Optimal Usage of Robot Manipulators

Behnam Kamrani[1], Viktor Berbyuk[2], Daniel Wäppling[3],
Xiaolong Feng[4] and Hans Andersson[4]
[1]*MSC.Software Sweden AB, SE-42 677, Gothenburg*
[2]*Chalmers University of Technology, SE-412 96, Gothenburg*
[3]*ABB Robotics, SE-78 168, Västerås*
[4]*ABB Corporate Research, SE-72178, Västerås*
Sweden

1. Introduction

Robot-based automation has gained increasing deployment in industry. Typical application examples of industrial robots are material handling, machine tending, arc welding, spot welding, cutting, painting, and gluing. A robot task normally consists of a sequence of the robot tool center point (TCP) movements. The time duration during which the sequence of the TCP movements is completed is referred to as cycle time. Minimizing cycle time implies increasing the productivity, improving machine utilization, and thus making automation affordable in applications for which throughput and cost effectiveness is of major concern. Considering the high number of task runs within a specific time span, for instance one year, the importance of reducing cycle time in a small amount such as a few percent will be more understandable.

Robot manipulators can be expected to achieve a variety of optimum objectives. While the cycle time optimization is among the areas which have probably received the most attention so far, the other application aspects such as energy efficiency, lifetime of the manipulator, and even the environment aspect have also gained increasing focus. Also, in recent era virtual product development technology has been inevitably and enormously deployed toward achieving optimal solutions. For example, off-line programming of robotic work-cells has become a valuable means for work-cell designers to investigate the manipulator's workspace to achieve optimality in cycle time, energy consumption and manipulator lifetime.

This chapter is devoted to introduce new approaches for optimal usage of robots. Section 2 is dedicated to the approaches resulted from translational and rotational repositioning of a robot path in its workspace based on response surface method to achieve optimal cycle time. Section 3 covers another proposed approach that uses a multi-objective optimization methodology, in which the position of task and the settings of drive-train components of a robot manipulator are optimized simultaneously to understand the trade-off among cycle time, lifetime of critical drive-train components, and energy efficiency. In both section 2 and 3, results of different case studies comprising several industrial robots performing different tasks are presented to evaluate the developed methodologies and algorithms. The chapter is concluded with evaluation of the current results and an outlook on future research topics on optimal usage of robot manipulators.

2. Time-Optimal Robot Placement Using Response Surface Method

This section is concerned with a new approach for optimal placement of a prescribed task in the workspace of a robotic manipulator. The approach is resulted by applying response surface method on concept of path translation and path rotation. The methodology is verified by optimizing the position of several kinds of industrial robots and paths in four showcases to attain minimum cycle time.

2.1 Research background

It is of general interest to perform the path motion as fast as possible. Minimizing motion time can significantly shorten cycle time, increase the productivity, improve machine utilization, and thus make automation affordable in applications for which throughput and cost effectiveness is of major concern.

In industrial application, a robotic manipulator performs a repetitive sequence of movements. A robot task is usually defined by a robot program, that is, a robot pathconsisting of a set of robot positions (either joint positions or tool center point positions) and corresponding set of motion definitions between each two adjacent robot positions. *Path translation* and *path rotation* terms are repeatedly used in this section to describe the methodology. Path translation implies certain translation of the path in x, y, z directions of an arbitrary coordinate system relative to the robot while all path points are fixed with respect to each other. Path rotation implies certain rotation of the path with θ, φ, ψ angles of an arbitrary coordinate system relative to the robot while all path points are fixed with respect to each other. Note that since path translation and path rotation are relative concepts, they may be achieved either by relocating the path or the robot.

In the past years, much research has been devoted to the optimization problem of designing robotic work cells. Several approaches have been used in order to define the optimal relative robot and task position. A *manipulability measure* was proposed (Yoshikawa, 1985) and a modification to Yoshikawa's manipulability measure was proposed (Tsai, 1986) which also accounted for proximity to joint limits. (Nelson & Donath, 1990) developed a gradient function of manipulability in Cartesian space based on explicit determination of manipulability function and the gradient of the manipulability function in joint space. Then they used a modified method of the steepest descent optimization procedure (Luenberger, 1969) as the basis for an algorithm that automatically locates an assembly task away from singularities within manipulator's workspace.

In aforementioned works, mainly the effects of robot kinematics have been considered.Once a robot became employed in more complex tasks requiring improved performance, e .g., higher speed and accuracy of trajectory tracking, the need for taking into account robot dynamics becomes more essential (Tsai, 1999).

A study of time-optimal positioning of a prescribed task in the workspace of a 2R planar manipulator has been investigated (Fardanesh & Rastegar, 1988). (Barral et al., 1999) applied the simulated annealing optimization method to two different problems: robot placement and point-ordering optimization, in the context of welding tasks with only one restrictive working hypothesis for the type of the robot. Furthermore, a state of the art of different methodologies has been presented by them.

In the current study, the dynamic effect of the robot is considered by utilizing a computer model which simulates the behavior and response of the robot, that is, the dynamic models of the robots embedded in ABB's IRC5 controller. The IRC5 robot controller uses powerful, configurable software and has a unique dynamic model-based control system which provides self-optimizing motion (Vukobratovic, 2002).

To the best knowledge of the authors, there are no studies that directly use the response surface method to solve optimization problem of optimal robot placement considering a general robot and task. In this section, a new approach for optimal placement of a prescribed task in the workspace of a robot is presented. The approach is resulted by path translation and path rotation in conjunction with response surface method.

2.2 Problem statement and implementation environment

The problem investigated is to determine the relative robot and task position with the objective of time optimality. Since in this study a relative position is to be pursued, either the robot, the path, or both the robot and path may be relocated to achieve the goal. In such a problem, the robot is given and specified without any limitation imposed on the robot type, meaning that any kind of robot can be considered. The path or task, the same as the robot, is given and specified; however, the path is also general and any kind of path can be considered. The optimization objective is to define the optimal relative position between a robotic manipulator and a path. The optimal location of the task is a location which yields a minimum cycle time for the task to be performed by the robot.

To simulate the dynamic behavior of the robot, RobotStudio is employed, that is a software product from ABB that enables offline programming and simulation of robot systems using a standard Windows PC. The entire robot, robot tool, targets, path, and coordinate systems can be defined and specified in RobotStudio. The simulation of a robot system in RobotStudio employs the ABB Virtual Controller, the real robot program, and the configuration file that are identical to those used on the factory floor. Therefore the simulation predicts the true performance of the robot.

In conjunction with RobotStudio, Matlab and Visual Basic Application (VBA) are utilized to develop a tool for proving the designated methodology. These programming environments interact and exchange data with each other simultaneously. While the main dataflow runs in VBA, Matlab stands for numerical computation, optimization calculation, and post processing. RobotStudio is employed for determining the *path admissibility* boundaries and calculating the cycle times. Figure 1 illustrates the schematic of dataflow in the three computational environments.

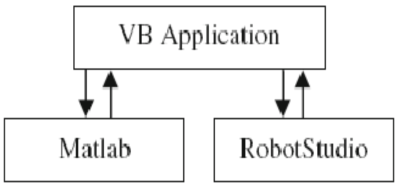


Fig. 1. Dataflow in the three computational tools

2.3 Methodology of time-optimal robot placement

Basically, the path position relative to the robot can be modified by translating and/or rotating the path relative to the robot. Based on this idea, translation and rotation approaches are examined to determine the optimal path position. The algorithms of both approaches are considerably analogous. The approaches are based on the response surface method and consist of following steps. First is to pursue the *admissibility boundary*, that is, the boundary of the area in which a specific task can be performed with the same robot configuration as defined in the path instruction. This boundary is obviously a subset of the general robot operability space that is specified by the robot manufacturer. The computational time of this step is very short and may take only few seconds. Then experiments are performed on different locations of admissibility boundary to calculate the cycle time as a function of path location. Next, optimum path location is determined by using constrained optimization technique implemented in Matlab. Finally, the sensitivity analysis is carried out to increase the accuracy of optimum location.

Response surface method (Box et al., 1978; Khuri & Cornell, 1987; Myers & Montgomery, 1995) is, in fact, a collection of mathematical and statistical techniques that are useful for the modeling and analysis of problems in which a response of interest is influenced by several decision variables and the objective is to optimize the response. Conventional optimization methods are often cumbersome since they demand rather complicated calculations, elaborate skills, and notable simulation time. In contrast, the response surface method requires a limited number of simulations, has no convergence issue, and is easy to use.

In the current robotic problem, the decision variables consist of x, y, and z of the reference coordinates of a prescribed path relative to a given robot base and the response of interest to be minimized is the task cycle time. A so-called full factorial design is considered by 27 experiment points on the path admissibility boundaries in three-dimensional space with original path location in center. Figure 2 graphically depicts the original path location in the center of the cube and the possible directions for finding the admissibility boundary.

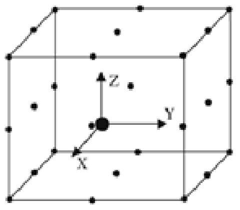


Fig. 2. Direction of experiments relative to the original location of path

Three-dimensional bisection algorithm is employed to determine the path admissibility region. The algorithm is based on the same principle as the bisection algorithm for locating the root of a three-variable polynomial. Bisection algorithm for finding the admissibility boundary states that each translation should be equal to half of the last translation and translation direction is the same as the last translation if all targets in the path are admissible; otherwise, it is reverse. Herein, targets on the path are considered admissible if the robot manipulator can reach them with the predefined configurations. Note that in this step the robot motion between targets is not checked.

Since the target admissibility check is only limited to the targets and the motion between the targets are not simulated, it has a low computational cost. Additionally, according to practical experiments, if all targets are admissible, there is a high probability that the whole path would also be admissible. However, checking the target admissibility does not guarantee that the whole path is admissible as the joint limits must allow the manipulator to track the path between the targets as well. In fact, for investigating the path admissibility, it is necessary to simulate the whole task in RobotStudio to ascertain that the robot can manage the whole task, i.e., targets and the path between targets.

