Automatic Re-Initialization and Failure Recovery for Aggressive Flight with a Monocular Vision-Based Quadrotor

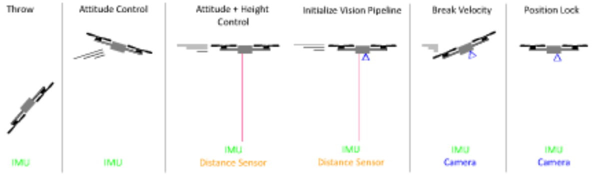
University of Zurich[UZH]

Department of Informatics

Matthias Faessler - Robotics and Perception Group - University of Zurich

ROBOTICS & PERCEPTION GROUP

Recovery Stages



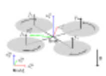
State Estimation

IMU-Based Attitude Estimation

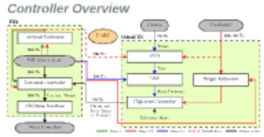
Vision-Based State Estimation

Control

Dynamical Model

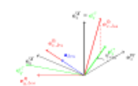


Controller Overview

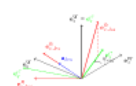


Position Controller

Roll and Pitch Controller



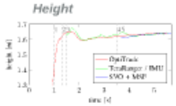
Yaw Controller



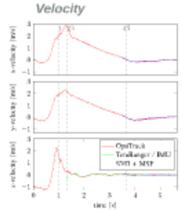
Low-Level Controller

Results

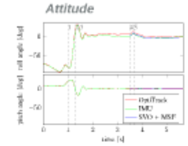
Height



Velocity



Attitude



Authors

Matthias Faessler
faessler@ifi.uzh.ch

Christian Forster
faessler@ifi.uzh.ch

Flavio Fontana
ffontana@ifi.uzh.ch

Davide Scaramuzza
sdavide@ifi.uzh.ch

http://rpg.ifi.uzh.ch

References

[1] C. Forster, M. Pizzoli, and D. Scaramuzza, **SVO: Fast semi-direct monocular visual odometry**, in IEEE Intl. Conf. on Robotics and Automation (ICRA), 2014.

[2] S. Lynen, M. Achtelik, S. Weiss, M. Chli, and R. Siegwart, **A robust and modular multi-sensor fusion approach applied to MAV navigation**, in IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2013.

[3] D. Brescianini, M. Hehn, and R. D'Andrea, **Nonlinear quadrocopter attitude control**, Department of Mechanical and Process Engineering, ETHZ, Tech. Rep., Oct. 2013.

[4] C. Mayhew, R. Sanfelice, and A. Teel, **Quaternion-based hybrid control for robust global attitude tracking**, Automatic Control, IEEE Transactions on, vol. 56, no. 11, pp. 2555–2566, Nov 2011.

[5] N. Trawny and S. I. Roumeliotis, **Indirect Kalman filter for 3D attitude estimation**, University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep. 2005-002, Mar. 2005.

[6] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, **Autonomous, vision-based flight and live dense 3D mapping with a quadrotor MAV**, J. of Field Robotics, 2015.

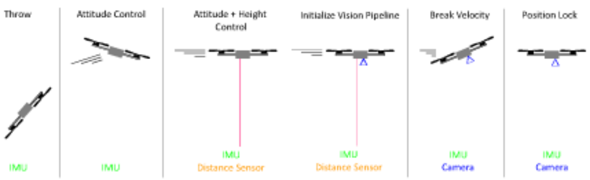
Automatic Re-Initialization and Failure Recovery for Aggressive Flight with a Monocular Vision-Based Quadrotor

University of Zurich
Department of Informatics

Matthias Faessler - Robotics and Perception Group - University of Zurich

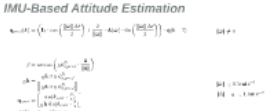
ROBOTICS & PERCEPTION GROUP

Recovery Stages



State Estimation

IMU-Based Attitude Estimation

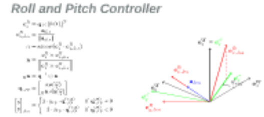


Vision-Based State Estimation

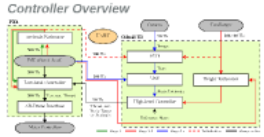
Control

Dynamical Model

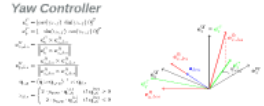
Roll and Pitch Controller



Controller Overview

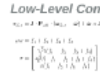


Yaw Controller



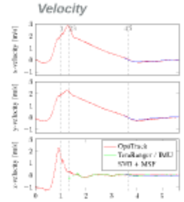
Position Controller

Low-Level Controller

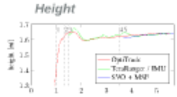


Results

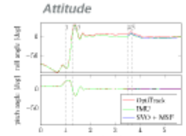
Velocity



Height



Attitude



Authors

Matthias Faessler
faessler@ifi.uzh.ch

Christian Forster
faessler@ifi.uzh.ch

Flavio Fontana
ffontana@ifi.uzh.ch

Davide Scaramuzza
sdavide@ifi.uzh.ch

http://rpg.ifi.uzh.ch

References

[1] C. Forster, M. Pizzoli, and D. Scaramuzza, **SVO: Fast semi-direct monocular visual odometry**, in IEEE Intl. Conf. on Robotics and Automation (ICRA), 2014.

[2] S. Lynen, M. Achtelik, S. Weiss, M. Chli, and R. Siegwart, **A robust and modular multi-sensor fusion approach applied to MAV navigation**, in IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2013.

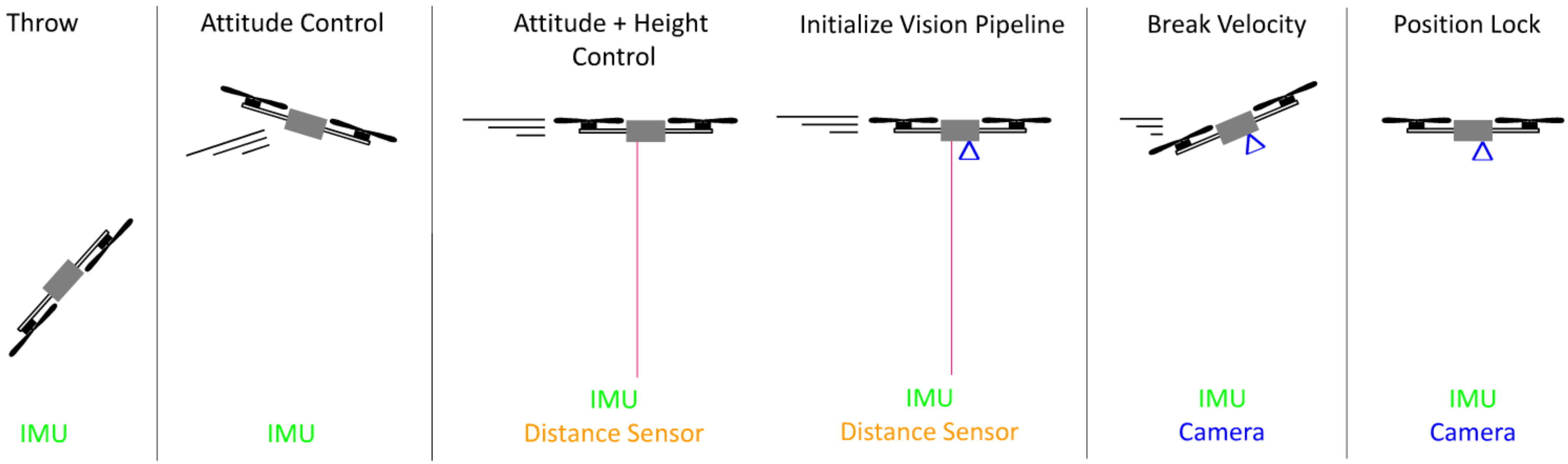
[3] D. Brescianini, M. Hehn, and R. D'Andrea, **Nonlinear quadrocopter attitude control**, Department of Mechanical and Process Engineering, ETHZ, Tech. Rep., Oct. 2013.

