Input-Output Feedback Linearization for Euler Angle Parameterized Attitude Control

Gabriel Colangelo
Purdue University
AAE 666 Final Paper

4/26/2024

Contents

Introduction

Feedback linearization is a common approach used in the control of non-linear systems, as it provides a way to directly handle the known non-linearities in a system. This paper looks at a specific type of feedback linearization called input-output feedback linearization, which is known as Dynamic Inversion in the aerospace industry [1]. Dynamic Inversion as a control technique has been applied in control design for quadrotors [2] and in the X-62 Vista aircraft [3]. A major advantage of input-output feedback linearization is that it allows for the prescription of linear closed loop dynamics [4]. This linearity allows the use of linear control techniques such as pole placement & LQR to stabilize the closed loop error dynamics.

Input-Output Feedback Linearization

Consider a n^{th} order non-linear system, that is linear/affine in its control, u, and that is also square such that the number of control inputs is equal to the number of system outputs. [5,1]. Such a system would take the form of

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{1}$$

For a tracking control problem, it is desired that the system output, y, follow a desired reference trajectory, r. A satisfactory tracking control law can be designed using input-output feedback linearization if a direct relationship between the output and input can be generated. To do this, the output is differentiated until the control input appears in the obtained derivative [5,1]. The differentiation is performed using Lie derivatives. The notation for Lie derivatives is as follows. Given a vector field with its Jacobian ∇f, and a scalar function $h(x)$, the Lie derivative of h with respect to f is $L_f h$, which is equivalent to $\nabla h\, f$. Applying this to the nonlinear system given by (1), the first Lie derivatives are

$$\dot{y} = \nabla h\,(f(x) + g(x)u) = L_f h(x) + L_g h(x)u \tag{2}$$

If $L_g h(x)$ is non-singular, then a relationship between the output and the input has been found. If $L_g h(x)$ is singular, then a second differentiation must be performed. This second differentiation is given by

$$\ddot{y} = \nabla(L_f h(x))\,(f(x) + g(x)u) = L_f^2 h(x) + L_g L_f h(x)u \tag{3}$$

Again, if $L_g L_f h(x)$ is non-singular, then a relationship between the output and the input has been found. If $L_g L_f h(x)$ is singular, then another differentiation must be performed. This continues r_d times, until $L_g L_f^{r_d-1} h(x)$ is invertible. The number of differentiations that are performed until the input appears is known as the relative degree of the system, r_d [5]. Once a relationship between the output and input has been obtained, the differentiated output has the following form

$$y^{(r_d)} = L_f^{r_d} h(x) + L_g L_f^{r_d-1} h(x) u \tag{4}$$

If the system output is to follow a desired trajectory, r, the error between this desired trajectory and the output is defined as $e = r - y$. Then the r_d^{th} derivative of the error is given by $e^{(r_d)} = r^{(r_d)} - y^{(r_d)}$. We now define the control law, u, which has the following form, where v is a to be determined auxiliary input [5, 1].

$$u = (L_g L_f^{r_d-1} h(x))^{-1}(-L_f^{r_d} h(x) + r^{(r_d)} + v) \tag{5}$$

$r^{(r_d)}$ is known as a feed-forward term, and in the case of $\dot{r}$ or $\ddot{r}$, it is referred to as a velocity feed-forward or an acceleration feed-forward respectively. Applying the control law (5) to (4) yields the following closed loop dynamics:

$$y^{(r_d)} = r^{(r_d)} + v \tag{6}$$

Using the previously defined error definition and performing simple manipulations, the closed loop error dynamics are as follows

$$e^{(r_d)} = -v \tag{7}$$

Now, v can be chosen such that the closed loop error dynamics are prescribed to be linear. With the obtained linear error dynamics, linear system design techniques such as LQR and pole placement can be used to stabilize the error. However, the stabilization of the error dynamics does not guarantee stability of the full closed loop system as the input-output linearization renders states not related to the output to be "unobservable" [5]. To analyze the stability of the full closed loop system, one must look at zero-dynamics of the nonlinear system. This is done by selecting the control input such that the system output is zero, thus also making the time derivatives of the output also zero ($\dot{y} = \ddot{y} = 0 \therefore v = -r^{(r_d)}$) [5]. For the system output to be zero, the control input must then be given by

$$u_0 = (L_g L_f^{r_d-1} h(x))^{-1}(-L_f^{r_d} h(x)) \tag{8}$$

Applying this control law to the nonlinear original system yields the systems zero dynamics

$$\dot{x}_0 = f(x) - g(x)(L_g L_f^{r_d-1} h(x))^{-1}(L_f^{r_d} h(x)) \tag{9}$$

The stability of the above zero dynamics can then be analyzed using techniques such as Lyapunov's direct and indirect methods. For the stabilization problem, if the zero-dynamics are asymptotically stable, the control law derived from the input-output feedback linearization can be shown to make the closed loop system locally asymptotically stable [5]. For the tracking control problem, if the zero-dynamics are asymptotically stable, the system is BIBO stable (the states remain bounded) and the tracking error converges to zero exponentially [5]. However, if $r_d < n$ and the zero-dynamics are unstable, then dynamic inversion cannot be used to stabilize the system with the current chosen output. One may choose a new set of output variables and repeat the analysis to try and stabilize the system using dynamic inversion. It should be noted, that if the relative degree of the system is equal to the system order, then there are no zero-dynamics as the system has been linearized via a form of input-state feedback linearization [5]. Therefore, it is only necessary to check the zero-dynamics of a nonlinear system if the order of the error dynamics (relative degree) is less than the system order.

