

Lecture Notes 5.C¹

Motion Control: Why Order Matters

Naomi Ehrich Leonard
Department of Mechanical and Aerospace Engineering
Princeton University, Princeton NJ 08544

1 Matrix Multiplication and Geometry

In Lecture Notes 5.B, we defined a matrix and its interpretation as a linear transformation. A linear transformation assigns to every point in two (or three) dimensional space a new location in two (or three) dimensional space. We saw that linear transformations can be used to describe, for example, translation or rotation of an object such as a spacecraft.

Our next step is to define multiplication of matrices so that the product of two matrices, call them A and B , is the matrix $C = B \times A = BA$ that is equivalent to the linear transformation that results if we first transform by A and then transform by B . In other words, we want multiplication of matrices to be our language for successive transformations.

The rules for multiplying two 2×2 matrices (discussed in the lab) are as follows:

$$\begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a'a + b'c & a'b + b'd \\ c'a + d'c & c'b + d'd \end{pmatrix}. \quad (1.1)$$

For example,

$$\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}.$$

Now, if we first rotate the planar spacecraft by θ degrees then the corresponding linear transformation is

$$A = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}.$$

If we then rotate the spacecraft by an additional θ' degrees, the corresponding linear transformation for this second step is

$$B = \begin{pmatrix} \cos \theta' & -\sin \theta' \\ \sin \theta' & \cos \theta' \end{pmatrix}.$$

Certainly, after we have performed the two successive rotations, the net result is that the spacecraft has been rotated by $(\theta + \theta')$ degrees. The transformation corresponding to rotation

¹©by Naomi Ehrich Leonard, 2000

by $(\theta + \theta')$ degrees is given by the matrix

$$C = \begin{pmatrix} \cos(\theta + \theta') & -\sin(\theta + \theta') \\ \sin(\theta + \theta') & \cos(\theta + \theta') \end{pmatrix}.$$

Using the rules of multiplication of matrices defined by (1.1), we can check that as desired $BA = C$, i.e., the product of the two individual matrices gives the matrix that corresponds to the net transformation:

$$\begin{aligned} BA &= \begin{pmatrix} \cos \theta' & -\sin \theta' \\ \sin \theta' & \cos \theta' \end{pmatrix} \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \\ &= \begin{pmatrix} \cos \theta' \cos \theta - \sin \theta' \sin \theta & -(\cos \theta' \sin \theta + \sin \theta' \cos \theta) \\ \sin \theta' \cos \theta + \cos \theta' \sin \theta & -\sin \theta' \sin \theta + \cos \theta' \cos \theta \end{pmatrix} \\ &= \begin{pmatrix} \cos(\theta + \theta') & -\sin(\theta + \theta') \\ \sin(\theta + \theta') & \cos(\theta + \theta') \end{pmatrix} = C. \end{aligned}$$

The rules for multiplying two 3×3 matrices are as follows:

$$\begin{aligned} &\begin{pmatrix} a' & b' & h' \\ c' & d' & i' \\ j' & k' & l' \end{pmatrix} \begin{pmatrix} a & b & h \\ c & d & i \\ j & k & l \end{pmatrix} \\ &= \begin{pmatrix} a'a + b'c + h'j & a'b + b'd + h'k & a'h + b'i + h'l \\ c'a + d'c + i'j & c'b + d'd + i'k & c'h + d'i + i'l \\ j'a + k'c + l'j & j'b + k'd + l'k & j'h + k'i + l'l \end{pmatrix} \end{aligned}$$

Thus, we can *compute* what happens if we first rotate the spacecraft about the x -axis by 90 degrees and then about the y -axis by 90 degrees. Rotation by $\alpha_1 = 90$ degrees about the x -axis is given by the matrix

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Rotation by $\alpha_2 = 90$ degrees about the y -axis is given by the matrix

$$P = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}.$$

The sequence of rotations yields the net transformation given by

$$C = PR = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{pmatrix}.$$

This transformation C takes points (x, y, z) into points

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} y \\ -z \\ -x \end{pmatrix}.$$

For instance, the point $(1, 0, 0)$, i.e., the nose of the spacecraft, gets transformed into the point $(0, 0, -1)$. This is easily checked by picking up a book and performing the above two rotations (in the correct order!). If you keep track of the “nose” of the book, you will see that the transformation matrix C does indeed give the correct final coordinates of the nose.

