Collision detection and avoidance method for two cooperative robot manipulators

A A Kabanov* and D A Tokarev

Sevastopol State University, 33, Universitetskaya str., Sevastopol, 299053, Russia

* KabanovAleksey@gmail.com

Abstract. The paper examines a collisions detection and avoidance problem for two cooperative robot manipulators. The proposed collision detection method is based on the division of workspace into small discrete volume spaces. Each of the discrete volumes have the status “occupied” or “free”. The control system generates restriction signals when the motion along the calculated path will bring the manipulator into occupied volume spaces. A step-by-step algorithm of the collision detection system is given.

1. Introduction

Prevention of collisions and execution of safe actions while working is a fundamental problem in the control systems design of redundant manipulators. It is necessary to ensure avoidance of collisions of the manipulator with the possible obstacles of the environment as well as evasion of critical states of the manipulator. A transition to the critical state may lead to the damage of the devices of the manipulator. It can often be seen in the literature that the problem of searching for critical states is provided for the case when only one manipulator is used. Methods of analysis of manipulator critical states are reduced to the analysis of manipulator kinematic model and searching of the coordinates and their combinations under which critical states appear [1-6]. The control systems are designed in such way that the redundant manipulator will not damage itself avoiding the critical states or collisions with other objects [1-5].

The solution of the problem of determination and prevention of dangerous movement of the manipulator is getting considerably more difficult in case when robot has several manipulators. Mutual interactions of manipulators must be highlighted: direct physical contact (collision) of the manipulators; interaction through the object of manipulation; interaction through the working environment (relevant to underwater robots).

There is no common approach to defining the intersection area of working manipulators and to analyze their possible collisions. In each case of specific robot a design of control system is held. The designed control system excludes such collisions of the manipulators under their cooperative work. As the result constraints on the positions and roll angles of the manipulators, links and joints are introduced.

In [5] the authors presented the problem of dual-arm manipulation control with the help of the method of constraint based programming. The collision avoidance problem is reduced to the analysis of interaction of only the end-effectors of two manipulators. This method is used for generating online motion plans for the dual arm robot manipulator. The developed method provides the avoidance of collisions with exterior obstacles as well as the avoidance of end-effectors collision. It is evident that this approach does not guarantee prevention of collisions between the links of the manipulators.

Another way is based on the searching of control signals which exclude the movement of two
manipulators with the intersection of the trajectories of all the manipulators links. In [7] an approach which is based on the modelling of the manipulator and the surrounding objects with the help of simple geometry primitives like “cylinder” or “sphere” is suggested. Using such geometric representations the authors introduced the properties of different types of collisions. A simple method of detection of such collisions is obtained. In [8] a so-called collision map is suggested in the capacity of collision zone for detection of the manipulators collision and their representation in two-dimensional coordinates frame. At that collision prevention is achieved by means of time scheduling of the robot trajectory which has received a command to move and by using an escape instruction for the robot which becomes an obstacle in the path of the other one. For real-time generation of the movements of the manipulators which allow avoiding collisions a robust reactive algorithm named “The skeleton algorithm” was suggested in [9]. According to this algorithm, the robot structure is divided into segments, so that random points can be chosen and controlled. The algorithm defines collision points on such segments and generates the suitable control for collision avoidance. This algorithm requires sensor information about the position of each link in order to schedule the trajectory of avoidance of the possible collisions.

So, the problem of collision detection and avoidance for two cooperative manipulators is necessary for defining dynamically changing constraints for robot drives. Let us highlight some important aspects for solving this problem. One of the ways to avoid manipulators collision is the division of the workspace so that the trajectories of cooperative manipulators will not cross. As the cooperative manipulators work in the united workspace, the space division is realized nominally in the form of defining the prohibited zones of each manipulator. Implementation of this approach is performed by software tools. The workspace is divided into two subspaces where each of the manipulators can work without direct contact with the other one. This workspace division approach can solve the considered problem, but it limits control system capabilities. An extension of this approach is the online dynamic realignment of the division boundary of the workspace taking into account the current position of the manipulators.