To clarify the method, an example is presented here. Let's assume an initial translation by 1.0 *m* in positive direction of *x* axis of reference coordinate system is considered. If all targets after translation are admissible, then the next translation would be 0.5 *m* and in the same (+*x*) direction; otherwise in opposite (–*x*) direction. In any case, the admissibility of targets in the new location is checked and depending on the result, the direction for the next translation is decided. The amount of new translation would be then 0.25 *m*. This process continues until a location in which all targets are admissible is found such that the last translation is smaller than a certain value, that is, the considered tolerance for finding the boundary, e.g., 1 mm.

After finding the target admissibility boundary in one direction within the decided tolerance, a whole task simulation is run to measure the cycle time. Besides measuring the cycle time, it is also controlled if the robot can perform the whole path, i.e., investigating the *path admissibility* in addition to targets admissibility. If the path is not admissible in that location, a new admissible location within a relaxed tolerance can be sought and examined. The same procedure is repeated in different directions, e.g. 27 directions in full-factorial method, and by that, a matrix of boundary coordinates and vector of the corresponding cycle times are casted.

A quadratic approximation function provides proper result in most of response surface method problems (Myers & Montgomery, 1995), that is:

$$\begin{aligned} f(x,y,z) = \; & b_0 + b_1x + b_2y + b_3z + \ldots && \text{(linear terms)} \\ & b_4xy + b_5yz + b_6xz + \ldots && \text{(interaction terms)} \\ & b_7x^2 + b_8y^2 + b_9z^2 && \text{(quadratic terms)} \end{aligned} \tag{1}$$

By applying the following mapping:

$$\begin{aligned} x = x_1 \;;\quad y = x_2 \;;\quad z = x_3 \\ xy = x_4 \;;\; xz = x_5 \;;\; yz = x_6 \\ x^2 = x_7 \;;\; y^2 = x_8 \;;\; z^2 = x_9 \end{aligned} \tag{2}$$

Eq. 1 can be expressed in linear form and by matrix notation as:

$$Y = XB + e \tag{3}$$

where Y is the vector of cycle times, X is the design matrix of boundaries, B is the vector of unknown model coefficients of {b0, b1, b2, …, b9}, and e is the vector of errors. Finally, B can be estimated using the least squares method, minimizing of $L=e^Te$, as:

$$B = (X^TX)^{-1}X^TY \tag{4}$$

In the next step of the methodology, when the expression of cycle time as a function of a reference coordinate (*x, y, z*) is given, the minimum of the cycle times subject to the determined boundaries is to be found. The *fmincon* function in Matlab optimization toolbox is used to obtain the minimum of a constrained nonlinear function. Note that, since the cycle time function is a prediction of the cycle time based on the limited experiments data, the obtained value (for the minimum of cycle time) does not necessarily provide the global minimum cycle time of the task. Moreover, it is not certain yet that the task in optimum location is kinematically admissible. Due to these reasons, the minimum of the cycle time function can merely be considered as an '*optimum candidate*.'

Hence, the optimum candidate must be evaluated by performing a confirmatory task simulation in order to, first investigate whether the location is admissible and second, calculate the actual cycle time. If the location is not admissible, the closest location in the direction of the translation vector is pursued such that all targets are admissible. This new location is considered as a new optimum candidate and replaced the old one. This procedure may be called sequential backward translation.

Due to the probability of inadmissible location and as a work around, the algorithm, by default, seeks and introduces several optimum candidates by setting different search areas in *fmincon* function. All candidate locations are examined and cycle times are measured. If any location is inadmissible, that location is removed from the list of optimum candidate. After examining all the candidates, the minimum value is selected as the final optimum. If none of the optimum candidates is admissible, the shortest cycle time of experiments is selected as optimum. In fact, and in any case, it is always reasonable to inspect if the optimum cycle time is shorter than all the experiment cycle times, and if not, the shortest cycle time is chosen as the local optimum.

As the last step of the methodology the sensitivity analysis of the obtained optimal solution with respect to small variations in x, y, z coordinates can be interesting to study. This analysis can particularly be useful when other constraints, for example space inadequacy, delimit the design of robotic cell. Another important benefit of this analysis is that it usually increases the accuracy of optimum location, meaning that it can lead to finding a precise local optimum location.

The sensitivity analysis procedure is generally analogous to the main analysis. However, herein, the experiments are conducted in a small region around the optimum location. Also, note that since it is likely that the optimum point, found in the previous step, is located on (or close to) the boundary, defining a cube around a point located on the boundary places some cube sides outside the boundary. For instance, when the shortest cycle time of the experiments is selected as the local optimum, the optimum location is already on the admissibility boundary. In such cases, as a work around, the nearest admissible location in the corresponding direction is considered instead.

Note that the sensitivity analysis may be repeated several times in order to further improve the results. Figure 3 provides an overview of the optimization algorithm.

As was mentioned earlier, the path position relative to the robot can be modified by translating as well as rotating the path. In path translation, the optimal position can be achieved without any change in path orientation. However, in path rotation, the optimal path orientation is to be sought. In other words, in path rotation approach the aim is to obtain the optimum cycle time by rotating the path around the x, y, and z axes of a local frame. The local frame is originally defined parallel to the axes of the global reference frame on an arbitrary point. The origin of the local reference frame is called the rotation center. Three sequential rotation angles are used to rotate the path around the selected rotation center. To calculate new coordinates and orientations of an arbitrary target after a path rotation, a target of T on the path is considered in global reference frame of X–Y–Z which is demonstrated in Fig. 4. The target T is rotated in local frame by a rotation vector of (θ, ϕ, ψ) which yields the target T′.

If the targets in the path are not admissible after rotating by a certain rotation vector, the boundary of a possible rotation in the corresponding direction is to be obtained based on the bisection algorithm. The matrices of experiments and cycle time response are built in the same way as described in the path translation section and the cycle time expression as a function of rotation angles of (θ, ϕ, ψ) is calculated. The optimum rotation angles are obtained using Matlab *fmincon* function. Finally, sensitivity analyses may be performed. A procedure akin to path translation is used to investigate the effect of path rotation on the cycle time.

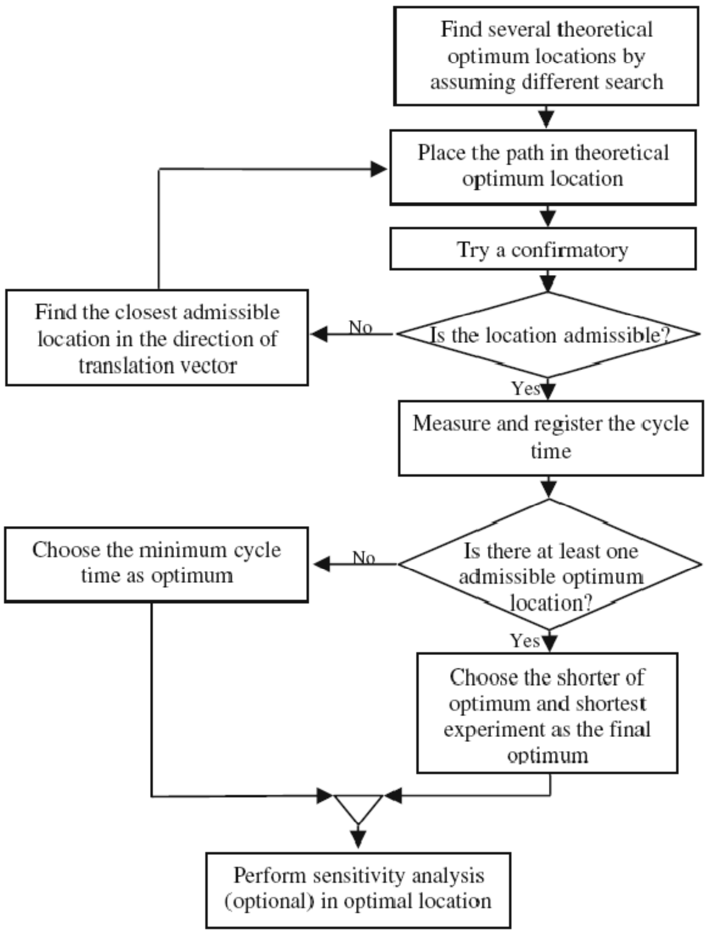

Find several theoretical optimum locations by assuming different search
Place the path in theoretical optimum location
Try a confirmatory
Is the location admissible?
No
Find the closest admissible location in the direction of translation vector
Yes
Measure and register the cycle time
Is there at least one admissible optimum location?
No
Choose the minimum cycle time as optimum
Yes
Choose the shorter of optimum and shortest experiment as the final optimum
Perform sensitivity analysis (optional) in optimal location

Fig. 3. Flowchart diagram of the optimization algorithm

Although the algorithm of path rotation is akin to path translation, two noticeable differences exist. Although the algorithm of path rotation is akin to path translation, two noticeable differences exist. First, in the rotation approach, the order of rotations must be observed. It can be shown that interchanging orders of rotation drastically influences the resulting orientation. Thus, the order of rotation angles must be adhered to strictly (Haug, 1992). Consequently, in the path rotation approach, the optimal rotation determined by sensitivity analysis cannot be added to the optimal rotation obtained by the main analysis, whereas in the translation approach, they can be summed up to achieve the resultant translation vector. Another difference is that, in the rotation approach, the results logically depend on the selection of the rotation center location, while there is no such dependency in the path translation approach. More details concerning path rotation approach can be found in (Kamrani et al., 2009).

