ABSTRACT
Designing a drive train for an industrial robot is a demanding task where a set of design variables need to be determined so that optimal performance is obtained for a wide range of different duty cycles. The paper presents a method where singular value decomposition (SVD) is used to reduce the design variable set. The application is a six degree of freedom serial manipulator, with nine drive train parameters for each axis and the objective is to minimize the cycle time on 122 representative design cycles without decreasing the expected lifetime of the robot. The optimization is based on a simulation model of the robot and conducted on a reduced set of the initial duty cycles and with the design variables suggested by the SVD analysis. The obtained design reduces the cycle time with 1.6% on the original design cycles without decreasing the life time of the robot.

INTRODUCTION
Optimal design of industrial robots has challenged engineers and researchers in both industry and academia for decades. The widely acceptable design approaches consist of general purpose design and task-based or cycle-based design. The general purpose robot design may be referred to as a design approach in which robot is designed based on dynamic performance at discrete points in workspace. The cycle-based design is referred to as a design approach in which robot is designed based on motion cycles defining tasks to be conducted by the robot.

For general purpose robot design, characterization of dynamic performance of a robot is essential. A great deal of efforts has been devoted to characterize the dynamic performance of a manipulator. Yoshikawa [1] and [2] developed the well-known dynamic manipulability ellipsoid. Graettinger and Krogh [3] proposed the acceleration radius. Ma and Angeles [4] presented the concept of dynamic isotropy. Bowling and Khatib [5] extended the acceleration isotropy by including not only isotropic linear end-effector velocity and accelerations but also isotropic angular velocity and accelerations. A commonality among these studies is the recommendation that end effector acceleration capability should be made as isotropic as possible, both locally and over the workspace. This is considered to be of great importance because this results in greater controllability due to the manipulators’ increased ability to response equally well to different controller commands targeted at different points in robots workspace.

The challenge to general purpose robot design, however, is the difficulties for:
- designing for time performance, execution or cycle time cannot be predicted even for a simple point-to-point motion;
- designing for thermal performance, no information on motion characteristics of drive-train components (motors and gearboxes) are available for thermal design; and
- designing for component lifetime, due to the lack of information on motion characteristics of drive-train components.

In contrary to general purpose robot design, cycle-based design may be conducted using “exact” motion information, in particular when the simulated robot controller from a robot provider may be employed.

In many situations, robot manipulators can be designed to perform certain classes of motions within a specified task space, for example, in arc welding, spot welding, machine tending, or material handling applications. The goal of this approach then becomes to determine the optimal kinematics and dynamic design of the robot for robot motions typical in these applications.

Cycle-based design overcomes obviously the drawbacks that the general purpose design experiences, due to the full...
access to the motion information of the robot. In addition, concurrent design optimization of manipulators mechanical components e.g. gearboxes, motors, drives and controller is feasible.

The biggest challenge of cycle-based design however is how to select robot motion cycles that are representative to a target application. Generally, a large number of design cycles are used to ensure the representativeness of pre-defined design cycles. This however results in significantly increased simulation time.

The second challenge of the cycle-based design is to find optimal settings for robot drive-train components consisting of motors and gearboxes, including normally a large set of design variables. The drive-train design is important because it is the major source of robot system performance and manipulator cost. The nature of the drive-train design optimization, involving normally both objective (e.g. cycle time) and constraints (e.g. lifetime of some components or thermal performance) demands large number of design iterations. This is the second source of long simulation time.

This article treats both challenges in cycle-based robot drive-train design optimization, i.e. to select and use a smaller set of representative design cycles and to use SVD analysis to identify and use a smaller set of most significant design variables, which will reduce the number of iterations during the optimization process. Thus the proposed method will reduce the computational time by several orders of magnitude.

The paper is organized as follows. Section two gives a description of the computational framework, including modeling, simulation and optimization. The following section describes the design application and the proposed solution approach. Next the results are presented and lastly the discussion concludes the paper. The object of study is a six axis robot from ABB Robotics, which is a leading supplier of industrial robots.

