Extended Simulations for the Observer-based Control of Position and Tension for an Aerial Robot Tethered to a Moving Platform

Marco Tognon, Antonio Franchi

Extended Simulations for the Observer-based Control of Position and Tension for an Aerial Robot Tethered to a Moving Platform

Technical Attachment to:
M. Tognon, S. S. Dash, and A. Franchi
"Observer-based Control of Position and Tension for an Aerial Robot Tethered to a Moving Platform"
published in the *IEEE Robotics and Automation Letters* and additionally presented at *2016 IEEE ICRA conference*

Marco Tognon[1,2] and Antonio Franchi[1,2]

This document is a technical attachment to [1] as an extension of the simulation's section.

I. MODELING WITH LAGRANGIAN FORMALISM

To derive the dynamic equations of $\mathbf{q}$ using the Lagrangian approach, we first compute the kinetic and potential energies, which are

$$K = \frac{1}{2} m_R \dot{\mathbf{p}}_R^W \cdot \dot{\mathbf{p}}_R^W + \frac{1}{2} J_W \dot{\vartheta}_W^2 = \frac{1}{2} m_R \dot{\mathbf{p}}_R^W \cdot \dot{\mathbf{p}}_R^W + \frac{1}{2} \bar{J}_W \dot{l}^2$$
$$V = m_R g \mathbf{p}_R^W \cdot \mathbf{e}_3,$$

respectively, where $\mathbf{p}_R^W = \mathbf{p}_C^W + l\mathbf{R}_C\mathbf{d}^C$ and

$$\dot{\mathbf{p}}_R^W = \mathbf{R}_C \left(\dot{\mathbf{p}}_C^C + l\boldsymbol{\Omega}_C\mathbf{d}^C + \dot{l}\mathbf{d}^C + l\dot{\mathbf{d}}^C \right), \tag{1}$$

$\boldsymbol{\Omega}_C = [\boldsymbol{\omega}_C]_\times$, $\bar{J}_W = J_W / r_W^2$, $\mathbf{e}_3 = [0\ 0\ 1]^T$, and $g \approx 9.81$ is the gravitational constant. Finally, the generalized forces that perform work on $\mathbf{q}$ are

$$\mathbf{Q} = \begin{bmatrix} -\frac{\partial \mathbf{p}_R^W}{\partial \mathbf{q}}^T \mathbf{R}_R \mathbf{e}_3 & \mathbf{e}_1 \end{bmatrix} \mathbf{u}_t = \mathbf{G}(\mathbf{q}, \mathbf{R}_R, \mathbf{R}_C)\mathbf{u}_t,$$

where $\mathbf{e}_3 = [0\ 0\ 1]^T$, $\mathbf{e}_1 = [1\ 0\ 0]^T$, $\mathbf{u}_t = [f_R\ \bar{\tau}_W]^T$ and $\bar{\tau}_W = \tau_W / r_W$. Applying the Euler-Lagrange equation [2] to K and V we obtain

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{c} + \mathbf{g} + \mathbf{n} = \mathbf{G}\mathbf{u}_t, \tag{2}$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{3\times 3}$ is the positive definite inertia matrix, $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{p}}_C^C, \boldsymbol{\omega}_C)$ contains all the centrifugal/Coriolis terms, $\mathbf{g}(\mathbf{q}, \mathbf{R}_C)$ contains all the gravity terms and $\mathbf{n}(\mathbf{q}, \ddot{\mathbf{p}}_C^C, \dot{\boldsymbol{\omega}}_C)$ contains the terms depending on the acceleration of the moving platform. In equation (6) of Table I we report the full expression of all the terms in (2).

[1]LAAS-CNRS, 7 Avenue du Colonel Roche, F-31400 Toulouse, France. mtognon@laas.fr, antonio.franchi@laas.fr
[2]Univ de Toulouse, LAAS, F-31400 Toulouse, France
This work has been partially funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 644271 AEROARMS.

II. EXTENDED SIMULATIONS

In this section we provide several additional detailed simulation results in order to test the validity of the proposed method. We consider an aerial vehicle with nominal mass $m_R = 1[\text{Kg}]$ and inertia $\mathbf{J}_R = \text{diag}(0.25, 0.25, 0.25)[\text{Kg m}^2]$. The nominal winch radius and inertia are $r_W = 0.2[\text{m}]$ and $J_W = 0.15[\text{Kg m}^2]$, respectively.

We set $\mathbf{k}_i$ and $\mathbf{k}_j$ such that the error dynamics $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$ have poles in $(-1,-2,-3,-4)$ and $(-1,-2)$ respectively. While for the observer we choose $\varepsilon = 0.1$ and (α_1, α_2) such that $s^2 + \alpha_1 s + \alpha_2$ has roots $(-3,-4)$. Those values guarantees the stability and ensure a sufficiently fast exponential tracking.

In the simulation we reproduce a possible real application scenario in which the task consists to let the moving platform and the aerial vehicle cooperatively patrol an area. The platform follows a certain smooth trajectory in the 3D space with $\mathbf{x}_C$ tangent to the curve and $\mathbf{y}_C$ perpendicular to $\mathbf{z}_W$, as if, e.g., a ground vehicle is following a mountain road. We require the aerial vehicle at time t_0 to takeoff from the moving platform, at time t_{circ} to circle above the platform at a certain altitude, and at time t_{land} to land at the same takeoff point on the moving platform. Moreover we ask that the yaw angle of the aerial vehicle follows the one of the platform, in this way the two vehicles always head to the same direction. Finally, during the takeoff the desired stress must go from a small initial tension of $0.5[\text{N}]$ to a steady-state value of $3[\text{N}]$ that is kept for the whole circling phase. Finally the tension has to go back to the initial value during the landing.

In order to fully validate our control strategy we tested it on several different *non ideal* conditions, reported in the following:

- Case A) initial position and estimation errors,
- Case B) parametric variations,
- Case C) partial measurements of the moving platform trajectory,
- Case D) noisy sensor measurements,
- Case E) non zero offset between the tether and the center of gravity of the aerial vehicle,
- Case F) non diagonal inertia matrix $\mathbf{J}_R$,
- Case G) saturation of the inputs,

