Verification of the altitude controller of the nano quadcopter Crazyflie

Research project

IP Paris
Émilie THOMÉ

February 26, 2021

Outlines

1 Introduction
- Project objective
- Formal verification

2 Formal methods
- Lyapunov stability
- Reachability analysis

3 Model
- Modelling the environment
- Modelling the altitude controller
- Equation system
- Linearizing model equations around equilibrium

4 Lyapunov function
- Method
- Candidate

5 Simulation
- Simulation of the dynamics
- Simulations of the Lyapunov candidate

6 Verification of the candidate
- An SOS problem
- Reachability analysis
- Checking the candidate

7 Conclusion

8 References

9 Appendix

Introduction

Project objective

The main objective of this project is to **demonstrate the stability** of the Crazyflie UAV's altitude controller.

It will be necessary to:

- **define how to ensure stability** of a system,
- **model** this controller,
- **demonstrate altitude controller stability** of the UAV using formal methods.

Formal verification

Formal verification is proving the correctness of a system using formal methods.
Formal methods are techniques, **mathematically based**, to validate suppositions.

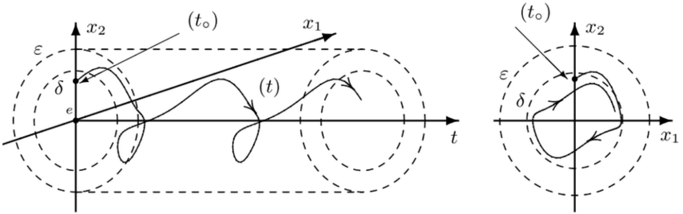
Method used to prove the correctness of the Crazyflie's altitude controller:

- **study over drone's stability** will give conditions to achieve a certain property,
- **reachability analysis** to select situations that meet the previous conditions.

Formal methods

Lyapunov stability

A system is **Lyapunov stable** for an equilibrium point x_e if all movement in the neighborhood of x_e remains in the neighborhood of this point.



Lyapunov stability

Lyapunov's stability theorem:

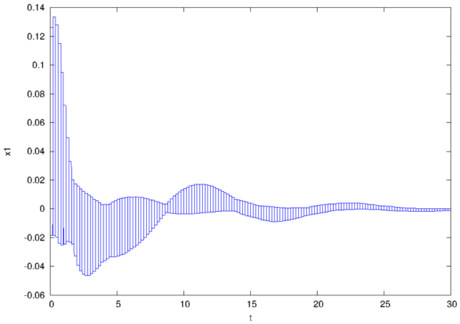
Let a system described by a differential equation of the type $\dot{x} = f(x, t)$.
WLOG: the equilibrium point of interest is $x_e = 0$*.*

- If there is a function $V : (x, t) \to V(x, t)$ continuously differentiable defined positive such that $\dot{V}$ is semi defined negative and that $V(x, t) \xrightarrow[||x||\to\infty]{} \infty$ then 0 is a **stable equilibrium point**.

NB: These conditions are similar to those that potential energy must verify for an equilibrium point of physical system to be stable.

Reachability analysis

A Lyapunov function is **specific to a dynamics and its operating boundaries**.
We will bound the state study space, using the reachability analyzer **Flow*** [3].



Flow* computes a finite number of Taylor models containing **all reachable states in a time Δ**, it over-approximates. If a given state is not included then it is **absolutely unreachable**.

