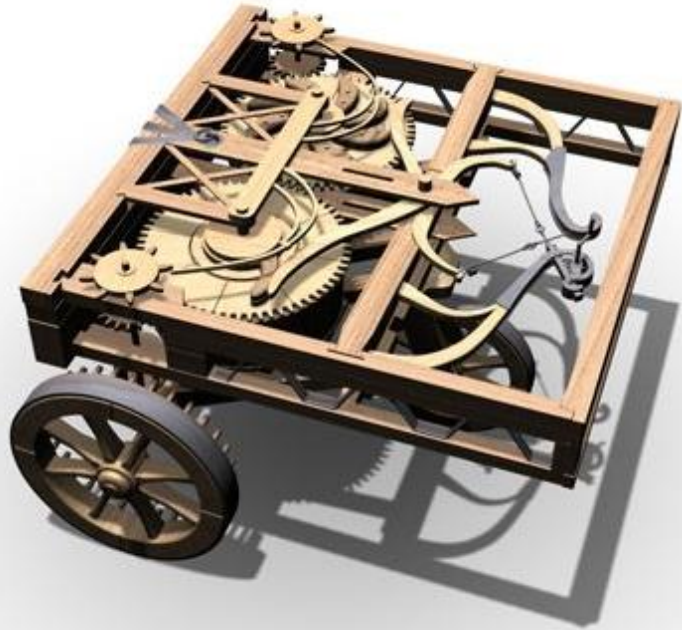# COURSE Review 1
MOBILE ROBOTICS
CENG 5437-01, CENG 4391-02 SPRING 2022

1. Introduction to the Course To the videos – History, Cars, Applications. ROS robots

2. Physics, Inertia, URDF models

3. Selecting Motors for Wheeled Robots

4. PID control of Wheeled Robots

http://www.universalleonardo.org/worklarge.php?id=512&image=0&trail=0&trailCount=&name=
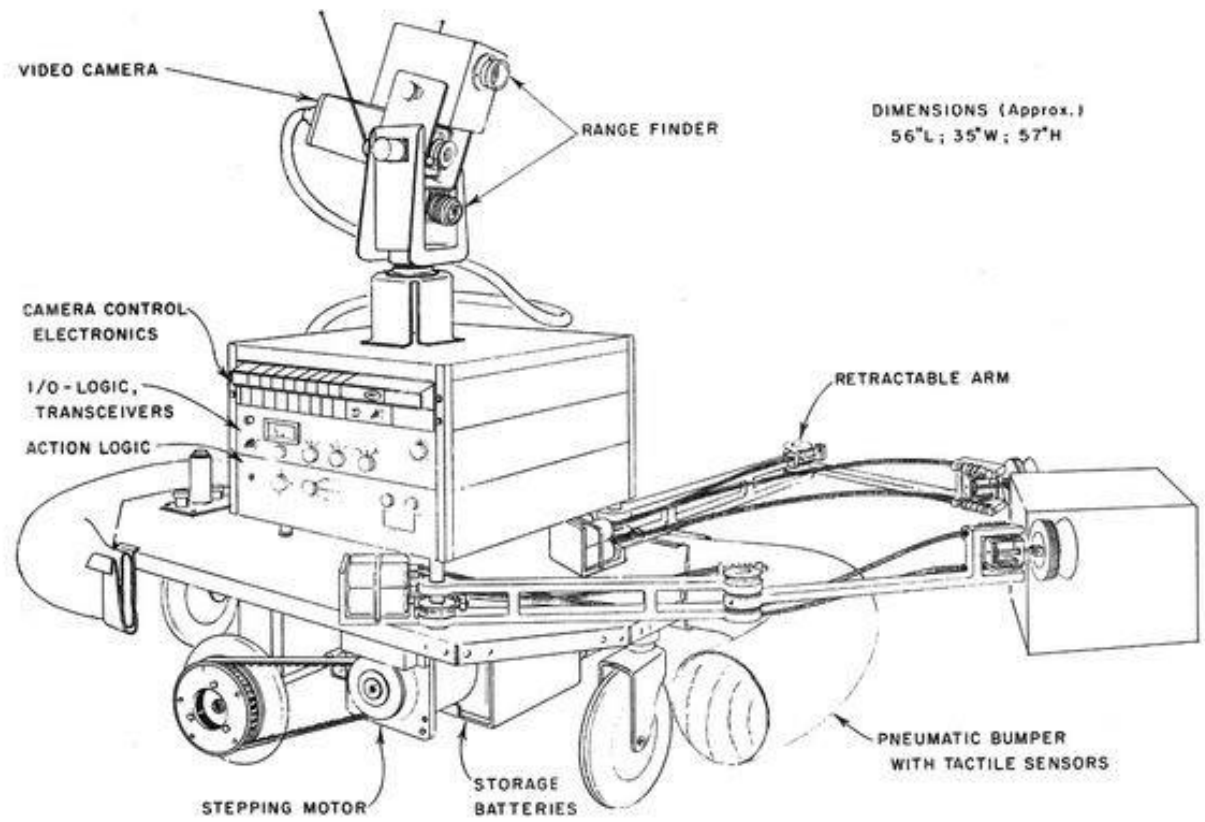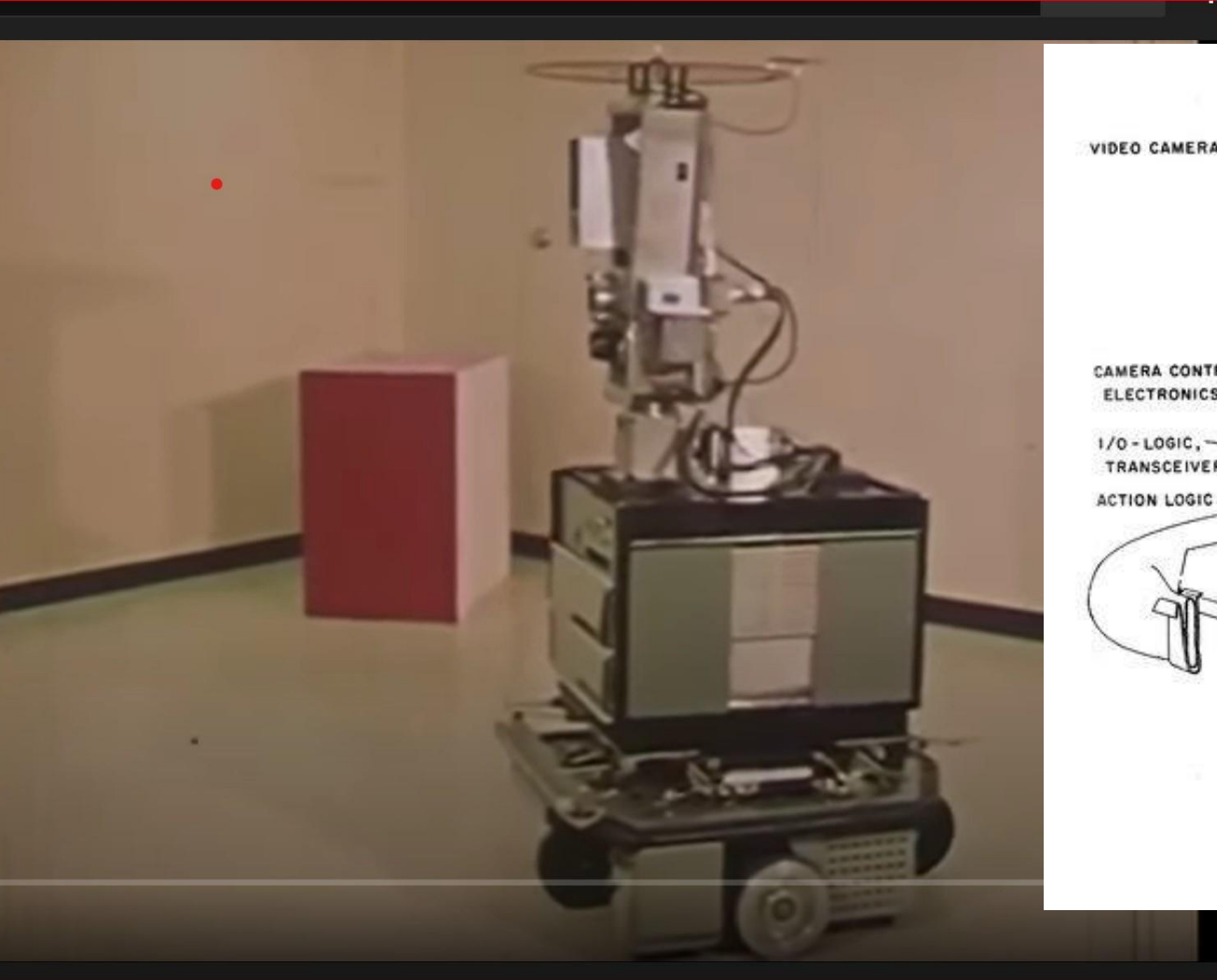
SHAKEY STARTS THE REVOLUTION!

