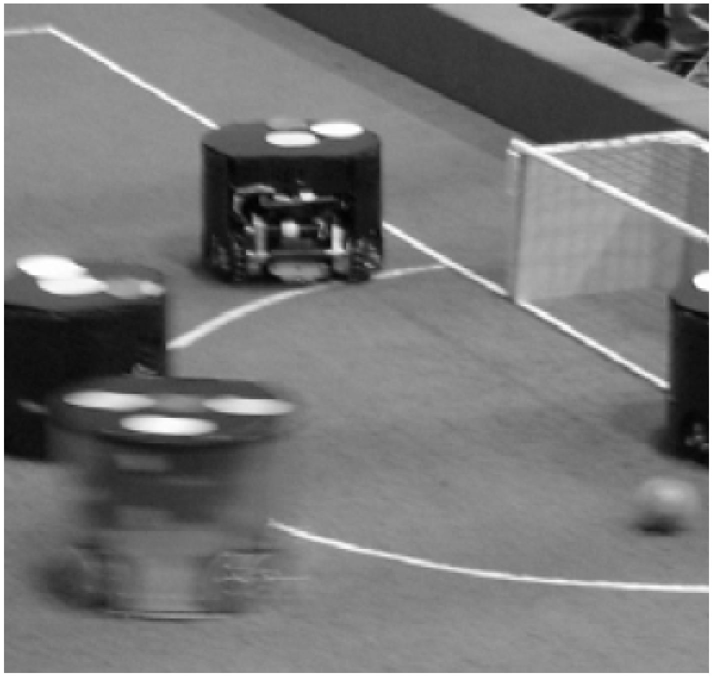


# Chapter 4
# Dynamics

## Part 3

4.4 Aspects of Linear Systems Theory

4.5 Predictive Modelling and System Identification

# Introduction

- Nonlinear dynamical systems are the closest thing to the engineering "theory of everything."
- Applies to:
  - growth of bacteria
  - chemical reactions
  - financial markets
  - motion of the planets
- Most important and general model of a mobile robot.

# Outline

- 4.3 Aspects of Linear Systems Theory
  - 4.4.1 Linear Time Invariant Systems
  - 4.3.2 State Space Representation of Linear Systems
  - 4.3.3 Nonlinear Dynamical Systems
  - 4.3.4 Perturbative Dynamics of Linear Systems
  - Summary
- 4.5 Predictive Modelling and System Identification

# Linear Time Invariant ODE s

- These are of the form:

"Forcing Function"
"Input"
"Control"

$$\frac{d^{(n)}y}{dt^n} + a_{n-1}\frac{d^{(n-1)}y}{dt^{n-1}} + \ldots + \frac{dy}{dt} + y = u(t)$$

- Establishes a relationship between system state x(t) and its derivatives.
  - Implies that such a system will move (even when u(t) is not present)
  - Called a dynamical system

# First Order System

- Behavior governed by:

"Time Constant"

$$\tau\frac{dy}{dt} + y = u(t)$$

- Consider the discrete time equivalent:

$$\frac{\tau}{\Delta t}(y_{k+1} - y_k) + y_{k+1} = u_{k+1}$$

$$y_{k+1} = y_k + \frac{\Delta t}{\tau}(u_{k+1} - y_{k+1})$$

- Hence output changes by an amount proportional to the distance-to-go.

# Step Response

$$\tau\frac{dy}{dt} + y = u(t)$$

- Useful to describe behavior of a few special inputs.
- Step response is response to constant input applied for t >= 0.
- Unforced response. Assume $y(t) = e^{st}$
- Substitute into ODE: $e^{st}(\tau s + 1) = 0$
- Characteristic equation: $\tau s + 1 = 0$
- The roots of this equation play a crucial role in determining system behavior.

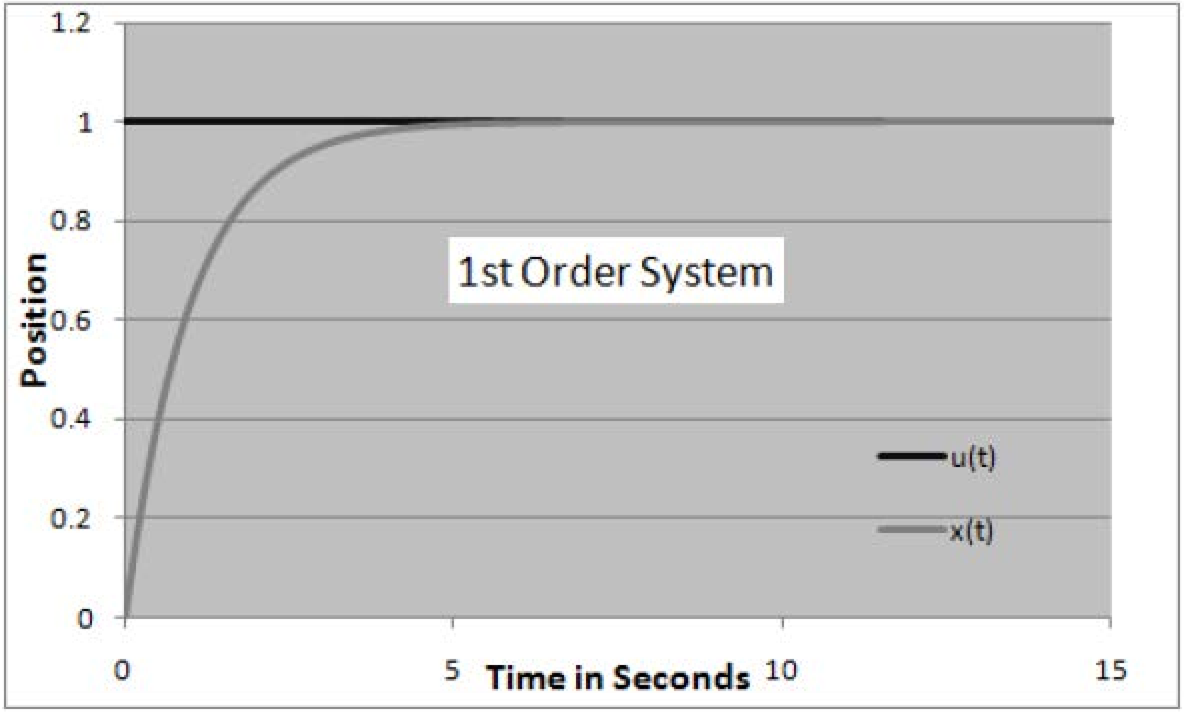
$$s = -1/\tau$$

# Solution

- Unforced solution: $y(t) = Ae^{-t/\tau}$
- Forced solution: $y(t) = y_{ss}$
- Complete solution: $y(t) = Ae^{-t/\tau} + y_{ss}$
- For $y(0) = 0$ we must have $A = -y_{ss}$
- Total Solution:

$$y(t) = y_{ss}(1 - e^{-t/\tau})$$

# Solution




- When t=τ, the system has moved …

$$1 - 1/e \text{ or } 63\%$$

- … of the total distance to the goal.

# Laplace Transform

- Extraordinarily powerful for manipulating compounded ODEs intuitively.
- Definition:

$$y(s) = \mathcal{L}[y(t)] = \int_0^{\infty} e^{-st} y(t) dt$$

- s is a "complex frequency"

$$s = \sigma + j\omega$$

- The kernel is a damped sinusoid:

$$e^{-st} = e^{-\sigma t} e^{-j\omega t} = e^{-\sigma t}[\cos(\omega t) - j\sin(\omega t)]$$

# Laplace Transform

- For a particular value of s:

$$y(s) = \mathcal{L}[y(t)] = \int_0^{\infty} e^{-st} y(t) dt$$

- is a (function) dot product with a damped sinusoid.
- y(s) encodes the projections for every value of s.
- Its just like a Fourier transform but for complex frequency s.

# Derivatives

- Most important property for our purpose:

$$\mathcal{L}[\dot{y}(t)] = s\mathcal{L}[y(t)] = sy(s)$$

- Good news!
  - Differentiation in the time domain is equivalent to multiplication by s.
- Bad news!
  - This is why differentiation amplifies noise.

# Transforming ODEs

- Recall the first order system ODE

$$\tau \frac{dy}{dt} + y = u(t)$$

- Transform the ODE itself:

$$\tau s y(s) + y(s) = u(s)$$

ODEs in the time domain become **algebraic eqns** in the Laplace domain

# Transfer Function

- Defined as the ratio of output to input:

$$T(s) = \frac{u(s)}{y(s)} = \frac{1}{1 + \tau s}$$

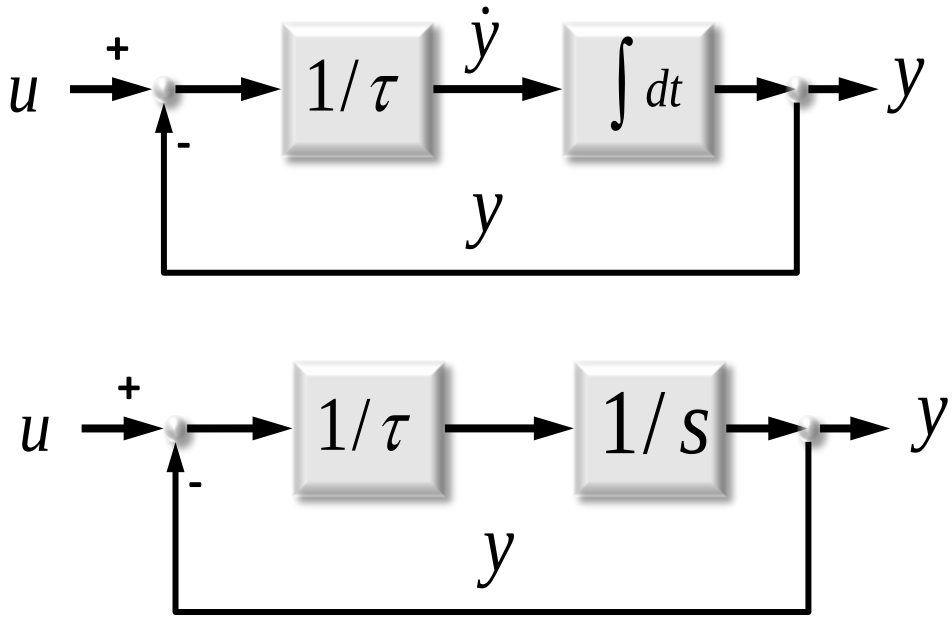
Characteristic polynomial!

- The roots of the characteristic polynomial always appear in the denominator of the transfer function.
- Known as the poles of the system.
- An n-th order ODE has n poles.

# Block Diagrams

- ODEs can be represented graphically as block diagrams.

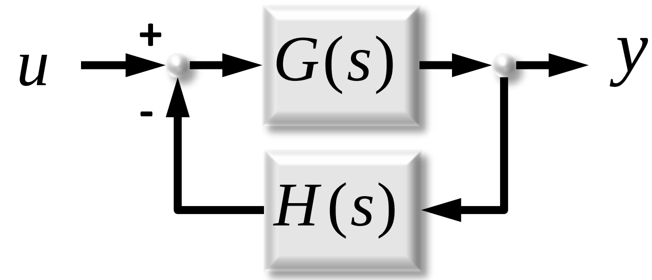
$$\tau\frac{\mathrm{d}y}{\mathrm{d}t} + y = u(t)$$



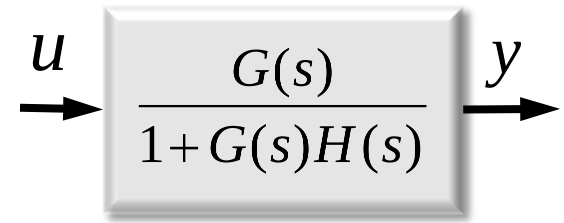

- Top is time domain, bottom is Laplace domain.