Model

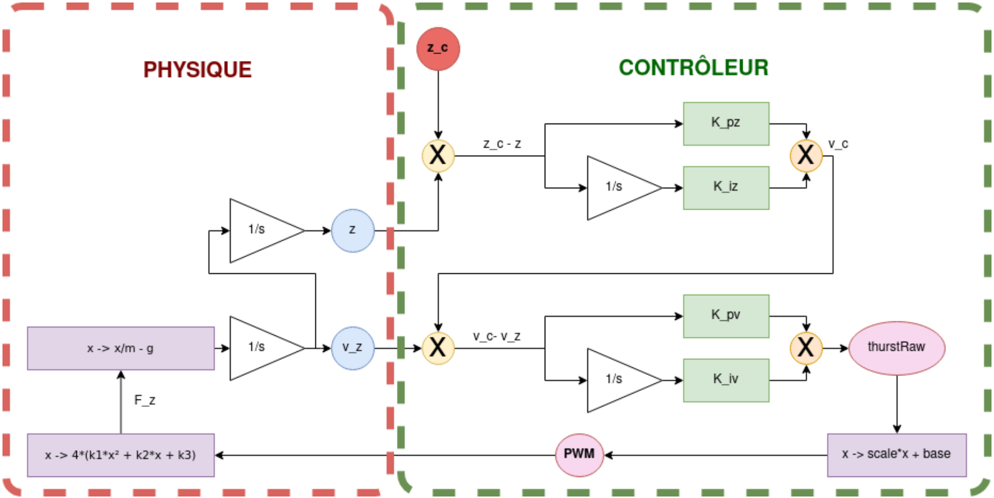
Modelling the environment

- **Fundamental principle of dynamics:** $\sum F = m\dot{v_z}$ donc $\dot{v_z} = \frac{F_z}{m} - g$
- **Force generated by each propeller:** $T_i = C_T\omega_i^2$
- **By symmetry:** $F_z = 4 * C_T\omega^2$.
- **Link between motor signal and angular speed:** $\omega = C_1 * PWM + C_2$
- **Then:** $F_z = (4C_T C_1^2) * PWM^2 + (8C_T C_1 C_2) * PWM + 4C_T C_2^2$

However, we will use the values found experimentally in [2] :

$$F_z = 4(k_1 * PWM^2 + k_2 * PWM + k_3)$$

Modelling the altitude controller



Modelling the altitude controller

The equations we can draw from the **C code of the Crazyflie firmware** are the following:

- $v_c = K_{pz}(z_c - z) + K_{iz} \int (z_c - z) dt$
- $thrustRaw = K_{pv}(v_c - v_z) + K_{iv} \int (v_c - v_z) dt$
- $PWM = \mathrm{Scale} * thrustRaw + \mathrm{Base}$

Equation system

Added states: $u_1 = \int(2(z_c - z) + 0.5\int(z_c - z)dt - v_z)dt$ et $u_2 = \int(z_c - z)dt$
New constants: $K_p = \mathrm{Scale} * K_{pv}$ et $K_i = \mathrm{Scale} * K_{iv}$
Centred altitude: $z \leftarrow z - z_c$

$$\begin{cases} \dot{z} = v_z \\ \dot{v_z} = \frac{4k_1}{m} * PWM^2 + \frac{4k_2}{m} * PWM + \frac{4k_3}{m} - g \\ \dot{u_1} = -2z + 0.5u_2 - v_z \\ \dot{u_2} = -z \end{cases}$$

$$PWM = K_p \dot{u_1} + K_i u_1 + \mathrm{Base}$$

Equilibrium of the system

The state equilibrium is reached when every derivative is zero.

$$\begin{cases} 0 = v_{ze} \\ 0 = 4(k_1 * PWM_e{}^2 + k_2 * PWM_e + k_3) - mg \\ 0 = -2z_e + 0.5u_2 - v_{ze} \\ 0 = -z_e \\ PWM_e = K_i u_{1e} + \mathrm{Base} \end{cases}$$

Then the equilibrium is:

$$\begin{cases} PWM_e \approx 37287 \\ z_e = 0 \\ v_{ze} = 0 \\ u_{1e} = \frac{PWM_e - \mathrm{Base}}{K_i} \approx 0.0858 \\ u_{2e} = 0 \end{cases}$$

Linearizing model equations around equilibrium

Taylor's first-order expansion:

$$\begin{cases} \Delta \dot{z} = \Delta v_z \\ \Delta \dot{v_z} = \frac{8k_1}{m} * PWM_e * \Delta PWM + \frac{4k_2}{m} * \Delta PWM \\ \Delta \dot{u_1} = -2\Delta z + 0.5\Delta u_2 - \Delta v_z \\ \Delta \dot{u_2} = -\Delta z \\ \Delta PWM = K_p \Delta \dot{u_1} + K_i \Delta u_1 \end{cases}$$

New equation system

Noting $\begin{cases} \Delta x = [\Delta z \ \Delta v_z \ \Delta u_1 \ \Delta u_2]^t \\ \Delta \dot{x} = [\Delta \dot{z} \ \Delta \dot{v_z} \ \Delta \dot{u_1} \ \Delta \dot{u_2}]^t \end{cases}$ then:

$$\begin{cases} \Delta \dot{x} = A * \Delta x \\ \Delta PWM = B * \Delta x \end{cases}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -2K_p * \alpha & -K_p * \alpha & K_i * \alpha & \frac{K_p}{2} * \alpha \\ -2 & -1 & 0 & 0.5 \\ -1 & 0 & 0 & 0 \end{bmatrix} \text{ with } \alpha = \frac{8(2k_1 * PWM_e + k_2)}{m}$$

$$B = \begin{bmatrix} -2 * K_p & -K_p & K_i & 0.5 * K_p \end{bmatrix}$$

Lyapunov function

Method

Equilibrium linearization is used to **construct a Lyapunov function**. Suppose P and Q are symmetric and defined positive such that :

$$A^t P + PA = -Q$$

Then the function $V : x \rightarrow x^t Px$ is defined positive and $-\dot{V} : x \rightarrow -\nabla VAx = x^t Qx$ too.
We then have to solve this so-called **Lyapunov equation**.

