Bohumil Honzík

CONTROL OF THE KINEMATICALLY REDUNDANT MOBILE MANIPULATOR

ŘÍZENÍ KINEMATICKY REDUNDANTNÍHO MOBILNÍHO MANIPULÁTORU

SHORT VERSION OF PHD THESIS

Branch of study: Cybernetics and Informatics
Supervisor: Doc. Ing. František Zezulka, CSc.
Opponents: Doc. Ing. Vladimír Řeřucha, CSc.
Prof. Ing. Ladislav Jurišica, CSc.

Thesis defended on May 31, 2001
KEYWORDS
mobile manipulator, kinematic redundancy, redundancy resolution, collision avoidance, generalized potential, linear programming

KLÍČOVÁ SLOVA
mobilní manipulátor, kinematická redundantce, inverzní kinematika, vyhýbání kolizím, zobecněný potenciál, lineární programování

Thesis is available at the Department of Control, Measurement and Instrumentation, FEECS BUT
Contents

1 Introduction 5

2 State of the Art 6

3 Dissertation Goals 9

4 Approaches to Manipulator Control 9
  4.1 Redundancy Resolution 12
  4.2 Collision Avoidance 13
  4.3 World Representation 13

5 Results 14
  5.1 Redundancy Resolution Techniques 14
    5.1.1 Tested Methods 14
    5.1.2 Analysis 15
    5.1.3 Synthesis of the Suitable Method 16
  5.2 Collision Avoidance Using Generalized Potentials 17
    5.2.1 Analysis 17
    5.2.2 Solution Using Linear Programming 17
    5.2.3 Extension to 3D 20
  5.3 Interconnection with Evidence Grids 21
  5.4 Other Control Issues 23

6 Conclusion 24

References 25

Curriculum Vitae 28

List of Related Publications 29

Abstrakt 30
1 Introduction

Mobile manipulators\textsuperscript{1}, i.e., manipulators mounted on mobile platforms, are attracting significant interest in many areas, such as material transfer, warehouse management, nuclear plant maintenance, handling with chemicals, construction tasks, fire fighting, military, law enforcement, space and ocean exploration and many others. The growing field of service robots demands new systems with increased flexibility. This can be achieved in many different ways. Mobile manipulation – the coordinated use of manipulation capabilities and mobility – is an approach to increase robots’ versatility with regard to their motion skills. Unfortunately, mobile manipulators are very complicated machines which exhibit some particular characteristics, such as kinematic redundancy, coupled degrees of freedom (DOF) or connected systems with different dynamics. For that reason modeling, motion planning, and control of mobile manipulators will probably attract attention of researchers for several years.

Usually, autonomous robots are mobile in order to perform tasks at different locations. Manipulation capabilities are not necessary for tasks like information acquisition (inspection tasks) or for transportation tasks, where there are external handling devices that load and unload the robot (e.g., automatically guided vehicles in automobile industry). Other types of mobile robots are equipped with special devices for tasks like grass mowing or floor cleaning. If the robot needs to handle objects of different geometry and size, a universal handling device – a manipulator with a universal gripper, should be used instead of a specialized handling device. Because most mobile robots are designed to run with on-board energy sources (usually a battery), only small manipulators with little energy consumption can be used in mobile systems. Due to their small size, these manipulators have a rather restricted workspace. The range of operation of a mobile platform, however, is only constrained by objects (obstacles) in the environment. This is how the idea of mobile manipulation came up: combining mobility and manipulation capabilities to enlarge the workspace of a manipulator. Moreover, with their typically high degree of kinematic redundancy, created by the addition of the platform degrees of freedom to those of the manipulator, mobile manipulators can accomplish a great variety of tasks. Each one of these tasks is typically associated with a particular motion mode (platform only, manipulator only, combined motion), particular task requirements or optimization objectives (minimum time motion, optimal strength configuration, minimum spent energy, maximum dexterity, etc.) and particular constraints (obstacles, joint limits, coupled degrees of freedom of the platform, etc.).

\textsuperscript{1}It is necessary to distinguish terms mobile robot and mobile manipulator, since the former corresponds only to the subsystem of the latter.
The ultimate goal of robotics is to develop autonomous robots – machines that will be able to accept high-level descriptions of tasks and execute them without further human intervention. Such input descriptions will specify what the user wants to do rather than how to do it. Unfortunately, at present it seems to be impossible with the state-of-the-art artificial intelligence. Nevertheless, in many cases, such as the work in environments inaccessible to or very hazardous for humans, or in rehabilitation robotics, it is not necessary to use an “absolutely autonomous” robot, and the solution can be found with the telemanipulation approach. Thus, perception, decision-making and control may be divided into two layers: the high-level tasks can be made by the human (global planning, vision and scene analysis, gripper control), while the low-level processes may be accomplished by the computer (redundancy resolution, collision avoidance). Design of these low-level control algorithms is the main subject of this work.

