# Error Propagation for a UR5 Robotic Manipulator

530.645 Kinematics Final Project Can Kocabalkanli

## Abstract

In this project, two error propagation formulas (from [1]) were recursively implemented in Mathematica and were evaluated at different magnitudes of error in the context of the UR5 robot arm. Their accuracies were assessed by numerically comparing the propagated error at the end effector resulting from errors from each individual joint of the UR5, and comparing these propagations to the actual error observed.

## Introduction

### UR5 Manipulator

The UR5 is a flexible collaborative robot arm manipulator built by Universal Robots. It features six joints that can each rotate through ± 360° [2]. It is commonly used for lightweight industrial and commercial applications and for educational purposes. Students like myself are expected to use the UR5 in some of their robotics classes, which makes error propagation for this manipulator a personally relevant and useful kinematics application.

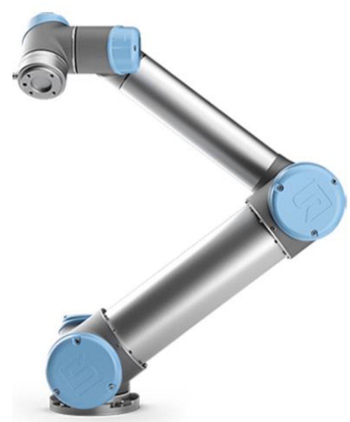

*Figure 1: UR5 Manipulator [2]*

### Rigid Body Motions in SE(3)

The motion of the UR5 is conveyed by the Euclidean motion group SE(3), represented with a 4 x 4 homogeneous transformation matrix $g = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$ where R is an SO(3) rotation matrix and t is a translation in three dimensions.

### Adjoint Operators in SE(3) (Ad and ad)

The error propagation formulae explored in this project make use of the adjoint operators. The Adjoint Operator Ad(g) for an SE(3) transformation matrix g is Ad(g) = $\begin{pmatrix} R & 0_3 \\ TR & R \end{pmatrix}$ , where T = $t^M$, the matrix representation of the translation t. Meanwhile, if we define $X = \log(g) = \begin{pmatrix} \Omega & v \\ 0^T & 0 \end{pmatrix}$, the adjoint ad(X) for X is ad(X) = $\begin{pmatrix} \Omega & 0 \\ V & \Omega \end{pmatrix}$, where V = $v^M$.

### Basis Elements

Basis elements of the Lie Algebra corresponding to SE(3) are used in the relevant error propagation formulae, and therefore are defined here:

$$E_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} E_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} E_3 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$E_4 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad E_5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad E_6 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

## Forward Kinematics of UR5

The Denavit-Hartenberg (D-H) convention can be used to attach reference frames to the six links of the UR5 and construct the SE(3) transformation matrices from thereon. The reference frames for each link are attached as seen in Figure 2. The D-H parameters of these frames are presented under Table 1.

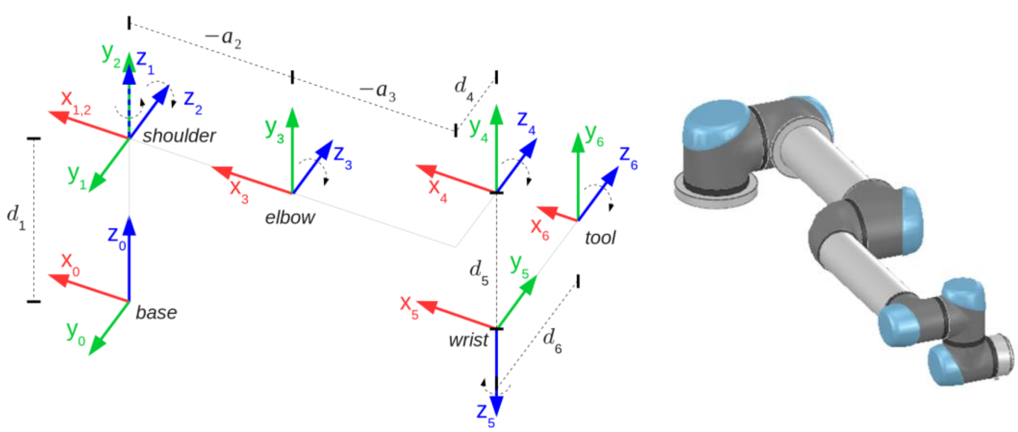


*Figure 2: D-H Frames for each link and visualization of the UR5 at this configuration* [3]

