Drone Network Decomposition

Ethan LoCicero

November 7, 2024

Our controller converts roll, pitch, yaw and/or their rates to torques in the same dimensions. From Brogliado, we know that Lagrangian systems are passive from torque inputs to angular velocity outputs[1]. That means from torques to roll, pitch, and yaw rates, the drone must be passive. If we ignore motor dynamics and say that the controller can provide torque actuation exactly, and then we could design the controller to be strictly passive to ensure stability. This is visualized in Figure 1.

However, the motors introduce non-trivial dynamics that must be accounted for, and these likely violate passivity. In reality, the drone can be decomposed as visualized in Figure 2. Here, we see that the controller mappings are the same, but the commanded torques, $\overline{\tau}_{(\cdot)}$ (and commanded lift force, $\overline{F}_L$) are converted to squares of commanded rotor speeds using the matrix Γ^{-1}, where

$$\Gamma = \begin{bmatrix} k & k & k & k \\ 0 & lk & 0 & -lk \\ lk & 0 & -lk & 0 \\ -b & b & -b & b \end{bmatrix}$$

The motor dynamics, $\mathcal{M}$, then convert the commanded motor speeds, $\overline{\omega}$, to actual motor speeds, ω, before Ω coverts these motor speeds squared back to actual torques (and lift force) that effect the drone frame dynamics, $\mathcal{G}$. (Notice, the lift force doesn't effect roll, pitch, and yaw, so it's excluded from the drone inputs, and x,y,z position and their rates are also currently excluded). The motor dynamics are given by

$$\begin{aligned} v &= Ri + L\frac{di}{dt} + k_e\omega \\ J\dot{\omega} &= -k_T i - b\omega^2, \end{aligned}$$

where v is voltage, i is current, ω is rotor speed, R is resistance, L is inductance, k_e is the back-EMF coefficient, k_T is the torque constant, J is the inertia, and b is the rotor drag constant. The voltage is restricted as $v \in [0, v_{\max}(t)]$, where $v_{\max}(t) \leq 12.6$. This is because the minimum voltage applied to the motors is 0, and the maximum is the instantaneous maximum voltage of the battery, which can be charged to at most $12.6V$. We want the voltage to equal the demanded rotor speed when the motors are at equilibrium, which is satisfied by $v = \frac{Rb}{k_T}\overline{\omega}^2 + k_e\overline{\omega}$, where $\overline{w}$ is the commanded rotor speed (as opposed to actual). Substituting the relation for voltage and letting $x^T = [i \ \omega]$, $u = \overline{\omega}^2$, and $y = \omega^2$, we have

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -\frac{R}{L} & -\frac{k_e}{L} \\ \frac{k_T}{J} & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & -\frac{b}{J} \end{bmatrix} x^2 + \frac{1}{L}\mathrm{sat}_{[0,v_{\max}(t)]}\left(\begin{bmatrix} \frac{Rb}{k_T} \\ 0 \end{bmatrix} u + \begin{bmatrix} k_e \\ 0 \end{bmatrix} \sqrt{u} \right) \\ y &= \begin{bmatrix} 0 & 1 \end{bmatrix} x^2. \end{aligned}$$

The matrices Γ and Γ^{-1} are linear, so this network fits into the framework of Vidyasagar's Network Dissipativity Theorem (VNDT). As before, the drone frame can be viewed as passive since it is a Lagrangian system, provided we consider the control inputs to be the roll pitch and yaw rates. (If the control inputs are just the angles themselves, we would need a different dissipativity characterization of the frame). The motor dynamics have nonlinear input and output functions, so we will need a clever way of analyzing them, possibly using Reza, Amy, and Frank's triangulation method, or possibly using Ethan's data methods.

Now, consider the case where the controllers have access to roll, pitch, and yaw rates as well as the rotor speeds (via back-EMF measurement). Then Figure 2 looks more like Figure 3.

[1] I never found this result written out, but Leila references it all the time, and I didn't look that hard because even taking it as true, there were more pressing things to solve. But we need to find it eventually.

