4

Principles of Planar Mechanical System Modeling

Preview

In this chapter, we deal with the dynamic behavior of translational and rotational planar motions. The basic physical law governing these types of systems can be either expressed in terms of Newton's law for translational and rotational motions, or in terms of the d'Alembert principle. The concepts will be demonstrated by means of a number of practical examples such as a crane crab system and an inverted pendulum. Towards the end of the chapter, we discuss electro–mechanical transducers.

4.1 Introduction

Mechanical systems are quite similar to electrical systems. Some of the basic principles are exactly the same for both types of systems. In electrical circuit modeling, we have learned that the sum of all currents in a node must always be zero. In modeling rigid mechanical systems, we shall see that all internal forces and torques at any point of the body must add up to zero. In electrical circuit modeling, we have learned that the potentials of all connecting branches at a node must be the same. In mechanical systems, it is true that the positions, velocities, and accelerations, both translational and rotational must be the same at any connecting point in the system.

In fact, some electrical engineers like to convert mechanical systems to equivalent electrical circuits, and thereafter treat those using the techniques that were discussed in Chapter 3. We shall not do so, but we shall see later (namely in Chapter 7) that a deeper truth lies behind these similarities.

However, differences exist also which make it a little harder to deal with mechanical systems than with electrical circuits. Let me summarize these dissimilarities:

(1) Geometry plays an important role in mechanical systems. This is not so in electrical systems unless one goes to very high frequency (microwave frequencies) or to very small dimensions (integrated circuits). Mechanical systems operate in a three–dimensional space which is difficult to capture on two–dimensional drawings. Therefore, it is a little more difficult to grasp the functioning of most mechanical devices. However, this is compensated by the fact that electrical circuits often contain thousands of circuit elements. Mechanical systems are always rather simple devices.

(2) To model electrical circuits, only two types of signals were required: *voltages* and *currents*. Every facet of an electrical circuit can be described in terms of these two quantities. In mechanical systems, each body can be exerted simultaneously by a *force* (which usually has an x, a y, and a z component), and by a *torque* (again with three components). As a consequence, we need to operate on a larger number of variables in order to capture all mechanical properties of a system.

(3) Mechanical systems are always subjected to *constraints*. Masses can bump into each other, or can fall down; springs cannot be compressed or pulled to an arbitrary extent; bodies are not infinitely stiff in reality, but can sag or can be deformed otherwise (elastically or even plastically). This is, of course, also true for electrical systems, but to a much lesser extent. Electrical systems are much "cleaner" than mechanical systems.

What this really means is that electrical systems exhibit a more crisp separation of the various governing physical phenomena both in the *time* as well as in the *space* dimensions. Most electrical circuits operate in the *kHz* to *MHz* range. In this frequency range, the electrical phenomena as studied in Chapter 3 are strongly dominant. Mechanical and thermal side effects are much slower, usually too slow to be considered in the model except for DC analysis. Quantum mechanical effects are much faster, usually so fast that they can either be ignored altogether, or at least, their influence can be aggregated to a statistical description (noise analysis). Also, in the *kHz* to *MHz* range, the geometry of the circuit layout can still be ignored (except inside an integrated circuit chip). On the other hand, mechanical systems operate in the *Hz* range. Therefore, it may be necessary to consider thermal side effects. More important, however, are the geometrical (spatial) influences. A simulation system which does not allow us to formulate the geometrical constraints inherent in the model is therefore virtually worthless.

However, let us start with the most simple principle that governs the mechanical behavior of rigid bodies. This principle has been formulated first by Sir Isaac Newton, and describes the dynamics of a rigid body both in translational and rotational terms.

4.2 Newton's Law for Translational Motions

Newton's law is often quoted as follows: The sum of all forces exerted on a rigid body equals the mass of the body multiplied by its acceleration, i.e.:

$$m \cdot a = \sum_{\forall i} f_i \tag{4.1}$$

However a little more precisely, the law should be written as:

$$\frac{d(m \cdot v)}{dt} = \sum_{\forall i} f_i \tag{4.2}$$

where the term $m \cdot v$ (the mass multiplied by the velocity) is sometimes called the *momentum* $\mathcal{I}$ of the rigid body, i.e.:

$$\frac{d\mathcal{I}}{dt} = \sum_{\forall i} f_i \tag{4.3}$$

This distinction becomes important if the body is not all that "rigid" after all, but loses mass on the way, such as our lunar landing module from Chapter 2. Fig.4.1 illustrates Newton's law as applied to our lunar landing module.

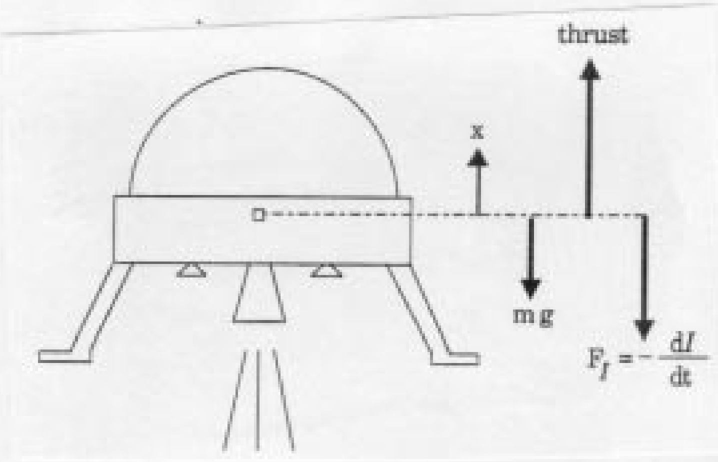


Figure 4.1. Lunar landing module

For this system, Newton's law can be written as:

$$\frac{d(m \cdot v)}{dt} = \frac{dm}{dt} \cdot v + m \cdot \frac{dv}{dt} = thrust - m \cdot g \tag{4.4}$$

which can be rewritten as:

$$m \cdot a = thrust - m \cdot g - \frac{dm}{dt} \cdot v \tag{4.5}$$

which is the form that we had used in Chapter 2.

In an alternative approach, we can introduce a fictitious "mass force":

$$F_{\mathcal{I}} = -\frac{d\mathcal{I}}{dt} \tag{4.6}$$

and, adding this "force" to our set of forces acting on the rigid body, we can reformulate Newton's law as follows:

$$\sum_{\forall i} f_i = 0 \tag{4.7}$$

In this modified form, Newton's law is known as the *d'Alembert principle.* The two formulations are equivalent.

Let us exercise Newton's law by means of the simple mechanical system that is depicted in Fig.4.2.

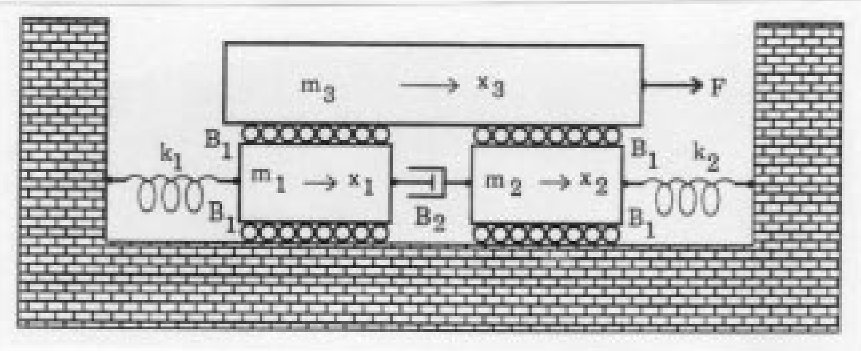


Figure 4.2. Simple translational problem

Let us assume that all elements of this mechanical system operate in their linear range. The top body m_3 does not fall down, neither does it lose any of its wheels. Also, the top body does not sag, and always covers all the wheels that separate it from the two lower bodies m_1 and m_2. The springs k_1 and k_2 do not overexpand or overcontract, and the same is true for the hydraulic cylinder B_2. Let us assume furthermore that, for all times $t < 0$, the driving force F is zero, and the system is in an equilibrium state in which all three positions x_1, x_2, and x_3 are defined as zero.

This problem is very simple. One approach to tackle such problems is to freeze all bodies but one, and see what happens to that body when we try to move it. Let us start with body m_3. If we apply a force F at time $t = 0$ pulling m_3 to the right, the two frictions between m_3 and the two other bodies m_1 and m_2, which are told not to move, will *oppose* our attempts. This settles the question of the sign, and we find immediately the equation:

$$m_3 \frac{d^2 x_3}{dt^2} = F - B_1(\frac{dx_3}{dt} - \frac{dx_2}{dt}) - B_1(\frac{dx_3}{dt} - \frac{dx_1}{dt}) \tag{4.8a}$$

The *reaction* to these friction forces (i.e., the same forces but with opposite signs) are responsible for getting the bodies m_1 and m_2 moving. Thus, we can for instance write the equation for body m_2:

$$m_2 \frac{d^2 x_2}{dt^2} = B_1(\frac{dx_3}{dt} - \frac{dx_2}{dt}) - B_1 \frac{dx_2}{dt} - B_2(\frac{dx_2}{dt} - \frac{dx_1}{dt}) - k_2\ x_2 \tag{4.8b}$$

and finally for body m_1:

$$m_1 \frac{d^2x_1}{dt^2} = B_1(\frac{dx_3}{dt} - \frac{dx_1}{dt}) - B_1 \frac{dx_1}{dt} + B_2(\frac{dx_2}{dt} - \frac{dx_1}{dt}) - k_1\ x_1 \qquad (4.8c)$$

In order to obtain a set of state–space equations, we need to solve eq(4.8a-c) for their highest derivatives, and reduce the second order differential equations to sets of first order differential equations. The resulting state equations are as follows:

$$\dot{x}_1 = v_1 \qquad (4.9a)$$

$$\dot{v}_1 = \frac{1}{m_1}[-k_1\ x_1 - (2B_1 + B_2)v_1 + B_2\ v_2 + B_1\ v_3] \qquad (4.9b)$$

$$\dot{x}_2 = v_2 \qquad (4.9c)$$

$$\dot{v}_2 = \frac{1}{m_2}[B_2\ v_1 - k_2\ x_2 - (2B_1 + B_2)v_2 + B_1\ v_3] \qquad (4.9d)$$

$$\dot{x}_3 = v_3 \qquad (4.9e)$$

$$\dot{v}_3 = \frac{1}{m_3}[B_1\ v_1 + B_2\ v_2 - 2B_3\ v_3 + F(t)] \qquad (4.9f)$$

which can be coded directly in any of the simulation systems that were introduced in Chapter 2.

