Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-2005-703

Real-time Simulation of Hybrid Aerospace Systems

A.A. ten Dam, W.F. Lammen and G. Rozema*

* University of Groningen

This report has been based on a paper presented at ENOC-2005, Eindhoven, 7 – 12 August, 2005.

Customer:	National Aerospace Laboratory NLR
Working Plan number:	AV.1.C2.2
Owner:	National Aerospace Laboratory NLR
Division:	Aerospace Vehicles
Distribution:	Unlimited
Classification title:	Unclassified
	February 2006

Approved by:

Author	Reviewer	Managing department
	27-2-'06	3/3'06

Summary

Aerospace vehicle systems are by nature, complex systems. The interaction between systems and their environment, and between systems with very different dynamical behavior adds to the complexity. In this paper we give an overview of selected aerospace applications in which modeling, simulation and control play an essential role.

Simulation studies require suitable mathematical models of all components of the system under consideration. When a simulator is also used for verification and validation, and/or training purposes, it is advantageous to take real-time aspects into account right from the start. As such real-time simulators require special computational algorithms. This paper will focus on the real-time aspects, i.e. computational methods, of selected non-smooth aerospace systems.

Contents

(19 pages in total)

1 Introduction

Aerospace vehicle systems are by nature, complex systems. The interaction between systems and their environment, and between systems with very different dynamical behavior adds to the complexity. During design and test of aerospace systems, software representations of (sub) systems are often replaced by hardware components. Moreover, training simulators, with humans-in-the-loop pose extra requirements on timely responses. In particular, embedded training, the amalgamation of simulation and training becomes more and more important.

This paper gives an overview of selected aerospace applications in which modeling, simulation and control play an essential role. Industrial designs of systems are usually driven by operational requirements. Based on these requirements, critical items are identified and feasibility studies are executed, supported by real-time simulation studies. These simulation studies require suitable mathematical models of all components of the system under consideration. We aim at generic models suitable for use in engineering as well as training simulators. As a consequence, proper as well as timely responses of the models are crucial. Therefore, the mathematical models for aerospace systems must behave very much like the systems would do in the real world. When a simulator is also used for verification and validation, and/or training purposes, it is advantageous to take real-time aspects into account right from the start. As such real-time simulators require special computational algorithms.

In the last decade, motivated by acquired capabilities of design and training in the field of vehicle dynamics and robotics, generic multi-body models of a wide range of systems were made. The acquired mathematical models are realistic representations of mechanical systems, comprising for example, elasticity and damping of the tire, wheel suspension, chassis stiffness characteristics, and roll, pitch and yaw motions. Of special interest is the interaction of a vehicle with rugged terrain: the impact phenomenon gives rise to discrete-event modeling in combination with continuous-time dynamics. In this case, building a model from sub-models has the advantage that model adaptations are easy, which is important in aerospace design studies.

Satellite systems offer another challenging field of research. The behavior of cooling and fuel liquid can have a considerable impact on performance and controller demands. The interaction of liquid with the rest of the satellite, or, here on earth, the interaction between a liquid cargo and the truck that is carrying the cargo, requires an extension of impact simulation between mechanical systems as done with robotic systems, to systems of a very different physical character.

The remainder of this paper is organized as follows. In section 2 we introduce some aerospace systems, and look for a common (mathematical) denominator that can be used to build generic models. For this we first focus on rigid-body systems. Next, in section 3 we briefly discuss numerical techniques that have been applied successfully in real-time simulation studies of aerospace systems. In sections 4 and 5 emphasis is on applications. Section 4 focuses on ongoing research in modeling and simulation of vehicles. In February 2005 the Dutch mini-satellite Sloshsat FLEVO was launched. In section 5 liquid-slosh and its interaction with a satellite or a vehicle on rugged terrain is described. Conclusions can be found in section 6.

2 Aerospace Systems: Examples

From a mathematical point of view, interactions of mechanical systems with their environment fall into the class non-smooth dynamical systems. At NLR there are many applications that deal with non-smooth dynamical systems. Examples are:

- robotic manipulators: simulation and control of constrained motions of the European Robotic Arm ERA;
- wind tunnel experiments: contact between an aircraft model with the walls of the test section in a wind tunnel must be avoided at all times to prevent damage;
- wheeled vehicles: simulation and control of a wheeled planetary rover under several terrain conditions;
- aircraft landing gear: structural design and simulation of wheel/runway impact and contact;
- satellites: investigations in modeling and control of the effects of liquid slosh on the behavior of satellites like Sloshsat FLEVO.

The design and fabrication of aerospace systems is by no means a trivial task. The projects usually take a number of years, technology ages during the project, new scientific discoveries must be taken into account, and last but not least, a system must remain operational for (at least) the time it has been designed for.

