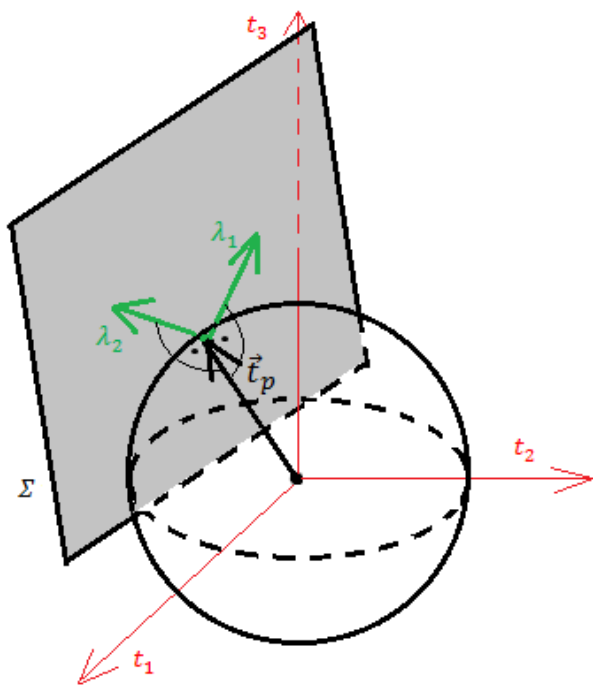
$$\vec{t}_{min} \leq \vec{t}_p + \mathbf{N}\vec{\lambda} \leq \vec{t}_{max}. \quad (3.21)$$

Pro výpočet matice \mathbf{W} byly použity vektory \vec{u}_i (resp. jednotkové vektory stejného směru), které představují manipulační lana směřující od platformy k navijáku. Tím pádem velikosti sil t_i působících zřejmě ve směru těchto vektorů nabývají kladných hodnot v případě, že jde o tahové účinky na platformu, a záporných hodnot v případě účinků tlakových. Interval možných zátěží pohonů s lany má tedy tvar $[0, t_{max}]$, nicméně pro obecnost budu dále pokračovat se značením $[t_{min}, t_{max}]$.

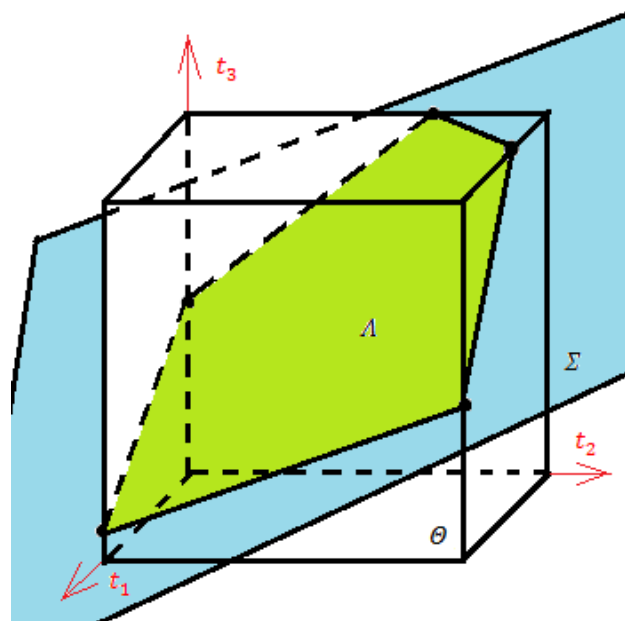
Přepíšu-li soustavu nerovností (3.21) na tvar

$$\vec{t}_{min} - \vec{t}_p \leq \mathbf{N}\vec{\lambda} \leq \vec{t}_{max} - \vec{t}_p, \quad (3.22)$$

dostanu soustavu nerovností vždy s jednou stranou známých sčítanců a druhou stranou neznámou. Interval $[\vec{t}_{min}, \vec{t}_{max}]$ je totiž neměnný pro celou úlohu a vektor minimálního řešení \vec{t}_p lze pro požadovaný silový účinek \vec{Q} a aktuální polohu platformy jednoznačně určit.



obr. 7 - Parametrická plocha Σ



obr. 8 - Dosažitelný polygon Δ

Celou problematiku hledání řešení vektoru sil \vec{t} si lze představit v m -dimenzionálním prostoru $\vec{t} \in \mathbb{R}^m$, ale pro jednoduchost je na obr. 7 znázorněn třídímenzionální prostor, ve kterém situaci analogicky modelují pomocí jedné rovnice o třech neznámých.

Ortonormální vektory N v takovém prostoru tvoří bázi 2D podprostoru. Množina Σ všech řešení soustavy rovnic (3.13) je v m -dimenzionálním prostoru právě rovinou o bázi N , takže veškeré myslitelné vektory \vec{t} spadající do množiny Σ směřují od počátku do některého bodu této roviny. Euklidova norma vektoru \vec{t}

$$\|\vec{t}\| = \sqrt{t_1^2 + \dots + t_m^2} \quad (3.23)$$

představuje jeho velikost, takže pomocí pseudoinverzí nalezeného řešení \vec{t}_p , které je právě minimem všech řešení dle této normy, určuje nejbližší bod roviny Σ od počátku – tedy průnik roviny Σ a m -dimenzionální hyperkoule se středem v počátku a zároveň tečné k této rovině.

To znamená, že vektor \vec{t}_p je kolmý na rovinu Σ , platí

$$N^T \vec{t}_p = \vec{0} \quad (3.24)$$

a tento bod roviny Σ je dle rovnice (3.14) zřejmě zároveň počátkem 2D souřadnicového systému roviny Σ s osami značenými parametry $\lambda_{1,2}$. Množina Θ v tomto m -dimenzionálním

prostoru tvoří jakousi hyperkrychli, jejíž stěny tvoří m dvojic protilehlých rovnoběžných rovin vzniklých omezením prostoru na interval $[t_{min}, t_{max}]$ pro každou souřadnici t_i . Je-li u lanových manipulátorů spodní hranice intervalu kvůli zajištění tahu stanovena na $t_{min} = 0$, jedná se o hyperkrychli, u které je m stěn tvořeno rovinami ohraničujícími první kvadrant m -dimenzionálního prostoru. Hledanou množinou $\Lambda = \Sigma \cap \Theta$ je tedy průnik právě roviny Σ s hyperkrychlí Θ a výsledkem je rovinný a zřejmě konvexní polygon Λ (obr. 8), jehož vnitřní plocha představuje v parametrické ploše Σ vektory vyjádřené pomocí všech parametrů $\vec{\lambda}$, které jsou bijektivním zobrazením množiny vektorů \vec{t} splňujících soustavu rovnic (3.13) a zároveň soustavu nerovnic (3.22). Tento polygon budu nadále nazývat polygonem dosažitelnosti.

Z výše psaného je zřejmé, že řešení dané úlohy nemusí ani při plné hodnosti matice W existovat, tedy že $\Lambda = \Sigma \cap \Theta = \emptyset$. V praxi si lze uvedení manipulátoru do takového stavu představit snadno. Buď jde o požadavek na tlakové silové účinky lan, jehož nejtriviálnější formou by byla statická poloha platformy nad úrovní všech navijáků, nebo jde o požadavek na příliš velké hodnoty přenášených sil, kterých nejsou pohony schopny, což může být zřejmě způsobeno příliš vysokou hmotností platformy, nepřiměřeným okamžitým zrychlením nebo třeba statickou polohou platformy u „stropu“ pracovního prostoru, kde síly v lanech pro polohy platformy ve směru osy z neúměrně rostou. Obě tyto příčiny překračující limity pohonů v jednom či druhém směru mohou být pro všechna lana libovolně vícenásobně nakombinovány. Ve skutečnosti k tomu ale často dochází při pouhém nestandardním požadavku na polohu a orientaci platformy v pracovním prostoru – i když se zdánlivě jedná o bezproblémovou situaci, 6 stupňů volnosti a 8 lan je již příliš složitý systém, aby dosažitelnost některých poloh byla na první pohled zřejmá.

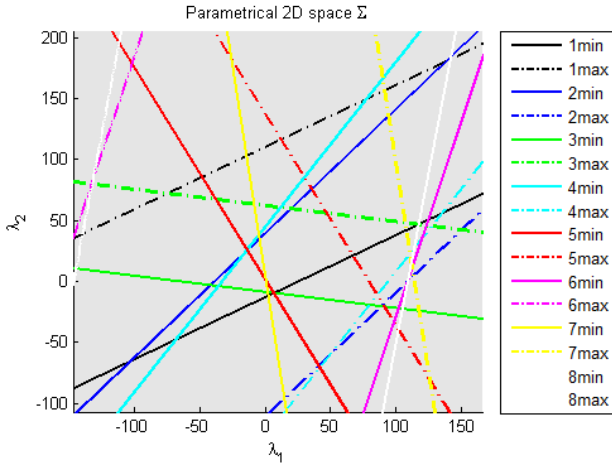
S představou popsanou v této kapitole je již zároveň možné snáz diskutovat nad redundancí u lanových systémů. Zásadní rozdíl oproti systémům schopným svými pohony vyvozovat tah i tlak je ten, že většina lanových manipulátorů v reálném světě musí počítat s gravitací. Nejtriviálnějším případem je klasická kladka, která díky tahu v laně umožní zvedat těleso do výše, ale pro opačný pohyb již využívá gravitačního zrychlení. Kdybychom gravitaci odmysleli – např. jednoduše uvažováním kladky pro horizontální pohyb tělesem – potřebovali bychom pro kladkový systém o 1 stupni volnosti 2 lana – každé pro pohyb resp. zrychlení tělesa do jednoho směru. Lehce složitější situace nastává při manipulaci tělesem v horizontální ploše. Stačí sice dvě lana nevycházející z jednoho místa, abychom dokázali

vyvodit silový účinek v obou osách, ale je zapotřebí ještě třetího vhodně umístěného lana, abychom tělesem dokázali pohybovat v obou směrech obou os. Tímto způsobem lze indukovat problematiku pro libovolný počet stupňů volnosti n , který tak bude vždy potřebovat $m = n + 1$ vhodně umístěných lan, čímž se manipulátor stane právě plně zavazbeným a teprve každé další lano již bude znamenat redundantní pohon. Termín „vhodné umístění lan“ potom neznamená nic jiného, než odkaz na existenci netriviálního průniku množin $\Lambda = \Sigma \cap \Theta$. V případě $m = n + 1$ pohonů se sice množina Σ stává pouhou přímkou, nicméně při nenáročných požadavcích na polohu a zrychlení je reálná šance, že tato přímka protne první kvadrant m -dimenzionálního prostoru a to v rozumných mezích.

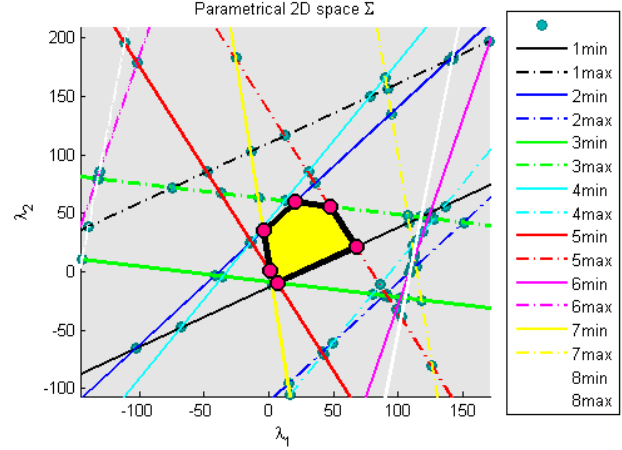
Důvodů, proč jsem pro svou práci zvolil systém o 6 stupních volnosti s 8 pohony – tedy s právě jedním redundantním pohonem, je více. Prvním je výrazné rozšíření pracovního prostoru ve všech osách, protože průnik roviny s hyperkrychlí je výrazně pravděpodobnější než v případě přímky. Druhým důvodem je symetričnost umístění pohonů v prostoru a připevnění lan na platformě, jelikož u 8 pohonů lze snadno dosáhnout spousty symetrických kombinací. Zdá se sice, že další navyšování počtu pohonů by situaci buď opět zlepšovalo, nebo případně alespoň nezhoršovalo. Faktem ale je, že při vyšších počtech už by příliš rostla pravděpodobnost vzájemné kolize lan nebo jejich kontakt s některou částí platformy mimo svůj bod úchyty.

