

# TAČR Centrum kompetence CIDAM

SW Tools and methodology for optimization of mechatronic systems in terms of structure, parameters and control

SW knihovna funkcí a funkčních bloků pro parametrickou optimalizaci robotů

Výzkumná zpráva

Martin Švejda

13. prosince 2017

**T A**  
**Č R**



**CIDAM**

*Center for Intelligent Drives  
and Advanced Machine Control*

## Abstrakt

Předložená zpráva se zabývá popisem SW knihovny funkcí a funkčních bloků implementovaných v prostředí Matlab, Simulink/SimMechanics, které vznikly jako dílčí výsledek výzkumu metod optimalizace robotických architektur. Konkrétně se jedná o oblast parametrické optimalizace, tedy nalezení optimálních kinematických parametrů manipulátoru (Denavit-Hartenbergovy parametry). Knihovna obsahuje předimplementované funkce pro řešení diskretního optimalizačního problému (nalezení globálního optima). Dále jsou v knihovně zahrnuty předimplementované funkce a funkční bloky pro tvorbu zobecněných virtuálních simulačních modelů sériových manipulátorů v případě, kdy každé rameno manipulátoru může obsahovat až 4 nezávislé aktuátory (max. 4 Denavit-Hartenbergovy parametry). Tato část knihovny tvoří podporu pro nový směr parametrické/strukturální optimalizace, kdy pro vzniklé redundantní manipulátory lze hledat optimální trajektorie pohybu (v důsledku nekonečně mnoho řešení inverzní kinematiky) a následně tyto nalezené optimální trajektorie dále analyzovat za účelem realizace dodatečných parametrických či strukturálních změn v architektuře manipulátoru (např. volba vhodných aktuátorů, jejich umístění, atd.).

## Obsah

1	Úvod	4
2	Algoritmy jako podpora diskrétní optimalizace	5
3	Algoritmy jako podpora optimalizace s využitím optimálního pohybu redundantních manipulátorů	8
4	Závěr	13

# 1 Úvod

Předložená zpráva se zabývá popisem vyvinutých algoritmů určených k optimalizaci kinematických architektur robotů, konkrétně se jedná zejména o parametrickou optimalizaci, tedy nalezení geometrických návrhových parametrů. Návrhovými parametry rozumíme takové kinematické parametry manipulátoru, které nejsou kloubovými souřadnicemi, a zůstávají tedy konstantní během pohybu robotu (typickým případem jsou ty Denavit-Hartenbergovy parametry [2], které nereprezentují pohybu kloubů manipulátoru).

Standardně lze na problém parametrické optimalizace manipulátoru nahlížet jako na problém nelineární optimalizace se zvolenou kriteriální funkcí a s definovanými omezujícími podmínkami:

$$\begin{aligned} \xi^* &= \operatorname{argmax}_{\xi \in \Xi} (J(\mathbf{X}_{\text{opt}}, \xi)) \\ \text{w.r.t.: } \quad \mathbf{Eq}(\mathbf{X}, \xi) &= 0, \quad \mathbf{Ineq}(\mathbf{X}, \xi) \geq 0 \end{aligned} \tag{1}$$

kde  $J$  je kriteriální funkce,  $\xi$  je vektor optimalizovaných kinematických parametrů z přípustné množiny  $\Xi$ ,  $\mathbf{X}$  je stav robotu (např. poloha/rychlost/zrychlení koncového efektoru) z požadovaného pracovního prostoru  $\mathbf{X}_{\text{opt}}$ ,  $\mathbf{Eq}$  resp.  $\mathbf{Ineq}$  jsou rovnice omezení dané rovnostmi resp. nerovnostmi.

Optimalizační úloha (1) bývá často zjednodušena na tvar optimalizační úlohy bez omezení

$$\xi^* = \operatorname{argmax}_{\xi \in \Xi} (J(\mathbf{X}_{\text{opt}}, \xi)) \tag{2}$$

zahrnutím omezujících rovnic  $\mathbf{Eq}$ ,  $\mathbf{Ineq}$  do hodnoty kriteriální funkce  $J$  metodou **penalizací** - přičtením hodnoty penalizační funkce  $J_{\text{pen}}$  k hodnotě účelové funkce  $J_{\text{obj}}$  (vlastní kritérium optimalizace) v případě porušení omezení. Výslednou kriteriální funkci lze pak definovat následovně:

$$J(\mathbf{X}, \xi) = \frac{1}{J_{\text{pen}} + J_{\text{obj}}} \in \langle 0, 1 \rangle \tag{3}$$

Pro penalizační funkci  $J_{\text{pen}}$  potom platí, že  $J_{\text{pen}} \rightarrow 0$  odpovídá splnění všech předepsaných kritérií,  $J_{\text{pen}} \rightarrow 1$  odpovídá úplnému porušení všech kritérií. Poznamenejme, že dílčí sčítaná omezení optimalizace mohou být různě vážena (dle nastavených podmínek a potřeb optimalizační úlohy, tzn. vážení nízkými hodnotami koeficientu odpovídá tzv. „měkkým“ omezením, vážení vysokými hodnotami koeficientu odpovídá „tvrdým“ omezením optimalizace).