Mobile manipulators are a special class of kinematically redundant manipulators with specific requirements. Although it is possible to build a nonredundant mobile manipulator as well, there are several good reasons to include kinematic redundancy into the system. Redundant manipulator (static or mobile) has the ability to avoid obstacles, joint limits and singular configurations, while fulfilling the given task. In addition to this task, such systems are capable of optimizing various performance criteria as minimization of joint velocities, joint accelerations, joint torques, energy, etc. Important is also the ability of the kinematically redundant mobile manipulator to overcome the coupled degrees of freedom (DOF) of the mobile platform, which would otherwise necessarily reduce the space of realizable end-effector trajectories.

2 State of the Art

There are many redundancy resolution techniques presented in the literature (good review can be found in [24]). Most of them are related to fixed-base serial-chain redundant manipulators with redundancy introduced by extra joints. Fundamental approaches can be categorized as local and global. Most of the methods reported up to date are of the local type, [15], [2], [8], [22], [23]. Formulation of local methods is usually based on differential kinematics and linear optimization technique. These approaches are relatively simple and require no a priori information about the end-effector motion. For that reason they are suitable for real-time implementations. Some drawbacks of local methods such as motion instability and excessive torque peaks in particular joints may be partially eliminated by various balancing and damping techniques, [18], [17], [30], but the other problems like nonconservative motion and lack of global optimality can be solved by the global optimization
only. However, global methods require a priori knowledge about the motion of the end-effector and obstacles, and also result in huge computation which rules out these approaches of the real-time applications, [21], [32], [11]. It was shown that in some cases global optimization may be converted into local optimization, [11], [23]; however, two-point boundary value problem of global optimization is thus transformed to an initial value problem which is solved locally, and natural boundary condition at one boundary does not apply anymore. Hence, the resulting path is only a “weak” minimum. An attempt to design a method which may transform some of the global optimization problems to their local equivalents using the theory of exponential dichotomies is presented in [19].

Nonetheless, kinematic redundancy of the mobile manipulator is introduced by the platform mobility, which is different from that introduced by extra joints. It is therefore not possible to generalize these results and conclusions for the class of mobile manipulators unconditionally.

Mobile manipulators are being studied approximately since the mid-1980s. Mobility and controllability characteristics of various types of mobile platforms with regard to their possible utilization in mobile manipulator systems are analyzed in [33]. There are several works that deal with the global path planning. One possibility is to formulate the mobile manipulator path planning as an optimization problem in which the decision variables for mobility (platform position) are separated from the manipulator joint variables in the cost function. The resulting numerical problem is nonlinear, with nonconvex, unconnected feasible regions in the decision space, and may be solved off-line using simulated annealing or genetic algorithms, [5], [36]. Similar approach that allows mobile manipulator to efficiently commute from one task to another in the accomplishment of sequences of tasks with widely varying requirements and modes of motion is presented in [27]. This approach calls for the planning of commutation configurations which are compatible with the requirements of both the finishing and initiating tasks.

Except the kinematical approaches mentioned above there is a variety of methods dealing with dynamic aspects of mobile manipulator control. To overcome the effects of the unknown interactive forces between the platform and the manipulator, a decentralized robust controller for a mobile manipulator is proposed in [16], where the platform and the manipulator are considered as two separate systems. Another approach to compensate for dynamic interaction – an extended Jacobian transpose control method, is presented in [9]. In [20], the task vector is augmented by the redundant DOF of the platform so that the manipulator motions can be determined by means of the optimal control theory. The values of the augmented coordinates are then derived by optimizing robot arm kinematics under the dynamic conditions of
the platform. A configuration control formalism is used in [29] to augment the basic task of the end-effector motion by a set of user-defined additional tasks in order to exploit the redundancy introduced by the platform mobility. The DOF of the platform and the manipulator are treated equally and they both contribute to the execution of the basic and additional tasks. The mobile manipulator control problem and the effect of dynamic interaction on the tracking performance is thoroughly analyzed in [34]. Four different control strategies are investigated: (1) with full compensation of the dynamic interaction between the mobile platform and the manipulator arm; (2) with the platform compensating the dynamic interaction caused by the manipulator; (3) with the manipulator compensating the dynamic interaction caused by the platform; and (4) without any compensation.

Techniques for coordinated task execution of a human and a mobile manipulator by means of the force control are presented in [7], [35]. In this case the human operator takes an initiative of the task execution. The task trajectory can be modified arbitrarily by the human, and the mobile manipulator follows the trajectory while executing the task together, e.g., when transporting a heavy object. Another issue that is also dealt with is the problem of stability of mobile manipulators, [7], [10].

