# ON INVERSE KINEMATICS AND KINETICS OF REDUNDANT SPACE MANIPULATOR SIMULATION

Bernd Schäfer, Rainer Krenn and Bernhard Rebele
Institute of Robotics and Mechatronics
German Aerospace Center (DLR), 82234 Wessling, Germany
bernd.schaefer@dlr.de

[Received: June 14, 2002]

**Abstract.** Redundant manipulators are nowadays favoured in several future space mission scenarios in order to enhance the skill and flexibility of the entire system. DLR, since many years, has been engaged in the development of light-weight robotic systems in modular design. Typical tasks for space applied robotics are to define robot kinematics and to calculate joint kinetics very rapidly in order to support the whole space mission design from the very beginning. Redundant manipulators then require the solution of the inverse kinematics problem for more than 6 degrees of freedom. Equivalently, the respective joint torques and forces are to be calculated by forward and backward recursions. Rather than applying conventional schemes based on pseudo-inverse matrix methods, we favour optimization with equality constraints, based upon the well-known Lagrange formalism. The optimization criteria are chosen to represent the underlying physical meaning, such as minimization of joint velocities, accelerations, torques or power, or even an optimization criterion that maintains the entire robot configuration during motion very close to a reference configuration. Simultaneously, this procedure also takes care that the joint loads and stresses in structural arm links do not exceed upper bounds. Two examples of light-weight robot design for space applications are presented that very clearly show the efficiency of the underlying algorithms.

*Mathematical Subject Classification:* 70B15
*Keywords*: inverse kinematics and kinetics, redundant manipulators, space robotics, optimization with constraints

## 1. Introduction

Future robotic-based satellite missions will be unimaginable without the demand for strong reduction of weight and for enhanced autonomy and intelligence. For many years DLR has paved the way for these challenging novel technologies. Currently, DLR is developing very light-weight robotic systems of its $3^{rd}$ generation in modular design, which are used as the design baseline for future space robot system scenarios of various kind [1]. Several studies are presently in progress (Figure 1) in order to prepare future robotized missions and to benefit from the use of robotic manipulators in space [2],[3]. It stands to reason that the desire to use light-weight and slender robotic systems is very strong since it promises very new application scenarios. Especially, this is more than ever true in all cases where the number of degrees of freedom is increased beyond the traditional maximum number of six. Such redundant manipulators are able to enhance the skill and flexibility of the entire robotic system, and hence are able to increase the operational use and to obtain the maximum use of the robotic workspace. Moreover, avoiding collision with obstacles, and the avoidance of and passing through singularities are two more major aspects that make the use of redundancy in manipulators for space applications very preferable.

One of the prevailing design tasks for a given space mission scenario including robotics is the definition of the robot kinematics, as well as the number of joints and the sequence of joint axes orientations. Moreover, the knowledge of the joint kinetics states, i.e., forces and torques for given motions, is very important in order to rapidly support the whole space mission design from the very beginning. Modifications in manipulator design are then feasible at an early stage of the whole mission design in cases where upper bounds for manipulator loads and stresses are exceeded.

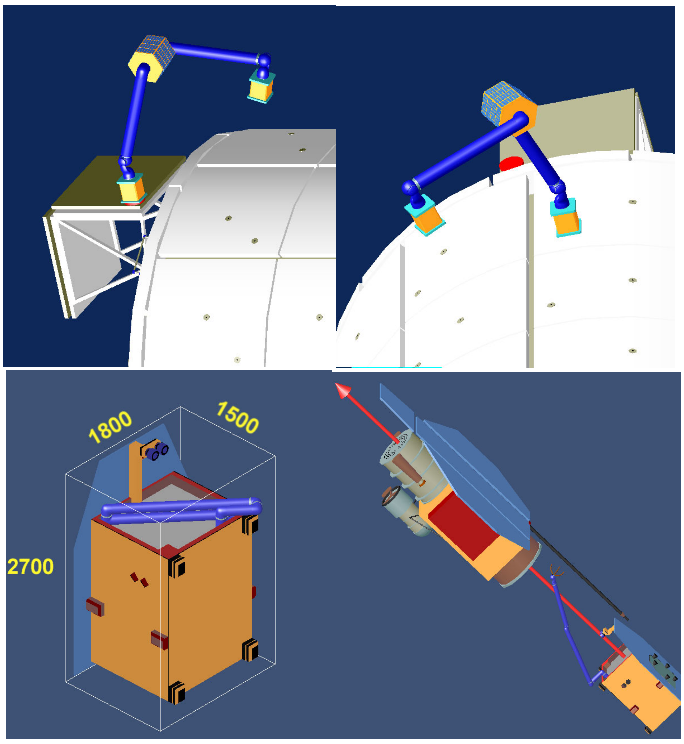


Figure 1. Two examples of near-future robotic-based space missions: MISSISS, Mobile Inspection and Service System for the International Space Station ISS (upper two figures), manipulator walk-over system operating on ISS Columbus external shell. Rescue satellite (lower two figures), manipulator used to capture a malfunctioning satellite (Rosat, the German Roentgen satellite, in orbit since 1990) followed by a de-orbiting scenario

However, the problem of solving the inverse kinematics increases according to the number of joints. Conventional solutions to this problem area are mostly based upon the use of pseudo-inverse matrix methods, where the physical meaning of the underlying method is not apparent at all. We have followed quite a different way for the inverse kinematics and kinetics problem solving. Currently, in our approach two different solutions are provided within our simulation environment, which both have their specific physical meanings. The first one is an optimization algorithm that minimizes, for example, the joint speed and acceleration subject to the end effector misalignment. The second one makes use of the differential equations describing the system of the robot motion whereby the robot is a passive chain of joints and limbs. For the latter one, the solution in joint coordinates is solved elegantly by the governing differential equations, rather than by solving algebraic equations and using minimization procedures, as in the first approach [2]. In this second case the motion is initiated by force and torque at the end effector and the joint motion can be calculated by solving the differential equations. Then, the dynamics parameters of the chain (e.g. joint friction, limb mass) affect the overall robot motion analogously to the optimization goal of the first method. This method is very appealing because it can be easily implemented in available multi-body dynamics simulation software. However, in this paper we will focus on the optimization method with equality constraints using Lagrange formalism for obtaining an optimal solution.

