A Framework for Modeling Underwater Vehicles in Modelica

SHASHANK SWAMINATHAN F.W. OLIN COLLEGE OF ENGINEERING

DR. SRIKANTH SARIPALLI TEXAS A&M UNIVERSITY

Presentation Outline

Introduction and Overview

Modeling Framework: Underwater Rigid Body Library (URBL)

Applying the URBL

Testing the Applied Model

Results

Conclusion

Connection to Robot Operating System (ROS)

Framework Goals

1

Aid the prototyping and testing of vehicle design and controls.

2

Be readily integrated with common control and feedback mechanisms, specifically ROS.

3

Visualize prototype design and test results via three-dimensional animation.

Framework Language

The language used by the modeling tool is **Modelica**.

- This is a **non-proprietary**, **domain-specific**, **object-oriented modeling language**
- In use by industry since 2000

Meant for modeling the dynamic behavior of technical systems via a **convenient, component-based approach**

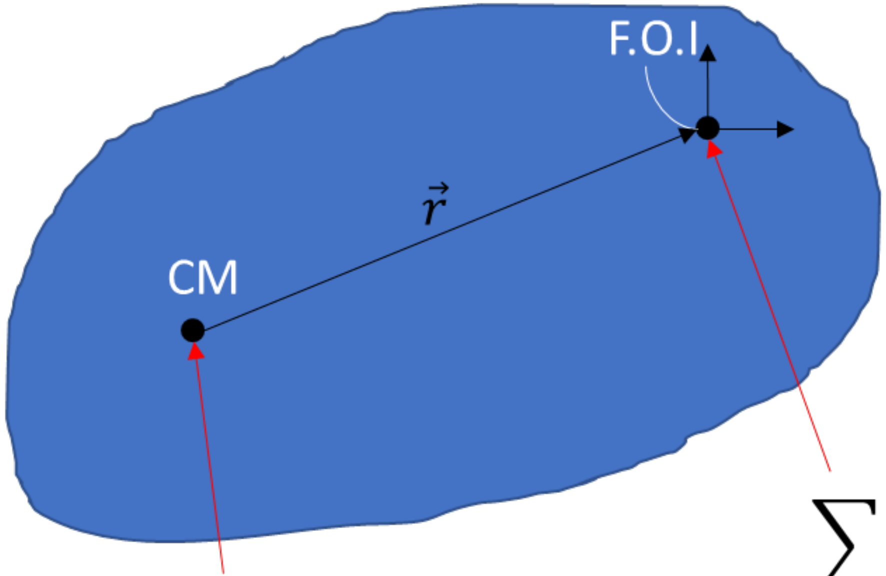
- Tools utilize graphical and textual editors for modeling

Models are described via discrete, algebraic, and/or differential equations;

- Modelica allows the user to **abstract out the modeling process** to just the equations depicting the physical system's nature
- No PDE or FEM representation, but can utilize results from third-party tools

Follows the **FMI Standard**

- Can import or export **FMUs** for co-simulation or model exchange



$$\sum \vec{F}_{CM} = \vec{F}_{buoyant} + \vec{F}_{drag}$$

$$\sum \vec{F}_{FOI} = \sum \vec{F}_{CM}$$

$$\sum \vec{\tau}_{FOI} = \vec{r} \times \sum \vec{F}_{CM}$$

URBL Field Model

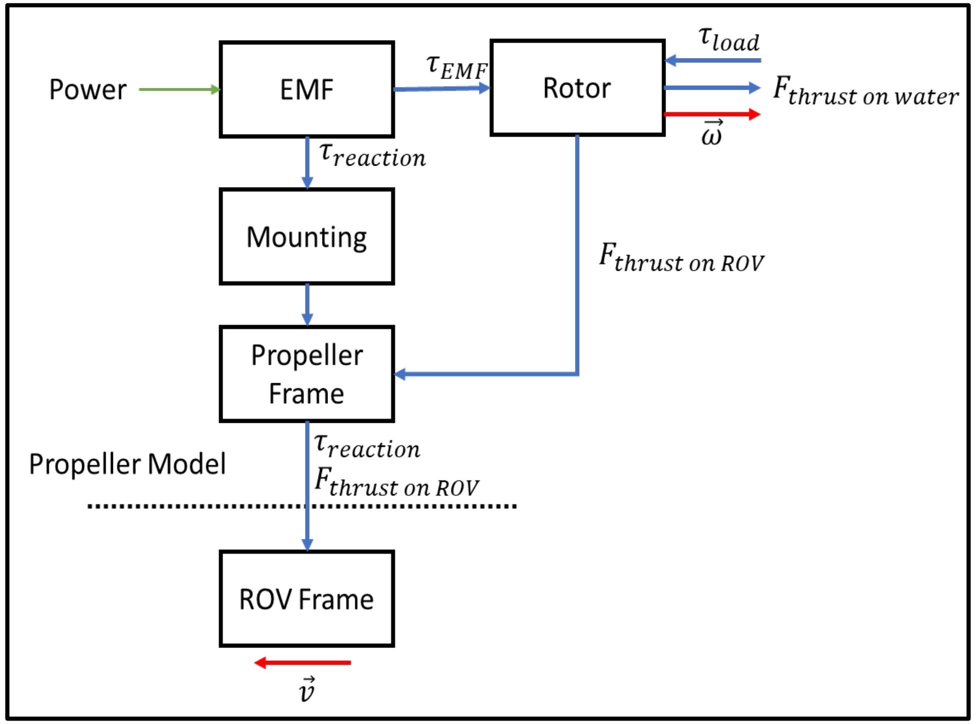
DIAGRAM OF FIELD INTERACTIONS WITH RIGID BODY

URBL Propeller Template

Motivated from the designs of other underwater vehicles modeled in other papers, as well as that of the BlueROV2

Creates a template for propeller use:

- Has dynamics detailing thrust on body, and load torque from water
- Has inertial mass component for interacting with ROV
- Can be implemented along any axis of motion
- Depends on EMF for actuation – source of voltage to the propeller is not provided



Propeller Dynamics - Thrust

Thrust can be written as follows:

- $F_{thrust} \propto \omega^2 K_T(J^*)$
- Where ω is the rotor's angular velocity, and $K_T(J^*) = \beta_1 - \beta_2 J^*$
  - $J^* = \frac{v}{\omega}$
- This means that F_{thrust} can be rewritten as $F_{thrust} \propto \omega^2 \left(\beta_1 - \beta_2 \frac{v}{\omega}\right)$

Expanding and regrouping constants gives

- $\vec{F}_{thrust} = -k_m|\omega|(k_r\vec{\omega} - \hat{\omega} \cdot \vec{v})\, b_{dir}$
  - $\vec{v}$ is the relative velocity between the ROV and the water
  - k_m, k_r are appropriate constants of proportionality derived from β_1 and β_2
  - b_{dir} indicates the direction of the propeller's mounting

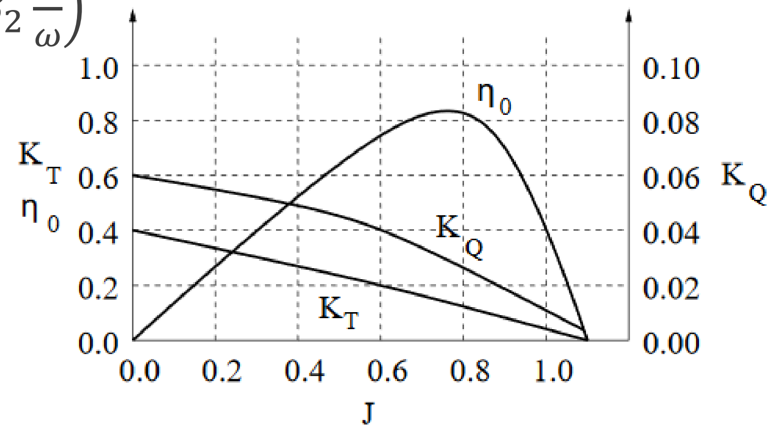


Figure 4: Typical thrust and torque coefficients.

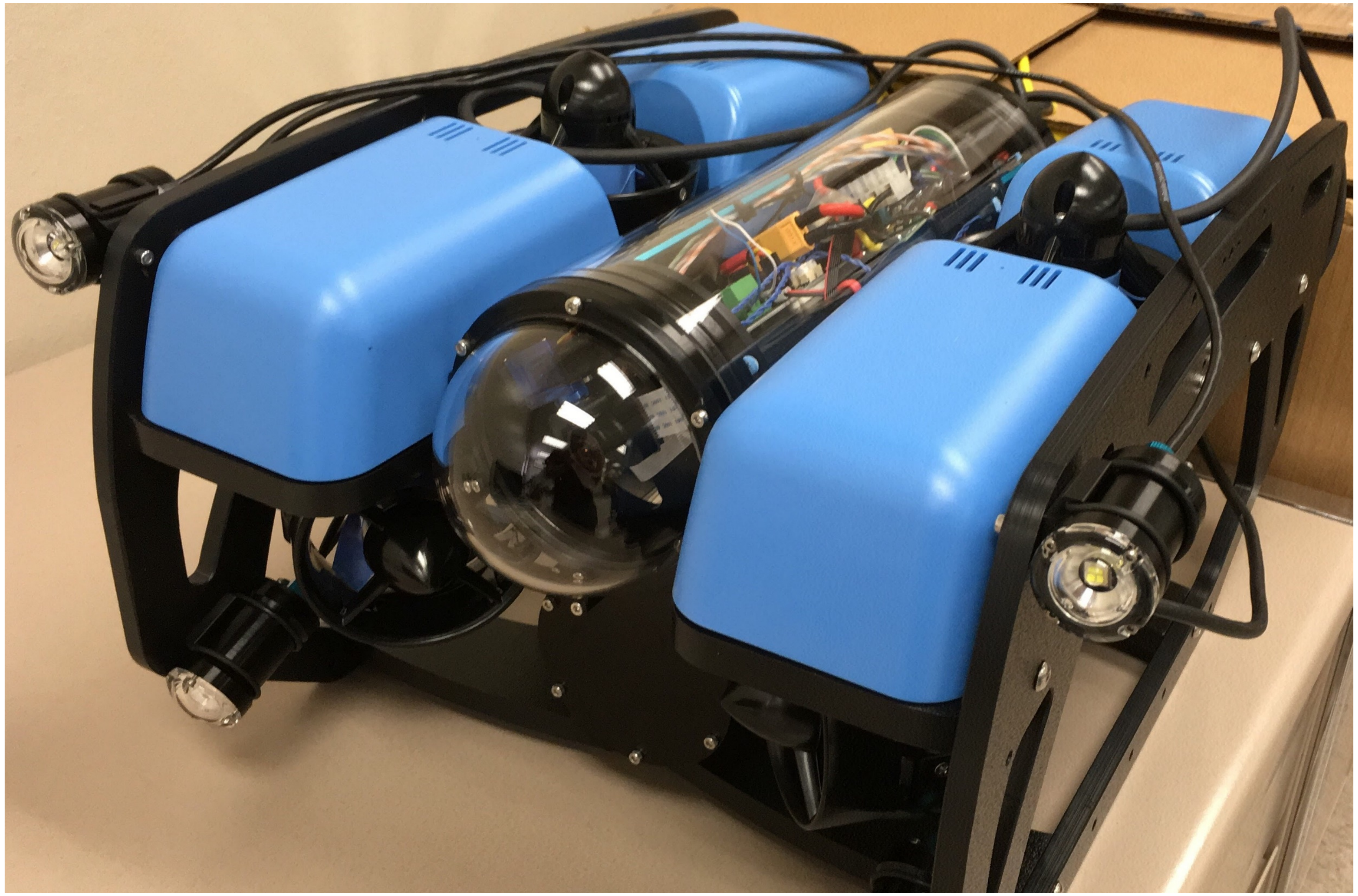
Propeller Dynamics – Load Torque

The load torque on the propeller due to moving water is found via power balance:

- $\vec{\tau}_{load} = -\frac{\vec{F}_{thrust}\,\widehat{\omega}\cdot\vec{v}}{|\vec{\omega}|\eta}$
  - $\vec{F}_{thrust}$ is thrust
  - $\vec{v}$ is the relative velocity between the ROV and the water
  - $\vec{\omega}$ is the rotor's angular velocity
  - η is the efficiency of the power balance

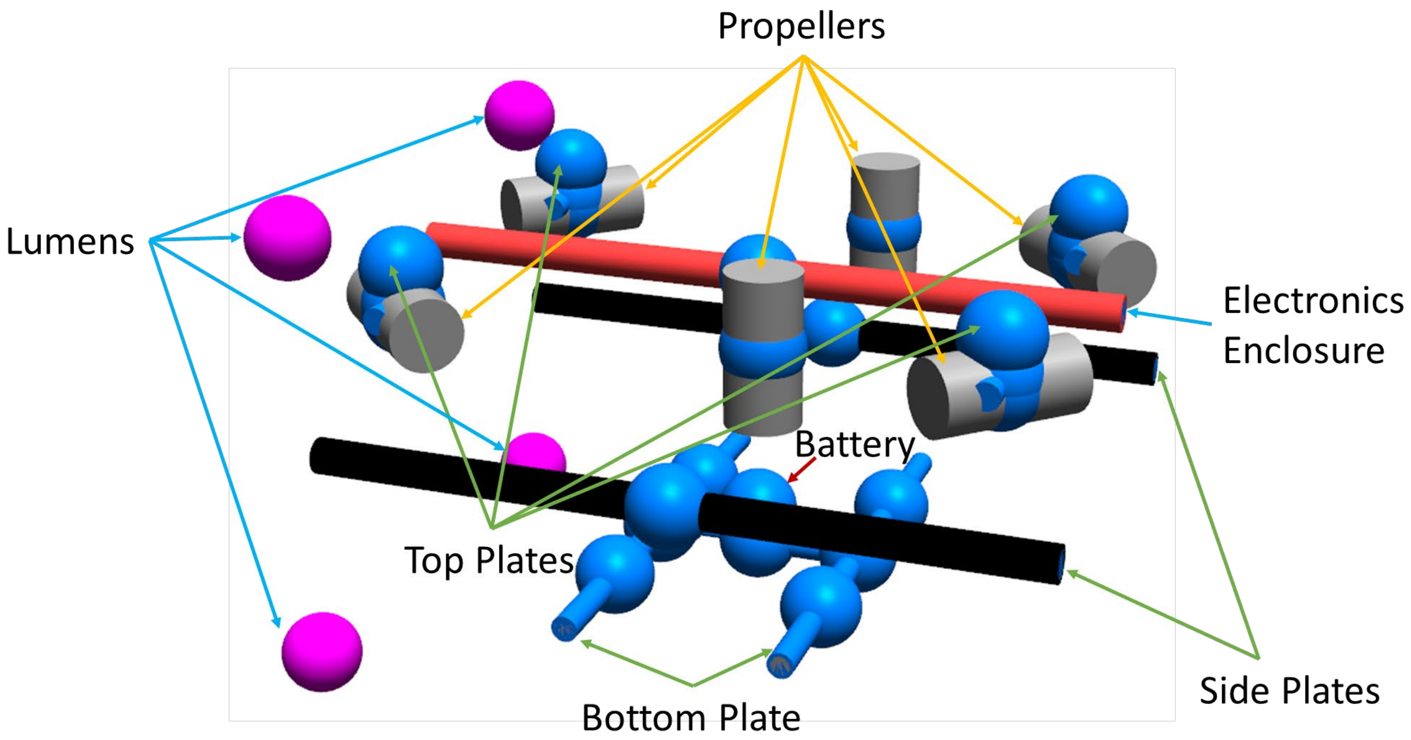
To better handle when $\vec{\omega}$ goes to zero, this equation is rewritten by expanding the thrust term:

- $\vec{\tau}_{load} = -k_m\vec{v}(k_r\vec{\omega} - \widehat{\omega}\cdot\vec{v})\, b_{dir} - k_{loss}\vec{\omega}$
- $k_{loss}\vec{\omega}$ is added to represent loss purely due to the rotor's motion.



