MRS UAV System for Real-world Testing and Development with Multirotor Aerial Vehicles

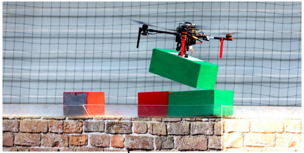
From control theory to practical use

Dr. Tomas Baca

Multi-Robot Systems group, Faculty of Electrical Engineering
Czech Technical University in Prague

Simulation-to-reality gap

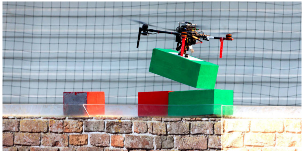
Simulation



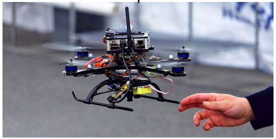
Reality

Simulation-to-reality gap

Simulation



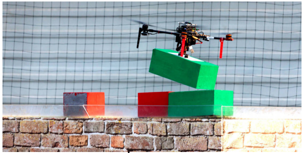
Reality



Worked once in a lab

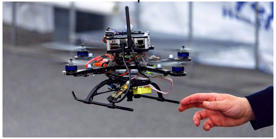
Simulation-to-reality gap

Worked once outside



Simulation

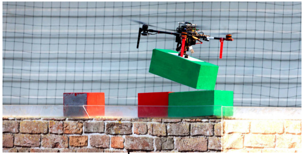
Reality



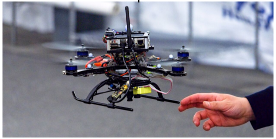
Worked once in a lab

Simulation

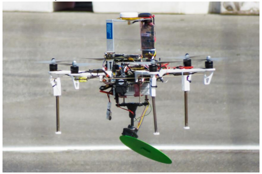
Worked once outside



Reality

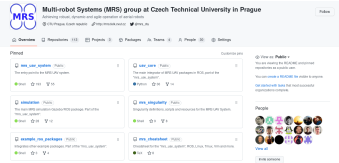


Worked once in a lab

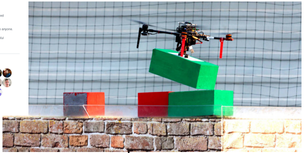


Repeatable on-demand deployment

Replicable research

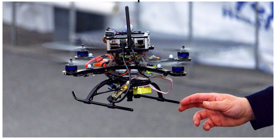


Worked once outside

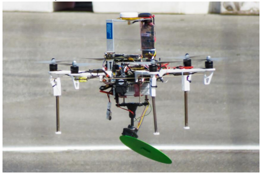


Simulation

Reality



Worked once in a lab



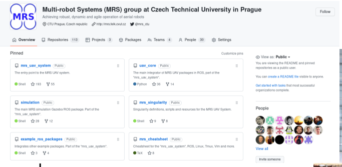
Repeatable on-demand deployment

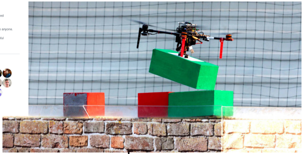
FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