New aerospace systems are usually high-tech, and novel scientific demands require novel approaches to system design [Kos et.al, 1999]. Nevertheless, the list below, adapted from [Brogliato, 1999], gives a good idea of the difficulties that are encountered in many cases of aerospace system design.

- Determine wellposedness of impact dynamical systems, i.e. properties of solutions. This is important to obtain a unified treatment of contact problems across boundaries of application domains, and for enabling real-time simulation of constrained multi-body mechanical systems. Part of NLR's expertise is presented in [Ten Dam et. al., 1997].

- Derive impact rules between two rigid bodies via macroscopic laws that relate the motion after and before the shocks. A particular solution is described in [Ten Dam and Willems, 1997].
- Develop new models of contact-impact laws for specific applications using engineering competence available in an organization. An example is the derivation of impact laws for liquid-slosh in a cavity [Sloshsat, 2005].
- Establish impact control methods of flexible systems like ERA on the International Space Station.
- Find adequate numerical algorithms to integrate systems subject to unilateral constraints. Simulation of non-smooth mechanical systems is known to be liable to numerical instabilities. A procedure for real-time simulation has been developed at NLR, proved to be widely applicable, and is described briefly in section 3.
- Design control strategies to improve the behavior of hybrid systems subject to repeated impacts with the environment. Control of constrained systems and liquid slosh is part of ongoing research at NLR. Control of liquid slosh has been one the main reasons for the launch of Sloshsat FLEVO.

Impact research at NLR took a major step forwards when design and control issues arose for the European Robotic Arm ERA. Space-borne manipulators like ERA are typically used for moving payloads. Whenever a space-borne manipulator comes in contact with its environment, e.g. a satellite or the international space station, it may not damage itself, the space station, or the payload. This makes the design of for example the control system for the robotic manipulator critical.

For over a two decades NLR, together with Dutch Space, has worked on modeling, control and simulation of a robotic manipulator that interacts with its environment. In the early nineties, a Matlab based robot simulation environment (see figure 1) called TRaCE was originally developed

- to investigate and solve the numerical problems of a robotic manipulator that is in or comes into contact with its environment, and
- to evaluate control strategies for automatic control of the manipulator under such conditions.

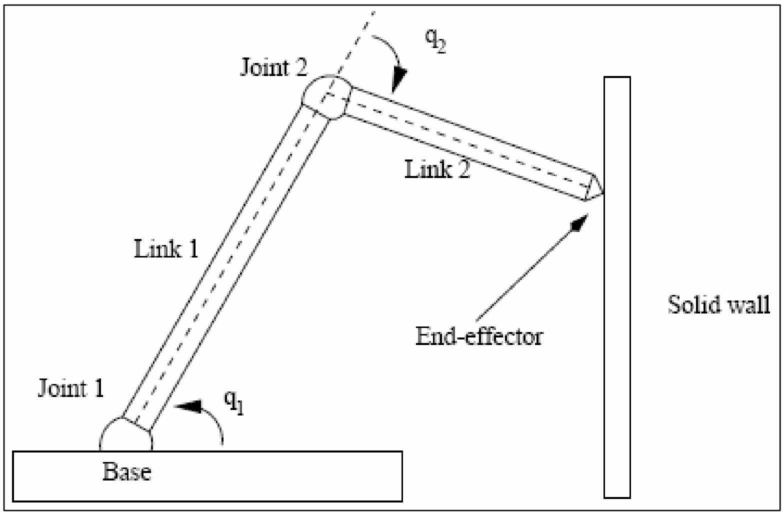

Figure 1 Simplification of hybrid space-borne robotic manipulator.

NLR employees used TRaCE to execute feasibility studies on control engineering topics that were identified both at NLR and in industry, whereas university students examined novel control paradigms on their merits via simulation studies.
Knowledge that has been gathered through TRaCE has been explored in the contributions of NLR to the development of the ERA Simulation Facility (ESF) for real-time simulation of the European Robotic Arm (ERA) on the international space station. The robotic manipulator has been simulated throughout the various motion phases and events:

- pre-impact phase, also called the free motion phase;
- impact phase, the phase in which the effect of making contact are notable in the manipulator and/or the other object;
- post-impact phase, the phase in which the robot remains in contact with another object;
- release phase, the phase in which a controlled release is made.

These phases are by now commonly recognized, but at the start of our research, literature on the subject was far less extensive than it is today and industrial applications where few, at least those available in the open literature.

Based on the above characterization of research issues and decomposition of impact phases, it is evident that for successful controller design, some basic mathematical issues need to be solved.

