Kendo Robot
(Robotics Final Project)

R11631045 Wei-Hsuan Cheng,
B10901042 Wen Perng,
R12921008 Che-Jung Chuang,
R12922135 Cheng-Yen Yu

National Taiwan University, Taipei, Taiwan

Abstract – Kendo, a well known martial art originated from Japan, requires training in both the sword skills and the way of the swords for its practitioners. To contribute to a successful attack, it requires the correct orientation and positioning of the sword upon striking one's enemy. In this project, we utilize the TM5M-700 robot arm for the grasping of the sword, to execute the appropriate attacks. A mathematical model - the conformal geometric algebra is used to calculate the inverse kinematics of the robot arm. A full interactive interface is also generated for enticing user experience.[a]

[a] Github link to all source code is here.

25 December 2023

1. INTRODUCTION

Kendo (剣道) is a martial art originated from Japan. Descended from the Japanese *kenjutsu*, it is a practice of both the sword skills and swordsmanship of its practitioners. As a modern sport, many different ways of scoring it has been incorporated. For a strike on an enemy to count as a successful attack, many criteria are considered, including: (1) whether the kendo sword (*shinai*) is of correct orientations, (2) whether or not you strike on the allowed attack positions, and (3) whether you, the attacker, exhibits the will of attack as a swordsman. The former two criteria requires sophisticated and accurate motion as well as real-time adaptation of the sword trajectory, hence, it is a perfect task for a robot to accomplish.

In this project, we utilized the TM5M-700 robot arm to accomplish the task of wielding a kendo sword. To perform the correct trajectories for the swords, we use the *conformal geometric algebra (CGA)* as a mathematical tool for analytic solutions to the inverse kinematics (IK) problem for the robot arm. Combined with ganja.js, an interactive GA arithmetic library, we are able to present the solutions to the IK problem in real time. We used a RealSense D435 RGBD camera to detect the enemy in its field of view. The LOGO-CAP algorithm is used to detect the body frame of the enemy, and the data is then feed into ganja.js to output the command for the robot arm. The robot arm is able

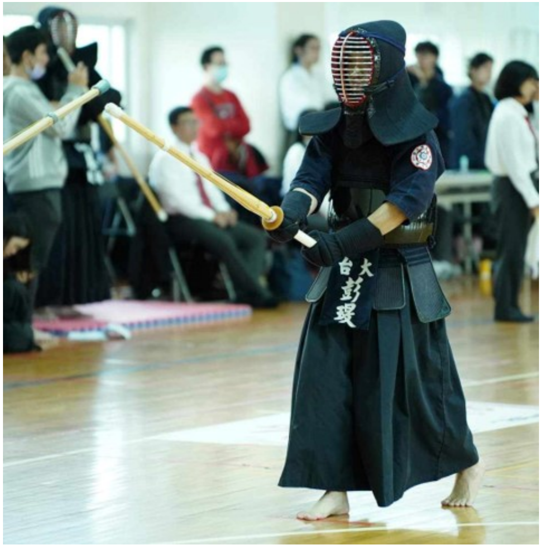


FIGURE 1: Mid Level Stance

to perform real-time tracking of the enemy position, and correctly hit the four allowed attack positions.

1.1. Notations

- $\boldsymbol{x}$: bold fonts for vectors.
- $|\boldsymbol{x}|$: vector 2-norm.
- e_i: lower-case serif font for orthonormal set of basis vectors.
- $\mathbb{R}$: the reals; $\mathcal{G}$: the geometric algebra.

2. PROJECT OBJECTIVES

2.1. Details on Kendo

Kendo as a sport is a one-versus-one game, attacking the other opponent with the sword. There are multiple rules that constitute a successful attack:

1. the sword should aim at the right attacking positions (elaborated later),
2. the sword should have the right orientation when striking the opponent,
3. the attackers should show their vigor and will of attack while also remaining in focus after an attempt of attack.

While the last one is more about the swordsmanship within Kendo, requiring years of training to understand the concrete meaning behind the vague guidelines, the former two are easily understandable and even replicable by a robot arm. We shall hence elaborate more on what the first two requirements are about.

First, the most prominant starting pose in kendo is known as the "*Chūdan* (中段)" or the "mid level stance" (see Figure 1). This pose ensures that the center line between you and your opponent is occupied by your sword: your enemy cannot get closer to you abruptly while many of the attack positions of your enemy are left wide open and vulnerable to your attack. Secondly. there are four allowed position for attack, namely: *men* (面, the head), *kote* (小手, the right wrist), *do* (胴, abdomen) and *tsuki* (突き, throat), as illustrated in Figure 2.

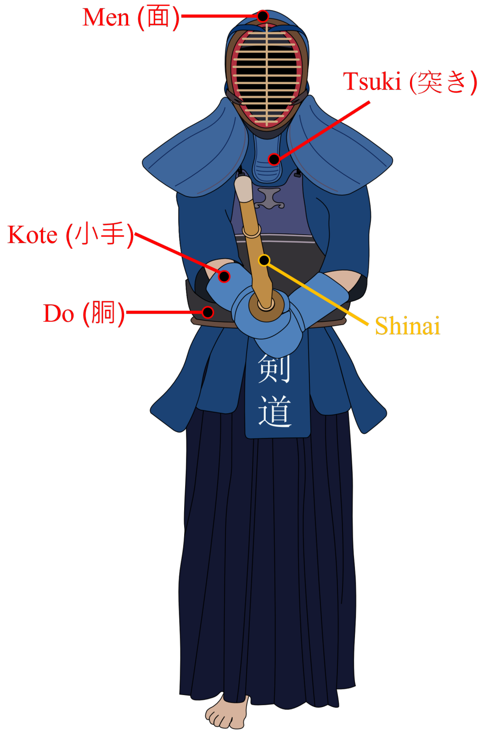


FIGURE 2: Attack Positions (source: Wikipedia Commons)

When striking *men* and *kote*, the sword moves in an arc upwards then down, with the blade of the sword slicing through the corresponding positions from top to down. When striking *do*, the blade should be going in a sideways arc-like motion, so as to slice through the abdomen of the opponent. Lastly, when attacking *tsuki*, it is a simple thrust at the throat of the opponent.

2.2. Objectives

Our project will be utilizing a RealSense D435 RGBD camera (see Figure 3) as the eyes of the whole system, detecting the position of an opponent. Next, command the TM5M-700 robot arm (see Figure 4) to perform tasks an operator wants. The robot arm shall be able to perform two modes:

- tracking mode: track the opponent's position and remain in the mid level pose while waiting for the operator's command;
- attacking mode: attack the four allowed positions under the command of the operator.

The followings are goals we plan to complete for the project, each point is listed in the sequence of

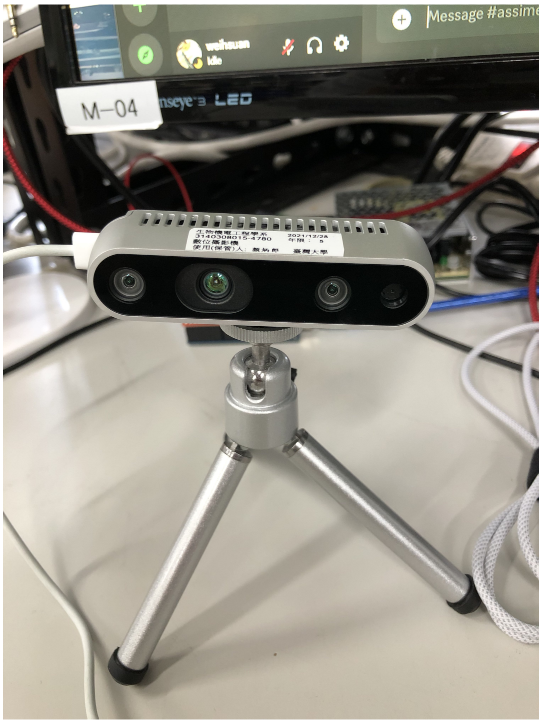


FIGURE 3: RealSense D435 Depth Camera

completion.

1. A suitable inverse kinematics model for the trajectory planning. We utilized a mathematical model called conformal geometric algebra for this task ,more in section 3.
2. Setup of the hardware and software environments, more in subsection 2.3.
3. Design of the end-effector for the TM robot arm to held the sword in place (see Figure 5).
4. Robot vision implemented for pose estimation of the opponent.
5. Integration of the software (pose estimation, inverse kinematics) and the hardware (the camera, the robot arm).
6. A user interface for the operator to gain easy access to information such as whether an attack position is available, which mode the robot is in, and where the opponent is at relative to the robot arm.

