# A Physical Modelling Approach to ARGESIM Benchmark C11 'SCARA Robot' using Dymola

Boris Malinowsky[1], Oliver Höftberger[1], Florian Miksch[2 *]

[1] Inst. for Analysis and Scientific Computing, Vienna Univ. of Technology, Wiedner Haupstraße 8-10, 1040 Vienna
[2] dwh Simulation Services, Neustiftgasse 57-59, 1070, Vienna, Austria; * *Florian.Miksch@@drahtwarenhandlung.at*

**Simulator.** Dymola is an object oriented environment for simulation based on MODELICA language. Due to the object orientation, defined components can be fully reused. Equations can be stated in neutral form (DAEs), and algorithms are used to derive efficient equation code (compiled from C language) for simulation runs. This allows acausal modeling to be used without consideration of computational order. For simulation, Dymola uses the implicit DASSL solver.

**Modelling:** In the definition of the benchmark implicit equations describe the problem. Graphical elements in Dymola's model editor are parameterized through the component dialogs. A few equations given in the problem definition are directly written in textual form into the model. Those textual additions might be edited at a later point from the components dialog as well. The editing of the graphical forms automatically creates and adjusts the corresponding model text descriptions. Connect statements are created for the model by drawing connector lines in the editor. Libraries support modeling, here the *Modelica. Mechanics.MultiBody* Robot drives, control and collision avoidance control can be added by using standard library components.

**A-Task: Mechanical Modelling Approach.** The SCARA mechanic parts were put into own model component and instanced once for the robot, parameterized accordingly. The same was done for the motors and control logic. Listing 1 shows the structure of the generated textual model description, which is flattened to be called from the DASSL solver.

```
model scaraMechanics
  Inner Modelica.Mechanics.MultiBody.World…….
  Modelica.Mechanics.MultiBody.Parts. …….
  Modelica.Mechanics.MultiBody.Joints.
    ActuatedRevolute actuatedRevolute a;[...]
equation
  connect(world.frame_b, b0.frame_a) a;
  connect(b0.frame_b, actuatedRevolute.frame_a) …
end scaraMechanics;
```

**Listing 1.** Textual model generated by Dymola.

In Figure 1, the whole layout of the SCARA robot is shown. On the left side in the middle of the figure, the control and collision avoidance component is located. Default control is applied later in Task b, collision avoidance in Task c. The component outputs the maximum voltage to apply for the servo motors of link 1 and 2, and the destination angle position of the two robot joints 1 and 2. Two inputs are used for the current angle positions; its values are dependent on the movement of the rotational drives. The third input is the current vertical position of the tool tip (the joint distance in z-direction). On the right side in the middle of the figure, the object of the mechanical structure (mechanics component) is located.

Figure 2 shows the mechanics component, modeling the robot arm. The link and joint objects are instanced from predefined parts of the *Modelica.Mechanics. MultiBody* library. The component contains 3 inputs, 2 rotational angle connectors for the absolute rotational angle (joint 1 and 2), and 1 translational angle connector for the absolute positioning (joint 3).

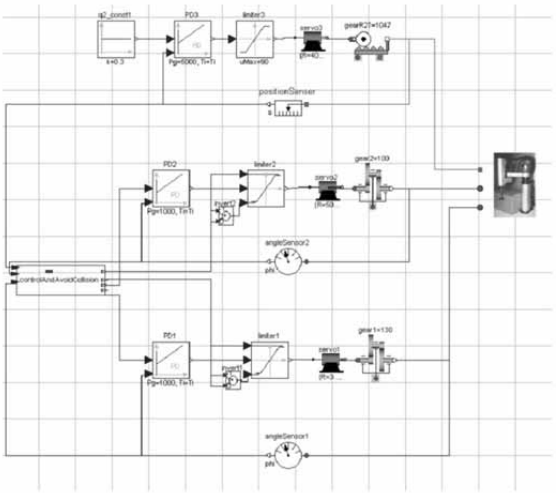

**Figure 1.** Layout of the model in the graphical editor of Dymola.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 21(1), 2011, 53-54 | doi: 10.11128/sne.21.bn11.10050