***Table 1: D-H Parameters of UR5*** **[4]**

| Joint (i) | ϑ | α | a | d |
| --- | --- | --- | --- | --- |
| 1 | Θ1 | π/2 | 0 | 0.08916 |
| 2 | Θ2 | 0 | -0.425 | 0 |
| 3 | Θ3 | 0 | -0.39225 | 0 |
| 4 | Θ4 | π/2 | 0 | 0.10915 |
| 5 | Θ5 | -π/2 | 0 | 0.09465 |
| 6 | Θ6 | 0 | 0 | 0.823 |

The parameters are defined as:

$\Theta_i$ : Angle between $X_{i-1}$ and $X_i$ about $Z_i$

$\alpha_i$ : Angle between $Z_i$ and $Z_{i+1}$ about $X_i$

$d_i$ : Distance between $X_{i-1}$ and $X_i$ along $Z_i$

$a_i$ : Distance between $Z_i$ and $Z_{i+1}$ along $X_i$

After setting each reference frame at the joints, a general expression for the transformation matrix between each consecutive frame (i-1 and i) can be written. Then, the transformation between the robot's base and any joint i can be written as ${}^{0}_{i}T = {}^{0}_{1}T\,{}^{1}_{2}T \ldots {}^{i-1}_{i}T$ (1). This general expression looks like this [3]:

$$
{}^{i\text{-}1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i\cos(\alpha_{i-1}) & \cos\theta_i\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -\sin(\alpha_{i-1})d_i \\ \sin\theta_i\sin(\alpha_{i-1}) & \cos\theta_i\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & \cos(\alpha_{i-1})d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}
$$

# Error Propagation in Serial Linkages

Intuitively, if two serial joints independently display error, these two error distributions would stack to cause greater error at the end effector of the serial linkage. Given the two error probability density functions, the error distribution at the end effector is their convolution [1].

The mean μ of a sample distribution is defined as (3) [5]:

$$
\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i \tag{3}
$$

While the covariance of the distribution about this mean is given by (4) [1]:

$$
\Sigma = \frac{1}{N}\sum_{i=1}^{N} log\left(\mu^{-1}\circ g_{ee}^{i}\right) log\left(\mu^{-1}\circ g_{ee}^{i}\right)^{T} \tag{4}
$$

There are two error propagation formulas given by Chirikjian to approximate the propagation of the means and covariances of the PDFs of each linkages along the series that result from the first and second order expansions of the probability density function using the Baker-Campbell-Hausdorff Formula [1].

The first order propagation formula is defined as (5) [6]:

$$\Sigma_{1*2} = \mathrm{Ad}(g_2^{-1})\Sigma_{1*2}Ad^T(g_2^{-1}) + \Sigma_2 \tag{5}$$

The second order propagation formula is defined as (6) [6]:

$$\Sigma_{1*2} = A + B + F(A,B) \tag{6}$$

$$A = \mathrm{Ad}(g_2^{-1})\Sigma_{1*2}Ad^T(g_2^{-1}), \quad B = \Sigma_2 \tag{7}$$

$$F(A,B) = \frac{1}{4}\sum_{i,j=1}^{d} ad(E_i)\, B\, ad^T(E_j)A_{ij} + \frac{1}{12}\left\{\left[\sum_{i,j=1}^{d} A_{ij}''\right]B + B^T\left[\sum_{i,j=1}^{d} A_{ij}''\right]^T\right\} + \frac{1}{12}\left\{\left[\sum_{i,j=1}^{d} B_{ij}''\right]A + A^T\left[\sum_{i,j=1}^{d} B_{ij}''\right]^T\right\} \tag{8}$$

$$A_{ij}'' = ad(E_i)ad(E_j)A_{ij}, \quad B_{ij}'' = ad(E_i)ad(E_j)B_{ij}, \tag{9}$$

Where Ad(g) and ad(X) are the adjoint operators and $E_i$ are the basis elements described earlier.

These particular formulae concern the stacking of two links. The formulae can be used recursively for manipulators with more links such as the UR5. In the case of the first order formula for instance, the covariance for the first three links of the serial manipulator would be $\Sigma_{1*2*3} = \mathrm{Ad}(g_3^{-1})\Sigma_{1*2}Ad^T(g_3^{-1}) + \Sigma_3$, and the covariance for the end effector can be recursively calculated as $\Sigma_{1*2*3*4*5*6} = \mathrm{Ad}(g_6^{-1})\Sigma_{1*2*3*4*5}Ad^T(g_6^{-1}) + \Sigma_6$. The second order formula is used similarly in a recursive manner.