## 2. **Inverse kinematics and kinetics**

2.1. **Description of kinematics.** Figures 2 and 3 give a representation of the various coordinate systems and notations used to describe multi-body kinematics. The desired (index $d$ within the formulas) end effector position in Cartesian coordinates is given by

$$\mathbf{r}_d = \begin{pmatrix} r_{d,x}, & r_{d,y}, & r_{d,z} \end{pmatrix}^{\mathrm{T}} \tag{1}$$

and the desired orientation of the end effector by

$$\mathbf{A}_d = \begin{pmatrix} A_{d,xx} & A_{d,xy} & A_{d,xz} \\ A_{d,yx} & A_{d,yy} & A_{d,yz} \\ A_{d,zx} & A_{d,zy} & A_{d,zz} \end{pmatrix}. \tag{2}$$

Then the inverse kinematics problem has to find a set of angular positions for rotary joints and a set of translational positions for prismatic joints (e.g. telescopic arms), which can be combined by a vector $\mathbf{q}$,

$$\mathbf{q} = \begin{pmatrix} q_1, & q_2, & \ldots, & q_n \end{pmatrix}^{\mathrm{T}} \tag{3}$$

with $n$ degrees of freedom, and $n > 6$ for redundant manipulators. Both end effector position and orientation (hereafter called 'pose') can be combined by the homogeneous transformation matrix $\mathbf{B}_d$,

$$\mathbf{B}_d = \begin{pmatrix} A_{d,xx} & A_{d,xy} & A_{d,xz} & r_{d,x} \\ A_{d,yx} & A_{d,yy} & A_{d,yz} & r_{d,y} \\ A_{d,zx} & A_{d,zy} & A_{d,zz} & r_{d,z} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{4}$$

leading finally to the problem

$$\mathbf{q} = f\left(\mathbf{B}_d\right) \tag{5}$$

to be solved. In the following, we will restrict our problem to rotary joints only. Unique solutions for the joint angles $\mathbf{q}$ exist only for $n = 6$, disregarding some special robot configurations like elbow up or down, etc.. For the general case of $n > 6$ we have to consider additional constraint equations that

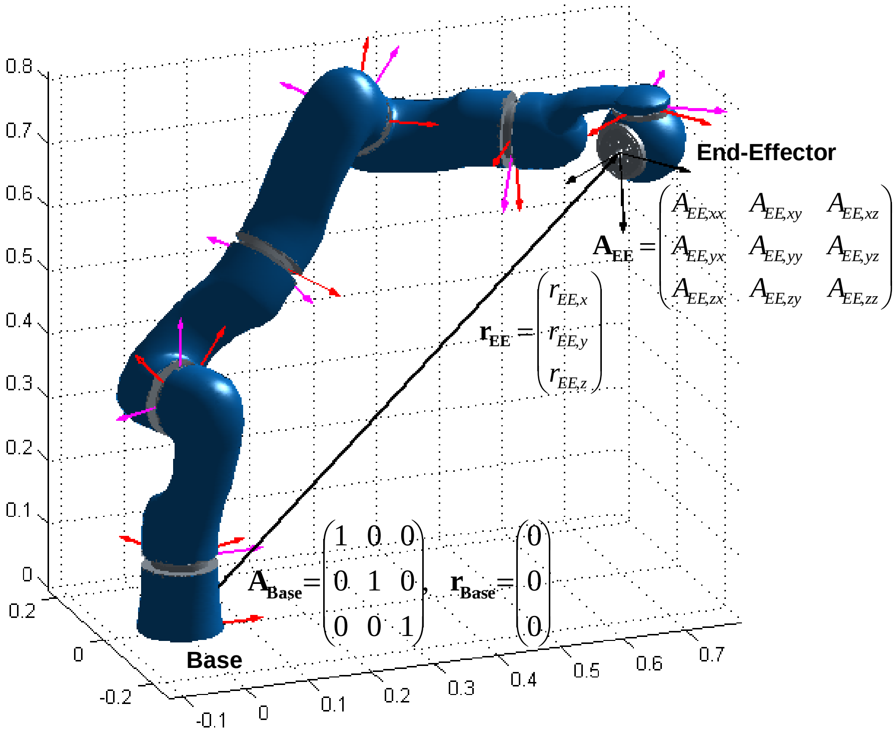

End-Effector
$\mathbf{A}_{\mathrm{EE}} = \begin{pmatrix} A_{EE,xx} & A_{EE,xy} & A_{EE,xz} \\ A_{EE,yx} & A_{EE,yy} & A_{EE,yz} \\ A_{EE,zx} & A_{EE,zy} & A_{EE,zz} \end{pmatrix}$
$\mathbf{r}_{\mathrm{EE}} = \begin{pmatrix} r_{EE,x} \\ r_{EE,y} \\ r_{EE,z} \end{pmatrix}$
$\mathbf{A}_{\mathrm{Base}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \mathbf{r}_{\mathrm{Base}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$
Base
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.2
0
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

Figure 2. Kinematic structure and coordinate systems for the entire robot. The numbers at coordinates axes are in meters

guarantee a unique solution. We will not build upon common approaches used throughout the many examples for terrestrial applications, such as the well-known 'pseudo inverse method'. Rather, we make use of an optimization procedure with constraints based upon Lagrange formalism, which accounts for a real physical interpretation of the underlying method used here.

2.2. **Optimization based inverse kinematics.** The constraint given is that the solution of the inverse kinematics problem has to guarantee coincidence between the actual pose $\mathbf{B}$ and the desired pose $\mathbf{B}_d$. For mathematical reasons this deviation can be better represented by a homogenous transformation matrix $\mathbf{\Delta}$ (see e.g. [4])