Figure labels: VIDEO CAMERA, RANGE FINDER, DIMENSIONS (Approx.) 56"L; 35"W; 57"H, CAMERA CONTROL ELECTRONICS, I/O - LOGIC, TRANSCEIVERS, ACTION LOGIC, RETRACTABLE ARM, STEPPING MOTOR, STORAGE BATTERIES, PNEUMATIC BUMPER WITH TACTILE SENSORS, FIG. 2 AUTOMATON VEHICLE
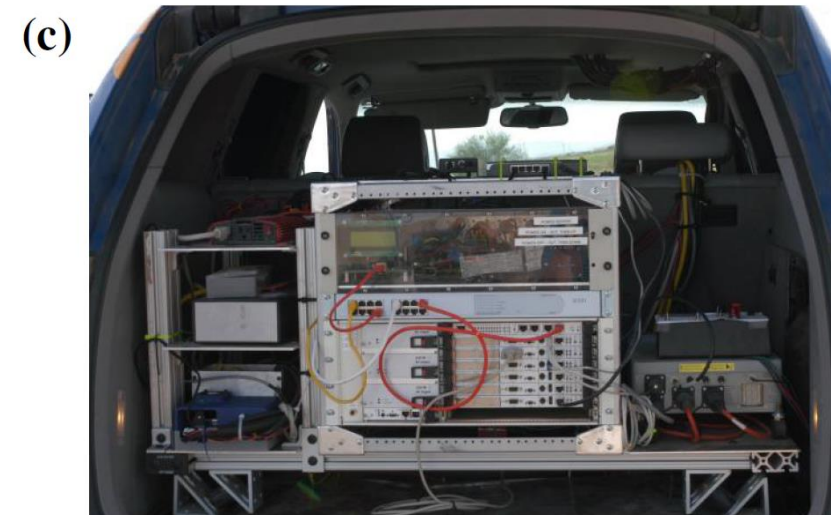
Figure 1: (a) The robot shown at approximately 1:40pm on Oct 8, 2005, as it successfully completes the 132-miles course. (b) View of the car and the roof rack with sensors. (c) The computing system in the vehicle's trunk.
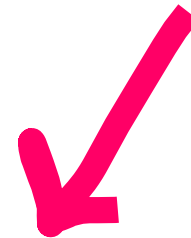
1. **Automated Delivery**

https://uh.edu/news-events/stories/2019/november-2019/11102019-starship-robots.php

Starship Autonomous Food Delivery Robots Deployed at University of Houston

StarshipPatentUS10732641.pdf

Good Info

https://patentimages.storage.googleapis.com/39/5e/9e/553ed587b69901/US10732641.pdf

(12) **United States Patent**
Heinla et al.

(10) **Patent No.:** **US 10,732,641 B2**
(45) **Date of Patent:** **Aug. 4, 2020**

(54) **MOBILE ROBOT SYSTEM AND METHOD FOR GENERATING MAP DATA USING STRAIGHT LINES EXTRACTED FROM VISUAL IMAGES**

(71) Applicant: **Starship Technologies OÜ**, Tallinn (EE)

(72) Inventors: **Ahti Heinla**, Tallinn (EE); **Kalle-Rasmus Volkov**, Tallinn (EE); **Lindsay Roberts**, Tallinn (EE); **Indrek Mandre**, Tallinn (EE)

(73) Assignee: **STARSHIP TECHNOLOGIES OÜ**, Tallinn (EE)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **15/968,802**

(22) Filed: **May 2, 2018**

(58) Field of Classification Search
CPC .. G05D 1/0274; G05D 1/0212; G05D 1/0088; G05D 1/0251; G05D 2201/0216;
(Continued)

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,015,831 B2   3/2006 Karlsson et al.
7,873,448 B2   1/2011 Takeda et al.
(Continued)

FOREIGN PATENT DOCUMENTS

WO   WO 99/30270   6/1999

OTHER PUBLICATIONS

Gaspar et al., "Vision-based Navigation and Environmental Representations with an Omni-directional Camera," IEEE Transactions on Robotics and Automation (Dec. 2000) vol. 16, No. 6, pp. 890-898.

(Continued)

RODNEY
BROOKS
CHANGES
PARADIGM

reason about behavior of objects

plan changes to the world

identify objects

monitor changes

Sensors ⟶ build maps ⟶ Actuators

explore

wander

avoid objects
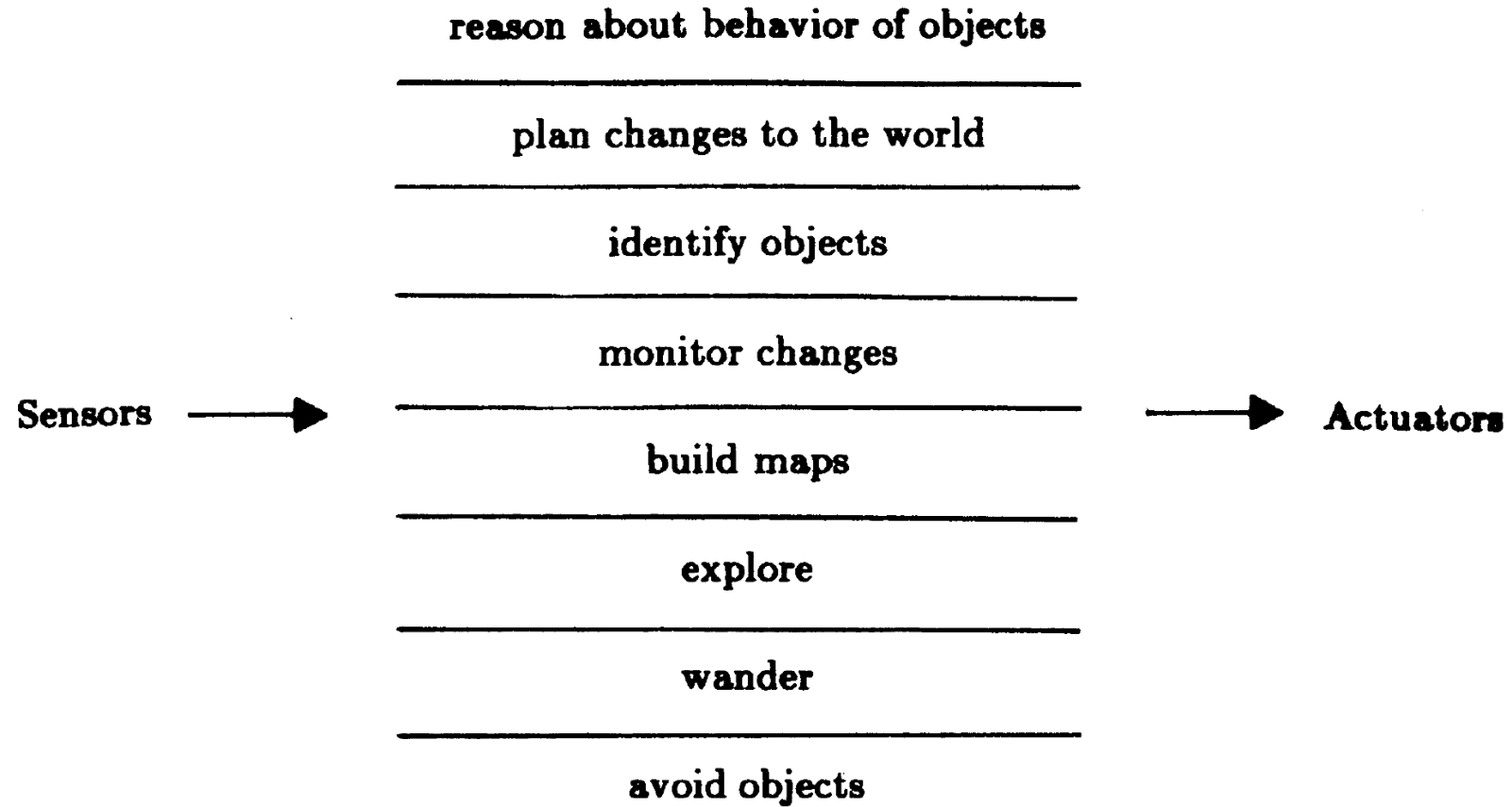
Figure 2. A decomposition of a mobile robot control system based on task achieving behaviors.