3 Real-time Simulation of Impact

Numerical simulation of impact (phases) require verified and validated simulation models of aerospace systems. In the case of ERA, many man- years have been put in the construction of such a validated model of a flexible space-borne robotic manipulator. Although the model was originally designed for free-motion phase only, it was clear that the model must be re-used for the study of impact and movement of payloads. The sheer costs of developing a new constrained motion model from the start were simply to high to be economically affordable.

This led us to search for approaches that would yield dynamical models of hybrid systems that are relatively easy to modify, and easy to extend. Moreover, the target was (in still is) to find approaches that can also be used across project boundaries and across applications. The numerical approach described below does just that. Originally developed for rigid systems, it proved pivotal in the study for the flexible robot-arm ERA, and is now being used in the further study of liquid-slosh. Details of this approach can be found in [Ten Dam, 1992] and [Brogliato et al, 2002].

Lagrange multiplier formulations of constrained mechanical systems can be obtained via so-called first-principles modeling. It is of interest to note that for constrained mechanical systems, Lagrange formulations can be derived that are particular useful for real-time simulation. Let x denote the generalized co-ordinates used to describe the ‘state’ of the system under consideration. For simplicity assume that the mechanical systems can be represented by

$$M(x)\ddot{x} + N(x,\dot{x}) = u. \qquad (1)$$

Here *M(.)* denotes the generalized mass matrix, *N(.,.)* denotes is a vector function that characterizes the Coriolis, centrifugal and gravitational load, and u denotes generalized inputs. Equation (1) can be obtained from a model as in figure 1 by using the forward kinematics relation $x=H(q)$ for a system dependent H. For simplicity, assume that there is a single constraint manifold, modeled by

$$h(x) = 0. \qquad (2)$$

Usually, equation (2) is obtained by modeling the environment in the area of interest. The region in which the end-effector is allowed to move can now be give as

$$h(x) \geq 0. \qquad (3)$$

Using equation (1) and (2), a constrained motion model can be given as

$$M(x)\ddot{x} + N(x,\dot{x}) = u + G^T(x)\lambda, \qquad (4)$$

where λ is the Lagrange multiplier, and *G(.)* represents the contact force matrix.

An additional advantage of the use of a Lagrange multiplier is that during simulation studies, an expression of a Lagrange multiplier can be used as a model for a force sensor or simply as a nonlinear expression for the contact force. The combination of first principles modeling and the relevance of the Lagrange multiplier for controller design has also been a motivation to continue research in this direction.

The introduction of the Lagrange multiplier makes detection of controlled contact and controlled release during simulation studies nontrivial. The following rules apply in case contact has already been established.

contact when

$$\left.\begin{array}{l} h(x) = 0 \ and \\ \dfrac{\partial h}{\partial x}(x)\dfrac{dx}{dt} = 0 \end{array}\right\} \qquad (5)$$

release when

$$\lambda < 0. \qquad (6)$$

For numerical simulation, a discrete version of the Lagrange multiplier is used. The actual derivation is beyond the scope of this paper, and we refer to [Ten Dam, 1997]. For the purpose of the present paper it suffices to mention that the discrete Lagrange multiplier differs from the continuous-time Lagrange multiplier only in a term that contains the expression

$$\frac{h(x_n)}{\Delta t}. \qquad (7)$$

Here x_n denotes the solution at time t_n and Δt denotes the (current) time-step used in the numerical solver.

One of the remaining difficulties for successful simulation is the discretization of the contact and release strategy given in equations (5) and (6), and the determination of the (simulated) discrete contact points and/or sets. Recall that our prime concern is real-time simulation, i.e. in case of hardware or humans in the loop, the numerical simulation must satisfy stringent timing requirements. To keep on the safe side, the following rules are applied a priori:

- step-back (in time) is not allowed,
- algorithms that use iterations are forbidden.

Since calculations inevitably involve the dynamics model there may simply not be enough (simulation) time for iterative computations. The solution to these difficulties is threefold. First, all real-time calculations must be as explicit as possible, i.e. it must be possible to provide a reliable estimate of execution times in case of impact. Second, it must be known a priori how to transfer the (simulated) state of the system when collision takes place (i.e. how a system will rebound at the point of contact). The solution we follow has been outlined in [Ten Dam et. al., 1997]. And thirdly, the methods and calculations are validated by more time-consuming numerical algorithms, and/or by experiments.