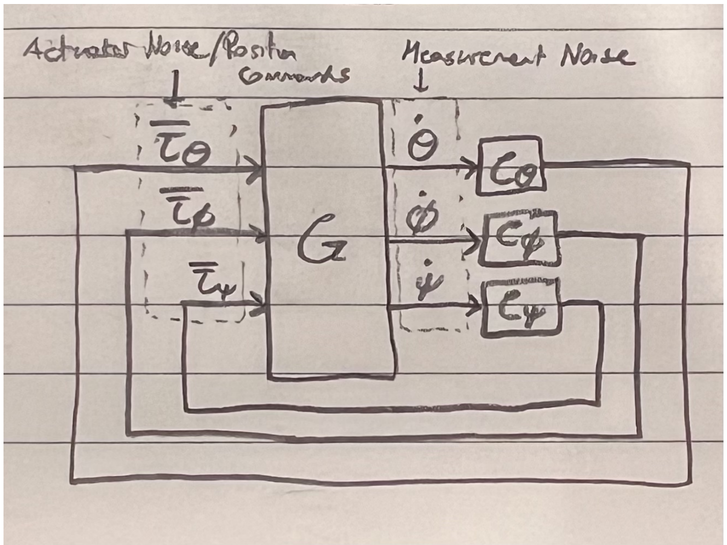

Figure 1: Drone in feedback with controller, neglecting motor dynamics. The overlines on τ's indicate the commanded torque. Roll, pitch, and yaw are θ, ϕ, and ψ, respectively, and $\mathcal{G}$ is the drone dynamics. (Note: where it says "Actuator noise/Position commands" and "Measurement noise", it means those exogenous signals are added there).

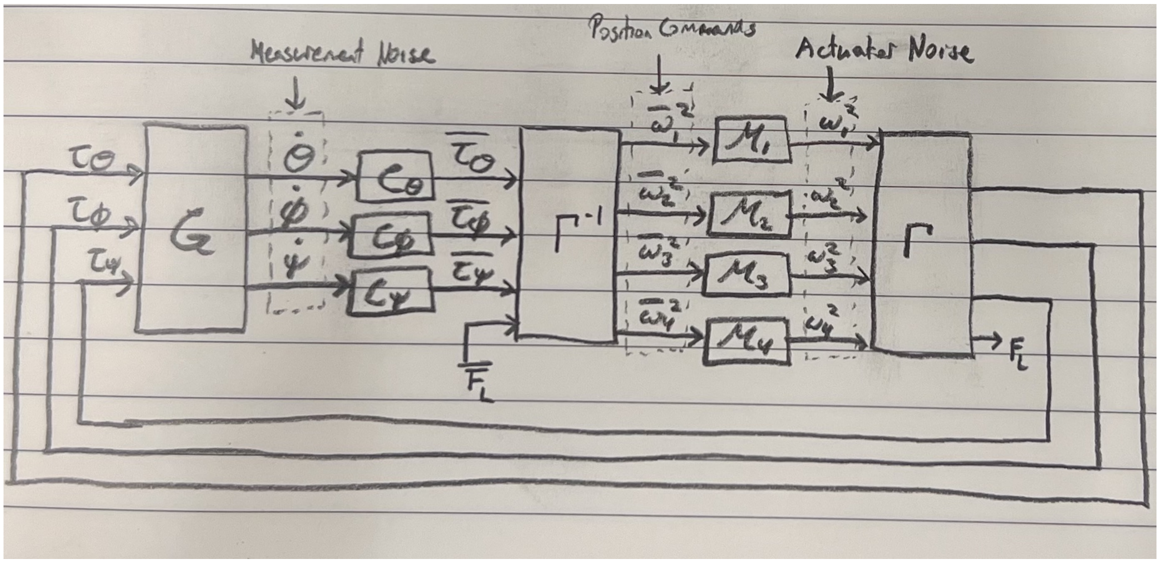

Figure 2: Drone decomposed into network with motor dynamics. (Note, where it says "Measurement noise", "Position commands", and "Actuator Noise", it means those exogenous signals are added in those places).