The above approach works well for simple problems, but it can become confusing when more parts are involved that move in all directions and rotate at the same time. In those more complicated cases, another approach works better. This will be illustrated next.

In the second approach, we start again by identifying parts of the system that can be moved without the rest of the system moving with them. We now cut the system open at the interface between the moving subsystem and the frozen subsystems, and replace the influence of the frozen subsystems on the moving subsystem by an equivalent force acting on the moving subsystem, and the *reaction* (i.e., the influence of the moving subsystem on the frozen subsystems) by an equivalent force acting on the frozen subsystems. These two *internal forces* are always of the same size but of opposite direction (i.e., they annihilate each other when the system is recombined). Fig.4.3 demonstrates this approach for the case of our simple mechanical system.

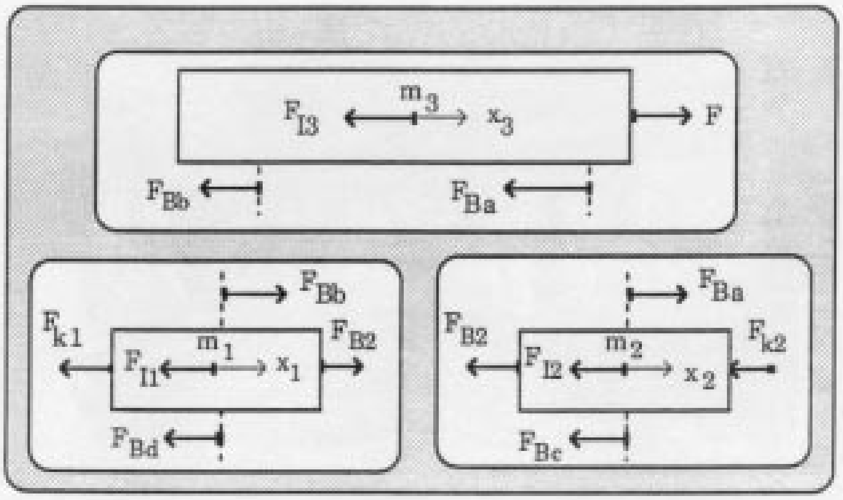


Figure 4.3. Simple translational problem cut open

Now, we can apply the *d'Alembert principle* to the three bodies separately, and write those equations down together with the equations governing the behavior of the individual forces:

$$F(t) = F_{I_3} + F_{B_a} + F_{B_b} \tag{4.10a}$$

$$F_{B_a} = F_{I_2} + F_{B_c} + F_{B_2} + F_{k_2} \tag{4.10b}$$

$$F_{B_b} + F_{B_2} = F_{I_1} + F_{B_d} + F_{k_1} \tag{4.10c}$$

$$F_{I_1} = m_1 \frac{dv_1}{dt} \tag{4.10d}$$

$$\frac{dx_1}{dt} = v_1 \tag{4.10e}$$

$$F_{I_2} = m_2 \frac{dv_2}{dt} \tag{4.10f}$$

$$\frac{dx_2}{dt} = v_2 \tag{4.10g}$$

$$F_{I_3} = m_3 \frac{dv_3}{dt} \tag{4.10h}$$

$$\frac{dx_3}{dt} = v_3 \tag{4.10i}$$

$$F_{B_a} = B_1(v_3 - v_2) \tag{4.10j}$$

$$F_{B_b} = B_1(v_3 - v_1) \tag{4.10k}$$

$$F_{B_c} = B_1 \; v_2 \tag{4.10l}$$

$$F_{B_d} = B_1 \; v_1 \tag{4.10m}$$

$$F_{B2} = B_2(v_2 - v_1) \tag{4.10n}$$
$$F_{k1} = k_1 \; x_1 \tag{4.10o}$$
$$F_{k2} = k_2 \; x_2 \tag{4.10p}$$

Notice that the directions of the arrows of the internal forces are arbitrary (as were the directions of currents and voltages in electrical circuits). However, we must adjust the equations to our conventions. This is demonstrated in Fig.4.4. If friction forces and/or spring forces have the opposite direction to the *position* (and velocity and acceleration) of a rigid body, the contribution of the body itself is counted positively in the force equation while the contribution of the environment is counted negatively. If the directions are the same, the contribution of the environment is counted positively, while the contribution of the body itself is counted negatively. If the direction of the inertial force is opposite to the direction of the position, the equation is entered with a plus sign, otherwise with a minus sign.

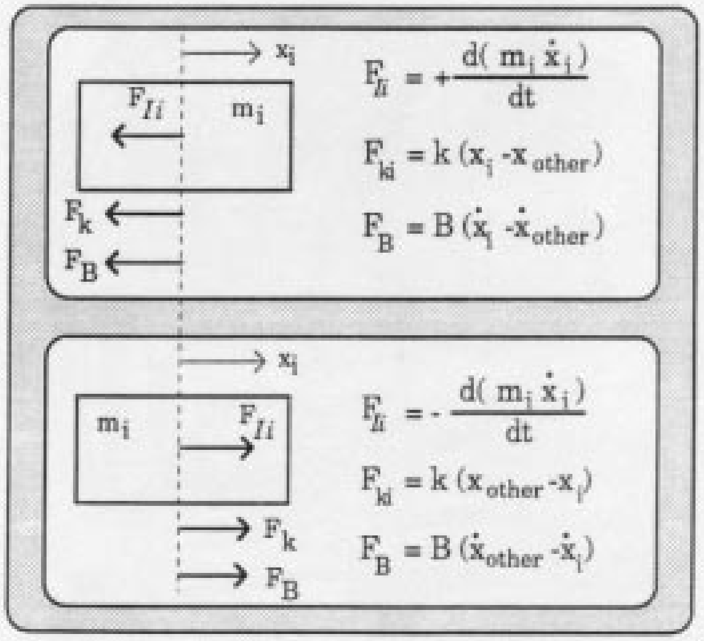

x_i
F_{Ii}
m_i
F_k
F_B
$F_{Ii} = +\frac{d(m_i \dot{x}_i)}{dt}$
$F_{ki} = k(x_i - x_{other})$
$F_B = B(\dot{x}_i - \dot{x}_{other})$
x_i
m_i
F_{Ii}
F_k
F_B
$F_{Ii} = -\frac{d(m_i \dot{x}_i)}{dt}$
$F_{ki} = k(x_{other} - x_i)$
$F_B = B(\dot{x}_{other} - \dot{x}_i)$

Figure 4.4. Convention for direction of forces

Now, we can try to *solve* these equations using exactly the same methodology that had been advocated in Chapter 3. As in Chapter 3, we start by solving all differential equations for their derivative terms, and proceed until all equations and unknowns have been used up.

$$F(t) = [F_{I3}] + F_{Ba} + F_{Bb} \tag{4.11a}$$

$$F_{Ba} = [F_{I2}] + F_{Bc} + F_{B2} + F_{k2} \tag{4.11b}$$

$$F_{Bb} + F_{B2} = [F_{I1}] + F_{Bd} + F_{k1} \tag{4.11c}$$

$$F_{I1} = m_1[\frac{dv_1}{dt}] \tag{4.11d}$$

$$[\frac{dx_1}{dt}] = v_1 \tag{4.11e}$$

$$F_{I2} = m_2[\frac{dv_2}{dt}] \tag{4.11f}$$

$$[\frac{dx_2}{dt}] = v_2 \tag{4.11g}$$

$$F_{I3} = m_3[\frac{dv_3}{dt}] \tag{4.11h}$$

$$[\frac{dx_3}{dt}] = v_3 \tag{4.11i}$$

$$[F_{Ba}] = B_1(v_3 - v_2) \tag{4.11j}$$

$$[F_{Bb}] = B_1(v_3 - v_1) \tag{4.11k}$$

$$[F_{Bc}] = B_1 \; v_2 \tag{4.11l}$$

$$[F_{Bd}] = B_1 \; v_1 \tag{4.11m}$$

$$[F_{B2}] = B_2(v_2 - v_1) \tag{4.11n}$$

$$[F_{k1}] = k_1 \; x_1 \tag{4.11o}$$

$$[F_{k2}] = k_2 \; x_2 \tag{4.11p}$$

which can then be rewritten as:

$$F_{I3} = F(t) - F_{Ba} - F_{Bb} \tag{4.12a}$$

$$F_{I2} = F_{Ba} - F_{Bc} - F_{B2} - F_{k2} \tag{4.12b}$$

$$F_{I1} = F_{Bb} + F_{B2} - F_{Bd} - F_{k1} \tag{4.12c}$$

$$\frac{dv_1}{dt} = F_{I1}/m_1 \tag{4.12d}$$

$$\frac{dx_1}{dt} = v_1 \tag{4.12e}$$

$$\frac{dv_2}{dt} = F_{I2}/m_2 \tag{4.12f}$$

$$\frac{dx_2}{dt} = v_2 \tag{4.12g}$$

$$\frac{dv_3}{dt} = F_{I3}/m_3 \tag{4.12h}$$

$$\frac{dx_3}{dt} = v_3 \tag{4.12i}$$
$$F_{Ba} = B_1(v_3 - v_2) \tag{4.12j}$$
$$F_{Bb} = B_1(v_3 - v_1) \tag{4.12k}$$
$$F_{Bc} = B_1 \; v_2 \tag{4.12l}$$
$$F_{Bd} = B_1 \; v_1 \tag{4.12m}$$
$$F_{B2} = B_2(v_2 - v_1) \tag{4.12n}$$
$$F_{k1} = k_1 \; x_1 \tag{4.12o}$$
$$F_{k2} = k_2 \; x_2 \tag{4.12p}$$

which again can be programmed immediately using any of the previously introduced simulation languages.

