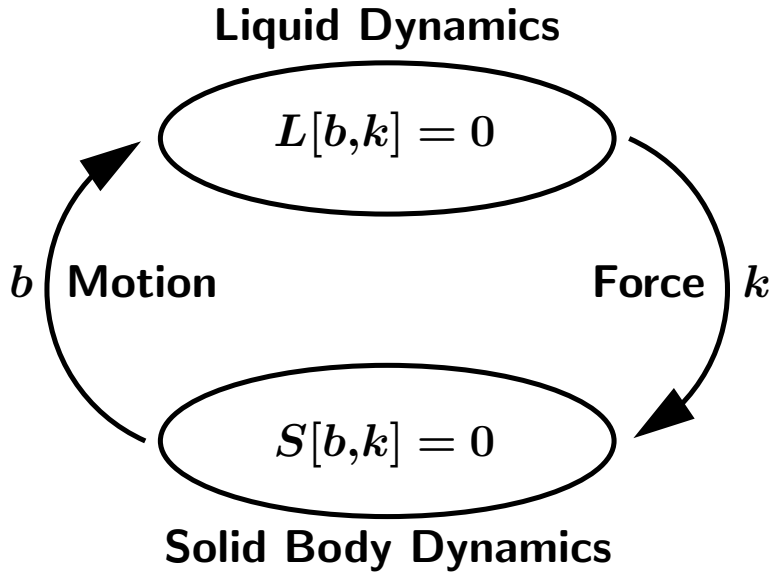
Modeling and Simulation of Vehicles Carrying Liquid Cargo

Gerk Rozema



Department of Mathematics

RuG

Master's Thesis

Modeling and Simulation of Vehicles Carrying Liquid Cargo

Gerk Rozema

Supervisor:
Prof.dr. A.E.P. Veldman
Department of Mathematics
University of Groningen
P.O. Box 800
9700 AV Groningen

July 2004

Contents

Chapter 1

Introduction

The ride behavior and stability of vehicles carrying liquid cargo may be affected by the liquid motion. In a partially filled tank the liquid is allowed to move from side to side, affecting for example cornering and rollover behavior. Also, liquid motion may become exaggerated due to driver inputs or excitations generated by the road surface, which in turn can have substantial effects on the motion of the vehicle.

The main objective of this thesis is to investigate the dynamical interaction between the liquid dynamics and the vehicle dynamics using simulation studies. Of most interest are simulations of realistic maneuvers. Simulation results of a cornering maneuver (rollover performance) and a bumpy road (exaggerated liquid motion) are presented in this thesis, enabling us to study the effects of the sloshing liquid on the motion of the vehicle.

In order to be able to perform these simulations, a simple vehicle model is developed in Matlab/Simulink, based on the conservation of linear and angular momentum of a solid body. For simulating the liquid dynamics the computer program ComFlo is used. ComFlo is capable of simulating sloshing liquids. The model for the fluid dynamics used in ComFlo is based on an equation for conservation of mass (continuity equation) and an equation for conservation of momentum (Navier-Stokes equation). These equations are solved for the fluid pressure and velocity field. To account for the vehicle motion, the liquid velocity is considered with respect to a moving reference frame (fixed with respect to the vehicle).

The vehicle is modeled in Matlab/Simulink, whereas ComFlo is a Fortran computer program. To execute this Fortran code from Simulink, the vehicle model is expanded with a block containing an adapted version of the ComFlo program. The force and torque that the fluid exerts on the tank are computed in ComFlo and used as input for the vehicle model. The vehicle motion in turn is used as input for ComFlo.

The actual vehicle model (based on dynamical interaction) is developed in chapter 3, followed by simulation results in chapter 4. Before that, the necessary rigid body dynamics are explained in chapter 2. As an example, the free rotation of a rectangular plate is modeled and simulated in Matlab/Simulink.

At the National Aerospace Laboratory NLR, Collaborative Engineering Systems department, the work on coupled solid-liquid vehicle dynamics is continued by Marc van den Raadt. The discussions with Marc van den Raadt at NLR are acknowledged.

Chapter 2

Rigid Body Dynamics

2.1 Equation for Linear Momentum

To describe the motion of a rigid body, we use two coordinate systems: An inertial frame $\boldsymbol{x}'$ and a coordinate system $\boldsymbol{x}$ fixed with respect to the body (figure 2.1). The center of mass of a rigid body moves as if it were a single particle, of mass equal to the total mass of the body, acted on by the total external force [4]. Newton's equation for the motion of a particle of mass m reads

$$\boldsymbol{F} = m\boldsymbol{a}_{\mathrm{f}} \tag{2.1}$$

where $\boldsymbol{a}_{\mathrm{f}}$ is the acceleration vector of the particle with respect to the inertial (fixed) reference frame (Newton's equation is valid only in an inertial frame of reference) and $\boldsymbol{F}$ is the force acting on the particle. The velocity relative to the fixed coordinate system of a particle fixed in the body system (figure 2.1) is given by

$$\boldsymbol{v}_{\mathrm{f}} = \boldsymbol{q} + \boldsymbol{\omega} \times \boldsymbol{r} \tag{2.2}$$

where $\boldsymbol{q}$ is the linear velocity of the origin of the body system, $\boldsymbol{\omega}$ is the angular velocity of the body system and $\boldsymbol{r}$ is the constant radius vector of the particle in the body system. The radius vector of the particle in the fixed system is denoted by $\boldsymbol{r}'$. Differentiating equation (2.2), we have

$$\boldsymbol{a}_{\mathrm{f}} = \left(\frac{d\boldsymbol{v}_{\mathrm{f}}}{dt}\right)_{\mathrm{f}} = \left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{f}} + \left(\frac{d\boldsymbol{\omega}}{dt}\right)_{\mathrm{f}} \times \boldsymbol{r} + \boldsymbol{\omega} \times \left(\frac{d\boldsymbol{r}}{dt}\right)_{\mathrm{f}} \tag{2.3}$$

The subscript f (fixed) is included to indicate that the time rate of change is measured in the fixed coordinate system. The time rates of change of $\boldsymbol{q}$,

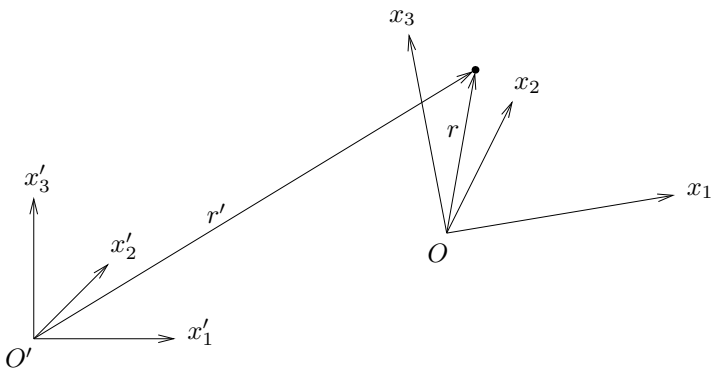


Figure 2.1: *Fixed coordinate system $\boldsymbol{x}'$ and body coordinate system $\boldsymbol{x}$. The particle is fixed in the body coordinate system.*

$\boldsymbol{r}$ and $\boldsymbol{\omega}$ as measured in the fixed system are given by

$$\left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{f}} = \left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{b}} + \boldsymbol{\omega} \times \boldsymbol{q} \tag{2.4}$$

$$\left(\frac{d\boldsymbol{r}}{dt}\right)_{\mathrm{f}} = \left(\frac{d\boldsymbol{r}}{dt}\right)_{\mathrm{b}} + \boldsymbol{\omega} \times \boldsymbol{r} = \boldsymbol{\omega} \times \boldsymbol{r} \tag{2.5}$$

$$\left(\frac{d\boldsymbol{\omega}}{dt}\right)_{\mathrm{f}} = \left(\frac{d\boldsymbol{\omega}}{dt}\right)_{\mathrm{b}} + \boldsymbol{\omega} \times \boldsymbol{\omega} = \left(\frac{d\boldsymbol{\omega}}{dt}\right)_{\mathrm{b}} = \dot{\boldsymbol{\omega}} \tag{2.6}$$

In these equations the subscript b (body) indicates that the time rate of change is measured in the body system. Substituting equations (2.4), (2.5) and (2.6) into equation (2.3), we obtain

$$\boldsymbol{a}_{\mathrm{f}} = \left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{b}} + \boldsymbol{\omega} \times \boldsymbol{q} + \dot{\boldsymbol{\omega}} \times \boldsymbol{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}) \tag{2.7}$$

Combining equations (2.1) and (2.7), the final equation for a particle of mass m fixed in the body coordinate system becomes

$$\boldsymbol{F} = m\left(\left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{b}} + \boldsymbol{\omega} \times \boldsymbol{q} + \dot{\boldsymbol{\omega}} \times \boldsymbol{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r})\right) \tag{2.8}$$

2.2 Equation for Angular Momentum

To obtain the equation for angular momentum, it is advisable to view the rigid body as a system of particles $\{i\}$ of masses m_i. The radius vectors $\boldsymbol{r}_i$ and $\boldsymbol{r}'_i$ are measured from O and O' respectively (figure 2.1). According to Newton's equation, $m_i\ddot{\boldsymbol{r}}'_i$ is equal to the force applied to the mass m_i, and its cross product with $\boldsymbol{r}_i$ is the moment about O:

$$\boldsymbol{N} = \sum \boldsymbol{N}_i = \sum \boldsymbol{r}_i \times m_i\ddot{\boldsymbol{r}}'_i \tag{2.9}$$

Using equations (2.7) and (2.4), the acceleration of particle i can be written as

$$\ddot{\boldsymbol{r}}'_i = \left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{f}} + \dot{\boldsymbol{\omega}} \times \boldsymbol{r}_i + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_i) \tag{2.10}$$

Substituting equation (2.10) into equation (2.9), we obtain

$$\boldsymbol{N} = \sum [m_i\boldsymbol{r}_i \times \left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{f}} + \boldsymbol{r}_i \times (\dot{\boldsymbol{\omega}} \times m_i\boldsymbol{r}_i) + \boldsymbol{r}_i \times (\boldsymbol{\omega} \times m_i(\boldsymbol{\omega} \times \boldsymbol{r}_i))] \tag{2.11}$$

The first term can be written as:

$$\sum \left(m_i\boldsymbol{r}_i \times \left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{f}}\right) = \left(\sum m_i\boldsymbol{r}_i\right) \times \left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{f}} = m_s\boldsymbol{r}_s \times \left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{f}} \tag{2.12}$$

where m_s is the total mass of the system and $\boldsymbol{r}_s$ is the vector defining the position of the system's center of mass in the body system. The second and third term of (2.11) can be written as [6]

$$\sum \boldsymbol{r}_i \times (\dot{\boldsymbol{\omega}} \times m_i\boldsymbol{r}_i) = \boldsymbol{I}_s\dot{\boldsymbol{\omega}} \tag{2.13}$$

$$\sum \boldsymbol{r}_i \times (\boldsymbol{\omega} \times m_i(\boldsymbol{\omega} \times \boldsymbol{r}_i)) = \boldsymbol{\omega} \times \boldsymbol{I}_s\boldsymbol{\omega} \tag{2.14}$$

where $\boldsymbol{I}_s$ is the inertia tensor of the body in the body coordinate system. Substituting equations (2.12), (2.13) and (2.14) into (2.11), the final equation becomes

$$\boldsymbol{N} = m_s\boldsymbol{r}_s \times \left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{f}} + \boldsymbol{I}_s\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}_s\boldsymbol{\omega} \tag{2.15}$$

2.3 Orientation of Body Coordinate System

To determine the orientation of the body coordinate system, we now introduce a series of rotations, which transforms the $\boldsymbol{x}'$ system into the $\boldsymbol{x}$ system (except for a translation). The rotation matrix $\boldsymbol{\lambda}$ describes the relative orientation of the two systems:

$$\boldsymbol{x} = \boldsymbol{\lambda}\boldsymbol{x}' \tag{2.16}$$

The first rotation is through an angle ϕ about the x_3'-axis (figure 2.1) to transform the fixed coordinate system $\boldsymbol{x}'$ into $\boldsymbol{x}''$. Positive angles are given by the right hand rule. The transformation matrix is

$$\boldsymbol{\lambda}_\phi = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.17}$$

$$\boldsymbol{x}'' = \boldsymbol{\lambda}_\phi \boldsymbol{x}' \tag{2.18}$$

The second rotation is through an angle θ about the x_2''-axis to transform $\boldsymbol{x}''$ into $\boldsymbol{x}'''$. The transformation matrix is

$$\boldsymbol{\lambda}_\theta = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \tag{2.19}$$

$$\boldsymbol{x}''' = \boldsymbol{\lambda}_\theta \boldsymbol{x}'' \tag{2.20}$$

The third and final rotation is through an angle ψ about the x_1'''-axis to transform the $\boldsymbol{x}'''$ system into the body coordinate system $\boldsymbol{x}$. The transformation matrix is

$$\boldsymbol{\lambda}_\psi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{pmatrix} \tag{2.21}$$

$$\boldsymbol{x} = \boldsymbol{\lambda}_\psi \boldsymbol{x}''' \tag{2.22}$$

The complete transformation from the $\boldsymbol{x}'$ system to the $\boldsymbol{x}$ system is given by

$$\boldsymbol{x} = \boldsymbol{\lambda}_\psi \boldsymbol{x}''' = \boldsymbol{\lambda}_\psi \boldsymbol{\lambda}_\theta \boldsymbol{x}'' = \boldsymbol{\lambda}_\psi \boldsymbol{\lambda}_\theta \boldsymbol{\lambda}_\phi \boldsymbol{x}' \tag{2.23}$$

and the rotation matrix $\boldsymbol{\lambda}$ is

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_\psi \boldsymbol{\lambda}_\theta \boldsymbol{\lambda}_\phi \tag{2.24}$$

The components of this matrix are

$$
\begin{aligned}
\lambda_{11} &= \cos\phi\cos\theta \\
\lambda_{21} &= \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\
\lambda_{31} &= \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\
\\
\lambda_{12} &= \sin\phi\cos\theta \\
\lambda_{22} &= \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi \\
\lambda_{32} &= \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi \\
\\
\lambda_{13} &= -\sin\theta \\
\lambda_{23} &= \cos\theta\sin\psi \\
\lambda_{33} &= \cos\theta\cos\psi
\end{aligned}
\tag{2.25}
$$

The angular velocities $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$ are directed along $x_3' = x_3''$, $x_2'' = x_2'''$ and $x_1''' = x_1$ respectively. The relationship between these angular velocities and the angular velocity vector $\boldsymbol{\omega}$ can be determined by collecting the components of $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$ along the body coordinate axes:

$$
\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = \begin{pmatrix} \dot{\psi} \\ 0 \\ 0 \end{pmatrix} + \boldsymbol{\lambda}_\psi \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \boldsymbol{\lambda}_\psi \boldsymbol{\lambda}_\theta \begin{pmatrix} 0 \\ 0 \\ \dot{\phi} \end{pmatrix} \tag{2.26}
$$

This equation can be written as

$$
\boldsymbol{\omega} = \boldsymbol{\mu}^{-1}\dot{\boldsymbol{a}} \tag{2.27}
$$

where $\boldsymbol{a}$ is the vector containing the angles ϕ, θ and ψ. The matrix $\boldsymbol{\mu}^{-1}$ is given by

$$
\boldsymbol{\mu}^{-1} = \begin{pmatrix} -\sin\theta & 0 & 1 \\ \sin\psi\cos\theta & \cos\psi & 0 \\ \cos\psi\cos\theta & -\sin\psi & 0 \end{pmatrix} \tag{2.28}
$$

The inverse of equation (2.27) is

$$
\dot{\boldsymbol{a}} = \boldsymbol{\mu}\boldsymbol{\omega} \tag{2.29}
$$

and

$$
\boldsymbol{\mu} = \begin{pmatrix} 0 & \frac{\sin\psi}{\cos\theta} & \frac{\cos\psi}{\cos\theta} \\ 0 & \cos\psi & -\sin\psi \\ 1 & \sin\psi\tan\theta & \cos\psi\tan\theta \end{pmatrix} \tag{2.30}
$$

In equation (2.29) a singularity occurs for $\theta = \pm\pi/2$. This condition only occurs when the vehicle is heading in the vertical direction (the rotation is about the x_2''-axis). Such a situation will not be included in our simulation studies.

The angles ϕ, θ and ψ are Yaw-Pitch-Roll angles. We find it convenient to use these angles because the projection of the x_1-axis on the horizontal (inertial) plane is completely determined by the first rotation through the angle ϕ about the x_3'-axis (figure 2.1). This way, it's easy to determine the heading of the vehicle.

2.4 Free Rotation of Rectangular Plate

Consider a rectangular plate of mass m_s fixed in the body coordinate system (figure 2.2). We let the origin of the body coordinate system coincide with the center of mass of the plate ($\boldsymbol{r}_s = \mathbf{0}$). The axes of the body coordinate system are chosen to coincide with the principal axes for the plate (figure 2.2). Hence, the moment of inertia tensor $\boldsymbol{I}$ consists only of diagonal elements I_1, I_2 and I_3.

