Supplemental Material for: Hierarchical Dual Quaternion-Based Recurrent Neural Network as a Flexible Internal Body Model

Malte Schilling
Center of Excellence Cognitive Interaction Technology,
Bielefeld University, Germany
mschilli@techfak.uni-bielefeld.de

Abstract

In the article, a hierarchical Mean of Multiple Computation neural network is presented that is based on an axis-angle representation of joint movements using dual quaternions. It is shown in detail how the network provides a solution for the forward kinematic problem applied for the case of a seven degrees of freedom robot manipulator. Furthermore, it is used in a complex scenario of a bimanual movement task. This demonstrates how the MMC approach can be easily scaled up from a representation of a single arm to a complex model of a complete body.

This supplement provides additional information for

- the definition of the variables
- illustration of the posture variations for the forward kinematic task
- the multiple redundant equations

Code implementation in Python can be found in https://github.com/malteschilling/hierarchicalQuaternionMMC.git .
More details on the Mean of Multiple Computation principle can be found at

- On dual quaternion representation in [1].
- On hierarchical MMC networks in [2], [3].

This research/work was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

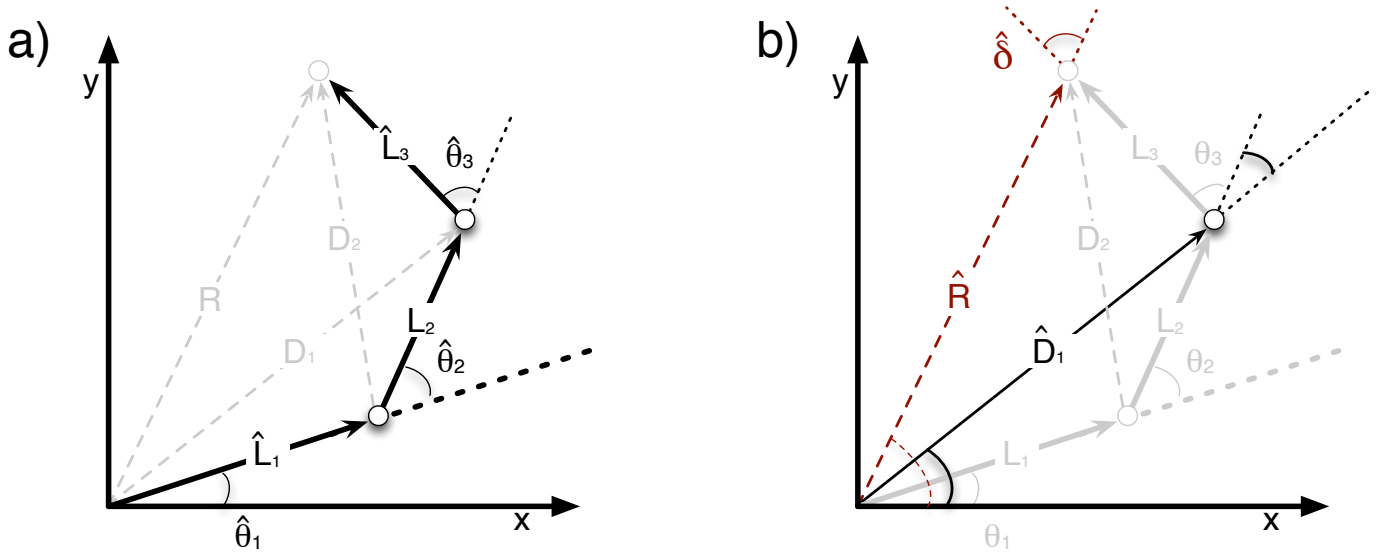


Fig. 1. Application of the MMC principle to a dual quaternion representation of a three segmented arm. The arm consists of three segments and has seven DoFs. In a) and b) different views of the arm are shown (segments are all unit length).

I. Dual Quaternion Variables and Equations

As an example, we will setup two equations for the first joint (list of all equations can be found in the appendix https://github.com/malteschillin The first joint contains three DoFs and is therefore described through an axis and an angle of rotation. It can be computed following two different ways (Fig. 1).

$$\begin{aligned}\hat{\theta}_{\mathbf{1}} &= \hat{\mathbf{D}}_{\mathbf{1}} * \left(\hat{\mathbf{L}}_{\mathbf{1}} * \hat{\theta}_{\mathbf{2}} * \hat{\mathbf{L}}_{\mathbf{2}}\right)^{-1} \\ \hat{\theta}_{\mathbf{1}} &= \hat{\mathbf{R}} * \hat{\delta} * \left(\hat{\mathbf{D}}_{\mathbf{2}} * \hat{\mathbf{L}}_{\mathbf{1}}\right)^{-1}\end{aligned} \tag{1}$$

For notation, we use greek letters for rotations; and for diagonals, end effector and segments the name of the vectors as used earlier. Importantly, the diagonals and end effector dual quaternions represent combined rotations and translations which align the orientations of the different connected coordinate system.

