

# Reducing Modelica Models Connection Errors in Dymola

Raees B. K. Parambu – Systems (Project) Engineer

**Software/Language used:** Dymola 2018 FD01, MSL 3.2.2

**Keywords:** Modelica, Cyber, Physical, Model, Connection, Error, Dymola.

**Abstract:** This blog post presents a simple method for reducing connection errors while developing or using cyber-physical systems (CPS) models in Dymola. For demonstrating this method, a synchronous machine permanent magnet (SMPM) model from Modelica Standard library (MSL) is used. It is shown that by carefully adding a correct unit to a Real input causal connector of the SMPM model ensures that a signal generated from the physical part of the model passes only to the correct cyber part of the same model.

## Introduction

For accelerating model development processes, Dymola provides a powerful graphical user interface (GUI). In Dymola, both plant (physical) and control models can be created and simulated from a single model. In this blog post, such a model is called as a cyber-physical model, like that in Figure 1.

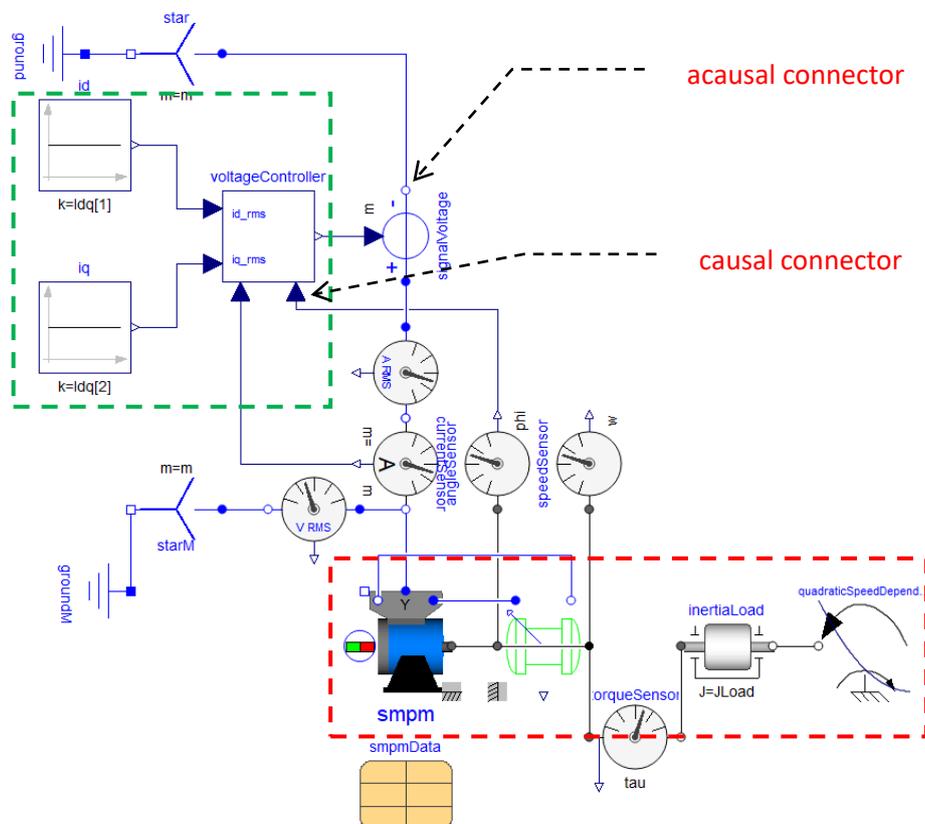


Figure 1– Diagram layer of MSL synchronous machine permanent magnet (SMPM) model. Path: Modelica.Electrical.Machines.Examples.SynchronousInductionMachines.SMPM\_VoltageSource

In Figure 1, the green square is the controller part and the red rectangle is the physical part of SMPM model. Communication between components in this model is achieved with the help of Modelica Connector (a specialised class in Modelica modelling language [1]). It can be broadly classified based on its function and flow direction of its information as shown in Figure 2.