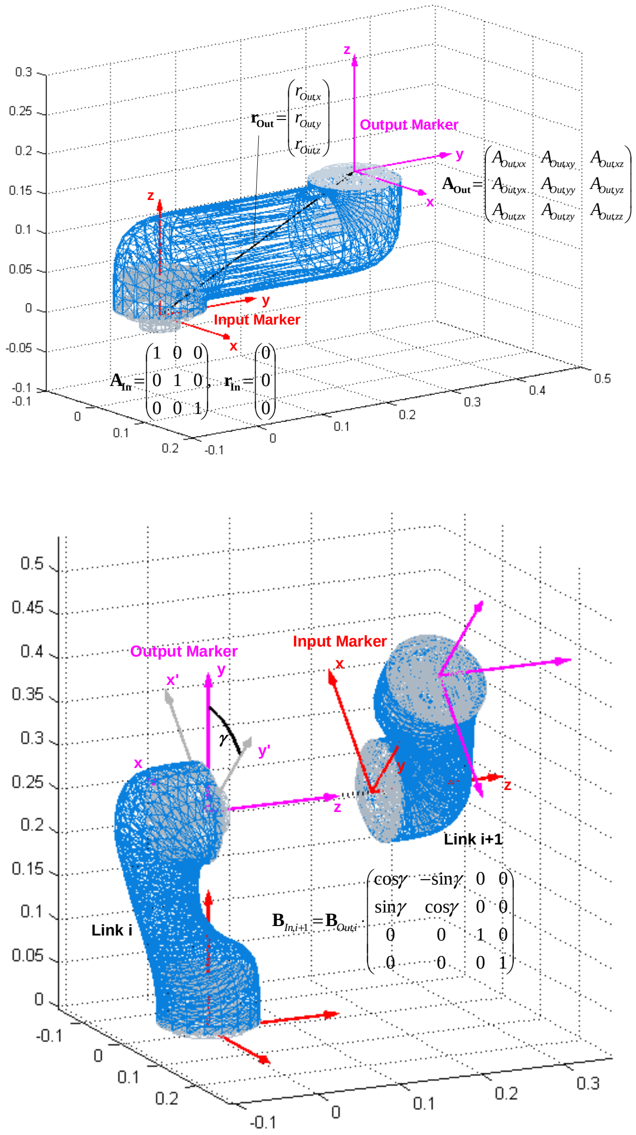

0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.1
0
0.1
0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
z
Output Marker
y
x
r_Out = (r_Outx, r_Outy, r_Outz)
A_Out = (A_Outxx A_Outxy A_Outxz; A_Outyx A_Outyy A_Outyz; A_Outzx A_Outzy A_Outzz)
z
y
Input Marker
x
A_In = (1 0 0; 0 1 0; 0 0 1), r_In = (0; 0; 0)
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.1
0
0.1
0.2
-0.1
0
0.1
0.2
0.3
Output Marker
x'
y
γ
y'
x
z
Input Marker
x
y
z
Link i+1
Link i
B_In,i+1 = B_Out,i · (cosγ −sinγ 0 0; sinγ cosγ 0 0; 0 0 1 0; 0 0 0 1)

Figure 3. Coordinate systems for a single link and two adjacent joints with input and output marker. The numbers at coordinates axes are in meters

according to

$$\mathbf{\Delta} = \mathbf{B}_d^{-1} \cdot \mathbf{B} = \begin{pmatrix} \Delta_{11} & \Delta_{12} & \Delta_{13} & \Delta_{14} \\ \Delta_{21} & \Delta_{22} & \Delta_{23} & \Delta_{24} \\ \Delta_{31} & \Delta_{32} & \Delta_{33} & \Delta_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad (6)$$

Should the desired pose be reached, then we will end up with $\mathbf{\Delta} = \mathbf{E}$, where $\mathbf{E}$ is the unity matrix. Then the problem to be solved can be transformed from matrix to vector form,

$$\mathbf{h} = \left( \Delta_{14}, \quad \Delta_{24}, \quad \Delta_{34}, \quad \Delta_{12} - \Delta_{21}, \quad \Delta_{23} - \Delta_{32}, \quad \Delta_{31} - \Delta_{13} \right)^{\mathrm{T}} = \mathbf{0}. \qquad (7)$$

The first three elements are translational components whereas the last three represent rotational ones. This formulation of the inverse kinematics problem is free of singularities and therefore numerically stable. It can be shown [5] that for an incremental robot motion simulation the upper left 3x3 sub-matrix of $\mathbf{\Delta}$ will turn into a skew symmetric one. In the limit for small angles and positions going to zero, $\mathbf{\Delta}$ then goes to the identity matrix.

Considering the optimization criteria, a physical interpretation of the underlying method is favoured. Several different approaches can be followed, which may be distinguished by two kinds of goal functions $f$ to be minimized: the first one is related to an optimization that focuses on joint drive kinematics and kinetics (joint drive oriented optimization), the second one takes care of the entire robot system configuration (configuration oriented optimization).

*Joint drive oriented optimization criteria.* The following criteria concern minimum loads and stresses applied to the robotic joints, and hence to the arm structures. For mathematical reasons, they are usually formulated as quadratic expressions of kinematic or kinetic states:

(a) Minimization of all the joint angular velocities $\dot{\mathbf{q}}$,

$$f_1 = \dot{\mathbf{q}}^{\mathrm{T}} \cdot \dot{\mathbf{q}}. \qquad (8)$$

Mechanically, this may be interpreted as minimizing some linear damping force proportional to the joint velocities.

(b) Minimization of all the joint angular accelerations $\ddot{\mathbf{q}}$,

$$f_2 = \ddot{\mathbf{q}}^{\mathrm{T}} \cdot \ddot{\mathbf{q}}. \qquad (9)$$

This optimization criterion may be favoured especially in cases where the robot is operates in a neighbourhood close to singularities, and where otherwise very high joint accelerations are required. This criterion will keep critical joints below their maximum load limits.

(c) Minimization of joint torques $t$,

$$f_3 = \mathbf{t}^{\mathrm{T}} \cdot \mathbf{t}. \qquad (10)$$

(c) Minimization of power $P_i = t_i \omega_i$ consumed in each joint drive with number $i$,

$$f_4 = \mathbf{P}^{\mathrm{T}} \cdot \mathbf{P}, \qquad (11)$$

where $\omega_i = \dot{q}_i$ and $\mathbf{P} = (P_1, P_2, ..., P_i, ..., P_n)^{\mathrm{T}}$ has been set. The latter two criteria require inverse kinetics calculation in order to obtain knowledge of the torques to be applied according to a given trajectory. Equations (8) to (11) also allow to weight each joint individually by a special weighting factor, in order to be more flexible in assigning each single joint more or less importance during the optimization run.

*Configuration oriented optimization criterion.* The motivation for this criterion is to allow the robot a high degree of manoeuvrability. This means that during motion, each joint should keep its respective configuration far apart from its joint angle limits $q_{i,\max}$ and $q_{i,\min}$. This can also be defined by a reference angle according to $q_{ref,i} = (q_{i,\max} + q_{i,\min})/2$, whereas the robot will try to stay in a neighbourhood close to the desired configuration. Moreover, this reference position may also be defined by restrictions given by the environment where the robot is operating. A typical example is to prescribe a reference configuration for a collision-free motion within the robot's working space, which the robot has to follow very closely. As a minimization criterion, again a quadratic expression is given

$$f_5 \;=\; \Delta\mathbf{q}^{\mathrm{T}} \cdot \Delta\mathbf{q}\,, \tag{12}$$

where $\Delta\mathbf{q} = \mathbf{q} - \mathbf{q}_{ref}$ and $\mathbf{q}$ contains the actual angular positions of the joints. From a mechanical point of view, this procedure can be interpreted as minimizing stiffness forces acting in the joints, which originate in virtual springs somehow depending upon spring displacements $\Delta\mathbf{q}$.

*Minimization of satellite base forces and torques.* For space applications, the reaction forces and torques exerted on the satellite base by the manipulator motion, are in many cases undesirable or even jeopardize mission requirements. This is the case where micro-gravity conditions have to be maintained over a longer period. Minimizing the base torques and forces is therefore a primary goal for manipulator path and trajectory design.

In order to be be more general, all these criteria can be combined and appropriately weighted by special factors $\alpha_j$

$$f \;=\; \sum \alpha_j\, f_j\,. \tag{13}$$

However, the best suited goal function has to be adapted to the respective application case, and cannot be generally determined.