These two equations provide different ways to compute the first joint variable. Following the MMC principle these multiple computations are integrated. In contrast to other representation formalisms, dual quaternions provide a meaningful way [4] to interpolate between different transformations while following the shortest path and being unambiguous [4], [5]. We employ Dual quaternion Linear Blending (DLB) which interpolates dual quaternions on a component basis and afterwards normalizes the result to produce a valid dual quaternion [6]. Also included in the integration is the current value of the variable which is fed back into the system through the recurrent connection. In this way, we end up with a new value for a variable after a time step:

$$\begin{aligned}\hat{\theta}_{\mathbf{1}}(\mathbf{t}+\mathbf{1}) = DLB([\{w_1, \hat{\mathbf{D}}_{\mathbf{1}} * (\hat{\mathbf{L}}_{\mathbf{1}} * \hat{\theta}_{\mathbf{2}} * \hat{\mathbf{L}}_{\mathbf{2}})^{-1}\}, \\ \{w_2, \hat{\mathbf{R}} * \hat{\delta} * (\hat{\mathbf{D}}_{\mathbf{2}} * \hat{\mathbf{L}}_{\mathbf{1}})^{-1})\}, \{w_3, \hat{\theta}_{\mathbf{1}}(\mathbf{t})\}]);\ \sum_i w_i = 1\end{aligned} \tag{2}$$

