Caroline Chow– CE 291 F Final Project Presentation 12/04/07

EXTREMUM SEEKING USING A UNICYCLE MODEL

EXTREMUM SEEKING

- Miroslav Krstic

EXTREMUM SEEKING

- Control technique

- Goal:
  - Force an output to its maximum

- Also referred as SOURCE SEEKING

- Applications:
  - Track an object emitting a signal
  - Find a contaminant…

SIGNAL MODEL

- The signal model is NOT known.

- Non-linear distribution on the (x,y) plane:

$$T = f(x,y)$$

- Signal strength decays away from a source
  - e.g. diffusion phenomenon

HEAT EQUATION

- 1D HEAT EQUATION

$$\frac{\partial T}{\partial t} = D\frac{\partial^2 T}{\partial x^2}$$

or

$$\frac{\partial T}{\partial t} = \Delta T$$

HEAT EQUATION

- Deriving:
  - Plugging Fourier's law

$$q = -K\frac{\partial T}{\partial x}$$

  - Into the continuity equation

$$c_p\frac{\partial T}{\partial t} = -\frac{\partial q}{\partial x}$$

HEAT EQUATION

- Solving:
  - Using the self-similar variable:

$$\eta = \frac{Ax}{t^n} \qquad \text{and} \qquad T(x,t) \equiv f(\eta)$$

HEAT EQUATION

- Solving:
  - Using the self-similar variable:

$$\eta = \frac{Ax}{t^n} \qquad \text{and} \qquad T(x,t) \equiv f(\eta)$$

  - and artistic choices…

$$A = 1/\sqrt{2D} \qquad \text{and} \qquad n = 0.5$$

HEAT EQUATION

- Solving:
  - Using the self-similar variable:

$$\eta = \frac{Ax}{t^n} \qquad \text{and} \qquad T(x,t) \equiv f(\eta)$$

  - and artistic choices…

$$A = 1/\sqrt{2D} \qquad \text{and} \qquad n = 0.5$$

  - Get the 2nd order ODE:

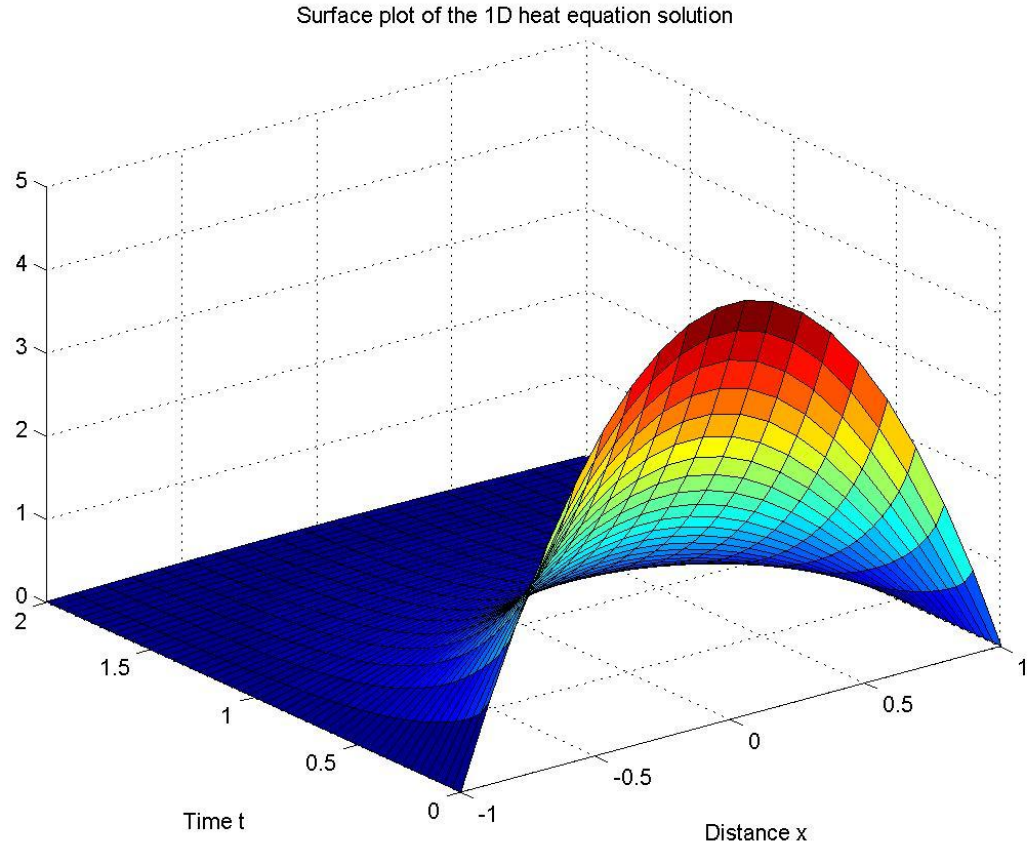
$$f''(\eta) + \eta f'(\eta) = 0$$

HEAT EQUATION

- MATLAB solving: using pdepe.m

HEAT EQUATION

- MATLAB solving: using pdepe.m



2-D DIFFUSION EQUATION

- $$\frac{\partial T}{\partial t} = \Delta T \qquad \text{or} \qquad \frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}$$