MRS MULTI-ROBOT SYSTEMS GROUP

Replicable research

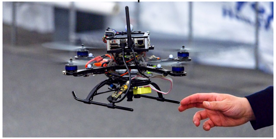
Worked once outside



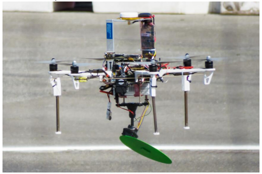


Simulation

Reality



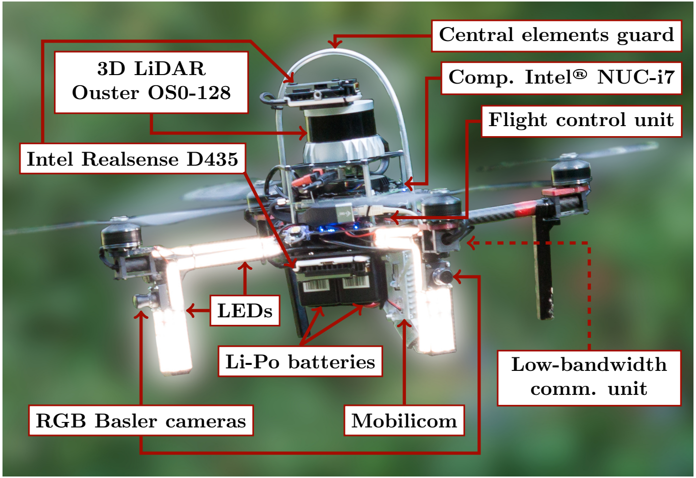
Worked once in a lab



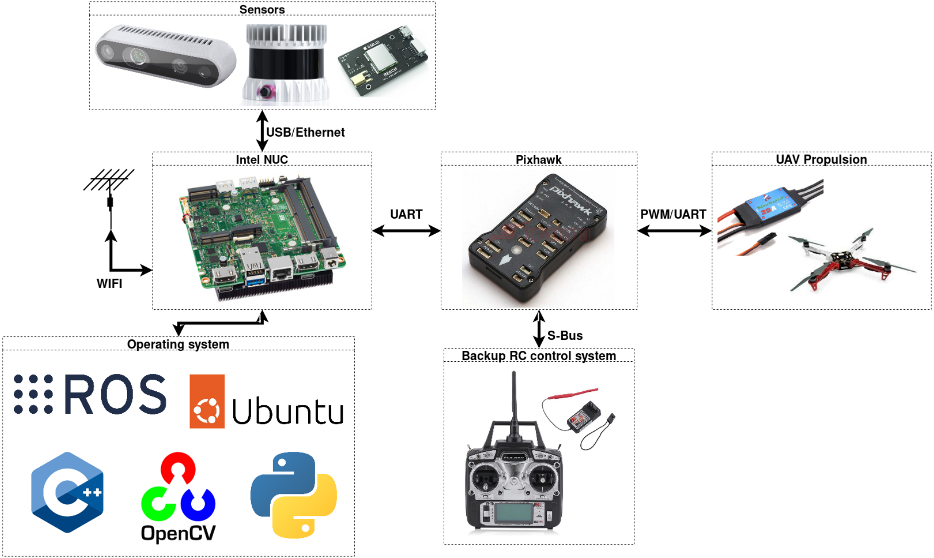
Repeatable on-demand deployment

Replicable real-world on-demand Multi-UAV deployment

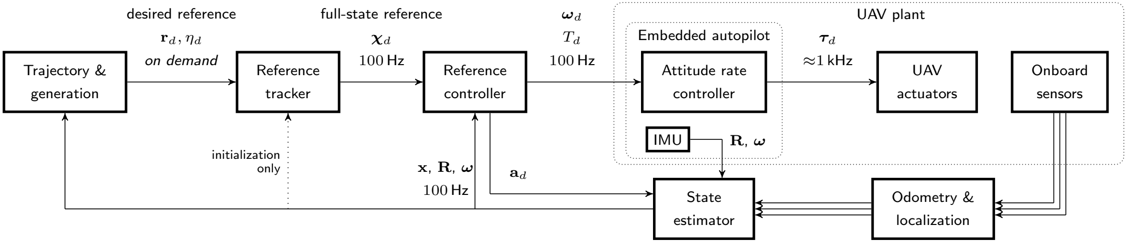
A typical pipeline structure revolves around a Linux computer



Research drone (flying robot) hardware diagram

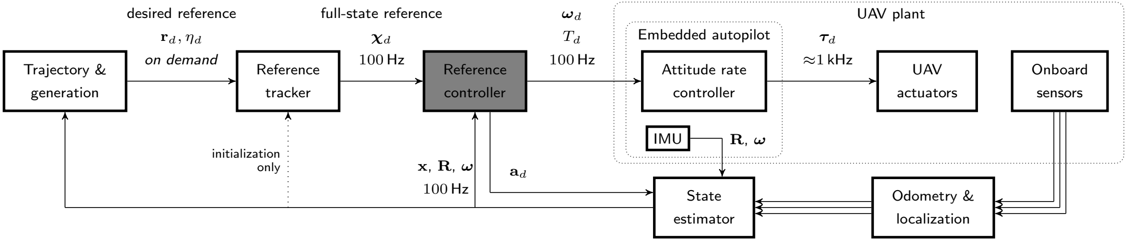


MRS UAV System block diagram



[1] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021

MRS UAV System block diagram



[1] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE | MRS MULTI-ROBOT SYSTEMS GROUP

Heading-compliant control design

Heading $\eta = \mathrm{atan2}\left(\hat{\mathbf{b}}_1^\intercal \hat{\mathbf{e}}_2, \hat{\mathbf{b}}_1^\intercal \hat{\mathbf{e}}_1\right) = \mathrm{atan2}\left(\mathbf{h}_{(2)}, \mathbf{h}_{(1)}\right)$ as the 4^{th} DOF (instead of *yaw*).

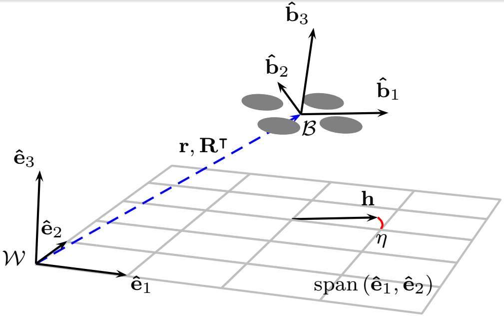
SO(3) attitude controller [2]

- Custom heading-compliant desired orientation:

$$\mathbf{R}_d = \left[\hat{\mathbf{b}}_{1d}, \hat{\mathbf{b}}_{2d}, \hat{\mathbf{b}}_{3d}\right], \tag{1}$$

$$\mathbf{b}_{1d} = \mathbf{O}(\mathbf{P}^\intercal\mathbf{O})^{-1}\mathbf{P}^\intercal\,\hat{\mathbf{h}}_d,\ \hat{\mathbf{b}}_{1d} = \frac{\mathbf{b}_{1d}}{\|\mathbf{b}_{1d}\|}, \tag{2}$$

$$\hat{\mathbf{b}}_{2d} = \hat{\mathbf{b}}_{3d} \times \hat{\mathbf{b}}_{1d}. \tag{3}$$



[1] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE | MRS MULTI-ROBOT SYSTEMS GROUP

Geometric tracking on *SE(3)* [2]

$$\mathbf{f}_d = \overbrace{-m_e \mathbf{k}_p \circ \mathbf{e}_p}^{\text{position feedback}} + \overbrace{-m_e \mathbf{k}_v \circ \mathbf{e}_v}^{\text{velocity feedback}} + \overbrace{m_e \ddot{\mathbf{r}}_d}^{\text{reference feedforward}} + \underbrace{m_e g \hat{\mathbf{e}}_3}_{\text{gravity compensation}} + \underbrace{-\mathbf{d}_w \circ \begin{bmatrix}1\\1\\0\end{bmatrix}}_{\text{world disturbance compensation}} + \underbrace{-\mathbf{d}_b \circ \begin{bmatrix}1\\1\\0\end{bmatrix}}_{\text{body disturbance compensation}}, \tag{4}$$

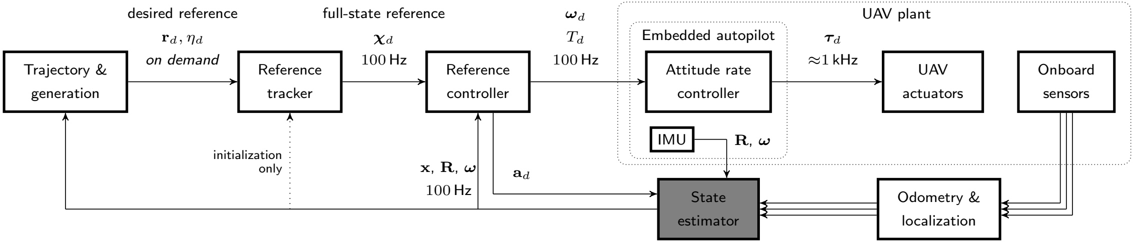
Linear MPC

$$\mathbf{f}_d = \overbrace{m_e \ddot{\mathbf{r}}_d}^{\text{reference feedforward}} + \overbrace{m_e \mathbf{c}_d}^{\text{MPC feedforward}} + \overbrace{m_e g \hat{\mathbf{e}}_3}^{\text{gravity compensation}} + \underbrace{-\mathbf{d}_w \circ \begin{bmatrix}1\\1\\0\end{bmatrix}}_{\text{world disturbance compensation}} + \underbrace{-\mathbf{d}_b \circ \begin{bmatrix}1\\1\\0\end{bmatrix}}_{\text{body disturbance compensation}}. \tag{5}$$

MPC feedforward: desired acceleration from a constrained linear MPC.

