Control Algorithms for On-Line Obstacle Avoidance for Redundant Robots

Leon Žlajpah and Bojan Nemec

Institute Jožef Stefan, Jamova 39, 1000 Ljubljana, Slovenia
leon.zlajpah@ijs.si

Abstract: The paper deals with on-line obstacle avoidance algorithms. We compare some obstacle avoidance strategies and group them into velocity and force strategies. The emphasis in the paper is given to the redundancy resolution and integration of these algorithms into the close-loop control. We address the problem of singular configurations and propose an algorithm based on artificial forces which avoids the problem of singular configurations. The efficiency of the proposed control algorithm is illustrated by the simulation.

1 Introduction

One of the goals of the robotic research is to provide control algorithms which allow robots to move in an unstructured environment. Using this ability, the robot can operate in an environment with unknown obstacles where collisions of the robot and obstacles can occur. The collisions or contacts may be part of the task, e.g. in the assembly operations, or an undesired event. In the later case, the problem is how to maneuver a kinematically redundant manipulator in the presence of obstacles in such a way that the end-effector tracks a given task space trajectory and at the same time the collisions between the manipulator body and the obstacles are avoided.

Generally, this problem, in literature addressed as the collision avoidance problem, can be solved by two kinds of strategies. One deals with high-level path planning in which the path is planned off-line in such a way that the obstacles in the workspace are avoided. The alternative is to solve the collision avoidance problem on-line by the low-level control. This strategy, which is closer to our approach, has been studied by many researchers, including [7, 5, 2, 3, 1, 4, 13].

The obstacle avoidance approach presented in this paper is to assign each point on the body of the manipulator, which is close to the obstacle, a velocity or artificial force component in a direction away from the obstacle. We compare some algorithms for obstacle avoidance. The emphasis in the paper is given to the redundancy resolution and integration of these algorithms into the close-loop control. The presented formulation of obstacle avoidance task includes the criteria that are defined in the Cartesian space and when the necessary joint velocities are calculated some problems arise when a part of the manipulator is in singular configuration. To avoid this problem, we propose an algorithm based on artificial forces. The efficiency of the proposed control algorithm is illustrated by the simulation.

2 Background

The robotic systems under study are n degrees of freedom serial redundant manipulators. The kinematics can be described by the following equations

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q}), \quad \dot{\boldsymbol{q}} = \mathbf{J}^{\#}\dot{\boldsymbol{x}} + \mathbf{N}\dot{\boldsymbol{q}}, \quad \ddot{\boldsymbol{q}} = \mathbf{J}^{\#}(\ddot{\boldsymbol{x}} - \dot{\mathbf{J}}\dot{\boldsymbol{q}}) + \mathbf{N}\ddot{\boldsymbol{q}} \tag{1}$$

where $\boldsymbol{f}$ is a vector function representing the manipulator forward kinematics, $\mathbf{J}^{\#}$ is the generalized inverse of the Jacobian matrix $\mathbf{J}$ and $\mathbf{N}$ is a matrix representing the projection into the null space of $\mathbf{J}$, $\mathbf{N} = (\mathbf{I} - \mathbf{J}^{\#}\mathbf{J})$. To decouple the task and the null-space it is reasonable to use the dynamically consistent pseudoinverse $\bar{\mathbf{J}}$, $\mathbf{J}^{\#} = \bar{\mathbf{J}} = \mathbf{H}^{-1}\mathbf{J}^{T}(\mathbf{J}\mathbf{H}^{-1}\mathbf{J}^{T})^{-1}$ [6].

For the redundant manipulators the static relationship between the m-dimensional generalized force in task space $\boldsymbol{F}$, and the corresponding n-dimensional generalized joint space force $\boldsymbol{\tau}$ is

$$\boldsymbol{\tau} = \mathbf{J}^{T}\boldsymbol{F} + \bar{\mathbf{N}}^{T}\boldsymbol{\tau} \tag{2}$$

where $\mathbf{N}^{T}$ is a matrix representing the projection into the null space of $\mathbf{J}^{T}$.