3.3.2. Efektivní využití redundantních pohonů

Abychom mohli polygon Λ v parametrické ploše Σ využívat, je nejprve třeba pochopit, co vlastně tento konvexní obrazec znázorňuje. Na obr. 8 je znázorněno, že polygon je v ploše Σ omezen limity minimálního a maximálního silového zatížení motoru $[t_{min}, t_{max}]$, takže pro každou osu t_i (resp. motor) existují dvě rovnoběžné roviny kolmé na danou osu. Průnik všech takových dvojic rovin s plochou Σ pro m -dimenzionální prostor je znázorněn na obr. 9. Pokud bychom každou nerovnici ze soustavy (3.22) přepsali na rovnici, dostaneme pro každý řádek soustavy jednobarevnou dvojici rovnoběžných čar – plnou pro minimum a čerchovanou pro maximum – ohraničující pás, do kterého musí spadat výsledné řešení. Průnik všech pásů jakožto řešení celé soustavy nerovnic (3.22) je pak na obr. 10 žlutě zvýrazněn právě polygonem dosažitelnosti.



obr. 9 - Soustava nerovnic



obr. 10 - Dosazitelný polygon Λ

Z rovnic (3.14) a (3.24) plyne, že vektor \vec{t}_p je kolmý na rovinu Σ a to v místě počátku souřadnicového systému $\vec{\lambda} = [0 \ 0]^T$, kde se tedy nachází energeticky optimální řešení pro obecnou úlohu. U lanového manipulátoru se ovšem toto optimum bude ve většině případů nacházet mimo polygon dosažitelnosti a je proto potřeba najít optimální řešení z množiny Λ . Hledáme tedy minimum Euklidovy normy (resp. bez újmy na principu – minimum druhé mocniny Euklidovy normy)

$$\min_t \|\vec{t}\|^2 \quad \text{tak, že } \vec{t} \in \Lambda. \quad (3.25)$$

S dosazením rovnice (3.14) provedu úpravy

$$\|\vec{t}\|^2 = \|\vec{t}_p + \mathbf{N}\vec{\lambda}\|^2 \quad (3.26)$$

$$\|\vec{t}\|^2 = \left(\sqrt{\sum_{i=1}^m (t_{p_i} + N_{i1}\lambda_1 + N_{i2}\lambda_2)^2} \right)^2 \quad (3.27)$$

$$\|\vec{t}\|^2 = \sum_{i=1}^m t_{p_i}^2 + 2t_{p_i}(N_{i1}\lambda_1 + N_{i2}\lambda_2) + (N_{i1}\lambda_1 + N_{i2}\lambda_2)^2 \quad (3.28)$$

$$\|\vec{t}\|^2 = \|\vec{t}_p\|^2 + 2\vec{t}_p^T \mathbf{N}\vec{\lambda} + \|\mathbf{N}\vec{\lambda}\|^2. \quad (3.29)$$

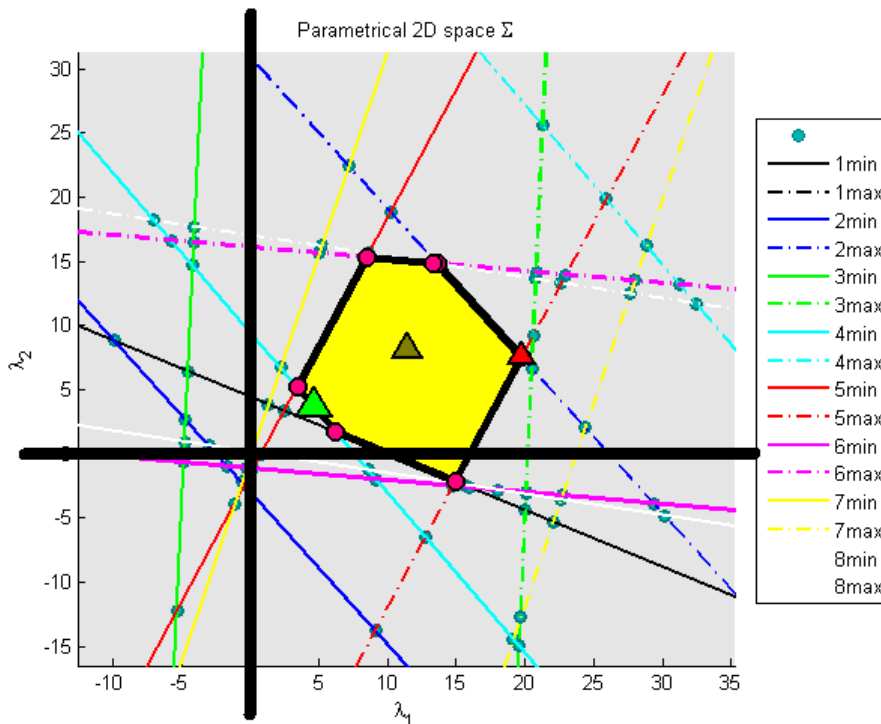
Jelikož je ale pro zadanou konfiguraci vektor \vec{t}_p konstantní a dle (3.24) je člen $2\vec{t}_p^T \mathbf{N}\vec{\lambda}$ nulový, lze s pomocí (3.15) převést problém nalezení řešení (3.25) na

$$\min_{\vec{\lambda}} \|\vec{\lambda}\|^2 \quad (3.30)$$

tak, že $\vec{t}_p + N\vec{\lambda} \in A$.

V praxi to znamená nalezení takového bodu polygonu, který je nejbližší počátku $\vec{\lambda} = [0 \ 0]^T$, což lze do výpočetního algoritmu snadno implementovat.

Stejně tak lze užívat požadavku co největšího silového působení v lanech, což má za důsledek hledání nejvzdálenějšího bodu polygonu od počátku. Takové maximum ale nemusí být jednoznačné a při symetriích konfigurace manipulátoru jich může být více, zatímco úloha minima má díky konvexnosti polygonu řešení vždy nejvýše jedno. Dále je možné zvolit některou z průměrových metod distribuce sil, jako např. aritmetický (případně) vážený průměr všech vrcholů polygonu nebo třeba co největší vzdálenost od všech hran polygonu apod. Řešení vzhledem k minimu, aritmetickému průměru a maximum jsou znázorněna na obr. 11 pomocí (po řadě) světle zeleného, tmavě zeleného a červeného trojúhelníku.



obr. 11 - Distribuce sil skrze polygon

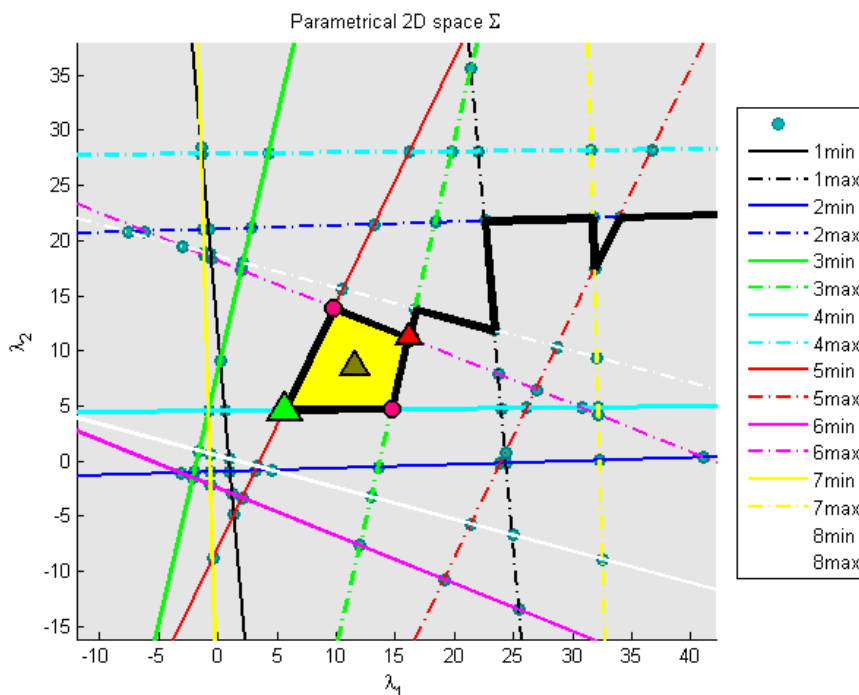
Zároveň je ale nutné dbát na spojitost takové metody mezi různými polohami manipulátoru. Zatímco minimum je v polygonu vždy jednoznačné a s každým dalším výpočetním cyklem modelu se posune jen o nejmenší nutnou vzdálenost v závislosti na velikosti časového kroku, u ostatních distribucí je tomu jinak. Například maximum se může během jediného časového kroku při jemné změně tvaru polygonu posunout o celou délku hrany na vedlejší vrchol. Při úloze průměru ze všech vrcholů může zase v dalším kroku snadno vzniknout zcela nový

vrchol nebo naopak některý zmizet – což lze pozorovat i na obr. 11, kde jsou na pozici [15,15] dva průsečíky těsně vedle sebe, které vychylují tmavě zelený trojúhelník mimo pozici, kde by byl jakožto průměr zdánlivě očekáván – čímž také dojde k velkému skoku v distribuci sil mezi časovými kroky.

Pro doplnění představy pouze uvedu, že pokud se platforma nachází pod úrovní všech navijáků – tedy je na všech lanech zavěšená – nachází se řešení energetického minima většinou uvnitř polygonu a $\min_t \|\vec{t}\|^2 \Rightarrow \vec{t} = \vec{t}_p$. V ostatních případech se minimum nachází na hraně polygonu nebo v některém z jeho vrcholů. V praxi se jedná o situace, kdy je jedno (pro hranu) lano či dvě (pro vrchol) lana možno zcela odlehčit a celou zátěž ponechat na lanech zbylých. Jednou z anomálií, která ovšem vzhledem k symetričnosti manipulátoru nemusí být až tak neobvyklá, je průnik 3 a více čar v některém z vrcholů polygonu. V takovém případě můžou na platformu působit kupříkladu pouze 4 lana a zbylá je možno prověsit.

3.3.3. Výpočet polygonu dosažitelnosti

Polygon dosažitelnosti musí být nalezen v každém kroku výpočetního cyklu modelu, což pro procesor představuje relativně velkou zátěž. Nejjednodušším řešením je zřejmě nalézt vzájemné průsečíky všech čar nerovností, kterých je v tomto případě 112. To znamená nutnost vyřešit 112 soustav dvou rovnic o dvou neznámých a vybrat pouze ta řešení, která splňují všechny nerovnosti ze soustavy (3.22). Pro některé hrubé nespojitě formy distribuce sil by tento krok byl postačující, v ostatních případech je potřeba průsečíky ještě seřadit, aby vznikl výsledný obrazec polygonu. To v řídicím modelu zajišťují skrze algoritmus, který nejdříve spočítá průměrnou hodnotu všech vrcholů polygonu a poté seřadí vrcholy podle úhlu, který svírá jejich spojnice s průměrnou hodnotou a osa λ_1 . Celý algoritmus pro výpočet polygonu je tak vcelku náročný, ale na druhou stranu je i robustní. Článek [5] nabízí rychlejší algoritmus, který zvolí a spočítá libovolný průsečík a z něj se pak po čarách pohybuje k dalším nejbližším, přičemž zkoumá, kolik nerovnic ze soustavy (3.22) je v každém bodě aktuálně splněno a dále se pohybuje právě tak, aby stávající vyhovující nerovnice zůstaly splněny a případně přibývaly další vyhovující nerovnice.



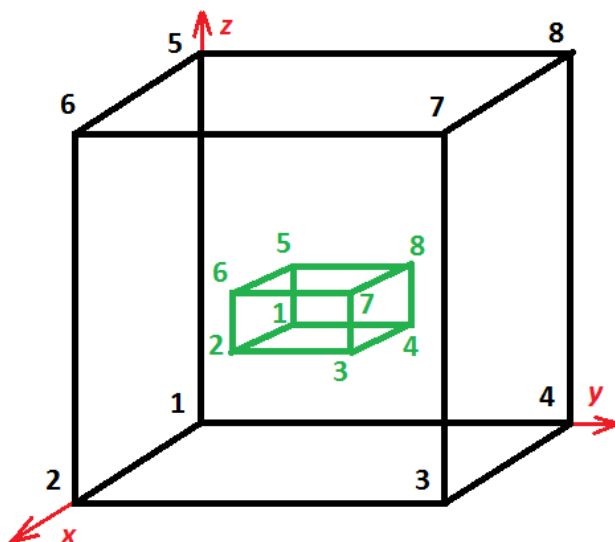
obr. 12 - Algoritmus stopování polygonu

Tímto postupem se algoritmus dopracuje až ke kýženému polygonu, který obkrouží, a po navštívení již dříve spočítaného průsečíku výpočet ukončí, jak je ilustrováno na obr. 12. Lze si všimnout, že stopovací algoritmus se tak zabýval skutečně jen zlomkem průsečíků (plus některými dalšími ležícími na navštívených čarách) a rychle dospěl k řešení. Rozhodovací proces, který v každém průsečíku určuje další postup grafem, je poměrně složitý [5]. Je u něj ale potřeba ošetřit degenerativní případy, kdy se v jednom průsečíku protíná 3 a více čar (v horším případě mohou některé z nich být díky symetrii systému i souběžné). Navíc je tento proces náchylnější na fatální chyby způsobené chybou numerického zaokrouhlování. Ve výsledku bylo obtížné dosáhnout rychlosti a zároveň robustnosti takového algoritmu a ve své práci proto budu používat algoritmus výpočtu všech průsečíků.

3.4. Optimalizace umístění lan

Kinematické vlastnosti celého manipulátoru závisí pouze na tom, ke kterým částem platformy jsou lana připevněna a ve kterých místech na okrajích pracovního prostoru jsou umístěny navijáky. Pro zachování maximální velikosti pracovního prostoru budu optimalizaci provádět s navijáky umístěnými v rozích krychlové konstrukce (v praxi jsou ovšem navijáky modulární a lze je upínat na libovolná místa konstrukce). Jedinými parametry jsou tak pozice upevnění lan na platformě, kterou uvažuji jako kvádr s úchyty libovolně umístěnými na jeho hranách.

