

Model Driven Engineering Approach to Design Sensing and Actuation Subsystems

Fernando S. Gonçalves*
Leandro B. Becker*

* Automation and Systems Department – DAS
Federal University of Santa Catarina - UFSC



Universidade Federal
de Santa Catarina





Universidade Federal
de Santa Catarina

Outline



- Motivation
- Proposed Approach
- Conclusions

PROVant Project

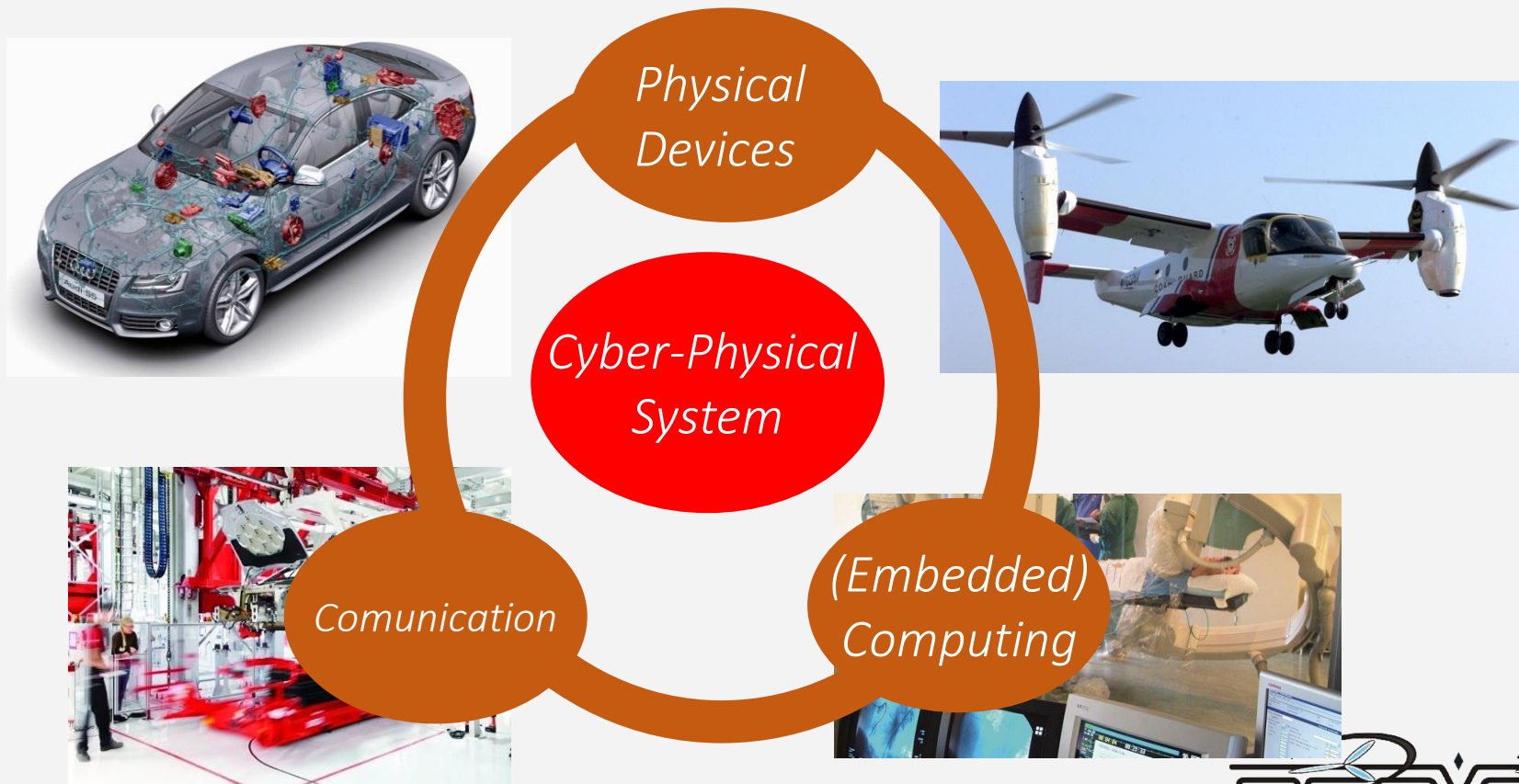
- Focus on design autonomous bi-rotor UAVs;
- Integrate different design teams (Control, Aeronautic, Software);
- Composed by at least 18 people (PhD, Master, and Undergrad students).