3 Obstacle avoidance

The natural strategy to avoid obstacles would be to move the manipulator away form the obstacle. However, this strategy can be applied only to redundant manipulators. Namely, without changing the motion of the end-effector, the reconfiguration of the manipulator into a collision-free configuration can be done only if the manipulator has redundant degrees-of-freedom (DOF). The flexibility depends on the degree-of-redundancy, i.e. on the number od redundant DOF. A high degree-of-redundancy is important especially when the manipulator is working in an environment with many potential collisions with obstacles.

We assume that the environment is unstructured and dynamic, and some sensors are used to measure the distance between the obstacles and body of the manipulator. There are different types of sensor systems which can be used to detect objects in the neighborhood of the manipulator such as tactile and proximity sensors or vision systems. We assume in the following that the manipulator is equipped with a sensor which detects distances to obstacles and gives the position of the critical points.

Let A_o be the critical point in the neighborhood of an obstacle (see Fig. 1). To avoid a possible collision we could assign to it such a velocity that it moves away from the obstacle or to push the critical point away by a force acting in A_{F}.

3.1 Velocity strategy

When using the "velocity" strategy the joint velocities can be calculated as

$$\dot{\boldsymbol{q}} = \mathbf{J}^{\#}\dot{\boldsymbol{x}}_e + \mathbf{N}\dot{\boldsymbol{\varphi}} \tag{3}$$

The null-space velocity $\dot{\boldsymbol{\varphi}}$ is obtained from the desired end-effector velocity $\dot{\boldsymbol{x}}_e$ and the desired velocity $\dot{\boldsymbol{x}}_o$ in point A_o using the equations

$$\mathbf{J}\dot{\boldsymbol{q}} = \dot{\boldsymbol{x}}_e, \quad \mathbf{J}_{\mathrm{F}}\dot{\boldsymbol{q}} = \dot{\boldsymbol{x}}_o \tag{4}$$

where $\mathbf{J}_{\mathrm{F}}$ is a Jacobian matrix associated with the point point A_{F}. Maciejewski and Klein [7] propose to substitute (3) into (4) which yields

$$\dot{\boldsymbol{\varphi}} = [\mathbf{J}_{\mathrm{F}}\mathbf{N}]^{\#}(\dot{\boldsymbol{x}}_o - \mathbf{J}_{\mathrm{F}}\mathbf{J}^{\#}\dot{\boldsymbol{x}}_e) \tag{5}$$

and gives the final solution in the form

$$\dot{\boldsymbol{q}} = \mathbf{J}^{\#}\dot{\boldsymbol{x}}_e + [\mathbf{J}_{\mathrm{F}}\mathbf{N}]^{\#}(\dot{\boldsymbol{x}}_o - \mathbf{J}_{\mathrm{F}}\mathbf{J}^{\#}\dot{\boldsymbol{x}}_e) \tag{6}$$

Note that the above solution guarantees to achieve exactly the desired $\dot{\boldsymbol{x}}_o$ only if the degree of redundancy of the manipulator is sufficient. Another problem arises if one part of the manipulator is near the singular position. In such configurations some components of the pseudoinverse $[\mathbf{J}_{\mathrm{F}}\mathbf{N}]^{\#}$ may become very large and special methods should be applied to minimize these effects [7].

Another possible solution for $\dot{\boldsymbol{\varphi}}$ is to calculate joint velocities which satisfy the secondary goal as

$$\dot{\boldsymbol{\varphi}} = \mathbf{J}_{\mathrm{F}}^{\#}\dot{\boldsymbol{x}}_o \tag{7}$$

