C. Knoll, X. Jia, R. Heedt

Trajectory Planning for Closed Kinematic Chains Applied to Cooperative Motions in Health Care

Kassel (virtual), 2021-03-17

Introduction

- Important situation in health care: standing up from sitting position

Source: https://www.robotik-produktion.de/allgemein/roboter-in-der-pflege/

Introduction

- Important situation in health care: standing up from sitting position
- Usual strategy: physical support from care giver

Source: https://www.robotik-produktion.de/allgemein/roboter-in-der-pflege/

Introduction

- Important situation in health care: standing up from sitting position
- Usual strategy: physical support from care giver
- Challenge: variety of possible motion patterns

$\rightarrow$ Different stress levels for care giver and receiver

- Optimal motion? $\rightarrow$ mathematical model necessary

Source: https://www.robotik-produktion.de/allgemein/roboter-in-der-pflege/

Introduction

- Important situation in health care: standing up from sitting position
- Usual strategy: physical support from care giver
- Challenge: variety of possible motion patterns

$\rightarrow$ Different stress levels for care giver and receiver

- Optimal motion? $\rightarrow$ mathematical model necessary
- Simplifications:
  - Humans $\hat{=}$ planar kinematic chains

Source: https://www.robotik-produktion.de/allgemein/roboter-in-der-pflege/

Introduction

- Important situation in health care: standing up from sitting position
- Usual strategy: physical support from care giver
- Challenge: variety of possible motion patterns

$\rightarrow$ Different stress levels for care giver and receiver

- Optimal motion? $\rightarrow$ mathematical model necessary
- Simplifications:
  - Humans $\triangleq$ planar kinematic chains
  - Support point: revolute joint at shoulders

Source: https://www.robotik-produktion.de/allgemein/roboter-in-der-pflege/

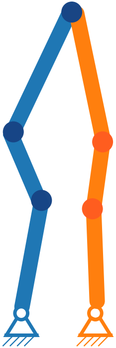
Introduction

- Important situation in health care: standing up from sitting position
- Usual strategy: physical support from care giver
- Challenge: variety of possible motion patterns

$\rightarrow$ Different stress levels for care giver and receiver

- Optimal motion? $\rightarrow$ mathematical model necessary
- Simplifications:
  - Humans $\hat{=}$ planar kinematic chains
  - Support point: revolute joint at shoulders

$\rightarrow$ Motion planning for **closed kinematic chains** necessary

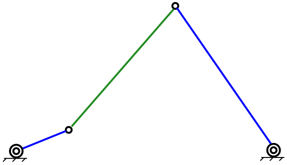


Outline

- ☑ Motivation
- → **Modelling of Closed Kinematic Chains**
- ☐ Motion Planning for Kinematic Chains
- ☐ Simplification via Quasi-Stationary Trajectory Planing
- ☐ Conclusion and Outlook

Simple Case: Four Bar Linkage

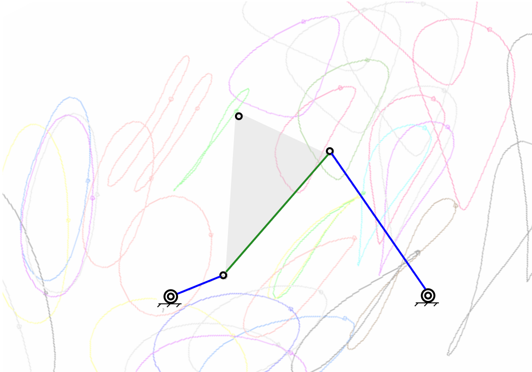
- 4 revolute joints
- 4 links (including basis)
- Closed kinematic chain
- One degree of freedom



Source: https://en.wikipedia.org/wiki/Four-bar_linkage

Simple Case: Four Bar Linkage

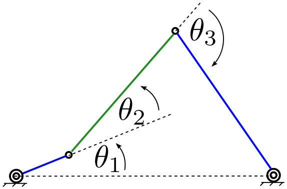
- 4 revolute joints
- 4 links (including basis)
- Closed kinematic chain
- One degree of freedom
- Quite “complex” motion of the 2nd link



Source: https://en.wikipedia.org/wiki/Four-bar_linkage

Simple Case: Four Bar Linkage

- 4 revolute joints
- 4 links (including basis)
- Closed kinematic chain
- One degree of freedom (e. g. θ_1)
- Quite “complex” motion of the 2nd link
- $\exists$ functional dependencies “$\theta_2(\theta_1)$” and “$\theta_3(\theta_1)$” in closed form



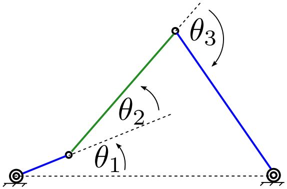
Source: https://en.wikipedia.org/wiki/Four-bar_linkage

Four Bar Linkage – Modeling Options

Option 1: Use functional dependencies $\theta_2(\theta_1)$, $\theta_3(\theta_1)$

- One independent variable: θ_1

$\Rightarrow$ Lagrange Eq. of 2nd kind applicable $\rightarrow$ ODE System ($n = 2$)