# Special Block Diagram

- This diagram:




- Is equivalent to this diagram




- Derivation:

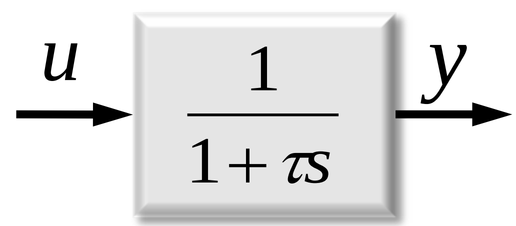
$$y(s) = G(s)(u(s) - H(s)y(s))$$

- So…

$$y(s)(1 + G(s)H(s)) = G(s)u(s)$$

$$T(s) = \frac{G(s)}{1 + G(s)H(s)}$$

- For the 1st order system:



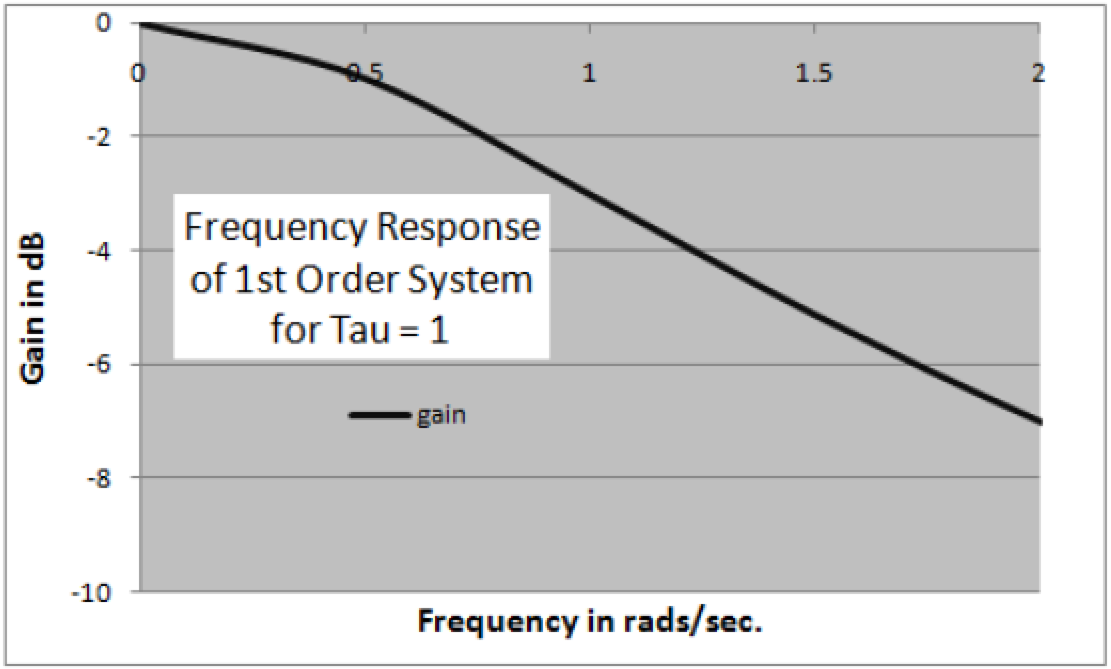
# Frequency Response

- Expresses the gain of the transfer function as function of frequency:
- Substitute into T(s): $s = j\omega$
- For 1st order system:

$$T(j\omega) = \frac{u(j\omega)}{x(j\omega)} = \frac{1}{1+\tau(j\omega)}$$

$$|T(j\omega)| = \frac{1}{|1+\tau(j\omega)|} = \boxed{\frac{1}{\sqrt{1+(\tau\omega)^2}}}$$

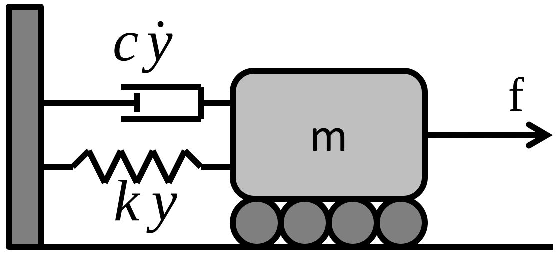
# Frequency Response




- Huh? Decibels?
  - dB = 20 log10(amplitude) = 10 log10(power)

# Second Order System

- One physical manifestation is a damped oscillator:




- Newton's second law:

$$m\ddot{y} = f - c\dot{y} - ky$$

- Rewrite:

$$\ddot{y} + \frac{c}{m}\dot{y} + \frac{k}{m}y = \frac{f}{m}$$ Physicists form

$$\ddot{y} + 2\zeta\omega_0\dot{y} + \omega_0^2 y = u(t)$$ Mathematicians form

# Simulation

- Simulate with:

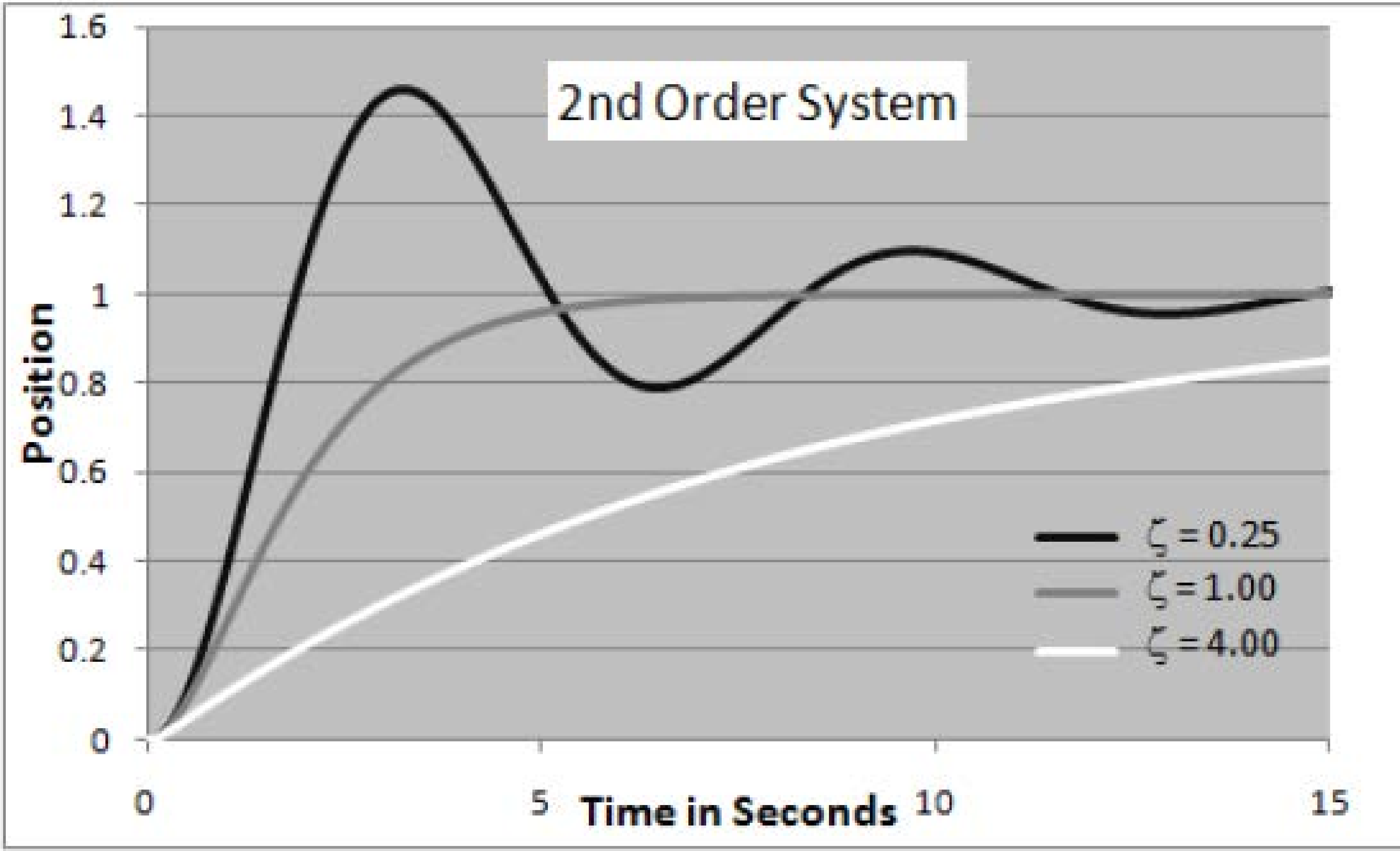
$$\ddot{y}_{k+1} = u_{k+1} - 2\zeta\omega_0\dot{x}_k - \omega_0^2 x_k$$

$$\dot{y}_{k+1} = \dot{y}_{k+1} + \ddot{y}_{k+1}\Delta t$$

$$y_{k+1} = y_{k+1} + \dot{y}_{k+1}\Delta t$$

- Truthoid: You can teach yourself controls if you can write a dynamic simulator like the above.

# 2nd Order Step Response



# 2nd Order Step Response

- Take Laplace transform of 2nd order ODE:

$$\ddot{y} + 2\zeta\omega_0\dot{y} + \omega_0^2 y = u(t)$$

$$s^2 y(s) + 2\zeta\omega_0 s y(s) + \omega_0^2 y(s) = u(s)$$

- Transfer function:

$$T(s) = \frac{y(s)}{u(s)} = \frac{1}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

- Behavior depends on the roots of the characteristic equation.

$$s = -\zeta\omega_0 \pm \omega_0\sqrt{\zeta^2 - 1} = \boxed{\omega_0(-\zeta \pm \sqrt{\zeta^2 - 1})}$$

# General 1st Order Solution

- For the more general 1st order time-varying system: $\dot{x}(t) = f(t)x(t) + g(t)u(t)$
- The following integrating function exists:

$$\phi(t, t_0) = \exp\left[\int_{t_0}^{t} f(\tau)d\tau\right]$$

- The general solution therefore is:

$$x(t) = \phi(t, t_0)x(t_0) + \int_{t_0}^{t} \phi(t, \tau)g(\tau)u(\tau)d\tau$$

- When f(t) is constant:

$$\phi(t, t_0) = \exp[f \cdot (t - t_0)] = e^{f \cdot (t - t_0)}$$

# Outline

- 4.3 Aspects of Linear Systems Theory
  - 4.4.1 Linear Time Invariant Systems
  - 4.3.2 State Space Representation of Linear Systems
  - 4.3.3 Nonlinear Dynamical Systems
  - 4.3.4 Perturbative Dynamics of Linear Systems
  - Summary
- 4.5 Predictive Modelling and System Identification

# State Space

- Remember the special representation used for Runge Kutta?

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{u}(t), t)$$

- State space = a minimal set of variables which can be used to predict future state given inputs:
  - Number of initial conditions in a differential equation.

# Conversion of an LTI ODE

- Consider the second order LTI ODE:

$$\frac{d^2 y}{dt^2} + a_1 \frac{dy}{dt} + a_0 y = u(t)$$

- Choose the state variables to be:

$$x_1(t) = y(t)$$

$$x_2(t) = \dot{y}(t) = \dot{x}_1(t)$$

# Conversion of an LTI ODE

- Rewrite the second and the original ODE as:

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = -a_1 x_2(t) - a_0 x_1(t) + u(t)$$

- This is of the form:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

$$\dot{\underline{x}}(t) = F\underline{x}(t) + Gu(t)$$

# Example: Damped Oscillator

- By inspection: $\ddot{y} + 2\zeta\omega_0\dot{y} + \omega_0^2 y = u(t)$

$$a_1 = 2\zeta\omega_0 \quad a_1 = \omega_0^2$$

- Hence, the system is of the form:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

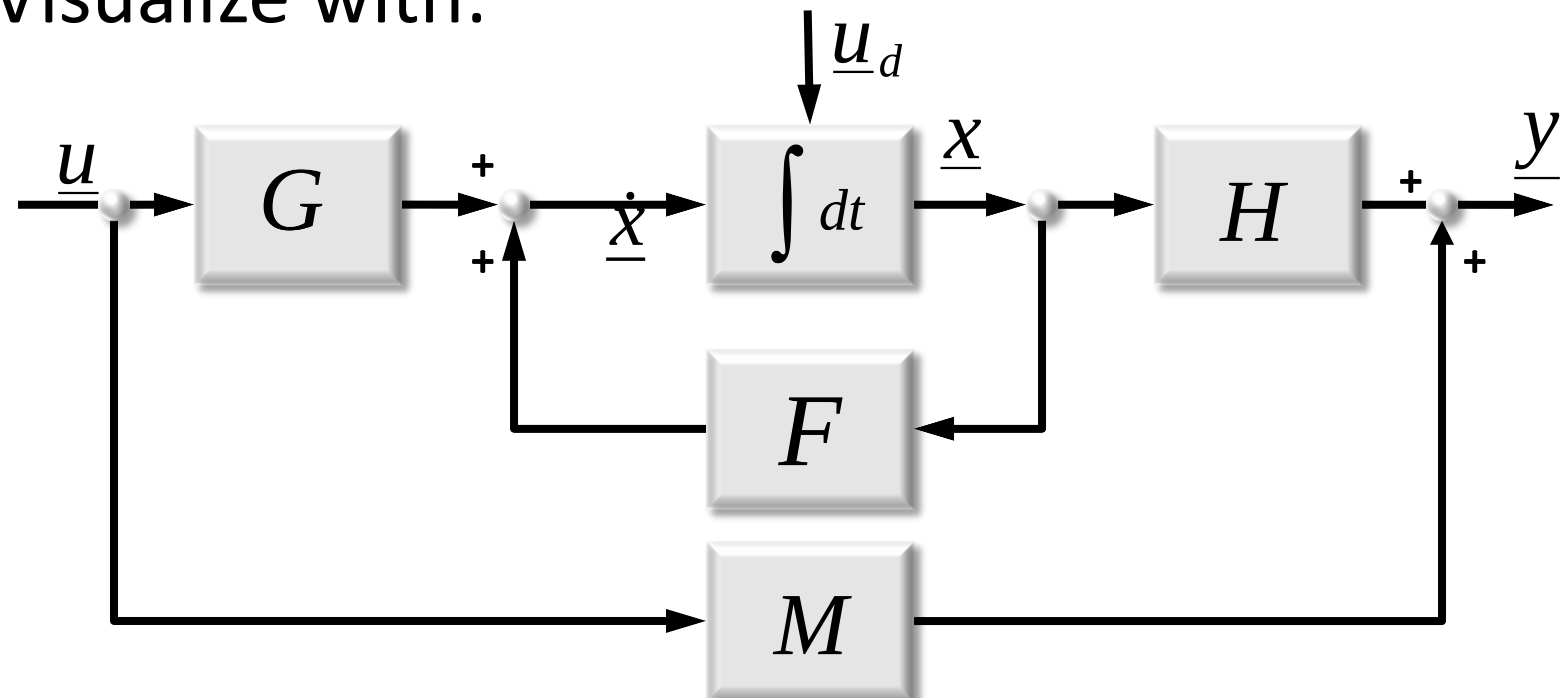
- Where x1 is the position and x2 is the velocity.

