Optimization of the control effort for quadrotors using a varying-gain quasi-continuous sliding mode control influenced by a wind estimator

Gabriele Perozzi

Optimization of the control effort for quadrotors using a varying-gain quasi-continuous sliding mode control influenced by a wind estimator

Gabriele Perozzi

Abstract

The problem of position tracking of a mini drone subjected to wind perturbations is investigated. Using the real-time on-board wind estimation and adapting properly the control gains, the quadrotor can follow a reference trajectory ensuring an optimal control effort on the rotors and a good robustness against the wind perturbations. In this way, control gains are modified in real-time according to the requested trajectory and to the requested disturbance compensation. The analysis process is explained using a toolbox, implementing the control and wind estimation algorithms and the fully configurable quadrotor model. Final simulations conclude the note, enlightening the importance of including wind estimates in the control algorithm.

1. Implemented algorithms

In this section, the algorithms implemented in the Toolbox are briefly illustrated. For the explanation of the control see [1], [2], for the explanation of the wind estimation see [3].

1.1 Control

The control for z and for $i = x,\ y$ are given by

$$U_z = \frac{m}{\cos\theta\cos\phi}\Big(g - \ddot{z}_{des} + \alpha_z\dot{e}_z + \Big(\frac{1}{2}(\nu(X)^2 + 2\varrho(X) + \nu(X)\sqrt{\nu^2(X) + 4\varrho(X)}) + \frac{L_z}{b} - L_z\alpha_z + \varpi_z\Big)\frac{\lceil\dot{S}_z\rfloor^2 + S_z}{\varrho_z + |\dot{S}_z|^2 + |S_z|}\Big),$$

$$U_i = \frac{m}{U_z}\Big(-\Big(d_{xx} + 2d_{yy} + 2d_{zz} + \varpi_i\Big)\frac{\lceil\dot{e}_i\rfloor^2 + e_i}{\varrho_i + |\dot{e}_i|^2 + |e_i|} + \ddot{i}_{des} - \alpha_i\dot{e}_i\Big),$$

The controls for roll, pitch and yaw respectively are given by

$$U_\phi = I_{xx}\Big(-qr\frac{I_{yy} - I_{zz}}{I_{xx}} - \frac{1}{I_{xx}}\frac{\lceil\dot{S}_\phi\rfloor^2 + S_\phi}{\varrho_\phi + |\dot{S}_\phi|^2 + |S_\phi|}\Big(\tilde{K}_\phi(f_{\phi1}(X) + D_{\phi1})\sqrt{|U_z|} + \bar{K}_\phi(f_{\phi2}(X) + D_{\phi2}) + \varpi_\phi + \alpha_\phi\delta_\phi L_\phi - \frac{\delta_\phi}{b}L_\phi\Big) - \alpha_\phi\dot{e}_\phi + \ddot{\phi}_{des}\Big),$$

$$U_\theta = I_{yy}\Big(-pr\frac{I_{zz} - I_{xx}}{I_{yy}} - \frac{1}{I_{yy}}\frac{\lceil\dot{S}_\theta\rfloor^2 + S_\theta}{\varrho_\theta + |\dot{S}_\theta|^2 + |S_\theta|}\Big(\tilde{K}_\theta(f_{\theta1}(X) + D_{\theta1})\sqrt{|U_z|} + \bar{K}_\theta(f_{\theta2}(X) + D_{\theta2}) + \varpi_\theta + \alpha_\theta\delta_\theta L_\theta - \frac{\delta_\theta}{b}L_\theta\Big) - \alpha_\theta\dot{e}_\theta + \ddot{\theta}_{des}\Big),$$

$$U_\psi = I_{zz}\Big(-pq\frac{I_{xx} - I_{yy}}{I_{zz}} - \frac{1}{I_{zz}}\frac{\lceil\dot{S}_\psi\rfloor^2 + S_\psi}{\varrho_\psi + |\dot{S}_\psi|^2 + |S_\psi|}\Big(\tilde{K}_\psi(f_{\psi1}(X) + D_{\psi1})\sqrt{|U_z|} + \bar{K}_\psi(f_{\psi2}(X) + D_{\psi2}) + \varpi_\psi + \alpha_\psi\delta_\psi L_\psi - \frac{\delta_\psi}{b}L_\psi\Big) - \alpha_\psi\dot{e}_\psi + \ddot{\psi}_{des}\Big),$$