4.3 Newton's Law for Rotational Motions

This version of Newton's law is often quoted as follows: The sum of all torques exerted on a rigid body equals the inertia of the body multiplied by its angular acceleration, i.e.:

$$J\dot{\omega} = \sum_{\forall i} \tau_i \tag{4.13}$$

However a little more precisely, the law should be written as:

$$\frac{d(J\omega)}{dt} = \sum_{\forall i} \tau_i \tag{4.14}$$

where the term $J\omega$ (the inertia multiplied by the angular velocity) is the *twist* $\mathcal{T}$ of the rigid body, i.e.:

$$\frac{d\mathcal{T}}{dt} = \sum_{\forall i} \tau_i \tag{4.15}$$

which is sometimes also called the *angular momentum*. As before, we can introduce a fictitious "inertial torque" $\tau_{\mathcal{T}}$:

$$\tau_T = -\frac{dT}{dt} \tag{4.16}$$

and reformulate Newton's law using the d'Alembert principle as:

$$\sum_{\forall i} \tau_i = 0 \tag{4.17}$$

Let me illustrate the modeling of rotational motions by means of another simple mechanical system as illustrated in Fig.4.5.

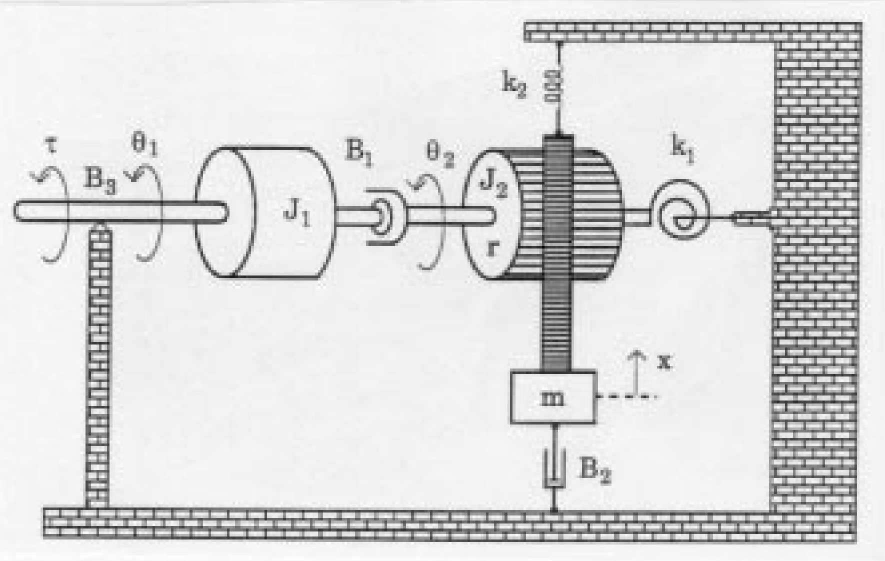


Figure 4.5. A simple rotational system

Since the system is sufficiently simple, we can proceed along the first route, and write second order differential equations right away. However, we first need to understand what the gear is doing to our system. This is illustrated in Fig.4.6, which describes the transformation of the rotational subsystem to the translational subsystem. The equations that govern the gear are the same irrespective of whether the cause is a torque applied to the pinion, or whether the cause is a force applied to the rack.

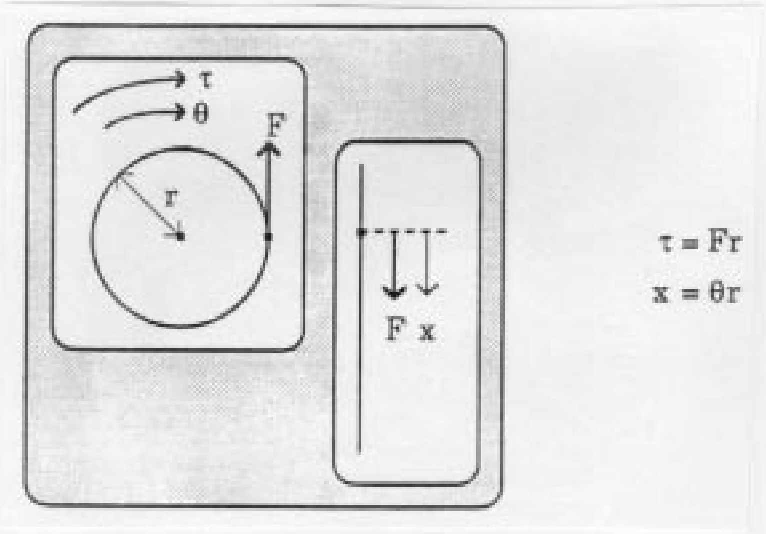


Figure 4.6. Modeling mechanical gears

Now, we are ready to write down the equations. Let us begin with the subsystem J_1:

$$J_1 \frac{d^2\theta_1}{dt^2} = \tau - B_1\left(\frac{d\theta_1}{dt} - \frac{d\theta_2}{dt}\right) - B_3 \frac{d\theta_1}{dt} \tag{4.18a}$$

Let me introduce two additional variables, namely a torque τ_G denoting the influence of the rack on the pinion J_2, and a force F_G denoting the influence of the pinion J_2 on the rack. This allows us to write an equation for the second subsystem (J_2):

$$J_2 \frac{d^2\theta_2}{dt^2} = B_1\left(\frac{d\theta_1}{dt} - \frac{d\theta_2}{dt}\right) - k_1\ \theta_2 - \tau_G \tag{4.18b}$$

and for the third subsystem:

$$m \frac{d^2x}{dt^2} = F_G - m\ g - B_2 \frac{dx}{dt} - k_2\ x \tag{4.18c}$$

Now, we only need to describe the gear:

$$\tau_G = r\ F_G \tag{4.18d}$$

$$x = r\ \theta_2 \tag{4.18e}$$

Of course, from eq(4.18e), we can immediately derive two more equations:

$$\frac{dx}{dt} = r\frac{d\theta_2}{dt} \tag{4.19a}$$

$$\frac{d^2x}{dt^2} = r\frac{d^2\theta_2}{dt^2} \tag{4.19b}$$

Let us eliminate x and F_G from eq(4.18c) by replacing these terms with θ_2 and τ_G:

$$m\ r\frac{d^2\theta_2}{dt^2} = \frac{1}{r}\tau_G - m\ g - B_2\ r\frac{d\theta_2}{dt} - k_2\ r\ \theta_2 \tag{4.20}$$

which can be solved for τ_G:

$$\tau_G = m\ r^2\frac{d^2\theta_2}{dt^2} + B_2\ r^2\frac{d\theta_2}{dt} + k_2\ r^2\ \theta_2 + m\ g\ r \tag{4.21}$$

Plugging eq(4.21) into eq(4.18b), and rearranging the terms, we find:

$$[J_2 + m\ r^2]\frac{d^2\theta_2}{dt^2} = B_1\frac{d\theta_1}{dt} - [B_1 + B_2\ r^2]\frac{d\theta_2}{dt} - [k_1 + k_2\ r^2]\theta_2 - m\ g\ r \tag{4.22}$$

The term $[J_2 + m\ r^2]$ is the *apparent inertia* of the body J_2, i.e., the inertia that is visible when we measure the inertia from the rotational end of the gear. Similarly, the terms $[B_1 + B_2\ r^2]$ and $[k_1 + k_2\ r^2]$ denote *apparent friction and spring coefficients.*

We are now ready to generate a set of state equations:

$$\dot{\theta}_1 = \omega_1 \tag{4.23a}$$

$$\dot{\omega}_1 = \frac{1}{J_1}[-(B_1 + B_2)\omega_1 + B_1\omega_2 + \tau(t)] \tag{4.23b}$$

$$\dot{\theta}_2 = \omega_2 \tag{4.23c}$$

$$\dot{\omega}_2 = \frac{1}{J_2 + m\ r^2}[B_1\omega_1 - (k_1 + k_2\ r^2)\theta_2 - (B_1 + B_2\ r^2)\omega_2 - m\ g\ r] \tag{4.23d}$$

$$x = r\theta_2 \tag{4.23e}$$

which can directly be coded in any of the simulation languages. Eq(4.23e) denotes an *output equation.* It is not needed in order to solve the set of differential equations, but is computed only for the purpose of display on output. Consequently, it does not make sense to code this equation inside the DERIVATIVE section of the program. To demonstrate this new concept, let me write down an excerpt of an ACSL [4.8] program implementing this model:

```
PROGRAM Rotational Mechanical System
  INITIAL
    constant
      "Place values for all constants here"
    J1inv  = 1.0/J1
    B13    = -(B1 + B3)
    J2apin = 1.0/(J2 + m * r * *2)
    kap    = k1 + k2 * r * *2
    Bap    = B1 + B2 * r * *2
    mgr    = m * g * r
  END $ "of INITIAL"
  DYNAMIC
    DERIVATIVE
      th1dot = om1
      om1dot= J1inv * (B13 * om1 + B1 * om2 + tau)
      th2dot = om2
      om2dot= J2apin * (B1 * om1 - kap * th2 - Bap * om2 - mgr)
      th1    = integ(th1dot, th10)
      om1    = integ(om1dot, om10)
      th2    = integ(th2dot, th20)
      om2    = integ(om2dot, om20)
    END $ "of DERIVATIVE"
    x = r * th2
    termt (t.ge.tmx)
  END $ "of DYNAMIC"
END $ "of PROGRAM"
```

This program demonstrates new concepts. Since the DERIVATIVE segment of the program is being executed over and over again, it is important to keep this segment *as short as possible* by throwing out all computations that are not necessary for the solution of the differential equations, in order to make the execution of the simulation program fast. In this context, all *constant expressions* should be moved into the INITIAL segment, and all *output equations* should be moved out of the DERIVATIVE block. Output equations will then be evaluated once per communication interval only, namely immediately before the output variables are stored in the data base.

DESIRE [4.7] and DARE–P [4.14] offer equivalent features. In DESIRE, *output equations* can be placed below the *OUT* statement, and in DARE–P, they can be coded in a separate $D2 block.

Let me now demonstrate the other approach. Fig.4.7 shows the same system after it has been decomposed into three subsystems, and after all internal and fictitious forces/torques have been introduced.