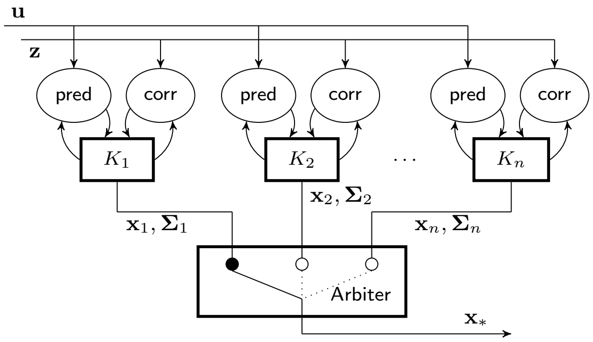
[1] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, *et al.*, “The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles,” *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021

MRS UAV System block diagram



[1] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021

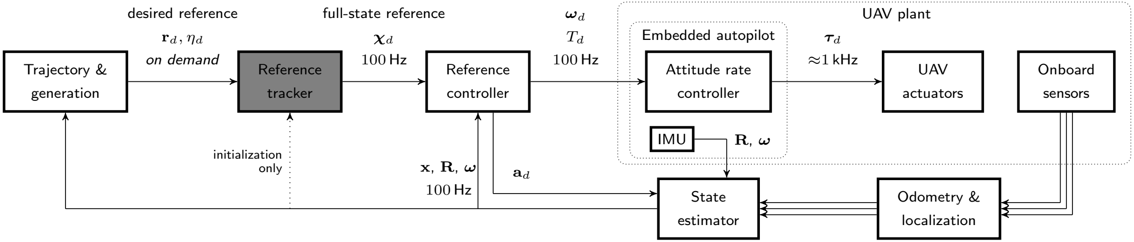
Bank of filters



- simultaneous estimation of UAV state in multiple frames of reference
- automatic & manual switching of the main estimator
- automatic detection of estimator failures
- switching of an estimator is synchronizes throughout the control pipeline

[3] M. Petrlik, T. Baca, D. Hert, M. Vrba, T. Krajnik, and M. Saska, "A Robust UAV System for Operations in a Constrained Environment," *IEEE Robotics and Automation Letters*, vol. 5, 2 Apr. 2020

MRS UAV System block diagram



[1] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021

Feedback and Feed-forward multicopter controllers benefit from a smooth and feasible control reference. A step in desired position or velocity is not a feasible reference.

Problem

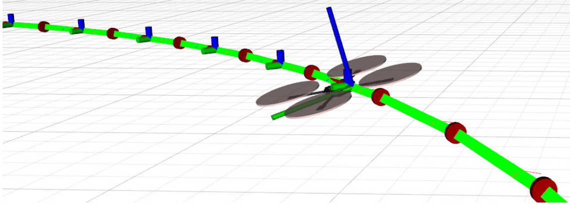
- real-time reference generation
- 100 Hz full state reference for controllers: $\dot{\mathbf{x}}, \ddot{\mathbf{x}}, \dddot{\mathbf{x}}, \ddddot{\mathbf{x}}$, where $\mathbf{x} = \left(r_x, r_y, r_z, \eta\right)^\intercal$ (pose and heading)
- UAV dynamics constraints satisfaction
- user input: trajectory consisting of poses sampled at regular intervals

Solution

- onboard real-time simulation of the linear translational UAV dynamics
- linear MPC (LCQP) solved at 100 Hz
- states of the simulated model sampled and taken as control reference
- 8 s prediction horizon used for inter-UAV collision avoidance

[4] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model Predictive Trajectory Tracking and Collision Avoidance for Reliable Outdoor Deployment of Unmanned Aerial Vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2018, pp. 6753–6760

Model Predictive Control Tracker



$$\min_{\mathbf{u}_{[1:n]}} \quad \mathrm{V}\left(\mathbf{x}_{[1:n]}, \mathbf{u}_{[1:n]}\right) = \frac{1}{2}\sum_{i=1}^{n}\left(\mathbf{e}_{[i]}^{T}\mathbf{Q}\mathbf{e}_{[i]} + \mathbf{u}_{[i]}^{T}\mathbf{P}\mathbf{u}_{[i]}\right)$$

$$\begin{aligned}
\text{s.t.}\ \mathbf{x}_{[t]} &= \mathbf{A}\mathbf{x}_{[t-1]} + \mathbf{B}\mathbf{u}_{[t]}, && \forall t \in \{0, \ldots, n\} \\
\mathbf{x}_{[t]} &\leq \mathbf{x}_{\max[t]}, && \forall t \in \{1, \ldots, n\} \\
\mathbf{x}_{[t]} &\geq \mathbf{x}_{\min[t]}, && \forall t \in \{1, \ldots, n\}
\end{aligned}$$

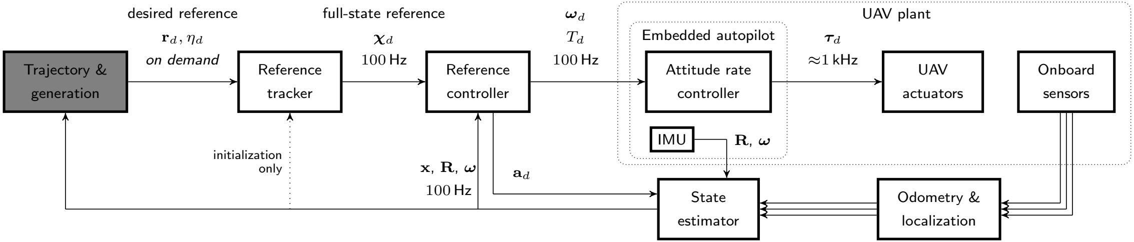
Properties

- Linear MPC is solved in real time
- Mutual collision avoidance prevent damage during experimentation
- The LCQP can be solved reliably
- The tracker can handle unfeasible trajectory references from a user