When designing the control system, apart from the division of workspace, one should also prioritize the robot command execution. In order to take into account the mutual influence of the manipulators during their cooperative work it is reasonable to use the master-slave approach [2]. According to this approach one of the manipulators is appointed a “master” and the other becomes a “slave”. If the master is in the interaction area, the prohibition zone of the slave manipulators changes according to the positions of the master manipulator. Thus, the slave manipulators free the interaction area of workspace in case it is occupied by the master. The advantage of the master-slave approach is that the realignment of the areas happens online during the cooperative work of the robots.

Apart from the master-slave approach, an occupied-free approach can also be applied. Some analogy of the occupied-free approach can be seen in [1] (with the provision that the approach is applicable for a two-dimensional space). In our opinion, the extension of the occupied-free approach into three-dimensional workspace is admissible. According to this approach, a conventional state of the common interaction area is introduced into the system in the form of two options “occupied” and “free”. If one of the manipulators is in the common interaction area, then this area gets the status “occupied”. At that, the given area is defined as a prohibited zone for the other manipulators. If the occupying manipulator leaves the common area, this area gets the status “free”.

In this research the issues of the development of collisions detection and avoidance algorithm for two cooperative robot manipulator are considered. The workspace of the manipulators is divided into small discrete volume spaces. Each of the discrete volumes can have the status “occupied” or “free”. The control system generates restrictions when the motion along the calculated path will bring the manipulator into the occupied volume spaces. This approach has several advantages: a simple implementation, checking the position of all manipulators joints and links, an avoidance of the manipulators collision between themselves and self-collision. Also the proposed method has no need to replace the manipulators joints and links with volumetric geometric primitives.

The rest of the paper is as follows. Section 2 contains the statement of the problem, the description of the method of detection and avoidance of manipulators collisions is presented in Section 3, the main conclusions is given in section 4.
2. Statement of the problem
The control problem of the two manipulators cooperative work consists in achieving the set position in the workspace by the manipulators end-effectors under additional conditions:
- avoid collisions of manipulators;
- avoid damage caused by manipulators self-collisions.

Assumption 1. The coordinate frame is attached to the basis where manipulators are fixed. Motor drives have a sufficient power margin ensuring reasonable dynamic characteristics of the manipulators (using low-level controllers). This allows ones to build a control on the basis of the kinematic model without taking into account the dynamics. Robot has feedback sensors which allows defining the current coordinates of the manipulators joints in workspace.

The kinematic model of the two cooperative robot manipulators can be presented in the form [3]:

\[ q_i = u_i, \quad i = 1, n, \]

where \( q_i \) are the manipulator joint angles, \( u_i \) are the joint velocities, \( i \) is a serial number of the manipulator joint. We assume that the number of the first manipulator joints is \( n_1 \), the second manipulator joints is \( n_2 \), so total number of joints is \( n_1 + n_2 = n \).

The joint velocities of manipulators \( u_i \) is an input control signal for manipulators control system design. Setting the velocities vector \( u \), the control of robot manipulators is performed. At that, the goal of the control is the transition of the joints from the current position into the set one. Thus, we introduce a function that connects the manipulators joint velocities and their position in workspace. Taking this into account, let us rewrite (1) in the form:

\[ \dot{q}_i = u_i, \quad u_i = F(a_i), \]

where \( F(a_i) \) is a function, define connection the spin rate \( u_j \) and the position of the joint \( a_j \).

Expressions (1) and (2) get constraints in the form of maximum values of the roll angles \( |q_i| \leq Q_j \) and velocities \( |u_j| \leq U_j \), \( Q_j \) and \( U_j \) are the boundaries of the angles and velocities correspondingly. The velocities constraints are determined by the physical properties of the drives. The angles constraints are determined by the construction of manipulators (singular positions). Additional constraints are placed on the critical positions under which the collision of the manipulators will happen.

In the common case the angles constraints are connected with the position of the manipulators joints \( a_j \) in workspace \( \mathbb{R}^3 \). Thus, the statement of the problem comes to searching of the control signals \( u_j \) which will put the system into the set final position of the joints \( a_j \) without manipulators collisions.