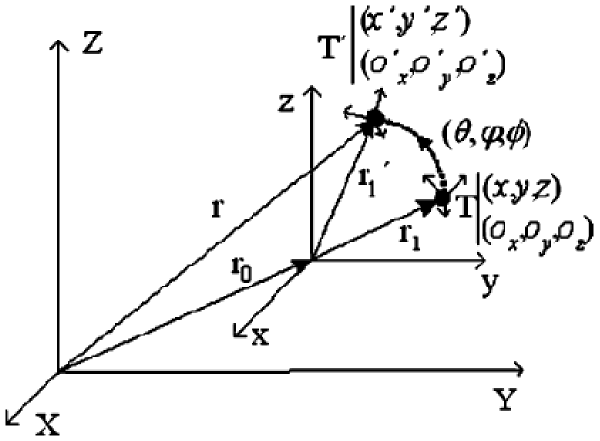


Fig. 4. Rotation of an arbitrary target T in the global reference frame

2.4 Results on time-optimal robot placement

To evaluate the methodology, four case studies comprised of several industrial robots performing different tasks are proved. The goal is to optimize the cycle time by changing the path position. A coordinate system with its origin located at the base of the robot, x-axis pointing radially out from the base, z-axis pointing vertically upwards, is used for all the cases below.

2.4.1 Path Translation

In this section, obtained by path translation approach are presented.

2.4.1.1 Case 1

The first test is carried out using the ABB robot IRB6600-225-175 performing a spot welding task composed of 54 targets with fixed positions and orientations regularly distributed around a rectangular placed on a plane parallel to the *x*-*y* plane (parallel to horizon). A view of the robot and the path in its original location is depicted in the Fig. 5. The optimal location of the task in a boundary of (±0.5 *m*, ±0.8 *m*, ±0.5 *m*) is calculated using the path translation approach to be as (Δx, Δy, Δz) = (0 *m*, 0.8 *m*, 0 *m*). The cycle time of this path is reduced from originally 37.7 seconds to 35.7 seconds which implies a gain of 5.3 percent cycle time reduction. Fig. 6 demonstrates the robot and path in the optimal location determined by translation approach.

2.4.1.2 Case 2

The second case is conducted with the same ABB IRB6600-225-175 robot. The path is composed of 18 targets and has a closed loop shape. The path is shown in the Fig. 7 and as can be seen, the targets are not in one plane. The optimal location of the task in a boundary of (±1.0 *m*, ±1.0 *m*, ±1.0 *m*) is calculated using the path translation approach to be as (Δx, Δy, Δz) = (-0.104 *m*, -0.993 *m*, 0.458 *m*). The cycle time of this path is reduced from originally 6.1 seconds to 5.6 seconds which indicates 8.3 percent cycle time reduction.

2.4.1.3 Case 3

In the third case study, an ABB robot of type IRB4400L10 is considered performing a typical machine tending motion cycle among three targets which are located in a plane parallel to the horizon. The robot and the path are depicted in the Fig. 8. The path instruction states to start from the first target and reach the third target and then return to the starting target. A restriction for this case is that the task cannot be relocated in the y-direction relative to the robot. The optimal location of the task in a boundary of (±1.0 *m*, 0 *m*, ±1.0 *m*) is calculated using the path translation approach to be as (Δx, Δy, Δz) = (0.797 *m*, 0 *m*, -0.797 *m*). The cycle time of this path is reduced from originally 2.8 seconds to 2.6 seconds which evidences 7.8 percent cycle time reduction.

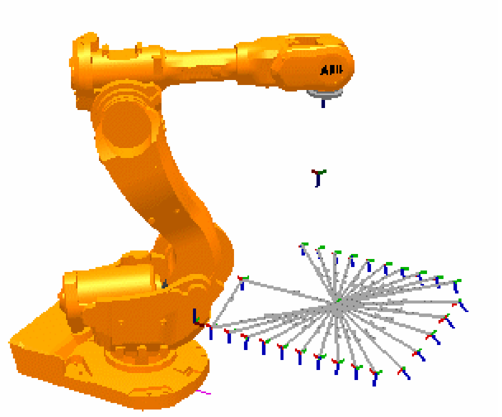


Fig. 5. IRB6600 ABB robot with a spot welding path of case 1 in its original location

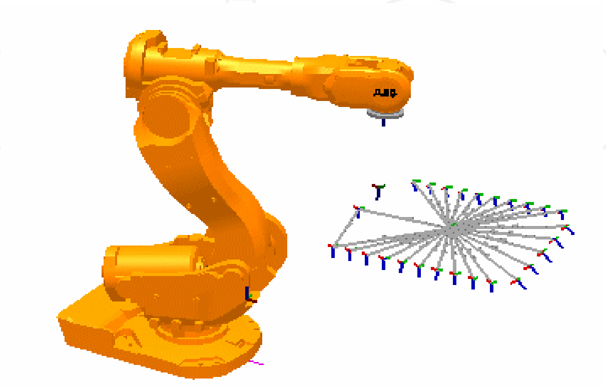


Fig. 6. IRB6600 ABB robot with a spot welding path of case 1 in optimal location found by translation approach

2.4.1.4 Case 4

The forth case is carried out using an ABB robot of IRB640 type. In contrast to the previous robots which have 6 joints, IRB640 has merely 4 joints. The path is shown in the Fig. 9 and comprises four points which are located in a plane parallel to the horizon. The motion instruction requests the robot to start from first point and reach to the forth point and then return to the first point again. The optimal location of the task in a boundary of (±1.0 *m*, ±1.0 *m*, ±1.0 *m*) is calculated using the path translation approach to be as (Δx, Δy, Δz) = (0.2 *m*, 0.2 *m*, -0.8 *m*). The cycle time of this path is reduced from originally 3.7 seconds to 3.5 seconds which gives 5.2 percent cycle time reduction.

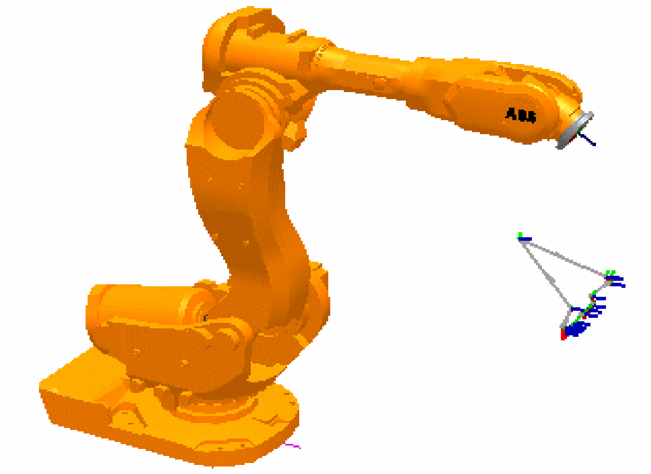


Fig. 7. IRB6600 ABB robot with the path of case 2 in its original location

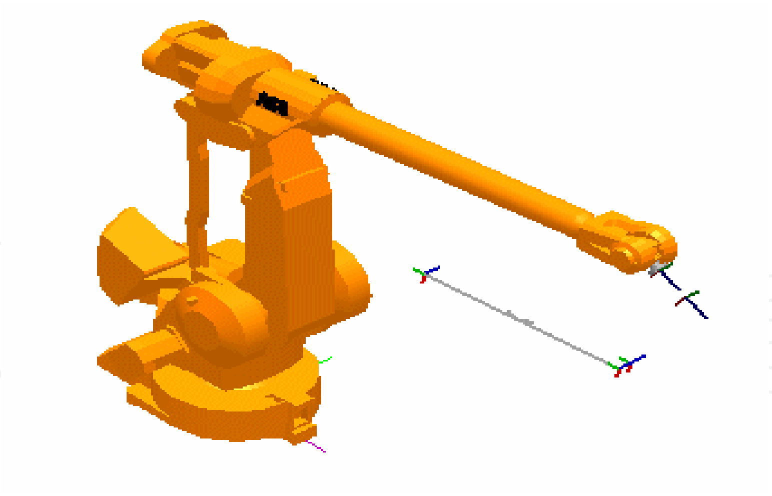


Fig. 8. IRB4400L10 ABB robot with the path of case 3 in its original location

2.4.2 Path Rotation

In this section, results of path rotation approach are presented for four case studies. Herein the same robots and tasks investigated in path translation approach are studied so that comparison between the two approaches will be possible.

2.4.2.1 Case 1

The first case is carried out using the same robot and path presented in section 2.4.1.1. The central target point was selected as the rotation center. The optimal location of the task in a boundary of (±45°, ±45°, ±30°) is calculated using the path rotation approach to be as (Δθ, Δφ, Δψ) = (45°, 0°, 0°). The path in the optimal location determined by rotation approach is shown in Fig. 10. The task cycle time was reduced from originally 37.7 seconds to 35.7 seconds which implies an improvement of 5.3 percent compared to the original path location.