## ♣ Příklad:

Omezení na maximální výsuv lineárních aktuátorů robotu může být dáno vztahem:

$$J_{\text{pen}} = K_1 \cdot P_1 + K_2 \cdot P_2 \tag{4}$$

kde

$$P_i = \begin{cases} 0 & \text{pro: } (\|d_i\| - d_i^{\text{max}}) < 0 \\ (\|d_i\| - d_i^{\text{max}}) & \text{pro: } (\|d_i\| - d_i^{\text{max}}) \geq 0 \end{cases}, \quad i = 1, 2$$

a  $d_i$  jsou aktuální hodnoty kloubových souřadnic výsuvů lineárních aktuátorů pro dané umístění koncového efektoru do bodu  $\mathbf{X}$  v pracovním prostoru,  $d_i^{\text{max}}$  jsou maximální povolená vysunutí aktuátorů a  $K_i > 0$  jsou příslušné penalizační konstanty.

♣

Účelová funkce  $J_{\text{obj}}$  zohledňující požadované vlastnosti manipulátoru, které mají být optimalizovány a vyjadřuje míru optimality dle zvoleného hlediska,  $J_{\text{obj}} \rightarrow 0$  odpovídá optimálnímu případu (hodnota účelové funkce je minimální možná),  $J_{\text{obj}} \rightarrow +\infty$  odpovídá nejvíce vzdálenému případu od optimálního.

V rámci výzkumu a vývoje algoritmů optimalizace byly prozkoumány dvě alternativy, a to klasický přístup parametrické optimalizace tak, jak byl definován výše, vedoucí na **globálního diskrétního algoritmu optimalizace (diskrétní optimalizace)**, viz Kapitola 2, a nová idea využívající **optimalizace pohybu redundantních manipulátorů** za účelem posouzení optimality uvažované kinematické architektury manipulátoru a možnosti jejího následného zlepšení, viz Kapitola 3.

## 2 Algoritmy jako podpora diskrétní optimalizace

Optimalizační problém definovaný v (2, 3) lze v diskretizované podobě psát jako:

$$\begin{aligned} J^*(\mathbf{X}, \boldsymbol{\xi}^*) &= \max_{i=1\dots N} \left( \min_{j=1\dots M} J_{\text{val}}(j, i) \right) \\ i^* &= \operatorname{argmax}_{i=1\dots N} \left( \min_{j=1\dots M} J_{\text{val}}(j, i) \right), \quad \boldsymbol{\xi}^* = \Xi\{i^*\} \end{aligned} \quad (5)$$

kde diskrétní prostor (matice) je dána jako:

$$J_{\text{val}} = \begin{bmatrix} J(\mathbf{X}_{\text{opt}\{1\}}, \Xi\{1\}) & J(\mathbf{X}_{\text{opt}\{1\}}, \Xi\{2\}) & \dots \\ J(\mathbf{X}_{\text{opt}\{2\}}, \Xi\{1\}) & J(\mathbf{X}_{\text{opt}\{2\}}, \Xi\{2\}) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

$\mathbf{X}_{\text{opt}} = \{\mathbf{X}_1 \dots \mathbf{X}_M\}$ ,  $\mathbf{X}_j \in \mathbb{R}^m$ ,  $\Xi = \{\boldsymbol{\xi}_1 \dots \boldsymbol{\xi}_N\}$ ,  $\boldsymbol{\xi}_i \in \mathbb{R}^p$  kde  $\mathbf{X}_j$  resp.  $\boldsymbol{\xi}_i$  reprezentuje vybraný diskrétní bod v pracovním prostoru robotu (např. polohu koncového efektoru robotu) resp. jednu realizaci (vektor) kinematických parametrů a  $J(\mathbf{X}, \boldsymbol{\xi})$  je kriteriální funkce jejíž minimální hodnota přes vybraný (optimalizovaný) pracovní prostor má být maximalizována (diskrétní problém optimalizace typu MINIMAX).

Poznamenejme, že výpočetní tvar kriteriální funkce  $J(\mathbf{X}, \boldsymbol{\xi})$  je definován uživatelem s ohledem na podstatu vykonávané optimalizace (zvolené kritérium optimalizace).

