Extending the Cooperative Dual-Task Space in Conformal Geometric Algebra

Tobias Löw*† and Sylvain Calinon*†
*Idiap Research Institute, Martigny, Switzerland
†EPFL, Lausanne, Switzerland

***Abstract*—In this work, we are presenting an extension of the cooperative dual-task space (CDTS) in conformal geometric algebra. The CDTS was first defined using dual quaternion algebra and is a well established framework for the simplified definition of tasks using two manipulators. By integrating conformal geometric algebra, we aim to further enhance the geometric expressiveness and thus simplify the modeling of various tasks. We show this formulation by first presenting the CDTS and then its extension that is based around a cooperative pointpair. This extension keeps all the benefits of the original formulation that is based on dual quaternions, but adds more tools for geometric modeling of the dual-arm tasks. We also present how this CGA-CDTS can be seamlessly integrated with an optimal control framework in geometric algebra that was derived in previous work. In the experiments, we demonstrate how to model different objectives and constraints using the CGA-CDTS. Using a setup of two Franka Emika robots we then show the effectiveness of our approach using model predictive control in real world experiments.**

***Index Terms*—Geometric Algebra, Dual-Arm Manipulation, Optimal Control**

I. INTRODUCTION

With the increasing desire to deploy robots in human environemnts, the need for robots to have human-like manipulation capabilities arises. One inherent ability that humans have is to manipulate objects using both their hands and arms, which is needed for example when objects that are either too large or too heavy need to be manipulated. In order to match these capabilities and to be able to mimic them, robotic systems also need to be able to cooperatively control two arms in order to perform tasks in human environments.

Apart from dual-arm systems being more human-like in terms of form factor, they also have some technical advantages. One can have the stiffness and strength of parallel manipulators combined with the flexibility and dexterity of serial manipulators [1]. Furthermore, since they increase the redundancy in the task-space due to their high number of degrees of freedom, they are better suited for intricate tasks that require a high manipulability such as screw assembly [2] and dishwashing [3]. Other applications of bimanual systems are manipulating articulated objects [4] or cables [5]. More advantages and examples are listed in [6].

This work was supported by the State Secretariat for Education, Research and Innovation in Switzerland for participation in the European Commission's Horizon Europe Program through the INTELLIMAN project (https://intelliman-project.eu/, HORIZON-CL4-Digital-Emerging Grant 101070136) and the SESTOSENSO project (http://sestosenso.eu/, HORIZON-CL4-Digital-Emerging Grant 101070310).

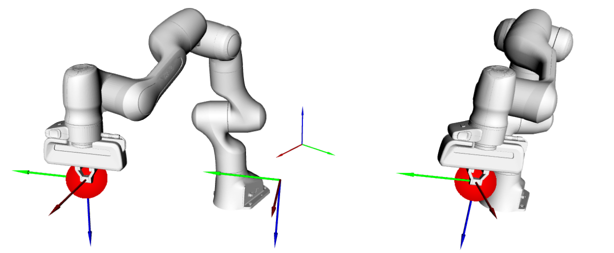


Fig. 1: Cooperative dual-task space using conformal geometric algebra. The figure shows the two manipulators as well as their individual, relative and absolute motors. Additionally, it shows the cooperative pointpair.

Since many problems in robotics boil down to optimization problems that can be solved efficiently with various state-of-the-art solvers, it is of great interest to facilitate the modeling and increase the expressiveness of the formulations. Choosing the correct representation can make a huge difference in terms of how much prior knowledge we can embed into the formulation of those optimization problems. These are in robotics often very geometric, hence it is very beneficial to choose representations that intuitively allow to incoporate the geometry of the problem. For the case of dual-arm manipulation, the cooperative dual-task space (CDTS) [7] was proposed. This approach uses dual quaternion algebra (DQA), which not only unifies the treatment of position and orientation, it also allows the representation of various geometric primitives [8]. These primitives can then be used to simplify the modeling of the tasks.

Dual quaternions have a strong connection to geometric algebra, especially the variants known as projective (PGA) and conformal (CGA) geometric algebra [9], since dual quaternions are isomorphically embedded in their sub-algebras [10]. The geometric algebras, however, are richer algebras that offer more geometric primitives and, more importantly, they offer the geometric construction of primitives based on operations such as intersections [10]. This leads to new possibilities when formulating objectives and constraints based on the geometric primitives in the CGA-CDTS compared to the DQ-CDTS, which we will show in the experiments.

In this article, we formulate the CDTS in conformal geometric algebra and show how this formulation naturally extends the DQ-CDTS. The resulting CGA-CDTS retains the same properties of a compact representation of two arm system as the DQ-CDTS, while adding a useful geometric primitive, the cooperative pointpair, that represents both end-effector positions simultaneously. Furthermore, we demonstrate how the CGA-CDTS can be used in the optimal control formulation with geometric primitives for manipulators that we presented in [11]. Hence, this article aims to explain the basic mathematical formulations of various optimization problems using the CGA-CDTS.