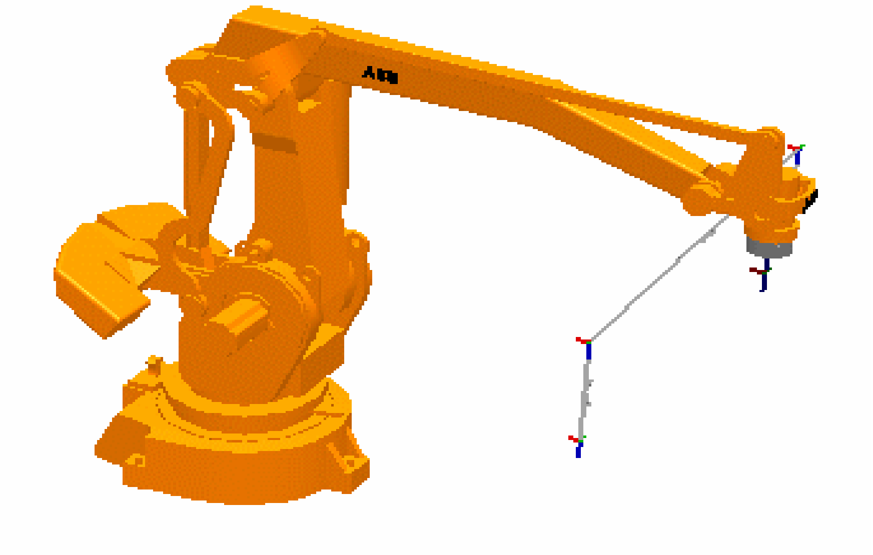


Fig. 9. IRB640 ABB robot with the path of case 4 in its original location

2.4.2.2 Case 2

The second case study is conducted with the same robot and path presented in 2.4.1.2. An arbitrary point close to the trajectory was selected as the rotation center. The optimal location of the task in a boundary of (±45°, ±45°, ±30°) is calculated using the path rotation approach to be as (Δθ, Δφ, Δψ) = (45°, 0°, 0°). The cycle time of this path is reduced from originally 6.0 seconds to 5.5 seconds which indicates 8.3 percent cycle time reduction.

2.4.2.3 Case 3

In the third example the same robot and path presented in section 2.4.1.3 are studied. The middle point of the long side was selected as the rotation center. To fulfill the restrictions outlined in section 2.4.1.3, only rotation around y-axis is allowed. The optimal location of the task in a boundary of (0°, ±90°, 0°) is calculated using the path rotation approach to be as (Δθ, Δφ, Δψ) = (0°, -60°, 0°). Here the sensitivity analysis was also performed. The cycle time of this path is reduced from originally 2.8 seconds to 2.2 seconds which evidences 21 percent cycle time reduction.

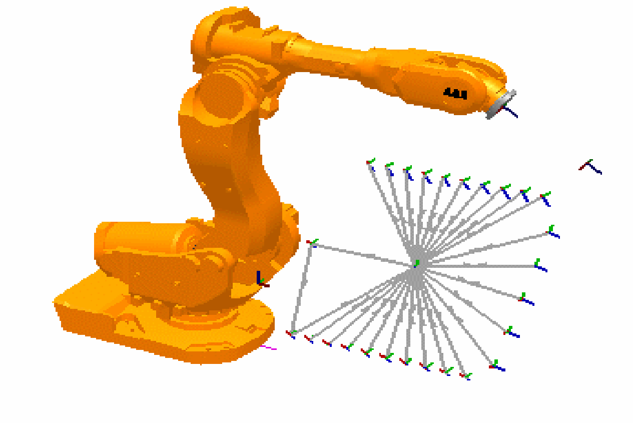


Fig. 10. IRB6600 ABB robot with a spot welding path of case 1 in optimal location found by rotation approach

2.4.2.4 Case 4

The forth case study is carried out with the same robot presented in 2.4.1.4. The point in the middle of a line which connects the first and forth targets was chosen as the rotation center. Due to the fact that the robot has 4 degrees of freedom, only rotation around the z-axis is allowed. The optimal location of the task in a boundary of (0°, 0°, ±45°) is calculated using the path rotation approach to be as $(\Delta\theta, \Delta\varphi, \Delta\psi) = (0°, 0°, 16°)$. In this case the sensitivity analysis was also performed. The cycle time of this path is reduced from originally 3.7 seconds to 3.6 seconds which gives 3.5 percent cycle time reduction.

2.4.3 Summary of the Results of Section 2

The cycle time reduction percentages that are achieved by translation and rotation approaches compared to longest and original cycle time are demonstrated in Fig. 11. The longest cycle time which corresponds to worst performance location is recognized as an existing admissible location that has the longest cycle time, i.e., the longest cycle time among experiments. As can be perceived, a cycle time reduction in range of 8.7 – 37.2 percent is achieved as compared to the location with the worst performance.

Results are also compared with the cycle time corresponding to original path location. This comparison is of interest as the tasks were programmed by experienced engineers and had been originally placed in proper position. Therefore this comparison can highlight the efficiency and value of the algorithm. The results demonstrate that cycle time is reduced by 3.5 - 21.1 percent compared with the original cycle time.

Fig. 11 indicates that both translation and rotation approaches are capable to noticeably reduce the cycle time of a robot manipulator.

A relatively lower gain in cycle time reduction in case four is related to a robot with four joints. This robot has fewer joint than the other tested robots with six joints. Generally, the fewer number of joints in a robot manipulator, the fewer degrees of freedom the robot has. The small variation of the cycle time in the whole admissibility area can imply that this robot has a more homogeneous dynamic behavior. Path geometry may also contribute to this phenomenon.

Also note that cycle time may be further reduced by performing more experiments. Although doing more experiments implies an increase in simulation time, this cost can reasonably be neglected by noticing the amount of time saving, for instance 20 percent in one year. In other word, the increase in productivity in the long run can justify the initial high computational burden that may be present, noting that this is a onetime effort before the assembly line is set up.

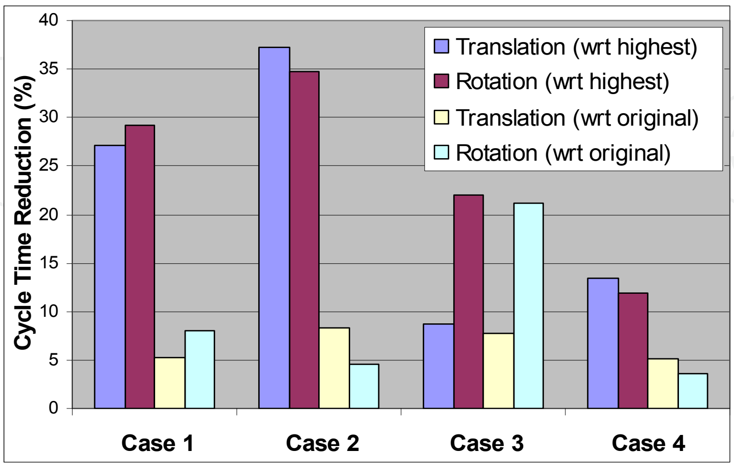


Fig. 11. Comparison of cycle time reduction percentage with respect to highest and original cycle time in four case studies

3. Combined Drive-Train and Robot Placement Optimization

3.1 Research background

Offline programming of industrial robots and simulation-based robotic work cell design have become an increasing important approach for the robotic cell designers. However, current robot programming systems do not usually provide functionality for finding the optimum task placement within the workspace of a robot manipulator (or relative placement of working stations and robots in a robotic cell). This poses two principal challenges: 1) Develop methodology and algorithms for formulating and solving this type of problems as optimization problems and 2) Implement such methodology and algorithms in available engineering tools for robotic cell design engineers.

In the past years, much research has been devoted to the methodology and algorithm development for solving optimization problem of designing robotic work cells. In Section 2, a robust and sophisticated approach for optimal task placement problem has been proposed, developed, and implemented in one of the well-known robot offline programming tool RobotStudio from ABB. In this approach, the cycle time is used as the objective function and the goal of the task placement optimization is to place a pre-defined task defined in a robot motion path in the workspace of the robot to ensure minimum cycle time.

In this section, firstly, the task placement optimization problem discussed in Section 2 will be extended to a multi-objective optimization problem formulation. Design space for exploring the trade-offs between cycle time performance and lifetime of some critical drive-train component as well as between cycle time performance and total motor power consumption are presented explicitly using multi-objective optimization. Secondly, a combined task placement and drive-train optimization (combined optimization will be termed in following texts throughout this chapter) will be proposed using the same multi-objective optimization problem formulation. To authors' best knowledge, very few literature has disclosed any previous research efforts in these two types of problems mentioned above.

3.2 Problem statement

Performance of a robot may be modified by re-setting robot drive-train configuration parameters without any need of modification of hardware of the robot. Performance of a robot depends on positioning of a task that the robot performs in the workspace of the robot. Performance of a robot may therefore be optimized by either optimizing drive-train of the robot (Pettersson, 2008; Pettersson & Ölvander, 2009; Feng et al., 2007) or by optimizing positioning of a task to be performed by the robot (Kamrani et al., 2009).

Two problems will be investigated: 1) Can the task placement optimization problem described in Section 2 be extended to a multi-objective optimization problem by including both cycle time performance and lifetime of some critical drive-train component in the objective function and 2) What significance can be expected if a combined optimization of a robot drive-train and robot task positioning (**simultaneously optimize a robot drive-train and task positioning**) is conducted by using the same multi-objective optimization problem formulation.

In the first problem, additional aspects should be investigated and quantified. These aspects include 1) How to formulate multi-objective function including cycle time performance and lifetime of critical drive-train component; 2) How to present trade-off between the conflicting objectives; 3) Is it feasible and how efficient the optimization problem may be solved; and 4) How the solution space would look like for the cycle time performance vs. total motor power consumption.

In the second problem investigation, in addition to those listed in the problem formulation for the first type of problem discussed above, following aspects should be investigated and quantified: 1) Is it meaningful to conduct the combined optimization? A careful benchmark work is requested; 2) How efficient the optimization problem may be solved when additional drive-train design parameters are included in the optimization problem? Will it be applicable in engineering practice?

It should be noted that, focus of this work presented in Section 3 is on methodology development and validation. Therefore implementation of the developed methodology is not included and discussed. However, the problem and challenge for future implementation of the developed methodology for the combined optimization will be clarified.

3.3 Methodology

3.3.1 Robot performance simulation

A special version of the ABB virtual controller is employed in this work. It allows access to all necessary information, such as motor and gear torque, motor and gear speed, for design use. Based on the information, total motor power consumption and lifetime of gearboxes may be calculated for used robot motion cycle. The total motor power is calculated by summation of power of all motors present in an industrial robot. The individual motor power consumption is calculated by sum of multiplication of motor torque and speed at each simulation time step. The lifetime of gearbox is calculated based on analytical formula normally provided by gearbox suppliers.