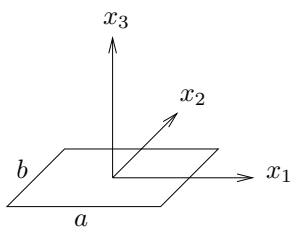


Figure 2.2: *Rectangular plate in body coordinate system (principal axes). The origin of the body coordinate system coincides with the center of mass of the plate.*

Because $\boldsymbol{r}_s = \mathbf{0}$, the equation for angular momentum (2.15) simplifies to

$$\boldsymbol{N} = \boldsymbol{I}_s\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}_s\boldsymbol{\omega} \tag{2.31}$$

Because we have chosen the axes of the body coordinate system to coincide with the principal axes of the body, the components of this equation (along the body axes) simplify to

$$\begin{aligned} I_1\dot{\omega}_1 - (I_2 - I_3)\omega_2\omega_3 &= N_1 \\ I_2\dot{\omega}_2 - (I_3 - I_1)\omega_3\omega_1 &= N_2 \\ I_3\dot{\omega}_3 - (I_1 - I_2)\omega_1\omega_2 &= N_3 \end{aligned} \tag{2.32}$$

Equations (2.32) are the Euler equations for the rotational motion of a rigid body. These equations can be written as

$$\begin{aligned}\dot{\omega}_1 &= (N_1 + (I_2 - I_3)\omega_2\omega_3)/I_1 \\ \dot{\omega}_2 &= (N_2 + (I_3 - I_1)\omega_3\omega_1)/I_2 \\ \dot{\omega}_3 &= (N_3 + (I_1 - I_2)\omega_1\omega_2)/I_3\end{aligned} \tag{2.33}$$

To determine the orientation of the body coordinate system (and of the body itself), equation (2.29) is used. The translational motion (the motion of the center of mass of the plate) is described by Newton's equation (2.1):

$$\boldsymbol{F} = m_s \boldsymbol{a}_{\mathrm{f}} \tag{2.34}$$

The Simulink model for numerically solving equations (2.33), (2.34) and (2.29) is shown in appendix B.1.

As an example the torque $\boldsymbol{N}$ is set to $\mathbf{0}$ (free rotation). The Euler equations (2.32) then read

$$\begin{aligned}I_1\dot{\omega}_1 &= (I_2 - I_3)\omega_2\omega_3 \\ I_2\dot{\omega}_2 &= (I_3 - I_1)\omega_3\omega_1 \\ I_3\dot{\omega}_3 &= (I_1 - I_2)\omega_1\omega_2\end{aligned} \tag{2.35}$$

According to these equations, if two components of the angular velocity vector $\boldsymbol{\omega}$ are 0, the third component is constant. A constant angular velocity vector directed along a principal axis corresponds to permanent rotation about that axis. These permanent rotations are stable about the axes of maximum and minimum moments of inertia, and unstable about the axis of intermediate moment of inertia. Stability here means that if a small perturbation is applied to the system, the motion will either return to its former mode or will perform small oscillations about it [4]. This is demonstrated in the following example.

Consider a plate with dimensions $a = 0.3$ m and $b = 0.2$ m (figure 2.2). The mass of the plate is set to $m_s = 1$ kg. From these parameters the three principal moments of inertia can be computed, namely

$$\begin{aligned}I_1 &= \frac{1}{12}mb^2 = \frac{0.04}{12}\ \mathrm{kg{\cdot}m^2} \\ I_2 &= \frac{1}{12}ma^2 = \frac{0.09}{12}\ \mathrm{kg{\cdot}m^2} \\ I_3 &= \frac{1}{12}m(a^2 + b^2) = \frac{0.13}{12}\ \mathrm{kg{\cdot}m^2}\end{aligned} \tag{2.36}$$

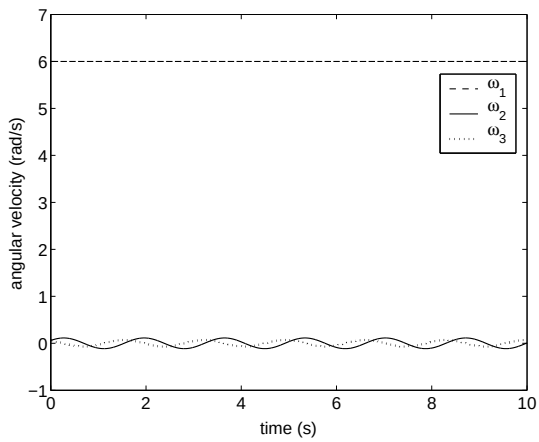


Figure 2.3: *Free rotation of rectangular plate around the x_1-axis (the principal axis corresponding to the smallest moment of inertia). Time evolution of the angular velocity vector $\boldsymbol{\omega}$ for perturbed initial condition $\boldsymbol{\omega}_0 = (6 \;\; 0.06 \;\; 0.06)$. The motion performs small oscillations about its initial condition.*

We start with rotation around the x_1-axis (the principal axis corresponding to the smallest moment of inertia) and apply a small perturbation, for example

$$\boldsymbol{\omega}_0 = (6 \;\; 0.06 \;\; 0.06) \tag{2.37}$$

The time evolution of the angular velocity vector $\boldsymbol{\omega}$ for initial condition (2.37) is displayed in figure 2.3. The motion performs small oscillations about its initial condition. Similar results are obtained for rotation around the x_3-axis (the principal axis corresponding to the greatest moment of inertia). When the rotation takes place around x_2 (the axis corresponding to the intermediate moment), however, the results are quite different (figure 2.4). The motion goes into a tumble! For more (analytical) information on this subject, see [6].

2.5 Rigid Body Motion

In this section, the axes of the body coordinate system do not necessarily need to coincide with the principal axes of the body. Hence, the moment of inertia tensor may also contain off-diagonal elements. However, the origin of the body coordinate system still is chosen to be the center of mass of the rigid body ($\boldsymbol{r}_s = \mathbf{0}$). The equation for angular momentum (2.15) therefore

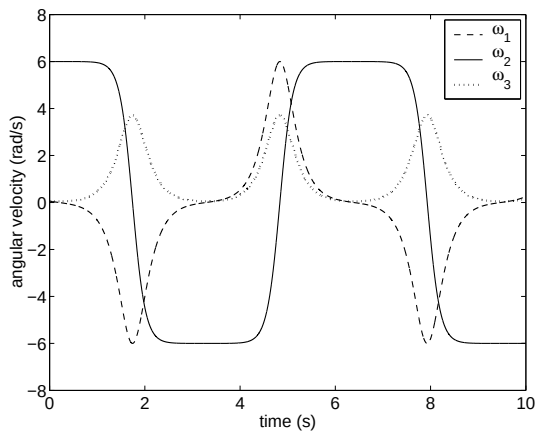


Figure 2.4: *Free rotation of rectangular plate around the x_2-axis (the principal axis corresponding to the intermediate moment of inertia). Time evolution of the angular velocity vector $\boldsymbol{\omega}$ for perturbed initial condition $\boldsymbol{\omega}_0 = (0.06 \;\; 6 \;\; 0.06)$. The motion goes into a tumble!*

simplifies to

$$\boldsymbol{N} = \boldsymbol{I}_s \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}_s \boldsymbol{\omega} \tag{2.38}$$

The center of mass of the body moves as if it were a single particle, of mass equal to the total mass of the body, acted on by the total external force [4]. Thus for the linear momentum of the body we may use equation (2.8) with $\boldsymbol{r} = \boldsymbol{r}_s = \boldsymbol{0}$ and $m = m_s$:

$$\boldsymbol{F} = m_s \left(\left(\frac{d\boldsymbol{q}}{dt} \right)_{\mathrm{b}} + \boldsymbol{\omega} \times \boldsymbol{q} \right) \tag{2.39}$$

The equations for linear momentum (2.39) and angular momentum (2.38) can be written as

$$\left(\frac{d\boldsymbol{q}}{dt} \right)_{\mathrm{b}} = \frac{1}{m_s} \boldsymbol{F} + \boldsymbol{q} \times \boldsymbol{\omega} \tag{2.40}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{I}_s^{-1} (\boldsymbol{N} - \boldsymbol{\omega} \times \boldsymbol{I}_s \boldsymbol{\omega}) \tag{2.41}$$

To determine the orientation of the body coordinate system (and of the body itself), equation (2.29) is used. The Simulink model for numerically solving equations (2.40), (2.41) and (2.29) is shown in appendix B.2. In this model, gravitational force is included. The rotation matrix $\boldsymbol{\lambda}$ is used to compute the components of the gravitational field vector $\boldsymbol{g}$ along the body coordinate

axes. The force and torque due to gravity are

$$\boldsymbol{F} = \sum \boldsymbol{F}_i = \sum m_i \boldsymbol{g} = m_s \boldsymbol{g} \tag{2.42}$$

$$\begin{aligned} \boldsymbol{N} = \sum \boldsymbol{N}_i &= \sum \boldsymbol{r}_i \times \boldsymbol{F}_i = \sum \boldsymbol{r}_i \times m_i \boldsymbol{g} \\ &= \left(\sum m_i \boldsymbol{r}_i \right) \times \boldsymbol{g} = m_s \boldsymbol{r}_s \times \boldsymbol{g} \end{aligned} \tag{2.43}$$

In this section, the torque due to gravity vanishes because $\boldsymbol{r}_s = \boldsymbol{0}$. Therefore, only equation (2.42) is included in the Simulink model.

Chapter 3

Vehicle Model

3.1 Introduction

The sloshing liquid induces an extra force and torque on the vehicle, thereby influencing the motion of the vehicle. On the other hand, the motion of the liquid is influenced by the motion of the tank in which it is contained. In other words, we are dealing with dynamical interaction between the solid body dynamics and liquid dynamics, i.e. coupled solid-liquid dynamics. The vehicle model consists of a model for the solid body dynamics S and a model for the liquid dynamics L. The model for the solid body dynamics takes the form of a relation between force and motion:

$$S[b, k] = 0 \tag{3.1}$$

where k represents the force exerted by the fluid on the vehicle and b represents the motion of the vehicle. The force k and motion b are also related by the model for the liquid dynamics:

$$L[b, k] = 0 \tag{3.2}$$

In this chapter the model S for the solid body dynamics is developed and combined with the model L for the liquid dynamics to obtain a model for the coupled solid-liquid vehicle dynamics:

$$\begin{cases} S[b, k] = 0 \\ L[b, k] = 0 \end{cases} \tag{3.3}$$

3.2 Mathematical Model

3.2.1 Geometry

The vehicle is modeled as a solid body (figure 3.1) consisting of a rectangular plate (dimensions $a \times b$, mass m_p) and a rectangular tank (dimensions $a \times b \times c$, mass m_t) with a uniform mass density. The mass of the solid body is given by $m_s = m_p + m_t$. The plate and tank are fixed with respect to the body coordinate system. We let the origin of the body system coincide with the center of mass of the plate. The position of the center of mass of the tank is defined by the vector $\boldsymbol{r}_t$, measured from the origin of the body coordinate system and directed along the x_3-axis. The solid body (tank + plate) center of mass vector $\boldsymbol{r}_s = m_t \boldsymbol{r}_t / m_s$ is also directed along the x_3-axis.

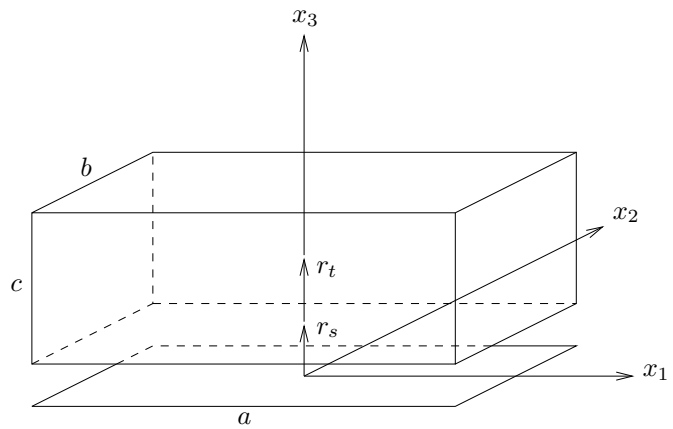


Figure 3.1: *Solid body (tank + chassis) fixed in body coordinate system* $\boldsymbol{x}$. *The position of the center of mass of the solid body is defined by the vector* $\boldsymbol{r}_s$. *The center of mass of the chassis coincides with the origin of the body coordinate system.*

The moment of inertia about the x_3-axis of the rectangular tank is given by

$$I_{33} = \frac{m_t[ab(a^2 + b^2) + c(a + b)^3]}{12(ab + ac + bc)} \tag{3.4}$$

Using Steiner's theorem, the moments of inertia of the tank about the x_1- and x_2-axes are

$$I_{11} = \frac{m_t[bc(b^2 + c^2) + a(b + c)^3]}{12(ab + ac + bc)} + m_t|\boldsymbol{r}_t|^2 \tag{3.5}$$

$$I_{22} = \frac{m_t[ac(a^2+c^2)+b(a+c)^3]}{12(ab+ac+bc)} + m_t|\boldsymbol{r}_t|^2 \tag{3.6}$$

Adding to these the moments of inertia of the rectangular plate, the moments of inertia of the solid body (tank + plate) become

$$I_{11} = \frac{m_t[bc(b^2+c^2)+a(b+c)^3]}{12(ab+ac+bc)} + \frac{m_p b^2}{12} + m_t|\boldsymbol{r}_t|^2 \tag{3.7}$$

$$I_{22} = \frac{m_t[ac(a^2+c^2)+b(a+c)^3]}{12(ab+ac+bc)} + \frac{m_p a^2}{12} + m_t|\boldsymbol{r}_t|^2 \tag{3.8}$$

$$I_{33} = \frac{m_t[ab(a^2+b^2)+c(a+b)^3]}{12(ab+ac+bc)} + \frac{m_p(a^2+b^2)}{12} \tag{3.9}$$

The products of inertia vanish because the axes of the body coordinate system are the principal axes for the solid body (the body is symmetrical under reflections through the x_1x_3- and x_2x_3-planes).

3.2.2 Solid Body Dynamics

The model for the solid body motion consists of an equation for linear momentum and an equation for angular momentum. The center of mass of the body moves as if it were a single particle, of mass equal to the total mass of the body, acted on by the total external force [4]. Thus for the linear momentum of the body we may use equation (2.8) with $\boldsymbol{r} = \boldsymbol{r}_s$ and $m = m_s$:

$$m_s\left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{b}} + \dot{\boldsymbol{\omega}} \times m_s\boldsymbol{r}_s = -m_s\boldsymbol{\omega}\times\boldsymbol{q} - \boldsymbol{\omega}\times(\boldsymbol{\omega}\times m_s\boldsymbol{r}_s) + \boldsymbol{F} + \boldsymbol{F}_l \tag{3.10a}$$

and using equation (2.15) for the angular momentum of the body, we have

$$m_s\boldsymbol{r}_s \times \left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{b}} + \boldsymbol{I}_s\dot{\boldsymbol{\omega}} = -m_s\boldsymbol{r}_s\times(\boldsymbol{\omega}\times\boldsymbol{q}) - \boldsymbol{\omega}\times\boldsymbol{I}_s\boldsymbol{\omega} + \boldsymbol{N} + \boldsymbol{N}_l \tag{3.10b}$$

In these equations $\boldsymbol{F}_l$ and $\boldsymbol{N}_l$ are respectively the force and torque that the fluid, via pressure (normal stress) and viscous effects (tangential stress), exerts on the boundary of the solid body [2]. Direct discretization of the system (3.10) would result in a method that is not stable for arbitrary liquid/solid mass ratios [2]. In particular it will become unstable when the liquid mass exceeds the solid body mass, i.e. when $m_l > m_s$. Therefore, the

system for the solid body dynamics is rewritten first. In [2] it is explained that $\boldsymbol{F}_l$ and $\boldsymbol{N}_l$ can be written as

$$\boldsymbol{F}_l = -m_l \left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{b}} - m_l \boldsymbol{\omega} \times \boldsymbol{q} - \dot{\boldsymbol{\omega}} \times m_l \boldsymbol{r}_l - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times m_l \boldsymbol{r}_l) - \boldsymbol{A}_l \quad (3.11)$$

$$\boldsymbol{N}_l = -m_l \boldsymbol{r}_l \times \left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{b}} - m_l \boldsymbol{r}_l \times (\boldsymbol{\omega} \times \boldsymbol{q}) - \boldsymbol{I}_l \dot{\boldsymbol{\omega}} - \boldsymbol{\omega} \times \boldsymbol{I}_l \boldsymbol{\omega} - \boldsymbol{B}_l \quad (3.12)$$

where $\boldsymbol{r}_l$ is the center of mass vector of the liquid and $\boldsymbol{I}_l$ is the moment of inertia tensor of the liquid. $\boldsymbol{A}_l$ and $\boldsymbol{B}_l$ are integrals over the volume V of the tank:

$$\boldsymbol{A}_l = \int_V \rho \left(\frac{D\boldsymbol{u}}{Dt} + 2\boldsymbol{\omega} \times \boldsymbol{u} - \boldsymbol{g}\right) dV \quad (3.13)$$

$$\boldsymbol{B}_l = \int_V \rho \boldsymbol{r}_v \times \left(\frac{D\boldsymbol{u}}{Dt} + 2\boldsymbol{\omega} \times \boldsymbol{u} - \boldsymbol{g}\right) dV \quad (3.14)$$

In these equations $\boldsymbol{u}$ is the velocity of a liquid particle with respect to the moving body coordinate system, $\boldsymbol{r}_v$ is the position of the liquid particle in the body system and the vector $\boldsymbol{g}$ represents acceleration due to gravity. Substituting the expressions for $\boldsymbol{F}_l$ and $\boldsymbol{N}_l$ in the system (3.10) and combining solid and liquid terms gives an alternative form of the model for the solid body dynamics, namely

$$m_c \left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{b}} + \dot{\boldsymbol{\omega}} \times m_c \boldsymbol{r}_c = -m_c \boldsymbol{\omega} \times \boldsymbol{q} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times m_c \boldsymbol{r}_c) - \boldsymbol{A}_l + \boldsymbol{F} \quad (3.15a)$$

$$m_c \boldsymbol{r}_c \times \left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{b}} + \boldsymbol{I}_c \dot{\boldsymbol{\omega}} = -m_c \boldsymbol{r}_c \times (\boldsymbol{\omega} \times \boldsymbol{q}) - \boldsymbol{\omega} \times \boldsymbol{I}_c \boldsymbol{\omega} - \boldsymbol{B}_l + \boldsymbol{N} \quad (3.15b)$$

In these equations $m_c = m_s + m_l$ is the total mass, $\boldsymbol{I}_c = \boldsymbol{I}_s + \boldsymbol{I}_l$ is the moment of inertia tensor of the coupled system, and $\boldsymbol{r}_c = (m_s \boldsymbol{r}_s + m_l \boldsymbol{r}_l)/m_c$ is the center of mass of the coupled system. The latter two quantities are time dependent because of the fluid motion with respect to the body coordinate system. The crucial point for numerical stability is that now in the left-hand side of (3.15) the total mass of the system appears, instead of only the solid body mass as in (3.10). In matrix form the system (3.15) reads

$$\boldsymbol{M} \begin{pmatrix} (d\boldsymbol{q}/dt)_b \\ \dot{\boldsymbol{\omega}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \end{pmatrix} \quad (3.16)$$

where

$$\boldsymbol{M} = \begin{pmatrix} m_c \boldsymbol{E} & \boldsymbol{H} \\ -\boldsymbol{H} & \boldsymbol{I}_c \end{pmatrix} \quad (3.17)$$

and

$$\boldsymbol{H} = \begin{pmatrix} 0 & m_c r_{c,3} & -m_c r_{c,2} \\ -m_c r_{c,3} & 0 & m_c r_{c,1} \\ m_c r_{c,2} & -m_c r_{c,1} & 0 \end{pmatrix} \tag{3.18}$$

In system (3.16) $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ refer to the right hand sides of equations (3.15a) and (3.15b). The matrix $\boldsymbol{E}$ in equation (3.17) is the identity matrix. To determine the orientation of the body coordinate system (and of the body itself), equation (2.29) is used.

