Robust State-Feedback H_∞ Control of Quadrotor

Sheng-Wen Cheng and Hsin-Ai Hung

***Abstract*—This paper presents an H_∞ control design for quadrotors to enhance robustness against unmodeled disturbances. In optimal control, the algebraic Riccati equation (ARE) frequently arises in topics like Linear Quadratic Regulator (LQR) or H_∞ control. The H_∞ control minimizes the maximal gain between the disturbance input and the control output, which can be solved by acquiring the solution of ARE while finding the minimal H_∞ norm of the corresponding transfer function. Thus, a simple bisection algorithm is also proposed for computing the stabilizing ARE solution of the H_∞ control, which combines a state-of-the-art algorithm for the ARE called the structure-preserving doubling algorithm (SDA).**

I. Introduction

The applications of quadrotors are versatile. This new technology has increasing demands and keeps emerging in new scenarios. Unfortunately, most quadrotors can only be operated in moderated conditions and cannot resist harsh environments, especially for designs like PID control [1], [2]. Disturbances like winds, unbalanced payload, or motor degeneration can all affect flying performance, sometimes devastating. One may overcome them by applying a robust controller like H_∞ control, which can attenuate the disturbance response to the control output.

H_∞ control is a topic strongly related to the eigenvalue problems of the Hamiltonian matrix. The stable invariant subspace of the Hamiltonian matrix forms the solution of the algebraic Riccati equation (ARE), which is the key to solving extensive optimal control problems like LQR [3], [4], or H_∞ control. In 1979, Laub proposed a Schur method [5], [6] for solving the ARE using the QR algorithm and a reordering scheme. Later, researchers created structure-preserving algorithms [7], [8], [9], which utilize the orthogonal symplectic transformations. A class of works named the doubling algorithm, inspired by the fixed-point iteration method, was investigated in the period of 1970s and 1980s. By 2005, a novel algorithm called the structure-preserving doubling algorithm (SDA) [10] was proposed. It is numerically stable and capable of generating doubled convergence sequence by utilizing the Cayley transform. Another algorithm worth mentioning is the matrix sign function method (MSFM) [11], an iterative-based method. Additionally, H_∞ control problems can be solved by calculating the solution of ARE while minimizing the H_∞ norm of the transfer function simultaneously, where [12] proposed a bisection and secant method.

Sheng-Wen Cheng is with the Graduate Degree Program of Robotics, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan, E-mail: shengwen.c@nycu.edu.tw.

Hsin-Ai Hung is with the Department of Mechanical Engineering, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan, E-mail: louise.c@nycu.edu.tw.

The rest of the paper is organized as follows: Section II provides a preliminary knowledge of H_∞ control and proposes a bisection algorithm for solving the optimal H_∞ control input. Section III formulates the control model of the quadrotor. Section IV presents the simulation result. Finally, Section V concludes all works done in this paper.

II. Preliminaries of the H_∞ Control

Fig. 1 shows the control block diagram of a typical H_∞ control system, which is formulated as

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + \begin{bmatrix} B_1 & B_2 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} \\ \begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix} &= \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & D_{12} \\ D_{21} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix}, \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the system state, $\mathbf{y}$ is the measured output, $\mathbf{w}$ is the exogenous input (disturbance), $\mathbf{u}$ is the control input, and $\mathbf{z}$ is the performance output (i.e., penalty error) to be minimized.

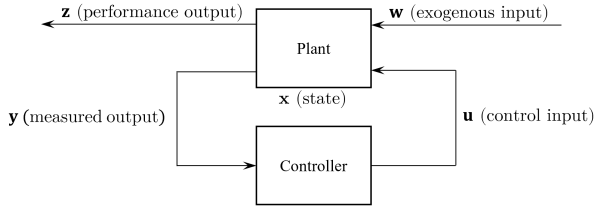

Fig. 1. Control block diagram of a typical H_∞ controller

The goal of the H_∞ control is to stabilize the system while inhibiting the disturbance effect; this can be formulated as an optimization problem such that the supremum of the penalty error divided by the disturbance is bounded under r:

$$\sup_{\mathbf{w}\neq 0} \frac{\|\mathbf{z}\|_2}{\|\mathbf{w}\|_2} \leq r \rightarrow \min_r \left\{ r^2 \|\mathbf{w}\|_2^2 - \|\mathbf{z}\|_2^2 \right\} \geqslant 0,\ \forall \mathbf{w} \neq 0. \tag{2}$$

Consider the following system:

$$\begin{cases} \dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \\ \mathbf{y} = C\mathbf{x} + D\mathbf{u} \end{cases}. \tag{3}$$

Taking the Laplace transform of (3) and rearranging the result with input-output representation yields:

$$\mathbf{y}(s) = \left[C\left(sI - A\right)^{-1} B + D \right] \mathbf{u}(s) \coloneqq T(s)\mathbf{u}(s), \tag{4}$$

where $T(s)$ is called the transfer function and commonly denoted as

$$T(s) \coloneqq \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]. \tag{5}$$

2022 International Automatic Control Conference (CACS) | 978-1-6654-9646-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/CACS55319.2022.9969787

The H_∞ norm of a transfer function is defined as

$$\|T\|_\infty \coloneqq \sup_{\omega\in\mathbb{R}} \|T(i\omega)\|_2 \,, \tag{6}$$

where $i = \sqrt{-1}$. Consider (4) by letting $\mathbf{u} = \mathbf{w}$ (i.e., input with pure disturbance); the H_∞ norm is defined as the largest magnitude over the whole frequency spectrum; it is designed to be minimized by the controller such that the gain between the disturbance and the control output is minimal.

Furthermore, the Hamiltonian matrix is defined as:

$$\mathscr{H} := \begin{bmatrix} A & -G \\ -H & -A^{\mathrm{T}} \end{bmatrix}, \tag{7}$$

which obeys the following lemma of the Riccati function [13].