COMPUTATIONAL FRAMEWORK

In order to be able to efficiently develop such complex mechatronic products as industrial robots a comprehensive computational framework is needed. The contents of the proposed framework are presented in this section. Firstly, to study the dynamic properties of an industrial robot, a dynamic simulation model of the complete robot is needed. Secondly, to facilitate an efficient development process an optimization algorithm is coupled to the simulation environment in order to speed up the search for an optimal design. Finally in order to reduce the design variable space, the concept of Singular Value Decomposition is used to limit the design variables to those which have the greatest influence on the design outcome.

Simulation environment

ABB virtual controller, which is part of the ABB Offline software RobotStudio, is a core component in the robot drive-train design optimization performed in this work. As input data, ABB virtual controller loads a robot configuration file, a cycle file, and an analysis file defining what analysis the virtual controller performs in a simulation. When the simulation is completed, simulation results are exported automatically into a text file. A schematic illustration of the ABB virtual controller simulation is shown in Figure 1.

Design parameters

Electric motors and gearboxes are major components in the drive-train of an industrial robot. The maximum allowed usage of these components is defined by a set of drive-train parameters. In this work, maximum allowed torques and speeds of motors and gearboxes in a robot drive-train were selected as design variables. In total, 9 design variables are used to determine the torques in speed space for both gears and motors of each axis in this study. The details on how these 9 parameters affect the torque-speed relation could not be described due to proprietary reasons. However, the implication of the parameters is to limit the torque for both motors and gearboxes as functions of rotational speed. Parameters $x_{1}$-$x_{9}$ are associated with axis 1, $x_{10}$-$x_{18}$ with axis 2 and so on.

The axes of the robots are numbered from 1 to 6 starting at the base. Thus axis 1 rotates the entire robot, axis 2 bends the lower arm, and so on.

Robot cycles

Tasks conducted by an industrial robot are normally defined by a robot program. Rapid syntax, for example, is a robot programming language for programming an ABB robot. A robot program is referred to as a robot cycle in this work. Robot cycles that are used in cycle-based design are referred to as design cycles. In this work, a total number of 122 design cycles
are included in the analysis. The design cycles are taken from real world spot welding applications.

14 representative design cycles are selected from the above 122 cycles in order to develop and verify methods of SVD analysis and multiple cycle based robot drive-train design optimization. Further on, results obtained using the 14 selected design cycles are verified on the 122 design cycles to confirm the representativeness of the optimization results.

Optimization framework
An integrated framework for the ABB virtual controller simulation and the optimization strategy has been developed in MATLAB®. It is possible to conduct automatic multiple cycle simulation and analysis in this environment.

![Figure 3: Optimization framework.](image)

A typical function evaluation loop in the optimization is detailed below: first a set of design variable values is generated and the same set of design variables in the initial robot configuration file is replaced by the generated new set of variable values. The new configuration file and a set of design cycles are loaded into ABB virtual controller, a simulation is conducted and system characteristics are obtained. Objective and constraint functions are then evaluated based on the simulated system characteristics. The optimization strategy analyzes the objective and constraint function values and determines how the design variables should be changed, which results in a new set of design variable values, and the loop continues. This loop is repeated until a set of optimal design variable values is found, which fulfill the constraints and the objective function value can not be further improved based on some pre-defined optimization criteria.

In this work, torque and speed limits of motors and gearboxes in a robot drive-train were selected as design variables. Cycle-time and lifetime were mainly used as objective functions and constraints.

Optimization algorithms
The optimization carried out in this work is based on simulation results from a comprehensive dynamic simulation model where no analytical derivatives of the objective function are available. The choice is therefore to use a non-gradient based method or to estimate the gradients using finite differences. In this study, the non-gradient Complex method was evaluated against a Quasi-Newton line search technique, namely the function ‘fmincon’ in MATLAB®.