Feb 1.

Pero Profe, cuándo voy a usar todo esto????

Why Learn Physics - Prof - When would we use this stuff??

SHOULD HAVE LISTENED TO THE PHYSICS PROF!

SPRING

$\downarrow$ $k x_1$

$\rightarrow x$

m

$\rightarrow F$
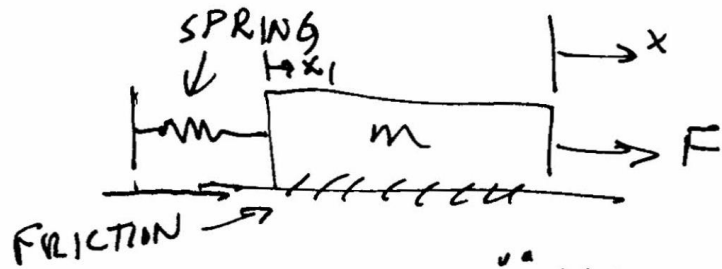
FRICTION $\rightarrow$

M CONSTANT

$F(t) = m\ddot{x}(t)$ NEWTON $M = \dfrac{W}{g} \left(\dfrac{lbs}{ft/s^2}\right)$

$F = K x_1(t)$ LINEAR SPRING

FRICTIONS (PLOT)

a) STATIC (STARTING OR STICTION)

$$F_s(t) = \pm F_s \Big|_{\dot{x}(t) = 0} \; ; \; \vec{F} = \mu_s \vec{N}$$

b) COULOMB FRICTION

$$F(t) = F_c \, sgn(t)$$

OPPOSES CONSTANT VELOCITY MOTION

c) VISCOUS FRICTION

$$F(t) = B \dot{x}(t)$$ DEPENDS ON VELOCITY

$$m\ddot{x}(t) + \beta\dot{x}(t) + kx(t) = f(t)$$

(HARMAN P221)

WE EXPECT

$$J\ddot{\theta}(t) + B\dot{\theta}(t) + K\theta(t) = T_{Applied}$$

$$J = \iiint \rho r^2 dV \qquad \text{IN GENERAL} \quad J = c\,MR^2$$
$$\text{volume} \qquad\qquad\qquad\qquad 0 < c < 1.$$

The Rules of Energy and Work are Similiar

LINEAR

$$E_k = \frac{1}{2} m v_2$$

$$E_p = \frac{1}{2} k x^2$$

$$Work = \int F(s) ds$$

$$= FS$$

If F is constant in direction s

$$Power = \frac{dw}{dt}$$

$$P(t) = F(t) v(t)$$

ROTATION

$$E_k = \frac{1}{2} J \omega^2$$

$$E_p = \frac{1}{2} k \theta^2$$

$$W = T\theta$$

$$P(t) = T(t) \omega(t)$$

(Δ Kinetic Energy)
+Δ Potential Energy

EQ 3.2.36
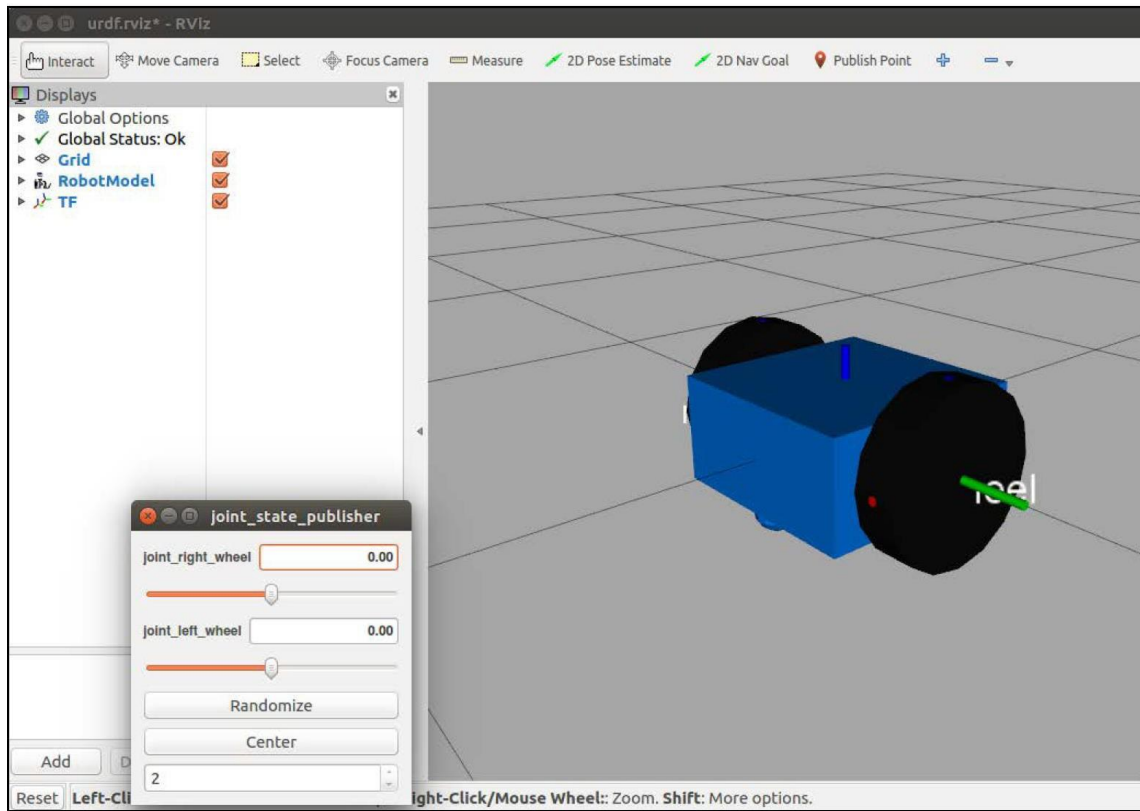P118

$$WATTS \rightarrow \frac{Kwatts \cdot Hours}{Time}$$

Inertia

Stopping_Distances

MotorSelection_Videos

Inertia and URDF Chapter 2 in Textbook (5435 Web – Text)

dd_robot5.urdf in rviz

PAGE 58 IN TEXTBOOK URDF FOR DD ROBOT

```xml
<?xml version='1.0'?>
<robot name="dd_robot">

    <!-- Base Link -->
    ...
    <inertial>
        <mass value="5"/>
        <inertia ixx="0.13" ixy="0.0" ixz="0.0"
                 iyy="0.21" iyz="0.0" izz="0.13"/>
    </inertial>

    <!-- Caster -->
    ...
    <inertial>
        <mass value="0.5"/>
        <inertia ixx="0.0001" ixy="0.0" ixz="0.0"
                 iyy="0.0001" iyz="0.0" izz="0.0001"/>
    </inertial>
</link>

<!-- Right Wheel -->
...
    <inertial>
        <mass value="0.5"/>
        <inertia ixx="0.01"  ixy="0.0" ixz="0.0"
                 iyy="0.005" iyz="0.0" izz="0.005"/>
    </inertial>
...
<!-- Left Wheel -->
...
    <inertial>
        <mass value="0.5"/>
        <inertia ixx="0.01" ixy="0.0" ixz="0.0"
                 iyy="0.005" iyz="0.0" izz="0.005"/>
    </inertial>
    ...
</robot>
```

https://navigation.ros.org/setup_guides/urdf/setup_urdf.html
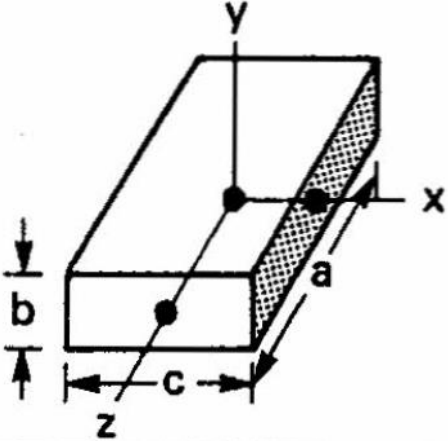
Setting Up The URDF¶
For this guide, we will be creating the Unified Robot Description Format (URDF) file for a simple differential drive robot to give you hands-on experience on working with URDF. We will also setup the robot state publisher and visualize our model in RVIZ.