### ♣ Poznámka:

V případě MINIMAX optimalizace lze pak snadno využít např. nové kritérium optimalizace publikované v [5], které zavádí účelovou funkci (3)  $J_{\text{obj}}(\mathbf{X}, \boldsymbol{\xi})$  jako hodnotu normy síly/silových momentů kloubů manipulátoru  $\boldsymbol{\tau}$  v definované poloze  $\mathbf{X}$  koncového efektoru manipulátoru potřebnou k pohybu z klidu do libovolného směru v pracovním prostoru s maximálním zrychlením  $a_{\text{max}}$ :

$$\begin{aligned} \forall \mathbf{X} : \quad \|\dot{\mathbf{X}}\| = 0, \|\ddot{\mathbf{X}}\| \leq a_{\text{max}} &\Rightarrow \|\boldsymbol{\tau}\| \leq \|\boldsymbol{\tau}\|_{\text{max}} \Rightarrow J_{\text{obj}}(\mathbf{X}, \boldsymbol{\xi}) = \|\boldsymbol{\tau}\|_{\text{max}} \\ &\|\boldsymbol{\tau}\|_{\text{max}} = \sigma_{\text{max}}(\mathbf{M}(\mathbf{Q}) \cdot \mathbf{J}^{-1}(\mathbf{Q})) \cdot a_{\text{max}} + \|\mathbf{G}(\mathbf{Q})\| \end{aligned} \quad (6)$$

kde  $\mathbf{J}(\mathbf{Q})$  je jakobián manipulátoru,  $\mathbf{M}(\mathbf{Q})$ ,  $\mathbf{G}(\mathbf{Q})$ ,  $\mathbf{C}(\mathbf{Q}, \dot{\mathbf{Q}})$  jsou dynamické matice manipulátoru odpovídající dynamickému modelu:

$$\mathbf{M}(\mathbf{Q}) \cdot \ddot{\mathbf{Q}} + \mathbf{C}(\mathbf{Q}, \dot{\mathbf{Q}}) \cdot \dot{\mathbf{Q}} + \mathbf{G}(\mathbf{Q}) = \boldsymbol{\tau}$$

kde  $\mathbf{Q}$  je vektor kloubových souřadnic manipulátoru vztažený k zobecněným souřadnicím (poloze konc. efektoru)  $\mathbf{X}$  prostřednictvím přímé a inverzní kinematické úlohy.

Poznamenejme, že kinematický jakobián  $J(\mathbf{Q})$ , dynamické matice  $M(\mathbf{Q})$ ,  $G(\mathbf{Q})$ ,  $C(\mathbf{Q}, \dot{\mathbf{Q}})$  a řešení přímé kinematiky lze získat prostřednictvím funkcí implementovaných v SW balíku *SW knihovna funkcí a funkčních bloků pro modelování robotu*, viz [6].

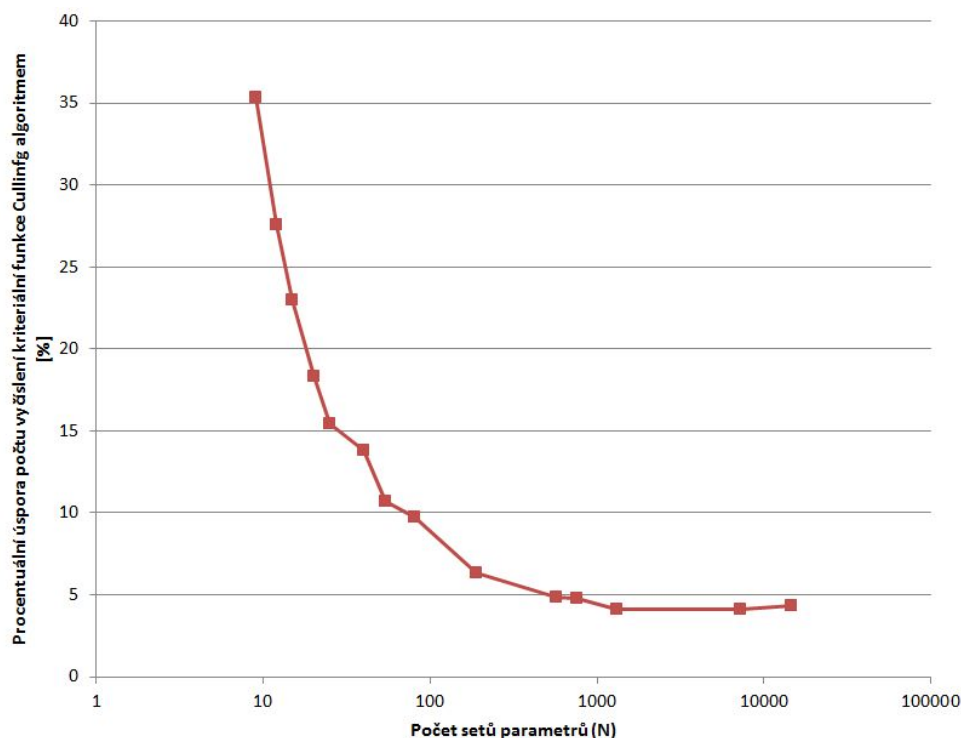


## Předimplementované funkce - Matlab (m-files)

Podrobné detaily k výpočetním jádrům předložených algoritmů lze nalézt v [4].