The Complex method was developed by Box [6] in the mid-sixties as an extension to the Neelder-Mead Simplex method [7]. In the Complex method the word complex refers to a geometric shape with $2kn+1$ points in an $n$-dimensional space, where $n$ is the number of optimization variables. These $k$ points are called vertices of the complex and the starting points are generated using random numbers. The main idea of the algorithm is to replace the worst point in the complex by a new point obtained by reflecting the worst point through the centroid of the remaining points by a reflection factor $\alpha > 1$. If the new point is better than the worst, it replaces it and the procedure starts over by reflecting the point that is worst in the new complex. If the new point is still the worst it is moved halfway towards the centroid until it stops repeating as the worst point. The procedure outlined is carried out until the complex has converged or until a pre-described number of evaluations is reached. The implementation used here is a modified version of the Complex method called Complex-RF, as is described in [8].

The gradient based method is a Quasi-Newton line search technique implemented in the built-in MATLAB® function fmincon [9]. Fmincon uses a search direction obtained by estimating first and second order derivatives of the objective function. In each iteration, the method takes a step in the search direction with a step length that yields the minimum value in that direction. However, since the derivatives are obtained using numerical approximations they depend on a smooth objective function. For this application though the objective function is far from smooth. Firstly, it contains numerical noise originating from the simulation model. Secondly, for some parameter there is a sort of dead-band so that the derivative is zero for small parameter changes, e.g. increasing the maximum allowed torque does not yield a lower cycle time unless the torque is larger than an unknown threshold. These issues taken together make it hard to obtain reliable numerical derivatives.

The two methods have been compared on a representative duty cycle with the objective to minimize the cycle time without decreasing the life time on axis 2 which is the critical component. Multiple optimization runs were conducted with different starting points and algorithm parameter settings for each optimization method. When evaluating the two methods, best results were always found by the Complex method, and in average 33% better results were obtained. It was robust in locating a good optimum even though it was more computational expensive. In average the Complex method used three times more evaluations than Fmincon. However, it was concluded that the Complex method were more suitable for this type of problem, and hence it is the method that has been used in the forthcoming optimizations.

Sensitivity analysis and Singular Value Decomposition
An elementary problem for a designer is to find the optimal values of the design variables in order to fulfill the desired specifications and/ or improve the identified system performance. The existing complex trade-offs between numerous design variables on one side and the multiple performance criteria on the other side is a challenging task in the design process. The designer must be able to find effective
design changes in the large design variable space and be aware
of interactions and trade-offs among multiple variables and
criteria. One technique used frequently in this context is
utilizing the so called sensitivity design matrix, see [10].

The sensitivity design matrix in our approach is a \( m \times n \)
matrix. If \( \mathbf{P} = \begin{bmatrix} p_1(x), p_2(x), \ldots, p_m(x) \end{bmatrix} \) is a vector
describing \( m \) different measures on system performance
criteria, and \( \mathbf{x} = \begin{bmatrix} x_1, x_2, \ldots, x_n \end{bmatrix} \) the design vector consisting of
\( n \) independent design variables, then the often rectangular
shaped sensitivity design matrix \( \mathbf{S} \) is defined as:

\[
\mathbf{S} = \begin{bmatrix}
\frac{\partial p_1}{\partial x_1} & \cdots & \frac{\partial p_1}{\partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial p_m}{\partial x_1} & \cdots & \frac{\partial p_m}{\partial x_n}
\end{bmatrix}
\]

Using the sensitivity matrix \( \mathbf{S} \), it is possible to calculate the
impact of small changes of the design variables \( \mathbf{x} \) on the
system characteristic vector \( \mathbf{P} \). Thus considering small
variations \( \Delta \mathbf{x} \) on design variables causing \( \Delta \mathbf{P} \) changes on
system characteristics we obtain: \( \Delta \mathbf{P} = \mathbf{S} \Delta \mathbf{x} \), i.e.

