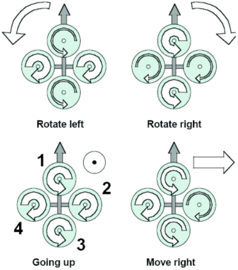
Making Flips with Quadrotors in Constrained Environments

Elie Hatem

Advisors: Dr. Sebastien Briot, Dr. Isabelle Fantoni

August 24, 2021

(a) DJI Phantom quadcopter (UAV)[1]



(b) Quadrotor Concept. Width of the arrows is proportional to the angular speed of the propellers[2]

Figure: Commercial quadrtotor platform (left) and quadrotor concept (right).

[1] `https://en.wikipedia.org/wiki/Quadcopter#/media/File:Quadcopter_camera_drone_in_flight.jpg`

[2] Design and control of quadrotors with application to autonomous flying, 2007, S. Bouabdallah

Over the last few years, quadrotors have gained large popularity in academia and industry. Because, they are:

- Simple to design and assemble using relatively cheap components.
- Different use cases: aerial photography, agriculture, surveillance tasks, etc.
- Quite agile and maneuverable during flight, especially when compared to other types of UAVs.

One of the main challenges

Designing control and planning methods to allow tracking aggressive trajectories.

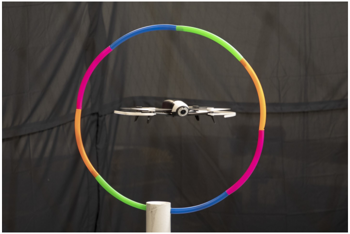
This difficulty is due to:

- The fast dynamics associated with the small dimensions of such agile quadrotors.
- Several dynamic effects will become important to consider during aggressive flight maneuvers.
- The motors will be commanded high speeds and accelerations, which will cause them to saturate and introduce delays.

The goals of the master thesis:

- Study of multi-flip maneuvers.
- Implement Model Predictive Control to solve the presented issues.
- Perform the maneuvers in a constrained environment.

(a) Quadrotor performing a triple flip[3]



(b) Quadrotor going though a loop[4]

Figure: Representation of the issues to be tackled in this master thesis.

[3] Adaptive fast open-loop maneuvers for quadrocopters, 2012, S. Lupashin and R. D'Andrea

[4] `https://newatlas.com/drones/muscle-signals-drone-control/`

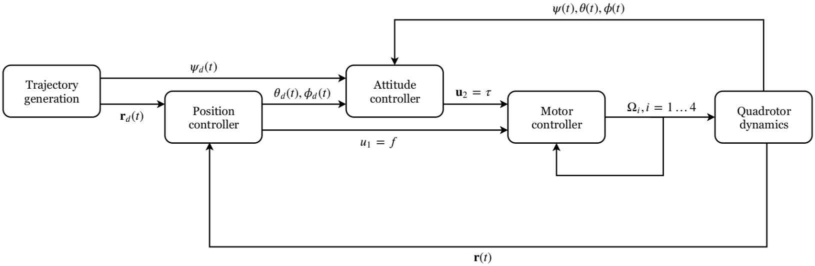


Figure: Diagram of the general control architecture of a quadrotor.

- Position controller: slow rise time - drives translational dynamics errors to 0.
- Attitude controller: faster rise time - drives rotational dynamics errors to 0.
- Motor controller: fastest rise time - maps the control inputs to motor speeds.

Remark

- The designed controller cannot be faster than the one at a lower level.
  - The orientation cannot be controlled any faster than the motors can be controlled.