Lastly, we will be adding some kinematic properties to our robot URDF to prepare it for simulation purposes. These steps are necessary to represent all the sensor, hardware, and robot transforms of your robot for use in navigation.
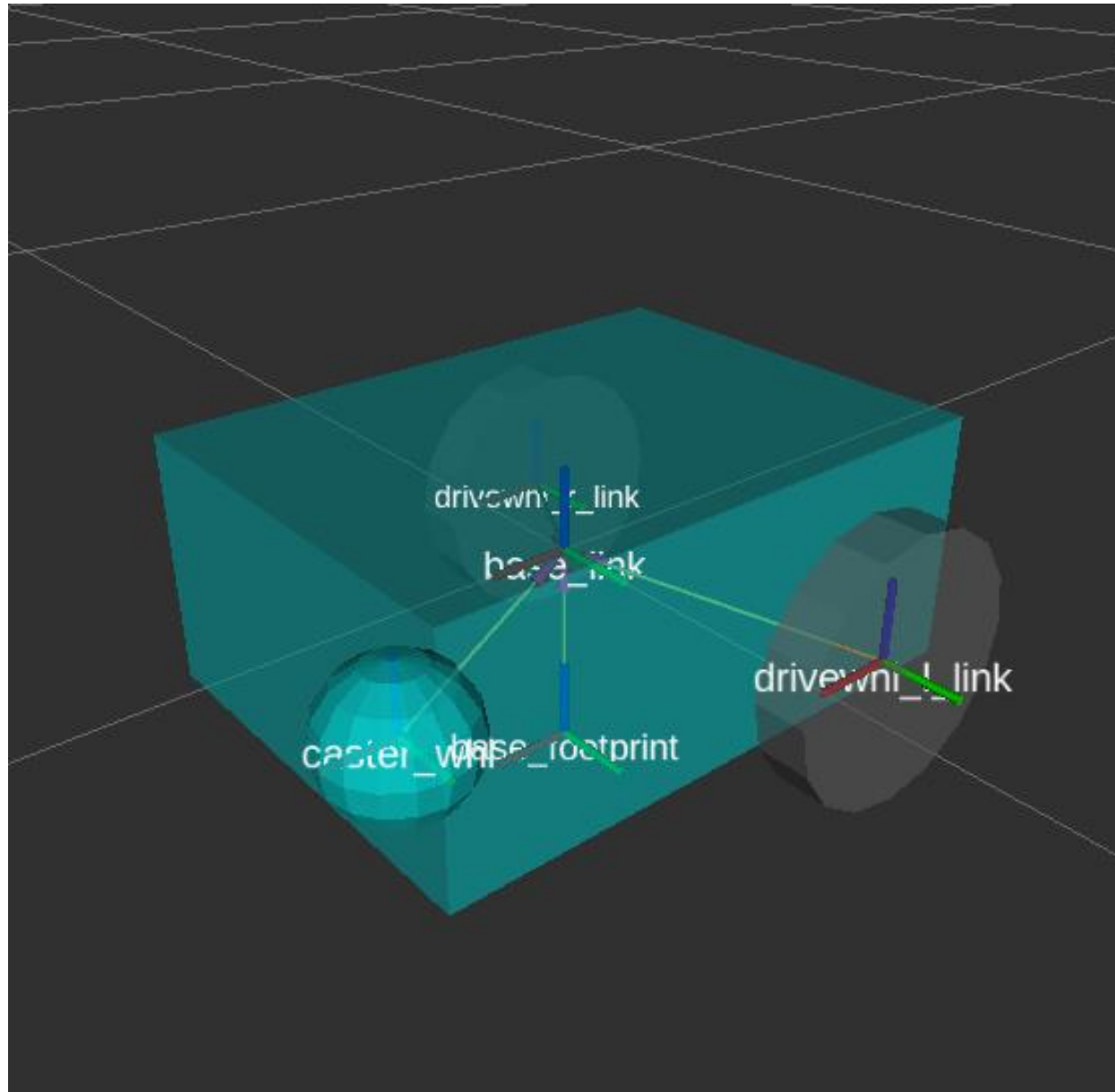
```xml
<!-- Define robot constants -->
  <xacro:property name="base_width"
value="0.31"/>
  <xacro:property name="base_length"
value="0.42"/>
  <xacro:property name="base_height"
value="0.18"/>

  <xacro:property name="wheel_radius"
value="0.10"/>
  <xacro:property name="wheel_width"
value="0.04"/>
  <xacro:property name="wheel_ygap"
value="0.025"/>
  <xacro:property name="wheel_zoff"
value="0.05"/>
  <xacro:property name="wheel_xoff"
value="0.12"/>

  <xacro:property name="caster_xoff"
value="0.14"/>
```