1.2 Wind estimator

Wind estimator algorithm for quadrotor translational dynamics is defined by

$$\begin{aligned}\tilde{\Delta}_a - \hat{\Delta}_a &= \Omega_a(d_w - \hat{d}_w) + \epsilon_a,\\ \dot{\hat{d}}_w &= \gamma_a\,\Omega_a^T\lceil\tilde{\Delta}_a - \hat{\Delta}_a\rfloor^{\alpha_a}.\end{aligned}$$

Wind estimator algorithm for quadrotor rotational dynamics is defined by

$$\begin{aligned}\dot{\hat{\Delta}}_g &= f_{0g} + \Omega_g \hat{d}_w + \ell'_g(\tilde{\Delta}_g - \hat{\Delta}_g) + \Xi\dot{\hat{d}}_w,\\ \dot{\Xi} &= -\ell'_g\Xi + \Omega_g,\\ \dot{\hat{d}}_w &= \gamma'_g\Xi^T\left\lceil\tilde{\Delta}_g - \hat{\Delta}_g\right\rfloor^{\alpha'_g}.\end{aligned}$$

2. Setting the toolbox

CW-Quad simulink toolbox (see [4], [5] for detailed explanations) is used for the simulations.

2.1 Quadrotor model

In the toolbox, the Quadrotor model is implemented inside of the block in Fig. 1. *Rotors* is the rotors velocity, *Wind* is the wind velocity, *State* is the state vector of the system, going to the Sensor model to evaluate the accelerometer and gyroscope noise.

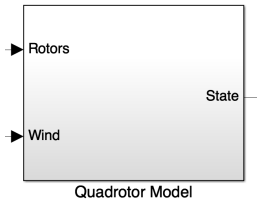

Figure 1: Quadrotor model

2.2 Rotors model

In the toolbox, the Quadrotor model is implemented inside of the block in Fig. 2. *Control* is the control signal, *omega* is the rotors velocity affected by the time-delay caused by the rotor transfer function, going to the Quadrotor model.

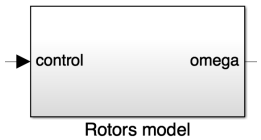

Figure 2: Rotors model

2.3 Control model

In the toolbox, the Quadrotor model is implemented inside of the block in Fig. 3. *state* is the state vector of the quadrotor system, *state des* is the desired 3D trajectory, *wind* is the wind estimates, *omega* is the rotors velocity, *U* is the control signal going to the Rotors model.

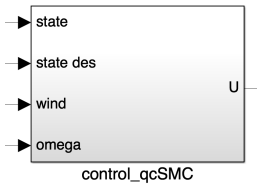

Figure 3: Control model

2.4 Wind estimator model

In the toolbox, the Quadrotor model is implemented inside of the block in Fig. 4. *rotors* is the rotors velocity, *acc* is the accelerometer measurements, *gyro* is the gyroscope measurements, *angles* is the angles of the quadrotor, *uav lin vel* is the translational quadrotor velocity, *wind control* is the wind estimates in body frame going to the control model, *wind est* is the wind estimates in earth frame.