Lemma 1. *Suppose $\mathscr{H} \in \mathrm{dom}(\mathrm{Ric})$ and $X = \mathrm{Ric}(\mathscr{H})$, i.e., $\mathscr{H} \mapsto X$ is a function, then:*

1) X is symmetric.
2) X satisfies the continuous-time algebraic Riccati equation (CARE):

$$A^{\mathrm{T}}X + XA - XGX + H = 0. \tag{8}$$

3) $A + GX$ is stable.

Next, rewrite the Hamiltonian matrix with the form of

$$\mathscr{H}(r) = \begin{bmatrix} A & r^{-2}BB^{\mathrm{T}} \\ -C^{\mathrm{T}}C & -A^{\mathrm{T}} \end{bmatrix}, \tag{9}$$

and the following lemma is essential for H_∞ control.

Lemma 2 (Bounded Real Lemma)**.** *The following statements are equivalent:*

1) $\|T\|_\infty < r$.
2) $\Lambda(\mathscr{H}(r)) \cap \mathrm{Im} = \emptyset$ ($\Lambda(\cdot)$: spectrum of a matrix).
3) $\mathscr{H}(r) \in \mathrm{dom}(\mathrm{Ric})$.
4) $\mathscr{H}(r) \in \mathrm{dom}(\mathrm{Ric})$ and $X \coloneqq \mathrm{Ric}(\mathscr{H}(r)) > 0$.

The optimal H_∞ control input can be calculated from the CARE solution X; the lemma indicates that the corresponding Hamiltonian matrix must have no eigenvalues on the imaginary axis, and the H_∞ norm of the transfer function is bounded.

In addition, the following assumptions are needed:

1) (A, B_1) is stabilizable and (C_1, A) is detectable.
2) (A, B_2) is stabilizable.
3) $D_{12}^{\mathrm{T}} \begin{bmatrix} C_1 & D_{12} \end{bmatrix} = \begin{bmatrix} 0 & I \end{bmatrix}$.

Then by [13], for the state-feedback H_∞ control problem (i.e., $C_2 = I, D_{21} = 0$) :

$$\begin{cases} \dot{\mathbf{x}} = A\mathbf{x} + B_1\mathbf{w} + B_2\mathbf{u} \\ \mathbf{z} = C_1\mathbf{x} + D_{12}\mathbf{u} \\ \mathbf{y} = \mathbf{x} \end{cases}, \tag{10}$$

the stabilizing control input $\mathbf{u}$ can be solved by

$$\mathbf{u} = -B_2^{\mathrm{T}}X\mathbf{x}, \tag{11}$$

where X is the CARE solution of the following Hamiltonian matrix:

$$\mathscr{H}(r) = \begin{bmatrix} A & r^{-2}B_1B_1^{\mathrm{T}} - B_2B_2^{\mathrm{T}} \\ -C_1^{\mathrm{T}}C_1 & -A^{\mathrm{T}} \end{bmatrix}, \tag{12}$$

and r is to be determined.

A. Bisection Method for Computing H_∞ Norm

[14] shows that the H_∞ norm of a transfer function can be computed by Algorithm 1 with the assumption of $D = 0$. Let r_{lb} and r_{ub} be the lower and upper bound of the bisection algorithm; the bounds can be set empirically or using the analytical result by Enns and Glover [15], [16]:

$$r_{\mathrm{lb}} = \sigma_{\mathrm{H}1},\ r_{\mathrm{ub}} = 2\sum_i \sigma_{\mathrm{H}i}, \tag{13}$$

where $\sigma_{\mathrm{H}i}$ are the Hankel singular values.

Algorithm 1 Bisection Method for Computing H_∞ Norm

Input $\{A,\ B,\ C,\ D(=0)\};\ \epsilon$
Output $r\ (= \|T\|_\infty)$

```
r_L ← r_lb;
r_H ← r_ub;
repeat
    r ← (r_L + r_H)/2;
    H(r) = [ A          r^{-2} B B^T ]
           [ -C^T C     -A^T         ];
    if Λ(H(r)) ∩ Im = ∅ then
        r_H ← r;
    else
        r_L ← r;
    end if
until r_H − r_L ≤ 2ε · r_L
```

B. Structure-Preserving Doubling Algorithm (SDA)

Algorithm 2 demonstrates the procedure of the structure-preserving doubling algorithm (SDA). The SDA converts CAREs (continuous-time algebraic Riccati equation) to DAREs (discrete-time algebraic Riccati equation) by Cayley transform and turns the Hamiltonian matrix into a symplectic matrix pair with the standard symplectic form (SSF). In each iteration, the SDA produces a new symplectic matrix pair in the SSF and eventually yields the doubled convergence sequence $\{X_k\}$. The numerical stability of the SDA is competitive and promising since SSF is a property stronger than symplectic. Therefore, the SDA is fast and accurate. By experiments, the accuracy of the SDA can reach 10^{-7} levels and approximately 4 times faster than the built-in care function of MATLAB. Notice that a parameter $\hat{\gamma}$ is required for Cayley transform in the SDA, and detailed investigations are given by [10]. In practice, it can be set between 2.1 and 2.6.