# Numerical Error Propagation

This project aims to evaluate the accuracy of each of these formulae at different distributions of error. The kinematic errors are introduced through the joint angles of the UR5; each angle is deviated from its ideal value by absolute error values of $\pm\varepsilon$, so that each joint angle is sampled at values of $\theta - \varepsilon$, $\theta$, $\theta + \varepsilon$ as done in [1].

Such an introduction of error generated $N = 3^6$ frames of references $\{g^i_{ee}\}$ for the end effector clustered around $g_{ee}$, the ideal reference frame that denotes the position and orientation of the end effector relative to the manipulator base.

For the first order theory, since the reference frames $\{g^i_{ee}\}$ are assumed to be clustered around $g_{ee}$ very tightly, the mean $\mu_{ee}$ is assumed to be equal to $g_{ee}$. However, this assumption is not precise enough for the second order approximation theory, and the mean of the reference frames might not be equal to the ideal position of the end effector. Therefore, we need to update the mean reference frame $\mu_{ee}$ like this [1]:

$$\mu_{ee} = g_{ee} \circ exp\left[\frac{1}{N}\sum_{i=1}^{N} \log(g_{ee}^{-1} \circ g_{ee}^{i})\right] \tag{10}$$

To evaluate the accuracy of each error approximation method, for different values of ε the covariance matrix Σ for the end effector is calculated using the first order approximation, the second order approximation, and by using brute force enumeration that serves as the ground truth that each approximation is compared to. Brute force covariance is calculated using equation (4), which is rewritten here as [1]:

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x_i}\boldsymbol{x_i^T} \tag{11}$$

Where $\boldsymbol{x_i} = \left[log\left(\mu^{-1} \circ g_{ee}^{i}\right)\right]^{v}$, v denoting the vector notation of the resulting matrix inside the brackets and N = $3^6$ for the end effector and N = 3 for individual joints.

Each of these methods are implemented in a Mathematica script, where the user specifies the error ε and the desired joint angle values. First, the forward kinematics of the UR5 manipulator is numerically evaluated between each consecutive joint for the angles of values of θ - ε, θ, θ + ε. Using the forward kinematics between individual consecutive joints, a covariance matrix Σ is calculated for each joint using Equation (11). Since the distribution is symmetric around the angle θ, the mean for each joint is equal to the matrix g evaluated at θ for each joint.

Then the transformations between the links for each distribution are multiplied with each other as seen in Equation (1) to yield the $3^6$ different reference frames for the end effector. The mean and covariance of the end effector distribution is calculated using Equation (11) as the ground truth. For the second order approximation, the mean is updated once using Equation (10), and the covariance of the end effector is recalculated using this mean with Equation (11). On a side note, the updated mean reference frames were found to be the same as the old mean reference frames for individual joint angles and hence need not to be updated in covariance calculations for the second order approximation.

Using the individual covariance matrices at each joint previously calculated, Equations (6) and (7) are used to evaluate the first and second order approximations respectively. Finally, the script evaluates the deviation between the approximations and the actual brute force covariance for the end effector calculated using the Frobenuis norm like the following:

$$deviation = \frac{\left\|\Sigma_{approx} - \Sigma_{actual}\right\|}{\left\|\Sigma_{actual}\right\|} \tag{12}$$

Where the brackets denote the Frobenius norm for the matrices inside them, $\|A\|_F = \sqrt{Tr(AA^H)}$ (13) where $A^H$ is the conjugate transpose of A.

The calculated deviation is a good measure of accuracy for the error propagation methods that we explore here. The figure below presents the deviation of the first order and second order approximations for different values of error, ε.