2 Order Matters

In checking the sequence of two rotation of the spacecraft (or book), we noted above that it is important to keep track of the order in which the two rotations are performed. It is easy to check that if you first rotate the book about the y axis by 90 degrees and then about the x axis by 90 degrees you won't get the same result as in the previous section where you did the rotations in the reverse order. This turns out to be critical for generating motion from non-reciprocal shape change!

When the order of application of two transformations makes a difference in the end result, we say that the two transformations *do not commute*. This lack of commutation in transformations is reflected in the lack of commutation in multiplication of the corresponding transformation matrices. That is, for consistency it is necessary that

$$RP \neq PR.$$

We can easily check that this is the case. We compute

$$D = RP = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \neq \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{pmatrix} = PR = C.$$

Thus, we have confirmed that P and R do not commute, i.e., that rotation about the x axis and rotation about the y axis do not commute.

Recall from Lecture Notes 5.A that we described a non-reciprocal shape change sequence as one in which the way the shape variables were returned to their original place was not simply an undoing of the initial contortion in reverse. In successfully achieving a net re-orientation of her whole body, the astronaut first opens her left arm and then opens her right arm. In returning to her original “shape” in which both arms are folded in, she first closes her *left* arm and then closes her right arm. This is a non-reciprocal shape change. A reciprocal shape change would have been one in which in returning to her original “shape”, she first closed her *right* arm and then her left arm.

Clearly, in this example, order matters, i.e., the non-reciprocal and the reciprocal shape changes do not yield the same final motion of the astronaut. In fact, we make deliberate use of the fact that the order of opening and closing of individual arms matters in constructing the shape change sequence. Since order matters, we can choose a return contortion that differs from the original contortion simply by swapping order of the individual shape change maneuvers. If order did not matter, i.e., if the shape change maneuvers commuted with one

another, then a non-reciprocal shape change sequence would be no different than a reciprocal shape change sequence and we wouldn't be able to get anywhere new!

3 Generating Motion from Shape Change

We can summarize now how all the pieces fit together to complete the story of generating motion from shape change. First, we have introduced the notion of a linear transformation which “moves” a figure around on a plot. To each linear transformation is associated an array of numbers which we call a matrix. Thus, we can use a matrix to represent motion, and more particularly shape change. In the parallel parking example developed in the lab, there are two shape change variables. One of these variables is the roll angle ϕ of the wheel which is equivalent to the forward (or backward) drive distance e . The second variable is the steering angle θ . A matrix P_f , expressed as a function of e , that describes translation of a figure in the plane can be used to represent the shape change associated with “drive” and another matrix P_r , expressed as a function of θ , that describes rotation of a figure in the plane can be used to represent the shape change associated with “steer”. We define

$$P_f(e) = \begin{pmatrix} 1 & 0 & -e \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad P_r(\theta) = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad P_s(f) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -f \\ 0 & 0 & 1 \end{pmatrix}.$$

Here, P_s is an additional linear transformation that corresponds to a sideways translation of the car a distance f .

We note that one should be careful in how one “applies” each of these matrices. In particular, these are not to be applied to the car itself, but rather to the “world” axes, i.e., to the figure that consists of the x -axis and y -axis that are fixed in the world. The transformation should be applied to the world axes with respect to a set of axes fixed in the car, e.g., call them the X -axis and the Y -axis. We take this perspective to allow drive forward to always mean drive in the forward direction of the car (rather than in the world x -axis direction). Similarly, rotation in this context means rotation about the center of the car and not about the center of the world axes. So, for example, applying P_f to the world axes will move it backwards a distance e in the X -direction relative to the car. That is, to a driver in the car, it will look like the world is moving backwards. However, to an observer on the street (i.e., standing still in the world), it will look, of course, as if the car has driven forward a distance e . Similarly, application of P_r to the world axes for a positive angle θ , rotates them right (clockwise), relative to the car's center. Thus, to an observer on the street, the car looks like it is turning left by θ degrees. See the lab (starting on page 2 of Section 5) for further discussion of this perspective.