2.3. Project Structure

Considering the computation load of the LOGO-CAP algorithm, we decided to dedicate one laptop (laptop A) to human pose estimation, and assign the rest of the tasks to a separate laptop (laptop B) to achieve maximum performance. This leaves us with four machines to chain together: one camera, two laptops, and the PC that drives the robot. The camera is first connected to laptop A via USB, and the latter two connections are made using ethernet cables. The architecture of the whole system is shown in Figure 6.

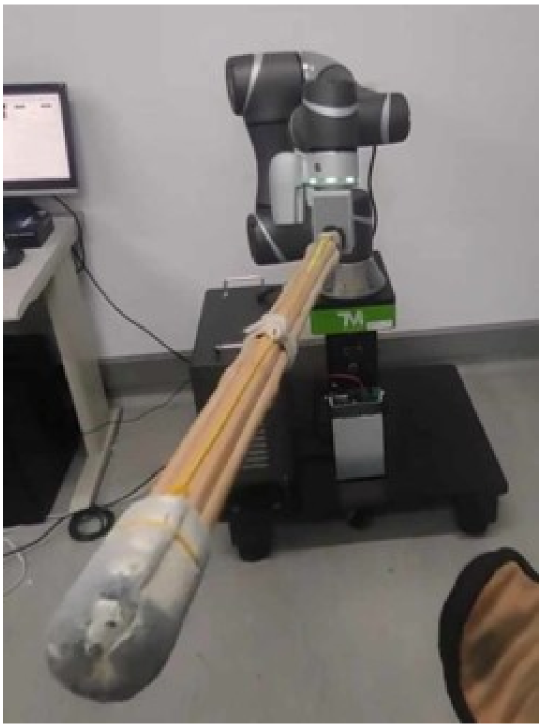


FIGURE 4: TM5M-700 Robot Arm with End-Effector Attached and Sword Equipped

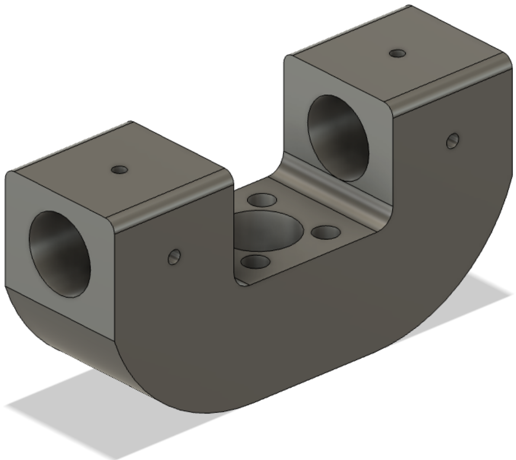


FIGURE 5: Sword Gripper (End-Effector) of the Robot Arm

The implementation of individual tasks span different programming languages and environments, including plain Python, Python running under ROS, and JavaScript running in the browser. This motivates the use of TCP/IP as a bridge across programs. The WebSocket protocol is used on top of TCP/IP when the browser is involved, since the browser does not support raw TCP/IP.

Out of the currently available software packages that implement GA, we chose `ganja.js` for its graphical, interactive nature, which is fitting considering that CGA is based on geometric primitives. Running in the

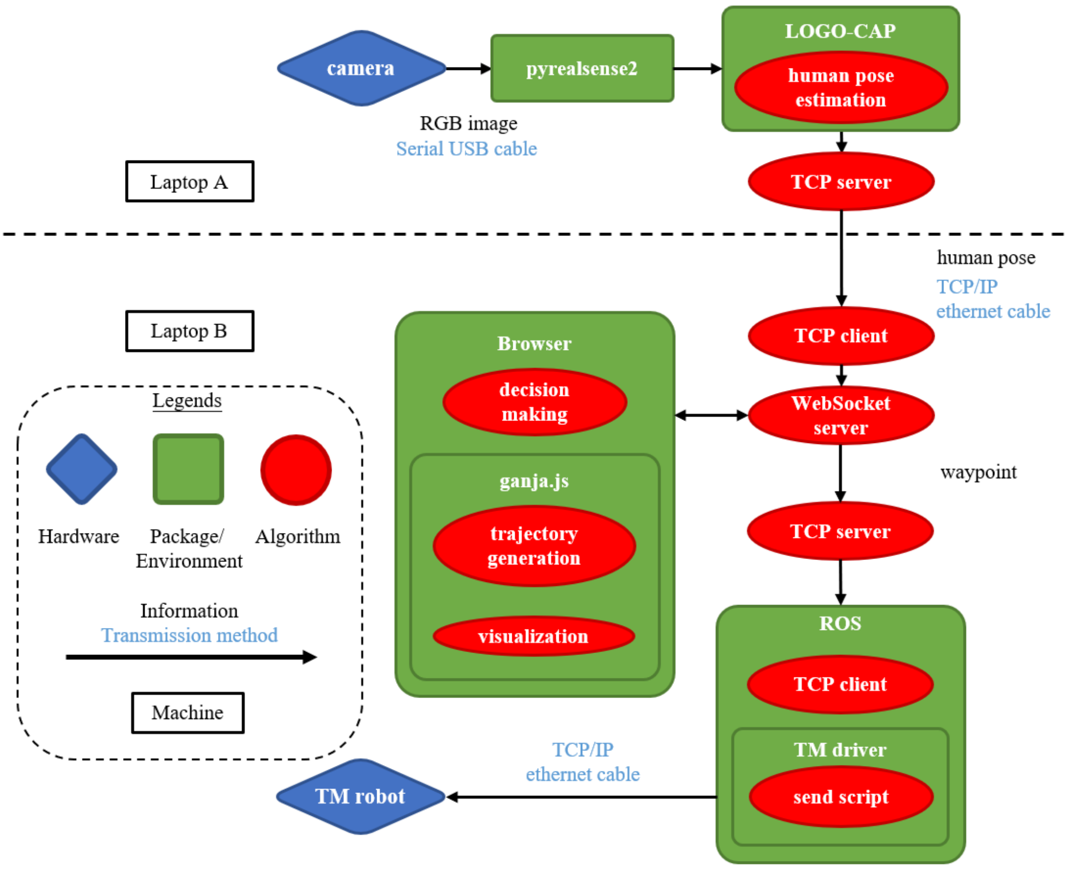


FIGURE 6: System Architecture

browser, it couples nicely with a live code editor as well as other native web UI components such as buttons and checkboxes. This enables rapid iteration of our CGA algorithm development.

Finally, since the provided interface to the robot arm is built with ROS, the output waypoints of our IK are encapsulated as messages before being sent as ROS service calls.

3. INVERSE KINEMATICS WITH CONFORMAL GEOMETRIC ALGEBRA

Traditional methods on solving inverse kinematics require solving systems of trigonometric equations, often requiring recursions to obtain numerical solutions to the *D-H parameters* controlling the robot.

Here we introduce a mathematical tool termed the *conformal geometric algebra (CGA)* that will be able to provide us with analytic solutions to the inverse kinematics problem of the robot (in our case, this is a 6-DoF robot arm). This mathematical description of IK also allows for intuitive relations for one to do trajectory planning. Detailed introduction to geometric algebra and conformal geometric algebra are provided in Appendix A.1 and Appendix A.2, respectively.

3.1. Geometric Ideology

The following ideas are from [KE16; Isi22; Ant04], where [KE16] proposed a closed-form solution of IK problem of a collaborative robot arm UR5, [Isi22] proposed and formulated closed-form solutions of a set of serial robots with a spherical wrist, and [Ant04] proposed a clear formula of how the 2 points can be extracted individually from a point pair (see A.38 for details). In this text, the IK solution is based on the combination of the ideas taken from the above three literature.

3.2. Analytic Solution to IK

3.2.1. Denavit–Hartenberg Parameters of TM5M-700

The inverse kinematics (IK) problem first starts with the Denavit–Hartenberg (D-H) parameters of our robot TM5M-700, provided in Table 1. Noted that the *modified* D-H parameters are used in this text. The solution to the IK problem is based on the use of the D-H parameters listed in Table 1.

It is interesting to point out that in CGA-based methods for IK problems, the origin position of each frame among the seven (6 axis + 1 base) frames are first solved, and then the joint variables are solved afterwards. This is the main difference between the

i	α_{i-1} [rad]	a_{i-1} [mm]	d_i [mm]	θ_i [rad]
1	0	0	145.2	θ_1
2	$-\frac{\pi}{2}$	0	0	θ_2
3	0	329.0	0	θ_3
4	0	311.5	-122.3	θ_4
5	$-\frac{\pi}{2}$	0	106	θ_5
6	$-\frac{\pi}{2}$	0	113.15	θ_6

TABLE 1: D-H table of TM5M-700

traditional geometric methods based on matrices and the CGA-based methods we are using in this text.

