



**Wydział Matematyki  
i Nauk Informatycznych**

POLITECHNIKA WARSZAWSKA

# Opracowanie wirtualnego środowiska do symulacji dynamiki lotu bezzałogowych statków powietrznych

Wojciech Gajda   Igor Faliszewski

28 listopada 2023

**Politechnika  
Warszawska**



# Agenda

## 1. Wprowadzenie

### 1.1 Motywacje

### 1.2 Cel projektu

## 2. Wstęp teoretyczny

### 2.1 Dynamika statku powietrznego

### 2.2 Sterowanie statkiem powietrznym

### 2.3 Grafika komputerowa

## 3. Demo

# Motywacja

# Motywacja



Motywacja



Politechnika  
Warszawska

Motywacja



Politechnika  
Warszawska

# Motywacja



# Motywacja





# Motywacja



Politechnika  
Warszawska



**BETAFLIGHT**



# Motywacja



Politechnika  
Warszawska



# Motywacja

# Motywacja



# Motywacja



# Motywacja



# Motywacja



# Cel projektu

- First item.



# Cel projektu

- ▶ First item.
- ▶ Second item.

# Cel projektu

- ▶ First item.
- ▶ Second item.
- ▶ Third item.

# Cel projektu

- ▶ First item.
- ▶ Second item.
- ▶ Third item.
- ▶ Fourth item.

# Cel projektu

- ▶ First item.
- ▶ Second item.
- ▶ Third item.
- ▶ Fourth item.
- ▶ Fifth item.

# Cel projektu

- ▶ First item.
- ▶ Second item.
- ▶ Third item.
- ▶ Fourth item.
- ▶ Fifth item. Extra text in the fifth item.

# Wstęp teoretyczny

## Wstęp teoretyczny

“ There is nothing so practical as a good theory.

Lewin Kurt

## Wstęp teoretyczny

“ There is nothing so practical as a good theory.

Lewin Kurt

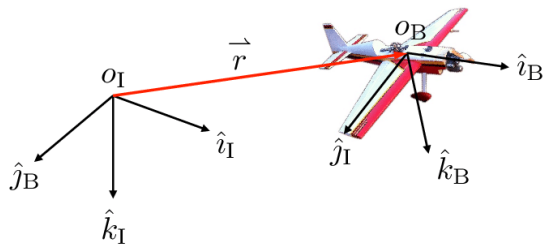
“ Nie ma osobnej ani teorii, ani praktyki inżynierskiej, jest tylko wspólna sztuka inżynierska.

prof. Jan Oderfeld

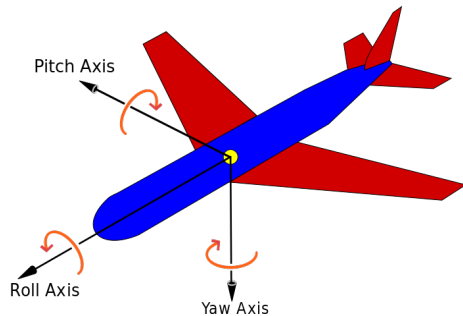
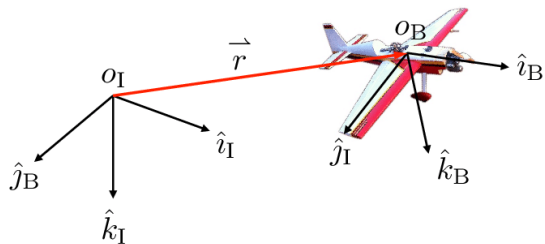


# Dynamika lotu

# Dynamika lotu



# Dynamika lotu



## Równania stanu

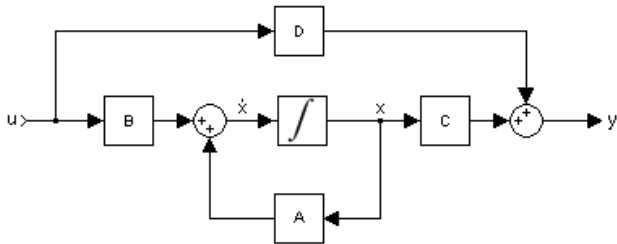
$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) \end{cases}$$