$$
\begin{aligned}
\mathbf{M} &= \begin{bmatrix} \bar{J}_W + m_R & 0 & 0 \\ 0 & -l^2 m_R \cos^2\delta & 0 \\ 0 & 0 & l^2 m_R \end{bmatrix} \\
\mathbf{c} &= [c_1 \ c_2 \ c_3]^T \\
c_1 &= -m_R(l\dot{\delta}^2 + l\omega_{Cx}^2 + l\omega_{Cz}^2 + \dot{x}_C\omega_{Cz}\sin\delta - \dot{z}_C\omega_{Cx}\sin\delta + l\dot{\varphi}^2\cos^2\delta + l\omega_{Cy}^2\cos^2\delta - l\omega_{Cz}^2\cos^2\delta - 2l\dot{\delta}\omega_{Cz}\cos\varphi \\
&\quad + 2l\dot{\delta}\omega_{Cx}\sin\varphi + \dot{y}_C\omega_{Cz}\cos\delta\cos\varphi - \dot{z}_C\omega_{Cy}\cos\delta\cos\varphi - 2l\dot{\varphi}\omega_{Cy}\cos^2\delta + \dot{x}_C\omega_{Cy}\cos\delta\sin\varphi - \dot{y}_C\omega_{Cx}\cos\delta\sin\varphi \\
&\quad - l\omega_{Cx}^2\cos^2\delta\cos^2\varphi + l\omega_{Cz}^2\cos^2\delta\cos^2\varphi - 2l\dot{\varphi}\omega_{Cx}\cos\delta\cos\varphi\sin\delta + 2l\omega_{Cx}\omega_{Cy}\cos\delta\cos\varphi\sin\delta \\
&\quad - 2l\dot{\varphi}\omega_{Cz}\cos\delta\sin\delta\sin\varphi + 2l\omega_{Cy}\omega_{Cz}\cos\delta\sin\delta\sin\varphi - 2l\omega_{Cx}\omega_{Cz}\cos^2\delta\cos\varphi\sin\varphi) \\
c_2 &= -lm_R\cos\delta(\dot{z}_C\omega_{Cy}\sin\varphi - \dot{y}_C\omega_{Cz}\sin\varphi - 2\dot{\varphi}\dot{l}\cos\delta + 2\dot{l}\omega_{Cy}\cos\delta + \dot{x}_C\omega_{Cy}\cos\varphi - \dot{y}_C\omega_{Cx}\cos\varphi + 2l\dot{\delta}\dot{\varphi}\sin\delta \\
&\quad + l\omega_{Cx}\omega_{Cz}\cos\delta - 2l\dot{\delta}\omega_{Cy}\sin\delta + 2\dot{l}\omega_{Cx}\cos\varphi\sin\delta + 2\dot{l}\omega_{Cz}\sin\delta\sin\varphi + l\omega_{Cx}^2\cos\delta\cos\varphi\sin\varphi \\
&\quad - l\omega_{Cz}^2\cos\delta\cos\varphi\sin\varphi + 2l\dot{\delta}\omega_{Cx}\cos\delta\cos\varphi + 2l\dot{\delta}\omega_{Cz}\cos\delta\sin\varphi + l\omega_{Cy}\omega_{Cz}\cos\varphi\sin\delta \\
&\quad - l\omega_{Cx}\omega_{Cy}\sin\delta\sin\varphi - 2l\omega_{Cx}\omega_{Cz}\cos\delta\cos^2\varphi) \\
c_3 &= lm_R(2\dot{\delta}\dot{l} + 2\dot{l}\omega_{Cx}\sin\varphi + (l\dot{\varphi}^2\sin(2\delta))/2 + (l\omega_{Cy}^2\sin(2\delta))/2 - (l\omega_{Cz}^2\sin(2\delta))/2 - \dot{x}_C\omega_{Cz}\cos\delta + \dot{z}_C\omega_{Cx}\cos\delta \\
&\quad - 2\dot{l}\omega_{Cz}\cos\varphi + l\omega_{Cx}\omega_{Cy}\cos\varphi + l\omega_{Cy}\omega_{Cz}\sin\varphi + \dot{y}_C\omega_{Cz}\cos\varphi\sin\delta - \dot{z}_C\omega_{Cy}\cos\varphi\sin\delta - l\dot{\varphi}\omega_{Cy}\sin(2\delta) \\
&\quad + \dot{x}_C\omega_{Cy}\sin\delta\sin\varphi - \dot{y}_C\omega_{Cx}\sin\delta\sin\varphi - l\omega_{Cx}^2\cos\delta\cos^2\varphi\sin\delta + l\omega_{Cz}^2\cos\delta\cos^2\varphi\sin\delta + 2l\dot{\varphi}\omega_{Cx}\cos^2\delta\cos\varphi \\
&\quad - 2l\omega_{Cx}\omega_{Cy}\cos^2\delta\cos\varphi + 2l\dot{\varphi}\omega_{Cz}\cos^2\delta\sin\varphi - 2l\omega_{Cy}\omega_{Cz}\cos^2\delta\sin\varphi - 2l\omega_{Cx}\omega_{Cz}\cos\delta\cos\varphi\sin\delta\sin\varphi) \\
\mathbf{g} &= m_R g(\mathbf{R}_C\frac{\partial \mathbf{p}_R^W}{\partial \mathbf{q}})^T\mathbf{e}_3 \\
\mathbf{n} &= [n_1 \ n_2 \ n_3]^T \\
n_1 &= m_R(\ddot{x}_C\cos\delta\cos\varphi - \ddot{y}_C\sin\delta + \ddot{z}_C\cos\delta\sin\varphi) \\
n_2 &= -lm_R\cos\delta(\ddot{x}_C\sin\varphi - \ddot{z}_C\cos\varphi + l\dot{\omega}_{Cy}\cos\delta + l\dot{\omega}_{Cx}\cos\varphi\sin\delta + l\dot{\omega}_{Cz}\sin\delta\sin\varphi) \\
n_2 &= -lm_R(\ddot{y}_C\cos\delta - l\dot{\omega}_{Cx}\sin\varphi + \ddot{x}_C\cos\varphi\sin\delta + \ddot{z}_C\sin\delta\sin\varphi + l\dot{\omega}_{Cz}\cos\varphi)
\end{aligned}
\tag{6}
$$

TABLE I: Full expression of the terms in (2), where $\mathbf{p}_C = [x_C \ y_C \ z_C]^T$ and $\boldsymbol{\omega}_C = [\omega_{Cx} \ \omega_{Cy} \ \omega_{Cz}]$.

Case H) motor time constant,

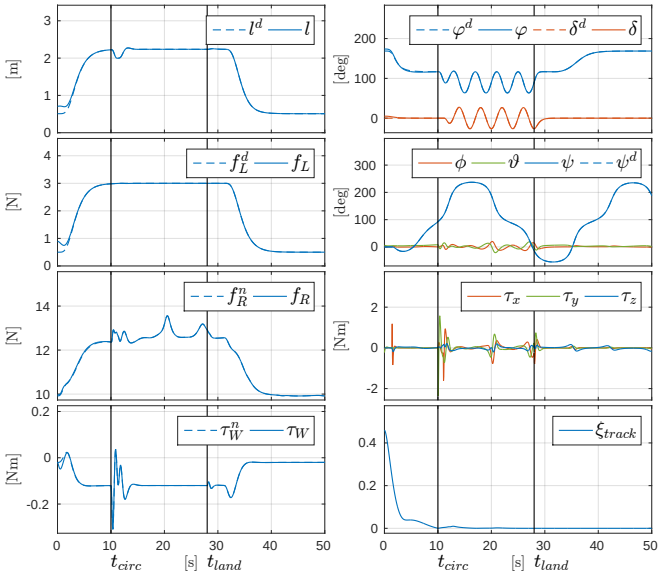
Case I) comparison with a controller based on standard hierarchical techniques.

For each case we show the control performances plotting the tracking of each output of interest, the global tracking error ξ_{track} computed as the sum of each errors, and the inputs. Concerning f_R and τ_W we also show the nominal input coming from the flatness, ${f_R}^n$ and τ_W^n, that should be applied to obtain the desired output tracking in the nominal case. We also show the observer performances comparing the estimated state and the actual one. The estimation error $e^{estimation}$ is simply calculated as the sum of the estimation error for each entry of the state. Finally we display the trajectories of the aerial vehicle and of the moving platform in the world frame and with respect to $\mathscr{F}_C$. In the 3D plots the position of the moving platform and of the aerial vehicle in some particular instants are represented with a triangle and a square respectively.

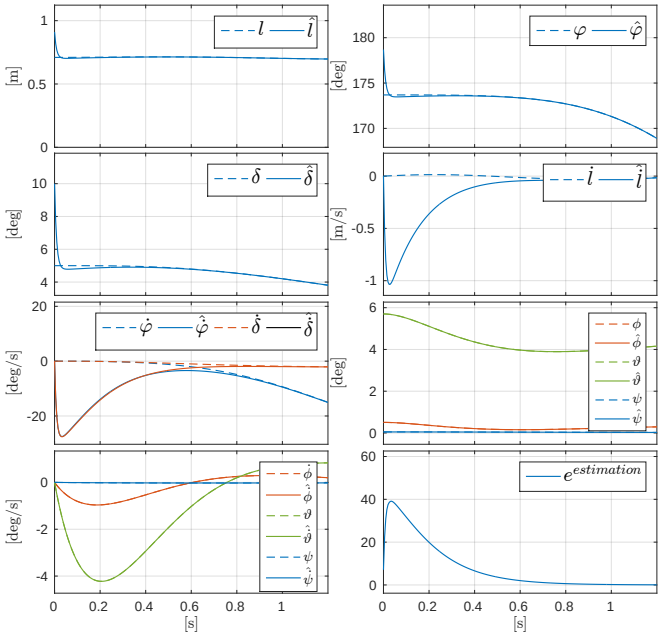
A. *Initial errors*

In this section we want to show the closed loop stability of the system in dynamic condition even with some initialization error. The system starts with an error on l of 0.1[m], on φ and δ of 2[°] and on f_L of 0.5[N]. Similarly the initialization of the observer is done with an error of 0.2[m] on $\hat{l}$, of 5[°] on $\hat{\varphi}$ and $\hat{\delta}$, while their velocity are initialized to zero.

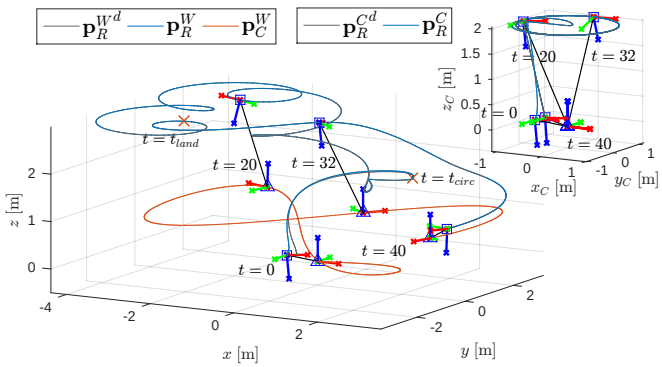
In Fig. 1 one can see that after the convergence of the observer, that takes less than one second, the controller exponentially steers the outputs along the desired trajectories, while the moving platform is following its own dynamic trajectory.



