

Two Dimensional Transformations and Its Application to Vehicle Motion Control and Analysis

Yutaka Kanayama
David MacPherson
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943

Gary Krahn
Department of Mathematical Sciences
United States Military Academy
West Point, NY 10996

Abstract

This paper proposes a new 2D transformation group in order to represent positioning of rigid body vehicles working in a plane. This group is a variation of the 3D homogeneous transformation group. The relationship between this transformation group and the group of all invertible 4×4 matrices is discussed. One of the advantages of this formulation is that the orientation of the vehicle is explicitly given, not by its trigonometric function values. This group theory is applied to describe the motion of vehicles and odometry error analysis for vehicles. This theory has already been implemented in the MML software system for the autonomous mobile robot Yamabico-11, in which only the "transformation" data structure, the "composition" function, and the "inverse" function.

1 Introduction

The 3D homogeneous transformation groups and quaternions are widely used in analysis and design of robot manipulators [1, 2, 3]. Likewise, we need a 2D transformation group to represent positioning of rigid body vehicles placed in a plane. However, the formulation to be given here is not merely the 2D version of existing transformation groups. We would like to express a robot's orientation θ explicitly to avoid losing complete information beyond the range of $[-\pi, \pi]$. This formulation has the same advantage as 3D homogeneous transformations, i.e. translation and rotation are described in a single mathematical structure, a configuration. This algebraic system is a variation of the 3D homogeneous transformation group. However, the system does not have a point of singularity, which was one of the drawbacks of the homogeneous transformations. Detailed analysis is given in Section 6.

This theory has made calculus related to vehicle motion design and control transparent and easy, including the analysis of odometry errors described in Section 4. This algebraic system was already implemented in the high level mobile robot language, MML for the autonomous mobile robot Yamabico-11 [4, 5]. All the definitions and the basic functions (composition, inverse, symmetric property, and so forth) provide a powerful and simple user-interface.

2 Transformation Group

Let \mathfrak{R} denote the set of all real numbers. A transformation q is

$$q \equiv \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad x, y, \theta \in \mathfrak{R}. \quad (1)$$

The set of all transformations is denoted by $\mathcal{T} (= \mathfrak{R}^3)$. For instance, $(2, 1, \pi/6)^T$ and $(2, 4, \pi/4)^T$ are examples of transformations (M^T means the transposition of a matrix M). Obviously, a transformation q is interpreted as a two dimensional coordinate transformation from the global Cartesian coordinate system \mathcal{F}_0 to another coordinate system \mathcal{F} .

Let $q_1 \equiv (x_1, y_1, \theta_1)^T$ and $q_2 \equiv (x_2, y_2, \theta_2)^T$. The composition $q_1 \circ q_2$ of these transformations is defined as

$$q_1 \circ q_2 \equiv \begin{pmatrix} x_1 + x_2 \cos \theta_1 - y_2 \sin \theta_1 \\ y_1 + x_2 \sin \theta_1 + y_2 \cos \theta_1 \\ \theta_1 + \theta_2 \end{pmatrix}. \quad (2)$$

We write $q_1 = q_2$ if and only if $x_1 = x_2, y_1 = y_2$ and there exists an integer n such that $\theta_1 = \theta_2 + 2n\pi$. The interpretation of $q_1 \circ q_2$ in the domain of 2D coordinate transformation is the composition of the coordinate

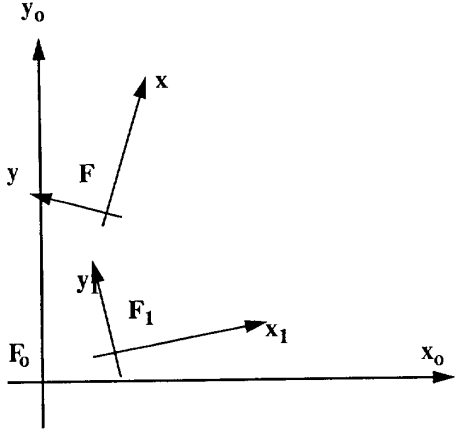


Figure 1: Composition

transformations p_1 and p_2 (Fig. 1). The following is one of immediate results from the definition above.