Candidate

$$Q = I \text{ imply that } P = \begin{bmatrix} 2.01656529 & 0.08342414 & 0.11486412 & -1.29171207 \\ 0.08342414 & 0.07182875 & -0.08901516 & -0.04613902 \\ 0.11486412 & -0.08901516 & 1.35166103 & -0.56806025 \\ -1.29171207 & -0.04613902 & -0.56806025 & 2.44792802 \end{bmatrix}.$$

The eigen values of P are 3.65640661, 1.40539696, 0.76532165 and 0.06085787. So the matrix is definitely defined positive.

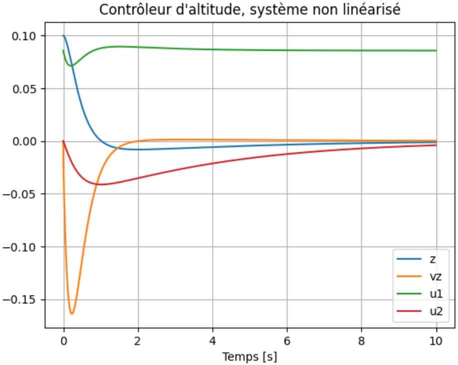
We therefore have a potential candidate for the Lyapunov function of the initial system.

Simulation

Simulation of the dynamics

Simulation of the initial system dynamics:

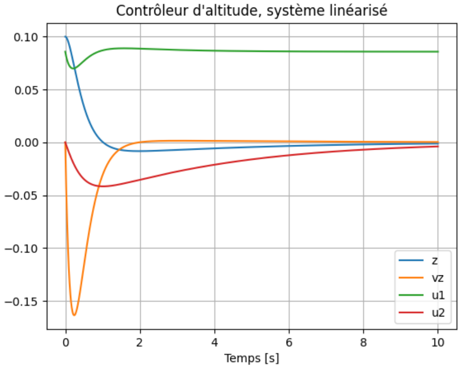
The altitude controller seems stable:



Simulation of the dynamics

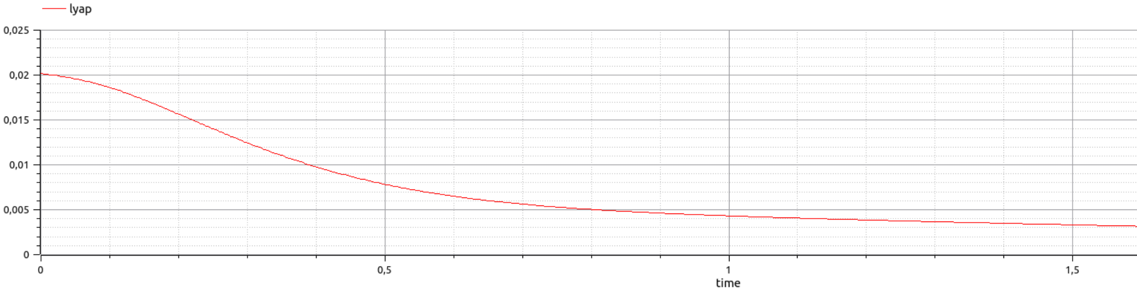
Simulation of the linearized system dynamics:

The linearized system is very similar to the initial one.
We can therefore hope that the **Lyapunov function found is a Lyapunov function** for the initial system.



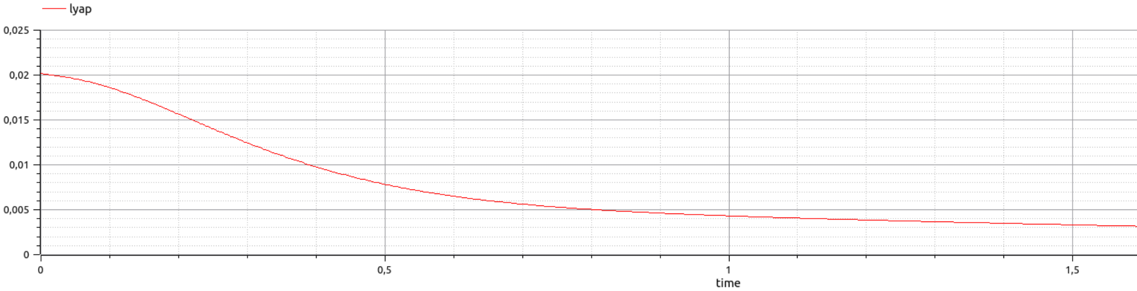
Simulations of the Lyapunov candidate

We can see that the computed Lyapunov function can **check the conditions** to describe the stability of the linearized system.



