Supplemental Materials

Tao Du MIT CSAIL
Adriana Schulz MIT CSAIL
Bo Zhu MIT CSAIL
Bernd Bickel IST Austria
Wojciech Matusik MIT CSAIL

1 Derivation of state-space representation

We choose to use *manipulator form*, described in [Tedrake 2014] to derive the state-space representation of the multicopter system from its equations of motion:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}(\mathbf{q})\mathbf{u} \tag{1}$$

where $\mathbf{q}$ is defined as:

$$\mathbf{q} = \begin{bmatrix}\mathbf{p}\\ \mathbf{e}\end{bmatrix} \tag{2}$$

Recall that $\mathbf{p}$ denotes the copter position and $\mathbf{e} = [\phi, \theta, \psi]^\top$ is an Euler angle vector representing roll, pitch and yaw.

In this section we give the definition of $\mathbf{H}$, $\mathbf{C}$, $\mathbf{G}$ and $\mathbf{B}$ in the manipulator form. Our derivation is originated from [Joubert et al. 2015] but it applies to more general multicopters without the gimbal.

Euler angles In our implementation we choose to use the Euler angles in the following order to represent the rotation matrix $\mathbf{R}$ that maps from the body frame to the world frame:

$$\begin{aligned}\mathbf{R} &= \mathbf{R}_\psi\mathbf{R}_\theta\mathbf{R}_\phi\\ &= \begin{bmatrix} c_\psi & -s_\psi & 0\\ s_\psi & c_\psi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix} c_\theta & 0 & s_\theta\\ 0 & 1 & 0\\ -s_\theta & 0 & c_\theta\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & c_\phi & -s_\phi\\ 0 & s_\phi & c_\phi\end{bmatrix}\end{aligned} \tag{3}$$

where $c_a = \cos(a)$ and $s_a = \sin(a)$.

The angular velocity ω, defined in the body frame, can be related to the Euler angle rates $\dot{\mathbf{e}} = [\dot\phi, \dot\theta, \dot\psi]^\top$ below:

$$\begin{aligned}\omega &= \begin{bmatrix}\dot\phi\\0\\0\end{bmatrix} + \mathbf{R}_\phi^\top\begin{bmatrix}0\\\dot\theta\\0\end{bmatrix} + \mathbf{R}_\phi^\top\mathbf{R}_\theta^\top\begin{bmatrix}0\\0\\\dot\psi\end{bmatrix}\\ &= \begin{bmatrix}\dot\phi\\0\\0\end{bmatrix} + \begin{bmatrix}0\\c_\phi\dot\theta\\-s_\phi\dot\theta\end{bmatrix} + \mathbf{R}_\phi^\top\begin{bmatrix}-s_\theta\dot\psi\\0\\c_\theta\dot\psi\end{bmatrix}\\ &= \begin{bmatrix}\dot\phi\\0\\0\end{bmatrix} + \begin{bmatrix}0\\c_\phi\dot\theta\\-s_\phi\dot\theta\end{bmatrix} + \begin{bmatrix}-s_\theta\dot\psi\\s_\phi c_\theta\dot\psi\\c_\phi c_\theta\dot\psi\end{bmatrix}\\ &= \begin{bmatrix}1 & 0 & -s_\theta\\ 0 & c_\phi & s_\phi c_\theta\\ 0 & -s_\phi & c_\phi c_\theta\end{bmatrix}\begin{bmatrix}\dot\phi\\\dot\theta\\\dot\psi\end{bmatrix}\end{aligned} \tag{4}$$

We use $\mathbf{L}$ to denote the matrix on the right so $\omega = \mathbf{L}\dot{\mathbf{e}}$, and therefore $\dot\omega = \dot{\mathbf{L}}\dot{\mathbf{e}} + \mathbf{L}\ddot{\mathbf{e}}$, where $\dot{\mathbf{L}}$ is:

$$\dot{\mathbf{L}} = \begin{bmatrix}0 & 0 & -c_\theta\dot\theta\\ 0 & -s_\phi\dot\phi & c_\phi c_\theta\dot\phi - s_\phi s_\theta\dot\theta\\ 0 & -c_\phi\dot\phi & -s_\phi c_\theta\dot\phi - c_\phi s_\theta\dot\theta\end{bmatrix} \tag{5}$$

