Dynamics modeling of structure-varying kinematic chains for free-flying robots

Roberto Lampariello, Satoko Abiko, Gerd Hirzinger

Institute of Robotics and Mechatronics

German Aerospace Center (DLR)

82234 Weßling, Germany

Roberto.Lampariello@dlr.de

Abstract— **A new method for the computation of the dynamics of structure-varying kinematic chains is proposed. This is based on the complete redefinition of the system connectivity deriving from a given structural change. The derived computational efficiency is then described with examples of typical motion planning tasks and structure changes for free-flying robots. These include open branched chains and closed loops, in free and grappled conditions. The method may then contribute to the efficiency of motion planning for robots which may require different kinematic structures for a given task.**

I. Introduction

This paper presents an algorithm for the efficient computation of the dynamics of structure-varying kinematic chains. As major application, multiple-arm free-flying space robots are considered. Other applications may however also be of interest. Structure variants include open branched chains and closed loops, in free and grappled conditions (see Fig. 1).

The aim of this work is to support motion planning methods, to be implemented on robots which may require different kinematic structures for different tasks, or even for single complex tasks. As is already argued in [1], it is generally useful to be able to handle structure changes without switching between different models and algorithms. Furthermore, the alternative of locking non-active joints, leads to computational inefficiency.

The goal of computing the dynamics of structure-varying kinematic chains was previously treated in [1], where the proposed method was based on two important features: firstly, changes to the connectivity, which describes the kinematic structure of the system, are minimized; secondly, the use of virtual links is made. Examples are given for handling closed loops and grappling of a free-floating robot to an inertially fixed structure. It is then argued that the advantage in this approach is that it is simple to implement and that it can be easily parallelized.

The method proposed here instead, is based on full connectivity variation. In fact, despite the extra implementation complexity, it is shown that this approach present an advantage in the computational efficiency.

As mentioned above, another advantage of the structure-varying modeling is to avoid unnecessary computations. This
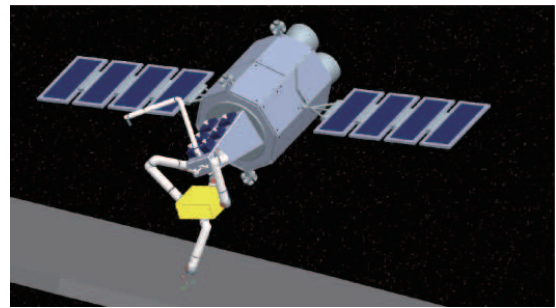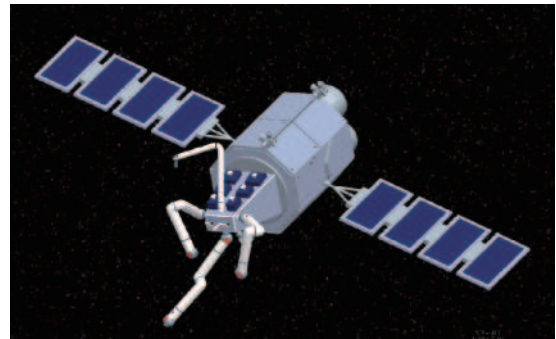

Fig. 1. Free-flying robot with four arms: free open branched chain (above); grappled closed loop (below)

is because the dynamic model of a given system can be reduced to only possess the degrees of freedom which are necessary to plan or perform a given task. Other degrees of freedom, which would normally have to be locked, are thus omitted in the computation. This point is demonstrated with some examples.

Structural changes are handled on-line, which practically means that they can take place without necessity to recompile the algorithm. This is a mandatory feature for treating tasks during which the structure of the system varies. It also opens the possibility to eventually tackle the problem of optimizing the kinematic structure for a given system and a given task.

The modeling of the free-flying base is derived from basic principles of free rigid-body dynamics, as opposed to the alternative six degree-of-freedom virtual link approach. Closed loops are also treated in [2], where Lilly and Bonaventura

proposed a dynamic formulation for a space robot with one robot arm, constrained at the end-effector to be fixed with the base. In this paper, we consider a more general formulation for free-flying space robots, to allow for one or more closed loops between any of the system links and also to treat branching.