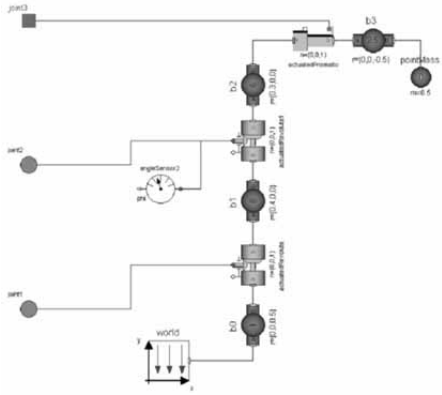

**Figure 2.** The modeled SCARA mechanic component using Modelica MultiBody parts.

**B-Task: Simulation of a point-to-point movement.** The robot contains 3 servo motors to drive the joints that are controlled by a single axis PD-controller each. The PD-controller is a modified standard component from the MODELICA library. A limiter block limits the voltage to a positive and negative maximum while the motor limits the maximum armature current in its electric/mechanic transformer (EMF, component limited EMF) using an if-condition. Figure 3 shows simulation results, start position is 0 for all joints (q1, q2, q3 equals 0), end position is 2 rad for q1 and q2, and 0.3m for q3.

The standard DAE solver DASSL worked efficiently. Dymola offers also other DAE solvers like LSODAR, a multi-step solver for stiff and non-stiff DAEs with root finder, and DOPRI45, a Runge-Kutta method with local extrapolation. While DOPRI45 was significantly slower thatn DASSL (factor 2.8), the LSODAR solver was slightly faster – factor 0.8. It must be expected, that LSODAR is faster if state events as the possible collision (Task C) must be recognized, but also without state events the algorithm has its benefits.

**C-Task: Collision avoidance feature.** For detecting a collision of the robot tool tip with an obstacle, a collision detector component is modeled. It controls maximum voltage and angular target positions. Collision avoidance can be enabled (for this task) or disabled (for Task b). A graphical structure setup in the component is used to calculate the tool tip position xtip (green graph in Figure 3). A logical network with boolean expressions checks for the critical position.

In case of moving into a critical position (possible collision in the future), the a boolean condition becomes true, it freezes tool tip position and stops angular movement by setting the temporary target position for motor 1 and 2 to current position.

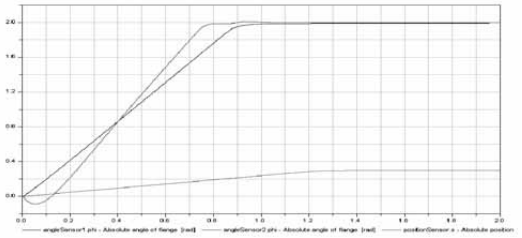

**Figure 3.** Joint positions without collision.

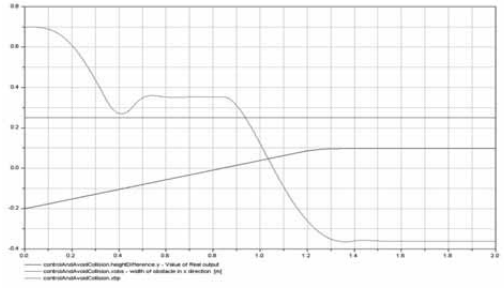

**Figure 4.** Tool-tip position and obstacle (in x-direction), and difference over time with collision avoidance.

It also switches output signal from regular maximum voltage to maximum emergency voltage. If the tool tip raised above obstacle height, targets and voltages are set back to regular. These actions are synchronized with the DAE solver (state event finding) by means of IF or WHILE constructs in DYMOLA.

The simulation results are shown in Figure 4, with simulation time 2 seconds, and distances and lengths drawn in meters.

Dymola also provides the functionality to animate and render graphical objects representing physical components during simulation runs (Figure 5).

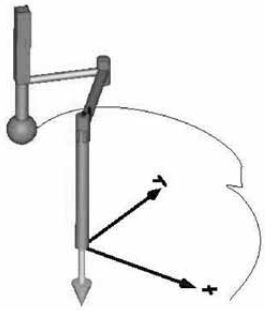

**Figure 6.** Animation view with tool-tip track (blue line) after collision avoiding

**Resumé.** Dymola provides appropriate predefined components in the MODELICA mechanics library and the standard library to build up the SACRA Robot and to implement the requested behavior as well as collision avoidance feature. Making use of plotting and rendering functionality offers the possibility to visualize the simulation and results very well.

Received: December 2009
Revised: July 20, 2010
Accepted: November 30, 2010