

# PSI 3442 - Projeto de Sistemas Embarcados

---



Fernando Zolubas Preto



# Objetivos das aulas práticas

---

Exemplificar a definição e projeto de sistemas ciberfísicos complexos.

Objeto de estudo: UAV (Unmanned aerial vehicle) / Drone

Por que?

- Complexo
- Aplicações diversas



# Objetivos das aulas práticas

---

## Eixos temáticos

A - Entender um sistema embarcado complexo

B - Simulação de sistemas embarcados

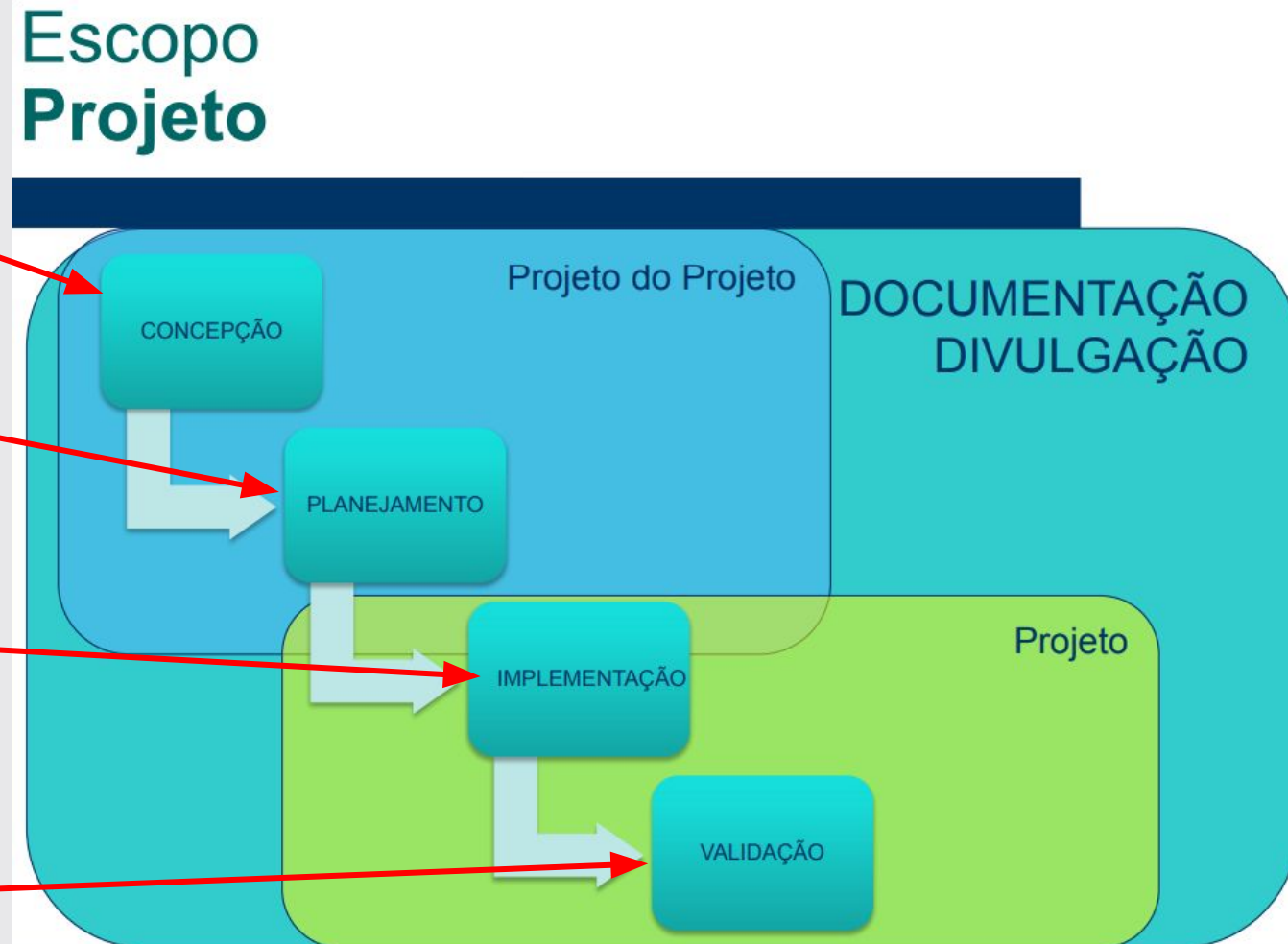
C - Traduzir requisitos de desempenho em algoritmos de controle para sistemas

D - Integrar Software + Hardware



# Objetivos das aulas práticas

- Problema de Engenharia
  - Solução e requisitos
- Dimensionamento do drone
- Simulações
  - PX4+Gazebo+ROS+Controle
- Drone Físico



# Na aula de hoje

---

- Sistema Embarcado
- Sistema Ciber-físico
- Exemplo: Drone
  - Complexidade
  - Estados (Graus de Liberdade)
  - Movimentação
  - Estimação de estados
  - Hardware
  - Software/Controle
  - Aplicações



