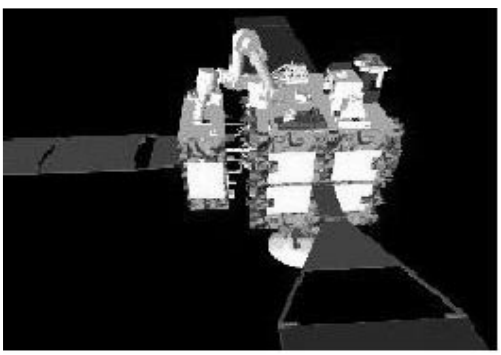
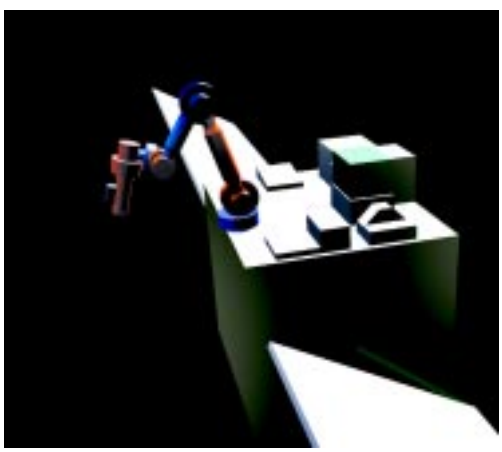
Roberto Lampariello
DLR, Oberpfaffenhofen

Modelling and Real-time Simulation of a Two Arm Free-Flying Robot with SIMPACK

The research and development in the field of free-flying robots at the Robotics and Mechatronics Institute of DLR has been supported, ever since its beginning, by the SIMPACK multibody dynamics software tool. Both modelling and simulation of these systems in the SIMPACK environment have been of aid in performing tasks such as path planning, parameter identification and control strategy development. The greatest contribution however, due to the symbolic code export capability, has lead to the development of a simulator, to run in a Unix environment, for a two arm robot system. This application was devised for tele-operation thus requiring real-time capabilities of the underlying software.



ETS - VII free-flying robot: Orbital set-up (courtesy of NASDA)



ETS - VII free-flying robot: SIMPACK simulation model

The motivation behind the development of a free-flying robot simulator lies in the fact that such systems are still at an early stage of development and optimal strategies of operation as well as feasible operational tasks are still to be determined. The Robotics and Mechatronics Institute of DLR (Deutsches Zentrum für Luft- und Raumfahrt) was actively involved with what was the first free-flying robot to be flown in Low Earth Orbit in 1998, the ETS-VII experimental satellite of the Japanese Space Agency NASDA (see figure on the left site). Further research at the Institute is now looking at two arm systems, attempting to promote its light weight robot arm and robot hand, as well as the free-flying robot concept. Possible tasks which can be envisaged for free-flying robots are: satellite repair and maintenance; satellite refuelling; in-orbit construction; de-orbiting of non-co-operative objects (such as defected satellites or space debris).

The major characteristic of free-flying robots from a dynamics point of view is the fact that, unlike ground robots, the base is not inertially fixed. This is evident from the fact that the robot is mounted on a satellite. As a result of this, an interaction between the robot and the base motion takes place which gives rise to a particular kind of dynamic behaviour to the system.

From a modelling point of view, the robot can be represented in a straightforward manner. All that is different from a conventional robot is the fact that extra degrees of freedom are introduced by the free moving base. The system, which is generally composed of rigid bodies, will then have $3+n$ degrees of freedom, where n is the number of joints of the robot(s). Note in fact that 3 degrees of freedom are eliminated by the conservation of linear momentum, meaning that the system is completely described by the robot motion and only three of the states of the base body (satellite).

With redundant robots arms, which generally have seven degrees of freedom, a two arm free-flying robot will then have 17 degrees of freedom in all. This of course is not taking the end-effectors into consideration.