# General Linear Dynamical Systems

- State Equations:

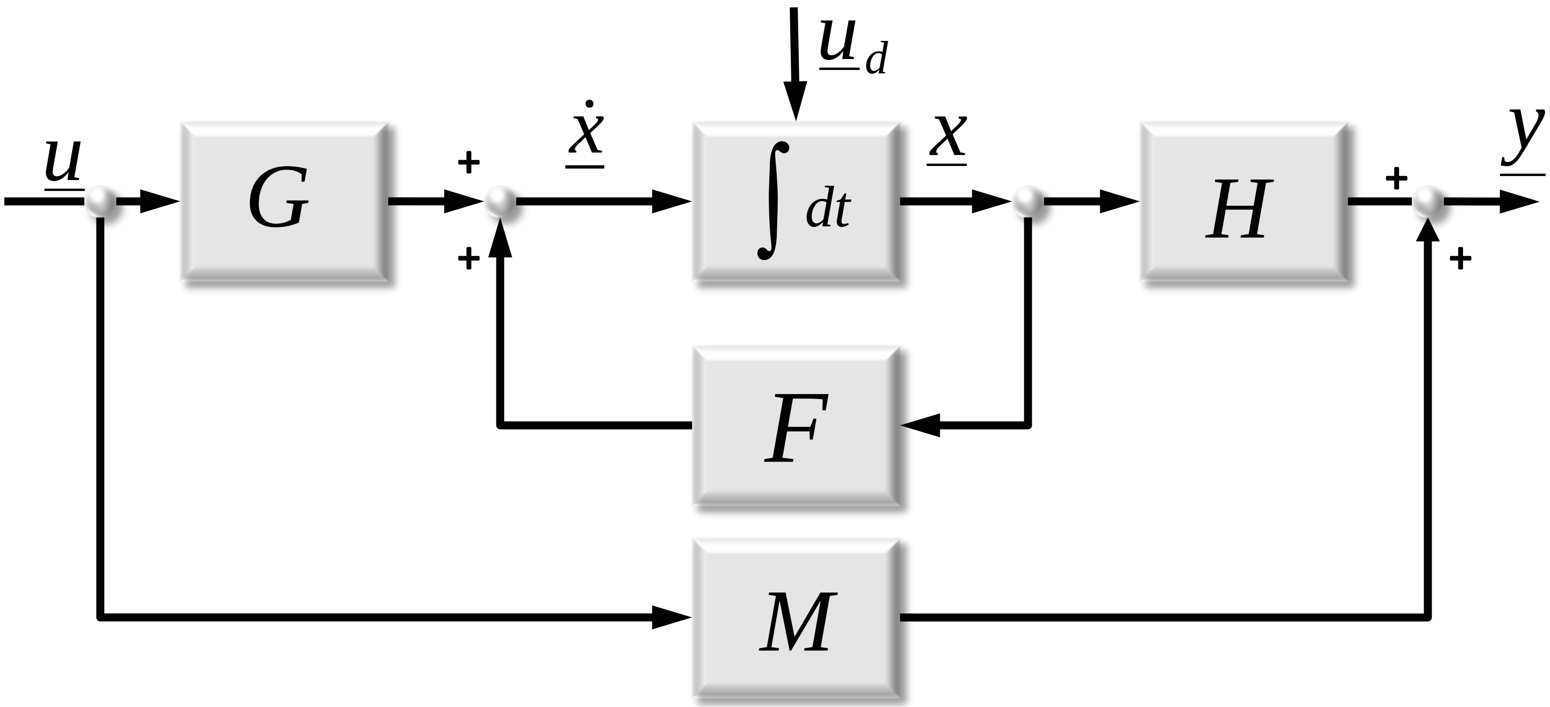
$$\dot{\underline{x}}(t) = Fx(t) + Gu(t)$$
$$y(t) = Hx(t) + Mu(t)$$

- Visualize with:






Carnegie Mellon
THE ROBOTICS INSTITUTE


$\underline{u}_d$
$\underline{u}$
$G$
+
$\underline{\dot{x}}$
+
$\int dt$
$\underline{x}$
$H$
+
$\underline{y}$
+
$F$
$M$

# Vector Case – Constant Coefficient

- When the system dynamics matrix F(t) is constant wrt time:

$$\Phi(t, \tau) = e^{F(t-\tau)}$$

Matrix Exponential

- Recall: by definition (for any matrix A):

$$e^{A} = \exp(A) = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$$

# Solution – Vector Case

- Knowing the transition matrix is equivalent to knowing the solution because:

$$\underline{x}(t) = \Phi(t, t_0)\underline{x}(t_0) + \int_{t_0}^{t} \Phi(t, \tau)G(\tau)\underline{u}(\tau)d\tau$$

Vector Convolution Integral

- This is the general solution to:

$$\dot{\underline{x}}(t) = F(t)\underline{x}(t) + G(t)\underline{u}(t)$$

# Outline

- 4.3 Aspects of Linear Systems Theory
  - 4.4.1 Linear Time Invariant Systems
  - 4.3.2 State Space Representation of Linear Systems
  - <u>4.3.3 Nonlinear Dynamical Systems</u>
  - 4.3.4 Perturbative Dynamics of Linear Systems
  - Summary
- 4.5 Predictive Modelling and System Identification

# Nonlinear Dynamical System

- Takes the form:

Nonlinear differential equation

State Equations
System Model
Process Model

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{u}(t), t)$$

Nonlinear algebraic equation

Measurement Model
Observer

$$\underline{z}(t) = \underline{h}(\underline{x}(t), \underline{u}(t), t)$$

State

Inputs
Forcing Function

# Solutions

- Closed form solutions need not exist at all for nonlinear equations.
- With computers though, we can always integrate like so:

$$\underline{x}(t) = \underline{x}(0) + \int_0^t \underline{f}(\underline{x}(\tau), \underline{u}(\tau), \tau)\, d\tau$$

- This case subsumes the linear case so anything true of nonlinear systems is true of a linear one.
  - Including the next few slides…..

# Relevant Properties

- Homogeneity (for some constant k):

$$f[\underline{x}(t), k \times \underline{u}(t)] = k^n \times f[\underline{x}(t), \underline{u}(t)]$$

- We say system is "homogeneous to degree n wrt u(t)".
- u(t) must occur in f() as a factor like so:

$$\underline{f}[\underline{x}(t), \underline{u}(t)] = \underline{u}^n(t) g(\underline{x}(t))$$

- As a result, all terms of the Taylor series of f() over u(t) of order less than n vanish.

# Drift Free

- All homogeneous systems are drift free.Their zero input response is zero.

$$\underline{u}(t) = 0 \Rightarrow \dot{\underline{x}}(t) = 0$$

- Such systems can be stopped instantly by nulling the inputs.

- Similar to “drift-free” designation in control theory.

# Reversibility & Monotonicity

- Odd degree homogeneity implies a reversible system.

$$\underline{u}_2(t) = -\underline{u}_1(\tau - t) \Rightarrow \underline{f}_2(t) = -\underline{f}_1(\tau - t)$$

- Even degree homogeneity implies monotonicity. Sign of derivative irrelevant to sign of u().

$$\underline{u}_2(t) = -\underline{u}_1(t) \Rightarrow \underline{f}_2(t) = \underline{f}_1(t)$$

# Outline

- 4.3 Aspects of Linear Systems Theory
  - 4.4.1 Linear Time Invariant Systems
  - 4.3.2 State Space Representation of Linear Systems
  - 4.3.3 Nonlinear Dynamical Systems
  - <u>4.3.4 Perturbative Dynamics of Linear Systems</u>
  - Summary
- 4.5 Predictive Modelling and System Identification

# Linearizing a Nonlinear Diff Eq

- Consider again:

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{u}(t), t)$$

- Suppose some u(t) generates some solution x(t). This is the "reference trajectory".
- Suppose we want a solution for:

perturbed input

$$\underline{u}'(t) = \underline{u}(t) + \delta\underline{u}(t)$$

input perturbation

- The solution can be written as:

perturbed state

$$\underline{x}'(t) = \underline{x}(t) + \delta\underline{x}(t)$$

state perturbation

- Defines the state perturbation dx(t) as the difference in solutions.

# Linearizing a Nonlinear Diff Eq

- If the perturbed solution is a solution, then:

$$\dot{\underline{x}}'(t) = \dot{\underline{x}}(t) + \delta\dot{\underline{x}}(t) = \underline{f}[\underline{x}(t) + \delta\underline{x}(t), \underline{u}(t) + \delta\underline{u}(t), t]$$

- Write a truncated Taylor Series at each point in time for the derivative f():

$$\underline{f}[\underline{x}(t) + \delta\underline{x}(t), \underline{u}(t) + \delta\underline{u}(t), t] \approx \underline{f}[\underline{x}(t), \underline{u}(t), t] + F(t)\delta\underline{x}(t) + G(t)\delta\underline{u}(t)$$

- Where the two new matrices are the Jacobians:

$$F(t) = \left.\frac{\partial}{\partial \underline{x}}\underline{f}\right|_{\underline{x},\underline{u}} \qquad G(t) = \left.\frac{\partial}{\partial \underline{u}}\underline{f}\right|_{\underline{x},\underline{u}}$$

# Linearizing a Nonlinear Diff Eq

- At this point we have:

$$\dot{\underline{x}}(t) + \delta\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{u}(t), t) + F(t)\delta\underline{x}(t) + G(t)\delta\underline{u}(t)$$

- Recall the original differential equation:

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{u}(t), t)$$

- Cancel it from the top one:

$$\delta\dot{\underline{x}}(t) = F(t)\delta\underline{x}(t) + G(t)\delta\underline{u}(t)$$

Linear Perturbation Equation

- If you know the Jacobians, you know the perturbative dynamics – the dynamics of error.
- If you know the transition matrix of that, you know the closed form solution to error dynamics.

# Next year

- Move slide 60 (or so) on perturbative dynamics od State Est 1 here. The example will be used later in State Est1 to derive Integrated Heading error dynamics in dead reckoning.
- Also move slide 61, 62 on transition matrix.

# Outline

- 4.3 Aspects of Linear Systems Theory
  - 4.4.1 Linear Time Invariant Systems
  - 4.3.2 State Space Representation of Linear Systems
  - 4.3.3 Nonlinear Dynamical Systems
  - 4.3.4 Perturbative Dynamics of Linear Systems
  - Summary
- 4.5 Predictive Modelling and System Identification

# Summary

- Nonlinear dynamical systems cannot be solved in closed form in general.
- The general solution for linear, even time-varying, dynamical systems exists.
  - Solution rests on Transition matrix
- Perturbative techniques linearize nonlinear differential equations
  - makes them solveable.

# Outline

- 4.3 Aspects of Linear Systems Theory
- 4.5 Predictive Modelling and System Identification

# Introduction – the "ives"

- Mobile robots must often be:
  - Deliberative – decide among options
  - Perceptive – aware of the surroundings
  - Reactive – capable of fast action
- They must be both
  - smart and
  - fast
- … doing that involves tradeoffs.

