Initial Research Proposal

Jongann Lee

October 15, 2023

1 Introduction

2 Previous Research

Before starting this proposal, it is important to clarify the coordinate systems used. Both the world axis and body axis will follow the NED convention. The most important statement here is that down will be the positive z-axis direction.

This proposal aims to combine two previously established controllers to create an adaptive quadrotor controller.

The controller will be first and foremost be based on the geometric tracking control proposed by Taeyoung Lee[1]. This controller is based on the dynamical model of the quadrotor, and uses proportional error feedback in both the translational and rotational variables to control both the vehicle's position and orientation. The errors are defined as follows.

$$e_x = x - x_d \tag{1}$$

$$e_v = v - v_d \tag{2}$$

$$e_R = \frac{1}{2}(R_d^T R - R^T R_d)^\vee \tag{3}$$

$$e_\Omega = \Omega - R^T R_d \Omega_d \tag{4}$$

The vee map $\vee : \mathfrak{so}(3) \to \mathbb{R}^3$ in equation 3 is defined as the inverse of the hat map.

From the desired trajectory and orientation $x_d(t), \vec{b}_{1d}(t)$, the desired orientation $R_d \in SO(3)$ can be calculated as follows.

$$\vec{b}_{3d} = \frac{-k_x e_x - k_v e_v - mge_3 + mg\ddot{x}_d}{|-k_x e_x - k_v e_v - mge_3 + mg\ddot{x}_d|} \tag{5}$$

From this equation we can then create the desired orientation as follows.

$$R_d = \left[\vec{b}_{2d} \times \vec{b}_{3d}, \vec{b}_{2d}, \vec{b}_{3d}\right] \tag{6}$$

where

$$\vec{b}_{2d} = \frac{\vec{b}_{3d} \times \vec{b}_{1d}}{|\vec{b}_{3d} \times \vec{b}_{1d}|} \tag{7}$$

The control inputs are then given as follows

$$f = -(-k_x e_x - k_v e_v - mge_3 + m\ddot{x}_d) \cdot Re_3 \tag{8}$$

$$M = -k_R e_R - k_\Omega e_\Omega + \Omega \times J\Omega - J(\hat{\Omega} R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d) \tag{9}$$

This controller is made adaptive using the natural adaptive control law proposed by Taeyoon Lee[2]. This is an adaptive control method that aims to improve performance by respecting the physical consistency of estimated inertial parameters.

Inertial parameters are often put in a 10 dimensional vector $\phi_b \in \mathbb{R}^10$ of the following form.

$$\phi_b = [m, h_x, h_y, h_z, I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{yz}, I_{xz}] \in \mathbb{R}^{10} \tag{10}$$

While this is the most intuitive form to organize these parameters, using this vector directly during optimization causes problems because the set of physically consistent ϕ_b is a subset of $\mathbb{R}^{10}$. Not only does the I_C with respect to the center of mass have to be positive definite, but a triangle inequality between the eigenvalues of I_C must hold as well.

This cumbersome requirement can be turned into a much simpler requirement by converting ϕ_b into a 4×4 matrix. Define a 4×4 symmetric matrix $P_b \in S(4)$ as follows

$$P_b = \int \begin{bmatrix} \vec{r_b} \\ 1 \end{bmatrix} (\begin{bmatrix} \vec{r_b} \\ 1 \end{bmatrix})^T \rho(\vec{r_b}) dV_b = \begin{bmatrix} \Sigma_b & h_b \\ h_b^T & m \end{bmatrix} \tag{11}$$

Where $\Sigma_b = \frac{1}{2} tr(I_b)\mathbf{1} - I_b$. It is shown in [3] that the physical consistency requirement is equivalent to $P_b \succ 0$ as in being positive definite. A one to one linear mapping $f : \mathbb{R}^{10} \to S(4)$ exists where $f(\phi_b) = P_b$