The paper is then structured as follows: in section II the mathematical model used for a free-flying robotic system is presented. In section III, the necessary functions to perform structure changes are described. Examples are then treated in section IV, including one discussed in [1] and others for 3D free-flying robots. Section V gives the conclusions.

II. MODELING

A. Open branched chains

In this section the modeling of the dynamic equations is introduced. We first consider the simplest case, that of the open kinematic chain. We assume a free-flying base, as opposed to one fixed to an inertial point. We will later arrive at this latter case, by virtue of the proposed modeling method. The system of interest in depicted in Fig. 2.

1) Equations of motion: The equations of motion are the following:

$$
\begin{bmatrix} \boldsymbol{H}_b & \boldsymbol{H}_{bm} \\ \boldsymbol{H}_{bm}^T & \boldsymbol{H}_m \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_b \\ \ddot{\boldsymbol{\phi}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{c}_m \end{bmatrix}
$$
$$
= \begin{bmatrix} \mathcal{F}_b \\ \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_b^T \\ \boldsymbol{J}_m^T \end{bmatrix} \mathcal{F}_e, \quad (1)
$$

where $\boldsymbol{x}_b$ and $\boldsymbol{\phi}$ are the generalized coordinates of the base body and the robot joints respectively. The remaining

TABLE I
MAIN NOTATION IN THE DYNAMIC EQUATIONS

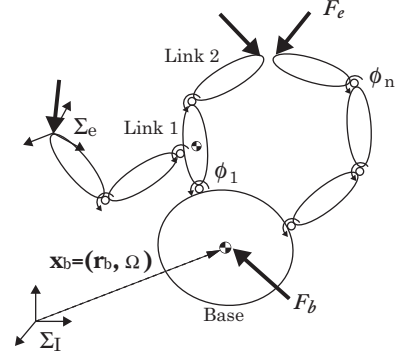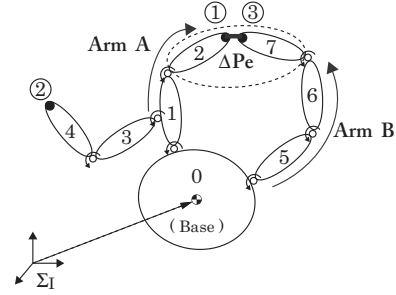|		
$\boldsymbol{H}_b$	$\in R^{6 \times 6}$	: inertia matrix of the base
$\boldsymbol{H}_m$	$\in R^{n \times n}$	: inertia matrix of the robot arms
$\boldsymbol{H}_{bm}$	$\in R^{6 \times n}$	: coupling inertia matrix between the base and the arms
$\boldsymbol{c}_b$	$\in R^{6 \times 1}$	: non-linear velocity dependent term on the base
$\boldsymbol{c}_m$	$\in R^{n \times 1}$	: non-linear velocity dependent term of the arms
$\mathcal{F}_b$	$\in R^{6 \times 1}$	: force and moment exerted on the base
$\mathcal{F}_e$	$\in R^{6 \times 1}$	: force and moment exerted on the end-effector
$\boldsymbol{\tau}$	$\in R^{n \times 1}$	: torque on the joints
$\boldsymbol{J}_b$	$\in R^{6 \times 6}$	: Jacobian matrix for the base
$\boldsymbol{J}_m^i$	$\in R^{6 \times n}$	: Jacobian matrix for the ith arm
$\boldsymbol{J}_C$	$\in R^{6 \times n}$	: Jacobian matrix for the closed loop

Fig. 2.   Open Tree Structure

Fig. 3.   Closed Loop Chain

variables are described in table I. A gravity-free environment is assumed.

In implementing these, the *Articulated Body Algorithm* (ABA) proposed by Featherstone [3] is used. This is well applicable, since it directly addresses free-floating systems. The ABA provides $O(n)$ in both the forward and the inverse dynamics.

2) Connectivity: The connectivity is described as follows, following the standard approach. The i-th body is attributed an index, $parent(i)$ which defines its parent. The first body is generally the base body, with index 0.

Furthermore, one or more bodies are defined as an end-effector, by means of a second index, $end - effector(i)$, which is non-zero if affirmative and 0 otherwise. For more end-effectors, the index is also used to number them.