It is appropriate to point out the potential issues with Dynamic Inversion as a control scheme. First, it is clear from the derived control law that full state feedback is necessary, which may be overcome with sufficient sensors or state estimators such as an extended Kalman filter [7]. There is also the issue of choosing appropriate outputs/control variables, as they may lead to unstable zero dynamics. The control law also requires an exact model of the plant dynamics, which may not always be available or if it is, the model may contain modeling errors. Thus dynamic inversion is not guaranteed to be robust to unmodeled dynamics or parameter uncertainty. Adaptive control has been used to help overcome the issues of parameter uncertaintiy, see ref [4] for more information.

ATTITUDE TRACKING CONTROL

Dynamic Inversion/Input-Output feedback linearization will now be applied to the attitude tracking control problem. The attitude will be parameterized using the 3-2-1 Euler Angle set, where yaw, pitch, and roll are denoted by ψ, θ, ϕ respectively. The plant dynamics are given by the following equations [6], where the body angular velocity vector (relative to the inertial frame) is denoted by $\omega_{B/I}^{B}$, u is the control input, $\mathbf{C}$ is the matrix containing the attitude kinematics for the 3-2-1 Euler Angle set (see appendix), and $\mathbf{I}$ is the moment of inertia tensor (which is assumed to be invertible, a good assumption. See appendix for the form of the inertia tensor)

$$\begin{aligned} \alpha &= [\phi, \theta, \psi]^T = y \\ \dot{\alpha} &= \mathbf{C}\omega_{B/I}^{B} \\ \dot{\omega}_{B/I}^{B} &= \mathbf{I}^{-1}[u - (\omega_{B/I}^{B} \times (\mathbf{I}\omega_{B/I}^{B}))] \end{aligned} \tag{10}$$

The outputs for this system are the Euler Angles themselves, which in a real life application maybe attained via an INS package or from a real-time estimate via an EKF. We can reorganize the system into a form of $\dot{x} = f(x) + g(x)u$.

$$\begin{aligned} x &= \begin{bmatrix} \alpha \\ \omega_{B/I}^{B} \end{bmatrix} \\ \dot{x} &= \begin{bmatrix} \mathbf{C}\omega_{B/I}^{B} \\ -\mathbf{I}^{-1}[(\omega_{B/I}^{B} \times (\mathbf{I}\omega_{B/I}^{B})) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I}^{-1} \end{bmatrix} u \end{aligned} \tag{11}$$

With the system in the same form as (1), we now begin the control law derivation by taking the first derivative of the output, where the following is obtained

$$\begin{aligned}\dot{y} &= \nabla h(f+gu) = [I_{3x3}\, 0_{3x3}] \begin{bmatrix} \mathbf{C}\omega^B_{B/I} \\ -\mathbf{I}^{-1}[(\omega^B_{B/I} \times (\mathbf{I}\omega^B_{B/I}))\end{bmatrix} + [I_{3x3}\, 0_{3x3}] \begin{bmatrix}\mathbf{0} \\ \mathbf{I}^{-1}\end{bmatrix} u \\ \dot{y} &= L_f h(x) + L_g h(x) u = \mathbf{C}\omega^B_{B/I} + 0_{3x3}\end{aligned} \tag{12}$$

Clearly $L_g h(x)$ is singular, so another time derivative is taken. This yields

$$\begin{aligned}\ddot{y} &= \nabla(L_f h(x))\,(f(x)+g(x)u) = L_f^2 h(x) + L_g L_f h(x) u \\ \ddot{y} &= [\frac{\partial \mathbf{C}\omega^B_{B/I}}{\partial \alpha} \quad \mathbf{C}] \begin{bmatrix} \mathbf{C}\omega^B_{B/I} \\ -\mathbf{I}^{-1}[(\omega^B_{B/I} \times (\mathbf{I}\omega^B_{B/I}))\end{bmatrix} + [\frac{\partial \mathbf{C}\omega^B_{B/I}}{\partial \alpha} \quad \mathbf{C}] \begin{bmatrix}\mathbf{0} \\ \mathbf{I}^{-1}\end{bmatrix} u \\ \ddot{y} &= \Gamma \mathbf{C}\omega^B_{B/I} - \mathbf{C}\mathbf{I}^{-1}(\omega^B_{B/I} \times (\mathbf{I}\omega^B_{B/I})) + \mathbf{C}\mathbf{I}^{-1} u\end{aligned} \tag{13}$$