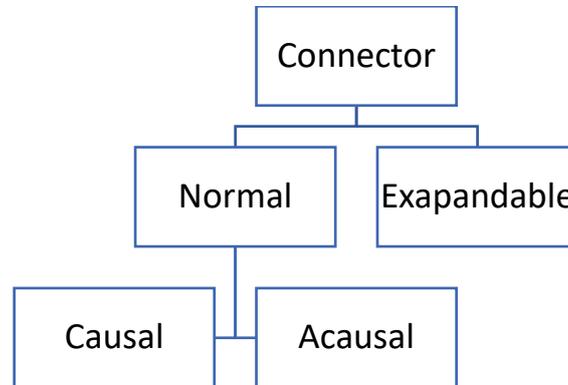


Figure 2 – Types of connectors in Modelica language based on [2].

In Figure 2, the first classification is based on its function, whereas, the second layer is based on its direction of the flow of information. In the case of a causal connector (shown in Figure 1), the flow of information is unidirectional, whereas, an acausal (physical or domain-specific) connector (shown in Figure 1) can pass information in both directions. Moreover, the causal connector can further be classified based on its primitive types such as Real input or output, Boolean and Integer. For more details about these connectors, please refer to [1, 2].

In Dymola, an error message is given if anyone attempts to connect two different types of acausal connectors together; for example, the left side connector of **inertiaLoad** and negative connector of **signalVoltage** in Figure 1. However, this is not the case with causal connectors they are more general. Say, if someone tried to interchange connection between **id\_rms** and **iq\_rms** of the **voltageController** in Figure 1, Dymola does not provide any error message. That means if a Real output causal connector is unitless, then you can connect them to any another Real input causal connector. However, we get a wrong simulation result. Therefore, this blog post is aimed to present a simple method to avoid such an error occurring and the wrong information being passed from the physical part of a CPS model to control part of the same model.

## Methodology

For reducing the connection error in a CPS model, the following steps can be used.

1. Open Dymola and create a new package called **CPS** in the Package Browser.
2. Duplicate the model, **SMPM\_VoltageSource** available in MSL, and place into the newly created package, as shown in Figure 3.