Cyber-Physical Systems (CPS)

CPS combines computation and physical systems, providing proper control

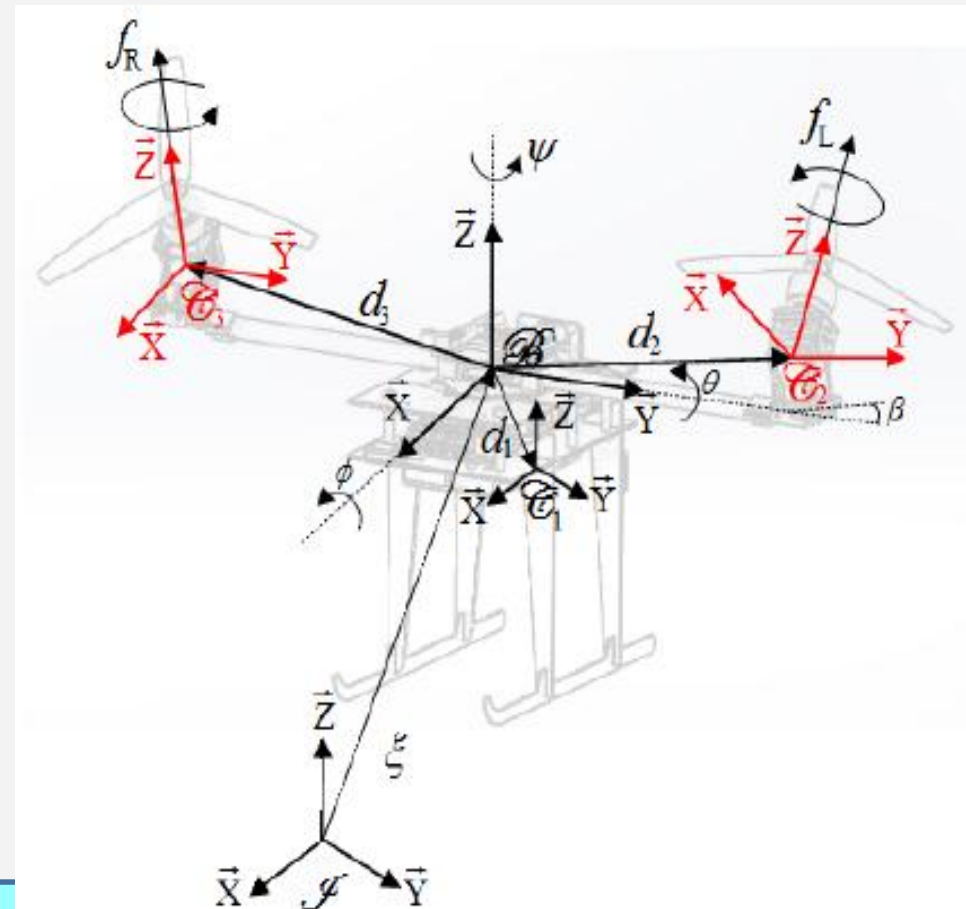


CPS Design Process

Different types of models:

(1/3) Physical Plant Model

- Analytical model from the device to be controlled

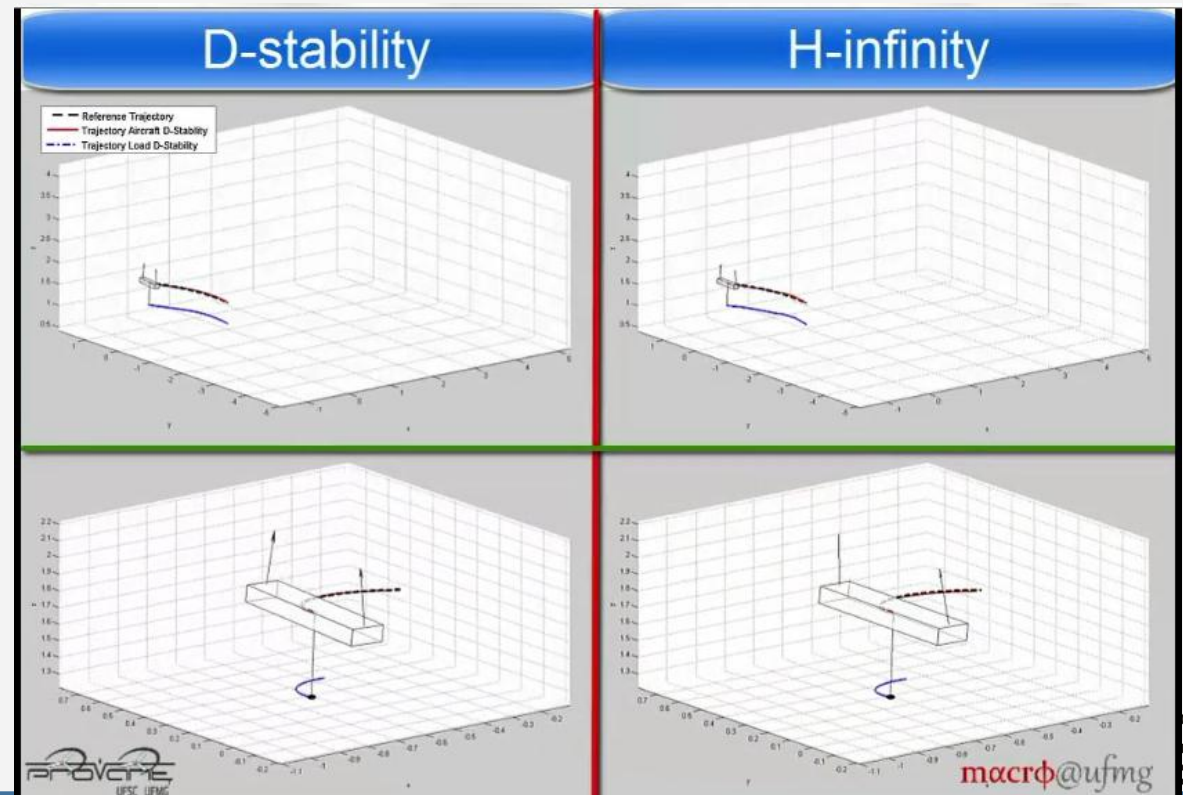


CPS Design Process

Different types of models:

(2/3) Functional Model

- System Operating Modes
- Control Algorithms



CPS Design Process

Different types of models:

(3/3) Architectural Model

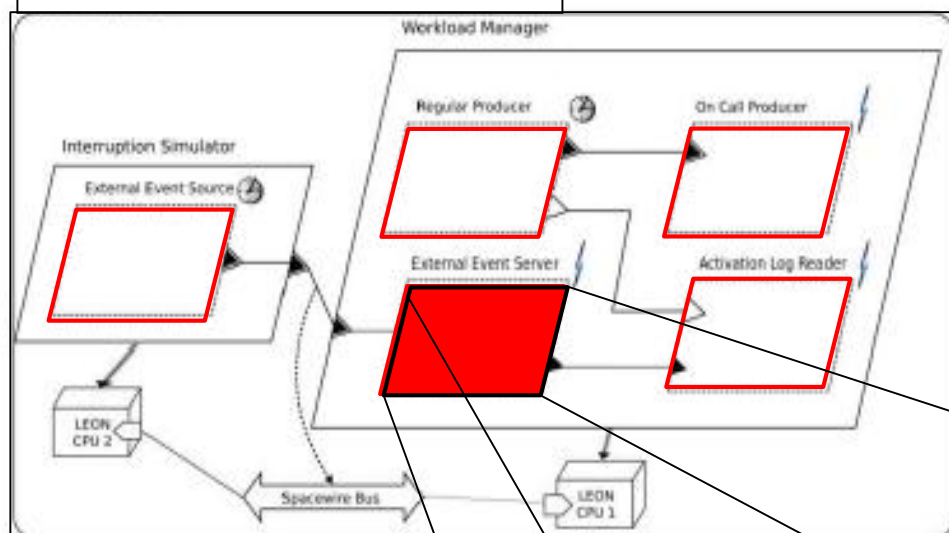
- Software organization (processes, threads, communication)
- Target architecture
- Software deployment (on target architecture)



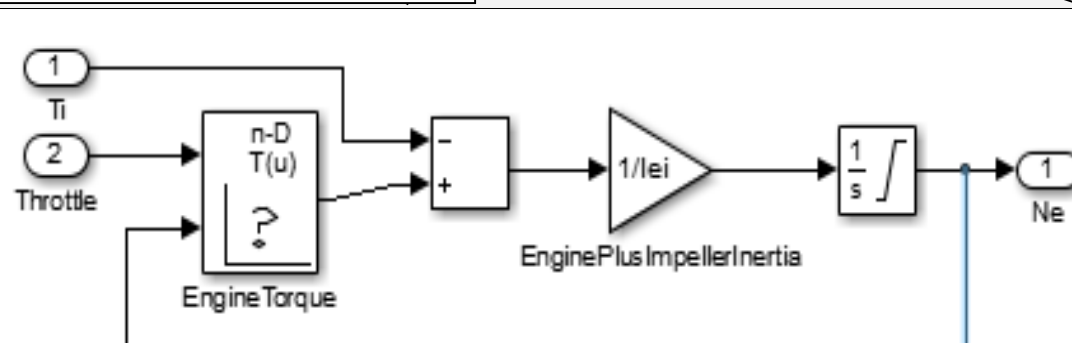
CPS Design Process

(3/3) Architectural Model

Architectural Model



Functional Model



Paper Scope

- Address how to perform the **transition from simulation model to the implementation model**;
- **Replace the analytical (physical) model**, that represents the physical devices, **for the real sensors and actuators**;
- Provide **adequate tool support** for the design process;



Why Simulink?

- Robust tool that support the design of discrete and continuous systems;
- Component library to support the systems design;
- Provide interface with the embedded platforms;
- Support system simulations and analysis;