- `varargout = culling(varargin)`:  
`varargin = {genCoords,robotParameters_kinematic,optParametersIndex,weight,...`  
`...optPar_range,optPar_dim,otherParameters,numOfOptims}`  
`varargout = {robotParameters_kinematic_vect,ksi_opts_val}`

Algoritmus diskrétní optimalizace, problém definován v (5), pro nalezení globálního extrému kriteriální funkce  $J(\mathbf{X}, \xi)$  nad diskretizovaným pracovním prostorem  $\mathbf{X} \in \mathbf{X}_{\text{opt}}$  z diskretizované množiny přípustných kinematických parametrů manipulátoru  $\xi \in \Xi$ . Algoritmus je ze třídy algoritmů *branch and bound*, který z uvažovaného prohledávaného prostoru  $\Xi$  vyřezává apriori neoptimální podmnožiny setů parametrů a výrazně tak redukuje výpočetní náročnost, viz Obrázek 1. Algoritmus je inspirován originálním algoritmem v [1], [3]. Algoritmus byl dále upraven pro možnosti vyhledávání 2., 3. atd. globálního optima a implementován do funkce v Matlabu, další detaily včetně ilustračních příkladů jsou k nalezení v [4].



Obrázek 1: Výpočetní náročnost algoritmu (vyčíslení hodnoty kriteriální funkce) v porovnání s algoritmem prohledávání hrubou silou pro diskretizaci pracovního prostoru  $\mathbf{X}_{\text{opt}}$  ( $M = 45$ ) a proměnný počet přípustných (diskretizovaných) setů kinematických parametrů  $\Xi$  ( $N$ ).

<u>Vstupy:</u>	
<code>genCoords</code>	Matice složená ze sloupců $\mathbf{X}_i$ reprezentující uvažované diskrétní body pracovního prostoru $\mathbf{X}_{opt}$
<code>robotPar..._kinematic</code>	Kinematické parametry manipulátoru $\xi$ (počáteční odhad - volba blízko k optimu urychlí proces prořezávání)
<code>optParametersIndex</code>	Vektor indexů definující jaké kinematické parametry z $\xi$ budou optimalizovány (z úplného vektoru kinematických návrhových parametrů <code>robotParameters_kinematic</code> , ne všechny parametry musí být optimalizovány)
<code>weight</code>	Vektor váhových penalizačních koeficientů $K_i$
<code>optPar_range</code>	Vektor přípustného rozsahu (sloupec $[\xi_i^{\min}; \xi_i^{\max}]$ ) pro každý optimalizovaný kinematický parametr (přípustná množina)
<code>optPar_dim</code>	Vektor $[\text{Dim}(\xi_1), \text{Dim}(\xi_2), \dots]$ počet diskretizovaných hodnot z rozsahu každého opt. kin. parametru (rozlišení diskretizace)
<code>otherParameters</code>	Pomocné parametry (typicky pro vyčíslení kritériální funkce)
<code>numOfOptims</code>	Počet hledaných globálních optim (v pořadí za sebou 2., 3., atd. gl. optimum pro možnosti spuštění následné lokální optimalizace z různých poč. podmínek)
<u>Výstupy:</u>	
<code>robotPar..._kin..._vect</code>	Vektor nalezených optimálních hodnot kinematických parametrů $\xi^*$
<code>ksi_opts_val</code>	Hodnota kritériální funkce přes pracovní prostor

$$J(\mathbf{X}_{opt}, \xi^*) = \min_{\mathbf{X} \in \mathbf{X}_{opt}} J(\mathbf{X}, \xi^*)$$

pro optimální parametry  $\xi^*$

- `critFunVal = critFunction(varargin)`:

`varargin = {genCoords, robotParameters_kinematic, weights, otherParameters}`

Vyčíslení hodnoty kritériální funkce. Definuje pouze rozhraní pro implementaci kritériální funkce  $J(\mathbf{X}, \xi)$  (tělo funkce je implementováno uživatelem - např. s využitím funkcí z knihovny *SW knihovna funkcí a funkčních bloků pro modelování robotu*, viz [6]). Funkce `critFunction` je využívána funkcí `culling`.

<u>Vstupy:</u>	
<code>genCoords</code>	Stav robotu $\mathbf{X}$ (koncová poloha koncového efektoru, zobecněné souřadnice)
<code>robotPar..._kinematic</code>	Kinematické parametry manipulátoru $\xi$
<code>weight</code>	Vektor váhových penalizačních koeficientů $K_i$
<code>otherParameters</code>	Pomocné parametry (pro vyčíslení krit. funkce)
<u>Výstupy:</u>	
<code>critFunVal</code>	Hodnota kritériální funkce $J(\mathbf{X}, \xi)$ .