C. Bisection Method for State-Feedback H_∞ Control

Solving the state-feedback H_∞ control problem requires finding $r_{\mathbf{x}}$, the infimum of r that satisfies Lemma 2.4. In fact, $r_{\mathbf{x}}$ is known to be inside the open interval $(r_{\mathrm{lb}}, \infty)$. The lower bound r_{lb} is characterized by

$$r_{\mathrm{lb}} = \|G_z\|_\infty \,, \tag{14}$$

Algorithm 2 Structure-Preserving Doubling Algorithm (SDA)

Input $\mathscr{H} = \begin{bmatrix} A & -G \\ -H & -A^{\mathrm{T}} \end{bmatrix} \in \mathbf{H}$ with $\Lambda(\mathscr{H}) \cap \mathrm{Im} = \emptyset; \epsilon$

Output The stabilizing solution $X = X^{\mathrm{T}} \geq 0$ to the CARE

Find a appropriate $\hat{\gamma} > 0$;
$A_{\hat{\gamma}} \leftarrow A - \hat{\gamma} I$;
$\hat{A}_0 \leftarrow I + 2\hat{\gamma}(A_{\hat{\gamma}} + G A_{\hat{\gamma}}^{-\mathrm{T}} H)^{-1}$;
$\hat{G}_0 \leftarrow 2\hat{\gamma} A_{\hat{\gamma}}^{-1} G (A_{\hat{\gamma}}^{\mathrm{T}} + H A_{\hat{\gamma}}^{-1} G)^{-1}$;
$\hat{H}_0 \leftarrow 2\hat{\gamma}(A_{\hat{\gamma}}^{\mathrm{T}} + H A_{\hat{\gamma}}^{-1} G)^{-1} H A_{\hat{\gamma}}^{-1}$;
$j \leftarrow 0$;
loop until converged
    $\hat{A}_{j+1} \leftarrow \hat{A}_j (I + \hat{G}_j \hat{H}_j)^{-1} \hat{A}_j$;
    $\hat{G}_{j+1} \leftarrow \hat{G}_j + \hat{A}_j \hat{G}_j (I + \hat{H}_j \hat{G}_j)^{-1} \hat{A}_j^{\mathrm{T}}$;
    $\hat{H}_{j+1} \leftarrow \hat{H}_j + \hat{A}_j^{\mathrm{T}} (I + \hat{H}_j \hat{G}_j)^{-1} \hat{H}_j \hat{A}_j$;
    $j \leftarrow j + 1$;
    if $\|\hat{H}_j - \hat{H}_{j-1}\| \leq \epsilon \|\hat{H}_j\|$ **then** Stop;
    end if
end loop
Set $X \leftarrow \hat{H}_j$;

Algorithm 3 Bisection Method for State-Feedback H_∞ Control

Input $\{A, B_1, B_2, C_1, D_{12}\}; \epsilon;$ precision

Output X, $r_{\mathbf{x}}$

$r_{\mathrm{L}} \leftarrow r_{\mathrm{lb}}$;
$r_{\mathrm{H}} \leftarrow r_{\mathrm{ub}}$;
repeat
    $r \leftarrow (r_{\mathrm{L}} + r_{\mathrm{H}})/2$;
    $\mathscr{H}(r) = \begin{bmatrix} A & r^{-2} B_1 B_1^{\mathrm{T}} - B_2 B_2^{\mathrm{T}} \\ -C_1^{\mathrm{T}} C_1 & -A^{\mathrm{T}} \end{bmatrix}$;
    if $\Lambda(\mathscr{H}(r)) \cap \mathrm{Im} = \emptyset$ **then**
        $\hat{X} = \mathrm{SDA}(\mathscr{H}(r))$;
        residual $\leftarrow \|A^{\mathrm{T}} \hat{X} + \hat{X} A - \hat{X} G \hat{X} + H\|$;
        if residual $\leq$ precision **then**
            $X \leftarrow \hat{X}$;
            $r_{\mathrm{H}} \leftarrow r$;
        else
            $r_{\mathrm{L}} \leftarrow r$;
        end if
    end if
until $r_{\mathrm{H}} - r_{\mathrm{L}} \leq 2\epsilon \cdot r_{\mathrm{L}}$
$r_{\mathbf{x}} \leftarrow r$;

where

$$G_z(s) = \left[\begin{array}{c|c} \left(A - Z C_1^{\mathrm{T}} C_1\right)^{\mathrm{T}} & C_1^{\mathrm{T}} \\ \hline B_1^{\mathrm{T}} & 0 \end{array}\right], \tag{15}$$

and $Z = \mathrm{Ric}(H_2)$ with

$$H_2 = \begin{bmatrix} A^{\mathrm{T}} & -C_1^{\mathrm{T}} C_1 \\ -B_2 B_2^{\mathrm{T}} & -A \end{bmatrix}. \tag{16}$$

According to [12], the upper bound r_{ub} can be approximated by

$$r_{\mathrm{ub}} = \frac{\sigma_{\max}(B_1)}{\sigma_0(B_2)}, \tag{17}$$

where $\sigma_{\max}$ is the biggest singular value, and σ_0 is the smallest nonzero singular value. Notice that (17) requires prior checking of the feasibility; for instance, when $B_1 = 0$, the bisection method can not be performed since $r_{\mathrm{ub}} = 0$. Given the interval, $r_{\mathbf{x}}$ can be found by applying Algorithm 3, which utilizes the residual of CARE solved by the SDA as the criteria for the bisection method.

III. Quadrotor Control

A. Rigid-body Rotation

Fig. 2 illustrates the frame definition of the quadrotor system, where $\{e_1, e_2, e_3\}$ is the inertial frame, i.e., the frame that is not accelerating in the space, and $\{b_1, b_2, b_3\}$ is the body-fixed frame that is constantly changing with the movement of the quadrotor.

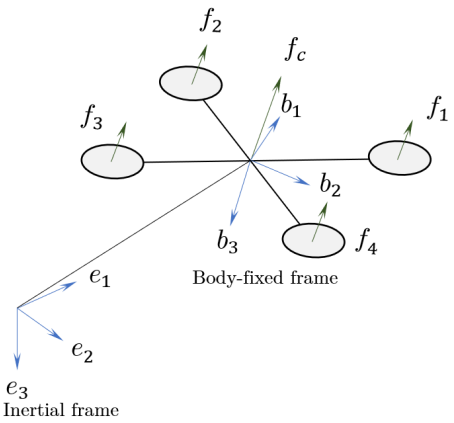

Fig. 2. Definition of the quadrotor coordinate frame

Denote $\mathrm{s} = \sin$, $\mathrm{c} = \cos$ and $\mathrm{t} = \tan$. The rigid-body rotation can be represented by rotation matrices parameterized by Euler angles $[\phi, \theta, \psi]^{\mathrm{T}}$; this requires three successive transformations, where the first rotation on the x-axis with the roll angle ϕ is given by

