Simulating Mobile Robots Using Simulink

Roy Featherstone

School of Engineering
Australian National University

Why Use Simulink?

- widely used simulator
- good user interface
- good documentation
- tightly coupled with Matlab
- general discrete/continuous event simulator
- variable-step integrators
- both interactive and programmable
- supports model libraries, sharing and hierarchical model decomposition

Why Not Use Simulink?

- expensive
- slow (compared with best special-purpose simulators)
- designed for systems with unidirectional signals

Programmability

From inside a Matlab function you can call

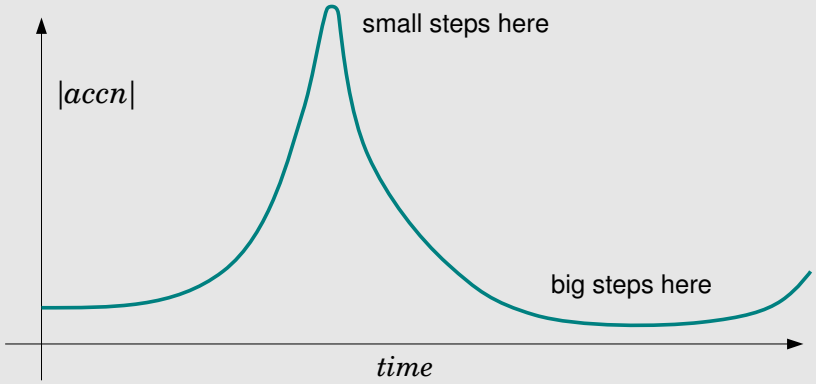
sim(*modelname*, *parameters*);

to make simulink run a specified simulation with specified parameters. Results can be output to the Matlab workspace.

This is useful (e.g.) for automatic optimization of parameter values.

Variable-Step Integrators

Rigid-body dynamics tends to produce spiky accelerations. A variable-step integrator can improve both efficiency and accuracy by adapting the step size.



Discrete/Continuous Simulation

- discrete events are instantaneous – they take place at instants in time
- continuous dynamics is simulated (by numerical integration) in the time intervals between discrete events
- there are two types of discrete event:

<u>data driven</u>

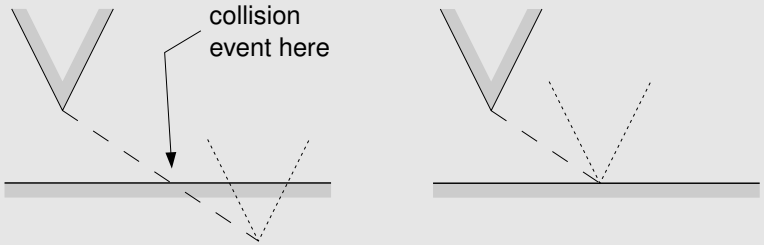
- gain/loss of contact
- stick/slip transitions
- hitting a saturation limit

<u>scheduled</u>

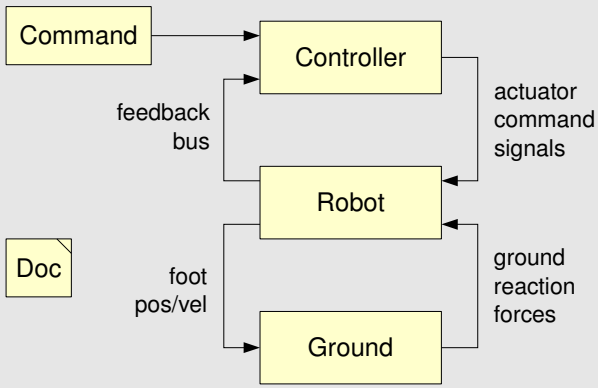
- scheduled finish time
- regular execution of controlling software (e.g. servo cycles)

Discrete/Continuous Simulation

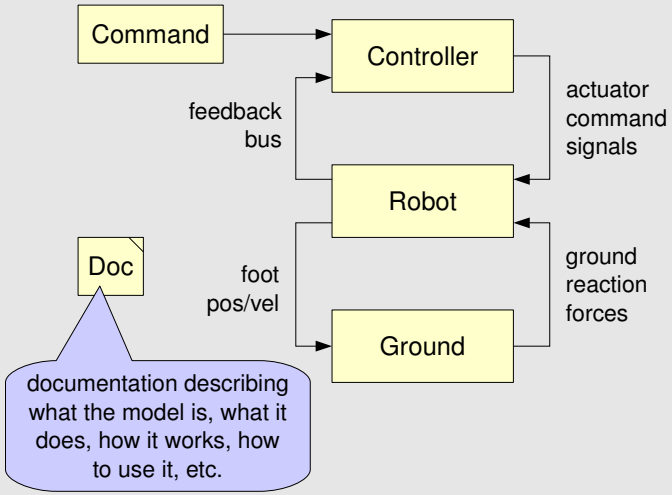
whenever a discrete event occurs, the current integration step is truncated (if necessary) so as to end at exactly the instant when the event occurs