3.2.3 Liquid Dynamics

The motion of a Newtonian, incompressible fluid with density ρ and molecular viscosity μ is governed by an equation for conservation of mass (continuity equation)

$$\nabla \cdot \boldsymbol{u} = 0 \tag{3.19}$$

and an equation for conservation of momentum (Navier-Stokes equation)

$$\frac{D\boldsymbol{u}}{Dt} = -\frac{1}{\rho}(\nabla p - (\nabla \cdot \mu \nabla)\boldsymbol{u}) + \boldsymbol{g} + \boldsymbol{f} \tag{3.20}$$

where $\boldsymbol{u}$ is the velocity of a liquid particle relative to the body coordinate system, p denotes the liquid pressure and $\boldsymbol{g}$ represents acceleration due to gravity. The material derivative in equation (3.20) is given by

$$\frac{D\boldsymbol{u}}{Dt} = \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \tag{3.21}$$

The extra term $\boldsymbol{f}$ in the Navier-Stokes equation (3.20) represents (virtual) acceleration due to the motion of the body coordinate system:

$$\boldsymbol{f} = -\left(\frac{d\boldsymbol{q}}{dt}\right)_{\mathrm{f}} - \dot{\boldsymbol{\omega}} \times \boldsymbol{r}_v - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_v) - 2\boldsymbol{\omega} \times \boldsymbol{u} \tag{3.22}$$

In this equation $\boldsymbol{r}_v$ is the position of the liquid particle in the body coordinate system. Equation (3.22) is similar to equation (2.7). The last term however is a totally new quantity that arises from the motion of the liquid particle in the body coordinate system (in equation (2.7) the particle was fixed in the body system). This term is called the Coriolis acceleration [4].

3.2.4 Suspension Model

The terms $\boldsymbol{F}$ and $\boldsymbol{N}$ in system (3.15) represent the force and torque due to gravity and vehicle suspension. The force and torque due to gravity are given by $m_s\boldsymbol{g}$ and $m_s\boldsymbol{r}_s \times \boldsymbol{g}$ respectively (equations (2.42) and (2.43)). For modeling the suspension of the vehicle, spring and damper combinations are used. The four vertices of the plate are connected to the ground contact points. The ground contact points are the ground points directly beneath the vertices of the plate.

ASSUMPTION A
The total mass of the wheels is uniformly distributed across the plate.

Assumption A implies that we are simply connecting the vertices to the ground contact points via massless spring and damper combinations.

ASSUMPTION B
The forces exerted by the spring and damper combinations on the vertices of the plate are directed vertically (in the x_3' direction, figure 2.1).

The restoring force is a linear function of the displacement, the damping force is a linear function of the velocity of the displacement. Because the forces exerted by the spring and damper combinations are assumed to be directed vertically, they can be computed from the vertical positions and velocities of the vertices and ground contact points. In this section the expressions for the absolute positions and velocities of the vertices are determined. The ground contact points are considered in section 4.3.

Consider the vertex located at $\boldsymbol{r}_V = (\pm a/2,\ \pm b/2,\ 0)$ in the body coordinate system. The radius vector $\boldsymbol{r}_V'$ (measured from the origin O' of the inertial coordinate system) is given by

$$\boldsymbol{r}_V' = \boldsymbol{r}_O' + \boldsymbol{\lambda}^{-1}\boldsymbol{r}_V \tag{3.23}$$

where $\boldsymbol{r}_O'$ is the vector defining the position of the origin O of the body coordinate system. Using equation (2.2), the velocity relative to the fixed (inertial) coordinate system of the vertex is

$$\boldsymbol{v}_V = \boldsymbol{\lambda}^{-1}(\boldsymbol{q} + \boldsymbol{\omega} \times \boldsymbol{r}_V) \tag{3.24}$$

The components of the velocity vector $\boldsymbol{v}_V$ are directed along the inertial axes (the rotation matrix $\boldsymbol{\lambda}^{-1}$ transforms the velocity vector $\boldsymbol{q} + \boldsymbol{\omega} \times \boldsymbol{r}_V$

into the inertial coordinate system). The magnitude of the force exerted by the spring and damper combination on the vertex is given by

$$F_V = -k(h - \xi - l) - d(v - \zeta) \tag{3.25}$$

where h and v are the x_3'-components of $\boldsymbol{r}_V'$ and $\boldsymbol{v}_V$ (i.e. the vertical position and velocity of the vertex), ξ and ζ are the vertical position and velocity of the ground contact point, k and d are the spring and damper constants and l is the rest length of the springs. The vector representation of this force (directed vertically and expressed in the body coordinate system) is given by

$$\boldsymbol{F}_V = \boldsymbol{\lambda} \begin{pmatrix} 0 \\ 0 \\ F_V \end{pmatrix} \tag{3.26}$$

The corresponding torque:

$$\boldsymbol{N}_V = \boldsymbol{r}_V \times \boldsymbol{\lambda} \begin{pmatrix} 0 \\ 0 \\ F_V \end{pmatrix} \tag{3.27}$$

Finally, the total force $\boldsymbol{F}$ and torque $\boldsymbol{N}$ due to suspension are obtained by summing these expressions over the four vertices.

3.2.5 Constraint Equation

In this section the constraint equation is formulated: The linear velocity vector $\boldsymbol{q}$ is constrained to lie in the vertical (inertial) plane through the x_1-axis of the body coordinate system and $\dot{\phi}$ is given by an input function to steer the vehicle in the desired direction.

The projection of the x_1-axis on the horizontal (inertial) plane is completely determined by the first rotation through the angle ϕ about the x_3'-axis. The direction of this projection is given by the vector

$$\boldsymbol{d} = \begin{pmatrix} \cos\phi \\ \sin\phi \\ 0 \end{pmatrix} \tag{3.28}$$

Using equations (2.29) and (2.30), the time derivative of ϕ can be written as

$$\dot{\phi} = \frac{\sin\psi}{\cos\theta}\omega_2 + \frac{\cos\psi}{\cos\theta}\omega_3 \tag{3.29}$$

Consider the constraint $\dot{\phi} = z$, where z is a time-dependent input function. Using equation (3.29), this constraint can be written as

$$\omega_2 \sin\psi + \omega_3 \cos\psi = z \cos\theta \tag{3.30}$$

The projection $\boldsymbol{q}_h$ of the linear velocity vector $\boldsymbol{q}$ on the horizontal (inertial) plane is given by the first two components of the vector $\boldsymbol{\lambda}^{-1}\boldsymbol{q} = \boldsymbol{\lambda}^t\boldsymbol{q}$, where $\boldsymbol{\lambda}^t$ is the transpose of $\boldsymbol{\lambda}$ (the transpose and the inverse of the rotation matrix $\boldsymbol{\lambda}$ are identical):

$$\boldsymbol{q}_h = \begin{pmatrix} \lambda_{11}q_1 + \lambda_{21}q_2 + \lambda_{31}q_3 \\ \lambda_{12}q_1 + \lambda_{22}q_2 + \lambda_{32}q_3 \\ 0 \end{pmatrix} \tag{3.31}$$

Next, consider the constraint $\boldsymbol{q}_h = k\boldsymbol{d}$, where k is an unspecified time-dependent function. The components of this constraint equation are

$$\begin{aligned} \lambda_{11}q_1 + \lambda_{21}q_2 + \lambda_{31}q_3 &= k\cos\phi \\ \lambda_{12}q_1 + \lambda_{22}q_2 + \lambda_{32}q_3 &= k\sin\phi \end{aligned} \tag{3.32}$$

Multiplying the first and second of these equations with $\sin\phi$ and $\cos\phi$ respectively, we obtain

$$\begin{aligned} (\lambda_{11}q_1 + \lambda_{21}q_2 + \lambda_{31}q_3)\sin\phi &= k\sin\phi\cos\phi \\ (\lambda_{12}q_1 + \lambda_{22}q_2 + \lambda_{32}q_3)\cos\phi &= k\sin\phi\cos\phi \end{aligned} \tag{3.33}$$

Combining these equations and rearranging terms, we have

$$\begin{pmatrix} \lambda_{11}\sin\phi - \lambda_{12}\cos\phi \\ \lambda_{21}\sin\phi - \lambda_{22}\cos\phi \\ \lambda_{31}\sin\phi - \lambda_{32}\cos\phi \end{pmatrix} \cdot \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} = 0 \tag{3.34}$$

Using the expressions for the components of the rotation matrix $\boldsymbol{\lambda}$ (equation (2.25)), equation (3.34) simplifies to

$$\begin{pmatrix} 0 & -\cos\psi & \sin\psi \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} = 0 \tag{3.35}$$

Combining the constraints (3.35) and (3.30), the final constraint equation becomes

$$\boldsymbol{L}\begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{\omega} \end{pmatrix} = \boldsymbol{y} \tag{3.36}$$

where

$$\boldsymbol{L} = \begin{pmatrix} 0 & -\cos\psi & \sin\psi & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sin\psi & \cos\psi \end{pmatrix} \tag{3.37}$$

and

$$\boldsymbol{y} = \begin{pmatrix} 0 \\ z\cos\theta \end{pmatrix} \tag{3.38}$$

Because the motion of the vehicle is constrained, certain forces must exist that maintain the constraint condition. Therefore, an extra term $\boldsymbol{\gamma}$ is added to the inverse of system (3.16):

$$\begin{pmatrix} (d\boldsymbol{q}/dt)_b \\ \dot{\boldsymbol{\omega}} \end{pmatrix} = \boldsymbol{M}^{-1} \begin{pmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \end{pmatrix} + \boldsymbol{\gamma} \tag{3.39}$$

The expression for $\boldsymbol{\gamma}$ is derived in section 3.3.3.

3.3 Numerical Model

3.3.1 Coupled Solid-Liquid Dynamics

System (3.10) is of the form

$$S[b,k] = 0 \tag{3.40}$$

where k represents the force exerted by the fluid on the vehicle and b represents the motion of the vehicle. Note that in the rewritten system (3.16) the force $\boldsymbol{F}_l$ and torque $\boldsymbol{N}_l$ are replaced by $\boldsymbol{A}_l$ and $\boldsymbol{B}_l$. Moreover, the center of mass and inertia matrix refer to the coupled system. The model for the solid body dynamics (3.16) is still of the form $S[b,k] = 0$, but now k represents $\boldsymbol{A}_l$ and $\boldsymbol{B}_l$ as well as $\boldsymbol{r}_l$ and $\boldsymbol{I}_l$. Therefore, the model for the coupled solid-liquid dynamics is defined as

$$\begin{cases} S[b,k] = 0 \\ L[b,k] = 0 \end{cases} \tag{3.41}$$

where

$$k = \{\boldsymbol{A}_l, \boldsymbol{B}_l, \boldsymbol{r}_l, \boldsymbol{I}_l\} \tag{3.42}$$

and

$$b = \{\boldsymbol{q}, \boldsymbol{\omega}, (d\boldsymbol{q}/dt)_b, \dot{\boldsymbol{\omega}}\} \tag{3.43}$$

We will solve this system iteratively. First we choose the order in which to do this. In reality, the interaction between S and L is a continuous process.

The choice of an order therefore is somewhat artificial, based on algorithmic considerations. The model for the liquid dynamics L is applied first, calculating k and passing it to the model for the solid body dynamics S. Then the model for the solid body dynamics is applied, calculating the motion b and passing it to the model for the liquid dynamics (figure 3.2). This iterative method is stable because the combined solid-liquid mass $m_c = m_s + m_l$ is always larger than the liquid mass m_l (see section 3.2.2 and [2]).

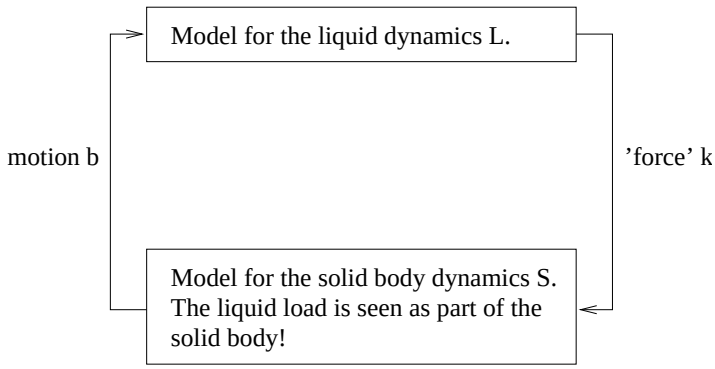


Figure 3.2: *Iterative method for solving the coupled solid-liquid dynamics. At each time step the liquid load is combined with the solid body to form the alternative solid body to which the numerically stable equations of motion apply.*

3.3.2 Solid Body Dynamics

Let's assume the linear and angular velocity vectors $\boldsymbol{q}$ and $\boldsymbol{\omega}$ as well as the position and orientation of the body coordinate system are known at time level n and are to be calculated at the new time level $n+1$. The model for the liquid dynamics is applied first, resulting in new values for $\boldsymbol{r}_c$, $\boldsymbol{I}_c$ (and thus $\boldsymbol{M}$), $\boldsymbol{A}_l$ and $\boldsymbol{B}_l$. Then, the model for the solid body dynamics is applied. The force $\boldsymbol{F}$ and torque $\boldsymbol{N}$ are computed and the matrix $\boldsymbol{M}$ is inverted to solve for $(d\boldsymbol{q}/dt)_b$ and $\dot{\boldsymbol{\omega}}$ at the new time level $n+1$:

$$\begin{pmatrix} (d\boldsymbol{q}/dt)_b \\ \dot{\boldsymbol{\omega}} \end{pmatrix}^{n+1} = (\boldsymbol{M}^{-1})^{n+1} \begin{pmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \end{pmatrix}^{n} + \boldsymbol{\gamma}^n \tag{3.44}$$

where

$$\boldsymbol{h}_1^n = -m_c \boldsymbol{\omega}^n \times \boldsymbol{q}^n - \boldsymbol{\omega}^n \times (\boldsymbol{\omega}^n \times m_c \boldsymbol{r}_c^{n+1}) - \boldsymbol{A}_l^n + \boldsymbol{F}^n \tag{3.45}$$

$$\boldsymbol{h}_2^n = -m_c \boldsymbol{r}_c^{n+1} \times (\boldsymbol{\omega}^n \times \boldsymbol{q}^n) - \boldsymbol{\omega}^n \times \boldsymbol{I}_c^{n+1} \boldsymbol{\omega}^n - \boldsymbol{B}_l^n + \boldsymbol{N}^n \tag{3.46}$$

and

$$\boldsymbol{A}_l^n = \int_V \rho \left(\frac{D\boldsymbol{u}^{n+1}}{Dt} + 2\boldsymbol{\omega}^n \times \boldsymbol{u}^{n+1} - \boldsymbol{g}^n \right) dV \tag{3.47}$$

$$\boldsymbol{B}_l^n = \int_V \rho \boldsymbol{r}_v^{n+1} \times \left(\frac{D\boldsymbol{u}^{n+1}}{Dt} + 2\boldsymbol{\omega}^n \times \boldsymbol{u}^{n+1} - \boldsymbol{g}^n \right) dV \tag{3.48}$$

From equation (2.29), the angular velocities $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$ are computed as

$$\dot{\boldsymbol{a}}^n = \boldsymbol{\mu}^n \boldsymbol{\omega}^n \tag{3.49}$$

The position and orientation of the body coordinate system are computed at time level $n+1$ by integration of $(\boldsymbol{\lambda}^{-1})^n \boldsymbol{q}^n$ (linear velocity expressed in fixed coordinate system) and $\dot{\boldsymbol{a}}^n$ respectively. The linear and angular velocity vectors $\boldsymbol{q}^{n+1}$ and $\boldsymbol{\omega}^{n+1}$ are obtained by integration of $(d\boldsymbol{q}/dt)_b^{n+1}$ and $\dot{\boldsymbol{\omega}}^{n+1}$. The Simulink model for solving system (3.44) is shown in appendix B.3.