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}\phi & -\mathrm{s}\phi \\ 0 & \mathrm{s}\phi & \mathrm{c}\phi \end{bmatrix}, \tag{18}$$

the second rotation on the y-axis with the pitch angle θ is given by

$$R_y(\theta) = \begin{bmatrix} \mathrm{c}\theta & 0 & \mathrm{s}\theta \\ 0 & 1 & 0 \\ -\mathrm{s}\theta & 0 & \mathrm{c}\theta \end{bmatrix}, \tag{19}$$

and the third rotation on the z-axis with the yaw angle ψ is given by

$$R_z(\psi) = \begin{bmatrix} \mathrm{c}\psi & -\mathrm{s}\psi & 0 \\ \mathrm{s}\psi & \mathrm{c}\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{20}$$

The rotation matrix from the body-fixed frame to the inertial frame is now defined as

$$\begin{aligned} R &= R_z(\psi)R_y(\theta)R_x(\phi) \\ &= \begin{bmatrix} \mathrm{c}\theta\mathrm{c}\psi & \mathrm{s}\phi\mathrm{s}\theta\mathrm{c}\psi - \mathrm{c}\phi\mathrm{s}\psi & \mathrm{c}\phi\mathrm{s}\theta\mathrm{c}\psi + \mathrm{s}\phi\mathrm{s}\psi \\ \mathrm{c}\theta\mathrm{s}\psi & \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi + \mathrm{c}\phi\mathrm{c}\psi & \mathrm{c}\phi\mathrm{s}\theta\mathrm{s}\psi - \mathrm{s}\phi\mathrm{c}\psi \\ -\mathrm{s}\theta & \mathrm{s}\phi\mathrm{c}\theta & \mathrm{c}\phi\mathrm{c}\theta \end{bmatrix}, \end{aligned} \tag{21}$$

where the multiplying order must be consistent over the whole time since the commutative property of rotation matrices does not hold, and there are six distinct combinations with different physics meanings. Also, $R^{-1} = R^{\mathrm{T}}$ since $R \in \mathrm{SO}(3)$, i.e., rotation matrices belong to the special orthogonal group.

B. Angular Velocity and Rate Change of Euler Angles

Successive transformations of (18), (19), and (20) imply ϕ, θ, and ψ are not on the same coordinate frame. Therefore, to calculate the rate change $\left[\dot{\phi}, \dot{\theta}, \dot{\psi}\right]^{\mathrm{T}}$ from the angular velocity $\Omega = [p, q, r]^{\mathrm{T}}$, the following transform is necessary:

$$\Omega = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + R_z \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_z R_y \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} = T \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \tag{22}$$

Conversely, the angular velocity Ω can be calculated from the rate change $\left[\dot{\phi}, \dot{\theta}, \dot{\psi}\right]^{\mathrm{T}}$ using:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T^{-1}\Omega, \tag{23}$$

where T^{-1} is given by

$$T^{-1} = \begin{bmatrix} 1 & \mathrm{s}\phi\mathrm{t}\theta & \mathrm{c}\phi\mathrm{t}\theta \\ 0 & \mathrm{c}\phi & -\mathrm{s}\phi \\ 0 & \frac{\mathrm{s}\phi}{\mathrm{c}\theta} & \frac{\mathrm{c}\phi}{\mathrm{c}\theta} \end{bmatrix}. \tag{24}$$

C. Derivation of Quadrotor Dynamics

The derivation of quadrotor dynamics starts from expanding (23) for the dynamics of the angular velocity:

$$\begin{cases} \dot{\phi} = p + r(\mathrm{c}\phi\mathrm{t}\theta) + q(\mathrm{s}\phi + \mathrm{t}\theta) \\ \dot{\theta} = q(\mathrm{c}\phi) - r(\mathrm{s}\phi) \\ \dot{\psi} = r\frac{\mathrm{c}\phi}{\mathrm{c}\theta} + q\frac{\mathrm{s}\phi}{\mathrm{c}\theta} \end{cases}. \tag{25}$$

The coordinate transformation of the velocity is given by

$$v_I = Rv_B, \tag{26}$$

where $v_B = [u, v, w]^{\mathrm{T}}$ is the velocity on the body-fixed frame, and $v_I = [\dot{x}, \dot{y}, \dot{z}]^{\mathrm{T}}$ is the velocity on the inertial frame. By expanding (26), the dynamics of the velocity are derived:

$$\begin{cases} \dot{x} = w(\mathrm{s}\phi\mathrm{s}\psi + \mathrm{c}\phi\mathrm{c}\psi\mathrm{s}\theta) - v(\mathrm{c}\phi\mathrm{s}\psi - \mathrm{c}\psi\mathrm{s}\phi\mathrm{s}\theta) \\ \qquad +u(\mathrm{c}\psi\mathrm{c}\theta) \\ \dot{y} = v(\mathrm{c}\phi\mathrm{c}\psi + \mathrm{s}\phi\mathrm{s}\psi\mathrm{s}\theta) - w(\mathrm{c}\psi\mathrm{s}\phi - \mathrm{c}\phi\mathrm{s}\psi\mathrm{s}\theta) \\ \qquad +u(\mathrm{c}\theta\mathrm{s}\psi) \\ \dot{z} = w(\mathrm{c}\phi\mathrm{c}\theta) - u(\mathrm{s}\theta) + v(\mathrm{c}\theta\mathrm{s}\phi) \end{cases}. \tag{27}$$

