CPSwarm

Final Workshop

Towards Smart Autonomous Cyber-Physical Systems: Unmanned Aerial/Ground Vehicles and Robots

Simulation and Optimization

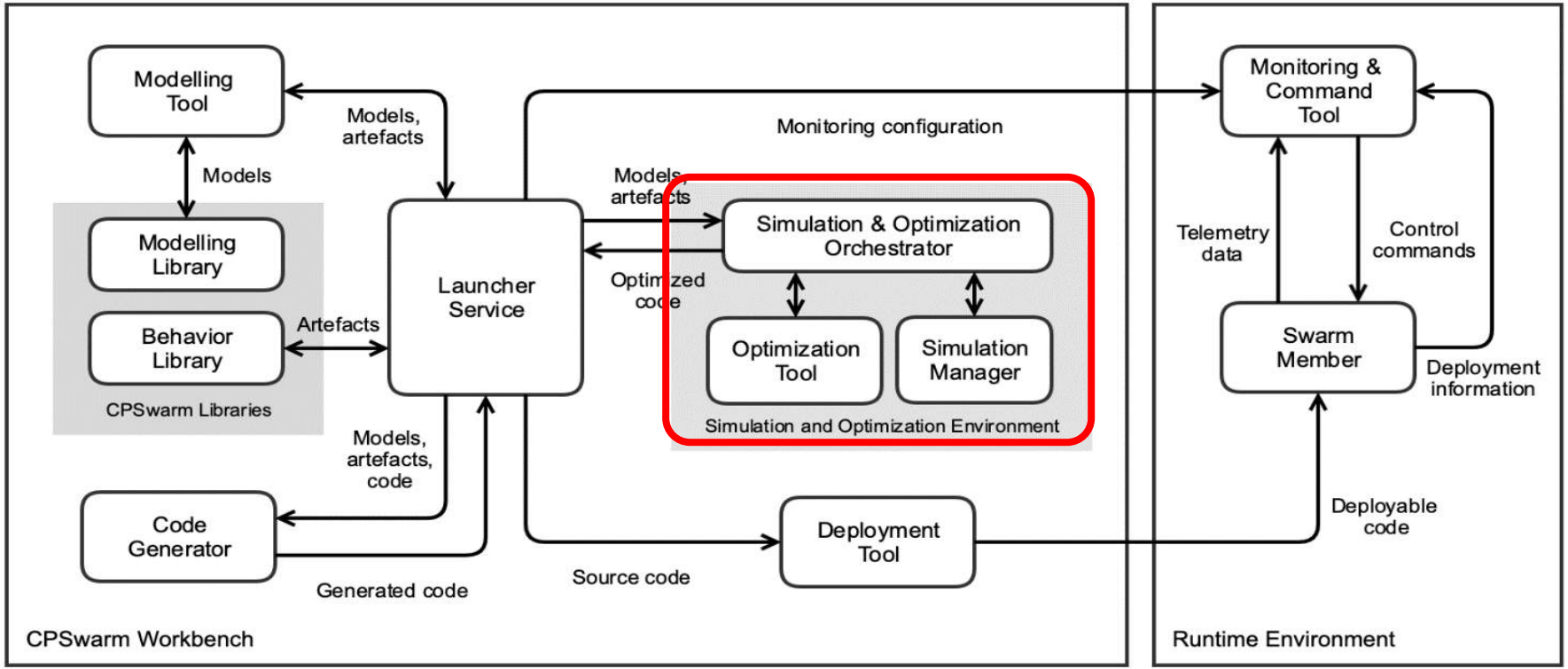
Davide Conzon, Links Foundation, CPSwarm