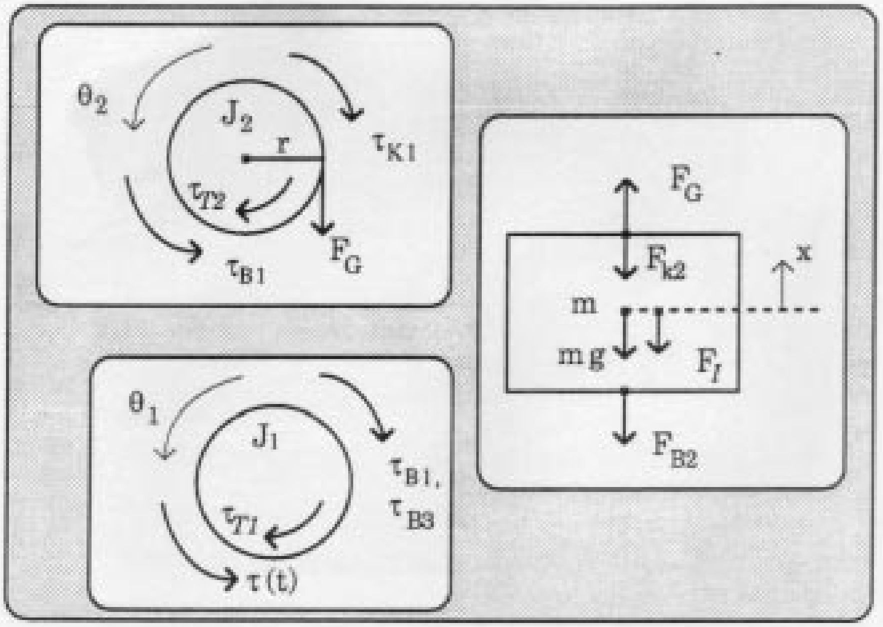


Figure 4.7. Rotational system after decomposition into parts

Previously, it was stated that subsystems should be selected such that each subsystem contains exactly one independently movable body. In this example, we did not adhere to this rule. The subsystems J_2 and m cannot be moved independently from each other. It will be demonstrated how this decision will affect our model.

We are now ready to write simulation equations directly. As before, the simulation equations comprise the equations resulting from the d'Alembert principle, as well as the equations describing the individual forces/torques (equivalent to the "branch equations" of electrical circuits).

$$\tau(t) = \tau_{T1} + \tau_{B1} + \tau_{B3} \tag{4.24a}$$

$$\tau_{B1} = \tau_{T2} + \tau_{k1} + \tau_{G} \tag{4.24b}$$

$$F_G = F_I + F_{k2} + F_{B2} + m\ g \tag{4.24c}$$

$$\tau_{T1} = J_1 \frac{d\omega_1}{dt} \tag{4.24d}$$

$$\frac{d\theta_1}{dt} = \omega_1 \tag{4.24e}$$

$$\tau_{T2} = J_2 \frac{d\omega_2}{dt} \tag{4.24f}$$

$$\frac{d\theta_2}{dt} = \omega_2 \tag{4.24g}$$

$$F_I = m\frac{dv}{dt} \tag{4.24h}$$

$$\frac{dx}{dt} = v \tag{4.24i}$$

$$\tau_G = r\ F_G \tag{4.24j}$$

$$x = r\theta_2 \tag{4.24k}$$

$$\tau_{B1} = B_1(\omega_1 - \omega_2) \tag{4.24l}$$

$$\tau_{B2} = B_2\omega_1 \tag{4.24m}$$

$$F_{B3} = B_3\ v \tag{4.24n}$$

$$\tau_{k1} = k_1\theta_2 \tag{4.24o}$$

$$F_{k2} = k_2\ x \tag{4.24p}$$

Among these equations, we find six differential equations. However, we know already that this is a fourth order system. The reason for this discrepancy is easily understood by looking at eq(4.24k). x and θ_2 are related to each other in a linear fashion, i.e., those two variables do not qualify for separate state variables. The problem was caused by the fact that we ignored the rule that systems which cannot be moved independently should not be split in two. Such a decision will always create *structural singularities.* Nothing is wrong with this approach though, we must only be prepared to do some extra work in the end in order to come up with an executable simulation model.

Rather than proceeding with this example, I prefer to demonstrate the learned concepts by means of a realistically complex problem, namely the analysis of a crane crab system.

4.4 The Crane Crab Example

Fig.4.8 shows a crane crab system that is used in a mechanical shop to move heavy loads from one place to another. A cart moves horizontally on a bridge. The cart is pulled with a non–elastic rope. The rope is moved by the motor $M1$. The load hangs on another rope the length of which can be controlled by motor $M2$. It is assumed that the masses of the ropes are negligible, that both ropes are ideally stiff (they don't exhibit either an elastic or a plastic deformation), and also the bridge is ideally stiff (no sag).

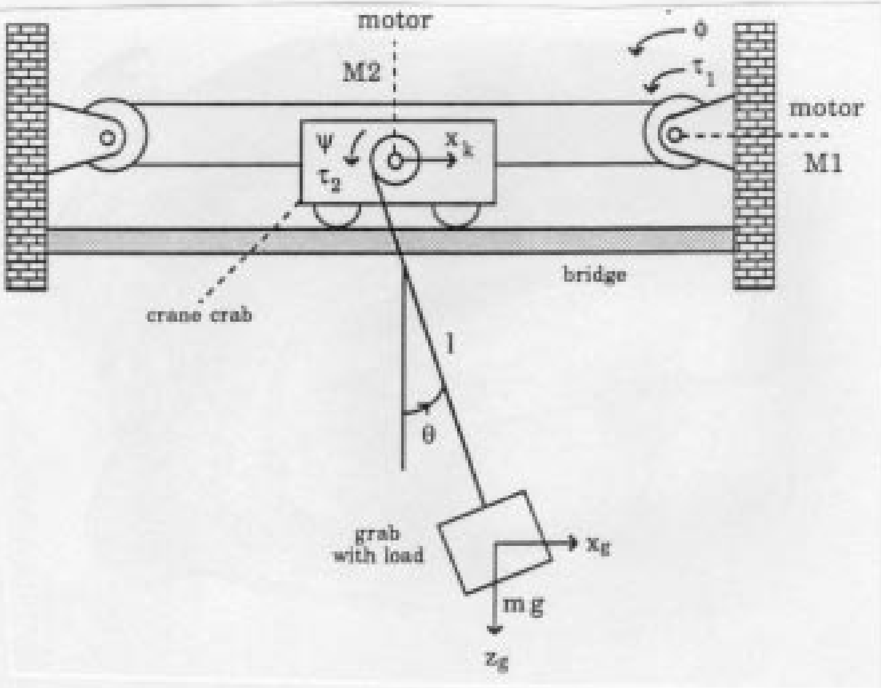


Figure 4.8. Crane crab system

Since the system is fairly complex, it was decided to use the decomposition technique. We decompose the system into four separate subsystems describing (a) the crane crab, (b) the motor $M1$, (c) the motor $M2$, and (d) the grab with the load. This is demonstrated in Fig.4.9.

The first subsystem (the crane crab) exhibits a translational movement in horizontal direction only (the vertical forces were also drawn, but they must add up to zero at all times). The second subsystem exhibits a rotational movement only, as does the third subsystem. The fourth subsystem exhibits translational movements in two directions. This seems to indicate that the system is of 10^{th} order. However, linear dependencies exist between the subsystems (we again cut the system into smaller portions than can be moved independently). The position x_k determines the angle ϕ completely. Also, the angle ψ influences both the x_g and the z_g coordinates of the load (for a constant value of ψ, the load has only one (circular) path along which it can move. Consequently, the system order will have to be reduced to six, and structural singularities will pop up between our initially chosen model variables.

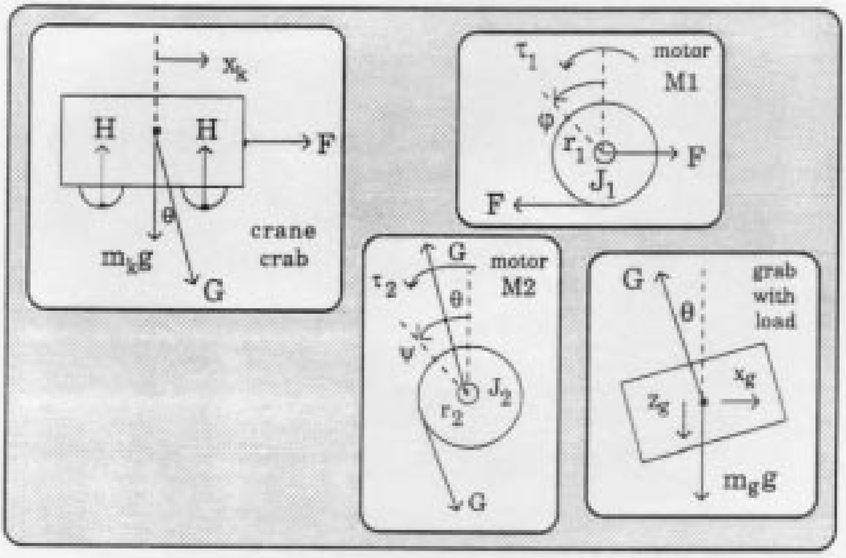


Figure 4.9. Crane crab system decomposed

Let us write an initial set of simulation equations now. For a change, we did not introduce any fictitious forces/torques, and shall operate on Newton's law directly.

$$m_k \, \ddot{x}_k = F + G \, sin\theta \tag{4.25a}$$
$$J_1 \, \ddot{\phi} = \tau_1(t) - r_1 \, F \tag{4.25b}$$
$$J_2 \, \ddot{\psi} = \tau_2(t) + r_2 \, G \tag{4.25c}$$
$$m_g \, \ddot{x}_g = -G \, sin\theta \tag{4.25d}$$
$$m_g \, \ddot{z}_g = m_g \, g - G \, cos\theta \tag{4.25e}$$
$$x_k = r_1 \phi \tag{4.25f}$$
$$\ell = r_2 \psi \tag{4.25g}$$
$$x_g = x_k + \ell \, sin\theta \tag{4.25h}$$
$$z_g = \ell \, cos\theta \tag{4.25i}$$