A well-established finding is that the dynamic model of a quadrotor is differentially flat:

- The system with state $\boldsymbol{x} \in \mathbb{R}^n$ and input $\boldsymbol{u} \in \mathbb{R}^m$ has flat outputs $\boldsymbol{y} \in \mathbb{R}^m$ which have the following form:

$$\mathbf{y} = \mathbf{y}(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, ..., \mathbf{u}^{(p)}) \tag{1}$$

With,

$$\begin{cases} \mathbf{x} = \mathbf{x}(\mathbf{y}, \dot{\mathbf{y}}, ..., \mathbf{y}^{(q)}) \\ \mathbf{u} = \mathbf{u}(\mathbf{y}, \dot{\mathbf{y}}, ..., \mathbf{y}^{(r)}) \end{cases} \tag{2}$$

- Very useful property in under-actuated systems where $m < n$.
- Allows to generate trajectories in the lower dimensional space m.
- The trajectories can then be mapped into the full dimensional space n.

The standard choice of flat outputs for the quadrotor are:

$$\mathbf{y} = \begin{bmatrix} x & y & z & \psi \end{bmatrix}^\intercal \tag{3}$$

As a result:

- Trajectories can be designed in the 4-dimensional space.
- They can then be mapped to the 6-dimenstional space.
  - This is due to the fact that the rotational and translational dynamics are tightly coupled.

General idea of MPC:

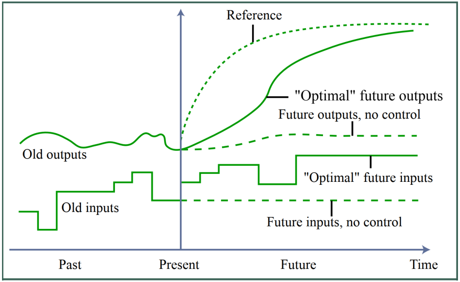


Figure: Basic idea of MPC[5]

- It is a **feedback control** algorithm.
- It uses a model to **predict** future outputs.
- It **solves an online optimization problem** to select the optimal control.

[5] Principles of Optimal Control, 2008, J. How

MPC Design parameters:

- Sample time.
- Prediction horizon.
- Control horizon.
- Constraints.
- Weights.

Choosing proper values for these parameters is important as they affect:

- The controller performance.
- The computational complexity of the MPC algorithm.

MPC Design parameters:

- Sample time.
- Prediction horizon.
- Control horizon.
- Constraints.
- Weights.

Choosing proper values for these parameters is important as they affect:

- The controller performance.
- The computational complexity of the MPC algorithm.

The most popular uses for MPC in quadrotors are:

- Centralized MPC: Single control loop for the system.
- Non-centralized MPC: Cascaded control consisting of more than 1 control loop. Examples:
  - MPC_{master}-MPC_{slave}
  - MPC-PD-P
  - Other options can be used for the inner loop.

Remark

- Centralized MPC: More accurate, high computation cost.
- Non-centralized MPC: Less accurate, lower computation cost.

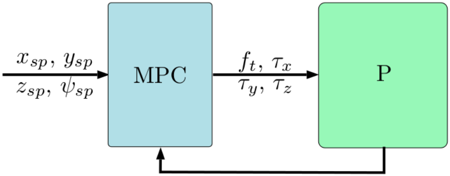


Figure: Centralized MPC

Inputs of centralized MPC:

- Desired x, y and z positions and the yaw angle ψ.

Outputs of centralized MPC:

- Total thrust f_t.
- Torques: τ_x, τ_y and τ_z

Another version of the centralized MPC exists with:

- Added ϕ and θ angles as outputs.

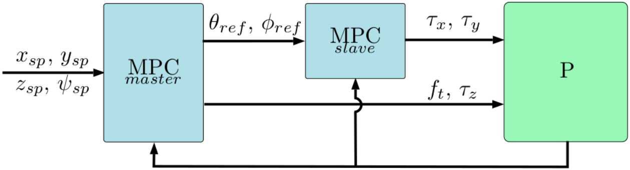


Figure: Non-centralized MPC (MPC$_{master}$-MPC$_{slave}$)

Outer-loop: master MPC

- Inputs: Desired x, y, z, ψ.
- Outputs: f_t, τ_z, θ_{ref}, ϕ_{ref}.

Inner-loop: slave MPC

- Inputs: θ_{ref}, ϕ_{ref}.
- Outputs: τ_x, τ_y.