[4] C. Mayhew, R. Sanfelice, and A. Teel, **Quaternion-based hybrid control for robust global attitude tracking**, Automatic Control, IEEE Transactions on, vol. 56, no. 11, pp. 2555–2566, Nov 2011.

[5] N. Trawny and S. I. Roumeliotis, **Indirect Kalman filter for 3D attitude estimation**, University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep. 2005-002, Mar. 2005.

[6] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, **Autonomous, vision-based flight and live dense 3D mapping with a quadrotor MAV**, J. of Field Robotics, 2015.

Recovery Stages



State Estimation

State Estimation

IMU-Based Attitude Estimation

$$\hat{\mathbf{q}}_{pred}(k) = \left(\mathbf{I}_4 \cdot \cos\left(\frac{\|\hat{\boldsymbol{\omega}}\|\,\Delta t}{2}\right) + \frac{2}{\|\hat{\boldsymbol{\omega}}\|} \cdot \boldsymbol{\Lambda}(\hat{\boldsymbol{\omega}}) \cdot \sin\left(\frac{\|\hat{\boldsymbol{\omega}}\|\,\Delta t}{2}\right)\right) \cdot \hat{\mathbf{q}}(k-1) \qquad \|\hat{\boldsymbol{\omega}}\| \neq 0$$

$$\begin{aligned}
\beta &= \arccos\left({}_B\hat{\mathbf{e}}^B_{z,pred} \cdot \frac{\bar{\mathbf{a}}}{\|\bar{\mathbf{a}}\|}\right) \\
{}_B\mathbf{h} &= \frac{{}_B\bar{\mathbf{a}} \times {}_B\,\hat{\mathbf{e}}^B_{z,pred}}{\left\|{}_B\bar{\mathbf{a}} \times {}_B\,\hat{\mathbf{e}}^B_{z,pred}\right\|} \\
\mathbf{q}_{corr} &= \begin{bmatrix} \cos(k_{corr} \cdot \frac{\beta}{2}) \\ {}_B\mathbf{h}\sin(k_{corr} \cdot \frac{\beta}{2}) \end{bmatrix} \\
\hat{\mathbf{q}}(k) &= \hat{\mathbf{q}}_{pred}(k) \otimes \mathbf{q}_{corr}
\end{aligned}$$

$$\|\bar{\boldsymbol{\omega}}\| < 0.5\,\mathrm{rad\,s^{-1}}$$

$$\left|\|\bar{\mathbf{a}}\| - g\right| < 1.0\,\mathrm{m\,s^{-2}}$$

Vision-Based State Estimation

State

IMU-Based Attitude Estimation

$$\hat{\mathbf{q}}_{pred}(k) = \left(\mathbf{I}_4 \cdot \cos\left(\frac{\|\hat{\boldsymbol{\omega}}\|\,\Delta t}{2}\right) + \frac{2}{\|\hat{\boldsymbol{\omega}}\|} \cdot \boldsymbol{\Lambda}(\hat{\boldsymbol{\omega}}) \cdot \sin\left(\frac{\|\hat{\boldsymbol{\omega}}\|\,\Delta t}{2}\right)\right) \cdot \hat{\mathbf{q}}(k-1) \qquad \|\hat{\boldsymbol{\omega}}\| \neq 0$$

$$\begin{aligned}
\beta &= \arccos\left({}_B\hat{\mathbf{e}}^B_{z,pred} \cdot \frac{\tilde{\mathbf{a}}}{\|\tilde{\mathbf{a}}\|}\right) \\
{}_B\mathbf{h} &= \frac{{}_B\tilde{\mathbf{a}} \times {}_B\,\hat{\mathbf{e}}^B_{z,pred}}{\left\|{}_B\tilde{\mathbf{a}} \times {}_B\,\hat{\mathbf{e}}^B_{z,pred}\right\|} \\
\mathbf{q}_{corr} &= \begin{bmatrix} \cos(k_{corr} \cdot \frac{\beta}{2}) \\ {}_B\mathbf{h}\sin(k_{corr} \cdot \frac{\beta}{2}) \end{bmatrix} \\
\hat{\mathbf{q}}(k) &= \hat{\mathbf{q}}_{pred}(k) \otimes \mathbf{q}_{corr}
\end{aligned}$$

$$\|\tilde{\boldsymbol{\omega}}\| < 0.5\,\mathrm{rad\,s^{-1}}$$

$$|\|\tilde{\mathbf{a}}\| - g| < 1.0\,\mathrm{m\,s^{-2}}$$

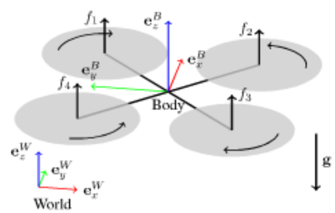
Vision-Based State Estimation

Control

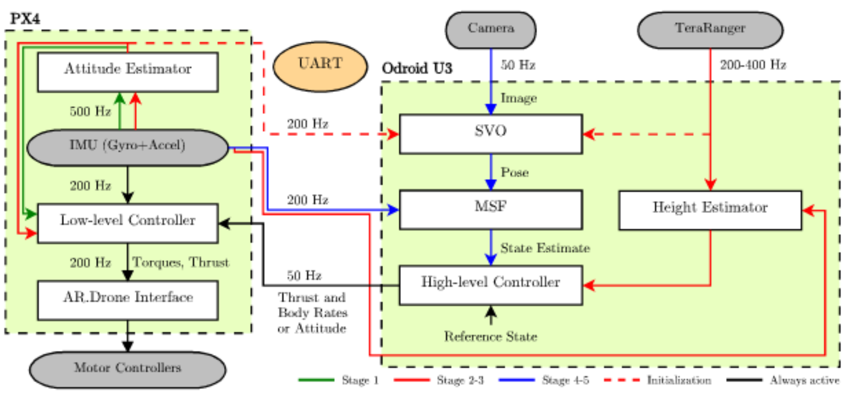
Dynamical Model

$$\dot{\mathbf{r}} = \mathbf{v}$$
$$\dot{\mathbf{v}} = \mathbf{g} + \mathbf{q} \odot \mathbf{c}$$
$$\dot{\mathbf{q}} = \boldsymbol{\Lambda}(\boldsymbol{\omega}) \cdot \mathbf{q}$$
$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1} \cdot (\boldsymbol{\tau} - \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega})$$