II. BACKGROUND

A. Cooperative Dual-Task Space

We will introduce the cooperative dual-task space here conceptually, and not mathematically, since its original definition uses dual quaternion algebra. In contrast to that, we will be defining and extending it using conformal geometric algebra. So, for the sake of conciseness, we are keeping this introduction on a high level.

The cooperative dual-task space was proposed as a compact and singularity-free representation of a two-arm system [7]. It is defined using two poses that depend on the end-effector poses of the two manipulators, one is the relative and the other one is the absolute pose. All poses are represented using unit dual quaternions, which have advantages compared to other representations, such as coupled position and orientation, singularity-free representation and efficient computation [12].

Based on the compact representation of the CDTS, various control strategies have been proposed. In [13], a coupled task-space admittance controller was presented, that allowed for a geometrically consistent stiffness term. A reactive control strategy was developed in [14] that leveraged geometric primitves for task relaxations and priorities. Task priorities for control in the CDTS were also proposed in [15]. In order to exploit human demonstration that allow the teaching of cooperative motions, motion primitives for bimanual systems were presented based on the CDTS [16].

Note that the results of the research that is based on the DQ-CDTS can also be used with the CGA-CDTS, albeit with mathematical changes due to the different algebra. Furthermore, the mentioned advantages of DQA also apply to CGA, since dual quaternions and the corresponding subalgebra in CGA, i.e. the motors, are isomorphic [17].

B. Geometric Algebra

Geometric algebra is a single algebra for geometric reasoning, alleviating the need of utilizing multiple algebras to express geometric relations. In this article, we are using the variant known as conformal geometric algebra (CGA). We use the following notation: x to denote scalars, $\boldsymbol{x}$ for vectors, $\boldsymbol{X}$ for matrices, X for multivectors and $\boldsymbol{\mathcal{X}}$ for matrices of multivectors.

A general element for computation in geometric algebra is called a multivector. There are three main products that can be used with multivectors: the geometric product XY, the inner product $X \cdot Y$ and the outer product $X \wedge Y$. The trivial vector case shows that the geometric product combines the inner $\cdot$ and the outer $\wedge$ product

$$\boldsymbol{ab} = \boldsymbol{a} \cdot \boldsymbol{b} + \boldsymbol{a} \wedge \boldsymbol{b}. \tag{1}$$

The outer product is a spanning operation that allows the creation of geometric primitives from points P_i. For example, two points $P_1 \wedge P_2$ yield a point pair, three points $P_1 \wedge P_2 \wedge P_3$ a circle and four points $P_1 \wedge P_2 \wedge P_3 \wedge P_4$ a sphere. There are more primitives, that we will introduce when we use them in the experiments.

Rigid body transformations are represented by motors M. They can be applied to multivectors by a sandwiching operation, similar to how quaternions rotate vectors

$$Y = MX\widetilde{M}, \tag{2}$$

where $\widetilde{M}$ stands for the reverse of a motor. Motors are exponential mappings of so-called bivector (i.e. the subspaces spanned by the outer product of two vectors), the inverse operation is the logarithmic map

$$M = \exp(B) \quad \iff \quad B = \log(M). \tag{3}$$

The motor $M(\boldsymbol{q})$ corresponding to the forward kinematics of a kinematic chain can be computed as the product of the individual joint motors

$$M(\boldsymbol{q}) = \prod_{i=1}^{N} M_i(q_i). \tag{4}$$

The analytic Jacobian can then be found as the derivative of the forward kinematics motor defined in Equation (4), i.e.

$$\boldsymbol{\mathcal{J}}^A(\boldsymbol{q}) = \frac{\partial M(\boldsymbol{q})}{\partial \boldsymbol{q}} = \begin{bmatrix} \frac{\partial M(\boldsymbol{q})}{\partial q_1} & \dots & \frac{\partial M(\boldsymbol{q})}{\partial q_N} \end{bmatrix}. \tag{5}$$

At this point, we only introduced the most important concepts for understanding the proposed method. For a complete introduction we refer interested readers to [18] and [19].

III. METHOD

The CDTS in conformal geometric algebra is an extension of the CDTS in dual quaternion algebra. We first present the basic reformulation of the CDTS in CGA and then its extension by using the additional geometric primitives. Lastly, we present how the CGA-CDTS can be used within an optimal control framework using geometric algebra that we previously proposed for manipulation tasks.