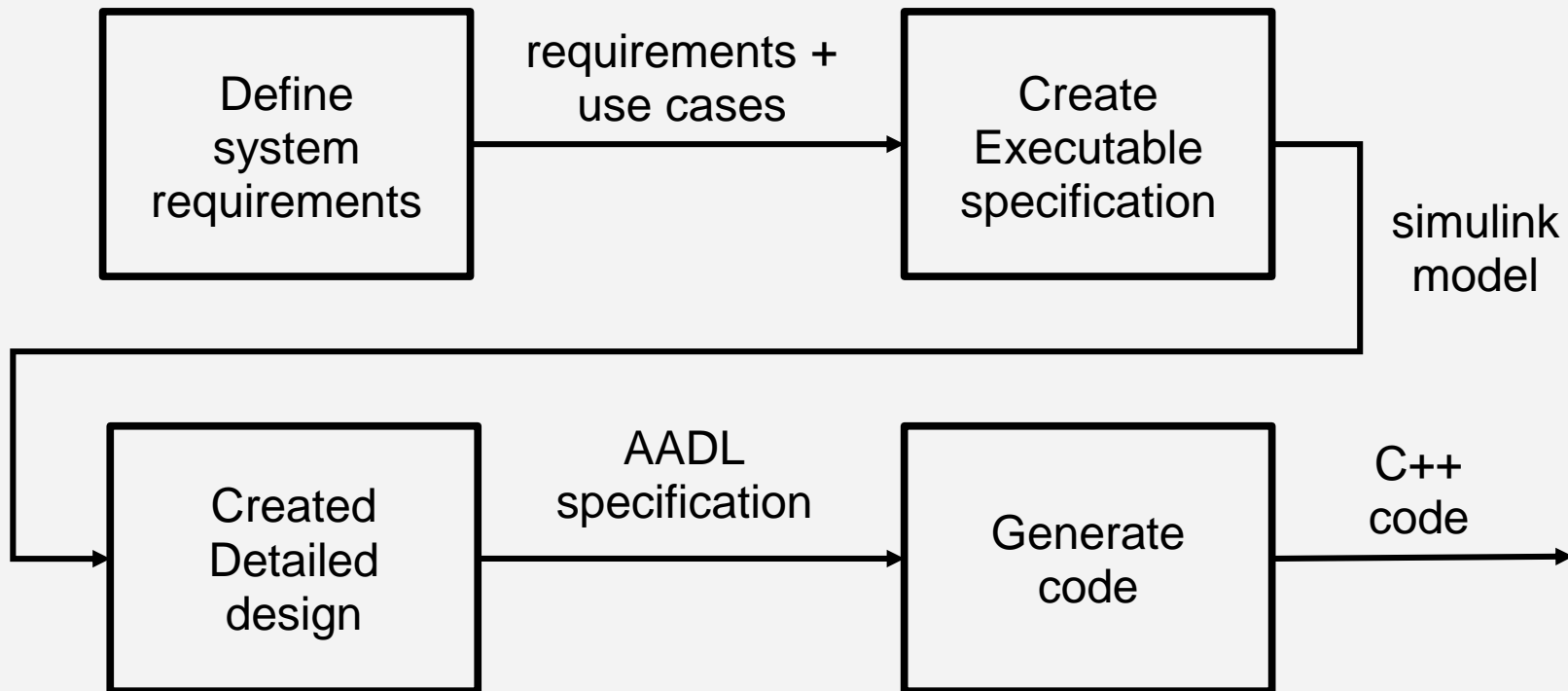


Why AADL?

- Language that support the specification of the system architecture;
- Provide structures to represent and integrate software and hardware components;
- Support analysis of different system properties;



Research Contextualization



Main activities and artifacts of the adopted method to develop CPS (from Passarini et. al 2015)

- Approach does not comprise the generation of the set of sensors and actuators needed by the CPS;





