FAKULTÄT FÜR ELEKTROTECHNIK UND INFORMATIONSTECHNIK
Faculty of Electrical Engineering and Information Technology

ACIN
AUTOMATION & CONTROL INSTITUTE
INSTITUT FÜR AUTOMATISIERUNGS- & REGELUNGSTECHNIK

Computational Performance of the Forward and Inverse Kinematics of an Anthropomorphic Robot Arm

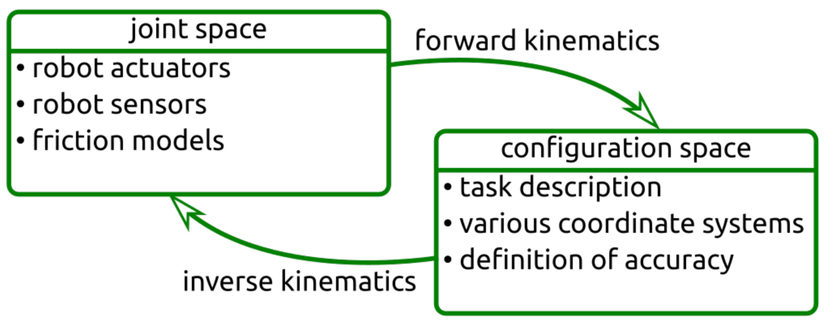
Christian Hartl-Nesic, Martin Meiringer
{hartl,meiringer}@acin.tuwien.ac.at

TU Wien
Automation and Control Institute (ACIN)
Group for Complex Dynamic Systems

Supervisor: Georg Nawratil
Institute of Discrete Mathematics and Geometry
Group for Differential Geometry and Geometric Structures

Introduction

In robotics, two fundamental coordinate spaces, the joint space and the configuration space are utilized.



Modern applications require fast and efficient kinematic algorithms.

Methods

The computation performance in terms of costs and exectution times are compared for the forward and inverse kinematics of a KUKA LWR using different approaches.

The computation cost is quantified by the number of additions, multiplications and function calls required to execute an algorithm

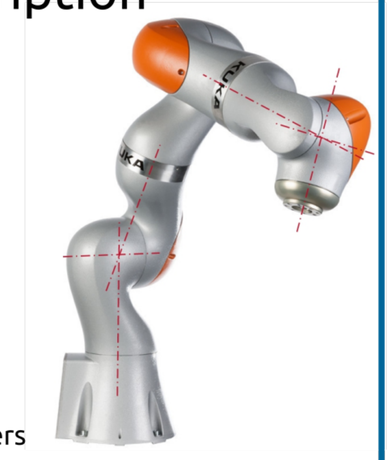
Software
- Maple 2018
- MATLAB 2017b

Hardware
- Office PC
- Intel Core i7-6700K
- 16 GB RAM

System Description

i	a_i	α_i	d_i	θ_i
1	0	$\pi/2$	0	q_1
2	0	$\pi/2$	0	q_2
3	0	$\pi/2$	l_1	q_3
4	0	$\pi/2$	0	q_4
5	0	$\pi/2$	l_2	q_5
6	0	$\pi/2$	0	q_6
7	0	0	0	q_7

Table I. Denavit-Hartenberg parameters



The kinematics of a KUKA LWR 7DOF serial robot is investigated in this work. This serial chain comprises a spherical 'sholder' joint, a rotational 'elbow' joint and a spherical 'wrist'.

Homogeneous Coordinates

A discrete transformation comprising a rotation and a translation between two coordinate frames using homogeneous coordinates is given by the 4x4 matrix $\mathbf{T}$

$$\mathbf{T} = \begin{bmatrix} 1 & \mathbf{0}^{\mathrm{T}} \\ \mathbf{d} & \mathbf{R} \end{bmatrix} \qquad \begin{aligned} \mathbf{d} &\ldots \text{translation vector} \\ \mathbf{R} &\ldots \text{rotation matrix} \end{aligned}$$

forward kinematics

$$\mathbf{f}_H(\boldsymbol{\theta}) = \ \mathbf{T}_0^N(\boldsymbol{\theta}) = \mathbf{T}_0^1(\theta_1)\mathbf{T}_1^2(\theta_2)\cdots\mathbf{T}_{N-1}^N(\theta_N)$$

inverse kinematics

In the inverse kinematics problem, a given desired transformation $\mathbf{T}^\star$ is solved for the joint angles $\boldsymbol{\theta}$

$$\boldsymbol{\theta} = \mathbf{f}_H^{-1}(\mathbf{T}^\star)$$

The method presented in M. Pfurner, *"Closed form inverse kinematics solution for a redundant anthropomorphic robot arm", Oct. 2016*, is utilized for calculation of the forward and inverse kinematics in homogeneous coordinates. This method exploits the constructional benefits and intuitive geometric relations. For an algebraic solution, the trigonometric functions are substituted using the Weierstraß substitution.