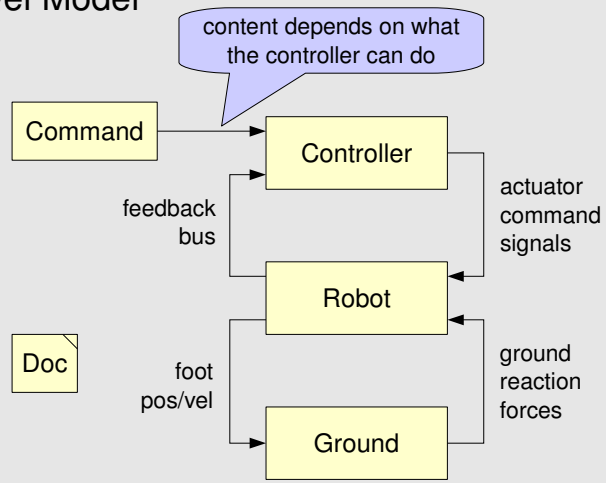
Top-Level Model



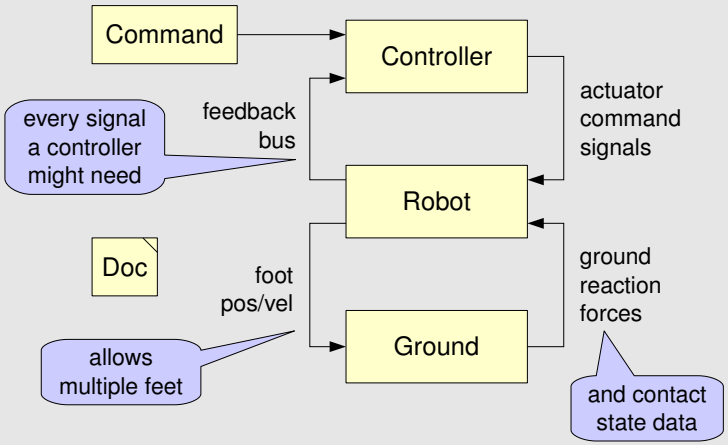
Top-Level Model



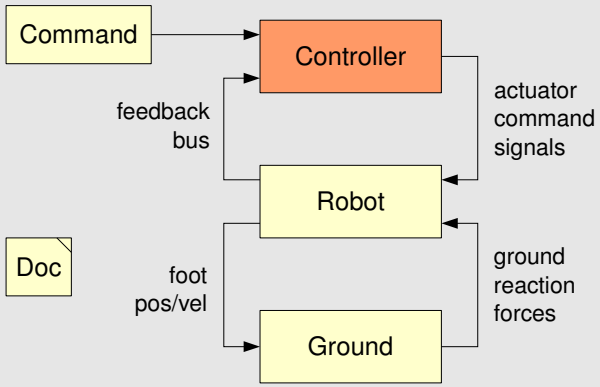
Top-Level Model



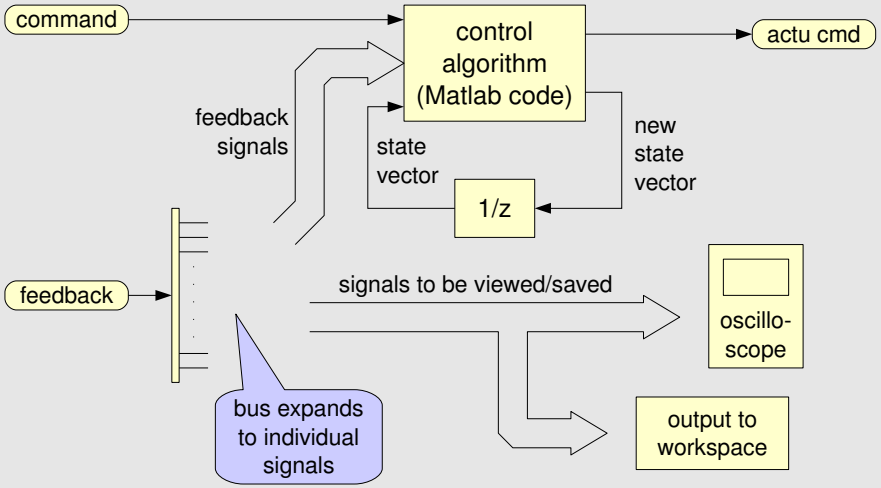
Top-Level Model



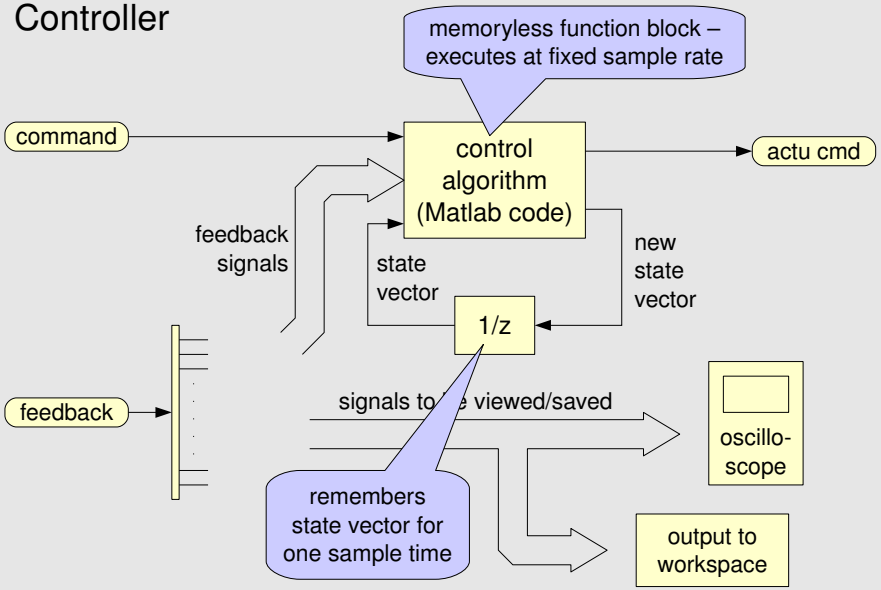
Controller



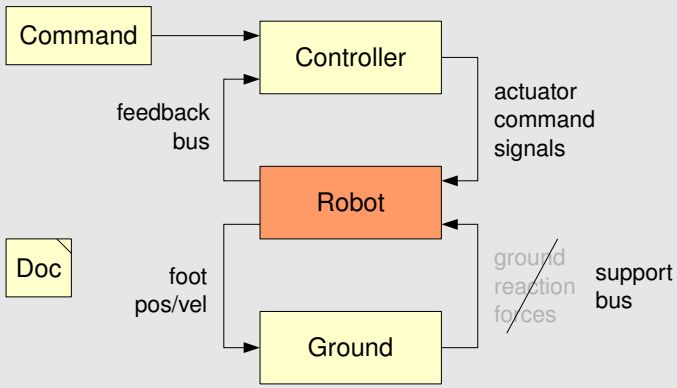