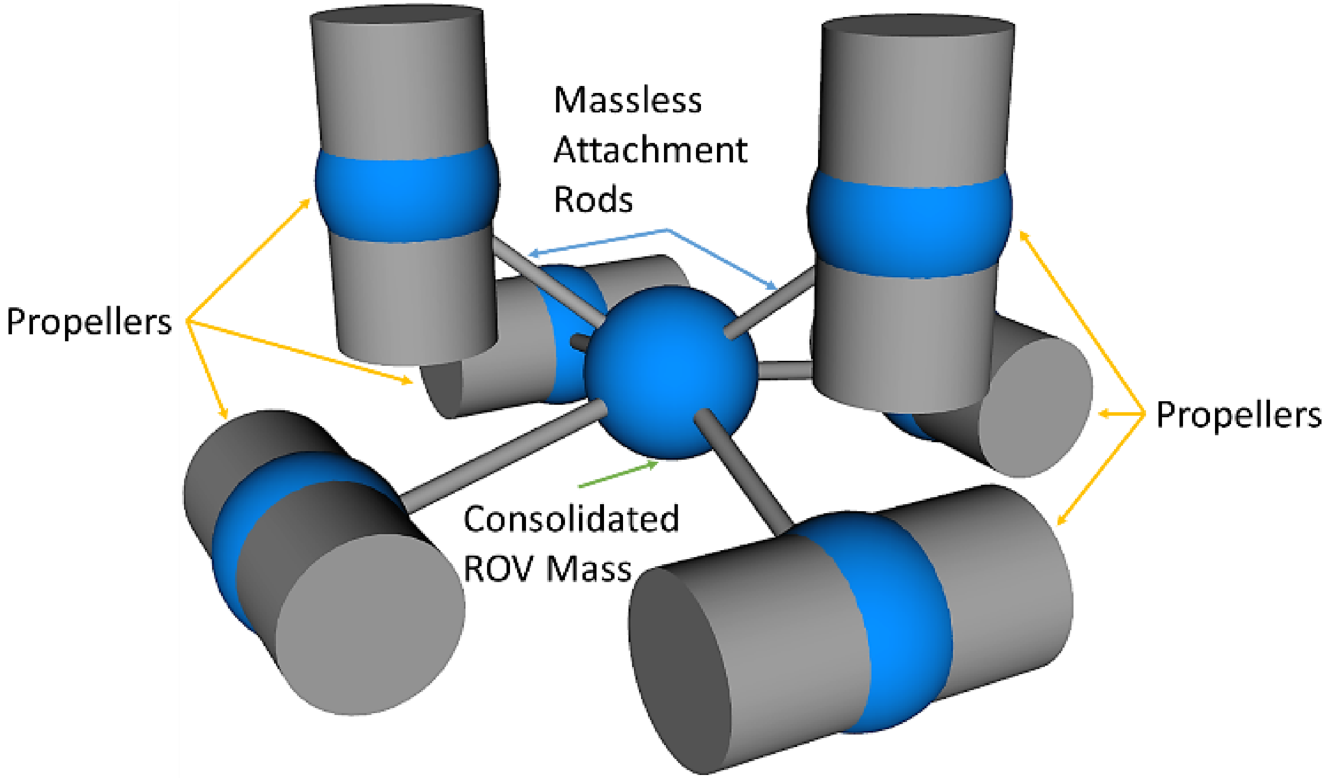
Application of the URBL

The framework is used to model a commercially available ROV, the BlueROV2 from Blue Robotics



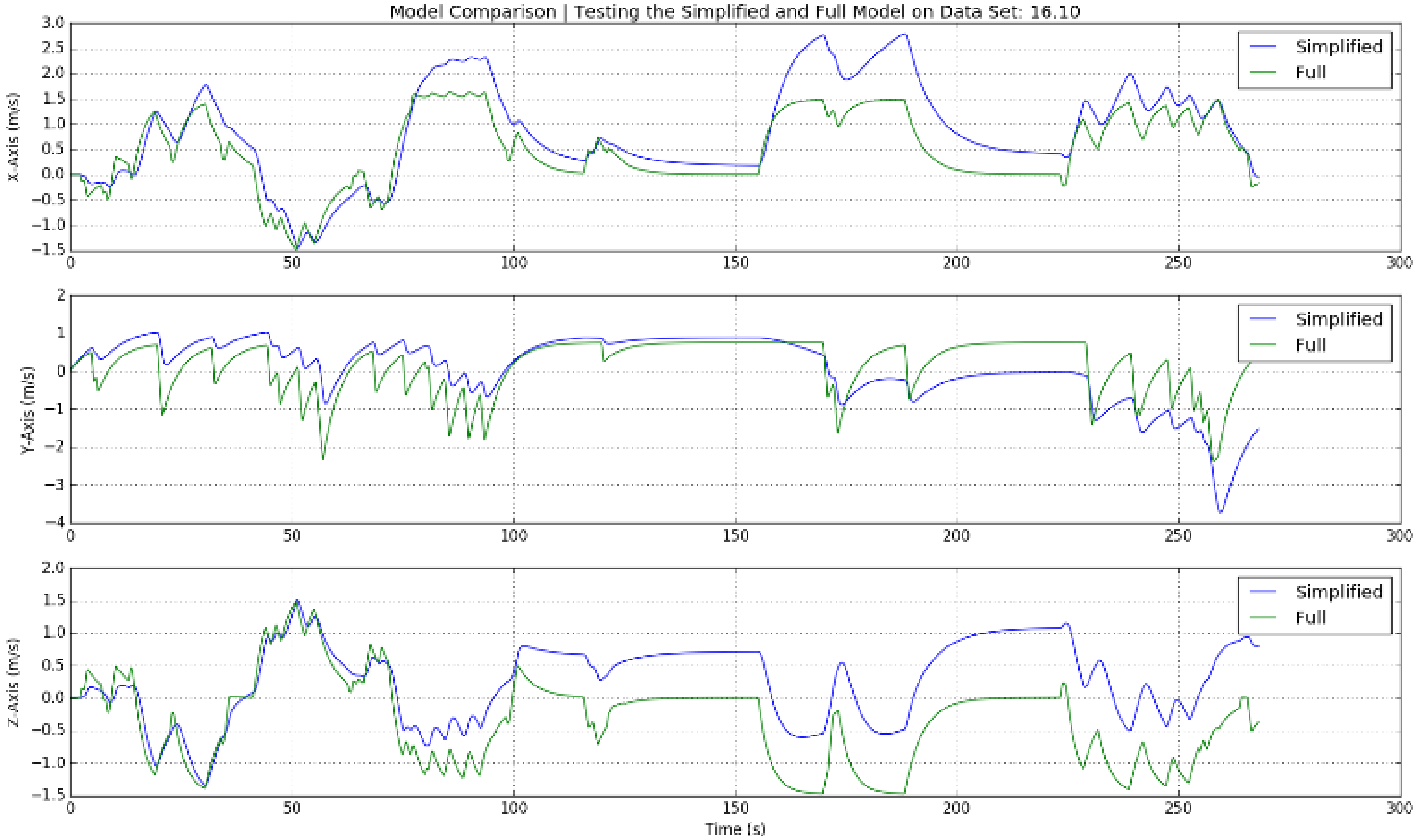
Animation of BlueROV2

GENERATED VIA SYSTEMMODELER



Simplified BlueROV2 Model

GENERATED VIA OPENMODELICA



Model Comparison – Full vs. Simplified

Experimental Procedure

In [1], a control matrix was derived to determine the motor command values from tele-op input.