# Aula 1: Sistema Embarcado

---

- É um Sistema Computacional
- Processador, memória, entrada/saída de periféricos
- Função específica em sistema Físico (ex: Mecatrônico)
- Tipicamente usado em controle de sistemas físicos
  - Computação de tempo real
  - Restrições de consumo energético
  - Computação intensiva
  - Tamanho reduzido



Apollo Guidance Computer

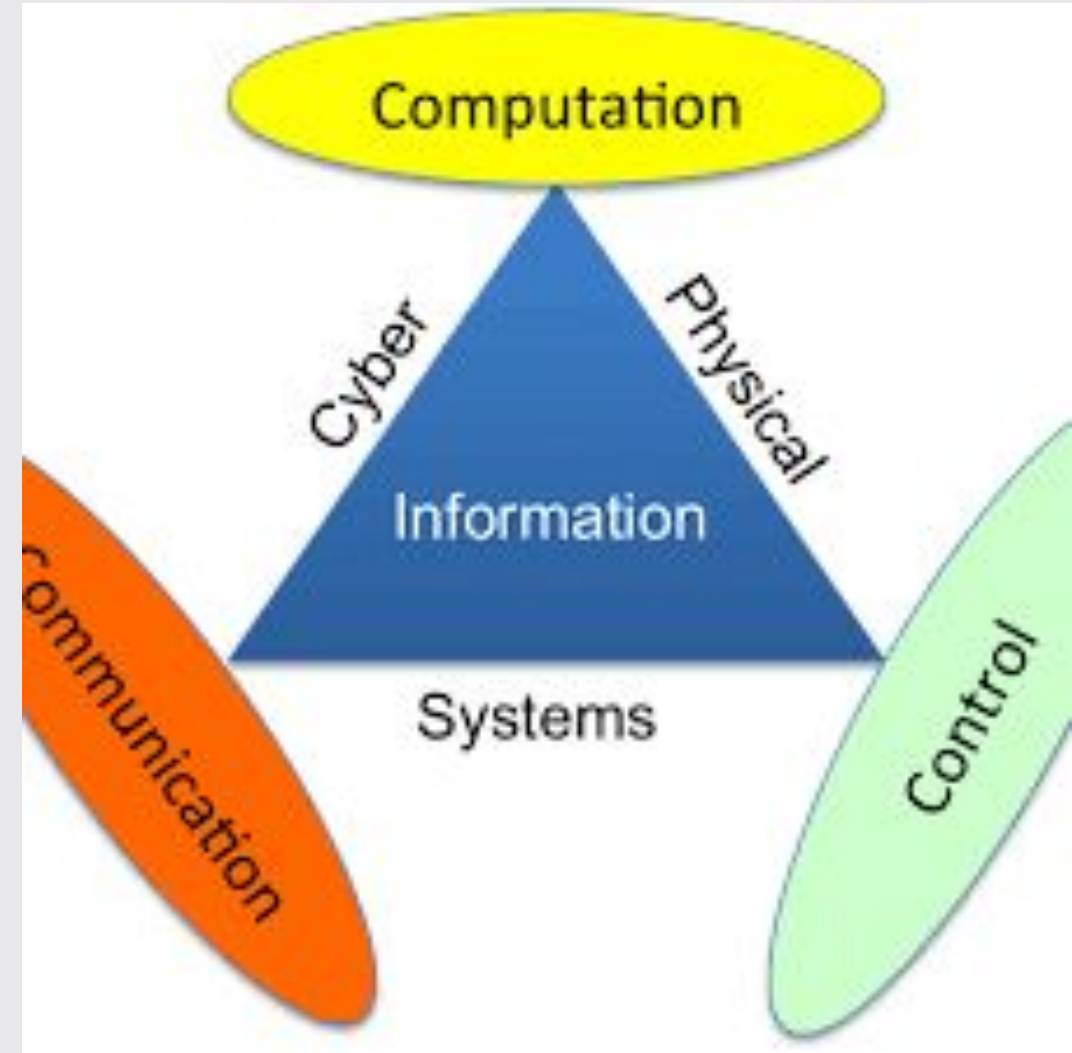


Um dos primeiros exemplos:



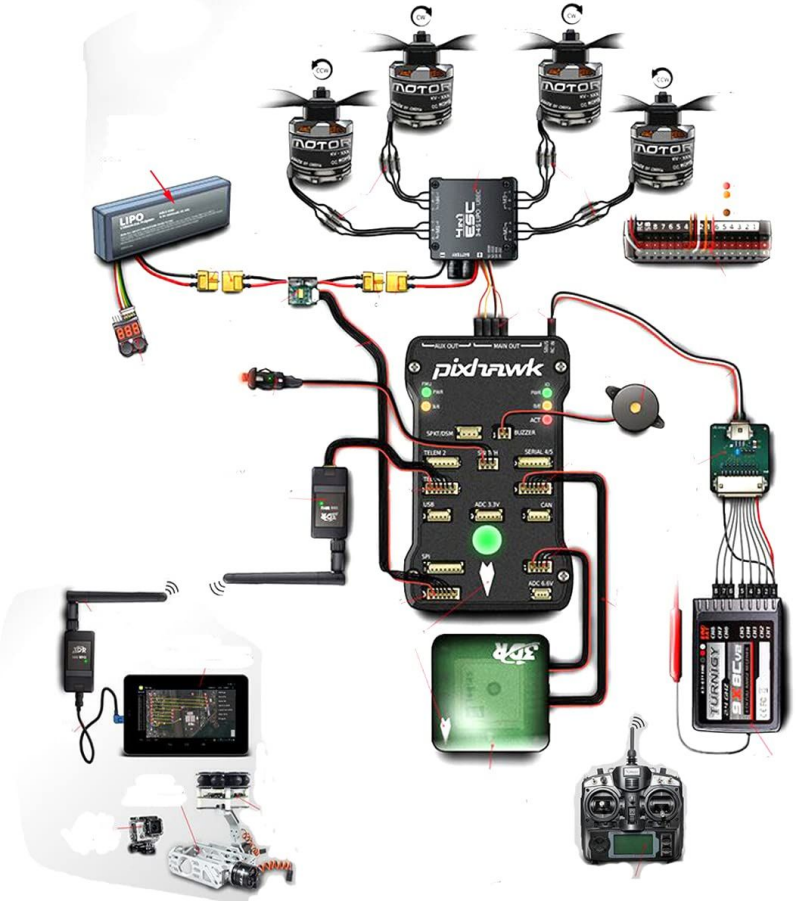
# Aula 1: Sistema Ciber-físico

Sistema Ciber-físico: “Cyber-physical system (CPS)” é uma integração de computação com processos físicos cujo comportamento somente pode ser definido pelo conjunto integrado das partes cibernéticas e físicas do sistema.



# Aula 1: Exemplo de Sistema Ciber-físico

Hardware