Controller



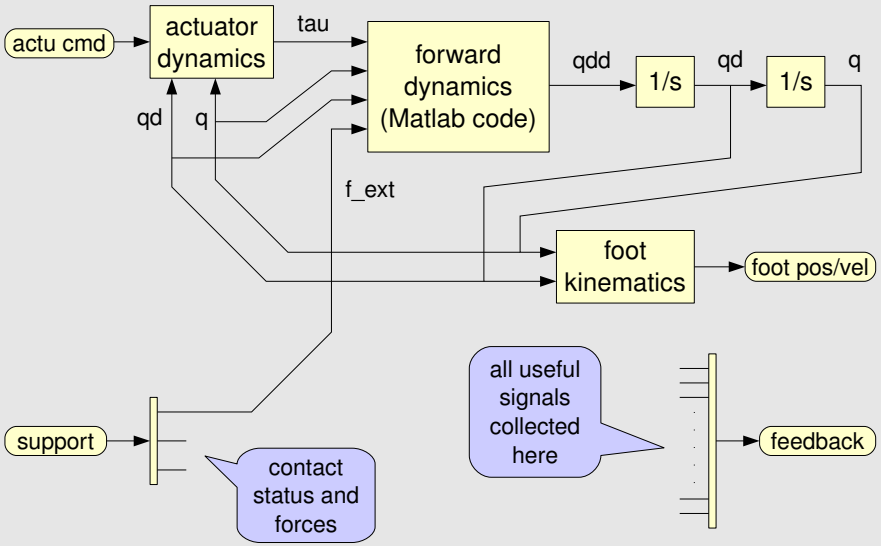
Controller



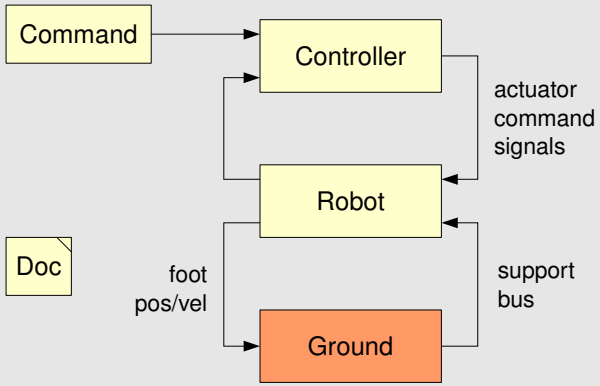
Robot



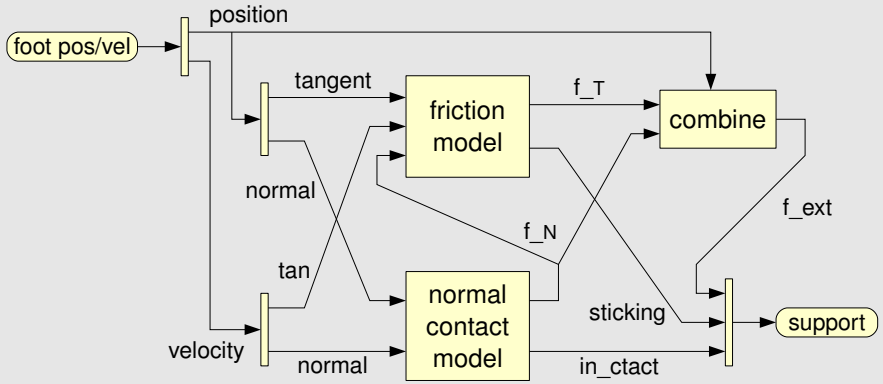
Robot



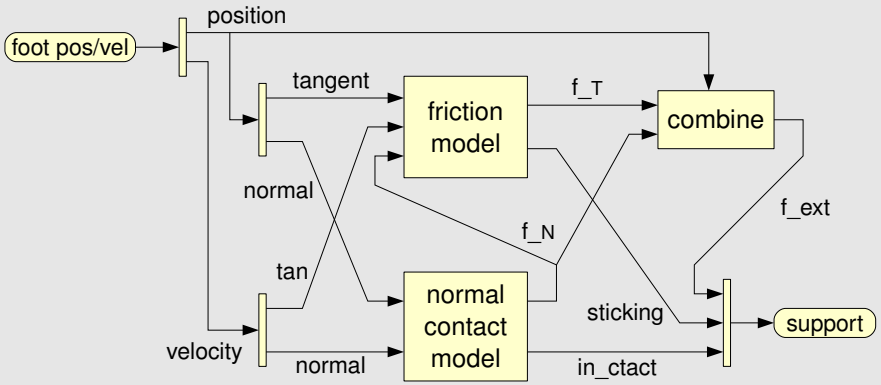
Ground



Ground

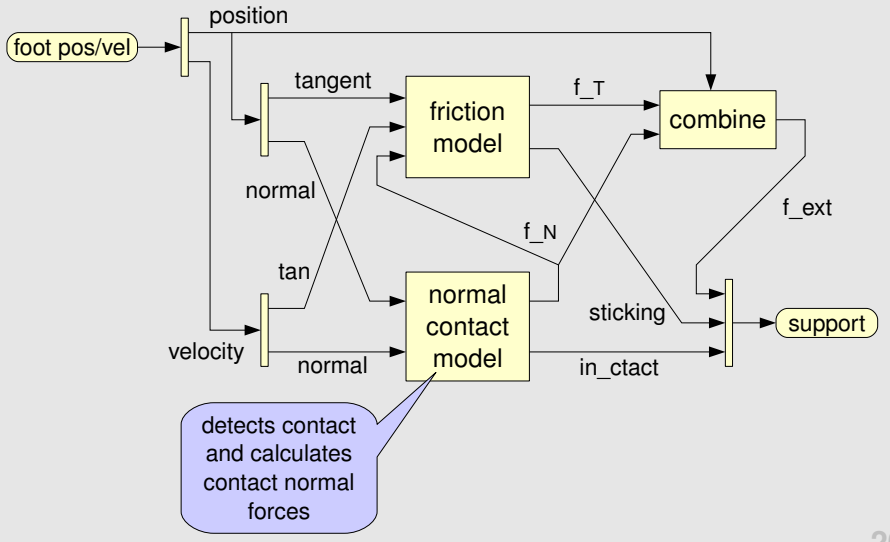


Ground

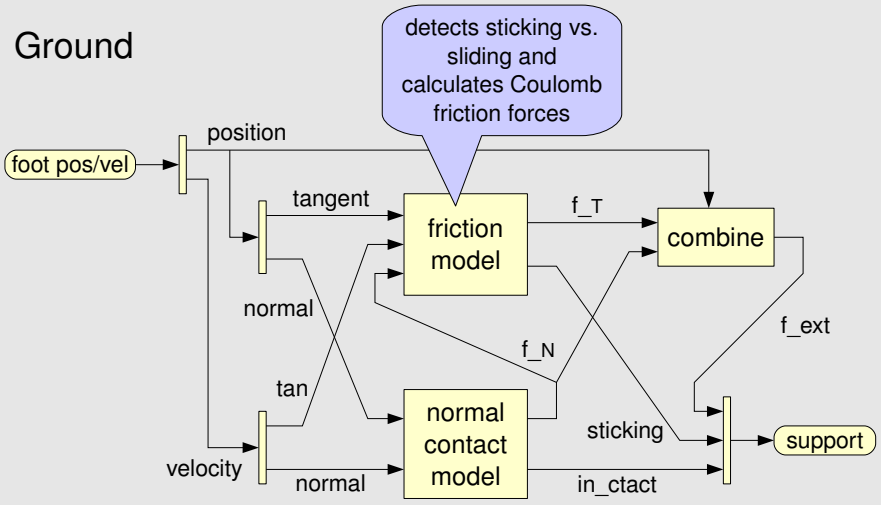