[4] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model Predictive Trajectory Tracking and Collision Avoidance for Reliable Outdoor Deployment of Unmanned Aerial Vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2018, pp. 6753–6760

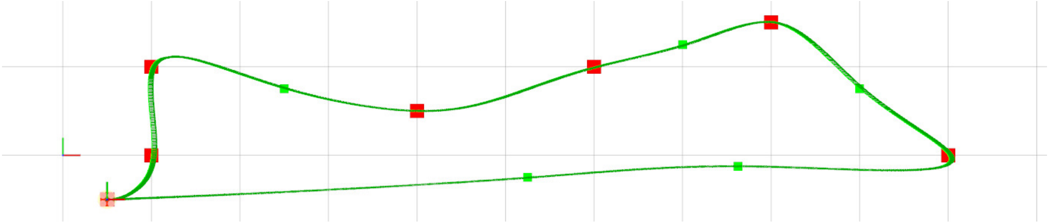
FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE
MRS MULTI-ROBOT SYSTEMS GROUP

MRS UAV System block diagram



[1] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021

Path → Feasible Trajectory



Fork of ethz-asl/mav_trajectory_generation

[5] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research*, Springer, 2016, pp. 649–666

[6] M. Burri, H. Oleynikova, M. W. Achtelik, and R. Siegwart, "Real-Time Visual-Inertial Mapping, Re-localization and Planning Onboard MAVs in Unknown Environments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015

- fixed poorly-implemented lower bound segment time initialization
- fixed poorly-implemented trajectory sampling
- recursive segment sub-sectioning for meeting desired corridor constraints
- + smooth continuation of the prior UAV motion (Requires the *MPC Tracker*)
- + Fallback solution if the `QP` optimization fails
- + asynchronous execution and timeouting (with fallback solution)
- + result sanitization

[1] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021

ROS – Robot Operating System

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

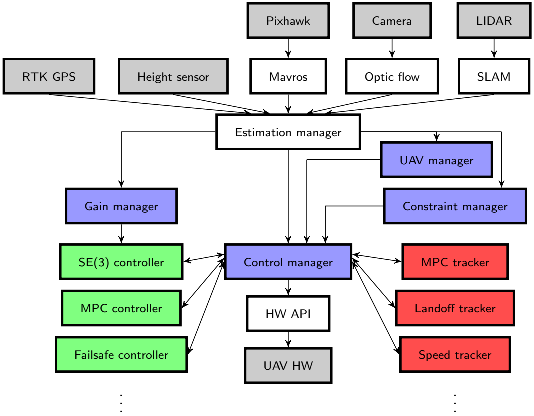
MRS MULTI-ROBOT SYSTEMS GROUP

- middleware allowing communication between programs
- integrates with `C++`, Python, Bash and Zshell
- makes the transition from *theory* to *bare metal* bearable
- universally supported by sensor manufacturers
- used globally by the research community
- integration through the Linux terminal
- out of the box: time and clock management, logging, recording onboard data, visualization and plotting, parameter loading, static and dynamic transformations, etc.
- integrates to robotic simulators: Gazebo, Coppelia (V-REP), AirSim

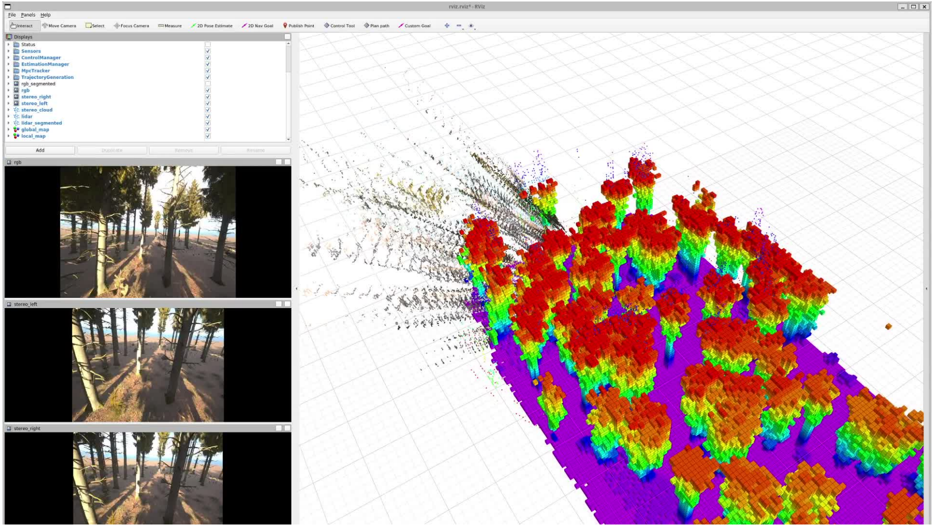
ROS.org

- Modularity promotes collaborative development
- Modularity leads to separation of concerns
- Allows component substitution and mocking
- Makes building abstract interfaces easier
- Allows for distributed execution across more machines
- Makes introspection easier
- makes runtime performance worse
- proper data exchange is major part of the system design

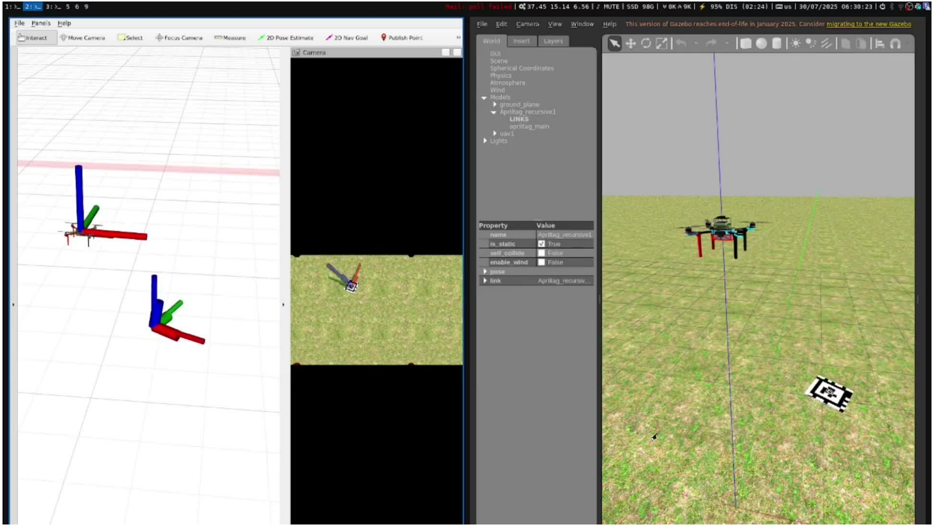
Modular system architecture



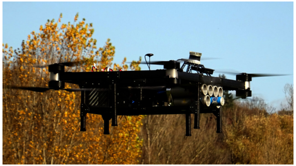
Realtime Mapping & Planning pipeline using LiDAR, RGBD