2-D DIFFUSION EQUATION

- $$\frac{\partial T}{\partial t} = \Delta T$$ or $$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}$$
- No MATLAB solver in the lab

2-D DIFFUSION EQUATION

- $$\frac{\partial T}{\partial t} = \Delta T \qquad \text{or} \qquad \frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}$$
- No MATLAB solver in the lab
- STEADY STATE SOLUTION

$$0 = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}$$

2-D DIFFUSION EQUATION

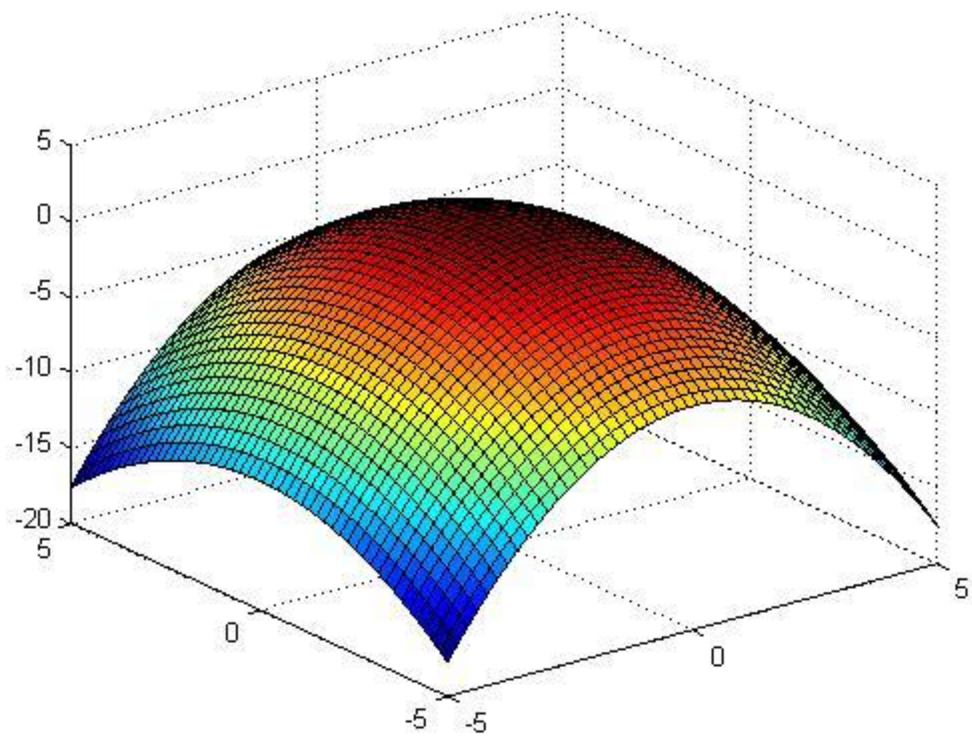
- STEADY STATE SOLUTION

$$T(x,y) = T^{*} - a(x - x^{*})^{2} - b(y - y)^{2}$$

2-D DIFFUSION EQUATION

- STEADY STATE SOLUTION

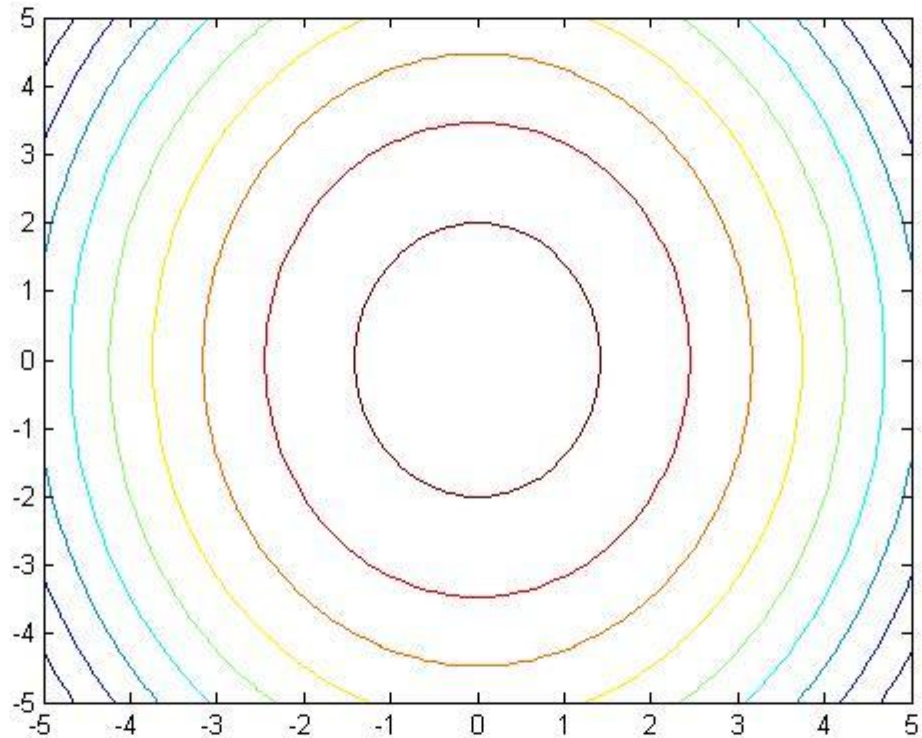
$$T(x,y) = T^* - a(x - x^*)^2 - b(y - y)^2$$