Four Bar Linkage – Modeling Options

Option 1: Use functional dependencies $\theta_2(\theta_1)$, $\theta_3(\theta_1)$

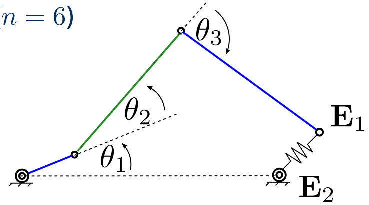
- One independent variable: θ_1

$\Rightarrow$ Lagrange Eq. of 2nd kind applicable $\rightarrow$ ODE System ($n = 2$)

Option 2: Chain closing via virtual spring

- Three independent variable: $\theta_1, \ldots, \theta_3$

$\Rightarrow$ Lagrange Eq. of 2nd kind applicable$\rightarrow$ (stiff) ODE System ($n = 6$)



Four Bar Linkage – Modeling Options

Option 1: Use functional dependencies $\theta_2(\theta_1)$, $\theta_3(\theta_1)$

- One independent variable: θ_1

$\Rightarrow$ Lagrange Eq. of 2nd kind applicable $\rightarrow$ ODE System ($n = 2$)

Option 2: Chain closing via virtual spring

- Three independent variable: $\theta_1, \ldots, \theta_3$

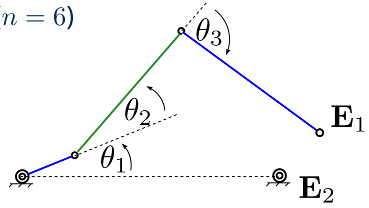
$\Rightarrow$ Lagrange Eq. of 2nd kind applicable$\rightarrow$ (stiff) ODE System ($n = 6$)

Option 3: Keep algebraic condition $\mathbf{E}_1 - \mathbf{E}_2 \overset{!}{=} \mathbf{0}$

- Three independent variables: $\theta_1, \ldots, \theta_3$

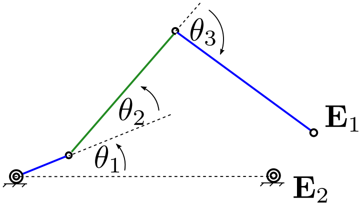
$\Rightarrow$ Lagrange Eq. of 1st kind necessary

$\Rightarrow$ DAE System, $n_\mathbf{x} = 6$, $n_\lambda = 2$, Index: 3



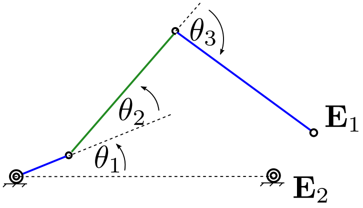
Lagrange Equations of 1st Kind

- Algebraic condition: $\mathbf{E}_1(\boldsymbol{\theta}) - \mathbf{E}_2(\boldsymbol{\theta}) =: \mathbf{a}(\boldsymbol{\theta}) \overset{!}{=} \mathbf{0}$



Lagrange Equations of 1st Kind

- Algebraic condition: $\mathbf{E}_1(\boldsymbol{\theta}) - \mathbf{E}_2(\boldsymbol{\theta}) =: \mathbf{a}(\boldsymbol{\theta}) \overset{!}{=} \mathbf{0}$
- Jacobian: $\mathbf{A}(\boldsymbol{\theta}) := \frac{\partial \mathbf{a}}{\partial \boldsymbol{\theta}}$

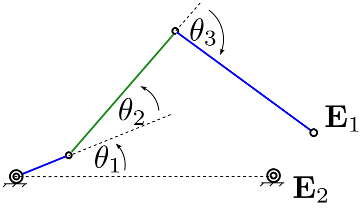


Lagrange Equations of 1st Kind

- Algebraic condition: $\mathbf{E}_1(\boldsymbol{\theta}) - \mathbf{E}_2(\boldsymbol{\theta}) =: \mathbf{a}(\boldsymbol{\theta}) \overset{!}{=} \mathbf{0}$
- Jacobian: $\mathbf{A}(\boldsymbol{\theta}) := \frac{\partial \mathbf{a}}{\partial \boldsymbol{\theta}}$
- Equations of motion:

$$\underbrace{\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{K}(\boldsymbol{\theta})}_{=:\mathbf{L}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})} = \boldsymbol{\tau} - \mathbf{A}^T(\boldsymbol{\theta})\boldsymbol{\lambda} = \mathbf{H}(\boldsymbol{\theta}) \underbrace{\begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{\lambda} \end{pmatrix}}_{\boldsymbol{\mu}},$$