Next, the relationship between the torque and angular velocity are described by Euler's equation of motion:

$$J\dot{\Omega} + \Omega \times J\Omega = \tau, \tag{28}$$

where $\tau = [\tau_x, \tau_y, \tau_z]^{\mathrm{T}}$ is the torque, and $\Omega = \begin{bmatrix} p, & q, & r \end{bmatrix}^{\mathrm{T}}$ is the angular velocity, both of them are on the body-fixed frame, and $J = \mathrm{diag}(J_x, J_y, J_z) \in \mathbb{R}^{9\times 9}$ is the inertia matrix. Additionally, the torque τ comprises the control input $\tau_b = [\tau_{b,x}, \tau_{b,y}, \tau_{b,z}]^{\mathrm{T}}$ and disturbance torque $\tau_w = [\tau_{w,x}, \tau_{w,y}, \tau_{w,z}]^{\mathrm{T}}$:

$$\tau = \tau_b + \tau_w. \tag{29}$$

By expanding (28) and (29), the dynamics of the torque are obtained:

$$\begin{cases} \tau_{b,x} + \tau_{w,x} = \dot{p}J_x - qrJ_y + qrJ_z \\ \tau_{b,y} + \tau_{w,y} = \dot{q}J_y + prJ_x - prJ_z \\ \tau_{b,z} + \tau_{w,z} = \dot{r}J_z - pqJ_x + pqJ_y \end{cases}. \tag{30}$$

For the dynamics of force, Newton's second law on the rotating frame is formulated as

$$m(\Omega \times v_B + \dot{v}_B) = f_B, \tag{31}$$

where m is the mass. Besides that, the net force $f_B \in \mathbb{R}^3$ on the body-fixed frame consists of gravity term, thrust term, and disturbance term $f_w = [f_{w,x}, f_{w,y}, f_{w,z}]^{\mathrm{T}}$:

$$f_B = \underbrace{R^{\mathrm{T}}mge_3}_{\text{gravity}} - \underbrace{f_c b_3}_{\text{thrust}} + \underbrace{f_w}_{\text{disturbance}}, \tag{32}$$

where g is the gravitational acceleration, $f_c \in \mathbb{R}$ is the collective thrust in the b_3 direction, and $e_3 = b_3 = [0, 0, 1]^{\mathrm{T}}$. Expanding (31) and (32) results in

$$\begin{cases} -mg(\mathrm{s}\theta) = m(\dot{u} + qw - rv) + f_{b,x} \\ mg(\mathrm{c}\theta\mathrm{s}\phi) = m(\dot{v} - pw + ru) + f_{b,y} \\ mg(\mathrm{c}\theta\mathrm{c}\phi) - f_c = m(\dot{w} + pv - qu) + f_{b,z} \end{cases}. \tag{33}$$

Finally, the full dynamics of the quadrotor are acquired by rearranging and combining the result from (25), (27), (30), and (33):

$$\mathbf{f}(\mathbf{x}, \mathbf{w}, \mathbf{u}) = \begin{cases} \dot{\phi} = p + r(\mathrm{c}\phi\mathrm{t}\theta) + q(\mathrm{s}\phi\mathrm{t}\theta) \\ \dot{\theta} = q(\mathrm{c}\phi) - r(\mathrm{s}\phi) \\ \dot{\psi} = r\frac{\mathrm{c}\phi}{\mathrm{c}\theta} + q\frac{\mathrm{s}\phi}{\mathrm{c}\theta} \\ \dot{p} = \frac{J_y - J_z}{J_x}rq + \frac{\tau_{b,x} + \tau_{w,x}}{J_x} \\ \dot{q} = \frac{J_z - J_x}{J_y}pr + \frac{\tau_{b,y} + \tau_{w,y}}{J_y} \\ \dot{r} = \frac{J_x - J_y}{J_z}pq + \frac{\tau_{b,z} + \tau_{w,z}}{J_z} \\ \dot{u} = rv - qw - g(\mathrm{s}\theta) + \frac{f_{w,x}}{m} \\ \dot{v} = pw - ru + g(\mathrm{s}\phi\mathrm{c}\theta) + \frac{f_{w,y}}{m} \\ \dot{w} = qu - pv + g(\mathrm{c}\theta\mathrm{c}\phi) + \frac{-f_c + f_{w,z}}{m} \\ \dot{x} = w(\mathrm{s}\phi\mathrm{s}\psi + \mathrm{c}\phi\mathrm{c}\psi\mathrm{s}\theta) \\ \qquad -v(\mathrm{c}\phi\mathrm{s}\psi - \mathrm{c}\psi\mathrm{s}\phi\mathrm{s}\theta) + u(\mathrm{c}\psi\mathrm{c}\theta) \\ \dot{y} = v(\mathrm{c}\phi\mathrm{c}\psi + \mathrm{s}\phi\mathrm{s}\psi\mathrm{s}\theta) \\ \qquad -w(\mathrm{c}\psi\mathrm{s}\phi - \mathrm{c}\phi\mathrm{s}\psi\mathrm{s}\theta) + u(\mathrm{c}\theta\mathrm{s}\psi) \\ \dot{z} = w(\mathrm{c}\phi\mathrm{c}\theta) - u(\mathrm{s}\theta) + v(\mathrm{c}\theta\mathrm{s}\phi) \end{cases}, \tag{34}$$

which requires linearization before applying H_∞ control.