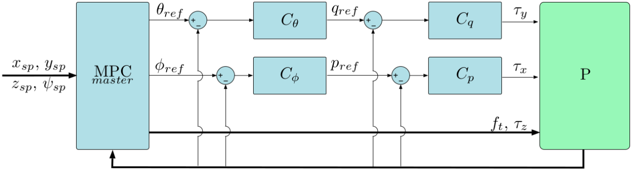


Figure: Non-centralized MPC (MPC-PD-P)

Outer-loop: master MPC

- Inputs: Desired x, y, z, ψ.
- Outputs: f_t, τ_z, θ_{ref}, ϕ_{ref}

Inner-loop: PD-P controller

- Inputs: θ_{ref}, ϕ_{ref}.
- Outputs: τ_x, τ_y.

Another example of a non-centralized MPC:

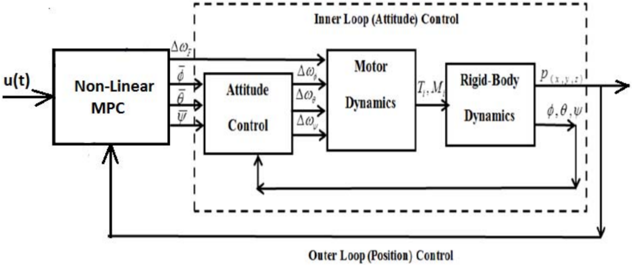


Figure: Non-centralized MPC

Remark

The inner-loop can remain fixed, while the outer loop can be reprogrammed to meet the required task.

The software used for implementing MPC controller is `acados`:

- Contains efficient optimal control algorithms implemented in C.
- Has a modular architecture enabling rapid prototyping of solution algorithms.
- Interfaces to `C++`, `Python` and `MATLAB`.
- Uses the high-performance linear algebra package `BLASFEO`.
- Compatible with `CasADi` expressions.
- Deployable on a variety of embedded devices.
- Free and open-source software.

Main drawback:

- Prediction horizon and control horizon must be of same length.
  - This issue can be solved using the real-time iteration (RTI) method.

The general form of the nonlinear program that can be handled by `acados` is:

$$
\begin{aligned}
\min_{\substack{x_0,\dots,x_N \\ u_0,\dots,u_{N-1} \\ z_0,\dots,z_{N-1} \\ s_0,\dots,s_N}} \quad & \sum_{k=0}^{N-1} l_k(x_k, u_k, z_k) + M(x_N) + \sum_{k=0}^{N} \rho_k(s_k) \\
\text{s.t.} \quad & \begin{bmatrix} x_{k+1} \\ z_k \end{bmatrix} = \phi_k(x_k, u_k) & k = 0, 1, \dots, N-1, \\
& 0 \geq g_k(x_k, z_k, u_k) - J_{s,k} s_k & k = 0, 1, \dots, N-1, \\
& 0 \geq g_N(x_N) - J_{s,N} s_N, \\
& 0 \leq s_k & k = 0, 1, \dots, N-1
\end{aligned} \tag{4}
$$

And,

$$
\rho_k(s_k) = \sum_{i=1}^{n_{s_k}} \alpha_k^i s_k^i + \beta_k^i s_k^{i^2} \tag{5}
$$

with $\alpha_k^i \in \mathbb{R}, \beta_k^i > 0$.

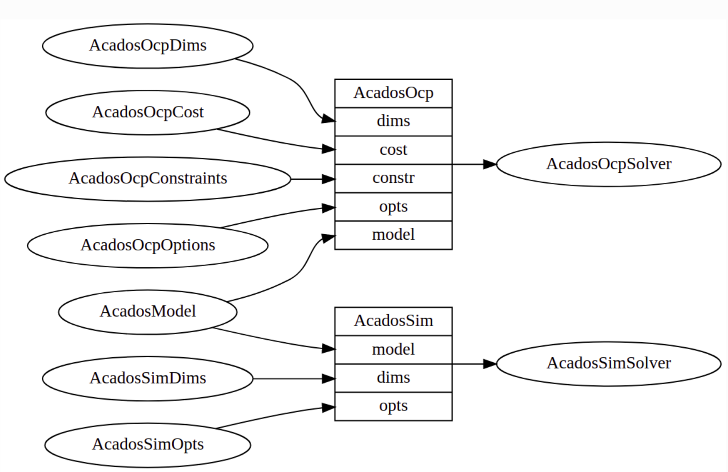


Figure: Overview of the Python API classes in acados[6]