The following contexts are all based on the mathematical fundamentals of CGA. See details of GA and CGA in Appendix A. Also for clearness of the context, the details of the whole IK solution is not presented. See the original literature [KE16] for more detailed development of the IK solution.

3.2.2. *Target Pose, Null Points, and Robot Configurations*

The i-th frame's origin position $\boldsymbol{x}_i$, of the total seven frames $(\{0\}, \dots, \{6\})$, is represented by a *null point* X_i in CGA, which is a 5D representation of a point in $\mathbb{R}^3$. The mapping and inverse mapping between the 3D and 5D points are shown below (see details in eqn.(A.27) and eqn.(A.29)),

$$X_i = \mathcal{X}(\boldsymbol{x}_i) \in \mathbb{R}^{4,1}, \tag{1}$$

$$\boldsymbol{x}_i = \mathcal{X}^{-1}(X_i) \in \mathbb{R}^3. \tag{2}$$

Given the *target pose* of the end-effector frame (set as the 6th frame $\{6\}$ in this text) with respect to the base frame, and solving the null points $(X_0, \dots, X_6)$, the 8 configurations of the 6-DoF robot arm can be explicitly determined (if reachable), and then the joint variables can be further calculated using simple geometries.

A simple procedure of solving the null points are listed below,

$$\underbrace{\boldsymbol{x}_6, ({}^0_6x, {}^0_6y, {}^0_6z)}_{\text{given target pose}} \to \underbrace{X_5, X_0, X_1}_{\text{trivial solution}} \to \underbrace{X_4, X_3 \to X_2}_{\text{CGA-based method}} .$$

Now we define the *configuration parameters* [KE16] of the robot, which will be used in the following derivations,

$$k_{ud} = \pm 1 \text{ (elbow up: 1, elbow down: -1)}, \tag{3}$$

$$k_{lr} = \pm 1 \text{ (should right: 1, should left: -1)}, \tag{4}$$

$$k_{fn} = \pm 1 \text{ (wrist not flipped: 1, wrist flipped: -1)}. \tag{5}$$

3.2.3. *IK problem- Solving the Null Points X_6,X_5,X_0, and X_1*

Given the target pose of frame $\{6\}$, where the target position is $\boldsymbol{x}_6$, and the target orientation is $({}^0_6x, {}^0_6y, {}^0_6z)$, the null points X_6 and X_5 are then easily determined,

$$X_6 = \mathcal{X}(\boldsymbol{x}_6), \tag{6}$$

$$X_5 = \mathcal{X}(\boldsymbol{x}_6 - d_6\, {}^0_6z). \tag{7}$$

The null points representing the origins of base frame $\{0\}$ and frame $\{1\}$ are trivial,

$$X_0 = \mathcal{X}(\boldsymbol{x}_0 = 0), \tag{8}$$

$$X_1 = \mathcal{X}(\boldsymbol{x}_0 + d_1 \mathsf{e}_3). \tag{9}$$

3.2.4. *IK problem- Solving the Null Points X_4 and X_3*

The null points X_4 and X_3 are solved based on strong geometric insights provided by CGA.

Intersecting 2 spheres, S_c and K_0, gives 1 circle C_{5k}, where

$$S_c = \left(X_5 - \frac{1}{2}d_4^2 \mathsf{e}_\infty\right)^\star \quad \text{(grade-4 sphere)}, \tag{10}$$

$$K_0 = \left(\mathsf{e}_0 - (S_c^\star \cdot \mathsf{e}_0)\mathsf{e}_\infty\right)^\star \quad \text{(grade-4 sphere)}, \tag{11}$$

$$C_{5k} = S_c \vee K_0 \quad \text{(grade-3 circle)}. \tag{12}$$

Intersect the circle C_{5k} and a horizontal plane passing through X_5 gives a point pair Q_c, where

$$Q_c = -(X_5 \wedge \mathsf{e}_1 \wedge \mathsf{e}_2 \wedge \mathsf{e}_\infty) \vee C_{5k} \quad \text{(grade-2 point pair)}. \tag{13}$$

Extracting each point in a point pair using the formula provided in eqn. (A.38) and [Ant04],

$$X_c = \left(1 + k_{lr}\frac{Q_c}{\sqrt{Q_c^2}}\right)(Q_c \cdot \mathsf{e}_\infty) \quad \text{(grade-1 null point)}. \tag{14}$$

This yields the 2 solutions of the null points X_c based on the *shoulder* configuration, where X_c is an auxiliary null point used for determining X_4.

Now form a vertical plane Π_c passing through X_c, where

$$\Pi_c = \mathsf{e}_0 \wedge \mathsf{e}_3 \wedge X_c \wedge \mathsf{e}_\infty \quad \text{(grade-4 plane)}. \tag{15}$$

Noted that X_c and Π_c are auxiliary null point and plane that can help solve the rest of the null points, since that X_0, X_1, X_2, and X_3 all lie in Π_c due to the geometric nature of the robot links,

$$X_0, X_1, X_2, X_3 \in \Pi_c. \tag{16}$$

Form another vertical plane $\Pi_{5\|}$ parallel to Π_c, which is shifted from Π_c to pass through both X_4 and X_5, where

$$\Pi_{5\|} = \left(\Pi_c^\star + (X_5 \cdot \Pi_c^\star)\mathsf{e}_\infty\right)^\star \quad \text{(grade-4 plane)}. \tag{17}$$

Form an auxiliary plane Π_{56} perpendicular to $\Pi_{5\|}$ and get the corresponding normal vector $\hat{n}_{56}$, where

$$\Pi_{56} = (X_5 \wedge X_6)^\star \wedge \mathsf{e}_\infty \quad \text{(grade-4 plane)}, \tag{18}$$

$$\hat{n}_{56} = -(\Pi_{56} \cdot \mathsf{e}_0) \cdot \mathsf{e}_\infty \quad \text{(grade-1 vector)}. \tag{19}$$

The plane Π_{56} and its normal vector $\hat{n}_{56}$ are used for calculating the plane $\Pi_{5\perp}$, which is not only perpendicular to $\Pi_{5\|}$ but also contains both X_4 and X_5, where

$$\Pi_{5\perp} = X_5 \wedge \hat{n}_{56} \wedge \mathsf{e}_\infty \qquad \text{(grade-4 plane).} \tag{20}$$

After defining a bundle of auxiliary planes, intersect the planes $\Pi_{5\|}$ and $\Pi_{5\perp}$, gives a line L_{54} passing through X_4 and X_5 (since the 2 planes both contain X_4 and X_5), where

$$L_{54} = \Pi_{5\|} \vee \Pi_{5\perp} \qquad \text{(grade-3 line).} \tag{21}$$

Define a sphere S_5 centred at X_5 with a radius of d_5,

$$S_5 = \left(X_5 - \frac{1}{2}d_5^2\mathsf{e}_\infty\right)^\star \qquad \text{(grade-4 sphere).} \tag{22}$$

Intersecting the sphere S_5 and the line L_{54} gives a point pair that contains X_4, where

$$Q_4 = L_{54}^\star \cdot S_5 \qquad \text{(grade-2 point pair).} \tag{23}$$

Now extract X_4 from the point pair Q_4, as we did in eqn. (14), where

$$X_4 = \left(1 + k_{fn}\frac{Q_4}{\sqrt{Q_4^2}}\right)(Q_4 \cdot \mathsf{e}_\infty) \quad \text{(grade-1 null point).} \tag{24}$$

This shows that X_4 also has 2 solutions based on the *wrist* configuration.

Then, X_3 is solved in a similar way, that is, intersecting a sphere centred at X_4 with a radius of d_4, and a line L_{34} passing through X_3 and X_4, gives a point pair Q_3 that contains the null point X_3, where

$$S_4 = \left(X_4 - \frac{1}{2}d_4^2\mathsf{e}_\infty\right)^\star \qquad \text{(grade-4 sphere),} \tag{25}$$

$$L_{34} = X_4 \wedge \Pi_c^\star \wedge \mathsf{e}_\infty \qquad \text{(grade-3 line),} \tag{26}$$

$$Q_3 = L_{34}^\star \cdot S_4 \qquad \text{(grade-2 point pair).} \tag{27}$$

Recall that since Π_c contains X_3, as mentioned in eqn. (16), it's not surprising that it can be used for solving L_{34} and X_3.

Now extract X_3 from the point pair Q_3, as we did in eqn. (14) and (24), where

$$X_3 = \left(1 + k_{lr}\frac{Q_3}{\sqrt{Q_3^2}}\right)(Q_3 \cdot \mathsf{e}_\infty) \quad \text{(grade-1 null point).} \tag{28}$$

This shows that X_3 also has 2 solutions based on the *shoulder* configuration.