3.3.2 Objective function formulation

The task placement optimization has been formulated as a multi-objective design optimization problem. The problem is expressed by

$$\min F(\boldsymbol{DV}) = w_1 \times CT_{norm}(\boldsymbol{DV}) + \ w_2 \times 1/LT_{norm}(\boldsymbol{DV}) \qquad (5)$$

where CT_{norm} is a normalzied cycle time, calculated by

$$CT_{norm} = CT/CT_{original} \qquad (6)$$

CT is the cycle time at each function evaluation in the optimization loop. $CT_{original}$ is the cycle time of the robot motion cycle with original task placement and original drive-train parameter setup for combined optimization. LT_{norm} is a normalized lifetime of gearbox of some selected critical axis. It is calculated by

$$LT_{norm} = LT/LT_{original} \qquad (7)$$

LT is the lifetime of some critical gearbox selected based on the actual usage of the robot at each function evaluation in the optimization loop. $LT_{original}$ is the lifetime of the selected gearbox of the robot motion cycle with original task placement and original drive-train parameter setup for combined optimization. w_1 and w_2 are two weighting factors employed in the weighted-sum approach for multi-objective optimization (Ölvander, 2001). $\boldsymbol{DV}$ is a design variable vector.

Two optimization case studies have been conducted. Robot task placement optimization with the design variable vector defined as

$$\boldsymbol{DV} = \ [\Delta X, \Delta Y, \Delta Z]^T \qquad (8)$$

and combined optimization with the design variable vector defined as

$$\boldsymbol{DV} = \ [\Delta X, \Delta Y, \Delta Z, DV_1, DV_2, \ldots DV_n]^T \qquad (9)$$

where $DV_1, DV_2, DV_3, \ldots\ DV_n$ are the drive-train configuration parameters, while $\Delta X, \Delta Y, \Delta Z$ are the *change* in translational coordinates of all robot targets defining the position of a task.

3.3.3 Optimizer: ComplexRF

The optimization algorithm used in this work is the Complex method proposed by Box (Box, 1965). It is a non-gradient method specifically suitable for this type of simulation-based optimization. Figure 12 shows the principle of the algorithm for an optimization problem consisting of two design variables. The circles represent the contour of objective function values and the optimum is located in the center of the contour. The algorithm starts with randomly generating a set of design points (see the sub-figure titled “Start”). The number of the design points should be more than the number of design variables. The worst design point is replaced by a new and better design point by reflecting through the centroid of the remaining points in the complex (see the sub-figure titled “1. Step”). This procedure repeats until all design points in the complex have converged (see last two sub-figures from left). This method does not guarantee finding a global optimum. In this work, an improved version of the Complex, or normally referred to as ComplexRF, is used, in which a level of randomization and a forgetting factor are introduced for improvement of finding the global optimum (Krus et al., 1992; Ölvander, 2001).

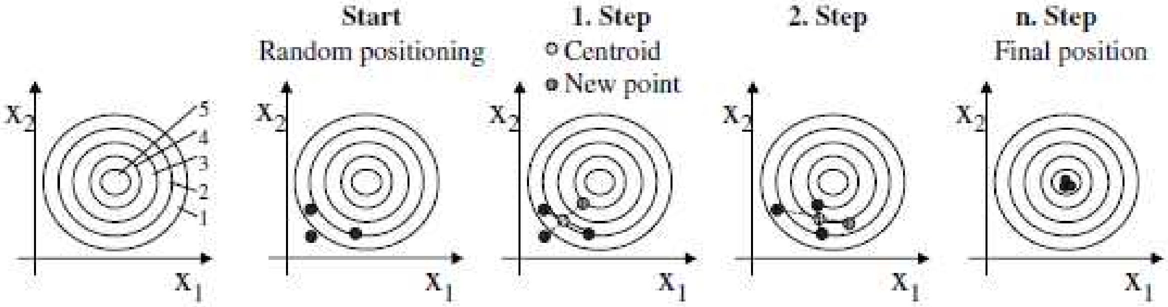


Fig. 12. The progress of the Complex method for a two dimensional example, with the optimum located in the center of the circles (Reprinted with permission from Dr. Johan Ölvander)

3.3.4 Workflow

The workflow of the proposed methodology starts with an optimizer generating a set of design variables. The variables defining robot task placement are used to manipulate the position of the robot task. The variables defining robot drive-train parameters are used to manipulate the drive-train parameters. The ABB robot motion simulation tool is run using the new task position and new drive-train setup parameters. Simulation results are used for computing objective function values. A convergence criterion is evaluated based on the objective function values. This optimization loop is terminated when either the optimization is converged or the limit for maximum number of function evaluations is reached. Otherwise, the optimizer analyzes the objective function values and proposes a new trial set of design variable values. The optimization loop continues until the convergence criterion is met.

3.4 Results on combined optimization

3.4.1 Case-I: Optimal robot usage for a spot welding application

In this case study, an ABB IRB6600-255-175 robot is used. The robot has a payload handling capacity of 175 kg and a reach of 2.55 m. A payload of 100 kg is defined in the robot motion cycle. The robot motion cycle used is a design cycle for spot welding application. The motion cycle consists of about 50 robot tool position targets. Maximum speed is programmed between any adjacent targets. A graphical illustration of the robot motion cycle is shown in Figure 5.

3.4.1.1 Task placement optimization

Only path translation is employed in the task placement optimization. Three design variables ΔX, ΔY, and ΔZ are used. They are added to all original robot targets so that the original placement of the robot task may be manipulated by ΔX in X coordinates, by ΔY in Y coordinates, and by ΔZ in Z corodinates. The limits for the path translation are

$$\begin{aligned} \Delta X &\in (-0.1\, m, 0.1\, m) \\ \Delta Y &\in (0\, m\,, 0.8\, m) \\ \Delta Z &\in (-0.1\, m, 0.1\, m) \end{aligned} \tag{10}$$

The weighting factors w_1 and w_2 in objective function (5) are set to $w_1 = 150$ and $w_2 = 100$ in this task placement optimization.

The convergence curve of the task placement optimization is shown in Figure 13(a). The optimization is well converged after about 100 function evaluations. The total optimization time is about 15 min on a portable PC with Intel(R) Core(TM) 2 Duo CPU T9600 @ 2.8 GHz.

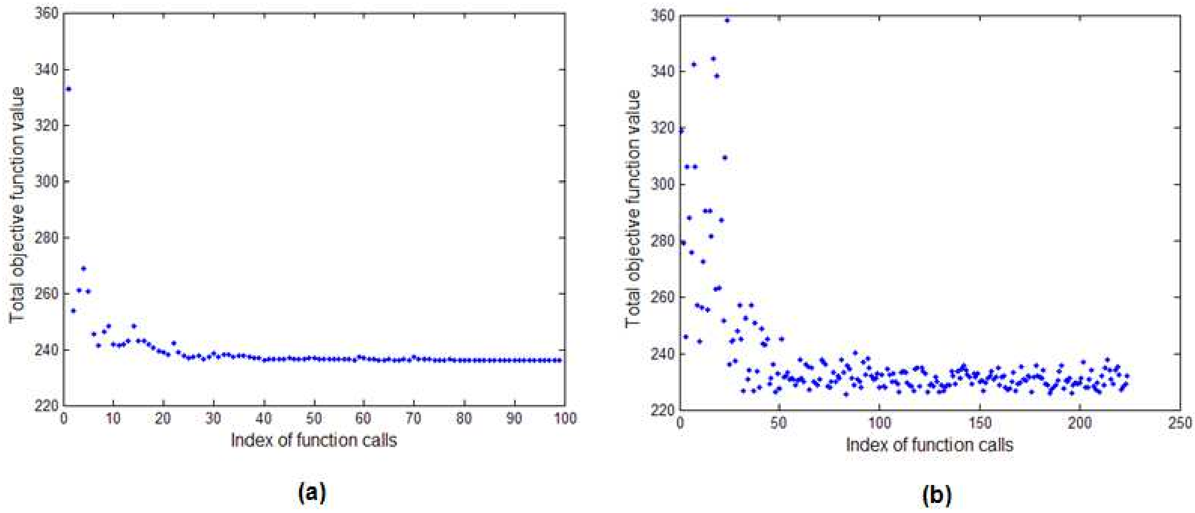


Fig. 13. Convergence curve. (a) for optimal task placement and (b) for combined optimization, (ABB IRB6600-255-175 robot)

Figure 14(a) shows the solution space of normalized lifetime of a critical gearbox as function of normalized cycle time. The cross symbol in blue color indicates the coordinate representing normalized lifetime and normalized cycle time obtained on the robot motion cycle programmed at original task placement. The results presented in the figure suggest one solution point with 8% reduction in cycle time (or improved cycle time performance) on the cost of about 50% reduction in the lifetime (point A1 in the figure 14(a)). Another interesting result disclosed in the figure is solution points in region A2, where about 20% increase in lifetime may be achieved with the same or rather similar cycle time performance.

Figure 15(a) shows the solution space of normalized total motor power consumption as function of cycle time. The normalized total motor power consumption is obtained by actual total motor power consumption at each function evaluation in the optimization loop divided by the total motor power consumption obtained on the robot motion cycle programmed at original task placement. The cross symbol in blue color indicates the coordinate representing normalized total motor power consumption and cycle time obtained on the robot motion cycle programmed at original task placement. The results presented in the figure disclose that the ultimate performance improvement point suggested by point A1 in figure 14(a) results in an increase of about 20% in total motor power consumption (point B1 in the figure 15(a)). Another interesting result disclosed in the figure is solution points in region B2, where about 5% saving of total motor power consumption may be achieved for the solution points presented in region A2 in figure 14(a). In other words, the solution points in region A2 in figure 15(a) suggest not only increase in lifetime but also saving of total motor power consumption.