Jelikož jde o značně diskrétní optimalizační úlohu, bude její princip spočívat v procházení jednotlivých možností. Vzhledem ke krychlovému pracovnímu prostoru a kvádrovému tvaru platformy lze očíslovat vrcholy těchto těles ve stejném charakteru – tedy pokud umístím malý kvádr doprostřed velké krychle, bude každému vrcholu krychle „odpovídat“ zřejmý vrchol kvádru, jak je znázorněno na obr. 13.



obr. 13 - Optimalizace - číslování vrcholů

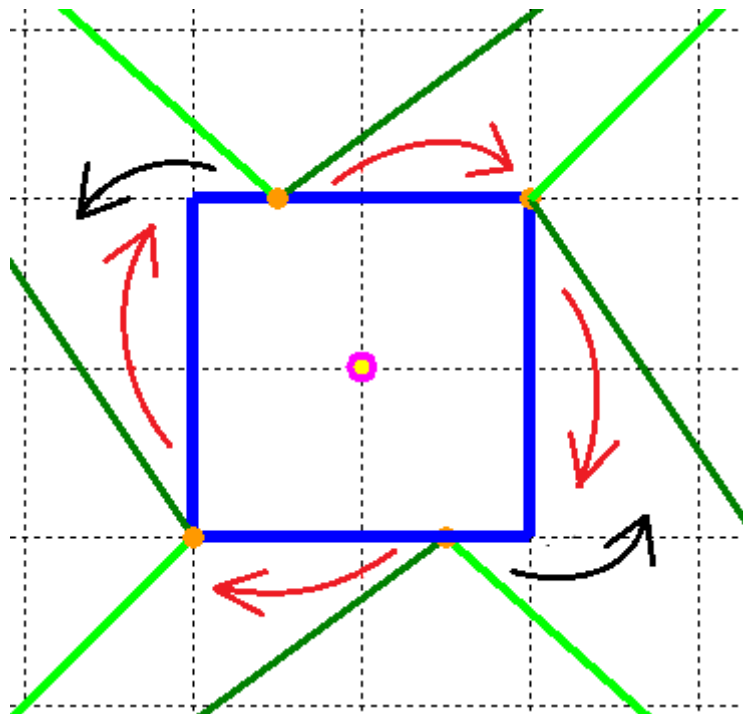
Princip optimalizační úlohy je projít všechna zobrazení, která bijektivně každému vrcholu krychlové konstrukce přiřadí právě jeden vrchol platformy. K procházení každé varianty pak stačí permutovat vektor celočíselných prvků 1 až 8. Každý prvek pak značí číslo vrcholu platformy, ke kterému je přivedeno lano vycházející z takového vrcholu konstrukce, jaká je pozice daného prvku ve vektoru. Každá taková konfigurace má pak druhý parametr, a to možný posuv o předem určenou vzdálenost po libovolné ze tří hran tvořících vrchol platformy. Cílová funkce se potom vztahuje k maximalizaci manipulovatelnosti systému a k minimalizaci možných kolizí lan navzájem či s platformou.

Manipulovatelnost lze snadno spočítat jako převrácenou hodnotu podmíněnosti Jacobiho matice

$$D = \frac{1}{\text{cond}(J)}. \quad (3.31)$$

Podmíněnost matice využívá singulárního rozkladu Jacobiho matice, kde singulární čísla vyjadřují míru ovladatelnosti platformy ve všech osách skrze pohony. Samotná podmíněnost

potom vrací poměr mezi nejvyšší a nejnižší hodnotou ze všech singulárních čísel (nabývá tedy hodnot v intervalu $(1, \infty)$), čímž poskytuje informaci o tom, jak moc se od sebe liší míry ovladatelnosti jednotlivých os navzájem. Požadavek na maximální manipulovatelnost poté zajišťuje, aby platforma byla ve všech osách ovladatelná v co nejpodobnější míře. Samotná manipulovatelnost ale nezávisí na žádné informaci o tom, že se jedná o lanový charakter manipulátoru. V každé kinematické konfiguraci tak uvažuje, že může skrze lana vyvíjet na platformu silový účinek jak tahový, tak i tlakový. Nás ale zajímá, jestli jsme v každé ose schopni s platformou pohybovat v obou směrech. Jak je znázorněno vrchním pohledem na platformu na obr. 14, rotaci kolem osy z (na obrázku je kolmá k nárysň) lze pomocí tahu v jednom směru zajistit skrze lepší páku (tedy větším momentem) než ve směru opačném.

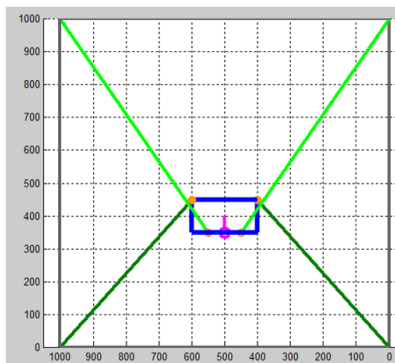


obr. 14 - Manipulovatelnost a lanový manipulátor

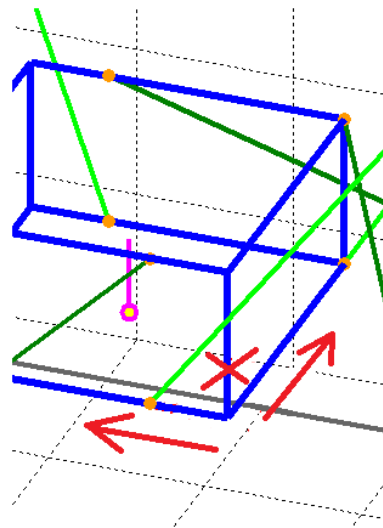
Abych se takovému nedostatku vyhnul, doplním cílovou funkci krom manipulovatelnosti také o velikost plochy polygonu dosažitelnosti. Ten je totiž ohraničen maximy a minimy možného namáhání lan. Pokud tedy při takové rotaci např. okolo osy z působí proti sobě dvě lana, mohou se teoreticky navzájem přetahovat až na hranici svých možností, aniž by s platformou rotovala. Nicméně jakmile proti sobě vzhledem k ose rotace působí přes různě velké páky, jedno z nich dosáhne svých maximálních limitů dříve než to druhé, což se projeví změnou tvaru polygonu – oproti symetrickému případu se polygon v kolmém směru některého z pásů

zploští. Požadavek maxima plochy polygonu tak původní manipulovatelnost doplní o ohled na ovladatelnost v obou směrech každé z os.

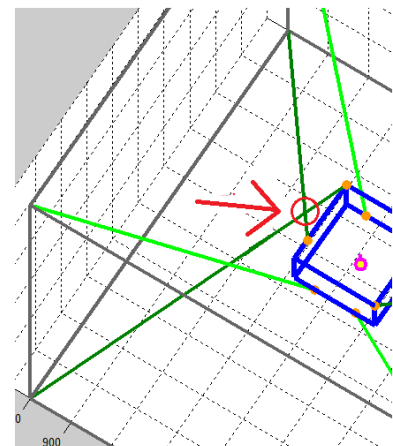
Při uvažování manipulátoru s 8 lany, které lze v různých kombinacích připevnit k 8 různým rohům platformy a v každém takovém rohu s úchytem ještě posouvat do 3 různých směrů, dostaneme přes 2,5 miliard možných kombinací. Výpočet Jacobiho matice, její podmíněnosti a navíc polygonu dosažitelnosti pro každou jednu kombinaci by průměrnému PC zabralo přinejmenším měsíce. Tento počet je proto třeba rozumně redukovat. Za prvé budu uvažovat, že lana z horních navijáků budou připevněna ke spodní části platformy a naopak (viz obr. 15), což je v jisté míře nutné pro obstojnou schopnost platformu naklápět. Za druhé zruším parametrický posuv úchytů lana na platformě v ose z (viz obr. 16), jelikož výška platformy je oproti zbylým dvěma rozměrům platformy značně menší.



obr. 15 - Zavěšení platformy

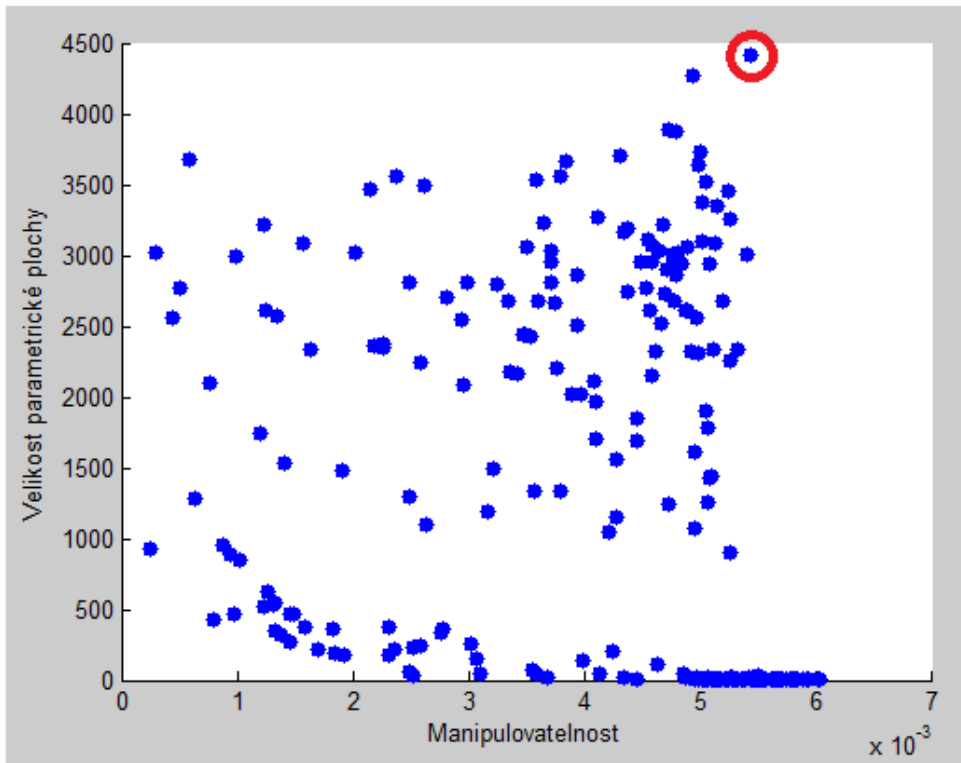


obr. 16 - Parametr posuvu

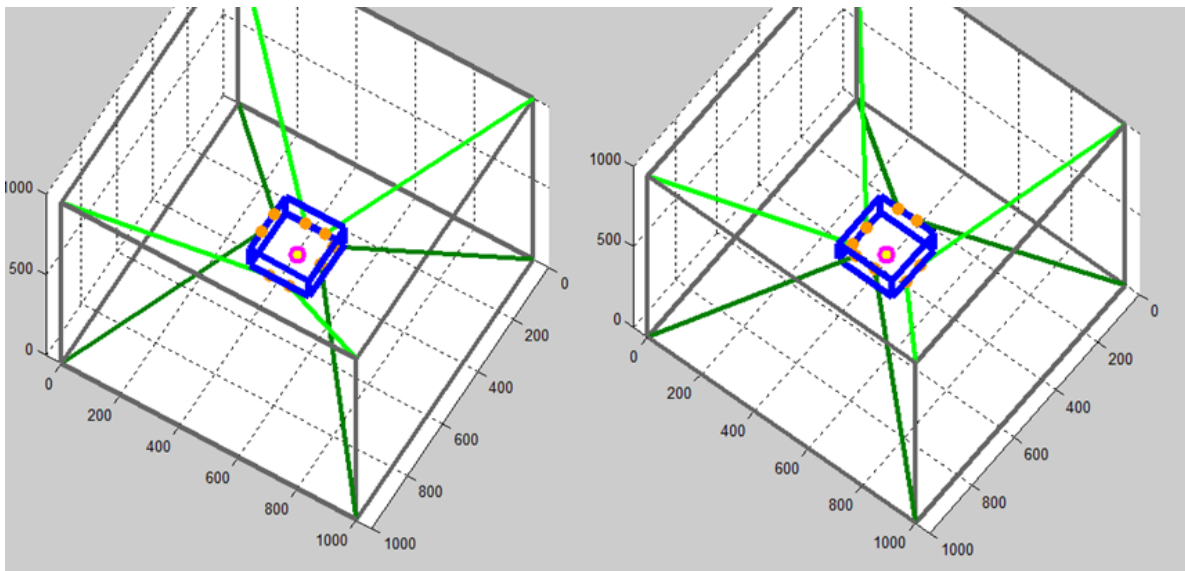


obr. 17 - Kolize lan

Tím se počet kombinací snížil na cca 4 miliony kombinací. Nakonec je vhodné vyřadit ty kombinace, ve kterých se již při neutrální poloze (všechna natočení – Eulerovy úhly – jsou nulová) některá lana navzájem kříží (viz obr. 17), případně procházejí skrz platformu apod. To vede již k výpočtově příznivým 6674 kombinacím. V každé konfiguraci jsem poté spočítal manipulovatelnost a velikost plochy polygonu. Výsledné hodnoty jsem zanesl do grafu na obr. 18. Ten lze snadno vyhodnotit a zpětně z parametrů získat odpovídající konfiguraci ilustrovanou na obr. 19 spolu se svým druhým symetrickým ekvivalentem.