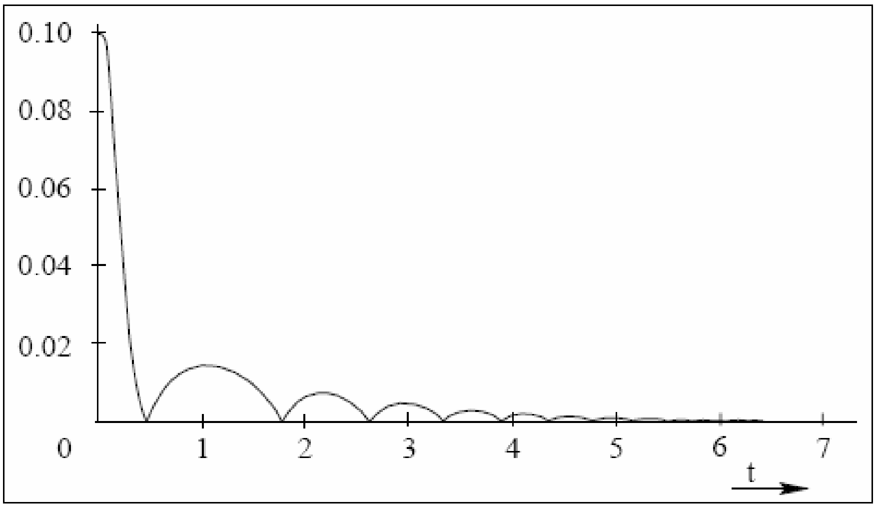

Figure 2 Simulated contact and rebound of a controlled collision of an end-effector with a rigid surface.

The result of all these carefully orchestrated international design and research exercises to answer a particular research question, can in the end be just a two-dimensional picture as shown in figure 2. Results like these can then be used to fine-tune controller design, or to further investigate and adjust material (properties).

4 Modeling Vehicles as Interaction of Parts

A case that is being investigated in detail at NLR is a generic behavior model of a vehicle moving over a rugged terrain. Possible applications of such dynamic behavior models are (real-time simulations of) trucks, personal cars, landing and taxiing aircraft, helicopters and planetary vehicles. Special attention is directed to the issues of proper representation of contact with the terrain underneath and a reliable numerical method to solve the associated equations of motion.

Following the idea of a multi-body system, the vehicle is split up into partial vehicles. A two- or four-wheeled vehicle is thought to consist of a corresponding number of partial vehicles (see figure 3). Each partial vehicle consists of a tire, a rim and a part of the chassis. These parts are modeled by point masses and in order to model the shock absorption, they are connected by springs and dampers. Applying appropriate geometrical constraints it is easy to connect these sub-models. In this way, a complex model based upon simple building-blocks results.

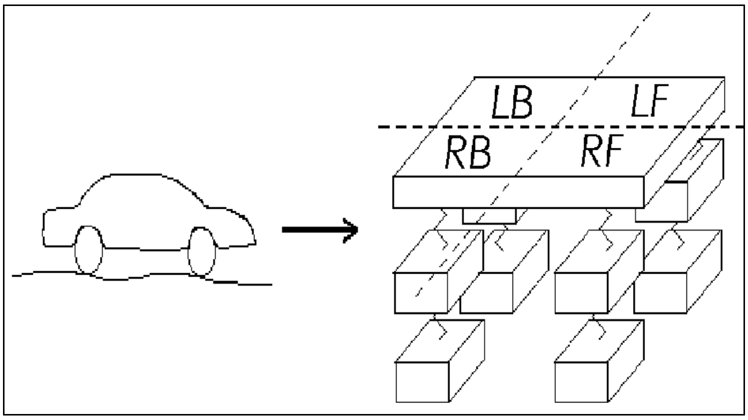

Figure 3 Vehicle that is build from parts.

Another advantage of the modular approach is that adding more partial vehicles only requires extra algebraic equations instead of rewriting the ordinary differential equation (ODE) completely. In fact, because of the use of point-masses a linear differential algebraic equation (DAE) system is derived, based on Newton’s law for forces for translational motion, instead of an ODE that includes the non-linear moment equation of the rotational motion of the chassis body of the vehicle.

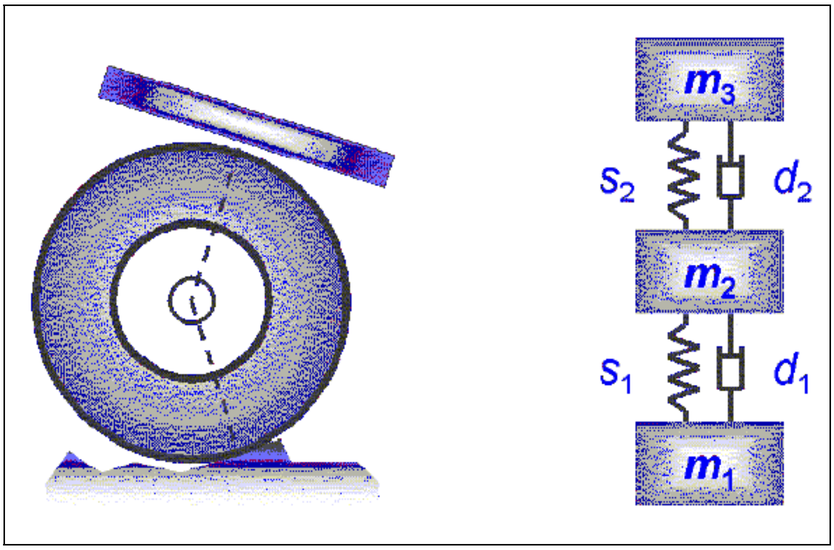

Figure 4 Two-dimensional partial vehicle representation.