$$\mathbf{a}(\boldsymbol{\theta}) = \mathbf{0}$$

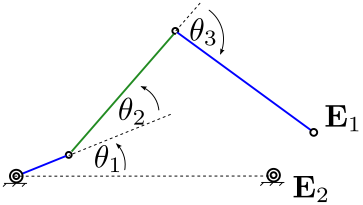


Lagrange Equations of 1st Kind

- Algebraic condition: $\mathbf{E}_1(\boldsymbol{\theta}) - \mathbf{E}_2(\boldsymbol{\theta}) =: \mathbf{a}(\boldsymbol{\theta}) \stackrel{!}{=} \mathbf{0}$
- Jacobian: $\mathbf{A}(\boldsymbol{\theta}) := \frac{\partial \mathbf{a}}{\partial \boldsymbol{\theta}}$
- Equations of motion:

$$\underbrace{\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{K}(\boldsymbol{\theta})}_{=:\mathbf{L}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}},\ddot{\boldsymbol{\theta}})} = \boldsymbol{\tau} - \mathbf{A}^T(\boldsymbol{\theta})\boldsymbol{\lambda} = \mathbf{H}(\boldsymbol{\theta}) \underbrace{\left(\begin{smallmatrix}\boldsymbol{\tau} \\ \boldsymbol{\lambda}\end{smallmatrix}\right)}_{\boldsymbol{\mu}},$$

$$\mathbf{a}(\boldsymbol{\theta}) = \mathbf{0}$$



- State space form of ODE-part: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\mu})$, with $\mathbf{x} = \begin{pmatrix}\boldsymbol{\theta} \\ \dot{\boldsymbol{\theta}}\end{pmatrix}$
- Index 3 $\Rightarrow$ Finding initial values and simulation nontrivial

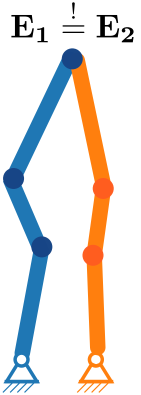
Lagrange Equations of 1st Kind

- Algebraic condition: $\mathbf{E}_1(\boldsymbol{\theta}) - \mathbf{E}_2(\boldsymbol{\theta}) =: \mathbf{a}(\boldsymbol{\theta}) \overset{!}{=} \mathbf{0}$
- Jacobian: $\mathbf{A}(\boldsymbol{\theta}) := \frac{\partial \mathbf{a}}{\partial \boldsymbol{\theta}}$
- Equations of motion:

$$\underbrace{\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{K}(\boldsymbol{\theta})}_{=:\mathbf{L}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})} = \boldsymbol{\tau} - \mathbf{A}^T(\boldsymbol{\theta})\boldsymbol{\lambda} = \mathbf{H}(\boldsymbol{\theta}) \underbrace{\left(\begin{smallmatrix}\boldsymbol{\tau} \\ \boldsymbol{\lambda}\end{smallmatrix}\right)}_{\boldsymbol{\mu}},$$

$$\mathbf{a}(\boldsymbol{\theta}) = \mathbf{0}$$

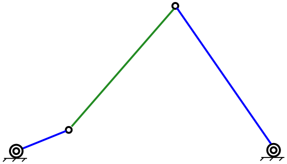
- State space form of ODE-part: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\mu})$, with $\mathbf{x} = \begin{pmatrix}\boldsymbol{\theta} \\ \dot{\boldsymbol{\theta}}\end{pmatrix}$
- Index 3 $\Rightarrow$ Finding initial values and simulation nontrivial



Side note: Degree of Freedom

General formula:

$$F = (N_{\text{links}} - 1) \cdot N_{\text{m. p.}} - \sum_{i=1}^{N_{\text{joints}}} (N_{\text{m. p.}} - f_i) \quad \text{with } N_{\text{m. p.}}\text{: motion possibilities}$$



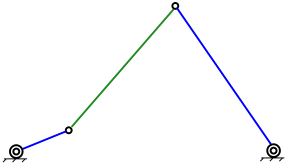
Side note: Degree of Freedom

General formula:

$$F = (N_{\text{links}} - 1) \cdot N_{\text{m. p.}} - \sum_{i=1}^{N_{\text{joints}}} (N_{\text{m. p.}} - f_i) \quad \text{with } N_{\text{m. p.}}\text{: motion possibilities}$$

Planar linkage with only revolute joints:

$$F = (N_{\text{links}} - 1) \cdot 3 - \sum_{i=1}^{N_{\text{joints}}} (3 - 1) = (N_{\text{links}} - 1) \cdot 3 - N_{\text{joints}} \cdot 2$$



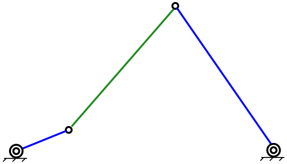
Side note: Degree of Freedom

General formula:

$$F = (N_{\text{links}} - 1) \cdot N_{\text{m. p.}} - \sum_{i=1}^{N_{\text{joints}}} (N_{\text{m. p.}} - f_i)$$ with $N_{\text{m. p.}}$: motion possibilities