obr. 18 - Optimalizace lanových konfigurací



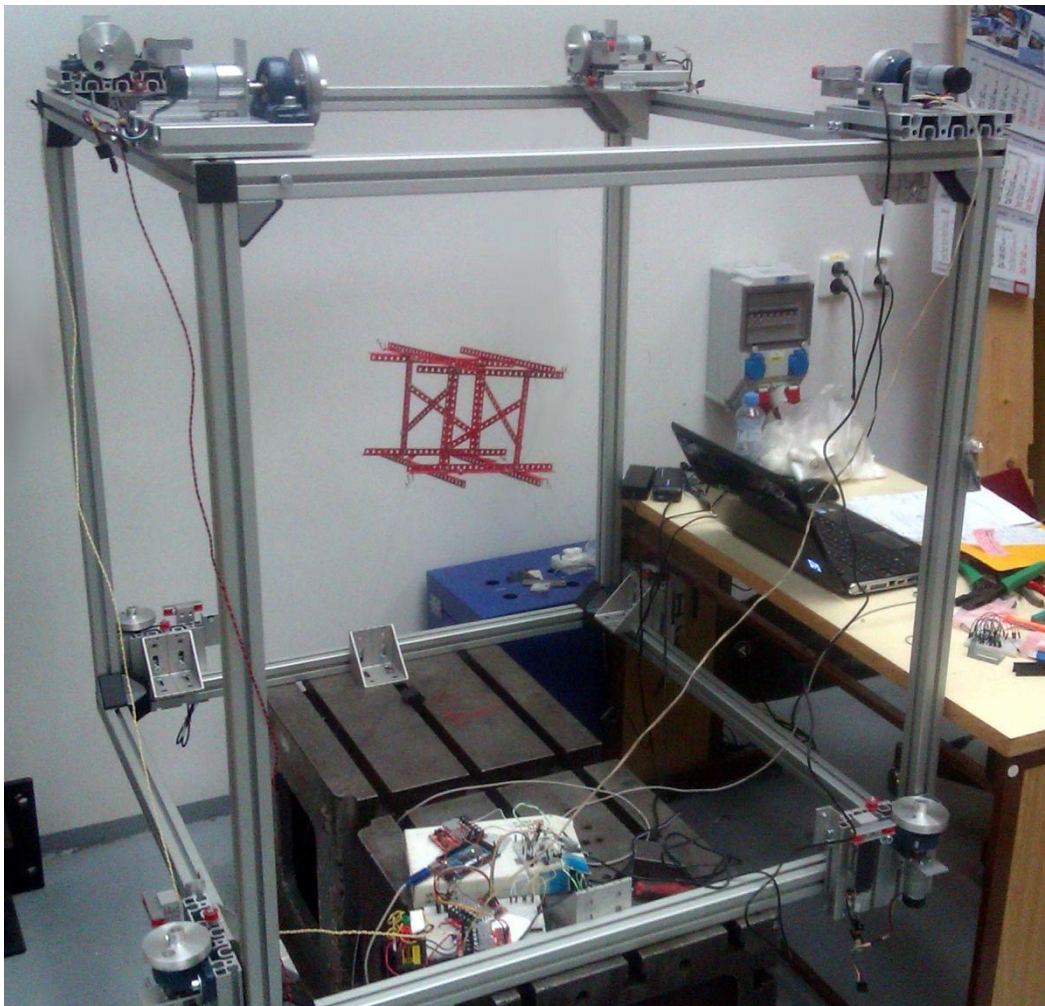
obr. 19 - Optimální konfigurace lan

4. Experimentální PLM

4.1. Topologie fyzického manipulátoru

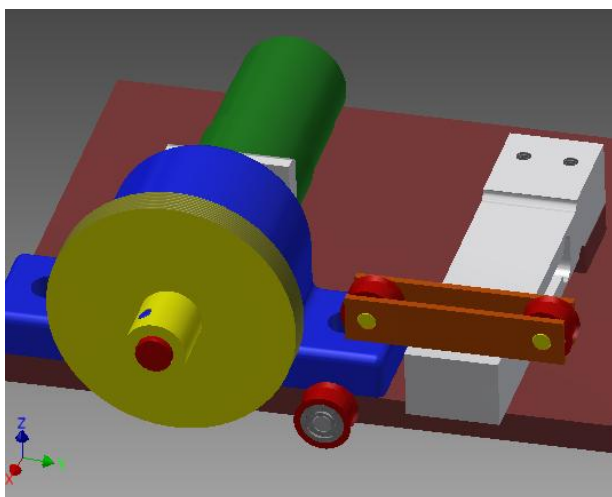
Posledním cílem této práce je sestavit reálný paralelní lanový manipulátor a implementovat do něj řídicí model. V rámci obecnějšího projektu jsem ve školní laboratoři sestavil experimentální manipulátor ve tvaru krychle o jednom kubickém metru (viz obr. 20).

Konstrukce se skládá z hliníkových profilů o rozměrech 40x40 mm. Z L-profilů stavebnice Merkur je sestavena také nesená platforma o celkových rozměrech 250x250x160 mm, ke které jsou na vhodných místech pomocí háčků připevněná lana.

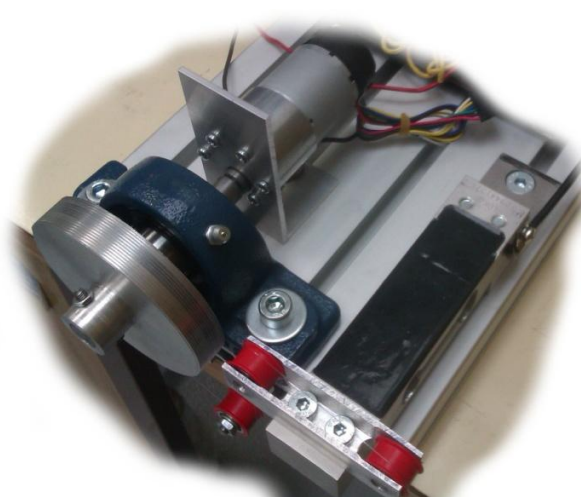


obr. 20 - Manipulátor – foto

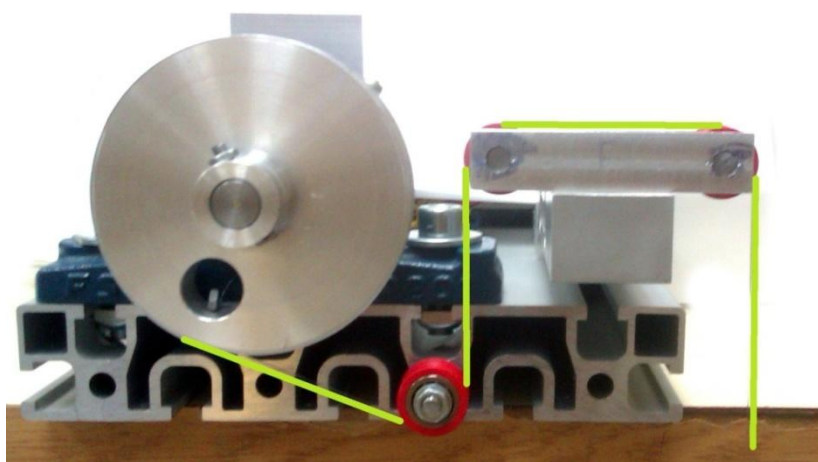
Navijáky lan (obr. 21, obr. 22 a obr. 23) jsou navrženy modulárně tak, aby bylo možné je přichytit v libovolném místě hlavní konstrukce manipulátoru a měnit tak charakter úlohy. Řídící model funguje na principu silového ovládnání motorů, a proto modul navijáku obsahuje kladkový systém připevněný na tenzometru, který je schopen měřit aktuální sílu přenášenou daným lanem a poskytovat tak zpětnou vazbu. Krom tenzometru obsahuje naviják také kvadratické enkodéry snímající aktuální polohu motoru, kterou lze přepočítat na délku odvinutého lana L_i . Tato informace ovšem neslouží pro přímé řízení motoru, protože ty lze řídit vždy buď pouze vzhledem k síle nebo vzhledem k poloze. Délky odvinutých lan ale řídicí model využívá pro výpočet polohy platformy v prostoru pomocí dopředné kinematiky (viz kapitola 3.2), na základě které v reálném čase počítá Jacobiho matici a poskytuje zpětnou vazbu pro PID regulátory řídicího modelu.



obr. 21 - Model navijáku



obr. 22 - Foto navijáku



obr. 23 - Naviják - vedení lana

Pracovní prostor výkonného bodu P manipulátoru je teoreticky celý objem vnější krychlové konstrukce a to s libovolnými rotacemi kolem všech os. V praxi je ovšem omezen několika faktory. Prvním z nich jsou kolize lan navzájem případně s platformou, což je jev, který nastává převážně při požadavcích na rotaci platformy o příliš velké úhlové výchylky. Druhým faktorem je kinematická dosažitelnost. Pokud by byl systém ovládán tuhými prvky schopných tahu i tlaku, nebylo by takové omezení příliš markantní, ale za použití lan snadno dojde k situaci, kdy manipulátor není v určité poloze schopen platformu zajistit ve všech 6 stupních volnosti. Poslední faktor je silového charakteru. Vzhledem ke gravitačnímu zrychlení totiž hmotnost platformy ovlivňuje maximální polohu vzhledem ke svislé ose z , jelikož průmět vektorů sil v lanech do osy z je ve vyšších polohách relativně malý a s každým dalším milimetrem výrazně klesá. Zatímco platformu není pomocí lan možné dostat do pozice nad všemi navijáky, nic systému nebrání ji spustit pod úroveň pracovního prostoru, ale dojde tak k výraznému snížení tuhosti vzhledem k její poloze.

4.2. Dimenzování

V této fázi je nutné se již zaměřit na dynamiku reálného manipulátoru a na základě požadavků zvolit správné komponenty – jako jsou motory, enkodéry a siloměry. Mezi konstantní parametry celého systému patří rozměry konstrukce, platformy a hmotnost platformy. Jelikož rozměry konstrukce laboratorního manipulátoru jsou $1 \times 1 \times 1$ m, vhodný tvar nesené platformy je kvádr o přibližných rozměrech $200 \times 200 \times 100$ mm. Platforma pak musí mít nejlépe co nejnížší hmotnost, nicméně samotná hmotnost není zdaleka jediným parametrem. Dalším z nich je výška, do které má být systém schopen platformu dostat, což je hmotností značně ovlivněno. V rámci probíhajícího projektu bude v budoucnu použita těžší platforma s možností připevnění další paralelní struktury o zatím neznámé hmotnosti, takže hrubý návrh obsahuje požadavek na schopnost systému vynést platformu o přibližné hmotnosti 1500 g do takové výše, aby se její spodní hrana nacházela cca 250 mm pod stropem pracovního prostoru (při nulových výchylkách natočení). Hodnotu síly pro lana pak lze snadno přepočítat (případně simulovat) a získat tak maximální zatížení pro jedno ze čtyř lan nesoucích platformu v takové statické poloze, které nabývá hodnoty 10 N. Posledním parametrem je potom maximální rychlost, kterou mají být navijáky schopny navíjet resp. odvíjet lano za předpokladu minimálního zatížení. Tu jsem stanovil na 300 mm/s, což je pro laboratorní účely nízkonákladového manipulátoru postačující. V tuto chvíli je možno hledat správnou kombinaci motoru, převodovky a navijecího bubnu pro navijáky. Jelikož je točivý moment

při volbě takové soustavy inverzní funkcí vzhledem k rychlosti otáček, lze tento problém řešit skrze klasickou pareto množinu.

Kromě limitů jednotlivých částí systému je také nutné stanovit určitou přesnost, se kterou má být systém schopen měřit jednotlivé veličiny. Vzhledem k nízkonákladovosti projektu není možné veličiny měřit s příliš extrémní přesností ani taková data v reálném čase dále zpracovávat. Zároveň je ale potřeba užít alespoň takové přesnosti, aby měřiče vzhledem k lanovému charakteru systému a šumu v elektrických obvodech do systému zaváděly relativně zanedbatelné chyby. Jako optimum jsem zvolil schopnost systému měřit síly v lanech s přesností alespoň 0,1 N a délku jejich odvíjení s přesností alespoň 0,1 mm.

Celý proces dimenzování ztěžuje fakt, že krom přesného měření musí být manipulátor přesný i sám o sobě. Například lana by se měla co nejméně protahovat, což znamená, že musí být z velice nepoddajného materiálu a/nebo nabývat určitého minimálního průměru vzhledem k vlastnostem materiálu. Dále je potřeba lano co nejpřesněji navíjet, čehož není možné dosáhnout klasickou jednoduchou drážkou, kde by se lano převíjelo přes sebe. Je proto nutné jej navíjet do závitové drážky. Ta ovšem musí mít určitou šířku, která závisí na průměru lana, a také navíjené délce, která zase plyne z velikosti obsluhovaného prostoru. Oba tyto parametry ovlivňují šířku navíjecího bubnu, která ale nesmí být příliš velká – je totiž složité a drahé vyrobit přesný naviják, který by disponoval sběrným očkem synchronně se pohybujícím podél závitů, aby zajistil přesné navíjení. Zvolil jsem proto pevný sběrný bod, který ovšem s sebou nese požadavek na ne příliš široký navíjecí buben, jinak by docházelo ke kosinové chybě při navíjení. Určitá minimální délka navinutého lana tak svazuje parametr šířky a průměru navíjecího bubnu dohromady. Velikost průměru bubnu ale zase ovlivňuje převodový poměr a tím i volbu převodovky atd.