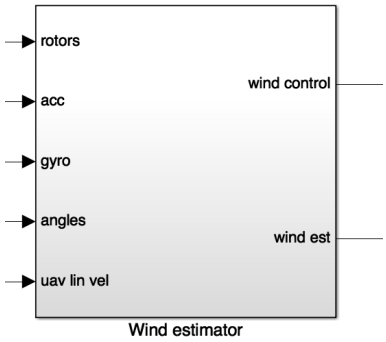

Figure 4: Wind estimator model

3. Results and conclusion

Values of the simulated model come from the *X4-MaG* drone, see [6]. The robustness properties against the external disturbances and against the model uncertainty (coming from the uncertainty of the the aerodynamic coefficients), and the finite-time wind estimation proof against the sensors noise are extensively discussed in [1], [3]. For this reason, the simulations below focus their attention on the advantages in having wind estimates as input to the controller, instead of covering as many example cases as possible.
Fig. 5 shows wind velocities in earth frame. Figures 6 ,7, 8, 9, 10, 11 show the (x, y, z) actual and reference positions, UAV linear velocities, UAV linear controls, (ϕ, θ, ψ) actual angles, UAV angular velocities, UAV angular controls for both fixed overestimated maximal wind velocity (tuned *a priori* to be (15, 15, 7) for wind velocities in (x, y, z) directions respectively), and for varying wind velocities (coming from the wind estimator). For a better comparison, control parameters are kept equals in the two cases.
Based on the results, the real-time wind estimates allow less control effort on the rotors (less chattering and smaller pitches) ensuring good trajectory tracking performance, due to the fact that the control gain is not anymore overestimated (less control effort avoids its saturation and corresponds to a lower wind value as input to the controller). As result, the controller with varying wind input presents a much smoother rotors behavior, see Figures 12, 13.

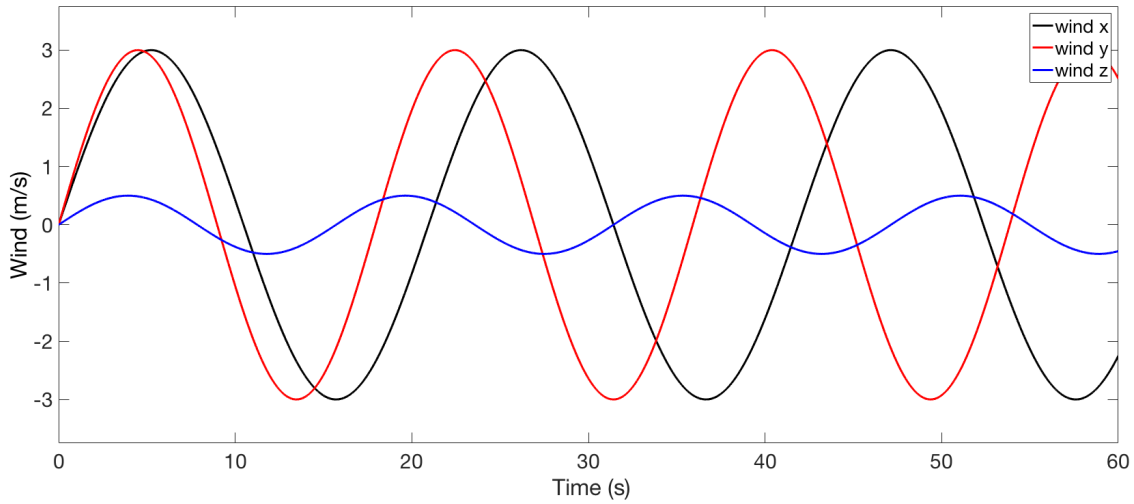

Figure 5: Wind velocities in earth frame.