Corollary 2.1 For any $q = (x, y, \theta)^T \in \mathcal{T}$,

$$\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} = \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} \circ \begin{pmatrix} 0 \\ 0 \\ \theta \end{pmatrix}. \quad (3)$$

Therefore, a transformation (x, y, θ) can always be decomposed into a *translation* $(x, y, 0)^T$ and a *rotation* $(0, 0, \theta)^T$. Notice, however, that in general $q \neq (0, 0, \theta)^T \circ (x, y, 0)^T$, since the composition function is non-abelian. The composition of the example transformations stated above is

$$\begin{pmatrix} 2 \\ 1 \\ \frac{\pi}{6} \end{pmatrix} \circ \begin{pmatrix} 2 \\ 4 \\ \frac{\pi}{4} \end{pmatrix} = \begin{pmatrix} \sqrt{3} \\ 2 + 2\sqrt{3} \\ \frac{5}{12}\pi \end{pmatrix}.$$

Lemma 2.1 (Closure) For any $q_1, q_2 \in \mathcal{T}$,

$$q_1 \circ q_2 \in \mathcal{T}. \quad (4)$$

Lemma 2.2 (Associativity) For any $q_1, q_2, q_3 \in \mathcal{T}$,

$$(q_1 \circ q_2) \circ q_3 = q_1 \circ (q_2 \circ q_3). \quad (5)$$

Lemma 2.3 (Identity) For all $q \in \mathcal{T}$,

$$q \circ (0, 0, 0)^T = (0, 0, 0)^T \circ q = q. \quad (6)$$

Therefore, $(0, 0, 0)^T \equiv e$ is the unique identity element in \mathcal{T} .

The following Lemma demonstrates the presence of the left and right inverse transformations for each $q \in \mathcal{T}$.

Lemma 2.4 (Right and Left Inverse)

Let $q \equiv (x, y, \theta)^T$ be given.

(1) The solution to an equation $q_1 \circ q = e$ is

$$q_1 = \begin{pmatrix} -x \cos \theta - y \sin \theta \\ x \sin \theta - y \cos \theta \\ -\theta \end{pmatrix}. \quad (7)$$

(2) The solution to an equation $q \circ q_2 = e$ is

$$q_2 = \begin{pmatrix} -x \cos \theta - y \sin \theta \\ x \sin \theta - y \cos \theta \\ -\theta \end{pmatrix}. \quad (8)$$

Since the left inverse q_1 and right inverse q_2 are equal there exists a unique inverse q^{-1} for each $q \in \mathcal{T}$.

Lemma 2.5 (Inverse)

For any transformation $q \equiv (x, y, \theta)^T$, there exists a unique inverse

$$q^{-1} \equiv \begin{pmatrix} -x \cos \theta - y \sin \theta \\ x \sin \theta - y \cos \theta \\ -\theta \end{pmatrix}. \quad (9)$$

For instance, the inverse of a transformation $(4, 2, \pi/6)^T$ is

$$\begin{pmatrix} 4 \\ 2 \\ \pi/6 \end{pmatrix}^{-1} = \begin{pmatrix} -1 - 2\sqrt{3} \\ 2 - \sqrt{3} \\ -\pi/6 \end{pmatrix} \approx \begin{pmatrix} -4.464 \\ 0.268 \\ -\pi/6 \end{pmatrix}.$$

Proposition 2.1 The set \mathcal{T} of transformation is a group with respect to the composition operation, denoted by $\langle \mathcal{T}, \circ \rangle$.