(a) Controller performances.



(b) Observer performances. We show here only the first second because after the estimated state follows the actual one with high fidelity for all the remaining simulation.



(c) Trajectories visualization.

Fig. 1: Simulation: initial errors.

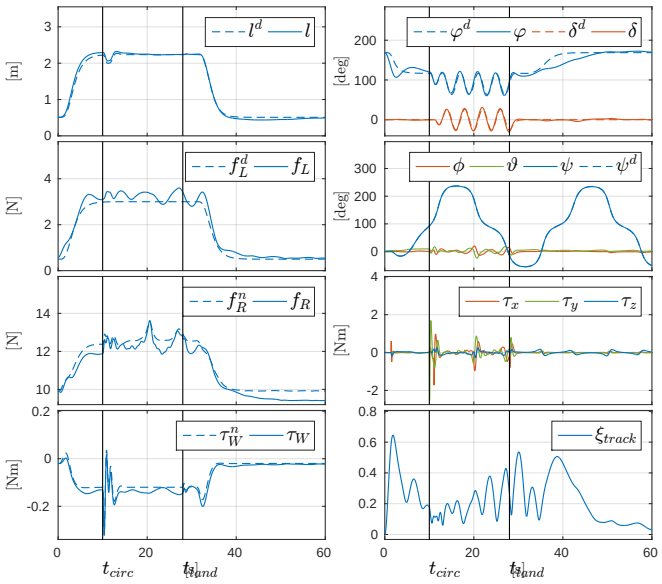
B. *Parametric variations*

The purpose of the next sections is to investigate the robustness of the proposed method. In particular in this one, we consider some parameter variation between the real model and controller/observer. Indeed in a real scenario we can not know exactly each parameter of the system, thus the controller and observer would be based on some parameter value different from the real one.

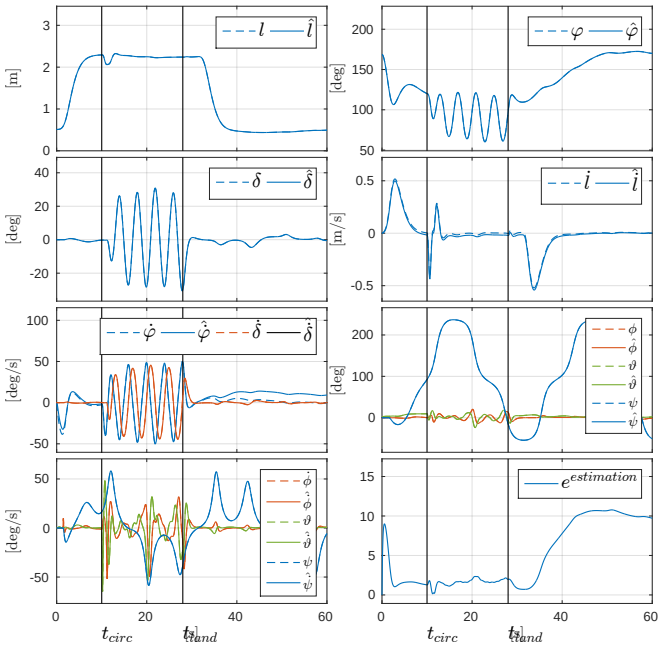
Fig. 2 displays the results of the simulation with a parametric variation of the 5% for each entry, i.e., m_R, $\mathbf{J}_R$, J_W and r_W. In order to partially compensate the effects of the uncertainties we added in the controller an integral term with gain $k_I = 3$.

We can notice that due to the uncertainty of the model we have some nonzero errors in the tracking and in the estimation of the state. Nevertheless the error system remains stable and thanks to the integrator terms, during the landing maneuver we obtained a decreasing tracking error that allows a correct landing of the aerial vehicle.

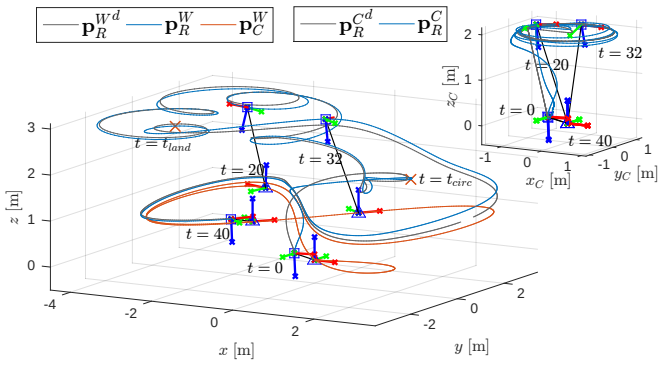
We performed additional extensive simulations in which we observed that the system remains stable up to a parametric variation of the 20%, after this value the system results unstable. However notice that in reality those parameters are very well measurable with small errors, certainly lower than the 20%.



(a) Controller performances.



(b) Observer performances.



(c) Trajectories visualization.

Fig. 2: Simulation: parametric variations.

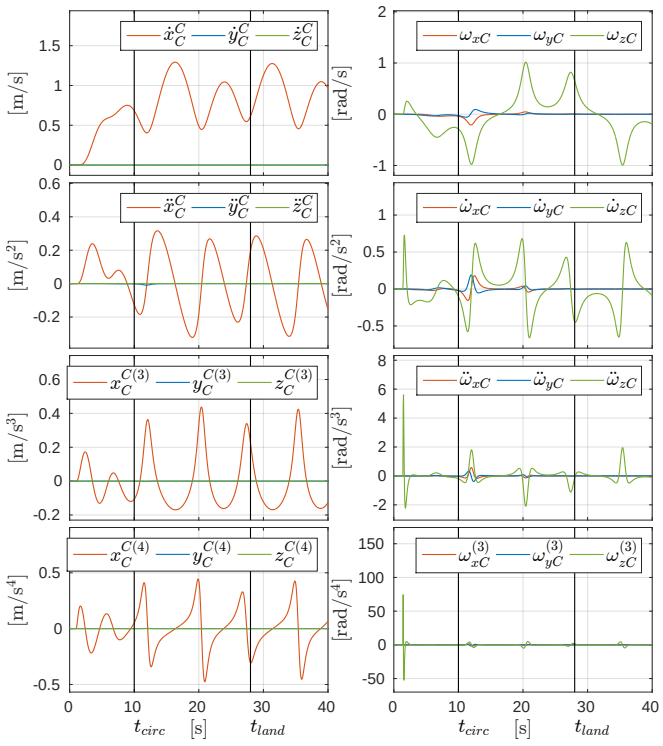


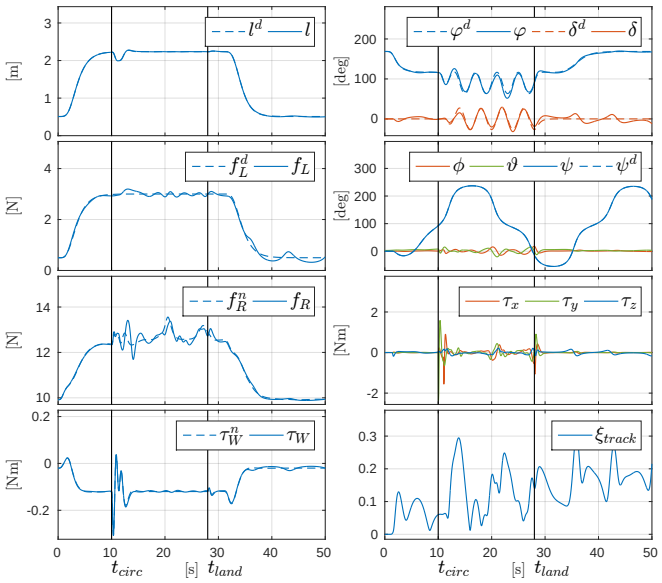
Fig. 3: Plot of $\mathbf{x}_C^i$, for $i = 1,2,3,4$. In Simulation C all the variables in the last five plots are considered zero by the controller and the observer.