$$\boldsymbol{\Lambda}(\boldsymbol{\omega}) = \frac{1}{2}\begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}$$
$$mc = f_1 + f_2 + f_3 + f_4$$
$$\boldsymbol{\tau} = \begin{bmatrix} \frac{\sqrt{2}}{2} l(f_1 - f_2 - f_3 + f_4) \\ \frac{\sqrt{2}}{2} l(-f_1 - f_2 + f_3 + f_4) \\ \kappa(f_1 - f_2 + f_3 - f_4) \end{bmatrix}$$



Controller Overview

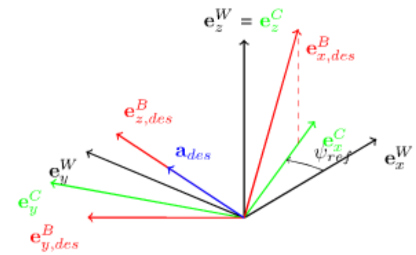


Position Controller

$$\mathbf{a}_{des} = \mathbf{P}_{pos} \cdot (\mathbf{r}_{ref} - \hat{\mathbf{r}}) + \mathbf{D}_{pos} \cdot (\mathbf{v}_{ref} - \hat{\mathbf{v}}) + \mathbf{a}_{ref} - \mathbf{g}$$
$$c_{des} = \mathbf{a}_{des} \cdot \mathbf{e}_z^B$$

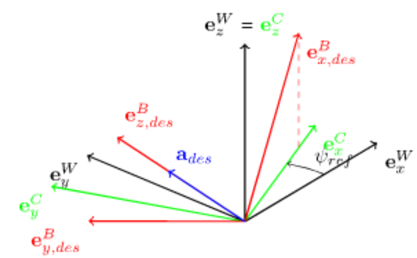
Roll and Pitch Controller

$$\hat{\mathbf{e}}_z^B = \hat{\mathbf{q}} \otimes [0\ 0\ 1]^T$$
$$\mathbf{e}_{z,des}^B = \frac{\mathbf{a}_{des}}{\|\mathbf{a}_{des}\|}$$
$$\alpha = \arccos(\hat{\mathbf{e}}_z^B \cdot \mathbf{e}_{z,des}^B)$$
$$\mathbf{n} = \frac{\hat{\mathbf{e}}_z^B \times \mathbf{e}_{z,des}^B}{\left\|\hat{\mathbf{e}}_z^B \times \mathbf{e}_{z,des}^B\right\|}$$
$${}_B\mathbf{n} = \hat{\mathbf{q}}^{-1} \odot \mathbf{n}$$
$$\mathbf{q}_{e,rp} = \begin{bmatrix} \cos(\frac{\alpha}{2}) \\ {}_B\mathbf{n} \sin(\frac{\alpha}{2}) \end{bmatrix}$$
$$\begin{bmatrix} p \\ q \end{bmatrix}_{des} = \begin{cases} 2 \cdot p_{rp} \cdot \mathbf{q}_{e,rp}^{(x,y)} & \text{if } \mathbf{q}_{e,rp}^{(w)} \geq 0 \\ -2 \cdot p_{rp} \cdot \mathbf{q}_{e,rp}^{(x,y)} & \text{if } \mathbf{q}_{e,rp}^{(w)} < 0 \end{cases}$$



Yaw Controller

$$\mathbf{e}_x^C = [\cos(\psi_{ref})\ \sin(\psi_{ref})\ 0]^T$$
$$\mathbf{e}_y^C = [-\sin(\psi_{ref})\ \cos(\psi_{ref})\ 0]^T$$
$$\mathbf{e}_{x,des}^B = \frac{\mathbf{e}_y^C \times \mathbf{e}_{z,des}^B}{\left\|\mathbf{e}_y^C \times \mathbf{e}_{z,des}^B\right\|}$$
$$\mathbf{e}_{y,des}^B = \frac{\mathbf{e}_{z,des}^B \times \mathbf{e}_{x,des}^B}{\left\|\mathbf{e}_{z,des}^B \times \mathbf{e}_{x,des}^B\right\|}$$
$$\mathbf{q}_{e,y} = (\hat{\mathbf{q}} \otimes \mathbf{q}_{e,rp})^{-1} \otimes \mathbf{q}_{des}$$
$$r_{des} = \begin{cases} 2 \cdot p_{yaw} \cdot \mathbf{q}_{e,y}^{(z)} & \text{if } \mathbf{q}_{e,y}^{(w)} \geq 0 \\ -2 \cdot p_{yaw} \cdot \mathbf{q}_{e,y}^{(z)} & \text{if } \mathbf{q}_{e,y}^{(w)} < 0 \end{cases}$$



Low-Level Controller

$$\boldsymbol{\tau}_{des} = \mathbf{J} \cdot \mathbf{P}_{att} \cdot (\boldsymbol{\omega}_{des} - \hat{\boldsymbol{\omega}}) + \hat{\boldsymbol{\omega}} \times \mathbf{J}\hat{\boldsymbol{\omega}}$$

$$mc = f_1 + f_2 + f_3 + f_4$$
$$\boldsymbol{\tau} = \begin{bmatrix} \frac{\sqrt{2}}{2} l(f_1 - f_2 - f_3 + f_4) \\ \frac{\sqrt{2}}{2} l(-f_1 - f_2 + f_3 + f_4) \\ \kappa(f_1 - f_2 + f_3 - f_4) \end{bmatrix}$$