In the above derivation, the matrix Γ is defined to be $\frac{\partial \mathbf{C}\omega^B_{B/I}}{\partial \alpha}$ (see appendix). $L_g L_f h(x)$ is non-singular as $\mathbf{C}\mathbf{I}^{-1}$ is invertible. It took two differentiations to obtain an invertible $L_g L_f h(x)$, therefore each output of the system has a relative degree of 2 and the system has a total relative degree of 6 (each output needs to be differentiated twice for the input to appear). The relative degree is equal to the order of our state vector, thus there are no zero-dynamics to check. With $L_f^2 h(x)$ and $L_g L_f h(x)$ determined, the control law (using equation 5) is found to be

$$\boxed{u = (\mathbf{I}\mathbf{C}^{-1})(-\Gamma\mathbf{C}\omega^B_{B/I} + \mathbf{C}\mathbf{I}^{-1}(\omega^B_{B/I} \times (\mathbf{I}\omega^B_{B/I})) + \ddot{r} + \nu)} \tag{14}$$

Applying this control law to the nonlinear system yields the following linear closed loop error dynamics

$$\begin{aligned}\ddot{y} &= [\ddot{\phi}, \ddot{\theta}, \ddot{\psi}]^T = \ddot{r} + \nu \\ \ddot{e} &= -\nu\end{aligned} \tag{15}$$

The auxiliary input was chosen to be $\nu = Kz$ (a simple PD controller), where $z = [e\ \dot{e}]^T$ and $K = [k_p\ k_d]$. This was done so that the LQR problem could be solved to obtain optimal control gains. Solving the LQR problem involves minimizing the cost function $J = \int_0^\infty (z^T Q z + w^T R w) dt$, subject to $\dot{z} = Az + Bw$, where $w = -Kz = -\nu$. Then, the algebraic Riccati equation will be solved for $P = P^T \succ 0$, which will yield the optimal feedback control $w = -R^{-1}BPz$ [1]. The matrices used in the LQR design are

$$A = \begin{bmatrix}0 & 1 \\ 0 & 0\end{bmatrix} \quad B = \begin{bmatrix}0 \\ 1\end{bmatrix} \quad Q = \begin{bmatrix}q_1 & 0 \\ 0 & q_2\end{bmatrix} \quad R = \begin{bmatrix}r_1 & 0 \\ 0 & r_2\end{bmatrix} \tag{16}$$

The resulting control gain matrix K will move the eigenvalues of $A - BK$ to the negative left hand plane, thus resulting in asymptotically stable error dynamics about zero. The control law derived above can be broken into four components. The first component is $L_f^2 h(x) = \Gamma\mathbf{C}\omega^B_{B/I} - \mathbf{C}\mathbf{I}^{-1}(\omega^B_{B/I} \times (\mathbf{I}\omega^B_{B/I})$, which serves as the inner feedback linearization loop. The feedback linearization loop makes the relationship between the auxiliary input and the output, a linear system. The second component is the outer tracking loop, represented by $v = Kz$. This tracking loop guarantees the tracking error will converge asymptotically to zero. The third component is the acceleration feed-forward command given by $\ddot{r}$, which is necessary for accurate closed loop tracking. Finally, the term $L_g L_f h(x) = (\mathbf{CI}^{-1})^{-1}$ is known as control mixing/control allocation which maps the previously mentioned components into a system input [1]. An example block diagram of the dynamic inversion controller is shown below.

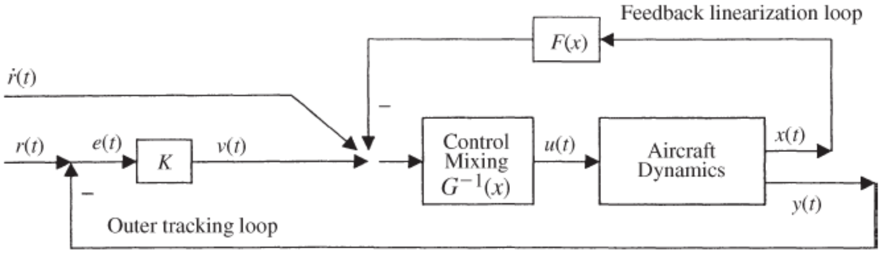

Figure 1: Nonlinear Dynamic Inversion Controller [1]

The control law given by equation 14 is not globally stabilizing nor bounded for all solutions as there exists a singularity in the **C** matrix. This singularity occurs at $\theta = \pm 90°$, and is inherent to the 3-2-1 Euler angle set. This singularity can be avoided by using a different attitude parameterization such as the quaternion [6]. The quaternion parameterization does have its own challenges as its state vector is 4th order. Therefore, one would need to analyze the zero dynamics. The quaternion error is also multiplicative, not additive, due to the unit norm constraint [7] so the error definition used in the above derivation would not hold.

