

## Trajectory Optimization of a Shuttle Mounted Robot

M. Glocker, M. Vögel, O. von Stryk

Lehrstuhl M2 Höhere Mathematik und Numerische Mathematik  
Technische Universität München, <http://www-m2.ma.tum.de>

### Abstract

Optimal path planning for a shuttle mounted remote manipulator system is investigated. Shuttle, robot and load are modelled as a multi-body system of rigid bodies whereby computer-aided modelling is used to derive the equations of motion. Minimum time and minimum energy trajectories of the robot are investigated which re-establish the shuttle's attitude at the beginning of the maneuver also at the end. The resulting constrained optimal control problems are solved by a recent version of the optimal control software DIRCOL [12] which is based on discretization of state and control variables, collocation and nonlinear programming (NLP). An improvement of internal gradient approximations, the full utilization of the discretized problem structure as well as of the sparsity of the NLP Jacobian are important factors improving the efficiency and robustness of the method. The resulting NLPs are solved by a recent version of the large-scale SQP method SNOPT [5]. A three-dimensional animation of the space robot trajectories is obtained using the graphics library SIGMA [1].

### Introduction

With respect to the present construction of an international space station, the investigation of space based robotics is an interesting subject. At the moment a lot of work has to be done by extra vehicular activity or space walks of astronauts. As these activities can also be quite dangerous the possibility of having them performed by robots becomes a valuable alternative. A manipulator arm of a space-shuttle is an example for such a robot.

When manipulating loads in space, one of the most important problems is that the manipulator base cannot be treated as inertially fixed. Thus, the shuttle will rotate and translate during robot motions. Hereby the linear shift can simply be evaluated from the angles of the manipulator joints at initial and final time, since the center of mass of the whole system does not move. In contrast to this, the final attitude is a function of the entire history of the joint angles.

A solution to this problem is to use the shuttle's attitude control system in order to handle the shuttle's rotation. But this would imply an unnecessary waste of the shuttle's limited attitude control fuel. Another, more advanced procedure is to include the re-establishment of the shuttle's attitude in the path planning [9]. Therefore, the initial and the desired final attitude are imposed as boundary constraints to the robot trajectory planning problem. Furthermore, the path can be optimized with respect to various performance indices such as time or energy.

The first part of the present paper introduces a dynamic model of the shuttle mounted robot. Optimal control problems for optimal trajectory planning are formulated next. The optimal control problems are solved numerically using a direct collocation method. Finally, the obtained solution is discussed and visualized.

## Formulation of the Optimal Path Planning Problem

### Dynamic Robot Model

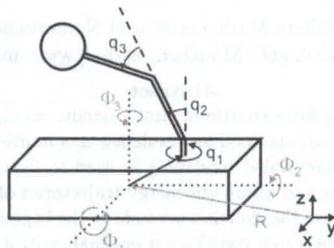


Figure 1: Kinematic model and state variables of the robotic manipulator.

For the dynamic model the following basic assumptions are made:

1. gravitational effects are neglected,
2. the system rests at initial time,
3. the robot arms and the load are three-dimensional, rigid bodies, and
4. the shuttle's attitude control is turned off.

These assumptions are basically the same as in [8, 9] besides the third one which goes significantly further. In [8] massless robot links and a point-mass load are considered. In our approach, the arms and the load are more realistically modelled as rigid bodies, i. e. with non-zero volumes [10]. This means that each of the robot arms and the load are described by a nonzero mass and a nonzero inertia tensor. The moments of inertia are computed based on the geometrical shape and data given in the following table:

	geometry	mass	data
shuttle	cuboid	104325 kg	$37.19 \times 8 \times 6.5 \text{ m}^3$
1. arm	cylinder	18.9 kg	$l = 0.6 \text{ m}, r = 0.2 \text{ m}$
2. & 3. arm	cylinder	200 kg	$l = 7.12 \text{ m}, r = 0.2 \text{ m}$
satellite	sphere	2500 kg	$r = 2 \text{ m}$

The first arm is mounted about 4.3 m away from the shuttle's center of gravity. In this mounting point there is a rotational joint denoted by the angle  $q_1$ . The shoulder and elbow joints are denoted by the angles  $q_2$  and  $q_3$ . Each joint is equipped with a motor. The torques, respectively the controls of the system, are denoted by  $(u_1, u_2, u_3)^T$ . Hereby the indices of the torques are corresponding to those of the joints. For the shuttle's attitude the angles  $(\phi_1, \phi_2, \phi_3)^T$  are used. The kinematic robot model and the state variables are depicted in Figure 1 where the vector  $R$  denotes the position of the center of gravity of the shuttle. The resulting equations of motion for the robot's joint angles and the shuttle's attitude angles are of the form

$$M(\phi, q) \begin{pmatrix} \ddot{\phi} \\ \ddot{q} \end{pmatrix} = b(\phi, q, \dot{\phi}, \dot{q}) + \begin{pmatrix} 0 \\ u \end{pmatrix}.$$

Here  $M \in \mathbb{R}^{6 \times 6}$  is a symmetric positive definite matrix representing the mass matrix. The data of the robotic manipulator system of the European Hermes project using the computer-aided dynamical equations have been used. The initial and final positions

### Optimal Control Problem

Two objectives for optimal path planning are considered:

For application of the minimum principle, the initial and final positions and times are given by

$$q(0) = q_0, \quad q(T) = q_f,$$

To return the shuttle to its initial position and attitude

is demanded. To ensure that the shuttle's attitude is controlled

must hold. To define and compute the optimal path is limited by their maximum values.