Important part of the mobile manipulator control problem is collision avoidance. There is a large number of methods solving this question for general robotic systems, but actually they are all based on one of the following approaches: roadmap, cell decomposition, or potential fields, [14]. Roadmap approach consists of capturing the connectivity of the robot’s free space in a network of one-dimensional curves, called the roadmap. Path planning is thus reduced to searching for the path in the roadmap connecting the initial and goal configurations, [25], [26], [3], [4]. In the cell decomposition method the robot’s free configuration space is decomposed into cells, such that any path between two configurations within the cell can be easily generated. Finally, a connectivity graph representing the adjacency between cells is constructed and searched, [28], [1]. While the roadmap and cell decomposition methods are usually used for global path planning, potential field approach is of the local type. The idea of the potential field technique is that the robot behaves like the particle moving under the influence of an artificial potential, produced by the goal configuration and obstacles. The repulsive potential generated by the obstacle at the given point depends on the distance to that obstacle, [12]. In generalized potential field technique, which is an extension of the classical potential field approach, the potential function depends not only on the position of the object, but also on its velocity, [13]. The main advantage of potential field methods is that they can be easily incorporated into redundancy resolution techniques based on differential kinematics.
3 Dissertation Goals

The task considered in this study may be described as follows: the motion of the manipulator end-effector is controlled in real-time (either by the human operator or by a superordinate control system), while the mobile platform and all particular links are moving accordingly to guarantee the desired trajectory following.

Hence, the main goals may be summarized as follows:

1. to design a suitable local redundancy resolution technique, and
2. to solve the collision avoidance problem, i.e., to avoid:
   (a) collisions between the mobile platform and static or moving obstacles,
   (b) joint limits and singular configurations of the manipulator, and
   (c) collisions between particular manipulator links (self-collisions).

There are several additional requirements imposed on the proposed algorithms. Although all the methods have to be of the local type, they should have the best possible global characteristics relative to energy consumption. Typical property of kinematically redundant manipulators is that zero vector of end-effector velocities may correspond to nonzero vector of joint velocities. This fact would complicate the manipulator control in the telemanipulation sense. Thus, manipulator body must stop when end-effector stops. To guarantee good controllability and motion precision, excessive joint torque peaks should be avoided.

4 Approaches to Manipulator Control

The term robot control is usually referred to the maintenance of the dynamic response of the manipulator in accordance with some prespecified performance criterion, given the dynamic equations of motion of a manipulator. Nonetheless, term control in this work is used in a more general sense, characterizing the whole control process from the task definition to the low-level joint control. The general control scheme of the mobile manipulator considered in this study is presented in Fig. 1. Description of its particular function blocks is as follows:

Task generator is responsible for task planning. Based on the information about the environment it generates high level control commands like “pick up the black cube and put it into the red box”.

9
**Task space trajectory generator** computes the desired end-effector trajectory from the high level commands.

**Joint space trajectory generator** solves the manipulator inverse kinematics. Since this problem is underdetermined, the most suitable solution is chosen by means of information about the robot state (joint positions, velocities, accelerations) and the environment (vicinity of obstacles, their motion etc.).

**Controller** realizes the control laws or strategies based on the knowledge of the dynamic model of the manipulator to achieve the desired system response and performance.

**Mobile manipulator** represents the kinematic chain of links connected by joints. It is described by means of the kinematic parameters (expressed, e.g., using the Denavit–Hartenberg notation) and dynamic parameters (weight, center of gravity, moment of inertia of particular links etc.)

**Environment** is the robot’s workspace. It contains objects to be manipulated, and static and moving obstacles that have to be taken into account during the control process.

**Sensors** are used to measure actual manipulator configuration and allow the controller to close the feedback control loop.

**Environmental sensors** like ultrasonic sensors, laser rangefinders, cameras, stereovision systems etc. gain the information about the manipulator environment to allow the task planning and the choice of the suitable joint space trajectory.

With regard to dissertation goals stated in Sect. 3, the main subjects of this work are function blocks related to the trajectory generation in task and joint spaces. Task space trajectory generation from high level task description is very complex problem which is out of the scope of this study. But even in the telemanipulation concept, which is considered here, there are some facts that must be taken into account. They are mainly related to the proper choice of the coordinate frame in which the end-effector is controlled, and to the way how the visual information about the manipulation task is provided to the human operator. Computation of the joint space trajectory from the given task space trajectory, i.e., solution of the manipulator inverse kinematics, is much more difficult, since it involves not only consideration of the kinematic structure of the redundant manipulator, but also the dealing with various types of obstacles, represented by joint limits, singular configurations, or “real” obstacles in the manipulator workspace.
Figure 1  Mobile manipulator control scheme. Gray function blocks are the main subjects of this study.
4.1 Redundancy Resolution

Redundancy resolution algorithms (algorithms solving the inverse kinematics of redundant manipulators) have to be of the local type in order to meet the requirement of real-time applicability. Straightforward is the utilization of closed-form inverse kinematic functions which contain explicit analytical expressions for joint variables. Unfortunately, definition of such functions may be difficult and is always specific for the given manipulator kinematic structure. Moreover, for most redundant manipulators these functions do not exist at all. For that reason the redundancy resolution algorithms are mostly based on the differential kinematics.

