# EE106A: Block 2 Supplement - Python / NumPy / SciPy Introduction*

September 12, 2016

---

## Goals

By the end of this supplement you should have:

- a working knowledge of Python
- gone through the ROS tutorials
- written kinematic functions in Python using NumPy / SciPy

---

## Contents



# 1 Motivation

To help you through the labs and projects, we are releasing this supplement to help you be more comfortable with Python and ROS and to prepare you for the Baxter labs. Without a working knowledge of Python / ROS, the lab assignments will only become more difficult. Note that while Sections 2 and 3 are optional, we highly recommend Section 2 if you do not have a working knowledge of Python and recommend (in general) Section 3 to give you a better working knowledge of ROS. Section 4 will need to be completed during Lab 3, though you can start working on it beforehand to improve your familiarity with array operations in Python.

# 2 Python (optional)

This section is for those who are unfamiliar with Python; if you are comfortable with Python, feel free to skip to the next section. As Python is the primary programming language of the lab and projects, you will need a working knowledge to be able to complete the labs in a reasonable amount of time. The great news is that Python is a very easy language to pick up if you already know how to program in C, C++, Java or even MATLAB. Here are some references that might be useful to you depending on your specific needs.

- `https://developers.google.com/edu/python`
- `http://learnpython.org/en/Welcome`
- `https://www.codecademy.com/en/tracks/python`

*Developed by Victor Shia and Jaime Fisac, Fall 2015

For those of you who are comfortable with MATLAB and want to carry your skills over to Python, look to these resources:

- `http://bastibe.de/2013-01-20-a-python-primer-for-matlab-users.html`
- `http://wiki.scipy.org/NumPy_for_Matlab_Users`

# 3 ROS Tutorials (optional)

On top of Python, ROS is the engine and essential framework of this robotics course. While Labs 1 and 2 go through the basics of ROS, a much more complete tutorial is available at: `http://wiki.ros.org/ROS/Tutorials`. We recommend going through the Beginner Level tutorials 2, 3, 4, 5, 6, 7, 10, 12, 13, 15, and 16.

FEAR NOT — You have already gone through much of the material in Labs 1 and 2. These tutorials will give you a deeper understanding of why things work the way they do. This should not take longer than 2 hours now (since you've seen a lot of it already) and going through this will help you greatly with the future labs. You can either use the provided VM or the lab computers to do this (outside of normal lab hours).

# 4 Kinematic Functions using NumPy and SciPy (to be completed for Lab 3)

Write Python functions to implement the following formulas from the book using the NumPy and SciPy libraries; you might find the *NumPy / SciPy Notes* posted in bCourses under `Labs > Resources` useful. You should use the `kin_func_skeleton.py` file included in `lab3_resources.zip` as a starting point. Keep in mind that as you implement functions, you can call them to simplify your implementation of subsequent functions. When you think your code is correct, you can test it by running `python kin_func_skeleton.py` at the command line.

- The `(^)` operator for rotation axes in 3D
  - Input: $3 \times 1$ vector, $\omega = [\omega_x, \omega_y, \omega_z]^T$.
  - Output: $3 \times 3$ matrix,
$$\hat{\omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$
- Rotation matrix in 2D as a function of $\theta$
  - Input: Scalar, $\theta$.
  - Output: $2 \times 2$ matrix,
$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$
- Rotation matrix in 3D as a function of $\omega$ and $\theta$
  - Input: $3 \times 1$ vector, $\omega = [\omega_x, \omega_y, \omega_z]^T$ and scalar, $\theta$.
  - Output: $3 \times 3$ matrix,
$$R(\omega, \theta) = e^{\hat{\omega}\theta} = I + \frac{\hat{\omega}}{\|\omega\|}\sin(\|\omega\|\theta) + \frac{\hat{\omega}^2}{\|\omega\|^2}(1 - \cos(\|\omega\|\theta))$$

  (This is the form for the Rodrigues formula when $\|\omega\| \neq 1$.)
- The `(^)` operator for twists in 2D
  - Input: $3 \times 1$ vector, $\xi = [v_x, v_y, \omega]^T$ (Note that $\omega$ is a scalar).
  - Output: $3 \times 3$ matrix,
$$\hat{\xi} = \begin{bmatrix} 0 & -\omega & v_x \\ \omega & 0 & v_y \\ 0 & 0 & 0 \end{bmatrix}$$

- The `(^)` operator for twists in 3D
  - Input: $6 \times 1$ vector, $\xi = [v^T, w^T]^T = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^T$.
  - Output: $4 \times 4$ matrix,
$$\hat{\xi} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\omega_z & \omega_y & v_x \\ \omega_z & 0 & -\omega_x & v_y \\ -\omega_y & \omega_x & 0 & v_z \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
- Homogeneous transformation in 2D
  - Input: $3 \times 1$ vector, $\xi = [v_x, v_y, \omega]^T$ and scalar $\theta$.
  - Output: $3 \times 3$ matrix,
$$g(\xi, \theta) = e^{\hat{\xi}\theta} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$$
    where
$$R = \begin{bmatrix} \cos(\omega\theta) & -\sin(\omega\theta) \\ \sin(\omega\theta) & \cos(\omega\theta) \end{bmatrix}$$
$$p = \begin{bmatrix} 1 - \cos(\omega\theta) & \sin(\omega\theta) \\ -\sin(\omega\theta) & 1 - \cos(\omega\theta) \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_x/w \\ v_y/w \end{bmatrix}$$
- Homogeneous transformation in 3D
  - Input: $6 \times 1$ vector, $\xi = [v^T, w^T]^T = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^T$ and scalar $\theta$.
  - Output: $4 \times 4$ matrix,
$$g(\xi, \theta) = e^{\hat{\xi}\theta} = \begin{bmatrix} e^{\hat{\omega}\theta} & \frac{1}{\|w\|^2}\left((I - e^{\hat{\omega}\theta})(\hat{\omega}v) + \omega\omega^T v\theta\right) \\ 0 & 1 \end{bmatrix}$$
- Product of exponentials in 3D
  - Input: $n$ $6 \times 1$ vectors, $\xi_1, \xi_2, \ldots, \xi_n$ and scalars, $\theta_1, \theta_2, \ldots, \theta_n$.
  - Output:
$$g(\xi_1, \theta_1, \xi_2, \theta_2, \ldots, \xi_n, \theta_n) = e^{\hat{\xi}_1\theta_1} e^{\hat{\xi}_2\theta_2} \ldots e^{\hat{\xi}_n\theta_n}$$