This is essential for minimizing the cost function in the Hamiltonian. It can be shown that the controls are constrained. The joint angles

This is necessary in order to avoid collisions between the shuttle and the satellite. The limitation of the joint angles

Here  $M \in \mathbb{R}^{6 \times 6}$  is a symmetric, positive definite inertia matrix and  $b \in \mathbb{R}^6$  a vector of gyroscopic accelerations.

The data of the robotic manipulator has been derived from data published for the remote manipulator system of the U.S. space shuttle and for the robotic manipulator of the European Hermes project [10]. The equations of motion have been derived using the computer-aided multibody systems simulation program SIMPACK [7]. The dynamical equations have not been derived in explicit form but as a subroutine that can be linked to the optimization method described further on.

### Optimal Control Problem

Two objectives for optimal robot trajectories are investigated

$$\text{energy } J_e[u] = \min \int_0^{t_f} \sum_{i=1}^3 (u_i(t))^2 dt \quad (2)$$

$$\text{and time } J_t[u] = \min t_f. \quad (3)$$

For application of the minimum energy objective, the final time  $t_f$  must be prescribed. Otherwise, a free final time may tend to infinity (cf. Theorem 8.1 in [11], pp. 95-96). The initial and final position of the robotic manipulator is given by the joint angles

$$q(0) = (0.9, 0.4683, 1.9)^T, \quad q(t_f) = (0, 0.994, 1.153)^T. \quad (4)$$

To return the shuttle to its original attitude

$$\phi(0) = 0, \quad \phi(t_f) = 0 \quad (5)$$

is demanded. To ensure that the system rests at initial and final time

$$\begin{aligned} \dot{\phi}(0) &= 0, & \dot{\phi}(t_f) &= 0 \\ \dot{q}(0) &= 0, & \dot{q}(t_f) &= 0 \end{aligned}$$

must hold. To define and compute a time optimal trajectory the controls have to be limited by their maximum possible motor torques

$$|u_i| \leq u_{i,\max}, \quad i = 1, 2, 3 \quad (6)$$

This is essential for minimum time trajectories because the controls appear linearly in the Hamiltonian. It can easily be shown that any time at least one motor will operate at its maximum capacity as long as the corresponding state variables are not constrained. The joint angles of the manipulator are limited as well by

$$\begin{pmatrix} -\pi \\ -0.2 \\ -3 \end{pmatrix} = q_{\min} \leq q \leq q_{\max} = \begin{pmatrix} \pi \\ 1.5 \\ 3 \end{pmatrix}. \quad (7)$$

This is necessary in order to avoid collisions among the manipulator arms themselves or between them and the shuttle. However, not all collisions can be excluded only by the limitation of the joint angles. Therefore, further state constraints must be

considered. Here, only collisions between the satellite and the left side of the shuttles bay are examined because of the special initial and final positions. The latter collision avoidance can be ensured by the non-negativity of a distance function

$$\Delta(W, p_{cs}) \geq 0 \quad (8)$$

with

$$\Delta(W, p_{cs}) = \begin{cases} \min \{\|w - p_{cs}\|_2 : w \in W\} - r, & \text{if } p_{cs} \notin W \\ -\min \{\|w - p_{cs}\|_2 : w \in \partial W\} - r, & \text{if } p_{cs} \in W. \end{cases} \quad (9)$$

Hereby the set  $W := \{x \in \mathbb{R}^3 : 2.3 < x_1 < 2.5, x_3 < 0.3\}$  approximately describes the left side of the shuttle's bay, and  $p_{cs}$  is the center of a sphere with radius  $r$  which contains the whole satellite, i. e., the load.

As long as the sphere does not penetrate the set  $W$ , the Euclidian distance between the surfaces of the sphere and the bay wall is used. Otherwise  $\Delta$  takes a negative value related to the magnitude of penetration. The function  $\Delta$  is greater or equal to zero as long as a safety distance is kept. It is also continuous and piecewise differentiable and is thus suited as a nonlinear collision avoidance constraint for numerical solution with the direct collocation method DIRCOL described in the sequel.

