A Methodology to Determine the High Performance Area of 6R Industrial Robot

Nianfeng Wang, Zhifei Zhang(✉), and Xianmin Zhang

Guangdong Provincial Key Laboratory of Precision Equipment and Manufacturing Technology, School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China
{menfwang,zhangxm}@scut.edu.cn, red13531@gmail.com

Abstract. To find an area where the robot can obtain a higher kinematic performance is a meaningful work for path planning and off-line programming. In this paper, a methodology was proposed to determine the high performance area (HPA) of a 6R industrial robot. Monte Carlo method was used to get a point cloud of the reference point of the end effector. The manipulability measure was selected as a performance index to filter the point cloud. A grid was defined to approach the filtered point cloud and its boundary was then extracted and smoothed. An example was presented to show that the proposed methodology is feasible to determine the HPA of a 6R industrial robot.

Keywords: High performance area · Performance index · Monte Carlo method

1 Introduction

The High Performance Area (HPA) was defined as, the cross-section of the high performance part of the workspace with a plane on which the first joint axis lies [10]. It was proposed as a measure of the high performance part of the workspace, and can provide designer with practical information about the performance of the robot structures. It can also provide guidance for path planning and workpiece placement.

To determine the high performance part of the workspace, a proper performance index must be selected first to tell this part of the workspace from the rest. Many performance indices were proposed in the previous works [5,7]. Yoshikawa proposed a measure of manipulability based on Jacobian of the robot [12]. The manipulability measure indicates the ability of the robot to move and apply forces in arbitrary directions. The condition index is another measure that was defined as the reciprocal of the condition number of the Jacobian and used to evaluate the control accuracy [8]. Other indices, like minimum singular value, isotropic index and so forth, were discussed in detail in this work [7].

The manipulability measure is one of the most commonly used performance indices [8]. Some researchers argued that the manipulability measure is a better indicator of dexterity than condition number or minimum singular value,

Y. Huang et al. (Eds.): ICIRA 2017, Part II, LNAI 10463, pp. 554–563, 2017.
DOI: 10.1007/978-3-319-65292-4_48

because the manipulability measure considers the motion of the end effector in all directions instead of one or two [9]. Based on these ideas, we choose the manipulability measure as the performance index.

There are many methods to determine the workspace of a robot, which can be roughly classified into two types. Analytical methods determine closed form descriptions of workspace boundary [11], but these methods are usually complicated because of the nonlinear equations and inverse kinematic computation [2]. Numerical methods can get a approximate boundary of the workspace without inverse kinematic and are easier to operate [1,6]. Therefore, the methodology proposed by this paper will base on numerical method.

2 Determining the HPA

The overall processes of the proposed methodology to determine the HPA will be discussed in detail in this part.

Since the HPA was defined as the cross-section of the high performance part of the workspace with a plane on which the first joint axis lies, we can choose the xoz plane of the base frame as the plane where we compute the HPA. And we can know from previous work that the manipulability index does not depend on the first DOF [4]. Therefore, to simplify the computation, the first joint variable will always be set to 0.

2.1 The Manipulability Measure

The manipulability measure of a robot was defined as a scalar value w by [12]

$$w = \sqrt{det(J(\theta)J^T(\theta))} \tag{1}$$

where $J(\theta)$ and $J^T(\theta)$ are the Jacobian and transpose of the Jacobian respectively, and the det means determinant of a matrix.

If the robot has no redundant degree of freedom, which is like the case of this paper, the 6R industrial robot, the equation above becomes

$$w = |J(\theta)| \tag{2}$$

The manipulation Jacobian with respect to the base frame is given by [3]

$$\begin{aligned} {}^0J &= \left[{}^0J_1(q)\ {}^0J_2(q)\ \dots\ {}^0J_n(q)\right] \quad \text{with} \\ {}^0J_1(q) &= \begin{bmatrix} z_{l-1} \times^{l-1} p_n \\ z_{l-1} \end{bmatrix} \quad \text{for a revolute joint} \\ {}^0J_1(q) &= \begin{bmatrix} z_{l-1} \\ 0 \end{bmatrix} \quad \text{for a prismatic joint,} \end{aligned} \tag{3}$$

where z_{l-1} is the z-axis vector of the frame of link $l-1$ expressed in the base frame, ${}^{l-1}p_n$ is the position of the link n with respect to the frame of link $l-1$ and expressed in the base frame, $\times$ is the vector cross product.

Taking the end effector into account, ${}^{l-1}p_n$ in Eq. (3) will be replaced by ${}^{l-1}p_{end}$. The latter is the position of the reference point of the end effector with respect to the frame of link $l-1$ and expressed in the base frame.