Direct kinematic mapping from the joint space to the task space can be written for the general manipulator as follows:

\[ r = f(q) , \]

where \( r \in \mathbb{R}^m \) is the vector of task variables, \( q \in \mathbb{R}^n \) is the vector of joint variables, \( f : \mathbb{R}^n \rightarrow \mathbb{R}^m \) is a continuous vector function, which is mostly highly nonlinear, and \( m \) and \( n \) denote the number of task variables and manipulator DOF, respectively. For kinematically redundant manipulator holds:

\[ m < n . \]  

Differentiating (1) with respect to time the relationship between end-effector velocities and joint velocities becomes linear:

\[ \dot{r} = J(q)\dot{q} , \]

where \( J(q) \), the Jacobian matrix of \( r \) with respect to \( q \), is given by

\[ J(q) = \frac{\partial r}{\partial q} = \frac{\partial f}{\partial q} \in \mathbb{R}^{m \times n} . \]

A general formulation of the inverse kinematics at the velocity level may be written as follows:

\[ \dot{q} = J^\# \dot{r} + (E - J^\#J)z , \]

where \( J^\# \) denotes the generalized inverse of the Jacobian matrix. The first term on the right hand side is the particular solution, which is used to realize the desired end-effector velocity \( \dot{r} \). The second term is the homogeneous solution, consisting of the null space projectional operator \( (E - J^\#J) \) and an arbitrary vector \( z \), which contributes to a motion in joint space only – so-called self-motion of the mechanical system. Similar equation may be formulated also at the acceleration level.
4.2 Collision Avoidance

Collision avoidance is solved by means of the artificial potential fields. In this approach the robot (represented by a point moving in the joint space – also called configuration space) behaves like a particle moving under the influence of an artificial potential. The goal configuration generates an attractive potential pulling the robot towards the goal, and the obstacles produce a repulsive potential which pushes the robot away from them. The negated gradient of the total potential is treated as an artificial force applied to the robot.

In contrast to classical potential fields, where the repulsive potential depends on the robot’s position only and is given by the shortest distance to the obstacle, generalized obstacle potential $p$ is defined as the inverse of the reserve avoidance time $t_r$:

\begin{align}
    p &= 1/t_r , \\
    t_r &= t_M - t_m ,
\end{align}

where the maximum avoidance time $t_M$ is the maximum time during which the velocity towards the obstacle may be brought to zero under constant deceleration without hitting the obstacle, and the minimum avoidance time $t_m$ is the time in which the velocity towards the obstacle can be brought to zero using maximum deceleration.

4.3 World Representation

Since the collision avoidance algorithm requires geometrical description of obstacles, important is the choice of the suitable world model which will allow to get the best from the available sensors and also makes the computation of analytical descriptions of obstacles as easy as possible.

The evidence grid approach represents the robot’s environment by a two or three-dimensional regular grid. In each cell the evidence (or probability) is stored, based on accumulated sensor readings, that the particular patch of space is occupied. Many sensors report the distance to the nearest object in a given direction. Given the robot’s position, probabilities in the cells near the indicated object are increased and probabilities between the sensor and the sensed object are decreased (since it is the first object in that direction). The exact amount of increase or decrease to the various cells in the vicinity of the sight line forms the sensor model. The grid method accumulates occupancy evidence for an array of spatial locations, slowly resolving ambiguities as the robot moves. This allows the robot to integrate disparate readings over time, taken from different locations and even with different sensors. An important advantage of the evidence grid approach is the straightforward way of data fusion from various sensor types.
5 Results

Several redundancy resolution and collision avoidance approaches were analyzed and verified using computer simulations. Since the performance of these algorithms is not satisfactory with respect to the use in the locally controlled mobile manipulator system, modifications and new solutions are suggested.

5.1 Redundancy Resolution Techniques

Totally five methods resolving the kinematical redundancy at the velocity and acceleration levels were investigated. Simulations were performed with the mobile manipulator consisting of the 6-DOF arm mounted on the 3-DOF mobile platform. The on-line controlled manipulator end-effector is following two U-shaped trajectories, denoted as ‘long’ and ‘short’. These trajectories were designed with respect to demands usually made on mobile manipulator systems: the former represents the task, when the gripper has to reach outside the manipulator arm boundary and platform motion is therefore necessary; the latter corresponds to the situation, when the major part of the manipulation task may be performed inside the manipulator arm workspace, and platform motion is undesirable.