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \\ \boldsymbol{y}(t) = \boldsymbol{g}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \end{cases}$$

# Równania stanu

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{cases}$$

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = \mathbf{g}(t, \mathbf{x}(t), \mathbf{u}(t)) \end{cases}$$



# Równania różniczkowe

- ▶ Przed zastosowaniem algorytmu obniżyć rząd równania różniczkowego.

# Równania różniczkowe

- ▶ Przed zastosowaniem algorytmu obniżyć rząd równania różniczkowego.
- ▶ Skorzystać z algorytmu jawnego lub niejawnego algorytmu całkowania RR

# Równania różniczkowe

- ▶ Przed zastosowaniem algorytmu obniżyć rząd równania różniczkowego.
- ▶ Skorzystać z algorytmu jawnego lub niejawnego algorytmu całkowania RR
- ▶ Algorytmy jawne:



# Równania różniczkowe

- ▶ Przed zastosowaniem algorytmu obniżyć rząd równania różniczkowego.
- ▶ Skorzystać z algorytmu jawnego lub niejawnego algorytmu całkowania RR
- ▶ Algorytmy jawne:
  - ▶ Euler:  $\boldsymbol{x}(t + \Delta t) = \boldsymbol{x}(t) + \Delta t \cdot \dot{\boldsymbol{x}}$

# Równania różniczkowe

- ▶ Przed zastosowaniem algorytmu obniżyć rząd równania różniczkowego.
- ▶ Skorzystać z algorytmu jawnego lub niejawnego algorytmu całkowania RR
- ▶ Algorytmy jawne:
  - ▶ Euler:  $\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \dot{\mathbf{x}}$
  - ▶ Rugge-Kutty 4 rzędu

# Model matematyczny statku powietrznego I

# Kolizje I

Odrzut

# Odrzut

Zasada zachowania pędu  $\vec{p}$  i zasada zachowania momentu pędu (krętu)  $\vec{L}$ .

# Odrzut

Zasada zachowania pędu  $\vec{p}$  i zasada zachowania momentu pędu (krętu)  $\vec{L}$ .

$$\begin{bmatrix} \vec{p}_{przed} \\ \vec{L}_{przed} \end{bmatrix} = \begin{bmatrix} \vec{p}_{po} \\ \vec{L}_{po} \end{bmatrix} + \begin{bmatrix} \vec{p}_{pocisku} \\ \vec{L}_{pocisku} \end{bmatrix}$$

# Odrzut

Zasada zachowania pędu  $\vec{p}$  i zasada zachowania momentu pędu (krętu)  $\vec{L}$ .

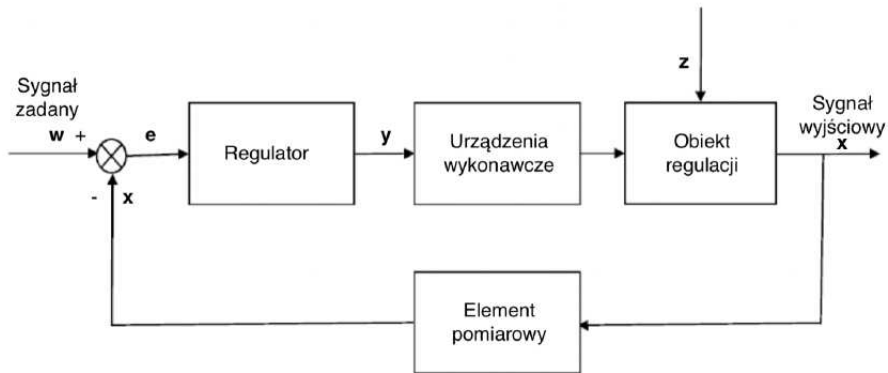
$$\begin{bmatrix} \vec{p}_{przed} \\ \vec{L}_{przed} \end{bmatrix} = \begin{bmatrix} \vec{p}_{po} \\ \vec{L}_{po} \end{bmatrix} + \begin{bmatrix} \vec{p}_{pocisku} \\ \vec{L}_{pocisku} \end{bmatrix}$$