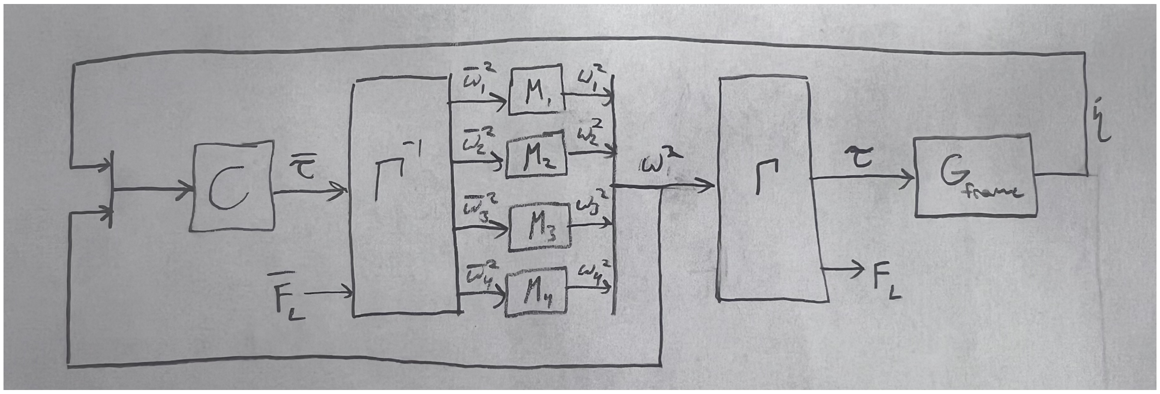

Figure 3: Drone decomposed into network with motor dynamics, with a controller that has access to roll, pitch, yaw rates and rotor speeds.

Converting Figure 3 into the form for VNDT, we have

$$
\begin{aligned}
G &= \operatorname{diag}(G_{\text{frame}},\, M,\, C) \\
M &= \operatorname{diag}(M_1,\, M_2,\, M_3,\, M_4) \\
\eta &= \begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix}, \qquad y = \begin{bmatrix} \dot{\eta} \\ \omega^2 \\ \overline{\tau} \end{bmatrix}, \qquad u = \begin{bmatrix} \tau \\ \overline{\omega}^2 \\ \dot{\eta} \\ \omega^2 \end{bmatrix} \qquad \tau = \begin{bmatrix} \tau_\theta \\ \tau_\phi \\ \tau_\psi \end{bmatrix} \qquad \omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \\
H &= \begin{bmatrix} 0 & \Gamma_0 & 0 \\ 0 & 0 & \Gamma_0^{-1} \\ I_3 & 0 & 0 \\ 0 & I_4 & 0 \end{bmatrix} \\
\Gamma_0 &= \begin{bmatrix} 0 & lk & 0 & -lk \\ lk & 0 & -lk & 0 \\ -b & b & -b & b \end{bmatrix}, \Gamma_0^{-1} = \begin{bmatrix} 0 & \frac{1}{2lk} & -\frac{1}{4b} \\ \frac{1}{2lk} & 0 & \frac{1}{4b} \\ 0 & -\frac{1}{2lk} & -\frac{1}{4b} \\ -\frac{1}{2lk} & 0 & \frac{1}{4b} \end{bmatrix} \\
Q &= \begin{bmatrix} Q_f & 0 & 0 \\ 0 & Q_M & 0 \\ 0 & 0 & Q_C \end{bmatrix} \\
S &= \begin{bmatrix} S_f & 0 & 0 & 0 \\ 0 & S_M & 0 & 0 \\ 0 & 0 & S_c^1 & S_c^2 \end{bmatrix} \\
R &= \begin{bmatrix} R_f & 0 & 0 & 0 \\ 0 & R_M & 0 & 0 \\ 0 & 0 & R_c^{11} & R_c^{12} \\ 0 & 0 & R_c^{12^T} & R_c^{22} \end{bmatrix} \\
SH &= \begin{bmatrix} 0 & S_f\Gamma_0 & 0 \\ 0 & 0 & S_M\Gamma_0^{-1} \\ S_c^1 & S_c^2 & 0 \end{bmatrix} \\
H^TRH &= \begin{bmatrix} R_c^{11} & R_c^{12} & 0 \\ R_c^{12^T} & R_c^{22} + \Gamma_0^T R_f \Gamma_0 & 0 \\ 0 & 0 & \Gamma_0^{-T} R_M \Gamma_0^{-1} \end{bmatrix} \\
\overline{Q} &= Q + SH + H^TS^T + H^TRH = \begin{bmatrix} Q_f + R_c^{11} & S_f\Gamma_0 + R_c^{12} & S_c^{1^T} \\ * & Q_M + R_c^{22} + \Gamma_0^T R_f \Gamma_0 & S_M\Gamma_0^{-1} + S_c^{2^T} \\ * & * & Q_c + \Gamma_0^{-T} R_M \Gamma_0^{-1} \end{bmatrix}.
\end{aligned}
$$

