# This function takes three inputs

x - a set of parameters t - the number of time-steps you wish to simulate

```
function f = policy_sliroutput_full(x,t)

% Here is a suggested framework for x.  However, you are free to
 deviate
% from this if you wish.

% set up transmission constants
k_infections = x(1);
k_fatality = x(2);
k_recover = x(3);
k_lockdown = x(4);
k_vac = .3;

% set up initial conditions
ic_susc = x(5);
ic_lockdown = x(6);
ic_inf = x(7);
ic_rec = x(8);
ic_fatality = x(9);

% Set up SIRD within-population transmission matrix
A = [1-(k_infections + k_lockdown + k_vac) 0.4 0 0 0;
    k_lockdown 0.6 0.5 0 0;
    k_infections 0 1-(0.5+k_fatality + k_recover) 0 0;
    k_vac 0 k_recover 1 0;
    0 0 k_fatality 0 1];


% The next line creates a zero vector that will be used a few steps.
B = zeros(5,1);

% Set up the vector of initial conditions
x0 = [ic_susc, ic_lockdown, ic_inf, ic_rec, ic_fatality];

A = sirpolicy(A, x0);
% Here is a compact way to simulate a linear dynamical system.
% Type 'help ss' and 'help lsim' to learn about how these functions
 work!!
sys_sir_base = ss(A,B,eye(5),zeros(5,1),1)
y = lsim(sys_sir_base,zeros(t,1),linspace(0,t-1,t),x0);

% return the output of the simulation
f = y;



end
```

```
Not enough input arguments.
```

```
Error in policy_sliroutput_full (line 11)
k_infections = x(1);
```

*Published with MATLAB® R2021a*