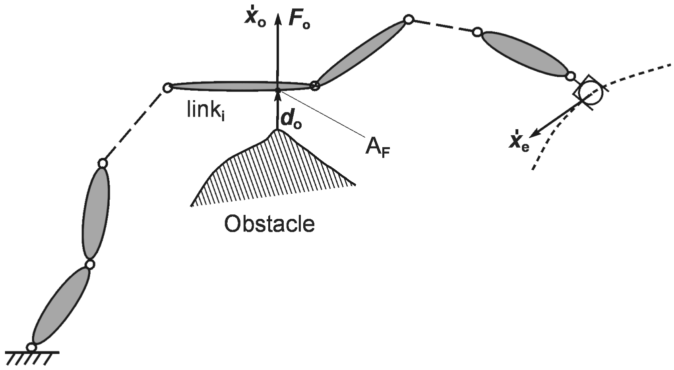

Figure 1. Manipulator in contact with obstacle

and then substitute $\dot{\boldsymbol{\varphi}}$ into (3) which yields

$$\dot{\boldsymbol{q}} = \mathbf{J}^{\#}\dot{\boldsymbol{x}}_e + \mathbf{N}\mathbf{J}_{\mathrm{F}}^{\#}\dot{\boldsymbol{x}}_o \tag{8}$$

This approach avoids the singularity problem of $[\mathbf{J}_{\mathrm{F}}\mathbf{N}]$. However, it does not guarantee to achieve exactly the desired $\dot{\boldsymbol{x}}_o$ even if the degree of redundancy is sufficient because $\mathbf{J}_{\mathrm{F}}\mathbf{N}\mathbf{J}_{\mathrm{F}}^{\#}\dot{\boldsymbol{x}}_o$ is not equal $\dot{\boldsymbol{x}}_o$ in general.

In both cases a special attention has to be given to the selection of the desired critical point velocity $\dot{\boldsymbol{x}}_o$ [7, 12]. Large values of $\dot{\boldsymbol{x}}_o$ could cause unstable motion. In [7] it is proposed to change in respect of the obstacle distance the magnitude of $\dot{\boldsymbol{x}}_o$ and the amount of the homogenous solution to be included in the total solution. For tactile sensor we propose [12] that the magnitude of the velocity depends on the duration of the contact and that the reached velocity is preserved for some time after the contact between the manipulator and the obstacle is lost.

3.2 Force strategy

Another strategy is to use artificial forces to move the manipulator away from the obstacle. Usually an artificial potential field is created around each obstacle which serves for the generation of artificial forces. Our approach is similar to those given in [5, 13], but differs in the way the artificial force is generated and how the obstacle avoidance algorithm is implemented in the close-loop control.

The artificial force $\boldsymbol{F}_o$ is based on the minimal distance vector $\boldsymbol{d}_o$ between the critical point A_{F} and the obstacle (see Fig. 1) using the equation

$$\boldsymbol{F}_o = \begin{cases} -K_s(K_{\mathrm{prox}} - \|\boldsymbol{d}_o\|)^p \dfrac{\boldsymbol{d}_o}{\|\boldsymbol{d}_o\|} & K_{\mathrm{prox}} > \|\boldsymbol{d}_o\| \\ 0 & K_{\mathrm{prox}} \geq \|\boldsymbol{d}_o\| \end{cases} \tag{9}$$

where K_{prox} is the critical distance to the obstacle and p is a scalar constant.

When the external force $\boldsymbol{F}_o$ acts anywhere on the body of the manipulator, then the static relation between the external force $\boldsymbol{F}_o$ and joint torques $\boldsymbol{\tau}$ is

$$\boldsymbol{\tau} = \mathbf{J}^T\boldsymbol{F} + \bar{\mathbf{N}}^T\boldsymbol{\tau}_{\mathrm{F}} = \mathbf{J}^T\boldsymbol{F} + \bar{\mathbf{N}}^T\mathbf{J}_{\mathrm{F}}^T\boldsymbol{F}_o \tag{10}$$

where $\boldsymbol{F}$ are task forces. The detailed analysis of the influence of external forces is given in [11].

4 Control algorithms

Most tasks performed by a redundant manipulator are broken down into several subtasks with different priority. In our case the task with the highest priority, referred to as the main task, is associated with the positioning of the end-effector in the task space, and other subtasks are associated with the obstacle avoidance and other additional tasks (if the degree-of-redundancy is high enough).