This essentially means that the set of all physically consistent parameters is equivalent to a manifold $\mathcal{M} \simeq \mathcal{P}(4)$. Based on the Riemannian geometry of this manifold, a new geodesic based distance metric can be defined[2]. For $P \in \mathcal{P}(4)$ and tangent vector $X, Y \in T_P\mathcal{P}(4)$ a coordinate invariant Riemannian metric invariant is given by

$$\langle X, Y \rangle_P = \frac{1}{2} tr(P^{-1} X P^{-1} Y) \tag{12}$$

The geodesic distance between two $P_1, P_2 \in \mathcal{P}(4)$ is then as follows

$$d_{\mathcal{P}(4)}(P_1, P_2) = \Big(\sum_{i=1}^{n} (log(\lambda_i))^2\Big)^{\frac{1}{2}} \tag{13}$$

Which means the distance metric between two inertial parameters can be defined as the following.

$$d_{\mathcal{M}}(^1\phi_b, ^2\phi_b) = d_{\mathcal{P}(4)}(^1P_b, ^2P_b) \tag{14}$$

However, using this metric to construct Lyapunov functions is not practical due to the its non-linearity. Instead, [2] proposes using the Bregman divergence of a log-det function $F(P) = -log(det(P))$ on $\mathcal{P}(4)$ as a distance metric. It is defined as follows.

$$D_{F(\mathcal{P}(4))}(P||Q) = \log(\frac{\det(Q)}{\det(P)}) + tr(Q^{-1}P) - 4 = \sum_{i=1}^{4} (-\log(\lambda_i) + \lambda_i - 1) \tag{15}$$

Where λ_i are the eigenvalues of $Q^{-1}P$.

The original control law in [2] was based on existing controllers whose time derivative of the trajectory error Lyapunov function is in the following form.

$$\dot{V}_t \leq \tilde{\phi}^T b \tag{16}$$

However, [4] has demonstrated that controllers can be designed even if equation 16 doesn't hold. It designed an adaptive attitude controller based on the geometric controller of [1] while using the Bregman divergence as a distance metric for adaptive control.

3 Proposal

The main proposal of this paper is to extend the work of [4] to create an adaptive controller that controls both attitude and trajectory. This will be done by combining the geometric tracking control from [1] with the Bregman divergence adaptive scheme of [2].

An error from [4] must first be stated. It uses the Bregman divergence metric on a vector consisting of the inertial parameters, and not the whole 4×4 converted inertial parameter matrix. Bregman divergence was formulated on a manifold consisting of the positive definite matrices in $S(4)$. Therefore, it is the judgement of this report that this is an incorrect use of Bregman divergence.

However, the overall mathematical structure is correct and therefore, it will be utilized in the creation of the combined controller.

The adaptive controller is as follows.

$$f = -(-k_x e_x - k_v e_v - \hat{m} g e_3 + \hat{m} g \ddot{x}_d) \cdot R e_3 \tag{17}$$

$$M = -k_R e_R - k_\Omega e_\Omega + \Omega \times \hat{J}\Omega - \hat{J}(\hat{\Omega} R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d) \tag{18}$$

Let's first start with the Lyapunov functions used in the creation of the geometric tracking controller. The Lyapunov stability analysis for this controller was done separately for translation and rotation. The Lyapunov function for translation is V_1 and the one for rotation is V_2. That makes the overall Lyapunov function $V = V_1 + V_2$. The value of these functions are as follows.

$$V_1 = \frac{1}{2} k_x |e_x|^2 + \frac{1}{2} m |e_v|^2 + c_1 e_x \cdot e_v \tag{19}$$

$$V_2 = \frac{1}{2} e_\Omega \cdot J e_\Omega + k_R \Psi(R, R_d) + c_2 e_R \cdot J e_\Omega \tag{20}$$

The third term of V_2 has been modified in accordance with [4] to simplify the math later on.