Simulations of the Lyapunov candidate

Hopping this function is also well suited for the initial system describing the altitude controller, the figure effectively suggests that **the function is a candidate**.



Verification of the candidate

An SOS problem

The candidate V found is polynomial of order 2, defined positive.

Only condition to satisfy: $-\dot{V}$ positive for your dynamics.
If we can prove that $-\dot{V}$ is a sum of squares (SOS) of polynomials, then V is a Lyapunov function. V might not be a Lyapunov function for the entire space.
Bounding constraints are needed.
If exist β SOS such that $-\dot{V}(x) = \beta_{1inf}(x) * (z - c_{1inf}) + [...] + \beta_{4sup}(x) * (c_{4sup} - u_2)$ then $-\dot{V}(x) > 0$ for all x in bounds.

SOSTOOLS [1] makes it possible to formulate and solve this problem.

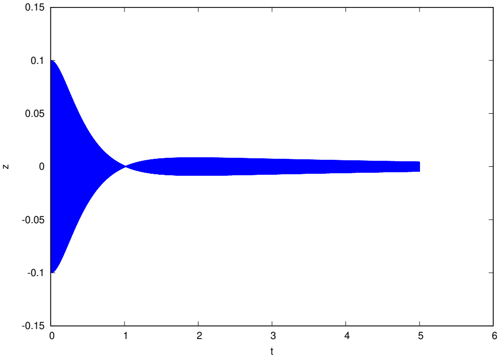
We now need to ensure that the chosen bounding **contains all reachable states**.

Reachability analysis

Boundings are found using **Flow***.
`unsafe` conditions are declared to be informed when states are out of bounds.

```
init
{ z in [-0.1, 0.1]        vz in [0,0]
   g in [9.80, 9.81]      u2 in [0,0]
  u1 in [0.0857,0.0858]    t in [0,0]}

unsafe
{z <= -0.15    z >= 0.15
  vz <= -0.3   vz >= 0.3
  u1 <= -0.08  u1 >= 0.1
  u2 <= -0.05  u2 >= 0.05}
```



I can now use those bounds to check the Lyapunov candidate with **SOSTOOLS**.

Checking the candidate

Bounds inputted, **SOSTOOLS** found the SOS polynomials β. The output of the solver is:

```
Residual norm: 5.4687e-10

         iter: 15
    feasratio: 1.0000
         pinf: 0
         dinf: 0
       numerr: 0
           r0: 1.1837e-11
       timing: [0.1272 0.1862 0.0099]
      wallsec: 0.3233
       cpusec: 0.2700
```

Precision: `Residual norm` smaller than incertitude on the matrix P (robust over perturbations).

We can now conclude that the candidate is a Lyapunov function.

The system is stable for an initial state near the equilibrium.

Conclusion

Conclusion

During this project I have:

- characterized a stability of a system
- modeled the Crazyflie's altitude controller
- linearized this system to simplify it
- found a Lyapunov candidate for the initial system
- and proved the Lyapunov candidate to be a Lyapunov function.

The next step in the formal verification of the Crazyflie's altitude controller system would be to **include this proof in the SPARK code of the Crazyflie**.

References

References

[1] G. Valmorbida S. Prajna P. Seiler A. Papachristodoulou J. Anderson and P. A. Parrilo. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*. Available from http://www.eng.ox.ac.uk/control/sostools, http://www.cds.caltech.edu/sostools and http://www.mit.edu/~parrilo/sostools. http://arxiv.org/abs/1310.4716, 2013.

[2] J. Förster. “System Identification of the Crazyflie 2.0 Nano Quadrocopter”. In: (2015).

[3] Erika Ábrahám Xin Chen Sriram Sankaranarayanan. *Flow**. Available from https://flowstar.org/. 2017.

Thank you.

Appendix

All recorded constants for the Crazyflie model

m	0.028
k_1	$2.130295e-11$
k_2	$1.032633e-6$
k_3	$5.484560e-4$
Scale	1000
Base	36000
K_{pz}	2
K_{iz}	0.5
K_{pv}	25
K_{iv}	15
K_p	25000
K_i	15000

The constants presented are in SI.

We can notice that k_3 and k_2 seem negligible in front of k_1 which is itself not very big. However the values of PWM are important, which makes k_i not dispensable.

Reminder: $PWM_e \simeq 37287$