Co

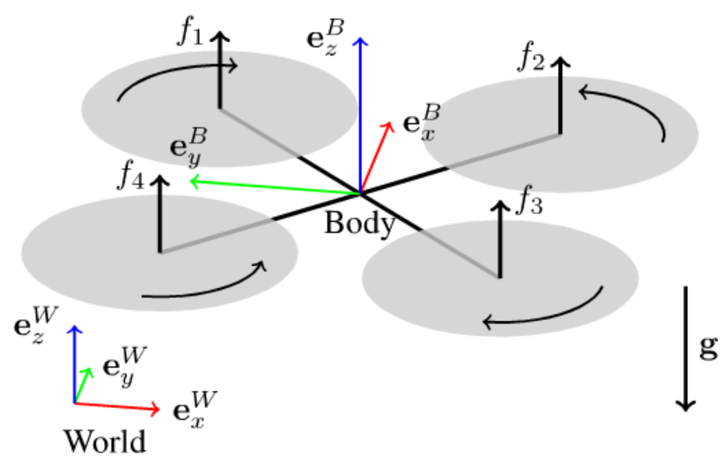
Dynamical Model

$$\dot{\mathbf{r}} = \mathbf{v}$$
$$\dot{\mathbf{v}} = \mathbf{g} + \mathbf{q} \odot \mathbf{c}$$
$$\dot{\mathbf{q}} = \mathbf{\Lambda}(\boldsymbol{\omega}) \cdot \mathbf{q}$$
$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1} \cdot (\boldsymbol{\tau} - \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega})$$

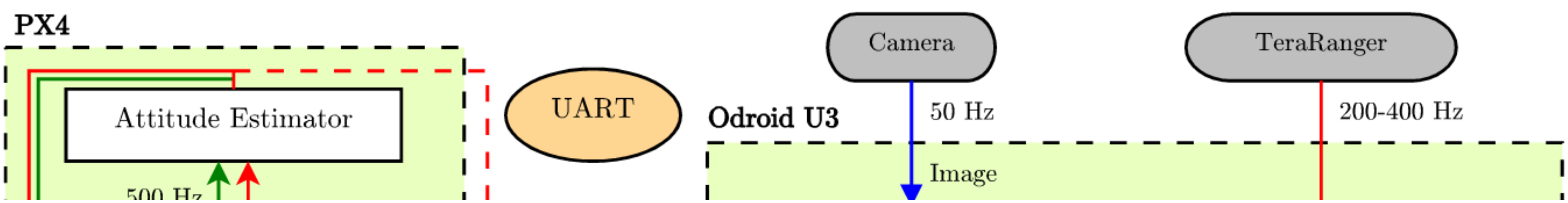
$$\mathbf{\Lambda}(\boldsymbol{\omega}) = \frac{1}{2}\begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}$$

$$mc = f_1 + f_2 + f_3 + f_4$$

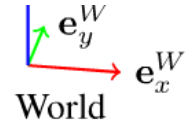
$$\boldsymbol{\tau} = \begin{bmatrix} \frac{\sqrt{2}}{2} l(f_1 - f_2 - f_3 + f_4) \\ \frac{\sqrt{2}}{2} l(-f_1 - f_2 + f_3 + f_4) \\ \kappa(f_1 - f_2 + f_3 - f_4) \end{bmatrix}$$



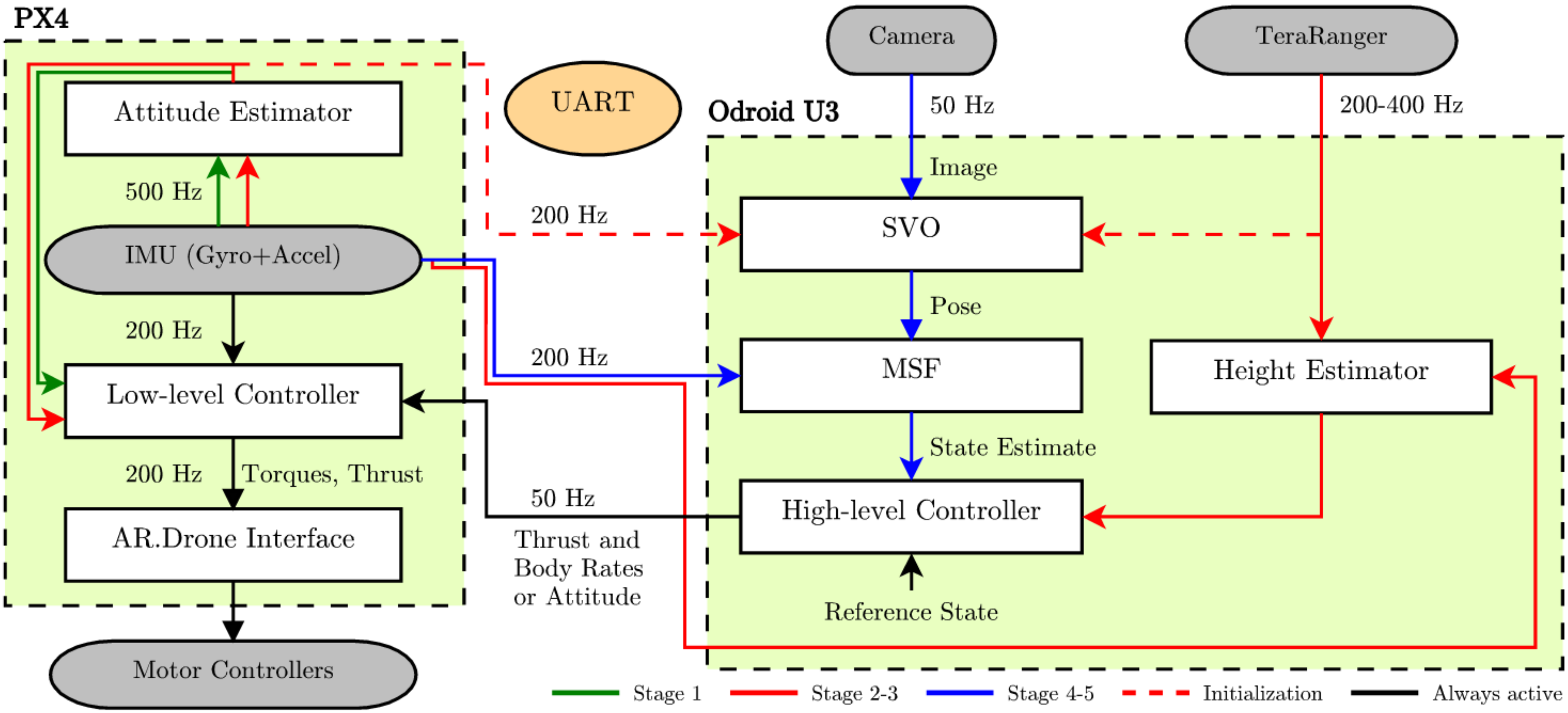
Controller Overview



$$\boldsymbol{\tau} = \begin{bmatrix} \frac{\sqrt{2}}{2} l(-f_1 - f_2 + f_3 + f_4) \\ \kappa(f_1 - f_2 + f_3 - f_4) \end{bmatrix}$$