[6] https://docs.acados.org/python_api/

Assumptions to considered when using the equations of motion in the planar and 3D case:

- Rigid structure.
- Symmetric structure.
- CoG and origin of body-fixed frame coincide.
- Propellers are rigid.
- Thrust and drag are proportional to the square of the velocity of the propeller.

The equations of motion of the planar quadrotor are:

$$\begin{cases} \ddot{y} = -\frac{u_1}{m} \sin \phi \\ \ddot{z} = -g + \frac{u_1}{m} \cos \phi \\ \ddot{\phi} = \frac{u_2}{I_{xx}} \end{cases} \tag{6}$$

With:

- u_1: Total thrust applied on the planar quadrotor.
- u_2: Torque applied by the planar quadrotor along the x-axis.

The equations of motion of a quadrotor:

$$\text{Translational model: } \begin{cases} \dot{\boldsymbol{p}}_i &= \boldsymbol{v}_i \\ \dot{\boldsymbol{v}}_i &= \frac{T}{m} \begin{bmatrix} 2(q_w q_y + q_x q_z) \\ 2(q_y q_z - q_w q_x) \\ 1 - 2(q_x^2 + q_y^2) \end{bmatrix} + \boldsymbol{g} \end{cases}$$

$$\text{Rotational model: } \begin{cases} \dot{\boldsymbol{q}}_i &= \frac{1}{2} \begin{bmatrix} 0 \\ \boldsymbol{\omega}_i \end{bmatrix} \otimes \boldsymbol{q}_i \\ \dot{\boldsymbol{\omega}}_i &= \boldsymbol{I}_i^{-1} \boldsymbol{\tau}_i - \boldsymbol{I}_i^{-1} (\boldsymbol{\omega}_i \times \boldsymbol{I}_i \boldsymbol{\omega}_i) \end{cases}$$

However, the MPC controller will not account for all of the equations (Non-centralized MPC):

$$\text{Treated by the MPC:} \begin{cases} \dot{\boldsymbol{p}}_i &= \boldsymbol{v}_i \\ \dot{\boldsymbol{v}}_i &= \frac{T}{m} \begin{bmatrix} 2(q_w q_y + q_x q_z) \\ 2(q_y q_z - q_w q_x) \\ 1 - 2(q_x^2 + q_y^2) \end{bmatrix} + \boldsymbol{g} \\ \dot{\boldsymbol{q}}_i &= \frac{1}{2} \begin{bmatrix} 0 \\ \boldsymbol{\omega}_i \end{bmatrix} \otimes \boldsymbol{q}_i \end{cases}$$

$$\text{Treated by the L.L. controller:} \ \left\{ \ \dot{\boldsymbol{\omega}}_i = \boldsymbol{I}_i^{-1} \boldsymbol{\tau}_i - \boldsymbol{I}_i^{-1} (\boldsymbol{\omega}_i \times \boldsymbol{I}_i \boldsymbol{\omega}_i) \right.$$

Remark

- A low level controller is assumed to exist to map the angular rates inputs $\boldsymbol{\omega}$ to the required torques $\boldsymbol{\tau}$.
- The equations above do not represent the complete model of the system.
  - However, they are sufficient to control the quadrotor.