In the development of our mathematical framework, we next introduced “multiplication” of two matrices which makes it possible to compute the result of a *sequence* of shape changes. For example, the product of two matrices A and B is given by $C = BA$. C is the matrix that describes the net transformation of a figure that is equivalent to having first applied

the transformation A to the figure and then the transformation B to the figure after its transformation by A . In the case of the parallel parking example, we can compute where the car will be after a sequence of drive and steer maneuvers. For example if we first turn (steer) by $\theta = \alpha$ and then drive a distance $e = k$, the matrix that describes the net motion of the car can be computed as

$$\begin{aligned} D &= P_f(e = k)P_r(\theta = \alpha) \\ &= \begin{pmatrix} 1 & 0 & -k \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \cos \alpha & \sin \alpha & -k \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned}$$

Before proceeding with an investigation or implementation of a non-reciprocal shape change sequence, we next check whether or not a non-reciprocal shape change sequence is even possible by checking to make sure that the individual shape change maneuvers do not commute with one another. For example, in the case of parallel parking, we should confirm that $P_fP_r \neq P_rP_f$. This can also be confirmed with a brief experiment with a roll of masking tape. Put the masking tape on its side as if it were a wheel. Roll it forward a little ways and then turn it left by some angle. Now mark the final position and orientation of the tape. Put the masking tape back to its original position and orientation and run the experiment with the shape change maneuvers in the opposite order, i.e., first turn the tape left by the same angle as in the first half of the experiment and then roll the tape forward by the same amount as earlier. Now mark the final position and orientation of the tape. When you compare this final position and orientation to the final position and orientation achieved with the first ordering of shape change maneuvers, you should see that the tape did not end up in the same configuration each time. That is, the order of rotation and translation matters!

We are now in a position to know that we can generate a “new” motion from a non-reciprocal shape change sequence (that is, in the case that we have demonstrated that the shape change maneuvers do not commute). Furthermore, the matrix algebra that we have developed makes it possible to *compute* exactly where the non-reciprocal shape change sequence moves the system! In the case of the car, the net motion on the car as a result of a

non-reciprocal shape change sequence is given by the matrix N where

$$\begin{aligned}
N &= P_f(e = -k)P_r(\theta = -\alpha)P_f(e = k)P_r(\theta = \alpha) \\
&= \begin{pmatrix} 1 & 0 & k \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -k \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & k(1 - \cos \alpha) \\ 0 & 1 & -k \sin \alpha \\ 0 & 0 & 1 \end{pmatrix} \\
&\approx \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -k\alpha \\ 0 & 0 & 1 \end{pmatrix} := P_s(f = k\alpha).
\end{aligned}$$

The last line above follows for small angles α , e.g., if α is less than 20 degrees. Note that this shape change sequence is a parallel park maneuver, i.e., it corresponds to turn left, drive forward, turn right, drive backward. We have shown that for α relatively small, N is the transformation that moves the car in the “sideways” direction by the amount αk (if α is expressed in degrees there is a conversion factor and the amount of net sideways translation that the car experiences is computed to be $\alpha k \pi / 180$).

In the case that two shape change maneuvers do commute, a non-reciprocal shape change sequence will not be possible. That is, we won’t be able to come up with a different way to get back to our original shape other than doing the original contortion in reverse since swapping the order of shape change maneuvers is no longer an option. An example is a car that consists of two wheels: one that drives the car forward and backward and the other that drives the car sideways left and right. Neither wheel can turn. Accordingly, one shape variable is e corresponding to the forward/backward distance traveled and a second shape variable is f corresponding to the sideways distance traveled. The first shape maneuver is described by the matrix $P_f(e)$ and the second shape maneuver is described by the matrix $P_s(f)$. As shown in the lab $P_f P_s = P_s P_f$, i.e., these two matrices commute. One can check this easily by noting that if you take two steps forward and then slide sideways two feet, you will be in the same position had you first slid sideways two feet and then taken two steps forward. In this example order doesn’t matter. It can be shown, as a result, that the non-reciprocal shape change sequence is equivalent to a reciprocal shape change sequence:

$$\begin{aligned}
M &= P_s(f = -l)P_f(e = -k)P_s(f = l)P_f(e = k) \\
&= P_f(e = -k)P_s(f = -l)P_s(f = l)P_f(e = k).
\end{aligned}$$

The second equality follows because of commutation of P_f and P_s , i.e., we can reverse the order of appearance of any two matrices without changing the result. The first expression takes the form of a non-reciprocal shape change sequence while the second expression takes the form of a reciprocal shape change sequence. As shown in the lab M is the transformation matrix that simply assigns $x' = x$ and $y' = y$.