Now it's time to relate everything to the manipulator matrices. For thrust we have:

$$m\ddot{\mathbf{p}} = m\mathbf{g} + \mathbf{R}\mathbf{M}_f\mathbf{u} \tag{6}$$

For torque we have:

$$\mathbf{J}(\dot{\mathbf{L}}\dot{\mathbf{e}} + \mathbf{L}\ddot{\mathbf{e}}) + (\mathbf{L}\dot{\mathbf{e}}) \times \mathbf{J}\mathbf{L}\dot{\mathbf{e}} = \mathbf{M}_t\mathbf{u} \tag{7}$$

Note that unlike equation (10) in [Joubert et al. 2015], our torque equation does not contain a rotation matrix on the left. This is because in their derivation they define ω to be the angular rate in the world frame, while we use ω to represent the body frame angular velocity. However, both world-frame and body-frame angular rates are linear on Euler angle rates, so they share similar representations.

Putting them together gives us the following manipulator matrices:

$$\begin{aligned}\mathbf{H}(\mathbf{q}) &= \begin{bmatrix}m\mathbf{I}_{3\times3} & \mathbf{O}_{3\times3}\\ \mathbf{O}_{3\times3} & \mathbf{J}\mathbf{L}\end{bmatrix}\\ \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) &= \begin{bmatrix}\mathbf{O}_{3\times3} & \mathbf{O}_{3\times3}\\ \mathbf{O}_{3\times3} & \mathbf{J}\dot{\mathbf{L}} + (\mathbf{L}\dot{\mathbf{e}})_\times\mathbf{J}\mathbf{L}\end{bmatrix}\\ \mathbf{G}(\mathbf{q}) &= \begin{bmatrix}-m\mathbf{g}\\ \mathbf{O}_{3\times1}\end{bmatrix}\\ \mathbf{B}(\mathbf{q}) &= \begin{bmatrix}\mathbf{R}\mathbf{M}_f\\ \mathbf{M}_t\end{bmatrix}\end{aligned} \tag{8}$$

where $\mathbf{I}_{3\times3}$ is a 3 by 3 identity matrix, and $\mathbf{O}_{m\times n}$ is an m by n zero matrix. An important consideration is that the manipulator matrices only depend on the Euler angles and Euler angular velocity. They don't rely on the position and linear velocity of the copter. This makes it possible to use LQR to control the copter position efficiently, as we don't need to redo Taylor expansion at different positions.

Recall that the relation between $\mathbf{x}$ in the state-space representation and $\mathbf{q}$ is:

$$\mathbf{x} = \begin{bmatrix}\mathbf{q}\\\dot{\mathbf{q}}\end{bmatrix}, \quad \dot{\mathbf{x}} = \begin{bmatrix}\dot{\mathbf{q}}\\\ddot{\mathbf{q}}\end{bmatrix} \tag{9}$$

Once we have the manipulator form, the state-space model can be rewritten as:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x},\mathbf{u})\\ &= \begin{bmatrix}\dot{\mathbf{q}}\\ \mathbf{H}^{-1}(\mathbf{q})(\mathbf{B}(\mathbf{q})\mathbf{u} - \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}))\end{bmatrix}\end{aligned} \tag{10}$$