A. Conformal Geometric Algebra Cooperative Dual-Task Space

The geometric algebra equivalent of the relative and absolute dual quaternions of the DQ-CDTS are defined using motors. Given the joint configurations of the two manipulators, $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, respectively, we can easily find their end-effector motors $M_1(\boldsymbol{q}_1)$ and $M_2(\boldsymbol{q}_2)$ using the forward kinematics in CGA. From this it is straightforward to formulate the relative motor as

$$M_r(\boldsymbol{q}_1, \boldsymbol{q}_2) = \widetilde{M}_2(\boldsymbol{q}_2) M_1(\boldsymbol{q}_1), \tag{6}$$

while its Jacobian, i.e. the relative analytic Jacobian, can be found as

$$\mathcal{J}_r^A(\boldsymbol{q}_1,\boldsymbol{q}_2) = \left[\widetilde{M}_2(\boldsymbol{q}_2)\mathcal{J}_1^A(\boldsymbol{q}_1) \quad \widetilde{\mathcal{J}}_2^A(\boldsymbol{q}_2)M_1(\boldsymbol{q}_1)\right]. \tag{7}$$

Similarly, the absolute motor can be found as

$$\begin{aligned} M_a(\boldsymbol{q}_1,\boldsymbol{q}_2) &= M_2(\boldsymbol{q}_2)M_{r/2}(\boldsymbol{q}_1,\boldsymbol{q}_2) \\ &= M_2(\boldsymbol{q}_2)\exp\left(\frac{1}{2}\log\left(M_r(\boldsymbol{q}_1,\boldsymbol{q}_2)\right)\right), \end{aligned} \tag{8}$$

with its corresponding absolute analytic Jacobian

$$\begin{aligned} \mathcal{J}_a^A(\boldsymbol{q}_1,\boldsymbol{q}_2) =& M_2(\boldsymbol{q}_2)\mathcal{J}_{M_{r/2}}^A(\boldsymbol{q}_1,\boldsymbol{q}_2) \\ &+ \left[\boldsymbol{0} \quad \mathcal{J}_2^A(\boldsymbol{q}_2)M_{r/2}(\boldsymbol{q}_1,\boldsymbol{q}_2)\right], \end{aligned} \tag{9}$$

where $\mathcal{J}_{M_{r/2}}^A(\boldsymbol{q}_1,\boldsymbol{q}_2)$ is the analytic Jacobian of the motor $M_{r/2}(\boldsymbol{q}_1,\boldsymbol{q}_2)$. It can be found as

$$\mathcal{J}_{M_{r/2}}^A(\boldsymbol{q}_1,\boldsymbol{q}_2) = \boldsymbol{J}_{\mathbb{B}\to\mathcal{M}}(B_{r/2})\boldsymbol{J}_{\mathcal{M}\to\mathbb{B}}(M_r)\boldsymbol{J}_r^A(\boldsymbol{q}_1,\boldsymbol{q}_2). \tag{10}$$

Here, $B_{r/2}$ is the logarithm of the motor $M_{r/2}$. The matrices $\boldsymbol{J}_{\mathbb{B}\to\mathcal{M}}$ and $\boldsymbol{J}_{\mathcal{M}\to\mathbb{B}}$ are the Jacobians of the exponential and logarithmic mapping respectively. We already showed the derivation of the Jacobian of the logarithmic mapping in the appendix of [11]. The Jacobian of the exponential mapping can be found in Appendix A.

Both the relative and the absolute analytic Jacobians are $1 \times 2N$ multivector matrices that contain motors as their elements. Hence, when expanding it to normal matrix algebra they become $8 \times 2N$ matrices.

Since motors in CGA can be used to transform any geometric primitive that is part of the algebra in a uniform way, it is easy to find cooperative geometric primitives. Their definition can be trivially found using Equation (2), where M is either the relative $M_r(\boldsymbol{q}_1,\boldsymbol{q}_2)$ or absolute $M_a(\boldsymbol{q}_1,\boldsymbol{q}_2)$ motor and X can be any geometric primitive. The corresponding Jacobians are then found using the respective Jacobians $\mathcal{J}_r^A(\boldsymbol{q}_1,\boldsymbol{q}_2)$ and $\mathcal{J}_a^A(\boldsymbol{q}_1,\boldsymbol{q}_2)$.

B. Cooperative Pointpair

In extension to the CDTS that was defined using dual quaternion algebra, the CDTS presented here using CGA also allows a geometric primitive that corresponds to both end-effector positions simultaneously. This cooperative pointpair is defined as the outer product of the two end-effector points, i.e.

$$P_{cdts} = M_1(\boldsymbol{q}_1)\boldsymbol{e}_0\widetilde{M}_1(\boldsymbol{q}_1) \wedge M_2(\boldsymbol{q}_2)\boldsymbol{e}_0\widetilde{M}_2(\boldsymbol{q}_2). \tag{11}$$