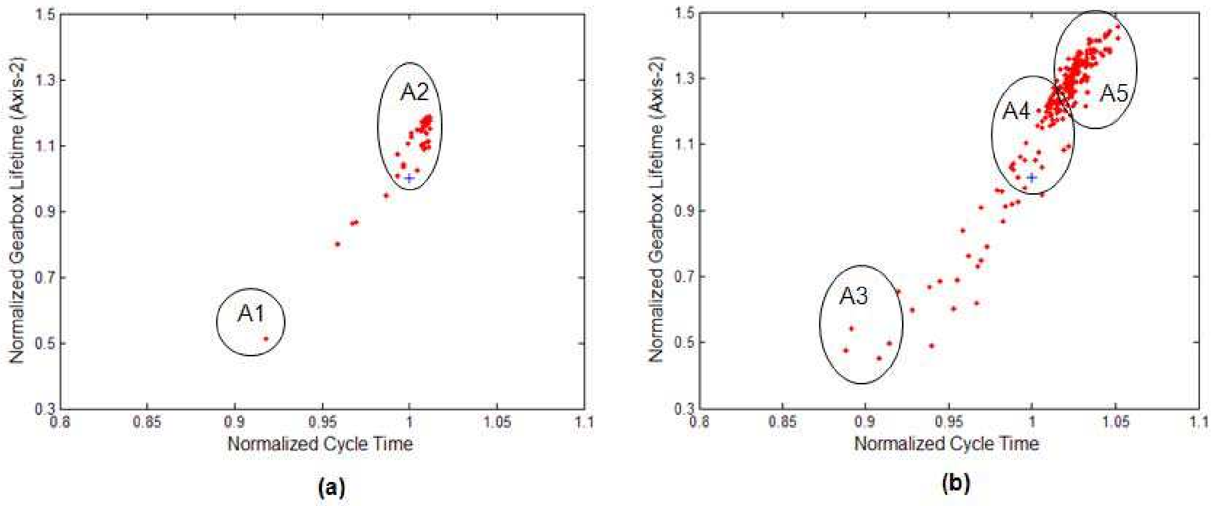


Fig. 14. Solution space of normalized lifetime of gearbox of axis-2 vs. normalized cycle time. (a) for optimal task placement and (b) for combined optimization, (ABB IRB6600-255-175 robot)

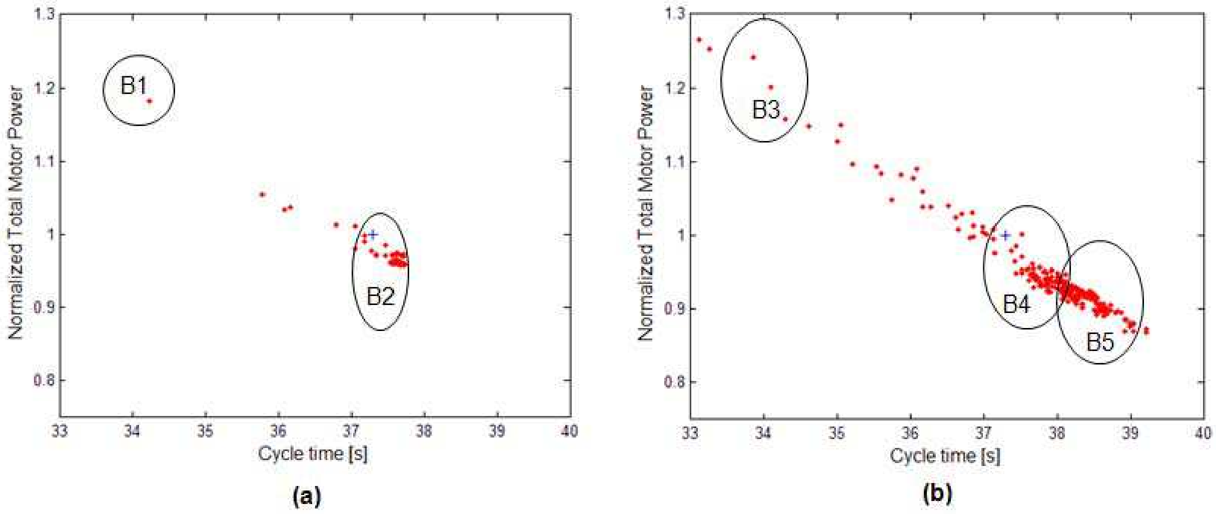


Fig. 15. Solution space of normalized total motor power vs. cycle time. (a) for optimal task placement and (b) for combined optimization, (ABB IRB6600-255-175 robot)

3.4.1.2 Combined task placement and drive-train optimization

The combined optimization involves both path translation for robot task placement and change of robot drive-train parameter setup. Two sets of design variables are used, the first set includes ΔX, ΔY, and ΔZ described in the task placement optimization; the second set includes nine design variables $DV_1, DV_2, \dots DV_9$ which are scaling factors to be multiplied to the original drive-train parameters of the three main axes (axes 1-3). The limits for the path translation are the same as those used in the task placement optimization, i.e., the same as in (10).

The limits for the $DV_1, DV_2, \ldots DV_9$ are

$$DV_i \in (0.9,1.2), where\ i = 1,2 \ldots ,9 \tag{11}$$

The weighting factors w_1 and w_2 are also set to $w_1 = 150$ and $w_2 = 100$ in this combined optimization.

To ease the benchmark work of task placement optimization and the combined optimization, the results of the combined optimization are presented in the same figures as those of task placement optimization. In addition, the figures are carefully prepared at the same scale.

Figure 13(b) shows the convergence curve of the combined optimization. The maximum limit of function evaluations for the optimizer is set to be 225. Optimization is interrupted after the maximum number of function evaluation limit is reached. The total optimization time is about 45 min on the same portable PC used in this work.

Figure 14(b) shows the solution space of normalized lifetime of the same critical gearbox as function of normalized cycle time. The cross symbol in blue color indicates the coordinate representing normalized lifetime and normalized cycle time obtained on the robot motion cycle programmed at original task placement and with original drive-train parameter setup values. The results presented in the figure suggest one solution point with more than 10% reduction in cycle time (or improved cycle time performance) on the cost of about 50% reduction in the lifetime (point A3 in the figure). Another result set disclosed in region A4 in the figure indicates up to 25% increase in lifetime that may be achieved with the same or rather similar cycle time performance. When a cycle time increase of up to 5% is allowed in practice, the lifetime of the critical gearbox may be increased by as much as close to 50% (region A5).

Figure 15(b) shows the solution space of normalized total motor power consumption as function of cycle time. The normalized total motor power consumption is obtained by actual total motor power consumption at each function evaluation in the optimization loop divided by the total motor power consumption obtained on the robot motion cycle programmed at original task placement and with original drive-train parameter setup values. The cross symbol in blue color indicates the coordinate representing normalized total motor power consumption and cycle time obtained on the robot motion cycle programmed at original task placement and with original drive-train parameter setup values. The results presented in the figure disclose that the ultimate performance improvement point suggested by point A3 in figure 14(b) results in an increase of about 20% in total motor power consumption (point B3 in the figure 15(b)). Another interesting result set disclosed in the figure is solution points in region B4, where about 5% saving of total motor power consumption may be achieved for the solution points presented in region A4 in figure 14(b). In other words, the solution points in region A4 in figure 14(b) suggest not only increase in lifetime but also saving of total motor power consumption. When a cycle time increase of up to 5% is allowed, not only the lifetime of the critical gearbox may be increased by as much as close to 50% (region A5) but also the total motor power consumption may be reduced by more than 10%.

3.4.1.3 Comparison between task placement optimization and combined optimization

When comparing the task placement optimization with combined optimization, it is evident that the combined optimization results in much large solution space. This implies in practice that robot cell design engineers would have more flexibility to place the task and setup drive-train parameters in more optimal way. However, the convergence time is also longer, due to the increase in number of design variables introduced in the combined optimization. In addition, changing drive-train parameters in robot cell optimization may pose additional consideration in robot design, so that the adaptation of drive-train in cell optimization would not result in unexpected consequence for a robot manipulator.

3.4.2 Case-II: Optimal robot usage for a typical material handling application

In this case study, an ABB IRB6640-255-180 robot is used. The robot has a payload handling capacity of 180 kg and a reach of 2.55 m. The payload used in the study is 80 kg. The robot motion cycle used is a typical pick-and-place cycle with 400 mm vertical upwards - 2000mm horizontal - 400mm vertical downwards movements – then reverse trajectory to return to the original position. Maximum speed is programmed between any adjacent targets.

3.4.2.1Task placement optimization

Only path translation is employed in the task placement optimization. Three design variables, ΔX, ΔY, and ΔZ are used to manipulate the task position in the same manner as discussed in the Case-I. The limits for the path translation are

$$\begin{array}{c} \Delta X \in (-0.1\,m, 0.1\,m) \\ \Delta Y \in (-0.1\,m\,, 0.1\,m) \\ \Delta Z \in (-0.2\,m, 0.5\,m) \end{array} \tag{12}$$

The weighting factors w_1 and w_2 are set to $w_1 = 100$ and $w_2 = 100$ in this task placement optimization.

The convergence curve of the task placement optimization is shown in Figure 16(a). The optimization is converged after 290 function evaluations. The total optimization time is about 40 min on the same portable PC used in this work.

Figure 17(a) shows the solution space of normalized lifetime of a critical gearbox as function of normalized cycle time. The cross symbol in blue color indicates the coordinate representing normalized lifetime and normalized cycle time obtained on the robot motion cycle programmed at original task placement. The results presented in the figure suggest one set of solution points with close to 6% reduction in cycle time (or improved cycle time performance) with somehow improved lifetime of the critical axis under study (region A6 in the figure). Another interesting result set disclosed in the figure is solution points in region A7, where about 20% increase in lifetime may be achieved with 3-4% improvement of cycle time performance. In engineering practice, 3-4% cycle time improvement can imply rather drastic economic impacts.