Because we have defined a new control law, a new Lyapunov function will be necessary. The Lyapunov function for the new adaptive control law is defined to be the same as the Lyapunov function for the geometric tracking controller, except for the addition of the Bregman divergence term. Mathematically, it looks like the following

$$\hat{V} = V_1 + V_2 + D_{F(\mathcal{P}(4))}(P || \hat{P}) \tag{21}$$

The time derivative of a Lyapunov function is crucial for Lyapunov analysis. In order to simplify the derivatives of the above Lyapunov functions, the following relations will be utilized.

$$\dot{e}_x = e_v \tag{22}$$

$$m\dot{e}_v = -k_x e_x - k_v e_v - X + (\hat{m} - m)(\ddot{x}_d - g e_3) \tag{23}$$

where

$$X = \frac{f}{e_3^T R_d^T R e_3}((e_3^T R_3^T R e_3) R e_3 - R_d e_3) \tag{24}$$

$$\dot{e}_R = \frac{1}{2}(tr[R^T R_d] I - R^T R_d) e_\Omega = C(R_d^T R) e_\Omega \tag{25}$$

$$J\dot{e}_\Omega = -k_R - k_\Omega + Y(\dot{\hat{J}} - \bar{J}) \tag{26}$$

$$\dot{\Psi}(R, R_d) = e_\Omega \cdot e_R \tag{27}$$

Here $Y = Y_1 + Y_2 \in \mathbb{R}^{3\times 6}$ and $\bar{J} \in \mathbb{R}^6$ is defined to linearize equation 26 in relation to the moment of inertia.

$$Y_1 = \begin{bmatrix} 0 & \omega_2\omega_3 & -\omega_2\omega_3 & \omega_1\omega_3 & -\omega_1\omega_2 & \omega_3^2 - \omega_2^2 \\ -\omega_1\omega_3 & 0 & \omega_1\omega_3 & -\omega_2\omega_3 & \omega_1^2 - \omega_3^2 & \omega_2\omega_1 \\ \omega_1\omega_2 & -\omega_2\omega_1 & 0 & \omega_2^2 - \omega_1^2 & \omega_3\omega_2 & -\omega_3\omega_1 \end{bmatrix} \tag{28}$$

$$Y_2 = \begin{bmatrix} \alpha_1 & 0 & 0 & \alpha_2 & \alpha_3 & 0 \\ 0 & \alpha_2 & 0 & \alpha_1 & 0 & \alpha_3 \\ 0 & 0 & \alpha_3 & 0 & \alpha_1 & \alpha_2 \end{bmatrix} \tag{29}$$

where $\alpha = \hat{\Omega} R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d \in \mathbb{R}^3$

$$\bar{J} = \begin{bmatrix} J_{xx} & J_{yy} & J_{zz} & J_{xy} & J_{xz} & J_{yz} \end{bmatrix} \tag{30}$$

Note that equation 23 and 26 contain terms at the end that arise from the difference between estimated inertia and real inertia. In the original geometric tracking controller, such terms did not exist. In fact, in [1] the derivative of the Lyapunov function looked as follows.

$$\dot{V}_1 = (c_1 - k_v)|e_v|^2 - \frac{c_1 k_x}{m}|e_x|^2 - \frac{c_1 k_x}{m} e_x \cdot e_v - X \cdot (\frac{c_1}{m} e_x + e_v) \tag{31}$$

$$\dot{V}_2 = -k_\Omega |e_\Omega|^2 + c_2 C(R, R_d) e_\Omega \cdot J e_\Omega - c_2 k_R |e_R|^2 - c_2 k_\Omega e_R \cdot e_\Omega \tag{32}$$

It has been shown that given a sufficiently small but positive c_1 and c_2, with the initial attitude error being less than 90 deg, the translation error and attitude error both exponentially converge to zero[1]. This means the Lyapunov functions V, V_1 and V_2 directly prove the exponential stability of the system.