For the case of branching, an example of which is shown at link 1 in figure 2, a parent simply possesses two (or more) children.

B. Closed loop

The dynamic model for the closed loop stems from that of the open chain structure, subject to the closed loop kinematic constraints. In a simple example in Fig. 3, the constraint between the arm A and the arm B can be described with

the Jacobian matrices:

$$\left[\begin{array}{cc} \boldsymbol{J}_m^A & -\boldsymbol{T}\boldsymbol{J}_m^B \end{array}\right]\left[\begin{array}{c} \dot{\boldsymbol{\phi}}_A \\ \dot{\boldsymbol{\phi}}_B \end{array}\right] = \boldsymbol{J}_C\dot{\boldsymbol{\phi}} = \boldsymbol{O}, \qquad (2)$$

where

$$\boldsymbol{T} = \left[\begin{array}{cc} \boldsymbol{E} & \widetilde{\Delta\boldsymbol{P}_e} \\ \boldsymbol{O} & \boldsymbol{E} \end{array}\right] \qquad (3)$$

and where $\Delta\boldsymbol{P}_e$ is the vector between the two end-effectors.

Furthermore, if m independent kinematic constraints exist, which may relate to $N \geq m$ linearly dependent closed loops, the constraint equation is expressed by:

$$\boldsymbol{J}_C\dot{\boldsymbol{\phi}} = \left[\begin{array}{cc} \boldsymbol{J}_G & \boldsymbol{J}_D \end{array}\right]\left[\begin{array}{c} \dot{\boldsymbol{\phi}}_G \\ \dot{\boldsymbol{\phi}}_D \end{array}\right] = \boldsymbol{O}, \qquad (4)$$

where the joint vector $\boldsymbol{\phi}$ is divided into the independent(active) joints vector $\boldsymbol{\phi}_G \in R^{(n-m)}$ and the dependent(passive) joints vector $\boldsymbol{\phi}_D \in R^m$. $\boldsymbol{J}_G \in R^{m \times (n-m)}$ and $\boldsymbol{J}_D \in R^{m \times m}$ denote the Jacobian matrix with respect to the joints $\boldsymbol{\phi}_G$ and $\boldsymbol{\phi}_D$, respectively.

By virtue of d'Alembert's principle and the above kinematic constraints, the dynamic equations of the closed loop system can be deduced from that of the tree-structure system (1) as follows:

$$\boldsymbol{M}\left[\begin{array}{c} \ddot{\boldsymbol{x}}_b \\ \ddot{\boldsymbol{\phi}}_G \end{array}\right] + \boldsymbol{C} = \left[\begin{array}{c} \mathcal{F}_b \\ \boldsymbol{\tau}_G \end{array}\right] + \boldsymbol{\Pi}^T\left[\begin{array}{c} \boldsymbol{J}_b^T \\ \boldsymbol{J}_m^T \end{array}\right]\mathcal{F}_e, \qquad (5)$$

where

$$\boldsymbol{M} = \boldsymbol{\Pi}^T\boldsymbol{H}\boldsymbol{\Pi},$$
$$\boldsymbol{C} = \boldsymbol{\Pi}^T\left[\begin{array}{c} \boldsymbol{c}_b \\ \boldsymbol{c}_m \end{array}\right] + \boldsymbol{\Pi}^T\boldsymbol{H}\dot{\boldsymbol{\Pi}}\left[\begin{array}{c} \dot{\boldsymbol{x}}_b \\ \dot{\boldsymbol{\phi}}_G \end{array}\right],$$
$$\boldsymbol{H} = \left[\begin{array}{cc} \boldsymbol{H}_b & \boldsymbol{H}_{bm} \\ \boldsymbol{H}_{bm}^T & \boldsymbol{H}_m \end{array}\right],$$
$$\boldsymbol{\Pi} = \left[\begin{array}{cc} \boldsymbol{E}^{6\times6} & \boldsymbol{O}^{6\times(n-m)} \\ \boldsymbol{O}^{(n-m)\times6} & \boldsymbol{E}^{(n-m)\times(n-m)} \\ \boldsymbol{O}^{m\times6} & -\boldsymbol{J}_D^{-1}\boldsymbol{J}_G \end{array}\right].$$

$\boldsymbol{M}$ and $\boldsymbol{C}$ denote the inertia matrix and the non-linear velocity dependent term for the closed loop, respectively. $\boldsymbol{\Pi}$ represents the transform matrix from the tree-structure to the closed loop system. Eq. (5) is a general formulation for the dynamics of the free-flying robot with closed loops. A more detailed derivation can be found in [4].