Controller Overview



Position Controller

Position Controller

$$\mathbf{a}_{des} = \mathbf{P}_{pos} \cdot (\mathbf{r}_{ref} - \hat{\mathbf{r}}) + \mathbf{D}_{pos} \cdot (\mathbf{v}_{ref} - \hat{\mathbf{v}}) + \mathbf{a}_{ref} - \mathbf{g}$$

$$c_{des} = \mathbf{a}_{des} \cdot \mathbf{e}_z^B$$

Roll and Pitch Controller

$$\hat{\mathbf{e}}_z^B = \hat{\mathbf{q}} \otimes [0\ 0\ 1]^T$$

$$\mathbf{e}_{z,des}^B = \frac{\mathbf{a}_{des}}{\|\mathbf{a}_{des}\|}$$

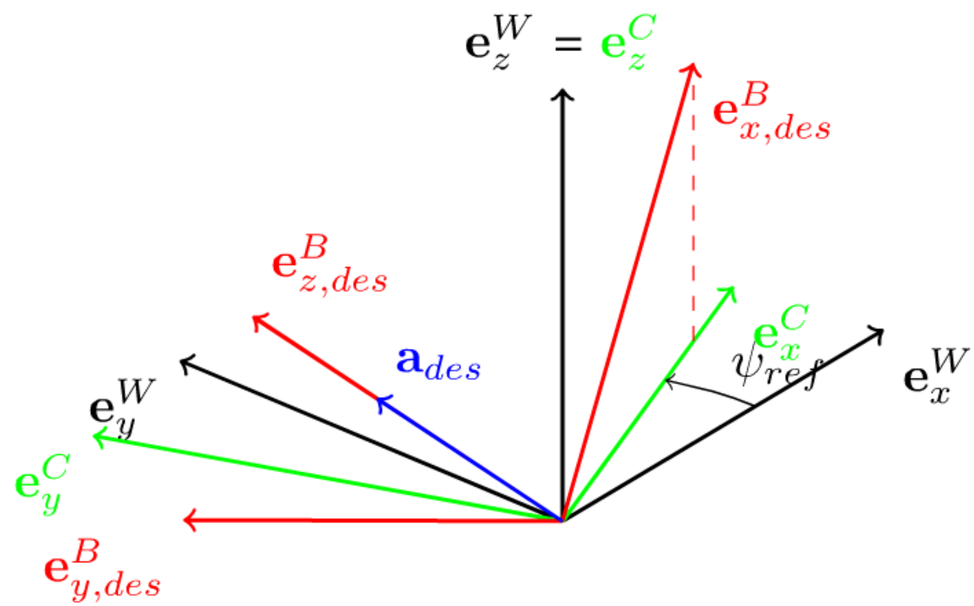
$$\alpha = \arccos(\hat{\mathbf{e}}_z^B \cdot \mathbf{e}_{z,des}^B)$$

$$\mathbf{n} = \frac{\hat{\mathbf{e}}_z^B \times \mathbf{e}_{z,des}^B}{\left\|\hat{\mathbf{e}}_z^B \times \mathbf{e}_{z,des}^B\right\|}$$

$${}_B\mathbf{n} = \hat{\mathbf{q}}^{-1} \odot \mathbf{n}$$

$$\mathbf{q}_{e,rp} = \begin{bmatrix} \cos(\frac{\alpha}{2}) \\ {}_B\mathbf{n}\sin(\frac{\alpha}{2}) \end{bmatrix}$$

$$\begin{bmatrix} p \\ q \end{bmatrix}_{des} = \begin{cases} 2 \cdot p_{rp} \cdot \mathbf{q}_{e,rp}^{(x,y)} & \text{if } \mathbf{q}_{e,rp}^{(w)} \geq 0 \\ -2 \cdot p_{rp} \cdot \mathbf{q}_{e,rp}^{(x,y)} & \text{if } \mathbf{q}_{e,rp}^{(w)} < 0 \end{cases}$$

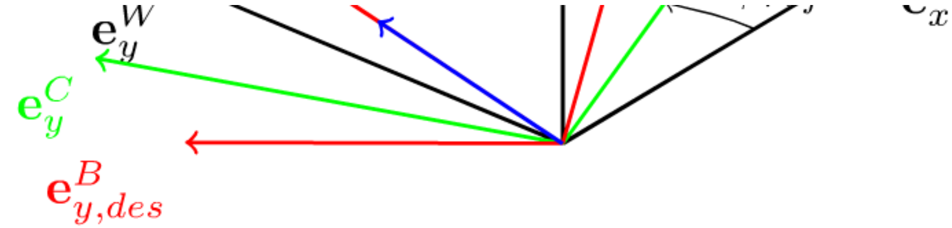


Yaw Controller

$$\mathbf{e}_x^C = [\cos(\psi_{ref})\ \ \sin(\psi_{ref})\ 0]^T$$

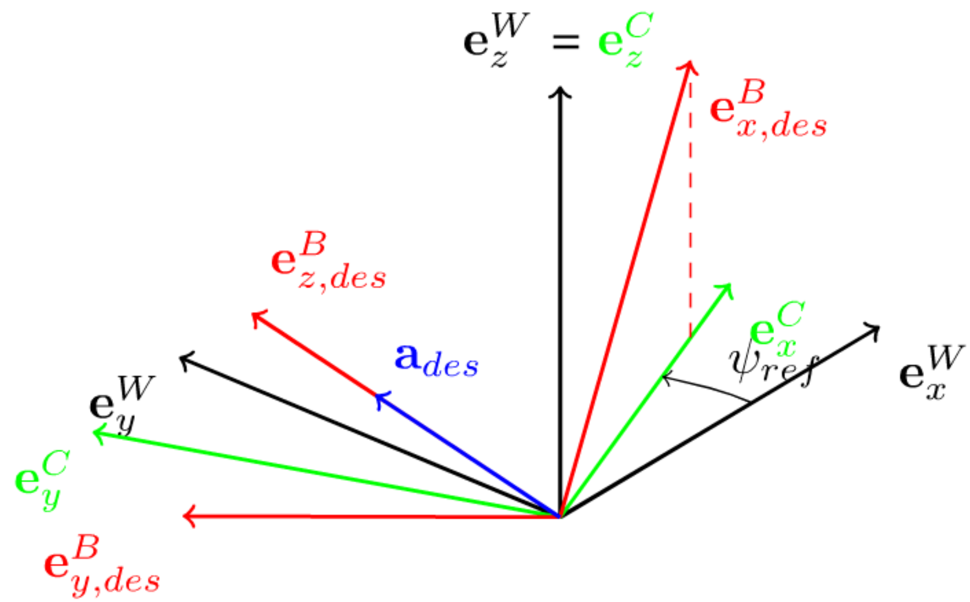
$$\mathbf{q}_{e,rp} = \begin{bmatrix} \cos(\frac{\alpha}{2}) \\ {}_B\mathbf{n}\sin(\frac{\alpha}{2}) \end{bmatrix}$$

$$\begin{bmatrix} p \\ q \end{bmatrix}_{des} = \begin{cases} 2 \cdot p_{rp} \cdot \mathbf{q}_{e,rp}^{(x,y)} & \text{if } \mathbf{q}_{e,rp}^{(w)} \geq 0 \\ -2 \cdot p_{rp} \cdot \mathbf{q}_{e,rp}^{(x,y)} & \text{if } \mathbf{q}_{e,rp}^{(w)} < 0 \end{cases}$$