4.3. Hardware

4.3.1. Výkonný a měřící hardware

Pohonnou jednotku navijáků tvoří motor Pololu 37Dx73L (viz obr. 24), který disponuje převodovkou v jeho přední části a kvadratickým enkodérem v části zadní. Na základě výsledků z dimenzování jsem z dostupné nabídky zvolil převodovku s poměrem 131,25:1, která spolu s navíjecím bubnem o průměru 80 mm zajišťuje maximální možnou tahovou sílu v lanech 11 N a maximální rychlost navíjení 335 mm/s. Výkonnostní parametry motoru sice umožňují vyvinout v lanech sílu až 44 N při elektrickém proudu 5 A, nicméně manipulátor

potřebuje vysokých hodnot sil dosahovat ve statických polohách a přílišné zatěžování kartáčových stejnosměrných motorů ve statických polohách by je mohlo poškodit. Výrobce doporučuje maximální proudové zatížení pro statické polohy je asi 25 % proudu maximálního, což – přepočítáno na sílu v lanech – odpovídá zhruba 11 N.



obr. 24 - Motor Pololu 131,25:1 37Dx73L

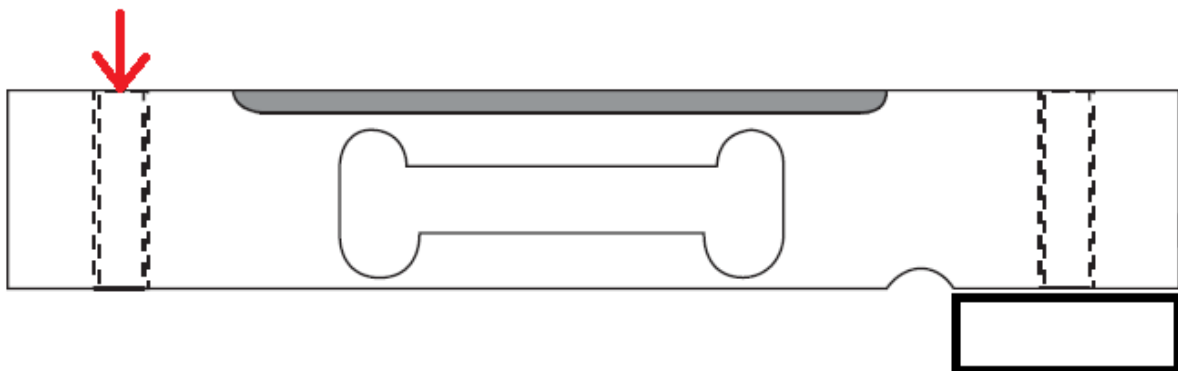
Jelikož motory disponují enkodéry ve své zadní části – tedy ještě před zpřevodováním rotačního pohybu – je třeba při volbě převodovky a navijecího bubnu dbát i na přesnost enkodérů. Ty měří s přesností 64 bodů na jednu otáčku hřídele motoru, což odpovídá 8400 bodům na otáčku výstupní hřídele převodovky. S průměrem bubnu 80 mm tak dostávám výslednou přesnost asi 0,03 mm, což spolehlivě splňuje minimální požadavek 0,1 mm.

Volba vhodných tenzometrů již není úloha příliš závislá na ostatních částech navijáku. Bohužel vysoká cena nedovoluje použít tenzometry určených přímo pro tah, na kterých by stačila pouhá jedna připevněná kladka. Zvolil jsem proto tzv. single-point tenzometry (viz obr. 25), které jsou namáhané na ohyb, nicméně je lze pro měření klasické síly použít pomocí kladkového systému umístěného tak, aby lano procházelo nad referenčním bodem tenzometru (viz obr. 26). Tento způsob kladkového systému má za následek, že tenzometr bude ve skutečnosti měřit sílu dvojnásobnou, než jsou lana skutečně namáhána. Navíc je třeba mít na vědomí, že ač pohony pracují se silou do 11 N, mohou při nestandardní situaci či chybě krátkodobě dosáhnout hodnot až čtyřnásobných a u tenzometru by v takovém případě nemělo

dojít k jeho destrukci. Jako vyhovující jsem proto zvolil tenzometr s rozsahem 10 kg – tedy přibližně 98,1 N – a s třídou přesnosti C3, tedy rozlišením 3000 bodů na celkový rozsah, což znamená přesnost 0,0327 N, která splňuje minimální požadavek 0,1 N.



obr. 25 - Tenzometr 10kg C3

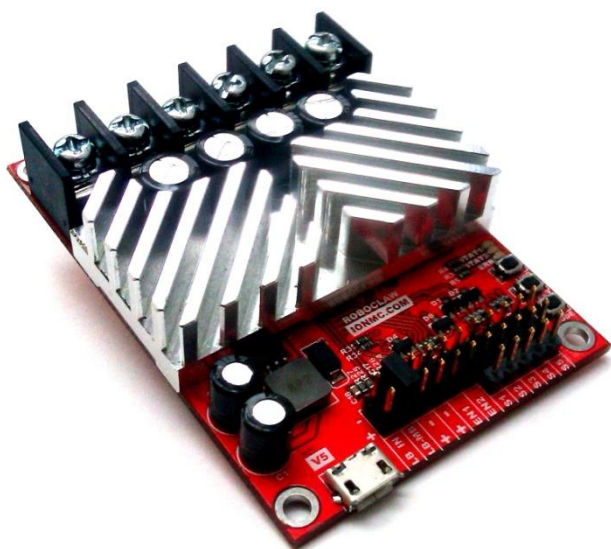


obr. 26 - Tenzometr - způsob namáhání

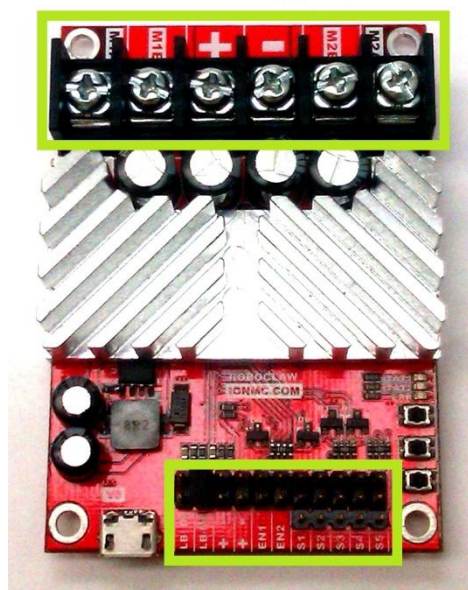
4.3.2. Výpočetní a řídicí hardware

Celý řídicí model systému obsahuje spoustu náročných procesů, jako jsou inverze matic, výpočet polygonu dosažitelnosti, případně Newtonovu iteraci apod., takže je potřeba výkonného výpočetního hardwaru. Low-cost charakter projektu směřuje k výběru některého z mikrokontrolérů/mikroprocesorů typu Arduino, BeagleBone, RaspberryPi či podobné desky. Ty ovšem buď nedisponují požadovaným výkonem a nebo nejsou dostatečně kompatibilní s vývojovým prostředím Matlab/Simulink. Hlavní část řídicího modelu proto poběží v klasickém PC v prostředí Matlab, což sice celý proces kvůli složitější komunikaci mírně zpomalí, ale na druhou stranu bude možno model interaktivně ovlivňovat.

Dále je potřeba napětově řídit stejnosměrné kartáčové motory a také čítat pulsy kvadratických enkodérů o frekvenci desítek kHz. Klasický low-cost přístup by byl vše řídit skrze některou z desek Arduino s přídavným Motor Shieldem a výstup enkodérů přivést na některý z digitálních pinů. Nicméně motory budou při náročnějších dynamických požadavcích potřebovat napájecí proud o několika ampérech a čítání pulsů kvadratického enkodéru v desítkách kHz skrze Arduino v interrupt módu již také není spolehlivé. Řešením je ale poloautomatická řídicí deska RoboClaw (viz obr. 27), která dokáže řídit motory s dostatečným výkonem a dekodovat enkodéry s vysokou frekvencí. RoboClaw se vyrábí v několika verzích lišících se v maximálním možném proudu dodávaném do motorů, který se pohybuje ve škále 5-60 A pro každý motor, přičemž umí obsluhovat dva motory najednou. Vstupní napětí desky se může pohybovat v rozsahu 6-34 V, což je vzhledem k 12V charakteru motorů vyhovující. Desku jsem označil jako „poloautomatickou“, protože ač neumožňuje vykonávání libovolného nahraného řídicího kódu (jako je tomu např. u Arduina), umí pracovat v několika přeprogramovaných módech. Tyto módy se liší převážně způsobem ovládání desky RoboClaw, ale dále je možno v nich nastavit vnitřní PID regulátory vzhledem poloze či rychlosti získané přímo z enkodérů, číst údaje o aktuální poloze enkodérů či hodnoty napájecího proudu motorů a další. Tuto univerzální desku tedy použiji pro zprostředkování pokynů z hlavního řídicího modelu do motorů a opačným směrem informací o hodnotách z enkodérů.

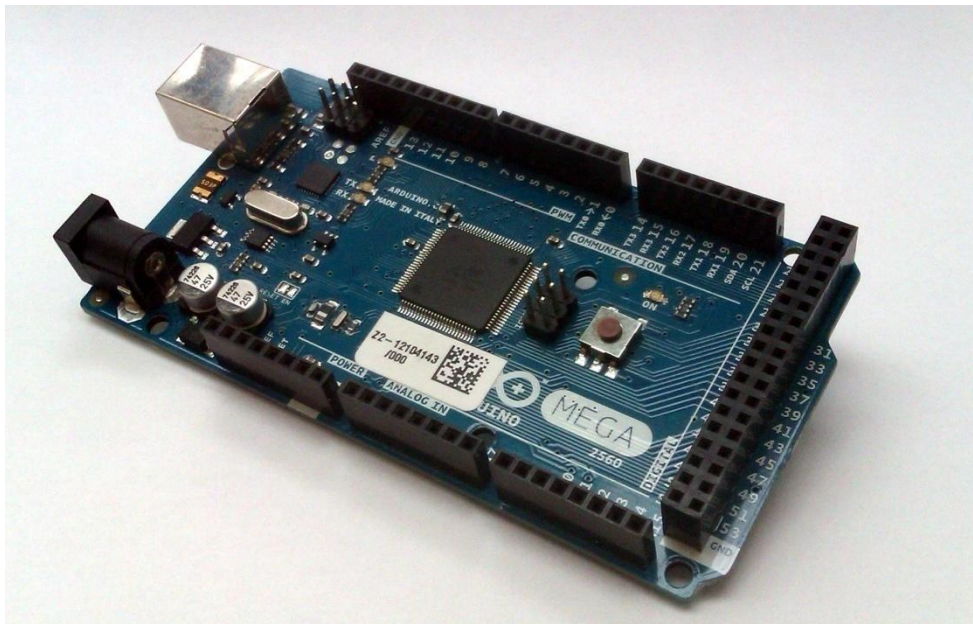


obr. 27 - RoboClaw 2x15A



obr. 28 - RoboClaw - terminály

Přímá komunikace PC s RoboClaw je ale obtížná a navíc motory nebudou mít každý ve svém řízení zpětnou vazbu vzhledem k poloze k enkodéru, ale k výstupu ze silového čidla. Za prvé tato zpětnovazební smyčka potřebuje vyšší frekvenci cyklů, než bude v hlavním řídicím modelu v PC, a za druhé je ještě nutné číst výstupní analogové hodnoty tenzometrů, takže je v každém případě potřeba ještě jednoho mezistupně zprostředkovávajícího tyto procesy. K tomu jsem se rozhodl využít vývojové desky Arduino Mega 2560 (viz obr. 29) disponující mikrokontrolérem o taktovací frekvenci 16MHz, digitálními piny, komunikačními kanály a hlavně analogovými vstupy vhodnými pro měření výstupních hodnot tenzometrů.

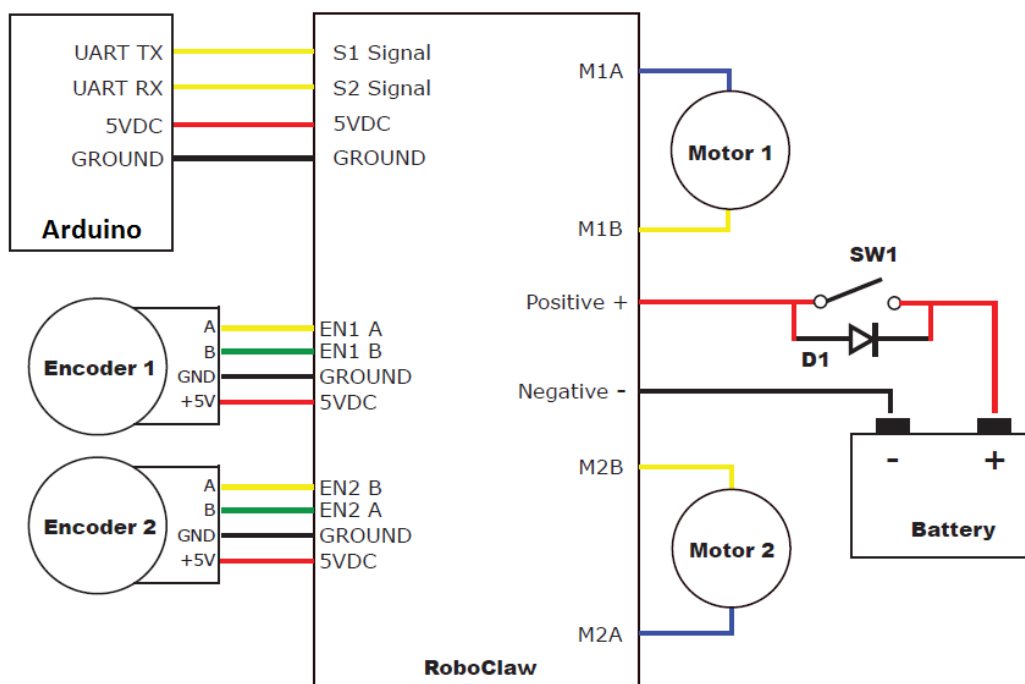


obr. 29 - Arduino Mega 2560

4.3.3. Zapojení a schémata

Pro tuto úlohu jsem vybral desku RoboClaw typu 2x15 A, jelikož krom dostatečného proudového limitu disponuje kvalitnějším pasivním chladičem než jeho slabší verze. Každá deska disponuje jedním šestinásobným šroubovacím terminálem (viz obr. 28) pro připojení hlavního napájení a přívodních kabelů k motorům, které jsou namáhané velkými proudy. Dále pak obsahuje sadu malých pinů pro 5V logiku – tedy pro příjem ovládacích signálů ve formě PWM či RC signálu, pro obousměrnou sériovou komunikaci a pro napájení a čtení enkodérů. Tato 5V logika může být snadno napájena přímo z hlavního zdroje napětí a nebo lze zajistit napájení externí. Připojení motorů přes šroubovací terminál je zřejmé a připojení enkodérů spolu s řídicí deskou Arduino je znázorněno na obr. 30. Na témže obrázku je také vidět, že jako zdroj energie je použita baterie se spínačem SW1 a usměrňovací diodou D1 na anodové

větvi. Důvodem přítomnosti spínače je fakt, že kladný vodič musí být při vypínání napájení odpojen vždy jako první, jinak by mohlo dojít k nestabilitě až poškození desky RoboClaw. Stejně tak je potřeba dbát na to, aby při vypnutém stavu desky RoboClaw nebyly vnější silou roztáčeny motory, protože energie v nich indukovaná by mohla poškodit desku RoboClaw. Pohybovat motory vnější silou v zapnutém stavu je možné, ale je potřeba, aby v motorech indukovaná energie měla možnost se vracet do baterie právě skrze usměrňovací diodu. Lehké nesnáze ovšem nastávají, pokud je deska napájena – nejčastěji spínaným zdrojem – ze sítě, kterým nelze energii do sítě zpět vracet. V takovém případě je možno desku RoboClaw předprogramovat do módu, kdy při roztáčení motorů vnější silou bude energii z nich přímo využívat na jejich brzdění – vzniká tak jakýsi „odpor“ vůči pohybu. Vzhledem k tomu, že motory budou mít silovou zpětnovazební smyčku, nebude toto bezpečnostní nastavení experiment nikterak ovlivňovat.