1) Connectivity Representation for Closed Loop Chains: To represent the connectivity for the closed loop chains, we introduce one additional index, namely *cflag*. The *cflag* index indicates that the i-th link forms a closed loop with the j-th link as *cflag(j) = i*.

III. STRUCTURAL CHANGES

In this section we describe the method for treating structural changes. The latter comprise the following possibilities:
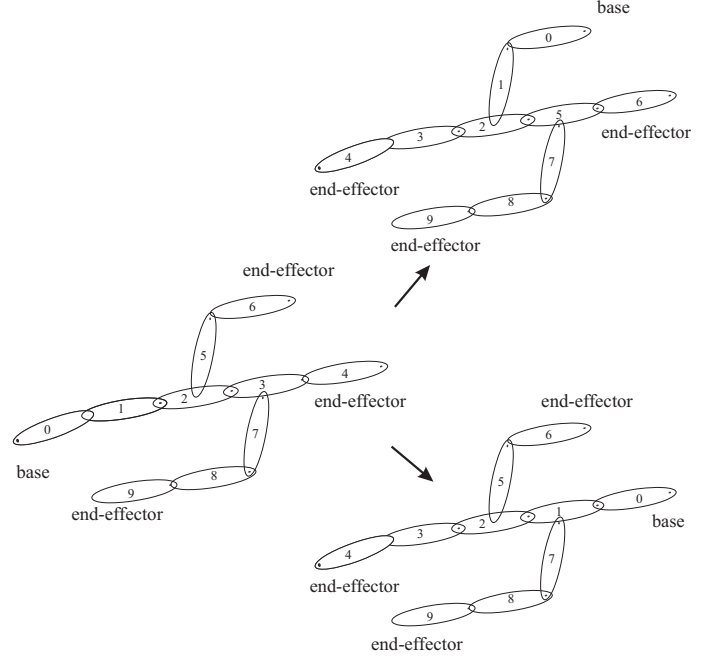

Fig. 4. Examples for chain inversion of a branched system

- given a system with n joints, to lock p of them, resulting in a reduction of DOF;
- given a free-flying system, to fix its base, or its end-effector, to an inertially fixed point;
- given an open kinematic chain, to close one or more loops.

To accomplish these changes, the following four functions are constructed.

A. New number of joints

This function calculates the new parameters which describe the system for its $n - p$ DOF. The parameters for the ith link consist of the inertial and the DH parameters. These parameters are of course known for the complete system.

The required transformation is function of the position of the locked joints. Practically, each locked joint is omitted in the new modeling description and the parameters relative to the links before and after it are updated, in function of its position. Mathematical rules for the updating are here omitted, but can be found in [5], for example, for the calculation of the new composite inertia.

In this process, the connectivity of the system is also updated. To represent the locked joints, we introduced an additional index, namely *connect*, which expresses the number of links which are connected together. Examples are given in section IV.

The equations of motion remain structurally identical to (1), whereas the dimension of $\boldsymbol{\phi}$ is reduced or increased to represent the new number of joints of the system.

TABLE II
LINK CONNECTIVITY REPRESENTATION FOR FIGURE 3

Link No.	parent	end-effector	connect	cflag
1	0	0	0	-1
2	1	1	0	-1
3	1	0	0	-1
4	3	2	0	-1
5	0	0	0	-1
6	6	0	0	-1
7	7	3	2	2

B. Invert chain

This function inverts the order of the system connectivity, such that a chosen end-effector becomes the new base. This is a simple procedure for an open kinematic chain.

For a branched system the procedure is more involved. Consider the example shown in Fig. 4. From it, the following rules follow:

- given a new base, construct the connectivity of an open chain moving along the system, until an end-effector is reached;
- then continue the numbering of the remaining branches, such that the first child belonging to a branch is connected to the new parent which has the same numbering as that which its old parent receives.