2.2 Obtain and Filter the Point Cloud

Now, we will use the Monte Carlo method to get a point cloud from a set of random joint vector. Every component of these joint vectors was obtained from the following equation

$$q_l = \begin{cases} q_{l,low} + Rand \times (q_{l,up} - q_{l,low}), & l \neq 1 \\ 0, & l = 1 \end{cases} \tag{4}$$

where $q_{l,low}$ and $q_{l,up}$ is the lower and upper bound value of the l-th joint, $Rand$ stands for a random value between 0 and 1.

After that, we can compute the position and orientation of the end effector reference point by the forward kinematic equation

$$T_{end} = \left(\prod_{l=0}^{dof} {}^{l}T_{l+1}(q_l) \right) {}^{dof}T_{end} \tag{5}$$

where dof is the degree of freedom that the robot has, ${}^{l}T_{l+1}$ is the homogeneous matrix between the frames of the consecutive link l and link $l+1$, and ${}^{dof}T_{end}$ is the homogeneous matrix of the end effector with respect to the frame of the last link.

For each point that was computed from Eq. (5), the Jacobian and manipulability measure of the robot in this point will be calculated. Then a threshold of manipulability measure will be introduced to handle those points. This step can simply be expressed by the following formula:

$$w = \begin{cases} w, & if\,(w > h) \\ 0, & otherwise \end{cases} \tag{6}$$

where h is the threshold, it is chosen in dependence on what kind of job the robot will do, which will not be discussed by this paper.

After the filtering, those points with a manipulability measure of 0 will be abandoned.

Although the first joint variable was set to 0, those points in the filtered point cloud may not locate at a single plane. This is because of the existence of the end effector, of which the reference point has a translation with respect to the frame of the last link.

Therefore, a transformation will be performed to transform these points into a single plane. The transformation is expressed by

$$p_s = \left[\sqrt{x_w^2 + y_w^2},\ 0,\ z_w \right]^T \tag{7}$$

where $p_w = [x_w, y_w, z_w]^T$ is the coordinates of a point from the filtered point cloud before the transformation, and p_s is the coordinates of this point after the transformation.

The nature of this transformation is a rotation around the z-axis of the base frame, or around the rotation axis of the first joint in terms of a 6R industrial robot. And as a result, all point will be transformed to xoz plane of the base frame. So actually this transformation will not affect the result we will get later because of the invariance of the manipulability measure and the symmetry characteristic of the workspace that we have discussed before.

2.3 Grid Handling

All those points in the point cloud are now at a single plane, i.e. xoz plane of the base reference frame. To define a grid in the plane that can overlap all points of the point cloud, the maximum dimensions of the point cloud along axis x and axis z have to be determined first. These two values can simply be calculated by the following formula:

$$\begin{cases} X_m = \max\limits_{1 \leqslant k \leqslant N}(x_k) - \min\limits_{1 \leqslant k \leqslant N}(x_k) \\ Z_m = \max\limits_{1 \leqslant k \leqslant N}(z_k) - \min\limits_{1 \leqslant k \leqslant N}(z_k) \end{cases} \tag{8}$$

where N is the number of points in the point cloud after filtering.

Now we define a grid with $m \times n$ cells, the size of each cell is Δx and Δz which can be considered as resolution dimensions. With lesser Δx and Δz we can obtain a more accurate result while the computation cost will be higher, so proper values should be chosen. In addition, each cell will have a flag value of either 0 or 1.

The number of cells along each axis can be derived by

$$\begin{cases} N_x = \left\lfloor \frac{X_m}{\Delta x} \right\rfloor + 1 \\ N_z = \left\lfloor \frac{Z_m}{\Delta z} \right\rfloor + 1 \end{cases} \tag{9}$$

where the operator $\lfloor\ \rfloor$ denotes the floor function that returns the nearest integer less than or equal to the real-valued argument inside the brackets.

After the grid was created, we decide which cell of the grid that a point p_k from the point cloud will locate in by

$$\begin{cases} i_k = \left\lfloor \frac{X_k}{\Delta x} \right\rfloor \\ j_k = \left\lfloor \frac{Z_k}{\Delta z} \right\rfloor \end{cases} \tag{10}$$

where i_k and j_k are the indices along x-axis and z-axis of the cell n_{i_k,j_k}, i.e. the cell where point p_k locates.

Then cell n_{i_k,j_k} will be colored and its flag value v_{i_k,j_k} will be set to 1. After all points from the point cloud have been handled like this, those cells with no point locates in will not be colored and their flag value remains 0.