Planar linkage with only revolute joints:

$$F = (N_{\text{links}} - 1) \cdot 3 - \sum_{i=1}^{N_{\text{joints}}} (3 - 1) = (N_{\text{links}} - 1) \cdot 3 - N_{\text{joints}} \cdot 2$$



Single-loop closed chain: $N_{\text{links}} = N_{\text{joints}}$

$$F = N_{\text{joints}} - 3$$

Side note: Degree of Freedom

General formula:

$$F = (N_{\text{links}} - 1) \cdot N_{\text{m.p.}} - \sum_{i=1}^{N_{\text{joints}}} (N_{\text{m.p.}} - f_i) \quad \text{with } N_{\text{m.p.}}\text{: motion possibilities}$$

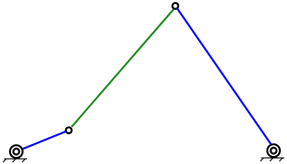
Planar linkage with only revolute joints:

$$F = (N_{\text{links}} - 1) \cdot 3 - \sum_{i=1}^{N_{\text{joints}}} (3 - 1) = (N_{\text{links}} - 1) \cdot 3 - N_{\text{joints}} \cdot 2$$

Single-loop closed chain: $N_{\text{links}} = N_{\text{joints}}$

$$F = N_{\text{joints}} - 3$$

$$F = 4 - 3 = 1$$



Side note: Degree of Freedom

General formula:

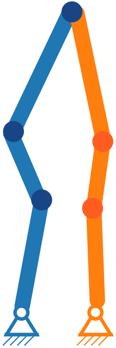
$$F = (N_{\text{links}} - 1) \cdot N_{\text{m.p.}} - \sum_{i=1}^{N_{\text{joints}}} (N_{\text{m.p.}} - f_i) \quad \text{with } N_{\text{m.p.}}\text{: motion possibilities}$$

Planar linkage with only revolute joints:

$$F = (N_{\text{links}} - 1) \cdot 3 - \sum_{i=1}^{N_{\text{joints}}} (3 - 1) = (N_{\text{links}} - 1) \cdot 3 - N_{\text{joints}} \cdot 2$$

Single-loop closed chain: $N_{\text{links}} = N_{\text{joints}}$

$$F = N_{\text{joints}} - 3$$



$$F = 7 - 3 = 4$$

Outline

☑ Motivation

☑ Modelling of Closed Kinematic Chains

→ **Motion Planning for Kinematic Chains**

☐ Simplification via Quasi-Stationary Trajectory Planing

☐ Conclusion and Outlook

Motion Planning via Optimization (1)

- Original DAE model: $$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \boldsymbol{\mu}) \quad \text{with } \mathbf{x} = \begin{pmatrix}\boldsymbol{\theta}\\ \dot{\boldsymbol{\theta}}\end{pmatrix} \text{ and } \boldsymbol{\mu} = \begin{pmatrix}\boldsymbol{\tau}\\ \boldsymbol{\lambda}\end{pmatrix}\\ \bar{\mathbf{a}}(\mathbf{x}) &= 0\end{aligned}$$

Motion Planning via Optimization (1)

- Original DAE model: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\mu})$ with $\mathbf{x} = \begin{pmatrix} \boldsymbol{\theta} \\ \dot{\boldsymbol{\theta}} \end{pmatrix}$ and $\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{\lambda} \end{pmatrix}$
  $\bar{\mathbf{a}}(\mathbf{x}) = 0$
- Euler discretization for N steps $k = 0, \ldots, N-1$:
  $\mathbf{x}_{k+1} = \Delta t \cdot \mathbf{f}(\mathbf{x}_k, \boldsymbol{\mu}_k)$ with $\mathbf{x}_k := \mathbf{x}(k\Delta t), \quad \boldsymbol{\mu}_k := \boldsymbol{\mu}(k\Delta t)$

Motion Planning via Optimization (1)

- Original DAE model: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\mu})$ with $\mathbf{x} = \begin{pmatrix} \boldsymbol{\theta} \\ \dot{\boldsymbol{\theta}} \end{pmatrix}$ and $\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{\lambda} \end{pmatrix}$
  $\bar{\mathbf{a}}(\mathbf{x}) = 0$
- Euler discretization for N steps $k = 0, \ldots, N-1$:
  $\mathbf{x}_{k+1} = \Delta t \cdot \mathbf{f}(\mathbf{x}_k, \boldsymbol{\mu}_k)$ with $\mathbf{x}_k := \mathbf{x}(k\Delta t)$, $\boldsymbol{\mu}_k := \boldsymbol{\mu}(k\Delta t)$
- Optimization variables: $\mathbf{Z} := (\mathbf{x}_0, \boldsymbol{\mu}_0, \ldots, \boldsymbol{\mu}_{N-1}, \mathbf{x}_N)$

Motion Planning via Optimization (1)