The equations of motion remain structurally identical to (1).

For the closed loop case, the same procedure applies, as shown in the example in Fig. 6.

C. Extract fixed-base equations

This function extracts the fixed-base system equations from the equations of the equivalent free-flying system. When referring to Eq. (1), the outcome is then simply the second row with $\ddot{x}_b = \dot{x}_b = 0$.

D. Close loop

An example is shown in switching from Fig. 2 to Fig. 3. The system has in total n DOF. When the open kinematic structure changes to the closed loop, two links are connected together, as shown in Fig. 3.

As mentioned in section II-B, the i-th link forms the closed loop with the j-th link by expressing *cflag(j) = i*. The corresponding model description for the system depicted in Fig. 3 is shown in Table II. In the example, *link 2* and *link 7* are connected to form the closed loop and *cflag(7) = 2* is set, as shown in the table.

The equations of motion are extended to those described in eqn. (5), by means of the transformation matrix $\mathbf{\Pi}$.

IV. EXAMPLES

A. Example 1: Open chain grappling to fixed inertial point

In this example, the first computational advantage of the proposed method can be seen. Consider the structure variation shown in Fig. 5, where a one joint free-flying robot grapples to a fixed inertial point by means of a one DOF fixture. In alternative to the approach proposed in [1], where the connectivity is preserved and a subsidiary link between the end-effector and ground is introduced, the following procedure is instead suggested, where the system is converted to a fixed-base system and its connectivity inverted:

- in order to allow for an extra joint at the end-effector, an extra link is first added to the system, by means of the *New number of joints* function;
- the system is inverted, by means of the *Invert chain* function;
- finally, the fixed-base equations are derived, by means of the *Extract fixed-base equations* function.

This example shows how this approach allows to represent a free-flying system and a fixed-base system with the same model. Furthermore, a fixed-base two-degree-of-freedom system results. This has a clear numerical advantage on the 5 degree-of-freedom system obtained in [1] (with also the additional geometric constraints on the end-effector). The numerical burden for calling the structure varying functions is of the order of milliseconds. This is neglegible, if compared to the efficiency gained in the thousands of function calls, necessary in motion planning procedures.

B. Example 2: Closed loop grappling to fixed inertial point

Consider the structure variation shown in Fig. 6, where a three-arm free-flying system grapples to a fixed inertial point forming two closed loops. The following procedure is applied:

- the first loop is closed, by imposing the necessary constraint on end-effector 2, to be connected to end-effector 1. This is achieved by setting the index $cflag(5) = 3$;
- the system is inverted, such that end-effector 1 becomes the base, by means of the *Invert chain* function. The change in connectivity can be seen in Fig. 6;
- the fixed-base equations are obtained by means of the *Extract fixed-base equations* function;
- finally, the third arm is connected to the base, by applying the closed loop constraint on its end-effector and by setting the index $cflag(7) = 0$. The appropriate choice of the vector ΔP_e (defined in figure 3) ensures the correct positioning of the point of connection to the fixed structure.