- `[kinPar_chains, dynPar_chains] = structurePar(varargin)`:

`varargin = {robotParameters_kinematic, robotParameters_dynamic, Q_home}`

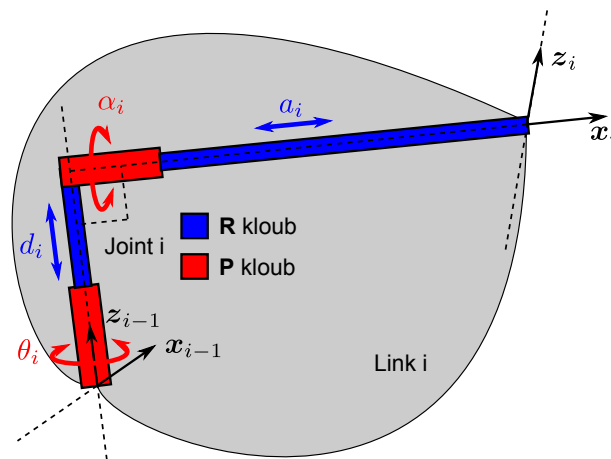
Vytvoření struktur `kinPar`, `dynPar` tak, jak jsou definovány ve funkci `robotSetup` (v knihovně *SW knihovna funkcí a funkčních bloků pro modelování robotu* [6]) z vektorů vybraných kinematických parametrů  $\xi$  (např. Denavit-Hartenbergovy parametry) a dynamických parametrů (např. hmotnosti, těžiště, setrvačnosti ramen) robotu. Definuje pouze rozhraní (tělo funkce implementováno uživatelem). Funkce `structurePar` je obvykle používána ve funkci `critFunction` za účelem konverze vektoru uvažovaných kin. parametrů (dyn. parametrů) do standardních struktur pro následné využití standardních funkcí pro kinematiku/dynamiku robotů, viz *SW knihovna funkcí a funkčních bloků pro modelování robotu* [6].

	<u>Vstupy:</u>
<code>robotParameters_kinematic</code>	Vektor kinematických parametrů robotu
<code>robotParameters_dynamic</code>	Vektor dynamických parametrů robotu
<code>Q_home</code>	Domovská poloha robotu (vektor kloubových souřadnic)
	<u>Výstupy:</u>
<code>kinPar_chains</code>	Struktura kinematických parametrů (.DHpar, .jointType, atd.)
<code>dynPar_chains</code>	Struktura dynamických parametrů (.mass, .inertiaTensor, atd.)

### 3 Algoritmy jako podpora optimalizace s využitím optimálního pohybu redundantních manipulátorů

Myšlenka využití redundance a následné optimalizace pohybu redundantních manipulátorů za účelem parametrické (částečně strukturální) optimalizace kinematických architektur je detailně popsána v práci [4]. Stručně ji lze shrnout následovně:

- Uvažujme neredundantní manipulátor a jeho standardní popis kinematiky prostřednictvím Denavit-Hartenbergových (D-H) parametrů, tzn. dimenze vektoru kloubových souřadnic  $\mathbf{Q}$  je shodná s dimenzí vektoru zobecněných souřadnic  $\mathbf{X}$ . Zbývající D-H parametry (vyjma těch reprezentující  $\mathbf{Q}$ ) představují geometrické (návrhové) parametry manipulátoru  $\xi$ .
- Dle potřeby mohou být některé z návrhových parametrů  $\xi$  „uvolněny“ a formálně přesunuty do vektoru kloubových souřadnic  $\mathbf{Q}$  - dostáváme tak redundantní manipulátor s daným stupněm redundance (dle počtu uvolněných parametrů). Jinými slovy, libovolný D-H parametr, viz Obrázek 2, nikoliv pouze parametr  $d_i$  (translační kloub) či  $\theta_i$  (rotační kloub), může reprezentovat aktuátor manipulátoru (celkem až 4 aktuátory  $d_i, \theta_i, a_i, \alpha_i$  na jedno rameno).



Obrázek 2: Popis D-H parametry s vyznačenými možnostmi reprezentace aktuátorů na ramenu manipulátoru (2x prismatický, 2x rotační aktuátor)

- Redundantní manipulátor  $\Rightarrow$  nekonečně možných řešení inverzní kinematiky.
- $\Rightarrow$  **Optimalizace pohybu redundantního manipulátoru** po zadané trajektorii pohybu  $\Rightarrow$  optimální pohyb (redundantních) kloubových souřadnic.
- Optimální trajektorie kloubových souřadnic lze dále analyzovat s ohledem na:



- **Parametrickou optimalizaci** - z optimálních průběhů kloubových souřadnic lze usuzovat, jaké kloubové souřadnice (D-H parametry) by bylo možné fixovat na konstantních hodnotách a dále identifikovat (nadřazení iterační algoritmy) tyto konkrétní hodnoty.
- **Strukturální optimalizaci** - z optimálních průběhů kloubových souřadnic lze usuzovat na vhodnost osazených aktuátorů manipulátoru, jinými slovy, zda-li není výhodné některé aktuátory nahradit jinými typy ( $\mathbf{P}$ ,  $\mathbf{R}$ ) či aktuátory přesunout na jiné pozice v kinematickém řetězci (na místo D-H parametrů  $d_i$ ,  $\theta_i$ ,  $a_i$ ,  $\alpha_i$ ).