Dynamic Inversion Simulation

The performance of the control law given by (14) is now to be investigated. The reference attitude commands to be followed were generated using the second order filters of the following form, where r_c is a step attitude command.

$$\begin{aligned} \frac{r}{r_c} &= \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \\ \frac{\dot{r}}{r_c} &= \frac{\omega_n^2 s}{s^2 + 2\zeta\omega_n s + \omega_n^2} \\ \frac{\ddot{r}}{r_c} &= \frac{\omega_n^2 s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} r_c \end{aligned} \tag{17}$$

The command filter parameters were set with $\zeta = 0.9$ and $\omega_n = \frac{\pi}{2}$. The plant true inertia's are given as $I_{xx} = 8800$ $[kg\ m^2]$, $I_{yy} = 35250$ $[kg\ m^2]$, $I_{zz} = 42000$ $[kg\ m^2]$, and $I_{xz} = 2900$ $[kg\ m^2]$. The LQR weighting matrices were chosen to be diagonal matrices with $r_1 = r_2 = 10$ (to penalize control inputs of large magnitude) and $q_1 = q_2 = 1$. Solving the LQR probelm with these matrices yielded control gains of $k_p = 0.3162$ $[s^{-2}]$ & $k_d = 0.8558$ $[s^{-1}]$. The same control gains were used in the error loops for yaw, pitch, and roll. The first simulation ran was assuming a perfect model (inertia's known perfectly) with no disturbances and the error loop turned off. The results can be shown in the figures below. The blue line represents the raw unfiltered command. The red line is the filtered command that is to be followed, which is also used for error calculations. Finally, the black line is the plant response.