3.3.3 Constraint Equation

System (3.39) is solved using the forward Euler integration method:

$$\begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{\omega} \end{pmatrix}^{n+1} = \begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{\omega} \end{pmatrix}^{n} + \Delta t[(\boldsymbol{M}^{-1})^{n+1} \begin{pmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \end{pmatrix}^{n} + \boldsymbol{\gamma}^n] \tag{3.50}$$

The constraint equation (3.36) is discretized as

$$\boldsymbol{L}^n \begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{\omega} \end{pmatrix}^{n+1} = \boldsymbol{y}^n \tag{3.51}$$

Insert (3.50) into (3.51) to obtain

$$\boldsymbol{L}^n \left[\begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{\omega} \end{pmatrix}^{n} + \Delta t[(\boldsymbol{M}^{-1})^{n+1} \begin{pmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \end{pmatrix}^{n} + \boldsymbol{\gamma}^n] \right] = \boldsymbol{y}^n \tag{3.52}$$

This equation can be written as

$$\Delta t \boldsymbol{L}^n \boldsymbol{\gamma}^n = -\boldsymbol{L}^n \left[\begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{\omega} \end{pmatrix}^{n} + \Delta t(\boldsymbol{M}^{-1})^{n+1} \begin{pmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \end{pmatrix}^{n} \right] + \boldsymbol{y}^n \tag{3.53}$$

We seek a solution in the form $\boldsymbol{\gamma}^n = (\boldsymbol{L}^t)^n \boldsymbol{p}^n$, for some vector $\boldsymbol{p}^n$. Substitute this form of $\boldsymbol{\gamma}^n$ into equation (3.53) and use $\boldsymbol{L}^n(\boldsymbol{L}^t)^n = \boldsymbol{E}$ to obtain

$$\boldsymbol{p}^n = -\boldsymbol{L}^n \left[\frac{1}{\Delta t} \begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{\omega} \end{pmatrix}^{n} + (\boldsymbol{M}^{-1})^{n+1} \begin{pmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \end{pmatrix}^{n} \right] + \frac{1}{\Delta t} \boldsymbol{y}^n \tag{3.54}$$

From this, the vector $\boldsymbol{\gamma}^n = (\boldsymbol{L}^t)^n \boldsymbol{p}^n$ can be computed. For more information, see [1].

3.3.4 Stability Analysis

The integration of system (3.39) is done using the first order forward Euler method (further adaptations to the vehicle model are necessary for higher order integration methods to work). In this section we investigate stability for the Euler method using system (3.15) in which $\boldsymbol{q}$, $\boldsymbol{\omega}$, $\boldsymbol{r}_c$ and $\boldsymbol{I}_c$ are replaced with scalars and $\boldsymbol{A}_l$, $\boldsymbol{B}_l$, $\boldsymbol{F}$ and $\boldsymbol{N}$ are omitted:

$$\begin{aligned} m_c\dot{q} + m_c r_c \dot{\omega} &= -m_c q\omega - m_c r_c \omega^2 \\ m_c r_c \dot{q} + I_c \dot{\omega} &= -m_c r_c q\omega - I_c \omega^2 \end{aligned} \tag{3.55}$$

This system is equivalent to

$$\begin{aligned} \dot{q} &= -q\omega \\ \dot{\omega} &= -\omega^2 \end{aligned} \tag{3.56}$$

Note that we now have a decoupled equation for ω. When solving a differential equation $\dot{x} = f(x)$ with forward Euler, the amplification factor g is given by

$$g = 1 + \frac{df}{dx}\Delta t \tag{3.57}$$

For the equations in system (3.56) the amplification factors are $1 - \omega\Delta t$ and $1 - 2\omega\Delta t$. For absolute stability we need $|g| \leq 1$. Since ω can be positive as well as negative, absolute stability is out of the question. The Euler method is zero stable however, since $|g| \leq 1 + O(\Delta t)$. Zero stability guarantees that for sufficiently small Δt the discrete solution becomes a good approximation of the continuous solution [7].

The integration of the Navier-Stokes equations and the iterative process of solving system (3.41) are stable also. The stability of the numerical coupling has been accomplished by rewriting the equations for linear and angular momentum in section 3.2.2 (see [2] for more information on this subject). Information on (the stability of) the discretized Navier-Stokes equations can be found in [2] and [7].

Chapter 4

Results

In this chapter the results of four different simulations are presented. The main goal of the simulations 'vertical motion' and 'inclined free surface' is to validate the model for the coupled solid-liquid dynamics. The simulations 'bumpy road' and 'cornering maneuver' are included to demonstrate the effects of dynamical interaction in realistic situations. The simulations are performed on a grid of $40 \times 20 \times 20$ cells (the computational grid used in ComFlo), except 'inclined free surface', for which we use a finer grid of $60 \times 30 \times 30$ cells. Integration in Simulink is done using the first order Euler method, with time step $\Delta t = 0.0005$ (further adaptations to the vehicle model are necessary for higher order integration methods to work).

4.1 Vertical Motion

First we set the vehicle parameters. Consider a tank with dimensions $a = 6$ m, $b = 2.5$ m, $c = 2$ m (figure 3.1) and mass $m_t = 10000$ kg. The center of mass of the (empty) tank is located at $\boldsymbol{r}_t = (0, 0, 1.5)$, i.e. the distance between the chassis and the bottom of the tank is equal to 0.5 m. The mass of the chassis is set to $m_p = 15000$ kg. The lower half of the tank is filled with liquid having a density of $\rho = 1000$ kg/m^3. Hence, the liquid mass is equal to $m_l = 15000$ kg.

Suspension parameters (restoring and damping) are set to $k = 500000$ N/m and $d = 20000$ N·s/m respectively for each of the four spring and damper combinations. The natural spring length is set to $l = 0.5$ m. In equilibrium position (the position at which the spring force equals the gravitational force), the spring length is reduced to 0.304 m.

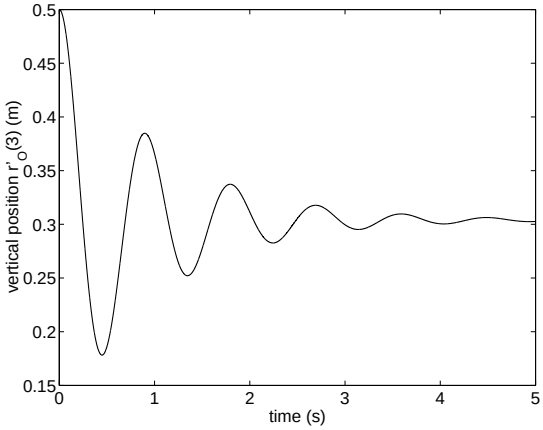


Figure 4.1: *Vertical position of the chassis for initial condition* $\boldsymbol{r}'_O(3) = 0.5$. *The equilibrium position is at* $x'_3 = 0.304$.

We start with simulating vertical motion on a flat ground surface. The flat ground surface is modeled by simply setting ξ and ζ (the vertical position and velocity of the ground contact points) to 0 for all of the four ground contact points. The vehicle is initialized with zero linear and rotational velocity. We set the initial position to $\boldsymbol{r}'_O = (0, 0, 0.5)$ (above the equilibrium position which is at $x'_3 = 0.304$). Damped oscillatory motion is expected in the vertical direction due to gravitational, restoring and damping forces. This is confirmed in figure 4.1, where the vertical position of the origin O of the body coordinate system is shown.

Since the acceleration of the tank does not exceed the gravitational acceleration, the fluid remains at rest with respect to the body coordinate system. Under these circumstances, the effects of the liquid on the motion of the vehicle are the same as those of a solid of equal mass and volume. Indeed, the vertical motion is simply that of a particle of mass $m_c = m_t + m_p + m_l$, acted on by the total force. The analytical solution is given by

$$x'_3(t) = e^{-t}[0.196\cos(7t) + \frac{0.196}{7}\sin(7t)] + 0.304 \tag{4.1}$$

The error in the numerical solution is plotted for various time steps in figure 4.2. When the time step is reduced by a factor two, the error in the numerical solution becomes twice as small due to the first order Euler time integration.

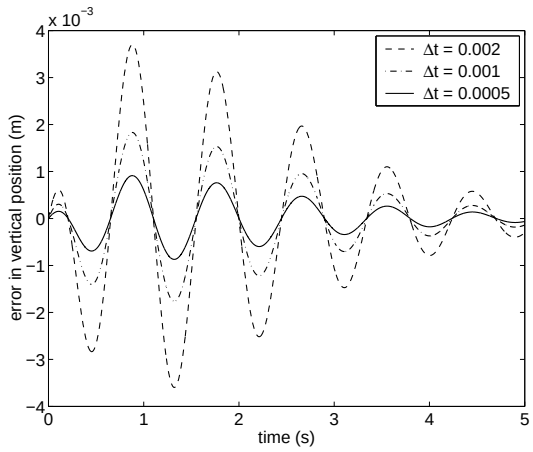


Figure 4.2: *Error in the vertical position for various time steps.*

4.2 Inclined Free Surface

In this test case the vehicle is initially at rest in its equilibrium position. The ground surface is flat. For the initial fluid configuration we select 'liquid on side of a plane': At $t = 0$ the fluid is on the lower side of the plane defined

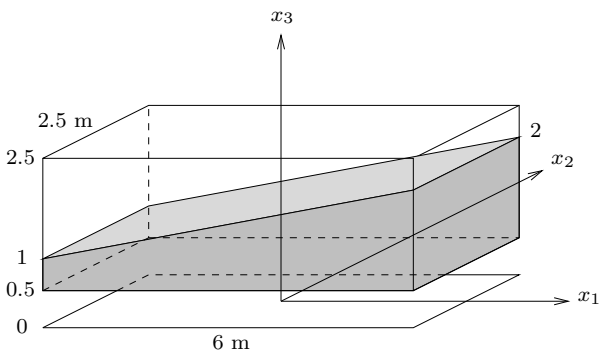


Figure 4.3: *Initial fluid configuration.*

by the equation $x_1 - 6x_3 = -9$ (figure 4.3). Gravity sets the fluid into motion and the vehicle accelerates due to the force exerted by the fluid. The liquid moves from side to side. This can be seen in figure 4.4, where the liquid height at the rear side of the tank is shown. The x_1'-component of the position $\boldsymbol{r}_O'$ of the vehicle and its pitch angle θ are shown in figures 4.5 and 4.6.

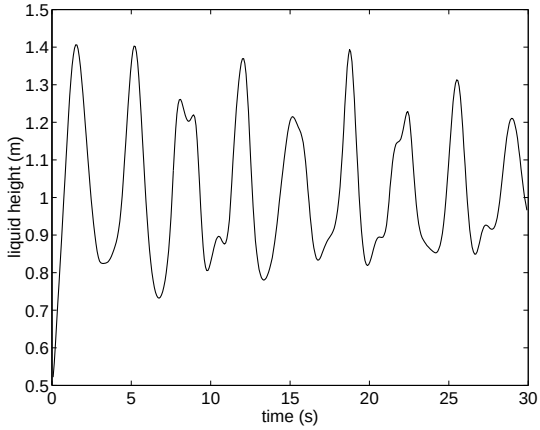


Figure 4.4: *Liquid height at the rear side of the tank*

Let's predict the frequency of the motion by looking at the sloshing motion of the liquid in a fixed tank. Then, the natural (angular) frequency ω for the oscillation of the liquid is given by [5]

$$\omega^2 = \frac{\pi g}{l} \tanh \frac{\pi h}{l} \tag{4.2}$$

where l is the length of the tank and h is the equilibrium position of the free surface. With $l = 6$ m, $h = 1$ m and $g = 9.8$ m/s^2, the natural period $2\pi/\omega$ for the oscillation of the liquid is approximately 4.0 s. The period of the vehicle motion is expected to be roughly the same. This is confirmed by the simulation results (the period is approximately 3.4 s).

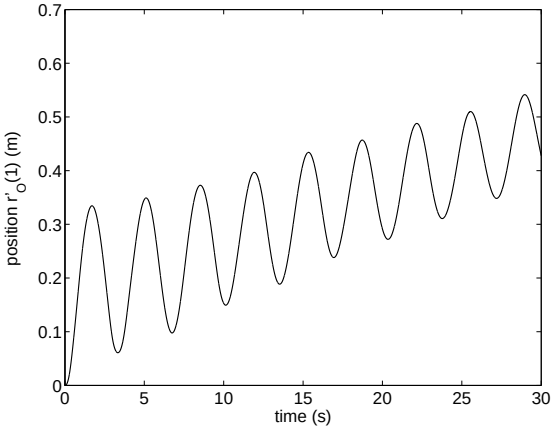


Figure 4.5: *Horizontal (forward) position of the chassis center of mass O.*

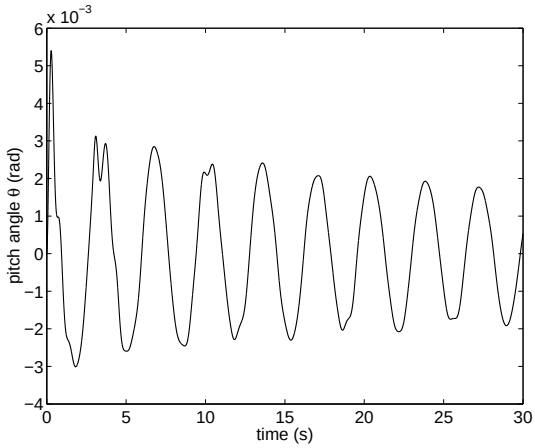


Figure 4.6: *Pitch motion of the vehicle (rotation about the x_2-axis).*

4.3 Bumpy Road

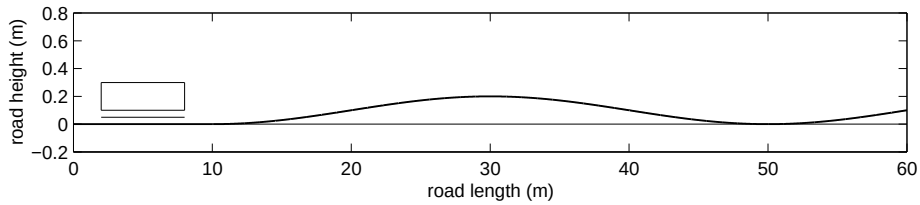


Figure 4.7: *Sinusoidal road surface. Note that the axes have different scales!*

In this section we let the vehicle drive straight ahead along a sinusoidal road surface of wavelength 40 m and height between 0 and 0.2 m (figure 4.7). The corresponding ground surface function is given by

$$\xi(x_1') = \begin{cases} \frac{1}{10}[1 + \sin(\frac{\pi}{20}(x_1' + 20))] & x_1' \geq 10 \\ 0 & x_1' < 10 \end{cases} \tag{4.3}$$

and its time derivative

$$\zeta(x_1', \frac{dx_1'}{dt}) = \begin{cases} \frac{\pi}{200}\frac{dx_1'}{dt}\cos(\frac{\pi}{20}(x_1' + 20)) & x_1' \geq 10 \\ 0 & x_1' < 10 \end{cases} \tag{4.4}$$

The vertical positions and velocities of the ground contact points are obtained by evaluating these functions in the x_1'-components of $\boldsymbol{r}_V'$ and $\boldsymbol{v}_V$ (the absolute positions and velocities of the vertices, equations (3.23) and (3.24)). We are interested in the effects of the sloshing liquid on the ride behavior of the vehicle. Therefore, simulations are repeated without the solid-liquid interaction. In order to be able to make a fair comparison the liquid is replaced with a solid beam of equal mass and volume. A separate Simulink model has been developed: The solid-liquid interaction is removed and the beam is included. The moments of inertia of the beam with dimensions $a \times b \times d$ and mass m_b are given by

$$I_{33} = \frac{1}{12}m_b(a^2 + b^2) \tag{4.5}$$

and, using Steiner's theorem

$$I_{11} = \frac{1}{12}m_b(b^2 + d^2) + m_b|\boldsymbol{r}_b|^2 \tag{4.6}$$

$$I_{22} = \frac{1}{12}m_b(a^2 + d^2) + m_b|\boldsymbol{r}_b|^2 \tag{4.7}$$

where $\boldsymbol{r}_b$ is the center of mass vector of the beam. We start with simulations for initial velocities 10, 11, 12 and 13 m/s. For these velocities the frequency of the excitation generated by the road surface is roughly the same as the natural frequency for the oscillation of the liquid, resulting in significant liquid slosh (the fluid responds in the excitation frequency). This can be seen in figures 4.8, 4.10, 4.12 and 4.14 where the liquid height at the rear side of the tank is shown (results for the front side are similar). Figures 4.9, 4.11, 4.13 and 4.15 show the pitch angle θ for both the rigid cargo and liquid cargo simulations.

As can be seen, the effects of the sloshing liquid on the ride behavior are opposite for the velocities 10, 11 m/s (increase in pitch motion) and 12, 13 m/s (decrease in pitch motion). For a velocity of 12 m/s the frequency of the excitation generated by the road surface is approximately the same as the 'natural' frequency for the vehicle motion as experimentally found in section 4.2. We would normally expect a resonance peak to occur for this frequency. The largest (steady response) amplitude of the liquid height is indeed found in figure 4.12, for a velocity of 12 m/s.

Simulation results for an initial velocity of 20 m/s are included to demonstrate that the liquid motion not always becomes as exaggerated as in the previous simulations, where the road excitation frequency and the natural liquid frequency were roughly the same. In figure 4.16 can be seen that for a velocity of 20 m/s the liquid height is almost unaffected by the excitation generated by the road surface. The pitch angle θ is shown for both rigid cargo and liquid cargo simulations in figure 4.17. As expected, the liquid has little effect on the motion of the vehicle.

We conclude this section with a simulation for initial velocity 23 m/s. The excitation frequency is now approximately twice as large as the natural frequency for the oscillation of the liquid. Subharmonic resonance effects are clearly visible in figures 4.18 and 4.19, which show the liquid height at the rear side of the tank and the pitch angle θ. It takes about 80 seconds for the fluid to reach a steady response. Moreover, whereas the fluid starts to respond in the excitation frequency, the final response is in the basic eigenfrequency. Since the excitation frequency (and thus the frequency of the pitch motion) is twice as large, the sloshing liquid has an alternating increasing and decreasing effect on the amplitude of the pitch motion, as can be seen in figure 4.19. For similar results, see [8].

