RoboJackets RoboNav

URC Manipulation - From an Electrical Viewpoint

Created at July 16, 2020

Last Edited at August 19, 2020

Contents

1 Introduction

As RoboNav moves away from IGVC and towards URC, understanding of the underlying principles for manipulation would be important for the electrical team. The study of manipulation has a long history in ECE and is arguably one of the most well-studied types of robot due to the ability to model the robot and the environment it operates in. Whether the task of planning (and / or) control of the manipulators eventually falls on the electrical team or not, a good grasp of the underlying principles of manipulation would always be a handy tool at hand for any unexpected happenings.

This guide borrowed much from the book "A Mathematical Introduction to Robotic Manipulation" by Dr. Richard Murray, Dr. Zexiang Li, and Dr. S. Shankar Sastry. *It is meant to provide a simplified introduction to manipulation that freshmen wouldn't need to spent more than a semester understanding the basis of manipulation.* We will start from rigid body motion, the right representation for rigid body motion and how it applies to the representation and planning of a manipulator.

2 Rigid Body Motions

The study of robot kinematics and controls has its core in the study of rigid body motions [1] , and we will attempt to approach rigid body motion using linear algebra and screw theory.

Michel Chases proved that a rigid body can be moved from any position to any other by a movement consisting of:

- A movement consisting of rotation about a straight line
- followed by translation parallel to that line.

One such motion is called a **screw motion**. The time derivative version of screw motion is called a **Twist**. Screw motion and twist play the central roles in the formulation of robot kinematics.

2.1 Rotational Motion in $\mathbb{R}^3$

2.1.1 Representation

Rotation Matrices

We begin the study by considering only the rotation aspect of rigid body motion. One of the most common method to describe the orientation of coordinate frame $\mathcal{B}$ relative to inertial frame $\mathcal{A}$ (as illustrated in Fig. 1) is to sequentially rotate about the z-axis of $\mathcal{B}$ by α, then y-axis of $\mathcal{B}$ by β, and finally along z-axis by γ. This yields a net rotation of $R(\alpha, \beta, \gamma)$ and α, γ, β are called the ZYZ Euler angles.

We use rotation matrix to represent a, well, rotation of coordinates. There are a number of reasons we want to use matrices to represent such transformation:

[1] think that the position of each joint on the manipulator is a movement away from the last joint

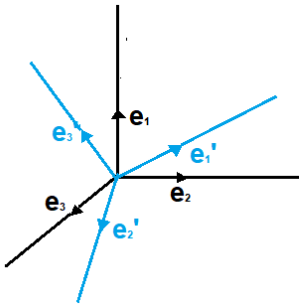

Figure 1: Example of Rotated Frames

- Modern computers are highly optimized for linear algebras.
- Rotation matrix is the cononical form for special orthogonal groups, which has proven properties that are useful to the representation, which we will briefly go over in section 2.1.2.

$$R_{\mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos & -\sin \\ 0 & \sin & \cos \end{bmatrix}, R_{\mathbf{y}} = \begin{bmatrix} \cos & 0 & \sin \\ 0 & 1 & 0 \\ -\sin & 0 & \cos \end{bmatrix}, R_{\mathbf{z}} = \begin{bmatrix} \cos & -\sin & 0 \\ \sin & \cos & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

A ZYZ Euler angle rotation would result in

$$R_{ab} = R_{\mathbf{z}}(\alpha) R_{\mathbf{y}}(\beta) R_{\mathbf{z}}(\gamma)$$

We use R_{ab} in the sense that it can transform coorindate points such that for point q currently attached on coordinate frame $\mathcal{A}$, and represented as $\mathbf{q}_a = [x, y, z]$. When coordinate frame $\mathcal{B}$ rotate to frame $\mathcal{B}$, point q's relative position in the frame $\mathcal{B}$ is still $\mathbf{q}_b = [x, y, z]$, since the point moved with the reference frame. However the current location of $\mathbf{q}_a = R_{ab}\mathbf{q}_b$