Proof. The algebraic structure $\langle \mathcal{T}, \circ \rangle$ satisfies the closure property by Lemma 2.1, the associative law by Lemma 2.2, the existence of the identity by Lemma 2.3, and the existence of inverses by Lemma 2.5. \square

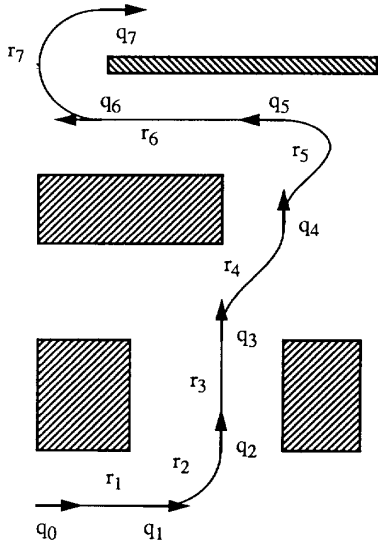


Figure 2: Smooth Path Generated by Configurations

3 Configuration of Vehicle

We assume a global Cartesian coordinate system \mathcal{F}_0 in a two dimensional plane \mathbb{R}^2 and a vehicle placed on this plane. The vehicle has a body-fixed Cartesian coordinate system \mathcal{F}_v attached to its body. The x-axis of \mathcal{F}_v is pointing out of the front of the vehicle. Since the vehicle moves, \mathcal{F}_v is a function of time. The vehicle's position in this plane is described by a *configuration* $q = (x, y, \theta)^T$, where (x, y) is the position of the origin of \mathcal{F}_v and θ the orientation of \mathcal{F}_v in the global coordinate system [2]. A vehicle configuration $(x, y, \theta)^T$ can be interpreted as a transformation $(x, y, \theta)^T$ which transforms \mathcal{F}_0 into \mathcal{F}_v . Under this interpretation, we can use the group theory for the control and analysis of vehicle motion.

There is a method for describing a vehicle's motion by a sequence of configurations [5]. We should select a set of configurations so that any path segment which is obtained by the smooth path planner passes through the constraints specified [4]. For instance, a path shown in Figure 2 is described by a sequence

$$\begin{aligned} & (q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7) \\ & = ((0, 0, 0)^T, (2, 0, 0)^T, (3, 1, \pi/2)^T, (3, 3, \pi/2)^T, \end{aligned}$$

$$(4, 5, \pi/2)^T, (4, 7, \pi)^T, (1, 7, \pi)^T, (1, 9, 0)^T).$$

of eight configurations, which are also shown in the figure. These configurations are constraints to the path and should be selected so that each path segment which is obtained by the smooth path planner will not have a conflict with the environment [4].

We may want to specify the next configuration relative to the current vehicle configuration, because it may be easier for us to calculate a relative configuration rather than a global configuration. For instance, when the current vehicle's configuration is q_0 , a relative configuration r is given so that the next configuration q_1 is given by the composition operation:

$$q_1 = q_0 \circ r. \quad (10)$$

In the previous example, each (absolute) configuration q_i is calculated by

$$q_i = q_{i-1} \circ r_i, \text{ for } i = 1, \dots, 7 \quad (11)$$

using a sequence of relative configurations:

$$\begin{aligned} & (r_1, r_2, r_3, r_4, r_5, r_6, r_7) \\ & = ((2, 0, 0)^T, (1, 1, \pi/2)^T, (2, 0, 0)^T, (2, -1, 0)^T, \\ & (2, 0, \pi/2)^T, (3, 0, 0)^T, (0, -2, \pi)^T). \end{aligned}$$

These relative configurations are also shown in Figure 2.

4 Odometry Error Analysis

One of the most frustrating problems in controlling autonomous mobile robots is the accumulation of odometry (dead reckoning) errors over time. If an error is excessive, the vehicle may become lost. The vehicle configuration q_o estimated by the on board odometry function is called the *odometry configuration*. Consider a situation in which the odometry configuration is q_o and its actual configuration is q_a . If there is no odometry error, $q_o = q_a$. Otherwise, there is a difference between where the vehicle thinks it is and where the vehicle really is (Figure 3). We propose to define an *error configuration* ϵ such that

$$\epsilon \circ q_o = q_a. \quad (12)$$

That is, this vehicle's best estimate of its configuration is q_o is correct only in the "erroneous frame" ϵ . If q_o and q_a are determined, the error configuration can be calculated by