## Numerical Solution Method

### Direct Collocation and Sparse Nonlinear Programming

Consider the general optimal control problem

$$\min J[u, p, t_f] = \phi(x(t_f), t_f) + \int_0^{t_f} L(x(t), u(t), p, t) dt \quad (10)$$

$$\text{unter } \dot{x} = f(x(t), u(t), p, t) \quad (11)$$

$$0 \leq g(x(t), u(t), p, t) \quad (12)$$

$$0 = h(x(t), u(t), p, t) \quad (13)$$

$$\begin{aligned} x : [0, t_f] &\rightarrow \mathbb{R}^{n_x} && \text{vector of state variables} \\ u : [0, t_f] &\rightarrow \mathbb{R}^{n_u} && \text{vector of control variables} \\ p \in \mathbb{R}^{n_p} &&& \text{vector of control parameters} \\ t_f &&& \text{free or given final time.} \end{aligned} \quad (14)$$

Furthermore, explicit or implicit, linear or nonlinear boundary conditions may be imposed on the state and control variables at initial time  $t = 0$ , final time  $t_f$ , or switching times  $t_{s_i}$ . The nonlinear algebraic constraint  $h = 0$  allows the treatment of semi-explicit differential-algebraic equations of index 1.

In a direct collocation approach, the state and control variables are approximated by elements of a finite dimensional function space [2, 13]. Here, piecewise polynomials have been chosen. The time interval  $[0, t_f]$  is divided into  $m \geq 1$  phases. Each phase is then subdivided by  $n_i + 1$  grid points,  $i = 1, \dots, m$  (Figure 2). In each phase, the control variables are approximated by continuous, piecewise linear polynomials, and the state variables by continuously differentiable, piecewise cubic polynomials (Figure 3).

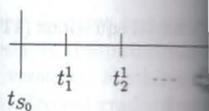


Figure 2

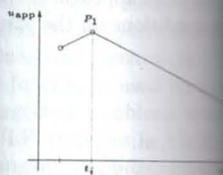


Figure 3: Appr

Figure 3: Appr

At switching points  $t_{s_i} = t_{s_i}^k$ , the control variables or the state variables are approximated by piecewise linear polynomials

$$u_{j,app}(t) = u_j(t_k^i) + (t - t_k^i) \cdot \dots$$

with  $j = 1, \dots, n_u$ , and the control variables  $x_{app}$  by

$$x_{j,app}(t) = \sum_{k=0}^{n_i} \dots$$

$$c_{0,k} = \dots$$

$$c_{1,k} = \dots$$

$$c_{2,k} = -3x_j(t_k^i) \dots$$

$$c_{3,k} = 2x_j(t_k^i) \dots$$

with  $j = 1, \dots, n_x$ . Hereby the

$$f_{j,k} := f_j(x(t_k^i), u(t_k^i), p, t_k^i) \dots$$

is used. On the basis of this approximation, the problem is transformed into a nonlinear optimization problem. In summary, the parameters of the optimization problem consist of the values of the control variables together with the control parameters

$$Y = (p_1, \dots, p_{n_p}, \dots)$$

$$u(t_0^i), \dots, u(t_0^m)$$

$$u(t_0^0), \dots, u(t_0^m)$$

$$\text{with } n_Y = (n_u + n_x + n_p) \dots$$

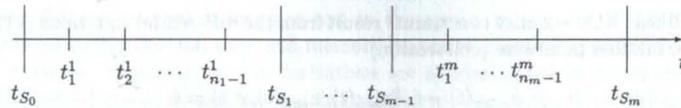


Figure 2: Discretization of the time interval.

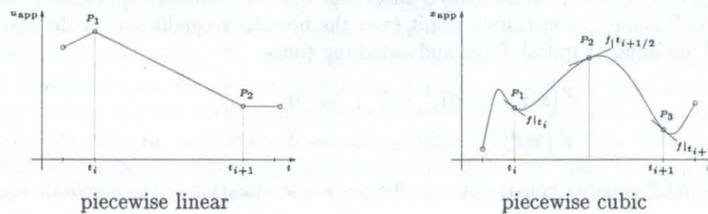


Figure 3: Approximation of control and state variables.

At switching points  $t_{S_i} = t_{n_i}^i = t_0^{i+1}$ ,  $i = 1, \dots, m - 1$ , discontinuities of the control variables or the state variables may be included in the discretization. The piecewise linear approximations  $u_{app}$  can be derived by the formula

$$u_{j,app}(t) = u_j(t_k^i) + (t - t_k^i)/(t_{k+1}^i - t_k^i) (u_j(t_{k+1}^i) - u_j(t_k^i)), \quad t \in [t_k^i, t_{k+1}^i]$$

with  $j = 1, \dots, n_u$ , and the continuously differentiable, piecewise cubic approximations  $x_{app}$  by

$$x_{j,app}(t) = \sum_{l=0}^3 c_{l,k} ((t - t_k^i)/(t_{k+1}^i - t_k^i))^l, \quad t \in [t_k^i, t_{k+1}^i] \quad (15)$$

$$\begin{aligned} c_{0,k} &= x_j(t_k^i) \\ c_{1,k} &= h_k f_{j,k} \\ c_{2,k} &= -3x_j(t_k^i) - 2h_k f_{j,k} + 3x_j(t_{k+1}^i) - h_k f_{j,k+1} \\ c_{3,k} &= 2x_j(t_k^i) + h_k f_{j,k} - 2x_j(t_{k+1}^i) + h_k f_{j,k+1} \end{aligned} \quad (16)$$

with  $j = 1, \dots, n_x$ . Hereby the abbreviation

$$f_{j,k} := f_j(x(t_k^i), u(t_k^i), p, t_k^i), \quad h_k := t_{k+1}^i - t_k^i$$

$$k = 0, \dots, n_i - 1, \quad i = 1, \dots, m$$

is used. On the basis of this approximation the optimal control problem is transformed into a nonlinear optimization problem, which is not convex in general.