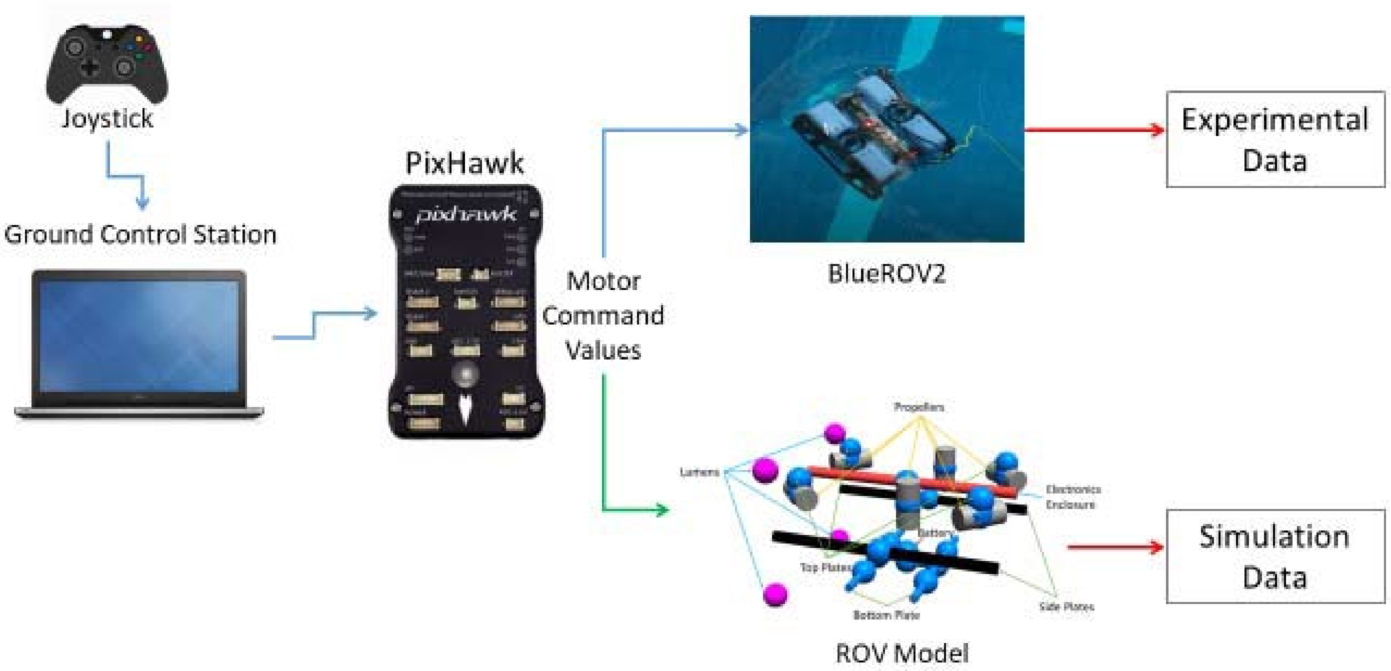
- It is difficult to determine if this control matrix matches that of the ROV
- Many approximations were made during its derivation

Instead, the testing relies on the motor control values

- These were pulled from the Pixhawk after each test
- 135 tests were done
- Only 59 of these tests contained usable data
  - 34 of the remaining datasets contained nontrivial data
  - The data was passed through a simple first-order filter to remove some measurement noise

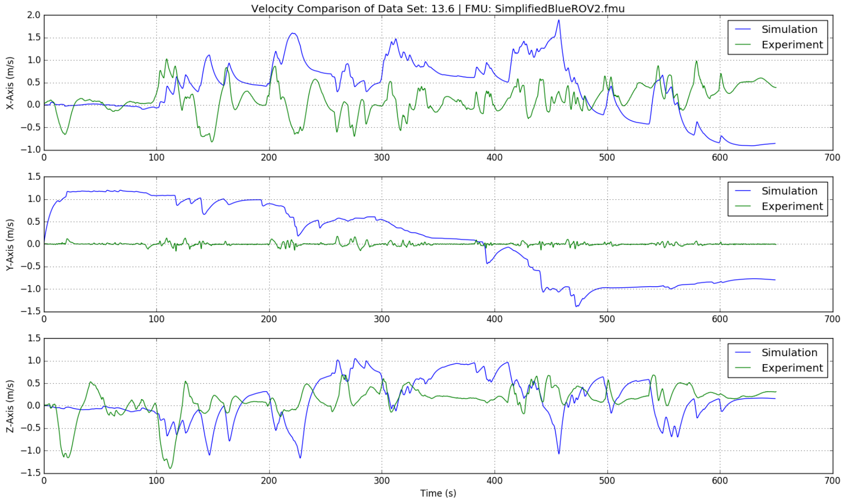
The motor command values were run through the simplified model

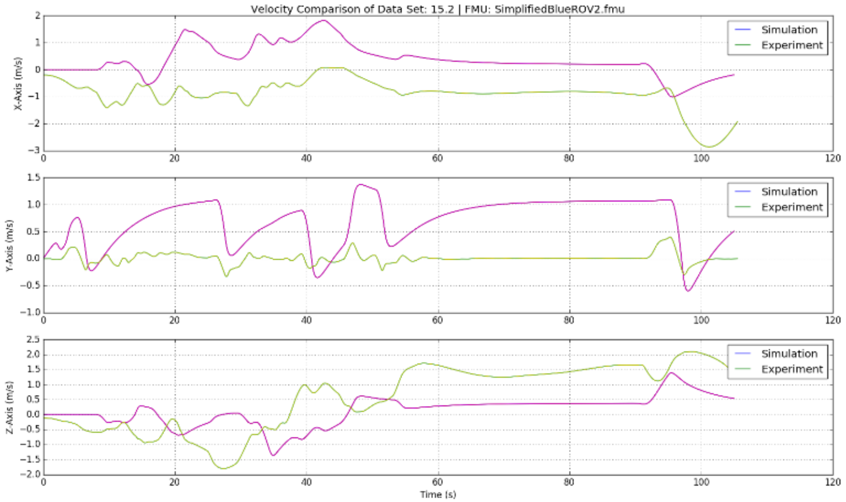
[1] Swaminathan, S., & Saripalli, S. (2018). Developing a Framework for Modeling Underwater Vehicles in Modelica. In Proceedings of the 2018 American Modelica Conference, October 9-10, 2018. Linköping University Electronic Press, in press.

