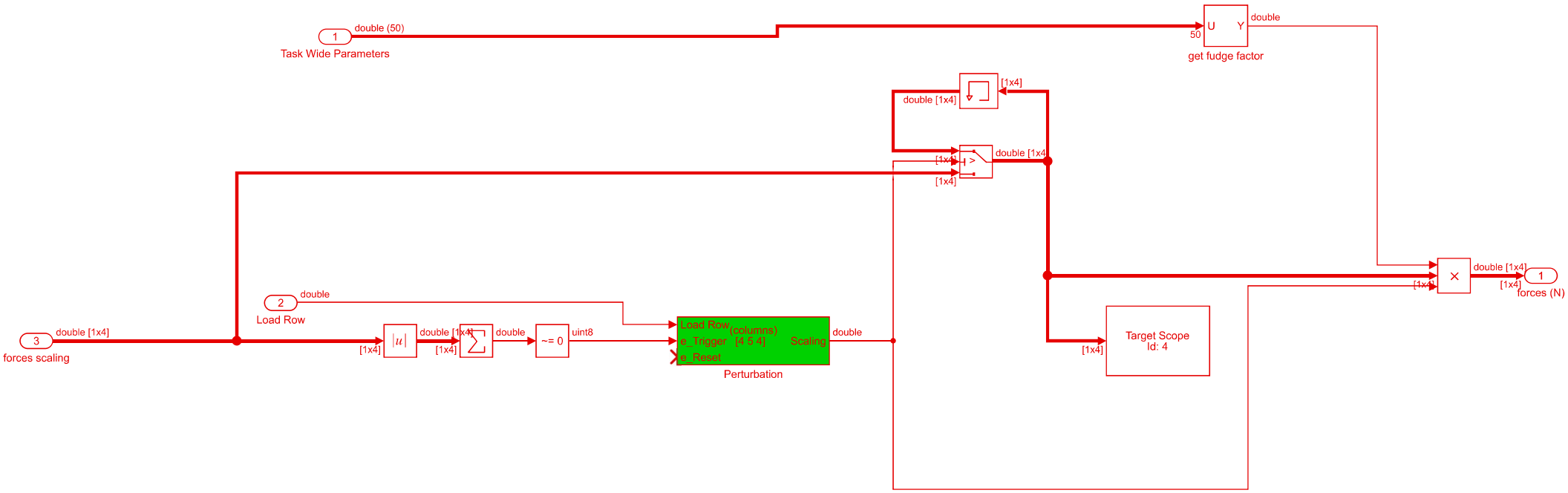
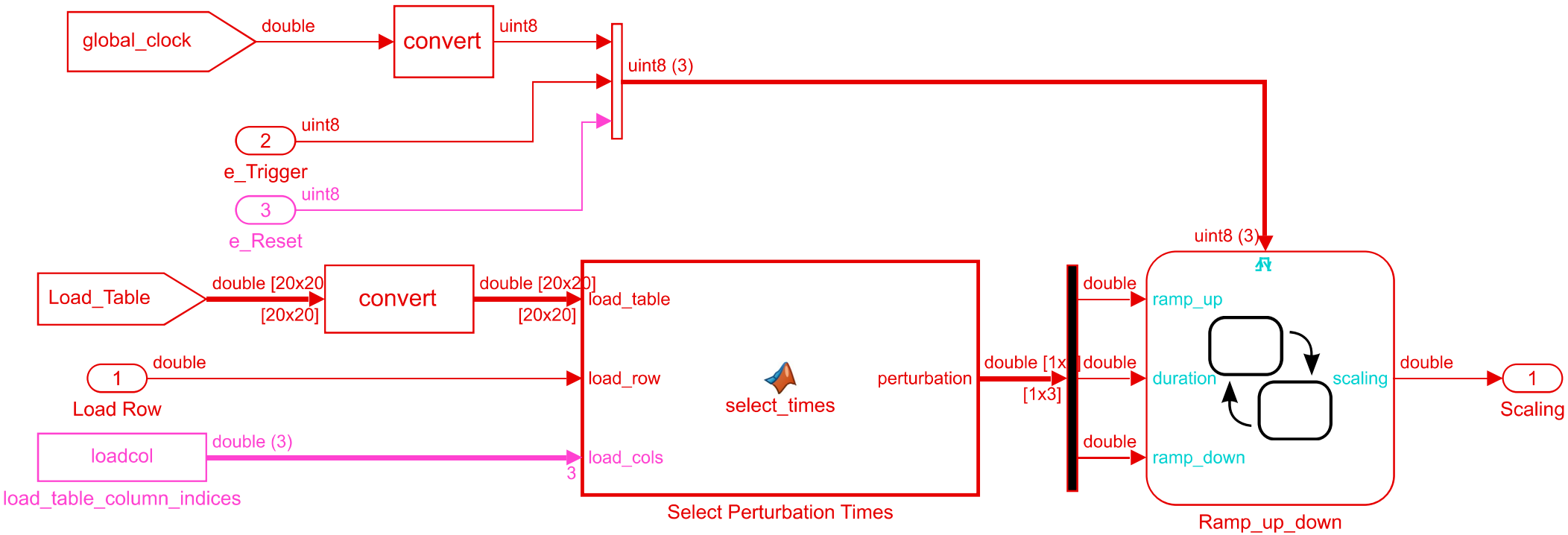
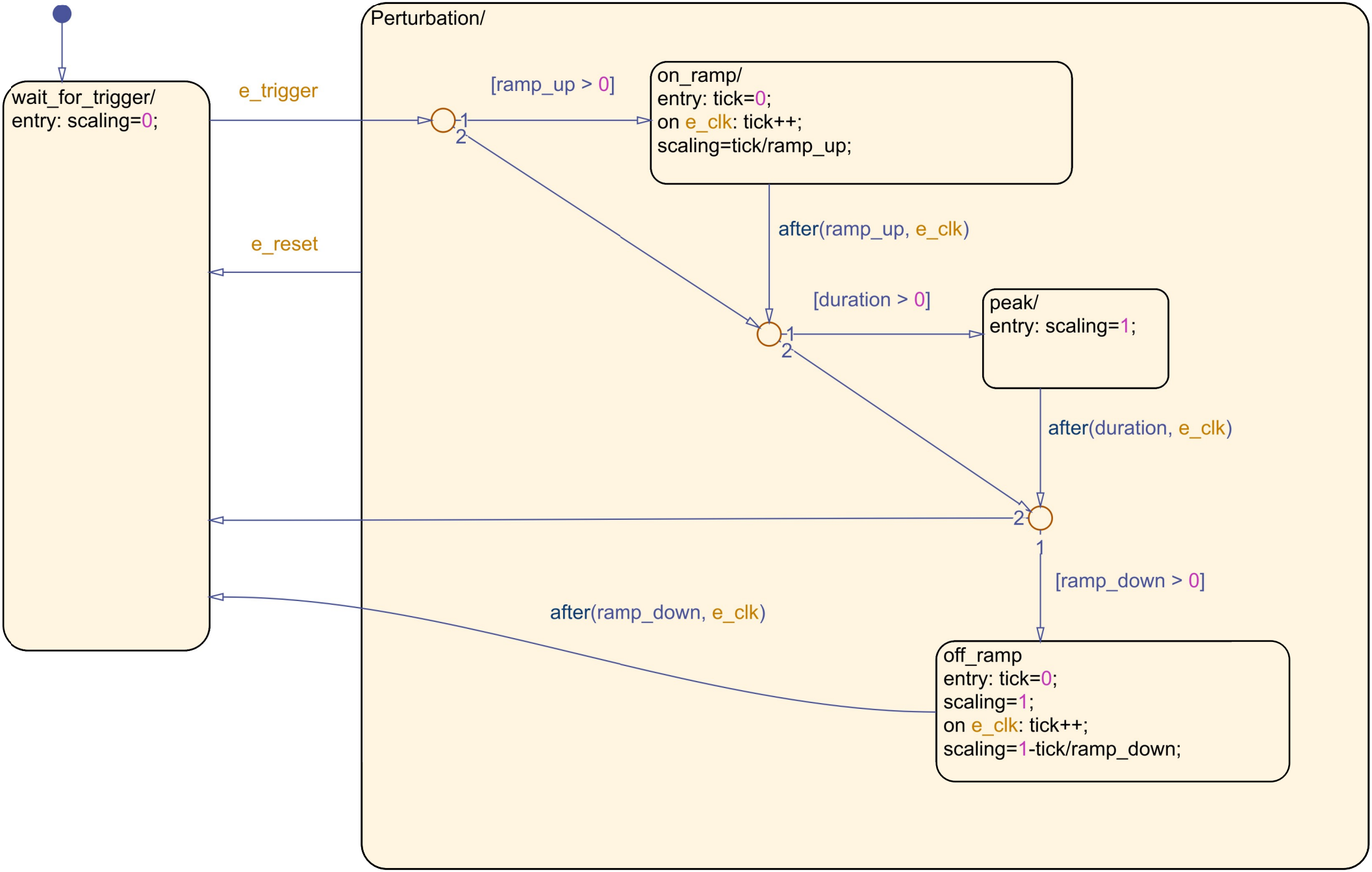
Creates the hand bump based on force scaling that comes from Collion Resolution

The basics here are:
-when force scaling changes from 0, a perturbation begins.
-the memory3 function holds the value of force_scaling at that moment until the perturbation is done
-the hand bump has a force value of: FORCE_MULTIPLIER * (force_scaling .* perturbation) N
-FORCE_MULTIPLIER is a fudge factor coming from Task Wide Parameters to make the bump a number of Newtons that feel nice






Perturbation/
wait_for_trigger/
entry: scaling=0;
e_trigger
e_reset
[ramp_up > 0]
on_ramp/
entry: tick=0;
on e_clk: tick++;
scaling=tick/ramp_up;
after(ramp_up, e_clk)
[duration > 0]
peak/
entry: scaling=1;
after(duration, e_clk)
[ramp_down > 0]
after(ramp_down, e_clk)
off_ramp
entry: tick=0;
scaling=1;
on e_clk: tick++;
scaling=1-tick/ramp_down;

```
function perturbation  = select_times(load_table, load_row, load_cols)

% Select perturbation ramp up, duration, and ramp down times from the
% from load table, or use 0 if an invalid load row and/or column is specified.

ramp_up = 0.0;
duration = 0.0;
ramp_down = 0.0;

if load_row > 0 && load_row <= size(load_table, 1)
    if load_cols(1) > 0
        ramp_up = load_table(load_row, load_cols(1));
    end
    if load_cols(2) > 0
        duration = load_table(load_row, load_cols(2));
    end
    if load_cols(3) > 0
        ramp_down = load_table(load_row, load_cols(3));
    end
end

perturbation = [ramp_up, duration, ramp_down];
```