2.4 Boundary Extraction

To extract the boundary of the cells that have been colored, two steps of sweeping operation will be applied.

In the first step, the sweeping operation is within a given row, each cells of this row will be swept to find the first and the last colored cells, which are called outer cells. The sweeping starts from cell $n_{0,j}$ to cell $n_{N_x,j}$. When we find a cell of which the index $i_{j,F}$ satisfies the following formula

$$\begin{cases} v_{i,j} = 0 & when \quad i < i_{j,F} \\ v_{i,j} = 1 & when \quad i = i_{j,F} \end{cases} \tag{11}$$

where $v_{i,j}$ is the flag value of cell $n_{i,j}$, then this cell $n_{i_{j,F},j}$ will be considered as the first colored cell of this row. Similarly, when we find a cell of which the index $i_{j,L}$ satisfies the following formula

$$\begin{cases} v_{i,j} = 0 & when \quad i > i_{j,L} \\ v_{i,j} = 1 & when \quad i = i_{j,L} \end{cases} \tag{12}$$

then this cell $n_{i_{j,L},j}$ will be considered as the last colored cell of the row. If all cells of a row have been swept and no proper $i_{j,F}$ and $i_{j,L}$ was found, then this row will be abandoned.

In the second step, the sweeping operation is within the whole grid, each row will be swept and its outer cells are considered to form the boundary. The indices of the outer cells of each row will be modified with the consideration of indices of the outer cells of the previous and following rows. This process expressed by

$$i_{j,F} = \begin{cases} i_{j,F}, & if \quad j < 2 \; or \; j > N_z - 2 \\ \min(i_{j-1,F}, i_{j+1,F}), & if \quad i_{j,F} < i_{j-1,F} \; and \; i_{j,F} < i_{j+1,F} \\ \max(i_{j-1,F}, i_{j+1,F}), & if \quad i_{j,F} > i_{j-1,F} \; and \; i_{j,F} > i_{j+1,F} \\ i_{j,F}, & otherwise \end{cases} \tag{13}$$

and

$$i_{j,L} = \begin{cases} i_{j,L}, & if \quad j < 2 \; or \; j > N_z - 2 \\ \min(i_{j-1,L}, i_{j+1,L}), & if \quad i_{j,L} < i_{j-1,L} \; and \; i_{j,L} < i_{j+1,L} \\ \max(i_{j-1,L}, i_{j+1,L}), & if \quad i_{j,L} > i_{j-1,L} \; and \; i_{j,L} > i_{j+1,L} \\ i_{j,L}, & otherwise \end{cases} \tag{14}$$

The purpose of this step is to ensure that there are no extreme sharps and hollows in the boundary we get.

Figure 1 illustrates the process of the two steps of operation.

After the two steps of operation, the center point of the outer cells of each row will be computed and added to an array, called boundary point array. Connecting points in this array in order with straight line, we will get a rough boundary of those colored cells.

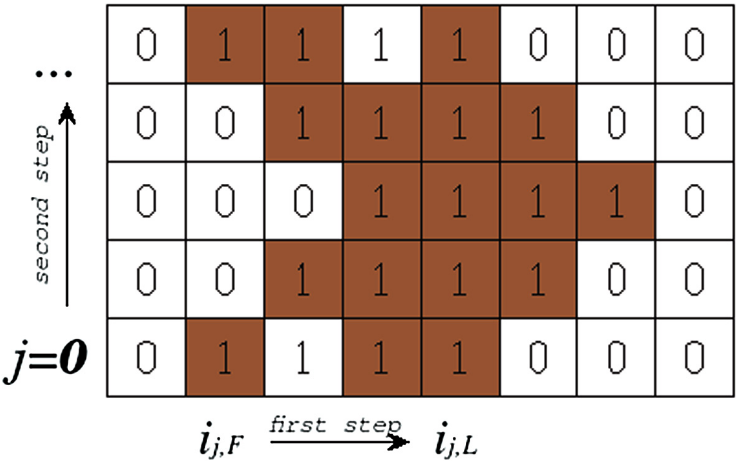


Fig. 1. The two steps of sweeping operation

2.5 Boundary Smoothing

The basic idea of smoothing the boundary is that processing the points in the boundary array with the least square method.