Turin, December 13th 2019

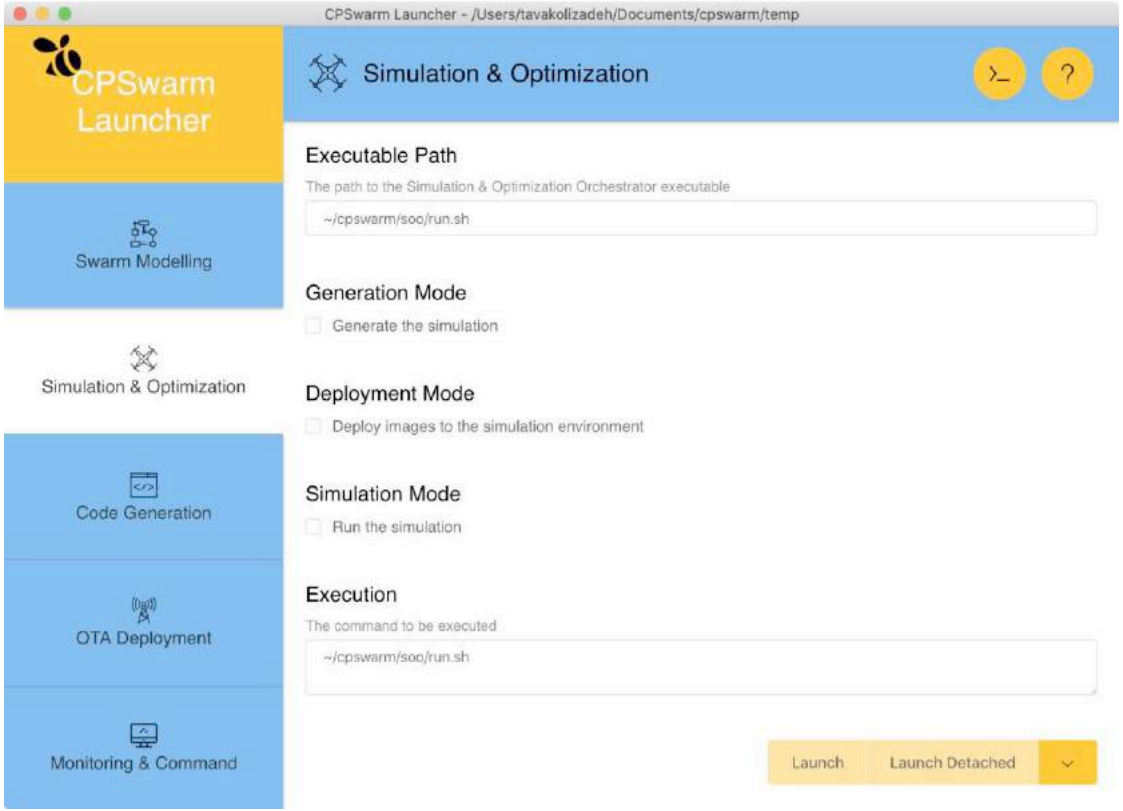
Simulation and Optimization

Role in the CPSwarm Workbench

CPSwarm Simulation and Optimization Environment



CPSwarm Optimization Environment – Launcher Interface

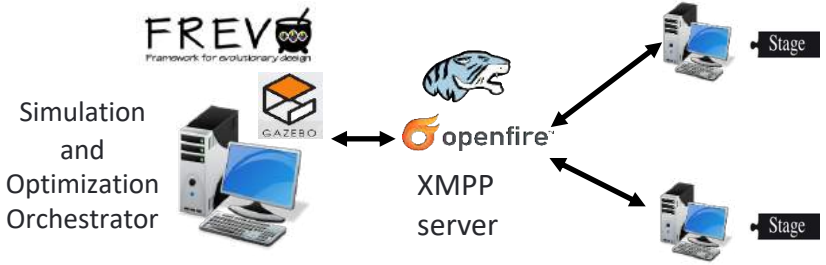


Simulation and Optimization

Main Outcomes

DISTRIBUTED ARCHITECTURE

- The Simulation and Optimization Environment **evaluates the performance** of a swarm solution.
- It can run a simulation or in combination with the Optimization Tool iteratively evolve parameters of the controller algorithm/module.
- During optimization, candidate parameter sets are ranked based on a **fitness score** computed by executing the controller in a predefined environment. Successful candidates are adapted to produce a new generation of controllers.
- The Simulation and Optimization Environment utilizes a **distributed architecture** based on the XMPP protocol which allows simulations to be executed in parallel on simulators such as **Stage** and **Gazebo**.



Distributed Simulation for Evolutionary Design of Swarms of Cyber-Physical Systems. ADAPTIVE 2018, February 2018

Scalable Distributed Simulation for Evolutionary Optimization of Swarms of Cyber-Physical Systems. SysMea, August 2019.

