Adding manipulation capabilities to ornithopters - GRIFFIN

José Ángel Acosta
Daniel Feliu

Internal Meeting
Adding manipulation capabilities to ornithopters

- We explore the feasibility of a new generation of aerial robots with the ability to perch and perform some task while the system is perched.

- The system shares similarities with a robot manipulator, but the joint at the bottom is passive with static friction, to emulate a claw.

- Putting both together underactuation and static friction makes it very difficult stabilize the system and perform some kind of task.
Adding manipulation capabilities to ornithopters

- We propose a bio-inspired controller in the sense that, it balances the body while the end-effector follows a trajectory. In addition, the controller shows robustness to disturbances.

- The proposed control strategy is based on a recent development in the stabilization of nonlinear systems with constraints.

- Verification with realistic simulations of a 5-link robot (claw, low/upper leg, body, neck, and beak) and experimental validation in a 3DOF prototype with a claw emulator.


A diffeomorphism approach for constrained mechanical systems

- Generalize the technique for Partially State-Constrained second order mechanical systems
- Improve the previous technique, assuring exponentially convergence of the state
- Allows us to use robust control techniques in order to increase the robustness of the system
- Motivation: use this control technique for under-actuated systems with hard nonlinear constraints and fully actuated manipulators

Dynamic of second order mechanical systems

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + P(q) + G(q) = \tau \]

A diffeomorphism approach for constrained mechanical systems

The technique has several steps:

Firstly: propose the construction of a diffeomorphism to map all the trajectories of the constrained dynamics into an unconstrained one.

\[
\begin{align*}
\Phi(x) &= \begin{pmatrix} \Phi_1(x_1) \\ x_2 \\ \Phi_2(x_2) \\ x_4 \end{pmatrix} \\
\Phi^{-1}(z) &= \begin{pmatrix} \Phi_1^{-1}(z_1) \\ z_2 \\ \Phi_2^{-1}(z_3) \\ z_4 \end{pmatrix} \\
\tilde{z}(t) &:= z(t) - z^*(t)
\end{align*}
\]

where \( \Phi_1(x_1), \Phi_2(x_2), ..., \Phi_n(x_n) \) is the diffeomorphism vector which transforms the constrained state \( |x_i| < a_i \) to an unconstrained one.

\[
\begin{align*}
\dot{x}_i &= x_{n+i}, \\
\dot{x}_{n+i} &= f_i(x) + g_i(x)u, \\
y_i &= x_i, & 1 \leq i \leq n; \\
|x_i(t)| &\leq a_i, & 1 \leq i \leq n;
\end{align*}
\]
A diffeomorphism approach for constrained mechanical systems

- Secondly: find a Lyapunov function in the transformed which demonstrates the exponential/asymptotic stability of the system

\[ \ddot{u}(\tilde{z}) = g(\tilde{z})^{-1} \left[ - \left( \frac{H(\tilde{z}_1 + z_1^*) \cdot \tilde{z}_1}{\mu \cdot \tilde{z}_2} \right) + \left( \begin{array}{c} 0 \\ \dot{\tilde{z}}_4^* \end{array} \right) - \left( \frac{k \cdot H(\tilde{z}_1 + z_1^*) \cdot \tilde{z}_3}{\gamma \cdot \tilde{z}_4} \right) - f(\tilde{z}) \right] \]

- Then, the closed-loop error dynamics becomes

\[ V = \tilde{z}_j^T P \tilde{z}_j \]

\[ H(z_1) = \frac{\partial_{x_s} \Phi_s(x_s)}{|x_\Phi^{-1}(z)} \]

- The system is globally exponential stable if

\[ \mu, \eta > 0 \]

\[ h_i(\tilde{z}_i + z_i^*) = h_i(z_i) \geq 1 \]
A diffeomorphism approach for constrained mechanical systems

- Transform the control law obtained in the unconstrained dynamics to the real one

\[
\bar{u}(\bar{z}) = g(\bar{z})^{-1} \left[ - \left( \frac{H(\bar{z}_1 + z^*_1) \cdot \bar{z}_1}{\mu \cdot \bar{z}_2} \right) + \left( \begin{array}{c} 0 \\ \dot{z}_4^* \end{array} \right) - \left( \frac{k \cdot H(\bar{z}_1 + z^*_1) \cdot \bar{z}_3}{\gamma \cdot \bar{z}_4} \right) - f(\bar{z}) \right]
\]

\[
z = \Phi(x) = \begin{pmatrix} \Phi_s(x_s) \\ x_r \\ \dot{x}_s \\ \dot{x}_r \end{pmatrix} ; \quad \Phi^{-1}(z) = \begin{pmatrix} \Phi_s^{-1}(z_1) \\ z_2 \\ z_3 \\ z_4 \end{pmatrix}
\]
Approach

We split the problem in two parts:

1) Design of the legs/claw (perching)
2) Design of arms/beak (perform manipulation)

• Manipulator for perching:

- Maintain the equilibrium while performing manipulation
- Help in the positioning in the perching maneuver to compensate inaccuracies in the flight
- Resist the impact in the perching maneuver, and allow us the perching maneuver

• Impact conditions in the perching: (requirements for the actual ornithopter)

- A total mass of the system around 1 Kg
- Speeds around 3 m/s
- Error around 30 cm
- Compliance/resistant to impacts
- Simple and very lightweight
- Precise and fast
Mechanism for perching

- The system weighs around 150 g (third prototype)
- The whole system compensates inaccuracies around 20-25 cm
- This can be improved by making the mechanism bigger but it is not recommendable
- A new prototype is under way (100-130g)
Mechanism for perching

Impact at 5 m/s

Set the experiments (speed, angle and height)

Force sensor
Mechanism for perching
Mechanism for perching
Mechanism for perching
Next stage: Scenario

- Throw in the launcher at a specified speed
- Glide some meters
- Perch in a branch
Flexible link with strain gauges (beak with sensing skills)

Ultralightweight manipulator

Allow us to interact with the environment

Possibility to estimate with accuracy the contact point along the manipulator

We measure the deflection in the base (implementation of force control systems)
Flexible link with force sensor (beak with sensing skills)

Assuming small deflections:

\[ EI \frac{d^4 y}{dx^4} = 0 \]

\[ y_i(x) = q_i,0 + q_i,1(x - l_i) + q_i,2(x - l_i)^2 + q_i,3(x - l_i)^3 \]

\[ q_{1,0} = q_{1,1} = 0 \]
\[ q_{1,2} = \frac{1}{2EI} (-ml^2\ddot{\theta}_i + \lambda l F_n) \]
\[ q_{1,3} = \frac{1}{6EI} (ml\dddot{\theta}_i - F_n) \]
\[ q_{2,0} = y_1(l_2) = \frac{ml^3\lambda^2}{6EI} \dddot{\theta}_i + \frac{2\lambda^3}{6EI} F_n \]
\[ q_{2,1} = \frac{dy_1(l_2)}{dt} = \frac{ml^3\lambda}{2EI} \dddot{\theta}_i + \frac{2\lambda^3}{2EI} F_n \]
\[ q_{2,2} = -\frac{ml^2(1-\lambda)}{2EI} \dddot{\theta}_i \]
\[ q_{2,3} = \frac{ml}{6EI} \dddot{\theta}_i \]
Estimate the contact point


Estimate the contact point

Propose an algorithm to detect the contact point based on the deflection of the link, we do not need a force sensor (more lightweight and much cheaper)


Thank you for your attention