The equations that describe the movement of one partial vehicle (see figure 4) are based on a layered mass-spring-damper system and are represented below.

Denote $d_{ij} := d(m_i, m_j)$, the absolute distance between point mass i and j in meters.

$$\left.\begin{aligned}
m_1\ddot{x}_1 &= k_1(d_{12} - z_{rel1})\underline{s}_1 + c_1\frac{\partial}{\partial t}d_{12}\underline{s}_1 + \underline{F}_{ex_1} + m_1 g \\
m_2\ddot{x}_2 &= -k_1(d_{12} - z_{rel1})\underline{s}_1 - c_1\frac{\partial}{\partial t}d_{12}\underline{s}_1 \\
&\quad + c_2\frac{\partial}{\partial t}d_{23}\underline{s}_2 + k_2(d_{23} - z_{rel2})\underline{s}_2 + \underline{F}_{ex_2} + m_2 g \\
m_3\ddot{x}_3 &= -k_2(d_{23} - z_{rel2})\underline{s}_2 - c_2\frac{\partial}{\partial t}d_{23}\underline{s}_2 + \underline{F}_{ex_3} + m_3 g
\end{aligned}\right\} \qquad (8)$$

Here

g : gravitational constant [m/s^2];

kj : spring constant of spring j=1,2 [N/m],

cj : damping constant of spring j=1,2 [kg/s],

z_{rel_j} : relaxed spring length of spring j=1,2 [m],

mi : the mass of point mass i =1,2,3 [kg],

x_i = (xj ,yj , zi)T : position of mass i =1,2,3 [m],

$\underline{F}_{ex_i}$: external force on mass i =1,2,3 [N].

The vectors s1 and s2 represent the directions of the lower and the upper spring. (See figure 5 for a three dimensional representation.)

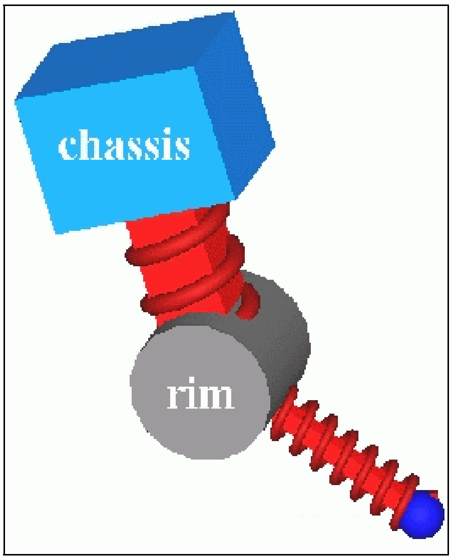

Figure 5 Three-dimensional partial vehicle representation.

To connect two partial vehicles into a two-wheeled vehicle, e.g. a motorcycle, a number of geometrical constraints must be posed. First, the absolute distance between the upper masses of partial vehicles must be equal to a certain value L to establish a chassis of the two-wheeled vehicle. Second, the direction of the suspension springs should be perpendicular to the direction of the chassis. The following equality constraints are derived:

$$\begin{aligned} &d\left(m_{13}, m_{23}\right) - L = 0 \\ &L^2 + d\left(m_{12}, m_{13}\right)^2 - d\left(m_{12}, m_{23}\right)^2 = 0 \\ &L^2 + d\left(m_{22}, m_{23}\right)^2 - d\left(m_{13}, m_{22}\right)^2 = 0 \end{aligned} \qquad (9)$$

Here mij denotes mass j of partial vehicle i. Additional constraints can be added, e.g., covering wheel-ground contact. Extension of the model to an N-wheeled vehicle is then straightforward ([Van der Raadt, 2004], [Klaasse, 2005]).

The behavior model of the vehicle is solved in a numerically stable way with an explicit fixed-step method (Runge-Kutta) and a discrete Lagrange multiplier, see section 3 ([Schram, 2004]). This method is therefore appropriate for real-time simulation models that can be used in training simulators.

Matlab/Simulink simulations have been derived for 2-, 3-, 4- and 6-wheeled vehicles. In the figures below stills of animations are shown of a 4-wheeled vehicle and a 6-wheeled vehicle.