$$\epsilon = q_a \circ q_o^{-1}. \quad (13)$$

For instance, if $q_o = (2, 1, 0)^T$ and $q_a = (4, 3, \pi/6)^T$,

$$\begin{aligned}\epsilon &= q_a \circ q_o^{-1} = \begin{pmatrix} 4 \\ 3 \\ \pi/6 \end{pmatrix} \circ \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}^{-1} \\ &= \begin{pmatrix} 9/2 - \sqrt{3} \\ 2 - \sqrt{3}/2 \\ \pi/6 \end{pmatrix} \approx \begin{pmatrix} 2.768 \\ 1.134 \\ \pi/6 \end{pmatrix}.\end{aligned}$$

i.e., the vehicle's odometry configuration $q_o = (2, 1, 0)^T$ is correct if it is interpreted as a local configuration in ϵ .

Let us analyze an error which is caused by a sequence of vehicle motion: A vehicle has traveled through a series of two configurations q_{o1} and q_{o2} in this order. These are estimated by odometry. Let q_o be their composition. Also, let the vehicle's actual movement as measured by an outside observer be q_{a1} and q_{a2} , where q_a is their composition.

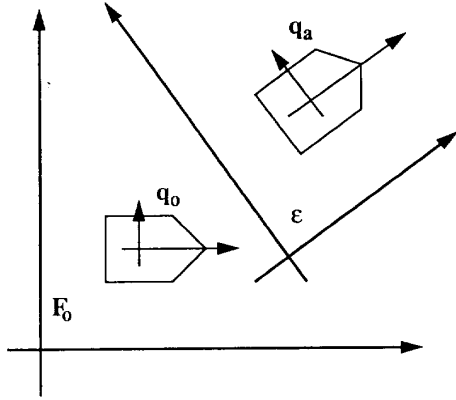


Figure 3: Vehicle Odometry Error

$$q_o = q_{o1} \circ q_{o2} \quad (14)$$

$$q_a = q_{a1} \circ q_{a2}. \quad (15)$$

The equations for the errors are:

$$q_{a1} = \epsilon_1 \circ q_{o1} \quad (16)$$

$$q_{a2} = \epsilon_2 \circ q_{o2} \quad (17)$$

$$q_a = \epsilon \circ q_o. \quad (18)$$

Therefore, if each odometry motion estimate and error are known, the total error is

$$\epsilon = q_a \circ q_o^{-1} = (q_{a1} \circ q_{a2}) \circ (q_{o1} \circ q_{o2})^{-1}$$

$$= \epsilon_1 \circ q_{o1} \circ \epsilon_2 \circ q_{o2} \circ q_{o2}^{-1} \circ q_{o1}^{-1} \quad (19)$$

$$= \epsilon_1 \circ q_{o1} \circ \epsilon_2 \circ q_{o1}^{-1}. \quad (20)$$

The involvement of q_1 in this equation makes the error analysis complex. Similarly, the total error equation for n consecutive motions is as follows:

$$\begin{aligned}\epsilon &= \epsilon_1 q_{o,1} \cdots \epsilon_{n-1} q_{o,n-1} \epsilon_n q_{o,n} q_{o,n}^{-1} q_{o,n-1}^{-1} \cdots q_{o,1}^{-1} \\ &= \epsilon_1 q_{o,1} \cdots \epsilon_{n-1} q_{o,n-1} \epsilon_n q_{o,n-1}^{-1} \cdots q_{o,1}^{-1}.\end{aligned} \quad (21)$$

Assume a special case in which $q_{o,1} = \cdots = q_{o,n-1} = e$; i.e., after each component motion q_{oi} , the vehicle is commanded to come back to the initial configuration. In this case, the error configuration becomes simply the composition of all the individual errors

$$\epsilon = \epsilon_1 \circ \epsilon_2 \circ \cdots \circ \epsilon_n. \quad (22)$$

5 Model/Sensor Based Error Detection

Not only the positioning of a vehicle, but also that of any object in the environment may be described by a configuration. For instance, Object A in Figure 4 may be assigned a body-fixed, local coordinate system and its position in this world is described using this local frame.

