An OO Numerical Solution to ARGESIM Comparison "C11 - SCARA Robot" using AnyLogic

F. Judex, F. Breitenecker, TU Vienna
efelo@fsmat.at

Simulator. AnyLogic is able to handle continuous, discrete and hybrid models. It is based on JAVA and therefore object-oriented. It offers drag-and-drop dialogues for the basic parts of the model's structure as well as for animation. Everything needed is created as an instance of the ActiveObject class, starting with the 'root' class which represents the model to state variables (the 'important' variables which can appear on the left-hand side of an ODE and can be plotted), statecharts and animation.

Task a: Modeling Method: AnyLogic can not handle implicit models; therefore the mass matrix was first inverted using Maple 8. Maple was also used to compute the maximal armature current. The components of the acceleration vector as well as the position vector and the armature currents – all variables at the left hand side of the ODE's – are ActiveObjects, and are embedded in the ActiveObject "Arm" along with all parameters (fig. 1). Collision is treated by introducing a state chart, where the transition condition `(x<xobs+0.1) && (z<hobs) && kol` allows to handle collisions with an obstacle at a certain position and height, if the Boolean variable `kol` is set to true.

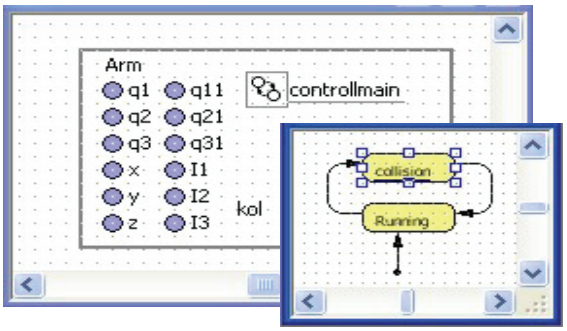

Fig. 1: AnyLogic model with statecharts

Curiously, AnyLogic has no limit function neither a limiting integrator; so a JAVA-function is used to keep the current and the voltage within the given limits.

The actual integration happens at two levels. The ODE's for the currents and the speed are solved in the statechart, while the positions are calculated in the ActiveObject "Arm". This is only possible because AnyLogic is sorting the equations during runtime. If a collision event (defined in the statechart) happens, equations are resorted.

Task b: Simulation: After the Model was set up, experimentation itself is a menu-driven object; also to get a plot of the position values. As all of them are ActiveObject variables, AnyLogic conveniently provides the plot shown as figure 2 via Drag-and-Drop dialogues.

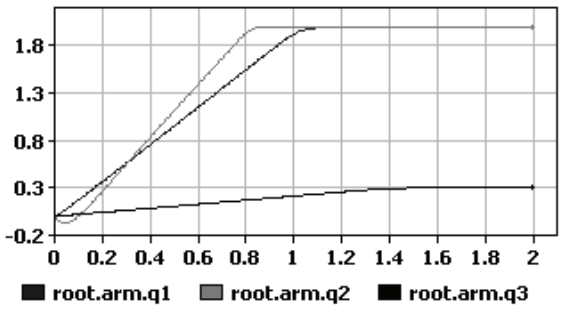

Figure 2: Results for positions without collision

Task c: Collision Avoidance: Once the Boolean variable `kol`, again an active object, is set to true, the transition between the "Running" state and the "collision" state is possible. In case of collision emergency, the maximal voltage of 230 volt is applied to the rotational motors, which control the horizontal movement, so their acceleration is set to the biggest negative value.

On the other hand the voltage for the linear motor is set to the positive maximum – also 230 volt – so the tip is raised at maximum speed. Once the tip is either high enough or reached a save distance from the obstacle, the transition "Running" state takes place. As seen in Figure 3, plotted with the same routines as number 2, one more emergency movement is necessary before the obstacle is passed.

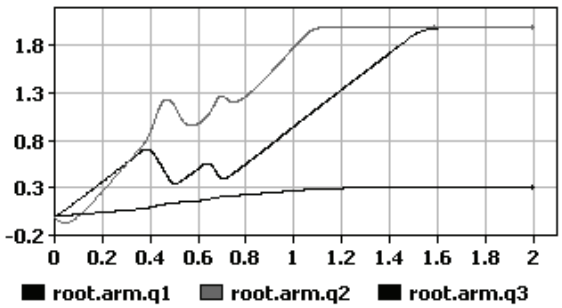

Figure 3: positions during collision avoidance

C9 Classification: OO Numerical Approach
Simulator: AnyLogic 4.5, Rel. 2003