In summary, the parameters  $Y$  of the resulting nonlinearly constrained optimization problem consist of the values of the state and control variables at the grid points, together with the control parameters, the final time and possible switching points

$$Y = (p_1, \dots, p_{n_p}, u(t_0^1), \dots, u(t_{n_1}^1), x(t_0^1), \dots, x(t_{n_1}^1), t_{S_1}, \dots, u(t_0^m), \dots, u(t_{n_m}^m), x(t_0^m), \dots, x(t_{n_m}^m), t_{S_m})^T \in \mathbb{R}^{n_Y}, \quad (17)$$

$$\text{with } n_Y = (n_u + n_x) \sum_{i=1}^m (n_i + 1) + n_p + m. \quad (18)$$

The nonlinear NLP equality constraints result from the differential equations (11) which are to be fulfilled pointwise (collocation)

$$\dot{x}_{j,\text{app}}(t) - f_j(x_{\text{app}}(t), u_{\text{app}}(t), p, t) = 0 \quad (19)$$

for  $t = 1/2(t_{k-1}^i + t_k^i)$ ,  $k = 1, \dots, n_i$ ,  $i = 1, \dots, m$ , and at  $t_{k-1}^i$  and  $t_k^i$ . The latter are implicitly fulfilled by the chosen ansatz for the state variable approximation (16). More NLP equality constraints result from the boundary conditions on the state and control variables at initial, final, and switching times

$$r^1(x(t_0^1), t_0^1, x(t_{n_m}^m), t_{n_m}^m) = 0 \quad (20)$$

$$r^i(x(t_{n_i-1}^i), x(t_0^i), t_{S_i-1}^i) = 0, \quad i = 2, \dots, m. \quad (21)$$

Further NLP equality constraints result from a discretization of the algebraic equation

$$h(x_{\text{app}}(t_k^i), u_{\text{app}}(t_k^i), p, t_k^i) = 0 \quad (22)$$

for  $k = 0, \dots, n_i$ ,  $i = 1, \dots, m$ . The resulting nonlinear inequality constraints are

$$g(x_{\text{app}}(t_k^i), u_{\text{app}}(t_k^i), p, t_k^i) \geq 0 \quad (23)$$

for  $k = 0, \dots, n_i$ ,  $i = 1, \dots, m$ . Finally, every variable must fulfill a box constraint

$$\begin{aligned} x_{j,\min}^i &\leq x_j(t_k^i) \leq x_{j,\max}^i, & j &= 1, \dots, n_x \\ u_{l,\min}^i &\leq u_l(t_k^i) \leq u_{l,\max}^i, & l &= 1, \dots, n_u \\ & & k &= 0, \dots, n_i, \quad i = 1, \dots, m \\ p_{l,\min} &\leq p \leq p_{l,\max}, & l &= 1, \dots, n_p \\ t_{S_i,\min} &\leq t_{S_i} \leq t_{S_i,\max}, & i &= 1, \dots, m. \end{aligned} \quad (24)$$

All in all, the nonlinear optimization problem (NLP)

$$\begin{aligned} &\text{minimize} && \Phi(Y), && Y \in \mathbb{R}^{n_Y} \\ &\text{subject to} && Y_{\min} \leq Y \leq Y_{\max}, \\ & && a(Y) = 0, \\ & && b(Y) \geq 0 \end{aligned} \quad (25)$$

is obtained. Typical properties of NLPs of discretized optimal control problems are:

- The NLPs are of large-scale with very many variables and very many constraints.
- Many of the NLP constraints are active at the solution, e.g., the equality constraints from collocation (19). Thus, the number of "free" variables of the NLP is much smaller than the total number of NLP variables  $n_Y$ .
- The objective of the optimal control problem can without loss of generality be assumed to be of Mayer type [12]. Then the NLP objective only depends on a few of the NLP variables whose number is independent of the number of grid points.
- The NLP Jacobians are sparse and structured.

All these features are that reduces computational earlier versions. Further lytically derived gradient robustness.