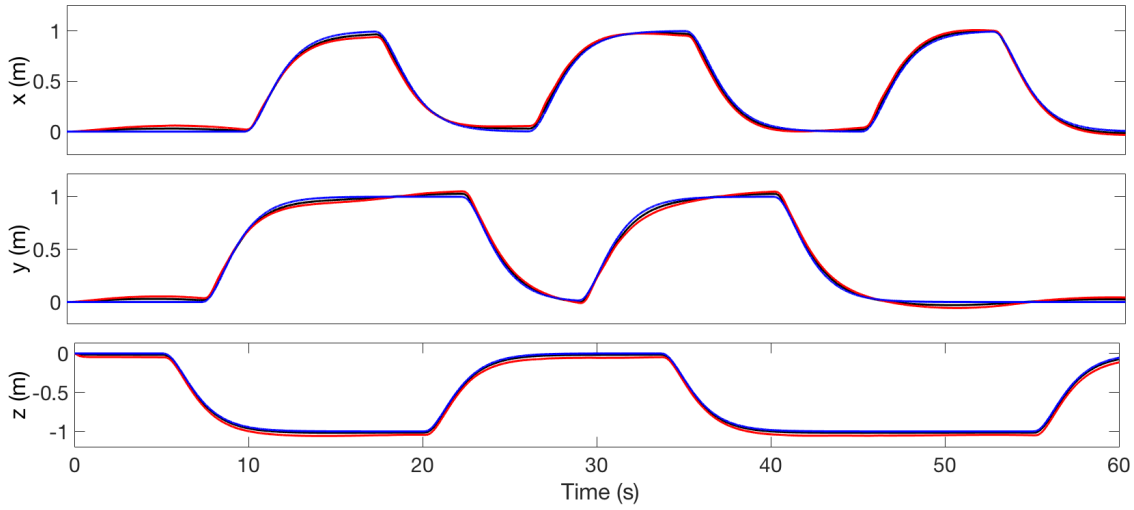

Figure 6: (x, y, z) UAV position. (—: Desired; —: controller with fixed wind; —: controller with varying wind).

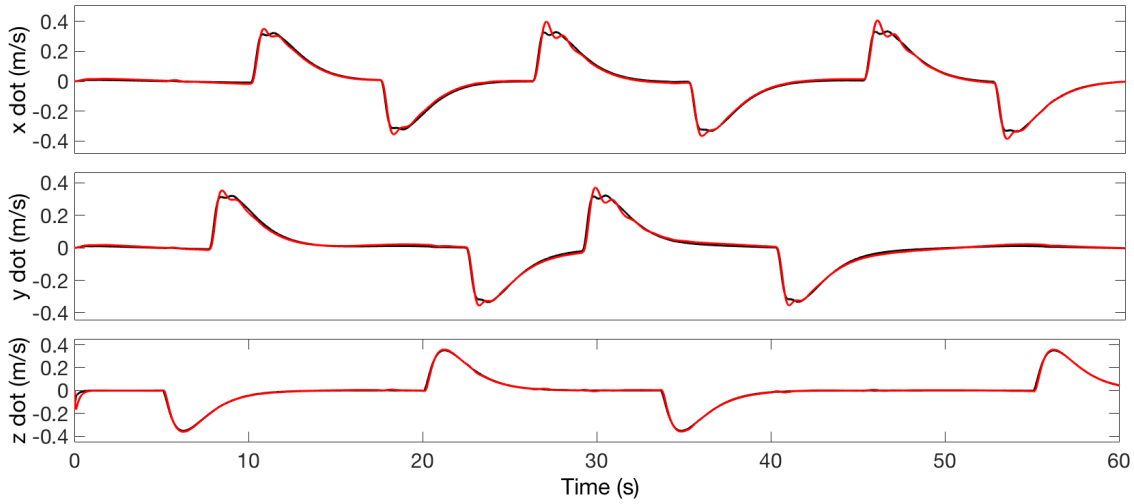

Figure 7: UAV linear velocities. (—: controller with fixed wind; —: controller with varying wind).

References

[1] G. Perozzi, D. Efimov, J.-M. Biannic, L. Planckaert, and P. Coton, "Wind rejection via quasi-continuous sliding mode technique to control safely a mini drone," in *7th European Conference for Aeronautics and Space Science (EUCASS)*, (Milan, Italy), July 2017.

[2] G. Perozzi, D. Efimov, J.-M. Biannic, L. Planckaert, and P. Coton, "On sliding mode control design for UAV using realistic aerodynamic coefficients," in *56th IEEE Conference on Decision and Control*, (Melbourne, Australia), pp. 5403–5408, Dec. 2017.

[3] G. Perozzi, D. Efimov, J.-M. Biannic, L. Planckaert, and P. Coton, "Wind estimation algorithm for quadrotors using detailed aerodynamic coefficients," in *American Control Conference*, (Milwaukee, WI, USA), June 2018.

[4] G. Perozzi, "A toolbox for quadrotors: from aerodynamic science to control theory," tech. rep., Jan. 2018. https://hal.inria.fr/hal-01696344.

[5] G. Perozzi, "CW-Quad Toolbox user's guide," tech. rep., Jan. 2018. https://hal.inria.fr/hal-01696347v2.

[6] "RT-MaG Project, an open-source toolbox for real-time robotic applications." http://www.gipsa-lab.fr/projet/RT-MaG/index.php.