2-D DIFFUSION EQUATION

- STEADY STATE SOLUTION

$$T(x,y) = T^* - a(x - x^*)^2 - b(y - y)^2$$



VEHICLE MODEL

- UNICYLE

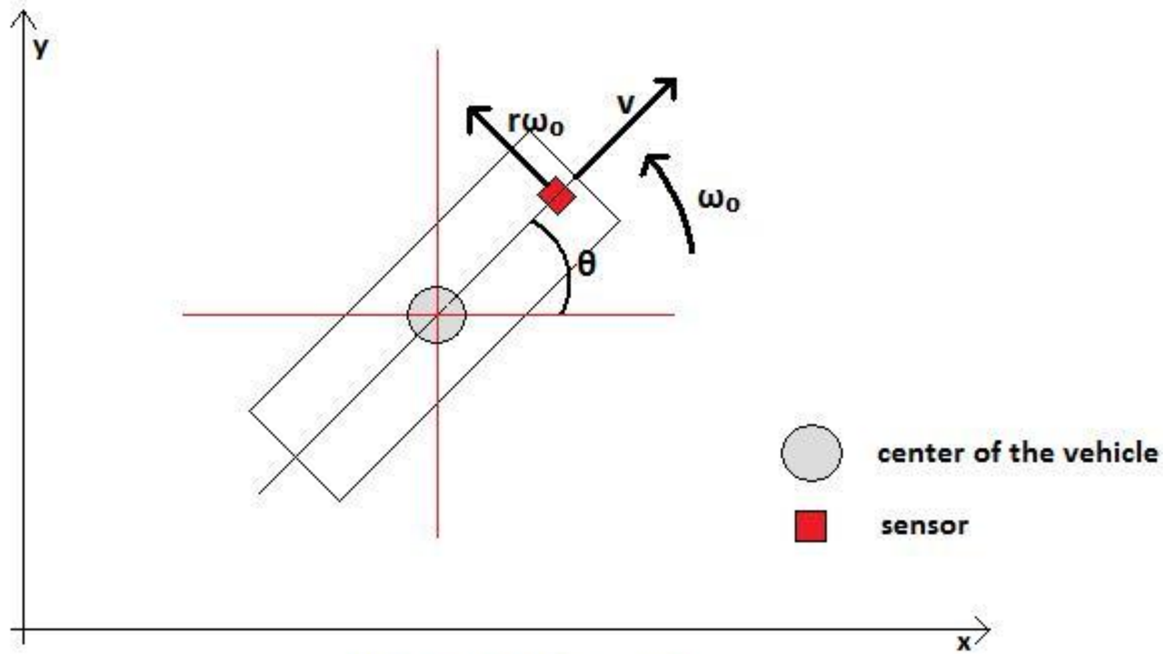


Figure 2 - Vehicle model

VEHICLE MODEL

- MOTION EQUATIONS
  - center of the vehicle:

$$x'_C = v\cos\theta$$
$$y'_C = v\sin\theta$$
$$\theta' = \omega_0$$

VEHICLE MODEL

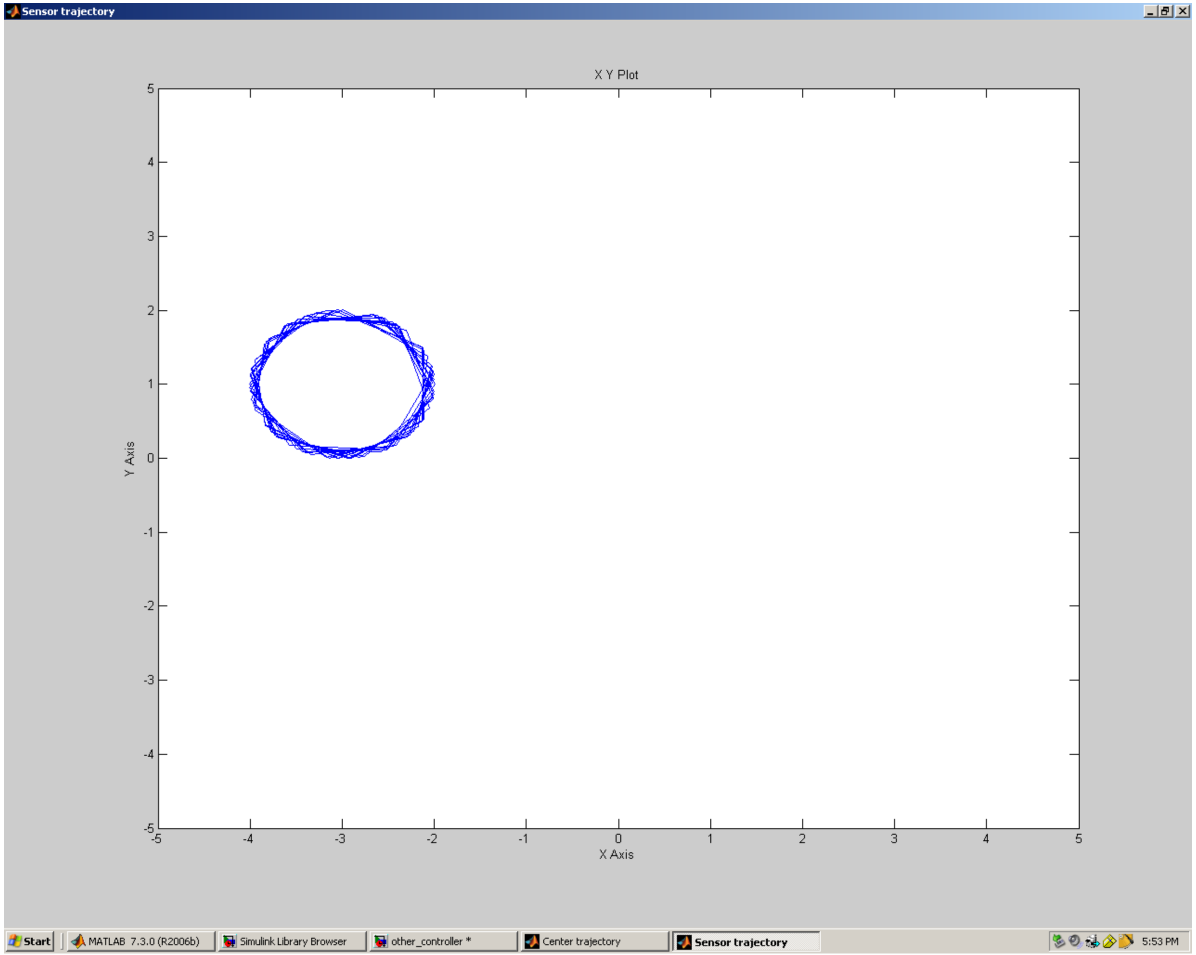
- MOTION EQUATIONS
  - center of the vehicle:

$$x_C' = v\cos\theta$$
$$y_C' = v\sin\theta$$
$$\theta' = \omega_0$$

  - sensor:

$$x_s = x_0 + r\,cos\,\theta$$
$$y_s = y_0 + r\,sin\,\theta$$

VEHICLE MOTION



EXTREMUM SEEKING ALGORITHM

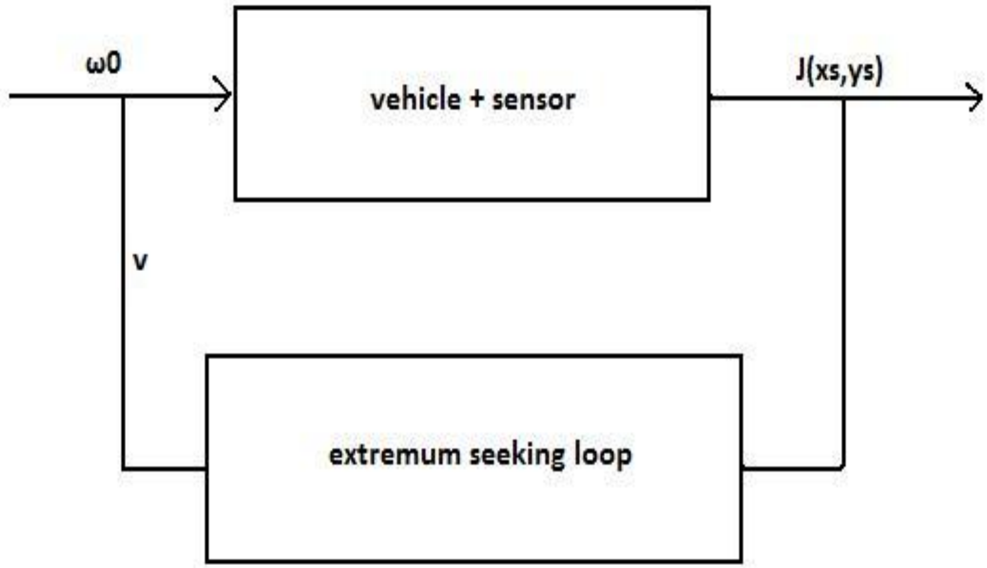


Figure 3. Very basic feedback controller

EXTREMUM SEEKING ALGORITHM

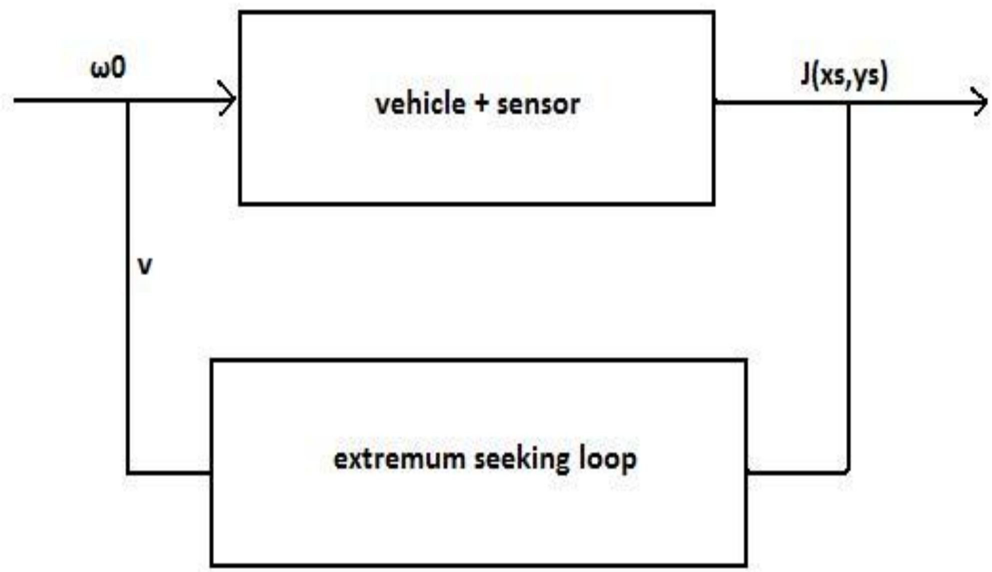