4.1 *Velocity strategy*

To decouple the task space and null-space motion the proposed controller is given in the form

$$\boldsymbol{\tau} = \mathbf{H}(\bar{\mathbf{J}}(\ddot{\boldsymbol{x}}_c - \dot{\mathbf{J}}\dot{\boldsymbol{q}}) + \bar{\mathbf{N}}(\boldsymbol{\phi} + \dot{\bar{\mathbf{J}}}\dot{\boldsymbol{x}}) + \mathbf{h} + \mathbf{g} \tag{11}$$

where $\ddot{\boldsymbol{x}}_c$ and $\boldsymbol{\phi}$ represent the task space and the null-space control law, respectively. The task space control $\ddot{\boldsymbol{x}}_c$ can be selected as

$$\ddot{\boldsymbol{x}}_c = \ddot{\boldsymbol{x}}_d + \mathbf{K}_v\dot{\boldsymbol{e}} + \mathbf{K}_p\boldsymbol{e} \tag{12}$$

where $\boldsymbol{e}$, $\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}$, is the tracking error, $\ddot{\boldsymbol{x}}_d$ is the desired task space acceleration, and $\mathbf{K}_v$ and $\mathbf{K}_p$ are constant gain matrices. The selection of $\mathbf{K}_v$ and $\mathbf{K}_p$ can be based on the desired task space impedance. To perform the additional subtask the vector $\boldsymbol{\phi}$ is given in the form

$$\boldsymbol{\phi} = \ddot{\boldsymbol{\varphi}} + \mathbf{K}_n\dot{\boldsymbol{e}}_n, \quad \dot{\boldsymbol{e}}_n = \bar{\mathbf{N}}(\dot{\boldsymbol{\varphi}} - \dot{\boldsymbol{q}}) \tag{13}$$

where $\dot{\boldsymbol{\varphi}}$ is the desired null space velocity and $\mathbf{K}_n$ is $n \times n$ diagonal gain matrix.

Next $\dot{\boldsymbol{\varphi}}$ has to be defined. We propose the desired null space velocity $\dot{\boldsymbol{\varphi}}$ to be defined as

$$\dot{\boldsymbol{\varphi}} = \sum_{i=1}^{k} \dot{\boldsymbol{\varphi}}_i \tag{14}$$

where k indicates the number of critical points and $\dot{\boldsymbol{\varphi}}_i$ are calculated using Eqs. 5 or 7.

4.2 *Force strategy*

Now the proposed controller is given in the form

$$\boldsymbol{\tau} = \mathbf{H}(\bar{\mathbf{J}}(\ddot{\boldsymbol{x}}_c - \dot{\mathbf{J}}\dot{\boldsymbol{q}}) + \bar{\mathbf{N}}(\boldsymbol{\phi} + \dot{\bar{\mathbf{J}}}\dot{\boldsymbol{x}}) + \bar{\mathbf{N}}^T\boldsymbol{\tau}_o + \mathbf{h} + \mathbf{g} \tag{15}$$

which is the same as Eq. 11 except that the term $\boldsymbol{\tau}_o$ is added representing the torques due to artificial forces. We define $\boldsymbol{\tau}_o$ as

$$\boldsymbol{\tau}_o = \sum_{i=1}^{k} \mathbf{J}_{\mathrm{F},i}^T \boldsymbol{F}_o, i \tag{16}$$

The main advantage of force control strategy compared to the velocity strategy can be seen in the above equation. Namely, when calculating $\dot{\boldsymbol{\varphi}}_i$ in Eq. 14 the pseudoinverse of $[\mathbf{J}_\mathrm{F}\mathbf{N}]$ or $\mathbf{J}_\mathrm{F}$ has to be calculated but in Eq. 16 only the transpose of $\mathbf{J}_\mathrm{F}$ is necessary.