Drone

Software

```
import rospy
from mavbase.MAV import MAV
import numpy as np

def go():

    rospy.init_node("mav_test")
    mav = MAV("1")

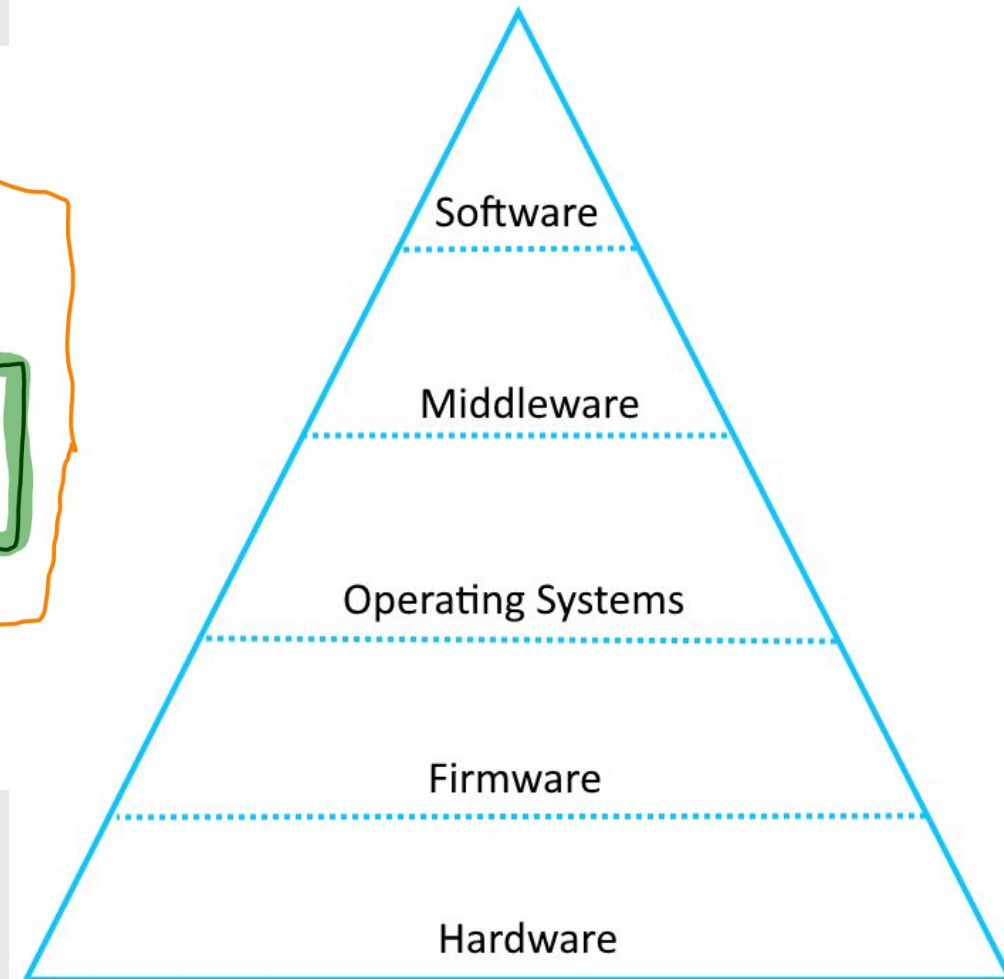
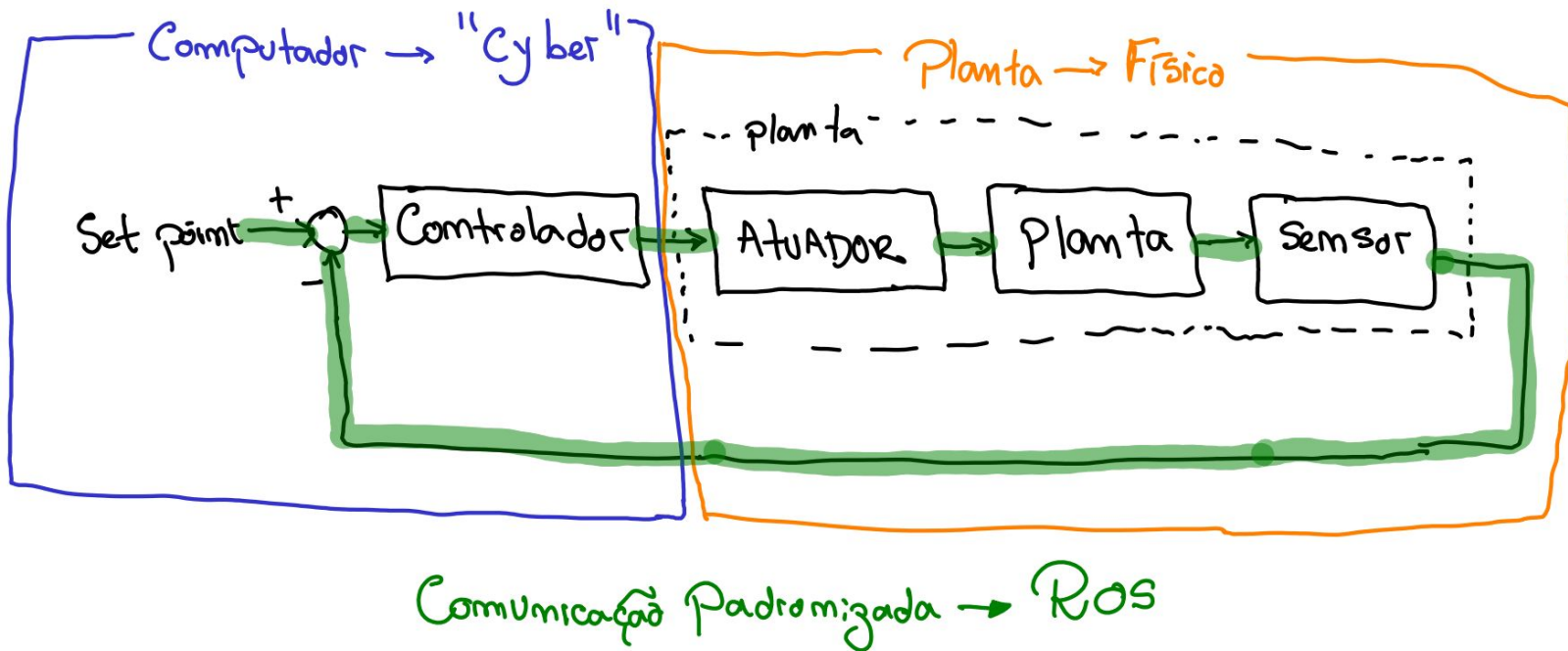
    takeoff_alt = 10
    goal_x = 5
    goal_y = 5
    altitude = 2

    mav.takeoff(takeoff_alt)
```