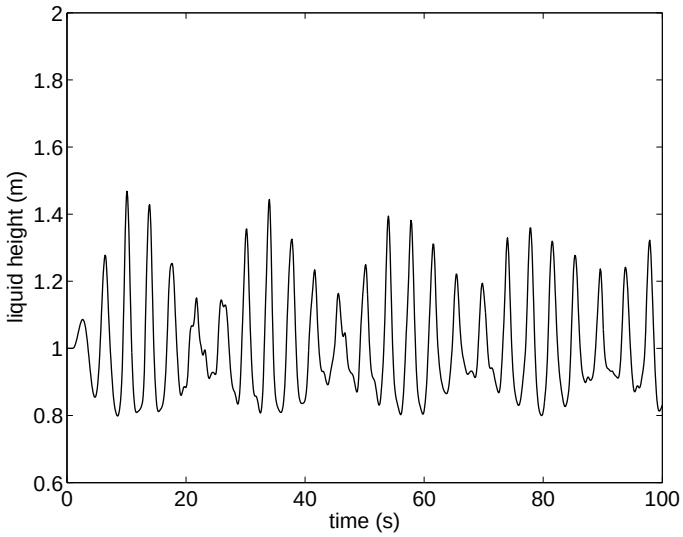


Figure 4.8: *Liquid height at the rear side of the tank for a velocity of* 10 *m/s.*

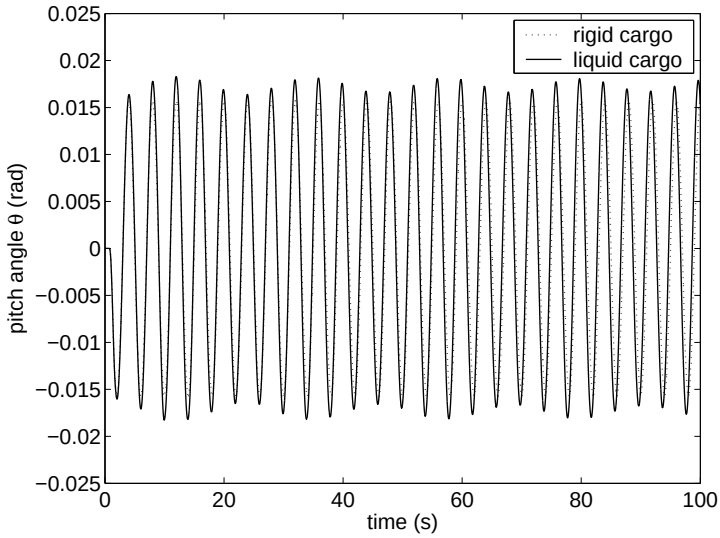


Figure 4.9: *Influence of the liquid cargo on the pitch motion of the vehicle for a velocity of* 10 *m/s.*

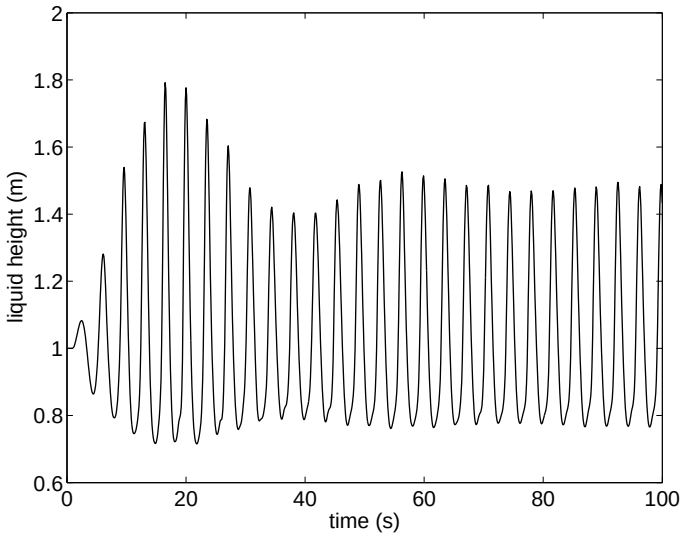


Figure 4.10: *Liquid height at the rear side of the tank for a velocity of* 11 *m/s.*

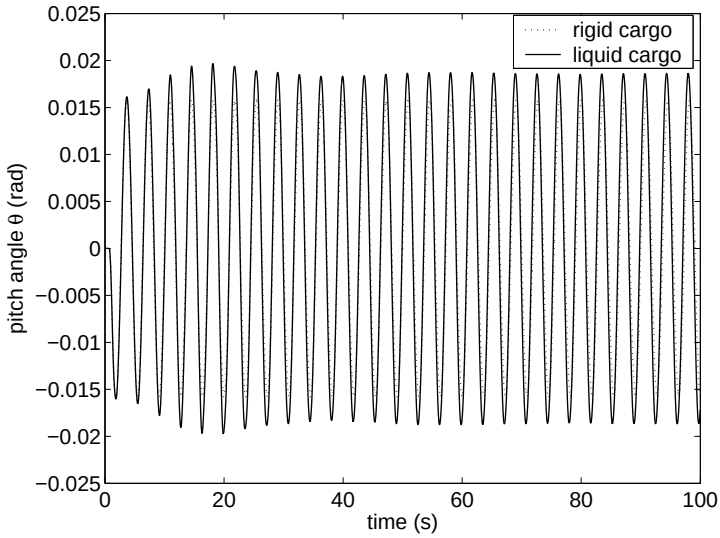


Figure 4.11: *Influence of the liquid cargo on the pitch motion of the vehicle for a velocity of* 11 *m/s.*

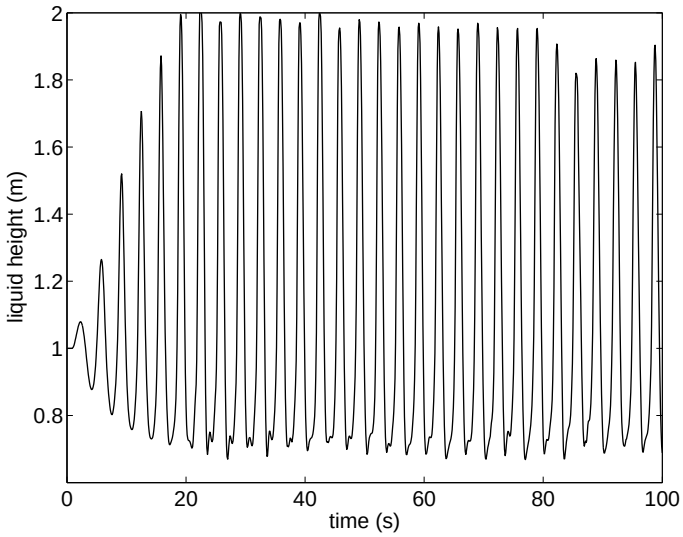


Figure 4.12: *Liquid height at the rear side of the tank for a velocity of* 12 *m/s.*

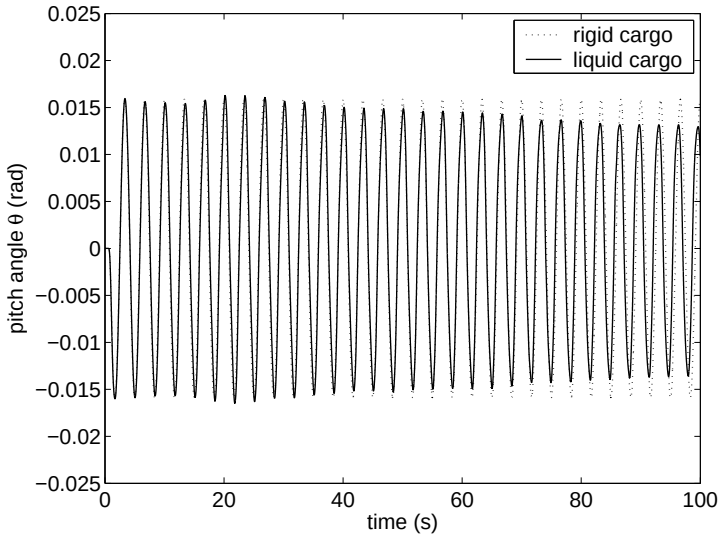


Figure 4.13: *Influence of the liquid cargo on the pitch motion of the vehicle for a velocity of* 12 *m/s.*

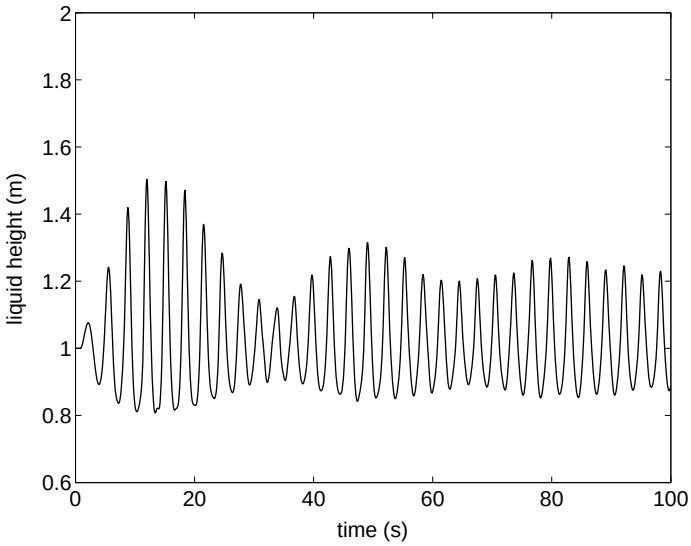


Figure 4.14: *Liquid height at the rear side of the tank for a velocity of* 13 *m/s.*

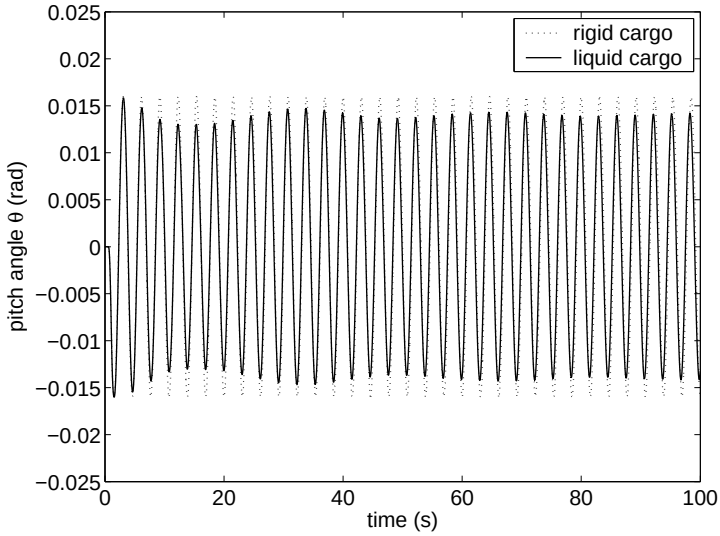


Figure 4.15: *Influence of the liquid cargo on the pitch motion of the vehicle for a velocity of* 13 *m/s.*

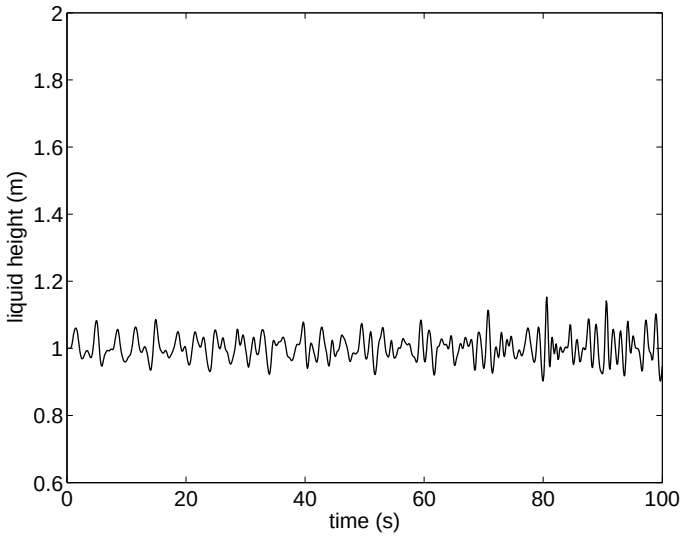


Figure 4.16: *Liquid height at the rear side of the tank for a velocity of* 20 *m/s.*

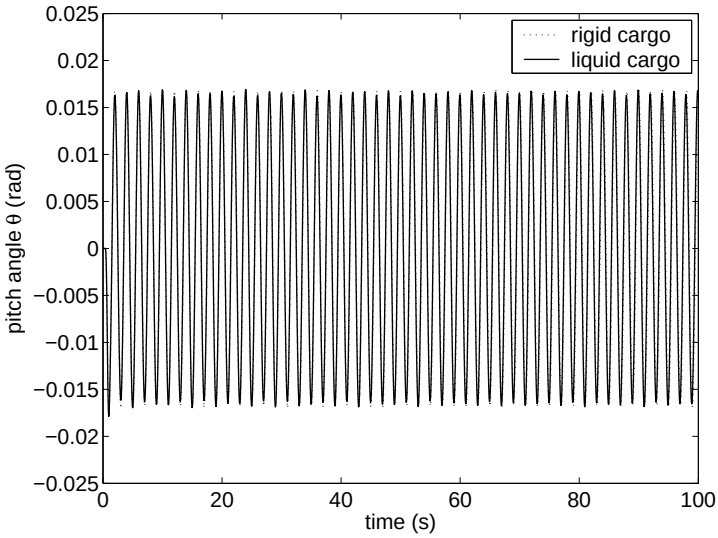


Figure 4.17: *Influence of the liquid cargo on the pitch motion of the vehicle for a velocity of* 20 *m/s.*

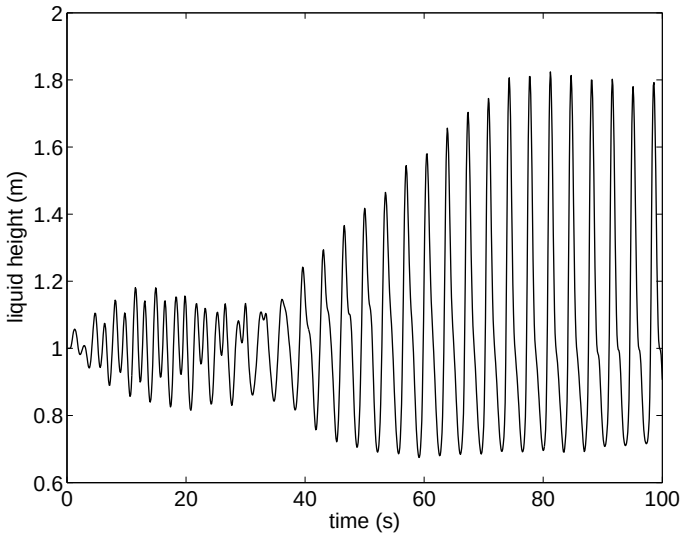


Figure 4.18: *Liquid height at the rear side of the tank for a velocity of* 23 *m/s.*

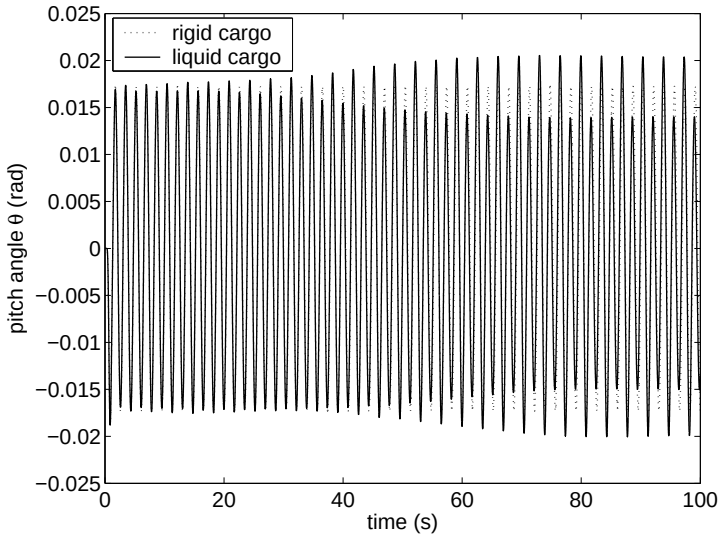


Figure 4.19: *Influence of the liquid cargo on the pitch motion of the vehicle for a velocity of* 23 *m/s.*

4.4 Cornering Maneuver

First we have to make an adaptation to the constraint equation because it prevents the centrifugal force from having an effect on the roll motion of the vehicle (rotation about the x_1-axis, figure 3.1). Until now, the centrifugal forces were small. When taking a bend however, the resulting centrifugal force is a very important quantity. Therefore, we have to find a way around this.

Note that $\boldsymbol{\gamma}$ in equation (3.39) represents acceleration. The corresponding force and torque are given by

$$\boldsymbol{M\gamma} = \begin{pmatrix} \boldsymbol{F}_\gamma \\ \boldsymbol{N}_\gamma \end{pmatrix} \tag{4.8}$$

The first three components of $\boldsymbol{M\gamma}$, denoted by $\boldsymbol{F}_\gamma$, can be seen as forces acting on the origin O of the body coordinate system. The last three components, denoted by $\boldsymbol{N}_\gamma$, represent torque. Since the angular velocity $\dot{\phi}$ is directed along the x_3'-axis (fixed coordinate system), we want the torque $\boldsymbol{N}_\gamma$ to be directed along the x_3'-axis also. However, the vector $\boldsymbol{\gamma}$ produces large amounts of additional torque about the x_1-axis (body coordinate system), in effect counteracting the centrifugal force. To circumvent this problem we omit the x_1'- and x_2'-components of $\boldsymbol{N}_\gamma$, i.e. we replace the torque $\boldsymbol{N}_\gamma$ by its projection $(\boldsymbol{N}_\gamma)_{x_3'}$ on the x_3'-axis:

$$(\boldsymbol{N}_\gamma)_{x_3'} = \boldsymbol{\lambda} \begin{pmatrix} 0 \\ 0 \\ (\lambda_{13}\ \lambda_{23}\ \lambda_{33}) \cdot \boldsymbol{N}_\gamma \end{pmatrix} \tag{4.9}$$

The force $\boldsymbol{F}_\gamma$ and torque $(\boldsymbol{N}_\gamma)_{x_3'}$ are applied to the inverse of system (3.16).