*Constraint optimization: Lagrange formalism.* To solve the inverse kinematics problem, the well-known Lagrange formalism for constrained optimization is applied

$$L = f\left(\mathbf{q}\right) + \lambda^T \cdot \mathbf{h}\left(\mathbf{q}\right) = f\left(\mathbf{q}\right) + g\left(\mathbf{q}, \lambda\right), \tag{14}$$

where $L$ is the Lagrange function, $f$ is the optimization criterion, $\mathbf{h}$ accounts for the equality constraints, and $\lambda$ is a vector consisting of six Lagrange multipliers. This equation contains $n + 6$ unknowns, which can be combined by a vector $\mathbf{z}$,

$$\mathbf{z} = \left(q_1,\; q_2, \cdots, q_n,\; \lambda_1, \cdots, \lambda_6\right)^{\mathrm{T}} \tag{15}$$

giving finally $L = L(\mathbf{z})$. Minimization of $L$ necessitates that the gradient $\nabla L$ goes to zero

$$\nabla L = \nabla\left(f(\mathbf{z}) + g(\mathbf{z})\right) = \\ = \left(\frac{\partial f}{\partial q_1} + \frac{\partial g}{\partial q_1},\ \frac{\partial f}{\partial q_2} + \frac{\partial g}{\partial q_2},\ ...,\ \frac{\partial f}{\partial q_n} + \frac{\partial g}{\partial q_n},\ h_1,\ h_2,\ ...,\ h_6\right)^{\mathrm{T}} \tag{16}$$

Equation (16) is a strongly non-linear system with $n + 6$ equations and unknowns. To solve it, the method based on an iteration according to Newton-Kantorowitsch (linearization by Taylor series expansion) may be favourably used. Given the Jacobi matrix

$$\mathbf{J} = \frac{\partial \mathbf{L}'}{\partial \mathbf{z}} \tag{17}$$

where $\mathbf{L}' = \nabla L$, a linearized system of $n + 6$ equations,

$$\mathbf{J}\left(\mathbf{z}^{(k)}\right) \cdot \Delta\mathbf{z}^{(k+1)} + \mathbf{L}'\left(\mathbf{z}^{(k)}\right) = \mathbf{0}, \qquad \text{and} \qquad \mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + \Delta\mathbf{z}^{(k+1)}\,, \tag{18}$$

is to be solved. The iteration will stop as soon as the deviation $\Delta\mathbf{z}^{(k+1)}$ remains below a given error limit. As initial value of $\mathbf{z}$, i.e. $\mathbf{z}^{(0)} = \mathbf{z}(t = 0)$, the current joint positions of the last, successfully terminated iteration step is taken, together with $\lambda^{(0)} = 0$.

2.3. **Inverse kinetics.** To solve the inverse kinetics problem, i.e., to obtain the forces and torques (kinetic states) acting in the joints, the dynamics description of the entire robot system, i.e., the differential equations of motion, has to be established. Based upon a multi-body system formulation, e.g. by Newton-Euler formulation, the joint motion q$(t)$ and its time derivatives can be assigned to the applied forces and torques. Given the end effector trajectory, the joint kinematic states are found by the inverse kinematics problem, and the kinetic states are obtained by applying forward and backward calculations. The forward kinematics calculation gives the respective values required, i.e., the absolute velocities and accelerations of joints and the center of mass of each link), beginning from the inertially fixed frame up to the end effector position and orientation. Then, by backward calculation the actual forces and torques can be determined, beginning at the end effector level.

## 3. Basic example of a 7 DOF robot: Simulation results and discussion

The 7 dof (degrees of freedom) robot with fixed base of Figure 2 has been taken as a basic example to demonstrate the efficiency of the inverse kinematics and kinetics problem for redundant manipulators. The initial robot configuration is given in Figure 4 achieving a total length of about 1.25 m. The starting configuration is the one already shown in Figure 2, i.e., all joints have been moved by the same angle of $45^o$. The end effector trajectory is prescribed: in our example we chose an elliptic planar path for the position within the three-dimensional working space, where the angle of the ellipse varies in time with a fifth order polynomial (see Figure 5 for the kinematic states). The orientation of the end effector is kept constant throughout

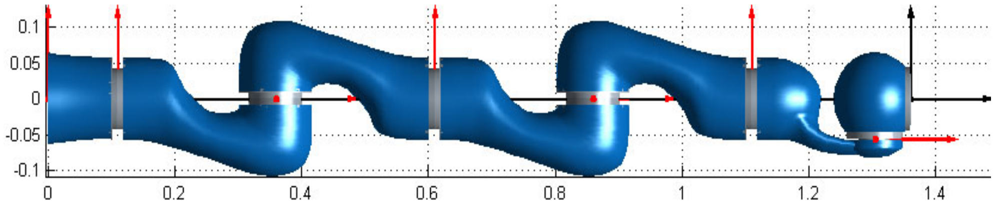


Figure 4. Initial robot configuration: 7 dof, kinematic configuration is (beginning from the fixed base) roll, pitch, roll, pitch, roll, pitch, roll; the numbers at both axes are in meters giving an overall length of about 1.25 m measured from the first roll axis

the elliptical motion. For comparison reasons, we have investigated three different optimization criteria: joint velocity, joint acceleration and configuration oriented minimization according to equations (8), (9) and (12). The two criteria based on kinetics states minimization, see equations (10) and (11), are still in the implementation phase. As the reference configuration for equation (12) we chose the starting configuration. It should be noted that in case of a non-redundant robot with 6 dof, the starting configuration of all the joints should be achieved again after terminating the elliptical motion (closed loop trajectory) of the end effector.