C. Example 3: DOF reduction for an open chain

A free-flying robot may consist of a multibody system with up to 37 DOF, when considering 6 DOF for the base, 7 DOF for each arm and 3 DOF for reaction wheels (see Fig. 1). It
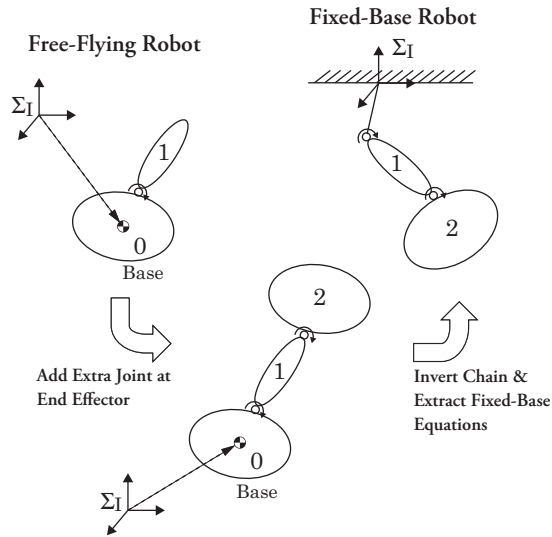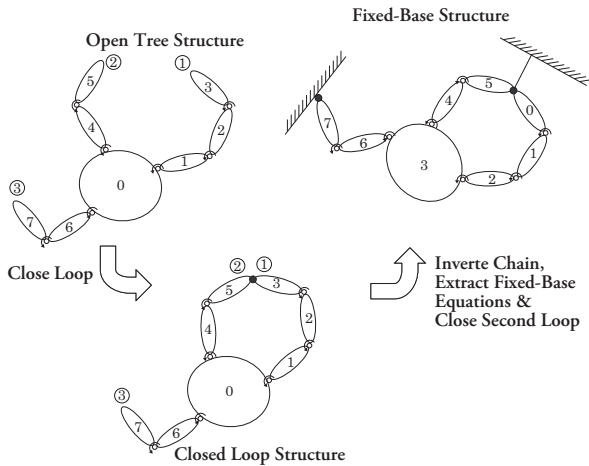
Fig. 5. Simple Example for Varying Structure

Fig. 6. Free-Flying Space Robot Application

TABLE III
LINK CONNECTIVITY REPRESENTATION FOR EXAMPLE IV-C: 2 ARM
SYSTEM WITH 2 ACTIVE JOINTS IN ARM 1

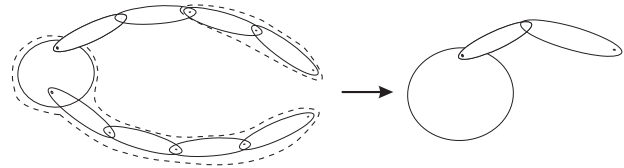| Link No.	parent	end-effector	connect	cflag
BEFORE				
1	0	0	0	-1
2	1	0	0	-1
3	2	0	0	-1
4	3	0	0	-1
5	4	1	2	-1
6	1	0	4	-1
7	6	0	4	-1
8	7	0	4	-1
9	8	2	4	-1
AFTER				
1	0	0	0	-1
2	1	0	0	-1
3	2	1	0	-1

Fig. 7. System description for Example IV-C: 2 arm system with 2 active joints in one arm

system and using a 37 DOF system with 28 locked joint is evident. Note that the computational burden between actuated and locked joints is equivalent, due to the recursive nature of the dynamics computation. The computation of the path planning problem is then linear in the number of joints. This is because it involves the computation of the system Jacobian matrices for the base and for the end-effector, and of the actuation forces, through the inverse dynamics. These are both linear functions in the number of joints for the algorithms applied here, as described in section II. As such, for this example the time for the reduced system is approximately one quarter of that for the full system. Typical running times for our current implementations range between 15 and 30 seconds on an Intel Pentium 4 (3.2 GHz), for the reduced system.

An example for a grasping maneuver of a tumbling target is given in Fig. 8. Here, the task is divided into: a waiting phase, where the target approaches the stationary robot; an approach phase, i.e. a point-to-point maneuver of the robot towards the target; a tracking phase, where the robot tracks the moving target to avoid impacts; a grasping and stabilization phase. The computational time for this optimal motion planning task is in the order of 30 seconds on an Intel Pentium 4 (3.2 GHz), for a 13 DOF robot system.
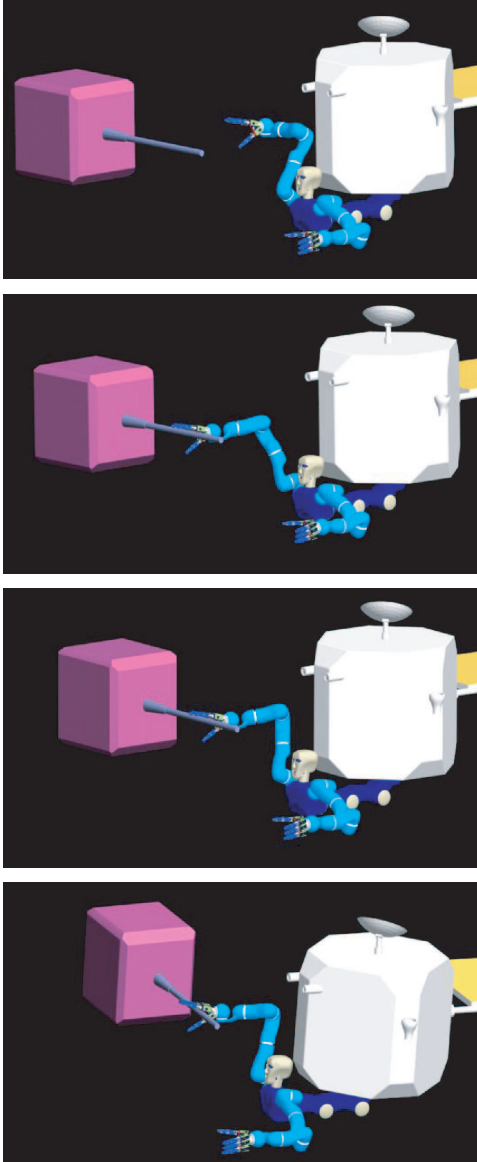
may however need to perform tasks which only require part of these DOF.