C. *Limited knowledge of* $\mathbf{p}_C^W(t)$

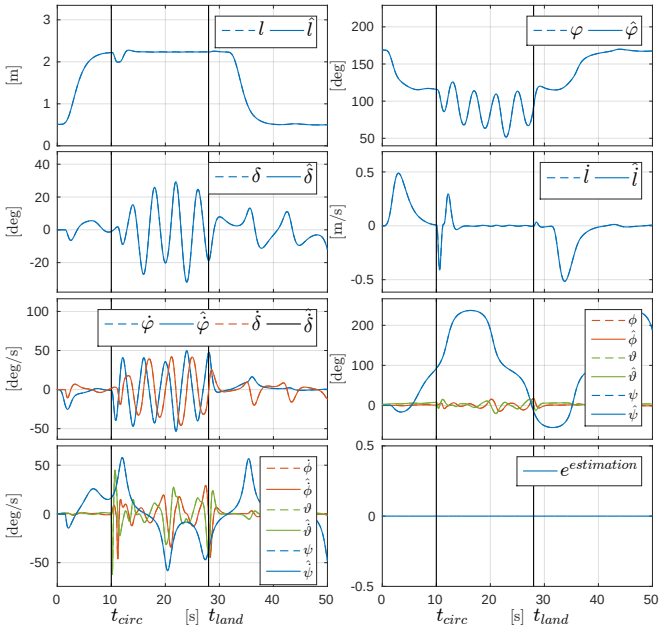
In the Sections III and IV of [1] we saw that the knowledge of $\mathbf{X}_C^4$ is needed in order to compute the control action. In other words, to obtain a perfect tracking one has to know the derivative of $\mathbf{p}_C^C(t)$ up to the fourth order and of $\boldsymbol{\omega}_C(t)$ up to the third order. Although those variables have to be known to obtain zero tracking error, actually, without a posteriori knowledge of the trajectory or the model and control inputs of the system, it is difficult to measure the higher-order derivatives. Nevertheless, in this section we want to show that even with only a partial measurement of $\mathbf{X}_C^4$ the system stays stable and the tracking error remains bounded.

In particular, Fig. 4 shows the results obtained considering measured only $\boldsymbol{\omega}_C(t)$, and $\mathbf{p}_C^C(t)$ up to its second derivative, i.e., assuming $\mathbf{w}_1 = \mathbf{X}_C^4$ where $\mathbf{x}_C^4 = \mathbf{x}_C^3 = (\mathbf{0}_3, \mathbf{0}_3)$. Indeed, for a real moving platform, a standard onboard sensorial configuration, such as optical flow, IMU and magnetometer, is sufficient to obtain $\boldsymbol{\omega}_C(t)$ and $\mathbf{p}_C^C(t)$ up to its second derivative.

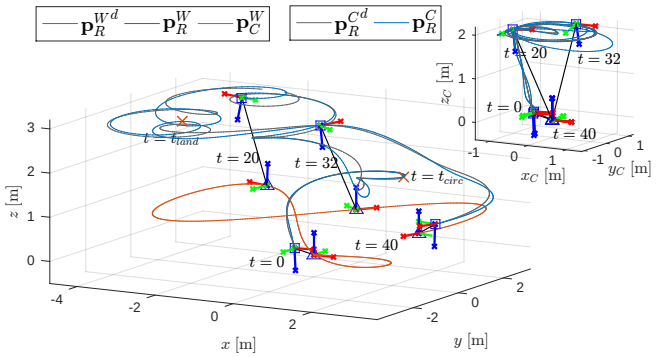
In Fig. 4b we can observe that the estimation error is almost constantly zero even if $\dot{\boldsymbol{\omega}}_C$ is assumed zero. While in Fig. 4a one can notice that the outputs oscillates around the desired value and the tracking error does not go to zero but remains bounded under a reasonable threshold. Nevertheless, with a more "aggressive" platform trajectory the negative effects would be more significants. In Fig. 3 the entries of $\mathbf{x}_C^i$ for $i = 1,2,3,4$ are plotted. The last five entries are assumed zero by the observer and the controller.



(a) Controller performances.



(b) Observer performances.



(c) Trajectories visualization.

Fig. 4: Simulation: limited measurements of the moving platform trajectory.

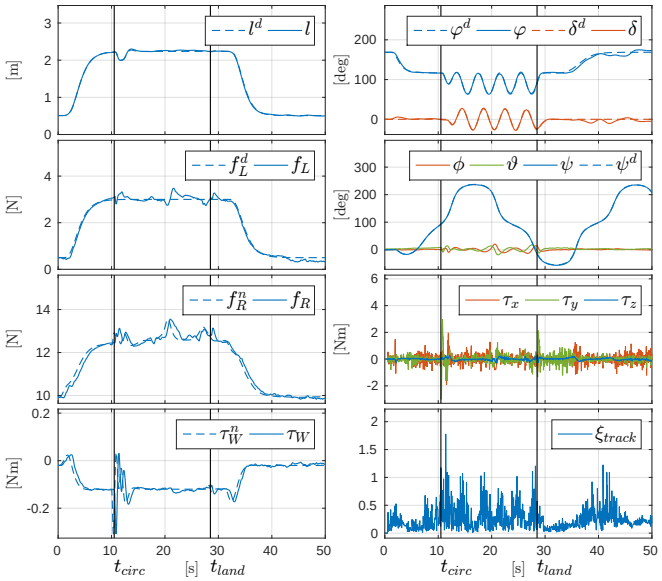
#	Type	Measurement	Noise variance
w_2	abs. encoder	$\vartheta_W \approx l$	0.008[rad]
w_3	abs. encoder	φ	0.008[rad]
w_4	abs. encoder	δ	0.008[rad]
$\mathbf{w}_5$	accelerometer	$\mathbf{R}_R(\ddot{\mathbf{p}}_R^W + g\mathbf{e}_3)$	-
$\mathbf{w}_6$	gyroscope	$\boldsymbol{\omega}_R$	0.01[rad/s]
$\mathbf{w}_7$	magnetometer	$\mathbf{R}_R\mathbf{h}^W$	-
$\mathbf{W}_R$	complementary filter	$\mathbf{R}_R$	0.001

TABLE II: List of sensors.

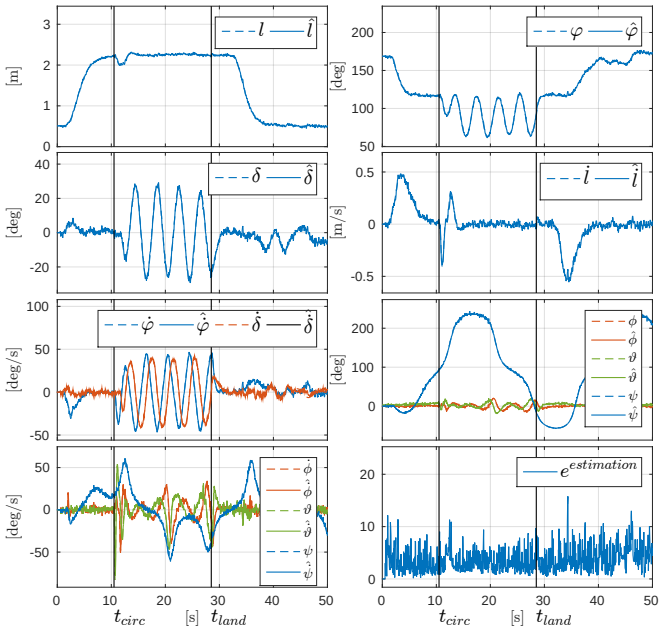
D. Noise on the measurements

In this last section we investigate the robustness of the proposed method with the presence of noise in the measurements. Tab. II gathers the variance magnitude set for each measurement. For the encoder and the gyroscope we set some reasonable value found in the literature [3]. On the other hand, instead of adding noise on $\mathbf{w}_5$ and $\mathbf{w}_7$ we preferred inserting the noise directly in the measure of the rotational matrix $\mathbf{R}_R$, i.e., in $\mathbf{W}_R$. This is done because the direct measure of $\mathbf{R}_R$ using the accelerometer and the magnetometer is normally filtered with the gyroscope [4], in order to obtain a less noisy estimation of both $\mathbf{R}_R$ and $\boldsymbol{\omega}_R$. The noise added directly to $\mathbf{R}_R$ is comparable to the one we would obtain after the filtering.

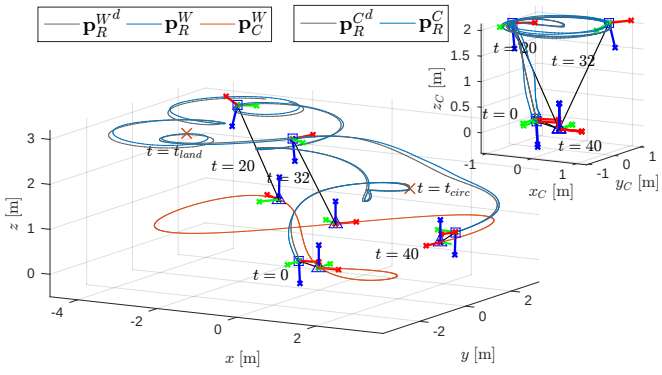
From Fig. 5 we can observe that the estimated state shows some noise but the corresponding error remains limited. Due to the noisy component on the estimated state the outputs presents some oscillation as well, especially on the stress that seams to be the more sensitive output to the noise. Nevertheless the tracking error remains small and always bounded.