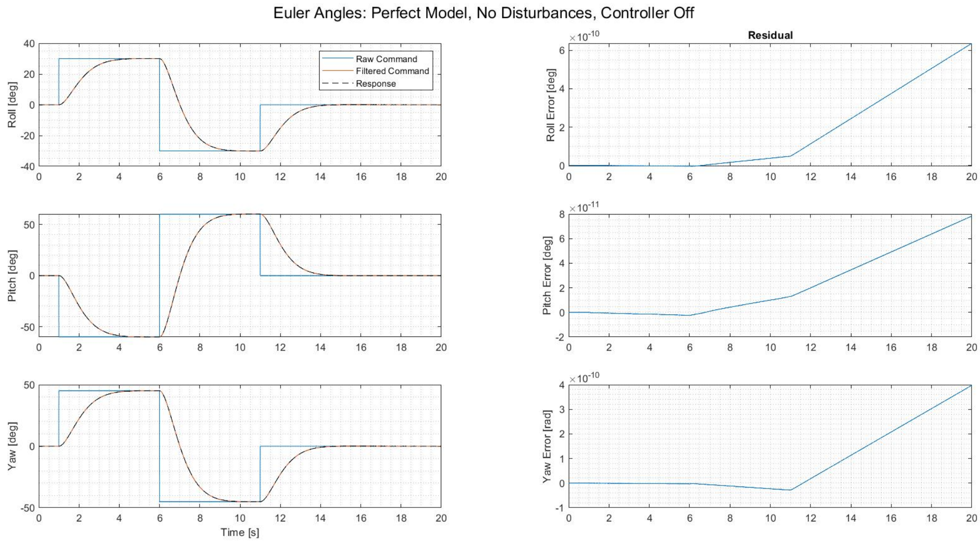

Figure 2: Perfect Model - Euler Angles

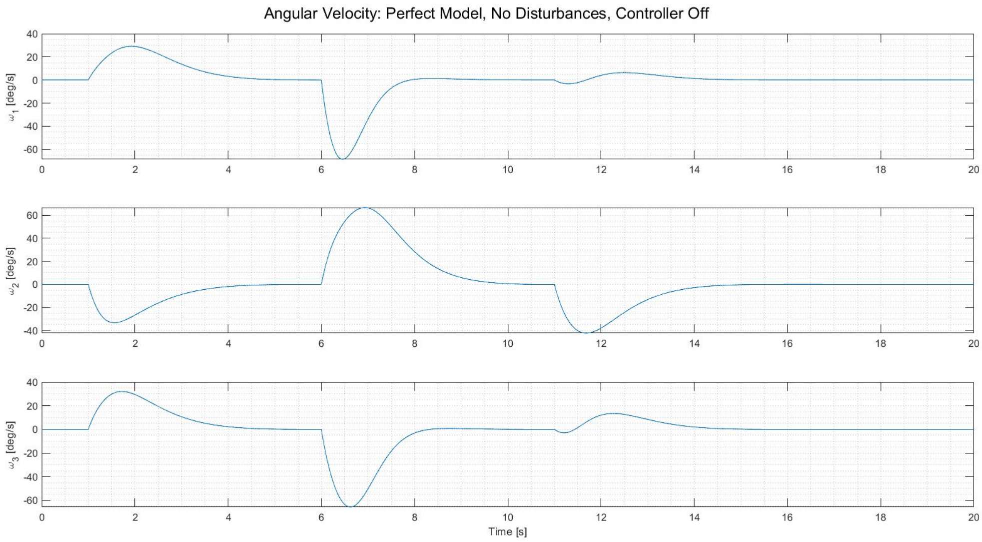

Figure 3: Perfect Model - Angular Velocities

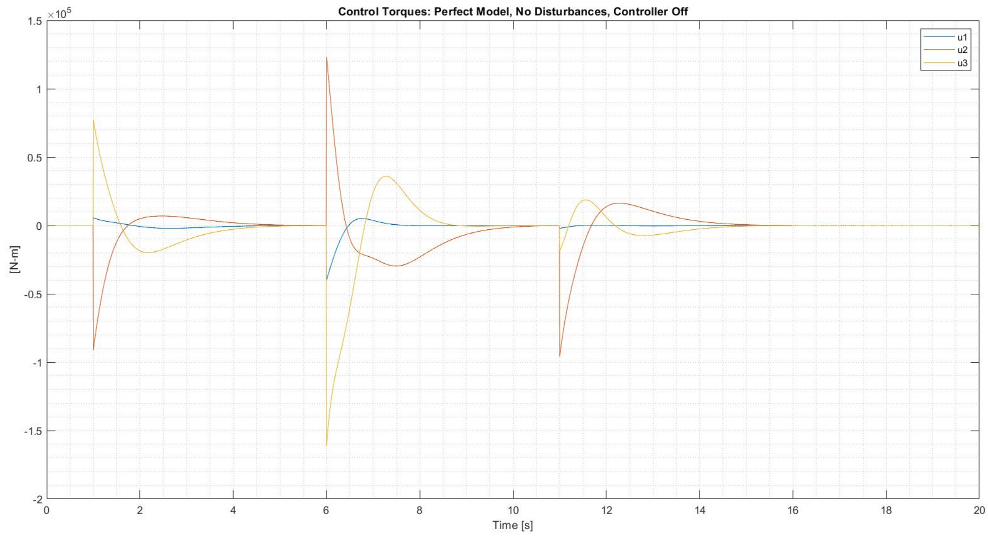

Figure 4: Perfect Model - Control Inputs

It is shown that with no initial condition or modeling errors, the outputs match the reference trajectory perfectly even without the error loops enabled. The angular velocity of the system is observed to be stable throughout the simulation, and eventually settles to zero at the end of the trajectory. The next simulation introduced initial condition error between the reference trajectory and the plant initial conditions. The plant inertia's were again perfectly modeled and there were no external disturbances. This simulation had the error loops enabled and the results are show in the following figures.