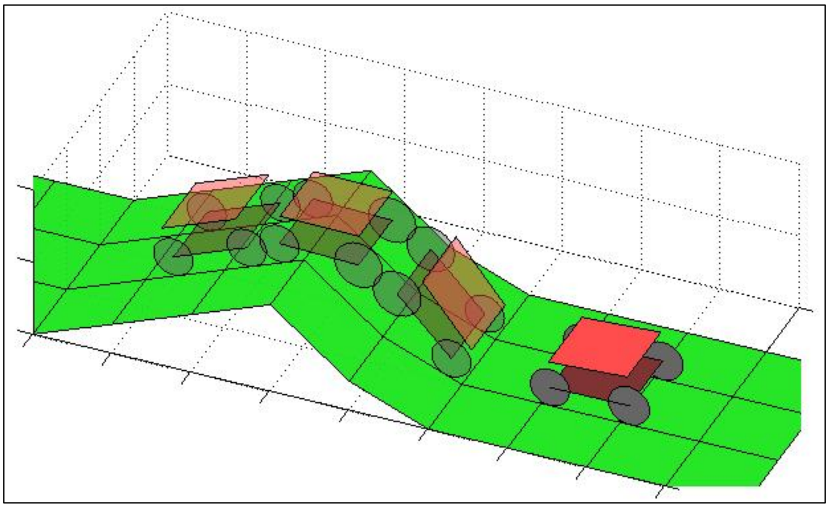

Figure 6 Animation layout of a Matlab-based simulation of a 4-wheeled vehicle on a rugged terrain.

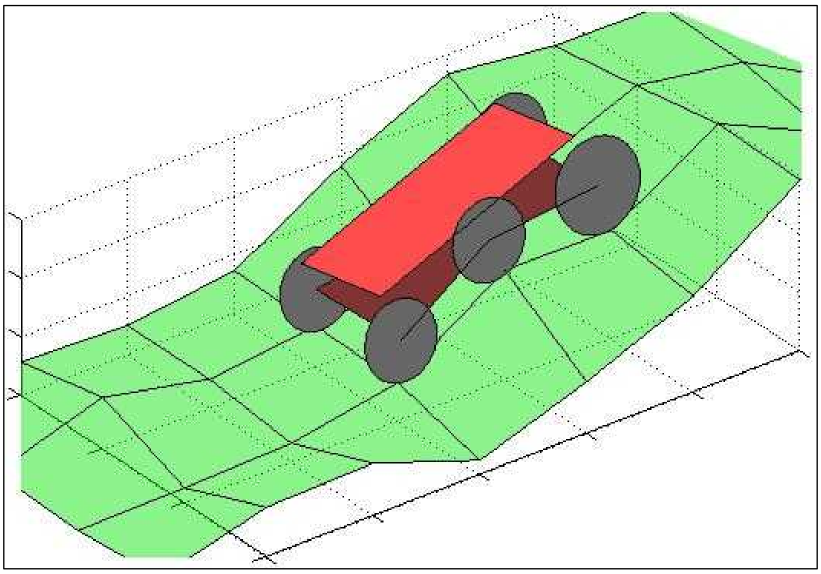

Figure 7 Matlab-based simulation of a 6-wheeled vehicle on a rugged terrain.

The Matlab/Simulink model of the vehicle can be automatically exported to a real-time simulation environment, e.g. with hardware and/or human-in-the-loop, using NLR's automatic model transfer tool MOSAIC [Lammen et. al., 2004].

5 Liquid-Slosh in a Hybrid Context

On February 12, 2005, the Dutch mini-satellite Sloshsat FLEVO went into orbit around the earth. It carried a large liquid tank designed to study sloshing behavior. Sloshsat FLEVO (Facility for Liquid Experimentation and Validation in Orbit) [Sloshsat, 2005] contained an 87-litre experimental plastic tank that transported 33.5 liters of water into space. All this water made the originally lightweight spacecraft - with a dry weight of 95 kilos - highly susceptible to the effects of liquid movement. By carrying out maneuvers with Sloshsat, researchers gained some fundamental understanding of the behavior of liquids in space and their interaction with the spacecraft.

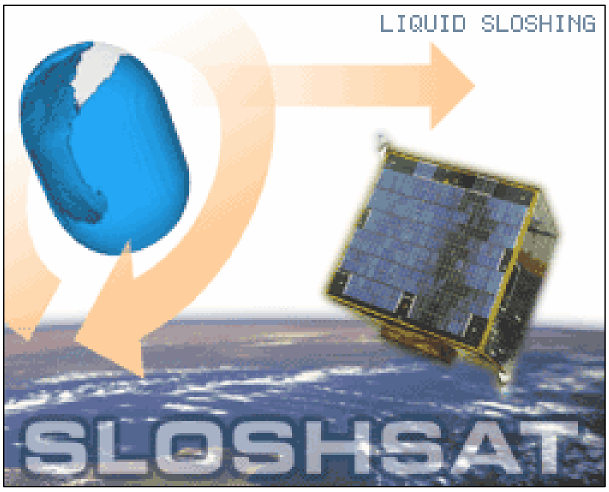

Figure 8 Sloshsat FLEVO: a small Dutch satellite to investigate how liquid sloshes around in the tanks in weightless conditions.