(a) Controller performances.



(b) Observer performances.



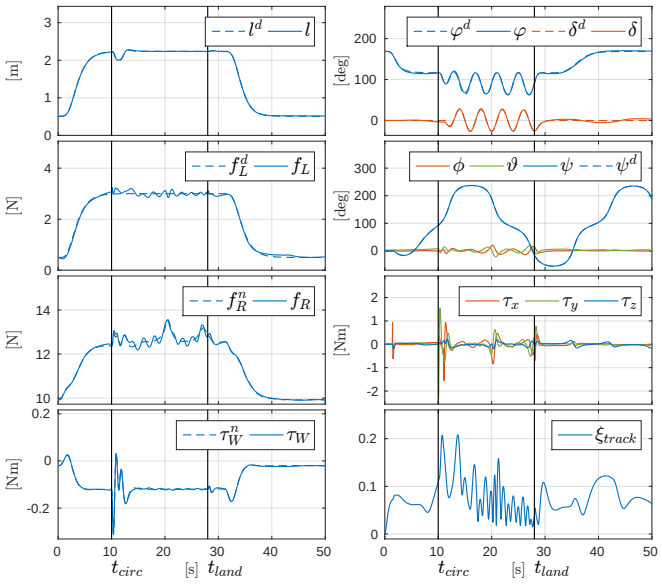
(c) Trajectories visualization.

Fig. 5: Simulation: noisy measurements.

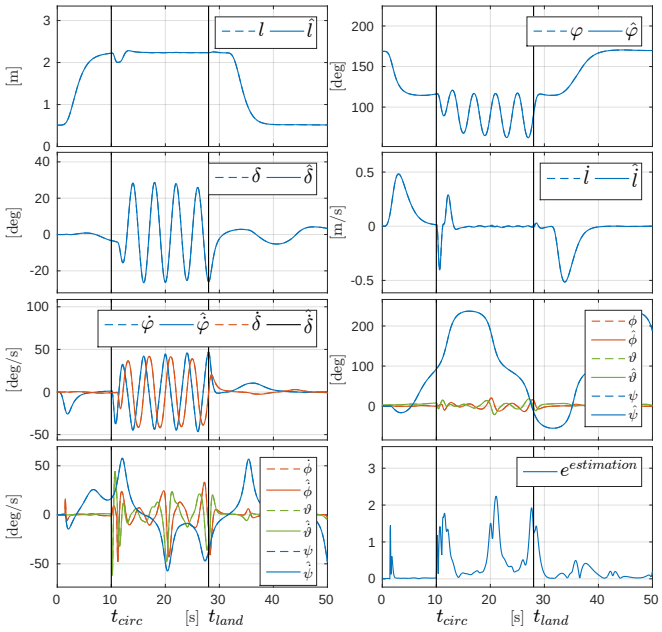
E. Tethered offset

Exact attachment of the link to the center of mass of the aerial vehicle is practically unfeasible. Therefore there will always be a non zero offset, although small, between the tether attachment and the center of gravity. This offset makes the translational and rotational dynamics of the aerial robot coupled and can potentially lead to the instability of the controlled system. In this section we want to show the robustness of the proposed method when the distance between the attaching point and the center of gravity of the aerial vehicle is non zero. In particular in this simulation the link is attached 5 [cm] vertically below O_R with respect to $\mathcal{F}_R$. As expected, the tracking error does not go to zero but however remains bounded, showing good tracking performances. Notice that the error is higher during the circling phase since this part of the global trajectory is very dynamical and the unmodeled effects due to the offset are larger. However we remark that a good mechanical design could make the tracking error almost negligible.

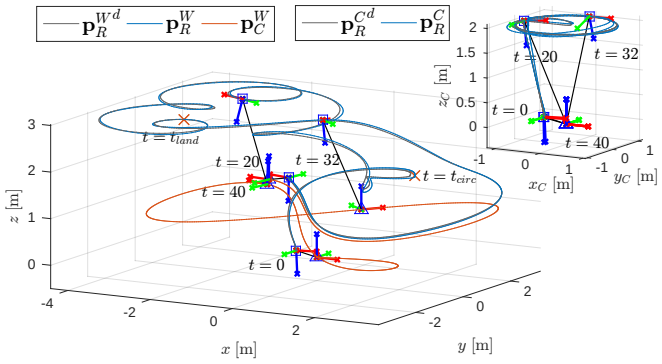
We tested the method with even larger offsets and we saw that the system remains stable up to a vertical offset of 30[cm], that is an exaggerated value for the system considered in the simulation (small-size quadrotor like vehicle). In fact, note that a larger quadrotor means a larger inertia which actually reduces the negative effects of the offset. In additional simulations, which are not reported here for the sake of brevity, we also tested the robustness of the method with a more general offset (not only vertical) noticing that, within some reasonable bounds, the system remains stable and with acceptable tracking performances.



(a) Controller performances.



(b) Observer performances.



(c) Trajectories visualization.

Fig. 6: Simulation: non zero offset between tether attachment and center of gravity of the aerial vehicle.

F. Nondiagonal inertia matrix

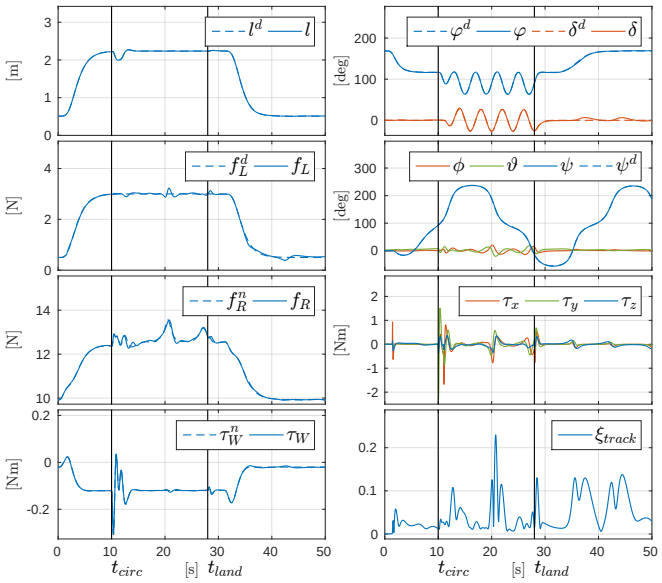
In the derivation of the model and of the controller as well, we assumed a diagonal inertia matrix. In this section we check the robustness of the method if the aerial vehicle has a non diagonal inertia matrix. In particular, in Fig. 7, we show the results for a test in which the real inertia matrix is

$$\mathbf{J}_R = \begin{bmatrix} 0.25 & 0.05 & 0.05 \\ 0.05 & 0.25 & 0.05 \\ 0.05 & 0.05 & 0.25 \end{bmatrix},$$

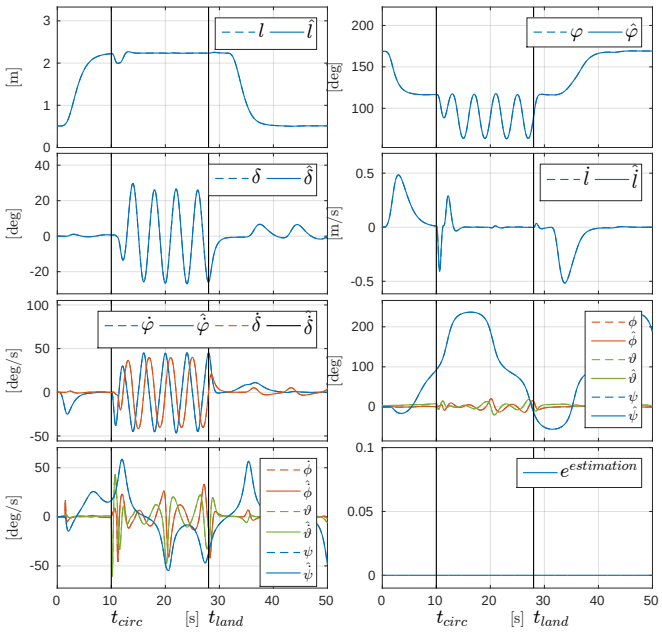
while the controller still assumes a diagonal inertial matrix.