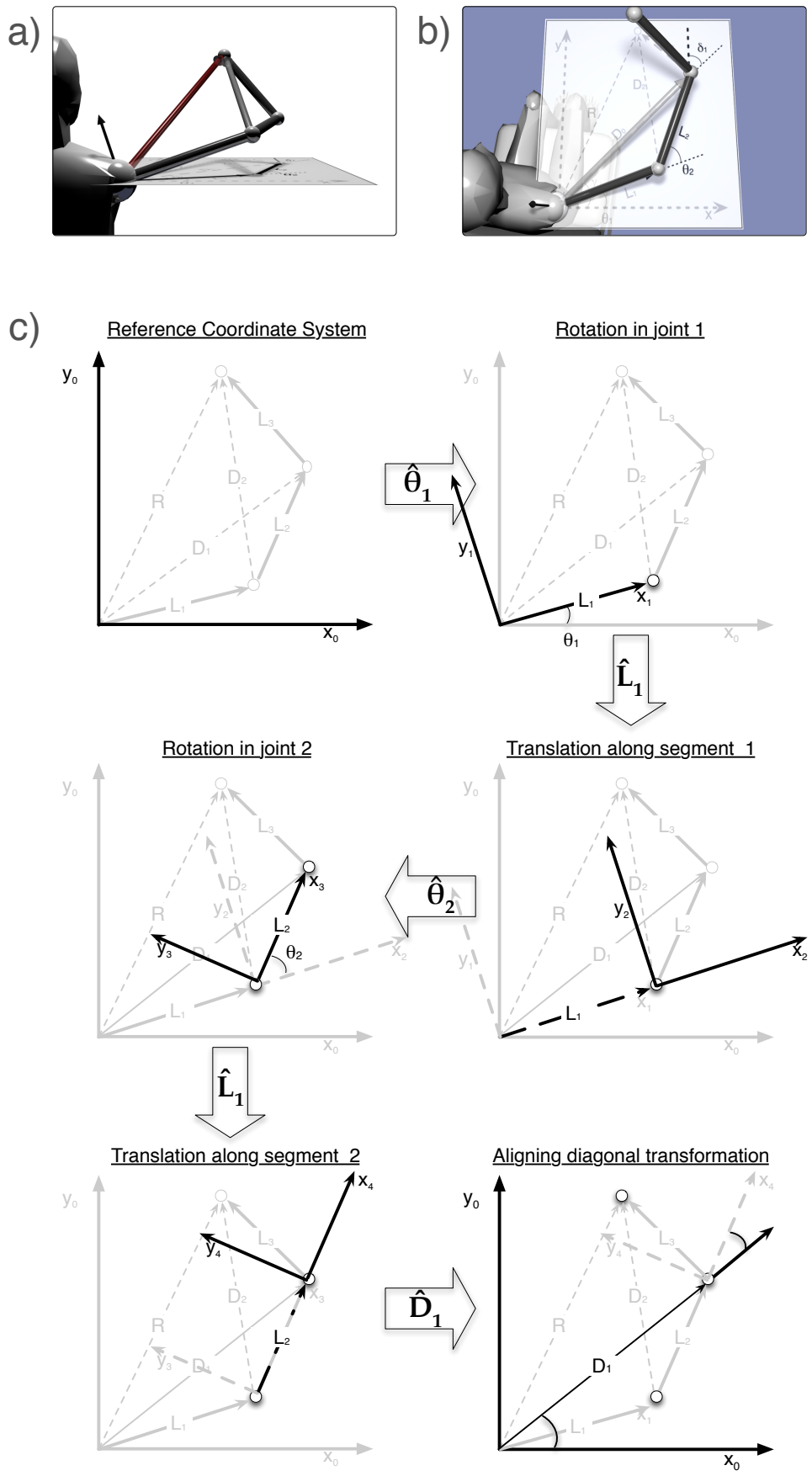


Fig. 2. Application of the MMC principle to a dual quaternion representation of a three segmented arm. The arm consists of three segments and has seven degrees of freedom. In a) and b) different views of the arm are shown (segments are all unit length). In c) one local equation is derived as a chain of transformations (subsequent rotations and translations).

II. Experiment: Convergence for Forward Kinematic Function

In the first set of simulations, a single human-like arm was used and controlled through given joint positions. The simulated arm consisted of three segments, all were assumed to be unit length. Overall, the arm had seven DoFs (like a human arm). The first joint connected the arm to a fixed base (the shoulder joint) and comprised three DoFs. The second or elbow joint was restricted to one degree of freedom and the third joint, again, could be freely moved in three dimensions. Given a joint configuration, the task for the internal model was to estimate the current position in space of the tip of the arm. We tested the internal model on a large number of movements between different postures. In each simulation run the internal model was driven from a starting posture to one of two end postures: On the one hand, a fully stretched out arm, leading for many starting postures to large movements across the workspace. On the other hand, a completely folded arm which is usually a difficult to reach posture requiring movements in all joints. The movements were controlled on the joint level. For each joint the start and end joint rotation was given and the movement had to be completed in 25 simulation steps. The in between joint rotations were obtained through interpolation using Spherical Linear Interpolation (SLERP) of the dual quaternions representing the rotation angle and axis [7]. In this way all the joints were driven for 25 iteration steps and the movement followed the shortest path. During the movement (and for an additional 25 time steps afterwards) we recorded the internal arm models estimate of the end position.

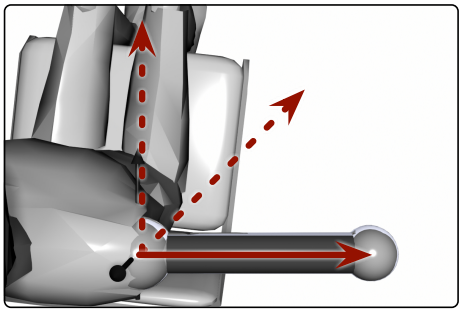
We defined a set of postures by variation of the degrees of freedom. The different postures and joint representations for the joints were derived through sequential concatenation of the individual transformations. For the first joint there are three DoFs and for each of those dimensions we used three different rotational positions (possible values for rotations: around z-axis 0°; 45°; 90°; around y-axis 0°; 45°; 90°; around x-axis -45°; 0°; 45°). For the elbow joint there is only one degree of freedom (possible values: 0°; 45°; 90°) and for the third joint we only looked at two DoFs as we are not interested in the orientation of the manipulator in this simulation (possible values of rotation: around z-axis 0°; 45°; 90°; around y-axis -45°; 0°; 45°). This results in 729 postures and, importantly, movements starting from these postures covers the whole working range.

The different possible values for each rotation are given in table I.

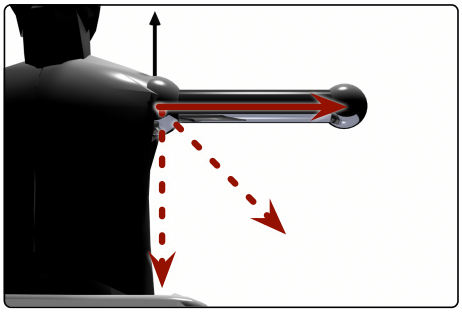
DoF number	Joint	Axis of Rotation	Values
1	Shoulder	z-axis	0°; 45°; 90°
2		y-axis	0°; 45°; 90°
3		x-axis	-45°; 0°; 45°
4	Elbow	z-axis	0°; 45°; 90°
5	Hand	z-axis	0°; 45°; 90°
6		y-axis	-45°; 0°; 45°

TABLE I
VARIATIONS OF INITIAL POSTURES IN THE DIFFERENT JOINTS.