# Role of Dynamics

- In support of the above, need to be …
  - Predictive – able to project consequences
  - Active – able to execute a plan of action
- You need dynamics models for both of these.

# Predictive Modeling

- Must model …
  - Information processing and propagation.
  - Physical vehicle / environment interaction.

- Often need to map …
  - what you can do (exert forces)
  - what you care about (trajectory through space).

- Latter requires integrating the dynamics.

# Outline

- 4.3 Aspects of Linear Systems Theory
- 4.5 Predictive Modelling and System Identification
  - 4.5.1 Braking
  - 4.5.2 Turning
  - 4.5.3 Vehicle Rollover
  - 4.5.4 Wheel Slip and Yaw Stability
  - 4.5.5 Parameterization and Linearization of Dynamic Models
  - 4.5.6 System Identification
  - Summary

# Reasons for Braking

- A) Last resort response to problems.
  - Collision is imminent due to
    - no solution or
    - inadequate planning or control.
- B) Deliberately slow down.
  - On slopes
  - The motion is finished.
  - In order to turn around.

# Avoiding Collision

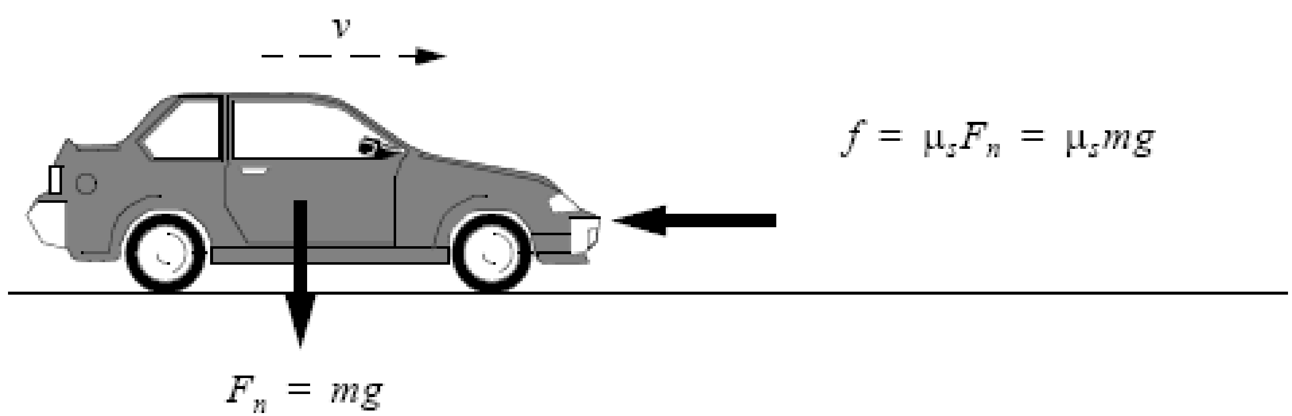
- Requires precise knowledge of the time and space required to react.

Why Care about time?

- These depend heavily on:
  - Speed (initial KE)
  - Friction (work done by friction)
  - Slope (change in PE)

# Braking Model

- Assume brakes are applied instantly:
- Free body diagram:
  - Friction and Weight are coupled.
- Do heavier vehicles take longer to stop?



# Simple Model

- Equate work done by external forces to initial Kinetic energy (assume it is all used up).

$$\frac{1}{2}mv^2 = \mu_s m g s_{brake}$$

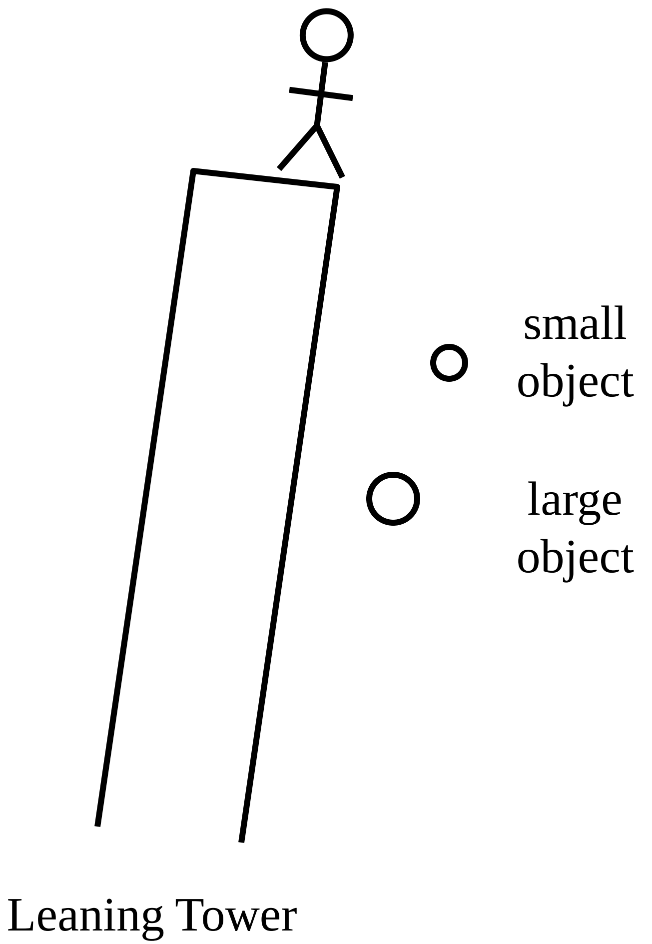
- Solve for braking distance:

$$s_{brake} = \frac{v^2}{2\mu_s g}$$

- Do heavier vehicles take more distance to stop?

# Tangent: Falling

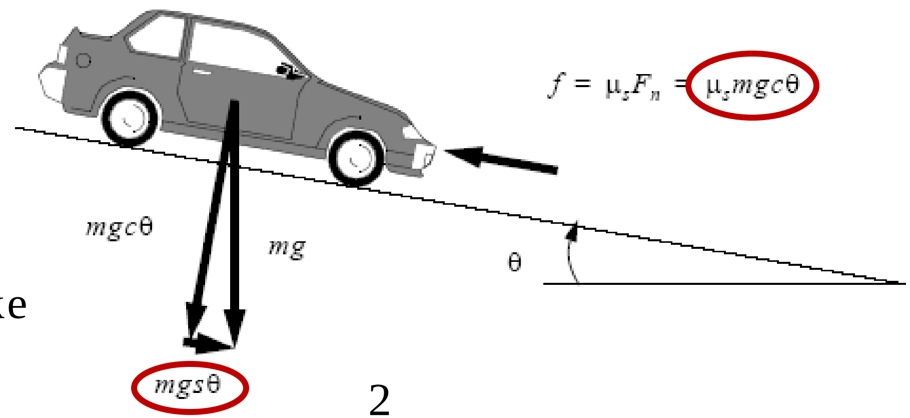
- Do heavier objects fall faster?



# Impact of Slopes

- Again equate work done to initial KE:

$$\frac{1}{2}mv^2 = (\mu_s mgc\theta - mgs\theta)s_{brake}$$




- Solve for distance:

$$s_{brake} = \frac{v^2}{2g(\mu_s c\theta - s\theta)}$$

- Effective coefficient of friction:

$$\mu_{eff} = (\mu_s c\theta - s\theta)$$

- Then, simply:

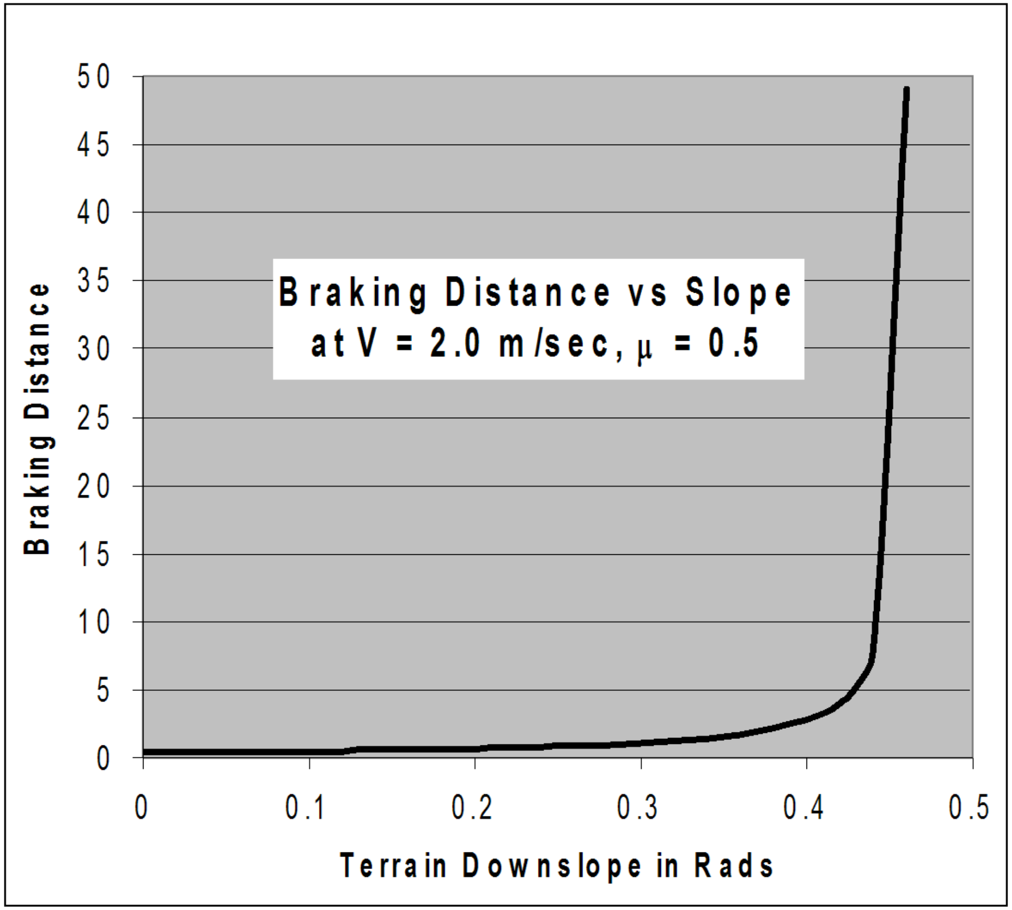
$$s_{brake} = \frac{v^2}{2\mu_{eff}g}$$

# Simple Model on Slopes

- Critical angle exists beyond which gravity overcomes friction….

$$\mu_s c\theta - s\theta = 0 \Rightarrow \tan\theta = \mu_s$$

- Atan() is highly nonlinear.



# General Case

- More generally:

$$\int_0^s \vec{F} \bullet \vec{ds} = \frac{1}{2} m v^2$$

- Robots can compute this.
  - The terrain shape is known.
  - Keep integrating until KE exhausted.
  - Final value of s is stopping distance.