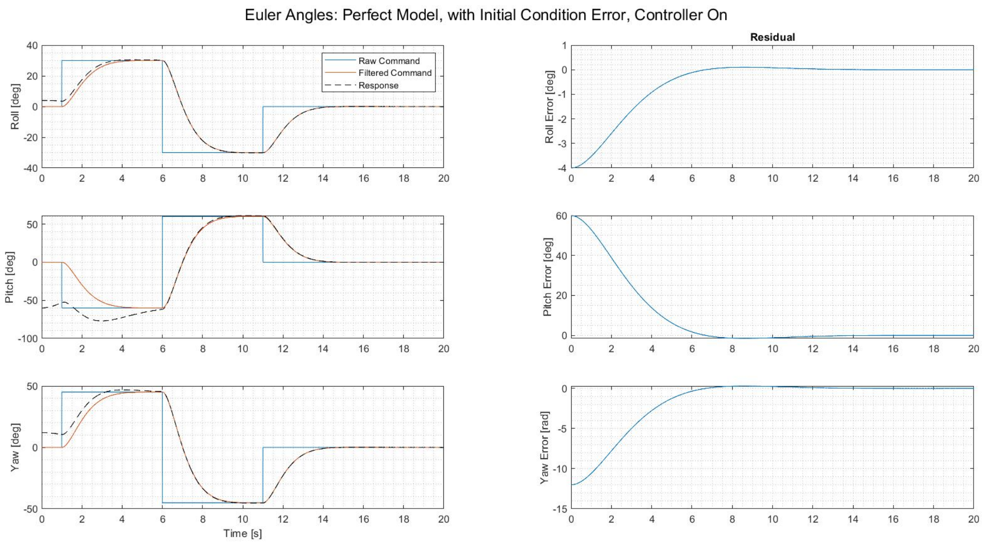

Figure 5: Perfect Model with IC Errors - Euler Angles

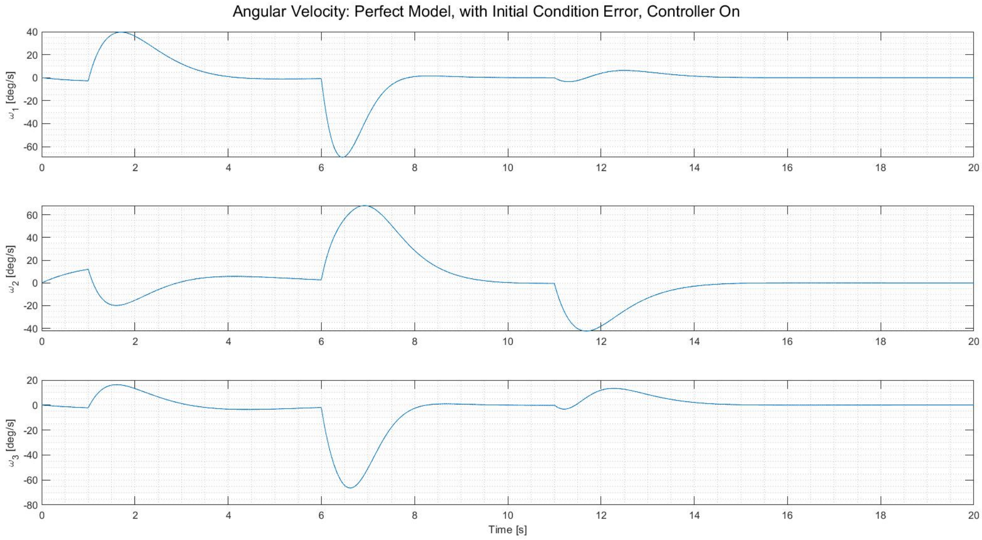

Figure 6: Perfect Model with IC Errors - Angular Velocity

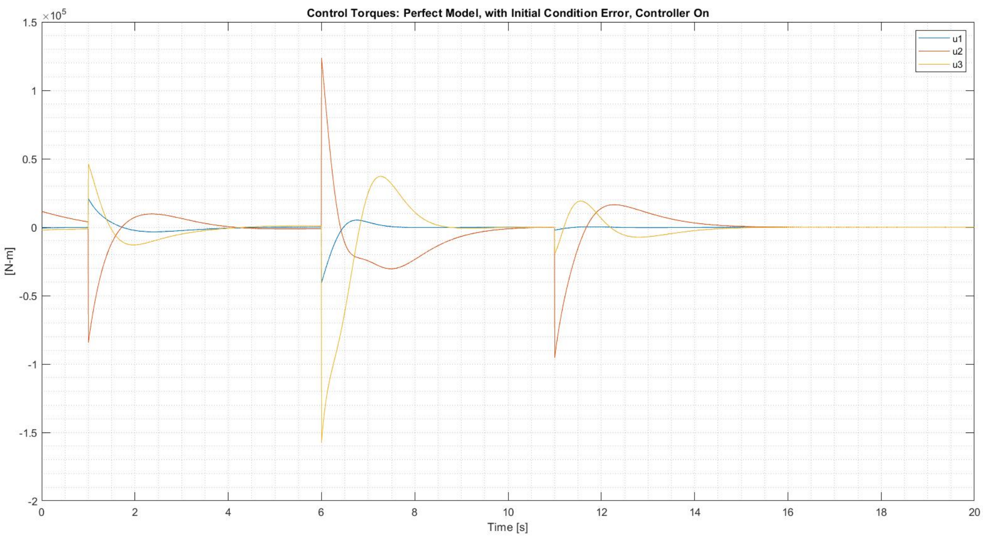

Figure 7: Perfect Model with IC Errors - Control Inputs

With the error loops enabled, the error induced by the initial condition errors asymptotically goes to zero, as expected from the derived closed loop error dynamics. Once the error transience dies out, the outputs track the reference command perfectly. Again the angular velocity state variables are stable and bounded. Dynamic Inversion appears to be a suitable control technique for attitude control if the plant is well modeled. For the final simulation the initial condition error is removed. However, an unmodeled sinusoidal disturbance torque with an amplitude of 300 $[N\ m]$ is applied to all three axes, and the plant moment of inertia's are modeled erroneously. The diagonal elements of the inertia matrix are modeled incorrectly, as they are selected to be 15% larger than their true values in order to test modeling error sensitivity. The results are plotted below.