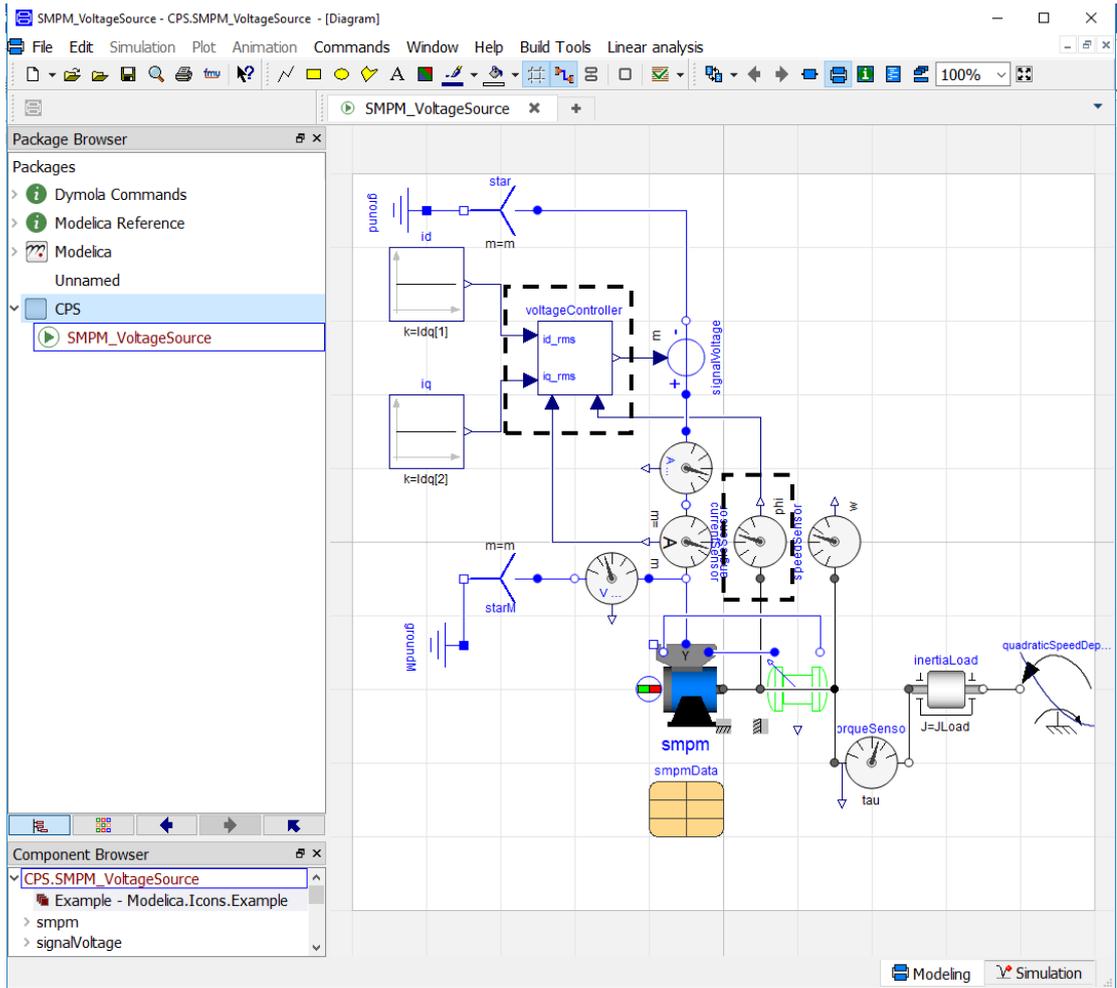


Figure 3 – A New package, CPS, with a model duplicated from MSL.

3. Right-click on the **voltageController** model, see Figure 3, and then select **Open Class** option, as shown in Figure 4, a voltage controller model will be displayed.

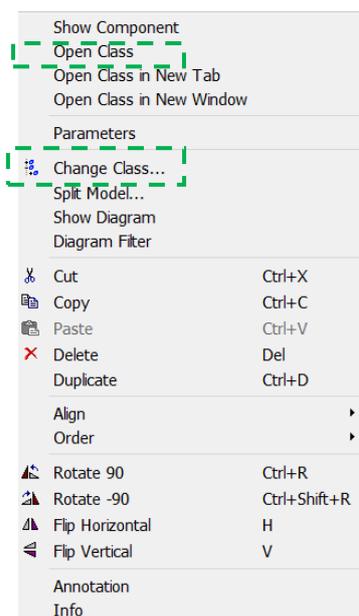


Figure 4 – Object right-click dialog box.

- Duplicate this **voltageController** model available in MSL, and place it into the newly created package, as in step 2 (Figure 5).