3. Development of the collision avoidance algorithm for manipulators
In our opinion, it is reasonable to perform an analysis of the manipulators collision with respect to their kinematic model supplemented by manipulators geometric parameters. As a result, each link of the manipulator can be presented as a set of points in workspace \( S \subset \mathbb{R}^3 \). In the common case the number of this points is unlimited in view of continuity of the obtained surface \( \partial S \) of the solid body (manipulator). This leads to the need for searching an analytical solution of bodies intersection problem. Taking into account that there are more than two bodies and their form can be unrestricted, the searching of such solution can be rather difficult. In the current paper in order to overcome these difficulties, an approach of searching the intersection (i.e. collision) of the manipulators links and joints has been suggested. It is based on discretisation of the workspace into unit volumes with sufficiently small sampling rate. The sampling rate size is determined by the required accuracy and accessible computational resources. Each manipulator in workspace appears as a finite number of the unit volumes. Thus, a transition from the manipulator in the form of a surface of solid body in workspace to the
manipulator in the form of a unit volumes set in workspace is made. These volumes will be characterized by their spatial coordinates $x, y, z$ in the obtained discrete workspace as well as a logical variable that defines the volume states (“occupied” and “free”):

$$v_k = (x_k, y_k, z_k, e_k).$$

where $x_k, y_k, z_k$ are spatial coordinates of $k$-th unit volume of the manipulators workspace, $e_k$ is a logical variable with two possible states

$$e_k = \begin{cases} 0 & \text{when the volume is free}, \\ 1 & \text{when the volume is occupied}. \end{cases}$$

Figure 1 images schematically a three-dimensional workspace with two manipulators in Cartesian coordinate frame divided into discrete unit volumes. Each of the obtained unit volumes has three spatial coordinates within the frames of the set discrete scale and a logical variable with the states “1” or “0” (occupied or free). Geometrical interpretation of the logical state can be highlighted in colour for descriptive reasons.

![Figure 1. Discrete workspace with schematic position of two manipulators.](image)

A schematic representation of the manipulators is the result of the fact that they are described by the kinematic model in Denavit–Hartenberg notation. This approach simplifies the representation of the real mechanism to its interpretation in the form of joints $p_i$ and connecting lines among them (by analogy with skeleton algorithm [9]). Connecting lines are constructed under assumption that the coordinates of the joints are known.

Each joint of the manipulator $p_i$ has spatial coordinates $(x_i, y_i, z_i)$, where $i = 1, 2, \ldots, n$. Transformation which connects $i$-th joint with $(i-1)$-th joint is described by the homogenous $4 \times 4$ matrix [3]:

$$T_i = \begin{bmatrix}
1 & 0 & 0 & x_i \\
0 & 1 & 0 & y_i \\
0 & 0 & 1 & z_i \\
0 & 0 & 0 & 1
\end{bmatrix}.$$
\begin{equation}
  i^{-1}A_i(\theta_i) = \begin{bmatrix}
  C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\
  S\alpha_i C\theta_i & C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\
  0 & S\theta_i & C\alpha_i & d_i \\
  0 & 0 & 0 & 1
\end{bmatrix},
\end{equation}

where $C\theta_i$ and $S\theta_i$ are $\cos(\theta_i)$ and $\sin(\theta_i)$, respectively, $\theta_i$ is an angle that the axis $X_{i-1}$ must be rotated around the axis $Z_{i-1}$, so that it will become codirected with the axis $X_i$, $\alpha_i$ is an angle that the axis $Z_{i-1}$ must be rotated around the axis $X_i$ so that it will become codirected with the axis $Z_i$. With the help of the given matrix we connect joint coordinates $p_i$ with the previous joint $p_{i-1}$:

\begin{equation}
  p_{i-1} = i^{-1}A_i p_i,
\end{equation}

where $p_{i-1} = (x_{i-1}, y_{i-1}, z_{i-1})^T$, $p_i = (x_i, y_i, z_i)^T$.

Homogenous matrix which determines the position of the $i$-th coordinate frame with respect to the basic coordinate frame is represented as a product of sequence of homogenous transformation matrixes $i^{-1}A_i$ [3]:

\begin{equation}
  T = \prod_{j=1}^{i} i^{-1}A_j(\theta_j) = \begin{bmatrix}
  0 & R_j & \theta_j \\
  0 & 1 & \theta_j
\end{bmatrix},
\end{equation}

where the top left submatrix $R_j$ has the dimension 3x3, $0p_i$ is a vector which connects the origin of the basic frame with the origin of the $i$-th frame, it has the dimension 3x1.