Figure: Crazyflie 2.1 with added mass

- $m_{no\_load} = 29.5g$
- $m_{with\_load} = 47g$
- $T_{max}^{actual} = (47 \times 10^{-3}kg)9.81\frac{m}{s^2} = 0.46N$
- $I_{xx} = 1.657171 \times 10^{-5}kg.m^2$
- $I_{yy} = 1.657171 \times 10^{-5}kg.m^2$
- $I_{zz} = 2.9261652 \times 10^{-5}kg.m^2$
- $L = 0.046m$

For the planar quadrotor case:

$$\begin{aligned} \min_{x,u} \quad & \frac{1}{2}\|\boldsymbol{V}_x\boldsymbol{x} + \boldsymbol{V}_u\boldsymbol{u} + \boldsymbol{V}_z\boldsymbol{z} - \boldsymbol{y}_{ref}\|^2_{\boldsymbol{W}} + \frac{1}{2}\|\boldsymbol{V}^e_x\boldsymbol{x} - \boldsymbol{y}^e_{ref}\|^2_{\boldsymbol{W}_e} \\ \text{s.t.} \quad & f(\boldsymbol{x}, \boldsymbol{u}) : \text{dynamics} \\ & 0 \leq u_1 \leq u_{1_{max}} \\ -u_{2_{max}} \leq & \; u_2 \leq u_{2_{max}} \end{aligned} \tag{7}$$

with:

- $\boldsymbol{V}_x \in \mathbb{R}^{n_y \times n_x}$
- $\boldsymbol{V}_u \in \mathbb{R}^{n_y \times n_u}$
- $\boldsymbol{V}_z \in \mathbb{R}^{n_y \times n_z}$
- $\boldsymbol{V}^e_x \in \mathbb{R}^{n_{ye} \times n_x}$
- $\boldsymbol{y}_{ref} \in \mathbb{R}^{n_y}$
- $\boldsymbol{y}_{ref_e} \in \mathbb{R}^{n_{ye}}$
- $\boldsymbol{W} \in \mathbb{R}^{n_y \times n_y}$
- $\boldsymbol{W}_e \in \mathbb{R}^{n_{ye} \times n_{ye}}$
- $\boldsymbol{Q}, \boldsymbol{Q}_e \in \mathbb{R}^{n_x \times n_x}$
- $\boldsymbol{R} \in \mathbb{R}^{n_u \times n_u}$
- $\boldsymbol{W} = diag(\boldsymbol{Q}, \boldsymbol{R})$
- $\boldsymbol{W}_e = \boldsymbol{Q}_e$

The state space variables:

$$\boldsymbol{x} = [y, z, \phi, v_y, v_z, \dot{\phi}]^\intercal \tag{8}$$

The state evolves according to:

$$\dot{y} = v_y \tag{9}$$

$$\dot{z} = v_z \tag{10}$$

$$\dot{\phi} = \dot{\phi} \tag{11}$$

$$\dot{v}_y = -\frac{u_1}{m}\sin(\phi) \tag{12}$$

$$\dot{v}_z = -g + \frac{u_1}{m}\cos(\phi) \tag{13}$$

$$\ddot{\phi} = \frac{u_2}{I_{xx}} \tag{14}$$

Control inputs: $\boldsymbol{u} = [u_1, u_2]^\intercal$

Initial condition:

$\boldsymbol{x}_0 = [y_0, z_0, \phi_0, v_{y_0}, v_{z_0}, \ddot{\phi}_0]^\intercal$

Desired states:

$\boldsymbol{x}_d = [y_d, z_d, \phi_d, v_{y_d}, v_{z_d}, \ddot{\phi}_d]^\intercal$

MPC parameters:

- $N = 100$
- $T_f = 1s$

Maximum thrust and torque:

- $u_{1_{max}} = 0.9\left(\frac{0.46N}{2}\right)$
- $u_{2_{max}} = 0.1\left(\frac{1}{2}u_{1_{max}}L\right)$