Figure 3. Very basic feedback controller

- Fixed angular velocity/ turn rate

EXTREMUM SEEKING ALGORITHM

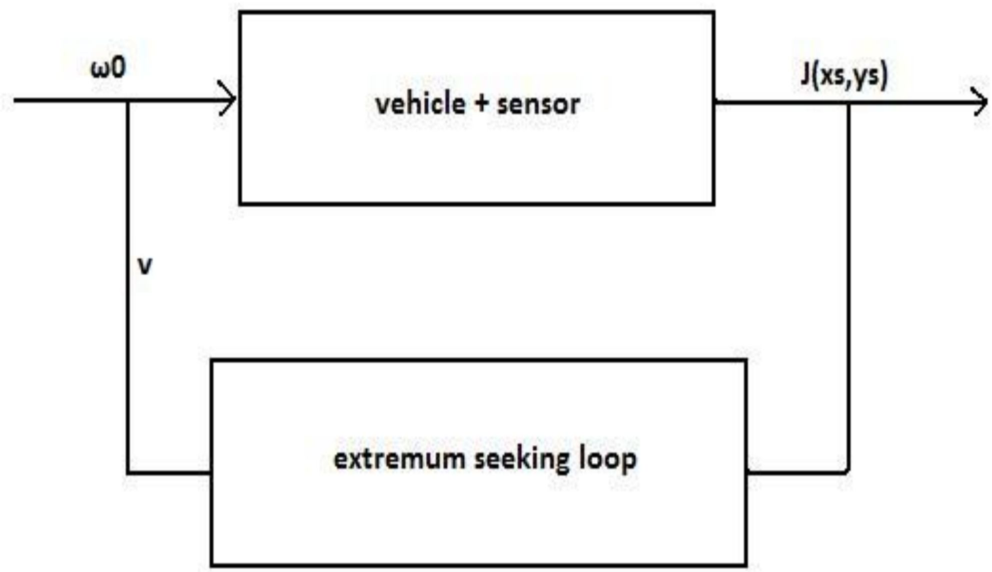


Figure 3. Very basic feedback controller

- Fixed angular velocity/ turn rate
  - OUPUT: VALUE OF SIGNAL

EXTREMUM SEEKING ALGORITHM

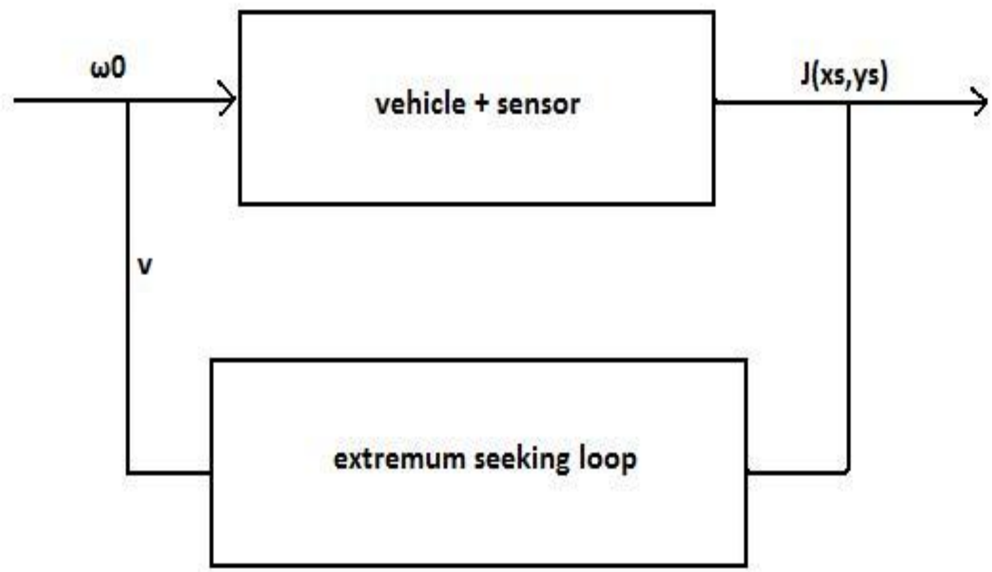
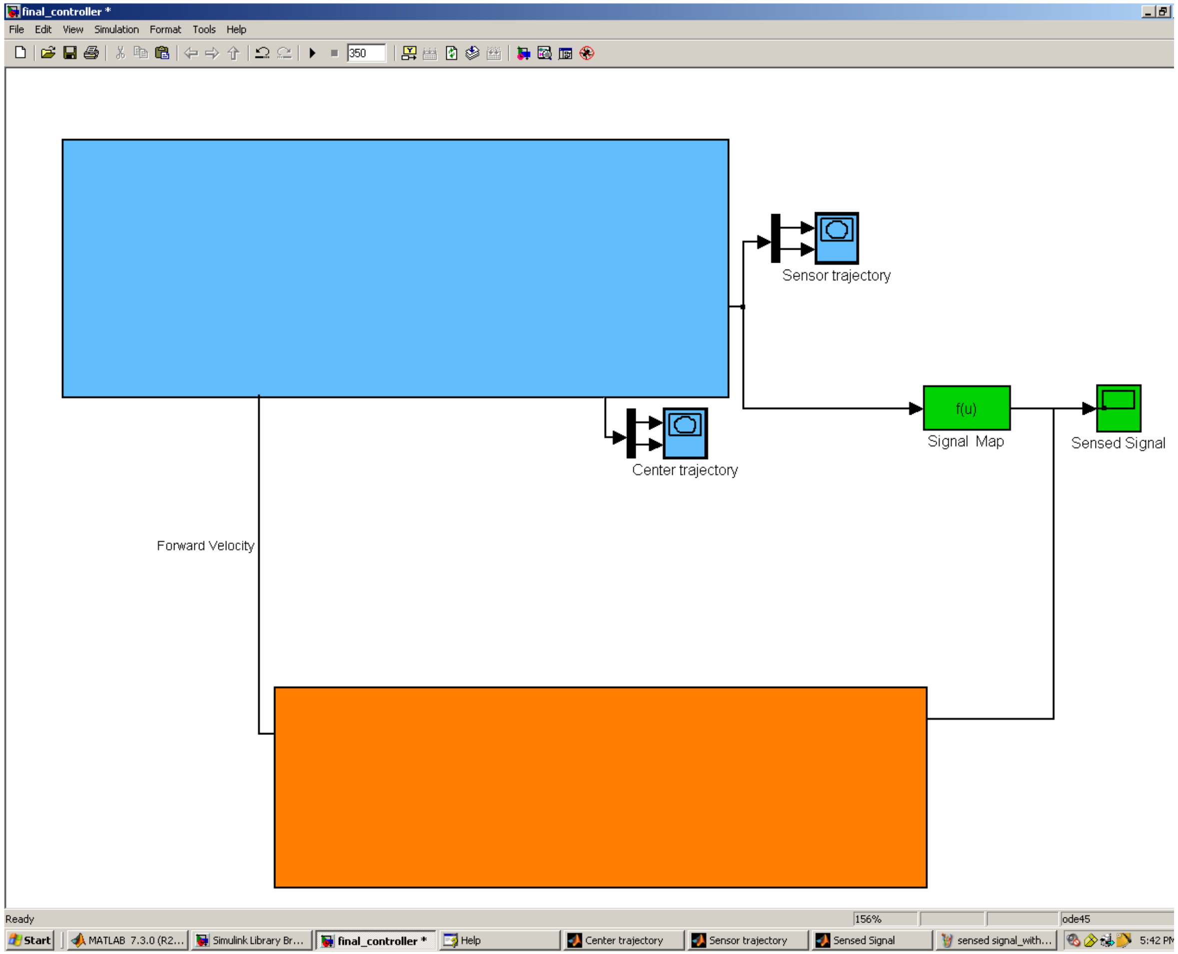