The ability to compute exactly where and how far the system will be moved as a result of a non-reciprocal shape change sequence is useful not only for analysis of the effect of shape

change on motion, but also for prescribing shape change sequences to achieve a particular motion of interest. That is, we can use our results to determine control laws to move a system about as desired. Recall that prescribing control laws (e.g., for automatically controlling systems) was one of the original objectives outlined in Lecture Notes 5.A. In the case of parallel parking, for example, we now know that if we perform the non-reciprocal shape change sequence described by N above, with $\alpha = 10$ degrees and $k = 2$ meters, then the car will experience a net sideways translation equal to $\alpha k \pi / 180 = 0.35$ meters. So, if we want to automate a car to parallel park into a spot that is 3.5 meters away, we would program the car to perform this shape change sequence 10 times. α and k should be chosen to respect the space constraints on the car. For example, if the car can only move forward and backward 1 meter rather than 2 meters, we would choose $k = 1$ meter. In this case, the net sideways translation in one sequence would be only $0.35/2 = 0.175$ meters and to move sideways 3.5 meters, we would have to repeat the sequence 20 times.

Of course, the methodology that we have now just summarized is relevant to a whole class of motion control problems, most usefully for those that are less intuitive than the parallel parking problem. For example, we could apply these ideas to the problem of controlling the orientation of a spacecraft with only two wheels. We make the following definitions:

$$R(\alpha_1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_1 & \sin \alpha_1 \\ 0 & -\sin \alpha_1 & \cos \alpha_1 \end{pmatrix},$$

$$P(\alpha_2) = \begin{pmatrix} \cos \alpha_2 & 0 & -\sin \alpha_2 \\ 0 & 1 & 0 \\ \sin \alpha_2 & 0 & \cos \alpha_2 \end{pmatrix},$$

$$Y(\alpha_3) = \begin{pmatrix} \cos \alpha_3 & \sin \alpha_3 & 0 \\ -\sin \alpha_3 & \cos \alpha_3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

These are matrices that describe roll, pitch and yaw rotations. Note, however, that the sign is different in these matrices as compared to the matrices described at the end of Lecture Notes 5.B. This is because these matrices are to be applied not to the spacecraft, but to the “world” axes. This is analogous to what we did above in the parallel parking problem. That is, we want to fix a set of axes, call them X, Y, Z axes in the spacecraft, so that, for example, roll of the spacecraft is always rotation about the spacecraft’s X -axis, not the world’s x -axis, and similarly for pitch and yaw. So, if we apply $R(\alpha_1 = \beta)$, with $\beta > 0$, to the world axes (with respect to the axes fixed in the spacecraft), someone on the spacecraft would see the world rolling to the right by β degrees. To someone sitting on the world axes, the spacecraft looks like it is rolling to the left by β degrees about its X -axis. If we then apply $R(\alpha_2 = \gamma)$, with $\gamma > 0$, to an observer on the spacecraft, the world axes would look like they were pitching to the right by γ degrees, and to someone on the world axes, the spacecraft would look like it were pitching left by γ degrees about its Y -axis.