2 Linear approximation

At a fixed point ($\mathbf{x}^*$, $\mathbf{u}^*$) such that $\mathbf{f}(\mathbf{x}^*, \mathbf{u}^*) = \mathbf{0}$, the linear approximation is given in [Tedrake 2014]:

$$\begin{aligned}\bar{\mathbf{x}} &= \mathbf{x} - \mathbf{x}^*\\ \bar{\mathbf{u}} &= \mathbf{u} - \mathbf{u}^*\\ \mathcal{A} &= \begin{bmatrix}\mathbf{O}_{6\times6} & \mathbf{I}_{6\times6}\\ \mathbf{H}^{-1}\sum_{i=1}^{n}\frac{\partial\mathbf{B}_i}{\partial\mathbf{q}}u_i^* & -\mathbf{H}^{-1}\mathbf{C}\end{bmatrix}\\ \mathcal{B} &= \begin{bmatrix}\mathbf{O}_{6\times n}\\ \mathbf{H}^{-1}\mathbf{B}\end{bmatrix}\\ \dot{\bar{\mathbf{x}}} &= \mathcal{A}\bar{\mathbf{x}} + \mathcal{B}\bar{\mathbf{u}}\end{aligned} \tag{11}$$

Here $\mathbf{B}_i$ is the i-th column of $\mathbf{B}$, and u_i^* is the i-th element in $\mathbf{u}^*$. Note that we omit the Jacobian of $\mathbf{G}$ as in our case it is constant.

The remaining thing is to compute the gradient of $\mathbf{B}$ over $\mathbf{q}$. Note that both $\mathbf{M}_f$ and $\mathbf{M}_t$ are independent from $\mathbf{q}$, and $\mathbf{R}$ only relies on $\mathbf{e}$, which greatly simplifies the computation. We first give the derivative of $\mathbf{R}$ over ϕ, θ, ψ:

$$\begin{aligned}
\frac{\partial \mathbf{R}}{\partial \phi} &= \mathbf{R}_\psi \mathbf{R}_\theta \begin{bmatrix} 1 & 0 & 0 \\ 0 & -s_\phi & -c_\phi \\ 0 & c_\phi & -s_\phi \end{bmatrix} \\
\frac{\partial \mathbf{R}}{\partial \theta} &= \mathbf{R}_\psi \begin{bmatrix} -s_\theta & 0 & c_\theta \\ 0 & 1 & 0 \\ -c_\theta & 0 & -s_\theta \end{bmatrix} \mathbf{R}_\phi \\
\frac{\partial \mathbf{R}}{\partial \psi} &= \begin{bmatrix} -s_\psi & -c_\psi & 0 \\ c_\psi & -s_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{R}_\theta \mathbf{R}_\phi
\end{aligned} \tag{12}$$

and the Jacobian of $\mathbf{B}_i$:

$$\begin{aligned}
\mathbf{B}_i &= \begin{bmatrix} \mathbf{R}(\mathbf{M}_f)_i \\ (\mathbf{M}_t)_i \end{bmatrix} \\
\frac{\partial \mathbf{B}_i}{\partial \mathbf{q}} &= \begin{bmatrix} \mathbf{O}_{3\times 3} & \frac{\partial \mathbf{R}}{\partial \phi}(\mathbf{M}_f)_i, \frac{\partial \mathbf{R}}{\partial \theta}(\mathbf{M}_f)_i, \frac{\partial \mathbf{R}}{\partial \psi}(\mathbf{M}_f)_i \\ \mathbf{O}_{3\times 3} & \mathbf{O}_{3\times 3} \end{bmatrix}
\end{aligned} \tag{13}$$

3 Choosing a fixed point

From $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}^*, \mathbf{u}^*) = \mathbf{0}$ we have:

$$\begin{aligned}
\mathbf{R}\mathbf{M}_f \mathbf{u}^* &= -m\mathbf{g} \\
\mathbf{M}_t \mathbf{u}^* &= \mathbf{0}
\end{aligned} \tag{14}$$

where $\mathbf{R}$ depends on the Euler angles only. The system above is generally under-determined as we allow $\mathbf{M}_f \mathbf{u}^*$ to rotate arbitrarily to align with the gravity. We first pick $\mathbf{u}^*$ by solving the following optimization problem:

$$\begin{aligned}
\min_{\mathbf{u}} \quad & \|\mathbf{u} - \mathbf{u}_0\|^2 \\
s.t. \quad & \mathbf{M}_t \mathbf{u} = \mathbf{0}
\end{aligned} \tag{15}$$

where $\mathbf{u}_0$ is the optimal thrust given by our optimization algorithm. We solve this convex problem by using CVX [Grant and Boyd 2014; Grant and Boyd 2008]. We then scale the optimal solution to get $\mathbf{u}^*$ such that $\mathbf{M}_f \mathbf{u}^*$ and $-m\mathbf{g}$ have the same length.