Furthermore, we consider a situation in which an ideal sensor mounted on the vehicle senses the configuration on an object in the environment. That is, the vehicle is able to sense the relative configuration of an object with respect to its own odometry configuration q_o with an infinite precision. Therefore, the vehicle's odometry error is superimposed upon the sensed object configuration.

We propose a method of detecting an vehicle odometry error by using an external landmark as a point of reference. In Figure 5, q_a is the vehicle's actual configuration, which is unknown, and p_a the configuration of an object A in the environment, which is obtained from an environmental model. The odometry configuration q_o is known, but may contain an error. The configuration p_o is the observed configuration of the object A , and may have some error, because this observation is made by the ideal sensor on board using the odometry configuration. As discussed in Section 4, a possible difference between p_a and p_o is due to the error ϵ in odometry.

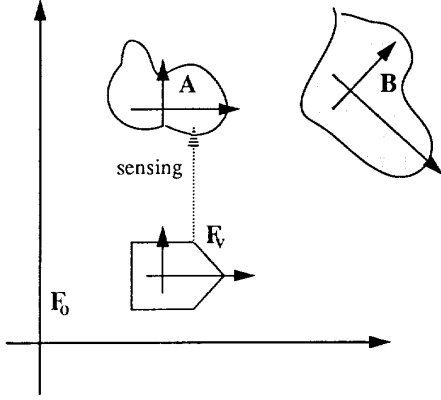


Figure 4: Object Configurations

The vehicle actually senses the object A at p_a , but since it thinks it is at q_o , it believes A is at p_o . Therefore, by introducing the error configuration ϵ ,

$$\epsilon \circ q_o = q_a \quad (23)$$

$$\epsilon \circ p_o = p_a. \quad (24)$$

From Equation 24, the error configuration is

$$\epsilon = p_a \circ p_o^{-1}. \quad (25)$$

Since both p_a and p_o are known, ϵ also becomes known. Therefore, from Equation 23,

$$q_a = \epsilon \circ q_o = p_a \circ p_o^{-1} \circ q_o. \quad (26)$$

Since p_a is obtained by the model and p_o is obtained by the sensor, the last equation formalizes the principle of the model/sensor based odometry error detection.

For instance, if $q_o = (2, 1, 0)^T$, $p_o = (2, 3, 0)^T$, and $p_a = (3, 3 + \sqrt{3}, \pi/6)^T$, then

$$\epsilon = p_a \circ p_o^{-1} = \begin{pmatrix} 3 \\ 3 + \sqrt{3} \\ \pi/6 \end{pmatrix} \circ \begin{pmatrix} 2 \\ 3 \\ 0 \end{pmatrix}^{-1} = \begin{pmatrix} 9/2 - \sqrt{3} \\ 2 - \sqrt{3}/2 \\ \pi/6 \end{pmatrix}$$

and

$$q_a = \epsilon \circ q_o = \begin{pmatrix} 9/2 - \sqrt{3} \\ 2 - \sqrt{3}/2 \\ \pi/6 \end{pmatrix} \circ \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \\ \pi/6 \end{pmatrix}.$$

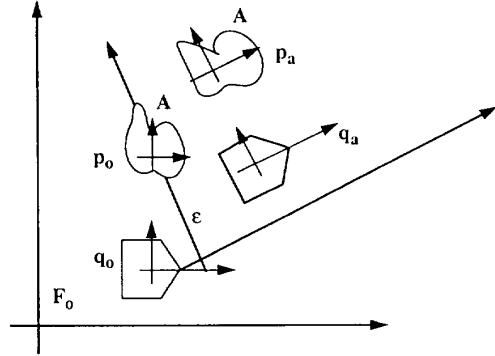


Figure 5: Vehicle Odometry Error Detection

6 Relation to Other Transformation Groups

The use of the three-dimensional homogeneous transformation group is common in the robotics field [1]. The general form of a homogeneous transformation is