$$M \begin{bmatrix} \vec{v}_{przed} \\ \vec{\omega}_{przed} \end{bmatrix} = M \begin{bmatrix} \vec{v}_{po} \\ \vec{\omega}_{po} \end{bmatrix} + m_{pocisku} \begin{bmatrix} \vec{v}_{pocisku} \\ \vec{r}_{pocisku} \times \vec{v}_{pocisku} \end{bmatrix}$$

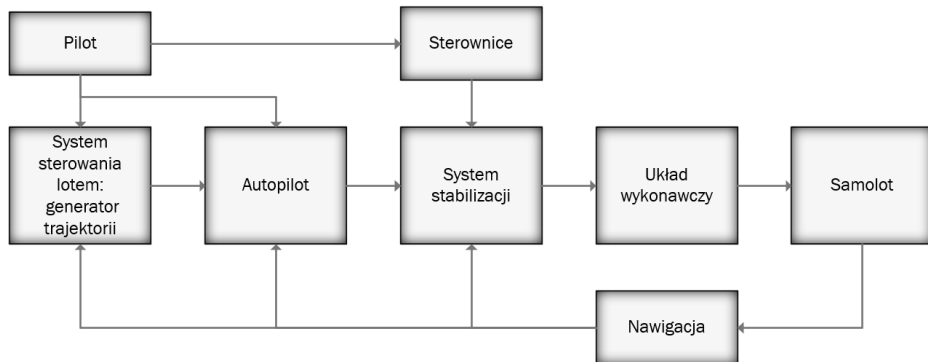


# Sterowanie statkiem powietrznym

# Sterowanie statkiem powietrznym

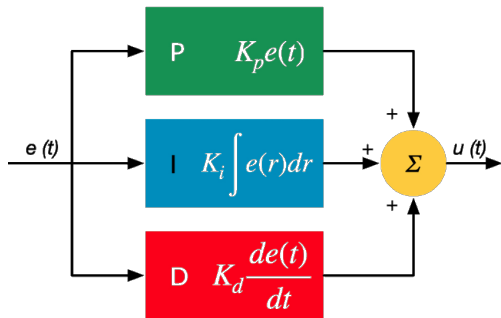


# Sterowanie statkiem powietrznym

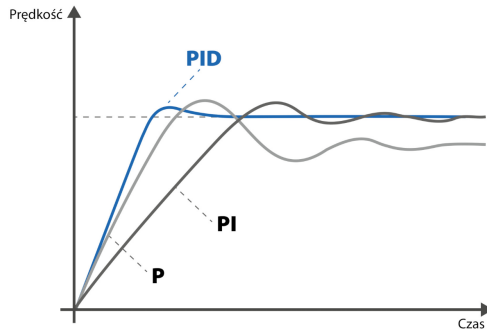
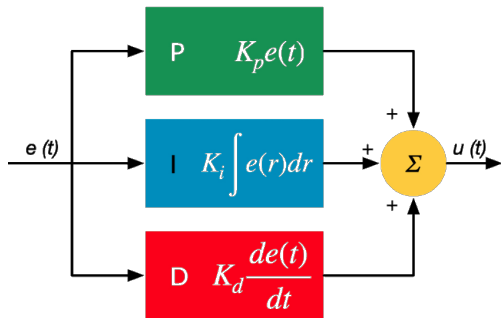