\[
\begin{bmatrix}
\Delta p_1 \\
\vdots \\
\Delta p_m
\end{bmatrix} = \begin{bmatrix}
\frac{\partial p_1}{\partial x_1} & \cdots & \frac{\partial p_1}{\partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial p_m}{\partial x_1} & \cdots & \frac{\partial p_m}{\partial x_n}
\end{bmatrix}
\begin{bmatrix}
\Delta x_1 \\
\vdots \\
\Delta x_n
\end{bmatrix}
\]

Further analysis of this sensitivity matrix enables the
designer to identify the most significant group of variables from
the system performance point of view. Some variables may
have no effect, while others may influence the system
characteristics considerably. The variables with minor impact
on the system characteristics have minor impact in optimization
procedure of the system. The sensitivity matrix is a
linearization in the point of the original design and therefore it
must be recalculated whenever the design has been changed.

In many cases the designer is given a certain specification
requirement/improvement on one or several system
characteristics and is aimed to fulfill the goal, experimenting on
different design variables. This tryout effort can now be
replaced using sensitivity matrix:

\( \Delta \mathbf{x} = \mathbf{S}^{-1} \Delta \mathbf{P} \)

where the inverse of the sensitivity matrix, \( \mathbf{S}^{-1} \) can be replaced
by any kind of pseudoinverse matrix calculation such as the
Moore-Penrose generalized matrix inverse \( \mathbf{S}^* \).

The inverse or the pseudo-inverse matrix produces the
smallest change in design variable space needed to reach the
given change in the system characteristics space. One drawback
of using the inverse of the sensitivity matrix is that all the
elements of \( \Delta \mathbf{P} \) must be given. One possible solution here is
to choose the remaining elements arbitrarily. One consequence
hereby can be an unnecessarily large recommended change in
the design variable space. This can be handled using the
optimization technique i.e. defining constraints in the system
characteristics space and the goal function to be minimized in
the design variable space, i.e. the required changes.

One possible approach would be to minimize the norm of
\( \Delta \mathbf{x} \). Constraints must be set on those elements of \( \Delta \mathbf{P} \), whose
desired change are known.

The considered optimization problem thus can be formulated as

\[
\min \| \Delta \mathbf{x} \|
\]

s.t. \( \Delta p_i(\Delta x) = \beta_i \)

where \( \beta_i \) is the desired change for system characteristic \( p_i \).

In order to decrease the number of design variables and
further identify those group of variables with major impact on
certain system characteristic, Singular Value Decomposition-
SVD- can be used.

SVD is the appropriate tool for analyzing a mapping from
one vector space into another vector space. Most systems of
simultaneous linear equations fall into this category. SVD
offers the possibility of solving the under/over determined
system of linear differential equations, i.e. SVD is a proper tool

Consider the sensitivity matrix \( \mathbf{S} \), then:

\[
\mathbf{S} = \mathbf{U} \sum \mathbf{V}^T
\]

where \( \mathbf{U} \in R^{m \times m}, \mathbf{V} \in R^{n \times n} \) and \( \sum \in R^{m \times n} \).

The column vectors in \( \mathbf{U} \) and \( \mathbf{V} \) are orthonormal vectors.
\( \sum \) is a diagonal matrix with the main diagonal composed of
singular values of the sensitivity matrix \( \mathbf{S} \):

\[
\sum = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_m | 0), \quad \sigma_1 > \sigma_2 > \cdots > \sigma_m.
\]

The physical interpretation of the singular value decomposition of the sensitivity matrix is that the vectors
\( \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\} \) in the design space are mapped to the
the corresponding vectors \( \{\sigma_1 \mathbf{u}_1, \sigma_2 \mathbf{u}_2, \ldots, \sigma_m \mathbf{u}_m\} \) in the system
characteristic space. This relation can be described mathematically by the equation:

\[
\mathbf{U} \sum = \mathbf{SV}
\]

which is equivalent to eq. (5).