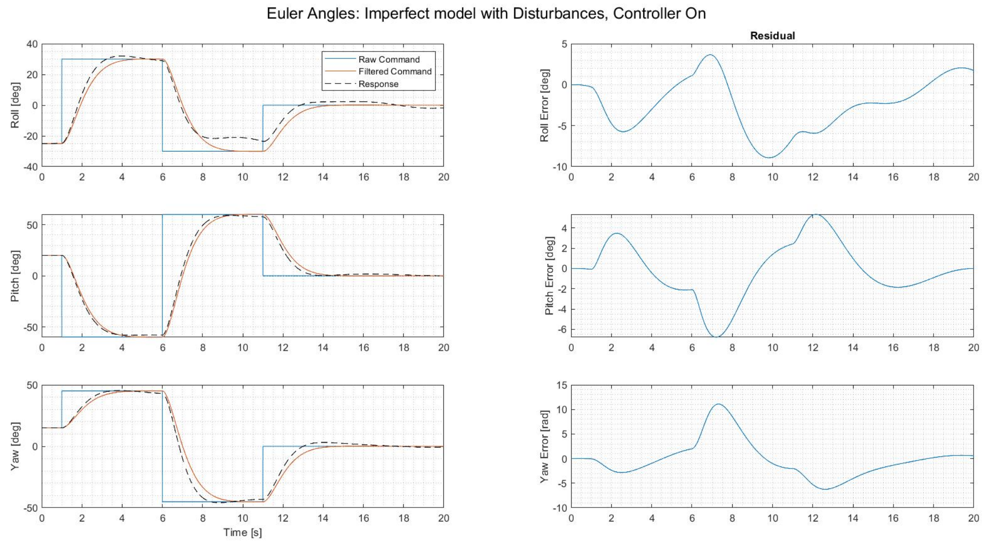

Figure 8: Imperfect Model - Euler Angles

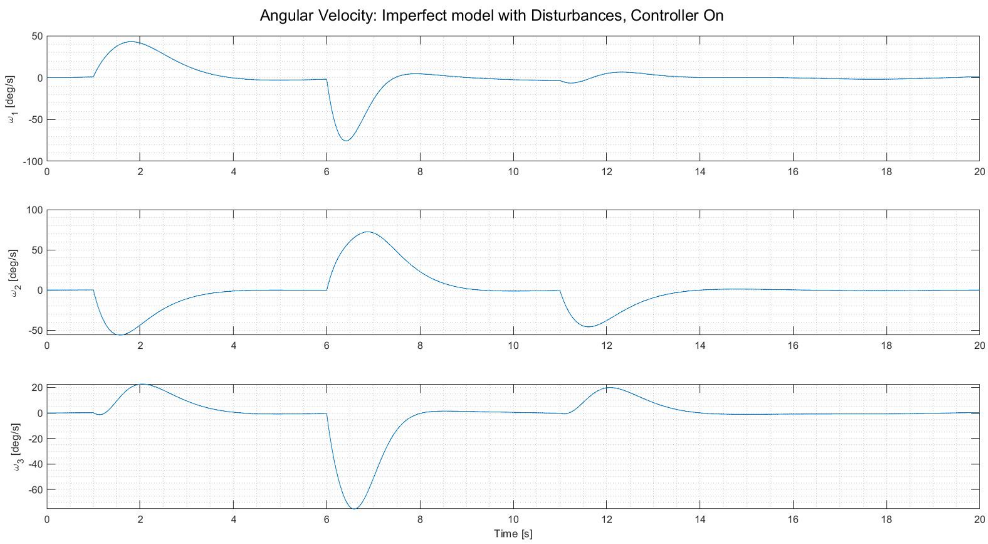

Figure 9: Imperfect Model - Angular Velocity

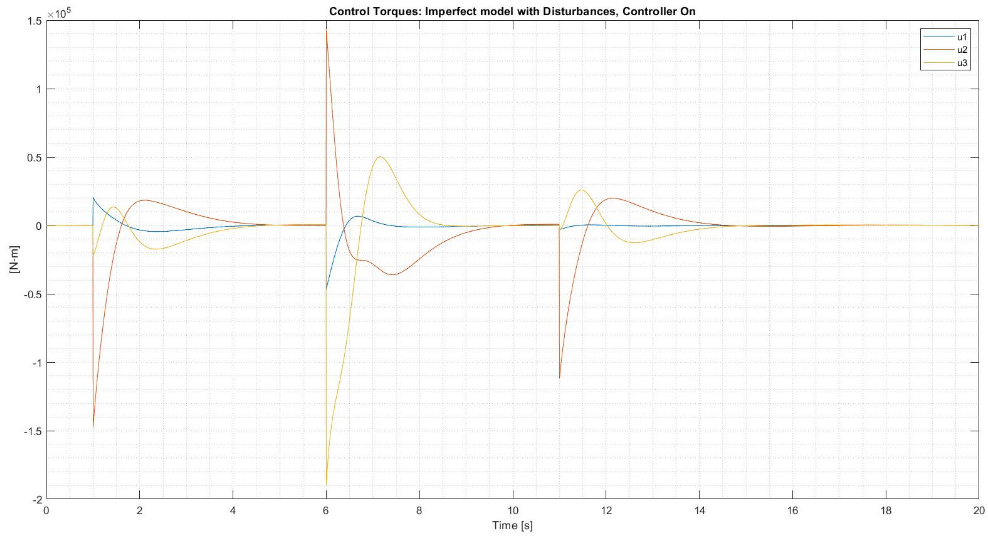

Figure 10: Imperfect Model - Control Inputs

The attitude tracking performance is the worst in the roll axis, but this is expected as the roll axis has the smallest moment of inertia so it is more susceptible to disturbance torques. The pitch and yaw tracking performance is marginally better as the desired trajectory is followed for the most part. The tracking error never settles to zero due to the unmodeled disturbance and the peaks of the desired response cannot be tracked correctly due to the erroneous inertia values. The control law does keep the angular velocity vector stable and bounded throughout the simulation, but the effects of modeling errors for dynamic inversion based control are clear in this simulation.

Conclusion

Input-output feedback linearization (or Dynamic Inversion) is a desirable nonlinear control method due to its ability to eliminate system non-linearities. It allows for the design of simple linear closed loop error dynamics, and has shown to be effective at tracking desired trajectories even when initial conditions errors may exist. A locally stabilizing attitude control law was derived using this input-output feedback linearization technique. The effectiveness of this control law was demonstrated for the attitude tracking control problem shown in the first two simulations. However, the quality of the model of the plant is inherent to the control law performance. This is due to the plant model being directly used in the control law derivation. Therefore, dynamic inversion as a control scheme is not robust to modeling errors, as shown in the last simulation. Dynamic inversion also requires full state feedback and that appropriate controlled variables be selected. These problems may be overcome by using advanced adaptive control techniques, which were not discussed in this report.