- Original DAE model: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\mu})$ with $\mathbf{x} = \begin{pmatrix} \boldsymbol{\theta} \\ \dot{\boldsymbol{\theta}} \end{pmatrix}$ and $\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{\lambda} \end{pmatrix}$
  $\bar{\mathbf{a}}(\mathbf{x}) = 0$
- Euler discretization for N steps $k = 0, \ldots, N-1$:
  $\mathbf{x}_{k+1} = \Delta t \cdot \mathbf{f}(\mathbf{x}_k, \boldsymbol{\mu}_k)$ with $\mathbf{x}_k := \mathbf{x}(k\Delta t)$, $\boldsymbol{\mu}_k := \boldsymbol{\mu}(k\Delta t)$
- Optimization variables: $\mathbf{Z} := (\mathbf{x}_0, \boldsymbol{\mu}_0, \ldots, \boldsymbol{\mu}_{N-1}, \mathbf{x}_N)$
- Cost function: $J(\mathbf{Z}) = \sum_{k=0}^{N-1} \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \boldsymbol{\mu}_k^T \mathbf{R} \boldsymbol{\mu}_k \rightarrow \min$

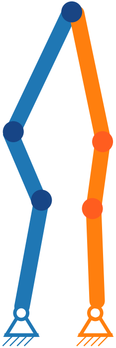
Motion Planning via Optimization (1)

- Original DAE model: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\mu})$ with $\mathbf{x} = \begin{pmatrix} \boldsymbol{\theta} \\ \dot{\boldsymbol{\theta}} \end{pmatrix}$ and $\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{\lambda} \end{pmatrix}$
  $\bar{\mathbf{a}}(\mathbf{x}) = 0$
- Euler discretization for N steps $k = 0, \dots, N-1$:
  $\mathbf{x}_{k+1} = \Delta t \cdot \mathbf{f}(\mathbf{x}_k, \boldsymbol{\mu}_k)$ with $\mathbf{x}_k := \mathbf{x}(k\Delta t)$, $\boldsymbol{\mu}_k := \boldsymbol{\mu}(k\Delta t)$
- Optimization variables: $\mathbf{Z} := (\mathbf{x}_0, \boldsymbol{\mu}_0, \dots, \boldsymbol{\mu}_{N-1}, \mathbf{x}_N)$
- Cost function: $J(\mathbf{Z}) = \sum_{k=0}^{N-1} \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \boldsymbol{\mu}_k^T \mathbf{R} \boldsymbol{\mu}_k \rightarrow \min$
- Constraints:
  - Discretized equations of $\mathbf{c}_{1,k}(\mathbf{Z}) := \Delta t \cdot \mathbf{f}(\mathbf{x}_k, \boldsymbol{\mu}_k) - \mathbf{x}_{k+1} \overset{!}{=} 0$
  - Chain closing condition: $\mathbf{c}_{2,k}(\mathbf{Z}) := \bar{\mathbf{a}}(\mathbf{x}_k) \overset{!}{=} 0$

Motion Planning via Optimization (2)

How big ist the problem (e. g. for $N = 100$)?

$$\dim \mathbf{c}_{1,k} = \dim \mathbf{x} = 2 \cdot \dim \boldsymbol{\theta} = 12,$$
$$\dim \mathbf{c}_{2,k} = \dim \mathbf{a} = 2$$

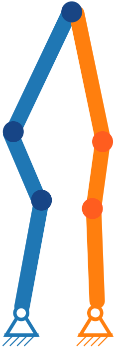


Motion Planning via Optimization (2)

How big ist the problem (e. g. for $N = 100$)?

$$\dim \mathbf{c}_{1,k} = \dim \mathbf{x} = 2 \cdot \dim \boldsymbol{\theta} = 12,$$
$$\dim \mathbf{c}_{2,k} = \dim \mathbf{a} = 2$$

Number of constraints: $N_\mathsf{c} = N \cdot \dim \mathbf{c}_{1,k} + N \cdot \dim \mathbf{c}_{2,k} = 1400$



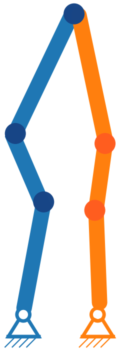
Motion Planning via Optimization (2)

How big ist the problem (e. g. for $N = 100$)?

$$\dim \mathbf{c}_{1,k} = \dim \mathbf{x} = 2 \cdot \dim \boldsymbol{\theta} = 12,$$
$$\dim \mathbf{c}_{2,k} = \dim \mathbf{a} = 2$$

Number of constraints: $N_\mathrm{c} = N \cdot \dim \mathbf{c}_{1,k} + N \cdot \dim \mathbf{c}_{2,k} = 1400$

Number of variables: $\dim \mathbf{Z} = (N+1) \cdot \dim \mathbf{x} + N \cdot \underbrace{\dim \boldsymbol{\mu}}_{6+2} = 2012$



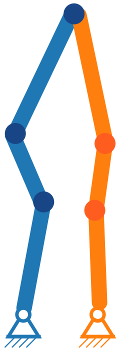
Motion Planning via Optimization (2)

How big ist the problem (e. g. for $N = 100$)?