https://navigation.ros.org/_images/base-bot_1.png

	Rectangular Parallelepiped	$\frac{1}{12} M(a^2 + b^2)$	$\frac{1}{12} M(a^2 + c^2)$	$\frac{1}{12} M(b^2 + c^2)$

```xml
<xacro:property name="caster_xoff" value="0.14"/>
<!-- Define intertial property macros  -->
  <xacro:macro name="box_inertia" params="m w h d">
    <inertial>
      <origin xyz="0 0 0" rpy="${pi/2} 0 ${pi/2}"/>
      <mass value="${m}"/>
      <inertia ixx="${(m/12) * (h*h + d*d)}" ixy="0.0"
ixz="0.0" iyy="${(m/12) * (w*w + d*d)}" iyz="0.0"
izz="${(m/12) * (w*w + h*h)}"/>
    </inertial>
  </xacro:macro>
```

```xml
<xacro:macro name="cylinder_inertia" params="m r
h">
  <inertial>
    <origin xyz="0 0 0" rpy="${pi/2} 0 0" />
    <mass value="${m}"/>
    <inertia ixx="${(m/12) * (3*r*r + h*h)}" ixy
= "0" ixz = "0" iyy="${(m/12) * (3*r*r + h*h)}" iyz
= "0" izz="${(m/2) * (r*r)}"/>
  </inertial>
</xacro:macro>


<xacro:macro name="sphere_inertia" params="m r">
  <inertial>
    <mass value="${m}"/>
    <inertia ixx="${(2/5) * m * (r*r)}" ixy="0.0"
ixz="0.0" iyy="${(2/5) * m * (r*r)}" iyz="0.0"
izz="${(2/5) * m * (r*r)}"/>
  </inertial>
</xacro:macro>
```
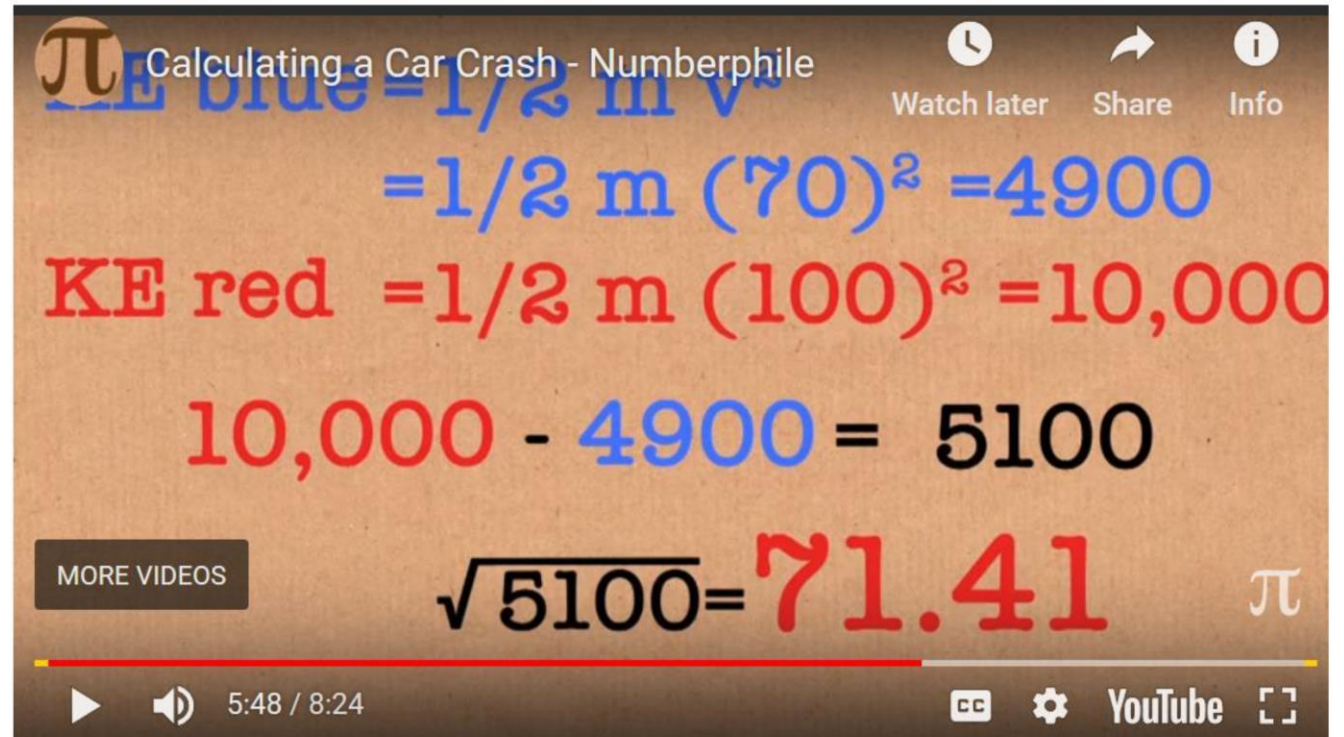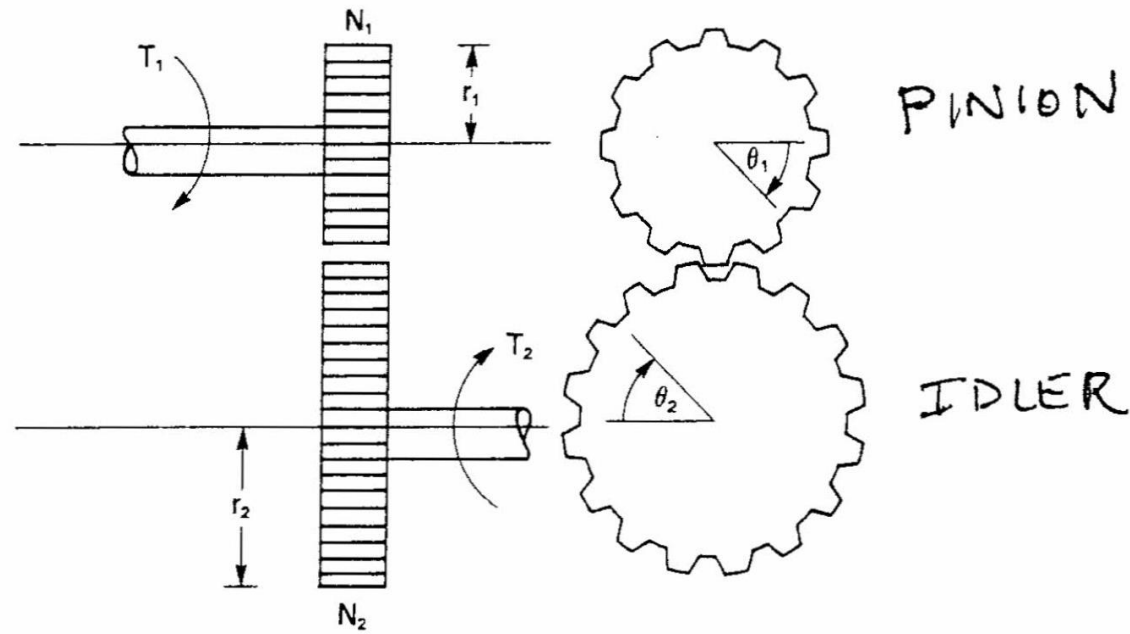
THINK AH

THINK AHEAD

$KE\ blue = 1/2\ m\ v^2$

$= 1/2\ m\ (70)^2 = 4900$

$KE\ red = 1/2\ m\ (100)^2 = 10,000$

$10,000 - 4900 = 5100$

$\sqrt{5100} = 71.41$

Loss in KE indicates 71MPH at crash!!

Finally, noting that since the two gear radii do not vary with time, if Eqs. (3.3.2) and (3.3.3) are differentiated with respect to time, their relationship still holds but with respect to $\dot{\theta}$ (i.e., the angular velocity ω^*) or $\ddot{\theta}$ (i.e., the angular acceleration, α). Using this concept, we may write

$$\frac{N_1}{N_2} = \frac{r_1}{r_2} = \frac{T_1}{T_2} = \frac{\theta_2}{\theta_1} = \frac{\omega_2}{\omega_1} = \frac{\alpha_2}{\alpha_1} \qquad (3.3.4)$$
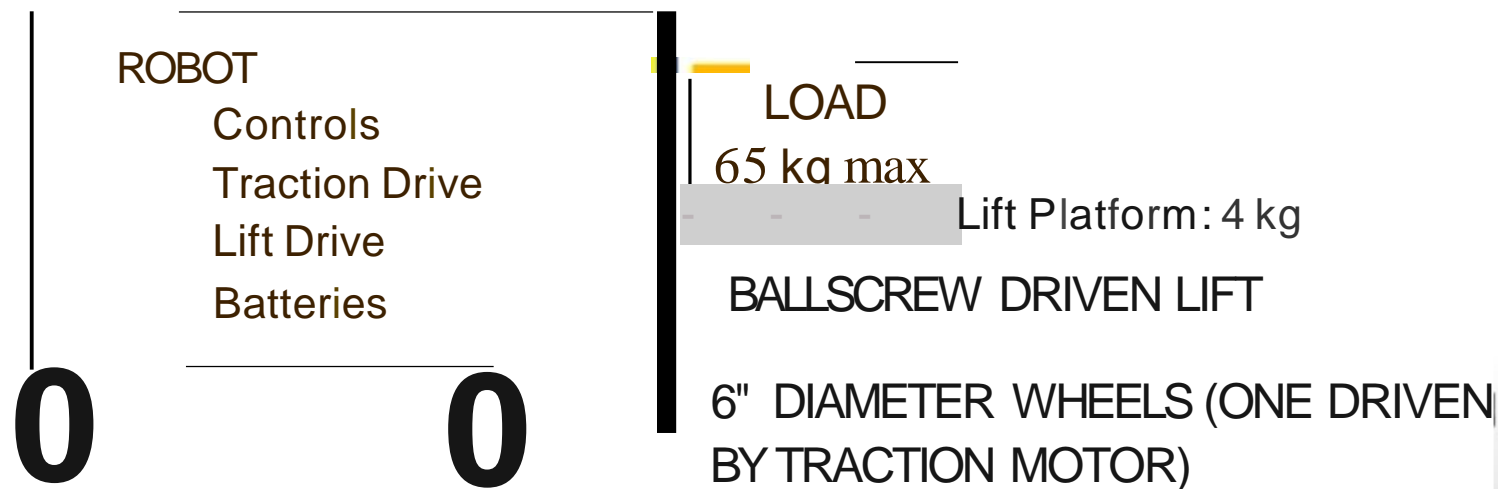
Example System

- Customer driven requirements include:
  - Max load weight (65kg)
  - Battery voltage (168Vdc)
  - Throughput, longest move time (≤ 40s)

https://www.youtube.com/watch?v=VJFDU31LQGM&app=desktop

Example System

- Autonomous Warehouse Fork Lift For Picking Up Pallets and Transporting Them

ROBOT
    Controls
    Traction Drive
    Lift Drive
    Batteries

LOAD
65 kg max

Lift Platform: 4 kg

BALLSCREW DRIVEN LIFT

6" DIAMETER WHEELS (ONE DRIVEN BY TRACTION MOTOR)