If point q rotates with frame $\mathcal{B}$ to a new frame $\mathcal{C}$, then

$$\mathbf{q}_a = R_{ab}\mathbf{q}_b = R_{ab}R_{bc}\mathbf{q}_c$$

There exists other types of euler angle parameterizations by using different ordered sets of rotation axes, including ZYX[2] and YZX. They avoided singularity at identity orientation, however do contain singularity at other orientations. We do not cover the details of singularity at this point, but more info at gimbal lock[3].

Quaternion

Quaternion works in a similar way that complex number works on the unit circle to represent planar rotations. They give a global parameterization of $SO(3)$ at the cost of

[2] roll pitch yaw

[3] `https://en.wikipedia.org/wiki/Gimbal_lock`

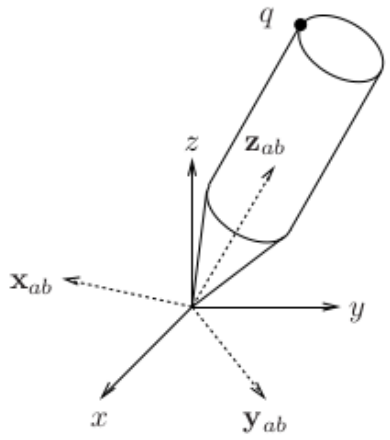

Figure 2: Rotated Reference Frame

using 4 numbers.

$$Q = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

w is the scalar component and $\mathbf{q} = (x, y, z)$ being vector component, a convenient shorthand notation being $Q = (w, \mathbf{q})$. Vector component satisfies the following relationship

$$\mathbf{i} \cdot \mathbf{i} = \mathbf{j} \cdot \mathbf{j} = \mathbf{k} \cdot \mathbf{k} = -1$$

$$\mathbf{i} \cdot \mathbf{j} = -\mathbf{j} \cdot \mathbf{i} = \mathbf{k}, \mathbf{j} \cdot \mathbf{k} = -\mathbf{k} \cdot \mathbf{j} = \mathbf{i}, \mathbf{k} \cdot \mathbf{i} = -\mathbf{i} \cdot \mathbf{k} = \mathbf{j}$$

The *conjugate* of a quaternion (recall imaginery numbers) $Q = (w, \mathbf{q})$ is $Q^* = (w, -\mathbf{q})$, and that the magnitude of a quaternion satisfies $||Q|| = Q \cdot Q^*$, and the inverse of a quaternion is

$$Q^{-1} = Q^*/||Q||^2$$

Unit quaternions are the subset of $Q \in \mathbb{Q}$ that $||Q|| = 1$, and typically are the only type of quaternion we operate on.

Similar to rotation matrices, if Q_{ab}, Q_{bc}, and Q_{ac} represents rotation between frame $\mathcal{AB}$, $\mathcal{BC}$, and $\mathcal{AC}$ respectively, the following equation holds.

$$Q_{ac} = Q_{ab} \cdot Q_{bc}$$

2.1.2 Understanding

Consider that we have a inertial frame $\mathcal{A}$, and a body frame $\mathcal{B}$ that has undergone a rotation about the origin point in inertial frame and no translation (Fig. 2). $\mathbf{x}_{ab}, \mathbf{y}_{ab}, \mathbf{z}_{ab}$ are the

coordinates of the principle axes of $\mathcal{B}$ in the inertial frame. Three vectors be vertical vector and the concatenation coordinate vectors obtain

$$R_{ab} = [\mathbf{x}_{ab}, \mathbf{y}_{ab}, \mathbf{z}_{ab}]$$

Since the column vectors of R_{ab} are from principle axes, dot product of the column vector with itself is 1, otherwise 0, yielding

$$R^T R = RR^T = 1 \Rightarrow \det R = \pm 1$$

Since $\det R = \mathbf{x}^T(\mathbf{y} \times \mathbf{z})$ and that $\mathbf{x} = \mathbf{y} \times \mathbf{z}$ under right-handed coordinate system, $\det R = 1$.