$$\dim \mathbf{c}_{1,k} = \dim \mathbf{x} = 2 \cdot \dim \boldsymbol{\theta} = 12,$$
$$\dim \mathbf{c}_{2,k} = \dim \mathbf{a} = 2$$

Number of constraints: $N_\mathsf{c} = N \cdot \dim \mathbf{c}_{1,k} + N \cdot \dim \mathbf{c}_{2,k} = 1400$

Number of variables: $\dim \mathbf{Z} = (N+1) \cdot \dim \mathbf{x} + N \cdot \underbrace{\dim \boldsymbol{\mu}}_{6+2} = 2012$



$\rightarrow$ “Medium sized” nonlinear optimal control problem

Calculation time: $t_\mathsf{calc} \approx 1\,\mathrm{min} \ldots 1\,\mathrm{h}$

(Setup: CasADi, IPOPT, PC@3.2GHz)

Outline

☑ Motivation

☑ Modelling of Closed Kinematic Chains

☑ Motion Planning for Kinematic Chains

→ **Simplification via Quasi-Stationary Trajectory Planing**

☐ Conclusion and Outlook

Reconsidering the Problem

Equations of motion :

$$\begin{aligned} \mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{K}(\boldsymbol{\theta}) &= \boldsymbol{\tau} - \mathbf{A}^T(\boldsymbol{\theta})\boldsymbol{\lambda} = \mathbf{H}(\boldsymbol{\theta}) \underbrace{\begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{\lambda} \end{pmatrix}}_{\boldsymbol{\mu}}, \\ \mathbf{a}(\boldsymbol{\theta}) &= \mathbf{0} \end{aligned} \qquad (*)$$

Reconsidering the Problem

Equations of motion (assuming slow motion):

$$\underbrace{\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})}_{\approx \mathbf{0}} + \mathbf{K}(\boldsymbol{\theta}) = \boldsymbol{\tau} - \mathbf{A}^T(\boldsymbol{\theta})\boldsymbol{\lambda} = \mathbf{H}(\boldsymbol{\theta}) \underbrace{\begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{\lambda} \end{pmatrix}}_{\boldsymbol{\mu}},$$

$$\mathbf{a}(\boldsymbol{\theta}) = \mathbf{0} \qquad (*)$$

Reconsidering the Problem

Equations of motion (assuming slow motion):

$$
\left.\begin{aligned}
\mathbf{K}(\boldsymbol{\theta}) &= \mathbf{H}(\boldsymbol{\theta})\boldsymbol{\mu}, \\
\mathbf{a}(\boldsymbol{\theta}) &= \mathbf{0}
\end{aligned}\right. \qquad (*)
$$

New (smaller) problem:

Find equilibrium configuration $\boldsymbol{\theta}$ and load $\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{\lambda} \end{pmatrix}$ such that $(*)$ is satisfied.

Reconsidering the Problem

Equations of motion (assuming slow motion):

$$\begin{aligned} \mathbf{K}(\boldsymbol{\theta}) &= \mathbf{H}(\boldsymbol{\theta})\boldsymbol{\mu}, \\ \mathbf{a}(\boldsymbol{\theta}) &= \mathbf{0} \end{aligned} \qquad (*)$$

New (smaller) problem:

Find equilibrium configuration $\boldsymbol{\theta}$ and load $\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{\lambda} \end{pmatrix}$ such that $(*)$ is satisfied.

Idea: patch neighboring equilibria together $\rightarrow$ definition of geometric path
$\rightarrow$ choose time scaling $t \mapsto \boldsymbol{\theta}(t)$ (additional design freedom)

Equilibrium Finding

Original cost function

$$J(\mathbf{Z}) = \sum_{k=0}^{N-1} \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \boldsymbol{\mu}_k^T \mathbf{R} \boldsymbol{\mu}_k$$

constraints

$\mathbf{c}_{1,k}, \mathbf{c}_{2,k}$ for $k = 1, \ldots, N$

Equilibrium Finding

Original cost function

$$J(\mathbf{Z}) = \sum_{k=0}^{N-1} \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \boldsymbol{\mu}_k^T \mathbf{R} \boldsymbol{\mu}_k$$

constraints

$\mathbf{c}_{1,k}, \mathbf{c}_{2,k}$ for $k = 1, \ldots, N$

New cost function

$$\bar{J}_k(\bar{\mathbf{Z}}_k, \bar{\mathbf{P}}_k) = \boldsymbol{\mu}_k^T \mathbf{R} \boldsymbol{\mu}_k + \sigma(\bar{\mathbf{P}}_k)$$

constraints

$(\bar{\mathbf{c}}_{1,k}, \bar{\mathbf{c}}_{2,k}, \bar{\mathbf{c}}_{3,k}(\bar{\mathbf{P}}_k))$

for k fixed

Equilibrium Finding

Original cost function

$J(\mathbf{Z}) = \sum_{k=0}^{N-1} \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \boldsymbol{\mu}_k^T \mathbf{R} \boldsymbol{\mu}_k$

constraints

$\mathbf{c}_{1,k}, \mathbf{c}_{2,k}$ for $k = 1, \ldots, N$

New cost function

$\bar{J}_k(\bar{\mathbf{Z}}_k, \bar{\mathbf{P}}_k) = \boldsymbol{\mu}_k^T \mathbf{R} \boldsymbol{\mu}_k + \sigma(\bar{\mathbf{P}}_k)$

with $\bar{\mathbf{Z}}_k := (\mathbf{x}_k, \boldsymbol{\mu}_k)$, $\dim \bar{\mathbf{Z}}_k = 14$