Any desired changes in the system characteristics space is feasible through changing the design variables in the subspace
spanned by the vectors \( \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\} \). These orthonormal
directions are called the design mode shapes. All possible
changes in the system characteristics space are made by the
combination of a finite number of design mode shapes, i.e. any
vector \( \Delta \mathbf{x} \) can be described as a linear combination of
\( \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\} \), thus:
\[ \Delta x = \sum_{i=1}^{n} \alpha_i \gamma_i \]  

(8)

All other changes in the design variable space are more or less redundant. The SVD analysis reduces the number of degrees of freedom to the number of non-zero singular values.

By studying the SVD components, conclusions can be drawn about couplings between different design parameters and system characteristics.

Now following the above discussion about optimization technique approach aimed to achieve a desired improvement on certain characteristic parameter with minimum changes on design variables, one can revisit the same problem this time using SVD of the sensitivity matrix in order to first reduce the number of design variables to the number of non-zero singular values.

Introduce a scale factor vector \( \alpha \in R^{m \times 1} \). Consider only the first \( m \) column vectors of \( \Sigma \in R^{n \times m} \) and \( V \in R^{n \times m} \), since the other vectors belong to the null space of the \( S \).

Equations (2), (7) and (8) lead to:

\[
\Delta p = U \sum \alpha 
\]

(9)

and

\[
\Delta x = V \alpha
\]

(10)

We can now formulate an optimization problem, which minimizes the necessary changes of the design variables and at the same time fulfil one or several goals in the system characteristics space

\[
\min \| \Delta x(\alpha) \| 
\]

s.t.

\[
\Delta p_i(\alpha) = \beta_i
\]

where \( \beta_i \) is the equality constraint on system characteristic \( i \).

Thus, given a certain improvement demand on system characteristics, \( \beta_i \), one can calculate the optimal linear combination parameter, \( \alpha \), of design mode shapes resulting to minimum changes on design variables \( \Delta x \), see “Optimization using a reduced variable set” below for an implementation example.

**DESIGN PROBLEM**

The most critical issue of optimal robot drive-train design is the simulation efforts needed due to the following challenges:

- a large number of design variables,
- a large set of design cycles in order to maximize the representativeness of the tasks which the robot is designed for,
- model-based simulation requiring direct search methods for design optimization which are generally computational expensive,
- the problem normally involves multiple objectives and constraints.

The paper investigates how these issues could be addressed, looking at mainly two different aspects. Firstly, could a subset of the original designs cycles be used in the optimization, and secondly, to use a subset of the available design variables.

Following steps have been performed to obtain an optimal drive-train design with maximum efficiency and with acceptable design accuracy.

**Step 1 – Select cycles:** A smaller set of, or 14, design cycles was identified based on a usage measure of the gearbox of axis 2, which is the critical component for the robot of study.

**Step 2 – Optimization:** Conduct design optimization based on an original set of drive-train design variables on the 14 representative design cycles. For this optimization all 9 variables on the three main axes were selected as optimization variables, giving a total of 27 design variables.

**Step 3 – Validation:** The optimization results, or the optimized design variable values, were then verified using all available 122 robot design cycles.

**Step 4 – SVD analysis:** SVD analysis was then conducted identifying 8 design variables as most significant ones, using the selected 14 design cycles.

**Step 5 – Optimization:** Drive-train design optimization was conducted using the 8 design variables identified by SVD analysis applied on the 14 representative design cycles. The same formulation of objective function and constraints as in Step 2 was employed, but with the design variables reduced from original 27 to 8.

**Step 6 – Validation:** The optimized design was verified using all 122 design cycles. This is done to verify the correctness of the optimized 8 design variables on the entire set of 122 design cycles.

Steps 1-3 above answer the question whether it is possible to use a reduced number of cycles in the optimization, and steps 4-6 show if it is possible to reduce the design variable space with a factor of three and still obtain a good design.

**RESULTS**

The results from the design application are described in this section.