3.2.5. *IK problem- Solving the Null Points X_2*

Since that X_3 and X_1 are already solved, X_2 can be easily determined. First, we intersect 2 spheres, one centred at X_3 with a radius of a_3, another centred at X_1 with a radius of a_2, to obtain a circle C_2, where

$$S_3 = \left(X_3 - \frac{1}{2}a_3^2\mathsf{e}_\infty\right)^\star \qquad \text{(grade-4 sphere),} \tag{29}$$

$$S_1 = \left(X_1 - \frac{1}{2}a_2^2\mathsf{e}_\infty\right)^\star \qquad \text{(grade-4 sphere),} \tag{30}$$

$$C_2 = S_1 \vee S_3 \qquad \text{(grade-3 circle).} \tag{31}$$

Recall again that since Π_c also contains X_2, as mentioned in eqn. (16), the former can help solve the latter. Now, intersect the circle C_2 and the plane Π_c to get the point pair Q_2 that contains X_2, where,

$$Q_2 = -\Pi_c \vee C_2 \qquad \text{(grade-2 point pair).} \tag{32}$$

Now extract X_2 from the point pair Q_2, as we did in eqn. (14), (24) and (28), where

$$X_2 = \left(1 + k_{ud}\frac{Q_2}{\sqrt{Q_2^2}}\right)(Q_2 \cdot \mathsf{e}_\infty) \quad \text{(grade-1 null point).} \tag{33}$$

This shows that X_2 also has 2 solutions based on the *elbow* configuration.

Browsing through the solutions of the 7 null points $(X_0, \dots, X_6)$, it can be easily seen that, depending on the signs of the configuration parameters k_{ud}, k_{lr}, and k_{fn}, the 8 robot configurations of a single target pose can be determined explicitly (if reachable); therefore, the CGA-based IK solution in this text can be considered a closed-form method.

3.2.6. *IK problem- Solving the Joint Variables θ_i*

Since all the null points as well as the geometric relations among all frames are determined, the joint variables can be easily solved based on simple geometries such as the calculating each angle between 2 consecutive lines. The calculation is not presented in this text due to page limit. For more details, see the original literature [KE16].

For the geometric pictures of each entities, also refer to the ganja.js example we wrote and the github here.

3.3. Trajectory Planning

Trajectory planning requires the change of the end-effector frame $\{6\}$ as time elapses. In the following, we shall set the "elapsed time" $t \in [0, 1]$. Note that t should be rescaled later to fit real moving speed of the sword.

The motion of the sword when attacking *tsuki* is the simplest, requiring a plain translational motion from the starting position to the end position. To obtain the two frames, we can first consider the orientation of the sword tip. Let the sword length be l_s, the position of the sword tip be $\boldsymbol{x}_{\mathrm{st}}$, the inclination of the sword with respect to the ground be θ_s, and the plane containing the naval-to-naval line $\hat{\boldsymbol{v}}_{\text{N-N}}$ and the universal z-axis be E. Note that the naval-to-naval line is equivalent to the translational vector from initial position to final position projected onto the horizontal plane:

$$\hat{\boldsymbol{v}}_{\text{N-N}} = -\left((\boldsymbol{x}_6^{\text{f}} - \boldsymbol{x}_6^{\text{i}}) \cdot (\mathsf{e}_1 \wedge \mathsf{e}_2)\right)(\mathsf{e}_1 \wedge \mathsf{e}_3). \tag{34}$$

Then the end-effector frame $\{6\}$ will have initial position

$$\boldsymbol{x}_6 = \boldsymbol{x}_{\text{st}} - l_s \cdot \mathrm{e}^{-E\theta_s/2}\hat{\boldsymbol{v}}_{\text{N-N}}\mathrm{e}^{E\theta_s/2}, \tag{35}$$

note that $\boldsymbol{x}_6$ is a function of $\boldsymbol{x}_{\text{st}}$ and θ_s. Given the initial and final sword tip position and inclination, we can obtain the respective end-effector frame $\boldsymbol{x}_6^{\text{i}}$ and $\boldsymbol{x}_6^{\text{f}}$. Since the initial $\boldsymbol{x}_{\text{st}}$ is targeted at the throat (the mid level stance), *tsuki* only requires a translational motion by the translator acting on $\boldsymbol{x}_6$:

$$T_{\text{trans}} = \mathrm{e}^{-(\boldsymbol{x}_6^{\text{f}} - \boldsymbol{x}_6^{\text{i}})\cdot t/2}, \tag{36}$$

and no further rotation is needed.

The motion of the sword when attacking *men* and *kote* are similar, both are a rotational motion with the center of rotation undergoing a translational motion. Suppose the sword swings upwards to a max inclination of $\theta_{s,\text{Max}} \approx 70°$ with center of rotation being $\boldsymbol{x}_6$. Then the motion of the end-effector frame can be described by the motor

$$M = T_{\text{trans}}\mathrm{e}^{-\theta_s E/2}. \tag{37}$$

θ_s is a function of time, going from the initial angle of θ_s^{i} to $\theta_{s,\text{Max}}$ in $t \in [0, 0.5]$ by an interpolating cubic spline (with boundary condition of $\dot{\theta}_s = 0$); going from $\theta_{s,\text{Max}}$ to $\theta_s^{\text{f}} \approx 20°$ in $t \in [0, 0.5]$.

The motion of the sword when attacking *do* is the most complicated. Since the blade of the sword needs to face sideways instead of downwards like all the previous ones, an additional rotation in the plane perpendicular to the naval-to-naval line is required. We can approximate the motion by including a sideways rotation:

$$M = T_{\text{trans}}\mathrm{e}^{-\theta_s E/2 - \theta_d \hat{\boldsymbol{v}}_{\text{N-N}} I_3^{-1}/2}. \tag{38}$$

Where θ_d is again a function of t, we set is as a parabola with $\theta_d = 0$ at $t = 0$ and $\theta_d > 0$ at $t = 1$. See the following ganja.js code for a better illustration.

At last, to control the TM robot arm, several points are sampled from the given trajectory and sent as commands for orientation of the robot.

4. HUMAN POSE ESTIMATION

To estimate human pose in three-dimensional space, we employ LOGO-CAP, a machine-learning based human pose estimation approach. LOGO-CAP can predict 17 two-dimensional keypoints for each pose. Notice that we use an RGBD camera, RealSense, as the main visual input. Combine estimated two-dimensional keypoints and depth map, we can infer human pose in three-dimensional space. To reach this, camera calibration and hand-eye calibration are done to acquire the transformation from camera coordinates to base coordinates.

5. RESULTS AND DEMONSTRATION

5.1. User Interface

The UI is shown in Figure 7. On the left, we have a 3D visualization of our system rendered using ganja.js. The objects drawn, such as points, lines, and planes, as well as our CGA algorithm, are defined in JavaScript written in the editor on the right. Controls are on the bottom left, which handles networking and controling the Kendo move of the robot.

In the visualization window, We update the human body frame in real time as new human pose data arrives. When we issue a move command, it displays an animation corresponding to the output sword trajectory, as shown in Figure 8. The window is interactive, and we can use the mouse to change the viewing angle and zoom level to better inspect the system.

Moreover, since words couldn't have possibly illustrate the vigor and enthusiasm of the robot, we have provided a video showcasing of our work on YouTube. It goes over the features and demonstrations in full detail.

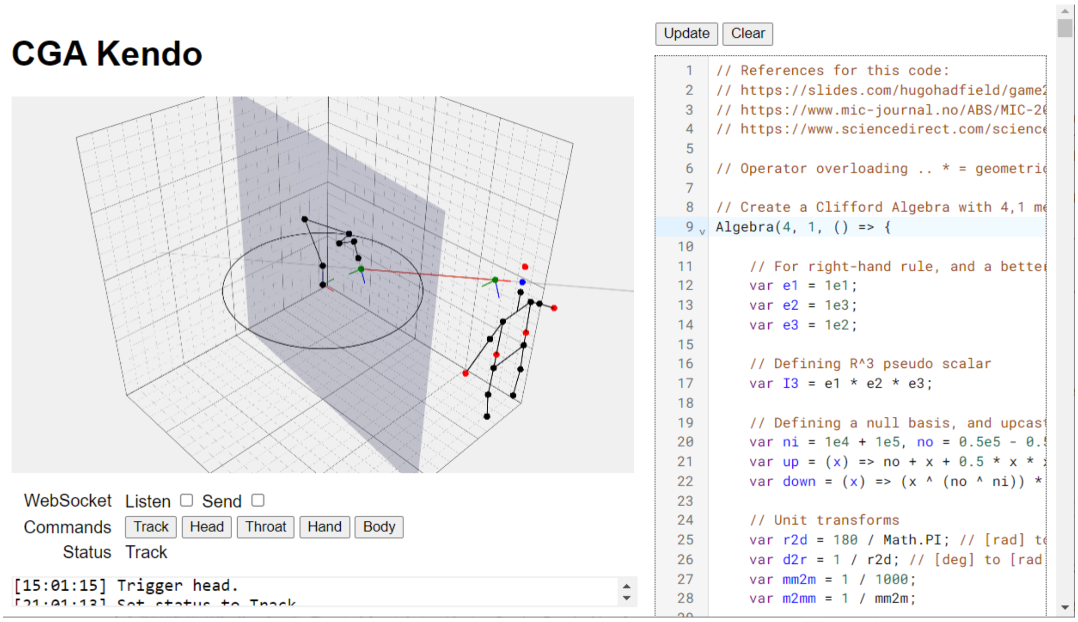
5.2. Division of Work

Wei-Hsuan Cheng	CGA theoretical development, Robot motion planning.
Wen Perng	CGA theoretical development, Kendo skill technical guidance.
Che-Jung Chuang	Robot motion planning, System integration.
Cheng-Yen Yu	Robot vision, System Integration.