One can observe that the tracking error is not exactly zero but is kept limited within a small bound. For the observer this does not constitute a non ideality, in fact the estimation error is constantly zero.

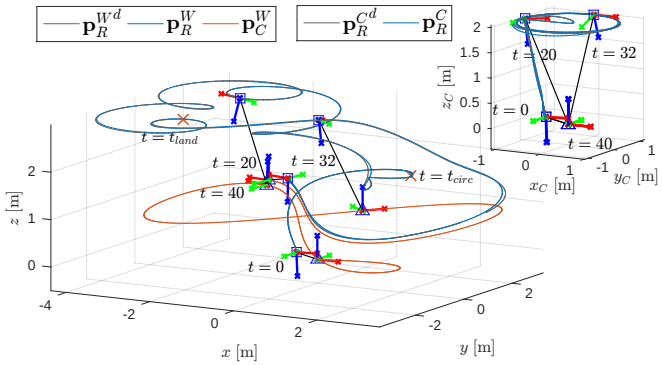
With further simulations we observed that the system remains stable up to a value of 0.15 in the off diagonal terms (60% of the values on the main diagonal). With larger values the system becomes unstable.



(a) Controller performances.



(b) Observer performances.



(c) Trajectories visualization.

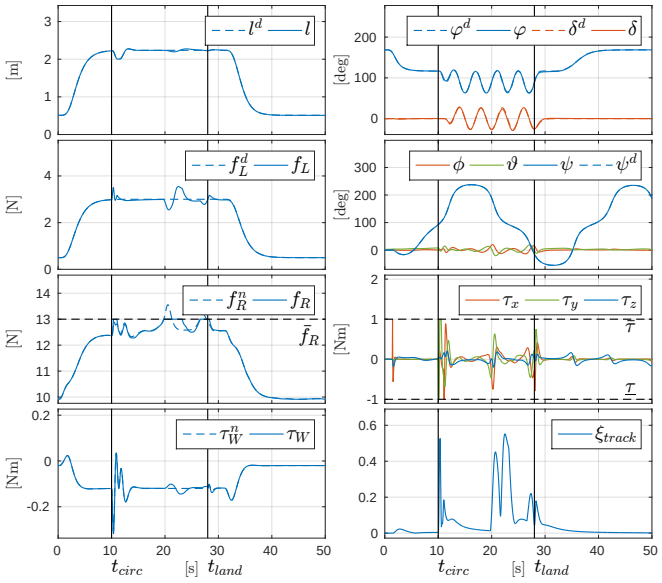
Fig. 7: Simulation: non-diagonal inertia matrix $\boldsymbol{J}_R$.

G. Input saturation

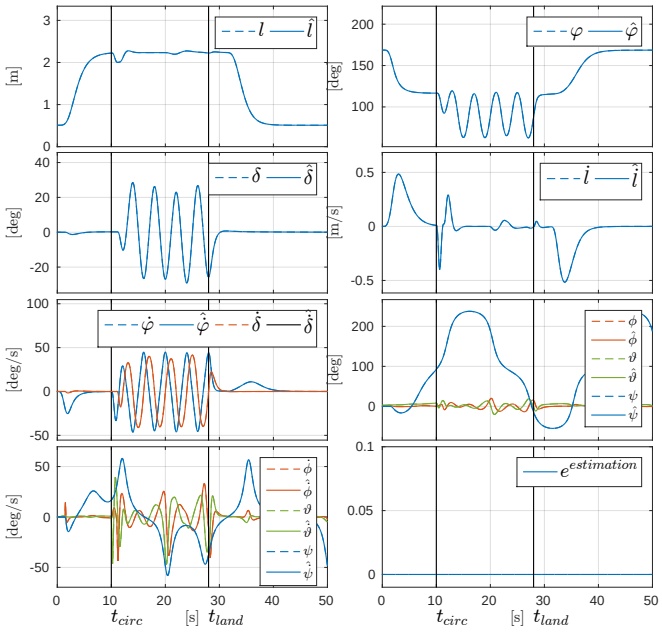
For how we planned the desired trajectory, the nominal input needed to track the desired outputs is always within the limits of the considered system. Indeed, exploiting the flatness, we are able to a priori check if the inputs exceed the minimum and maximum values. Nevertheless, in this section we want to show that the system is still stable if the inputs are hardly saturated for some instants. Thus we set some very restrictive limits on the input, i.e., $f_R \leq \bar{f}_R$ and $\underline{\tau} \leq \tau_i \leq \bar{\tau}$, where $i = x, y, z$, $\bar{f}_R = 13[\text{N}]$, $\underline{\tau} = -1[\text{Nm}]$ and $\bar{\tau} = 1[\text{Nm}]$. In order to let the saturation show up during execution we did not re-plan the desired trajectory.

In Fig. 8a it can be seen that the inputs are saturated for some time instants during the execution of the task. When the inputs are saturated the tracking error increases, but, as soon as the inputs come back within the limits, the error exponentially decreases to zero.

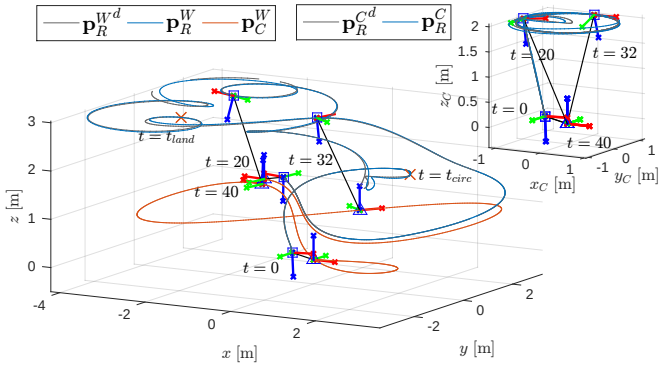
We stress again the fact that the saturation of the inputs can be avoided exploiting the flatness. Using the flatness one can check if the desired trajectory requires inputs that are too large. In the worst case one can re-plan the trajectory such that the input limits are respected.



(a) Controller performances.



(b) Observer performances.



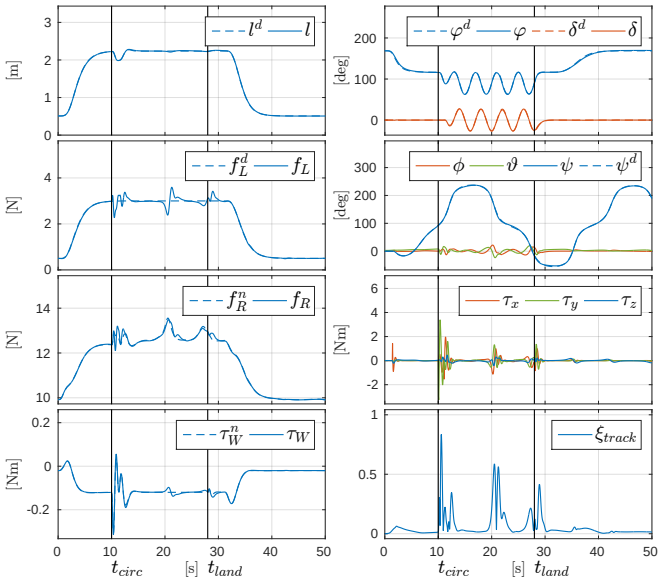
(c) Trajectories visualization.

Fig. 8: Simulation: saturation of the input.

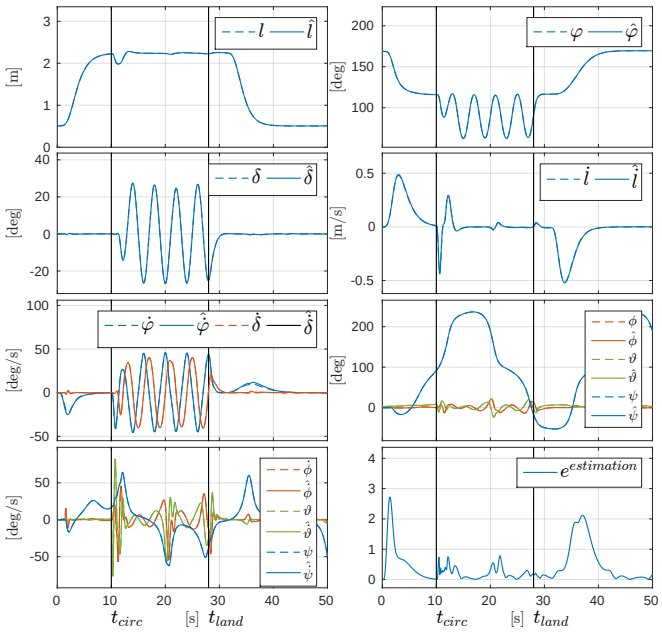
H. Motor time constant

With this simulation we want to further enlarge the set of non ideal models considered for the testing of the proposed control method. Considering an aerial vehicle actuated by rotating propellers, in this simulation we add the dynamical model of the motors described with a first order system characterized by a time constant of $\tau_M = 0.1[\text{s}]$. In practice the propeller dynamics inserts a frequency dependent phase shift between the commanded control input and the actuated one, whose amplitude depends on the time constant. In other words the models acts as a low pass filter on the commanded input, cutting its high frequency components. Those effects could dramatically decrease the performances or even make the system unstable. However, from Fig. 9, one can notice that our method is robust to the unmodeled effects of the propellers dynamics. Indeed, in some instant, where the trajectory is more dynamical and requires fast varying inputs, the tracking error increases but it is always bounded and at steady state converges to zero.