**Optimization using all design variables**

First the robot is optimized using 27 design variables (i.e. nine variables for the first three axes) for 14 different cycles. The objective is to minimize the cycle time for the 14 cycles without decreasing the life time on axis 2, which is the critical component. The optimization problem could thus be formulated according to (12).

\[
\min f(x) = \sum_{i=1}^{14} \frac{CT_i(x)}{CT_{i,orig}}
\]

\[
g(x) = LTaxis_{2_i} \geq LTaxis_{2_k,orig}, \quad k = 1, \ldots, 14
\]

\[
x_{x_j}^{lower} \leq x_j \leq x_{x_j}^{upper}, \quad j = 1, \ldots, 27
\]

Where \( i \) is the index for design cycles and \( j \) the index for design variables. \( f(x) \) and \( g(x) \) are the objective and constraint functions, respectively. \( LTaxis_{2_i} \) is the lifetime of
gearbox on axis 2 of cycle \( k \). The constraint function confines that the lifetime of the gearbox on axis 2 is not allowed to decrease compared to the original lifetime for each cycle. The predicted life of the gearboxes is based on a formula supplied by the gearbox manufacturer, see equation (13).

\[
L_o = K \left( \frac{N_o}{N_a} \right)^{10/3} \tag{13}
\]

where,
\( L_o \) is predicted lifetime in hours
\( K \) is a converting factor
\( N_o \) is rated output speed
\( T_o \) is rated output torque
\( N_a \) is average output speed
\( T_a \) is average output torque

The optimization yields a decrease in the average cycle time of 1.9% and an increase of the expected lifetime on axis two, see Figure 3. Table 1 below shows the variable limits and the obtained optimal values for the parameters associate with axis 2 expressed as percentage of the original value. The optimization algorithm required 5000 function evaluations to converge, which took 68 hours on a 3 GHz PC.

Table 1: Optimal parameter settings and variable limits for axis 2.

<table>
<thead>
<tr>
<th>Parameter axis 2</th>
<th>Lower bound</th>
<th>Upper bound</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>- 10.0 %</td>
<td>+ 29.0 %</td>
<td>+ 25.0 %</td>
</tr>
<tr>
<td>11</td>
<td>- 10.0 %</td>
<td>+ 29.0 %</td>
<td>- 9.4 %</td>
</tr>
<tr>
<td>12</td>
<td>- 48.0 %</td>
<td>+ 50.0 %</td>
<td>- 27.7 %</td>
</tr>
<tr>
<td>13</td>
<td>- 10.0 %</td>
<td>+ 450.0 %</td>
<td>+ 163.2 %</td>
</tr>
<tr>
<td>14</td>
<td>- 10.0 %</td>
<td>+ 0 %</td>
<td>- 8.0 %</td>
</tr>
<tr>
<td>15</td>
<td>- 10.0 %</td>
<td>+ 30.8 %</td>
<td>0.0 %</td>
</tr>
<tr>
<td>16</td>
<td>- 10.0 %</td>
<td>+ 30.8 %</td>
<td>+ 25.8 %</td>
</tr>
<tr>
<td>17</td>
<td>- 10.0 %</td>
<td>+ 30.8 %</td>
<td>- 8.1 %</td>
</tr>
<tr>
<td>18</td>
<td>- 10.0 %</td>
<td>+ 30.8 %</td>
<td>+ 18.2 %</td>
</tr>
</tbody>
</table>

In the optimal design the load, in terms of the allowed torque on axis 2, has been decreased, whereas the allowed torque on axes 1 and 3 have been increased. Therefore the cycle time could be lowered without a deterioration of the lifetime on axis 2. The lifetime on axes 1 and 3 are however decreased, but they are still above the critical limit.

The obtained optimized design is now validated by evaluating the cycle-time and lifetime on the 108 design cycles that were excluded in the optimization. In total 118 cycles shows a reduction in cycle-time, most of them between 1-2 % but one more than 5%, see Figure 4.

Figure 3. Changes in lifetime (blue, increase) and cycle-time (red, decrease).

Figure 4. Distribution of cycle-time changes for the 122 cycles.