As a first example, maneuvers where discussed in [6], where *three* joints of one arm were used to induce motions of the free-floating base for identification of the latter's inertial parameters, or those of the load (it was assumed that the inertial parameters of the arms were known from CAD data). This process involved solving a motion planning problem, based on the integration of the system equations of motion. The necessary DOF for this task however are only 6 DOF for the base + 3 DOF for the actuated arm. The model reduction is then from 37 to 9 DOF. For this example, the respective connectivity indexes before and after the structural change are described in table III. A simplified version of this operation is shown in Fig. 7.

The computational advantage between using a 9 DOF

case of locked joints, where these are simply omitted from the computation.

Since structure variations are handled on-line, there is no need to prepare every possible kinematic chain in advance, for a given system. This constitutes an important practical advantage. Furthermore, the method improves the efficiency of motion planning.

Examples are given for the application of the proposed method, which include typical motion planning tasks and structure changes for free-flying robots. Free-flying robots models are thus switched between open branched chains, closed loop, in the free and in the grappled states.

REFERENCES

[1] Nakamura, Y., Yamane, K.: *Dynamics Computation of Structure-Varying Kinematic Chains and Its Application to Human Figures*, IEEE Trans. on Robotics and Automation, Vol. 16, No. 2, pp. 124 – 134, April 2000.

[2] Lilly, K.W., Bonaventura, C.S.: *A Generalized Formulation for Simulation of Space Robot Constrained Motion*, The Proc. of the 1995 IEEE International Conference on Robotics and Automation, vol. 3, pp. 2835 – 2840, Nagoya, Japan, May 1995.

[3] Featherstone, R.: *Robot Dynamics Algorithms*, Kluwer Academic Publishers, 1987.

[4] Park, F.C., Choi, J., Ploen, S.R.: *Symbolic Formulation of Closed Chain Dynamics in Independent Coordinates*, Mechanism and Machine Theory, No. 34, pp. 731–751, 1999.

[5] Walker, M.W., Orin, D.E.: *Efficient Dynamic Computer Simulation of Robotic Mechanisms*, Journal of Dynamic Systems, Measurement and Control, vol. 104, pp. 205–211, September 1982.

[6] Lampariello, R., Hirzinger, G.: *Modeling and Experimental Design for the On-Orbit Inertial Parameter Identification of Free-Flying Space Robots*, ASME 2005, Long Beach, California, September 2005.

[7] Lampariello, R., Deutrich, K.: *Simplified Path Planning for Free-Floating Robots*, Technical Report 515-99-04, Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 1999.

Fig. 8. Grasping maneuver of tumbling target: wait, approach, track and grasp

For the above motion planning examples, the structure-varying approach may give a clear computational advantage. Furthermore, note that the same approach allows, in principle, to tackle the optimization problem of determining which structure may be best for performing a desired task.

V. CONCLUSION

A new method for the computation of the dynamics of structure-varying kinematic chains is proposed. A first computational advantage with respect to existing approaches, validated with examples, is gained by the complete redefinition of the system connectivity deriving from a given structural change. A second advantage is pointed out, for the