Figure 18(a) shows the solution space of normalized total motor power consumption as function of cycle time. The cross symbol in blue color indicates the coordinate representing normalized total motor power consumption and cycle time obtained on the robot motion cycle programmed at original task placement. The results presented in the figure disclose that the solution points with more than 4% cycle time performance improvement (region B6) result in at least 20% increase in total motor power consumption.

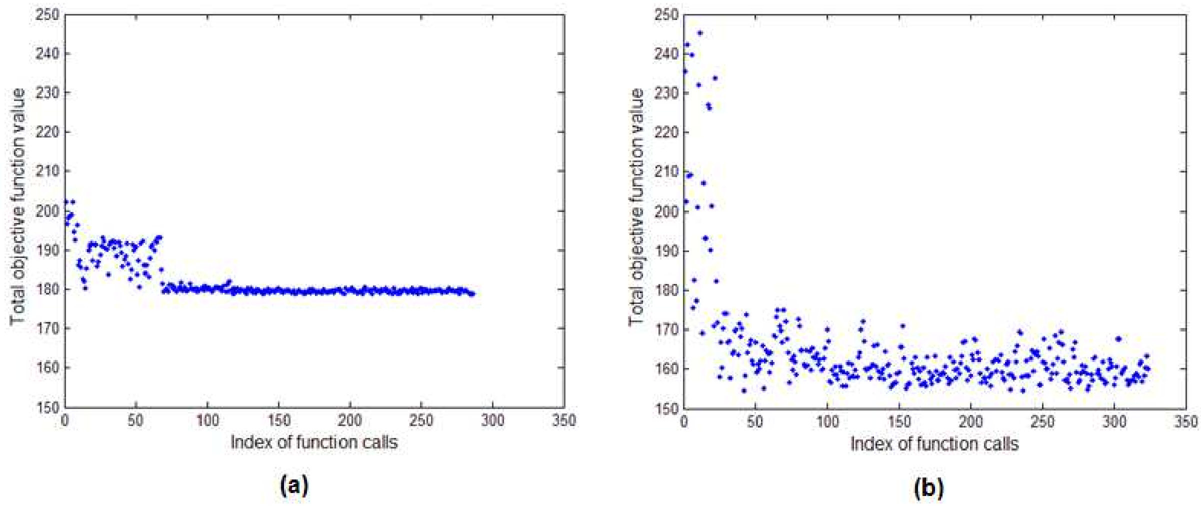


Fig. 16. Convergence curve. (a) for optimal task placement and (b) for combined optimization, (ABB IRB6640-255-180 robot)

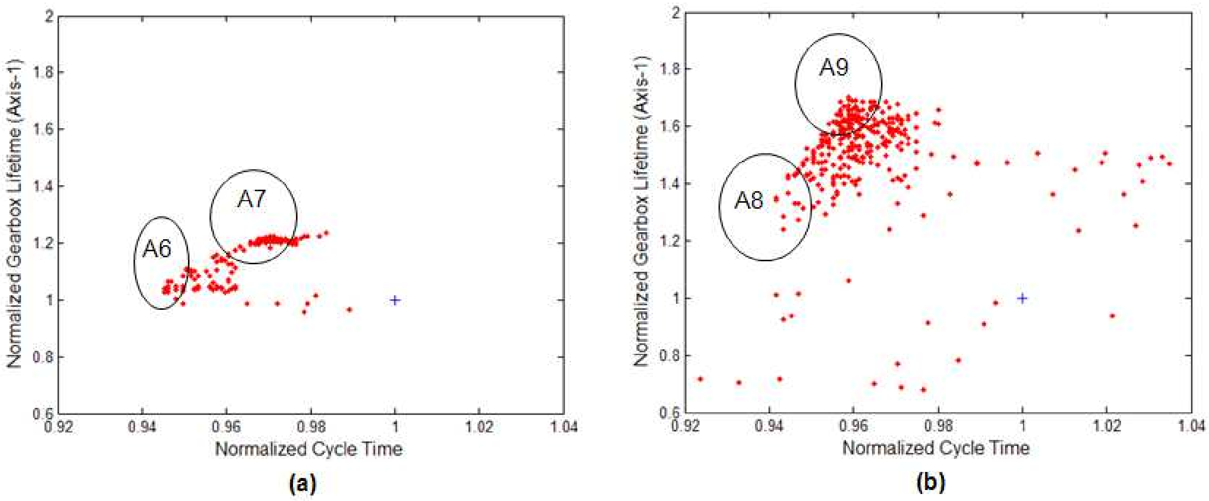


Fig. 17. Solution space of normalized lifetime of gearbox of axis-2 vs. normalized cycle time. (a) for optimal task placement and (b) for combined optimization, (ABB IRB6640-255-180 robot)

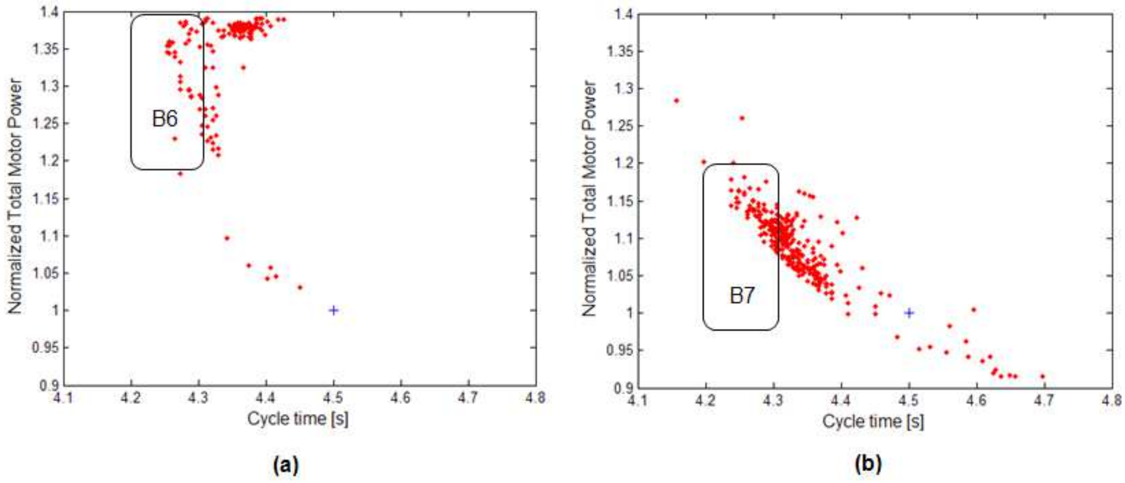


Fig. 18. Solution space of normalized total motor power vs. cycle time. (a) for optimal task placement and (b) for combined optimization, (ABB IRB6640-255-180 robot)

3.4.2.2 Combined task placement and drive-train optimization

As discussed in Case-I, the combined optimization involves both path translation for robot task placement and change of robot drive-train parameter setup. The same two sets of design variables are used. The limits for the path translation are the same as those used in the task placement optimization which are defined by (13).
The limits for the $DV_1, DV_2, \dots DV_9$ are

$$DV_i \in (0.9,1.2), where\ i = 1,2 \dots ,9 \tag{13}$$

The weighting factors w_1 and w_2 are also set to $w_1 = 100$ and $w_2 = 100$ in this combined optimization.
For the same reason, the results of the combined optimization are presented in the same figures as those of task placement optimization. In addition, the figures are carefully prepared at the same scale.
Figure 16(b) shows the convergence curve of the combined optimization. The maximum limit of function evaluations for the optimizer is set to be 325. Optimization is interrupted after the maximum number of function evaluation limit is reached. The total optimization time is about 65 min on the same portable PC used in this work.
Figure 17(b) shows the solution space of normalized lifetime of the same critical gearbox as function of normalized cycle time. The cross symbol in blue color indicates the coordinate representing normalized lifetime and normalized cycle time obtained on the robot motion cycle programmed at original task placement and with original drive-train parameter setup values. The results presented in region A8 in the figure suggest a set of solution points with close to 6% reduction in cycle time but with clearly more than 20% increase in the lifetime. Another result set disclosed in region A9 in the figure indicates more than 60% increase in lifetime and with 3-4% improved cycle time performance!
Figure 18(b) shows the solution space of normalized total motor power consumption as function of cycle time. The cross symbol in blue color indicates the coordinate representing normalized total motor power consumption and cycle time obtained on the robot motion cycle programmed at original task placement and with original drive-train parameter setup

values. The solution points disclosed in region B9 indicate that the solution points with more than 4% cycle time performance improvement result in maximum 20% increase in total motor power consumption.

3.4.2.3 Comparison between task placement optimization and combined optimization

Compared to task placement optimization, it is evident that the combined optimization results in much large solution space. This implies in practice that robot cell design engineers would have more flexibility to place the task and setup drive-train parameters in more optimal way. Even more significantly, the optimization results obtained on this typical pick-and-place cycle reveals more interesting observations. When the same cycle time improvement may be achieved, much more significant lifetime improvement may be achieved by combined optimization and the same is true for the total motor power consumption.

However, the convergence time is longer and optimization has to be interrupted using pre-defined maximum number of function evaluations, due to the increase in number of design variables introduced in the combined optimization. In addition, the same consequence is evident: changing drive-train parameters in robot cell optimization may pose additional consideration in robot design, so that the adaptation of drive-train in cell optimization would not result in unexpected consequence for a robot manipulator.

3.5 Summary of the Results of Section 3

Multi-objective robot task placement optimization shows obvious advantage to understand the trade-off between cycle time performance and lifetime of critical drive-train component.

Sometimes, it may be observed that the cycle time performance and lifetime can be simultaneously improved. When task placement optimization involving only path translation is conducted, reasonable optimization time can be achieved.