Universidade Federal
de Santa Catarina

Outline



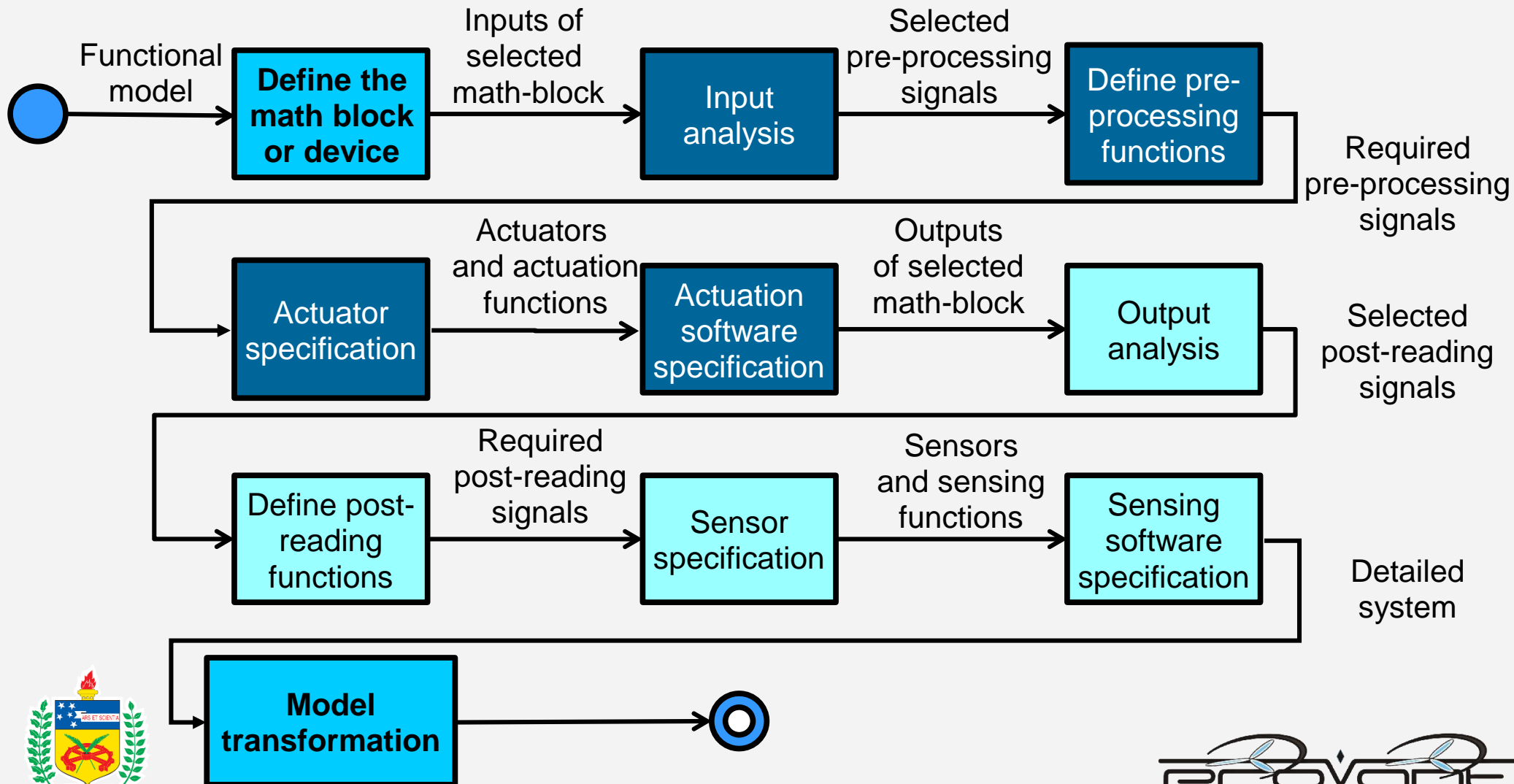
- Motivation
- **Proposed Approach**
- Conclusions

Proposed Approach - ECPSModel

- Extends previous approach, allowing to represent sensors and actuators characteristics before model transformation