# Aula 1: Sistema Ciber-físico

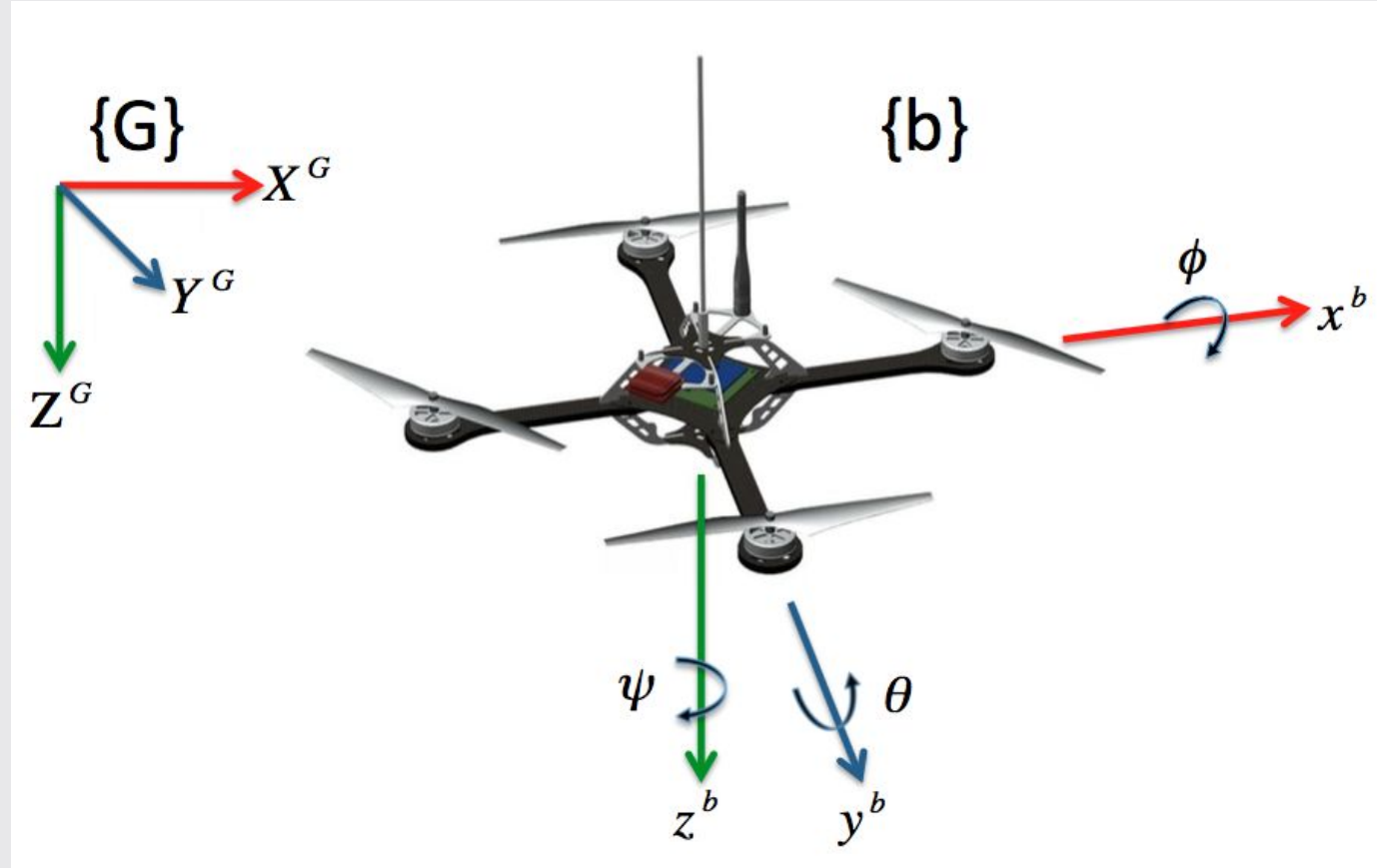
## Esquemáticos de sistemas ciber-físicos



# Aula 1: Exemplo de sistema embarcado complexo

## Drone Quadrotor

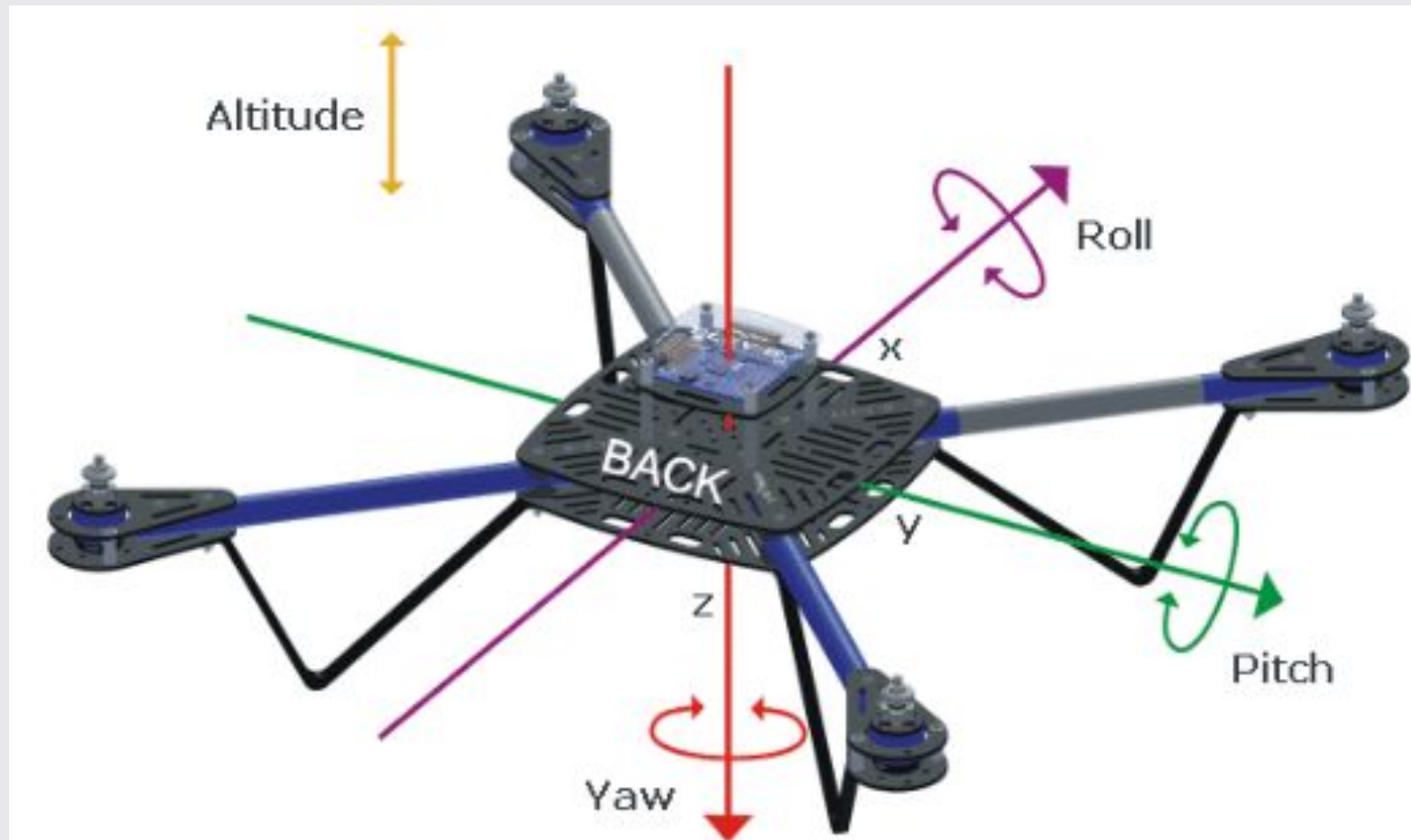
- Multivariável
  - Estimação do estado
    - Múltiplos sensores
- Equilíbrio instável
  - Controle constante
- Múltiplos graus de liberdade
  - 6DOF (Degrees of freedom)
  - Múltiplos atuadores