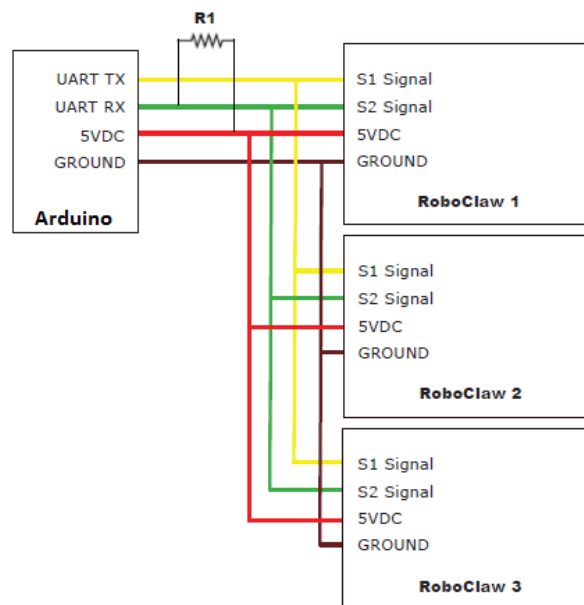


obr. 30 - RoboClaw – zapojení [8]

Jedna deska RoboClaw umí číst dva enkodéry a řídit dva motory, takže pro tento projekt bude potřeba 4 takových desek. Cílem sice není řídit motory vnitřní zpětnovazební smyčkou polohového charakteru, nicméně hodnotu enkodérů potřebujeme, aby bylo možno dopřednou kinematikou spočítat polohu platformy v prostoru. Informaci o hodnotě z enkodérů je proto nutné z desky RoboClaw získat pomocí sériové komunikace. Nevýhoda desky RoboClaw je,

že pokud je potřeba za jakýmkoli účelem použít sériovou komunikaci, budou tím obsazeny piny, které jsou jinak univerzální i pro ovládání motorů skrze PWM či RC signál, takže i pro motory bude nutné využít komunikace sériové, což proces mírně zpomalí, ale pro řízení motorů silovou zpětnou vazbou bude rychlost dostačující (viz kapitola 4.4.1) .

Vývojová deska Arduino Mega 2560 sice disponuje 4 linkami pro sériovou komunikaci, nicméně jedna z nich je použita pro spojení s řídicím počítačem. Vývojové desky RoboClaw lze ale snadno připojit ve větším počtu na jednu sériovou linku, protože každému z nich je možno přiřadit jednu až z 8 volacích adres. Stačí pouze společnou linku pro tok informací z RoboClaw do Arduina osadit pull-up rezistorem, jako je tomu na obr. 31, a skrze jednu desku Arduino pak pomocí adres postupně ovládat všech 8 motorů a číst všech 8 enkodérů.

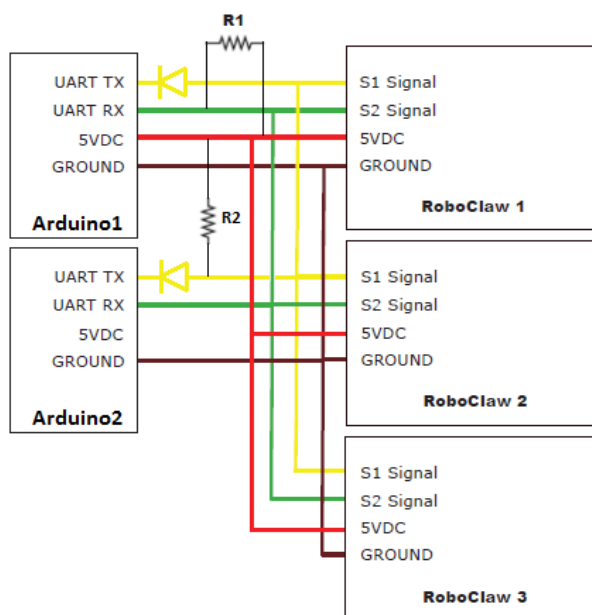


obr. 31 - RoboClaw - multi-unit zapojení [8]

Dostáváme tak situaci, kdy mezi jedním počítačem, 8 deskami RoboClaw a 8 tenzometrickými snímači zprostředkovává veškerou komunikaci jedna deska Arduino, což by mohla být zbytečná funkční zátěž. Samotný výpočetní výkon totiž netvoří, ale příliš mnoho sériové komunikace do všech možných směrů již ano. Vzhledem k tomu, že nejvyšší možná hodnota pro požadavek silového namáhání lana jakožto výstup hlavního výpočetního modelu bude vždy nezáporné číslo saturované na hodnotu 11 N (viz kapitola 4.2), vystačí na její popis 1 bajt přenesený komunikační linkou, čímž je dosaženo více než dostatečného rozlišení 0,043 N. Na druhou stranu hodnota enkodérů s rozlišením 0,03 mm bude výrazně vyšší a vyžaduje bajty 2, což vystačí na celý prostor uvnitř krychlové konstrukce. Pokud by vznikl

požadavek na spuštění platformy pod úroveň všech navijáků, bylo by třeba jeden bajt přidat nebo snížit rozlišení. Ať tak či tak – s přenosem polohové informace mezi PC a deskou Arduino nelze klesnout pod 2 bajty, přičemž samotná deska RoboClaw tak jako tak přenáší do desky Arduina bajty 4. Takže zatímco přenos informace pro pohon motorů je relativně nenáročný, mezi Arduinem a RoboClaw prakticky jednosměrný (RoboClaw vrací pouze jeden potvrzovací bajt) a navíc je pro pohon motorů potřeba co nejrychlejší zpětnovazební smyčky (přičemž čtení analogových vstupů Arduina pro získání informace z tenzometrů je proces velice rychlý), přenos informace o poloze je proces výrazně zdlouhavější a je potřeba pouze tak často, jak si jej hlavní řídicí program vyžádá. Takové množství přenášených informací v jednu chvíli pak způsobí příliš velké protahování časové periody pro PID regulátor, který tak může ztrácet stabilitu. Celá situace spěje k myšlence zprostředkovávat komunikaci pomocí dvou desek Arduino. Zatímco jedna se stará o rychlý cyklický chod silové zpětnovazební smyčky a s vysokou frekvencí tak zajišťuje, aby motory skrze lana skutečně působily požadovanou silou (kterou dostává z hlavního řídicího programu s výrazně nižší frekvencí), druhá má za úkol na pokyn PC pouze požádat desku RoboClaw o hodnoty enkodérů a přeposlat je zpět.

Takové rozdělení komunikačních a procesních služeb do dvou desek Arduino ale vede k problému, kdy je požadováno, aby obě desky Arduino byly schopny obousměrně komunikovat se čtyřmi deskami RoboClaw za použití jediné sériové komunikační linky. Ve skutečnosti jsou to problémy dva – jeden hardwarový a druhý softwarový. Když bychom totiž pouze paralelně napojili dvě desky Arduino na jednu sériovou linku, nemohla by žádná z nich odesílat informace skrze pin TX, protože druhá vždy v nečinnosti udržuje TX pin na logické jedničce. Tento problém lze ale řešit správně zapojenými usměrňovacími diodami a pull-up rezistorem na lince TX (viz obr. 32). Pokud totiž některá z desek Arduino bude chtít odesílat informace, bude díky propustnému směru diody moci uzemňovat signál na lince TX, přičemž logická jednička u druhé (neaktivní) desky bude zablokována závěrným směrem diody, a logickou jedničku bude vždy implicitně obstarávat pull-up rezistor.



obr. 32 - RoboClaw - multi-unit & multi-mcu zapojení

Správná úvaha by samozřejmě byla aplikovat tentýž princip i na TX (S2) pin všech desek RoboClaw, nicméně ty disponují tzv. multi-unit módem, ve kterém při činnosti (odesílání informací) pin S2 uzemňují, pokud jde o logickou nulu, a nechávají volně plovoucí, pokud jde o logickou jedničku, takže vše, co stačí ke správné funkci více paralelně zapojených desek RoboClaw, je pouze pull-up rezistor v doporučené hodnotě $1\text{ k}\Omega - 4,7\text{ k}\Omega$. Tím je zajištěno fyzicky funkční propojení všech desek Arduino i RoboClaw na jedné komunikační lince, nicméně to ještě nezaručuje, že se nebudou desky navzájem rušit, pokud budou vysílat ve stejnou dobu. S deskami RoboClaw je to snadné, jelikož vysílají pouze tehdy, jsou-li k tomu vyzvány. Pokud deska Arduino vyšle požadavek na nějakou informaci, počká, než se jí požadovaná hodnota vrátí, a pak teprve pokračuje dál v proceduře. I v případě, že jde pouze o jednosměrný pokyn pro běh motorů, vrací desky RoboClaw jeden potvrzovací bajt, že informaci přijaly (pokud je jim adresována). Problém tedy může nastat pouze tehdy, pokud by chtěly vysílat dvě desky Arduino najednou. To lze ale snadno ošetřit dvojitým propojením jejich digitálních pinů, po kterých si vždy sdělí požadavek o vysílání.

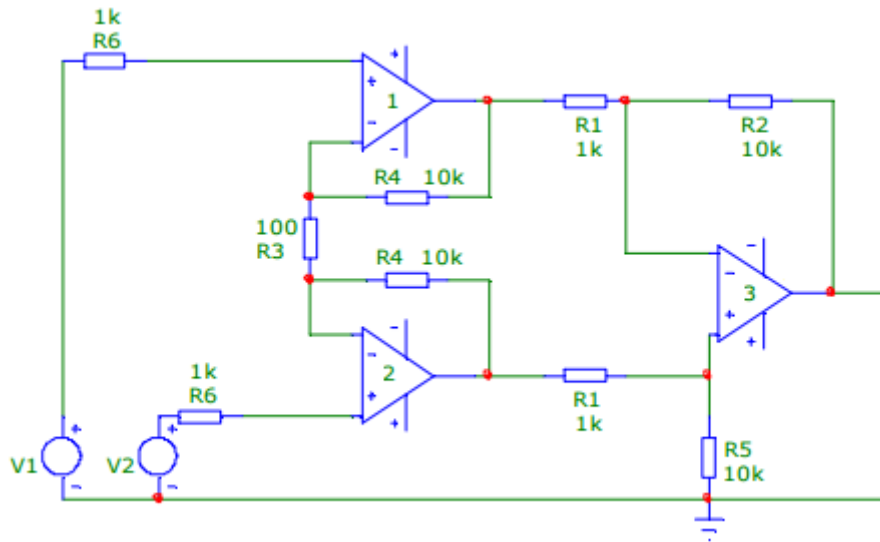
Mohlo by se zdát, že se dvěma deskami Arduino zprostředkovávající komunikaci bude situace stejná, jako s jednou, pokud si mají neustále žádat o uvolnění linky TX, nicméně klíčové ušetření času tkví v nezávislé komunikaci desek Arduino s PC. Celý proces pak lze efektivně sladit a postup může být např. následující:

- 1) Hlavní řídicí program z PC odešle silový požadavek do desky Arduino₁.
- 2) Arduino₁ nepřetržitě kontroluje buffer sériové linky s PC a při prvním příchozím bajtu předá desce Arduino₂ pomocí digitálního pinu informaci o tom, že nyní nebude potřebovat společnou komunikační linku.
- 3) Arduino₂ obsadí společnou linku požadavkem o aktuální poloze všech enkodérů, zatímco Arduino₁ přijímá informace z PC.
- 4) Jakmile Arduino₂ získá hodnoty ze všech enkodérů, předá desce Arduino₁ pomocí digitálního pinu informaci o uvolnění společné linky, na kterou Arduino₁ již vyčkává, protože byl zaneprázdněn přenášením menšího množství dat.
- 5) Arduino₁ obsadí společnou linku a pokračuje v rychlém PID řízení motorů pomocí silové zpětnovazební smyčky z analogových vstupů, zatímco Arduino₂ přeposílá informaci o poloze do PC, které již vyčkává.
- 6) Hlavní řídicí program v PC počítá nový silový požadavek, Arduino₁ zajišťuje skutečné dosažení aktuálně požadované síly a Arduino₂ vyčkává.