The Jacobian of the cooperative pointpair can be found as

$$\mathcal{J}_{P_{cdts}} = \left[\mathcal{J}_{P_{cdts},1} \quad \mathcal{J}_{P_{cdts},2}\right], \tag{12}$$

where

$$\begin{aligned} \mathcal{J}_{P_{cdts},1} =& \mathcal{J}_1^A(\boldsymbol{q}_1)\boldsymbol{e}_0\widetilde{M}_1(\boldsymbol{q}_1) \wedge M_2(\boldsymbol{q}_2)\boldsymbol{e}_0\widetilde{M}_2(\boldsymbol{q}_2) \\ &+ M_1(\boldsymbol{q}_1)\boldsymbol{e}_0\widetilde{\mathcal{J}}_1^A(\boldsymbol{q}_1) \wedge M_2(\boldsymbol{q}_2)\boldsymbol{e}_0\widetilde{M}_2(\boldsymbol{q}_2), \end{aligned} \tag{13}$$

and

$$\begin{aligned} \mathcal{J}_{P_{cdts},2} =& M_1(\boldsymbol{q}_1)\boldsymbol{e}_0\widetilde{M}_1(\boldsymbol{q}_1) \wedge \mathcal{J}_2^A(\boldsymbol{q}_2)\boldsymbol{e}_0\widetilde{M}_2(\boldsymbol{q}_2) \\ &+ M_1(\boldsymbol{q}_1)\boldsymbol{e}_0\widetilde{M}_1(\boldsymbol{q}_1) \wedge M_2(\boldsymbol{q}_2)\boldsymbol{e}_0\widetilde{\mathcal{J}}_2^A(\boldsymbol{q}_2). \end{aligned} \tag{14}$$

Note that the cooperative pointpair is a direct representation of both points and is not the same as stacking the two points. Therefore the Jacobian matrix is also different, which will lead to different solutions of optimization problems. An example of this is shown in Figure 2, where two Franka Emika robots are tasked to reach a plane, once individually (i.e. by stacking their end-effector points) and once cooperatively (i.e. by using the cooperative pointpair that is presented here). It can be seen that the corresponding solution configurations are not the same, which shows that the cooperative pointpair representation lets the two robots influence each other.

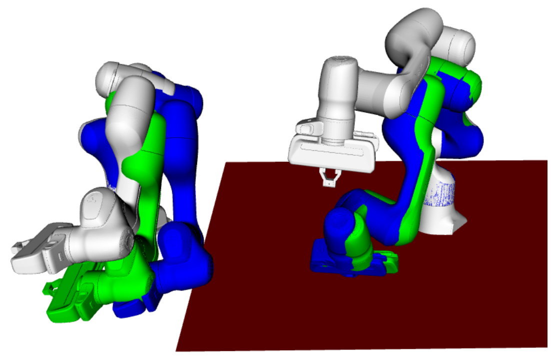


Fig. 2: Dual-Arm manipulator reaching a plane. The target plane is shown in red. The white Franka Emika robots show the initial configurations, the green ones are the result of individually reaching the plane, and the blue ones cooperatively.

Evidently, this cooperative pointpair Jacobian has a singularity in the case when both end-effector positions are equal. This will cause the outer product, by definition, to be zero. In practice, this can be avoided by posing a constraint on the distance between end-effectors, which usually is required anyways.

C. Using the CGA-CDTS for Optimal Control

Integrating the CGA-CDTS into the optimal control framework that we presented in [11] can be achieved by replacing the single end-effector motor with the relative and absolute motors, respectively. The objective of reaching a target motor is then formulated as

$$E_M(\boldsymbol{q}_1,\boldsymbol{q}_2) = \log\left(\widetilde{M}_{target}M(\boldsymbol{q}_1,\boldsymbol{q}_2)\right), \tag{15}$$

where $M(\boldsymbol{q}_1,\boldsymbol{q}_2)$ can be either $M_r(\boldsymbol{q}_1,\boldsymbol{q}_2)$ or $M_a(\boldsymbol{q}_1,\boldsymbol{q}_2)$.

Alternatively, we can define a residual multivector for reaching a geometric primitive X_d as

$$E_{X_d}(\boldsymbol{q}_1,\boldsymbol{q}_2) = X_d \wedge M(\boldsymbol{q}_1,\boldsymbol{q}_2)X\widetilde{M}(\boldsymbol{q}_1,\boldsymbol{q}_2), \tag{16}$$

again using either the relative or absolute motor. Deriving the respective Jacobians is straightforward using the definitions of the relative and absolute analytic Jacobians in Equations (7) and (9). With this, these residual multivectors of the CGA-CDTS can then directly be used to define objectives or constraints for optimal control problems, which would mean for example that a relative or absolute geometric primitive should be reached.

The cooperative pointpair can also be used in order to define tasks as optimization problems. The first way to do so is using the outer product in a similar way to the above relative and absolute residual multivectors, i.e.

$$E_{cdts}(\boldsymbol{q}_1, \boldsymbol{q}_2) = P_{target} \wedge P_{cdts}(\boldsymbol{q}_1, \boldsymbol{q}_2), \tag{17}$$

with the Jacobian

$$\boldsymbol{\mathcal{J}}_{E_{cdts}}(\boldsymbol{q}_1, \boldsymbol{q}_2) = P_{target} \wedge \boldsymbol{\mathcal{J}}_{P_{cdts}}(\boldsymbol{q}_1, \boldsymbol{q}_2), \tag{18}$$

This objective means that the two manipulators should cooperatively reach a single point. The implications and results of this objective are further detailed in the experiment section.