It was easy so far. We ended up with nine highly non-linear equations in the nine unknowns x_k, x_g, z_g, ϕ, ψ, θ, ℓ, F, and G. In order to come up with a set of simulation equations, we will need to analytically compute the second derivatives of eq(4.25f-i). This leads to:

$$\ddot{x}_k = r_1 \ddot{\phi} \tag{4.26a}$$
$$\ddot{\ell} = r_2 \ddot{\psi} \tag{4.26b}$$
$$\ddot{x}_g = \ddot{x}_k + \ell\ddot{\theta}cos\theta - \dot{\ell}\theta^2 sin\theta + 2\dot{\ell}\dot{\theta}cos\theta + \ddot{\ell}sin\theta \tag{4.26c}$$
$$\ddot{z}_g = -\ell\ddot{\theta}sin\theta - \dot{\ell}\theta^2 cos\theta - 2\dot{\ell}\dot{\theta}sin\theta + \ddot{\ell}cos\theta \tag{4.26d}$$

Now, let us eliminate the variables x_g and z_g by plugging eq(4.25d) into eq(4.26c), and by plugging eq(4.25e) into eq(4.26d).

$$-G\ sin\theta = m_g[\ddot{x}_k + \ell\ddot{\theta}cos\theta - \dot{\ell}\theta^2 sin\theta + 2\dot{\ell}\dot{\theta}cos\theta + \ddot{\ell}sin\theta] \tag{4.27a}$$
$$m_g\ g - G\ cos\theta = m_g[-\ell\ddot{\theta}sin\theta - \dot{\ell}\theta^2 cos\theta - 2\dot{\ell}\dot{\theta}sin\theta + \ddot{\ell}cos\theta] \tag{4.27b}$$

These equations can be simplified by the following operation:

$$eq(4.27a) \cdot cos\theta - eq(4.27b) \cdot sin\theta \Rightarrow eq(4.28a)$$
$$eq(4.27a) \cdot sin\theta + eq(4.27b) \cdot cos\theta \Rightarrow eq(4.28a)$$

This generates the equations:

$$-g\ sin\theta = \ddot{x}_k\ cos\theta + \ell\ddot{\theta} + 2\dot{\ell}\dot{\theta} \tag{4.28a}$$
$$m_g\ g\ cos\theta - G = m_g[\ddot{x}_k\ sin\theta - \dot{\ell}\theta^2 + \ddot{\ell}] \tag{4.28b}$$

We now have seven equations in seven unknowns, namely eq(4.25a-c), eq(4.26a-b), and eq(4.28a-b). This set of equations is solvable except for the fact that it contains algebraic loops. We must either continue to eliminate variables (we can eliminate F and G, for example) until the algebraic loops disappear, or we must place the entire set of equations into an *IMPL* construct as described in Chapter 2. In the given example, the latter approach may be more feasible.

4.5 Modeling Pulleys

Pulleys are frequently used elements to enable human operators to lift heavy loads. Fig.4.10 shows a four–pulley hoist which may serve as an example.

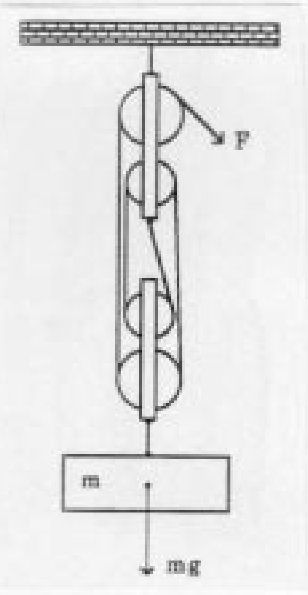


Figure 4.10. Four–pulley hoist

The question of interest is the following: Which force F is necessary in order to keep the system in an equilibrium state? The answer is trivial. If we cut the system in the middle, we realize that the internal forces (tensions) in the four ropes must add up to mg, otherwise, the lower two wheels would move either up or down. Furthermore, if we cut the system above the lowermost wheel, we see that the tensions in the two outermost ropes must be the same, otherwise, the lowermost wheel would rotate. We can thus conclude that the tensions in all four ropes must be equal, i.e. $\frac{mg}{4}$. Consequently, in order to prevent the uppermost wheel from rotating, we need to apply a force $F = \frac{mg}{4}$, i.e., we require only one fourth of the force to lift the heavy load m as compared to a direct lift.

4.6 The Inverse Pendulum Problem

Let us look at one more problem. A double pendulum is balanced on a cart. We would like to simulate what happens to the pendulum as a function of time if various types of control inputs F are applied to the system. The system is shown in Fig.4.11.

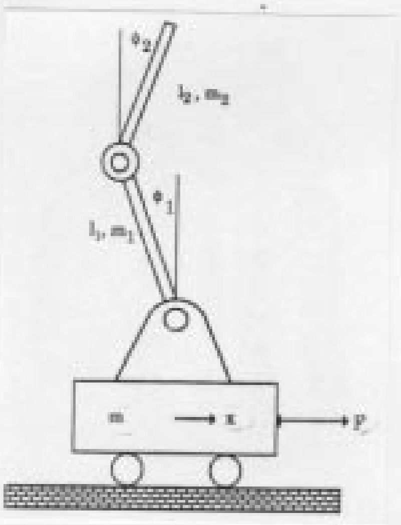


Figure 4.11. Double pendulum balanced on a cart

This system can be used to study a number of interesting control problems. The most interesting question is the following: Assume that we start out with both sticks in the upright position. Assume that a small disturbance moves the sticks away from the unstable equilibrium point. Can we balance the two sticks to return to the upright position simply by moving the cart back and forth? In other words: Can we find a control strategy which stabilizes the system around this steady–state point? Amazingly enough, the answer is yes. In fact, it has been proven that an infinitely large number of sticks can (at least theoretically) be balanced in this way. By making these infinitely many sticks infinitely short, we just reinvented the Indian magicians rope trick (no flute though).

In order to tackle the modeling task, we must first realize that we can replace the two sticks of lengths ℓ_1 and ℓ_2 and homogeneously distributed masses m_1 and m_2 by two other sticks with their masses concentrated in their centers of gravity. We can then decompose the system into three parts by introducing the internal forces at the cutting points. This is shown in Fig.4.12.

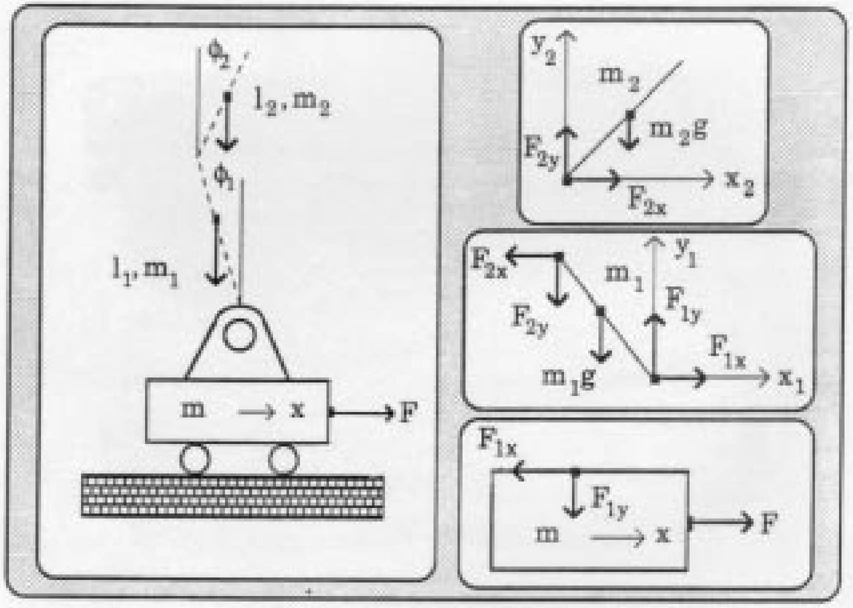


Figure 4.12. Double pendulum decomposed

Now, we are ready to write the differential equations for each of the three subsystems. Notice that the two angles ϕ_1 and ϕ_2 are counted positively clockwise from the vertical.

$$m\,\ddot{x} = F - F_{1x} \tag{4.29a}$$

$$m_1\,\ddot{x}_1 = F_{1x} - F_{2x} \tag{4.29b}$$

$$m_2\,\ddot{x}_2 = F_{2x} \tag{4.29c}$$

$$m_1\,\ddot{y}_1 = F_{1y} - F_{2y} - m_1\,g \tag{4.29d}$$

$$m_2\,\ddot{y}_2 = F_{2y} - m_2\,g \tag{4.29e}$$

$$J_1\ddot{\phi}_1 = -(F_{1x} + F_{2x})\frac{\ell_1}{2}cos\phi_1 + (F_{1y} + F_{2y})\frac{\ell_1}{2}sin\phi_1 \tag{4.29f}$$

$$J_2\ddot{\phi}_2 = -F_{2x}\frac{\ell_2}{2}cos\phi_2 + F_{2y}\frac{\ell_2}{2}sin\phi_2 \tag{4.29g}$$

$$x_1 = x + \frac{\ell_1}{2}sin\phi_1 \tag{4.29h}$$

$$y_1 = \frac{\ell_1}{2}cos\phi_1 \tag{4.29i}$$

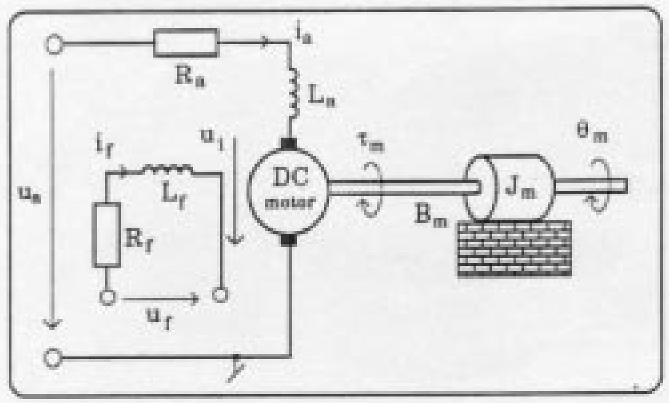
$$x_2 = x + \ell_1\,sin\phi_1 + \frac{\ell_2}{2}sin\phi_2 \tag{4.29j}$$

$$y_2 = \ell_1 cos\phi_1 + \frac{\ell_2}{2}cos\phi_2 \tag{4.29k}$$

The model consists of eleven highly non–linear equations in the eleven unknowns x, x_1, x_2, y_1, y_2, ϕ_1, ϕ_2, F_{1x}, F_{1y}, F_{2x}, and F_{2y}. As in the case of the crane crab system, we are plagued by structural singularities. Looking at the degrees of freedom of the system, we realize that we can move the cart in one direction, the first stick in one direction relative to the cart, and the second stick in one direction relative to the first stick, i.e., we have three independently movable bodies with one direction each, that is: the system must be of sixth order. However, looking into our equations above, we seem to have a 14^{th} order model here. This discrepancy can be explained (as before) by the four *linear constraints* eq(4.29h-k). In order to come up with a simulation model, we would again have to compute second derivatives for these four equations, detect the resulting algebraic loops, and eliminate variables until they disappear, or solve the equations in an *IMPL* block.