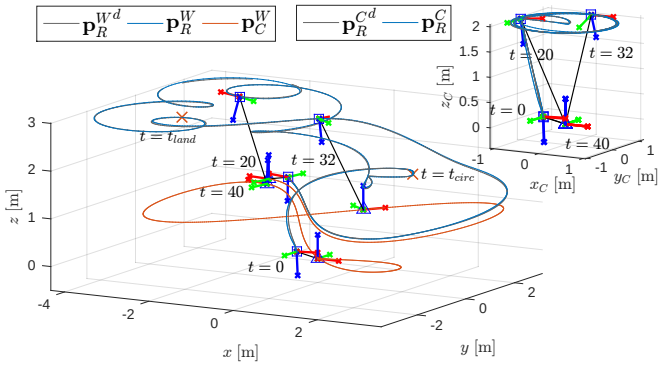
We remark that, if needed, one can increase the smoothness of the control inputs considering an higher order in the dynamic feedback control. Indeed adding more integrators on the control channels one can increase the degree of smoothness of the control input thus guarantying that it is always below the cutting frequency proper of the system, and in particular of the propellers. Another possible strategy is to exploit the flatness to plan a trajectory that fulfills the system limitations.



(a) Controller performances.



(b) Observer performances.



(c) Trajectories visualization.

Fig. 9: Simulation: system with motors dynamics.

I. Hierarchical control

In this section we compare our control method to a controller based on hierarchical techniques. Similar techniques were successfully implemented and tested in, e.g., [5], [6]. In fact, the methods in [5], [6] can not be directly applied to solve our problem because they are designed for different systems, although similar. Therefore in the following we design the best hierarchical controller we can conceive for controlling the position of the aerial vehicle and the tension on the link. The controller is based on the cascaded structure between the translational and rotational dynamics. We shall then compare its performances with respect our controller based on dynamic feedback linearization to show that a hierarchical approach performs much worse than our method in terms of both tracking precision and robustness to noise.

Given a desired position trajectory $\mathbf{p}_R^C(t)$, defined in terms of the generalized coordinates $\mathbf{q}(t)^d$ we define

$$\ddot{\mathbf{q}}^\star = \ddot{\mathbf{q}}^d + k_\mathbf{q}^D(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) + k_\mathbf{q}^P(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}),$$

where $k_\mathbf{q}^P, k_\mathbf{q}^D \in \mathbb{R}^+$. The vector $\ddot{\mathbf{q}}^\star$ could be seen as the desired acceleration that lets $\mathbf{q}$ follow the desired configuration $\mathbf{q}^d$ using a PD strategy.

Then, given a desired trajectory for the internal force of the link $f_L(t)^d$, and inverting the balance of momenta on the winch, we compute the winch torque as

$$\bar{\tau}_W = \bar{J}_W \ddot{l}^\star - f_L^d.$$

To finally implement $\ddot{\mathbf{q}}^\star$ we compute the desired thrust vector inverting the balance force equation on O_R,

$$\underbrace{f_R \mathbf{R}_C^T \mathbf{R}_R \mathbf{e}_3}_{\text{desired thrust vector}} = -\mathbf{a}_x - \mathbf{a}_g - f_L^d \mathbf{d}^C - m_R \mathbf{J}_\mathbf{q} \ddot{\mathbf{q}}^\star.$$

From the desired thrust vector we derive the input f_R as

$$f_R = \left\| -\mathbf{a}_x - \mathbf{a}_g - f_L^d \mathbf{d}^C - m_R \mathbf{J}_\mathbf{q} \ddot{\mathbf{q}}^\star \right\|,$$

and the desired z-axis of $\mathscr{F}_R$, i.e.,

$$\mathbf{z}_R^\star = \mathbf{R}_R \mathbf{e}_3 = \frac{\mathbf{R}_C(-\mathbf{a}_x - \mathbf{a}_g - f_L^d \mathbf{d}^C - m_R \mathbf{J}_\mathbf{q} \ddot{\mathbf{q}}^\star)}{f_R}.$$

The desired yaw angle ψ^d together with $\mathbf{z}_R^\star$ let us define the desired attitude of the vehicle described by $\mathbf{R}_R^\star$. In fact, given ψ^d we define $\mathbf{x}_R' = \mathbf{R}_\mathbf{z}(\psi^d)\mathbf{e}_1$ where $\mathbf{R}_\mathbf{z}(\psi^d)$ is the rotation matrix describing the rotation of ψ^d along $\mathbf{z}_W$. The axis $\mathbf{x}_R'$ represents the desired heading of the aerial vehicle. The desired attitude is computed creating an orthonormal basis using the vectors $\mathbf{x}_R'$ and $\mathbf{z}_R^\star$ that is given by $\mathbf{R}_R^\star = [\mathbf{x}_R^\star \ \mathbf{y}_R^\star \ \mathbf{z}_R^\star]$ where,

$$\mathbf{y}_R^\star = \frac{\mathbf{z}_R^\star \times \mathbf{x}_R'}{\|\mathbf{z}_R^\star \times \mathbf{x}_R'\|}, \qquad \mathbf{x}_R^\star = \frac{\mathbf{y}_R^\star \times \mathbf{z}_R\star}{\|\mathbf{y}_R^\star \times \mathbf{z}_R\star\|}.$$

This concludes the design of the outer loop control. Given the tracking error it computes the desired winch torque $\bar{\tau}_W$, the desired thrust intensity f_R and the desired attitude $\mathbf{R}_R^\star$.

Now we design the inner loop control to let the attitude follow the desired one. Let $\mathbf{e}_R \in \mathbb{R}^3$ (the attitude error) be computed as

$$[\mathbf{e}_R]_\times = -\frac{1}{2}(\mathbf{R}_R^{\star T}\mathbf{R}_R - \mathbf{R}_R^T \mathbf{R}_R^\star).$$

In order to steer $\mathbf{e}_R$ to zero we define the desired angular acceleration based on a PD controller,

$$\dot{\boldsymbol{\omega}}_R^\star = -k_\omega^D \boldsymbol{\omega}_R + k_\omega^P \mathbf{e}_R,$$

where $k_\omega^P, k_\omega^D \in \mathbb{R}^+$. Inverting the rotational dynamics we can finally find the input torque $\boldsymbol{\tau}_R$,

$$\boldsymbol{\tau}_R = -\mathbf{J}_R \boldsymbol{\omega}_R \times \boldsymbol{\omega}_R + \mathbf{J}_R \dot{\boldsymbol{\omega}}_R^\star.$$

If the inner loop is sufficiently faster than the outer loop, the asymptotic convergence of $\mathbf{q}$ to $\mathbf{q}^d$ is guaranteed.

In Fig. 10 the results of the hierarchical controller in ideal condition are reported. The desired trajectory and the initial tracking and estimation errors are the same of the ones in Sec. II-A. After a tuning phase we were able to get some good performances and a small bounded tracking error, even if the error does not converge exactly to zero. On the other hand, in the same conditions the controller based on dynamic feedback linearization is able to steer the output along the desired trajectory with zero error (see Fig. 1). However, to obtain good tracking performances with the hierarchical controller we had to set *very high* gains that make the system more reactive and thus able to follow the desired trajectory. Nevertheless this requirement has two main drawbacks.

The first drawback is that, due to the large control gains, the control effort increases thus possibly requiring an input that is out of the physical limits of the actuators. Indeed with this configuration we reach a maximum thrust and a maximum torque of about 15[N] and 2.5[Nm] respectively. This values are higher than the nominal inputs required to track the desired trajectory.

The second extremely serious issue arises in the presence of noise in the measurements and so in the estimated state. Indeed, the higher the gains, the larger the noise in the commands and the closer the controlled system is to instability. In fact, simulating the system with the same measurement noise described in Sec. II-D the closed loop system becomes unstable. In order to get a stable behavior we had to significantly lower the gains, an action that, however, clearly degrades the tracking performances. As we can see in Fig. 11 the performances with noise are much worse than the ones obtained using the dynamic feedback linearizing controller in the same noisy condition.