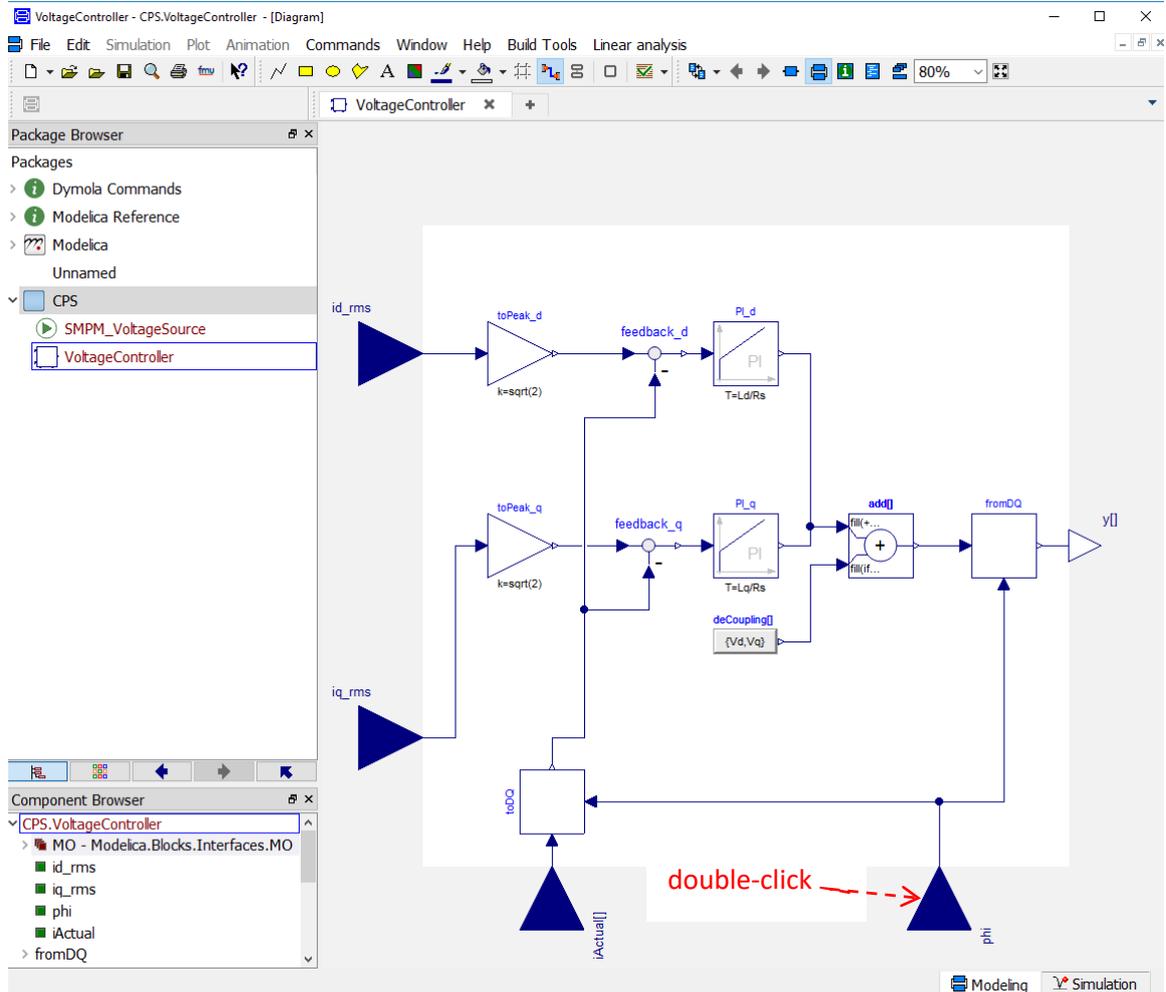


Figure 5 – The duplicated voltage controller model marked in Figure 3.

- The next step is to check the unit of the **phi** connector of the **angleSensor** in Figure 3. For this purpose, repeat step 3, but switch to Modelica Text layer as shown in Figure 6.