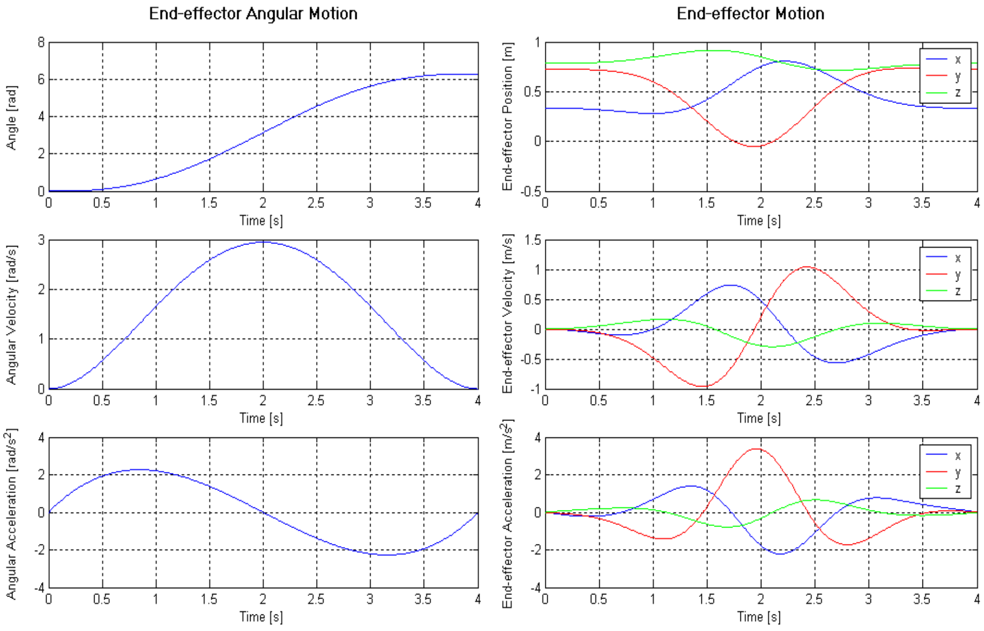


Figure 5. Kinematic states of the prescribed elliptical motion of the end effector

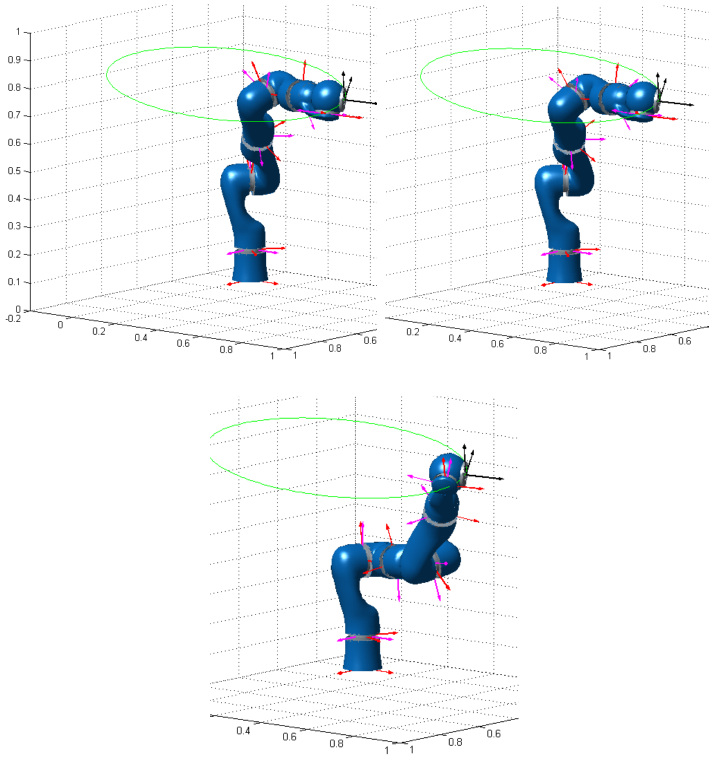


Figure 6. Final robot configuration achieved after one full elliptical loop: joint velocities optimized (left), reference configuration optimized (right), joint accelerations optimized (middle)

Figure 6 gives the comparison of the different final configurations achieved after one full elliptical loop. In the case of joint velocity and acceleration optimization we observe a great difference in the final configuration, whereas optimization of the reference, i.e. starting configuration does not exhibit a great deviation when compared to joint velocity minimization. Moreover, in the reference configuration case the final configuration is indeed identical to the initial, i.e., the reference configuration. The joint velocity optimization shows here some minor deviations from the initial configuration, whereas in joint acceleration minimization the significant joint angle deviations are surprising.

These differences in the kinematic states are presented graphically in Figure 7. Another striking behaviour in joint acceleration is to be noted: The joint velocities achieved after termination of the full elliptical trajectory are never zero. This is not surprising since the algorithm minimizes accelerations rather than velocities, although the joint accelerations achieved after trajectory termination are zero. For reasons of comparison, for each optimization criterion the respective corresponding kinematic state is presented graphically as well (Figure 8), together with the quadratic sum of the respective kinematic states.

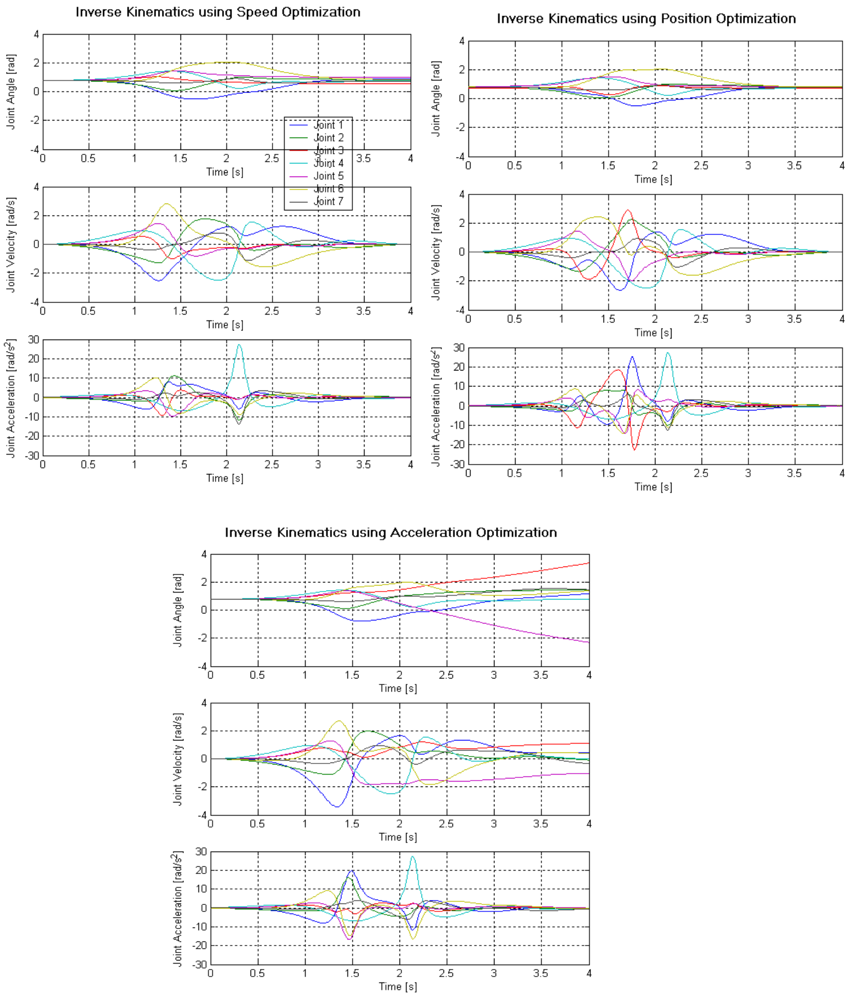


Figure 7. Kinematic states of each joint achieved by optimization

Finally, Figure 9 then shows an example of the calculated joint torque (w.r.t the joint axis) that is necessary in order to drive the respective joint. For the sake of compactness within this paper, the inertia parameters of each joint and link, which are required for the inverse kinetics calculation, are not presented within this paper.