Therefore **right-handed coordinate frame are represented by orthogonal matrices with determinant 1**. The set of all matrices in $n \times n$ dimension are denoted by $SO(n)$, as special orthogonal. $SO(3) \in \mathbb{R}^{3\times 3}$ is a **group** under matrix multiplication. A set G, with a binary operation $\circ$ defined on elements of G is considered a group if it satisfies the following axioms:

- Closure: $g_1 \circ g_2 \in G$ if $g_1, g_2 \in G$
- Identity: exists e for every $g \in G$ that $g \circ e = e \circ g = g$
- Inverse: for each $g \in G$ there exists a unique inverse, $g^{-1} \in G$ that $g \circ g^{-1} = g^{-1} \circ g = e$
- Associativity.

In the case of $SO(3)$, it satiefies the above criteria in that

- $AB = C$ where $A, B \in \mathbb{R}^3$ gives $C \in \mathbb{R}^3$ by definition.
- $IR = RI = R$.
- $R^T R = RR^T = I$.
- Associativity from matrix multiplication.

Consider the scenario in Fig. 3 that $\omega \in \mathbb{R}^3$ be a unit vector specifying direction of rotation (rotation axis) and $\theta \in \mathbb{R}$ be the angle of rotation (Rad). The velocity of the point can be written as

$$\dot{q}(t) = \omega \times q(t) = \widehat{w} q(t)$$
$$w = [w_1, w_2, w_3] \Rightarrow \widehat{w} = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}$$

$\widehat{\omega}$ is a conversion for cross product to be represented as a typical matrix multiplication, and the equation is a time-invariant linear differential that can be integrated to:

$$\begin{aligned} q(t) &= e^{\widehat{\omega} t} q(0) \\ \therefore R(\omega, \theta) &= e^{\widehat{\omega}\theta} \end{aligned} \tag{2}$$

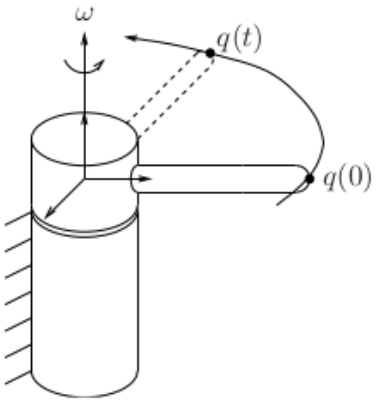

Figure 3: Showing rotation with exponential of ξ

$R\omega, \theta$ denotes rotation about axis ω at unit velocity for ω units of time. Expand $e^{\widehat{\omega}\theta}$ using taylor expansion:

$$e^{\widehat{\omega}\theta} = I + \widehat{\omega}\theta + \frac{(\widehat{\omega}\theta)^2}{2!} + \frac{(\widehat{\omega}\theta)^3}{3!} + ... \tag{3}$$

Since $\widehat{\omega}$ are skew-symmetric matrices, denote such matrices for $\mathbb{R}^3$ as $so(3), \forall S \in so(3) : S^T = -S$. Through this property, $\forall \widehat{a} \in so(3)$:

$$\widehat{a}^2 = aa^T - ||a||^2 I$$
$$\widehat{a}^3 = -||a||^2 \widehat{a}$$

Apply the above to equation 3:

$$e^{\widehat{\omega}\theta} = I + \left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - ...\right)\widehat{\omega} + \left(\frac{\theta^2}{2!} - \frac{\omega^4}{4!} + \frac{\omega^6}{6!} - ...\right)\widehat{\omega}^2$$

and hence

$$\boxed{e^{\widehat{\omega}\theta} = I + \widehat{\omega}\sin\theta + \widehat{\omega}^2(1 - \cos\theta)} \tag{4}$$

Commonly known as *Rodrigue's formula*, we may relate this back to the rotation matrix in equation 1 of section 2.1.1 that

$$R_{\mathbf{x}}(\theta) = e^{\widehat{\mathbf{x}}\theta} = I + \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \sin\theta + \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} (1 - \cos\theta)\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos & -\sin \\ 0 & \sin & \cos \end{bmatrix} \tag{5}$$

This similarly holds for rotation along other axis.