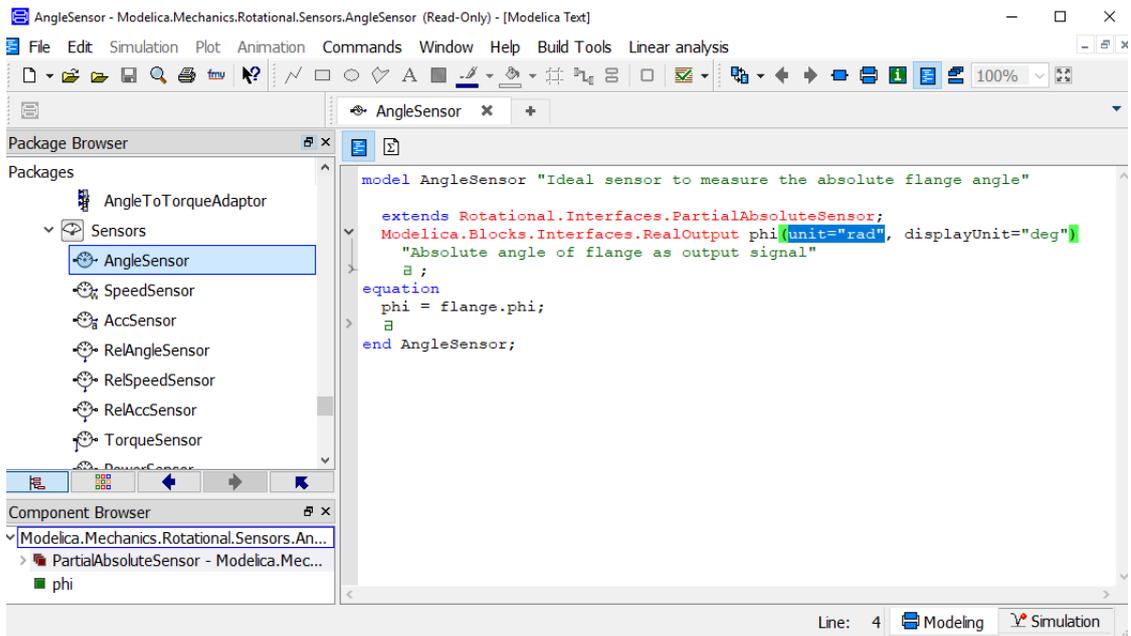


Figure 6 – Modelica Text layer of MSL AngleSensor.

6. As shown in Figure 6, the unit of the **phi** connector is “rad”. Copy the highlighted text in Figure 6.
7. Now, go to the duplicated **voltageController** model in **CPS** in Figure 5, and double-click on **phi**. A pop-up dialog box will appear and then click **Add modifiers** tab and paste the unit copied in the previous step to the empty field as shown in Figure 7. Then click OK.

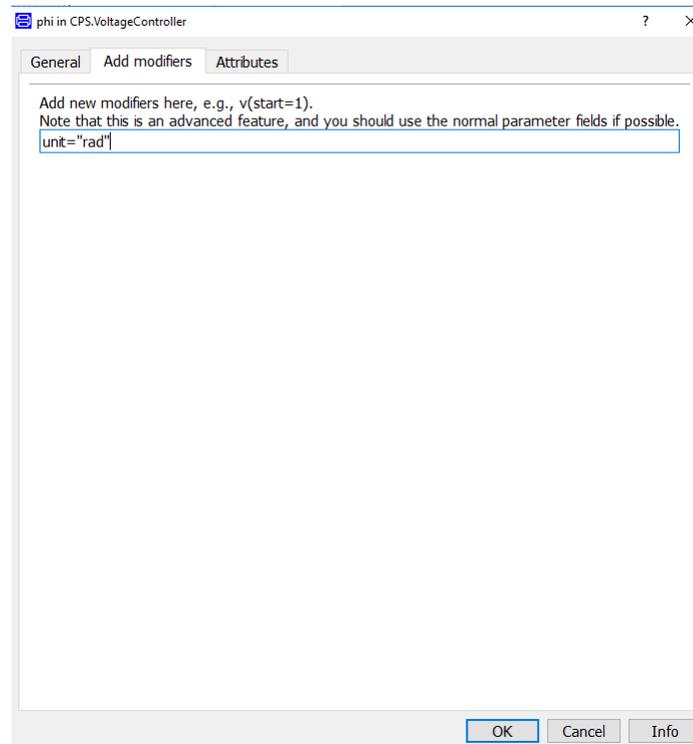


Figure 7 – Dialog box of CPS.VoltageController’s **phi** connector, with Add modified tab is displayed.