Suppose that the number of points in the boundary point array is N_B. Now we divide these points into N_l small point lists, each list will have $N_p = N_B/N_l$ points. For each list, the least square method is applied to determine the regression line.

$$Ls\colon \bar{z} = bx + a \quad \text{with} \quad \begin{cases} b = \dfrac{\sum\limits_{i=1}^{N_p}(x_i-\bar{x})(z_i-\bar{z})}{\sum\limits_{i=1}^{N_p}(x_i-\bar{x})^2} \\ a = \bar{z} - b\bar{x} \end{cases} \tag{15}$$

For two regression lines that their corresponding point lists are neighbor, their point of intersection will be computed. Then we will get N_l intersection points since we have N_l of regression lines. Connecting these intersection points in order with straight line and we will get the final approximate boundary of HPA. Moreover, rotating this boundary around the rotation axis of first joint, we will get the high performance part of the workspace.

3 An Example

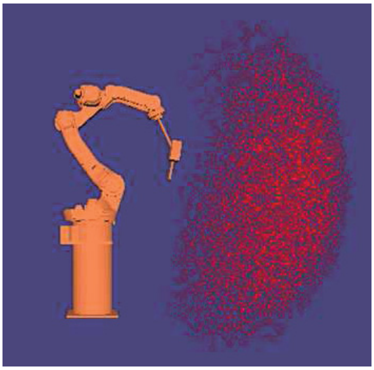
In this part, an example will be presented to show that the proposed methodology is feasible to determine the HPA of a 6R industrial robot. The 6R industrial robot that this example will study on is Motoman MH12, of which the D-H parameters are given by Table 1.

Firstly, 5,000,000 random points were generated, and a threshold was applied to those points. For some applications that need more dexterity, like welding,

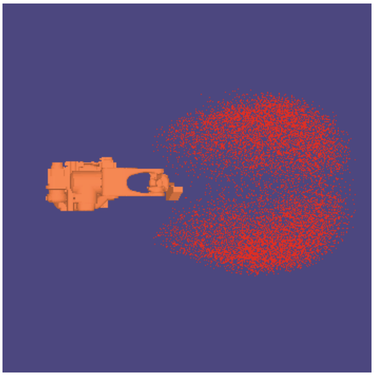
Table 1. The D-H parameters of Motoman MH12

Link	a/mm	$\alpha/\circ$	d/mm	$\theta/\circ$
1	0	0	0	−170/+170
2	155	−90	0	−180/+65
3	614	180	0	−85/+150
4	200	−90	−640	−150/+150
5	0	90	0	−135/+90
6	0	−90	0	−210/+210

the threshold h should be bigger, while for applications like carrying, a less h is proper. In this example, we will not aim at any application, so a moderate threshold 0.5 was chosen. Figure 2 shows the point cloud after being processed by the threshold.



(a) Front view



(b) Top view

Fig. 2. The point cloud filtered from 5,000,000 points by a manipulability threshold of 0.5

Then a grid of which the cell size is $10\,mm \times 10\,mm$ was created. Figure 3 shows the colored cells of the grid we have got after the grid handling step.

After that, the two steps of sweeping operation was applied, a boundary array was got and its points were connected in order with straight line. The rough boundary was illustrated in Fig. 4.

Finally, the points in the boundary array were divided into 100 small lists and their regression lines were computed. 100 intersection points were obtained and connected in order with straight line to form the final boundary of HPA, which was illustrated in Fig. 5.

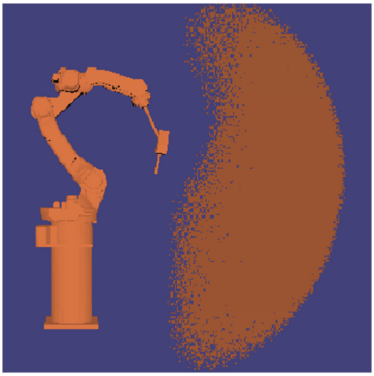


Fig. 3. The colored cells with the cell size of $10\,mm \times 10\,\text{mm}$ (Color figure online)

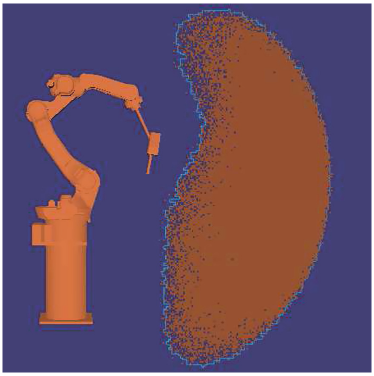


Fig. 4. Boundary before smoothing and the corresponding colored cells (Color figure online)

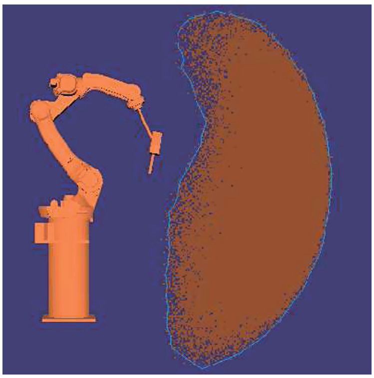


Fig. 5. The smoothed boundary and the corresponding colored cells (Color figure online)

The whole computation process takes about 10 min with a intel core i5 cpu, obtaining and filtering of the point cloud take the most of time, while grid handling and boundary extraction are much faster relatively.