 - Define the set of **input devices (sensors)** and **output devices (actuators)**
 - Specify the required sensing and actuating **functions**;
 - **Organize the functions and devices** in a software structures;

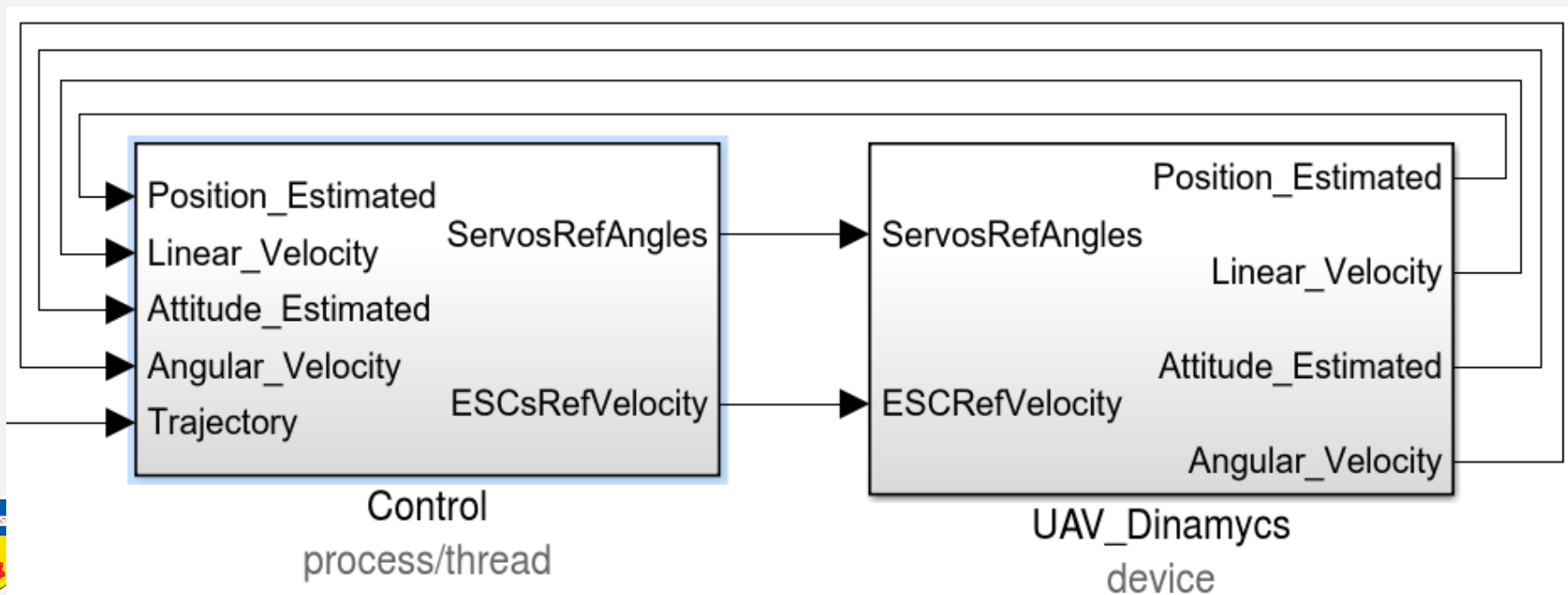
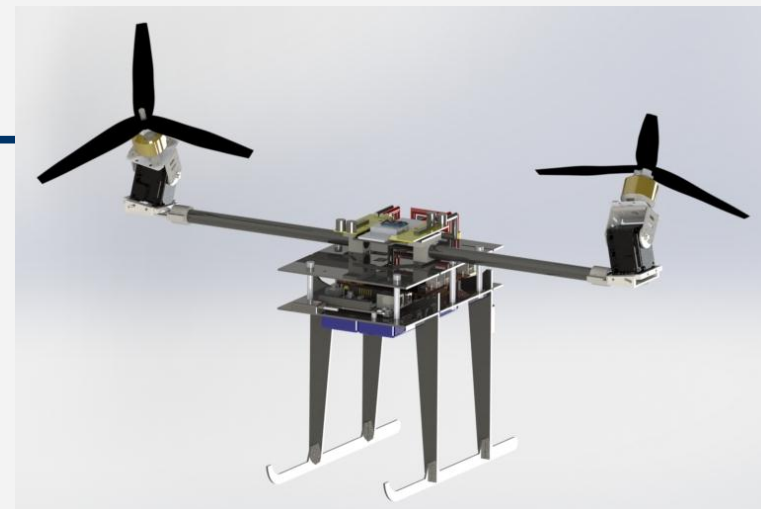