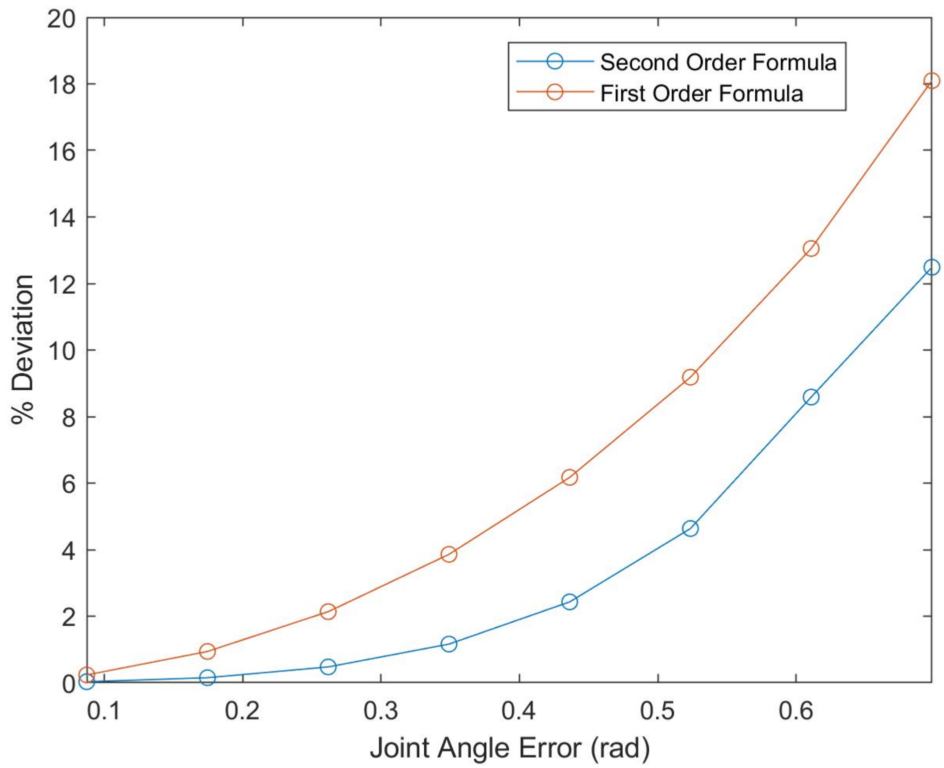


*Figure 3: %Deviation of First and Second Order Error Approximation vs Joint Angle Error in Radians*

This figure reveals that both approximation methods feature less than 5% deviation for about 0.35 radians of ε deviation in each of the joint angles, and that the second order method remains below 5% deviation for up to more than 0.5 radians of error. These results are in accordance with the results in [1], and by extending the range of error to more than 0.6 radians they provide a more comprehensive picture of how the exponential increase in deviation behaves at such large amounts of error.

Since even 0.35 radians is an unreasonably large amount of error for a decently manufactured robot's joints, both methods are quite viable for practical error propagation applications in the context of the UR5 and beyond. However, the deviation of the second order method stays almost negligible up until very large joint angle errors, and grows much more slowly compared to the first order approximation, making it the definite choice for applications where exactness and very high accuracy are needed.

## Conclusion

In this project a Mathematica Script was developed to implement the forward kinematics and error propagation for a UR5 robotic manipulator. The Denavit-Hartenberg method was used to represent and calculate the forward kinematics for each individual link and the end effector of the UR5. The first and second order error propagation formulas were both implemented and then compared to the actual error through the covariance matrix of the set of reference frames constituting the error distribution sample calculated by each method. Both propagation formulae provide accurate approximations for small amounts of error, about until 0.35 radians, and although afterwards the deviation from actual error rises exponentially for both methods, this rise is much slower for the second order method which remains very accurate up until about 0.5 radians of error. Overall, the second order method is a significant improvement over the first order method in terms of accuracy and robustness for any error, especially for larger errors. It successfully propagates error through the UR5 serial manipulator.

## References

[1] Chirikjian, Gregory S., and Yufeng Wang. *"Nonparametric Second-Order Theory of Error Propagation on Motion Groups."* The International Journal of Robotics Research, no. 27, 2008, p. 1258., DOI:10.1177/0278364908097583 .

[2] "Universal Robot UR5." *Applications for Collaborative Robotic Arms | Universal Robots*, 8 Jan. 2019, www.universal-robots.com/products/ur5-robot/.

[3] Andersen, Rasmus Skovgaard. *Kinematics of a UR5*. Aalborg University, 31 May 2018, rasmusan.blog.aau.dk/files/ur5_kinematics.pdf.

[4] "UR5." *UR5 - Rsewiki*, 8 Jan. 2019, rsewiki.elektro.dtu.dk/index.php/UR5.

[5] Anderson, T. W. *An Introduction to Multivariate Statistical Analysis*, 3rd edn, 2005, New York, Wiley.

[6] Li, X., Chirikjian, Gregory S., Book Chapter in: Riemannian Computing in Computer Vision, Editors: Pavan K. Turaga, Anuj Srivastava, pp. 165-186, Springer, 2016.