Under the dynamics created by the new adaptive controller, the derivative of the Lyapunov function $\hat{V}$ is as follows.

$$\dot{\hat{V}} = \dot{V}_1 + \dot{V}_2 + (\frac{c_1}{m} e_x + e_v) \cdot (\hat{m} - m)(\ddot{x}_d - g e_3) + (e_\Omega + c_2 e_R) \cdot Y(\hat{J} - J) + \dot{D}_{F(\mathcal{P}(4))}(P || \hat{P}) \tag{33}$$

where

$$\dot{D}_{F(\mathcal{P}(4))}(P||\hat{P}) = tr(\hat{P}^{-1}\dot{\hat{P}} + \hat{P}^{-1}\dot{\hat{P}}\hat{P}^{-1}P) = tr([\hat{P}^{-1}\dot{\hat{P}}\hat{P}^{-1}](\hat{P} - P)) \tag{34}$$

In order to simplify the notation, the following two error definitions $e_1, e_2 \in \mathbb{R}^3$ will be used.

$$e_1 = \frac{c_1}{m} e_x + e_v \tag{35}$$

$$e_2 = e_\Omega + c_2 e_R \tag{36}$$

The tilde notation will also be used for denoting the difference between the estimated and actual values, meaning $\tilde{m} = \hat{m} - m$ and $\tilde{\bar{J}} = \hat{\bar{J}} - \bar{J}$.

To organize this further, Lyapunov function terms related to the inertial parameter error will be organized into the following form.

$$e_1 \cdot (\tilde{m})(\ddot{x}_d - g e_3) + e_2 \cdot Y(\tilde{\bar{J}}) = tr(B\tilde{P}) \tag{37}$$

The specific value of B is as follows. First, a column vector $Z \in \mathbb{R}^6$ will be defined as $Z = e_2^T Y$. Then define $B \in \mathbb{R}^{4\times 4}$ is defined as follows

$$B = \begin{bmatrix} Z_2 + Z_3 & -\frac{1}{2}Z_4 & -\frac{1}{2}Z_5 & 0 \\ -\frac{1}{2}Z_4 & Z_1 + Z_3 & -\frac{1}{2}Z_6 & 0 \\ -\frac{1}{2}Z_5 & -\frac{1}{2}Z_6 & Z_1 + Z_2 & 0 \\ 0 & 0 & 0 & e_1 \cdot (\ddot{x}_d - g e_3) \end{bmatrix} \tag{38}$$

Based on these new definitions, the Lyapunov function derivative now becomes

$$\dot{\hat{V}} = \dot{V}_1 + \dot{V}_2 + tr(B\tilde{P} + [\hat{P}^{-1}\dot{\hat{P}}\hat{P}^{-1}]\tilde{P}) \tag{39}$$

The parameter update rule can then be set as the following

$$\dot{\hat{P}} = -\hat{P}B\hat{P} \tag{40}$$

This eliminates the last term from the Lyapunov function dynamics, making it $\dot{\hat{V}} = \dot{V}_1 + \dot{V}_2$. Following from [1], this means that the zero equilibrium of the tracking error are exponentially stable.

In conclusion this means the new control laws equations 19 and 20 combined with the inertial parameter update law equation 40 creates an adaptive control law that makes the zero equilibrium of the tracking error are exponentially stable.

4 Simulation

A simulation of the control law was performed using Matlab. The simulation involved making the quadcopter follow a trajectory in the form of a Lissajous curve. The desired trajectory is defined as below.

$$x_d(t) = \begin{bmatrix} \sin(\frac{1}{10}t) & \cos(\frac{1}{10}t) & \sin(\frac{1}{15}t + \frac{\pi}{4}) \end{bmatrix}^T, \psi_d(t) = -\frac{1}{10}t \tag{41}$$

The necessary $\dot{x}_d, \ddot{x}_d, R_d, \Omega_d$ was then found using the method from [5]. Modifications to the equations were made to fit the NED coordinates.

Two simulations were conducted. Both simulations used the geometric tracking controller with the initial conditions set as the following.