Yaw Controller

$$\mathbf{e}_x^C = [\cos(\psi_{ref}) \;\; \sin(\psi_{ref}) \;\; 0]^T$$
$$\mathbf{e}_y^C = [-\sin(\psi_{ref}) \;\; \cos(\psi_{ref}) \;\; 0]^T$$
$$\mathbf{e}_{x,des}^B = \frac{\mathbf{e}_y^C \times \mathbf{e}_{z,des}^B}{\left\| \mathbf{e}_y^C \times \mathbf{e}_{z,des}^B \right\|}$$
$$\mathbf{e}_{y,des}^B = \frac{\mathbf{e}_{z,des}^B \times \mathbf{e}_{x,des}^B}{\left\| \mathbf{e}_{z,des}^B \times \mathbf{e}_{x,des}^B \right\|}$$
$$\mathbf{q}_{e,y} = (\hat{\mathbf{q}} \otimes \mathbf{q}_{e,rp})^{-1} \otimes \mathbf{q}_{des}$$
$$r_{des} = \begin{cases} 2 \cdot p_{yaw} \cdot \mathbf{q}_{e,y}^{(z)} & \text{if } \mathbf{q}_{e,y}^{(w)} \geq 0 \\ -2 \cdot p_{yaw} \cdot \mathbf{q}_{e,y}^{(z)} & \text{if } \mathbf{q}_{e,y}^{(w)} < 0 \end{cases}$$



Low-Level Controller

Low-Level Controller

$$\boldsymbol{\tau}_{des} = \mathbf{J} \cdot \mathbf{P}_{att} \cdot (\boldsymbol{\omega}_{des} - \hat{\boldsymbol{\omega}}) + \hat{\boldsymbol{\omega}} \times \mathbf{J}\hat{\boldsymbol{\omega}}$$

$$\begin{aligned} mc &= f_1 + f_2 + f_3 + f_4 \\ \boldsymbol{\tau} &= \begin{bmatrix} \frac{\sqrt{2}}{2} l(f_1 - f_2 - f_3 + f_4) \\ \frac{\sqrt{2}}{2} l(-f_1 - f_2 + f_3 + f_4) \\ \kappa(f_1 - f_2 + f_3 - f_4) \end{bmatrix} \end{aligned}$$

$$\mathbf{a}_{des} = \mathbf{P}_{pos} \cdot (\mathbf{r}_{ref} - \hat{\mathbf{r}}) + \mathbf{D}_{pos} \cdot (\mathbf{v}_{ref} - \hat{\mathbf{v}}) + \mathbf{a}_{ref} - \mathbf{g}$$

$$c_{des} = \mathbf{a}_{des} \cdot \mathbf{e}_z^B$$

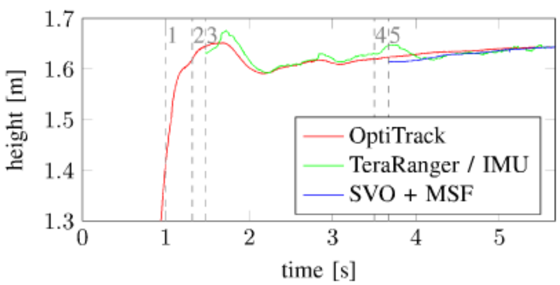
$$\boldsymbol{\tau}_{des} = \mathbf{J} \cdot \mathbf{P}_{att} \cdot (\boldsymbol{\omega}_{des} - \hat{\boldsymbol{\omega}}) + \hat{\boldsymbol{\omega}} \times \mathbf{J}\hat{\boldsymbol{\omega}}$$

$$mc = f_1 + f_2 + f_3 + f_4$$

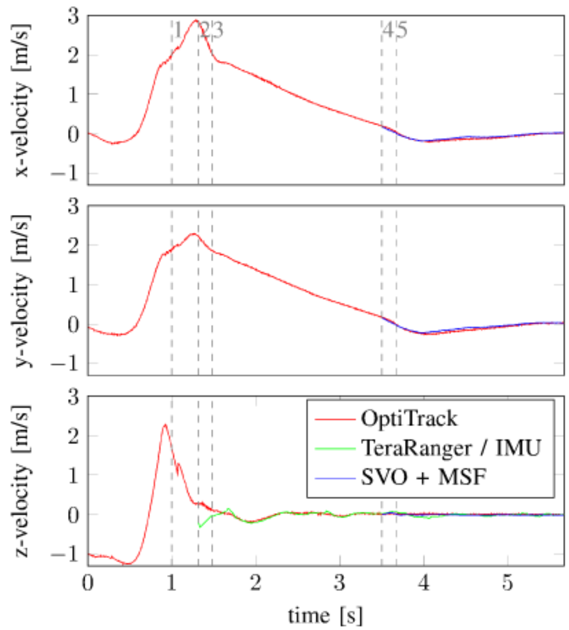
$$\boldsymbol{\tau} = \begin{bmatrix} \frac{\sqrt{2}}{2} l (f_1 - f_2 - f_3 + f_4) \\ \frac{\sqrt{2}}{2} l (-f_1 - f_2 + f_3 + f_4) \\ \kappa (f_1 - f_2 + f_3 - f_4) \end{bmatrix}$$