For the 3D quadrotor case:

$$\begin{aligned} \min_{x,u} \quad & \frac{1}{2}\|\boldsymbol{V}_x\boldsymbol{x} + \boldsymbol{V}_u\boldsymbol{u} + \boldsymbol{V}_z\boldsymbol{z} - \boldsymbol{y}_{ref}\|^2_{\boldsymbol{W}} + \frac{1}{2}\|\boldsymbol{V}^e_x\boldsymbol{x} - \boldsymbol{y}^e_{ref}\|^2_{\boldsymbol{W}_e} \\ \text{s.t.} \quad & f(\boldsymbol{x},\boldsymbol{u}) : \text{dynamics} \\ & 0 \leq T \leq T_{max} \\ & -4\pi \leq \omega_x, \omega_y, \omega_z \leq 4\pi \end{aligned} \tag{15}$$

with:

- $\boldsymbol{V}_x \in \mathbb{R}^{n_y \times n_x}$
- $\boldsymbol{V}_u \in \mathbb{R}^{n_y \times n_u}$
- $\boldsymbol{V}_z \in \mathbb{R}^{n_y \times n_z}$
- $\boldsymbol{V}^e_x \in \mathbb{R}^{n_{ye} \times n_x}$
- $\boldsymbol{y}_{ref} \in \mathbb{R}^{n_y}$
- $\boldsymbol{y}_{ref_e} \in \mathbb{R}^{n_{ye}}$
- $\boldsymbol{W} \in \mathbb{R}^{n_y \times n_y}$
- $\boldsymbol{W}_e \in \mathbb{R}^{n_{ye} \times n_{ye}}$
- $\boldsymbol{Q}, \boldsymbol{Q}_e \in \mathbb{R}^{n_x \times n_x}$
- $\boldsymbol{R} \in \mathbb{R}^{n_u \times n_u}$
- $\boldsymbol{W} = diag(\boldsymbol{Q}, \boldsymbol{R})$
- $\boldsymbol{W}_e = \boldsymbol{Q}_e$

The state space variables:

$$\boldsymbol{x} = [x, y, z, q_w, q_x, q_y, q_z, v_x, v_y, v_z,]^\intercal \tag{16}$$

Control inputs: $\boldsymbol{u} = [T, \omega_x, \omega_y, \omega_z]^\intercal$

Initial condition:

$\boldsymbol{x}_0 = [x_0, y_0, z_0, q_{w_0}, q_{x_0}, q_{y_0}, q_{z_0}, v_{x_0}, v_{y_0}, v_{z_0}]^\intercal$

Desired states:

$\boldsymbol{x}_d = [x_d, y_d, z_d, q_{w_d}, q_{x_d}, q_{y_d}, q_{z_d}, v_{x_d}, v_{y_d}, v_{z_d}]^\intercal$

MPC parameters:

- $N = 100$
- $T_s = 1s$

Maximum thrust and torque:

- $T_{max} = 0.9(0.46N)$

The states evolve according to:

$$\dot{x} = v_x$$
$$\dot{y} = v_y$$
$$\dot{z} = v_z$$
$$\dot{q}_w = \frac{1}{2}(-w_x q_x - w_y q_y - w_z q_z)$$
$$\dot{q}_x = \frac{1}{2}(w_x q_w + w_z q_y - w_y q_z)$$
$$\dot{q}_y = \frac{1}{2}(w_y q_w - w_z q_x + w_x q_z)$$
$$\dot{q}_z = \frac{1}{2}(w_z q_w + w_y q_x - w_x q_y)$$
$$\dot{v}_x = 2(q_w q_y + q_x q_z)\frac{T}{m}$$
$$\dot{v}_y = 2(q_y q_z - q_w q_x)\frac{T}{m}$$
$$\dot{v}_z = (1 - 2q_x^2 - 2q_y^2)\frac{T}{m} - g$$

The physics of a quadrotor flip can be divided into 4 parts:

- **Climb phase**: Maximum vertical acceleration is applied.
- **Multi-flip phase**: It ends when the desired $2n\pi$ are achieved.
- **Descent phase**: Maximum thrust for descent compensation.
- **Re-stabilization**: Altitude regulation to desired value.