The resulting NLP programming (SQP) method direction from a quadratic the linearized constraint merit function is used for large-scale problems the NLP variables in the Lagrangian is approximated Hessian algorithm is used the current working set

### Local Accuracy

A user of DIRCOL does Lagrange equations or the adjoint variables  $\lambda(t)$  are provided by DIRCOL NLP [11, 12]. Thus, the necessary conditions these estimates also local the objective can be computed Let  $\Phi$  be the "true" minimized discretized problem. Then

An estimate  $\tilde{\omega}$  of  $\omega$  can be obtained interval [11]

Let  $\tilde{\omega}$  denote the minimum interval  $[t_k^i, t_{k+1}^i]$  is now

$$\begin{aligned} \tilde{\omega}_{i,k} &= \sum_{j=1}^{n_x} \int_{t_k^i}^{t_{k+1}^i} \tilde{\lambda}_j(t) dt \\ &+ \sum_{j=1}^{n_u} \int_{t_k^i}^{t_{k+1}^i} \tilde{u}_j(t) dt \\ &+ \sum_{j=1}^{n_h} \int_{t_k^i}^{t_{k+1}^i} \tilde{p}_j(t) dt \end{aligned}$$

In DIRCOL [12] the integration is done over the absolute error is approximated

All these features are taken into account in the recent version 2.1 of DIRCOL [12] that reduces computational time and memory resources drastically when compared to earlier versions. Furthermore, the derivatives are approximated by a mixture of analytically derived gradients and tailored finite difference approximations which improves robustness.

The resulting NLP problems are solved by the large-scale sequential quadratic programming (SQP) method SNOPT [5]. SQP methods are based on evaluating the search direction from a quadratic approximation of the Hessian of the Lagrangian subject to the linearized constraints. To ensure global convergence, an augmented Lagrangian merit function is used for computation of the step size. SNOPT is especially efficient for large-scale problems with many active constraints [4]. It is based on partitioning the NLP variables in basic, superbasic and nonbasic variables. The Hessian of the Lagrangian is approximated by limited-memory quasi-Newton updates and a reduced Hessian algorithm is used for solving the QP subproblems. The null-space matrix of the current working set matrix is obtained from a sparse LU factorization.

### Local Accuracy Checks and Grid Refinement

A user of DIRCOL doesn't have to deal with adjoint differential equations, Euler-Lagrange equations or the Maximum Principle. Nevertheless, an estimation of the adjoint variables  $\lambda(t)$  and multipliers  $\eta(t)$  of equality and  $\nu(t)$  of inequality constraints are provided by DIRCOL which are obtained from the Lagrange multipliers of the NLP [11, 12]. Thus, additional checks of the computed solution with consistency of the necessary conditions resulting from the Maximum Principle are enabled. Based on these estimates also local optimality error estimates and an absolute error estimate of the objective can be computed.

Let  $\Phi$  be the "true" minimum objective value and  $\tilde{\Phi}$  the computed value for the discretized problem. Then the absolute error of the objective is given by

$$\omega = \Phi - \tilde{\Phi}. \quad (26)$$

An estimate  $\tilde{\omega}$  of  $\omega$  can be obtained by a sum of local error estimates in each discretization interval [11]

$$\tilde{\omega} = \sum_{i=1}^m \sum_{k=0}^{n_i-1} \tilde{\omega}_{i,k}. \quad (27)$$

Let  $\tilde{\cdot}$  denote the numerically computed solution. The local optimality error in the interval  $[t_k^i, t_{k+1}^i]$  is now estimated by

$$\begin{aligned} \tilde{\omega}_{i,k} = & \sum_{j=1}^{n_g} \int_{t_k^i}^{t_{k+1}^i} \tilde{\lambda}_j(t) (\dot{\tilde{x}}_j(t) - \mathbf{f}_j^i(\tilde{x}(t), \tilde{u}(t), \tilde{p}, t)) dt \\ & + \sum_{j=1}^{n_g} \int_{t_k^i}^{t_{k+1}^i} \tilde{\eta}_j(t) \mathbf{g}_j^i(\tilde{x}(t), \tilde{u}(t), \tilde{p}, t) dt \\ & + \sum_{j=1}^{n_h} \int_{t_k^i}^{t_{k+1}^i} \tilde{\nu}_j(t) \mathbf{h}_j^i(\tilde{x}(t), \tilde{u}(t), \tilde{p}, t) dt, \quad k = 0, \dots, n_i - 1, \quad i = 1, \dots, m. \end{aligned} \quad (28)$$

In DIRCOL [12] the integrals of (28) are approximated by trapezoidal sums. If the integration is done over the absolute values of the integrands, an upper bound of the absolute error is approximated.

Several other local error estimators are also implemented. Due to the approximation of the state variables, the differential equations are satisfied precisely only at the grid points and the centers in between. An absolute error function  $d(t)$  of the differential equations inside the discretization interval  $[t_k^i, t_{k+1}^i]$  is defined by a piecewise linear function at the points

$$t_{k+l/4}^i = t_k^i + l/4(t_{k+1}^i - t_k^i), \quad l = 0, \dots, 4 \quad (29)$$

using the formula

$$d(t_{k+l/4}^i) := \max \left\{ \left| \dot{x}_{j,\text{app}}(t) - \dot{f}_j^i(x_{\text{app}}(t), u_{\text{app}}(t), p_{\text{app}}(t)) \right| \text{ at } t = t_{k+l/4}^i, j = 1, \dots, n_x \right\}, \\ l = 0, \dots, 4, \quad k = 0, \dots, n_i - 1, \quad i = 1, \dots, m. \quad (30)$$

At the same check points of Equation (29) the violations of the inequality constraints are computed.