This kind of understanding is important when a satellite changes directions with partially filled fuel tanks. The best maneuvers will create a minimum of sloshing, vibration, and disturbance and result in minimum fuel consumption for the guidance rockets.

By taking advantage of the liquid motions, a spacecraft can save even more fuel, thereby extending the spacecraft's life (see figure 8). Understanding the movement of liquids is also useful for applications here on earth, for instance, in ships, and tank trucks.

Compared to figure 2, where single-point of contact has been modeled, figure 9 shows a discretization of infinitely many points of contact. Real-time simulations of Sloshsats control algorithms have been carried out with EuroSim, the Dutch real-time simulation engine [Lammen et.al., 2004].

Sloshsat will validate the University of Groningen's existing fluid-dynamic calculation program, ComFlo. This will make the first reliable Computational Fluid Dynamic (CFD) predictions possible for maneuvers in space. ComFlo is also used at NLR to study the interaction of a truck carrying liquid cargo. In a sense the latter is a more difficult problem from a hybrid systems point-of-view.

In case of a satellite, the cavity contains and interacts with the liquid only, whereas in the case of a truck, the motions of the truck also interact with the tank. Initial investigations have been reported in [Rozema, 2004].

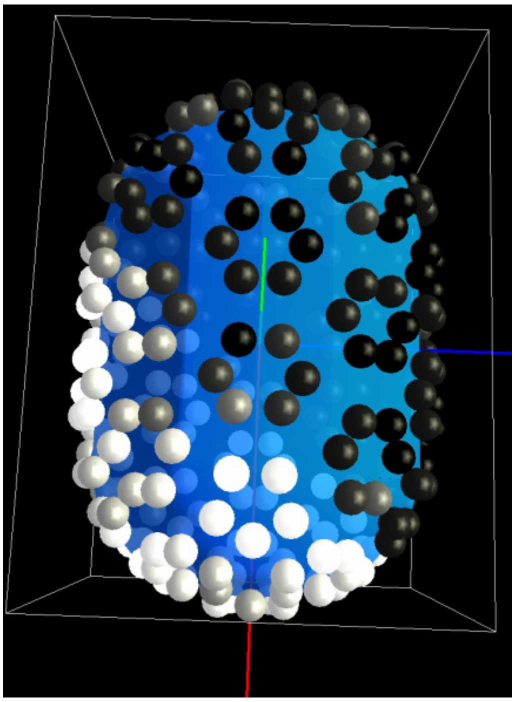

Figure 9 Visualisation of a real-world experiment of liquid slosh in a cavity. White balls indicate areas where liquid is present, black balls indicate areas where there is no liquid. Shades of grey are used to indicate areas where some liquid is present.

In order to apply our general theory to trucks that carry liquid cargo the following ingredients are needed:

- a dynamics model of a truck, similar in set up as the model in equation (1) and derived in a modular way as described in section 4,
- a dynamics model of the motion of the fluid, and
- a constraint equation that is used to relate the position and motion of the tank relative to the position and motion of the truck.

The dynamics of the liquid is described by a Navier-Stokes equation. This model constitutes a partial differential equation (PDE). And it is at this point that we can take full advantage of our approach to hybrid systems. One can keep the original model of the truck and the original model of the liquid motion – which are of a very different mathematical nature – and combine them by means of a Lagrange multiplier approach. An additional benefit is that we can use the discrete Lagrange multiplier to obtain numerically stable simulations.

The motion of the liquid can affect the ride behavior and stability of vehicles carrying liquid cargo. 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 due to the road surface, which in turn can have substantial effects on the motion of the vehicle.

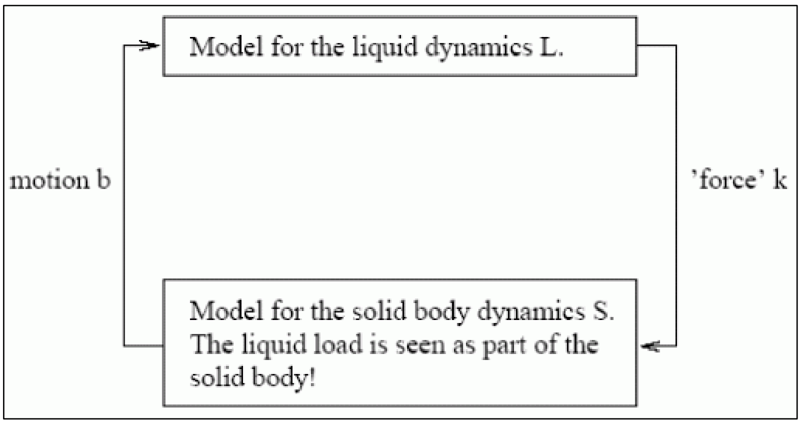

Figure 10 Information flow of liquid/solid-body simulation.