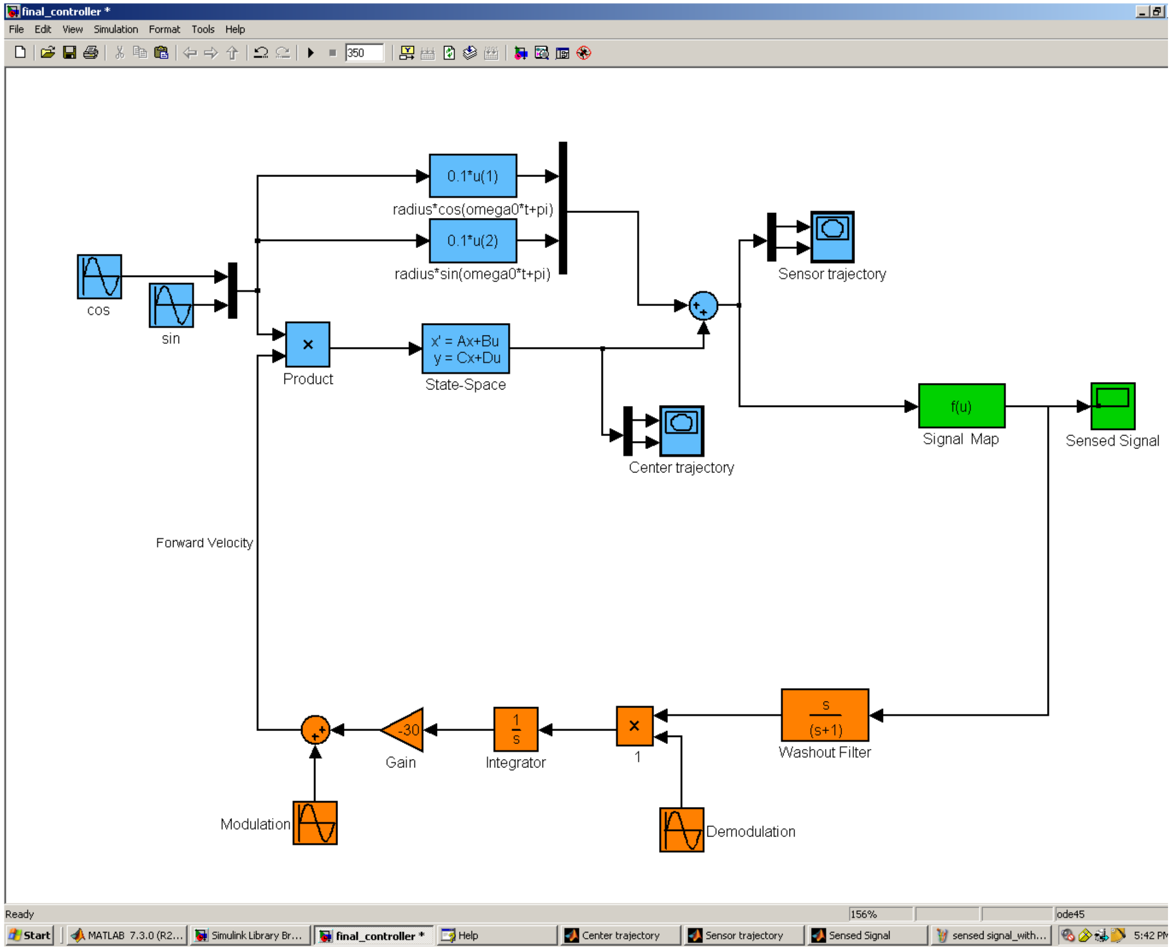
Figure 3. Very basic feedback controller

- Fixed angular velocity/ turn rate
- OUPUT: VALUE OF SIGNAL
- CONTROL/INPUT: FORWARD VELOCITY