D. Design of the Quadrotor H_∞ Controller

The state vector $\mathbf{x}$, disturbance vector $\mathbf{w}$, and control input vector $\mathbf{u}$ are defined as

$$\mathbf{x} = \left[\phi, \theta, \psi, p, q, r, u, v, w, x, y, z\right]^{\mathrm{T}} \tag{35}$$

$$\mathbf{w} = \left[f_{w,x}, f_{w,y}, f_{w,z}, \tau_{w,x}, \tau_{w,y}, \tau_{w,z}\right]^{\mathrm{T}} \tag{36}$$

$$\mathbf{u} = \left[f_c, \tau_{b,x}, \tau_{b,y}, \tau_{b,z}\right]^{\mathrm{T}}. \tag{37}$$

The state transition matrix A is obtained by computing the Jacobian matrix of (34):

$$A = \left.\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{w}, \mathbf{u})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}(t)}. \tag{38}$$

To pursue the best performance, the linearization is updated constantly. Also, due to the singularity of the Euler angles (i.e., gimbal lock), the roll angle has a restriction of $|\phi| < 90°$. Next, deriving the input matrix B_1 of the disturbance $\mathbf{w}$ gives:

$$B_1 = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{w}, \mathbf{u})}{\partial \mathbf{w}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{J_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{J_z} \\ \frac{1}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{39}$$

which appears to be a constant matrix and the same for the input matrix B_2 of the control input $\mathbf{u}$:

$$B_2 = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{w}, \mathbf{u})}{\partial \mathbf{u}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{J_x} & 0 & 0 \\ 0 & 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & 0 & \frac{1}{J_z} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{40}$$

The observation matrix C_1 of the penalty error $\mathbf{z}$ is designed as

$$\begin{aligned} C_1 &= [c_{ij}] \\ c_{1:10,3:12} &= \mathrm{diag}(\sigma_\psi, \sigma_p, \sigma_q, \sigma_r, \sigma_u, \sigma_v, \sigma_w, \sigma_x, \sigma_y, \sigma_z) \\ c_{1:10,1:2} &= 0 \\ c_{11:14,1:12} &= 0, \end{aligned} \tag{41}$$

where $(\sigma_\psi, ..., \sigma_z)$ are the state's penalty weights, emphasizing the controller's strength to regulate the penalty error to zero. The roll and pitch (i.e., ϕ and θ) are absent in (41) since they are affected by the translational control indirectly. The penalty input matrix D_{12} of the control input $\mathbf{u}$ is given by

$$D_{12} = \begin{bmatrix} 0_{10\times 4} \\ \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \end{bmatrix}_{14\times 4}. \tag{42}$$

Additionally, since the gravity term from the dynamics (32) vanished after calculating the partial derivatives, the following feedforward control term is designed for compensation:

$$\mathbf{u}_{\mathrm{ff}} = \left[\ \left\langle -(-mge_3), Re_3 \right\rangle, 0, 0, 0\ \right]^{\mathrm{T}}, \tag{43}$$

where $\langle \cdot, \cdot \rangle$ is the dot product. Combining (11) and (43) yields:

$$\mathbf{u} = -B_2^{\mathrm{T}} X \mathbf{x} + \mathbf{u}_{\mathrm{ff}}. \tag{44}$$

Finally, given the desired state (i.e., the setpoint) $\mathbf{x}_{\mathrm{d}}$, the complete control input can be obtained by rewriting (44) as

$$\mathbf{u} = -B_2^{\mathrm{T}} X \left(\mathbf{x} - \mathbf{x}_{\mathrm{d}}\right) + \mathbf{u}_{\mathrm{ff}}. \tag{45}$$

E. Thrust Allocation

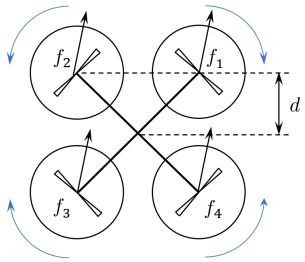

Fig. 3. Geometric configuration of the quadrotor

Fig. 3 shows the geometric configuration of the quadrotor model. The thrust allocation is given by

$$\begin{bmatrix} f_c \\ \tau_{b,x} \\ \tau_{b,y} \\ \tau_{b,z} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -d & d & d & -d \\ d & d & -d & -d \\ -c_{\tau,f} & c_{\tau,f} & -c_{\tau,f} & c_{\tau,f} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}, \tag{46}$$

which describes the conversion between the collective thrust value f_c, torque vector $\tau_b = [\tau_{b,x}, \tau_{b,y}, \tau_{b,z}]^{\mathrm{T}}$, and all individual motor thrusts $[f_1, \cdots, f_4]^{\mathrm{T}}$, where $c_{\tau,f}$ is the propeller coefficient, and d is the quadrotor length parameter.

IV. Simulation Results

The parameters of the drone are selected to resemble the F450 quadrotor as follows:

$$J = \mathrm{diag}(0.01466,\ 0.01466,\ 0.2848)\mathrm{kg}\cdot\mathrm{m}^2,$$

$$m = 1\mathrm{kg},\ d = 0.16\mathrm{m},\ c_{\tau,f} = 0.01\mathrm{m}.$$

The parameters of the H_∞ controller are chosen as

$$\sigma_\psi = 125,\ \sigma_p = 10,\ \sigma_q = 10,\ \sigma_r = 25,\ \sigma_u = 50,$$

$$\sigma_v = 50,\ \sigma_w = 100,\ \sigma_x = 200,\ \sigma_y = 200,\ \sigma_z = 160.$$

The initial conditions of the simulation are set as

$$\mathbf{x}(0) = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^{\mathrm{T}}.$$

The disturbances to the quadrotor, as shown in Fig. 4, are modeled as randomized Gaussian noise with the standard deviations given by

$$\sigma_{f_w} = 1.0\mathrm{kg} \cdot \mathrm{m} \cdot \mathrm{s}^{-2},\ \sigma_{\tau_w} = 0.5\mathrm{kg} \cdot \mathrm{m}^2 \cdot \mathrm{s}^{-2}.$$

The desired state vector is defined as

$$\mathbf{x}_{\mathrm{d}} = [0, 0, \psi_{\mathrm{d}}, p_{\mathrm{d}}, q_{\mathrm{d}}, r_{\mathrm{d}}, u_{\mathrm{d}}, v_{\mathrm{d}}, w_{\mathrm{d}}, x_{\mathrm{d}}, y_{\mathrm{d}}, z_{\mathrm{d}}]^{\mathrm{T}},$$

where a circular motion varies the desired translational states in the xy-direction and a linear motion in the z-direction:

$$\begin{cases} x_{\mathrm{d}} = 1.5\cos(0.25\pi t),\ u_{\mathrm{d}} = -1.5\sin(0.25\pi t) \\ y_{\mathrm{d}} = 1.5\sin(0.25\pi t),\ v_{\mathrm{d}} = 1.5\cos(0.25\pi t) \\ x_{\mathrm{d}} = 0.05t,\ w_{\mathrm{d}} = 0.05 \end{cases},$$

and the desired rotational states are given as

$$\psi_{\mathrm{d}} = 0,\ [p_{\mathrm{d}},\ q_{\mathrm{d}},\ r_{\mathrm{d}}]^{\mathrm{T}} = [0,\ 0,\ 0]^{\mathrm{T}}.$$