Results

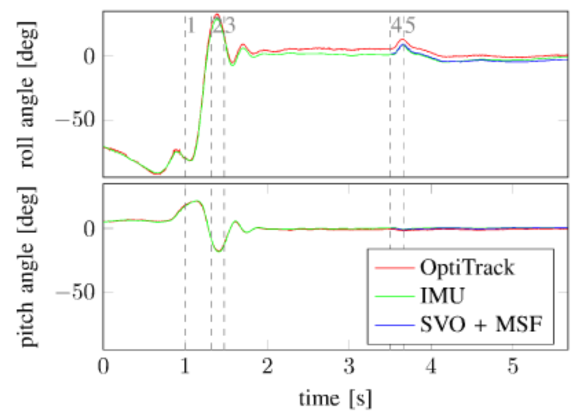
Height



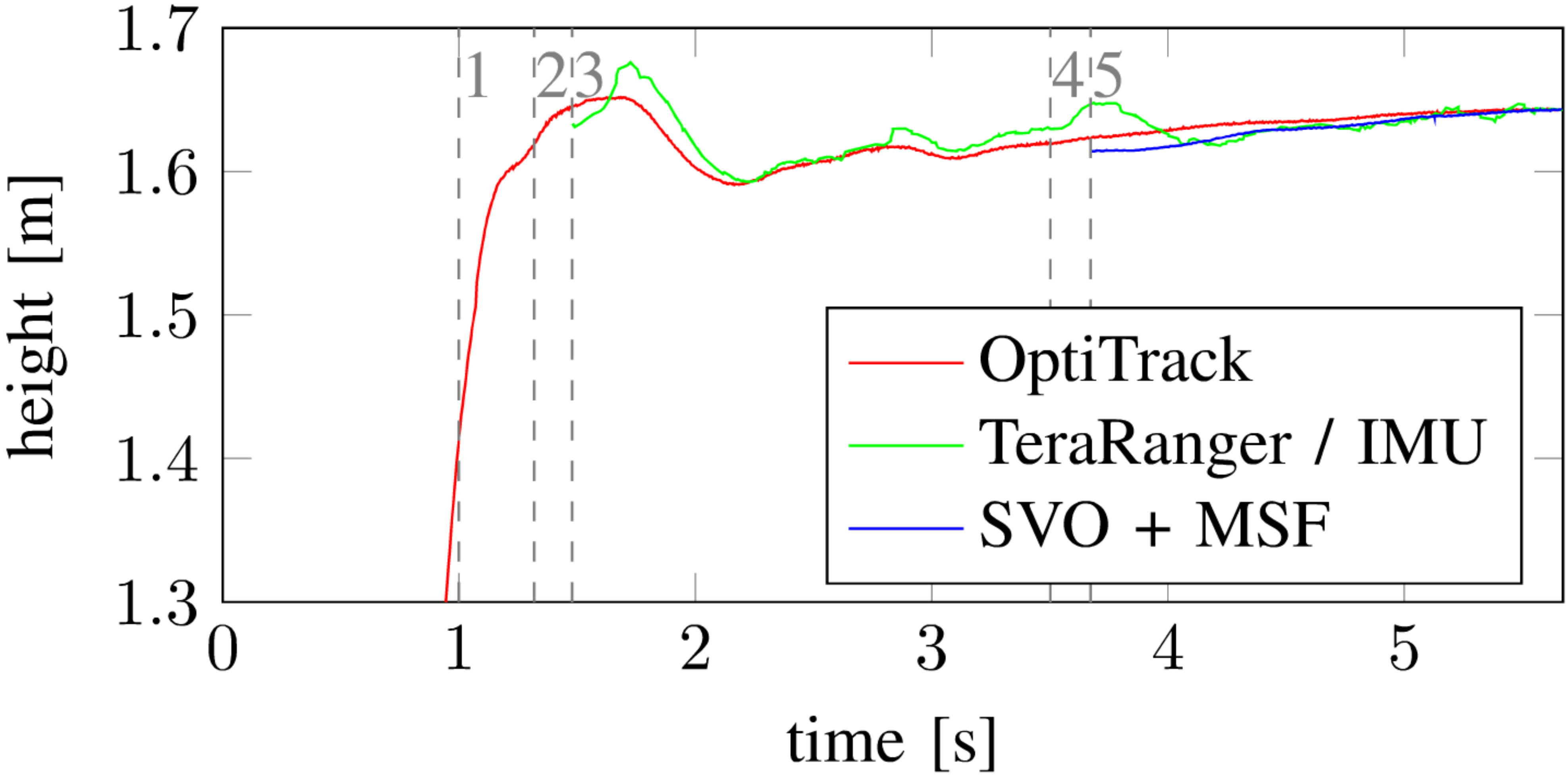
Velocity



Attitude

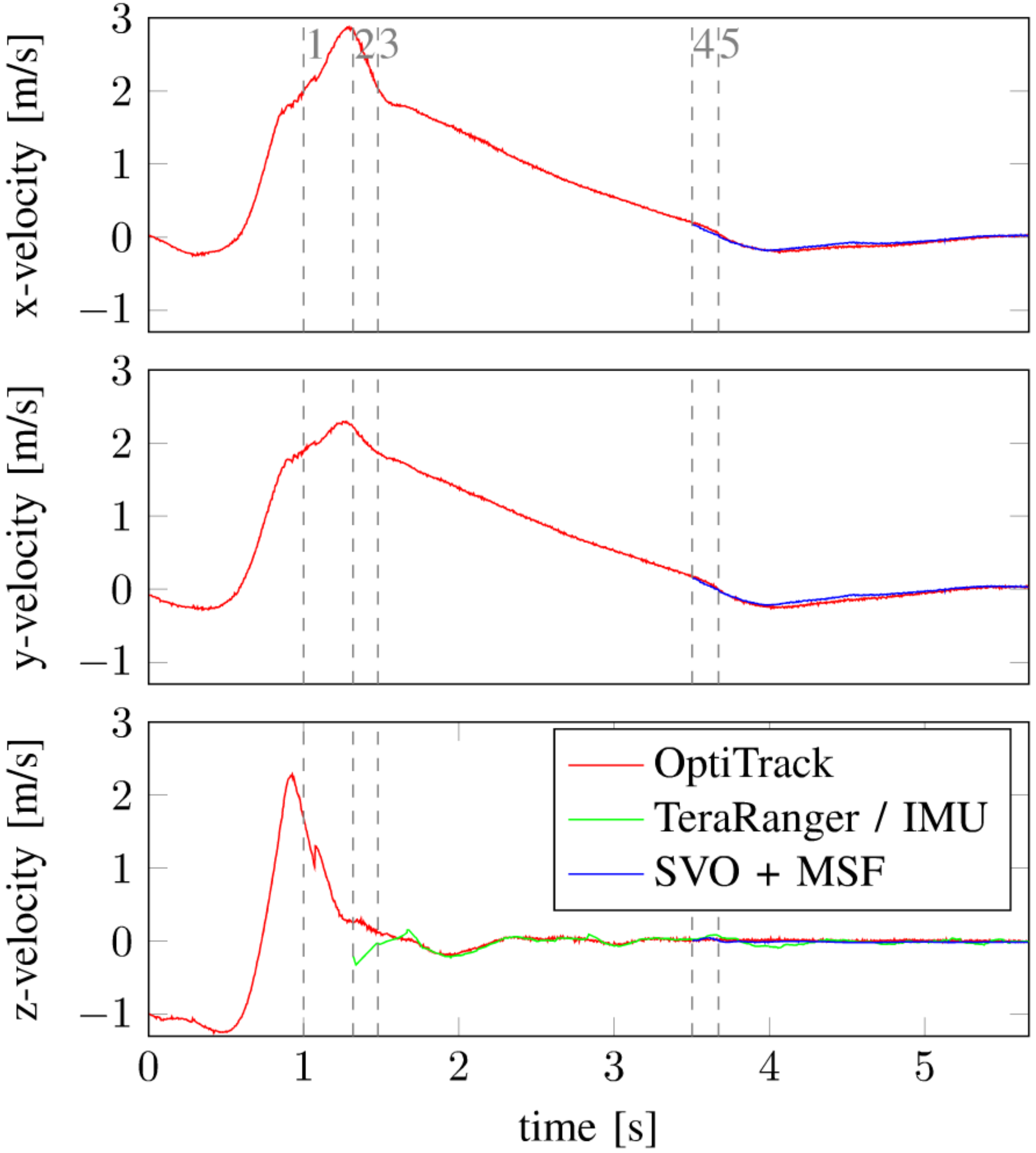


Height

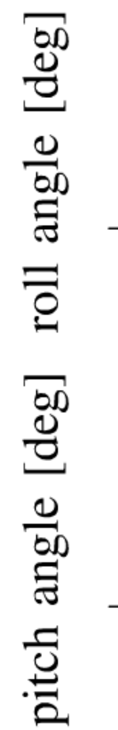


1.7
1.6
1.5
1.4
1.3
height [m]
0
1
2
3
4
5
time [s]
1
2
3
4
5
OptiTrack
TeraRanger / IMU
SVO + MSF

Velocity

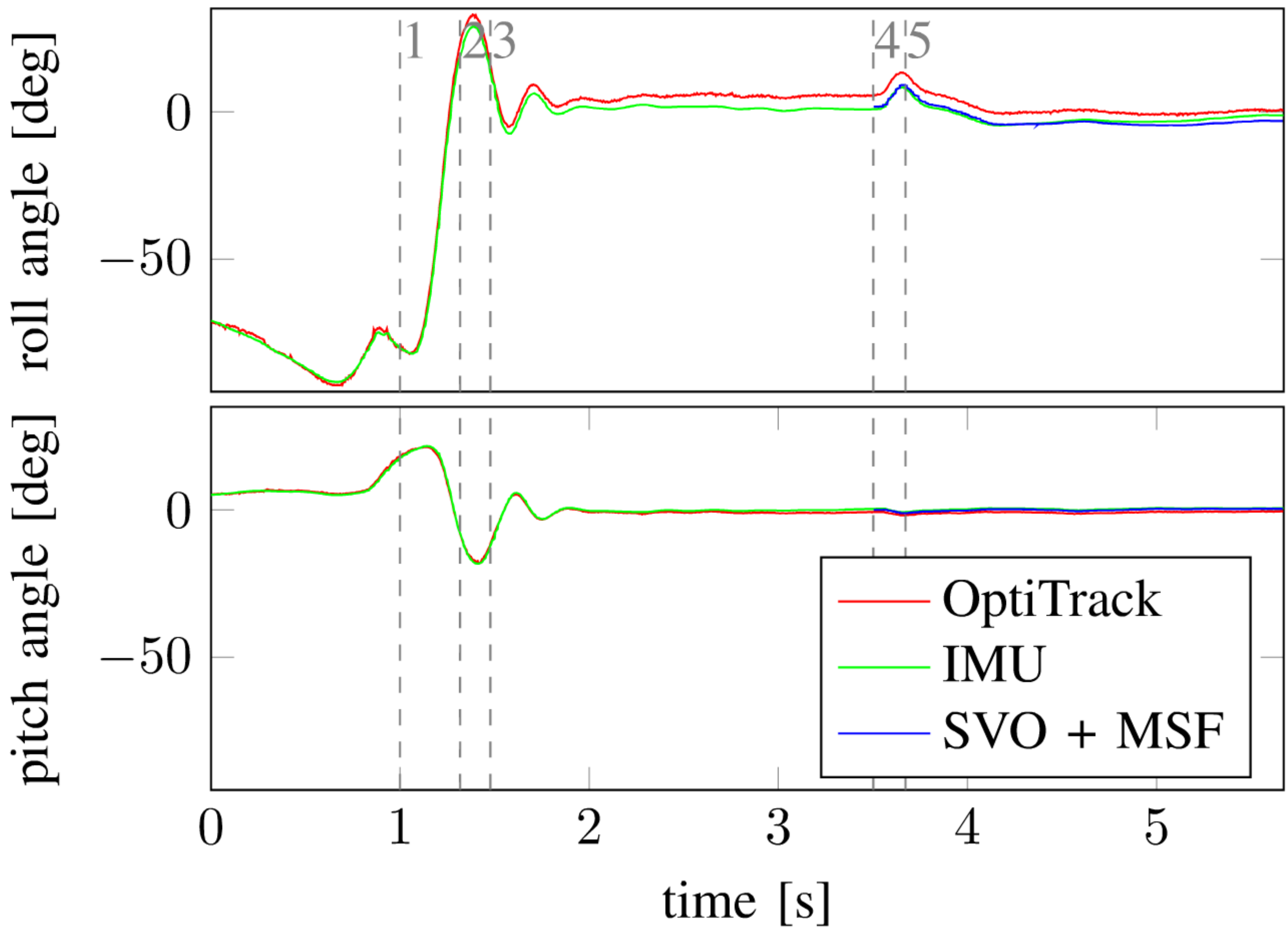


x-velocity [m/s]
y-velocity [m/s]
z-velocity [m/s]
time [s]
1
2
3
4
5
OptiTrack
TeraRanger / IMU
SVO + MSF



pitch angle [deg]
roll angle [deg]

Attitude



roll angle [deg]
pitch angle [deg]
time [s]
0
−50
0
−50
0
1
2
3
4
5
1
2
3
4
5
OptiTrack
IMU
SVO + MSF

Authors

Matthias Faessler
faessler@ifi.uzh.ch

Flavio Fontana
ffontana@ifi.uzh.ch

Christian Forster
faessler@ifi.uzh.ch

Davide Scaramuzza
sdavide@ifi.uzh.ch

http://rpg.ifi.uzh.ch

References

[1] C. Forster, M. Pizzoli, and D. Scaramuzza, **SVO: Fast semi-direct monocular visual odometry**, in IEEE Intl. Conf. on Robotics and Automation (ICRA), 2014.

[2] S. Lynen, M. Achtelik, S. Weiss, M. Chli, and R. Siegwart, **A robust and modular multi-sensor fusion approach applied to MAV navigation**, in IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2013.

[3] D. Brescianini, M. Hehn, and R. D'Andrea, **Nonlinear quadrocopter attitude control**, Department of Mechanical and Process Engineering, ETHZ, Tech. Rep., Oct. 2013.

[4] C. Mayhew, R. Sanfelice, and A. Teel, **Quaternion-based hybrid control for robust global attitude tracking**, Automatic Control, IEEE Transactions on, vol. 56, no. 11, pp. 2555–2566, Nov 2011.

[5] N. Trawny and S. I. Roumeliotis, **Indirect Kalman filter for 3D attitude estimation**, University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep. 2005-002, Mar. 2005.

[6] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, **Autonomous, vision-based flight and live dense 3D mapping with a quadrotor MAV**, J. of Field Robotics, 2015.