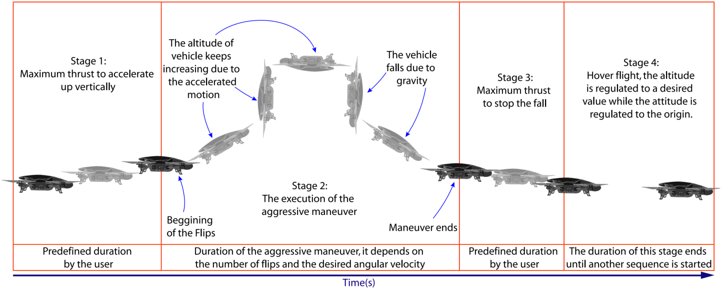


Figure: The phases needed for performing multi-flip maneuvers [7]

[7] Nonlinear ellipsoid based attitude control for aggressive trajectories in a quadrotor: closed-loop multiflips implementation, 2018, F. Oliva-Palomo et al.

The dynamic criterion of a planar quadrotor is[8]:

$$\ddot{y}\cos\phi + (\ddot{z} + g)\sin\phi = 0 \tag{17}$$

- It represents the dynamics constraints that must be satisfied to track a trajectory.

To design dynamically feasible flip trajectories:

- The trajectories along z and ϕ will be generated using polynomials of order 9.
- The trajectory along y will then be computed by integrating the dynamic criterion twice:

$$y(t) = -\iint (\ddot{z} + g)\tan\phi \, dtdt \tag{18}$$

[8] Making aggressive maneuvers with drones thanks to parallel singularity crossing approaches, 2019, M. Orsingher

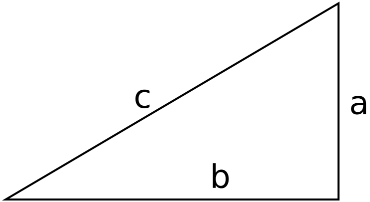
- Some
- Appearing
- Bullets
    - With sub-bullets

- Some
- Appearing
- Bullets
    - With sub-bullets

- Some
- Appearing
- Bullets
  - With sub-bullets

- Some
- Appearing
- Bullets
  - With sub-bullets

- Some
- Appearing
- Bullets
  - With sub-bullets



And an appearing figure.

- Some
- Appearing
- And disappearing
- Bullets
    - With sub-bullets
- That appear and disappear with their parent

- Some
- Appearing
- And disappearing
- Bullets
  - With sub-bullets
- That appear and disappear with their parent

- Some
- Appearing
- And disappearing
- Bullets
    - With sub-bullets
- That appear and disappear with their parent

- Some
- Appearing
- And disappearing
- Bullets
  - With sub-bullets
- That appear and disappear with their parent

- Some
- Appearing
- And disappearing
- Bullets
    - With sub-bullets
- That appear and disappear with their parent

1. Some
2. Numbers
    - With sub-bullets
3. That appear and the same time
4. Nicely spaced on the slide

- Should be done with \fullcite
  - O. S. Pythagoras (Feb. -580). “Theorem”. In: *Some old journal*
- You may also use \smallcite
  - O. S. Pythagoras (Feb. -580). “Theorem”. In: *Some old journal*
  - It takes less space...
- Check all imported references for:
  - Name / Journal name / year / editor (journals)
  - Carefully check conference name (IEEEexplore)

- Should be done with \fullcite
  - O. S. Pythagoras (Feb. -580). “Theorem”. In: *Some old journal*
- You may also use \smallcite
  - O. S. Pythagoras (Feb. -580). “Theorem”. In: *Some old journal*
  - It takes less space...
- Check all imported references for:
  - Name / Journal name / year / editor (journals)
  - Carefully check conference name (IEEExplore)

- Should be done with \fullcite
  - O. S. Pythagoras (Feb. -580). “Theorem”. In: *Some old journal*
- You may also use \smallcite
  - O. S. Pythagoras (Feb. -580). “Theorem”. In: *Some old journal*
  - It takes less space...
- Check all imported references for:
  - Name / Journal name / year / editor (journals)
  - Carefully check conference name (IEEEexplore)