Another common use-case are containment relationships for the cooperative pointpair with respect to other geometric primitives. These can then be used to define tasks where the dual-arm system should cooperatively reach a target. The mathematical formulation is using a product called the commutator product $\times$, i.e.

$$\begin{aligned} E_{cdts}(\boldsymbol{q}_1, \boldsymbol{q}_2) &= X_d \times P_{cdts}(\boldsymbol{q}_1, \boldsymbol{q}_2) \\ &= \frac{1}{2}(X_d P_{cdts}(\boldsymbol{q}_1, \boldsymbol{q}_2) - P_{cdts}(\boldsymbol{q}_1, \boldsymbol{q}_2) X_d). \end{aligned} \tag{19}$$

The Jacobian of this containment relationship can be found as

$$\boldsymbol{\mathcal{J}}_{E_{cdts}}(\boldsymbol{q}_1, \boldsymbol{q}_2) = X_d \times \boldsymbol{\mathcal{J}}_{E_{cdts}}(\boldsymbol{q}_1, \boldsymbol{q}_2). \tag{20}$$

The interpretation of the residual multivector $E_{cdts}(\boldsymbol{q}_1, \boldsymbol{q}_2)$ is that it should be reduced to zero if the cooperative pointpair is contained within the desired geometric primitive X_d, e.g. if both end-effector points lie on a circle.

The distance between the two end-effector can be constrained using the inner product between the two end-effector points, i.e.

$$E_d(\boldsymbol{q}_1, \boldsymbol{q}_2) = -2 M_1(\boldsymbol{q}_1) \boldsymbol{e}_0 \widetilde{M}_1(\boldsymbol{q}_1) \cdot M_2(\boldsymbol{q}_2) \boldsymbol{e}_0 \widetilde{M}_2(\boldsymbol{q}_2) - d^2, \tag{21}$$

where d is the desired distance.

IV. EXPERIMENTS

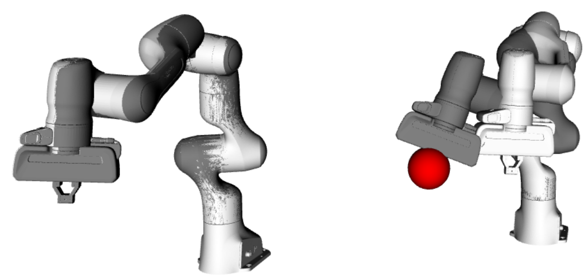
In this section, we are presenting various cooperative tasks that are defined in the CGA-CDTS. For each task we are providing the mathematical definition of the optimization problem. We are then solving those optimization problems using standard solvers such as Gauss-Newton. For simplicity, the problems in simulation are formulated essentially as inverse kinematics problems, the same objectives and constraints can, however, be used in optimal control problems that are for example then used for model predictive control. We are demonstrating this in the real world experiments, where the problems are then solved by using a variant of the iterative linear quadratic regulator (iLQR) [20]. Both the simulation and the real-world experiments use the same setup of two tabletop mounted Franka Emika robots. Additional material for the experiments as well as the videos of the real world experiments can be found on the accompanying website [1]. All geometric algebra computations are done using our open-source library *gafro*[2].

A. Cooperatively Reaching a Point

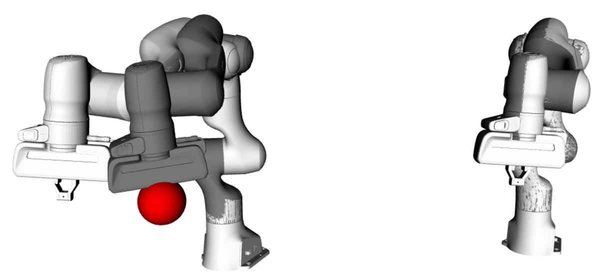
As mentioned before, the cooperative pointpair models both end-effector positions simultaneously. Hence, when formulating a simple optimization problem such as reaching a single point, the system automatically chooses which manipulator should perform the task. Since the information of both manipulators is encoded in one geometric primitive, it alleviates the need to construct the problem using conditional formulations. The problem can be expressed compactly using the outer product, i.e.

$$\boldsymbol{q}^* = \min_{\boldsymbol{q}} \Big\| P_{target} \wedge P_{cdts}(\boldsymbol{q}_1, \boldsymbol{q}_2) \Big\|_2^2. \tag{22}$$

An example of this task is shown in Figure 3. Notice how in both cases the manipulator that is closer to the point performs the reaching task while the other remains in its initial configuration.