The another idea behind this control law is to use open-loop control for the obstacle avoidance, i.e. the generated artificial forces push the manipulator into the safe configuration until they are zero. Hence, the null-space velocity controller can be used for other tasks. Suppose that p is a function representing the desired performance criterion. We can select $\dot{\boldsymbol{\varphi}}$ as [8]

$$\dot{\boldsymbol{\varphi}} = \mathbf{K}\nabla p \tag{17}$$

to optimize p. Here ∇p is the gradient of p and $\mathbf{K}$ is a gain matrix.

However, there is another difference between both control schemes regarding the selection of null-space gains. Namely, when velocity strategy is used, the null-space gains $\mathbf{K}_n$ should be high to assure good tracking of the desired null-space velocity. In case of force strategy these gains should be smaller so that the manipulator in more compliant in the null-space and the responses on artificial forces are faster.

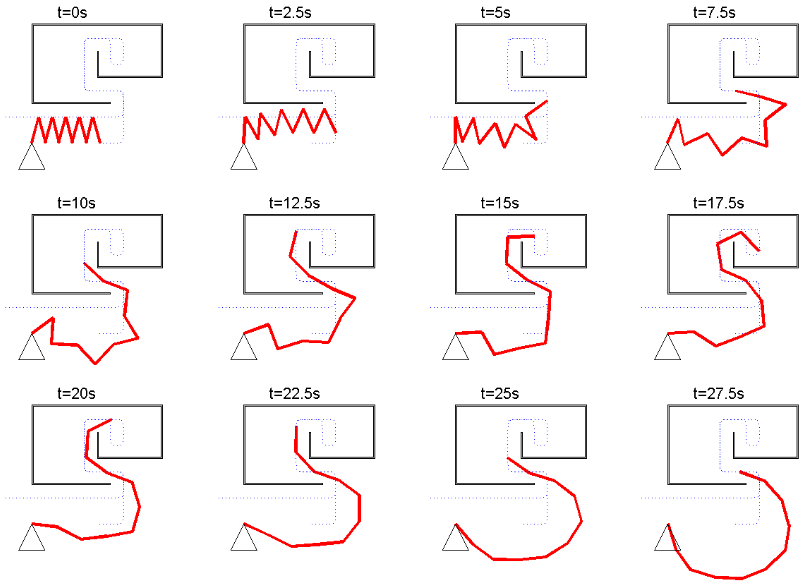

Figure 2. Tracking of a path around obstacles: manipulator configurations

5 Simulation example

The following simulation example illustrates the behavior of a 10-R planar manipulator when the manipulator is moving in an unstructured environment with obstacles. The simulation has been done in MATLAB/SIMULINK using the Planar Manipulators Toolbox [10]. The Planar Manipulators Toolbox is based on a kinematic and dynamic model of a planar manipulator with revolute joints and permits the simulation of manipulators with many DOF.

The manipulator has 10 revolute joints and is supposed to move in a plane x-y. The task coordinates $\boldsymbol{x}$ are the positions in x-y plane, $\boldsymbol{x} = [x, y]^T$. All the links are of equal lengths ($l = 1$m) and are modeled as rods ($m = 1$kg, $l_c = 0.5$m, $I = m\, l^2/12$ and $B_v = 0.5$Nms). There is additional load mass at the end of the last link ($m_l = 1$kg). The controller is based on the algorithm (15). The task space controller parameters are $\mathbf{K}_p = 1000\mathbf{I}\text{s}^{-2}$ and $\mathbf{K}_v = 80\mathbf{I}\text{s}^{-1}$, which are tuned to ensure good tracking of the task trajectory (stiff task space behavior). The desired null space velocity has been calculated to maximize the manipulability using the performance criterion $p = K_m\sqrt{\det(\mathbf{J}\mathbf{J}^T)}$ and the null space controller parameters are $K_m = 0.1$ and $\mathbf{K}_n = 30$ which ensure fair tracking of the null space velocity (compliant null space behavior). To avoid the obstacles the proximity sensor distance has been $K_{\text{prox}} = 0.2$ and the artificial forces has been calculated using $K_s = 1000$ and $p = 1$.