Thus, specifying a set of transformation matrices, we can calculate the current joint position $p_i$ of the manipulators. By substituting the values of the joint angles for the next position in these matrixes, we get the forecast of the manipulators joints positions.

In this paper we take the approach that one manipulator in relation to the other acts as a three-dimensional prohibited area. When the robot works, this area changes according to the current position of the manipulators. The calculation of the manipulators movement is performed with allowance for the dynamically changeable prohibition areas. The intersection of two bodies (manipulators collision) presented in the form of a set of unit volumes is detected by coincidence of the coordinates of two or more unit volumes. The changes of the coordinates of unit volumes which belong to one manipulator are performed with allowance for its kinematic model. The implementation of this method is reduced to writing an algorithm that searches for the same volume coordinates for two objects.

As an example in figure 1 joints $p_i$ of the left and right manipulators are denoted through $p_{L0}, p_{L1}, p_{L2}$ and $p_{R0}, p_{R1}, p_{R2}$ correspondingly. The coordinates of the manipulators joints must be converted into the coordinates of occupied volumes (in which these joints are situated). For this purpose, let us multiply the coordinates $p_i$ by the discretisation coefficient $K_d$ which equals the number of discrete volumes in units of the basic frame and drop the fractional part.

For instance, for the manipulator joint $p_{R0}$ in figure 1 we get:

\begin{equation}
  p_{R0} = (x_{R0}, y_{R0}, z_{R0})^T = (6.5, 4.5, 0.5, 1)^T.
\end{equation}

In the present example, the discretisation scale equals 1, consequently, the coefficient $K_d = 1$. Then we get the following coordinates of the unit volume occupied by the joint $p_{R0}$:

\begin{equation}
  v_{R0} = (6, 4, 0, 1)^T.
\end{equation}

For the rest of the joints let us act the same way.

Apart from the volumes occupied by the joints, there are volumes through which the connecting lines
go through. Let us also denote these unit volumes as occupied. The function according which the given 
lines are constructed is obtained from the equation of a straight line in space:

\[
\frac{y - y_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0} = \frac{z - z_0}{z_1 - z_0},
\]

where \((x_0, y_0, z_0), (x_1, y_1, z_1)\) are coordinates of joints \(p_0\) and \(p_1\) in three-dimensional coordinate frame 
correspondingly. For \(x \in [x_0, x_1]\) we get \(y(x)\) and \(z(x)\):

\[
\begin{cases}
y(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0} (x - x_0), \\
z(x) = z_0 + \frac{z_1 - z_0}{x_1 - x_0} (x - x_0).
\end{cases}
\]

As a result, in the workspace of manipulators \(V\) we get two discrete subspaces \(V_L, V_R\) occupied by 
left and right manipulators correspondingly. Thus, the problem of collisions detection is reduced to the 
problem of searching the intersection of discrete spaces \(V_L, V_R\). Given the discreteness of the 
coordinates of these subspaces, the search for their intersection is reduced to the search for at least two 
unit volumes with coinciding coordinates.

For a clear demonstration and ease of understanding of the proposed approach let us give an example 
for two manipulators in two-dimensional workspace. Figure 2 demonstrates a two-dimensional 
projection of the three-dimensional workspace with two manipulators. Each manipulator consists of two 
links and three joints. The joints are connected by the direct lines. Joints coordinates are calculated 
according to (3). In the given example the workspace discretization coefficient equals \(K_\alpha = 1\).

![Figure 2](image-url)
Using the proposed approach, we get three discrete areas for each manipulator joint (occupied discrete areas are marked in colour in figure 2a). The joints have connections in the form of direct lines. The areas, connecting lines pass through, should be denoted as occupied as well. Let us make the calculation of the coordinates of these discrete areas using the straight line equation in a plane:

\[ \frac{y - y_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0}, \]

where \((x_0, y_0),(x_1, y_1)\) are the coordinates of the joints \(p_0\) and \(p_1\) respectively. From the last expression for \(x \in [x_0, x_1]\) one can obtain

\[ y(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0} (x - x_0). \tag{4} \]

Using (4) with the numeric values of the coordinates of the joints \(p_{L0}, p_{L1}, p_{L2}\) and \(p_{R0}, p_{R1}, p_{R2}\) we obtain the coordinates of the points of lines connecting the joints of the manipulators. As a result, we get discrete areas occupied by the connecting lines. Let us mark them as occupied as well (figure 2b).