# Aula 1: Exemplo de sistema embarcado complexo

## Drone Quadrotor: Movimentação

- Rotação
  - Roll
  - Pitch
  - Yaw
- Translação
  - X
  - Y
  - Z



# Aula 1: Exemplo de sistema embarcado complexo

## Drone Quadrotor: Estimação de estados (DOF)

### Sensores

- Acelerômetro\* [x,y,z]
- Giroscópio\* [roll,pitch,yaw]
- Magnetômetro\* [yaw]
- Barômetro, LIDAR [z]
- GPS [x,y,z]
- Câmera (SLAM) [x,y,z]

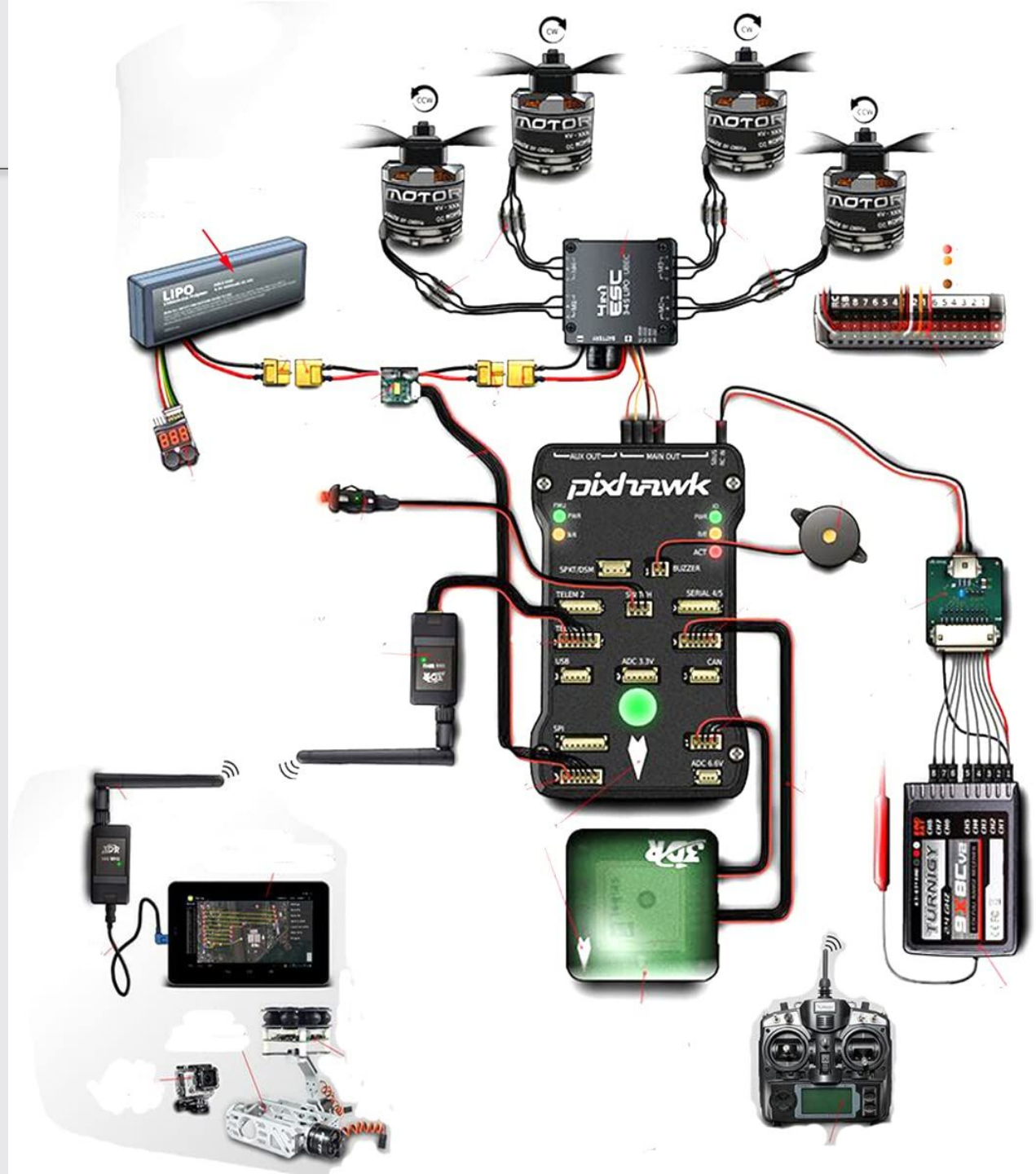
Observador de Estados + Filtro (EKF\*\*)