Fig. 2 shows the motion of the manipulator at subsequent time instants. The manipulator should move along the path defined by cubic splines shown as dashed line in Fig. 2. We can see that the manipulator is never in the contact with the obstacle. Additionally, if due to some reasons the contact with the obstacle would occur, the low null-space damping would allow the contact force to push the manipulator away [9]. Furthermore, the secondary task, i.e. the manipulability optimization, is performed along the path in the best way possible. The effectiveness of this method depends especially on the selection of null space gains. We have found out that making the null space behavior more compliant reduces the artificial forces necessary to move away from the obstacle but such gain also degrade the tracking of the desired null space velocity. To high artificial forces and low gain $\mathbf{K}_n$ may cause also some oscillations when moving in narrow passages between obstacles.

6 Conclusion

The obstacle avoidance algorithms presented in this paper assign each point on the body of the manipulator, which is close to the obstacle, a velocity or force component in a direction that is away from the obstacle. Some algorithms are compared regarding the way the manipulator is moved into safe position and the integration of these algorithms into the close-loop control is given. To avoid the problem of singular positions which plays an important role in velocities strategies we favor the force strategies. We propose an algorithm based on artificial forces which avoids the problem of singular configurations. Additionally, the proposed algorithm enables us to use the null-space velocity controller for additional subtasks like optimization of a performance criterion. The simulation example shows the efficiency of the proposed control algorithm. To obtain good behavior the gains have to be tuned so that the manipulator exhibits stiff behavior in the task space and is compliant in the null space.

7 References

[1] Y. Chen and M. Vidyasagar. Optimal control of robotic manipulators in the presence of obstacles. *J. of Robotic Systems*, 7(5):721 – 740, 1990.

[2] E. Cheung and V. Lumelsky. Motion planning for robot arm manipulators with proximity sensing. In *Proc. Intl. Conf. On Robotics and Automation*, pages 740 – 745, Philadelphia, Pennsylvania, 1988.

[3] R. Colbaugh, H. Seraji, and K.L. Glass. Obstacle avoidance for redundant robots using configuration control. *J. of Robotic Systems*, 6(6):721 – 744, 1989.

[4] Z.Y. Guo and T.C. Hsia. Joint trajectory generation for redundant robots in an environment with obstacles. *J. of Robotic Systems*, 10(2):119 – 215, 1993.

[5] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. of Robotic Research*, 5:90 – 98, 1986.

[6] O. Khatib. A unified approach for motion and force control of robot manipulators:the operational space formulation. *IEEE Trans. on Robotics and Automation*, 3(1):43 – 53, 1987.

[7] A.A. Maciejewski and C.A. Klein. Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *Int. J. of Robotic Research*, 4(3):109 – 117, 1985.

[8] D. N. Nenchev. Redundancy resolution through local optimization: A review. *J. of Robotic Systems*, 6(6):769 – 798, 1989.

[9] L. Žlajpah. Compliant Motion Control of Redundant Manipulators in Constraint Space. In M. Guglielmi, editor, *Preprits of 5^{th} IFAC Symp. on Robot Control, SYROCO'97*, pages 657 – 661, Nantes, 1997.

[10] L. Žlajpah. Simulation of *n*-r planar manipulators. *Simulation Practice and Theory*, 6(3):305 – 321., 1998.

[11] L. Žlajpah. Influence of external forces on the behaviour of redundant manipulators. *Robotica*, 17, 1999.

[12] L. Žlajpah. On Obstacle Avoidance for Redundant Robots Using Tactile Sensors. In I.J. Rudas and L. Madarász, editors, *1999 IEEE Int. Conf. on Intelligent Engineering Systems, INES'99*, pages 415 – 420, Stara Lesna, Slovakia, 1999.

[13] H.P. Xie, R.V. Patel, S. Kalaycioglu, and H. Asmer. Real-time collision avoidance for a redundant manipulator in an unstructured environment. In *Proc. Intl. Conf. On Intelligent Robots and Systems IROS'98*, pages 1925 – 1930, Victoria, Canada, 1998.