5.1.1 Tested Methods

Brief description of implemented algorithms and corresponding simulation results follow.

Minimization of joint velocities (MJV)

Instantaneous minimization of the term

\[ C_v = \dot{\mathbf{q}}^T \dot{\mathbf{q}}. \]  

Simulation result: manipulator motion along the whole trajectory is stable; joint velocities at the end of the motion are zero.

Minimization of joint accelerations (MJA)

Instantaneous minimization of the term

\[ C_a = \ddot{\mathbf{q}}^T \ddot{\mathbf{q}}. \]  

Simulation result: very similar to MJV, manipulator motion along the whole trajectory is stable; joint velocities at the end of the motion are nonzero.
Minimization of the kinetic energy (MKE)

Instantaneous minimization of the term

\[ C_e = \dot{q}^T H \dot{q} , \quad (10) \]

where \( H \) is the manipulator inertia matrix.

Simulation result: manipulator motion along the whole trajectory is stable, joint velocities at the end of the motion are zero; although this technique shows good performance, high torque peaks, which occur during the motion, are undesirable from the control point of view and degrade this strategy.

Minimization of joint torques (MJT)

Instantaneous minimization of the term

\[ C_t = \tau^T \tau , \quad (11) \]

where \( \tau \) is the vector of joint torques/forces.

Simulation result: this technique produces unstable manipulator motion with very large torques; joint velocities at the end of the motion are nonzero.

Minimization of inertia inverse weighted joint torques (MWJT)

Instantaneous minimization of the term

\[ C_w = \tau^T H^{-1} \tau . \quad (12) \]

Simulation result: similar to MJT, manipulator motion is unstable with very large torques; joint velocities at the end of the motion are nonzero.

5.1.2 Analysis

Minimization of the total energy consumed by the manipulator is equivalent to global minimization of joint torques (which correspond to the current in motors):

\[ I_c = \int \tau^T \tau dt . \quad (13) \]

Unfortunately, this is a typically global criterion, which cannot be converted to any local equivalent. Since MJV and MJA are purely kinematic methods, their influence on energy consumption is indirect.

In MKE as little mass as possible is moving at any moment. This method gives very good results, when the end-effector is able to follow the desired
trajectory without moving of the platform, because the mass of the platform is much larger in comparison with the arm and a lot of energy can be saved. Critical point comes at the moment when the end-effector arrives with nonzero velocity at the workspace boundary of the manipulator arm: the platform must in an infinitely short time produce an infinitely large acceleration, which results in undesirable torque peaks. Nonetheless, the manipulator motion is stable. The platform position is changed if, and only if, the end-effector has to reach outside the arm workspace.

Mobile manipulator motion, when utilizing MJT, is as follows: starting with zero joint velocities and moving the end-effector inside the manipulator arm workspace, mobile manipulator behaves like in MKE, i.e., the platform is not moving. The difference will be clear at the moment when the end-effector reaches the boundary of the manipulator arm workspace (at this moment the infinite torque peak as in MKE occurs): the platform starts to move to guarantee the target reachability, but does not change its velocity, until the boundary of the arm workspace is reached again. The reason for this is that the stop of the platform movement or change of its velocity requires exertion of certain torques, and joint torques are minimized in this strategy. Thus, MJT results in unstable motion.

The physical interpretation of MWJT is not as clear as in MJT. Nonetheless, even the weighting of joint torques by the inertia matrix inverse (which depends on the manipulator configuration) in the criterial function does not avoid motion instability.

5.1.3 Synthesis of the Suitable Method

It was shown that only MJV and MKE meet the requirement of zero joint velocities corresponding to zero end-effector velocities. Nonetheless, simulation results and also their physical interpretation imply that MKE is the right strategy, since it has better influence on the energy consumption. The only one difficulty are undesirable torque peaks which arise when the manipulator end-effector reaches the arm workspace boundary. Since this phenomenon is closely related to the elbow singularity of the arm (for the considered kinematic structure), torque peaks may be eliminated by avoiding of this singularity. This can be accomplished by the choice of proper scalar cost function $c_e(q)$, penalizing the distance to this singularity. Then, the gradient projection method is utilized – the negative gradient of $c_e(q)$ is used instead of the vector $z$ in (5), and singular configurations are compensated by the manipulator self-motion. Since this would cause the homogeneous solution of (5) to be nonzero also for zero vector of end-effector velocities $\dot{r}$, the whole homogeneous solution is weighted by the quadratic norm of this vector. Thus, resulting redundancy
resolution algorithm is given by

\[
\dot{q} = J_H^+ \dot{r} + (E - J_H^+ J) (\nabla c_e) \| \dot{r} \|
\]

(14)

where \( J_H^+ \) denotes the Moore–Penrose inverse of the Jacobian matrix weighted by the inertia matrix.