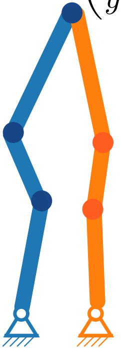
and parameters $\bar{\mathbf{P}}_k := (y_{\mathrm{E},k}, \bar{\mathbf{Z}}_{k-1}, \bar{\mathbf{Z}}_{k-2})$

constraints

$(\bar{\mathbf{c}}_{1,k}, \bar{\mathbf{c}}_{2,k}, \bar{\mathbf{c}}_{3,k}(\bar{\mathbf{P}}_k))$

for k fixed

$\mathbf{E} = \begin{pmatrix} x_{\mathrm{E}} \\ y_{\mathrm{E}} \end{pmatrix}$



Equilibrium Finding

Original cost function

$$J(\mathbf{Z}) = \sum_{k=0}^{N-1} \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \boldsymbol{\mu}_k^T \mathbf{R} \boldsymbol{\mu}_k$$

constraints

$\mathbf{c}_{1,k}, \mathbf{c}_{2,k}$ for $k = 1, \ldots, N$

New cost function

$$\bar{J}_k(\bar{\mathbf{Z}}_k, \bar{\mathbf{P}}_k) = \boldsymbol{\mu}_k^T \mathbf{R} \boldsymbol{\mu}_k + \sigma(\bar{\mathbf{P}}_k)$$

constraints

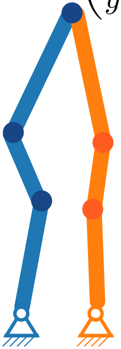
$(\bar{\mathbf{c}}_{1,k}, \bar{\mathbf{c}}_{2,k}, \bar{\mathbf{c}}_{3,k}(\bar{\mathbf{P}}_k))$ for k fixed

with $\bar{\mathbf{Z}}_k := (\mathbf{x}_k, \boldsymbol{\mu}_k)$, $\dim \bar{\mathbf{Z}}_k = 14$

and parameters $\bar{\mathbf{P}}_k := (y_{\mathsf{E},k}, \bar{\mathbf{Z}}_{k-1}, \bar{\mathbf{Z}}_{k-2})$

$\rightarrow$ For each k increase y_{E} (“standing up”) via $\bar{\mathbf{c}}_{3,k}$ and penalize changes in $\bar{\mathbf{Z}}$

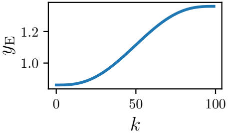
$$\mathbf{E} = \begin{pmatrix} x_{\mathrm{E}} \\ y_{\mathrm{E}} \end{pmatrix}$$



Trajectory Generation

For each $k = 0, \dots, N$:

1. Choose y_E from

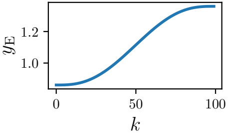


2. Solve small nonlinear optimization problem $\bar{J}_k(\bar{\mathbf{Z}}_k, \bar{\mathbf{P}}_k) \to \min$ s. t. $\bar{\mathbf{c}}_{i,k}$

Trajectory Generation

For each $k = 0, \ldots, N$:

① Choose y_E from



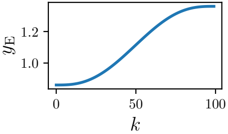
② Solve small nonlinear optimization problem $\bar{J}_k(\bar{\mathbf{Z}}_k, \bar{\mathbf{P}}_k) \to \min$ s. t. $\bar{\mathbf{c}}_{i,k}$

③ Calculate $\bar{\mathbf{P}}_{k+1}$ and $\bar{\mathbf{c}}_{i,k+1}$ according to results

④ $k := k + 1 \to$ ①

Trajectory Generation

For each $k = 0, \ldots, N$:

1. Choose y_E from 

2. Solve small nonlinear optimization problem $\bar{J}_k(\bar{\mathbf{Z}}_k, \bar{\mathbf{P}}_k) \to \min$ s. t. $\bar{\mathbf{c}}_{i,k}$

3. Calculate $\bar{\mathbf{P}}_{k+1}$ and $\bar{\mathbf{c}}_{i,k+1}$ according to results

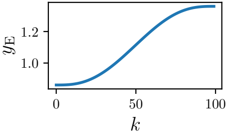
4. $k := k + 1 \to$ 1

Result: geometric path $k \mapsto \boldsymbol{\theta}_k$,
static torques $k \mapsto \boldsymbol{\mu}_k$