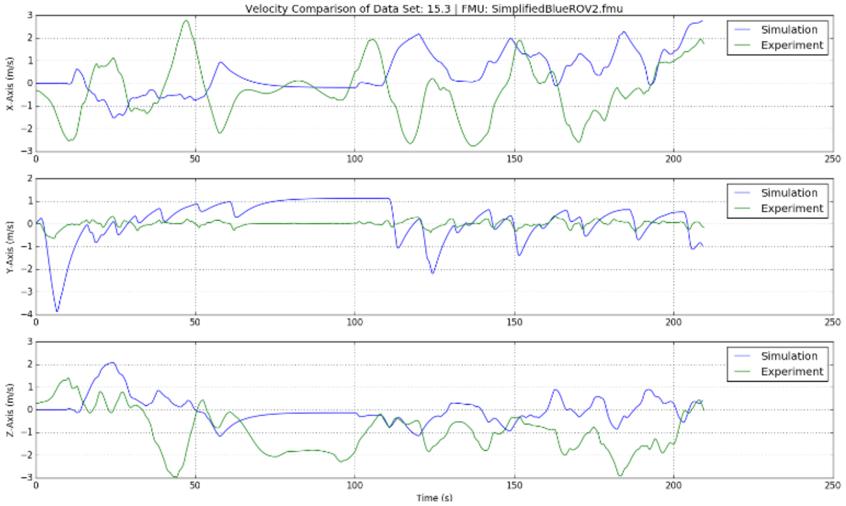


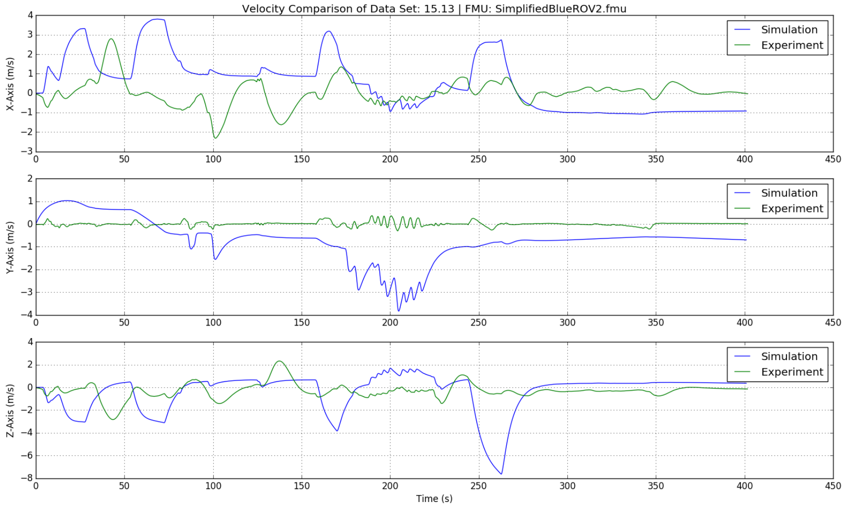
Testing Process

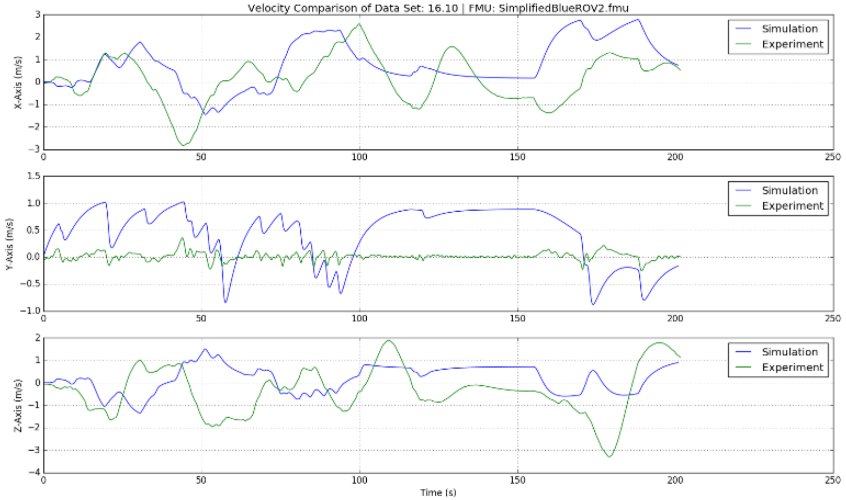
A SCHEMATIC OF THE EXPERIMENTAL PROCEDURE