Samotný proces nalezení optimální trajektorie redundantních manipulátorů je komplexním problémem a je diskutován podrobně v [4], kde jsou představeny různé možné strategie a přístupy. Za účelem možnosti vývoje takových algoritmů je však nutné standardní algoritmy pro řešení kinematických a dynamických úloh v robotice, tak jak jsou uvedeny v knihovně *SW knihovna funkcí a funkčních bloků pro modelování robotu*, viz [6], přepracovat tak, aby mohly být současně definovány pro každé rameno až 4 nezávislé aktuátory (klouby).

Následující výčet shrnuje předimplementované funkce v Matlabu a funkční bloky v SimMechanicsu jako stavební bloky pro vytváření virtuálních simulačních modelů, které slouží pro analýzu obecných redundantních manipulátorů vzniklých rozvolněním příslušných D-H parametrů ve standardním popisu kinematiky manipulátorů.

### Předimplementované funkce - Matlab (m-files)

Podrobné detaily k výpočetním jádrům předložených algoritmů lze nalézt v [4].

- `[Links, Joints] = robotSetup_redundant(kinPar, dynPar)`:  
Vytvoří strukturu kinematických a dynamických parametrů pro sériový zobecněný kinematický řetězec - využíváno dalšími funkčními bloky.

<u>Vstupy:</u>	
<code>kinPar</code>	Struktura kinematických parametrů manipulátoru
<code>.DHpar</code>	Matice D-H parametrů: $[d_1, \theta_1, a_1, \alpha_1; d_2, \theta_2, a_2, \alpha_2; \dots; d_n, \theta_n, a_n, \alpha_n]$
<code>.jointType</code>	Typy a umístění příslušných aktuátorů (až 4 DoF na jedno rameno), ve formátu:

$$[is^{d_1}, is^{\theta_1}, is^{a_1}, is^{\alpha_1}; is^{d_2}, is^{\theta_2}, is^{a_2}, is^{\alpha_2}; \dots; is^{d_n}, is^{\theta_n}, is^{a_n}, is^{\alpha_n}],$$

$is^{d_i} = 1$  pokud  $i$ -té zobecněné redundantní rameno obsahuje **P** aktuátor reprezentovaný D-H parametrem  $d_i$ , jinak  $is^{d_i} = 0$

$is^{\theta_i} = 1$  pokud  $i$ -té zobecněné redundantní rameno obsahuje **R** aktuátor reprezentovaný D-H parametrem  $\theta_i$ , jinak  $is^{\theta_i} = 0$

$is^{a_i} = 1$  pokud  $i$ -té zobecněné redundantní rameno obsahuje **P** aktuátor reprezentovaný D-H parametrem  $a_i$ , jinak  $is^{a_i} = 0$

$is^{\alpha_i} = 1$  pokud  $i$ -té zobecněné redundantní rameno obsahuje **R** aktuátor reprezentovaný D-H parametrem  $\alpha_i$ , jinak  $is^{\alpha_i} = 0$

<code>dynPar</code>	Struktura dynamických parametrů manipulátoru
<code>.mass</code>	Vektor hmotností ramen, ve formátu: $[m_1; m_2 \dots; m_n]$
<code>.inertiaTensor</code>	Vektor tensorů setrvačnosti ramen v těžišti vzhledem k s.s. příslušného ramene, ve formátu: $[\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_n]$
<code>.gravityCenter</code>	Matice vektorů těžišť ramen vzhledem k příslušnému s.s. ramene, ve formátu: $[\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_N]$
<code>.gravityVector</code>	Vektor gravitačního zrychlení vzhledem k s.s. $F_0$ , ve formátu: $[g_x; g_y; g_z]$

Výstupy:

<code>Links{j}</code>	Struktura obsahující informace o $j$ -tém ramenu ( $j = 1, 2, \dots, n$ ). Struktura je dále používána funkčními bloky.
<code>.base</code>	Počátek $j$ -tého ramene, tzn. počátek s.s. $F_{j-1}$
<code>.mass</code>	Hmotnost $j$ -tého ramene
<code>.inertiaTensor</code>	(Konstantní) tensor setrvačnosti $j$ -tého ramene, vzhledem k s.s. ramene $F_j$
<code>.gravityCenter</code>	(Konstantní) umístěné těžiště $j$ -tého ramene, vzhledem k s.s. ramene $F_j$
<code>Joints{j}</code>	Struktura obsahující informace o $j$ -tém zobecněném kloubu ( $j = 1, 2, \dots, n$ ). Struktura je dále používána funkčními bloky.
<code>.rangeOfJointCoords</code>	Index (pořadí) aktivních kl. souřadnic příslušející $j$ -tému zobecněnému kloubu, ve tvaru dvojice $[n_1; n_2]$ , $1 \leq n_1 \leq n_2 \leq \dim(\mathbf{Q})$
<code>.jointCoordDim</code>	Celkový počet aktivních kloubů manipulátoru, $\dim(\mathbf{Q})$
<code>.DHpar</code>	D-H parametry $j$ -tého ramene: $[d_j, \theta_j, a_j, \alpha_j]$
<code>.jointType</code>	Umístění aktuátorů $j$ -tého zobecněného kloubu: $[is^{d_j}, is^{\theta_j}, is^{a_j}, is^{\alpha_j}]$

- `[J,dJ] = kinJacobian_redundant(jointCoords,DHpar,jointType):`

Výpočet kinematického jakobiánu  $\mathbf{J}_n$  a jeho časové derivace  $\dot{\mathbf{J}}_n$ .