Instead of keeping $\dot{\phi} = 0$ to prevent a change of direction we will now force the vehicle to take a bend. The initial velocity is set to 15 m/s and during the first second of the simulation we steer the vehicle straight ahead by setting $\dot{\phi} = 0$ rad/s. After the first second, $\dot{\phi}$ is changed to 0.2 rad/s and the vehicle takes a bend. For these parameters, the radius of the curve is equal to $r = 75$ m. At the bend the centrifugal force causes the vehicle to rotate around the x_1-axis and the fluid is moving outwards. The roll angle ψ is shown for both rigid cargo and liquid cargo simulations in figure 4.20.

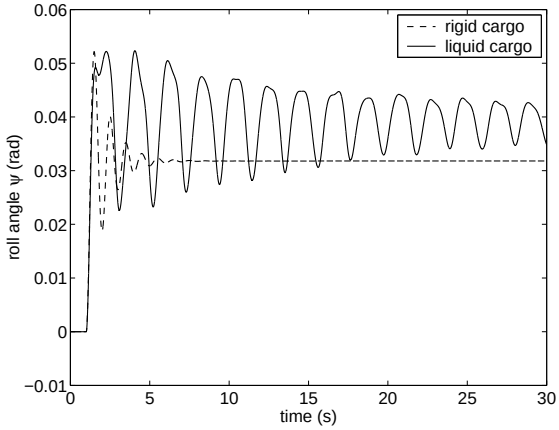


Figure 4.20: *Influence of the liquid cargo on the roll motion of the vehicle for a velocity of* 15 *m/s.*

The effects of the sloshing liquid on the motion of the vehicle are clearly visible (an increased, oscillating roll angle). The increase of the roll angle is caused by the lateral shift of the liquid load. Let's predict the frequency of the motion by looking at the sloshing motion of the liquid in a fixed tank. Using equation (4.2), now with $l = 2.5$ m, the natural period $2\pi/\omega$ for the oscillation of the liquid is approximately 1.9 s. We expect the period of the vehicle motion (the liquid induced oscillatory roll motion) to be roughly the same. This is confirmed by the simulation results.

The angle α between the free surface and the horizontal (inertial) plane can also be predicted. Since the centrifugal and gravitational acceleration are approximately $\dot{\phi}^2 r$ and -10, the angle α becomes

$$\alpha \approx \arctan \frac{\dot{\phi}^2 r}{10} = \arctan 0.3 \approx 0.29 \tag{4.10}$$

Since the roll angle ψ is approximately 0.04 rad (figure 4.20), the angle between the free surface and the bottom of the tank is estimated to be 0.33 rad. This means that the difference in liquid height at the left and right side of the tank should eventually become 0.86 m. Considering the difference graph in figure 4.21, this looks to be a reasonable estimate.

The simulation is repeated for a velocity twice as large ($q_1 = 30$ m/s, $\dot{\phi} = 0.4$ rad/s) to demonstrate effects on rollover performance. The centrifugal acceleration now exceeds the gravitational acceleration and the angle between

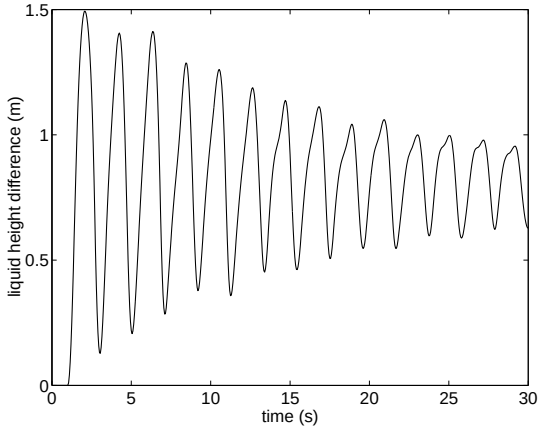


Figure 4.21: *Difference in liquid height at the left and right side of the tank for a velocity of* 15 *m/s.*

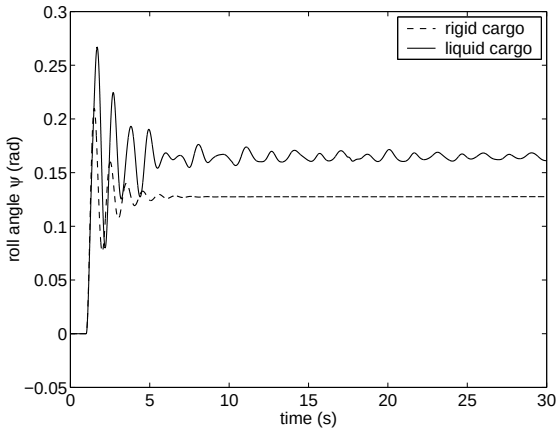


Figure 4.22: *Influence of the liquid cargo on the roll motion of the vehicle for a velocity of* 30 *m/s.*

the free surface and the horizontal (inertial) plane is approximately 0.9 rad. The roll angle ψ is shown for both rigid cargo and liquid cargo simulations in figure 4.22. As can be seen, the roll angle ψ is increased by the liquid load. Although the vehicle is assumed to stay in contact with the road (the vertices of the chassis are connected to the ground contact points via spring and damper combinations), it is clear that the liquid increases the chance of rollover.

Appendix A

Adaptations to ComFlo

The main program of ComFlo consists of three parts: The program setup, the time cycle and the termination. From these parts numerous subroutines are called:

```
      PROGRAM COMFLO
C
C  Program setup
C
      T = 0.0
C
      CALL SETPAR
      CALL SETCSA
      CALL BNDLAB
      CALL SETFLD
      IF (SLOSH .EQ. 0) CALL COEFL
      IF ((DTAVS .LE. TMAX) .AND. (T .EQ. 0.0)) CALL AVS
      IF ((DTMATL .LE. TMAX) .AND. (T .EQ. 0.0)) CALL MATLAB
      IF ((DTM3D .LE. TMAX) .AND. (T .EQ. 0.0)) CALL MATL3D
      IF ((DTPLIC .LE. TMAX) .AND. (T .EQ. 0.0)) THEN
        CALL PLIC
        CALL MLPLIC
      ENDIF
      IF ((DTVTK .LE. TMAX) .AND. (T .EQ. 0.0)) CALL VTK
      IF ((DTCSA .LE. TMAX) .AND. (T .EQ. 0.0)) CALL CSA
      CALL LIQPCT(1, PCT)
C
C  Time cycle
C
 10   CONTINUE
        CALL INIT
```

```
        CALL TILDE
        CALL SOLVEP
C
        IF (SLOSH .EQ. 1) CALL PLIC
        IF (SLOSH .EQ. 1) CALL VFCONV
        CALL BC
        IF (TUMBLQ .EQ. 1) CALL TUMBLE(0)
C
        T = T + DT
C
        CALL DTADJ
C
        IF ((T+0.5*DT .GE. DTPRNT*FLOAT(NRPRNT)) .OR. (CYCLE .LE. 10))
     &       CALL PRNT
        IF (T+0.5*DT .GE. DTAVS*FLOAT(NRAVS)) CALL AVS
        IF (T+0.5*DT .GE. DTMATL*FLOAT(NRMATL)) CALL MATLAB
        IF (T+0.5*DT .GE. DTM3D*FLOAT(NRM3D)) CALL MATL3D
        IF (T+0.5*DT .GE. DTPLIC*FLOAT(NRPLIC)) CALL MLPLIC
        IF (T+0.5*DT .GE. DTVTK*FLOAT(NRVTK)) CALL VTK
        IF (T+0.5*DT .GE. DTCSA*FLOAT(NRCSA)) CALL CSA
        IF (T+0.5*DT .GE. DTCOM*FLOAT(NRCOM)) CALL COM(1)
        IF (T+0.5*DT .GE. DTMOI*FLOAT(NRMOI)) CALL MOI(1)
        IF (T+0.5*DT .GE. DTTANK*FLOAT(NRTANK)) CALL TUMBLE(1)
        IF (T+0.5*DT .GE. DTFILL*FLOAT(NRFILL)) CALL FILLBX
        IF (T+0.5*DT .GE. DTFRC*FLOAT(NRFRC)) CALL FRCBX
        IF (T+0.5*DT .GE. DTFLUX*FLOAT(NRFLUX)) CALL FLUXBX
        IF (T+0.5*DT .GE. DTMNTR*FLOAT(NRMNTR)) CALL MNTR
        CALL STREAM
C
        LOADQ = 0
        CALL AUTOSV
C
        CALL STEER
C
      IF (T+0.5*DT .LT. TMAX) GOTO 10
C
C  Termination
C
      CALL LIQPCT(1, PCT)
C      OPEN(UNIT=11, FILE='dthist.dat', POSITION='append')
CRAY
      OPEN(UNIT=11, FILE='dthist.dat')
 20   CONTINUE
        READ(11,'(A1)',END=30)
      GOTO 20
```

```
 30   WRITE(11,'(E12.4,I10,I4,E12.4)') T, CYCLE+1, 0, DT
      CLOSE(11)
C
      STOP
      END
C
C  End of MAIN.
C
C  ---------------------------------------------------------------------
C
```

To execute the ComFlo code from Simulink, a level 1 Fortran-MEX S-function is used (see also [3]). The template file for Fortran MEX S-functions contains only subroutines and merely copies the input to the output. We will use this template file and edit it to perform the ComFlo operations. The template file subroutine *output* performs the output calculations (at each time step). This is where the time cycle part of ComFlo should be placed, without the time loop itself. The program setup and termination part of ComFlo can be included in the template file subroutines *initcond* and *stopcomflo* respectively:

```
C
C File:  sfuntmpl_fortran.f
C
C=====================================================================
      SUBROUTINE INITCOND(X0_S)
C
C  Program setup
C
      T = 0.0
C
      CALL SETPAR
      CALL SETCSA
      CALL BNDLAB
      CALL SETFLD
      IF (SLOSH .EQ. 0) CALL COEFL
      IF ((DTAVS .LE. TMAX) .AND. (T .EQ. 0.0)) CALL AVS
      IF ((DTMATL .LE. TMAX) .AND. (T .EQ. 0.0)) CALL MATLAB
      IF ((DTM3D .LE. TMAX) .AND. (T .EQ. 0.0)) CALL MATL3D
      IF ((DTPLIC .LE. TMAX) .AND. (T .EQ. 0.0)) THEN
        CALL PLIC
        CALL MLPLIC
      ENDIF
      IF ((DTVTK .LE. TMAX) .AND. (T .EQ. 0.0)) CALL VTK
```

```
      IF ((DTCSA .LE. TMAX) .AND. (T .EQ. 0.0)) CALL CSA
      CALL LIQPCT(1, PCT)
C
      RETURN
      END

C=====================================================================
      SUBROUTINE OUTPUT(T_S, X_S, U_S, Y_S)
C
C  Time cycle
C
      CALL INIT
      CALL TILDE(U_S)
      CALL SOLVEP
C
      IF (SLOSH .EQ. 1) CALL PLIC
      IF (SLOSH .EQ. 1) CALL VFCONV
      CALL BC
      IF (TUMBLQ .EQ. 1) CALL TUMBLE(0,Y_S)
C
      T = T + DT
C
      CALL DTADJ
C
      IF ((T+0.5*DT .GE. DTPRNT*FLOAT(NRPRNT)) .OR. (CYCLE .LE. 10))
     &       CALL PRNT
      IF (T+0.5*DT .GE. DTAVS*FLOAT(NRAVS)) CALL AVS
      IF (T+0.5*DT .GE. DTMATL*FLOAT(NRMATL)) CALL MATLAB
      IF (T+0.5*DT .GE. DTM3D*FLOAT(NRM3D)) CALL MATL3D
      IF (T+0.5*DT .GE. DTPLIC*FLOAT(NRPLIC)) CALL MLPLIC
      IF (T+0.5*DT .GE. DTVTK*FLOAT(NRVTK)) CALL VTK
      IF (T+0.5*DT .GE. DTCSA*FLOAT(NRCSA)) CALL CSA
      IF (T+0.5*DT .GE. DTCOM*FLOAT(NRCOM)) CALL COM(1)
      IF (T+0.5*DT .GE. DTMOI*FLOAT(NRMOI)) CALL MOI(1)
      IF (T+0.5*DT .GE. DTTANK*FLOAT(NRTANK))
     &       CALL TUMBLE(1,Y_S)
      IF (T+0.5*DT .GE. DTFILL*FLOAT(NRFILL)) CALL FILLBX
      IF (T+0.5*DT .GE. DTFRC*FLOAT(NRFRC)) CALL FRCBX
      IF (T+0.5*DT .GE. DTFLUX*FLOAT(NRFLUX)) CALL FLUXBX
      IF (T+0.5*DT .GE. DTMNTR*FLOAT(NRMNTR)) CALL MNTR
      CALL STREAM
C
      LOADQ = 0
      CALL AUTOSV
C
```

```
      CALL STEER
C
      RETURN
      END

C=====================================================================
      SUBROUTINE STOPCOMFLO
C
      CALL LIQPCT(1, PCT)
C      OPEN(UNIT=11, FILE='dthist.dat', POSITION='append')
CRAY
      OPEN(UNIT=11, FILE='dthist.dat')
 20   CONTINUE
        READ(11,'(A1)',END=30)
      GOTO 20
 30   WRITE(11,'(E12.4,I10,I4,E12.4)') T, CYCLE+1, 0, DT
      CLOSE(11)
C
      RETURN
      END
```

Note that in subroutine *output* the time loop is removed. The variable declarations in the main program of ComFlo are moved to the subroutines *initcond*, *output* and *stopcomflo*. ComFlo already contains a coupled solid-liquid dynamics subroutine. This subroutine (*tumble*) is executed after the model for the liquid dynamics has been completed (in the same time step). In *tumble* the center of mass of the liquid $\boldsymbol{r}_l$, the inertia tensor of the liquid $\boldsymbol{I}_l$ and the integrals over the liquid volume $\boldsymbol{A}_l$ and $\boldsymbol{B}_l$ are computed. These variables need to be inputs for the vehicle model in Simulink and are therefore assigned to the output vector `Y_S`:

```
      SUBROUTINE TUMBLE(MODE,Y_S)
C
C  Outputs to Simulink
C
        Y_S(1) = LQCOMX
        Y_S(2) = LQCOMY
        Y_S(3) = LQCOMZ
        Y_S(4) = LQMOIXX
        Y_S(5) = LQMOIYY
        Y_S(6) = LQMOIZZ
        Y_S(7) = LQMOIXY
        Y_S(8) = LQMOIXZ
        Y_S(9) = LQMOIYZ
```

```
C
C  Outputs to Simulink
C
            Y_S(10) = RHO * (ACCX + 2.0 * CORX - FRCX)
            Y_S(11) = RHO * (ACCY + 2.0 * CORY - FRCY)
            Y_S(12) = RHO * (ACCZ + 2.0 * CORZ - FRCZ)
            Y_S(13) = RHO * (RACCX + 2.0 * RCORX - RFRCX)
            Y_S(14) = RHO * (RACCY + 2.0 * RCORY - RFRCY)
            Y_S(15) = RHO * (RACCZ + 2.0 * RCORZ - RFRCZ)
C
```

The subscript S is included to differentiate between ComFlo and Simulink variables. Since the output vector `Y_S` is assigned a value in subroutine *tumble*, `Y_S` is added to the subroutine parameter list. In the Simulink model the motion of the vehicle is computed and assigned to the input vector `U_S` (replacing the solid body dynamics in *tumble*). In the ComFlo subroutine *tilde* these values are passed to the corresponding ComFlo variables (`U_S` is added to the subroutine parameter list):

```
      SUBROUTINE TILDE(U_S)
C
C  Inputs from Simulink
C
      OMETN(1) = OMET(1)
      OMETN(2) = OMET(2)
      OMETN(3) = OMET(3)
      QT(1) = U_S(1)
      QT(2) = U_S(2)
      QT(3) = U_S(3)
      OMET(1) = U_S(4)
      OMET(2) = U_S(5)
      OMET(3) = U_S(6)
      DQDT(1) = U_S(7)
      DQDT(2) = U_S(8)
      DQDT(3) = U_S(9)
      DOMEDT(1) = U_S(10)
      DOMEDT(2) = U_S(11)
      DOMEDT(3) = U_S(12)
      INTQ(1) = U_S(13)
      INTQ(2) = U_S(14)
      INTQ(3) = U_S(15)
      INTOME(1) = U_S(16)
      INTOME(2) = U_S(17)
      INTOME(3) = U_S(18)
C
```

Note that these ComFlo variables were previously computed in the ComFlo subroutine *tumble*. The ComFlo variable `OMETN` contains the value of `OMET` (the angular velocity $\boldsymbol{\omega}$) at the previous time step. Before setting a new `OMET`, the value of `OMET` is stored in `OMETN`. Since the variable `OMETN` is part of the common block `COSYS`, the whole block `COSYS` is declared in subroutine *tilde*.