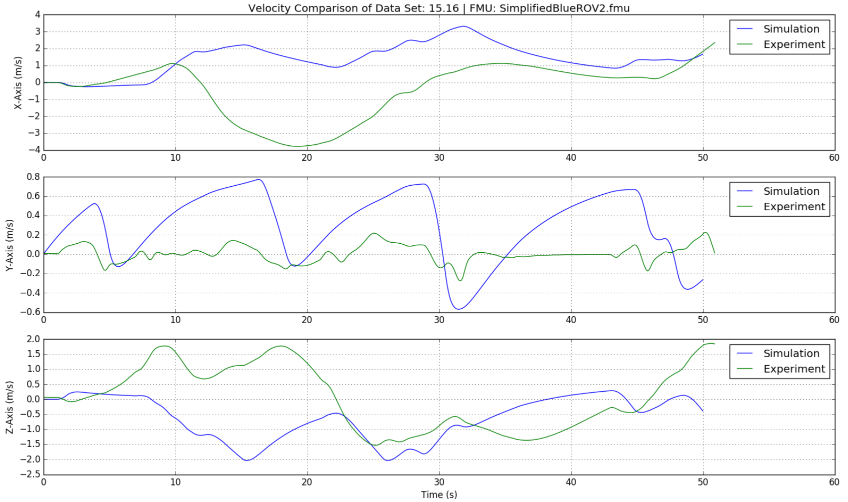




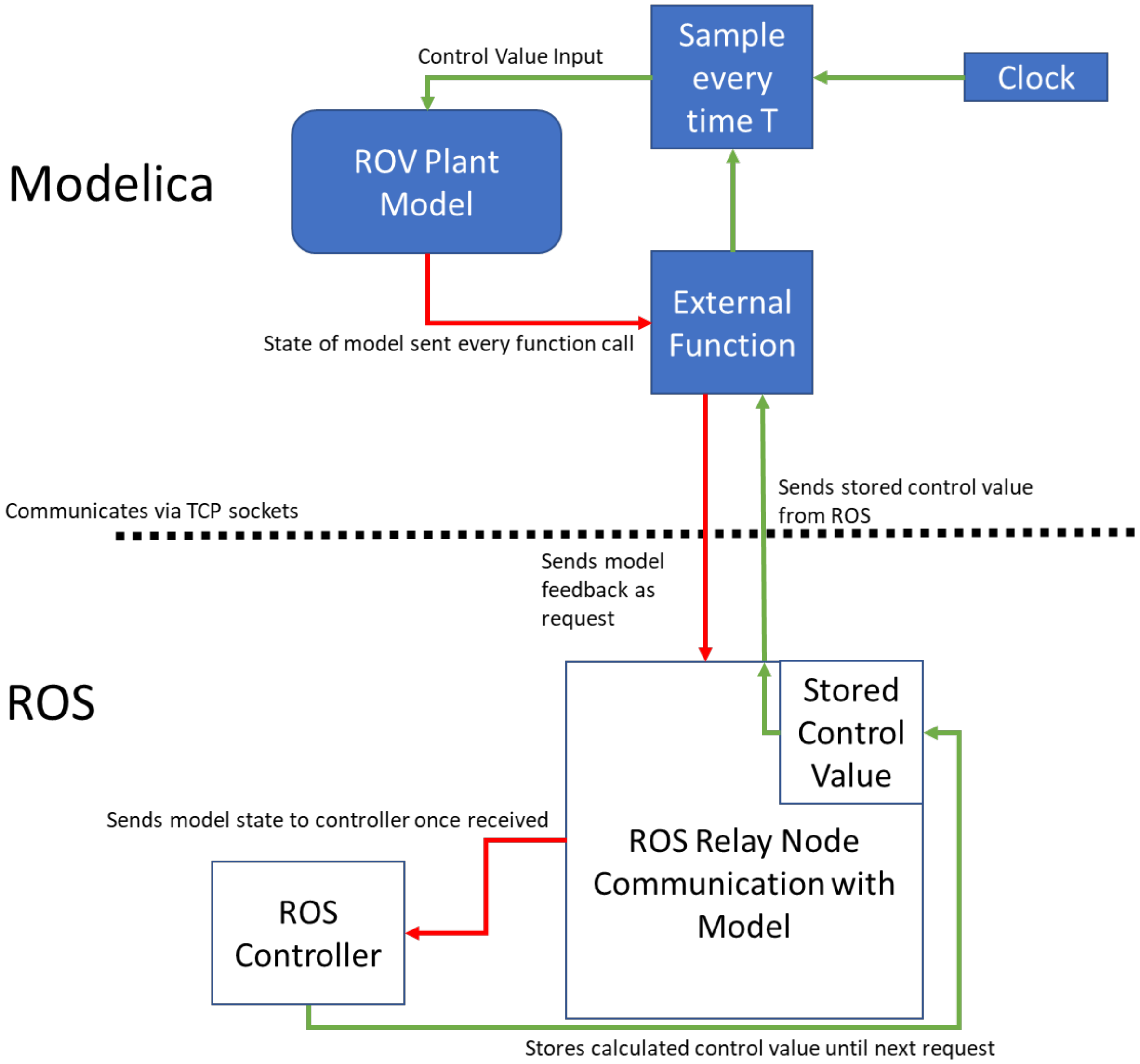








Results Comparison

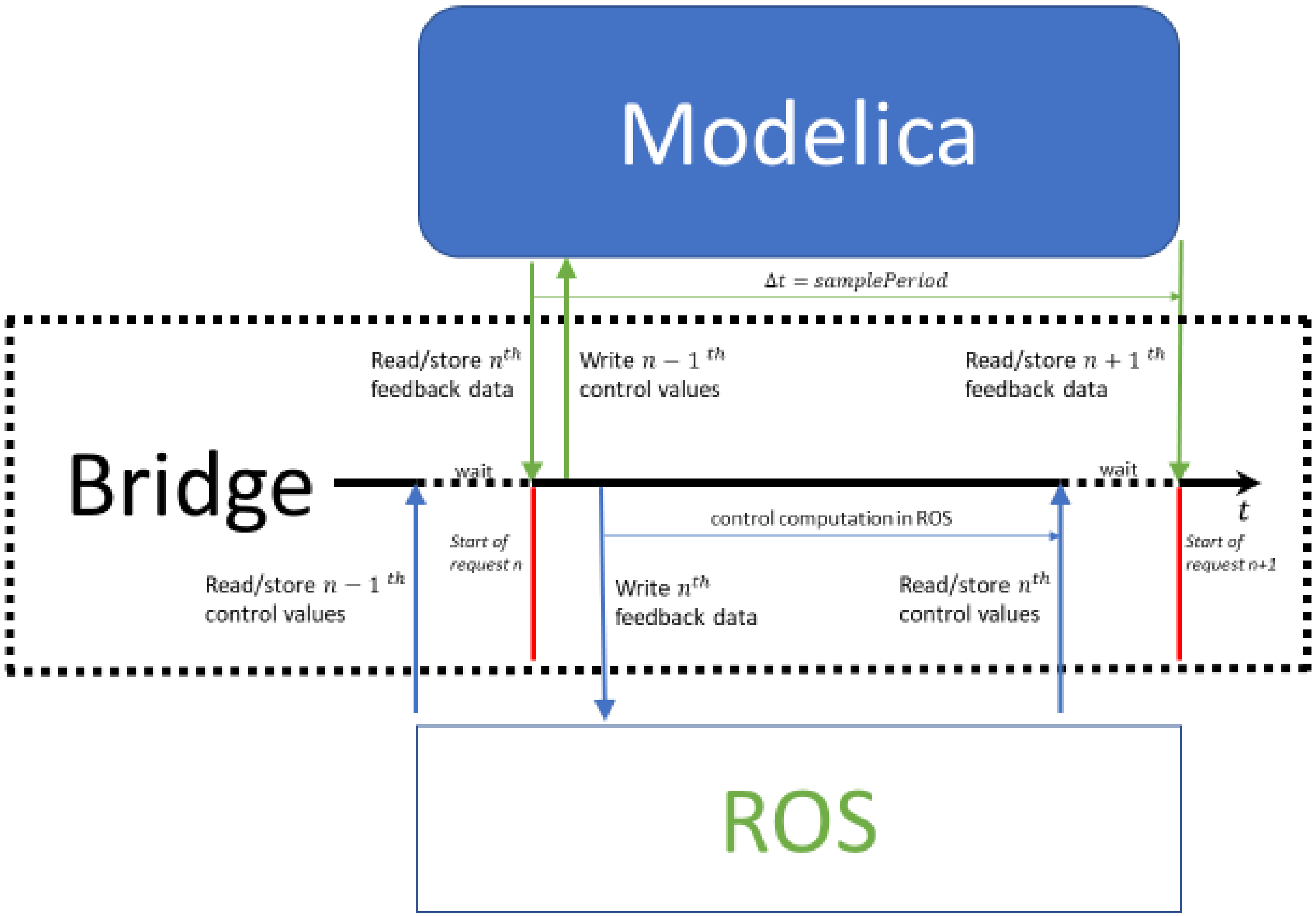


Bridge to ROS

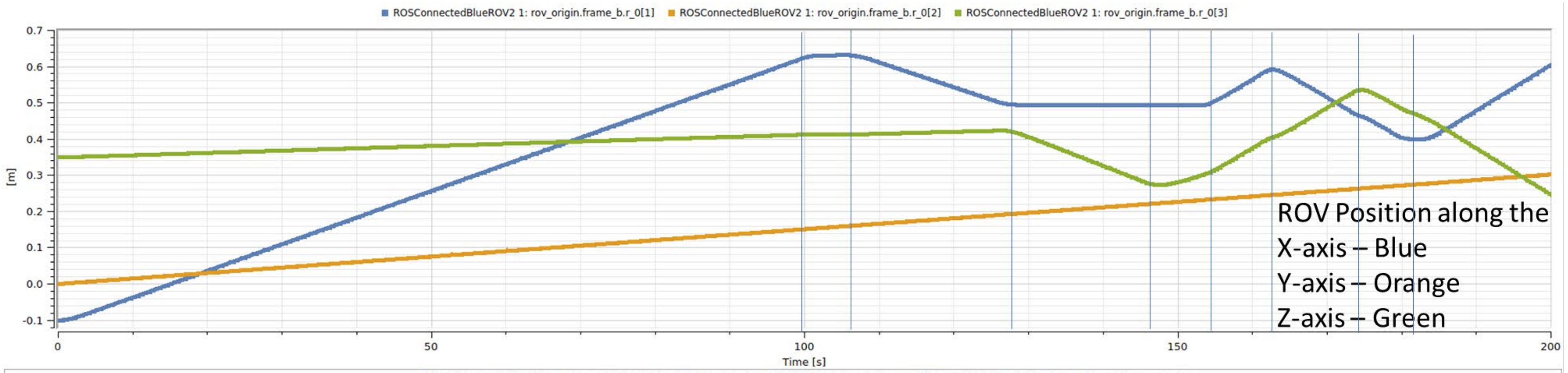
Diagram detailing the connection between Modelica and ROS

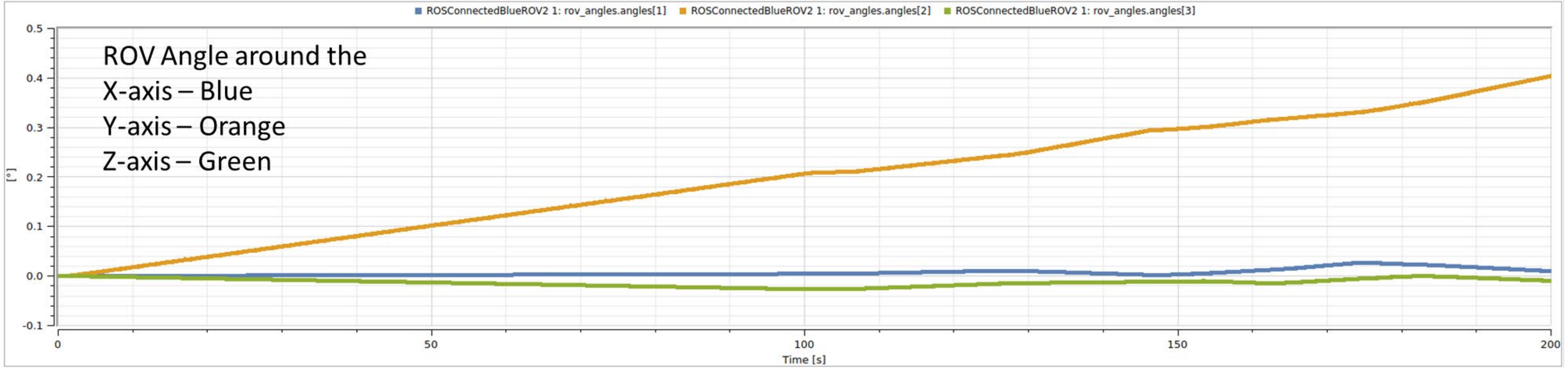
Communicates via TCP/IP sockets

Developed separately beyond paper's scope as *modelica_bridge*



Discrete Data Flow in Bridge





Simulation Results – Motor Inputs via ROS

Conclusions

The URBL was stably constructed to provide basic ROV modeling components, as well as ready-to-use integration with ROS

The URBL was successfully used to model an existing commercially available ROV design, the BlueROV2, and was validated against experimental data.

Future Work:

Library Improvements

- Viscous drag representation
- Hydrodynamic function replaceability

Model Improvements

- Better parameter analysis

Validation Improvements

- Utilize ocean data