# Rough Heuristic for Slopes

- Make small angle assumptions:

$$c\theta = 1 \quad s\theta = \theta$$

- Change in effective coefficient:

$$\mu_{eff}(\theta) = (\mu_s c\theta - s\theta) \approx \mu_s - \theta$$

10% slope reduces μ by 0.1

- Ratio of sloped to level stopping distance:

$$\frac{s_\theta}{s_0} = \left[\frac{1}{1 - \frac{\theta}{\mu_s}}\right] \approx \left[1 + \frac{\theta}{\mu_s}\right]$$

- Stopping distance increases or decreases by the factor $\theta / \mu_s$

# Outline

- 4.3 Aspects of Linear Systems Theory
- 4.5 Predictive Modelling and System Identification
  - 4.5.1 Braking
  - <u>4.5.2 Turning</u>
  - 4.5.3 Vehicle Rollover
  - 4.5.4 Wheel Slip and Yaw Stability
  - 4.5.5 Parameterization and Linearization of Dynamic Models
  - 4.5.6 System Identification
  - Summary

# Turning

- Goal is to cause terrain to exert a moment on the vehicle
  - By 3rd law, vehicle must exert a moment on the terrain.
- May actuate:
  - Wheel steering (Ackerman)
  - Wheel speeds (Differential, skid)

# Simple Motion Prediction

- For small steer angles:

$$\kappa(t) = \alpha(t)$$

- Integrate the differential equations using "back substitution":

The mapping from steer angle and velocity onto the path the robot follows. Assumes flat terrain.

$$\theta(t) = \theta_0 + \int_0^t V(t)\alpha(t)dt$$

$$x(t) = x_0 + \int_0^t V(t)\cos(\theta(t))dt$$

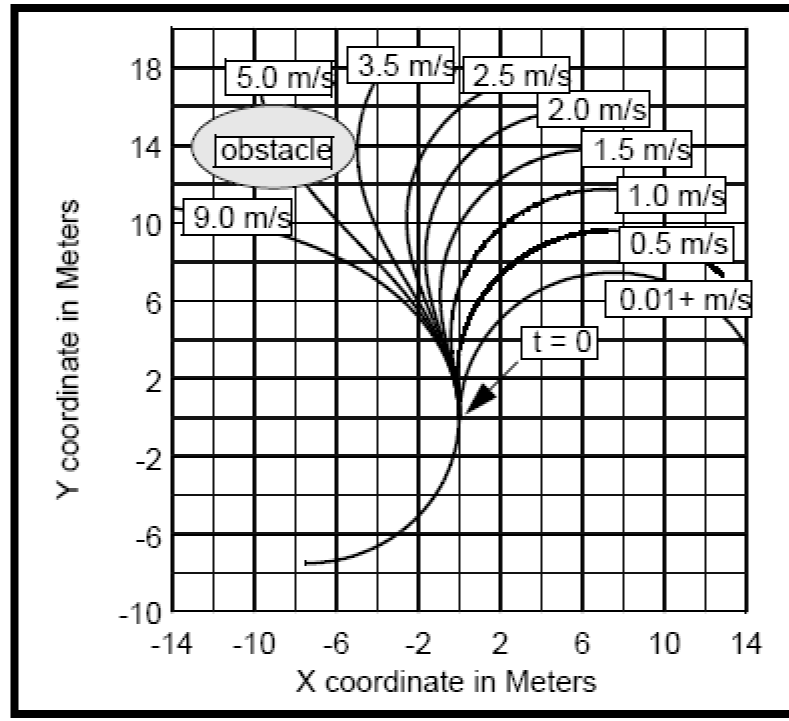
$$y(t) = y_0 + \int_0^t V(t)\sin(\theta(t))dt$$

Note <u>mapping from inputs to outputs are integrals.</u>

- Errors in steering are integrated twice to determine errors in predicted position.

# Reverse Turn @ Multiple Speeds

- A curvature step is the most ambitious maneuver.
- Not modeling steering response leads to collisions with obstacles above 3.5 m/sec speed.

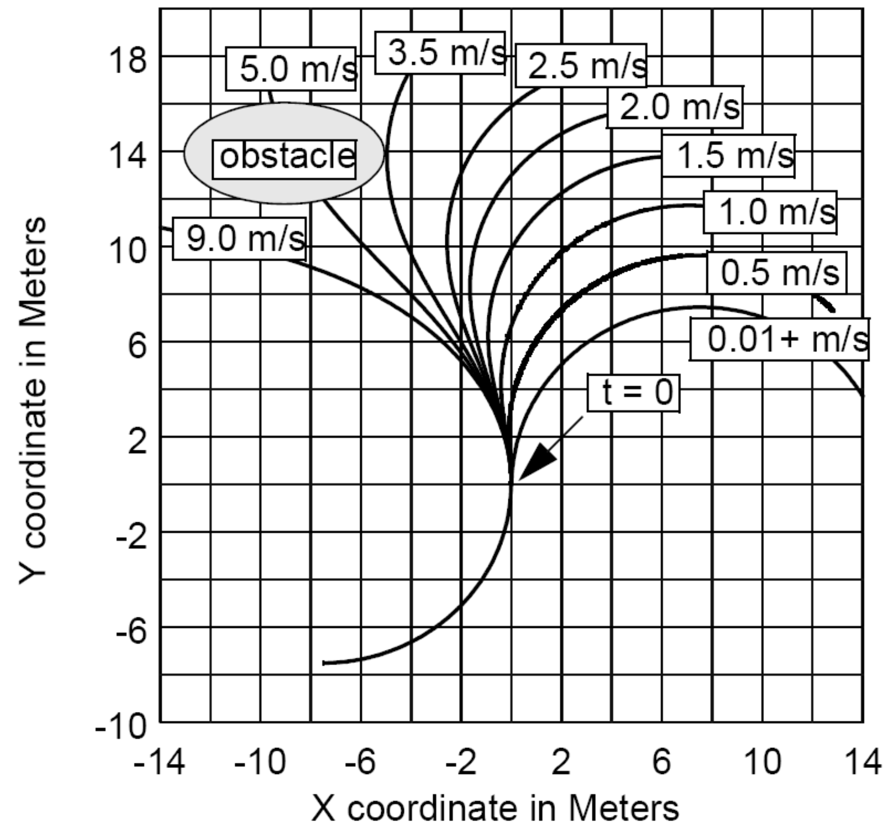



“Reverse Turn”

- One Curvature
- Various Speeds

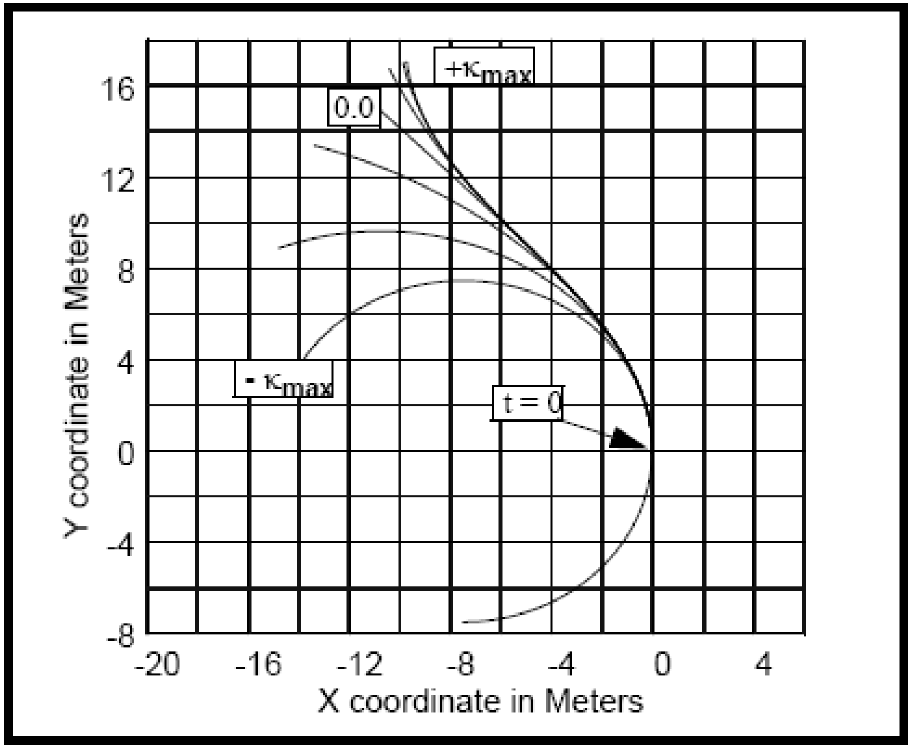
# Recall: Speed Coupling

- Due to vehicle dynamics…….
- The path followed is generally a function of speed.
- Therefore, they must be estimated together.



# Reverse Turn @ Multiple Curvatures

- Different steering commands. Same speed (5 m/s).
- It takes a long distance to cross the forward (y) axis.
  - Its longer the faster you are going.




"Reverse Turn"

- One Speed
- Various Curvatures

# Swerving

- Recall our typical 2D equations of motion:

$$\dot{\psi} = \kappa v$$

$$\psi = \psi_0 + \int v(t)\kappa(t)dt$$

$$x = x_0 + \int v(t)\cos(\psi(t))dt$$

$$y = y_0 + \int v(t)\sin(\psi(t))dt$$

# Swerving

- Assuming velocity is constant, and curvature rate is limited and constant, the yaw is given by:

$$\psi = \psi_0 + v\int \dot{\kappa}_{\max}\, t dt$$

$$\psi = \psi_0 + \frac{v\dot{\kappa}_{\max}}{2} t^2$$

- This gives the position coordinates as:

$$x = v\int \cos\left(\frac{v\dot{\kappa}_{\max}}{2} t^2\right) dt$$

"Clothoids"

$$y = v\int \sin\left(\frac{v\dot{\kappa}_{\max}}{2} t^2\right) dt$$

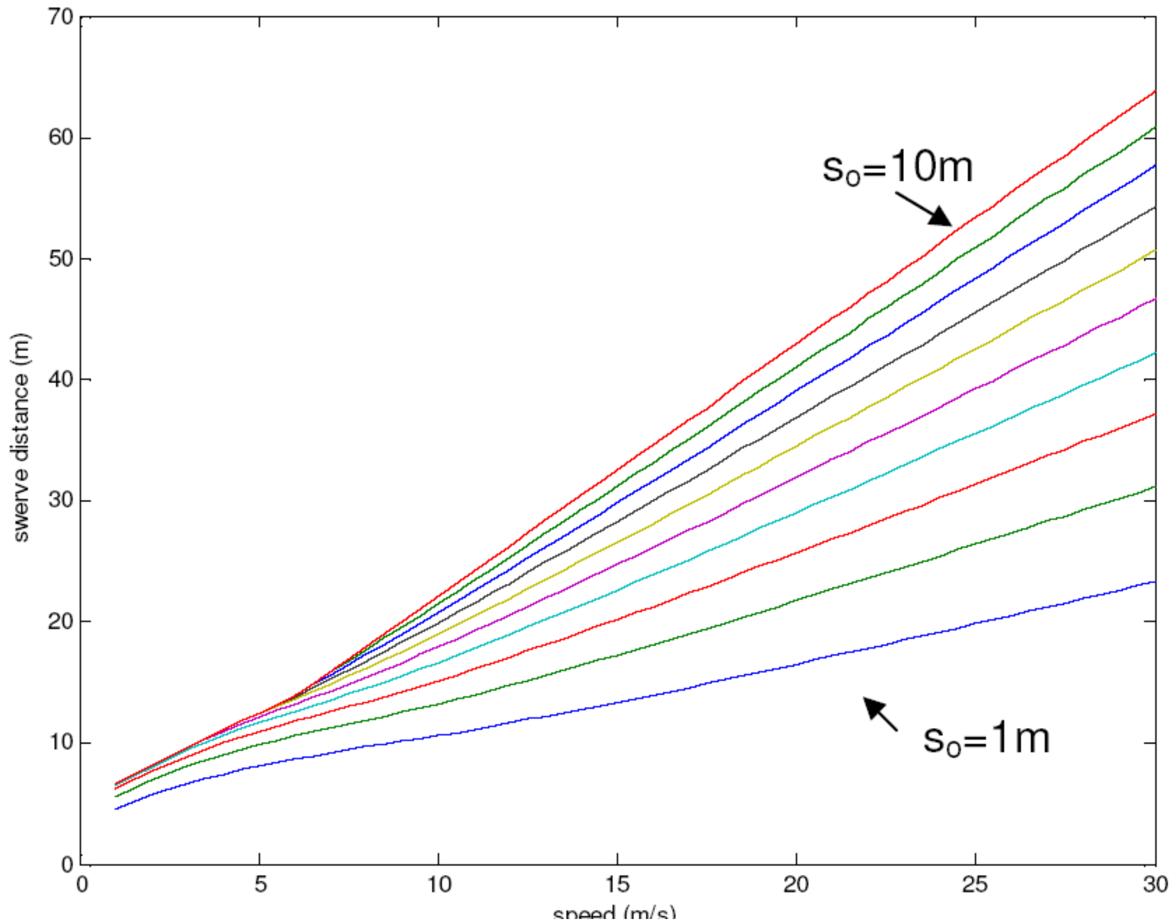
# Swerving

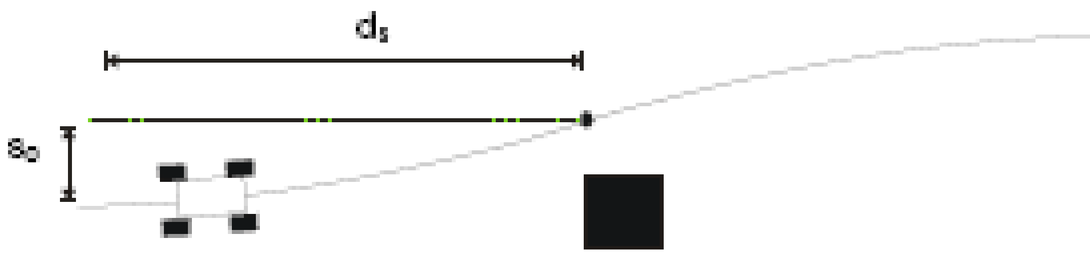
- Two limits on curvature (slipping and rollover) can be computed from:

$$\kappa_{slip} = \frac{\mu g}{v^2} \qquad \kappa_{roll} = \frac{T}{2hv^2} g$$

- Given all this, the equations for (x,y) can be integrated numerically to get….