6. CONCLUSIONS

By utilizing ganja.js we were able to calculate inverse kinematics and trajectory planning for the TM robot arm using the novel mathematical framework of conformal geometric algebra. Incorporating the pose estimation using the LOGO-CAP algorithm and RealSense camera, skeleton of the enemy can be displayed in an UI, allowing easy game-like control of the robot arm as if competing in a round of Kendo.

The whole system is linked up across platforms and seeing the real-time adaptation of the robot arm to opponent position is exciting. The whole experience, whether being the controller of the robot or the opponent facing the robot, is interesting and worth a try, allowing one to understand and try out the joy of Kendo.



WebSocket Listen □ Send □

Status Track

```
[15:01:15] Trigger head.
[21:01:13] Set status to Track
```

```
// References for this code:
// https://slides.com/hugohadfield/game2
// https://www.mic-journal.no/ABS/MIC-20
// https://www.sciencedirect.com/science

// Operator overloading .. * = geometric

// Create a Clifford Algebra with 4,1 me
Algebra(4, 1, () => {

    // For right-hand rule, and a better
    var e1 = 1e1;
    var e2 = 1e3;
    var e3 = 1e2;

    // Defining R^3 pseudo scalar
    var I3 = e1 * e2 * e3;

    // Defining a null basis, and upcast
    var ni = 1e4 + 1e5, no = 0.5e5 - 0.5
    var up = (x) => no + x + 0.5 * x * x
    var down = (x) => (x ^ (no ^ ni)) *

    // Unit transforms
    var r2d = 180 / Math.PI; // [rad] to
    var d2r = 1 / r2d; // [deg] to [rad
    var mm2m = 1 / 1000;
    var m2mm = 1 / mm2m;
```

FIGURE 7: Browser-based UI hosting our CGA algorithm (right), visualization (left top), and controls (left bottom).

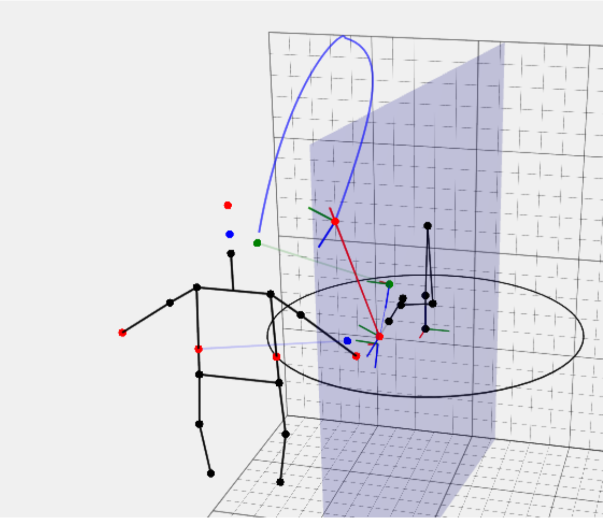


FIGURE 8: Animation of a sword trajectory targeting the body.

REFERENCES

[Ant04] Richard Wareham Anthony Lasenby Joan Lasenby. *A covariant approach to geometry using geometric algebra*. Department of Engineering - University of Cambridge, 2004.

[Per09] Christian Perwass. *Geometric Algebra with Applications in Engineering*. Springer Series in Geometry and Computing. Springer, 2009. ISBN: 9783540890676.

[KE16] A. Kleppe and O. Egeland. *Inverse kinematics for industrial robots using conformal geometric algebra*. Modeling, Identification and Control. 2016.

[Isi22] Joan Lasenby Isiah Zaplana Hugo Hadfield. *Closed-form solutions for the inverse kinematics of serial robots using conformal geometric algebra*. Mechanism and Machine Theory vol. 173. 2022.

APPENDIX A. MATHEMATICAL DETAIL

Appendix A.1. Geometric Algebra

Geometric algebra (GA) is equivalent to the *Clifford algebra* proposed by the English mathematician William Clifford in the 1800s. The former especially emphasized on the geometric interpretations provided by the algebraic structures.

An intuitive and mathematically non-rigorous introduction to GA is aimed to be given here, providing the readers with the basics to understand how to calculate the equations mentioned in the text. For a further understanding of the whole topic, please refer to [Per09].

Appendix A.1.1. Grade-1 Basis

Consider a *graded algebra* $\mathcal{G}_{p,q}$ termed the *geometric algebra* with signature (p, q), it is a vector space with its elements graded from grade-0 to grade-$(p+q)$ over the field $\mathbb{R}$. Its grade-0 subspace is the reals $\mathbb{R}$, also coined the *scalars*. Its grade-1 subspace is the vector space of $\mathbb{R}^{p,q}$, the pseudo-Euclidean space of signature (p, q), also called the *vectors*. The vector space $\mathbb{R}^{p,q}$ is spanned by an orthonormal basis with a total of $p+q$ basis vectors $\{\mathsf{e}_i\}_{i=1}^{p+q}$ defined by their inner product relation[1]:

$$\mathsf{e}_i \cdot \mathsf{e}_j = g_{ij} = \begin{cases} 1 & , i = j = 1, \cdots, p; \\ -1 & , i = j = p+1, \cdots, p+q; \\ 0 & , i \neq j. \end{cases} \tag{A.1}$$

This is the *canonical basis vectors* of $\mathbb{R}^{p,q}$. Let us construct the whole graded algebra $\mathcal{G}_{p,q}$ by the its grade-1 components.

Appendix A.1.2. Outer Product

A general element in $\mathcal{G}_{p,q}$ is termed a multivector. Consider a new operation termed the *outer product*. Denoted by $\wedge$, it is an associative bilinear product that distributes over addition of multivectors:

$$\wedge : \mathcal{G}_{p,q} \times \mathcal{G}_{p,q} \to \mathcal{G}_{p,q} \tag{A.2}$$
$$(M_1, M_2) \mapsto M_1 \wedge M_2. \tag{A.3}$$

Moreover, we can consider the graded-r subspace of $\mathcal{G}_{p,q}$ as $\mathcal{G}^r_{p,q}$, then the outer product satisfies the additivity of grade:

$$\wedge : \mathcal{G}^{r_1}_{p,q} \times \mathcal{G}^{r_2}_{p,q} \to \mathcal{G}^{r_1+r_2}_{p,q}. \tag{A.4}$$

Furthermore, to actually be able to calculate the outer product, we define their action on the grade-1 basis as *anti-symmetric*, satisfying

$$\mathsf{e}_i \wedge \mathsf{e}_j = -\mathsf{e}_j \wedge \mathsf{e}_i \quad (\forall i, j = 1, \cdots, p+q). \tag{A.5}$$

The calculation of outer products between two multivectors can be easily extended via associativity and bilinearity. And the outer product with the grade-0 elements, is simply the scalar scaling.