Proposed Approach - ECPSModel



Case Study

- Bi-rotor tilt-rotor UAV;
- Functional model is composed by two blocks;



Select the mathematical block

File Import Wizard

Sensing and Actuation Modeling

Define the System Mathematical Model:

Define the subsystem that represent the mathematical model:

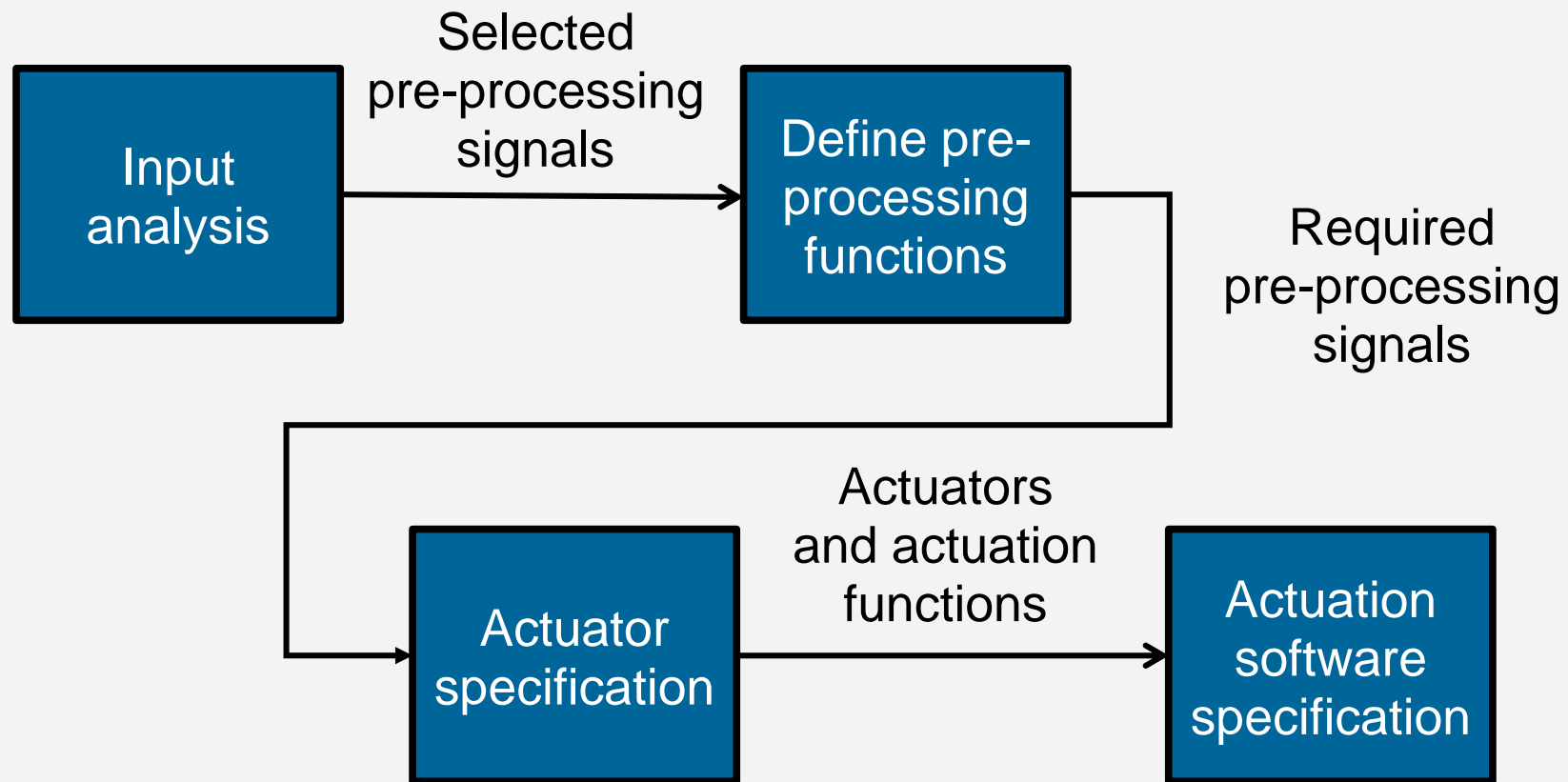
Output model: AADL Model

Subsystems
uav
control
uav_dinamycs

? < Back Next > Finish Cancel



Actuation definition



Input analysis

File Import Wizard

Actuation Analyze

Detail the Actuation subsystem:

Analyze the input ports of the mathematical model:

Input	Vector	Inputs Size	Pre-writing	Actuator
servosrefangles	<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	
escfvelocity	<input type="checkbox"/>	1	<input type="checkbox"/>	

ESC
Servo
Motor

? < Back Next > Finish Cancel



Actuators specification

File Import Wizard

Actuator Specification

Detail the system Actuators:

Specify the system actuator that compose the application:

Signal	Actuator	Protocol	Priority	Periodic	Period (...)
ForceR	ESC	I2C	1	<input checked="" type="checkbox"/>	5
ForceL	ESC	I2C	1	<input checked="" type="checkbox"/>	5
AngleR	Servo	Serial	1	<input checked="" type="checkbox"/>	5
AngleL	Servo	Serial	1	<input checked="" type="checkbox"/>	5

Edit Actuator

Actuator Specification

Signal: AngleL

Actuator: Servo

Protocol: Serial

Priority: 1

Periodic:

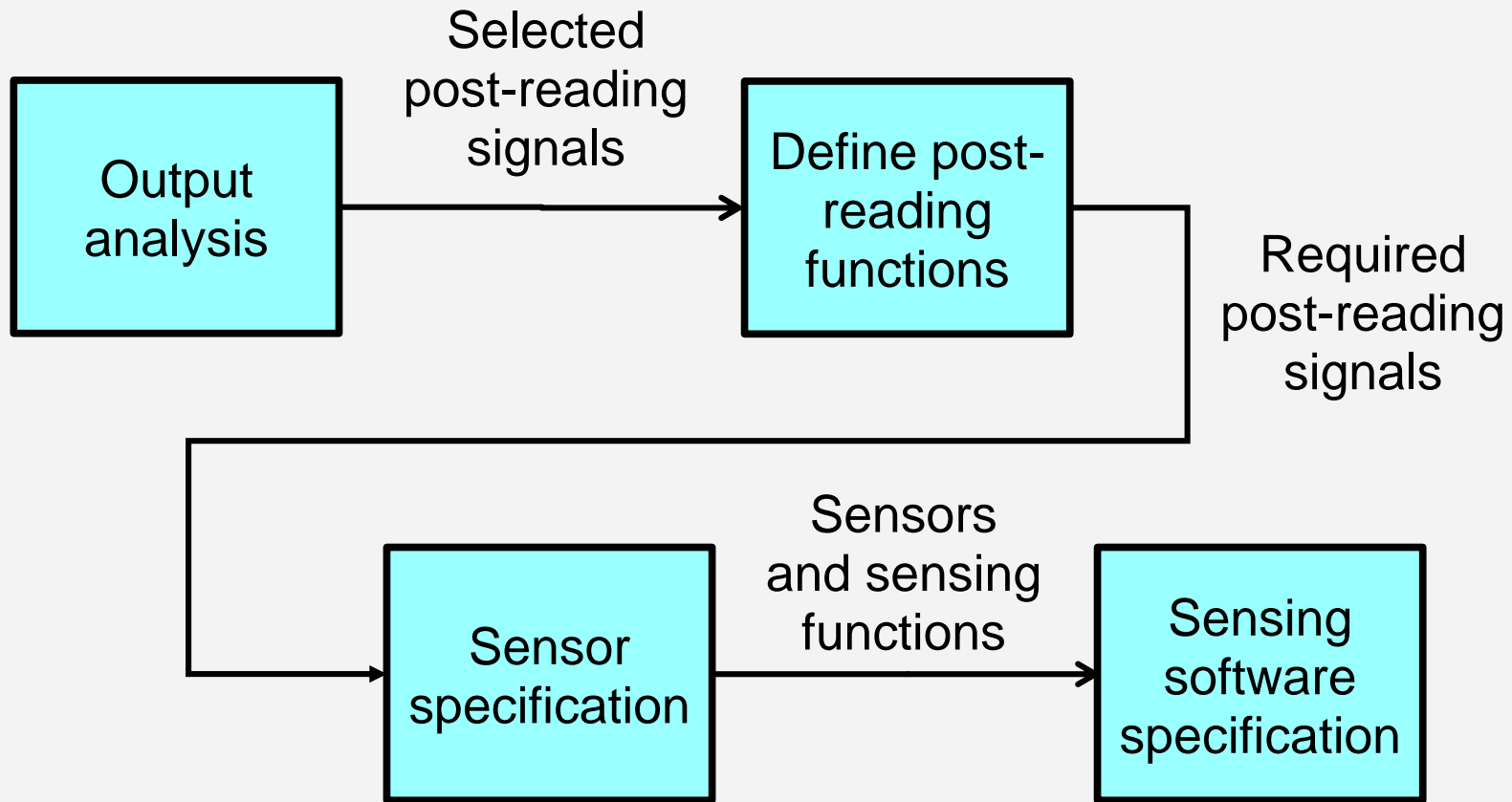
Period (ms): 5

OK Cancel

? < Back Next > Finish Cancel



Sensing definition



Output analysis

File Import Wizard

Sensing Analyze

Detail the Sensing subsystem:

Analyze the output ports of the mathematical model:

Input	Vector	Inputs Size	Post-reading	Sensor
position_estimated	<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	
linear_velocity	<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	
attitude_estimated	<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	
angular_velocity	<input type="checkbox"/>	1	<input type="checkbox"/>	

IMU
GPS
Sonar
Barometer
Encoder

? < Back Next > Finish Cancel



Sensor specification

File Import Wizard

Sensor Specification

Detail the Sensing Subsystem:

Specify the set of sensors that compose the system:

Signal	Sensor	Protocol	Priority	Periodic	Period (...)
ImuData	IMU	I2C	1	<input checked="" type="checkbox"/>	5
SonarData	Sonar	Serial	1	<input checked="" type="checkbox"/>	5
GpsData				<input type="checkbox"/>	

Sensor Specification

Signal: GpsData

Sensor: GPS

Protocol: Serial

Priority: 1

Periodic:

Period (ms): 100

OK Cancel

Edit Sensor

? < Back Next > Finish Cancel



Sensing software specification

Add Periodic Thread [X]

Thread:

Template:

Periodic: Period (ms): Priority:

Sensors **Functions**

Sensors	Thread Sensors
GPS	IMU
	Sonar

<< >>

OK Cancel

Add Periodic Thread [X]

Thread:

Template:

Periodic: Period (ms): Priority:

Sensors **Functions**

Functions	Thread Functions
PositionEstimation	BehaviorEstimation

<< >>

OK Cancel



Generated output model

```
1 SYSTEM IMPLEMENTATION UAV.impl
2 SUBCOMPONENTS
3 --PROCESS
4 pi_control: PROCESS p_control.impl;
5 pi_sen_act: PROCESS p_sen_act.impl;
6 --DEVICE
7 di_esc_r: DEVICE d_esc.impl;
8 di_esc_l: DEVICE d_esc.impl;
9 di_servo_r: DEVICE d_servo.impl;
10 di_servo_l: DEVICE d_servo.impl;
11 di_gps: DEVICE d_gps.impl;
12 di_sonar: DEVICE d_sonar.impl;
13 di_imu: DEVICE d_imu.impl;
14 CONNECTIONS
15 C1: PORT di_gps.position -> pi_sen_act.position;
... Here goes all the connections (lines 16 to 35)
35 END UAV.impl;
```



Generated output model

```
1 PROCESS IMPLEMENTATION p_est_act.impl
2 SUBCOMPONENTS
3   ti_signalTransformation: THREAD t_signalTransformation.impl;
4   ti_behaviorEst: THREAD t_behaviorEst.impl;
5   ti_positionEst: THREAD t_positionEst.impl;
6 CONNECTIONS
7   C1: PORT distance -> ti_behaviorEst.distance;
... Here goes all the connections (lines 8 to 25)
26 END p_est_act.impl;
```



Generated output model

```
1  THREAD IMPLEMENTATION t_behaviorEst.impl
2  CALLS
3  Mycalls: {
4    P_Spg : SUBPROGRAM behaviorEstimation;
5  };
6  PROPERTIES
7    Dispatch_Protocol => Periodic;
8    Period => 5ms;
9    Deadline => 5ms;
10 END t_behaviorEst.impl;
```





Universidade Federal
de Santa Catarina

Outline



- Motivation
- Proposed Approach
- **Conclusions**

Conclusions

- ECPSModeling tool was created to support the automated design of CPS;
- Systematize the design process of sensing and actuation subsystems;
- Generated AADL model integrate control algorithms, estimation filters and the system devices.
- Use of Design-Space Exploration techniques in order to generate more elaborated models;
- Integrate verification methods to evaluate and guarantee the system properties.



Thank you

Questions ?

fernando.goncalves@posgrad.ufsc.br

leandro.becker@ufsc.br



Universidade Federal
de Santa Catarina