(a) Left arm reaching for the point.



(b) Right arm reaching for the point.

Fig. 3: Two Franka Emika robots reaching for a single point in the cooperative dual-task space. The initial configurations are shown in white and the final ones in gray. The target point is shown in red.

[1] https://geometric-algebra.tobiloew.ch/cdts/
[2] https://github.com/idiap/gafro

We also show this experiment in using the real-world setup and the accompanying video can be found on our website.

B. Cooperatively Reaching a Circle

One of the geometric primitives that is available in CGA but not DQA is a circle. Hence, we can use the CGA-CDTS to define a task where a dual arm system should cooperatively reach a circle. An example application of reaching a circle would be holding a filled bucket with two manipulators. Mathematically, a circle is obtained by the outer product of three points.

The problem of cooperatively reaching a circle is formulated as a constrained optimization problem using the cooperative pointpair

$$\begin{aligned} \boldsymbol{q}^* = \min_{\boldsymbol{q}} & \left\| \log\left(\widetilde{M}_r(\boldsymbol{q}_0) M_r(\boldsymbol{q})\right) \right\|_2^2 \\ \text{s.t. } & C \times P_{cdts}(\boldsymbol{q}_1, \boldsymbol{q}_2) = 0. \end{aligned} \tag{23}$$

This optimization problem formulates the task of reaching a circle, while trying to maintain the initial relative motor. The result of this problem can be seen in Figure 4. Formulating the task in this manner circumvents the mentioned singularity issue in the Jacobian of the cooperative pointpair, since the system will reach circle while keeping as close as possible to their initial relative poses.

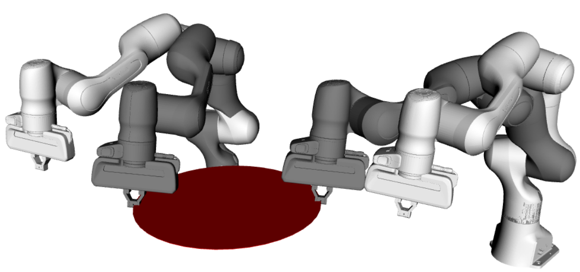


Fig. 4: Two Franka Emika robots reaching a circle in the cooperative dual-task space while trying to maintain the same relative motor.

C. Aligning Orientation Axis

When two manipulators are cooperatively manipulating an object, often it is necessary to partially constrain their relative orientation, which is necessary in nearly all dual-arm grasping scenarios of rigid objects. One way to do so is enforcing collinear lines in the desired direction at the end-effector level. This is again achieved by using the commutator product, i.e.

$$\begin{aligned} & E_{L_1 L_2}(\boldsymbol{q}_1, \boldsymbol{q}_2) \\ & \quad = M_1(\boldsymbol{q}_1) L_1 \widetilde{M}_1(\boldsymbol{q}_1) \times M_2(\boldsymbol{q}_2) L_2 \widetilde{M}_2(\boldsymbol{q}_2). \end{aligned} \tag{24}$$

In Figure 5, we show the results of minimizing $E_{L_1 L_2}(\boldsymbol{q}_1, \boldsymbol{q}_2)$, where $L_1 = L_2 = \boldsymbol{e}_0 \wedge (\boldsymbol{e}_0 + \boldsymbol{e}_1 + \frac{1}{2}\boldsymbol{e}_\infty) \wedge \boldsymbol{e}_\infty$, which corresponds to aligning two lines that pass through the x-axes of the frames at the end-effectors of the two manipulators.

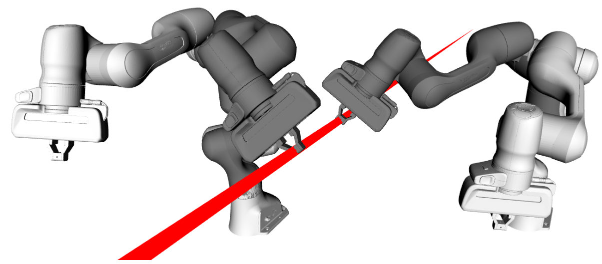


Fig. 5: Aligning the x-Axis.

There are, of course, other ways to achieve this behavior, e.g. by constraining the orientation part of the relative motor. We chose to demonstrate this method of aligning orientation, however, because it provides a lot of flexibility, since the two lines L_1 and L_2 can be chosen arbitrarily.