Precise landing on an April tag



Our flying robots

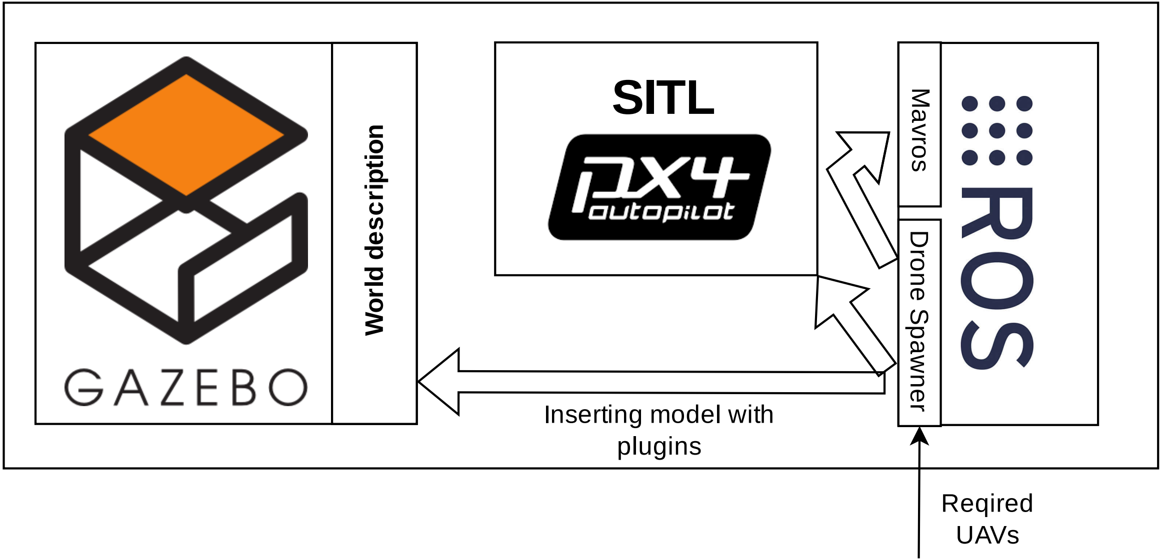


Gazebo/ROS simulation pipeline

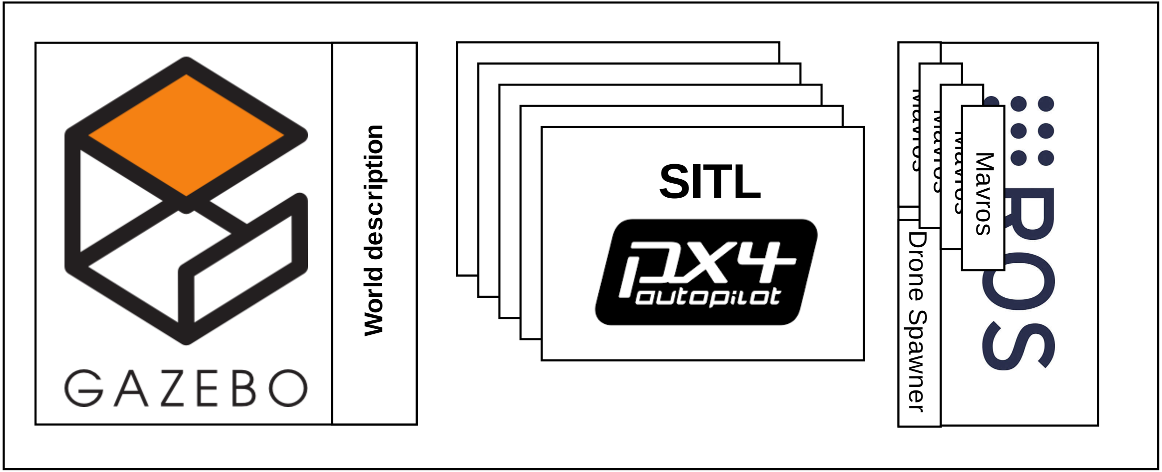
GAZEBO

GAZEBO

- scalable for any UAV configuration



- scalable for any UAV configuration



1. Definition of universal payloads

- Payload models with plugins defined in a separate file.

```
component_snippets.sdf.jinja                                                                 buffers

  <!-- ================================================================== -->
  <!-- ||  generic sensor plugin definitions (camera, rangefinder ...) || -->
  <!-- ================================================================== -->

+--483 lines: -----------------------------------------------------------------------------------------

<!-- ================================================================== -->
<!-- ||                      sensor definitions                      || -->
<!-- ================================================================== -->

<!-- ====================== rangefinder sensors ======================-->

+-- 58 lines: Macro to add a garmin (connected to the pixhawk) ------------------------------------

+-- 47 lines: Macro to add an external garmin (not connected to the pixhawk) ----------------------

+-- 44 lines: Macro to add a Teraranger One --------------------------------------------------------

+-- 45 lines: Macro to add a URM37 ultrasound ------------------------------------------------------

<!-- ======================== LIDAR sensors ======================== -->

+-- 77 lines: Macro to add a Scanse Sweeper---------------------------------------------------------

+-- 78 lines: Macro to add a RPlidar A3-------------------------------------------------------------

+--139 lines: Macro to add a Velodyne Lidar --------------------------------------------------------

+--290 lines: Macro to add an Ouster Lidar ---------------------------------------------------------

<!-- ======================= camera sensors ======================== -->

+--201 lines: Macro to add an Intel Realsense D435 -------------------------------------------------

+-- 34 lines: Macro to add a Bluefox camera --------------------------------------------------------

+-- 43 lines: Macro to add a Fisheye camera --------------------------------------------------------

+-- 34 lines: Macro to add a Mobius camera ---------------------------------------------------------

+-- 70 lines: Macro to add a thermal camera --------------------------------------------------------

+-- 40 lines: Macro to add a UV Camera -------------------------------------------------------------

+--155 lines: Macro to add a camera mounted on virtual servo ---------------------------------------

+-- 33 lines: Macro to add a fisheye camera with integrated imu -----------------------------------
```

1. Definition of universal payloads

- Payload models with plugins defined in a separate file.