Trajectory Generation

For each $k = 0, \ldots, N$:

1. Choose y_E from



2. Solve small nonlinear optimization problem $\bar{J}_k(\bar{\mathbf{Z}}_k, \bar{\mathbf{P}}_k) \to \min$ s. t. $\bar{\mathbf{c}}_{i,k}$
3. Calculate $\bar{\mathbf{P}}_{k+1}$ and $\bar{\mathbf{c}}_{i,k+1}$ according to results
4. $k := k + 1 \to$ 1

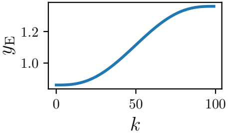
Result: geometric path $k \mapsto \boldsymbol{\theta}_k$, *static* torques $k \mapsto \boldsymbol{\mu}_k$

5. Choose time scaling $t \mapsto k(t)$
6. Calculate $t \mapsto (\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t), \ddot{\boldsymbol{\theta}}(t))$ (via interpolation)

Trajectory Generation

For each $k = 0, \ldots, N$:

1. Choose y_E from



2. Solve small nonlinear optimization problem $\bar{J}_k(\bar{\mathbf{Z}}_k, \bar{\mathbf{P}}_k) \to \min$ s. t. $\bar{\mathbf{c}}_{i,k}$
3. Calculate $\bar{\mathbf{P}}_{k+1}$ and $\bar{\mathbf{c}}_{i,k+1}$ according to results
4. $k := k + 1 \to$ 1

Result: geometric path $k \mapsto \boldsymbol{\theta}_k$, *static* torques $k \mapsto \boldsymbol{\mu}_k$

5. Choose time scaling $t \mapsto k(t)$
6. Calculate $t \mapsto (\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t), \ddot{\boldsymbol{\theta}}(t))$ (via interpolation)
7. Calculate actual *dynamic* torques

$$\boldsymbol{\mu}^\star(t) = \arg\min_{\boldsymbol{\mu}} \boldsymbol{\mu}^T \mathbf{R} \boldsymbol{\mu}$$

$$\text{s.t.}$$

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{K}(\boldsymbol{\theta}) = \mathbf{H}(\boldsymbol{\theta})\boldsymbol{\mu}$$

for each desired time step t

Results

Calculation time:

“1 big promblem ($N = 100$)”: $t_{\text{calc}} \approx 1\,\mathrm{min} \ldots 1\,\mathrm{h}$ (reminder)

Results

Calculation time:

"1 big promblem ($N = 100$)": $t_{\text{calc}} \approx 1\,\text{min} \ldots 1\,\text{h}$ (reminder)

$N = 100$ "small promblems": $t_{\text{calc}} \approx 0.5\,\text{s} \ldots 1\,\text{s}$

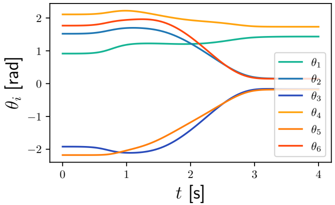
Results

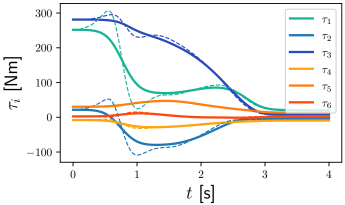
Calculation time:

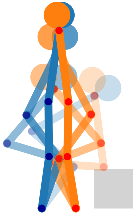
“1 big promblem ($N = 100$)”: $t_{\text{calc}} \approx 1\,\text{min} \ldots 1\,\text{h}$ (reminder)

$N = 100$ “small promblems”: $t_{\text{calc}} \approx 0.5\,\text{s} \ldots 1\,\text{s}$

Numerical Results:







See also: `https://github.com/TUD-RST/MPCKC`

Outline

- ☑ Motivation
- ☑ Modelling of Closed Kinematic Chains
- ☑ Motion Planning for Kinematic Chains
- ☑ Simplification via Quasi-Stationary Trajectory Planing
- → **Conclusion and Outlook**

Conclusion and Outlook

Summary:

- Modelling of closed kinematic chains via Lagrange Eq. of 1st kind
- Formulation as optimal control problem
- Simplification via sequential solution of quasi-stationary auxiliary problems
  $\rightarrow \approx 1000\times$ speedup
- Python based software frame work: `https://github.com/TUD-RST/MPCKC`

Conclusion and Outlook

Summary:

- Modelling of closed kinematic chains via Lagrange Eq. of 1st kind
- Formulation as optimal control problem
- Simplification via sequential solution of quasi-stationary auxiliary problems
  $\rightarrow \approx 1000\times$ speedup
- Python based software frame work: `https://github.com/TUD-RST/MPCKC`

Outlook:

- Parameter studies (lengths, masses, $\mathbf{R}_{ii}$)
- More realistic humans (e. g. with arms)
- Consideration of tilting (ZMP)