The above mentioned local accuracy checks are used to automatically refine the discretization grid by DIRCOL. Intervals with local error estimates larger than given tolerances are splitted up by inserting new grid points. The resulting new NLP is then solved starting from the previous solution of the coarser grid.

## Results

problem no.	$q_3(t_0)$	DIRCOL-1.2 with NPSOL	DIRCOL-2.1 with SNOPT	speed-up factor
1	2.205	3055 s	65 s	47
2	2.204	2841 s	107 s	27
3	2.202	2827 s	79 s	36
4	2.150	3072 s	182 s	17
5	2.100	3228 s	97 s	33
6	2.050	3573 s	100 s	36
7	2.000	4076 s	121 s	34
8	1.950	3001 s	27 s	111
9	1.930	2536 s	7 s	363
10	1.925	4768 s	17 s	280
11	1.900	3604 s	24 s	150
sum:		36581 s	826 s	44

Table 1: Comparison of the computing times of the older version 1.2 of DIRCOL (with the standard SQP method NPSOL-5.0-2) and the recent version 2.1 (with the large-scale SQP method SNOPT 5.3-5) for a family of minimum energy robot trajectory problems with different initial positions and given final time  $t_f = 100$  s. A discretization by  $n_1 = 24$  grid points yields NLP dimensions (25) of  $n_y = 363$ ,  $n_a = 299$ ,  $n_b = 24$ . The percentage of nonzero elements of the NLP Jacobian is 8.5%. The computing times refer to an SGI Indigo<sup>2</sup> workstation with a MIPS R4400/150 MHz processor which is about seven times slower than a PC with a recent Pentium II/500 MHz processor.

The semi-implicit second order system (1) is formally transformed into an explicit first order system (11) by  $x := (\phi, \dot{\phi}, q, \dot{q})^T$ , i. e.,  $n_x = 12$ ,  $n_u = 3$ , in order to apply DIRCOL.

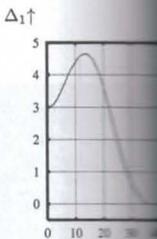


Figure 4: Histogram of collision avoidance.

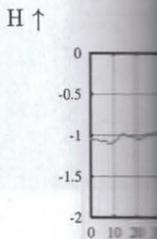


Figure 5: Estimate of minimum time.

A minimum energy trajectory is computed by a homotopy. Figure 6 shows the computed structures as it is likely to appear linearly in the history of the non-linear problem. A comparison of the performance of version 1.2 that doesn't use the speed-up of 44 for 24 grid points. A three-dimensional vis

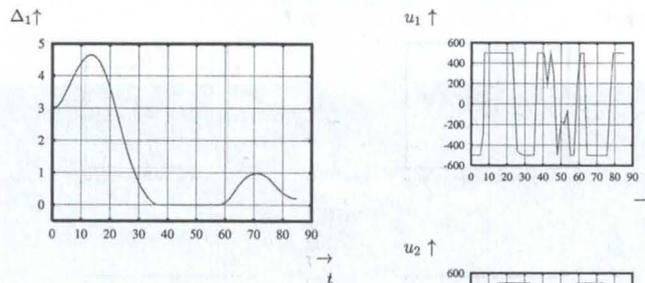


Figure 4: History of the nonlinear collision avoidance constraint.

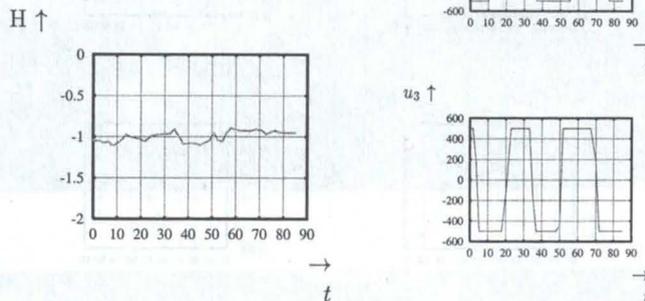


Figure 5: Estimated history of the minimum time Hamiltonian.

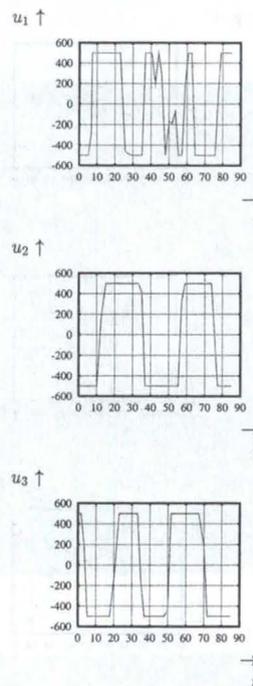


Figure 6: Minimum time controls.