2. Definition of platform-specific mounting points

- Arbitrary placement and configuration of particular payload for particular platform.

```
f450.sdf.jinja                                                              buffers

  <!-- ================================================================ -->
  <!-- ||                    optional sensor definitions              || -->
  <!-- ================================================================ -->

+-- 20 lines: {# Ground truth #}------------------------------------------------

  <!-- ===================== rangefinder sensors ======================-->

+-- 18 lines: {# Garmin #}------------------------------------------------------

+-- 17 lines: {# Garmin Up #}---------------------------------------------------

+-- 16 lines: {# Teraranger One #}----------------------------------------------

  <!-- ========================= LIDAR sensors ======================== -->

+-- 16 lines: {# Scanse Sweep #}------------------------------------------------

+-- 16 lines: {# Rplidar #}-----------------------------------------------------

+-- 24 lines: {# Velodyne #}----------------------------------------------------

+-- 21 lines: {# Ouster #}------------------------------------------------------

  <!-- ========================= camera sensors ======================= -->

+-- 21 lines: Fisheye camera placements-----------------------------------------

+-- 41 lines: Bluefox camera placements-----------------------------------------

+-- 75 lines: Realsense placements ----------------------------------------------

+-- 70 lines: Mobius placements ------------------------------------------------

+-- 35 lines: {# Dual UV cameras #}---------------------------------------------

+-- 20 lines: {# Back UV cameras #}---------------------------------------------

+-- 40 lines: {# Servo camera #}------------------------------------------------

+-- 42 lines: VIO placements --------------------------------------------------

  <!-- ========================= other sensors ======================== -->

+-- 19 lines: {# Teraranger Tower Evo #}----------------------------------------

+-- 15 lines: {# Magnetic gripper #}--------------------------------------------

+-- 16 lines: {# Timepix #}-----------------------------------------------------
```

1. Definition of universal payloads

- Payload models with plugins defined in a separate file.

2. Definition of platform-specific mounting points

- Arbitrary placement and configuration of particular payload for particular platform.

3. Dynamic assembly of the final UAV sdf structure

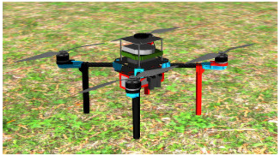
- The robot description file is assembled based on user's query.
- Any-to-any mapping between UAV frames and UAV payloads
- Queries are queued and executed in series.

Bonus facts

- Potential definition of sensor groups.
- Support for query parameters — additional customization.

Spawning query

```
rosservice call /mrs_drone_spawner/spawn "1
--x500 --pos 0 0 1 0 --enable-rangefinder
--enable-realsense-front --enable-ouster
model:=OS0-128 use_gpu:=True"
```

Model	Spawn argument	Simulation
DJI f330	`--f330`	
DJI f450	`--f450`	
Holybro x500	`--x500`	
DJI f550	`--f550`	
Tarot t650	`--t650`	
NAKI II	`--naki`	

- 7 UAV types: DJI f330, DJI f450, DJI f550, T-Motor x500, Tarot 650, T-drone m690, Fly4Future's RoboFly

Available sensors

- RGB camera: MatrixVision Bluefox, Mobius
- Stereo Cameras: Intel Realsense
- 1D LiDARs: Terabee Teraranger, Gamin Lite
- 2D LiDARs: Scanse Sweep, Garmin RPLidar
- 3D LiDARs: Velodyne Puck, Ouster
- Thermal cameras, UV cameras for UVDAR

Available actuation

- magnetic gripper
- parachute
- water gun

http://github.com/ctu-mrs/mrs_uav_gazebo_simulation

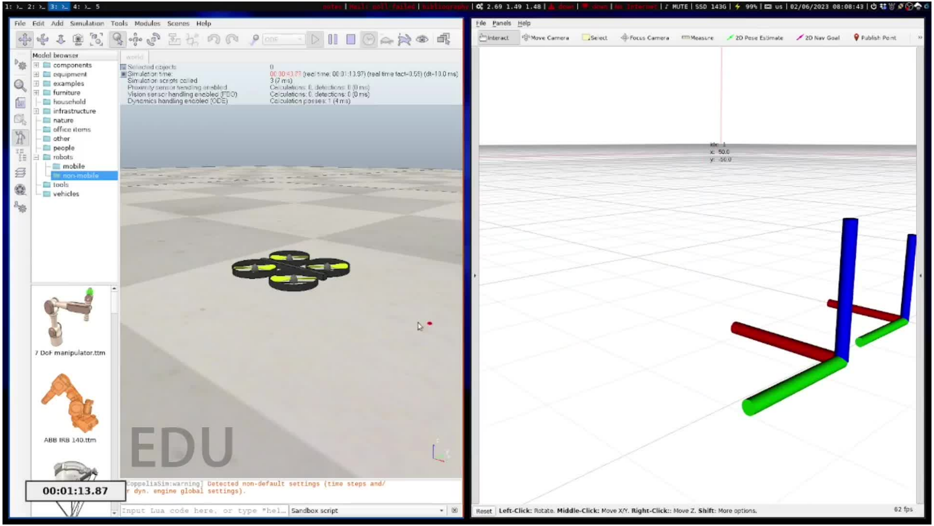
Is one simulator enough?