4.7 Modeling Electro–Mechanical Systems

We are now ready to model electro–mechanical devices. Electrical and mechanical systems can interact in several ways, the most prominent of which is through magnetic fields. This is how all electrical motors work. Let us look at one such motor in a little more detail, namely the DC–motor. Fig.4.13 shows an electro–mechanical diagram of this motor.



$$\tau_m = \psi . i_a$$
$$u_i = \psi . \dot{\theta}_m$$
$$\psi = k . i_f$$

Figure 4.13. Electro–mechanical diagram of a DC–motor

The motor has two separate coils, the *armature coil* which is mounted on the rotating part of the motor, and the *field coil* which is mounted on the stationary part of the motor. The current flowing through the field coil generates a magnetic field. If current flows through the armature coil as well, a force is generated in the armature coil which is responsible for the rotation of the cylinder which is anchored to the armature coil. The resulting torque τ_m is proportional to the applied field current, and also to the applied armature current:

$$\tau_m = k \cdot i_f \cdot i_a \tag{4.30}$$

Very often, the DC–motor is operated with a *constant field*, i.e., the angular velocity of the motor is controlled through a variation of the applied armature current. Such a configuration is called *armature control*. In that case, eq(4.30) can also be written as:

$$\tau_m = \psi \cdot i_a \tag{4.31}$$

where ψ is sometimes called the *torque constant*, and sometimes the *Back EMF constant*, since the same constant appears in a second equation:

$$u_i = \psi \cdot \omega_m \tag{4.32}$$

which describes the voltage induced in the armature coil under the influence of the rotation.

With these two equations, we can now model the DC–motor as a whole since it consists of the two electrical subsystems describing the field and the armature, the mechanical subsystem describing the inertia and friction of the rotating cylinder, plus the two coupling equations that connect the mechanical subsystem to the electrical subsystem. The equations should be self–explanatory by now.

$$u_f = R_f\ i_f + L_f \frac{di_f}{dt} \tag{4.33a}$$

$$u_a = R_a\ i_a + L_a \frac{di_a}{dt} + u_i \tag{4.33b}$$

$$J_m \frac{d^2\theta_m}{dt^2} = \tau_m - \tau_L - B_m \frac{d\theta_m}{dt} \tag{4.33c}$$

$$\tau_m = k\ i_f\ i_a \tag{4.33d}$$

$$u_i = k\ i_f\ \omega_m \tag{4.33e}$$

Fig.4.14 shows a block diagram of the DC–motor.

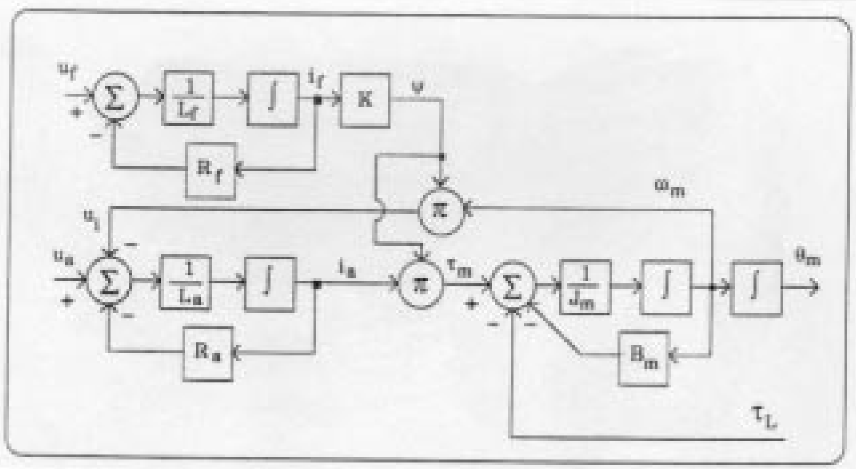


Figure 4.14. Block diagram of a DC–motor

The motor has three different inputs, namely the voltage applied to the field coil u_f, the voltage applied to the armature coil u_a, and the torque load τ_L which results from the machinery that is being driven by the motor. The block diagram also explains the popularity of armature control. If the field current is constant, the device degenerates to a *linear system* which is better amenable to an analytical treatment (although, for the purpose of simulation, we couldn't care less).

4.8 Summary

In this chapter, we have dealt with the problem of modeling planar mechanical systems in both translational and rotational coordinates. More can be said about mechanical systems than we were able to cover in this chapter. Unfortunately, we lack the necessary space in this text for an enhanced discussion. Notice that one of our major goals is to bridge the gap between the various application areas of differential equation models, and to come up with a consistent terminology and methodology to deal with such models. It is not the aim of this chapter to duplicate the tremendous effort that went into the design of textbooks in mechanics.

What are the topics that were left out of this chapter, and where are they discussed?

This chapter discussed planar systems only. A more general discussion of the subject matter should include three–dimensional mechanical problems. A free moving rigid body can translate along and rotate around the three spatial axes independently. Consequently, Newton's law (or the d'Alembert principle) must be formulated six times, once for each degree of freedom. Therefore, a free moving rigid body is described through a 12^{th} order state–space model. Examples of three–dimensional motions are presented in the two projects of this chapter.

A number of textbooks [4.1,4.10,4.11] deal with the modeling of mechanical systems, textbooks that are geared more towards the needs of mechanical engineering students. However, all these texts are junior level textbooks and not senior level textbooks. Consequently, while they cover the modeling of mechanical systems on a larger number of pages, they simply proceed at a somewhat slower pace, and do not really extend their coverage beyond our discussion. In particular, none of those textbooks covers general motions of mechanical systems in three space dimensions. Moreover, all these texts are geared towards *analytical modeling* rather than towards *simulation modeling*. Consequently, they stop with the derivation of the differential equations themselves, and don't bother to translate these differential equations into state–space models. Consequently, they don't discuss the concept of structural singularities and algebraic loops at all. However, since these texts are meant to be used by mechanical engineers, they provide nice chapters on *hydraulic system modeling* and on *pneumatic system modeling*, topics for which we lack the space in this textbook which has been written more with the electrical engineering and systems engineering students in mind.

A good selection of general Newtonian mechanics textbooks exists that indeed go far beyond our coverage of the topic [4.4,4.15,4.18]. All of these texts discuss three–dimensional motions in great detail. However, these textbooks deal with the physics of mechanical systems only, and are not really meant to be modeling textbooks.

Secondly, when modeling moving bodies (such as an aircraft), it is quite common to describe the motion of the body relative to a coordinate system that moves along with the body. The origin of the moving coordinate system is then often assumed to be the center of gravity of that body. Typical examples are bodies that move within an Earth–fixed coordinate system, but relative to the movement of planet Earth, or a robot's end–effector which moves relative to the position of its wrist. The total movement of a rigid body is thus decomposed into a movement relative to a moving coordinate frame, and an absolute movement of that coordinate frame itself. In this case, Newton's law (or the d'Alembert principle) must be slightly modified by including two additional fictitious forces, namely the *centripetal force* F_Z, and the *Coriolis force* F_C. Again, this topic is carefully discussed in all classical Newtonian mechanics textbooks. An example of a relative motion is presented in pr(P4.2).

Finally, a more modern approach to dealing with mechanical systems is through the use of the *Lagrange equation* which replaces and generalizes Newton's law. Newton's law, as it was discussed in this chapter, assumes that the equations of motion are described in *Cartesian coordinates*. This is not always practical. We can overcome this limitation by formulating the *total kinetic energy* of all bodies in the system in as many different velocities as the system contains independently moving bodies. For example, for the system of Fig.4.5, we find:

$$E_k = \frac{1}{2}J_1\dot{\theta}_1^2 + \frac{1}{2}J_2\dot{\theta}_2^2 + \frac{1}{2}m\dot{x}^2 = \frac{1}{2}J_1\dot{\theta}_1^2 + \frac{1}{2}(J_2 + mr^2)\dot{\theta}_2^2 \qquad (4.34)$$

We call the θ_i and x_i variables our *generalized displacements* q_i. In these variables, we can then reformulate Newton's law as follows:

$$\frac{d}{dt}(\frac{\partial E_k}{\partial \dot{q}_i}) - \frac{\partial E_k}{\partial q_i} = Q_i \qquad (4.35)$$

where Q_i stands for the sum of all *generalized forces* in the direction of the generalized displacements q_i. This formulation of Newton's law is more powerful than the previously used formulation since it is valid independently of the coordinate frame that is being used. Furthermore, with this approach, it is no longer necessary to cut the system into individually moving pieces by introducing coupling forces which we must later eliminate again. Therefore, this approach is often more economical than the direct application of Newton's law.

A special case is the set of *conservative systems*, i.e., systems without energy dissipation (i.e., the frictionless systems). For those systems, yet another formulation of the equations of motion can be found. This time, we consider also the potential energy E_p of the system. Thereafter, we build the so–called *Lagrangian* of the system which is the difference between the total kinetic energy and the total potential energy

$$L(q_1, \ldots, q_n, \dot{q}_1, \ldots, \dot{q}_n) = E_k - E_p \tag{4.36}$$

Using the Lagrangian, we can reformulate Newton's law as follows:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_i}) - \frac{\partial L}{\partial q_i} = 0 \tag{4.37}$$

Another formulation of eq(4.37) is through the use of the so-called *Hamiltonian* of the system which is equivalent to the total *free energy* of the system

$$H(q_1, \ldots, q_n, \dot{q}_1, \ldots, \dot{q}_n) = E_k + E_p \tag{4.38}$$

We can now replace the derivatives of the generalized displacements by *generalized momentums*. In translational coordinates, the generalized momentum of a displacement x_i is its momentum $p_i = \mathcal{I}_i = m_i \dot{x}_i$, whereas in rotational coordinates, the generalized momentum of an angle θ_i (a generalized displacement) is its twist $p_i = \mathcal{T}_i = J_i \dot{\theta}_i$. We can thus reformulate the Hamiltonian in terms of the new variables q_i and p_i. Using this version of the Hamiltonian, we can reformulate Newtons law as follows:

$$\dot{p}_i = -\frac{\partial H}{\partial q_i}, \quad \dot{q}_i = \frac{\partial H}{\partial p_i} \tag{4.39}$$

We shall resume this discussion in Chapters 7 and 8 of this text.

