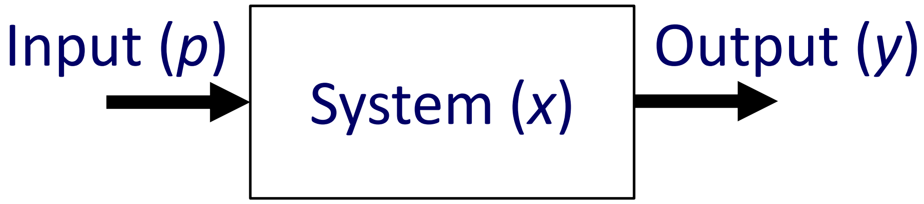
Part I: Dynamic Modeling

$$\text{minimize} \quad J(x, y, p)$$

$$\textit{subject to} \quad 0 = f\left(\frac{dx}{dt}, x, y, p\right)$$

$$0 \leq g\left(\frac{dx}{dt}, x, y, p\right)$$



Empirical | Hybrid | Fundamental

Data Regression
- Artificial Neural Networks
- Linear State Space Identification

Combination
- Parameter Estimation
- Linearized Fundamental Model

First Principles
- Molecular Dynamics
- Mass/Energy Balance

Automobile Velocity from Force Balance

- Dynamic Modeling
  - Velocity (*v (m/s)*)
  - Gas Pedal (*p (%)*)
  - Gain (*K*)
  - Time Constant (τ)

Input
(Gas Pedal)

Output
(Velocity)

First Order Linear System

$$\tau \frac{\partial v}{\partial t} = -v + Kp$$

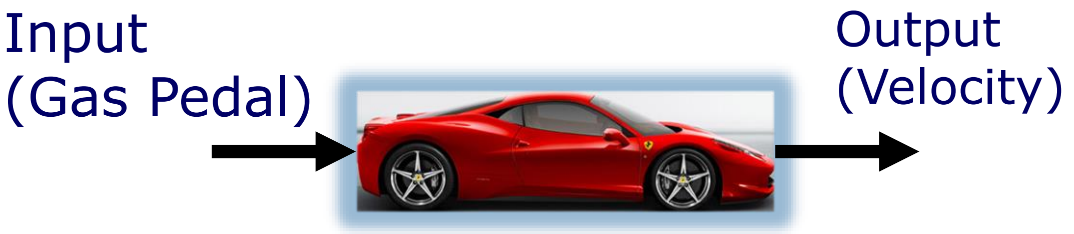
Input = P(s)

$$\frac{K}{\tau s + 1}$$

Output = V(s)

Dynamic Model from Fundamentals

- v = 25 m/s (56 mph) = Desired Velocity
- m = 500 kg (mass)
- b = 50 N-s/m (resistive coefficient)
- K = 1.0 m/s / (% gas pedal)
- p = ? (% gas pedal position)



Force Balance

$$\frac{m}{b}\frac{\partial v}{\partial t} = -v + K\,p$$

→

Linear First-Order Model

$$\tau\frac{\partial v}{\partial t} = -v + Kp$$

Dynamic Modeling, APMonitor Model

```
Constants
  m = 500 ! Mass (kg)

Parameters
  b = 20  ! Resistive coefficient (N-s/m)
  K = 0.8 ! Gain (m/s-%pedal)
  p = 0 >= 0 <= 100  ! Gas pedal position (%)

Variables
  v = 0

Equations
  m/b * $v = -v + K * p
```

$$\frac{m}{b}\frac{\partial v}{\partial t} = -v + K\ p$$

Dynamic Modeling, Solve with MATLAB/Python

MATLAB

```matlab
clear all; close all; clc % clear session
addpath('apm') % load APMonitor.com toolkit
y = apm_solve('ferrari'); z = y.x; % solve

% plot results
figure(1)

subplot(2,1,1)
plot(z.time,z.p,'r-','LineWidth',2)
legend('Pedal')
ylabel('Position (%)')

subplot(2,1,2)
plot(z.time,z.v,'b.-','LineWidth',2)
legend('Velocity')
ylabel('Velocity (m/s)')
xlabel('Time (sec)')
```

Python

```python
from apm import *  # load APMonitor.com toolkit
z = apm_solve('ferrari',7) # solve

# plot results
import matplotlib.pyplot as plt
plt.figure()

plt.subplot(211)
plt.plot(z['time'],z['p'],'r-')
plt.legend(['Pedal'])
plt.ylabel('Position (%)')

plt.subplot(212)
plt.plot(z['time'],z['v'],'b.-')
plt.legend(['Velocity'])
plt.ylabel('Velocity (m/s)')
plt.xlabel('Time (sec)')
plt.show()
```

Dynamic Modeling