References

[1] B. L. Stevens and F. L. Lewis, *Aircraft control and Simulation.* Hoboken, N.J.: J. Wiley, 2003.
[2] Umberto Saetti, J. F. Horn, Sagar Lakhmani, C. Lagoa, and T. Berger, "Design of Dynamic Inversion and Explicit Model Following Control Laws for Quadrotor Inner and Outer Loops," Jan. 2018.
[3] D. J. Caraway, Q. C. Harris, and M. C. Cotting, "VISTA X-62A Model Following Algorithm Overview," *AIAA SCITECH 2023 Forum*, Jan. 2023, doi: https://doi.org/10.2514/6.2023-1746.
[4] H. Schaub, M. R. Akella, and J. L. Junkins, "Adaptive Control of Nonlinear Attitude Motions Realizing Linear Closed Loop Dynamics," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 1, pp. 95–100, Jan. 2001, doi: https://doi.org/10.2514/2.4680.
[5] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control.* Pearson Education, 1991.
[6] Hanspeter Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems.* Reston, Va: American Institute Of Aeronautics And Astronautics, Inc, 2018.
[7] Crassidis, J.L. and Junkins, J.L., *Optimal Estimation of Dynamic Systems - Second Edition*, Chapman & Hall/CRC, Boca Raton, FL, 2012.

Appendix

The Simulink block diagram implementation of the dynamic inversion control scheme is shown below.

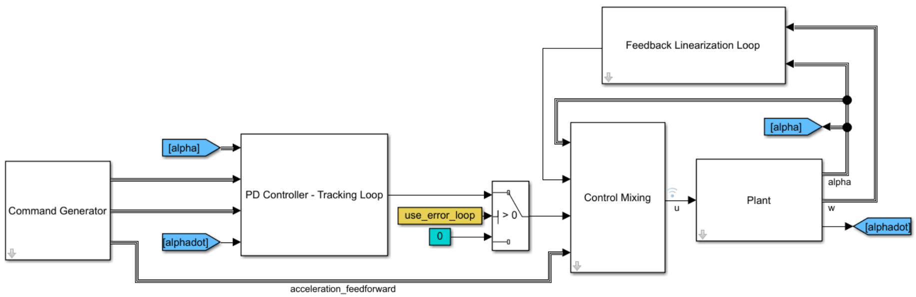

Figure 11: Nonlinear Dynamic Inversion Controller Simulink Implementation

Some matrices that were used in the control law derivation weren't properly defined. These matrices are defined here. Below are the Euler Angle Kinematic Differential Equations for the 3-2-1 Euler set.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \omega^B_{B/I} = \mathbf{C}\,\omega^B_{B/I} \tag{18}$$

The moment of inertia tensor is of the following form:

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{bmatrix} \tag{19}$$

The matrix Γ is defined to be $\frac{\partial \mathbf{C}\omega^B_{B/I}}{\partial \alpha}$. Γ is given below, with a singularity at $\theta = 90°$:

$$\Gamma = \begin{bmatrix} \omega_2 \cos\phi \tan\theta - \omega_3 \sin\phi \tan\theta & \frac{\omega_2 \sin\phi + \omega_3 \cos\phi}{\cos^2\theta} & 0 \\ -\omega_2 \sin\phi - \omega_3 \cos\phi & 0 & 0 \\ \frac{\omega_2 \cos\phi - \omega_3 \sin\phi}{\cos\theta} & \frac{\sin\theta}{\cos^2\theta}(\omega_2 \sin\phi + \omega_3 \cos\phi) & 0 \end{bmatrix} \tag{20}$$

Lastly, it will be shown that there are no zero dynamics for this system. To analyze the zero dynamics, the output, $y = \alpha$, and its time derivatives, $\dot{y}$ & $\ddot{y}$, will be zero. Therefore, $\ddot{e} = \ddot{r} = -v$. Applying this to the control law in equation 14 results in a control law of the form given by equation 8. This control input will force the system output to be zero. Applying this control law to the plant yields the system zero dynamics

$$\dot{x}_0 = \begin{bmatrix} \mathbf{C}\omega^B_{B/I} \\ -\mathbf{I}^{-1}[(\omega^B_{B/I} \times (\mathbf{I}\omega^B_{B/I})) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I}^{-1} \end{bmatrix} (\mathbf{IC}^{-1})(-\Gamma\mathbf{C}\omega^B_{B/I} + \mathbf{CI}^{-1}(\omega^B_{B/I} \times (\mathbf{I}\omega^B_{B/I}))) = \begin{bmatrix} \dot{\alpha} \\ \dot{\omega}^B_{B/I} \end{bmatrix} \tag{21}$$

Because the outputs and their derivatives are zero, $\mathbf{C}\omega^B_{B/I}$ must equal zero as $\dot{\alpha} = 0$. If the outputs are zero, then $\mathbf{C}$ will reduce to the identity matrix. This then forces $\omega^B_{B/I}$ to be zero. This in turn then forces $\dot{\omega}^B_{B/I}$ to also be zero. Therefore there are no zero dynamics for the system as $\dot{x}_0$ equals the zero vector.

A GitHub repository containing all work for this report is found here: `https://github.com/gpcolange/Graduate_Coursework/tree/main/AAE666/Project/Euler%20Angle%20NDI`