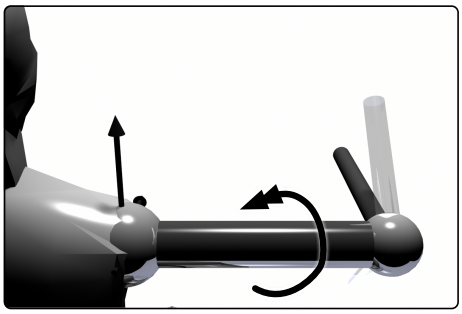
a) View from above, variation around z-axis in the first joint



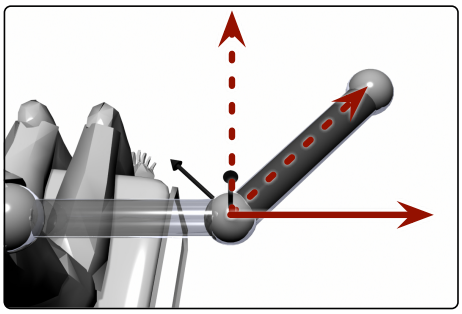
b) View from behind, variation around y-axis in the first joint



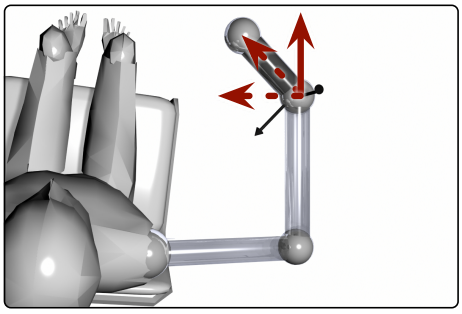
c) View from behind, variation around x-axis in the first joint (around segment)



d) View from above, variation around z-axis in the second joint



e) View from above, variation around z-axis in the third joint



f) View from the side, variation around y-axis in the third joint (up and down)

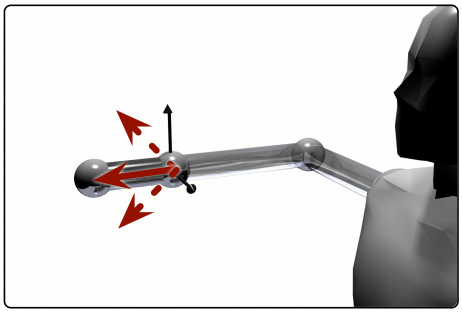


Fig. 3. Variation of the arm posture: produced through variation of joint angles. The red lines show the different postures for that dimension. a) to c) show variations of the first joint (shoulder joint): the different variations are subsequently applied (each variation defines a rotation): a) shows rotations around the z-axis (pointing upwards), b) rotations around y-axis (moving the arm up and down) and c) is a rotation around the segment (the x-axis). The second joint is restricted to one degree of freedom (z-axis) and is shown in d). Subfigure e) and f) depict the variation in the third joint: e) shows rotations around the z-axis (pointing upwards in the fully stretched out arm), f) rotation around y-axis (moving the hand up and down).

III. Appendix

Multiple computations of the different variables expressed in dual quaternions.
The first joint:

$$\begin{aligned}\hat{\theta}_{\mathbf{1}} &= \hat{\mathbf{D}}_{\mathbf{1}} * \left(\hat{\mathbf{L}}_{\mathbf{1}} * \hat{\theta}_{\mathbf{2}} * \hat{\mathbf{L}}_{\mathbf{2}}\right)^{-1} \\ \hat{\theta}_{\mathbf{1}} &= \hat{\mathbf{R}} * \hat{\delta} * \left(\hat{\mathbf{D}}_{\mathbf{2}} * \hat{\mathbf{L}}_{\mathbf{1}}\right)^{-1}\end{aligned} \tag{3}$$

The second joint:

$$\begin{aligned}\hat{\theta}_{\mathbf{2}} &= \hat{\mathbf{D}}_{\mathbf{2}} * \left(\hat{\mathbf{L}}_{\mathbf{2}} * \hat{\theta}_{\mathbf{3}} * \hat{\mathbf{L}}_{\mathbf{3}}\right)^{-1} \\ \hat{\theta}_{\mathbf{2}} &= \left(\hat{\theta}_{\mathbf{1}} * \hat{\mathbf{L}}_{\mathbf{1}}\right)^{-1} * \hat{\mathbf{D}}_{\mathbf{1}} * \hat{\mathbf{L}}_{\mathbf{2}}^{-1}\end{aligned} \tag{4}$$