Vetor de estados  $x = [x,y,z, \text{roll,pitch,yaw}]$

\*IMU: Inertial measurement unit, \*\* Extended Kalman Filter

# Aula 1: Hardware

- Sensores
  - IMU
  - GPS
  - Barômetro
  - Câmera
- Controlador
- Atuadores
- Comunicadores
  - Telemetria
  - Rádio
- Bateria



# Aula 1: Middleware/Firmware/Software

- Middleware
  - ROS (Robotic Operating System)
- Sistema Operacional
  - NuttX (RTOS pixhawk)
  - Linux
- Firmware
  - PX4
  - Ardupilot
- Software Simulação
  - Gazebo
- Linguagens de programação
  - Python/C++



# Aula 1: Por que desenvolver sistemas ciber-físicos?

---



# Aula 1: Software

---

- Implementação computacional de um algoritmo
- Algoritmo: Sequências de instruções que quando seguidas geram um produto/resultado desejado

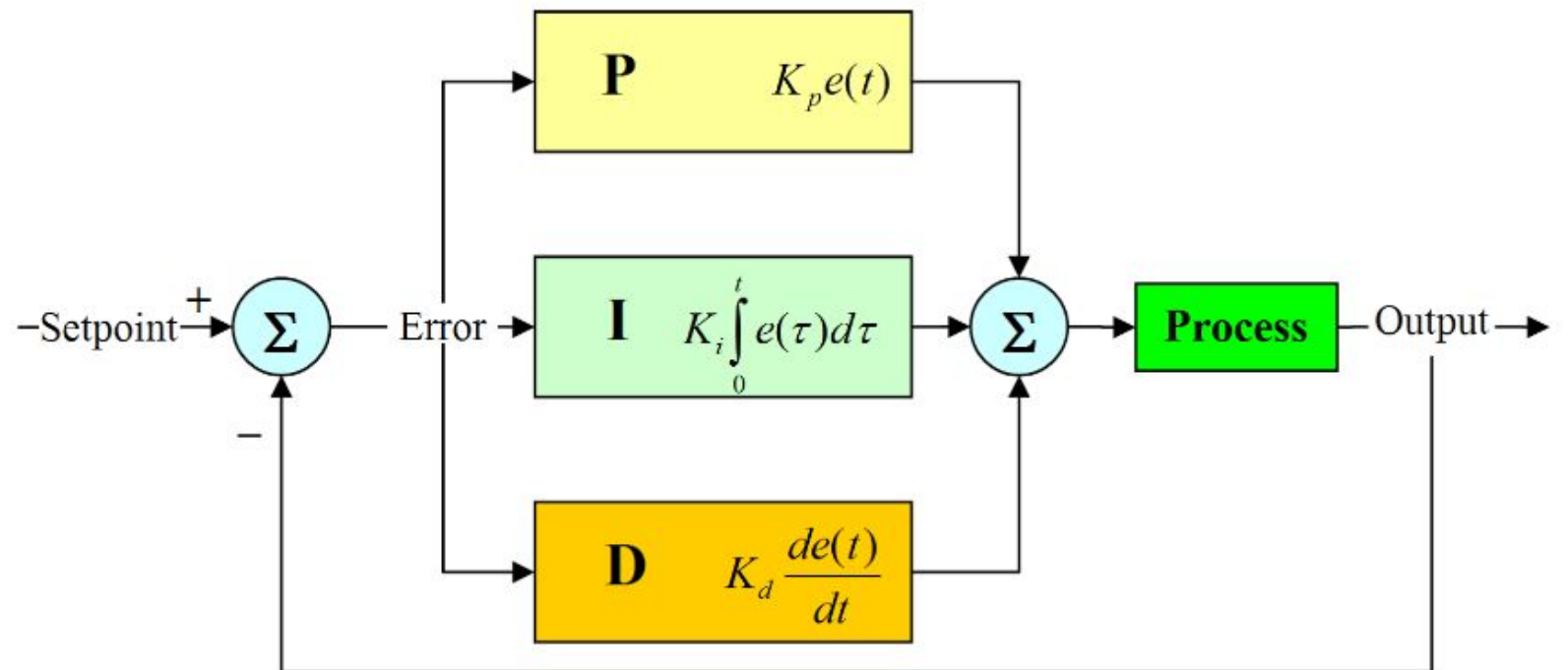
```
def go():  
  
    rospy.init_node("mav_test")  
    mav = MAV("1")  
  
    takeoff_alt = 10  
    goal_x = 5  
    goal_y = 5  
    altitude = 2  
  
    mav.takeoff(takeoff_alt)  
    mav.rate.sleep()  
    actual_x = mav.drone_pose.pose.position.x  
    actual_y = mav.drone_pose.pose.position.y  
    actual_dist = np.sqrt((goal_x - actual_x)**2 + (goal_y - actual_y)**2)  
    rospy.loginfo("Setting position to (%s, %s)" %(str(goal_x), str(goal_y)))
```





# Aula 1: Controle

- Uma lei de controle é um algoritmo cuja finalidade é alterar a dinâmica natural de um sistema de modo que sua nova dinâmica atenda a *requisitos de controle* desejado e de modo que o sistema resultante seja *estável e robusto à perturbações e incertezas de projeto*.