2.2 Rigid Motion in $\mathbb{R}^3$

2.2.1 Representation

As a screw motion requires a rotation and a translation component, we denote the pair of (p_{ab}, R_{ab}) as $SE(3)$ (special euclidean group) that $p \in \mathbb{R}^3, R \in SO(3)$. Similar to the case of rotation, for q_a, q_b be coordinate of q in frame A and B,

$$q_a = p_{ab} + R_{ab} q_b \tag{6}$$

We represent transformation in equation 6 in its homogenous form that

$$g = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \tag{7}$$

and if we append 1 to the coordinate of a point to yield the *homogeneous coordinates*

$$\bar{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ 1 \end{bmatrix} \tag{8}$$

$$\bar{q}_a = \begin{bmatrix} q_a \\ 1 \end{bmatrix} = \begin{bmatrix} p_{ab} + R_{ab} q_b \\ 1 \end{bmatrix} = \begin{bmatrix} R_{ab} & p_{ab} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q_b \\ 1 \end{bmatrix} = g_{ab} \bar{q}_b \tag{9}$$

Similar to rotation components, given g_{ab}, g_{ac}, g_{ac} can be given by the following equation

$$g_{ac} = g_{ab} g_{bc} = \begin{bmatrix} R_{ab} R_{bc} & R_{ab} p_{bc} + p_{ab} \\ 0 & 1 \end{bmatrix} \tag{10}$$

This relationship can be expanded to additional reference frames.

2.2.2 Understanding

We will first show that the set of rigid transformation is a group, and then link rigid motions and twists through exponential coordinates.

Similar to SO groups, this representation satisfies the following conditions:

- If $g_1, g_2 \in SE(3)$, then $g_1 g_2 \in SE(3)$
- The 4×4 identity matrix is a member of $SE(3)$
- For $\bar{g} \in SE(3)$, the inverse of $\bar{g}$ is simply its matrix inversion

$$\bar{g}^{-1} = \begin{bmatrix} R^T & -R^T p \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}^{-1}$$

- The composition rule is associative.

It is therefore a group.

Consider the scenario with the following joint that is translating and rotating at the same time, the velocity at the tip would be

$$\dot{p}(t) = \omega \times (p(t) - q)$$

we can rewrite this as

$$\begin{bmatrix} \dot{p} \\ 0 \end{bmatrix} = \begin{bmatrix} \widehat{\omega} & -\omega \times q \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} =$$

If we define ξ as body velocity, then

$$\widehat{\xi} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix}$$

therefore $\dot{p} = \widehat{\xi} p$ and differential equation solves to

$$p(t) = e^{\widehat{\xi} t} p(0)$$

which we may proceed similarly through taylor expansion as we did for $SO(3)$

2.3 Velocity of a Rigid Body

We will primarily show the process of calculating and transforming, velocity of particles in $SE(3)$ through the use of Twists in this section.