Dual Quaternions

A discrete transformation comprising a rotation and a translation between two coordinate frames using dual quaternions is given by the dual quaternion $\mathbf{u}$

$$\mathbf{u} = \mathbf{q}_R + \frac{1}{2}\varepsilon\mathbf{q}_R \otimes \mathbf{q}_d \qquad \begin{aligned} \mathbf{q}_d &\ldots \text{translation quaternion} \\ \mathbf{q}_R &\ldots \text{rotation unit-quaternion} \\ \varepsilon &\ldots \text{dual unit} \\ \otimes &\ldots \text{quaternion multiplication} \end{aligned}$$

forward kinematics

$$\mathbf{f}_{dQ}(\boldsymbol{\theta}) = \ \mathbf{u}_0^N(\boldsymbol{\theta}) = \mathbf{u}_0^1(\theta_1) \otimes \mathbf{u}_1^2(\theta_2)\cdots\otimes \mathbf{u}_{N-1}^N(\theta_N)$$

inverse kinematics

In the inverse kinematics problem, a given a desired transformation $\mathbf{u}^\star$ solved for the joint angles $\boldsymbol{\theta}$

$$\boldsymbol{\theta} = \mathbf{f}_{dQ}^{-1}(\mathbf{u}^\star)$$

The method presented by Pfurner is reformulated using dual quaternions. The proposed method still applies, but the resulting equations become polynomials of 4$^{\text{th}}$ order. Solving these yields multiple invalid solutions, which have to be eliminated.

Substitutions for Algebraic Unity

The forward kinematics equations derived using homogeneous coordinates and dual quaternions lead to analytically identical expressions by applying the two substitutions on the right. The rotational quaternion utilizes $\sin(\cdot)$ and $\cos(\cdot)$ of the half rotation angle, while homogeneous coordinates use the whole rotation angle.

Weierstraß substitution

$$t = \tan\left(\frac{x}{2}\right), \quad \sin(x) = \frac{2t}{1+t^2}$$

$$\cos(x) = \frac{1-t^2}{1+t^2}$$

square of sin and cos

$$\sin^2\left(\frac{x}{2}\right) = \frac{1}{2}(1-\cos(x))$$

$$\cos^2\left(\frac{x}{2}\right) = \frac{1}{2}(1+\cos(x))$$

Performance Comparison

calculation	add	mult	fcn. calls	CPU time
$\tilde{\mathbf{f}}_H(\boldsymbol{\theta})$	88	238	327	4.1 µs
$\mathbf{f}_H(\boldsymbol{\theta})$	40	87	14	3.6 µs
$\tilde{\mathbf{f}}_{dQ}(\boldsymbol{\theta})$	312	1576	880	19.5 µs
$\mathbf{f}_{dQ}(\boldsymbol{\theta})$	38	88	14	4.4 µs
$\mathbf{f}_H^{-1}(\mathbf{T}^\star)$	1461	2570	52	86.1 µs
$\mathbf{f}_{dQ}^{-1}(\mathbf{T}^\star)$	1070	2258	184	173.1 µs

Table II. Performance for the forward and inverse kinematics

In Table II and Figure I, the accent (~) denotes a non-optimized code generation of the equations (using Maple).
As the results show, especially for a formulation in dual quaternions this optimization is crucial.

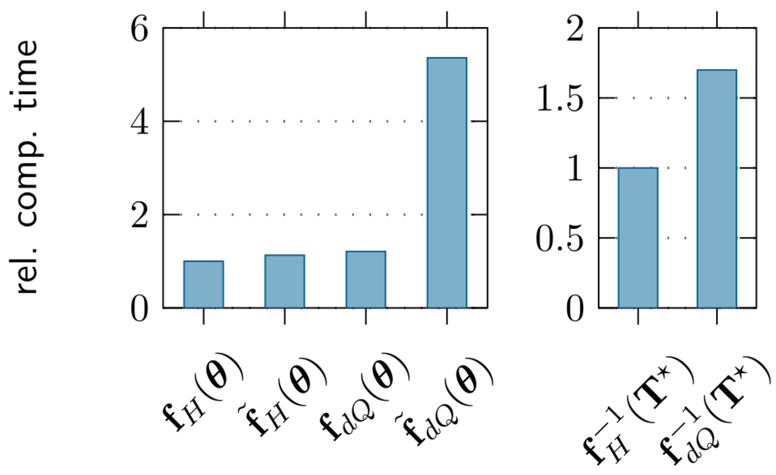


Figure I. Relative computation times for the forward and inverse kinematics