Appendix B

Simulink Models

B.1 Free Rotation of Rectangular Plate

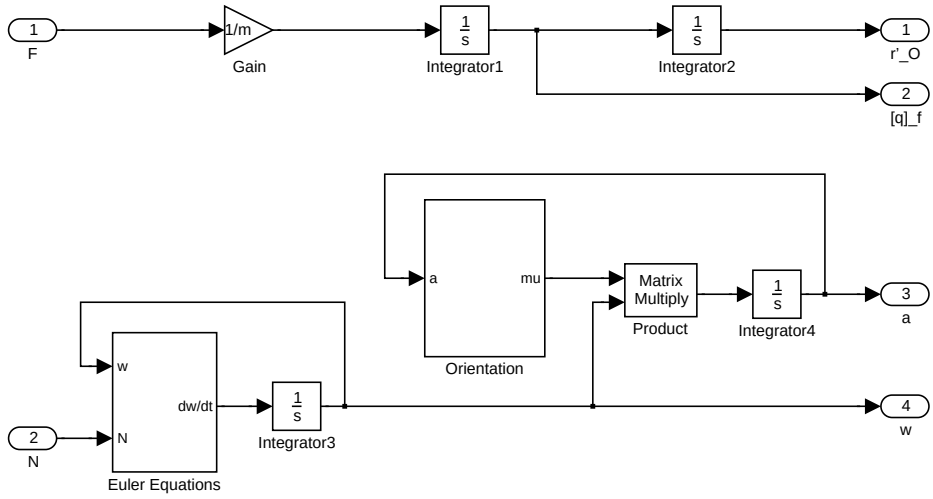


Figure B.1: *Rigid Body Dynamics*

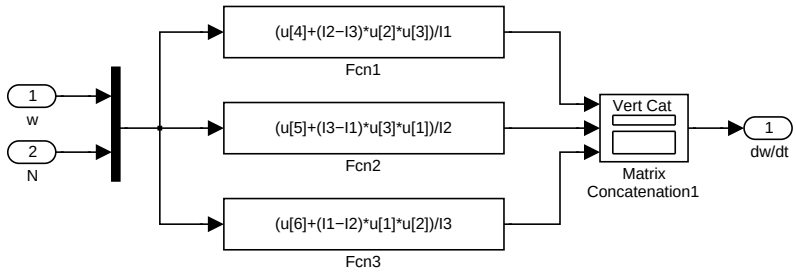


Figure B.2: *Euler Equations*

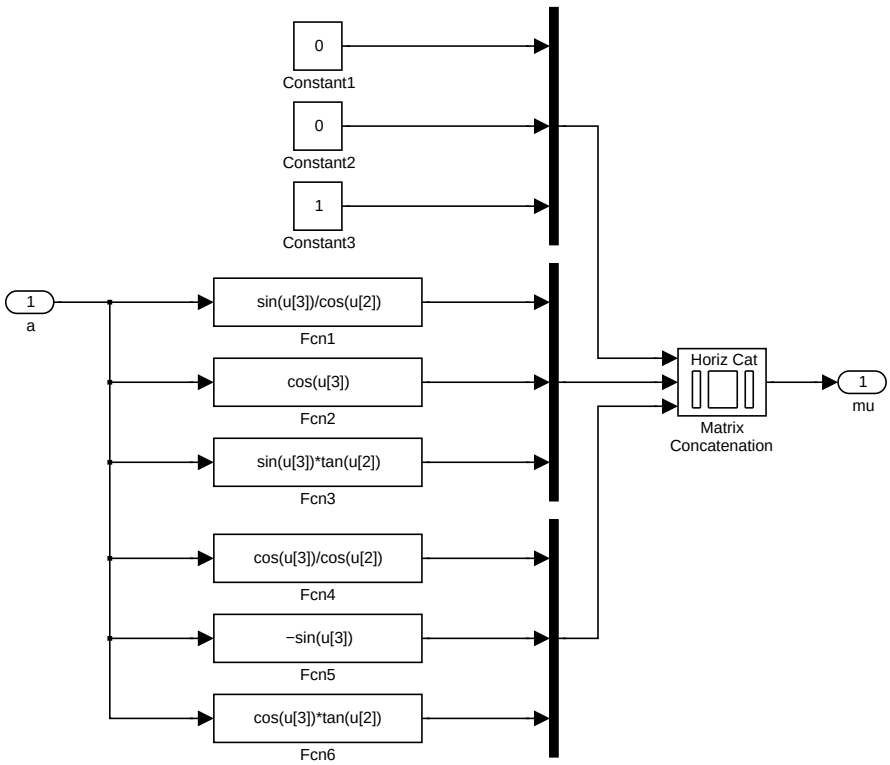


Figure B.3: *Orientation*

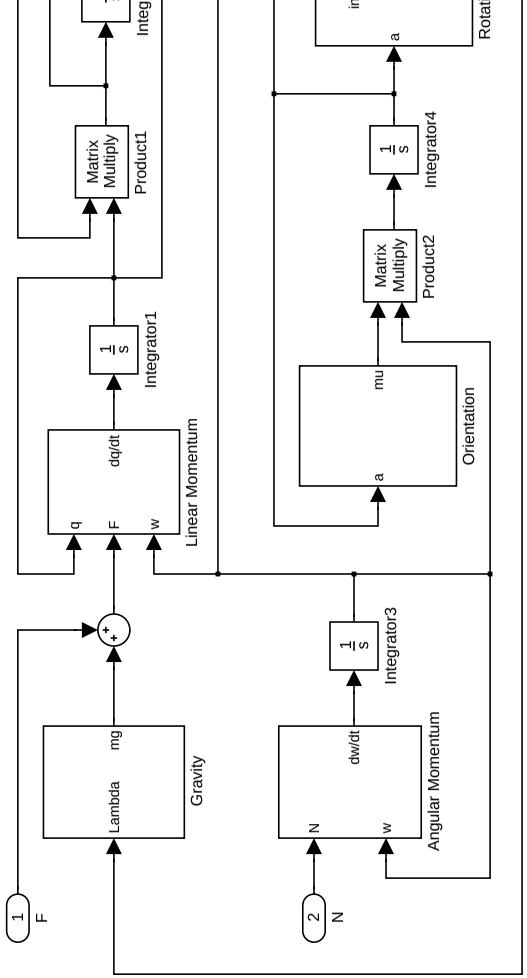


Figure B.4: *Rigid Body Dynamics*

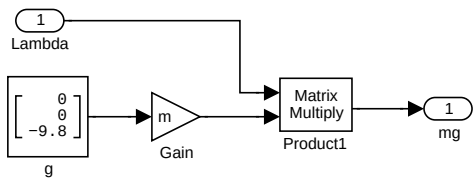


Figure B.5: *Gravity*

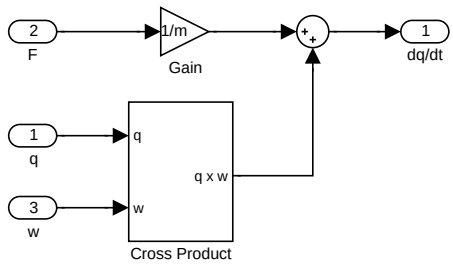


Figure B.6: *Linear Momentum*

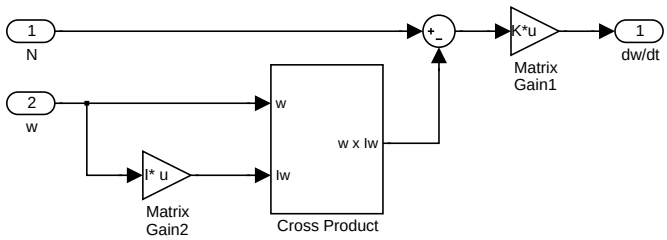


Figure B.7: *Angular Momentum*

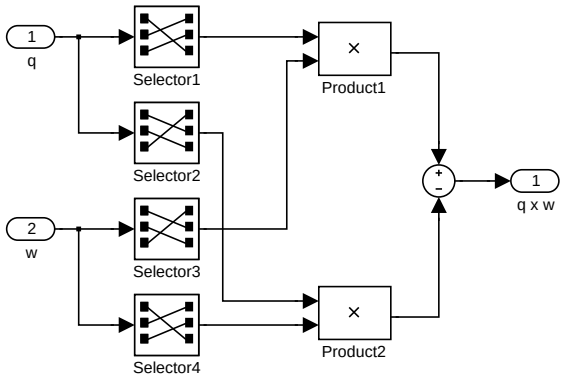


Figure B.8: *Cross Product*

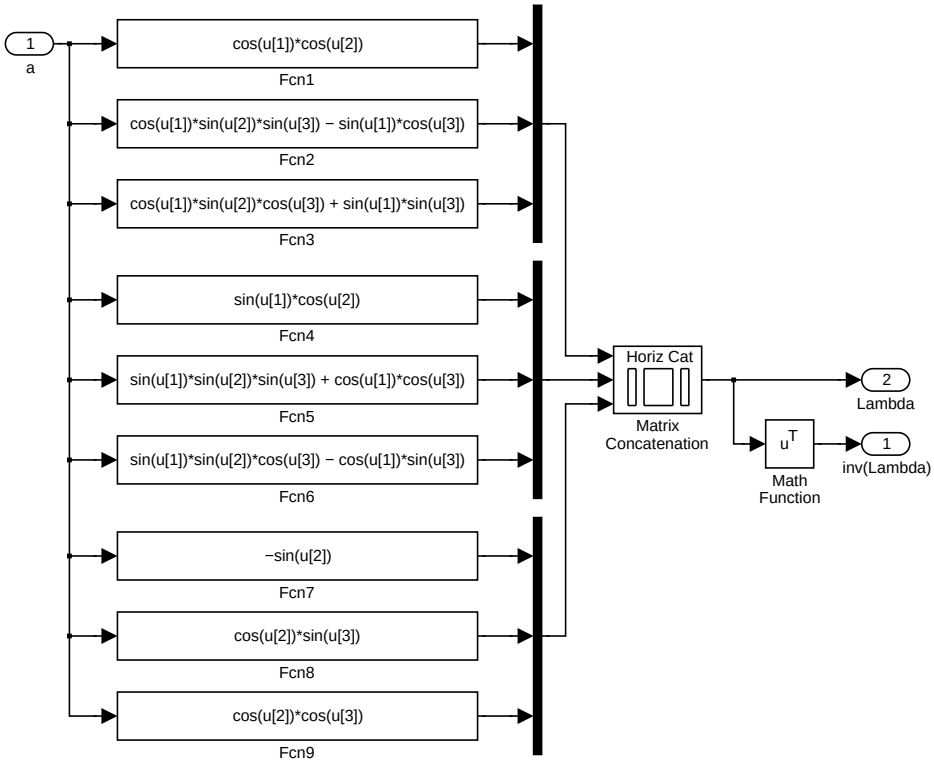


Figure B.9: *Rotation Matrix*

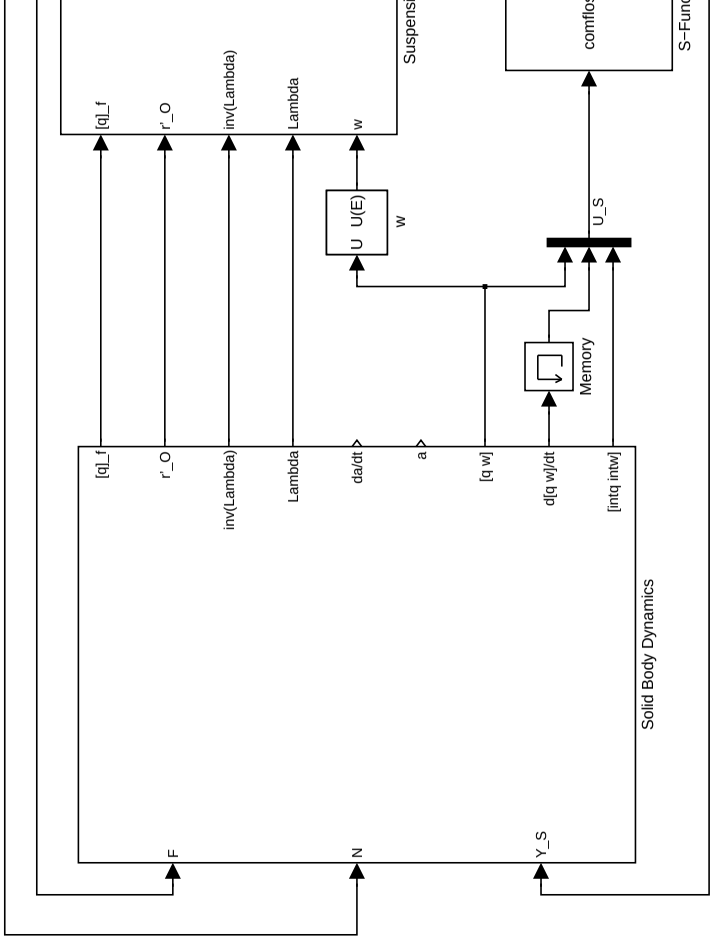


Figure B.10: *Coupled Solid-Liquid Dynamics*

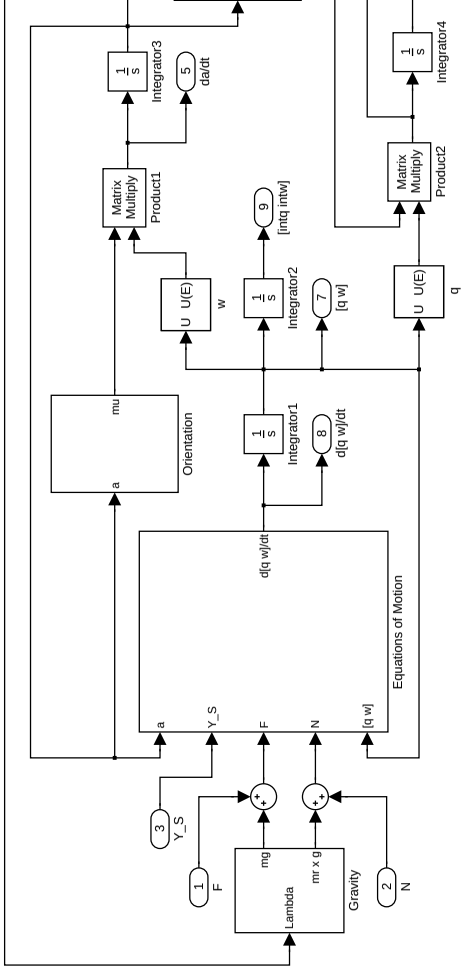


Figure B.11: *Solid Body Dynamics*

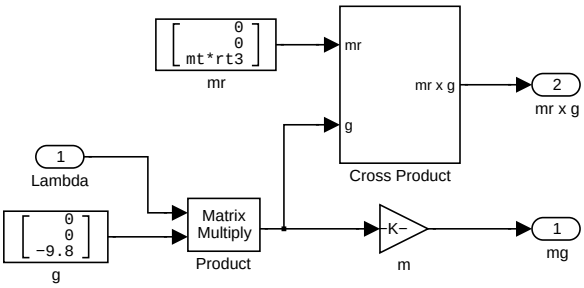


Figure B.12: *Gravity*

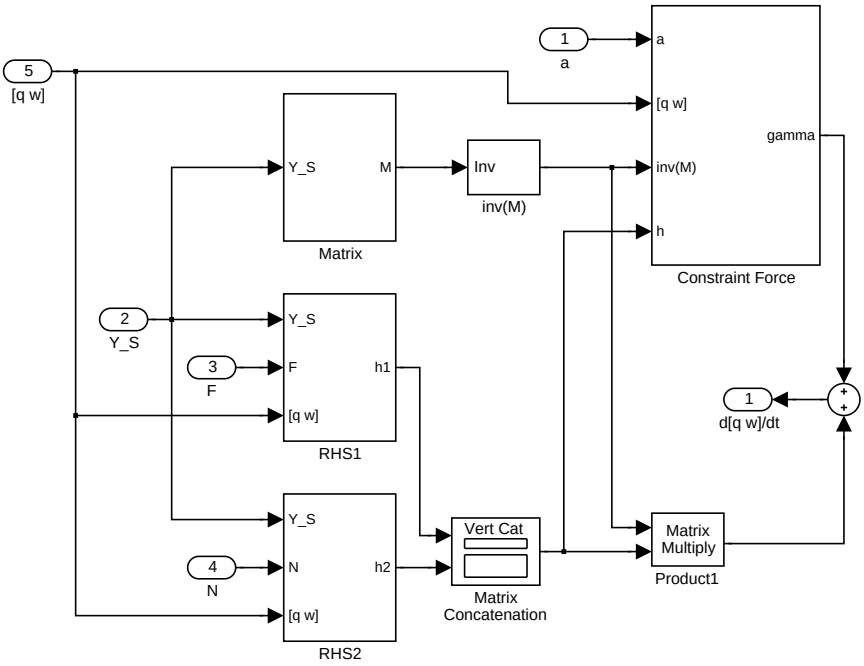


Figure B.13: *Equations of Motion*

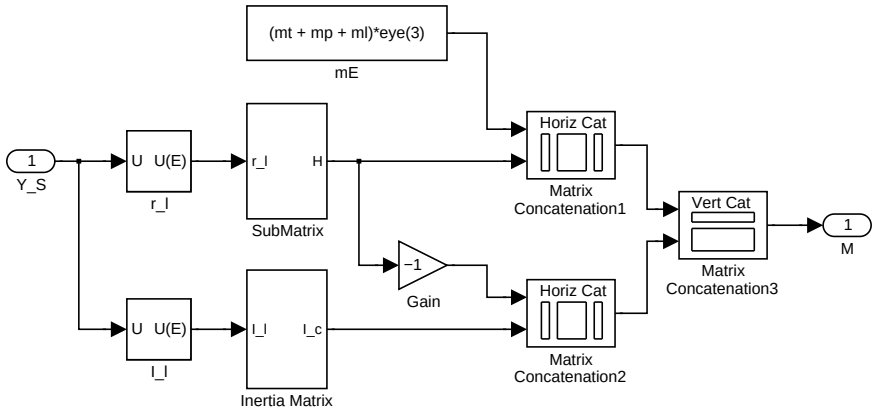


Figure B.14: *Matrix*

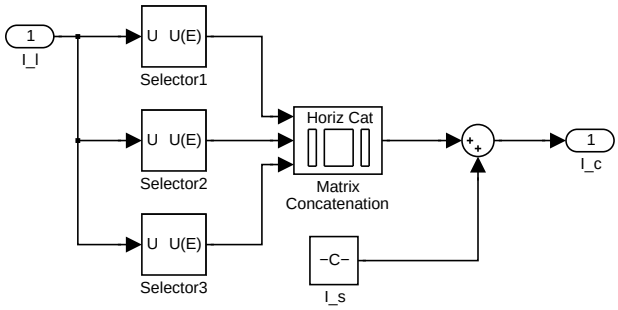


Figure B.15: *Inertia Matrix*

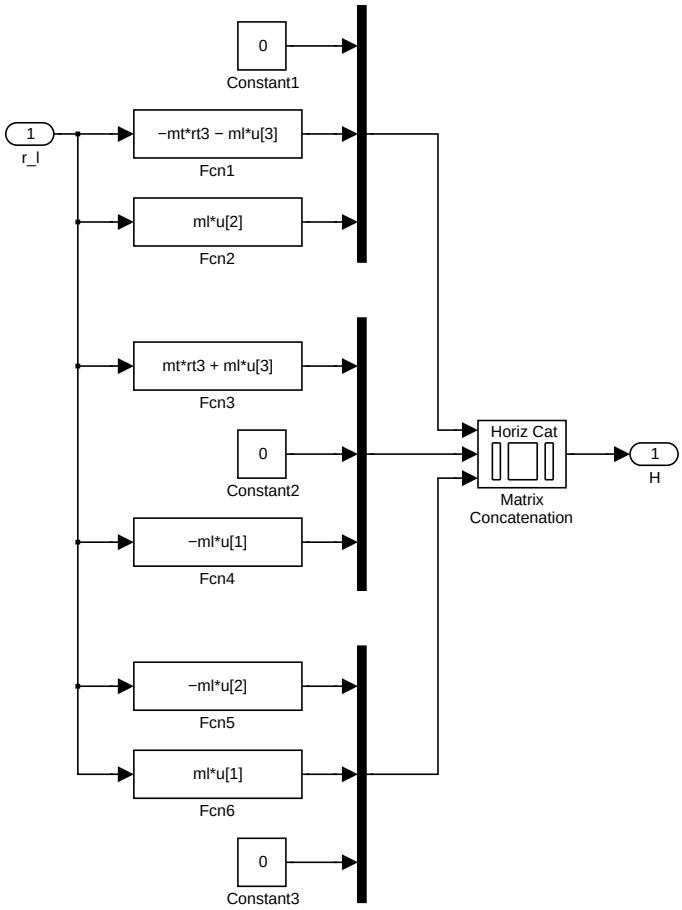


Figure B.16: *SubMatrix*

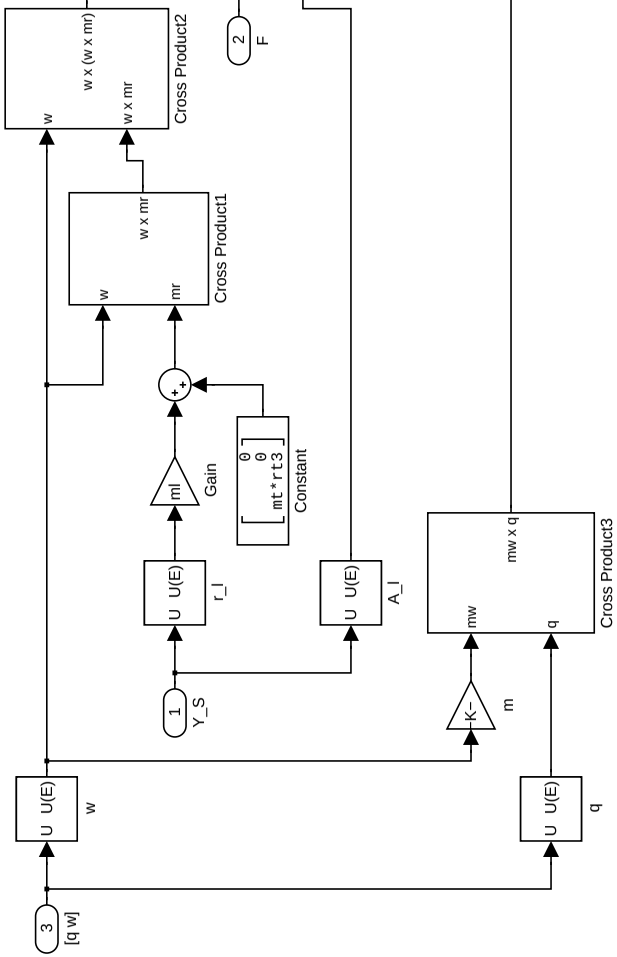


Figure B.17: *RHS1*

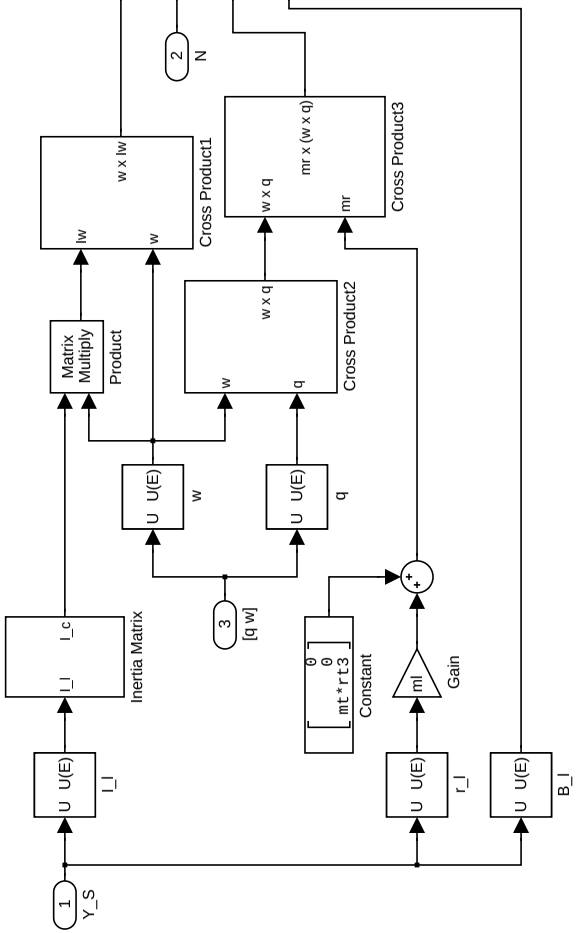


Figure B.18: *RHS2*

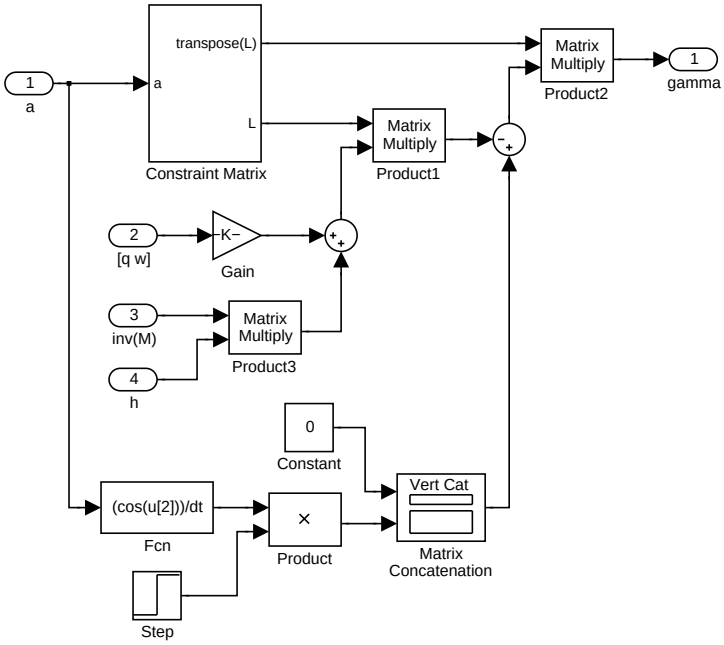


Figure B.19: *Constraint Force*