Large-scale swarms — require high parallelization within the simulator

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE
MRS MULTI-ROBOT SYSTEMS GROUP

High-fidelity visuals — requires capable rendering engine

CoppeliaSim — Ease of use and deployment



Is one simulator enough? No

Each simulator has pros and cons:

Is one simulator enough? No

Each simulator has pros and cons:

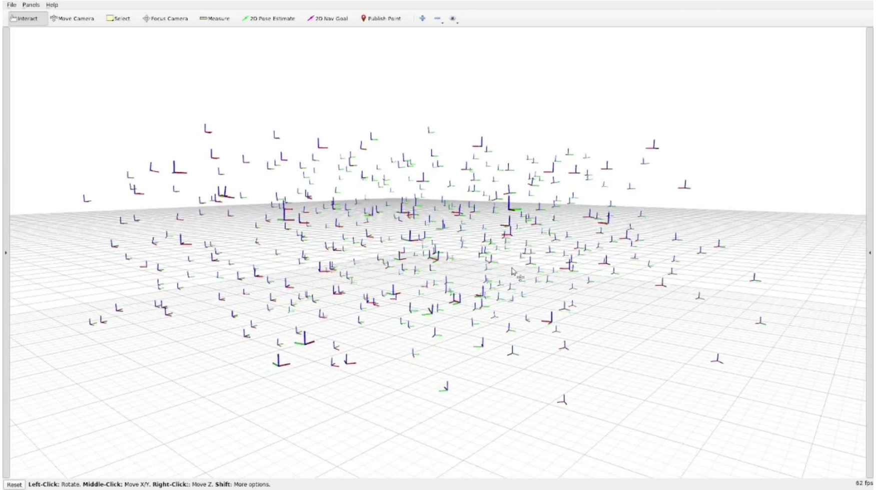
Is one simulator enough? No

Each simulator has pros and cons:

Is one simulator enough? No

Each simulator has pros and cons:

The MRS swarm simulator



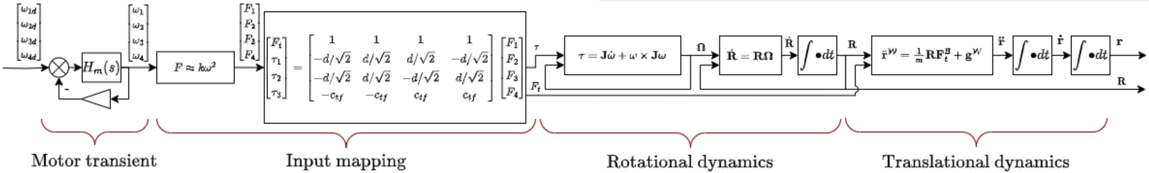
Video: `https://youtu.be/2UJ7aYaHOX0`

MRS multirotor simulator

- Full multirotor dynamics
- Embedded feedback controllers
- Fast C++ ODE solver
- **header-only library — intended for RL**
- available ROS integration
- minimum external dependencies
- tightly integrated into the MRS system's core

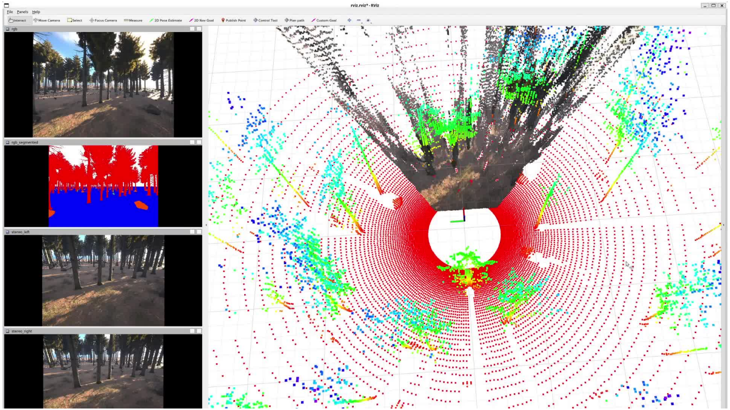
Available control inputs

- Position & heading
- Velocity & heading
- Velocity & heading rate
- Acceleration & heading
- Acceleration & heading rate
- Attitude & throttle
- Attitude rate & throttle
- Control group & throttle
- Individual actuators' throttle



FlightForge simulator (The MRS Multirotor + Unreal Engine 5)

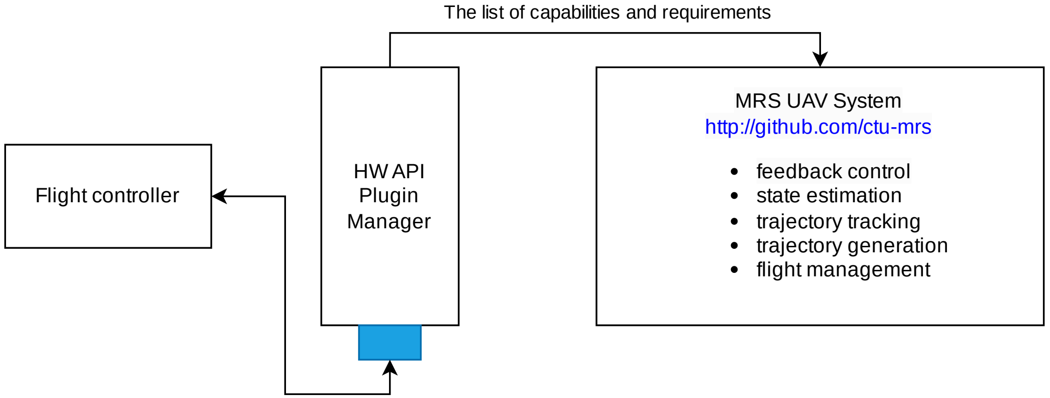
- RGB camera
- Stereo camera
- 3D LiDAR
- Ground truth segmentation in LiDAR and RGB
- Step-locked with our dynamics simulator (ROS)