Furthermore, instead of taking from a table the stored trajectory increments, a more common application in space robotics is to guide the robot by means of an input device, for example, a so-called 6-dof space ball. We call this mode of operations a tele-operated manipulation. This input device allows the human teleoperator to guide the robot end effector in 3 translational and in 3 rotational degrees of freedom. Again, the commanded trajectory given in some arbitrary manner by the human operator, is discretized by the space ball inherent software at a time

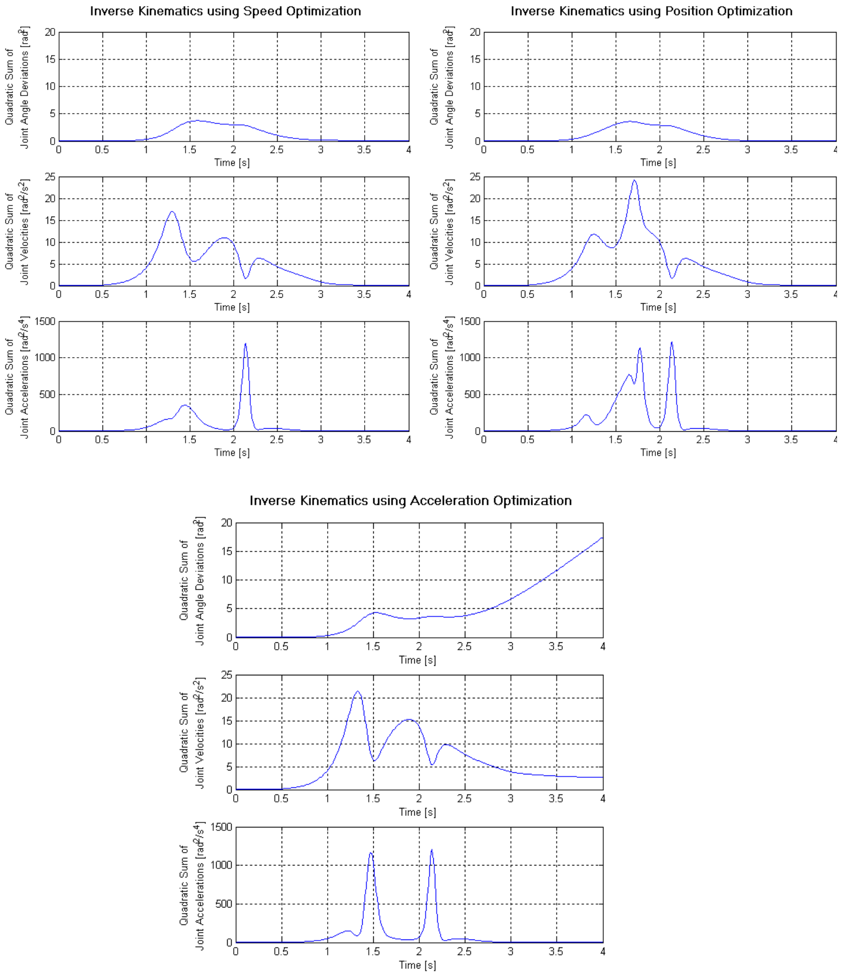
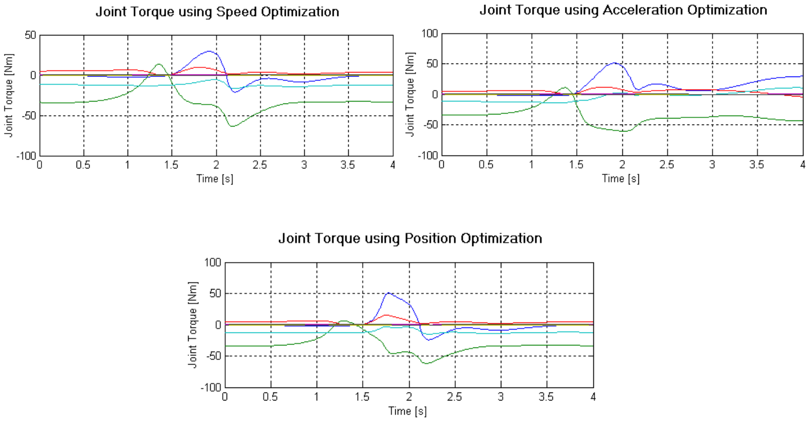


Figure 8. Quadratic sum of the respective kinematic joint states. Note, that in each optimization criterion only the respective quadratic sum at each time increment is optimal

interval equivalent or close to that of the table stored trajectory. Finally, for both kinds of robot guidance, i.e., either by a given trajectory stored in a table or by an input device based commanding via tele-operation, the underlying procedure for inverse kinematics and kinetics problem solving is highly efficient.

Final remarks have to be given to the optimization algorithm numerically implemented, and to the usefulness of the comparison of the different optimization criteria. The given trajectory is discretized by a fixed time interval of $T_s = 10$ ms. This interval is small enough to find a solution of the given inverse problem. Equivalently this means that the new prescribed end effector position and orientation does not change substantially from one time increment to the next. Otherwise, the procedure used


Joint Torque using Speed Optimization

Joint Torque using Acceleration Optimization

Joint Torque using Position Optimization

Figure 9. Calculated torques in each joint for the respective drive axis, for the different optimization criteria

here has to be modified appropriately (see specific example in Section 4), either by reducing the time interval or by changing the initial guesses for equation (18).

Concerning the comparison of results for different optimization criteria, it is found useless for two reasons. First, each optimization criterion naturally results in a configuration that is optimal in the given sense. The definition of any kind of measure with physical interpretation would be very helpful here. Second, the optimization based algorithm only gives an optimal result in terms of a new configuration that refers only to the configuration achieved in the time interval before. The reason is that the criterion is optimized only within the given time interval $T_s$. Therefore, the joint states are not optimized within the global time $T$, which covers the entire trajectory. This means, in general, that for the different optimization criteria used, the achieved new joint angles, velocities, accelerations (and torques, power, etc.) are different in nature for each time interval. Therefore, it makes no sense here to compare all criteria to each other in a sensible way.

However, a comparison may be considered sensible in the case where we let the optimization process cover the entire trajectory as a total. In that case, the optimization criteria defined in equations (8) to (12) have to be modified. For example, equations (8) then would read

$$f = \int_0^T \dot{\mathbf{q}}^T \dot{\mathbf{q}} \, \mathrm{d}t \tag{19}$$

and the optimization has to be accomplished off-line for the whole path.