$$T_3 = \begin{bmatrix} R_{11} & R_{12} & R_{13} & x \\ R_{21} & R_{22} & R_{23} & y \\ R_{31} & R_{32} & R_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (27)$$

where the left-top 3×3 submatrix represents rotation and the right-top 3×1 matrix a translation. Its two-dimensional version is

$$T_2 = \begin{bmatrix} R_{11} & R_{12} & x \\ R_{21} & R_{22} & y \\ 0 & 0 & 1 \end{bmatrix} \quad (28)$$

$$= \begin{bmatrix} \cos \theta & -\sin \theta & x \\ \sin \theta & \cos \theta & y \\ 0 & 0 & 1 \end{bmatrix}. \quad (29)$$

We may be able to use this transformation matrix T_2 to represent vehicle's configuration as described in Section 3. In order to obtain θ itself from T_2 , we need the evaluation

$$\theta = \text{atan2}(R_{21}, R_{11}), \quad (30)$$

where the range of the function atan2 is assumed $[-\pi, \pi]$. However, this method has a drawback. If the vehicle's accumulated rotation is beyond the range of $[-\pi, \pi]$, a part of the orientational information is lost.

For instance, if the vehicle rotates 2π counterclockwise, θ becomes 0 instead of 2π if we use Equation 30. In order to avoid this information loss, we should add an explicit θ term to T in Equation 28

$$T' = \begin{bmatrix} \cos \theta & -\sin \theta & x & 0 \\ \sin \theta & \cos \theta & y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \theta & 1 \end{bmatrix}. \quad (31)$$

The set T' of all 4×4 matrices of the form in Equation 31 under matrix multiplication is a subgroup of all invertible 4×4 matrices. Therefore $\langle T', \bullet \rangle$ is a group.

Proposition 6.1 $\langle T, \circ \rangle$ is isomorphic to $\langle T', \bullet \rangle$.

Although the groups $\langle T, \circ \rangle$ and $\langle T', \bullet \rangle$ are isomorphic, operations in the transformation group $\langle T, \circ \rangle$ are simpler to represent and are computationally more efficient than matrix multiplication in $\langle T', \bullet \rangle$. This computational enhancement is essential for transformation calculations performed by an autonomous robotic platform. Obviously, this transformation system does not have any singularities.

7 Conclusion

We have described a 2D configurational algebra based upon group theory. Group theory provides an elegant algebraic structure that make robot motion calculations more transparent. Motion analysis and robot odometry error determination are made possible using this algebra. The periodic detection and reduction of odometry errors allows an autonomous mobile robot to work for a sustained period with great precision.

We have applied this theory experimentally using Yamabico-11. The group theory functions described have been incorporated in the Model-Based Mobile Robot Language (MML). Several experiments were conducted in which odometry corrections were performed in real-time using landmarks with known configurations. This experiment was performed repeatedly at several different speeds. In the worst case, average odometry error was 2.54 cm in distance and 1.04 degrees in orientation over a 914 cm course.

One of the open problems related to this theory is whether there exists a similar theory in three dimensional transformations, i.e., how can we define

the “composition” and “inverse” for transformations $q = (x, y, z, \phi, \theta, \psi)^T$. Here, ϕ, θ and ψ are Euler angles [1].

References

- [1] Paul, R., “Robot Manipulators: Mathematics, Programming, and Control,” The MIT Press, 1984.
- [2] Lozano-Perez, T., “Spatial Planning: A Configuration Space Approach,” *IEEE Transaction on Computers*, vol. C-32, no. 2, pp. 108-119, Feb. 1983.
- [3] Bloch, N., *Abstract Algebra with Applications*, 1987, Prentice Hall, Inc.
- [4] Kanayama, Y. and Hartman, B., “Smooth Local Path Planning for Autonomous Vehicles,” IEEE Int. Conf on Robotics and Automation, (1989), pp. 1265-1270.
- [5] Kanayama, Y., and Onishi, M., “Locomotion Functions in the Mobile Robot Language, MML,” IEEE Int. Conf on Robotics and Automation, (1991), pp. 1110-1115.