# Swerving (Urmson)




1) Roughly linear!

2) Lower than stopping distance (v^2/10) at 10 m/s and beyond

Sometimes you can swerve in time even when you cannot stop.

# Outline

- 4.3 Aspects of Linear Systems Theory
- 4.5 Predictive Modelling and System Identification
  - 4.5.1 Braking
  - 4.5.2 Turning
  - <u>4.5.3 Vehicle Rollover</u>
  - 4.5.4 Wheel Slip and Yaw Stability
  - 4.5.5 Parameterization and Linearization of Dynamic Models
  - 4.5.6 System Identification
  - Summary

# Note

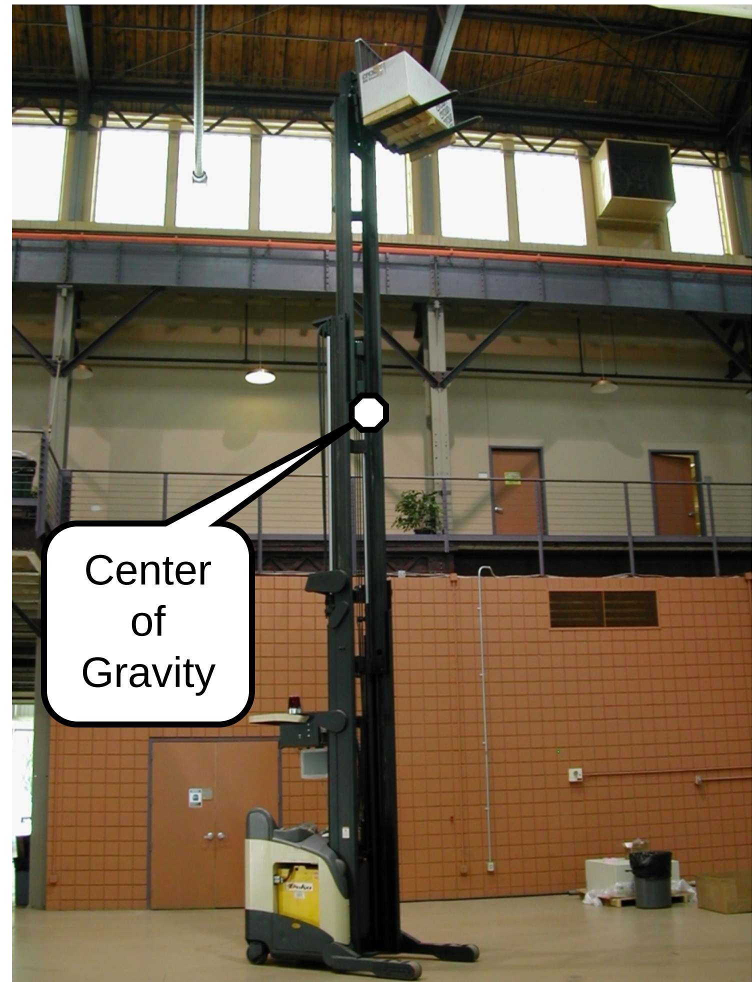
- There is plenty of content on rollver in dyn1 too. Check it all/

# Field Robots Motivation

- Contemporary mining, forestry, agriculture, and military vehicles, operate
  - on slopes and/or
  - at high speeds
- Field robots do rollover!
  - They at least need a reactive system if predictive elements fail.

# Industrial Robots Motivation

- Market forces reward manufacturers of industrial truck that:
  - Are narrower,
  - Lift heavier loads,
  - Lift them higher.
- Automated industrial trucks face the same challenges.

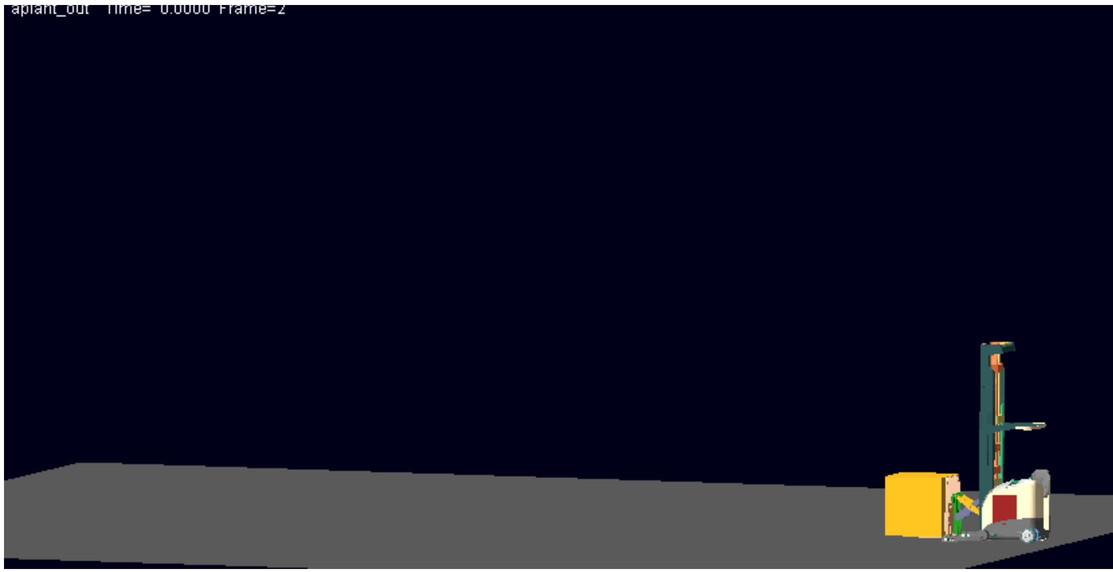


# PerceptOR - Yuma

- Some Robots live dangerously.
- Listen for the distinctive “Crunch” of a ladar sensor.

# UGCV – Roll Test

# Lift Truck Simulations



Governed

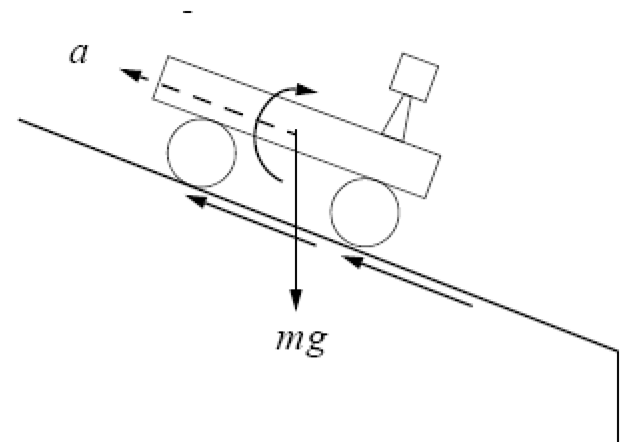
Ungoverned

# Rollover

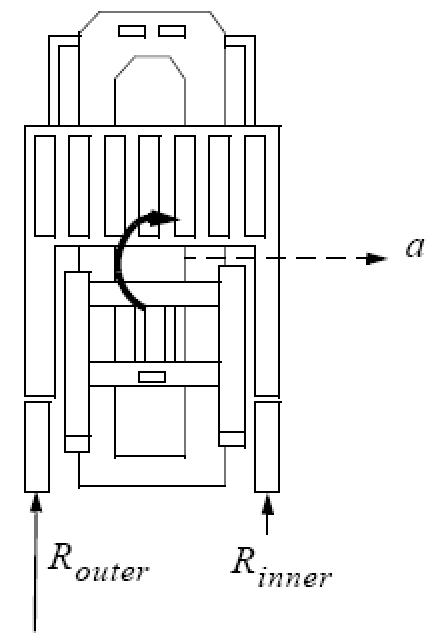
- More likely in factory and field robots.
- Happens due to combinations of:
  - narrow wheel spacing,
  - high centers of gravity
  - high inertial forces (speeds and curvatures)
  - steep slopes
- Incidents may be:
  - Terrain induced (slide sideways into a curb)
  - Maneuver induced (turn too sharp on a hill)

# Examples

- Tipover when stopping on a downslope.




- Rollover when turning sharply.



# Forms of Instability

- Must distinguish two events:
  - Point of wheel liftoff (still recoverable)
  - Point were cg passes over wheels (irrecoverable)
- The first occurs first and is easier to detect
  - Does not require knowledge of inertia.

# NOTE

- The book was updated to use a singel figure and to not reverse the direction of the reactions as the figures do here.
- That changed the signs in a few places so the figures and the math need to be updated here to be consistent with the book.

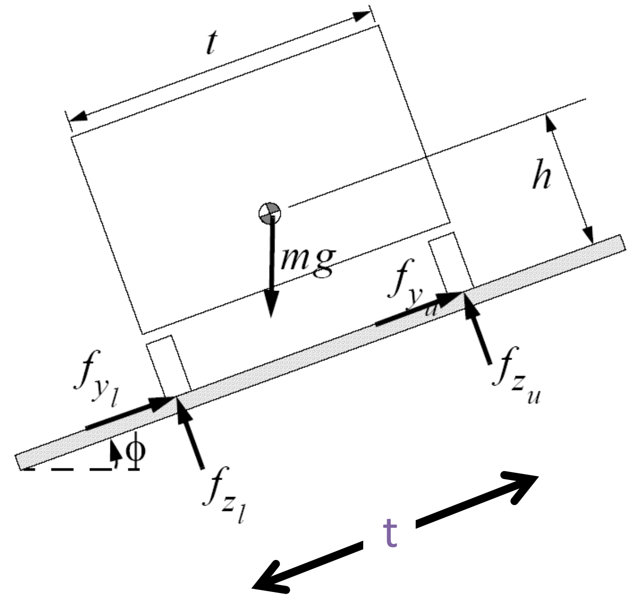
# Static Case

- For translational equilibrium:

$$f_{y_u} + f_{y_l} = mg\sin\phi$$

$$f_{z_u} + f_{z_l} = mg\cos\phi$$

- For rotational equilibrium:




$$f_{z_u} t + mg\sin\phi h = mg\cos\phi\frac{t}{2}$$

Sum moments about lower wheel. Do cross products (r x f) in body coordinates where it is easy

# Static Liftoff

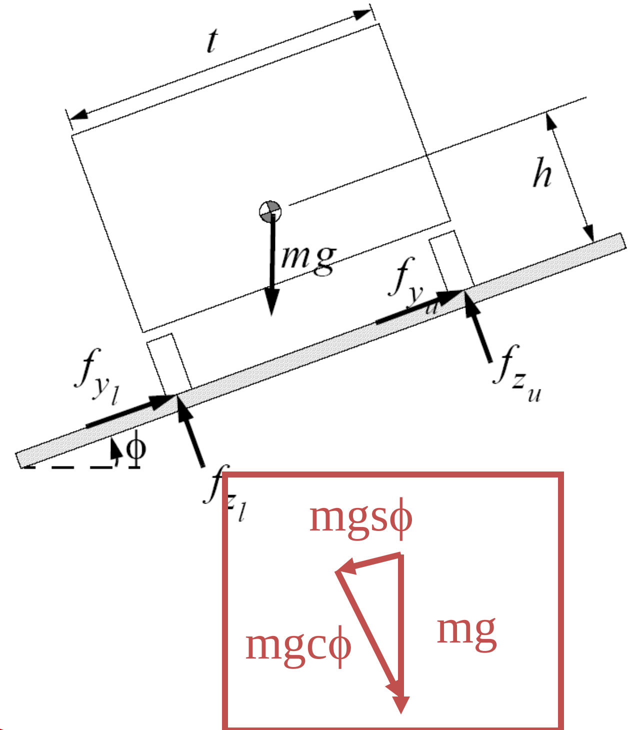
- Imagine raising the slope:
  - fzu decreases
  - fzl increases
- At some point $f_{zu}$=0 and the moment balance becomes:

$$mg\sin\phi h = mg\cos\phi\frac{t}{2}$$

- Can solve for the slope at which tipping occurs:

$$\tan\phi = \frac{t}{2h}$$

- Using this, can compute cg height using a tilt table.