The combined optimization of a robot drive-train and robot task placement, in comparison with task placement optimization, has disclosed even more advantages in achieving 1) wider solution space and 2) even more simultaneously improved cycle time performance and lifetime. Benefit of the combined optimization has been evident. Even though the optimization time can be nearly 2-3 times longer than task placement optimization, it can still be justified to be used in engineering practice; namely, earning from longer lifetime of a robot installation is greater than the calculation costs.Furthermore, this suggests that more efforts should be devoted in the future to; 1) better understanding of the multi-objective combined optimization problem and its impact on simulation-based robot cell design optimization; 2) improving efficiency of the optimization algorithms; 3) including collision-free task placement; and finally 4) sophisticated software implementation for engineering usage.

The plots of lifetime of critical component as function of cycle time performance and that of total motor power consumption as function of cycle time performance are also suggested in this work. This graphical representation of the solution space can further ease robot cell design engineers to better understand the trade-off between lifetime of critical drive-train component or total motor power consumption to cycle time performance and therefore choose better design solution that meets their goal.

4. Conclusions and Outlook

4.1 Single Objective Optimization

The results confirm that the problem of path placement in a robot work cell is an important issue in terms of manipulator cycle time. Cycle time greatly depends on the path position relative to the robot manipulator. Up to the 37.2% variation of cycle time has been observed which is remarkably high. In other words, the cycle time is very sensitive to the path placement. Algorithm and tool were developed to determine the optimal robot position by path translation and path rotation approaches. Several case studies were considered to evaluate and verify the developed tool for optimizing the robot position in a robotic work cell. Results disclose that an increase in productivity up to 37.2% can be achieved which is profoundly valuable in industrial robot application. Therefore, using this tool can significantly benefit the companies which have similar manipulators in use.

It is certain that employing this methodology has many important advantages. First, the cycle time reduces significantly and, therefore, the productivity increases. The method is easy to implement and the expense is only simulation cost, i.e., not any extra equipment is needed to be designed or purchased. The solution coverage is considerably broad, meaning that any type of robots and paths can be optimized with the proposed methodology. Another merit of the algorithm is that convergence is not an issue, i.e., reducing the cycle time can be assured. However, a disadvantage is that a global optimum cannot be guaranteed. The importance of the developed methodology is not confined only to the robot end-user application. Robot designers can also take advantage of the proposed methodology by optimizing the robot parameters such as robot structure and drive-train parameters to improve robot performance. As a design application example, the idea of optimum relative position of robot and path can be applied to the design of a tool such as welding device or glue gun which is erected on the mounting flange of the robot. The geometry of the tool can be optimized by studying design parameters to achieve shorter cycle time. Another possibility can be to use the developed methodology for optimal robot placement to realize other optimization objective in robots such as minimizing the torque, energy consumption, and component wear.

One interesting issue that can be investigated is to consider the general problem of finding the optimum by translation and rotation of the path simultaneously. What has been demonstrated in section 2 of the current chapter is to find the optimum path location by either translation or rotation of the path. Obviously, it is also possible to apply both these approaches at the same time. This would probably further shorten the cycle time in comparison to the case when only one approach is used. However, developing an optimal strategy for concurrently applying both approaches is an interesting challenge for future research.

Another important subject to be investigated is to take into account constraints for avoiding collisions. In a real application, a robot is not alone in the work cell as other cell equipments can exist in the workspace of the robot. Hence, in real robot application it is important to avoid collision.

4.2 Multi-Objective Optimization

It is noteworthy that although the methodology is implemented in RobotStudio, the algorithm is general and not dependent on RobotStudio. Therefore, the same methodology and algorithm can be implemented in any other robotic simulation software for achieving time optimality.

Multi-objective robot task placement optimization shows obvious advantage to understand the trade-off between cycle time performance and lifetime of critical drive-train components. The combined optimization of a robot drive-train and robot task placement, in comparison with task placement optimization, discloses even more advantages in achieving wider solution space and even more simultaneously improved cycle time performance and lifetime.

However, weighted-sum approach for formulating the multi-objective function has experienced difficulties in this work, since the weighting factors have been observed to significantly affect the final solution. Hence, an advanced formulation of multi-objective function and algorithms for multi-objective optimization need to be investigThe relativeated.

In combined optimization, the reachability is presumed to be satisfied as the purpose of this work is to rather explore the effect and feasibility of the method. Nevertheless, advanced and practical solutions exist for reachability checking that need to be implemented in the future work. In this study, while the task placement defined in a robot program is manipulated, the relative placements among sub-tasks (representing in practice the relative placements among different robotic stations in a robot cell) are kept unchanged. In the future work, relative placements of sub-tasks in a robot cell can also be optimized using the proposed methodologies.

5. References

Barral, D. & Perrin. J-P. & Dombre, E. & Lie'geois, A. (1999). Development of optimization tools in the context of an industrial robotic CAD software product, *International Journal of Advvanced Manufacturing Technology,* Vol. 15(11), pp. 822–831, doi: 10.1007/ s001700050138

Box, G.E.P. & Hunter, W.G. & Hunter, J.S. (1978). Statistics for experimenters: an introduction to design, data analysis and model building, Wiley, New York

Box, M. J., (1965). A New Method of Constrained Optimization and a Comparison with Other Methods, *Computer Journal*, Vol 8, pp. 42-52

Fardanesh, B. & Rastegar, J. (1988). Minimum cycle time location of a task in the workspace of a robot arm, *Proceeding of the IEEE 23rd Conference on Decision and Control*, pp. 2280–2283

Feng, X. & Sander, S.T. & Ölvander, J. (2007). Cycle-based Robot Drive Train Optimization Utilizing SVD Analysis, *Proceedings of the ASME Design Automation Conference*, Las Vegas, September 4-7, 2007

Haug, E.J. (1992). Intermediate dynamics, Prentice-Hall, Englewood Cliffs, NJ

Kamrani, B. & Berbyuk, V. & Wäppling, D. & Stickelmann, U. & Feng, X. (2009). Optimal Robot Placement Using Response Surface Method, *International Journal of Advvanced Manufacturing Technology*, Vol. 44, pp. 201-210

Khuri, A.I. & Cornell, J.A. (1987). Response surfaces design and analyses, Dekker, New York

Krus, P. & Jansson, A. & Palmberg, J-O. (1992). Optimization Based on Simulation for Design of Fluid Power Systems, *Proceedings of ASME Winter Annual Meeting*, Anaheim, USA

Luenberger, D.G. (1969). Optimization by vector space methods, Wiley, New York

Myers, R.H. & Montgomery, D. (1995). Response surface methodology: process and product optimization using designed experiments, Wiley, New York

Nelson, B. & Donath, M. (1990). Optimizing the location of assembly tasks in a manipulator's workspace, Journal of Robotic Systems, Vol 7(6), pp. 791–811, doi:10.1002/rob.4620070602

Pettersson, M. & Ölvander, J. (2009). Drive Train Optimization for Industrial Robots, *IEEE Transactions on Robotics*, to be published

Pettersson, M. (2008). A PhD Dissertation, Linköping University, Linköping, Sweden

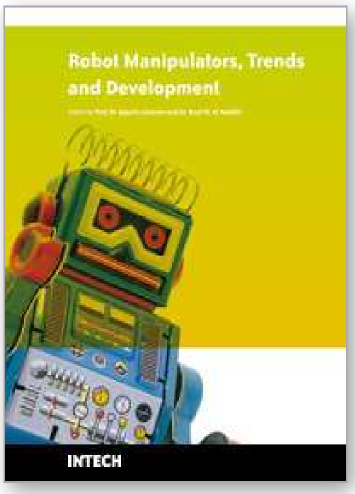
Tsai, L.W. (1999). Robot analysis, Wiley, New York

Tsai, M.J. (1986). Workspace geometric characterization and manipulability of industrial robot. Ph.D. Thesis, Department of Mechanical Engineering, Ohio State University

Vukobratovic, M. (2002). Beginning of robotics as a separate discipline of technical sciences and some fundamental results—a personal view, *Robotica,* Vol. 20(2), pp. 223–235

Yoshikawa, T. (1985). Manipulability and redundancy control of robotic mechanisms, *Proceeding of the IEEE Conference on Robotics and Automation*, pp 1004–1009, St. Louis

Ölvander J. (2001). Multiobjective Optimization in Engineering Design – Applications to Fluid Power Systems, A PhD Dissertation, No. 675 at Linköping University



Robot Manipulators Trends and Development

Edited by Agustin Jimenez and Basil M Al Hadithi

ISBN 978-953-307-073-5
Hard cover, 666 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

This book presents the most recent research advances in robot manipulators. It offers a complete survey to the kinematic and dynamic modelling, simulation, computer vision, software engineering, optimization and design of control algorithms applied for robotic systems. It is devoted for a large scale of applications, such as manufacturing, manipulation, medicine and automation. Several control methods are included such as optimal, adaptive, robust, force, fuzzy and neural network control strategies. The trajectory planning is discussed in details for point-to-point and path motions control. The results in obtained in this book are expected to be of great interest for researchers, engineers, scientists and students, in engineering studies and industrial sectors related to robot modelling, design, control, and application. The book also details theoretical, mathematical and practical requirements for mathematicians and control engineers. It surveys recent techniques in modelling, computer simulation and implementation of advanced and intelligent controllers.

Behnam Kamrani, Viktor Berbyuk, Daniel Wäppling, Xiaolong Feng and Hans Andersson (2010). Optimal Usage of Robot Manipulators, Robot Manipulators Trends and Development, Agustin Jimenez and Basil M Al Hadithi (Ed.), ISBN: 978-953-307-073-5, InTech, Available from: http://www.intechopen.com/books/robot-manipulators-trends-and-development/optimal-usage-of-robot-manipulators

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821