Once $\mathbf{u}^*$ is determined, $\mathbf{R}$ is computed from a rotation that rotates a vector around axis $\hat{\mathbf{v}}$ by angle α, which are defined below:

$$\begin{aligned}
\mathbf{v} &= \mathbf{M}_f \mathbf{u}^* \times (-m\mathbf{g}) \\
\hat{\mathbf{v}} &= \frac{\mathbf{v}}{\|\mathbf{v}\|} \\
\alpha &= \arccos\left(\frac{-m\mathbf{g}^\top \mathbf{M}_f \mathbf{u}^*}{\|\mathbf{M}_f \mathbf{u}^*\| \|m\mathbf{g}\|}\right)
\end{aligned} \tag{16}$$

The final $\mathbf{x}^*$ is:

$$\begin{bmatrix} \mathbf{p}^* \\ \mathbf{e}^* \\ \mathbf{O}_{6\times 1} \end{bmatrix} \tag{17}$$

where $\mathbf{p}^*$ can be any point and the Euler angles $\mathbf{e}^*$ are extracted from $\mathbf{R}$ directly.

4 Measurement

4.1 Mass and price of each component

We measure the weight and record the price of each component in the copter, including motors, connectors, battery, controlling board and rods. For parametric components like rods we measure its unit weight and compute its unit price.

Name	Mass(g)	Price(USD)
Motor	125	72
10-inch blade	9	20
14-inch blade	17	34
Battery	189	10
Connector	17	5
Controlling board	39	200
Carbon fiber rod (per meter)	59	74

Table 1: *Mass and price of each component.*

4.2 Motor and propeller properties

For a motor and propeller combination we measure the ratio λ between torque and thrust, the u-P_{ele} curve which shows the overall efficiency, the u-I curve, and the mapping from PWM control signal to thrust. We use RCBenchmark dynamometer [RCBenchmark 2016] to measure the thrust, torque, throttle, current and voltage directly from a motor and propeller combination, then fit them with properly chosen functions. The measurement data and fitting results can be found in Figure 1 and Figure 2.

References

GRANT, M., AND BOYD, S. 2008. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*. Springer, 95–110. http://stanford.edu/~boyd/graph_dcp.html.

GRANT, M., AND BOYD, S., 2014. CVX: MATLAB software for disciplined convex programming, version 2.1. http://cvxr.com/cvx.

JOUBERT, N., ROBERTS, M., TRUONG, A., BERTHOUZOZ, F., AND HANRAHAN, P. 2015. An interactive tool for designing quadrotor camera shots. *ACM Trans. Graph.* 34, 6, (November), 238:1–238:11.

RCBENCHMARK, 2016. Dynamometer. https://www.rcbenchmark.com/.

TEDRAKE, R., 2014. Underactuated robotics: algorithms for walking, running, swimming, flying, and manipulation (course notes for MIT 6.832). http://underactuated.csail.mit.edu/underactuated.html.