How did we get that? First, we pick the order of systems: frame, then motors, then controller. We pick the inputs and outputs correspondingly. Then to get H, we use the formula $u = Hy$. So the each u component should be some linear combination of the y components. For the first, $\tau = \Gamma_0\omega^2$, which is $u_1 = \Gamma_0 y_2$. The idea of Γ_0 (Γ_0^{-1}) is that it removes the row (column) that corresponds to the lift force, which is not currently incorporated in the controller or frame dynamics, since we're just working on attitude control. Repeating this process for each component of u yields H. The dimensions of Q_c, S_c, and R_c are a little tricky, due to the double feedback to C from different parts of the network. Since it has one output, Q_c is blockwise 1×1, and S_c has one block row. Since C has two inputs, S_c has two block columns, and R_c is blockwise 2×2. Then applying VNDT gives $\overline{Q}$, which must be negative definite to ensure stability.

The important open question is whether there exists a choice of Q_c, S_c, and R_c that make $\overline{Q} < 0$. Presuming the drone is passive, we have $Q_f = 0$, $S_f = 1$, $R_f = 0$, therefore,

$$
\overline{Q} = \begin{bmatrix} R_c^{11} & \Gamma_0 + R_c^{12} & S_c^{1^T} \\ * & Q_M + R_c^{22} & S_M\Gamma_0^{-1} + S_c^{2^T} \\ * & * & Q_c + \Gamma_0^{-T} R_M \Gamma_0^{-1} \end{bmatrix}.
$$

Immediately, we need $R_c^{11} \leq 0$, which is possible but somewhat concerning. We might be looking at an excone here for the controller. Let's choose $R_c^{12} = -\Gamma_0$, $S_c^{1^T} = 0$, and $S_c^{2^T} = -S_M\Gamma_0^{-1}$, so that we have

$$
\overline{Q} = \begin{bmatrix} R_c^{11} & 0 & 0 \\ * & Q_M + R_c^{22} & 0 \\ * & * & Q_c + \Gamma_0^{-T} R_M \Gamma_0^{-1} \end{bmatrix}.
$$

Now let $R_c^{11} = -\epsilon$, $R_c^{22} = -Q_M - \epsilon$, and $Q_c = -\Gamma_0^{-T} R_M \Gamma_0^{-1} - \epsilon$. Then we have $\overline{Q} = -\epsilon I < 0$ (which implies network stability) with

$$\begin{bmatrix} Q_c & S_c \\ * & R_c \end{bmatrix} = \begin{bmatrix} -\Gamma_0^{-T} R_M \Gamma_0^{-1} & 0 & -S_M \Gamma_0^{-1} \\ * & 0 & -\Gamma_0 \\ * & * & -Q_M \end{bmatrix} - \epsilon I. \tag{1}$$

The matrix in Equation 1 needs to be indefinite or positive definite to be realizable by some controller. Likewise,

$$\begin{bmatrix} Q_M & S_M \\ * & R_M \end{bmatrix}$$

must be indefinite or positive definite to be realizable. Suppose $Q_M < 0$, $R_M > 0$, and $S_M = 0$, which satisfies indefiniteness (this is a small gain characterization, which would be easy to find for the motor). Then $-\Gamma_0^{-T} R_M \Gamma_0^{-1} < 0$ and $-Q_M > 0$, which means $Q_c < 0$ and $R_c^{22} > 0$. So the controller QSR is indefinite as well. Therefore, if the frame is passive and the motors are gain-bounded, then there is at least one dissipativity characterization of the controllers that makes the network input-output stable!