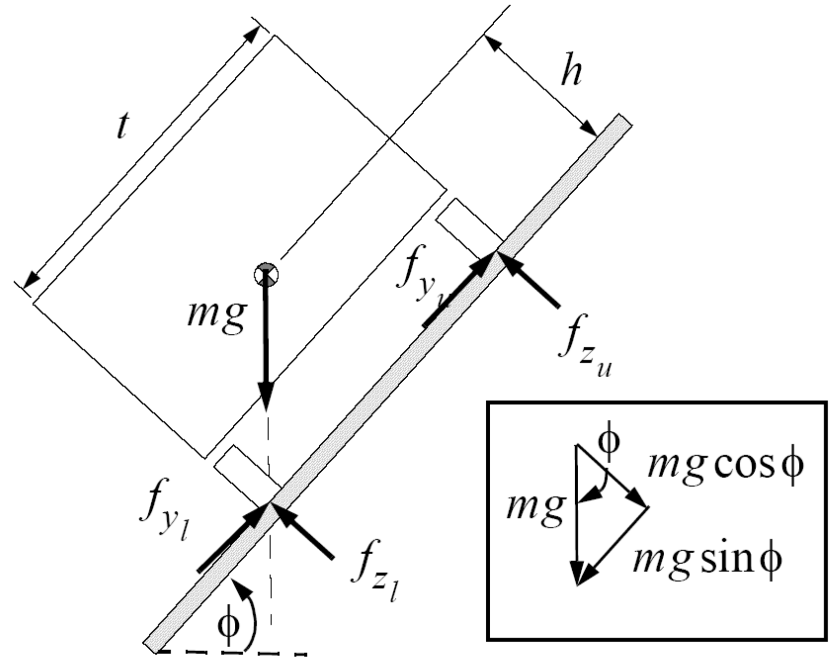



Gravity is the only force involved.

An important/famous vehicle design parameter affecting stability.

# Static Liftoff

- Since we are talking about a moment of a single force…
  - Result can be understood in terms of the direction of gravity.
- Liftoff criterion is first satisfied when gravity vector:
  - emanating from the center of gravity (cg)
  - points at the lower wheel contact point.



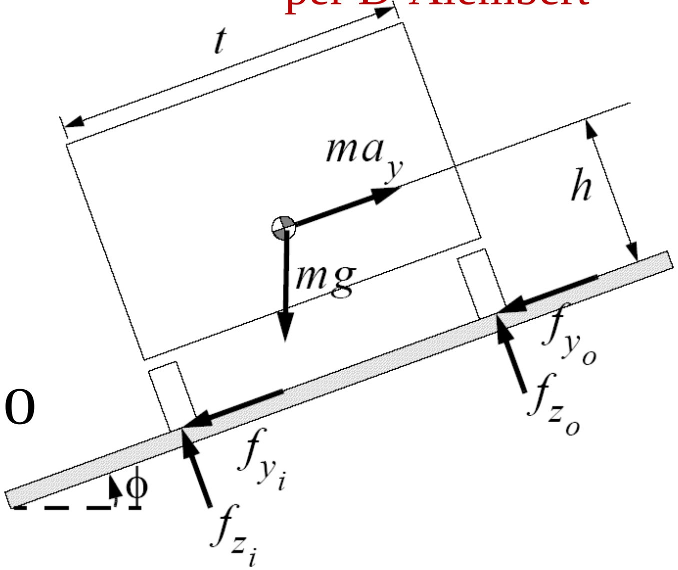



Carnegie Mellon THE ROBOTICS INSTITUTE

# Dynamic Case

Vehicle is turning left
Ma is reversed in sense
per D'Alembert

- Use D'Alemberts principle:
  - I.E. treat – ma like a real force.
- Moment balance:




$$-f_{z_i}t - ma_y h + mgs\phi h + mgc\phi\frac{t}{2} = 0$$

- Solve for lateral acceleration in g's:

$$\frac{a_y}{g} = \left[\frac{t}{2}c\phi + hs\phi - \frac{tf_{z_i}}{mg}\right]/h$$

- Set $f_{z_i} = 0$ to get lateral acceleration threshold.

$$\frac{a_y}{g} = \left[\frac{t}{2}c\phi + hs\phi\right]/h$$



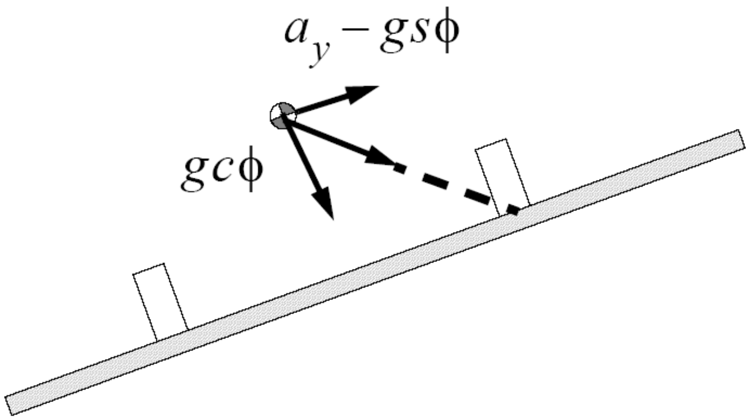
Carnegie Mellon THE ROBOTICS INSTITUTE

# Dynamic Case

- Rewrite last result:

$$\frac{a_y - gs\phi}{gc\phi} = \frac{t}{2h}$$

- Liftoff when net noncontac

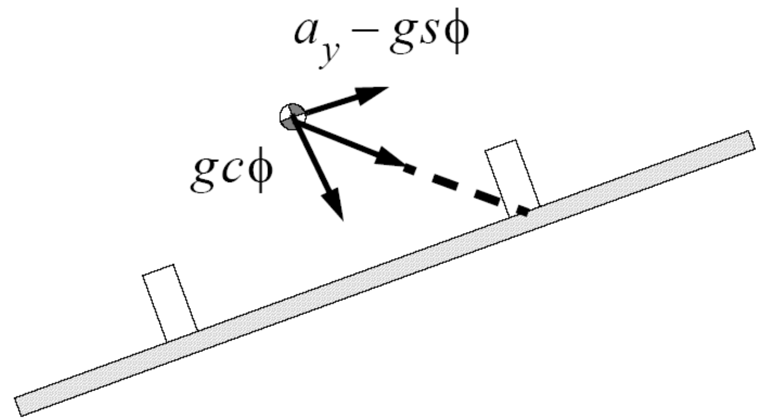



$$\vec{f} = \hat{g} - \hat{a}$$

- Points at the outside wheel contact point.
- A pendulum mounted at the cg aligns with this vector.

# Interpretations

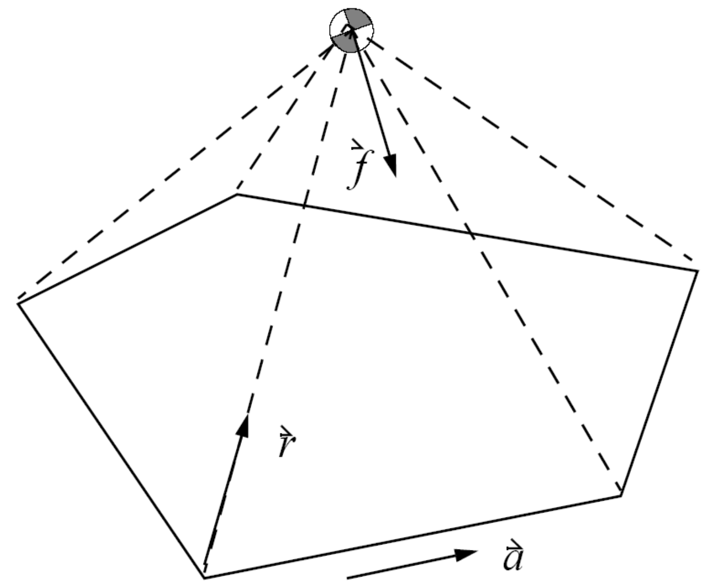
- Static case is just special case of dynamic (ay=0)
- Stability increases with:
  - Lower cg h
  - Wider tread t
  - Lowering slope
  - Decreasing acceleration
    - Slowing down
    - Reducing curvature




$$\frac{a_y - gs\phi}{gc\phi} = \frac{t}{2h}$$

# Stability Pyramid

- Theory generalizes to vehicles of any shape.
- Stability pyramid = the pyramid
  wheel contact points with the c
- Each edge is a potential tipover
  - Moment is:
  - Unbalanced when:




$$\vec{M} = \vec{r} \times \vec{f}$$

$$\vec{M} \bullet \vec{a} > 0$$

Wheels need not be in the same plane.

# Implementation

- Some vehicles articulate mass so the cg would have to be (re-)calculated in real time.
- An accelerometer or inclinometer works like a pendulum, but:
  - It probably cannot be placed at the cg.
  - So, acceleration transforms are necessary.

# Outline

- 4.3 Aspects of Linear Systems Theory
- 4.5 Predictive Modelling and System Identification
  - 4.5.1 Braking
  - 4.5.2 Turning
  - 4.5.3 Vehicle Rollover
  - <u>4.5.4 Wheel Slip and Yaw Stability</u>
  - 4.5.5 Parameterization and Linearization of Dynamic Models
  - 4.5.6 System Identification
  - Summary

# Slip Angle

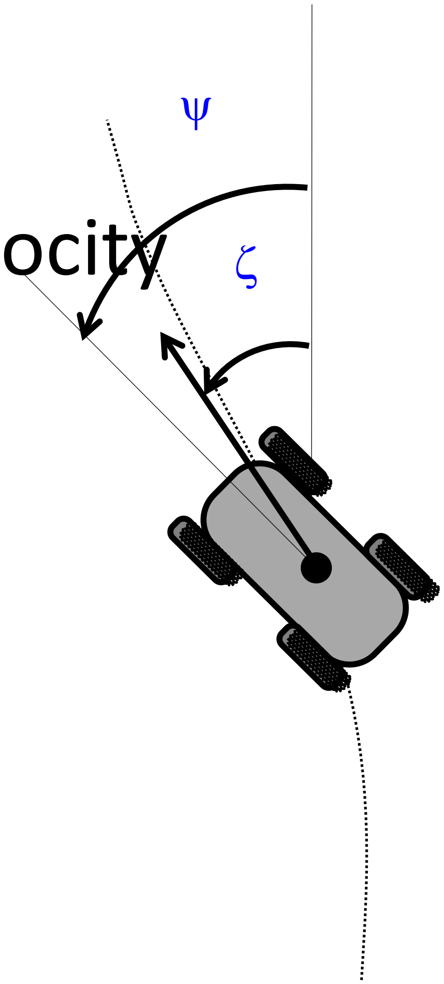
- Defined for a car as:

$$\beta = \psi - \zeta$$

- Alternatively using body frame velocity components:

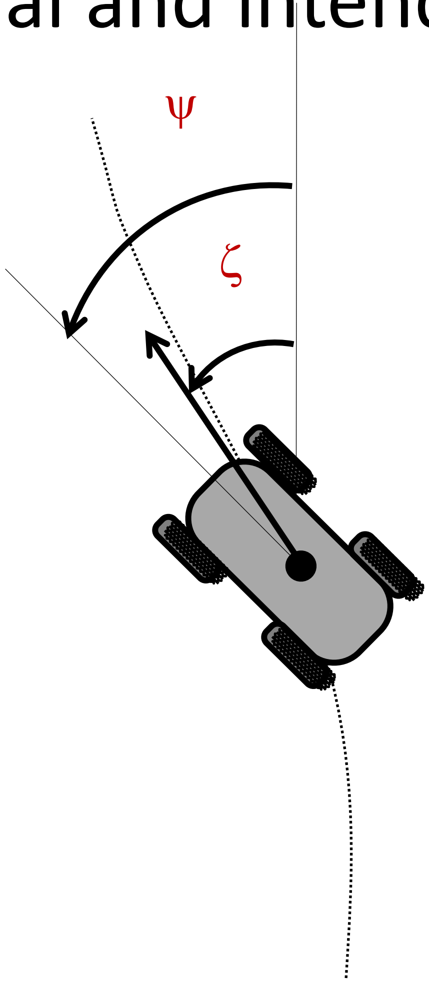
$$\beta = \mathrm{atan2}(V_y, V_x)$$

- Can be defined for wheels too.