TOTAL WEIGHT (ROBOT AND LOAD)= 418 p o u n d s - 190 kg mass
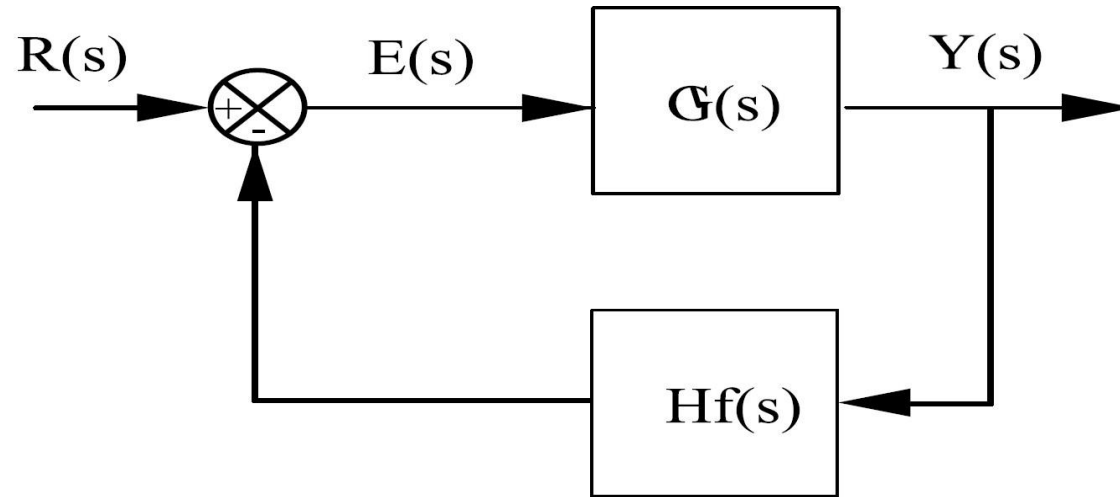
FEEDBACK

FIGURE 1.7 *Feedback control model*

If the input $R(s)$ is constant, we say the control system is a *regulator* since the object is to maintain the output at some constant value in the presence of disturbances. Temperature control or speed-control closed-loop systems are of this type. Other control systems are designed to allow the output to follow some time function as input. Such systems that control mechanical position or motion are called *servomechanisms*.

The transfer function $Y(s)/R(s)$ for the system in Figure 1.7 is derived by observing that

$$Y(s) = G(s)E(s)$$
$$E(s) = R(s) - H_f(s)Y(s)$$

and eliminating $E(s)$ from these equations to yield

$$G = 500 \qquad \text{so From Equation 1.16 in web}$$

$$TF_1: \text{Gain}_1 = \frac{Y(s)}{R(s)} = \frac{500}{1 + 500(0.09)} = \frac{500}{46} = 10.87 \quad \text{5pts}$$

$$TF_2 = \text{Gain}_2 = \frac{Y(s)}{R(s)} = \frac{450}{1 + 450(0.09)} = \frac{450}{41.5} = 10.84 \quad \text{5pts}$$

$$\Delta \text{Gain} = \frac{10.87 - 10.84}{10.87} = \frac{0.030}{10.87} = 0.0028$$

$$\text{OR} \quad 0.28\% \qquad \text{5pts}$$

$$\Delta G(s) = \frac{500 - 450}{500} = \frac{50}{500} = 0.100$$

$$\text{OR} \quad \underline{10\%} \qquad \text{5pts}$$

FEEDBACK IS GREAT

DC Motor

Let us find the poles [i.e., the roots of the denominator polynomial of $G_m(s)$] of a commercially available dc servomotor. For example, the parameters of an Electrocraft Corporation E530 motor are:

TE P719
OF MOTORS
PENDIX B

$K_T = 10.02$ oz-in./A   TORQUE CONST.

$K_E = 7.41$ V/1000 rpm   BACK EMF

$R_a = 1.64\ \Omega$ (including brush resistance)

$L_a = 3.39$ mH

$B = 0.1$ oz-in./1000 rpm
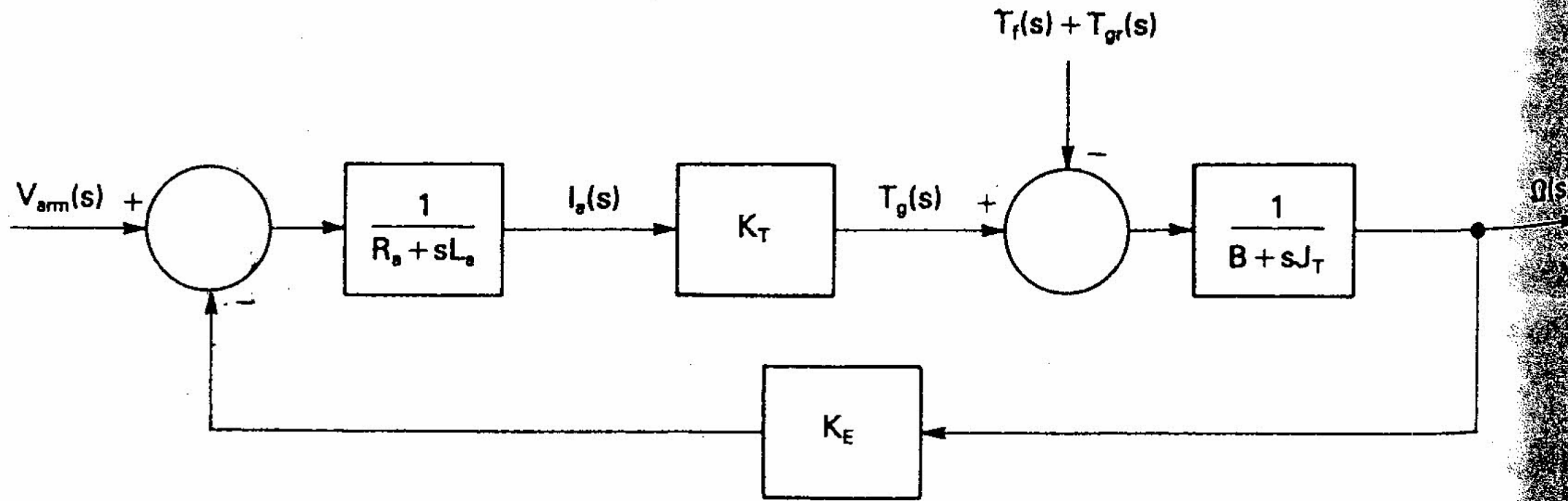
$J_M = 0.0038$ oz-in.-s²   ROTOR INERTIA

Figure 4.3.3. Block diagram of DC servomotor including gravitational and friction disturbance torques.

Klafter

$$G_m(s) = \frac{\Omega(s)}{V_{\text{arm}}(s)}$$

$$= \frac{K_T/L_aJ_T}{s^2 + [(R_aJ_T + L_aB)/L_aJ_T]s + (K_TK_E + R_aB)/L_aJ_T} \qquad (4.3.5)$$

$$G_m(s) = \frac{7.778 \times 10^5}{s^2 + 484.027s + 5.516 \times 10^4}$$

□ **EXAMPLE 5.15**   *Reduction of a Second-Order Equation*
Consider the second-order equation

$$m\ddot{x}(t) + b\dot{x}(t) + kx(t) = f(t).$$

Using the principles just defined, set $x_1(t) = x(t)$ and $x_2(t) = \dot{x}(t)$, so the first-order system becomes