We will start from the rotation only situation. Consider point q is fixed in frame B that is rotating along the origin of frame A:

$$q_A = R(t) q_B$$

We may describe the velocity as following

$$v_{q_a} = \omega^s_{ab}(t) \times q_a$$
$$v_{q_b} = \omega^b_{ab}(t) \times q_b$$

Where ω^s, ω^b represents the spatial and body velocity between frame A and B, where we may obtain them as follows.

$$\widehat{\omega}^s_{ab} = \dot{R}_{ab} R^{-1}_{ab}$$
$$\widehat{\omega}^b_{ab} = R^{-1}_{ab} \dot{R}_{ab}$$

Now we consider the general case where the motion is defined by a rigid body transformation $g(t) \in SE(3)$.

$$\widehat{V}^s_{ab} = \dot{g}_{ab} g^{-1}_{ab}$$
$$\widehat{V}^b_{ab} = g^{-1}_{ab} \dot{g}_{ab}$$

By doing the vee (unhat) operation, we may get the individual components of spatial velocity

$$V_{ab}^s = \begin{bmatrix} v_{ab}^s \\ \omega_{ab}^s \end{bmatrix}$$

And we may get the velocity of point q in frame A and B by

$$v_{q_a} = \widehat{V}_{ab}^s q_a$$
$$v_{q_b} = \widehat{V}_{ab}^b q_b$$

Further note that since both V_{ab}^s and V_{ab}^b are twists, they are related by the following relationship

$$\widehat{V}_{ab}^s = \dot{g}_{ab} g_{ab}^{-1} = g_{ab} g_{ab}^{-1} \dot{g}_{ab} g_{ab}^{-1} = g_{ab} \widehat{V}_{ab}^b g_{ab}^{-1}$$

Alternatively,

$$V_{ab}^s = \begin{bmatrix} v_{ab}^s \\ \omega_{ab}^s \end{bmatrix} = \begin{bmatrix} R_{ab} & \widehat{p}_{ab} R_{ab} \\ 0 & R_{ab} \end{bmatrix} \begin{bmatrix} v_{ab}^b \\ \omega_{ab}^b \end{bmatrix} \Bigg\} \rightarrow V_{ab}^b$$

The transformation that transforms twists from one coordiante frame to another is called *Adjoint transformation*, for one associated with g, denote Ad_g. By one of the following methods

$$Ad_g(h) = ghg^{-1}$$

Ad_g is invertible, and is given by the following

$$Ad_{g^{-1}} = \begin{bmatrix} R^T & -R^T \widehat{p} \\ 0 & R^T \end{bmatrix}$$

Knowing the velocity sets us up for planning manipulators later in the section.

Velocity transforming in 2D Changing the coordinate frames in 2D is a lot different and simplified than in 3D. In general, we follow the following rules.

$$Ad_g(\xi) = g\xi g^{-1} \text{ where } \xi = \begin{bmatrix} R(\omega) & v \\ 0 & 1 \end{bmatrix}$$

2.4 Application

Use the MATLAB scripts to confirm you understanding of the above material. `coordinate_frame.m` is intended for understanding of SO2 and SE2 groups, and `velocity_spatial_body.m` is for the section on body and spatial velocity.

3 Manipulator Kinematics

In this section we will briefly talks about how to utilize the knowledge in Section 2 to simulate a manipulator. There are typically six types of manipulator joints:

- Revolute: pure rotation along a single axis
- Prismatic: pure linear motion along a single axis
- Helical: allows linear and rotation motion that are fixed to each other along the same single axis[4].
- Cylindrical: allows independent linear and rotation motion along the same single axis
- Spherical: allows arbitrary rotation, also refered to as *ball and socket* joints
- Planar: allows for arbitrary translation and rotation in the plane. This is rarely used and they are typically constructed from a revolute joint attached to two independent prismatic joints.

Modern manipulators are typically formulated by connecting different joints together using rigid links. The core of the formulation is the **product of exponential** formula, which represents the kinematics as the product of exponential of twists.

This approach works uniformly regardless of the type of the joints and provides a global and geometric representation of the kinematics of a manipulator which greatly simplfies the analysis and parameterization process of a manipulator. Compared to its alternative, *Denavit–Hartenberg parameters*, it further has the advantage of only needing to define two reference frame and easier geometric intepretation of screw axis at each joint.