Moreover, we can get several boundaries by using different thresholds. Figure 6 shows boundaries with their corresponding thresholds are 0.5, 0.6 and 0.7 from outside to inside respectively. We can get a rough idea about the distribution of the dexterity of the robot from this figure.

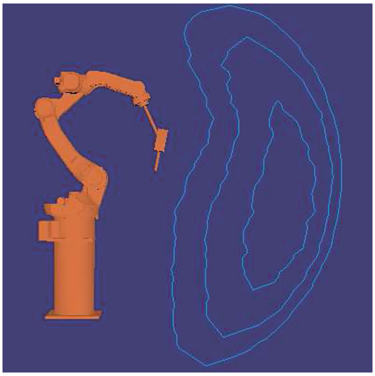


Fig. 6. Boundaries with the corresponding thresholds are 0.5, 0.6 and 0.7 from outside to inside respectively

4 Conclusion

In this paper, a methodology was proposed to determine the HPA of a 6R industrial robot. A point cloud was generated by Monte Carlo method and filtered by a manipulability threshold. A grid was defined to approach the filtered point cloud and the boundary was then extracted and smoothed with least square method. An example was presented to show that the proposed methodology is feasible to determine the HPA of a 6R industrial robot.

Acknowledgements. This work is supported by National Natural Science Foundation of China (Grant Nos. 51205134, 91223201), Science and Technology Program of Guangzhou (Grant No. 2014Y2-00217), Research Project of State Key Laboratory of Mechanical System and Vibration (MSV201405), the Fundamental Research Funds for the Central University (Fund No. 2015ZZ007) and Natural Science Foundation of Guangdong Province (S2013030013355).

References

1. Alciatore, D.G., Ng, C.C.D.: Determining manipulator workspace boundaries using the Monte Carlo method and least squares segmentation. ASME Robot.: Kinemat. Dyn. Control. **72**, 141–146 (1994)

2. Cao, Y., Qi, S., Lu, K., Zang, Y., Yang, G.: An integrated method for workspace computation of robot manipulator. In: International Joint Conference on Computational Sciences and Optimization, pp. 309–312 (2009)
3. Fu, K.S., Gonzalez, R.C., Lee, C.S.G.: Robotics: Control, Sensing, Vision, and Intelligence. Robotica (1987)
4. Gotlih, K., Troch, I.: Base invariance of the manipulability index. Robotica **22**(4), 455–462 (2004)
5. Kucuk, S., Bingul, Z.: Comparative study of performance indices for fundamental robot manipulators. Robot. Auton. Syst. **54**(7), 567–573 (2006)
6. Ottaviano, E.: A fairly general algorithm to evaluate workspace characteristics of serial and parallel manipulators. Mech. Based Des. Struct. Mach. **36**(1), 14–33 (2008)
7. Patel, S., Sobh, T.: Manipulator performance measures - a comprehensive literature survey. J. Intell. Robot. Syst. **77**(3), 1–24 (2015)
8. Salisbury, J.K., Craig, J.J.: Articulated handsforce control and kinematic issues. Int. J. Robot. Res. **1**(1), 4–17 (1981)
9. Tadokoro, S., Kimura, I., Takamori, T.: A dexterity measure for trajectory planning and kinematic design of redundant manipulators. In: Conference of IEEE Industrial Electronics Society, IECON 1989, vol. 2, pp. 415–420 (1989)
10. Valsamos, C., Moulianitis, V.C., Synodinos, A.I., Aspragathos, N.A.: Introduction of the high performance area measure for the evaluation of metamorphic manipulator anatomies. Mech. Mach. Theory **86**, 88–107 (2015)
11. Wang, Y., Chirikjian, G.S.: A diffusion-based algorithm for workspace generation of highly articulated manipulators. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, vol. 2, pp. 1525–1530 (2002)
12. Yoshikawa, T.: Manipulability of robotic mechanisms. Int. J. Robot. Res. **4**(4), 3–9 (1985)