The CPSwarm Technology for Designing Swarms of Cyber-Physical Systems. STAF 2019, July 2019

Simulation and Optimization Orchestrator

- The **Simulation and Optimization Orchestrator** is the **interface** between the Simulation and Optimization Environment and the rest of the Workbench. It receives the requests from the user (through the Launcher) and **orchestrates** the connected Simulation Tools (wrapped by the Simulation Managers) and the Optimization Tool.
- It supports several **different modes of execution**:

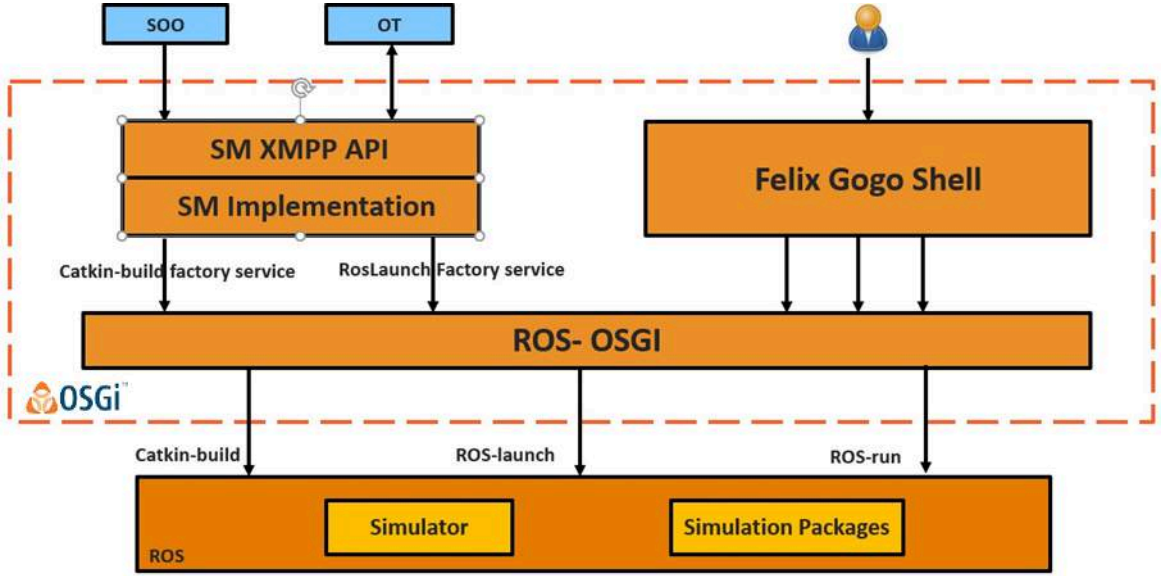
Generation: using a **SCXML** file generated by the Modelling Tool, the Simulation and Optimization Orchestrator can generate the **ROS simulation package** required for simulating and optimizing an algorithm.

Deployment: the Simulation and Optimization Orchestrator can **rapidly deploy** and **orchestrate** a set of Simulation Managers to parallelize the large number of simulations needed to complete an optimization.

Simulation: the Simulation and Optimization Orchestrator facilitates **running simulations (with optimization)** remotely, **automatically selecting** the Simulation Tool based on user's **requirements.**

Simulation Managers

- The Simulation Managers allow the **seamless integration** and (remote) **control** of heterogenous simulators.
- Currently, **ROS-based simulators** are supported, with Simulation Managers for **Stage** and **Gazebo** available on the CPSwarm GitHub repository.



Features

- Based on Standard **OSGi** technology and **ROS-OSGi** libraries for modular and standard implementation.
- Integration **with opensource OSGi tools** for **simulation control** via command line and web interface.
- Implementation of **XMPP API** to support the **remote configuration and control** of the wrapped simulation tools

Optimization Tool

- The **Optimization Tool** is in charge of optimizing the behaviour of algorithms, **iteratively evolving** parameters of the controller algorithm/module.
- The CPSwarm Workbench integrates the **FRamework for EVOolutionary design** (FREVO) into its toolset, which has been extended in the project to **support XMPP** communication with the Simulation and Optimization Orcestrator and Simulation Managers.

FREVO
Framework for evolutionary design

Parameter optimization

- Most algorithms have **parameters** that can be tuned to **increase system performance.**
- Tuning these by applying **evolutionary techniques** and **automated simulation** run on distributed simulators is **computationally efficient**.