A minimum energy trajectory for a fixed final time  $t_f = 100$  s is computed first. The minimum time trajectory with  $t_f = 83.77$  s is computed from this initial trajectory by a homotopy. Figure 5 shows an approximation of the history of the Hamiltonian computed from the approximations of state (Figure 7), control (Figure 6), adjoint and multiplier variables provided by DIRCOL for a refined grid of 54 non equidistant grid points. The problem is autonomous. Thus, the Hamiltonian must be (piecewise) constant. Furthermore, it must be equal to  $-1$  at  $t_f$  for the minimum time trajectory. The history of the nonlinear state constraint  $\Delta \geq 0$  for collision avoidance is depicted in Figure 4. It becomes active at its lower bound approximately within [36 s, 59 s].

Figure 6 shows the computed minimum time controls. They clearly exhibit bang-bang structures as it is likely for a minimum time optimal control problem when the controls appear linearly in the Hamiltonian. Furthermore,  $u_1$  exhibits a singular behavior along the active subarc of the nonlinear state constraint  $\Delta$ .

A comparison of the performance of the recent version 2.1 of DIRCOL with the older version 1.2 that doesn't utilize the problem structure is given in Table 1. The average speed-up of 44 for 24 grid points becomes even larger as the number of grid points is increased.

A three-dimensional visualization of the minimum time trajectory is displayed in Fig-

the approximation  
only at the grid  
of the differential  
a piecewise linear

(29)

$j = 1, \dots, n_x$ ,

(30)

quality constraints

ally refine the dis-

larger than given

g new NLP is then

speed-up  
factor

47

27

36

17

33

36

34

111

363

280

150

44

on 1.2 of DIRCOL  
version 2.1 (with the  
energy robot trajectory  
0 s. A discretization  
 $n = 299$ ,  $n_b = 24$ . The  
the computing times  
processor which is  
MHz processor.

into an explicit first  
er to apply DIRCOL.

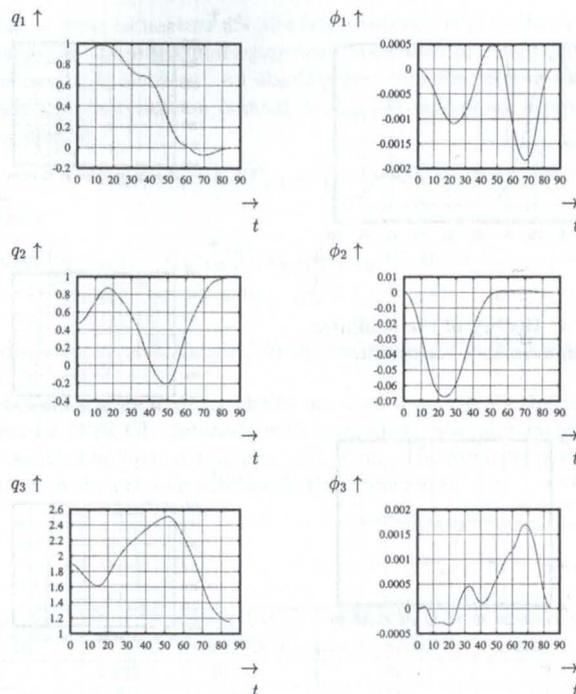


Figure 7: State variables of the minimum time trajectory.

ure 8 which has been obtained using the graphics library SIGMA [1]. SIGMA offers the possibility for photo-realistic visualization of material and surface structures and can also be used for animations.

## Summary

Optimal path planning for a shuttle mounted remote manipulator system is investigated. Equations of motion are derived using computer-aided modelling with SIMPACK where robot arms and load are modelled as three-dimensional, rigid bodies. Constraints are considered that avoid collisions between the robot arms and the shuttle and that re-establish the shuttle's attitude at the beginning of the maneuver also at the end. The resulting optimal control problems are solved by a direct collocation discretization and nonlinear programming. A recent version of DIRCOL takes into account the structure of the discretized problems and utilizes the sparse SQP method SNOPT for solving the resulting large-scale, sparse and structured nonlinear programming problems. The new method which utilizes much of the problem structure is orders of magnitude faster than the previous method without any loss of robustness. Typical minimum time and minimum energy trajectories are discussed in detail.