- Have a number if used with `\begin{equation}`

$$\forall \phi : \quad \cos^2 \phi + \sin^2 \phi = 1 \tag{19}$$

- Do not have a number if used with `\begin{equation*}`

$$\forall a, b : \quad (a + b)^2 = a^2 + 2ab + b^2$$

- Another useful environment is simply `\begin{center}`

$$\forall a, b : \quad (a - b)^2 = a^2 - 2ab + b^2$$

  - Probably more suited to slides as we use less equation references

- Can also be included in the text / bullets
  - $\forall \phi : \quad (\cos \phi + \sin \phi)^2 = 2 \cos \phi \sin \phi + 1$

- Have a number if used with `\begin{equation}`

$$\forall \phi : \quad \cos^2 \phi + \sin^2 \phi = 1 \tag{19}$$

- Do not have a number if used with `\begin{equation*}`

$$\forall a, b : \quad (a+b)^2 = a^2 + 2ab + b^2$$

- Another useful environment is simply `\begin{center}`

  $\forall a, b : \quad (a-b)^2 = a^2 - 2ab + b^2$

  - Probably more suited to slides as we use less equation references

- Can also be included in the text / bullets
  - $\forall \phi : \quad (\cos \phi + \sin \phi)^2 = 2 \cos \phi \sin \phi + 1$

- Have a number if used with `\begin{equation}`

$$\forall \phi : \quad \cos^2 \phi + \sin^2 \phi = 1 \tag{19}$$

- Do not have a number if used with `\begin{equation*}`

$$\forall a, b : \quad (a + b)^2 = a^2 + 2ab + b^2$$

- Another useful environment is simply `\begin{center}`

$$\forall a, b : \quad (a - b)^2 = a^2 - 2ab + b^2$$

  - Probably more suited to slides as we use less equation references

- Can also be included in the text / bullets
  - $\forall \phi : \quad (\cos \phi + \sin \phi)^2 = 2 \cos \phi \sin \phi + 1$

- Have a number if used with `\begin{equation}`

$$\forall \phi : \quad \cos^2 \phi + \sin^2 \phi = 1 \tag{19}$$

- Do not have a number if used with `\begin{equation*}`

$$\forall a, b : \quad (a + b)^2 = a^2 + 2ab + b^2$$

- Another useful environment is simply `\begin{center}`

$$\forall a, b : \quad (a - b)^2 = a^2 - 2ab + b^2$$

  - Probably more suited to slides as we use less equation references

- Can also be included in the text / bullets
  - $\forall \phi : \quad (\cos \phi + \sin \phi)^2 = 2 \cos \phi \sin \phi + 1$

- Have a number if used with `\begin{equation}`

$$\forall\phi: \quad \cos^2\phi + \sin^2\phi = 1 \tag{19}$$

- Do not have a number if used with `\begin{equation*}`

$$\forall a, b: \quad (a+b)^2 = a^2 + 2ab + b^2$$

- Another useful environment is simply `\begin{center}`

$$\forall a, b: \quad (a-b)^2 = a^2 - 2ab + b^2$$

  - Probably more suited to slides as we use less equation references

- Can also be included in the text / bullets
  - $\forall\phi: \quad (\cos\phi + \sin\phi)^2 = 2\cos\phi\sin\phi + 1$

- Have a number if used with `\begin{equation}`

$$\forall \phi : \quad \cos^2 \phi + \sin^2 \phi = 1 \tag{19}$$

- Do not have a number if used with `\begin{equation*}`

$$\forall a, b : \quad (a+b)^2 = a^2 + 2ab + b^2$$

- Another useful environment is simply `\begin{center}`

$$\forall a, b : \quad (a-b)^2 = a^2 - 2ab + b^2$$

  - Probably more suited to slides as we use less equation references

- Can also be included in the text / bullets
  - $\forall \phi : \quad (\cos \phi + \sin \phi)^2 = 2 \cos \phi \sin \phi + 1$