# Generalized Slip Angle

- Define the angle between the actual and intended velocity:

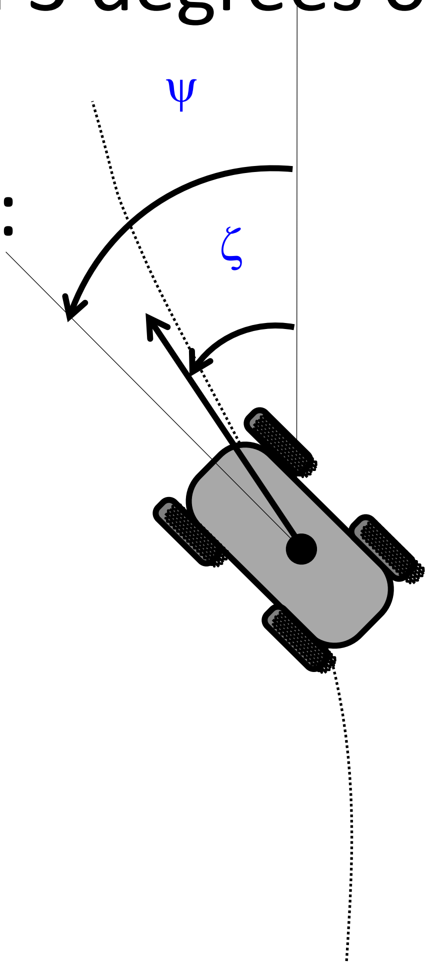



$$\beta = \mathrm{acos}[(\tilde{\underline{V}} \cdot \underline{V}) / (|\tilde{\underline{V}}||\underline{V}|)]$$

actual reference

# Generalized Slip Equation

- The velocity may be incorrect in all 3 degrees of freedom.
- Express errors in body coordinates:

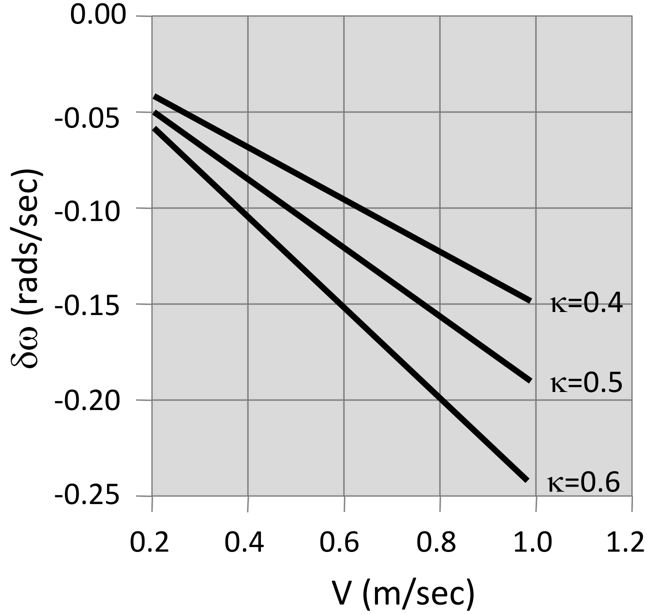



$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \left( \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} + \begin{bmatrix} \delta V_x \\ \delta V_y \\ \delta \omega \end{bmatrix} \right)$$

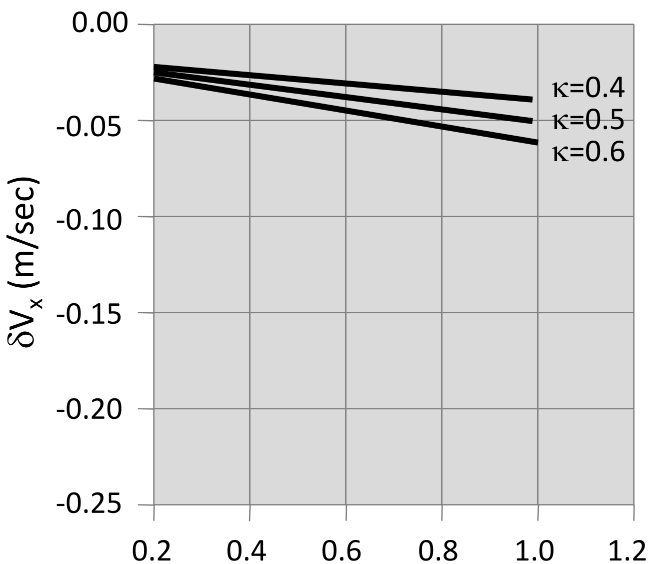
$$^{w}\underline{\tilde{V}} = R(\theta)(\underline{V} + \delta V)$$

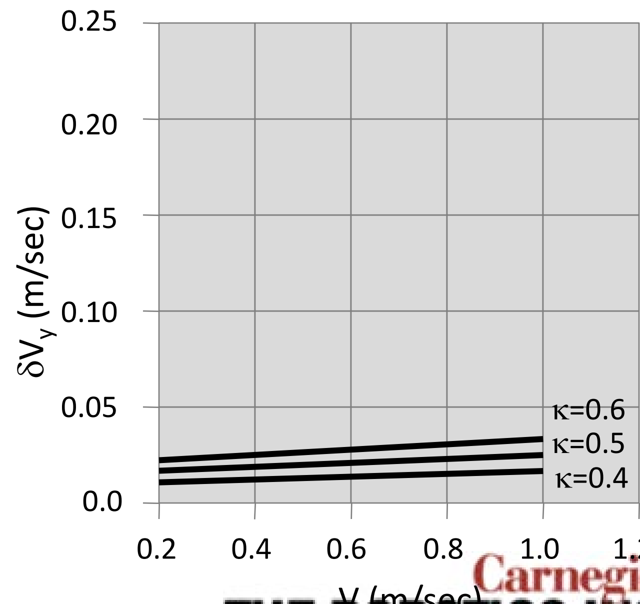
actual reference

# Wheel Slip Graphs



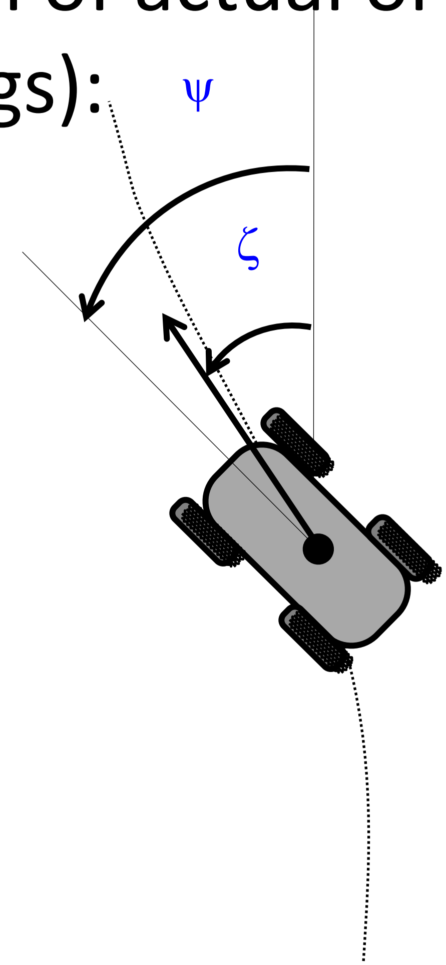




# Removing Slip with Prediction

- Slip can be expressed as a function of actual or reference velocity (and other things):
- Compensate in body coordinates.

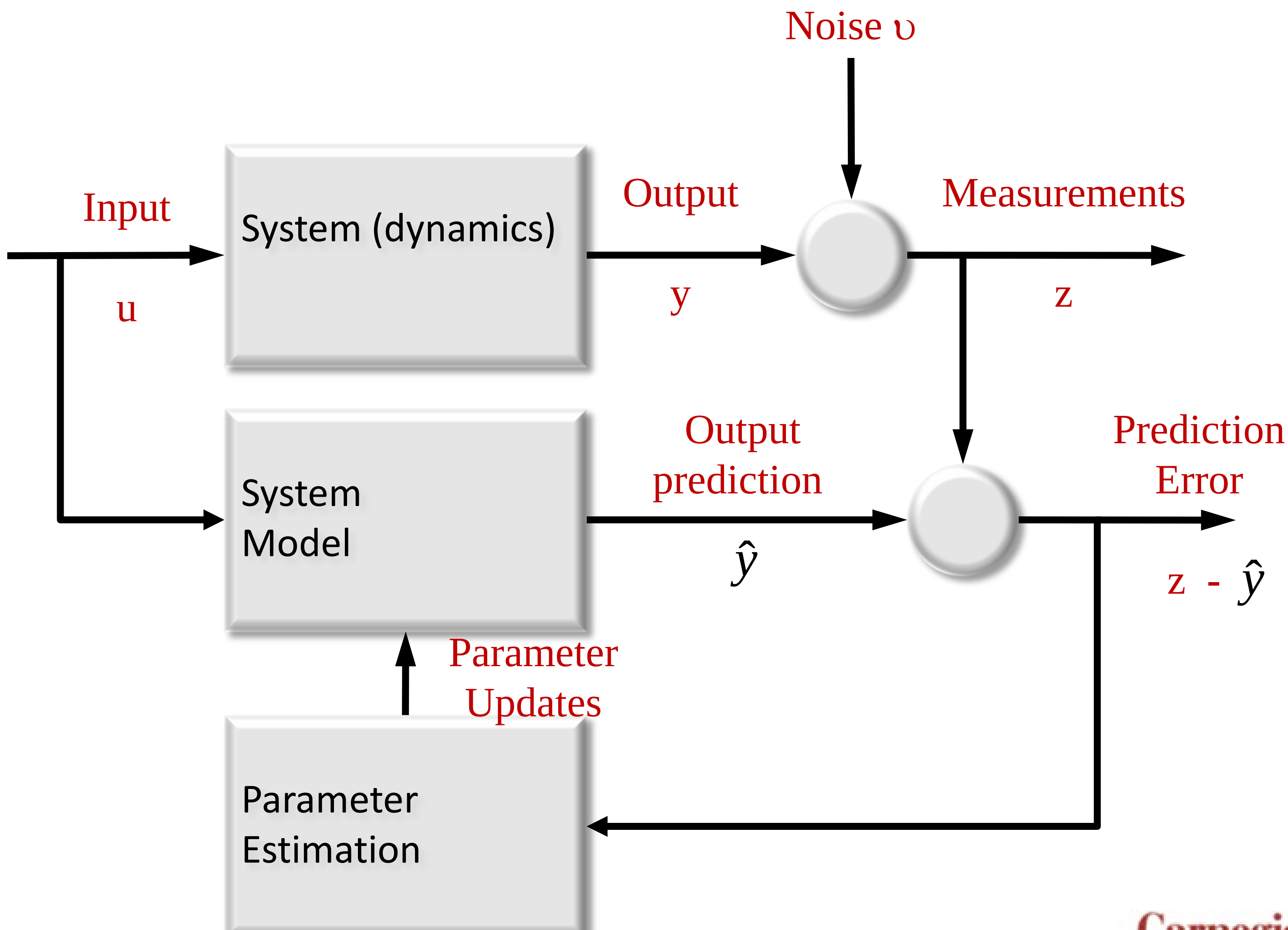



reference actual

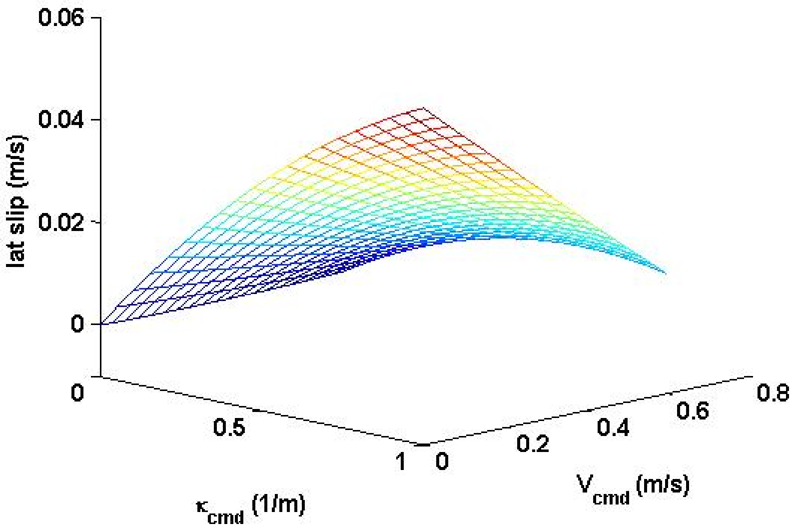
$$\underline{V} = R^{-1}(\theta)\,{}^{w}\underline{\tilde{V}} - \delta V$$

# Outline

- Introduction
- Wheel Slip
- Braking
- Turning & Swerving
- Rollover
- System Identification
- Summary


Noise υ
Input
System (dynamics)
Output
Measurements
u
y
z
System
Model
Output
prediction
Prediction
Error
$\hat{y}$
$z - \hat{y}$
Parameter
Updates
Parameter
Estimation

*Side Slip*



0.06
0.04
0.02
0
lat slip (m/s)
0
0.5
1
$\kappa_{cmd}$ (1/m)
0
0.2
0.4
0.6
0.8
$V_{cmd}$ (m/s)


Laser
Range
Finder
Video
Camera
SICK
Land Tamer II

# Summary

- Braking distance:
  - increases quadratically with initial speed
  - depends heavily on slope
- Turning and Swerving:
  - predicting steering maneuvers requires calibrated dynamic models.
- Rollover stability can be measured with a pendulum at the cg.