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

$$\dot{x}_2(t) = -\frac{k}{m}x_1(t) - \frac{b}{m}x_2(t) + \frac{f(t)}{m}.$$

The matrix equation is

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -\dfrac{k}{m} & -\dfrac{b}{m} \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \frac{f(t)}{m}.$$

$$\text{FOR } \ddot{x} + 16\dot{x} + 64x = u(t) \qquad \vec{x}(t) = \begin{cases} x_1(t) \\ x_2(t) \end{cases}$$

$$\dot{\vec{x}}(t) = \begin{bmatrix} 0 & 1 \\ -64 & -16 \end{bmatrix} \vec{x} + \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} u(t)$$

output is $x(t)$

$$x(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$
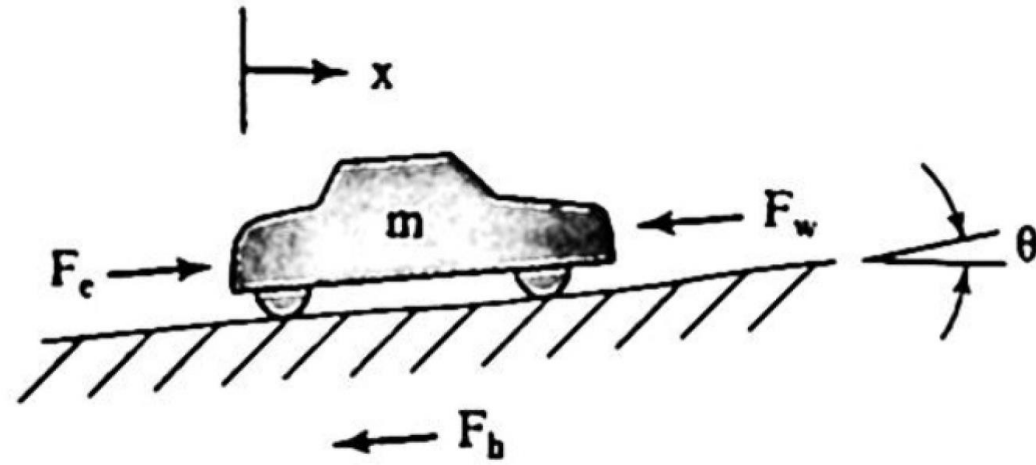
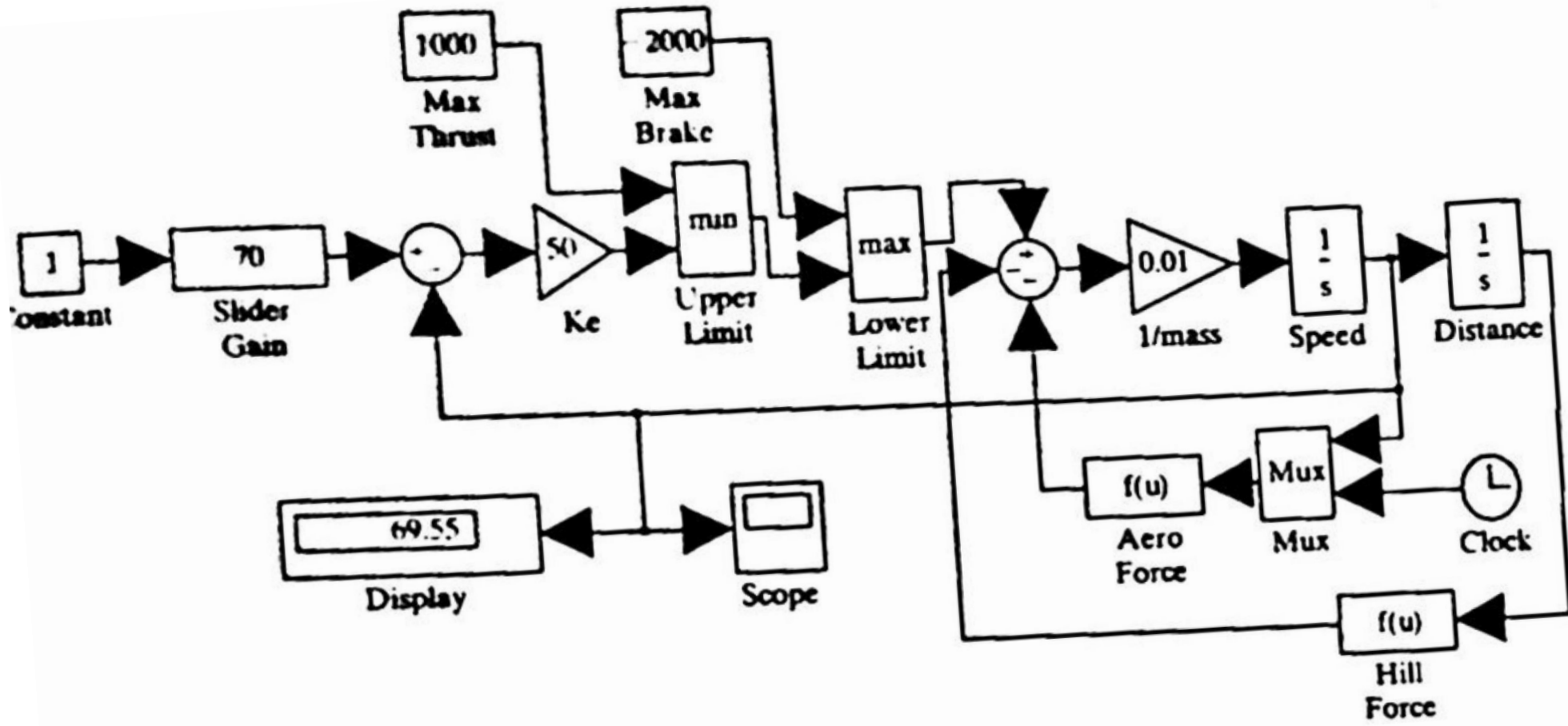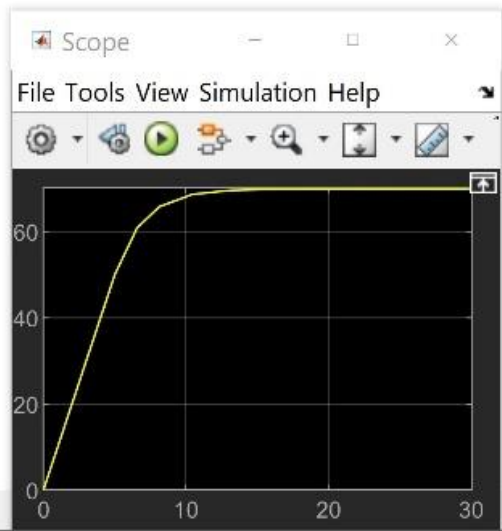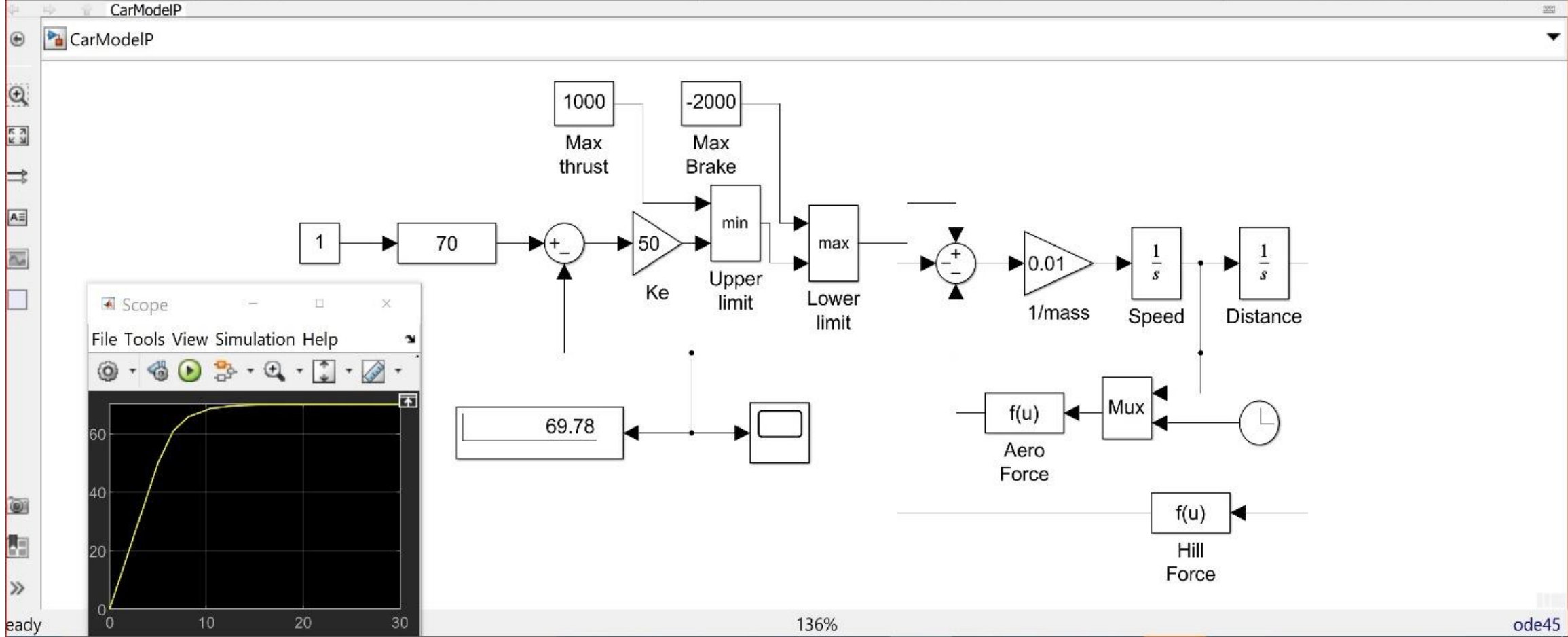$$m\ddot{x} = F_e - F_w - F_h$$