An example is as below: consider in $\mathcal{G}_{3,0}$,

$$\begin{aligned}(1+\mathsf{e}_1) \wedge (\mathsf{e}_2 + \mathsf{e}_1 \wedge \mathsf{e}_3) =& 1 \wedge \mathsf{e}_2 + \mathsf{e}_1 \wedge \mathsf{e}_2 \\ &+ 1 \wedge \mathsf{e}_1 \wedge \mathsf{e}_3 + \mathsf{e}_1 \wedge \mathsf{e}_1 \wedge \mathsf{e}_3 \\ =& \mathsf{e}_2 + \mathsf{e}_1 \wedge \mathsf{e}_2 + \mathsf{e}_1 \wedge \mathsf{e}_3 + 0.\end{aligned}$$

To reduce crowding of the subscripts, we conduct the convention below: given subscripts $\mathcal{I}$ and $\mathcal{J}$, then

$$\mathsf{e}_{\mathcal{I}} \wedge \mathsf{e}_{\mathcal{J}} =: \mathsf{e}_{\mathcal{I}\mathcal{J}}. \tag{A.6}$$

A multivector has mixed-grade components in it. To extract components with a given grade r, we shall define the *grade selection operator* $\langle \cdot \rangle_r$ that is linear and satisfies: for all multivector $M \in \mathcal{G}_{p,q}$,

$$\langle M \rangle_r \in \mathcal{G}^r_{p,q}, \tag{A.7}$$

and for all $M_r \in \mathcal{G}^r_{p,q}$, $\langle M_r \rangle_r = M_r$. Moreover, we denote the grade-0 selection operator as $\langle \cdot \rangle_0 \equiv \langle \cdot \rangle$.

Appendix A.1.3. Geometric Product

Between vectors, we have already defined the inner product; for multivectors, we have defined the outer product. It would be really convenient for us to extend the definition of the inner product to multivectors, or even better, combine the two products to obtain a more omnipotent product. Here we shall define the *geometric product*: for A, B, $C \in \mathcal{G}_{p,q}$ and $\boldsymbol{a} \in \mathcal{G}^1_{p,q} = \mathbb{R}^{p,q}$, the geometric product (denoted by concatenation) satisfies

1. closure: $AB \in \mathcal{G}_{p,q}$;
2. associativity: $A(BC) = (AB)C$;
3. distributivity: $A(B+C) = AB + AC$ and $(A+B)C = AC + BC$;
4. $\exists\, 1 \in \mathcal{G}_{p,q}$ such that $1A = A$;
5. $\boldsymbol{a}^2 := \boldsymbol{a} \cdot \boldsymbol{a}$.

Consider the following product:

$$\underbrace{(\boldsymbol{a}+\boldsymbol{b})^2}_{\in\mathbb{R}} = \underbrace{(\boldsymbol{a}^2 + \boldsymbol{b}^2)}_{\in\mathbb{R}} + (\boldsymbol{ab} + \boldsymbol{ba}),$$

hence we know that $\boldsymbol{ab} + \boldsymbol{ba} \in \mathbb{R}$. By simple geometric arguments, We can know that this symmetric product is in fact the inner product:

$$\boldsymbol{a} \cdot \boldsymbol{b} = \frac{1}{2}(\boldsymbol{ab} + \boldsymbol{ba}). \tag{A.8}$$

We then *define* the anti-symmetric sum as the outer product:

$$\boldsymbol{a} \wedge \boldsymbol{b} := \frac{1}{2}(\boldsymbol{ab} - \boldsymbol{ba}). \tag{A.9}$$

In fact, for multivectors $A_r = \langle A_r \rangle_r$ and $\langle B_s \rangle_s$, we can define their inner and outer product as:

$$A_r \cdot B_s := \langle A_r B_s \rangle_{|r-s|}, \tag{A.10}$$
$$A_r \wedge B_s := \langle A_r B_s \rangle_{r+s}. \tag{A.11}$$

[1] The inner product mentioned here is not necessarily non-negative. For two vectors $\boldsymbol{u} = u^i \mathsf{e}_i$ and $\boldsymbol{v} = u^j \mathsf{e}_j$ (Einstein summation convention used) in $\mathbb{R}^{p,q}$, their inner product is defined as $\boldsymbol{u} \cdot \boldsymbol{v} = u^i v^j g_{ij}$, where the metric is $g_{ij} = \mathrm{diag}(\underbrace{+1}_{\times p}, \underbrace{-1}_{\times q})$.

Grade	Name	Basis	
0	Scalar	1	
1	Vectors	e_i	$i \in \{1 \sim p+q\}$
2	Bivectors	e_{ij}	$i \neq j \in \{1 \sim p+q\}$
3	Trivectors	e_{ijk}	$i \neq j \neq k \in \{1 \sim p+q\}$
k	k-vectors	$\mathsf{e}_{\mathcal{I}}$	Index set: $\lvert\mathcal{I}\rvert = k$
$p+q$	Pseudoscalar	$I := \mathsf{e}_{12\cdots(p+q)}$	

TABLE A.1: Canonical Basis of $\mathbb{G}_{p,q}$

The results can be extended to all multivectors by defining the inner and outer product as bilinear operations.

Appendix A.1.4. Geometric Algebra

Consider the geometric product between the canonical basis vectors, they satisfy

$$\mathsf{e}_i\mathsf{e}_j = \begin{cases} \mathsf{e}_i \cdot \mathsf{e}_j = \mathsf{e}_j\mathsf{e}_i & , i = j; \\ \mathsf{e}_i \wedge \mathsf{e}_j = -\mathsf{e}_j\mathsf{e}_i & , i \neq j. \end{cases} \tag{A.12}$$

These relations between the basis vectors allow us to actually calculate the inner, outer and geometric product between any multivectors. Also, we can construct a set of *canonical basis* for the whole $\mathcal{G}_{p,q}$ as in Table A.1. As can easily be seen, the grade-k subspace has $\binom{p+q}{k}$ bases. The basis of the highest grade is often coined the *pseudoscalar*, denoted by I. For example, in $\mathcal{G}_3$, $I = \mathsf{e}_{123} =: I_3$.

Appendix A.1.5. Duality

The grade-k and grade-$(p+q-k)$ subspace of $\mathcal{G}_{p,q}$ has the same dimensions, and naturally, a duality between the two subspaces exists. We define the duality as

$$\star : \mathcal{G}^r_{p,q} \to \mathcal{G}^{p+q-r}_{p,q} \tag{A.13}$$

$$M \to M^\star := M/I. \tag{A.14}$$

Where the division by the pseudoscalar is defined as multiplication by the inverse of I, which is

$$I^{-1} = I^3 = (-1)^{(p+q-1)(p+q)/2+q} I. \tag{A.15}$$

The result can be easily extended to all multivectors by defining the duality as a linear operator. The inverse operator, undual, is defined by: $(M^\star)^{-\star} = M$ for all multivectors M.

Appendix A.1.6. Important Formulas

Before ending this subsection, a few important and useful formulas are introduced without proof. Consider $A_k = \boldsymbol{a}_1 \wedge \boldsymbol{a}_2 \wedge \cdots \wedge \boldsymbol{a}_k$ and $B_l = \boldsymbol{b}_1 \wedge \boldsymbol{b}_2 \wedge \cdots \wedge \boldsymbol{b}_l$, denote

$$[A_k/\boldsymbol{a}_i] = \boldsymbol{a}_1 \wedge \cdots \wedge \boldsymbol{a}_{i-1} \wedge \boldsymbol{a}_{i+1} \wedge \cdots \wedge \boldsymbol{a}_k. \tag{A.16}$$

For $k \leq l$, we have

$$A_k \cdot B_l = [A_k/\boldsymbol{a}_k] \cdot (\boldsymbol{a}_k \cdot B_l). \tag{A.17}$$

Let $\boldsymbol{a}$ be a vector, then

$$\boldsymbol{a} \cdot A_k = \sum_{i=1}^{k} (-1)^{i+1} (\boldsymbol{a} \cdot \boldsymbol{a}_i)[A_k/\boldsymbol{a}_i]. \tag{A.18}$$

Lastly, exponentials of a multivector M can be defined by the Taylor series expansion:

$$\mathrm{e}^M := \sum_{n=0}^{\infty} \frac{M^n}{n!}. \tag{A.19}$$

Appendix A.2. Conformal Geometric Algebra

The conformal geometric algebra (CGA) is a *flavour* of the geometric algebra, able to represent and work with geometric objects such as points, lines, circles, planes and spheres. Namely, it works with $\mathcal{G}_{4,1}$ to represent the geometric objects mentioned and their *meets* and *joins* as well as them under conformal transformations such as mirroring, rotation, translation and sphere inversion. Due to the algebra having spheres, lines and rigid body motion, it is useful in solving inverse kinematics problem in robotics.

The grade-1 canonical basis vectors of $\mathcal{G}_{4,1}$ is $\{\mathsf{e}_1, \mathsf{e}_2, \mathsf{e}_3, \mathsf{e}_+, \mathsf{e}_-\}$, where: $\mathsf{e}_1^2 = \mathsf{e}_2^2 = \mathsf{e}_3^2 = \mathsf{e}_+^2 = 1$, $\mathsf{e}_-^2 = -1$ and $\mathsf{e}_i \cdot \mathsf{e}_j = 0$ for $i \neq j$. The pseudoscalar of the algebra is defined as $I := \mathsf{e}_{123+-}$, $I^{-1} = -I$; the pseudoscalar of the $\mathbb{R}^3$ subspace is $I_3 := \mathsf{e}_{123}$, $I_3^{-1} = -I_3$.