5.2 Collision Avoidance Using Generalized Potentials

Generalized potential field approach is used for collision avoidance between the mobile platform and planar obstacles. This technique results in behavior such that the robot is not repulsed by the obstacle, if it is not moving towards it. Resulting potential is used for computation of the suitable cost function. Collision avoidance algorithm is then integrated with the redundancy resolution using the gradient projection method.

5.2.1 Analysis

The key problem when using generalized potentials is the computation of \( t_M \). Most mobile robots used in research have circular, hexagonal or polygonal footprint that can be circumscribed and approximated by a circle. In this case the robot’s orientation is not considered and solution of obstacle avoidance problem is simplified, because such robot can be modeled as a point in the plane, when the contour of obstacles is enlarged by the radius of the circumscribing circle. This method is utilizing configuration space approach, and enlarged obstacles are called \( C \)-obstacles. Hence, computation of \( t_M \) is straightforward, since the distance from the robot to the obstacle in the given direction can be easily computed, as shown in Fig. 2.

In some cases the robot’s shape cannot be approximated by a circle, and rotational degree of freedom must be taken into account. Then, the configuration space obstacles are of three-dimensional forms which are not easy to represent or calculate.

5.2.2 Solution Using Linear Programming

In order to avoid computation of configuration space obstacles, calculations will be done in the operational space of task-related parameters (space with Cartesian rather than joint coordinates).

The task may then be stated as follows. Define the mobile robot as a convex polygon and obstacles as constrained areas in the plane with coordinates \( x \) and \( y \). Let the line perpendicular to this plane be the time axis \( t \). This new space will be referred to a task-time space. As the mobile robot is moving, its edges are forming surfaces which constrain three-dimensional object –
Figure 2 Construction of the C-obstacle. (a) Mobile robot with hexagonal footprint. (b) Approximation of the robot’s shape by a circle (dotted line). (c) Mobile robot in configuration space; \( d \) is the distance between the robot (point) and C-obstacle (dotted line) in the given direction.

\textit{task-time image}. The same holds for the obstacles as well. If any of these objects intersect, collision occurs. Then, the intersection \( p_c = [x_c \ y_c \ t_c] \) with the lowest \( t \)-coordinate corresponds to the collision point, where \( x_c, y_c \) are its Cartesian coordinates and \( t_c \) denotes the time, at which collision occurs. Situation is shown in Fig. 3.

Consider the simplified case when all moving objects may only translate with constant velocities. Thus, the constraining surfaces are planes, and linear programming (LP) can be used for computation of \( t_c \). The linear program is then expressed as follows\(^2\):

\[
\begin{align*}
\text{minimize} \quad & \mathbf{c}^T \chi, \\
\text{subject to} \quad & \mathbf{A} \chi \geq \mathbf{b}, \\
& \quad t \geq 0.
\end{align*}
\]

where \( \chi = [x \ y \ t]^T \) is the vector of variables to be solved for, \( \mathbf{A} \) is a matrix of known coefficients and \( \mathbf{c} \) and \( \mathbf{b} \) are vectors of known coefficients. An ad-

\(^2\)To be consistent with notation common in LP and avoid confusion with \( c \) in (19) and (21), variables defining the linear program are typeset in bold sans serif: \( \mathbf{A}, \mathbf{b}, \) and \( \mathbf{c} \).
Additional constraint (17) must be added to guarantee the task-time images of the platform and the obstacle not to intersect in the negative time. Inequality (16) is given by constraining planes which are formed by the mobile robot and obstacles (Fig. 3). More specifically,

\[
\begin{align*}
A &= \begin{bmatrix}
    a_1 & b_1 & c_1 \\
    a_2 & b_2 & c_2 \\
    \vdots & \vdots & \vdots \\
    a_l & b_l & c_l
\end{bmatrix}, \\
\mathbf{b} &= \begin{bmatrix}
    -d_1 \\
    -d_2 \\
    \vdots \\
    -d_l
\end{bmatrix},
\end{align*}
\]

where

\[
\begin{align*}
a_i x + b_i y + c_i t + d_i &= 0, \\
i &= 1 \ldots k, k + 1 \ldots l.
\end{align*}
\]

Equation (19) describes the \(i\)-th constraining plane in the task-time space. Thus, there are \(k\) equations corresponding to the robot, and \(l - k\) equations corresponding to the obstacle. Because \(t\) is to be minimized, vector \(\mathbf{c}\) in the objective function (15) is given as \(\mathbf{c} = [0 \ 0 \ 1]^T\). The number of rows in matrix \(A\) corresponds to the number of constraining planes, and coefficients in (19) depend on robot’s and obstacles’ shape. It can be shown that \(t_M = 2t_c\).