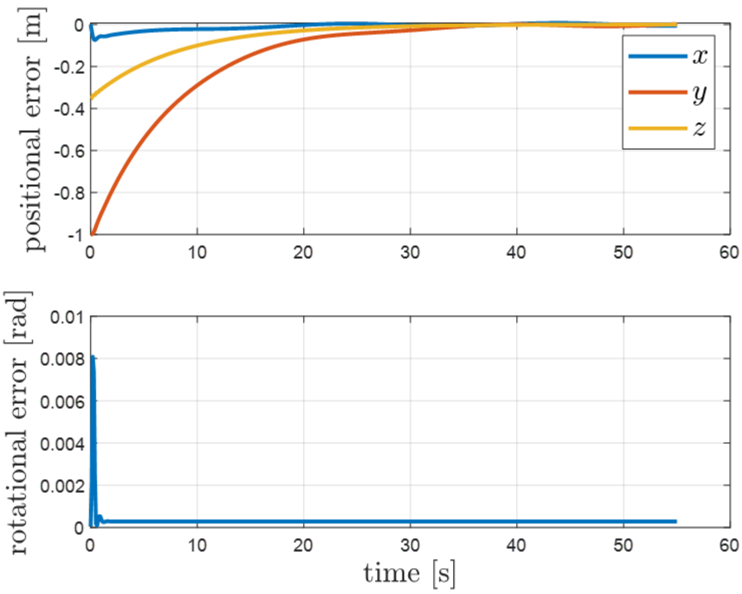

Figure 1: Simulation result with adaptive control law

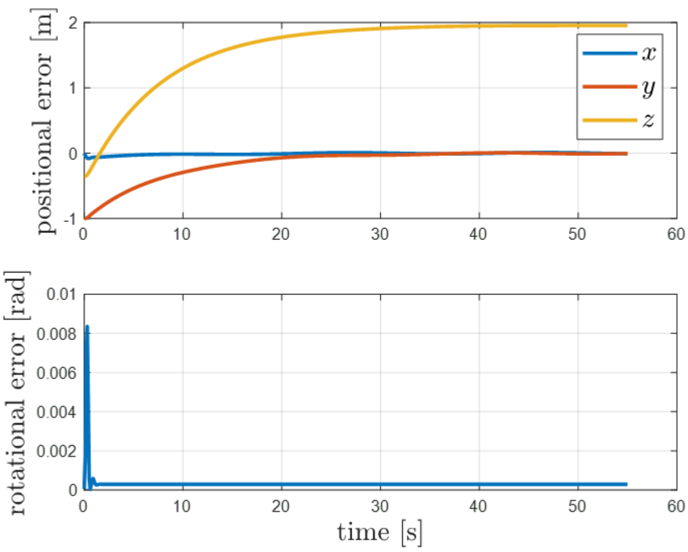

Figure 2: Simulation result without adaptive control law

$$m = 4kg, J = \begin{bmatrix} 0.08 & 0 & 0 \\ 0 & 0.08 & 0 \\ 0 & 0 & 0.14 \end{bmatrix}, \hat{m} = 3.6kg, \hat{J} = \begin{bmatrix} 0.07 & 0 & 0 \\ 0 & 0.07 & 0 \\ 0 & 0 & 0.13 \end{bmatrix} \quad (42)$$

$$x_0 = v_0 = 0, R_0 = I, \Omega_0 = [0\ 0\ 0]^T \quad (43)$$

$$k_x = 2, k_v = 16, k_R = 10, k_\Omega = 1, c_1 = c_2 = 0.5 \quad (44)$$

Figure 1 shows the result with the control law of equation 40 enabled, while figure 2 shows the result with the control law disabled.

References

[1] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," *49th IEEE conference on Decision and Control (CDC)*, 2010.

[2] T. Lee, J. Kwon, and F. C. Park, "A natural adaptive control law for robot manipulators," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[3] P. M. Wensing, S. Kim, and J.-J. E. Slotine, "Linear matrix inequalities for physically consistent inertial parameter identification: A statistical perspective on the mass distribution," *IEEE Robotics and Automation Letters*, 2018.

[4] K. Patnaik and W. Zhang, "Adaptive attitude control for foldable quadrotors," *IEEE Control Systems Letters, Vol. 7*, 2023.

[5] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," *2011 IEEE international conference on robotics and automation*, 2011.