We further define two vectors (not the basis) to make the notations of the following writing much clearer:

$$\mathsf{e}_\infty := \mathsf{e}_- + \mathsf{e}_+, \tag{A.20}$$

$$\mathsf{e}_0 := \frac{1}{2}(\mathsf{e}_- - \mathsf{e}_+). \tag{A.21}$$

They satisfy:

$$\mathsf{e}_\infty^2 = \mathsf{e}_\infty \cdot \mathsf{e}_\infty = 0 = \mathsf{e}_0 \cdot \mathsf{e}_0 = \mathsf{e}_0^2, \tag{A.22}$$

$$\mathsf{e}_\infty \cdot \mathsf{e}_0 = -1, \tag{A.23}$$

$$\mathsf{e}_{\infty 0} := \mathsf{e}_\infty \wedge \mathsf{e}_0 = \mathsf{e}_{+-}. \tag{A.24}$$

Appendix A.2.1. Up-Projection

Here we motivate the representation of points in CGA. To describe spheres and conformal transformations, it is natural for us to first *stereographically project* a point $\boldsymbol{x}$ from $\mathbb{R}^3$ to $S^3 \hookrightarrow \mathbb{R}^4$ by

$$\mathcal{S}(\boldsymbol{x}) := \frac{2\boldsymbol{x}}{\boldsymbol{x}^2+1} + \frac{\boldsymbol{x}^2-1}{\boldsymbol{x}^2+1}\mathsf{e}_\infty. \tag{A.25}$$

Moreover, in order for affine transformations be allowed in our algebra, we need to *homogenize* the objects by including a fifth dimension e_-. This is the *up-projection*, also called the *Hestenes' embedding*, defined by

$$\mathcal{X}(\boldsymbol{x}) := \frac{1}{2}(\boldsymbol{x}^2+1)(\mathcal{S}(\boldsymbol{x}) + \mathsf{e}_-). \tag{A.26}$$

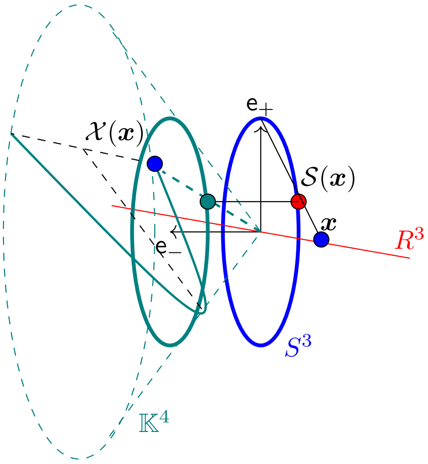


FIGURE A.1: Null Cone and Up-Projection

This embedding can up-project a point $\boldsymbol{x} \in \mathbb{R}^3$ to $\mathbb{R}^{4,1} = \mathcal{G}^1_{4,1}$ as

$$X \equiv \mathcal{X}(\boldsymbol{x}) := \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^2\mathsf{e}_\infty + \mathsf{e}_0. \tag{A.27}$$

Arbitrary scalings of the image of $\mathbb{R}^3 \cup \{\infty\}$ (the extended reals, by including the point(s) at infinity) by $\mathcal{X}$ is a 4-dimensional cone $\mathbb{K}^4$ in $\mathbb{R}^{4,1}$ termed the *null cone* (see Figure A.1). For all $\alpha \in \mathbb{R}$, they satisfy

$$\left(\alpha\mathcal{X}(\boldsymbol{x})\right)^2 = 0, \tag{A.28}$$

in fact, this is a necessary and sufficient condition for points on the null cone. For all points on the null cone, an inverse embedding, called the down-projection, exists to project the points back to $\mathbb{R}^3$:

$$\mathcal{X}^{-1} : X \mapsto \boldsymbol{x} = \frac{\sum_{i=1}^3 (X \cdot \mathsf{e}_i)\mathsf{e}_i}{-X \cdot \mathsf{e}_\infty}. \tag{A.29}$$

It is mentioned that $\mathcal{G}_{4,1}$ includes spheres, planes, circles and lines as algebraic objects, i.e., as multivectors, so how do we define and justify the geometric interpretation of multivectors? For a multivector $A \in \mathcal{G}_{4,1}$, we shall define two null spaces associated with it: the *inner product null space* and the *outer product null space*. They are defined, respectively, as

$$\mathcal{IN}(A) := \{\boldsymbol{x} \in \mathbb{R}^3 \mid \mathcal{X}(\boldsymbol{x}) \cdot A = 0\}, \text{ and} \tag{A.30}$$
$$\mathcal{ON}(A) := \{\boldsymbol{x} \in \mathbb{R}^3 \mid \mathcal{X}(\boldsymbol{x}) \wedge A = 0\}. \tag{A.31}$$

A geometric object can each be described using the inner product or the outer product null space representation. The inner product null space representation is easier to describe the meets (intersections) of geometric objects, while the outer product null space representation is easier to describe the joins (unions) of geometric objects. The two representations are interchangeable by dualizing the representing multivector.

Consider the null space of $A \wedge B$, where A, B are two multivectors in $\mathcal{G}_{4,1}$. First note that for A be grade-1, by utilizing eqn.(A.18),

$$\begin{aligned}\boldsymbol{x} \in \mathcal{IN}(A \wedge B) &\Leftrightarrow \mathcal{X}(\boldsymbol{x}) \cdot (A \wedge B) = 0\\ &\Leftrightarrow (\mathcal{X}(\boldsymbol{x}) \cdot A)B - A \wedge (\mathcal{X}(\boldsymbol{x}) \cdot B) = 0\\ &\Leftrightarrow \boldsymbol{x} \in \mathcal{IN}(A) \cap \mathcal{IN}(B).\end{aligned}$$

Hence, the outer product is equivalent to the meet of geometric objects under the inner product null space representation. Next, note that

$$\begin{aligned}\boldsymbol{x} \in \mathcal{ON}(A \wedge B) &\Leftrightarrow \mathcal{X}(\boldsymbol{x}) \wedge (A \wedge B) = 0\\ &\Leftrightarrow \boldsymbol{x} \in \mathcal{ON}(A) \cup \mathcal{ON}(B).\end{aligned}$$

Hence, the outer product is equivalent to the join of geometric objects under the outer product null space representation.

Moreover, it can be shown that for $A \in \mathcal{G}_{4,1}$ having grade not equal to 0 or 5, then

$$A \wedge \mathcal{X}(\boldsymbol{x}) = (A^\star \cdot \mathcal{X}(\boldsymbol{x}))^{-\star}. \tag{A.32}$$

Therefore,

$$\mathcal{ON}(A) = \mathcal{IN}(A^\star). \tag{A.33}$$

Finally, note that any multivector A when scaled by a nonzero scalar α, it's geometric representation doesn't change. We thus denote such equivalence relation as

$$A \sim \alpha A \quad (\alpha \neq 0). \tag{A.34}$$

Appendix A.2.2. Inner Product Null Space

We shall start from the multivectors with grades equal to 1 and build towards grade-4 objects.

Consider the vector

$$S = \mathcal{X}(\boldsymbol{a}) - \frac{1}{2}\rho^2\mathsf{e}_\infty, \tag{A.35}$$

where $\rho \in \mathbb{R}$, $\rho \geq 0$ and $\boldsymbol{a} \in \mathbb{R}^3$. Then

$$\mathcal{X}(\boldsymbol{x}) \cdot S = 0 \Leftrightarrow (\boldsymbol{x} - \boldsymbol{a})^2 = \rho^2. \tag{A.36}$$

Hence, eqn.(A.35) represents a sphere. The special case where $\rho = 0$ results in $\mathcal{X}(\boldsymbol{a})$ being a sphere with zero radius, i.e. a point. Another limit is by taking the center of sphere at$\boldsymbol{a} - \rho\hat{\boldsymbol{a}}$ and $\rho \to \infty$, obtaining

$$P \sim \hat{\boldsymbol{a}} + |\boldsymbol{a}|\mathsf{e}_\infty. \tag{A.37}$$

This is a plane intersecting $\boldsymbol{a}$ with normal $\hat{\boldsymbol{a}} := \boldsymbol{a}/|\boldsymbol{a}|$.