In many mechanical systems, the *kinematic constraints* are the dominating factors that determine the motion of the system. The system is so rigid and moves so slowly that the dynamics of the system are no longer considered important, and are therefore simply being ignored. Inputs to these systems are no longer forces and torques, but rather positions and angles. The goal of the investigation is to determine the positions and angles of all parts of the system in response to the applied inputs. The responses are considered *instantaneous*. Time appears only through the input functions themselves which are often assumed to be functions of time. Consequently, these models are not differential equation models, and are therefore outside the realm of this textbook.

Very little was said in this chapter about *software* for dealing with mechanical problems. Indeed, very little has been accomplished in this respect. A nicely written senior level textbook exists that describes in detail both the kinematics and the kinetics of two–dimensional (planar) and three–dimensional mechanical systems

[4.9]. However, while this textbook describes clearly how such systems are being modeled, and while it is also computer (simulation) oriented, it is somewhat disappointing with respect to the maturity of software concepts and user interfaces of the programs introduced to deal with the simulation of these systems. It contains no more than a bunch of Fortran coded subroutines that the student is encouraged to adapt to his or her needs. Unfortunately, this seems to be the state-of-the-art in mechanical system simulation.

Three hot research topics in mechanical system modeling can be named. One topic is related to *full digital flight simulators*. Such simulators are used for three separate purposes: (1) as parts of autopilots, (2) for pilot training, and (3) for system trouble–shooting. The equations of flight motion are straightforward, and a number of good textbooks exist that deal with those in great detail [4.3,4.5]. The major problem with flight simulators relates to their execution speed. Flight simulators must execute in *real time* which either calls for very fast computers (which are still fairly expensive), or special parallel processor architectures for which no good distributed real–time operating systems exist yet. Consequently, flight simulators still rely heavily on assembly programming, and special tricks to reduce the execution time required.

A second hot topic is *robot modeling* for the purpose of robot control. Again, good textbooks can be found that deal with robot modeling in great detail [4.12,4.13,4.16]. The major problem here is with the kinematic constraints. Robots are highly non–linear systems, and the algebraic loops created by the kinematic constraints can often not be solved analytically. In many cases, the robot dynamics are not considered at all, and the movement of the robot is dictated entirely by its kinematic constraints. Newer papers, in particular those dealing with dexterous hand movements and other sorts of fine motion planning include the dynamic equations of motion. The problems here are mostly concerned with the highly non–linear and non–measurable friction and backlash coefficients. Clearly, more research must be devoted to the development of user–friendly general purpose robotics software. A good amount of work went recently into the development of three–dimensional graphics engines which allow to display the three–dimensional motion of robots on a computer screen.

A third hot topic is related to the modeling of mechanical limbs for the purpose of *prosthesis design*. To my knowledge, no textbook exists yet that deals with this topic, but a series of very good research papers have been devoted to this topic recently.

References

[4.1] Charles M. Close, and Dean K. Frederick (1978), *Modeling and Analysis of Dynamic Systems*, Houghton Mifflin Company, Boston, MA.

[4.2] J. Denavit, and R. S. Hartenberg (1955), "A Kinematic Notation for Lower–Pair Mechanisms Based on Matrices", *ASME J. of Applied Mechanics*, **22**(2), pp. 215–221.

[4.3] Bernard Etkin (1982), *Dynamics of Flight*, John Wiley & Sons, Inc., New York.

[4.4] Herbert Goldstein (1980), *Classical Mechanics*, Second Edition, Addison–Wesley, Reading, MA.

[4.5] R. K. Heffley *et alia* (1972), *Aircraft Handling Qualities Data*, Report NASA–CR–2144, Systems Technology, Inc.

[4.6] Gene H. Hostetter, Clement J. Savant, Jr., and Raymond T. Stefani (1989), *Design of Feedback Control Systems*, Second Edition, Saunders College Publishing, New York.

[4.7] Granino A. Korn (1989), *Interactive Dynamic–System Simulation*, McGraw–Hill, New York.

[4.8] Edward E.L. Mitchell, and Joe S. Gauthier (1986), *ACSL: Advanced Continuous Simulation Language — User Guide / Reference Manual*, Mitchell & Gauthier Assoc., 73 Junction Square, Concord, MA 01742.

[4.9] Parviz E. Nikravesh (1988), *Computer–Aided Analysis of Mechanical Systems*, Prentice–Hall, Englewood Cliffs, N.J.

[4.10] Katsuhiko Ogata (1978), *System Dynamics*, Prentice–Hall, Englewood Cliffs, N.J.

[4.11] William J. Palm, III (1983), *Modeling, Analysis, and Control of Dynamic Systems*, John Wiley, New York.

[4.12] Richard P. Paul (1981), *Robot Manipulators: Mathematics, Programming, and Control — The Computer Control of Robot Manipulators*, MIT Press, Boston, MA.

[4.13] Mark W. Spong, and M. Vidyasagar (1989), *Robot Dynamics and Control*, John Wiley, New York.

[4.14] John V. Wait, and DeFrance Clarke, III (1976), *DARE–P User's Manual*, Version 4.1, Dept. of Electrical & Computer Engineering, University of Arizona, Tucson, AZ, 85721.

[4.15] Jens Wittenburg (1977), *Dynamics of Systems of Rigid Bodies*, Teubner Verlag, Stuttgart, FRG.

[4.16] William A. Wolovich (1987), *Robotics: Basic Analysis and Design*, Holt, Rinehart and Winston, Inc., New York.

[4.17] Bernard P. Zeigler, François E. Cellier, and Jerzy W. Rozenblit (1988), "Design of a Simulation Environment for Laboratory Management by Robot Organizations", *J. of Intelligent and Robotic Systems*, **1**, pp. 299–309.

[4.18] Hans Ziegler (1965), *Mechanics*, Addison–Wesley, Reading, MA.

Homework Problems

[H4.1] Lagrangian

In the system of Fig.4.5, replace all frictions (dissipative terms) temporarily by springs. Call the spring constants β_1, β_2, and β_3. In this way, we have transformed our previously dissipative system into another system which is conservative. Its potential energy can be computed as:

$$E_p = \frac{1}{2}\beta_3\theta_1^2 + \frac{1}{2}\beta_1(\theta_1 - \theta_2)^2 + \frac{1}{2}k_1\theta_2^2 + \frac{1}{2}k_2x^2 + \frac{1}{2}\beta_2x^2 + mgx - \tau\theta_1 \quad (H4.1)$$

Find the Lagrangian of this system, and plug it into eq(4.37). Derive a set of differential equations describing this conservative system. In the very end, replace each term of type $\beta_i q_j$ by the original friction term $B_i \dot{q}_j$, and show that you end up with the same set of differential equations as with our original approach. This cheap engineering trick allows us to use the Lagrangian and/or the Hamiltonian to analyze also dissipative systems.

[H4.2] Hamiltonian

Find a Hamiltonian for the modified system of hw(H4.1), replace the derivatives of the generalized displacements $\dot{q}_i$ by the generalized momentums p_i, and write down the modified expression for the Hamiltonian in the q_i and p_i variables. Then apply eq(4.39), and show that you end up with the same set of differential equations as in hw(H4.1) prior to replacing the dissipative terms back into the system equations.

[H4.3] Cervical Syndrom

Some people (such as my sister) suffer from a so–called cervical syndrom. Their neck is not sufficiently stiff to connect their head solidly with their upper torso. Therefore, if their upper torso is exposed to vibrations, such as when riding in a car, these people often react with severe headaches.

A car manufacturer wants to design a new car in which these problems are minimized. Resonance phenomena are to be studied with the purpose of avoiding resonance frequencies of the human body to appear as eigenfrequencies of the car.

Fig.H4.3 shows a mechanical model of a sitting human body [4.6]. The legs are left out since they do not contribute to potential oscillations of the upper body. The data are average data for a human adult.

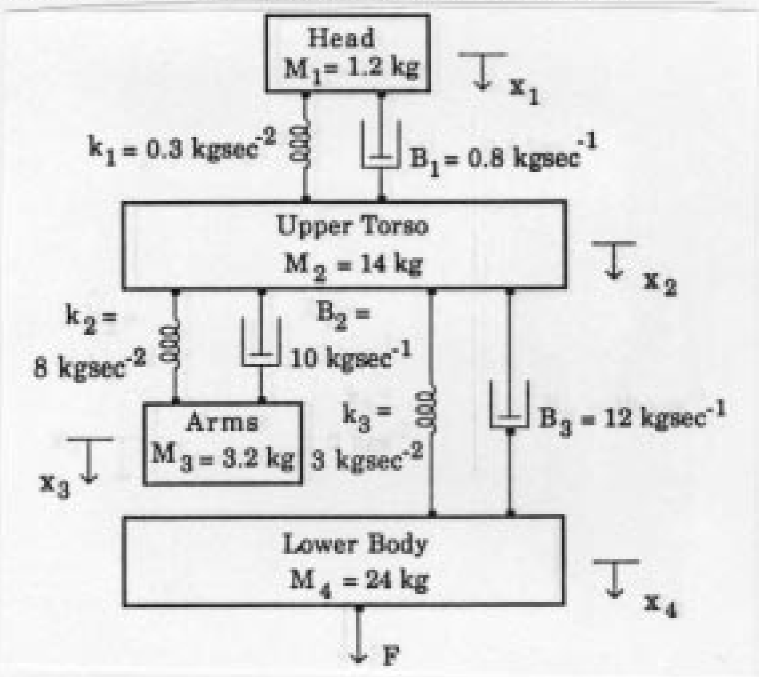


Figure H4.3. Mechanical model of a sitting human body

Derive a state–space model for this system. Since this is a linear time–invariant system, put it in linear state–space form, and simulate the system directly in CTRL–C (or MATLAB) using an AC force input of 1.5 *Hz*. The output of interest is the distance between the head and the upper torso.