3.1 Forward Kinematics

The forward kinematics of a manipulator determines the configuration of the end effector given the parameters of each joint in the manipulator. We attach two frames, the base frame S[5] and tool frame T. Given a set of angles $\theta \in Q$, the forward kinematics is a mapping $g_{st} := Q \to SE(3)$

Typically, to compute the configuration of the end-effector, we concatenate the rigid motion between frames. In the example below, the transformation can be represented as

$$g_{s,t}(\theta_1, \theta_2) = g_{s,l_1}(\theta_1)\, g_{l_1,l_2}(\theta_2) g_{l_2,T} \tag{11}$$

This can be extended to arbitrary number of joints on any manipulator and is a general formula for forward kinematics.

[4] `https://www.youtube.com/watch?v=fZqJopEL-k0`

[5] Not using B to avoid confusion with body frame

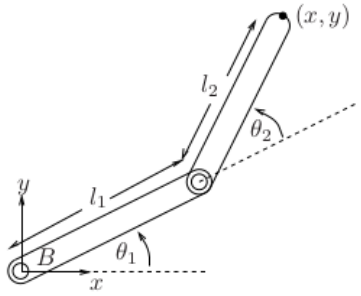

Figure 4: Example Revolute-Revolue Manipulator

3.2 Inverse Kinematics

3.2.1 Example IK Problem

Inverse kinematics looks at obtaining a feasible set of parameters for joints given a end-effector configuration. This is important since very often we receive manipulator problems with only start and goal configurations of the end effector. Inverse kinematics bridges the gap between the goal and our actuator. However note that inverse kinematics is a hard problem and general solution to inverse kinematics, especially given high number of joints and high DoF manipulators are active area of research and typically you won't be implementing one for yourself.

We here provide a simple 2D example of inverse kinematics of a RR manipulator (two revolute joints) in Fig 4. The end effector locaiton can be obtained by:

$$x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$
$$y = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

The inverse problem would be to given x, and y to solve for θs. Typically we approach the problem using polar coordinates and angles (see Fig 5), because trignometry. From the configuration visualized in above, we obtain θ_2 as following

$$\theta_2 = \pi \pm \alpha, \alpha = \cos^{-1}\left(\frac{l_1^2 + l_2^2 - r^2}{2 l_1 l_2}\right)$$

If $\alpha \neq 0$, then there are two configurations of the manipulator that would give the radius. The second is referreed to as the "flip solution". The value of θ_1 need to be solved for each possible value for θ_2, giving

$$\theta_1 = \arctan 2(y, x) \pm \beta, \beta = \cos^{-1}\left(\frac{r^2 + l_1^2 - l_2^2}{2 l_1 r}\right)$$

This example illustrates several important aspect of IK (inverse kinematics) problems.

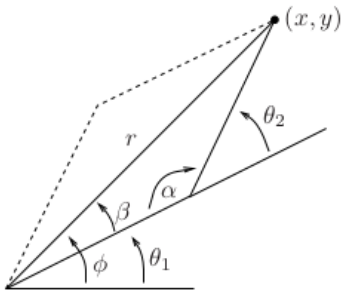

Figure 5: RR Manipulator in Polar Coordinates

1. One usually first divide the problem into specific subproblems.
2. Multiple solution occur when the desired configuration is within the workspace that maps to the same end-effector location.

There are typically two families of approach, closed-form and numeric solutions. Closed-form solutions are fast, analytical, and efficient calculators while numeric requires a back-and-forth process in determine the best solution configuration. Yet at the heart of both, they looks at the same set of equations that defines the inverse kinematics problem and solve them using geometric and algebraic identities.

3.2.2 Application

Use MATLAB scripts to confirm your understanding. `ik_rr.m` is an application of the above material.

3.3 Planning Manipulator Trajectory

We are near the end of this guide and we now consider the proper planning of manipulator via joint configuration given a end-effector trajectory.