Next, by meeting two grade-1 spheres S_1 and S_2, we obtain their intersection as a grade-2 circle $C = S_1 \wedge S_2$. If the spheres are taken to be planes $P_1 = \hat{\boldsymbol{a}} + \alpha\mathsf{e}_\infty$ and $P_2 = \hat{\boldsymbol{b}} + \beta\mathsf{e}_\infty$, then their meet will be the line $L = P_1 \wedge P_2 = \hat{\boldsymbol{a}} \wedge \hat{\boldsymbol{b}} + (\beta\hat{\boldsymbol{a}} - \alpha\hat{\boldsymbol{b}}) \wedge \mathsf{e}_\infty$.

We know that three spheres meet at a point pair. Hence for spheres S_1, S_2 and S_3, we have the point pair $Q = S_1 \wedge S_2 \wedge S_3$. When one point of the point pair is located at infinity, then we call this point pair a *homogeneous point*.

Lastly, four spheres meet at a single point. Hence a point can either be represented by a zero-radius sphere, or as $S_1 \wedge S_2 \wedge S_3 \wedge S_4$.

$\mathcal{IN}$	$\mathcal{ON}$	Grade	# of Basis	Basis
Sphere S (Point A), Plane P	Point A	1	5 4	e_1, e_2, e_3, e_∞, e_0
Circle $S_1 \wedge S_2$, Line $P_1 \wedge P_2$	Point Pair $A \wedge B$, Homo. Point $A \wedge \mathsf{e}_\infty$	2	10 6/4	e_{23}, e_{31}, e_{12}, $\underline{\mathsf{e}_{1\infty}}$, $\underline{\mathsf{e}_{2\infty}}$, $\underline{\mathsf{e}_{3\infty}}$, e_{10}, e_{20}, e_{30}, $\underline{\mathsf{e}_{0\infty}}$
Point Pair $S_1 \wedge S_2 \wedge S_3$, Homo. Point $P_1 \wedge P_2 \wedge P_3$	Circle $A \wedge B \wedge C$, Line $A \wedge B \wedge \mathsf{e}_\infty$	3	10 4/6	$\underline{\mathsf{e}_{23\infty}}$, $\underline{\mathsf{e}_{31\infty}}$, $\underline{\mathsf{e}_{12\infty}}$, e_{230}, e_{310}, e_{120}, $\underline{\mathsf{e}_{10\infty}}$, $\underline{\mathsf{e}_{20\infty}}$, $\underline{\mathsf{e}_{30\infty}}$, e_{123}
Point $S_1 \wedge S_2 \wedge S_3 \wedge S_4$	Sphere $A \wedge B \wedge C \wedge D$, Plane $A \wedge B \wedge C \wedge \mathsf{e}_\infty$	4	5 4	$\mathsf{e}_{123\infty}$, $\mathsf{e}_{230\infty}$, $\mathsf{e}_{310\infty}$, $\mathsf{e}_{120\infty}$, $\underline{\mathsf{e}_{1230}}$

∘ Note: Blue and teal objects are those that extended to infinity
The former being the inner product null space representation, and cannot contain e_0's;
the latter being the outer product null space representation, and must contain $\underline{\mathsf{e}_\infty}$.

TABLE A.2: Geometric Objects in Null Space Representations

Appendix A.2.3. Outer Product Null Space

By eqn.(A.33), we know that the inner and outer product null space representation of the same geometric object uses dual algebraic entities.

Since for all $A = \mathcal{X}(\boldsymbol{a})$, we have $A \wedge \mathcal{X}(\boldsymbol{a}) = 0$. The grade-1 point is represented by the same algebraic entity in both representations. Subtle difference between the two lies in the fact that the inner product null space relies on the signature while the outer product null space depends only on the algebraic structure.

By joining two points, we effectively obtain a grade-2 point pair $Q = A \wedge B$ representing the points $\boldsymbol{a}$ and $\boldsymbol{b} \in \mathbb{R}^3$, where $A = \mathcal{X}(\boldsymbol{a})$ and $B = \mathcal{X}(\boldsymbol{b})$. I.e. $\mathcal{ON}(A \wedge B) = \{\boldsymbol{a}, \boldsymbol{b}\}$. If one of the point is at infinity: $B = \mathcal{X}(\infty) = \mathsf{e}_\infty$, then we have the grade-2 homogeneous point $H = A \wedge \mathsf{e}_\infty = \boldsymbol{a} \wedge \mathsf{e}_\infty + \mathsf{e}_0 \wedge \mathsf{e}_\infty$, where $\mathcal{ON}(H) = \{\boldsymbol{a}\}$. To extract one of the points from a point pair $Q = A \wedge B$, we have the following useful formula:

$$\left(1 \pm \frac{Q}{\sqrt{Q^2}}\right)(Q \cdot \mathsf{e}_\infty) = -2A \text{ or } 2B \sim A \text{ or } B. \quad \text{(A.38)}$$

By joining three points, we uniquely defined a grade-3 circle $K = A \wedge B \wedge C$ going through the three points. A grade-3 line would require one of the points be at infinity or that they are co-linear.

Lastly, by joining four points we can obtain a grade-4 sphere $S = A \wedge B \wedge C \wedge D$. If one of the point is at infinity or the points are co-planar, we obtain a grade-4 plane.

All the representations of a geometric object is summarized in Table A.2. In practice, we often switch between the two representations of the geometric objects and opt for the representation with a lower grade for easier calculations.

Appendix A.2.4. Transformations

Geometric algebra (CGA especially) is useful in describing geometries not only because of the inclusion of geometric objects as algebraic entities, but also because it includes transformations that are described by the geometric objects that are left invariant. The following discussions are all in the inner product null space representation

The first of such transformation is reflection by plane P and *inversion*[2] by sphere S. Consider a geometric object A, its sandwich product by P is

$$PAP, \quad \text{(A.39)}$$

equivalent to A reflected across P. Similarly,

$$SAS \quad \text{(A.40)}$$

represents A inverted across the sphere S. Note that plane reflection is just a special case of sphere inversion for when the radius of the sphere goes to infinity.

We can further composite the sphere inversions to obtain all *conformal transformations*. We shall discuss those transformations that are useful for our purpose, i.e. translations and rotations that can actually be applied on rigid bodies[3]. Consider two consecutive reflection across parallel planes $P_1 = \hat{\boldsymbol{a}} + \alpha_1 \mathsf{e}_\infty$ and $P_2 = \hat{\boldsymbol{a}} + \alpha_2 \mathsf{e}_\infty$, all geometric entities should be translated by $\boldsymbol{t} = 2(\alpha_2 - \alpha_1)\hat{\boldsymbol{a}}$[4]. Algebraically, we have

$$A \mapsto P_2 P_1 A P_1 P_2 = \left(1 - \frac{\boldsymbol{t}}{2}\mathsf{e}_\infty\right) A \left(1 + \frac{\boldsymbol{t}}{2}\mathsf{e}_\infty\right) \quad \text{(A.41)}$$

$$= \mathsf{e}^{-\boldsymbol{t}\mathsf{e}_\infty/2} A \mathsf{e}^{\boldsymbol{t}\mathsf{e}_\infty/2}. \quad \text{(A.42)}$$

Without a doubt, the direction vector $\boldsymbol{t}$ is left invariant by the translation transformation. The exponential term is called the *translator*.

[2] Consider a sphere with radius ρ and center $\boldsymbol{c}$, then the inversion of the point $\boldsymbol{a}$ by the sphere is

$$\boldsymbol{c} + \frac{\rho^2}{|\boldsymbol{a} - \mathbf{c}|^2}(\boldsymbol{a} - \boldsymbol{c}).$$

[3] The compositions of translations and rotations form the special Euclidean group SE(3).

[4] Note the double distance!

If the two planes of reflection are not parallel, with $P_1 = \hat{\boldsymbol{a}}_1 + \alpha_1 \mathsf{e}_\infty$ and $P_2 = \hat{\boldsymbol{a}}_2 + \alpha_2 \mathsf{e}_\infty$. The two planes intersect at an angle $\frac{\theta}{2}$ across the line $L = P_1 \wedge P_2 / \sin\frac{\theta}{2}$. Then the consecutive reflection generates a rotation of θ[5] around the line L:

$$A \mapsto P_2 P_1 A P_1 P_2 = \mathrm{e}^{-L\theta/2} A \mathrm{e}^{L\theta/2}, \tag{A.43}$$

since

$$P_2 P_1 = \cos\frac{\theta}{2} - L \sin\frac{\theta}{2}.$$

The exponential term is called the *rotor*. The exponent of the rotor is the outer product of two linearly independent vectors lying on the plane of rotation. The dual of the exponent is the usual "cross product" direction of rotation. Once again, the rotation transformation is described by the line left invariant by it.

Composition of rotors and translators, as a generalization of the both, is often called a *screw* since it generates a screw motion in space. And in CGA, we often call translators, rotors and screws together as *motors*, objects that describes motion in space.

[5] Note the double angle!