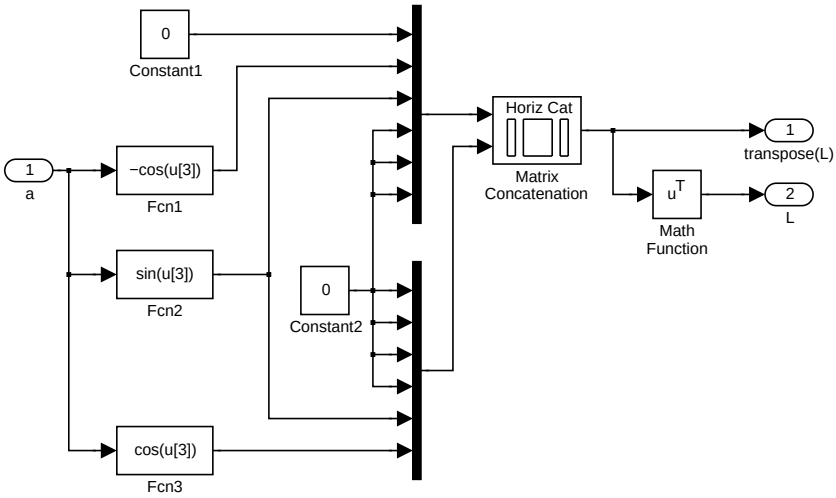


Figure B.20: *Constraint Matrix*

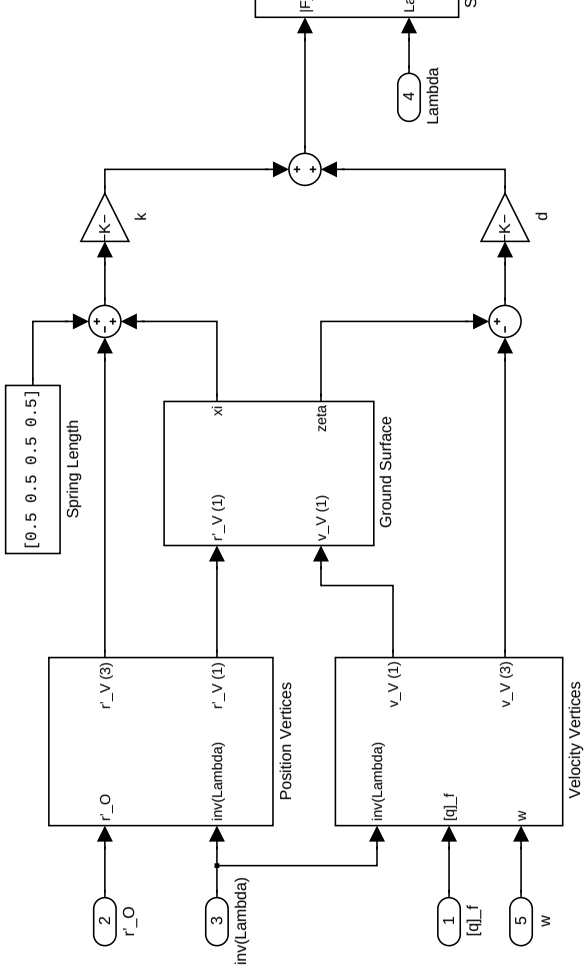


Figure B.21: *Suspension Model*

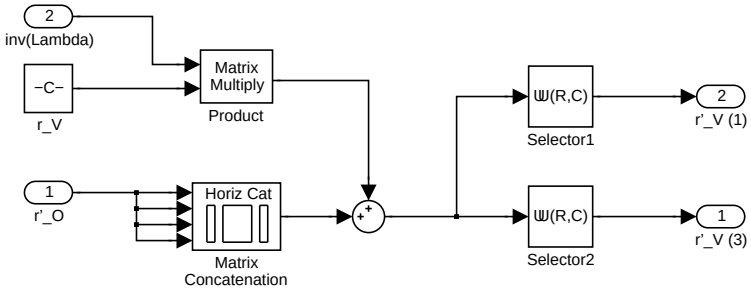


Figure B.22: *Position Vertices*

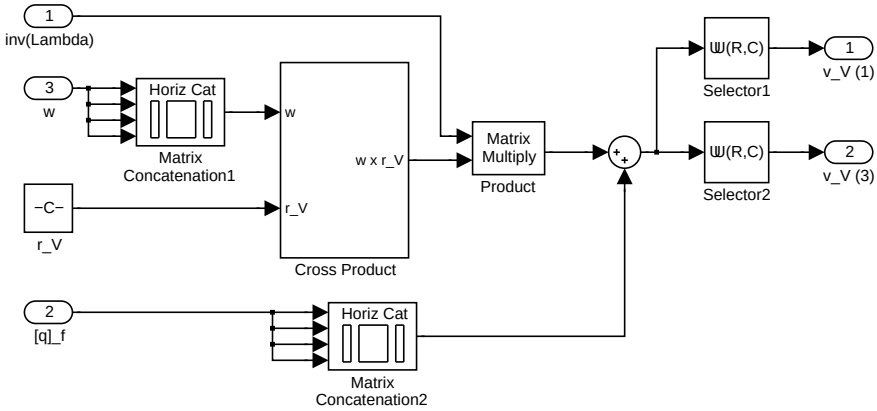


Figure B.23: *Velocity Vertices*

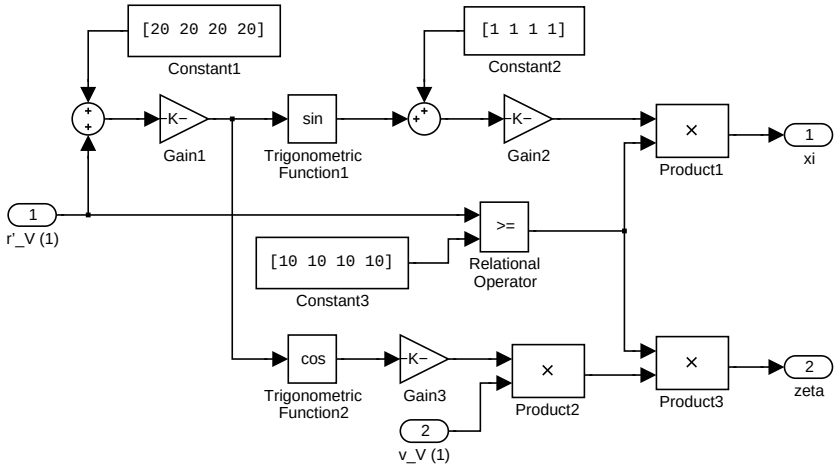


Figure B.24: *Ground Surface*

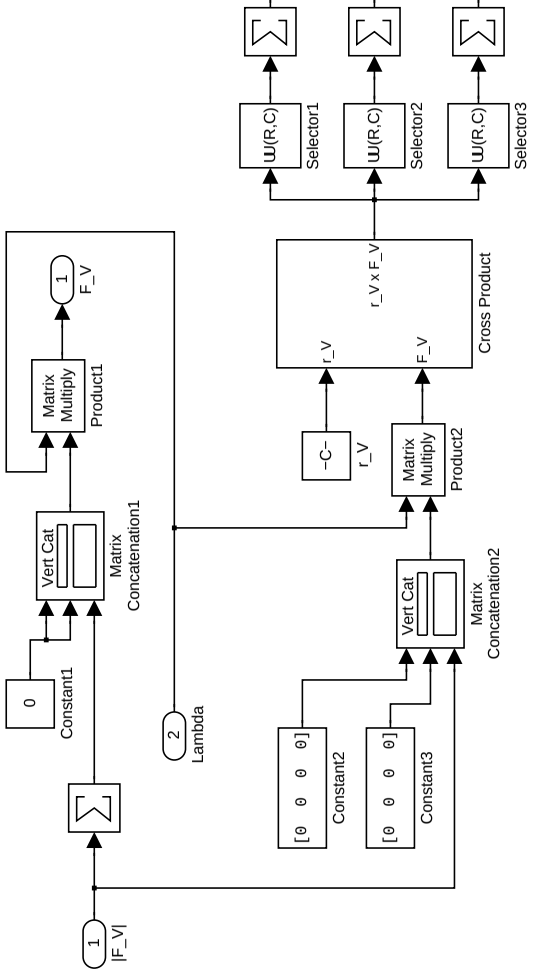


Figure B.25: *Suspension Force*

Bibliography

[1] A.A. ten Dam. *Stable numerical integration of dynamical systems subject to equality state-space constraints.* Journal of Engineering Mathematics **26**: 315-337, 1992.

[2] J.Gerrits. *Dynamics of Liquid-Filled Spacecraft, Numerical Simulation of Coupled Solid-Liquid Dynamics.* PhD thesis, Rijksuniversiteit Groningen, 2001.

[3] A. Kremer. *Coupling of computer models for aircraft cabin and temperature controller.* Master's thesis, National Aerospace Laboratory, 2002.

[4] J.B. Marion and S.T. Thornton. *Classical dynamics of particles and systems.* Harcourt, 1995.

[5] A.R. Paterson. *A first course in fluid dynamics.* Cambridge University Press, 1983.

[6] W.T. Thomson. *Introduction to Space Dynamics.* J. Wiley & Sons Inc., New York, 1961.

[7] A.E.P. Veldman. *Computational Fluid Dynamics.* Lecture notes, Rijksuniversiteit Groningen, 2001.

[8] A.E.P. Veldman and M.E.S. Vogels. *Axisymmetric Liquid Sloshing under Low-Gravity Conditions.* Acta Astronautica Vol. 11, No. 10/11, pp. 641-649, 1984.