When considering rotational degree of freedom of the robot, situation becomes much more difficult, since objects formed by the robot moving with constant velocity in the task-time space are generally helicoidal-shaped and the problem is no longer linear. To avoid using of nonlinear programming, these complex objects are properly approximated first. Instead of rotation of particular edges of the mobile robot, their movement is approximated by
Consideration of rotation. (a) Mobile robot translating and rotating with constant velocity. (b) Approximation of rotation – above view: helicoidal shaped task-time image to be approximated (dashed lines), and its approximation (solid lines)

translation in the direction perpendicular to these edges, as shown in Fig. 4. The linear programming approach may be used for two convex objects only – mobile platform and one obstacle. If there are more obstacles, they must be treated separately, i.e., $t_M$ has to be calculated for each of them, and the smallest value is then used for computation of the repulsive potential.

### 5.2.3 Extension to 3D

Previous sections have presented the calculation of generalized potentials for the case of planar objects. Nevertheless, this approach can be extended for 3D objects as well. This is achieved by simply adding one further dimension to the planar problem. Thus, (19) is replaced by the following equation:

$$a_i x + b_i y + c_i z + d_i t + e_i = 0.$$  \hspace{1cm} (21)

Equation (21) describes the $i$-th hyperplane in the task-time space. In other words, it defines the position of the $i$-th plane in the 3D task space at the time $t$. These planes constrain half-spaces, intersections of which form particular convex 3D objects. Matrix $A$ and vectors $b$ and $c$ are also altered accordingly:

$$A = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ \vdots & \vdots & \vdots & \vdots \\ a_l & b_l & c_l & d_l \end{bmatrix}, \quad b = \begin{bmatrix} -e_1 \\ -e_2 \\ \vdots \\ -e_l \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \hspace{1cm} (22)$$

The calculation of the maximum avoidance time $t_M$ is then accomplished in the same way as in Sect. 5.2.2.
5.3 Interconnection with Evidence Grids

In this study only the two-dimensional evidence grid approach is considered. It will be further assumed that the evidence grid is already built (or is being updated by the low-level preprocessing algorithm) and therefore available at every time instant as the binary matrix. Elements assumed to be occupied correspond to ones (denoted by black dots in Fig. 5), free cells are zeros (blank area). The task is now to gain the analytical description of obstacles from this world representation. All analytically defined (AD) obstacles must be convex areas. They need not to correspond exactly to the occupied grid cells, but should always contain them. If any occupied area in the evidence grid is nonconvex, it must be either decomposed in several AD obstacles or encapsulated in one AD obstacle.

The simplest brute force method is to treat every grid cell as one AD obstacle, i.e., a small square. Computing such AD obstacles is very easy, but there is one difficulty: above presented approach to computing $t_{M}$ using linear programming can be used for two convex objects only. Thus, every occupied cell would have to be treated separately, which can take a lot of time. It is therefore desirable to collect as many neighboring cells as possible together and consider them as one AD obstacle. The process of calculation of such AD obstacles can be split into three steps:

Calculation of the Constraining Polyline

When constructing the constraining polyline (CP), the grid is searched for all that occupied cells that are visible from the robot’s point of view $V$ (e.g., the center of robot’s footprint), as illustrated in Fig. 5a. Bresenham’s scan converting algorithm for lines [6] is used for computation of coordinates of points that lie on or near the straight line connecting the point $V$ with particular grid boundary points (denoted by diamonds). Once the calculated point corresponds with the occupied cell, it is considered as the visible one (denoted by triangles), and the scan-conversion routine is interrupted. This procedure is repeated until the whole area is scanned. Result of this process is the set of all visible points, which form vertices of the CP (Fig. 5b).

Approximation of the Constraining Polyline

Each line segment connecting two vertices of the CP corresponds to one inequality (16) in the linear program. Since the number of convex obstacles that have to be constructed from the CP depends on the number of vertices and the complexity of CP, it is advantageous to approximate its shape and reduce this way the computational effort. Approximation is performed using
the adapted Sklansky–Gonzalez’ algorithm [31]. The goal is to find a subset of a given set of data points such that the polygonal curve formed by joining every pair of successive points of the subset by a straight line segments lies within a given tolerance from all the points of the given set, Fig. 5c.

**Construction of Analytically Defined Obstacles**

Data points generated by the approximation procedure must be further processed into the form acceptable by the linear program solver that can handle only two convex task-time images. If either the mobile robot or obstacles have nonconvex footprint, they must be decomposed into convex polygons first. Task-time image of each convex part of the mobile robot is then searched for its intersection with task-time image of each convex obstacle. Creation of convex polygons is illustrated in Fig. 5d.