In order to achieve a real-time simulation environment for the above described system, the structure described in the lower figure on the right site was adopted. The lower figure on the right site can be described as follows:

- The input to the simulator is determined by two space mice, which control the state (position and orientation) of the end-effectors of the two robot arms. This input is given in terms of small variations of the states, described in the figure by the vector variable Δx^e. Further input is

determined by the variables Δm^e and ΔI^e which represent desired variations of the mass and inertia of the load on the end-effectors.

- The state variations Δx^e are processed by the 'Inverse kinematics' module, which generates the equivalent joint position variations of the robots, vector $\Delta\theta$ required to obtain the desired end-effector motions. This module solves the inverse kinematics as an optimisation problem, minimising the robot motion as well as posing some extra constraints to allow for preferred postures of the arm during its motion.

- The desired joint position variations are fed into the dynamic model of SIMPACK. The latter was obtained as a symbolic code export of a free-flying robot model. Further modifications of the model, with respect to the one described above, are the use of rheonomically driven joints (14 in all) to drive the two robots. This resulted in a reduction of the degrees of freedom of the system to only six. The rheonomic functions were programmed in the user routines as time excitations with the spline form shown in the upper figure on the right site. The output of the 'Dynamics' module is, after integration of the equations of motion of the system which is subject to the desired robot joint motions, the state variation of the satellite base, vector Δx^b.

- The new position of the free-flying robot is updated in the simulator viewer.

The results of this simulator structure are very good, in that the simulator runs in real-time on a standard Sgi machine (with extended graphics capabilities). The integration of the SIMPACK symbolic export code is performed with the explicit Euler integration method.

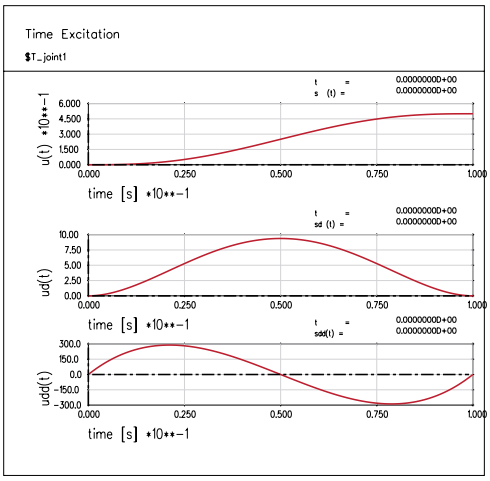
The feature of varying the mass and inertia of the load on the end-effectors of the robot (variables Δm^e and ΔI^e) was introduced in the simulator to demonstrate how the dynamic interaction between the robot and the satellite base motions varies in relation to the ratio between the size of the two. This feature was included in the symbolic code export by hand with some effort and was validated with an independent code of the robot dynamics. However, parameter dependent symbolic codes should become available in future SIMPACK versions, as the SIMPACK developers have confirmed.

The capacity to generate a symbolic code of multibody systems is a very powerful tool since it allows for a versatile means to incorporate models in independent working environments. This is in view of performing optimisation work as well as parameter identification, control and design. The advantage of saving the user a great deal of programming time is clearly evident. Furthermore, the efficient programming of the equations of motion of complex multibody systems in SIMPACK has proven to provide very good results for real-time computation.

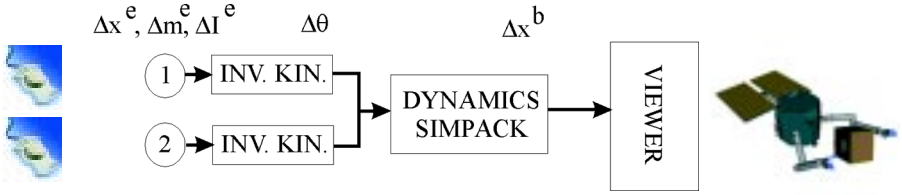
For any further information of free-flying robots research at the DLR please contact R. Lampariello at roberto.lampariello@dlr.de or refer to the web pages:

http://www.op.dlr.de/FF-DR/dr_mkd/staff/lampo/Roberto.Lampariello.html

http://www.op.dlr.de/FF-DR/dr_mkd/research/SpaceRobots.html



Time excitation for robot joints (time,velocity,acceleration)



Space mice Simulator flow diagram Robonaut free-flying satellite model