The lifetime for all 122 cycles was also evaluated. It could be seen in Figure 5 that cycles with low lifetime obtained an increased lifetime whereas cycles with high lifetime were decreased.
Optimization using a reduced variable set

The sensitivity analysis (see above), was performed including 54 design variables and 5 system characteristics. The chosen system characteristics were cycle time and the estimated lifetime of the gearboxes for different axis. In order to increase reliability of the calculated sensitivity matrix, 14 selected design cycles were used. Thus the final sensitivity matrix (eq. 2) is a result of a weighted sum of the fluctuations on 5 system characteristics and 54 design variables (9/ axis), subjected to small modifications. In practice, the sensitivity for each design variable, \( \frac{\partial p_i}{\partial x_j} \), is obtained by calculating \( p_i \) (using 14 different cycles) for a set of small changes of \( x_j \) keeping all other \( x \) constant. The sensitivity matrix thus represents a linearization around the current design point, which nevertheless captures the main effects of changes in the design variables.

Using equations (5) - (10), the optimal combinations, \( \alpha \), of the design mode shapes were calculated. The constraint was to reduce the cycle time with a certain amount. The corresponding changes of design variables \( \Delta x \) were calculated, see eq (9).

\[ \min f(x) = \sum_{i=1}^{14} \frac{CT_i(x)}{CT_i,\text{orig}} \]

\[ g(x) = LT \text{ axis } 2_j \geq LT \text{ axis } 2_k, \text{ orig } \]

\[ x_{j,\text{lower}} \leq x_j \leq x_{j,\text{upper}} \]

Using the reduced variable set the optimization yielded a decrease in cycle time of 0.9%, still with an increased lifetime on axis 2, see Figure 7. The optimization with the reduced parameter set converged in 1575 evaluations which is a reduction with a factor of 3 compared to the optimization with the full variable set.

The obtained optimal design was again validated by running all 122 design cycles. As can be seen in Figure 8 the cycle time was reduced by between 0-2% for a vast majority of the cycles.
CONCLUSIONS

The optimization with the reduced design variable set converged in 1575 evaluations which is a reduction with a factor of 3 compared to the optimization with 27 variables. It should however be noted that the convergence in variable space were 0.001% for the problem with 8 variables compared to 0.06% for the problem with 27 variables. Thus it could be concluded that the convergence of the Complex method scales linearly with the number of optimization variables. This implies that if the variable space is reduced with a factor of three, so is also the computational time.

Singular value decomposition was employed to determine a subset of design variables to use in the optimization in order to speed up convergence. With this approach the computational burden is reduced to one third. However, the results obtained with the reduced variable set are not as good as with the full variable set. Nevertheless, it is still a significant improvement. One reason for not obtaining better results is that the original design, which was the origin for the sensitivity analysis, is rather different than the optimal. This implies that the variables that had the greatest influence on the performance of the original design are not necessarily the same as for the optimized design.

When conducting cycle based design of industrial robots there is always the discussion of which cycles to consider. Should the robot be designed for one particular cycle or should a large number of different cycles be used when evaluating a design. Conceivably some cycles might be considered more important than others. In this paper we have shown that it is possible to select a fraction of the total set of cycles and use them in the optimization and still get good performance considering the whole set of cycles. In this approach the cycle time for each cycle were normalized by dividing it with the original cycle time. Thereby each cycle is given an equal weighting in the optimization. Naturally a different weighting of the cycles is possible.

ACKNOWLEDGEMENTS

Hans Andersson at ABB Robotics is acknowledged for providing with design cycles and helping with setting up ABB simulated controller for this work. Daniel Wäpling, Jakob Wessström, and Dr. Torgny Brogårth at ABB Robotics are acknowledged for a large number of technical discussions. Henrik Johansson and Gustav Torn are acknowledged for their help in conducting the calculations. Johan Ölvander would like to acknowledge financial support from VINNOVA, Swedish Governmental Agency for Innovation Systems.

REFERENCES