Therefore, the hierarchical approach presents a strictly penalizing tradeoff between applicability with noise and tracking performances. One cannot obtain both. Attainment of both objectives is instead possible with our proposed controller.

REFERENCES

[1] M. Tognon, S. S. Dash, and A. Franchi, "Observer-based control of position and tension for an aerial robot tethered to a moving platform," *IEEE Robotics and Automation Letters*, 2016.

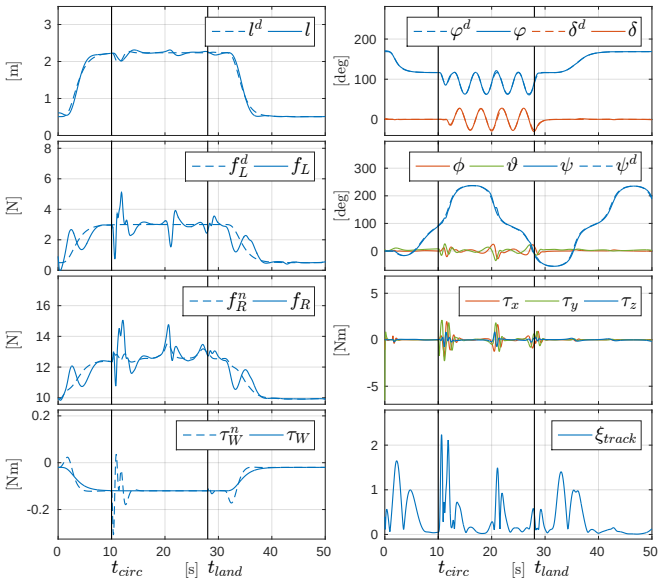
[2] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC, 1994.

[3] L. Sandino, D. Santamaria, M. Bejar, A. Viguria, K. Kondak, and A. Ollero, "Tether-guided landing of unmanned helicopters without GPS sensors," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May 2014, pp. 3096–3101.

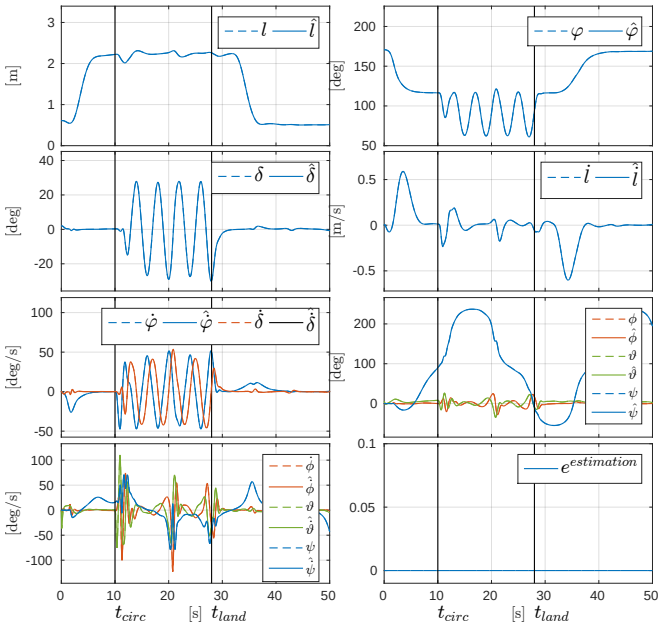
[4] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[5] L. Sandino, M. Bejar, K. Kondak, and A. Ollero, "A square-root unscented kalman filter for attitude and relative position estimation of a tethered unmanned helicopter," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, June 2015, pp. 567–576.

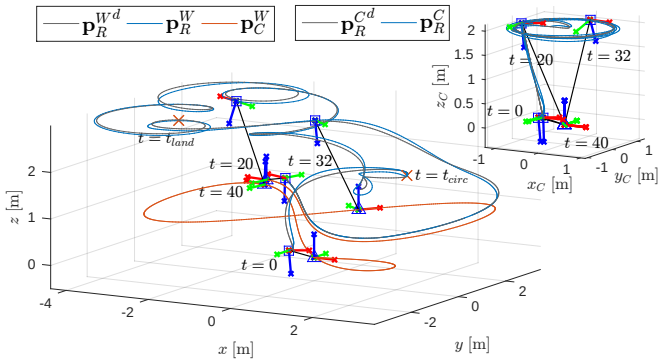
[6] S. Lupashin and R. D'Andrea, "Stabilization of a flying vehicle on a taut tether using inertial sensing," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, Nov 2013, pp. 2432–2438.



(a) Controller performances.

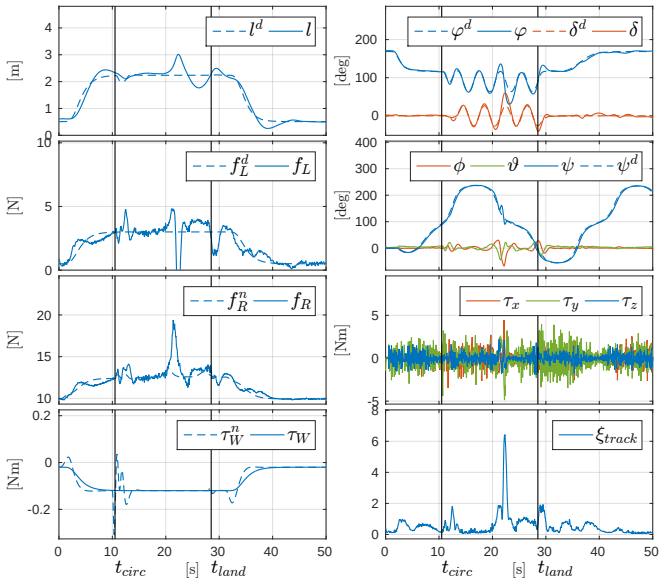


(b) Observer performances.

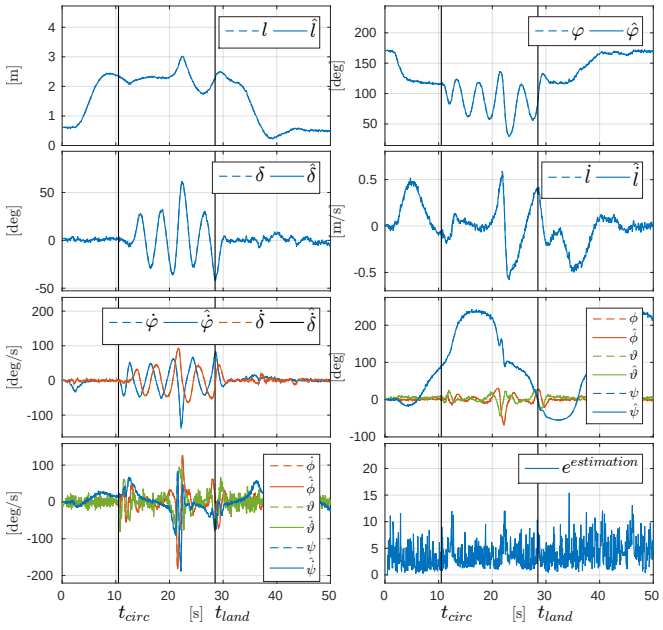


(c) Trajectories visualization.

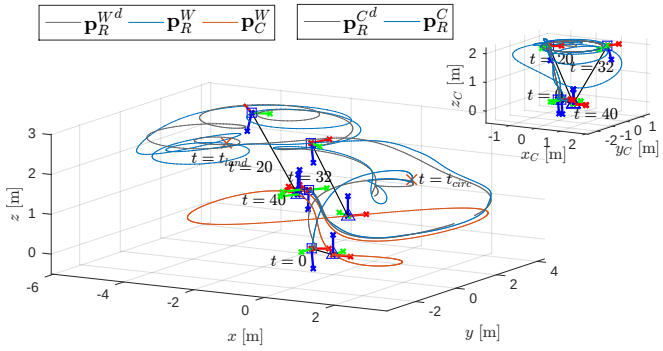
Fig. 10: Simulation: Hierarchical control. Good tracking performances are obtained only with very high gains in the inner and outer control loops



(a) Controller performances.



(b) Observer performances.



(c) Trajectories visualization.

Fig. 11: Simulation: Hierarchical control in the noisy case. To preserve stability lower gains have to be used with noise, therefore the performances are significantly degraded. The hierarchical controller presents a strictly penalizing tradeoff between tracking performances and robustness to noise.