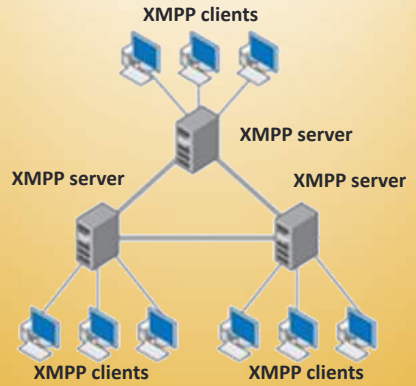
Fitness function integration

- A **fitness function** judges the performance of a set of parameters by **automatically assessing** the outcome of a simulation. It is used to **drive the optimization process.**
- CPSwarm has defined **guidelines** for fitness function definition.
- The Simulation and Optimization Environment integrates the fitness functions **generated** by the Modelling tool.

Simulator API

- API for **discovery**, **configuration** and **control** of a set of distributed Simulation Tools implemented using the **XMPP protocol**.
- XMPP is a an **open standard** (with **10 years** of maturity), providing proven features of **scalability**, **reliability** and **security**, with a **large community**. It supports: **presences**, **1-to-1 communication** and **file transfer**.

Decentralized architecture



Real-time presences

- **Discovery** of new simulators.
- **Addition/removal** of simulators at runtime.
- **Identification** of errors and connection problems in real time, for **recovery**.

Configuration and control

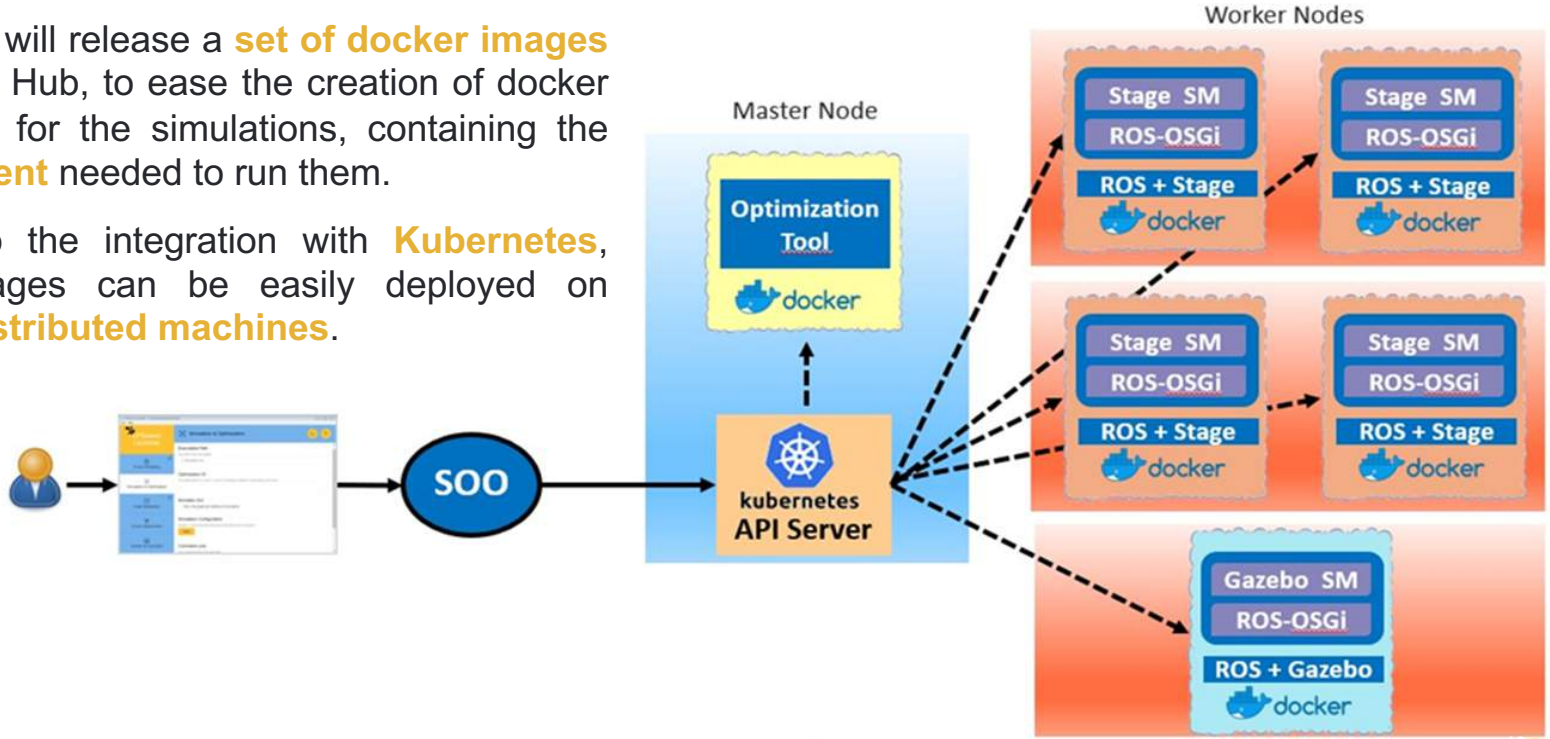
- **Remote configuration** of simulator tools.
- **API to run a simulation** on (remote) simulation tools with different parameters.
- **Integration** with an Optimization Tool to run optimization tasks using **distributed simulators**.