Finally, all these individual optimization criteria can be combined while providing different weighting factors to each of them. Obviously, a best suited goal function cannot be given for optimization; rather it has to be adjusted to the specific application scenario. The results we obtained underline the efficiency of the algorithms being computationally very fast (1-2 ms for a complete inverse calculation within $T_s$, using a personal computer with 1 GHz performance). More than ever, this last aspect is most important in all cases where need to perform an entire robotic motion simulation in real-time.

## 4. Application: Robotics-based mission scenario

To demonstrate the procedure for the inverse kinematics and kinetics calculation of redundant manipulators for a real scenario, a rescue satellite space mission is envisaged, according to Figure 1. Here, a large malfunctioning satellite (Rosat, the German Roentgen satellite in low earth orbit) is approached by a smaller rescue satellite that is equipped with a 2 m long manipulator of 7 dof. The very complex and delicate scenario comprises final approach, manipulator deployment, and finally grasping of the defective satellite by the manipulator. Our interest here is directed towards finding a suitable path for the manipulator's end effector, while avoiding any collision with obstacles in between. The approach and grasping strategy distinguishes between two kinds of operations: the first one assumes a fixed base of the manipulator, i.e., the rescue satellite is expected not to move during manipulator motion. This can be guaranteed by means of the satellite's AOCS (attitude and orbit control system)

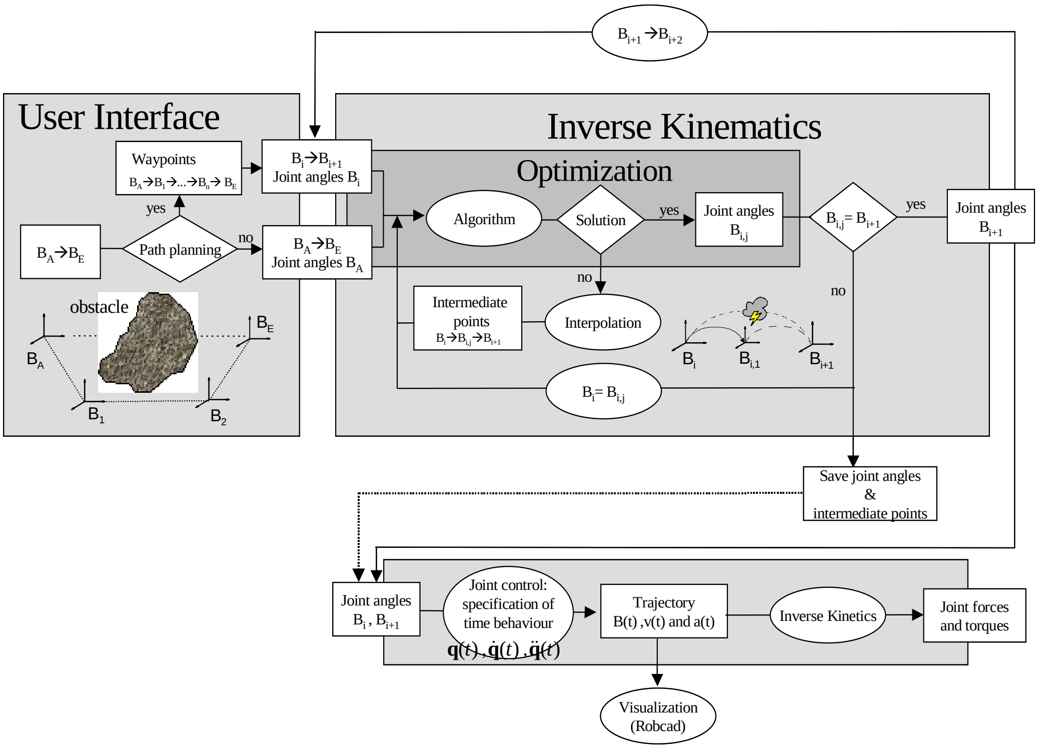


Figure 10. Structural diagram for computation of inverse kinematics and kinetics for fixed satellite base

being active. A second mode of operation supposes the rescue satellite's AOCS notto be active, leaving additional 6 degrees of freedom to the entire system, consisting of satellite and manipulator. Then our problem can be described as follows: the end effector starting point $A$ is given assigned with an initial robot configuration with all joint angles given, and represented by the matrix $\mathbf{B}_A$ equivalent to equation (4). Also, the end point $E$ to be reached is given, represented by the corresponding matrix $\mathbf{B}_E$, except for the respective joint angles that are not given. The overall task for the inverse kinematics problem then is to find the appropriate joint angles.

Applying first the procedure for the fixed satellite base, Figure 10 gives an idea how to solve this problem. Within the 'user interface' block the user defines whether to directly proceed from point $A$ to $E$ (direct path) or to define some so-called waypoints, which are appropriately chosen in between both points. The operator, by his experience or best knowledge about the entire system behaviour, may choose such points in cases where safety distances have to be maintained between manipulator and satellite parts, where obstacles have to be avoided or straight ahead of the final point to be grasped (in order to keep the manipulator in a halting mode before the final complex grasping operation is carried out). In kinematic description, these waypoints are equivalent to the final point $E$, i.e., the end effector position and orientation is represented by a given matrix $\mathbf{B}_i$, but the respective joint angles are not known.

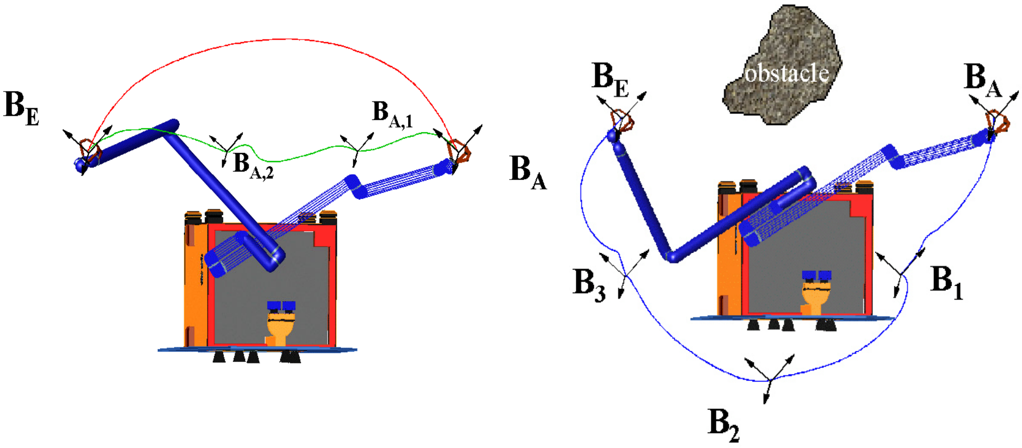


Figure 11. Left: Path and configuration for direct path (red line) or for allowing for intermediate points (green line). Right: Path and configuration for setting waypoints in between starting and end points. The 3 dimensional spatial paths are calculated using Robcad animation software package with built-in joint control algorithms. Note that although the starting configuration is the same in the left and right figures, this is not the case of the achieved final configurations, which are quite different