Given that we have defined matrices for our shape change maneuvers, and given that we know how to multiply these matrices, we can next check that the two available shape change

maneuvers (roll and pitch) do not commute.

$$\begin{aligned} R(\alpha_1)P(\alpha_2) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_1 & \sin \alpha_1 \\ 0 & -\sin \alpha_1 & \cos \alpha_1 \end{pmatrix} \begin{pmatrix} \cos \alpha_2 & 0 & -\sin \alpha_2 \\ 0 & 1 & 0 \\ \sin \alpha_2 & 0 & \cos \alpha_2 \end{pmatrix} \\ &= \begin{pmatrix} \cos \alpha_2 & 0 & -\sin \alpha_2 \\ \sin \alpha_1 & \cos \alpha_1 & \sin \alpha_1 \cos \alpha_2 \\ \cos \alpha_1 \sin \alpha_2 & -\sin \alpha_1 & \cos \alpha_1 \cos \alpha_2 \end{pmatrix}. \end{aligned}$$

$$\begin{aligned} P(\alpha_2)R(\alpha_1) &= \begin{pmatrix} \cos \alpha_2 & 0 & -\sin \alpha_2 \\ 0 & 1 & 0 \\ \sin \alpha_2 & 0 & \cos \alpha_2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_1 & \sin \alpha_1 \\ 0 & -\sin \alpha_1 & \cos \alpha_1 \end{pmatrix} \\ &= \begin{pmatrix} \cos \alpha_2 & \sin \alpha_1 \sin \alpha_2 & -\cos \alpha_1 \sin \alpha_2 \\ 0 & \cos \alpha_1 & \sin \alpha_1 \\ \sin \alpha_1 & -\sin \alpha_1 \cos \alpha_2 & \cos \alpha_1 \cos \alpha_2 \end{pmatrix}. \end{aligned}$$

So, $R(\alpha_1)P(\alpha_2) \neq P(\alpha_2)R(\alpha_1)$.

We can, therefore, proceed to see where we get from a non-reciprocal shape change sequence using α_1 and α_2 :

$$\begin{aligned} Q &= P(\alpha_2 = -\gamma)R(\alpha_1 = -\beta)P(\alpha_2 = \gamma)R(\alpha_1 = \beta) \\ &= \begin{pmatrix} \cos \gamma & 0 & \sin \gamma \\ 0 & 1 & 0 \\ -\sin \gamma & 0 & \cos \gamma \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & -\sin \beta \\ 0 & \sin \beta & \cos \beta \end{pmatrix} \begin{pmatrix} \cos \gamma & 0 & -\sin \gamma \\ 0 & 1 & 0 \\ \sin \gamma & 0 & \cos \gamma \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & \sin \beta \\ 0 & -\sin \beta & \cos \beta \end{pmatrix} \\ &\approx \begin{pmatrix} \cos(-\beta\gamma) & \sin(-\beta\gamma) & 0 \\ -\sin(-\beta\gamma) & \cos(-\beta\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} = Y(\alpha_3 = \beta\gamma). \end{aligned}$$

If β and γ are expressed as angles in degrees, replace the expression $\beta\gamma$ with $(\beta\gamma)(\pi/180)$ degrees. We have, then, shown that a non-reciprocal shape change sequence using two wheels produces a net rotation about the third axes, and we have provided a formula for how much net rotation the spacecraft will experience. As in the case of the car, we can use this result to prescribe control laws, i.e., we can program in advance how we want to use our wheels to make the spacecraft perform a desired orientation maneuver.

As suggested in the above example, the idea of shape change for motion control can be used to successfully provide motion (e.g., yaw of a spacecraft) when there has been a failure on board a system (e.g., the wheel on the spacecraft that would normally provide spin about the yaw axis has failed). This notion of using shape change to provide motion when compensating for a failure is quite powerful and can be applied to various other systems, for example, it can be used to move an underwater robotic vehicle around even in the face of a failure of one of its propellers.

The ideas developed in this module have been applied both to the study of biological systems and to the design of controllers for engineered (robotic) systems of all kinds. This

includes systems which have more than two shape variables (e.g., the snakeboard). Here are some relevant web links that you may enjoy investigating. Some of these include video of shape change for motion control in action!

<http://www.cc.gatech.edu/gvu/animation/Areas/humanMotion/humanMotion.html>

<http://robotics.eecs.berkeley.edu/~lara/net.html>

<http://robotics.eecs.berkeley.edu/~wents/movies.html>

<http://www.cds.caltech.edu/~marsden/movies/Krishnaprasad/rollerracer.mpg>

<http://www.laas.fr/~jpl/hiare-rem.html>

<http://www.laas.fr/~nic/uncert.html>

<http://www.soda.co.uk/soda/constructor/>