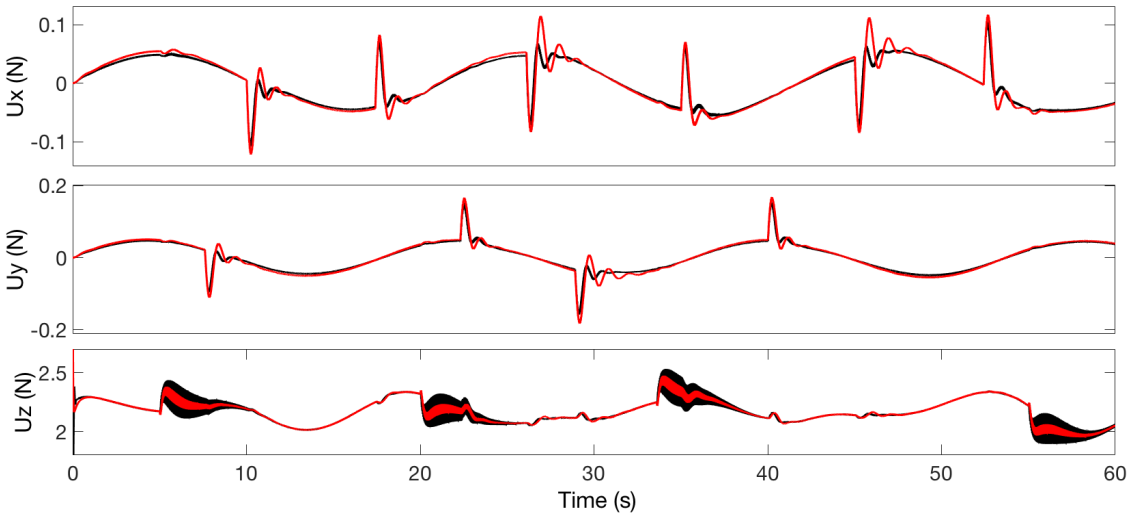

Figure 8: UAV linear controls. (—: controller with fixed wind; —: controller with varying wind).

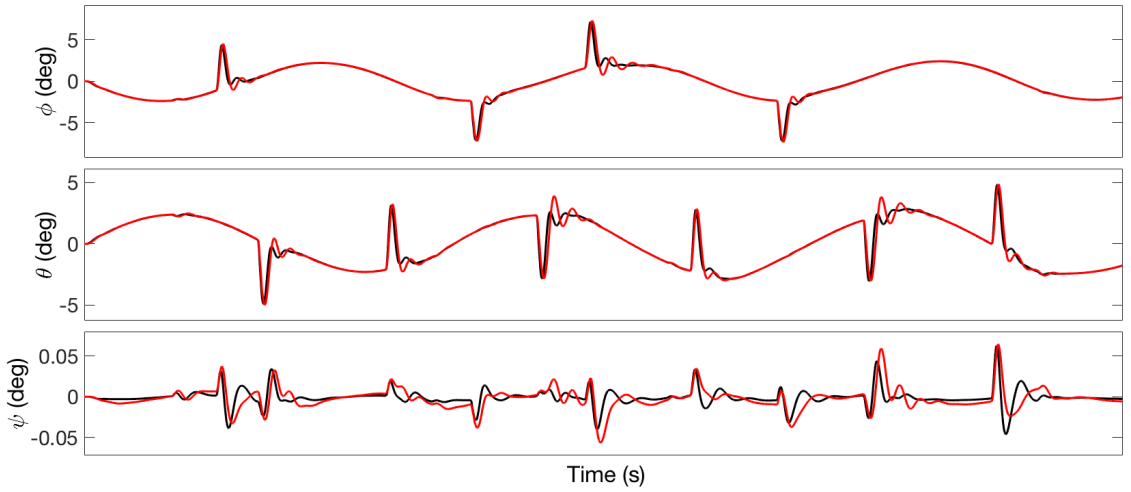

Figure 9: UAV angles. (—: controller with fixed wind; —: controller with varying wind).