- a foot is the union of one or more points or spheres
- the model accepts data on multiple feet

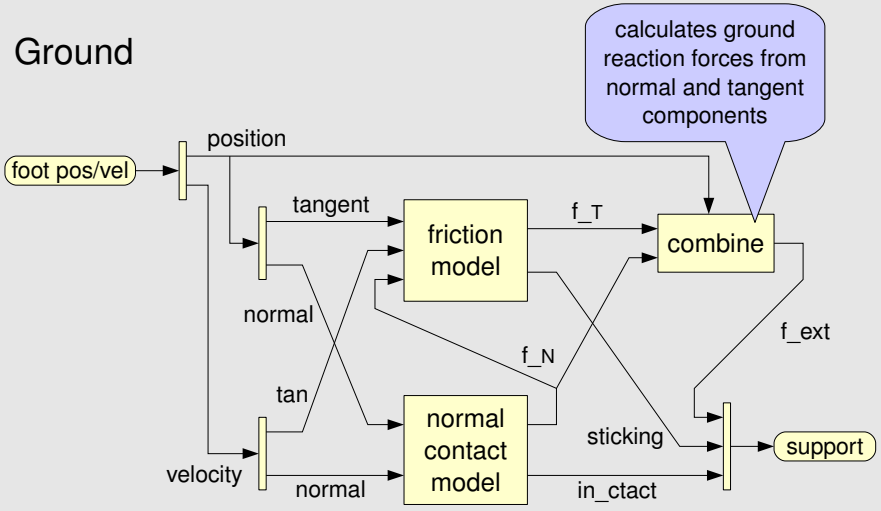
Ground



Ground



Ground



Results so Far

- students have implemented Raibert-style controllers on hoppers and runners in 2D and 3D
- new, improved contact normal model
- realistic models of lossy electric motors and drive trains
- 3D animation software for off-line analysis of simulation results
- S-functions (C/C++) for faster simulations

Future Work

- develop an *energy audit* facility to track all energy flows in a simulation
- use the software to support research in highly dynamic legged locomotion
- make the software available on the web