Existuje samozřejmě více variant a může se zdát, že ve výše uvedeném postupu je zbytečně mnoho vyčkávání. Hlavní řídicí program by např. mohl ještě před dokončením svého cyklu vyslat pokyn desce Arduino₂, aby si vyžádala společnou linku a začala sbírat informace o poloze enkodérů s předstihem. Potom ale záleží na správném načasování – příliš brzký pokyn by způsobil, že získané informace o poloze budou při použití již zastaralé, a nebo by mohla nastat nešikovná situace, kdy na moment uvolnění společné linky deskou Arduino₂ bude plynule navázána komunikace mezi PC a deskou Arduino₁, čímž by vždy vznikala příliš dlouhá proluka ve zpětnovazebním řízení motorů, a situace by se nelišila od modelu s jednou deskou Arduino.

Posledním hardwarovým prvkem je zesilovač analogového signálu tenzometrů. Silové měřiče jsou osazeny tenzometry zapojenými do můstku (viz obr. 34) s citlivostí 2 mV/V. To znamená, že celý měřicí rozsah (v tomto případě 10kg) odpovídá výstupnímu napětí 2 mV na každý 1 V napětí vstupního (napájecího). Měření tak probíhá v řádu μV a není možné jej přímo měřit pomocí analogového vstupu desky Arduino, která je schopna měřit rozsah 5 V s rozlišením 10 bitů. Deska Arduino sice disponuje referenčním pinem, na kterém externě přivedené napětí stanoví rozsah pro 10 bitové měření, ten ale akceptuje referenční napětí v rozsahu od 1 V do 5 V. Je proto nutné signál z tenzometrů pomocí operačních zesilovačů značně zesílit.

Nejdostupnější variantou je zesilovač vyrobit podle schématu [9] na obr. 33. To lze rozdělit na dvě části. První z nich se skládá z operačních zesilovačů 1 a 2 spolu s rezistory R_3 a R_4 a druhá z operačního zesilovače 3 a rezistorů R_1 a R_2 .



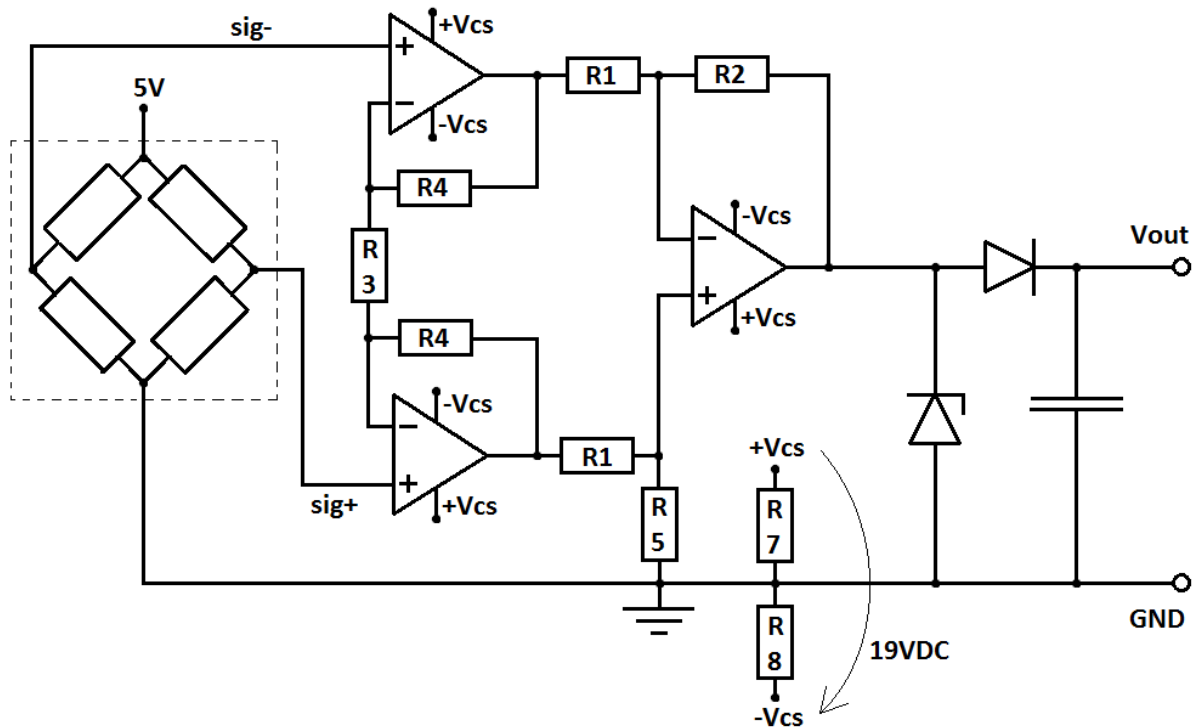
obr. 33 - Zesilovač - základní schéma [9]

Pomocí různých hodnot rezistorů lze měnit zesílení obou částí, které se pak spolu vynásobí do výsledného zesilovacího účinku, a to podle vztahu

$$K = \left(1 + \frac{2R_4}{R_3}\right) \frac{R_2}{R_1}. \quad (4.1)$$

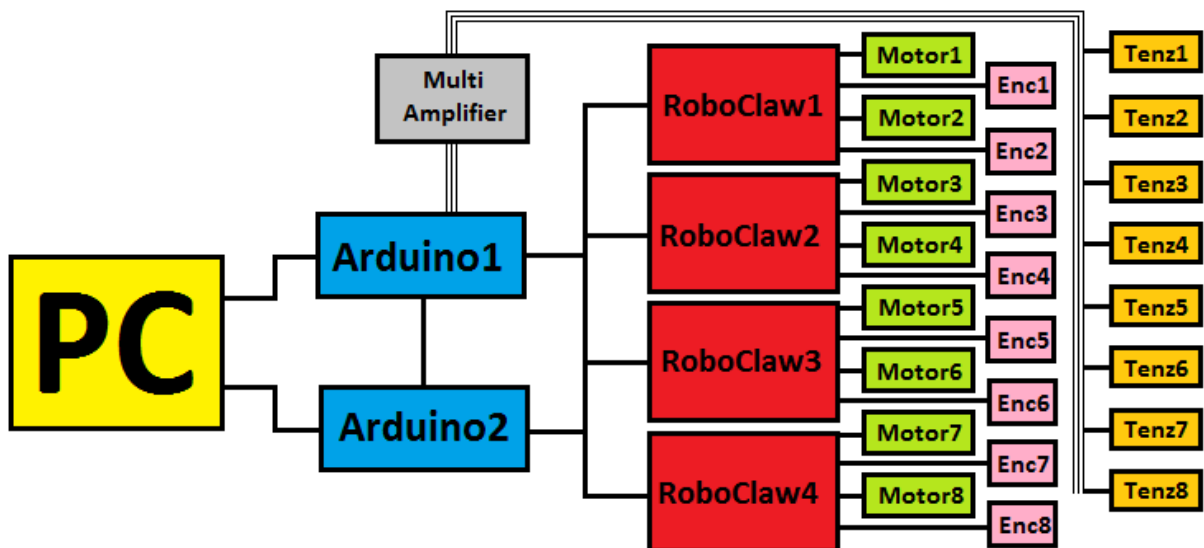
V prvním návrhu jsem např. zvolil hodnoty $R_1 = 2 \text{ k}\Omega$, $R_2 = 10 \text{ k}\Omega$, $R_3 = 100 \Omega$, $R_4 = 10 \text{ k}\Omega$, čímž získám výsledný zesilovací účinek $K = 1005$. Všechny operační zesilovače je ale nutné napájet symetrickým stejnosměrným napětím, které nám poskytne buď laboratorní zdroj, nebo klasický síťový adaptér spolu s děličkou napětí. Pomocí děličky napětí je tak možno vytvořit tzv. „umělý střed“, který bude představovat zem. Ať tak či tak, je potřeba dbát na to, aby symetrické napájecí napětí $\pm V_{cs}$ bylo o pár voltů (dle datasheetu) vyšší, než maximální výstupní napětí celého zesilovače, které je v tomto případě 5 V. Pro jistotu je pak nutné ošetřit výstup zesilovacího obvodu od předpětí, což lze snadno provést Zenerovou diodou zapojenou v závěrném směru s průrazným napětím 5,1 V, protože tolerance bezpečného přepětí analogových vstupů desky Arduino je 5,5 V. Symetricky napájené zesilovače jsou ale schopny produkovat záporné výsledné napětí. Stačí k tomu malý negativní offset některého ze zesilovačů, nebo lehké namáhání tenzometru v opačném směru. Jelikož by záporné napětí desku Arduino taktéž zničilo, je do obvodu vhodné zapojit

usměrňovací diodu v propustném směru. Nakonec je možné ještě obvod zbavit šumu analogovým filtrem s dolní propustí, k čemuž postačí jeden kondenzátor o vhodné kapacitě připojený mezi svorkami výstupu obvodu. Schéma celého zesilovacího obvodu je pak znázorněno na obr. 34.



obr. 34 - Zesilovač - celé schéma

Nyní je potřeba všechny komponenty vhodně pospojovat. Celý systém je pak znázorněn zjednodušeným topologickým schématem na obr. 35. Pro správnou komunikaci mezi komponentami a pro správné měření je nutné, aby všechny byly propojeny jednou společnou zemí. Desky Arduino jsou napájeny přes komunikační USB kabel z PC, desky RoboClaw jsou napájeny ze společného 12V zdroje s vysokým výkonem, zesilovač je napájen symetrickým napětím s umělou zemí ze síťového zdroje a enkodéry společně s tenzometry jsou po dvojicích (z důvodu limitů proudů v 5V logice desky RoboClaw) napájeny z každé desky RoboClaw. Nicméně pokud je pro všechna napájení použito spínaných zdrojů případně laboratorních zdrojů s oddělovacím transformátorem, jsou všechna napětí plovoucí, takže mezi nimi implicitně není žádná napěťová diference a je možno je společným zemnicím kabelem propojit.



obr. 35 - Schéma propojení HW

4.4. Testy a výsledky

Tato kapitola se bude zabývat ověřením základní funkčnosti reálného manipulátoru. Zprv je potřeba zjistit, zdali a jak je možno navijáky po hardwarové i softwarové stránce řídit silově, a zadruhé, je-li celý systém schopen mechanicky fungovat a platformu rozpochybovat.

4.4.1. Silové řízení navijáku

Jelikož bude paralelní manipulátor v budoucnu fungovat na principu silového řízení všech motorů, je nejprve nutné ověřit funkčnost jednoho modulu navijáku jako samostatného celku. Ten pomocí motoru pohání navíjecí buben se šroubovou drážkou pro lano, které je pak kladkovým systémem veden přes těleso tenzometru. Tím je možno získávat informace o aktuálním zatížení lana a použít je pro zpětnovazební smyčku řízení s PID regulátorem. K testu je potřeba jedna deska Arduino, jedna deska RoboClaw a jeden zesilovací obvod, přičemž čtení enkodérů v této situaci není třeba.

Jak pro silové řízení každého z motorů, tak pro účely celého budoucího řídicího systému je třeba užít diskretního PID regulátoru. Ve spojitém systému je rovnice PID regulátoru

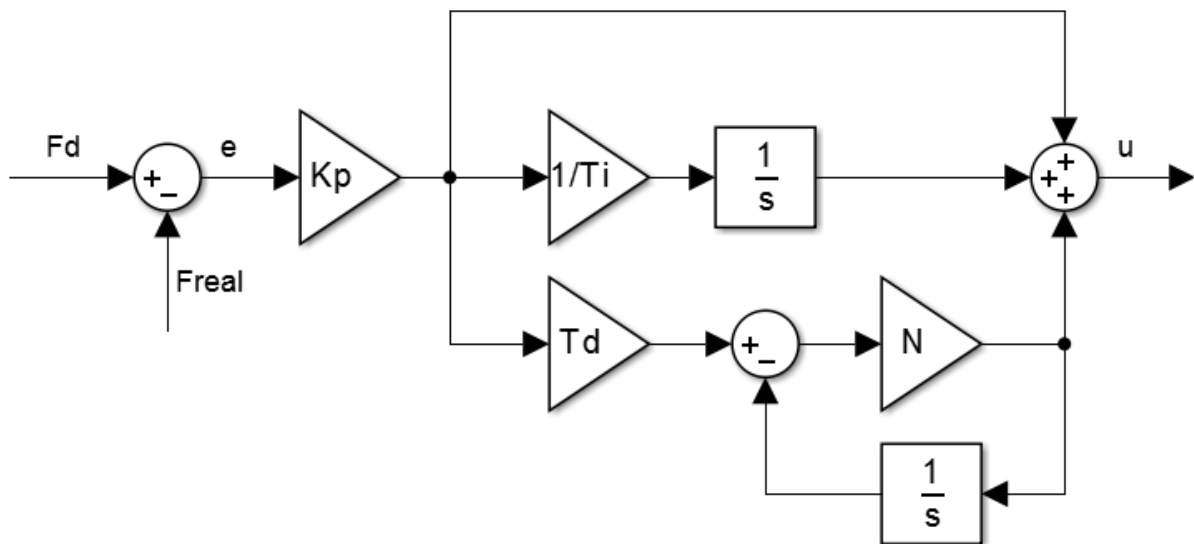
$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (4.2)$$

kde $e(t)$ je regulační odchylka, K_p , K_i a K_d jsou po řadě proporcionální, integrační a derivační konstanty a $u(t)$ je výstupní signál regulátoru. S použitím Laplaceovy transformace rovnici (4.2) přepíšu do tvaru spojitého obrazového přenosu

$$G(s) = K_p + \frac{K_i}{s} + K_d s, \quad (4.3)$$

přičemž z praktických důvodů [10], kdy je zpětnovazební signál často zašuměný (a tím tedy i regulační odchylka), se často používá derivační složka PID regulátoru ve tvaru filtru s dolní propustí a s jeho ladící konstantou N (viz obr. 36), takže obrazový přenos nabývá tvaru

$$G(s) = K_p + \frac{K_i}{s} + \frac{K_d N}{1 + \frac{N}{s}}. \quad (4.4)$$



obr. 36 - PID regulátor – schéma

Pro implementaci PID regulátoru do mikrokontroléru je nutné obrazový přenos (4.4) převést do diskretního tvaru. To jsem provedl pomocí Z-transformace a pro integrační i derivační složku regulátoru jsem zvolil zpětnou Eulerovu metodu tak, že

$$G(z) = K_p + \frac{K_i h z}{z - 1} + \frac{K_d N}{1 + \frac{N h z}{z - 1}}, \quad (4.5)$$

kde h značí časovou délku jednoho cyklu zpětnovazební smyčky. Pravou stranu rovnice (4.5) lze pak několika kroky upravit na tvar

$$G(z) = \frac{[K_p(1 + Nh) + K_i h(1 + Nh) + K_d N] + [K_p(-2 - Nh) - K_i h - 2K_d N]z^{-1} + [K_p + K_d N]z^{-2}}{[1 + Nh] + [-2 - Nh]z^{-1} + [1]z^{-2}}, \quad (4.6)$$

který lze substituuovat na

$$G(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{a_0 + a_1z^{-1} + a_2z^{-2}}. \quad (4.7)$$

