



C13 Crane and Embedded Control MATLAB

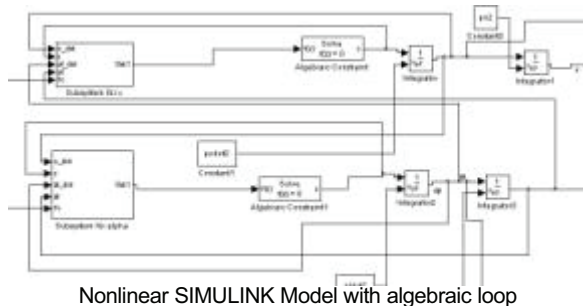
Hybrid Approach

Simulator: MATLAB is a widely used software tool based on numerical vector and matrix manipulation. SIMULINK is a graphical extension for block oriented simulation.

The model: Control and sensor diagnosis of the system was implemented in MATLAB, of linear and nonlinear dynamics were modelled in SIMULINK. The time base of the controller ($\Delta t=0.01$ sec) was used as discretisation basis of the controlled system: loop with i) sensor check, ii) emergency actions, iii) calculation of new discrete control (with brake) and iv) simulation of the dynamics upon next sampling interval (using the SIMULINK model of the linear dynamics):

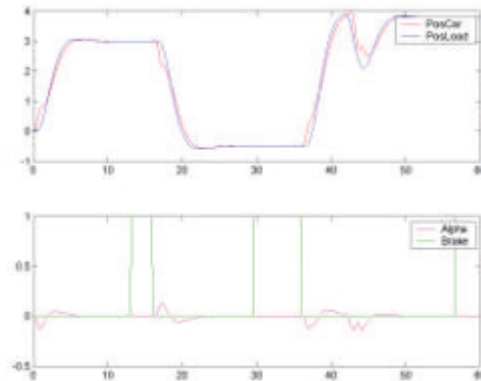
```
for systimeind = 0:endtimeind
    [ES,EM,BrakeTrigger] = sensorcheck
    (SensorErrorTime,SensorErrorValue);
    [VCneu,qneu] = control (alpha,PosCar,
    PosLoad,PosDesired,EM, VCalt,qalt);
    VCout = controlbreak(VCneu,ES,
    BrakeTrigger,newPosDesired,IEnable);
    sim('cl3sim_lindfe',[0 0.01]);
end %
```

Task a: Comparison of uncontrolled nonlinear and linear model. SIMULINK is a block-oriented graphical simulator, based on directed signal flows, so that implicit systems (i.e. an algebraic loop for the signal flow) cannot be modelled directly. In this solution the `solve` block was used to break the algebraic loop for the equations for car acceleration and angle acceleration. This block constrains the input $F(\ddot{\alpha}; \alpha, \dot{\alpha}, x_1, \dot{x}_1) = 0$ to zero wrt $\ddot{\alpha}$ (fed back as output to the input of the subsystem, which represents the implicit equations) - see next figure. It is to be noted, that in each integration step this iterative solver is called to calculate the derivatives. The table shows the differences between in x_1 of the linear and the nonlinear model after $t=600$ sec.



disturbance fd	-750	-800	-850
difference Δx_1	-262.89	-325.19	-391.05

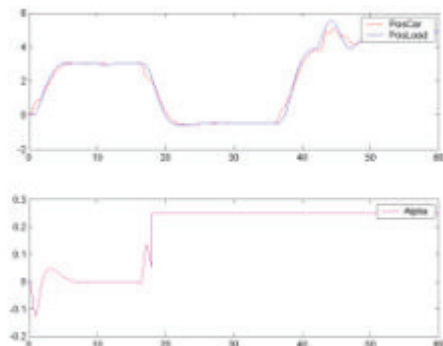
Task b: Simulation of the controlled system. Controller and brake logic were implemented in MATLAB m-functions, called in the loop shown before. The next figure position of the car and of the load, and the angle of the cable and the activation times of the brakes, which are: $t = 13.17, 29.63$ and 56.56 sec.



Task c: Simulation of controlled system with sensor diagnosis. Sensor diagnosis is implemented by a fourth action in the core loop (at first position). The last 10 states are scanned in order to perform diagnosis check. In MATLAB this check looks like:

```
function [ES,EM,BrakeTrigger]= sensor-
check(AngleSensorErrorTime,AngleSensorErrorValue)
for i=1:10,
    if ( (ErgPosCar(systimeind-i) < PosCarMin) |
    (ErgPosCar(systimeind-i) > PosCarMax) )
        diagRange(i,1)=1; end; .....
    .....
    if ( AngleSensorValue >= AlphaMax)
        diagAngle(i,1)=1; end;
```

The whole system goes into Emergency Mode at $t = 18.05$ and the brake was switched on at 13.17 and 32.64 , whereas Emergency stop was activated at 44.47 (see next figure)



Daniela Schachinger, Wolfgang Weidinger, TU Vienna
dschachi@fsmat.at, wieweit@fsmat.at