**Figure 5** Evidence grids. (a) Search for the visible cells. (b) Constraining polyline. (c) Approximation. (d) Construction of convex obstacles.
5.4 Other Control Issues

Besides the key problems mentioned above there are several issues that will be only shortly discussed. Their detailed description can be found in the dissertation thesis.

Joint limits. Since the motion of each joint of the manipulator arm is usually due to the mechanical construction restricted to certain working range, it is necessary to avoid any potential collisions with the joint limits. Cost function penalizing the distance to joint limits is formed, which is then incorporated using the gradient projection method into the redundancy resolution.

Selfcollisions. Collisions between the particular links of the mobile manipulator are avoided in the similar way as joint limits. Cost function may be formulated by means of generalized potentials.

Coupled DOF of the platform. For some kinematic structures of the mobile platform (e.g., with differential driven wheels), the geometry of the feasible free paths between two configurations is restricted, and the platform can move only along specifically shaped trajectories. This can be considered as an additional constraint task which is executed along with the original end-effector task.

Guarantee of realizability. Mobile manipulator inverse kinematics is calculated using numerical integration of (14). When the end-effector is driven to reach somewhere, where the platform is no longer able to follow, integration fails, e.g., when reaching too high, or when the platform’s path is obstructed by the obstacles that cannot be avoided by any self-motion. This fact must be considered in the redundancy resolution algorithm.

Telemanipulation. When performing difficult manipulation tasks, important topic is the choice of the proper coordinate frame, in which the end-effector will be controlled. In dependence on the character of the task the end-effector motion may be defined in the coordinate frame coupled either with the platform, the end-effector itself, or the camera. The last mentioned case can be considered, when telepresence approach is used. Thus, the information about the manipulation task and the environment is provided to the human operator over a vision system. Utilization of stereovision and head-mounted displays is recommended, since the ability to estimate distances between objects is very limited in single-camera systems, and any manipulation tasks are then difficult to complete successfully.
6 Conclusion

Resulting redundancy resolution and collision avoidance algorithms are integrated into a compact form and extended with correction terms that make them robust and resistive to improper control commands. Although the mobile manipulator used in this simulation study consists of the 6-DOF manipulator arm mounted on a 3-DOF mobile platform, presented approaches are independent on the kinematic structure of the robotic system. The design was restricted to algorithms of the local type in order to guarantee the possibility of their real-time implementation and incorporation of instantaneous sensor measurements.

Besides the analysis and utilization of some principles already known in the robotic research there are several novel approaches introduced in this dissertation: (1) Weighting of the homogeneous solution of the inverse kinematics by a scalar function of the norm of end-effector velocities to suppress the manipulator self-motion when the end-effector stops. (2) Algorithm for computation of the distance measure for collision avoidance in 2D and 3D based on linear programming. (3) Interconnection between the 2D case of the presented collision avoidance concept and the proper world model.

Although this work was an attempt to propose complete algorithms that could be directly applicable to a real system, it is still only a simulation study which lacks the qualities of the real robot. The realization of the presented redundancy resolution method should be possible without great troubles – there are many examples of similar strategies implemented and verified on real manipulators. The same holds for the avoidance of joint limits and singular configurations. Nonetheless, problems may occur when using the obstacle avoidance by means of the generalized potentials. The critical part of the system are the sensors used for acquisition of information about the environment. Since the generalized potential approach has the intrinsic tendency to oscillations, data from slow and inaccurate sensors may increase the probability of such behavior even if evidence grids are used to make the data more reliable.

Another subject to the future work, besides the issues related to implementation problems on a real system, is the interconnection between the evidence grid based world model and the 3D extension of the collision avoidance algorithm. The key issue in this case is the fact that separate handling of particular cells is not possible due to the large amount of data, while the approximation of such complex nonconvex three-dimensional objects is very difficult. Moreover, construction of 3D evidence grids is not easy, even though last research advances are very promising. For that reason is this method more suitable for collision avoidance between objects of known shape and motion.
References


Bohumil Honzík

* 1972 in Hranice, Czech Republic

Education
1987 – 1991   grammar school in Hranice, Czech Republic
1991 – 1996   study at the Dept. of Control and Instrumentation, Faculty of El. Engineering and Comp. Science, Brno University of Technology, Czech Republic, master degree in Cybernetics, Automation and Measurement
1996 – 1999   doctorate study at the Dept. of Control and Instrumentation, Faculty of El. Engineering and Comp. Science, Brno University of Technology, Czech Republic, specialization Cybernetics and Informatics

International Scholarships
October 1997 – July 1988   Control Systems Engineering Group, Department of Electrical Engineering, FernUniversität Hagen, Germany

Professional Interests
• simulation and visualization of robotic systems
• MATLAB, SIMULINK, programming in C, LaTeX, VRML

Languages
• German
• English
List of Related Publications


Abstrakt