Simulation package generation

- The Simulation and Optimization Orchestrator can **generate** the simulation to be run.
- Currently, the Simulation and Optimization Orchestrator supports only the generation of ROS packages to be used in ROS-based simulators, based on the **state machine** supplied by the Modelling Tool.
- The approach is similar to the one used by the CPSwarm Code Generator, the input is a description of a **Finite State Machine**, the output is the **ROS package** to be used in the simulation tool, for simulation and optimization (not all the code is generated, some parts have to be written manually or using external tools). The code generation process is driven by a **Java-based template engine** called Velocity.

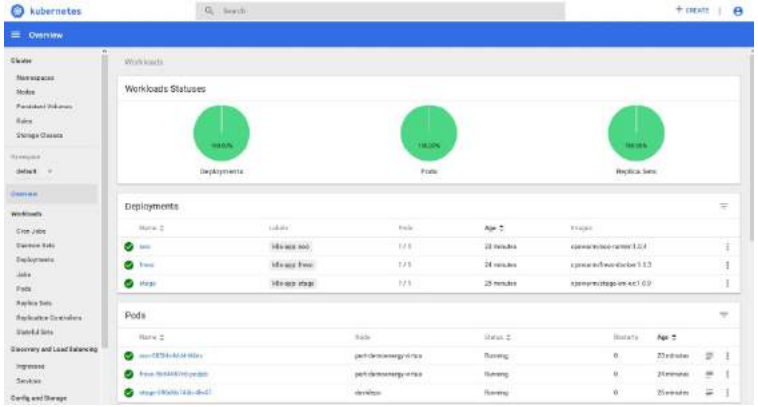
Deployment and orchestration

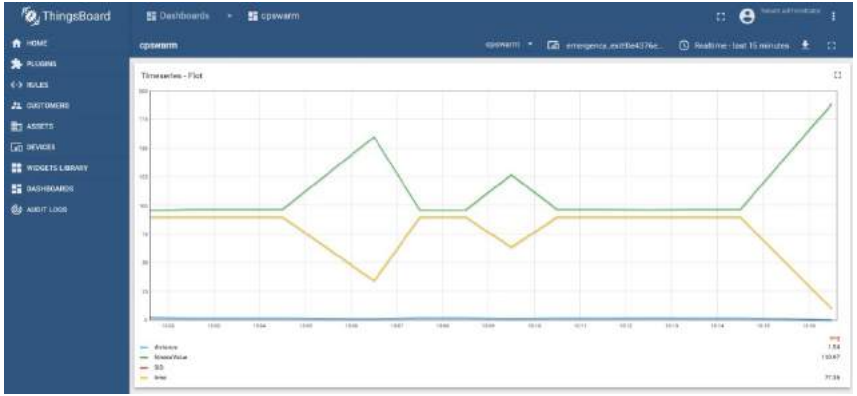
- Kubernetes is a solution for the **rapid deployment** and orchestration of containerized applications (i.e., **docker containers**) on a **cluster of distributed machines**. The Simulation and Optimization Orchestrator integrates a Kubernetes client to ease the deployment and setup of simulation servers.
- CPSwarm will release a **set of docker images** on Docker Hub, to ease the creation of docker containers for the simulations, containing the **environment** needed to run them.
- Thanks to the integration with **Kubernetes**, these images can be easily deployed on several **distributed machines**.



Monitoring

- Integration of **opensource** tools (i.e., Kubernetes and Thingsboard dashboards) to enable **orchestration, management and monitoring** of distributed cluster of simulation servers.