The simulation outcomes are presented in Fig. 5, 6, 7, and 8. The result shows that the quadrotor can precisely track the position and velocity trajectory under disturbed environments. As the only controllable quantity of the Euler angles, the yaw angle converged to the desired value with only $\pm 2^\circ$ of fluctuation. The H_∞ norm r_{lb} and $r_{\mathbf{x}}$, as shown in 9, are re-computed at every new iteration as the linearization keeps changing the system characteristic. Also, $r_{\mathbf{x}}$ is indeed always larger than r_{lb} as expected. Fig. 10 shows the CARE residual by the SDA solution, in which the value is very small, and the worst case is no larger than 10^{-7}. Finally, the source code is published on GitHub for the public and can be accessed via https://github.com/shengwen-tw/quadrotor-h-infty-ctrl-simulator (accessed on 26 May 2022).

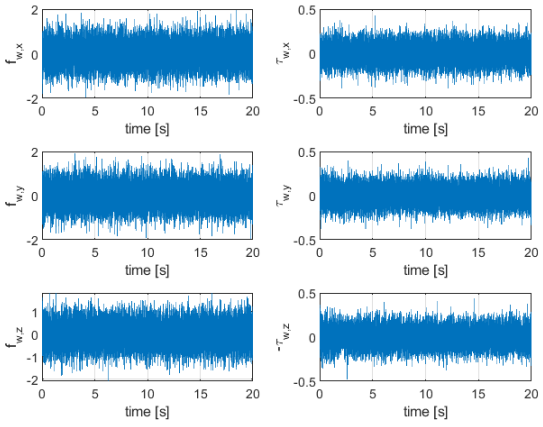

Fig. 4. Force and torque disturbances

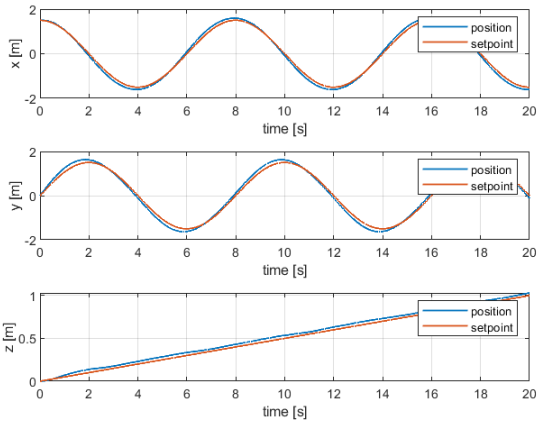

Fig. 5. Position

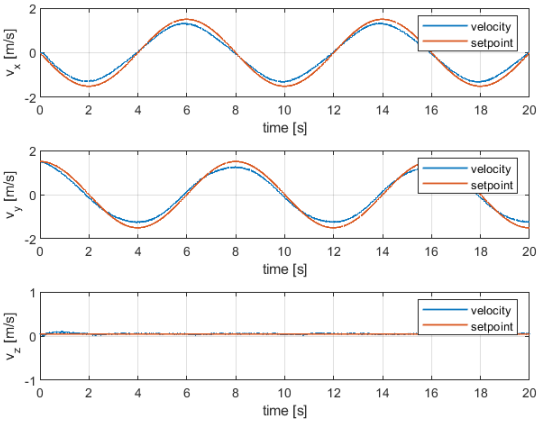

Fig. 6. Velocity

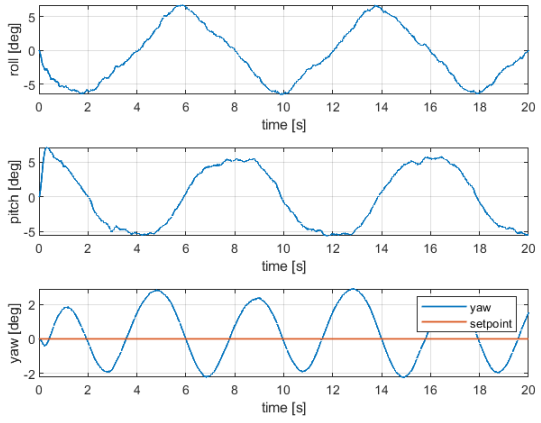

Fig. 7. Attitude (Euler angles)