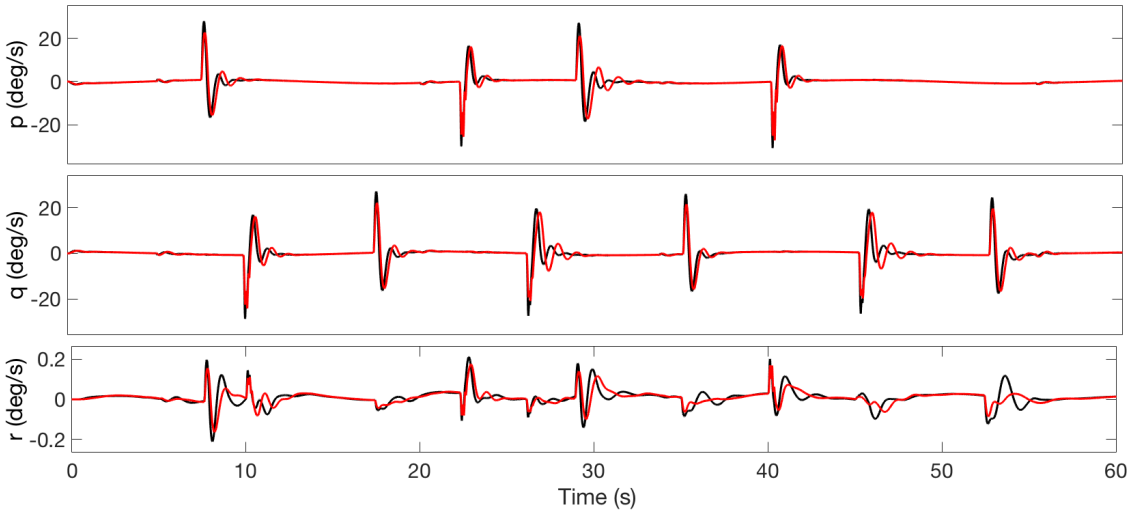

Figure 10: UAV angular velocities. (—: controller with fixed wind; —: controller with varying wind).

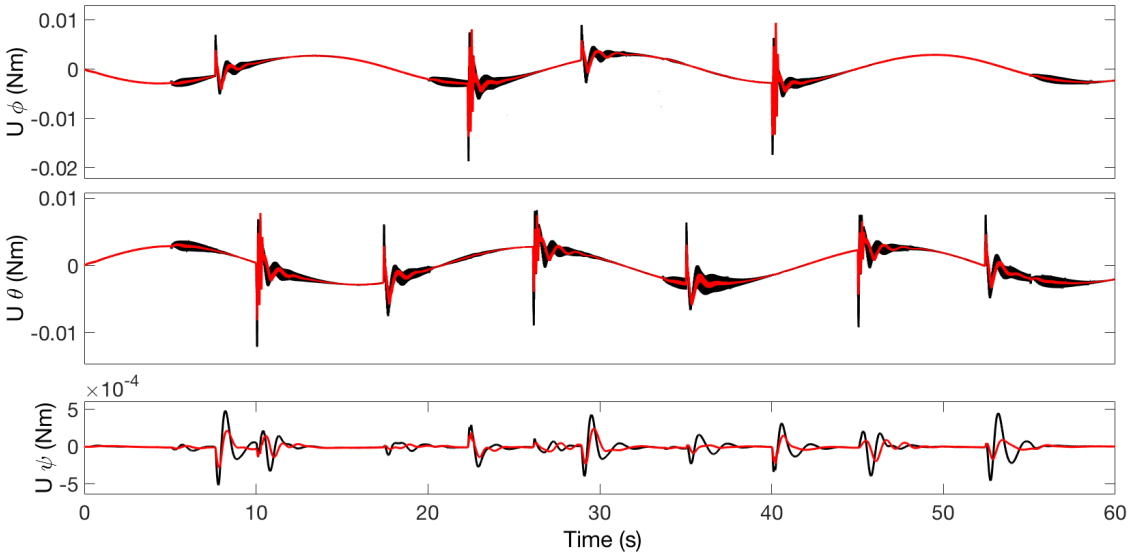

Figure 11: UAV angular controls. (—: controller with fixed wind; —: controller with varying wind).