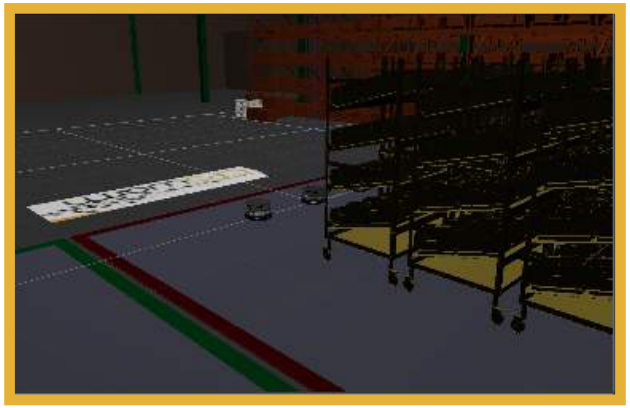


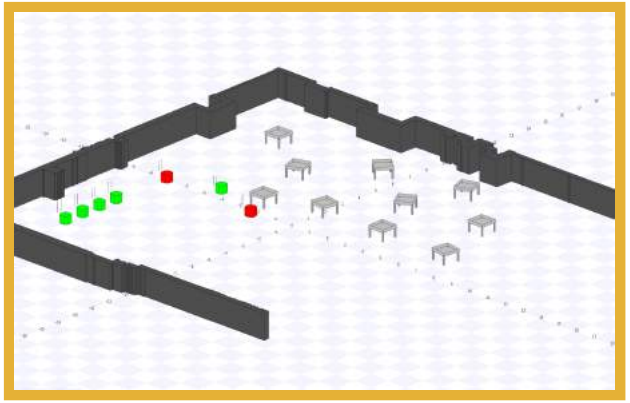


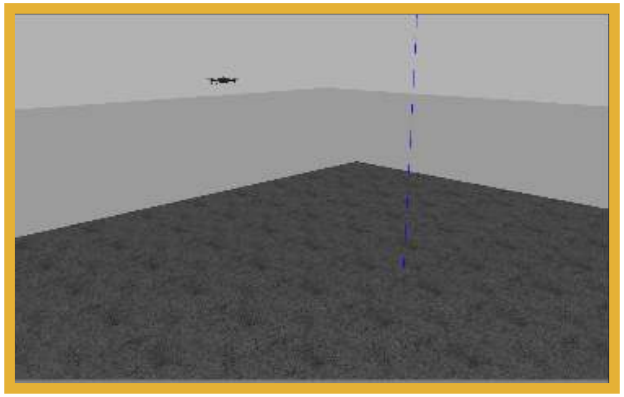
kubernetes

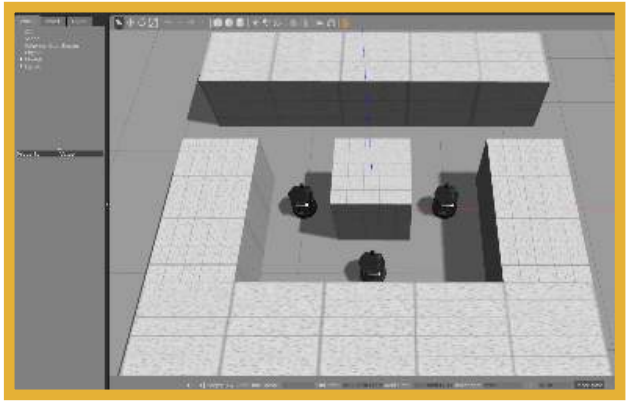
ThingsBoard

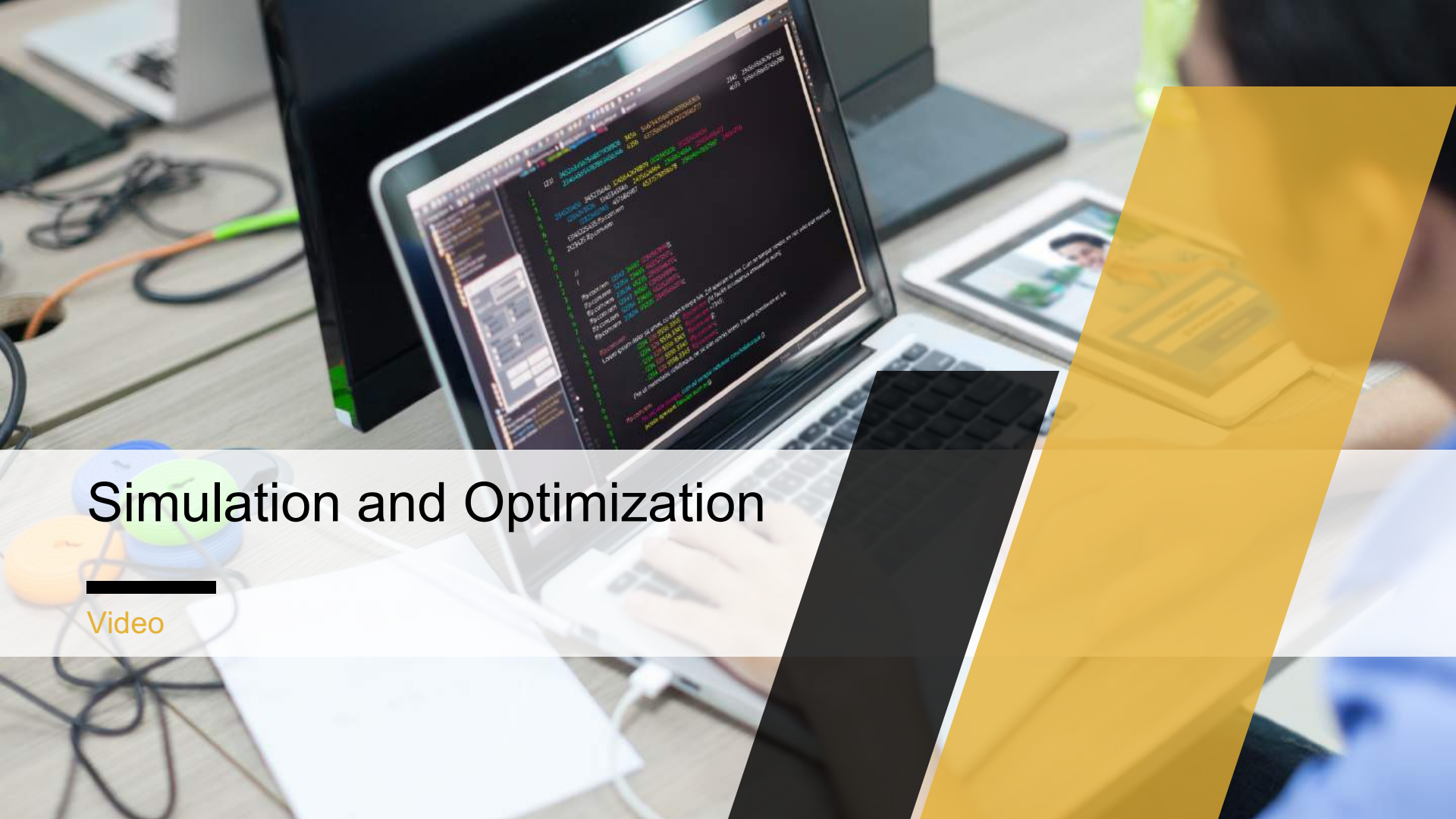
Simulations











Simulation and Optimization

Video

Demo video

THANK YOU!
QUESTIONS?

Web References

- Simulation and Optimization Orchestrator: https://github.com/cpswarm/SimulationOrchestrator
- Stage Simulation Manager: https://github.com/cpswarm/StageSimulationManager
- Gazebo Simulation Manager: https://github.com/cpswarm/GazeboSimulationManager
- FREVO: https://github.com/cpswarm/FREVO
- Stage: https://github.com/rtv/Stage
- Gazebo: http://gazebosim.org/
- XMPP: https://xmpp.org/
- Kubernetes: https://kubernetes.io/
- OSGi: https://www.osgi.org/
- ROS-OSGi: https://github.com/ibcn-cloudlet/rososgi
- SCXML: https://www.w3.org/TR/scxml/
- Docker: https://www.docker.com/
- Thingsboard: https://thingsboard.io/

CPSwarm

Final Workshop

Towards Smart Autonomous Cyber-Physical Systems: Unmanned Aerial/Ground Vehicles and Robots

Davide Conzon

LINKS Foundation – CPSwarm

davide.conzon@linksfoundation.com

www.cpswarm.eu

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731946

CPSwarm

CPSwarm

http://www.cpswarm.eu

Follow @CPSwarm_EU

Coordinator

Partners

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731946.