The choice of an execution order to cut the loop must be based on algorithmic and stability considerations. The model for the liquid dynamics L is applied first to calculate the force k and to pass it to the model for the solid body dynamics S. Subsequently, the model for the solid body dynamics is applied, calculating the motion b and passing it to the model for the liquid dynamics (see figure 10). This iterative method is stable because the liquid mass is considered part of the solid mass in the actual calculations. [Gerrits, 2001]).

Interesting control challenges are to counteract liquid-slosh that results from release of the satellite from the carrier (rocket), and control of liquid-slosh due to changes of directions of a satellite, or road vehicle.

6 Conclusions

This paper presented some examples of non-smooth aerospace systems. The systems are complex and built of sub-systems that often have a very different dynamic behavior, due to their different physical nature. Some sub-systems are usually influenced by impact phenomena during operation.

Real-time simulation of aerospace systems is essential in the design phase because of validation and training purposes. Mathematical methods have been described and applied to handle real-time simulation of impact phenomena and coupled systems. The methods use the Lagrange multiplier approach. Particularly the discrete Lagrange multiplier method proves to be very suitable for stable real-time simulation of non-smooth aerospace systems: it satisfies our needs for reliable numerical methods that can be used across project boundaries and across applications.

Current work is directed towards a further integration of vehicle models and liquid-slosh models, and controller design of vehicles carrying liquid loads. It is expected that the approach as well as algorithms will be reusable in other applications due to its mathematical foundation.

7 Acknowledgements

Aerospace system design and presentation is teamwork. The contributions from Jacco van Weert (section 5) and Zacharias Klaasse (section 4) are greatfully acknowledged.

8 References

Brogliato, B., Dam, A.A. ten, Paoli, L., Génot, F., and Abadie, M. (2002). Numerical simulation of finite dimensional multibody nonsmooth mechanical systems, *ASME Applied Mechanics Reviews*, Volume 55, Number 2, pages 107-150.

Brogliato, B. (1999). *Nonsmooth Mechanics*, second edition, Springer Verlag, Communications and Control Engineering Series, London.

Dam, A.A. ten, Dwarshuis K.F., and Willems, J.C. (1997). The contact problem for linear continuous-time dynamical systems: a geometric approach, *IEEE Transactions on Automatic Control*, AC-42, No. 4, pages 458-472.

Dam, A.A. ten, (1997). Unilaterally Constrained Dynamical Systems, PhD Thesis.

Dam, A.A. ten, (1992). Stable numerical integration of dynamical systems subject to equality state-space constraints, *Journal of Engineering Mathematics*, Vol.26, pages 315-337.

Dam, A.A. ten, and Willems, J.C. (1997). A system theoretical framework to study unilaterally constrained dynamical systems, *Proceedings European Control Conference ECC'97*, June 30 - July 4, Brussels, Belgium.

Klaasse, Z. (2005). Modelling of multi-wheeled vehicles on rough terrain, Masters' thesis, NLR internal report in preparation.

Gerrits, J. (2001). Dynamics of Liquid-Filled Spacecraft, Numerical Simulation of Coupled Solid-Liquid Dynamics, PhD Thesis.

Kos, J. and Dam, A.A. ten. (1999). Reducing conception times and promoting the use of distributed resources in space system design. In *Proc. 50th International Astronautical Congress*, Amsterdam, The Netherlands, 4-8 October, IAF-99-U.1.02. Also NLR-TP-2000-257.

Lammen, W.F., Zwartbol. T., Jansen, M, Dam, A.A. ten, Arcioni, M., Wijnands, Q. (2004). Automatic Model Transfer in Space Applications, NLR-TP-2004-453, *Proceedings SESP 2004*, ESTEC, Noordwijk, The Netherlands

Raadt, M.P.M.G. van den. (2004). Building a vehicle from parts: a case-study in model-instantiations, NLR memorandum AVCE-2004-030.

Rozema, G. (2004). Modeling and simulation of vehicles carrying liquid cargo, Masters' thesis, NLR-Memorandum AVCE-2004-040.

Schram, M. (2004) Modelling a vehicle with use of partial vehicles in three dimensions, NLR-Memorandum AVCE-2004-050

Sloshsat information. (2005). http://www.nlr.nl/ public/facilities/f183/f183.php