3.3.1 Spline Trajectory

Spline trajectory fits the manipulator by considering the pose and twist of start and end location. A common approach to this technique involves fitting the value of manipulator along the curve of a spline and its first and second order derivatives trajectories. We will not be covering this and would instead focus on the Jacobian methods as Jacobians takes an important role in robotics in general.

3.3.2 Manipulator Jacobian

Manipulators often make use of the relationship between the joint and end-effector velocities. This has the significance that once we know desired path of the end-effector, we may obtain a differential of joint angles to be added onto initial joint configuration and therefore an actionable path for the manipulator.

$$V_{st}^s = J_{st}^s(\theta)\dot{\theta}$$

where,

$$J_{st}^s(\theta) = \begin{bmatrix} \left(\frac{\partial g_{st}}{\partial \theta_1}\right)^\vee & \cdots \left(\frac{\partial g_{st}}{\partial \theta_n}\right)^\vee \end{bmatrix}$$

J_{st}^s is called a spatial manipulator jacobian that at each configuration θ it maps the joint velocity vector into the corresponding velocity of the end-effector. Jacobian in general shows the relationship between two variables in how the change of one affect the other.

This relationship between joint velocity and end-effector velocity can be used to move end-effector from one configuration to another, without calculating the inverse kinematics for the manipulator at all configuration. If J_{st} is invertible, then

$$\dot{\theta}(t) = (J_{st}^s(\theta))^{-1} V_{st}^s(t)$$

And we would be able to do integration over θ to obtain a trajectory for θ given a known current end-effector location g_{st} and known current joint configuration θ

3.3.3 Application

Use MATLAB scripts to confirm your understanding. Use `jacobian_planning.m` to perform a simple planning of manipulator. `theta_dot.m` is required for performing integration and you would need to fill that out as well.