The third joint:

$$\begin{aligned}\hat{\theta}_{\mathbf{3}} &= \left(\hat{\theta}_{\mathbf{2}} * \hat{\mathbf{L}}_{\mathbf{2}}\right)^{-1} * \hat{\mathbf{D}}_{\mathbf{2}} * \hat{\mathbf{L}}_{\mathbf{3}}^{-1} \\ \hat{\theta}_{\mathbf{3}} &= \hat{\mathbf{D}}_{\mathbf{1}}^{-1} * \hat{\mathbf{R}} * \hat{\delta} * \hat{\mathbf{L}}_{\mathbf{3}}^{-1}\end{aligned} \tag{5}$$

The two diagonals:

$$\begin{aligned}\hat{\mathbf{D}}_{\mathbf{1}} &= \hat{\theta}_{\mathbf{1}} * \hat{\mathbf{L}}_{\mathbf{1}} * \hat{\theta}_{\mathbf{2}} * \hat{\mathbf{L}}_{\mathbf{2}} \\ \hat{\mathbf{D}}_{\mathbf{1}} &= \hat{\mathbf{R}} * \hat{\delta} * \left(\hat{\theta}_{\mathbf{3}} * \hat{\mathbf{L}}_{\mathbf{3}}\right)^{-1}\end{aligned} \tag{6}$$

$$\begin{aligned}\hat{\mathbf{D}}_{\mathbf{2}} &= \hat{\theta}_{\mathbf{2}} * \hat{\mathbf{L}}_{\mathbf{2}} * \hat{\theta}_{\mathbf{3}} * \hat{\mathbf{L}}_{\mathbf{3}} \\ \hat{\mathbf{D}}_{\mathbf{2}} &= \left(\hat{\theta}_{\mathbf{1}} * \hat{\mathbf{L}}_{\mathbf{1}}\right)^{-1} * \hat{\mathbf{R}} * \hat{\delta}\end{aligned} \tag{7}$$

The variables representing the end effector position and orientation:

$$\begin{aligned}\hat{\mathbf{R}} &= \hat{\theta}_{\mathbf{1}} * \hat{\mathbf{L}}_{\mathbf{1}} * \hat{\mathbf{D}}_{\mathbf{1}} * \hat{\delta}^{-1} \\ \hat{\mathbf{R}} &= \hat{\mathbf{D}}_{\mathbf{1}} * \hat{\theta}_{\mathbf{3}} * \hat{\mathbf{L}}_{\mathbf{3}} * \hat{\delta}^{-1}\end{aligned} \tag{8}$$

$$\begin{aligned}\hat{\delta} &= \hat{\mathbf{R}}^{-1} * \hat{\theta}_{\mathbf{1}} * \hat{\mathbf{L}}_{\mathbf{1}} * \hat{\mathbf{D}}_{\mathbf{1}} \\ \hat{\delta} &= \hat{\mathbf{R}}^{-1} * \hat{\mathbf{D}}_{\mathbf{1}} * \hat{\theta}_{\mathbf{3}} * \hat{\mathbf{L}}_{\mathbf{3}}\end{aligned} \tag{9}$$

REFERENCES

[1] M. Schilling, "Universally manipulable body models — dual quaternion representations in layered and dynamic MMCs," *Autonomous Robots*, vol. 30, no. 4, pp. 399–425, 2011.

[2] M. Schilling, J. Paskarbeit, J. Schmitz, A. Schneider, and H. Cruse, "Grounding an internal body model of a hexapod walker – control of curve walking in a biological inspired robot," in *Proc. of IEEE/RSJ IROS 2012*, 2012, pp. 2762–2768.

[3] M. Schilling and H. Cruse, "Hierarchical MMC Networks as a manipulable body model," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2007), Orlando, FL*, 2007, pp. 2141–2146.

[4] N. A. Aspragathos and J. K. Dimitros, "A comparative study of three methods for robot kinematics," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 28, no. 2, pp. 135–145, 1998.

[5] J. Funda and R. Paul, "A computational analysis of screw transformations in robotics," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 348–356, 1990.

[6] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan, "Geometric skinning with approximate dual quaternion blending," *ACM Transactions on Graphics*, vol. 27, no. 4, p. 105, 2008.

[7] K. Shoemake, "Animating rotation with quaternion curves," in *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 1985, pp. 245–254.