A similar instance of this problem can be defined using the absolute motor of the CGA-CDTS. By using the absolute translator $T_a(\boldsymbol{q}_1, \boldsymbol{q}_2)$ we can move a desired line to the absolute position and require it to be identical to the line moved by the absolute motor. This defines a desired orientation with respect to the arbitrary axis that is defined by the line. The formulation hence is

$$\begin{aligned} & E_{T_L}(\boldsymbol{q}_1, \boldsymbol{q}_2) \\ & \quad = T_a(\boldsymbol{q}_1, \boldsymbol{q}_2) L \widetilde{T}_a(\boldsymbol{q}_1, \boldsymbol{q}_2) \times M_a(\boldsymbol{q}_1, \boldsymbol{q}_2) L \widetilde{M}_a(\boldsymbol{q}_1, \boldsymbol{q}_2). \end{aligned} \tag{25}$$

D. Balancing a Plate

In this real-world experiment we are combining several of the previous constraints in order to implement the task of balancing a plate. First, the robot lifts the plate to a height of 20cm above the table, then it will try to keep it in that position. This is formulated as constrained optimization problem, where the objective is to stay close to the initial configuration and the constraint is to keep the z-axis of absolute motor perpendicular to the xy-plane.

We formulate this task as an optimal control problem

$$\begin{aligned} \boldsymbol{u}^* = \min_{\boldsymbol{u}} & \left\| E_N(\boldsymbol{x}_N) \right\|_2^2 + \sum_{k=0}^{N-1} \left\| E_k(\boldsymbol{x}_k) \right\|_2^2 + \|\boldsymbol{u}_k\|_R^2 \\ \text{s.t. } & \boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k) \end{aligned} \tag{26}$$

where $\boldsymbol{x} = \left[\boldsymbol{q}_1^\top, \boldsymbol{q}_2^\top, \dot{\boldsymbol{q}}_1^\top, \dot{\boldsymbol{q}}_2^\top\right]^\top$ and $\boldsymbol{u} = \left[\ddot{\boldsymbol{q}}_1^\top, \ddot{\boldsymbol{q}}_2^\top\right]^\top$. As the state dependent cost we choose the residual multivectors of Equations (24) and (25), where the line L is chosen to be the z-axis, i.e. $L = \boldsymbol{e}_0 \wedge (\boldsymbol{e}_0 + \boldsymbol{e}_3 + \frac{1}{2}\boldsymbol{e}_\infty) \wedge \boldsymbol{e}_\infty$. This means that the two manipulators should simultaneously keep their relative grasping positions on the plate and to keep the plate horizontal.

This formulation is given to a model predictive controller that uses second order system dynamics to compute desired accelerations. Using inverse dynamics we then compute torque commands for the control of the two manipulators. We chose a short horizon of 10 timesteps with $\Delta t = 0.01$, in order to achieve a very reactive behavior of the controller. The model predictive controller is then run at 100Hz and the inverse dynamics controller at 1000Hz.

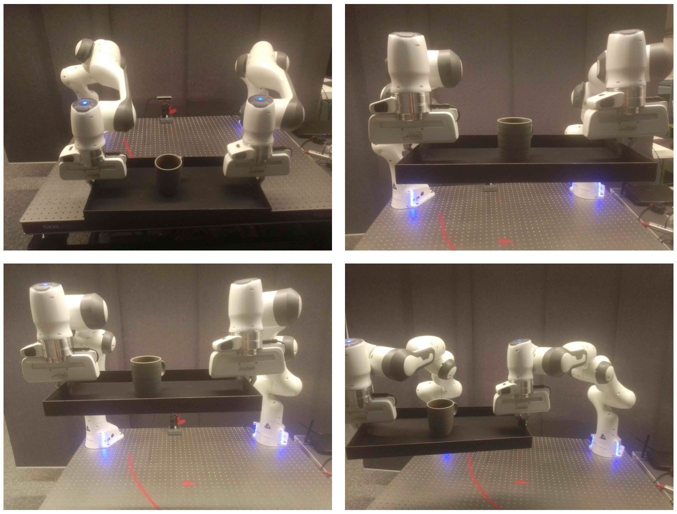


Fig. 6: Results of balancing a plate in different configurations.

Since there are only relative constraints in this task, the robots have a compliant behavior when one of them or the plate is moved by hand. The other one then adapts its configuration accordingly. We show several different configurations in Figure 6. If no external forces are applied, the robots stay in their current configuration. This experiment can also be seen in the accompanying video on our website.

V. Conclusion

In this article we presented an extension of the cooperative dual-task space in conformal geometric algebra, namely the CGA-CDTS. This extension keeps all the benefits of the original formulation that is based on dual quaternions, but adds more tools for geometric modeling of the dual-arm tasks. After reformulating the CDTS, we showed how the cooperative pointpair can be used to simultaneously represent both end-effector positions and how that can be exploited for cooperative reaching tasks and how the additional geometric primitives facilitate the modeling of dual-arm tasks. We then demonstrated the integration of the CGA-CDTS into an existing framework for optimal control using geometric algebra. In future work the ideas of the CGA-CDTS could be used to facilitate the modeling of the task spaces of robotic hands. For a 3-finger hand, for example, the cooperative pointpair could become a cooperative circle.