A Reference Output FIgures for MATLAB practice

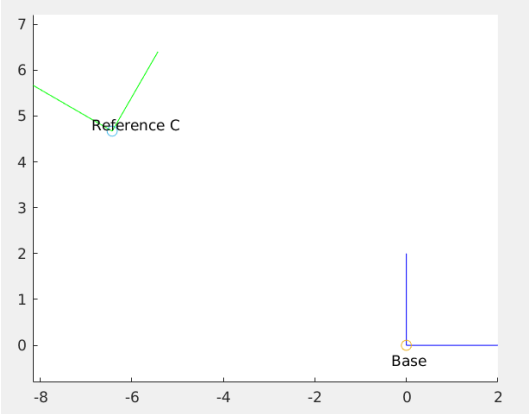

Figure 6: First Figure for `coordinate_frame.m`

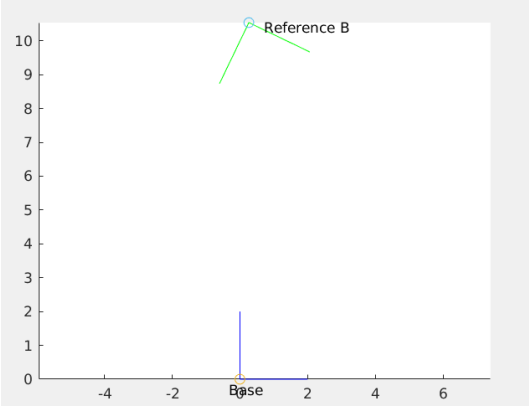

Figure 7: Second Figure for `coordinate_frame.m`

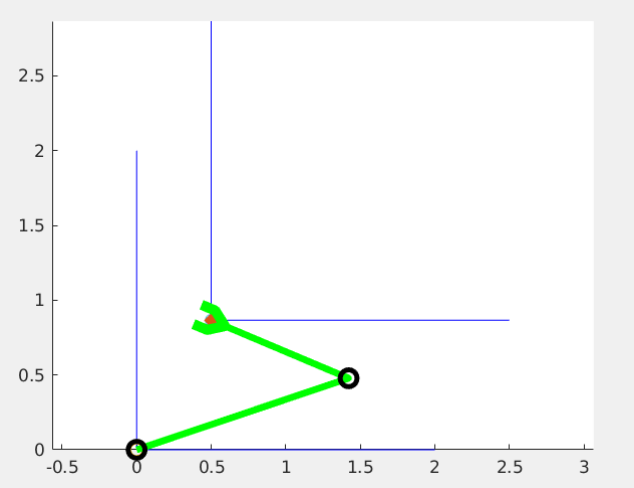

Figure 8: First Figure for `ik_rr.m`

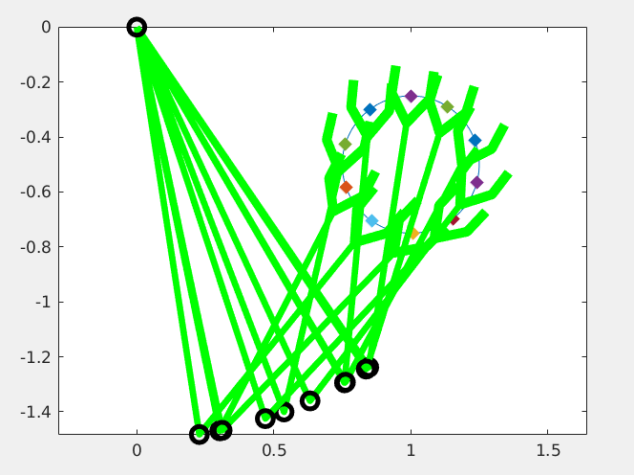

Figure 9: Second Figure for `ik_rr.m`

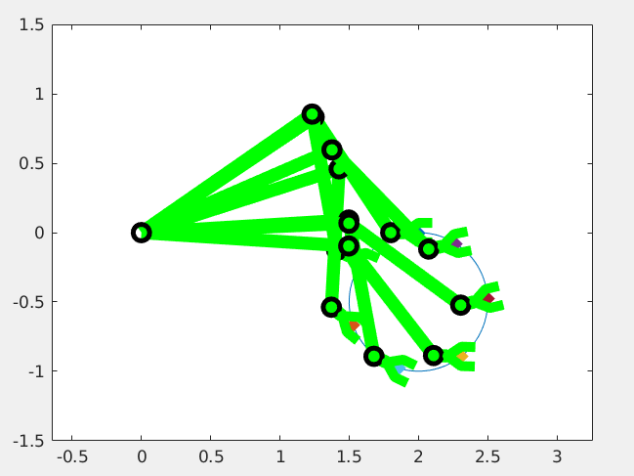

Figure 10: First Figure for `jacobian_planning.m`

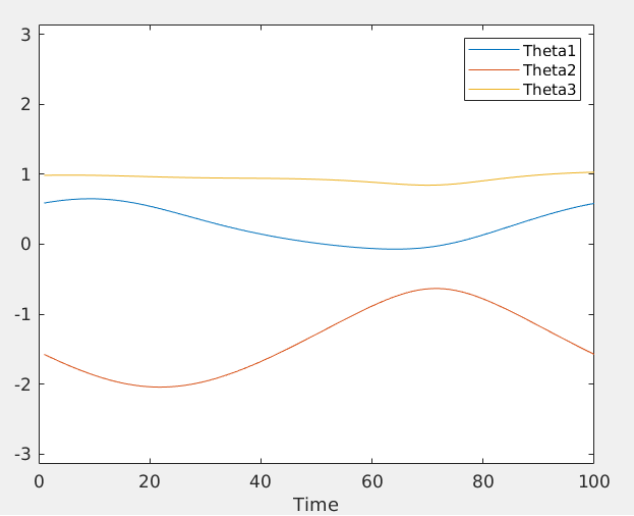

Figure 11: Second Figure for `jacobian_planning.m`