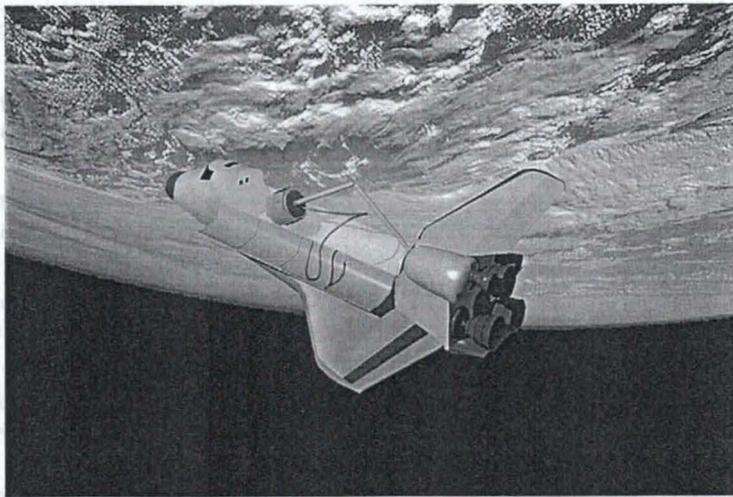
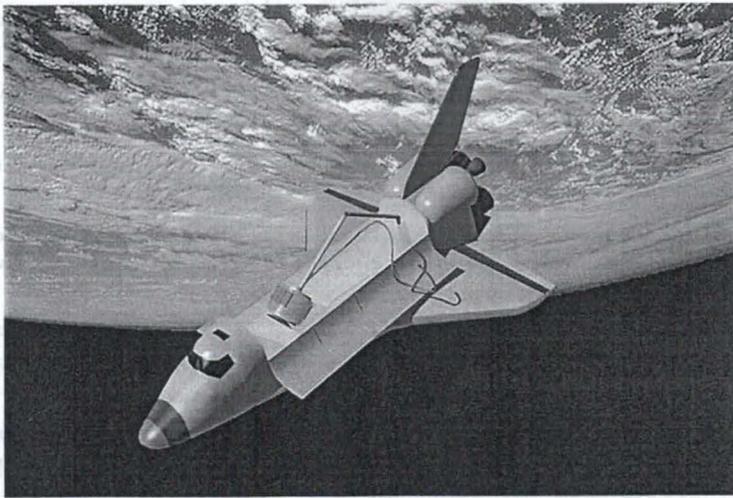


Figure 8: Visualization of the minimum time trajectory.

## References

- [1] M. Alefeld, J. Haber, A. Heim: *A new 3D graphics library: concepts, implementation and examples*, in: H.-C. Hege, K. Polthier (eds.): *Visualization and Mathematics*, pp. 211–225, Springer, Berlin, 1997.
- [2] J.T. Betts: *Survey of numerical methods for trajectory optimization*, AIAA J. Guidance, Control, and Dynamics **21**, 2, pp. 193–207, 1998.
- [3] R. Bulirsch, D. Kraft (eds.): *Computational Optimal Control*, International Series of Numerical Mathematics **115**, Birkhäuser, Basel, 1994.
- [4] P.E. Gill, W. Murray, M.A. Saunders: *SNOPT: An SQP algorithm for large-scale constrained optimization*, Report NA 97-2, Department of Mathematics, University of California, San Diego, 1997.
- [5] P.E. Gill, W. Murray, M.A. Saunders: *User's guide for SNOPT 5.3-5: a Fortran package for large-scale nonlinear programming*, Department of Mathematics, University of California, San Diego, 1999.
- [6] M. Glocker: *Berechnung und Visualisierung optimaler Trajektorien für einen Roboter im Weltraum*, Fortgeschrittenenpraktikum, Zentrum Mathematik, Technische Universität München, 1998.
- [7] Intec GmbH: *SIMPACK*, World Wide Web: <http://www.simpack.de/>
- [8] V.H. Schulz: *Reduced SQP methods for large-scale optimal control problems in DAE with application to path planning problems for satellite mounted robots*, Dissertation, Naturwiss.-Math. Gesamtfakultät, Universität Heidelberg, 1996.
- [9] V.H. Schulz, H.G. Bock, R.W. Longman: *Optimal path planning for satellite mounted robot manipulators*, Advances in the Astronautical Sciences **82**, pp. 311–329, 1993.
- [10] M. Vögel: *Modellierung und optimale Bewegungen für einen Roboter im Weltraum*, Fortgeschrittenenpraktikum, Zentrum Mathematik, Technische Universität München, 1996.
- [11] O. von Stryk: *Numerische Lösung optimaler Steuerungsprobleme: Diskretisierung, Parameteroptimierung und Berechnung der adjungierten Variablen*, Fortschritt-Berichte VDI, Reihe 8, Nr. 441, VDI-Verlag, Düsseldorf, 1995.
- [12] O. von Stryk: *User's guide for DIRCOL-Version 2.1: a direct collocation method for the numerical solution of optimal control problems*, Lehrstuhl M2 Numerische Mathematik, Technische Universität München, 1999.
- [13] O. von Stryk, R. Bulirsch: *Direct and indirect methods for trajectory optimization*, Annals of Operations Research **37**, pp. 357–373, 1992.
- [14] O. von Stryk, M. Schlemmer: *Optimal control of the industrial robot Manutec r3*, in [3], pp. 367–382, 1994.

APPLIC  
METHOD T  
AERO  
DETE

The paper  
aerodynamic  
tering proble  
shown. The n  
application. I  
tion and for t  
the field of co

## Introduction

The fundamental  
aerodynamic char  
from wind tunnel  
ods). Another way