The control system design requires computation of the areas of manipulators location for detecting and predicting the collisions. In case if the movement of the manipulator leads it to the occupied area, such control signal will not be applied and re-computation with regard to the detected collision will be required. An example of the intersection of two manipulators areas is depicted in figure 2c. A conflict area is marked in red colour.

The algorithm of the collision detection and avoidance system using this approach looks like this:

1. Formation of numerical values of the joint angles that will move the manipulator from the current position into the new one.
2. Computation of the space coordinates of the joints and the connecting lines.
3. Workspace discretization and searching of the coordinates of the unit volumes where there are more than one object (joint, connecting line etc.).
4. In case of absence of intersection, movement is permitted, otherwise, recalculation of control and repetition of steps 1-4 is performed.

Figure 3 shows a block-scheme of the described collision detection and avoidance algorithm.

**Figure 3.** A block-scheme of the collision detection and avoidance algorithm.
4. Conclusion
In the current work an algorithm of collision detection and avoidance for two robot manipulators under cooperative work is proposed. The given algorithm is based on the division of the manipulators workspace into small discrete volumes. These volumes are characterized by their spatial coordinates $x, y, z$ in the obtained discrete workspace as well as a logical variable that defines the volume states (“occupied” or “free”). Those volumes where the manipulators joints and the connecting lines are located are denoted as occupied. When computing the control, a forecast of the future position of the the manipulators joints is made. In case a new position of the manipulator leads it to an occupied area, re-computation is made.

Consideration of the geometrical features of the manipulators is a necessary upgrade of the current approach. Variation of workspace discretization under division into unit volumes can be applied with an allowance for the geometry of the links and joints of the manipulators. A kinematic model represents a number of joints with the known coordinates and links among them. With that, the real values of the drives, motors, junctions and other constructive elements are not taken into consideration. Addition of geometrical figures which describe the volume representation of these elements will allow taking into account the collision of the manipulators in conditions of a real design. Each of the figures is attached to the coordinates of the manipulator links. The following realization of the approach is similar to the mentioned above. This method allows one to predict the collision of two or more manipulators and exclude critical states at the stage of control design.

Acknowledgments
The reported study was supported by the Ministry of Education and Science of the Russian Federation (agreement № 14.578.21.0264, project RFMEFI57818X0264).

References
[1] Jefferies M E, Yeap W 2008 Robotics and cognitive approaches to spatial mapping (New York: Springer-Verlag)
[2] Siciliano B, Khatib O 2008 Springer handbook of robotics (New York: Springer-Verlag)
[3] Lewis F, Dawson D.M., Abdallah C T 2004 Robot manipulator control: theory and practice (New York, Marcel Dekker)
[4] Tarbouriech S, Navarro B, Fraisse P, Crosnier A, Cherubini A 2018 Dual-arm relative tasks performance using sparse kinematic control IROS: Intelligent Robots and Systems pp.6003-8
[5] Wang Y, Ogren P, Simth C, Vina F, Karayiannidis Y 2014 Dual arm manipulation using constraint based programming Proc. 19th World Congress IFAC (Cape Town) vol 19 pp 311-9
[6] Kabanov A A., Balabanov A N 2018 The modeling of an anthropomorphic robot arm MATEC Web of Conferences 224 02034
[7] Patel R V, Shadpey F, Ranjbaran F, Angeles J 2005 A collision-avoidance scheme for redundant manipulators: Theory and experiments Journal of Robotic Systems 22 (12) pp 737–57
[8] Afaghani A Y, Aiyama Y 2013 On-line collision avoidance between two robot manipulators using collision map and simple escaping method Proc. IEEE/SICE Int. Symposium on System Integration, (Kobe) pp 105-10
[9] Santis A, Albu-Schäffer A, Ott C, Siciliano B, Hirzinger G 2007 The skeleton algorithm for real-time collision avoidance of a humanoid manipulator interacting with humans Proc. 2007 IEEE/ASME international conference on advanced intelligent mechatronics (Zurich) 9871732