Now, follow step 8 to replace the **voltageController** model in the duplicated **SMPM\_VoltageSources** model in **CPS** package with the modified **VoltageController** model in the same package.

8. Right click on the **voltageController** model in Figure 3, the same dialog box in Figure 4 will appear, but, this time select the **Change Class ...** A new dialog box will open-up as in Figure 8. Browse through this dialog box and select the modified **VoltageController**. Then click OK.

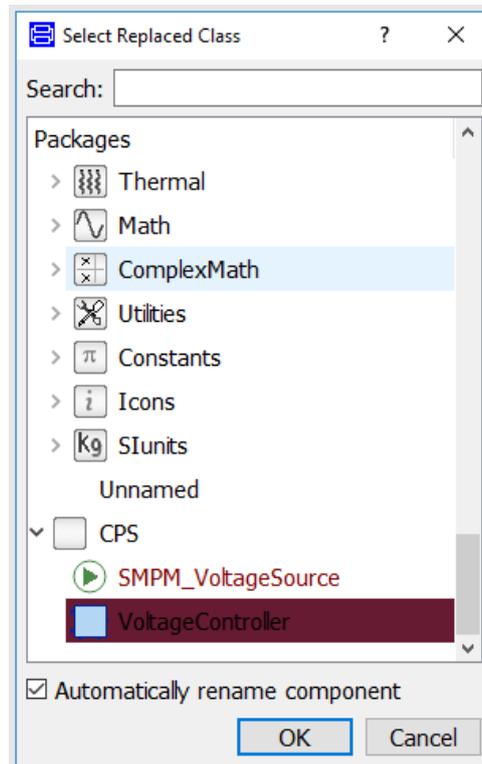


Figure 8 – *Change Class...* dialog box.

## Experimentation

This section will investigate the above-presented method. However, in Figure 9, for the better visualisation, a graphical representation of steps presented in the last methodology and the following experimentation sections are summarised.