# Recapitulando

---

- Sistema Embarcado
- O que é um Sistema Ciber-físico?
- Exemplo: Drone
  - Por que um drone é complexo?
  - Estados
  - Como funciona a movimentação?
  - Como se faz a estimação de estados?
  - Quais os componentes de Hardware de um drone?
  - Quais elementos de software vamos usar?
  - Quem sabe outra aplicação de drones?

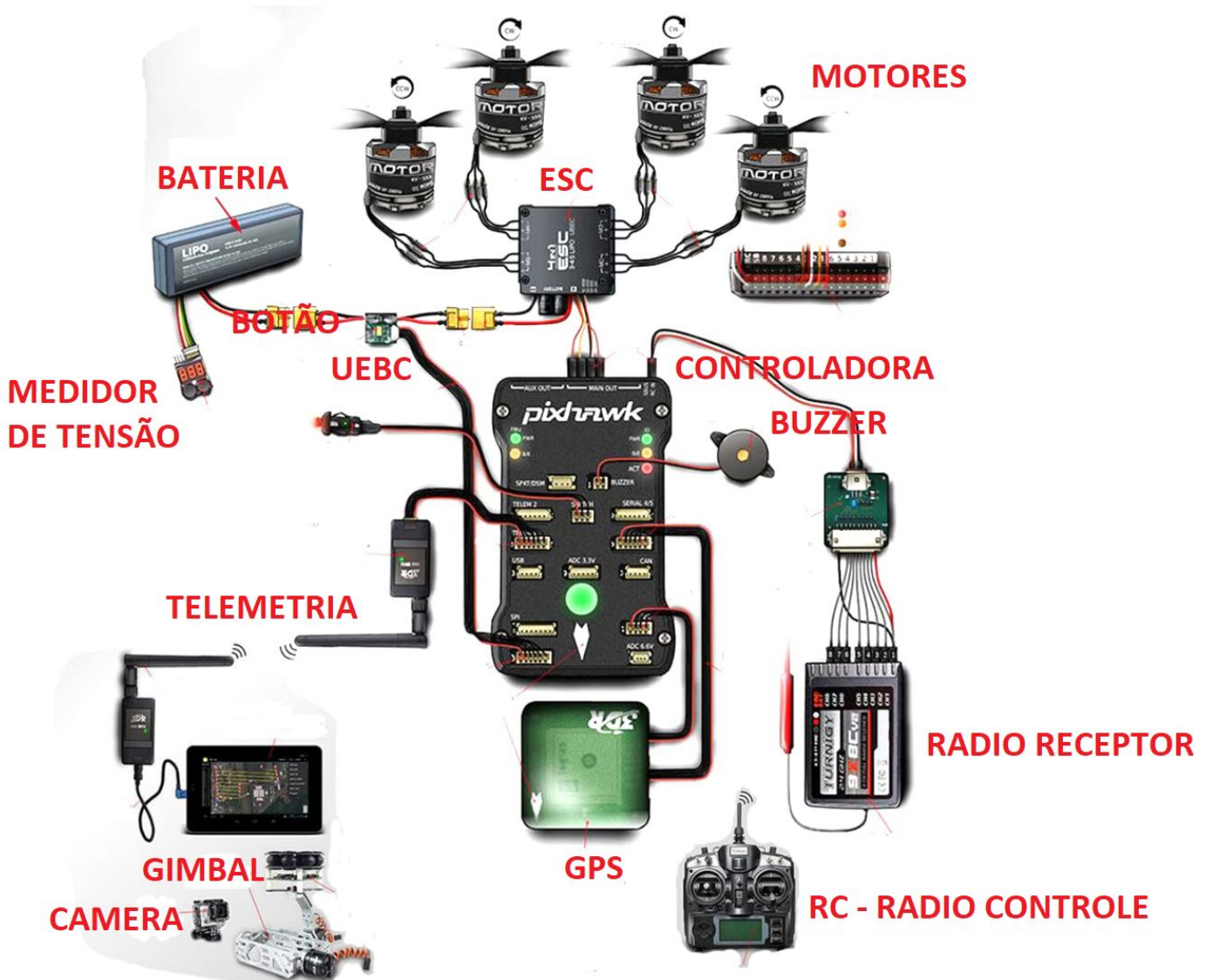


# Fim

---



# Extra: Hardware



# Extra: Hardware

Explicação de alguns componentes

- Ubec: regulador de tensão
- Telemetria: Informações de status do drone
- Gimbal: Orientação da câmera
- Buzzer: Sirene