EXTREMUM SEEKING ALGORITHM



EXTREMUM SEEKING ALGORITHM

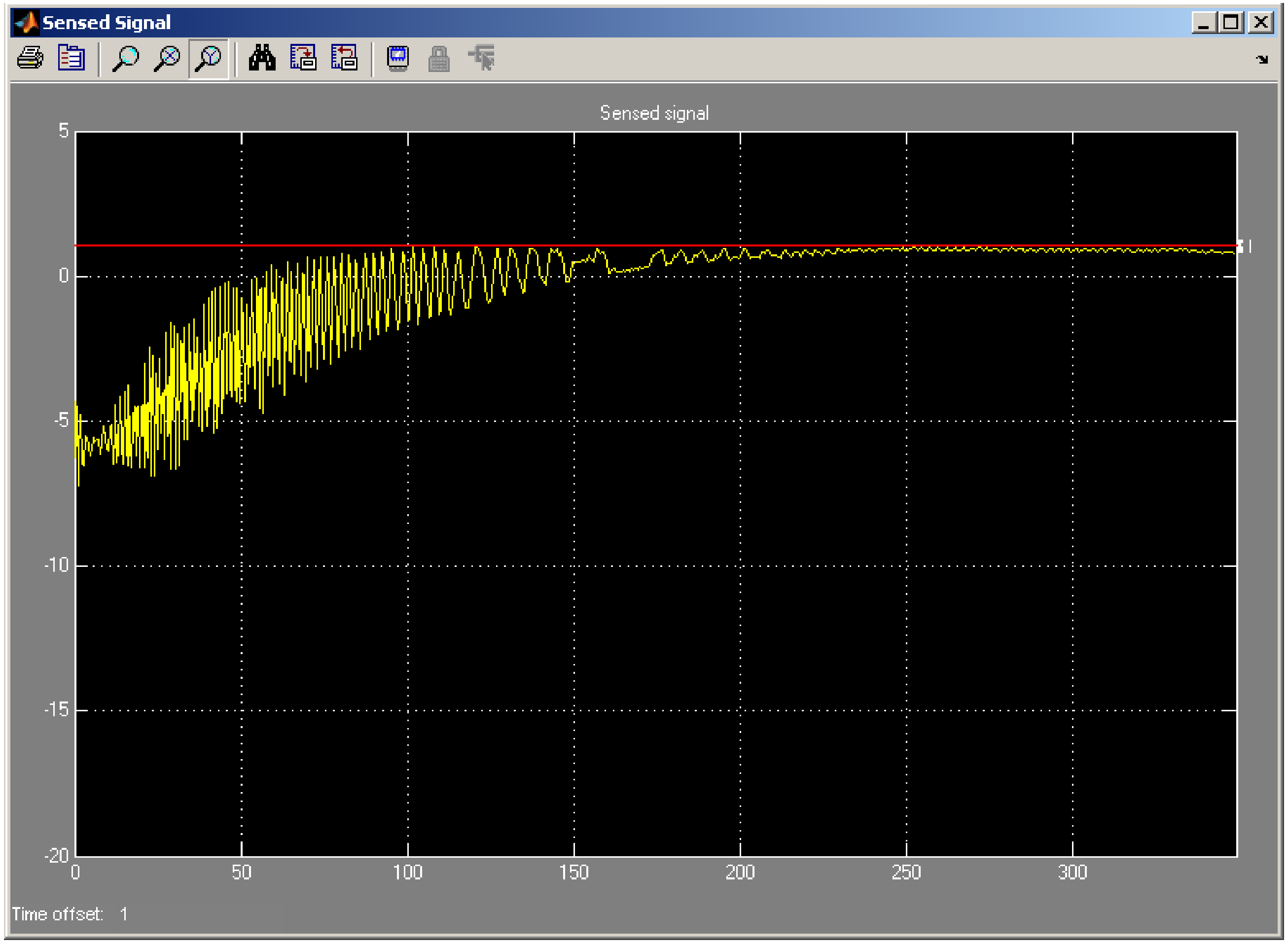


EXTREMUM SEEKING PARAMETERS

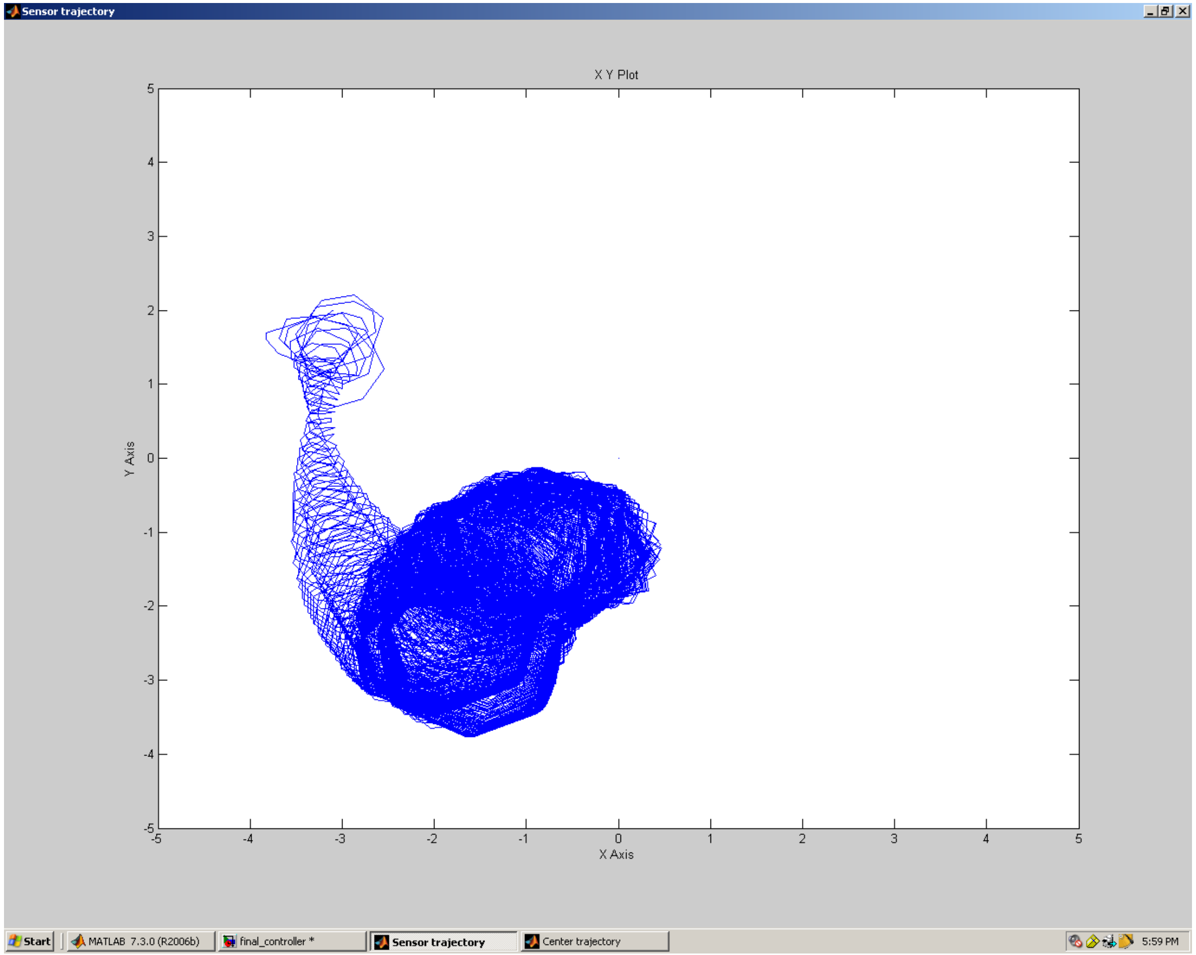
- SIMULINK IMPLEMENTATION with:
  - Signal & domain
  - Initial coordinates and heading
  - Angular velocity and radius
  - Perturbation frequency

EXTREMUM SEEKING CONVERGENCE

EXTREMUM SEEKING CONVERGENCE



VEHICLE MOTION



NEXT STEPS

- Get a pro version of MATLAB
- Moving source

NEXT STEPS

- Get a pro version of MATLAB
- Moving source