In order to analyze the resonance phenomena, we wish to obtain a *Bode diagram* of this system. Create a frequency base logarithmically spaced between 0.01 *Hz* and 100 *Hz* using CTRL–C's (MATLAB's) *Logspace* function. Then compute a Bode diagram using the *Bode* function. Convert the amplitude into Decibels, and plot, on two graphs, the magnitude and the phase in a semi–logarithmic scale using CTRL–C's (MATLAB's) *Plot* function. Both CTRL–C and MATLAB offer interactive *Help* on all these functions (in CTRL–C, you need also the *Window*, *Title*, *Xlabel*, and *Ylabel* functions; MATLAB offers similar facilities). Determine all resonance frequencies together with the maximum overshoot at these frequencies.

Finally, we wish to perform a *sensitivity analysis*. We want to study the variability of the spring constant and the damper between the head and the upper torso. For this purpose, we assume a variability of k_1 and B_1 of ±50%. Repeat the frequency analysis from above for the four worst case combinations, and determine the range of resonance frequencies to be avoided. Determine also the maximum overshoot to be expected in the worst case. I suggest that you create a CTRL–C (MATLAB) function which computes the **A**–matrix and the b–vector as a function of the two model parameters k_1 and B_1. This will allow you to generate the four models more easily. While you execute these simulations, keep a diary of what you are doing (using CTRL–C's or MATLAB's *Diary* function).

[H4.4] Electro–Mechanical System

The electro-mechanical system shown in Fig.H4.4 can represent either a microphone, a loudspeaker, or a vibrating table.

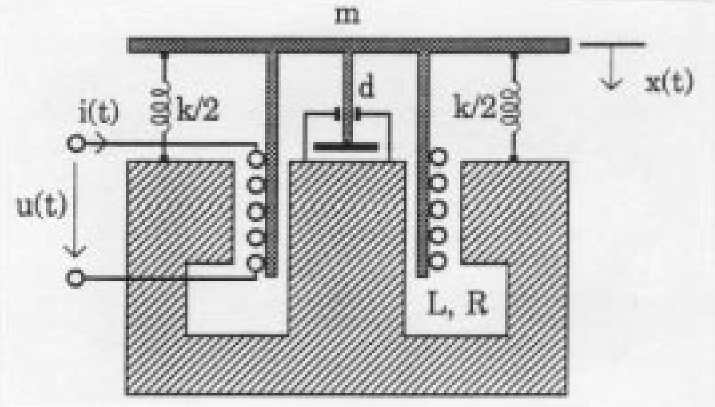


Figure H4.4. Electro–mechanical system

A moving induction coil is placed in the circular gap of a permanent magnet with the magnetic induction B. It can oscillate in axial direction. The coil has the inductance L and the resistance R. It consists of w windings with a radius r. The length of the coil is ℓ. Coupled to the coil is a mechanical system with a mass m, a spring constant k, and a damping factor d. The coupling between the electrical and the mechanical system can be described by the following two equations:

$$F = B \cdot i \cdot \ell \qquad (H4.4a)$$

$$u_i = B \cdot \dot{x} \cdot \ell \qquad (H4.4b)$$

where F denotes the force exerted on the mechanical system as a function of the current i that flows through the coil, and u_i denotes the voltage induced in the coil as a result of the mechanical movement.

Determine a state–space model for this electro–mechanical system, and draw a block diagram with $u(t)$ as input, and $x(t)$ as output.

[H4.5] Translational System

The translational mechanical system shown in Fig.H4.5 is to be modeled.

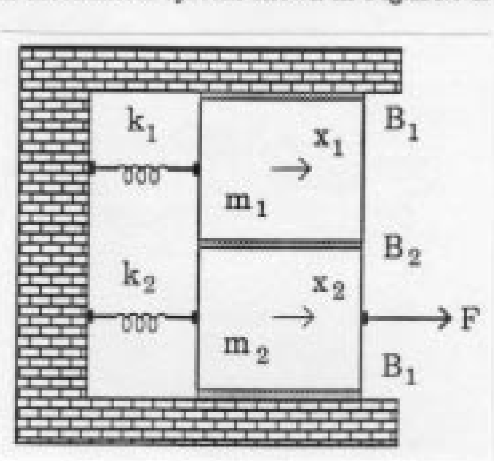


Figure H4.5. Translational mechanical system

Find a linear state–space model for this system with $F(t)$ as input, and $x_1(t)$ as output.

[H4.6]* Mixed Translational and Rotational System

The mixed translational and rotational system shown in Fig.H4.6 is to be modeled and simulated. This is a simplified version of a slipping clutch. A force F pulls (or pushes) a mass $m = 3.2\ kg$ along a floor. *Viscous friction* exists between the mass m and the floor. The friction force is:

$$F_{Fr} = B_1 \dot{x} \tag{H4.6a}$$

where $B_1 = 0.8\ kg\ sec^{-1}$. The mass m is also attached to the rear wall with a spring. The spring constant is $k_1 = 5\ kg\ sec^{-2}$. A cylinder with the inertia $J = 0.001\ kg\ m^2$ and the radius $r = 0.05\ m$ sits on top of the mass. It can either *roll* on the mass or *slip* over the mass. *Coulomb friction* exists between the mass m and the cylinder. The friction force is:

$$F_{Coul} \leq B_2\ sign(\dot{x}) \tag{H4.6b}$$

where $B_2 = 0.4$ *kg m sec*$^{-2}$. As long as the internal force at the contact point between the mass m and the cylinder is smaller than the maximum Coulomb friction, the cylinder will roll, and behaves exactly like a gear. However, as soon as the internal force becomes larger than the maximum Coulomb friction, the cylinder will start to slip. Thereafter, the cylinder will continue to slip until the two velocities at the contact point have equalized again. At this point in time, the cylinder will return to its rolling mode. The cylinder is attached to the two side walls with two rotational springs with the spring constants $k_2 = k_3 = 0.001$ *kg m*^{2} *sec*$^{-2}$.

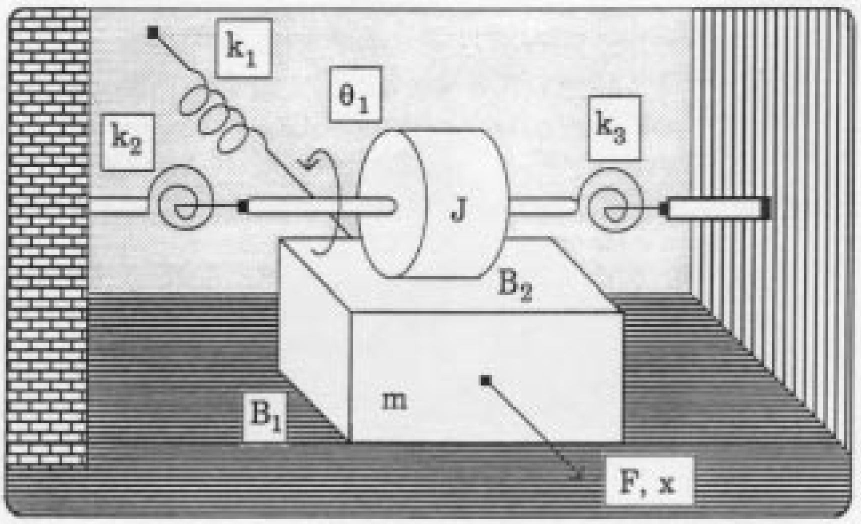


Figure H4.6. Mixed translational and rotational system

Model this non–linear system in ACSL using $F(t)$ as input, and both $x(t)$ and $\theta_1(t)$ as outputs. Apply a force:

$$F(t) = 0.1 \cdot t \cdot \sin(t) \qquad (H4.6c)$$

and simulate this system during 75 *sec*.

This simulation is not so simple. We need to toggle between two different models. The overall model is of second order whenever the cylinder rolls, but it is of fourth order when the cylinder slips. This is therefore a so–called *variable structure model.* In order to model the switching between the two modes correctly, we require two *state–events.* ACSL's state–event scheduler is called from within the DERIVATIVE segment. The two statements:

```
schedule goslip .xn. slpcon
schedule goroll .xn. rolcon
```

can be placed within the DERIVATIVE section. *Slpcon* is a real expression which triggers the execution of a DISCRETE section by the name of *goslip* to be executed whenever *slpcon* changes its sign from positive to negative. *Rolcon* works accordingly.

Projects

[P4.1] Aircraft Modeling

Flight stability can be studied through two independent models: longitudinal and lateral. Longitudinal motions can be modeled independently from the lateral ones if the following simplifying assumptions are valid:

(1) The airplane is perfectly symmetrical with respect to its median longitudinal plane.

(2) No gyroscopic effects of spinning masses (engine rotors, airscrews, etc) act on the aircraft.

In this project, we want to adopt the above assumptions, and consider the longitudinal model of a B747 aircraft in cruise flight at high altitude.

A longitudinal flight is characterized by the absence of forces and moments that would cause its lateral motion. (Notice the terminological confusion: the term "moment" is here used as a synonym for "torque", and not in the sense in which we have introduced that term before. However, I decided to stick to the conventional terminology since this is the one that you will commonly find when you scan through aerodynamics literature.) Furthermore, the aeroelastic nature of the airplane's structure is neglected as well, so that the *rigid body* equations of motion apply to the model.

The mathematical model described in this project reflects an essentially longitudinal flight restricted to longitudinal deviations from a trimmed reference flight condition. This reference flight is characterized by the requirement that the resultant force and torque acting on the aircraft's center of mass are zero.

We define a reference flight condition as being characterized by a steady longitudinal and horizontal flight where the resultant force and moment acting on the plane are zero. The headwind is assumed to be constant and horizontal.

The theory presented in this project is developed with respect to a set of body–fixed axes named the *stability axes*. The origin of this coordinate system is the center of gravity of the airplane: the x–axis points in the direction of the motion of the airplane in the reference flight condition, the z–axis points "downward", and the y–axis runs spanwise and points to the right.