# Regulator PID

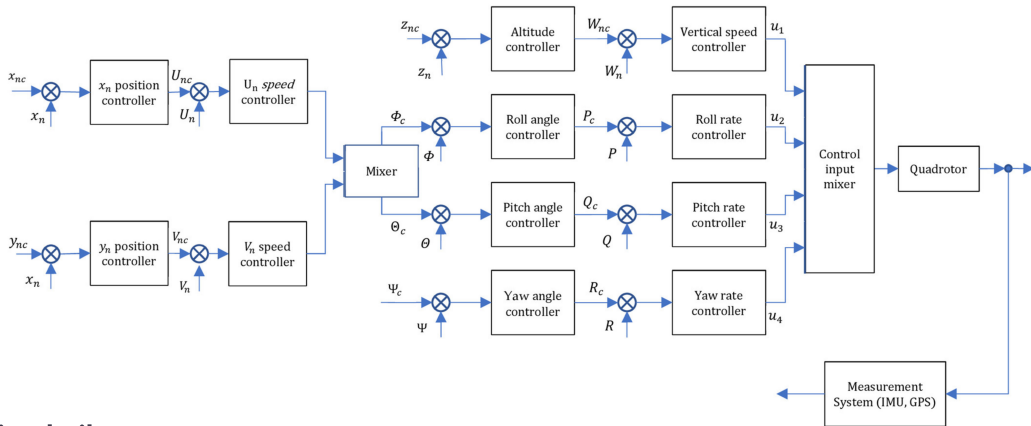
# Regulator PID



# Regulator PID



# Sterowanie statkiem powietrznym



# Nawigacja

## Czujniki:

- ▶ Żyroskop
- ▶ Akcelerometer
- ▶ Barometer
- ▶ Czujnik prędkości powietrza
- ▶ Nawigacja satelitarna
- ▶ Radar, sonar, lidar

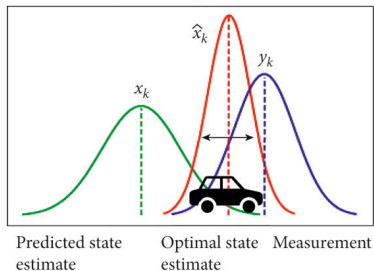


# Nawigacja

## Czujniki:

- ▶ Żyroskop
- ▶ Akcelerometer
- ▶ Barometer
- ▶ Czujnik prędkości powietrza
- ▶ Nawigacja satelitarna
- ▶ Radar, sonar, lidar

## Filtr Kalmana:



# Potok renderowania I

# Shadery I

# GPU I

# Cieniowanie i model oświetlenia I

# Renderowanie interfejsu I

# Obsługa kontrolera I

# Krzywa łańcuchowa I

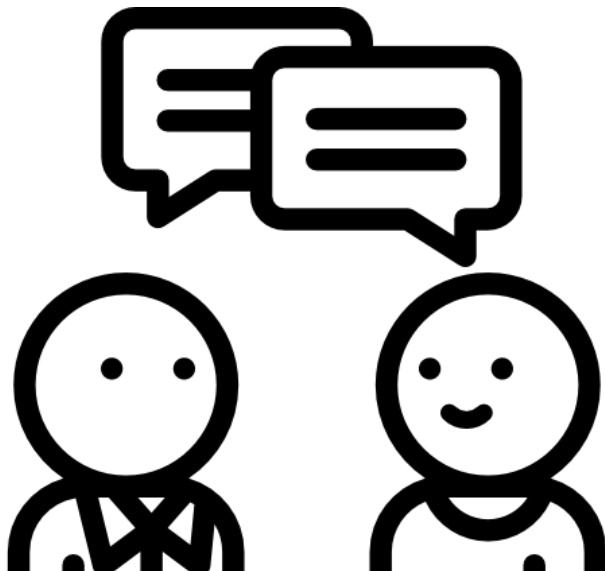


# Demo

Testy  $\alpha$



# Dyskusja



# Literatura



[Energies, 2022] Quadrotor Model for Energy Consumption Analysis  
Jacewicz, Mariusz and Żugaj, Marcin and Głębocki, Robert and Bibik,  
Przemysław

Dziękuję za uwagę!