In general, the spatial distance between $\mathbf{B}_A$ and $\mathbf{B}_E$, or also between two neighbouring waypoints $\mathbf{B}_i$ and $\mathbf{B}_{i+1}$, is by all means long compared to the very closely spaced and discretized trajectory points of the basic example of Section 3. Furthermore, the underlying algorithm is based on linearization that is expected to give solutions more for local, closely distant points. Therefore, it is supposed that a solution is not achieved for all cases of two neighbouring waypoints a solution is achieved. This suggestion proved true during simulation runs, and a modified search strategy therefore has been set up. We decided, between two waypoints, to set so-called intermediate points $\mathbf{B}_{i,j}$ that guarantee an appropriate solution (Figure 10). The way to find such a point is to determine a search algorithm, arbitrarily chosen, that takes care of finding a suitable interpolation ratio fast and avoiding intermediate points near singularities. Therefore, between the given matrices $\mathrm{B}_i$ and $\mathrm{B}_{i+1}$ a linear interpolation is applied to find the cartesian coordinates and the Euler angles for the end effector.

Figure 11 presents the final results for the two choices, either to go directly from $\mathbf{B}_A$ to $\mathbf{B}_E$ (red path; or with intermediate points in between, green paths) or to go via waypoints $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{B}_3$ in order to avoid an obstacle for example (blue path). The path between two waypoints is not a straight line at all. In fact, it depends upon the joint control chosen, i.e., upon the specification of the time behaviour of the kinematic states $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$ and $\ddot{\mathbf{q}}(t)$. Visualization and animation of the robot motions were performed by means of the software package Robcad (manufactured by Tecnomatix), which also provides the joint kinematics time specification (constant joint acceleration with upper bound as well as an upper bound for joint velocity).

Concerning the inverse kinematics and kinetics problem for a moving satellite base, the overall dynamics behaviour of the entire system has to be regarded. Figure 12 outlines the approach that will be followed for our further investigations. Since the manipulator motion causes a corresponding movement of the base, the algorithm for the fixed base has to be expanded by a part reflecting the influence of these additional 6 dof of the base, while taking the well-known laws for momentum and impulse also into consideration.

## 5. Concluding remarks and further activities

The inverse kinematics problem for redundant space manipulators has been solved by making use of the physical meaning of the underlying optimization criteria. Moreover, this method also accounts for the joint loads and stresses in structural arm links not to exceed upper bounds. Two different examples were presented, a more convential one for basic investigations of the quality of the approach, and a realistic one that simulates a typical space mission scenario where a robot attached to a rescue satellite is intended to grasp a second, malfunctioning satellite. The algorithms applied to both examples proved very efficient and they are computationally extremely fast. More than ever, the last aspect is most important in all cases where we have the need to perform an entire robotic motion simulation in real-time.

Should the inverse kinetics computation show that the joint torques and forces obtained are not below certain limits, then the robot design has to start again. This means that especially the mechanical sub-components as motors, gears, and link structures, have to be changed according to a CAD based data panel, for example. This

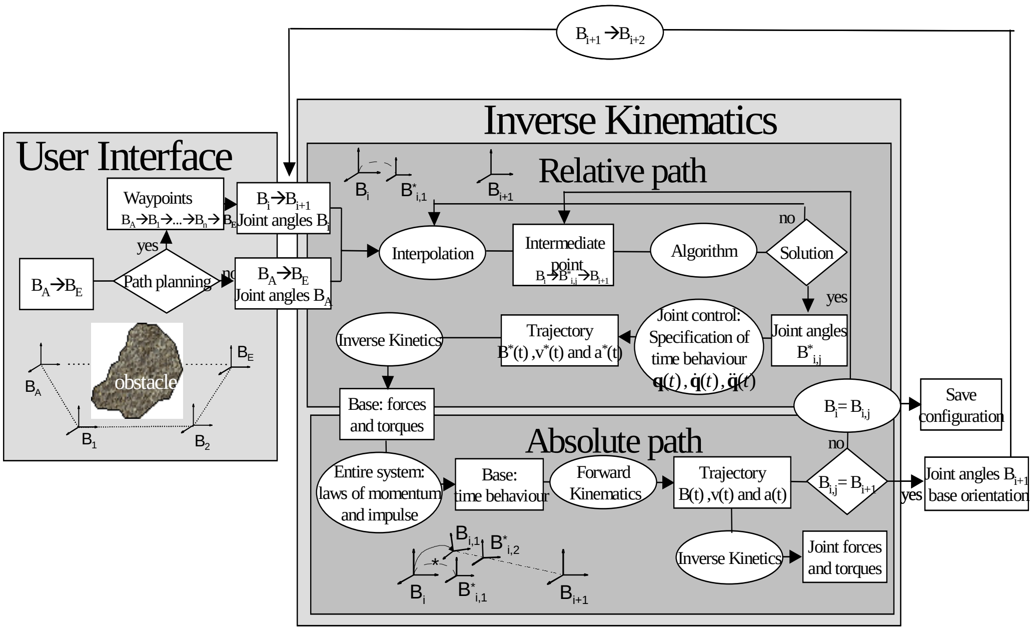


Figure 12. Structural diagram for computation of inverse kinematics and kinetics for moving satellite base

starts the inverse kinematics and kinetics problem solving again. However, there are still more problems to be solved in further investigations. One refers to the moving satellite base, where we suggested an approach for the inverse problems. Another important item concerns an optimal trajectory planning, which we have not addressed here, but mentioned one possible approach as given by equation (19). Moreover, the addressed optimization criteria based on kinetics states minimization (e.g. joint torque and power) will be followed further.

## REFERENCES

1. Schäfer, B., Krenn, R., Hirzinger, G. and da Silva, A.R.: *Light-weight space robotics: rapid design approach and efficient simulation environment*, MIROC Machine Intelligence and Robotic Control, **3**(3), (2001), 99-111.
2. Krenn, R. and Schäfer, B.: *Dynamics simulation and assembly environment for rapid manipulator design*, GAMM Scientific Annual Meeting, Augsburg, Germany, 25-28 March 2002.
3. Rebele, B., Schäfer, B. and Borrell Lopez-Doriga, D.: *Dynamics aspects in satellite capturing by robotic manipulator*, GAMM Scientific Annual Meeting, Augsburg, Germany, 25-28 March 2002.
4. Weiss, H.: *Genetic Algorithms and Optimum Robot Design*, DLR Internal Report, October 2001.

5. Spong, M.W. and Vidyasagar, M.: *Robot Dynamics and Control*, John Wiley, New York, 1989.