FIGURE 5.20: Automobile on a hilly road

FIGURE 5.21: Automobile model with proportional speed control

Linearization

Suppose the acceleration of an automobile can be described by the equation of motion

$$M\frac{dv(t)}{dt} = cu(t) - \alpha v^2(t), \qquad (12.33)$$

where the first term represents the acceleration caused by the engine at a throttle setting u and the second term is the drag caused by air resistance. Since this force is proportional to the square of the speed, the equation is nonlinear. Solving this by numerical techniques would not be difficult if the constants

Take (U_0, V_0) to be the "operating point," so that the car travels at a constant speed V_0 for a constant throttle position U_0. Inserting this condition into Equation 12.33 yields

$$M\frac{dV_0}{dt} = cU_0 - \alpha V_0^2(t) = 0,$$

or $V_0 = \sqrt{cU_0/\alpha}$. Let us assume that a small change in the throttle position leads to a small change in speed. Thus, we set

$$u(t) = U_0 + \Delta u, \quad v(t) = V_0 + \Delta v$$

and substitute again into the equation of motion. The result is

$$M\frac{d}{dt}[V_0 + \Delta v] = c[U_0 + \Delta u] - \alpha[V_0 + \Delta v]^2.$$

Using the result that $cU_0 = \alpha V_0^2$ and expanding the terms but neglecting the second-order term $-\alpha\Delta v^2$ leads to the approximation

$$M\frac{d}{dt}[\Delta v] \approx c[\Delta u] - 2\alpha V_0 \Delta v.$$

Writing $\Delta v = v_a(t)$ and $\Delta u = u_a(t)$ to indicate the linearized variables, the original differential equation of motion described by Equation 12.33 becomes the linear differential equation

$$M\frac{dv_a(t)}{dt} + 2\alpha V_0 v_a(t) = cu_a(t). \tag{12.34}$$

LINEAR MODEL – GOOD OVER A LIMITED RANGE

TWO-DIMENSIONAL TAYLOR SERIES

The notion of sequences and series of functions of a single variable as described in Chapter 6 can be extended to functions of several variables. For example, the power series expansion for a function of two variables $F(x, y)$ is

$$F(x, y) = \sum_{n=0}^{\infty} f_n(x, y) = f_0(x, y) + f_1(x, y) + \cdots + f_n(x, y) + \cdots, \quad (12.35)$$

with the terms

$$f_n(x, y) = c_{n,0}\, x^n + c_{n,1}\, x^{n-1} y + \cdots + c_{n,n-1}\, xy^{n-1} + c_{n,n}\, y^n.$$

IN MOST ROBOT PROBLEMS, WE DEAL WITH MULTIVARIATE FUNCTIONS; i,.e.   POSE AS   (X,Y, Φ) FOR DD-ROBOT

Control methods for Well- behaved systems – Particularly for linear time-invariant* (LTI) system.

Although **PID control** is the most common type of industrial controller, it does have limitations.

First, PID control is generally not suitable for systems with multiple inputs and multiple outputs (**MIMO**), as the transfer functions and differential equations used to represent the system become overly complex when more then one input (or output) is involved.

Second, PID control is based on constant parameters, so its effectiveness in controlling non-linear systems is limited.

https://www.motioncontroltips.com/what-is-state-space-control/

An alternative control method is state space control. The key difference between PID control (aka "transfer control") and state space control is that the state space method takes into **account the internal state of the system**, through what are referred to as "state variables."

These state variables describe the system and its response to any given set of inputs.

PID control, on the other hand, relies on an "observer," which estimates the internal state of the system based on **measured inputs and outputs.**
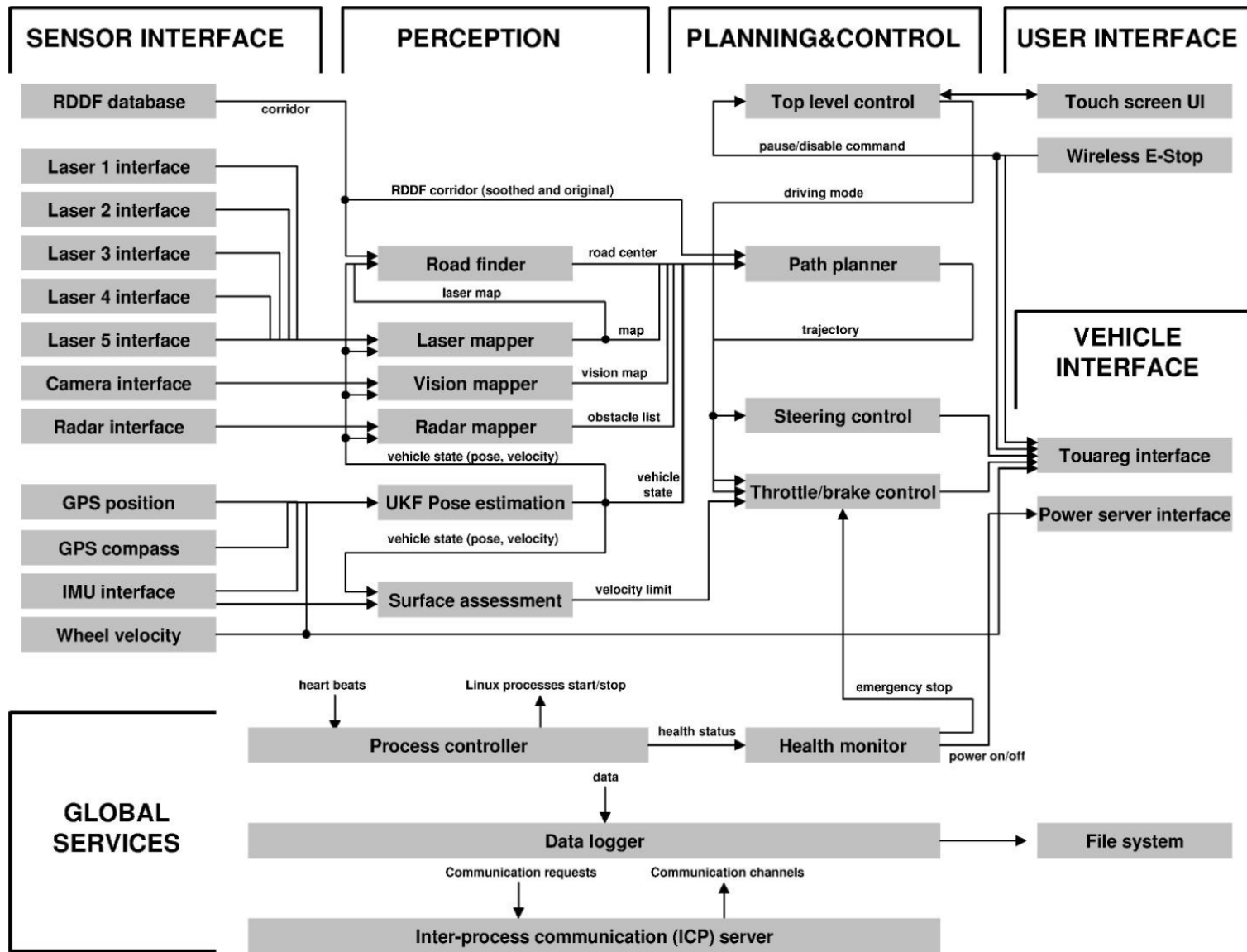
Figure 2: Software flowchart: The software is divided into six functional groups: sensor interface, perception, control, vehicle interface, user interface, and global services.