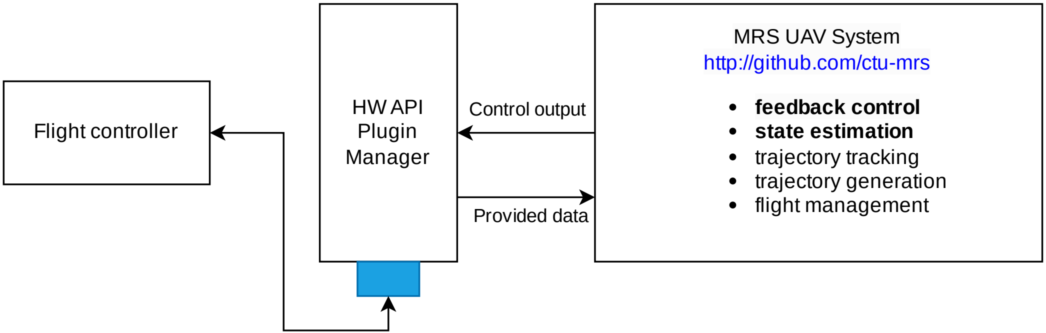
`https://github.com/ctu-mrs/mrs_uav_flightforge_simulator`

D. Capek, J. Hrncir, T. Baca, *et al.*, "FlightForge: Advancing UAV Research with Procedural Generation of High-Fidelity Simulation and Integrated Autonomy," in *IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, 2025

Interfacing with any simulator or flight controller



Interfacing with any simulator or flight controller



Information provided

Subset of the following:

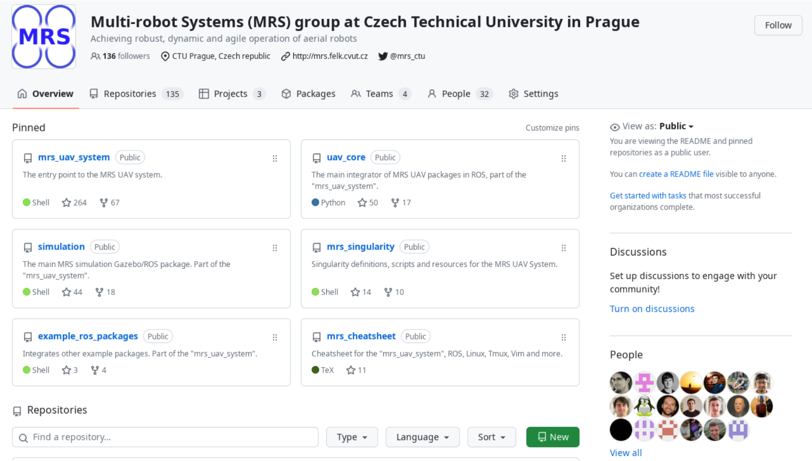
- IMU
- GNSS
- RTK GNSS
- Magnetometer
- AMSL measurement
- Ground truth pose
- Height measurement
- Angular rate
- RC channels
- Velocity
- Orientation
- 3D Pose

Control input accepted

Subset of the following:

- Position & heading
- Velocity & heading
- Velocity & heading rate
- Acceleration & heading
- Acceleration & heading rate
- Attitude & throttle
- Attitude rate & throttle
- Control group & throttle
- Individual actuators' throttle

MRS UAV System — all open-source



http://github.com/ctu-mrs

Native installation over ROS Noetic

```
curl https://ctu-mrs.github.io/ppa-stable/add_ppa.sh | bash
sudo apt install ros-noetic-mrs-uav-system-full
```

Apptainer container system + container wrapper

http://github.com/ctu-mrs/mrs_apptainer

Docker containers

http://github.com/ctu-mrs/mrs_docker

Sources

http://github.com/ctu-mrs/mrs_uav_system

Documentation

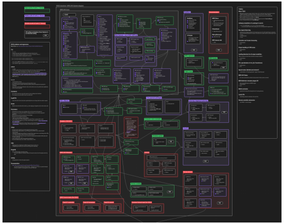
https://ctu-mrs.github.io/docs/1.5.0/introduction/

Transition to ROS2 - ROS2 Jazzy (Ubuntu 24.04)

Problematic at best

- realtime issues with DDSs ($\Longrightarrow$ Zenoh (2024))
- buggy client libraries (executors, timers)
- design regressions all over the board
- performance significantly worse than ROS1

State of transition



`https://ctu-mrs.github.io/ros2_obsidian_knowledgebase/main_canvas.html`

Available in ROS2

- MRS UAV Core
- PX4 HW API
- MRS Multirotor sim
- FlightForge sim
- some sensor drivers

To be done

- MRS Gazebo stack
- GNSS-denied flight (LiDAR, VIO)
- Octomap Mapping & Planning
- Precise landing
- Pointcloud utils
- Lot of QoL utilities
- **Documentation**

Native installation over ROS2 Jazzy

```
curl https://ctu-mrs.github.io/ppa2-stable/add_ppa.sh | bash
sudo apt install ros-jazzy-mrs-uav-system-full
```

Sources

http://github.com/ctu-mrs/mrs_uav_system/tree/ros2

Documentation

https://ctu-mrs.github.io/docs/2.0.0/introduction/

Thanks for your attention

Download pdf of this presentation

`https://github.com/ctu-mrs/presentation_mrs_uav_system`

My profile

`http://mrs.felk.cvut.cz/people/tomas-baca`

Cite the MRS UAV System if you use it for your research

T. Baca, M. Petrlik, M. Vrba, *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021

References I

[1] T. Baca, M. Petrlik, M. Vrba, *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021.

[2] T. Lee, M. Leoky, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *2010 IEEE Conference on Decision and Control*, IEEE, 2010, pp. 5420–5425.

[3] M. Petrlik, T. Baca, D. Hert, M. Vrba, T. Krajnik, and M. Saska, "A Robust UAV System for Operations in a Constrained Environment," *IEEE Robotics and Automation Letters*, vol. 5, 2 Apr. 2020, ISSN: 2169-2176.

[4] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model Predictive Trajectory Tracking and Collision Avoidance for Reliable Outdoor Deployment of Unmanned Aerial Vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2018, pp. 6753–6760.

[5] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research*, Springer, 2016, pp. 649–666.

[6] M. Burri, H. Oleynikova, M. W. Achtelik, and R. Siegwart, "Real-Time Visual-Inertial Mapping, Re-localization and Planning Onboard MAVs in Unknown Environments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[7] D. Capek, J. Hrncir, T. Baca, *et al.*, "FlightForge: Advancing UAV Research with Procedural Generation of High-Fidelity Simulation and Integrated Autonomy," in *IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, 2025.