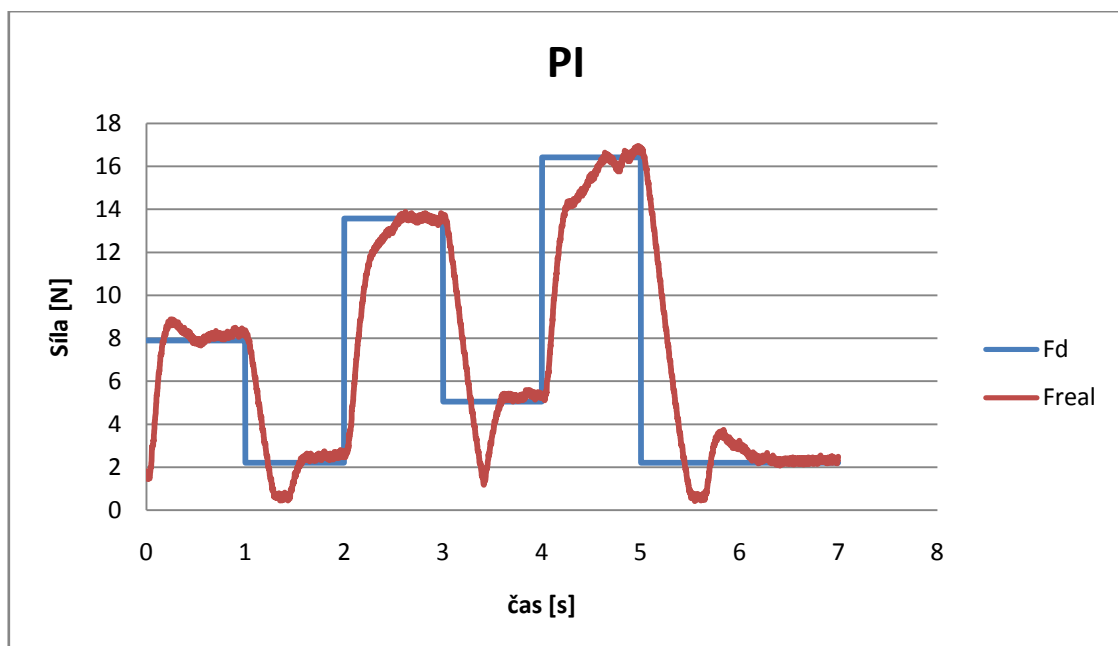
$a_0, a_1, a_2, b_0, b_1, b_2$ jsou konstantami obrazového přenosu $G(z)$ a pro výstupní hodnotu v časovém kroku k platí

$$u(k) = \frac{1}{a_0} [-a_1u(k-1) - a_2u(k-2) + b_0e(k) + b_1e(k-1) + b_2e(k-2)], \quad (4.8)$$

kde $u(k)$ a $e(k)$ jsou výstupy a regulační odchylky regulátoru v časovém kroku k . Pro PI regulátor bez derivační složky lze rovnici (4.8) také použít, ale konstanty budou vypadat následovně:

$$\begin{aligned} a_0 &= 1; & a_1 &= -1; & a_2 &= 0; \\ b_0 &= K_p + K_i h; & b_1 &= -K_p; & b_2 &= 0; \end{aligned}$$

Samotný test probíhal s poddajným lanem (navinutým podle obr. 23), na jehož konci bylo připevněno závaží o hmotnosti 1700g ležící na malé váze (pro kontrolu skutečných hodnot). Úloha řídicího programu nahraného na desce Arduino pak byla prostá. Každou sekundu měnil požadovanou sílu v rámci celého rozsahu, pomocí analogového vstupu četl zesílený signál z tenzometru a regulační odchylkou mezi těmito dvěma hodnotami pomocí PI regulátoru napěťově řídil motory s rozlišením 7 bitů v každém směru. Výsledky testu jsou graficky vykresleny na obr. 37.

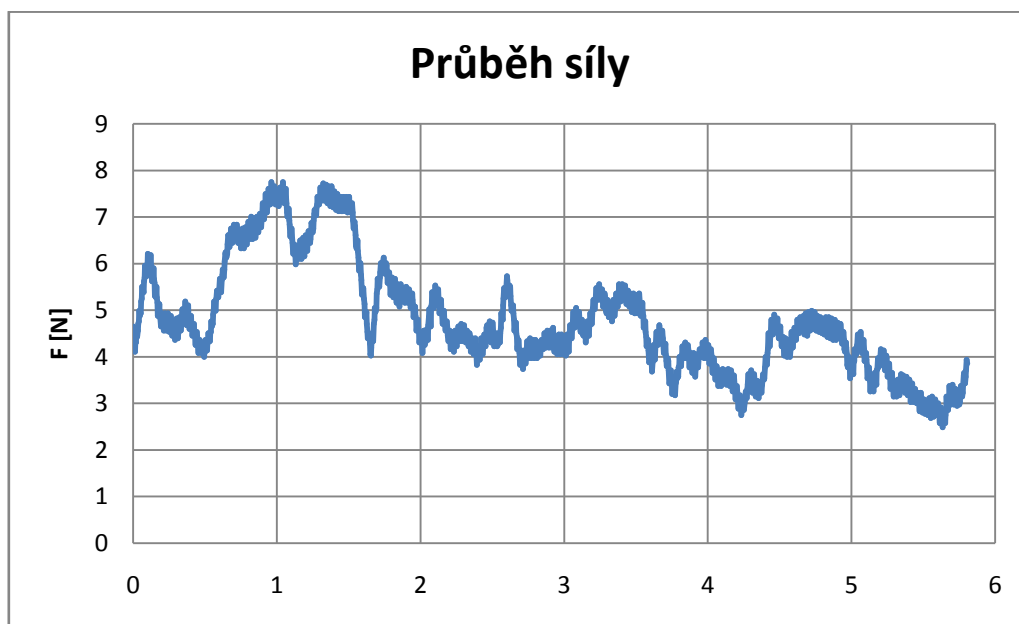


obr. 37 - PI regulace silově řízeného pohonu

Pro ladění konstant PI regulátoru byla použita Ziegler-Nicholsova metoda a poté experimentální iterace. Dlouhé lineární průběhy reálné síly na obr. 37 jsou způsobeny saturací rychlosti otáčení navijecího bubnu, takže s příliš poddajným lanem naviják vyrovnává větší rozdíly s časovou prodlevou.

4.4.2. Oživení lanového manipulátoru

Poté, co je celý manipulátor sestaven (obr. 20) je potřeba ověřit jeho mechanickou pohyblivost. Za tímto účelem bylo k platformě připevněno pouze 6 lan – 3 z nich byla ovládána výše položenými navijáky a zbylá 3 lana byla předepjata závažími o hmotnosti 80 g a vedena skrze spodní rohy pracovního prostoru. Pomocí jedné desky Arduino a dvou desek RoboClaw na jedné sériové lince poté byla provedena šestisekundová sekvence pohybů vzhůru, zpět dolů a kývavé rotace okolo horizontální úhlopříčně vedené osy. Mezi pohyby byly krátké pauzy. Pro ilustraci byla snímána síla v jednom z poháněných navijáků (účastnícího se všech pohybů), která je graficky znázorněna na obr. 38.



obr. 38 - Test pohyblivosti - Průběh síly

Pohyby vzhůru a dolů byly prováděny bez měkkých startů a doběhů, což se v grafickém znázornění projevilo. V čase 0 s je vidět strmý nárůst a následný pokles síly z důvodu energického startu. V druhé polovině první sekundy je možné pozorovat náhlý nárůst způsobený nepříznivým úhlem horních nosných lan při vyšších polohách platformy. V čase 1 s pak vidíme krátký pokles síly a její návrat na původní hodnotu, což je typické pro náhlé

zastavení, kdy platforma setrvačností pokračovala v pohybu vzhůru a tenzometru tak ulehčila. V čase 1,5 s byla platforma skokově spuštěna dolů, což mělo za následek pokles síly z důvodu setrvačností platformy v klidové poloze a – jelikož odvíjení lan bylo prováděno konstantní rychlostí – ráz platformy do zavěšených lan okolo času 1,7 s. V čase 2,5 s pak byla platforma energicky zastavena. V čase 3 s byl započat již pomalejší kývavý pohyb trvající až do konce měření.

Průběh měřené síly na obr. 38 se zdá být zašuměný, nicméně vzorkování probíhá na frekvenci cca 300 Hz (kvůli zpomalení způsobenému vypisováním do uživatelského prostředí) a střední frekvence (cca 6 Hz) vyskytující se v grafu mají systematický a až harmonický průběh. To je způsobeno tím, že platforma pro tento experiment nebyla plně zajištěna ve všech stupních volnosti, byla použita relativně poddajná lana, spodní lana byla předepínána silově (takže nedržela polohu) a závaží se mohla pod průvlakovým očkem volně houpat. Ve výsledku tak platforma během jednoduchých pohybů konala ještě pohyby kývavé s vlastní frekvencí.

5. Závěr

Diplomová práce se zabývá problematikou paralelních redundantních lanových manipulátorů a jejich redundancí. Bylo použito kinematických i dynamických modelů pro návrh celkového modelu řízení ve vývojovém prostředí Matlab, jenž funguje na základě silového ovládní všech pohonů. Díky redundanci systému pak řídicí model umožňuje experimentovat s různými parametrickými distribucemi sil mezi jednotlivé pohony. Práce dále obsahuje návrh reálného experimentálního manipulátoru, včetně jeho sestavení na modulární bázi, kdy je možno snadno přemísťovat pohony v rámci celé konstrukce a lana lze libovolně připevňovat do různých bodů ovládané platformy. Reálný manipulátor je zároveň nízkonákladový a skládá se ze snadno dostupných součástí. Byla provedena zkouška pohyblivosti celého systému a test silového zpětnovazebního řízení samotného navijákového modulu. Tím jsem splnil všechny stanovené cíle práce z kapitoly 1.2 a manipulátor je připraven k budoucím experimentům v rámci obecnějšího školního projektu.

6. Citovaná literatura

1. **Diao, Xiumin and Ma, Ou.** A method of verifying force-closure condition for general cable manipulators with seven cables. *Mechanism and Machine Theory*. 2007, Volume 42, Issue 12, pp. 1563 - 1576.
2. **Michael Valášek, Václav Bauma, Zbyněk Šika, Květoslav Belda, Pavel Píša.** Design-by-Optimization and Control of Redundantly Actuated Parallel Kinematics Sliding Star. *Multibody System Dynamics*. November 2005, Vol. 14, 3, pp. 251-267.
3. **Merlet, J.-P.** On the redundancy of cable-driven parallel robots. *New Trends in Mechanism and Machine Science*. 2015, pp. 31-39.
4. **Merlet, J.-P.** *Checking the cable configuration of cable-driven parallel robots on a trajectory*. Hong Kong : 2014 IEEE International Conference on Robotics & Automation (ICRA), 2014.
5. **Gouttefarde, Marc, et al.** A Versatile Tension Distribution Algorithm for n -DOF Parallel Robots Driven by n+2 Cables. *IEEE Transactions on Robotics*. 2015, Volume 31, Issue 6, pp. 1444 - 1457.
6. **Stejskal, V. and Valášek, M.** *Kinematics and Dynamics of Machinery*. New York : Dekker, 1996.
7. **P. Beneš, Z. Šika, P. Polach, M. Valášek, J. Zavřel, J. Volech.** *Kinematic analysis of possible concepts of multi-level mechanisms combining active structures and cable-driven mechanisms*. Špičák : Computational Mechanics, 2015.
8. **datasheet.** RoboClaw Series - Brushed DC Motor Controllers. [Online] 2015. http://downloads.ionmc.com/docs/roboclaw_datasheet_2x15a.pdf.
9. **Havlík, Jan and Horčík, Zdeněk.** Zesilovače biologických signálů. *Web Server Noel*. [Online] http://noel.feld.cvut.cz/vyu/x31zle/Lectures/04_Zesilovac.pdf.
10. Discrete-time PID Controller Implementation. *Control Systems Lab*. [Online] 8 8, 2014. <http://controlsystemslab.com/discrete-time-pid-controller-implementation/>.