Vstupy:  
**jointCoords** Hodnoty poloh kloubových souřadnic a odpovídající rychlosti, ve formátu:  $[Q, \dot{Q}]$  (více aktuátorů - kl. souřadnic na jedno rameno)  
**DHparam** Matice D-H parametrů, viz výše  
**jointType** Typy příslušných kloubů, viz výše

Výstupy:  
**J** Kinematický jakobián  $J_n \in \mathbb{R}^{6 \times n}$   
**dJ** Časová derivace kinematického jakobiánu  $\dot{J}_n \in \mathbb{R}^{6 \times n}$

- `[genCoords] = forwardKinematics_redundant(jointCoords, kinPar, N):`  
 Formálně shodná jako funkce `forwardKinematics` v knihovně *SW knihovna funkcí a funkčních bloků pro modelování robotu*, viz [6].
- `[tau] = inverseDynamicModel_redundant(jointCoords, endEff_force_moment, ... . . . kinPar, dynPar):`  
 Formálně shodná jako funkce `inverseDynamicModel` v knihovně *SW knihovna funkcí a funkčních bloků pro modelování robotu*, viz [6].
- `[ddq] = forwardDynamicModel_redundant(jointCoords, joint_force_moment, ... . . . endEff_force_moment, kinPar, dynPar):`  
 Formálně shodná jako funkce `forwardDynamicModel` v knihovně *SW knihovna funkcí a funkčních bloků pro modelování robotu*, viz [6].

## Implementované funkční bloky - Matlab/Simulink/SimMechanics

Viz Obrázek 3.

- **Manipulator Link:**  
 Blok reprezentující  $j$ -té rameno manipulátoru.  
Vstupy/Výstupy:  
 CS Fi-1, CS Fi Přípojně body ramena tvoří s.s. ramene  $F_j$  a s.s. ramene předcházejícího  $F_{j-1}$   
Parametry:  
**Link order** Pořadí umístění ramene v kinematickém řetězci.
- **Manipulator general Joint (up to 4 DoF - DH param) (Actuation: Motion):**  
 Blok reprezentující  $j$ -tý zobecněný kloub manipulátoru. Vzhledem k tomu, že uvažujeme obecné redundantní rameno - se zobecněným aktuátorem (až 4 nezávislé DoF:  $d_j, \theta_j, a_j, \alpha_j$ ), parametry ramena (`Links{j}`) definují pouze počátek ramena (`Links{j}.base`), je koncový bod ramene vypočten přímo z příslušných D-H parametrů a to následovně: Dle určených aktuátorů ramena (viz  $j$ -tý řádek matice `kinPar.jointType`) jsou všechny příslušné D-H parametry (a jejich časové derivace) reprezentující aktuátory nahrazeny polohami (rychlostmi, zrychleními) kloubových souřadnic, zbývající (konstantní) D-H parametry reprezentující kinematické návrhové parametry jsou nahrazeny zadanými D-H parametry (viz `kinPar.DHpar`, odpovídající rychlosti a zrychlení jsou nulové). Dostáváme tak transformaci, která přesouvá a orientuje  $j$ -té rameno na základě kloubových souřadnic  $Q_j \in \{d_j, \theta_j, q_j, \alpha_j\}$  (dle výběru aktuátorů příslušného ramene) a konstantních kinematických návrhových parametrů manipulátoru  $\xi_j = \{d_j, \theta_j, q_j, \alpha_j\} \setminus Q_j$  (zbývající D-H parametry).

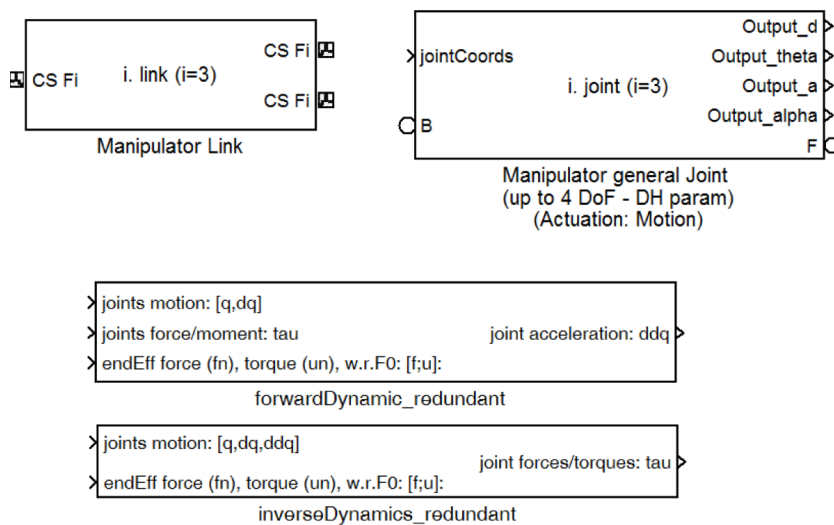
### Vstupy/Výstupy:

$B, F$	$Base, Follower$ - standardní přípojný body kloubů v SimMechanicsu
Output_d	= [Computed force; Reaction torque; Reaction force] pro <b>P</b> aktuátor odpovídající D-H parametru $d_i$
Output_theta	= [Computed torque; Reaction torque; Reaction force] pro <b>R</b> aktuátor odpovídající D-H parametru $\theta_i$
Output_a	= [Computed force; Reaction torque; Reaction force] pro <b>P</b> aktuátor odpovídající D-H parametru $a_i$
Output_alpha	= [Computed torque; Reaction torque; Reaction force] pro <b>R</b> aktuátor odpovídající D-H parametru $\alpha_i$

### Parametry:

Joint order Pořadí umístění kloubu v kinematickém řetězci.

- **forwardDynamics\_redundant:**  
Blok reprezentující řešení DDM založený na funkci `forwardDynamicModel_redundant` s odpovídajícími vstupy/výstupy. Bez parametrů. Řešení DDM je implementováno **pouze pro sériové manipulátory**.
- **inverseDynamics\_redundant:**  
Blok reprezentující řešení IDM založený na funkci `inverseDynamicModel_redundant` s odpovídajícími vstupy/výstupy. Bez parametrů. Řešení IDM je implementováno **pouze pro sériové manipulátory**.



Obrázek 3: Bloky knihovny „robotLib“ do prostředí SimMechanics pro zobecněný redundantní manipulátor

## Poděkování

Tento výzkum byl podpořen Technologickou agenturou České republiky z prostředků projektu Centra Kompetence CIDAM TE02000103.

## 4 Závěr

Knihovny předimplementovaných funkcí a funkčních bloků mohou být volně staženy zde:

### **Algoritmy jako podpora diskrétní optimalizace**

[http://home.zcu.cz/~msvejda/PhD\\_disertace/Algoritmy/robotLib/Optimal/](http://home.zcu.cz/~msvejda/PhD_disertace/Algoritmy/robotLib/Optimal/)

### **Algoritmy jako podpora optimalizace s využitím optimálního pohybu redundantních manipulátorů**

[http://home.zcu.cz/~msvejda/PhD\\_disertace/Algoritmy/robotLib/Redundant/](http://home.zcu.cz/~msvejda/PhD_disertace/Algoritmy/robotLib/Redundant/)

## Reference

- [1] *Mechanism design for global isotropy with applications to haptic interfaces*, 1997, cited By (since 1996): 5.  
URL [www.scopus.com](http://www.scopus.com)
- [2] Denavit, J.; Hartenberg, R. S.: A kinematic notation for lower-pair mechanisms based on matrices. *Trans. of the ASME. Journal of Applied Mechanics*, ročník 22, 1955: s. 215–221.  
URL <http://ci.nii.ac.jp/naid/10008019314/en/>
- [3] Stocco, L.; Salcudean, S. E.; Sassani, F.: Fast Constrained Global Minimax Optimization of Robot Parameters. *Robotica*, ročník 16, č. 6, Listopad 1998: s. 595–605, ISSN 0263-5747, doi:10.1017/S0263574798000435.  
URL <http://dx.doi.org/10.1017/S0263574798000435>
- [4] Švejda, M.: *Optimalizace robotických architektur*. Dizertační práce, Západočeská univerzita v Plzni, 2016.  
URL [http://home.zcu.cz/~msvejda/\\_publications/2016/4\\_SvejdaMartin\\_thesis\\_2016\\_06\\_14.pdf](http://home.zcu.cz/~msvejda/_publications/2016/4_SvejdaMartin_thesis_2016_06_14.pdf)
- [5] Švejda, M.: New Kinetostatic Criterion for Robot Parametric Optimization. In *2017 IEEE 4th Intl. Conference on Soft Computing & Machine Intelligence (ISCMi 2017)*, 2017, s. 66–70, doi:ISBN:978-1-5386-1313-9.
- [6] Švejda, M.: SW knihovna funkcí a funkčních bloků pro modelování robotu. Technická zpráva, Západočeská univerzita v Plzni, 2017.  
URL [http://home.zcu.cz/~msvejda/\\_publications/2017/11\\_SWknihovny\\_General.pdf](http://home.zcu.cz/~msvejda/_publications/2017/11_SWknihovny_General.pdf)