References

[1] S. Lee and S. Kim, "A self-reconfigurable dual-arm system," in *1991 IEEE International Conference on Robotics and Automation Proceedings*, Apr. 1991, 164–169 vol.1. DOI: 10.1109/ROBOT.1991.131573.

[2] R. Liza Ahmad Shauri and K. Nonami, "Assembly manipulation of small objects by dual-arm manipulator," *Assembly Automation*, vol. 31, no. 3, pp. 263–274, Jan. 1, 2011. DOI: 10.1108/01445151111150604.

[3] P. Ögren, C. Smith, Y. Karayiannidis, and D. Kragic, "A Multi Objective Control approach to Online Dual Arm Manipulation1," *IFAC Proceedings Volumes*, 10th IFAC Symposium on Robot Control, vol. 45, no. 22, pp. 747–752, Jan. 1, 2012. DOI: 10.3182/20120905-3-HR-2030.00032.

[4] D. Almeida and Y. Karayiannidis, "Cooperative Manipulation and Identification of a 2-DOF Articulated Object by a Dual-Arm Robot," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 5445–5451. DOI: 10.1109/ICRA.2018.8460511.

[5] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier, and A. Cherubini, "Dual-arm robotic manipulation of flexible cables," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid: IEEE, Oct. 2018, pp. 479–484. DOI: 10.1109/IROS.2018.8593780.

[6] C. Smith, Y. Karayiannidis, L. Nalpantidis, *et al.*, "Dual arm manipulation—A survey," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1340–1353, Oct. 2012. DOI: 10.1016/j.robot.2012.07.005.

[7] B. V. Adorno, P. Fraisse, and S. Druon, "Dual position control strategies using the cooperative dual task-space framework," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 3955–3960. DOI: 10.1109/IROS.2010.5650218.

[8] B. V. Adorno, "Robot Kinematic Modeling and Control Based on Dual Quaternion Algebra — Part I: Fundamentals.," p. 48,

[9] C. Gunn, "Geometric Algebras for Euclidean Geometry," *Adv. Appl. Clifford Algebras*, vol. 27, no. 1, pp. 185–208, Mar. 2017. DOI: 10.1007/s00006-016-0647-0.

[10] C. G. Gunn. "Projective geometric algebra: A new framework for doing euclidean geometry." (Aug. 18, 2020), [Online]. Available: http://arxiv.org/abs/1901.05873 (visited on 09/04/2023), preprint.

[11] T. Löw and S. Calinon, "Geometric Algebra for Optimal Control With Applications in Manipulation Tasks," *IEEE Transactions on Robotics*, pp. 1–15, 2023. DOI: 10.1109/TRO.2023.3277282.

[12] E. Sariyildiz and H. Temeltas, "A comparison study of three screw theory based kinematic solution methods for the industrial robot manipulators," in *2011 IEEE International Conference on Mechatronics and Automation*, Beijing, China: IEEE, Aug. 2011, pp. 52–57. DOI: 10.1109/ICMA.2011.5985630.

[13] M. d. P. A. Fonseca, B. V. Adorno, and P. Fraisse, "Coupled Task-Space Admittance Controller Using Dual Quaternion Logarithmic Mapping," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6057–6064, Oct. 2020. DOI: 10.1109/LRA.2020.3010458.

[14] R. Laha, L. F. Figueredo, J. Vrabel, A. Swikir, and S. Haddadin, "Reactive Cooperative Manipulation based on Set Primitives and Circular Fields," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 6577–6584. DOI: 10.1109/ICRA48506.2021.9561985.

[15] C. M. de Farias, Y. G. Rocha, L. F. C. Figueredo, and M. C. Bernardes, "Design of singularity-robust and task-priority primitive controllers for cooperative manipulation using dual quaternion representation," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, Aug. 2017, pp. 740–745. DOI: 10.1109/CCTA.2017.8062550.

[16] J. Vorndamme, J. Carvalho, R. Laha, *et al.*, "Integrated Bi-Manual Motion Generation and Control shaped for Probabilistic Movement Primitives," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, Nov. 2022, pp. 202–209. DOI: 10.1109/Humanoids53995.2022.10000149.

[17] E. Bayro-Corrochano and D. Kähler, "Motor algebra approach for computing the kinematics of robot manipulators," *Journal of Robotic Systems*, vol. 17, no. 9, pp. 495–516, 2000. DOI: 10.1002/1097-4563(200009)17:9<495::AID-ROB4>3.0.CO;2-S.

[18] C. Perwass, *Geometric Algebra with Applications in Engineering* (Geometry and Computing 4). Berlin: Springer, 2009, 385 pp.

[19] E. Bayro-Corrochano, *Geometric Algebra Applications Vol. II: Robot Modelling and Control*. Cham: Springer International Publishing, 2020. DOI: 10.1007/978-3-030-34978-3.

[20] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 4906–4913. DOI: 10.1109/IROS.2012.6386025.