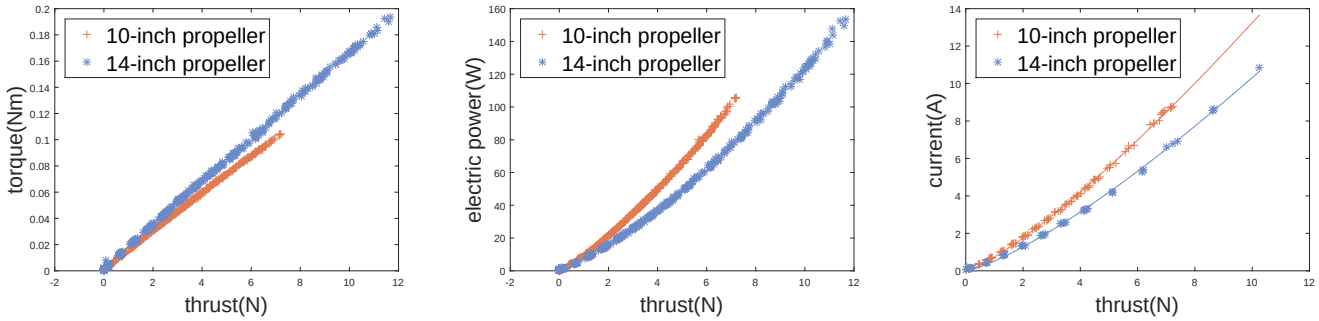

Figure 1: *Torque, thrust and power measurement for two propellers. Left: torque and thrust data. Fitting the measurement with a linear equation gives* $\lambda_{10} = 0.0147$ *and* $\lambda_{14} = 0.0167$. *Middle: electric power and thrust data for two propellers. It can be seen clearly from the plots that given the motor we use, 14-inch propeller is more efficient than its 10-inch counterpart. Right: current and thrust data for two propellers when battery is full. Solid curves denote the power functions fit from the data.*

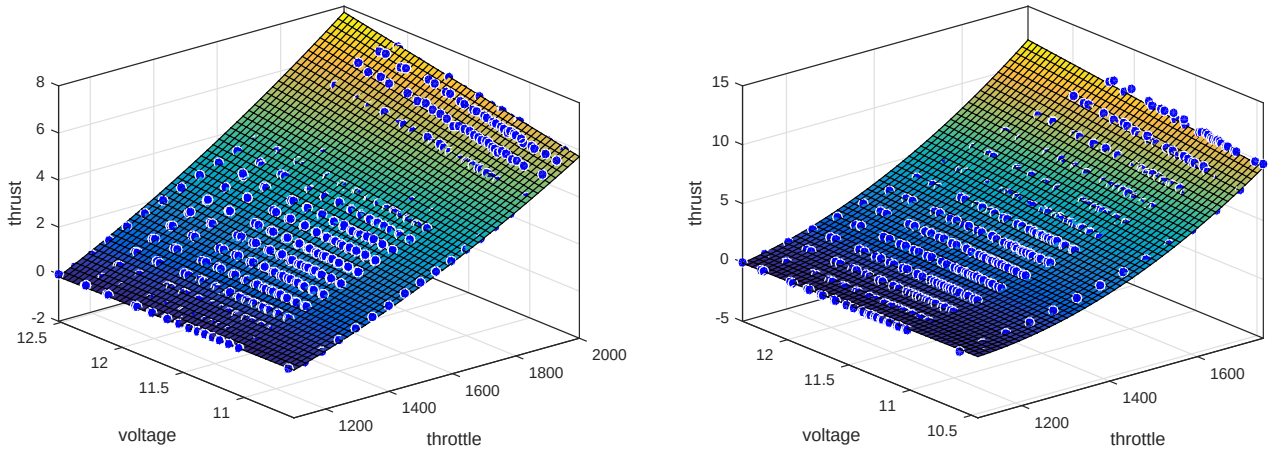

Figure 2: *Fitting voltage, throttle and thrust data with a quadratic model. Measurement data is represented as blue dots. The surface shows the quadratic model that fits best with the measurement. The default range of PWM is 1000 to 2000, and the typical battery value is between 10.5V and 12.5V. Left: 10-inch propeller. The fitting result has* R^2 *= 0.9936 and RMS error = 0.1604. Right: 14-inch propeller. The fitting result has* R^2 *= 0.9975 and RMS error = 0.1649.*