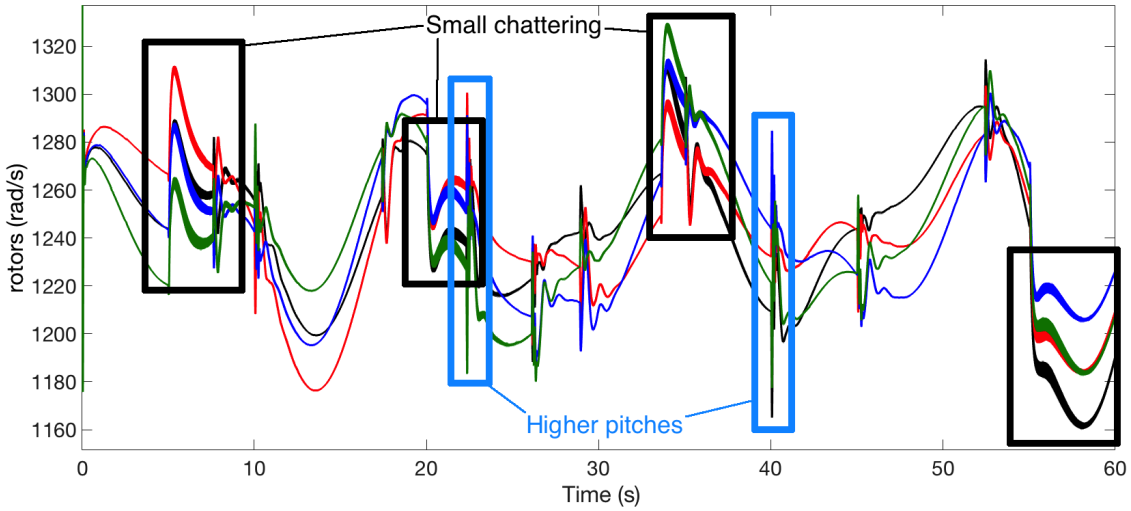

Figure 12: Angular rotor velocities for SMC controller with fixed wind values as input.

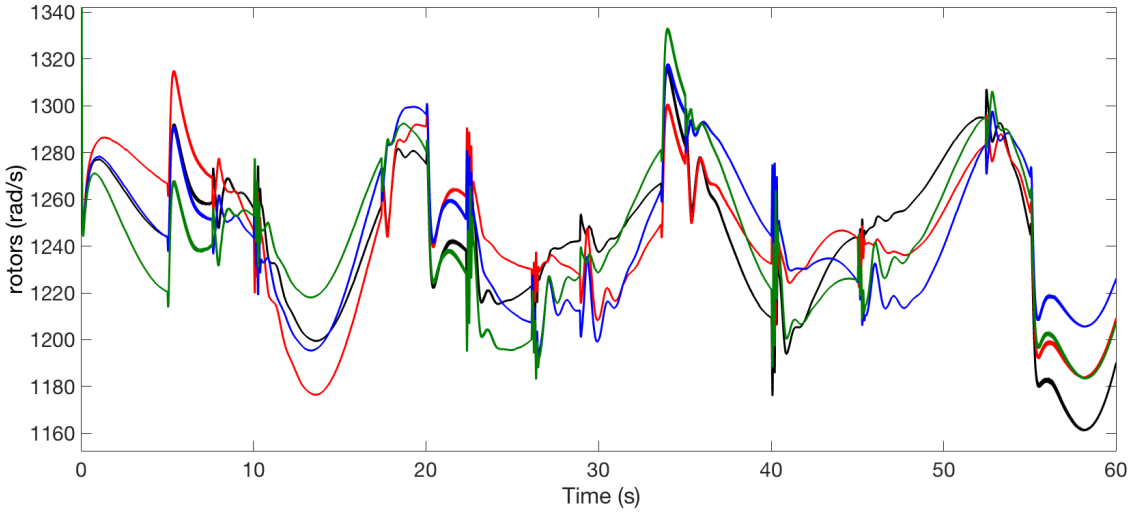

Figure 13: Angular rotor velocities for SMC controller with varying wind values as input.