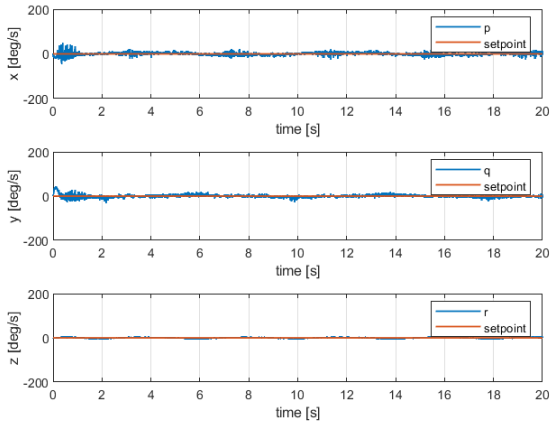

Fig. 8. Angular velocity

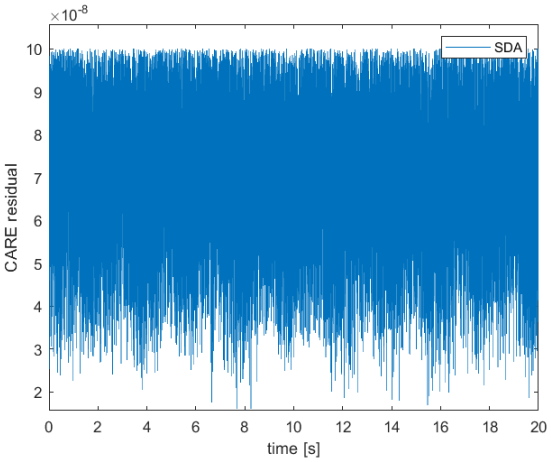

Fig. 10. CARE residual by the SDA solution

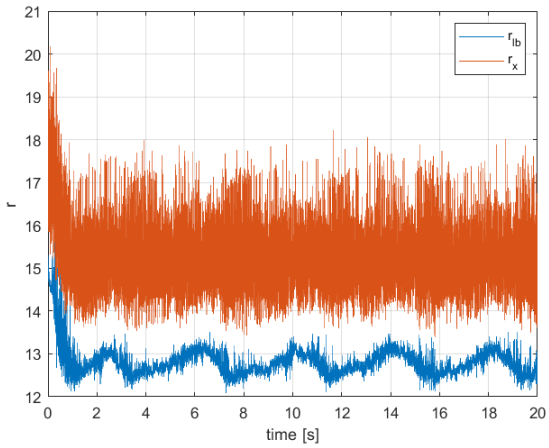

Fig. 9. The H_∞ norm $r_{\mathbf{x}}$ and r_{lb} of the H_∞ controller

V. Conclusion

This paper presents a robust quadrotor controller based on the state-feedback H_∞ control and parameterizes the quadrotor dynamics with Euler angles. A bisection method based on the structure-preserving doubling algorithm (SDA), a state-of-the-art CARE solver, is proposed to solve the optimal control input. The simulation result shows that the quadrotor can track the trajectory accurately under the given disturbances. However, the singularity of Euler angles at $\phi = \pm 90^\circ$ limits the maneuver of the quadrotor. Therefore, future work includes replacing the parameterization of the rotational dynamics with quaternion. Also, techniques like the state-dependent Riccati equation (SDRE) [17] can be considered to replace the linearization of the nonlinear dynamics for further improvement in accuracy.

Acknowledgment

The authors are grateful to professor Wen-Wei Lin from National Yang Ming Chiao Tung University for his inspiration to the topic.

References

[1] A. L. Salih, M. Moghavvemi, H. A. Mohamed, and K. S. Gaeid, "Modelling and pid controller design for a quadrotor unmanned air vehicle," in *Proc. IEEE Int. Conf. Autom. Quality Testing Robot. Conf.*, vol. 1, 2010, pp. 1–5.

[2] J. Li and Y. Li, "Dynamic analysis and pid control for a quadrotor," in *Proc. IEEE Int. Conf. Mechatronics Autom. Conf.*, 2011, pp. 573–578.

[3] F. Sabatino, "Quadrotor control: modeling, nonlinear control design, and simulation," Master's thesis, 2015.

[4] P. Foehn and D. Scaramuzza, "Onboard state dependent lqr for agile quadrotors," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 6566–6572.

[5] A. Laub, "A schur method for solving algebraic riccati equations," *IEEE Trans. Autom. Contr.*, vol. 24, no. 6, pp. 913–921, 1979.

[6] R. Patel, Z. Lin, and P. Misra, "Computation of invariant subspaces corresponding to stable eigenvalues of hamiltonian matrices," in *Proc. Amer. Control Conf.*, 1990, pp. 2565–2571.

[7] P. Benner, V. Mehrmann, and H. Xu, "A new method for computing the stable invariant subspace of a real hamiltonian matrix," *J. Comput. Appl. Math.*, vol. 86, no. 1, pp. 17–43, 1997.

[8] R. Byers, "A hamiltonian qr algorithm," *SIAM J. Sei. Stat. Comp.*, vol. 7, no. 1, pp. 212–229, 1986.

[9] P. Benner, V. Mehrmann, and H. Xu, "A numerically stable, structure preserving method for computing the eigenvalues of real hamiltonian or symplectic pencils," *Numerische Mathematik*, vol. 78, no. 3, pp. 329–358, 1998.

[10] E.-W. Chu, H.-Y. Fan, and W.-W. Lin, "A structure-preserving doubling algorithm for continuous-time algebraic riccati equations," *Linear Algebra Appl.*, vol. 396, pp. 55–80, 2005.

[11] R. Byers, "Solving the algebraic riccati equation with the matrix sign function," *Linear Algebra Appl.*, vol. 85, pp. 267–279, 1987.

[12] W.-W. Lin, C.-S. Wang, and Q.-F. Xu, "On the computation of the optimal h-infinity norms for two feedback control problems," *Linear Algebra Appl.*, vol. 287, no. 1-3, pp. 223–255, 1999.

[13] J. Doyle, K. Glover, P. Khargonekar, and B. Francis, "State-space solutions to standard h2 and h-infity control problems," in *Proc. Amer. Control Conf.*, 1988, pp. 1691–1696.

[14] S. Boyd, V. Balakrishnan, and P. Kabamba, "A bisection method for computing the h-infinity norm of a transfer matrix and related problems," *Math. Control, Signals, Syst.*, vol. 2, no. 3, pp. 207–219, 1989.

[15] D. F. Enns, "Model reduction with balanced realizations: an error bound and a frequency weighted generalization," in *Proc. 23rd IEEE Conf. Decision and Control*, 1984, pp. 127–132.

[16] K. Glover, "All optimal hankel-norm approximations of linear multivariable systems and their l-infinity error bounds," *Int. J. Control*, vol. 39, no. 6, pp. 1115–1193, 1984.

[17] J. R. Cloutier, "State-dependent riccati equation techniques: an overview," in *Proc. Amer. Control Conf.*, vol. 2, 1997, pp. 932–936.