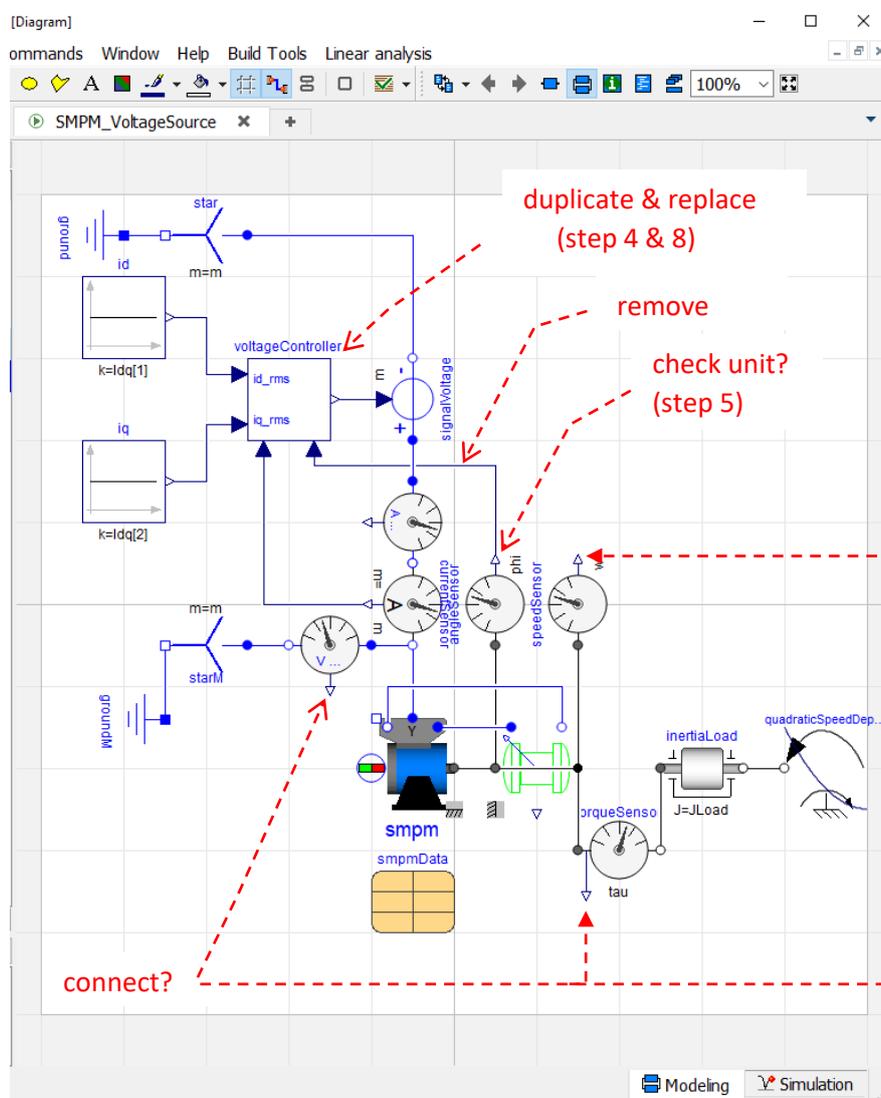


Figure 9 – Same as Figure 3 but modified to graphically summarise both methodology and experimentation steps.

In the modified **SMPM\_voltageSource** model in **CPS** package, remove the connection between **angleSensor.phi** and right bottom connector of **voltageController** (**voltageController.phi**). Now try to connect **speedSensor.w** to **voltageController.phi**. The following warning message will appear.

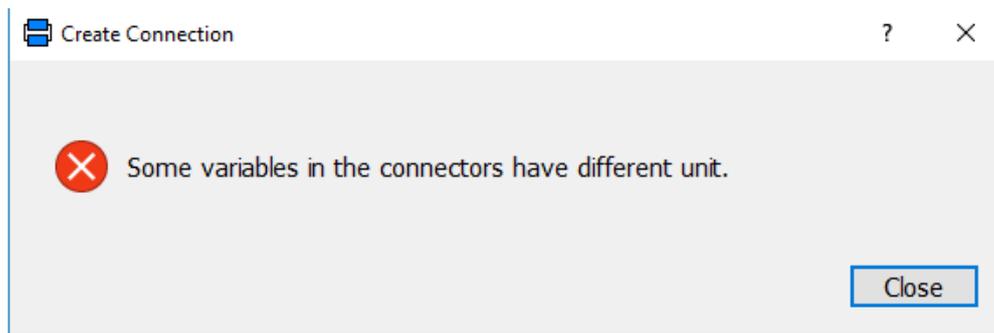


Figure 10 – Error message appeared when attempted to connect two causal connectors.

Repeat the above experiment with other connectors such as **voltageQuasiRMSSensor.V** and **torqueSensor.tau**. The same error message in Figure 10 will have displayed at each attempt. However, when connecting the **angleSensor.phi** and **voltageController.phi** again, no error message will be displayed because they have the same unit applied. For further investigation, one can simulate both models and then compare the simulation result especially the signals from the modified connectors.

## Conclusion

This blog post presented a simple method for reducing connection errors when connecting causal Real connectors of a cyber-physical systems (CPS) model in Dymola.

## References

1. Modelica language specifications version 3.3 Revision 1, Copyright © 1998–2014 Modelica Association July 11, 2014, <https://www.modelica.org/documents> (03 Oct. 17, last accessed).
2. Modelica by Example, [http://book.xogeny.com/components/connectors/connector\\_def/#connector-def](http://book.xogeny.com/components/connectors/connector_def/#connector-def) (10 April 18, last accessed).