Energy Efficient and Collision Free Motion of Industrial Robots using Optimal Control

Staffan Björkenstam[1], Daniel Gleeson[1], Robert Bohlin[1], Johan S. Carlson[1] and Bengt Lennartson[2]

Abstract— **In a production plant for complex assembled products there could be up to several hundred of robots used for handling and joining operations. Thus, improvement in robot motions can have a huge impact on equipment utilization and energy consumption. These are two of the most important aspects of sustainability in a production system. Therefore, this paper presents an algorithm for generating efficient and collision free motion of industrial robots using path planning and direct transcription methods for numerical optimal control. As a measure of efficiency for moving between configurations we use a combination of the energy norm of the applied actuator torques and the cycle time. Velocity and torque limits are handled and modeled as hard constraints. However, more general problems can be solved by the same approach.**

Our novel algorithm solves the problem in three steps; (i) first a path planning algorithm calculates an initial collision free path, (ii) a convex optimal control problem is then formulated to follow this path, and finally (iii) a nonlinear optimal control problem is solved to iteratively improve the trajectory. The resulting trajectory is guaranteed to be collision free by restrictions in the configuration space based on a local sensitivity analysis. The algorithm has been successfully applied to several industrial cases demonstrating that the proposed method can be used effectively in practical applications.

I. INTRODUCTION

Sustainable manufacturing is defined as “the creation of manufactured products that use processes that minimize negative environmental impacts, conserve energy and natural resources, are safe for employees, communities, and consumers and are economically sound” by The U.S. Department of Commerce 2010 [1]. This implies the need to consider and balance between economical, ecological and social factors to achieve a sustainable production system. A more detailed discussion on measures and awareness used at Chalmers University of Technology can be found in Johansson, Dagman, Rex, *et al.* [2]. By using Computer Aided Engineering (CAE), physical prototypes can be replaced by simulation, new products can be introduced faster, the efficiency of the production system can be optimized using mathematical methods and algorithms, and it can be done by simulation experts and production engineers in a safe and healthy environment.

The automotive industry is an example of an equipment and energy intensive manufacturing, where up to 28% of the vehicle life cycle energy is spent during production. For example, a typical automotive car body consists of about 300 sheet metal parts, joined by about 4000 welds. Typical joining methods are spot welding, arc welding, gluing and stud welding. In car body assembly plants, the welds are distributed to several hundred industrial welding robots, which are organized in up to 100 stations. The body shop is indeed investment intense, with the robots as the main consumer of energy (Meike and Ribickis [3]). In Almström, Andersson, Muhammad, *et al.* [4] it is highlighted how utilization affects different aspects of sustainable production, the link between utilization and productivity, as well as practical considerations when improving utilization in manufacturing industry. Therefore from a sustainability perspective it is highly motivated to develop new software methods and algorithms for further improvement of equipment utilization and energy efficiency of robotized manufacturing systems.

In Segeborn, Segerdahl, Carlson, *et al.* [5] it is shown that the balancing of weld work load between the executing stations and robots has a significant influence on achievable production rate and equipment utilization. Robot line balancing is a complex problem, where a number of welding robots in a number of stations are available to execute an overall weld load. Each weld is to be assigned to a specific station and robot, such that the line cycle time is minimized. Line balancing efficiency depends on station load balancing, robot welding sequencing, path planning and effectiveness of robot coordination for collision free execution within each other’s working envelopes. Robot coordination impairs cycle time by inserting waiting positions and signals into the original paths. At Volvo Cars it has been proven that by using automatic path planning and line balancing instead of standard off-line programming the cycle time in welding lines can be improved by as much as 25%. The next step for improving the automatic path planning and line balancing is to include detailed optimization of motion profiles between welds. This choice is also from an energy efficiency aspect motivated. Meike and Ribickis [3] investigated different strategies to operate robots in an energy efficient way. Motion profile optimization was one of the strategies pointed out among others such as automatic shut-down and start-up, reusing braking energy, and brake management.

Direct transcription methods have been applied to solve minimum time and minimum energy problems for industrial robots before, for example by Stryk and Schlemmer [6]. In this paper we present a novel method for collision free and energy-efficient control of an industrial robot. Our main contribution is the way we incorporate the collision avoidance into the optimal control problem. This is done by approximating the geometry in configuration space rather than $\mathbb{R}^3$ making the size of the resulting optimization

[1] Geometry and Motion Planning, Fraunhofer-Chalmers Centre, Chalmers Science Park, SE-412 88 Göteborg, Sweden

[2] Automation Research Group, Department of Signals and Systems, Chalmers University of Technology, SE-412 96 Göteborg, Sweden

978-1-4799-1515-6/13/$31.00 ©2013 IEEE

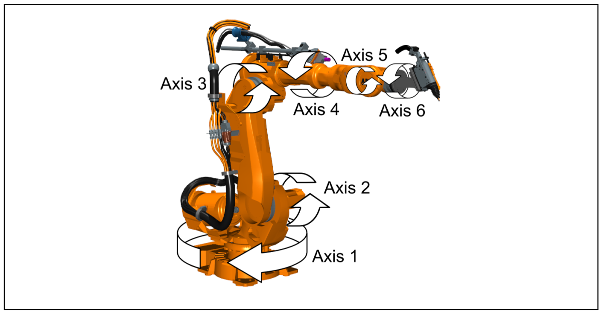


Fig. 1. Degrees of freedom of the industrial robot ABB IRB 6640.

problem independent of the complexity of the geometry which contrasts existing methods, for example Müller [7] and El Khoury, Lamiraux, and Taix [8].

Our algorithm solves the problem in three steps; (i) first a path planning algorithm calculates an initial collision free path, (ii) a convex optimal control problem is then formulated to follow this path, and finally (iii) a nonlinear optimal control problem is solved to iteratively improve the trajectory. The resulting trajectory is guaranteed to be collision free by restrictions in the configuration space based on a local sensitivity analysis.

The method described in this paper can also be used for creating motions for simulation of manual assemblies considering ergonomics and feasibility of humans, and is therefore also relevant for the social aspects of sustainability. An early version of the method can be found in Gleeson [9] where it is applied in the framework of Bohlin, Delfs, Hanson, *et al.* [10].

II. METHOD

In this section we describe the steps of the method we use to generate energy efficient and collision free robot motions. The method is generic, but for clarity we restrict our attention to a typical industrial robot with six degrees of freedom, where all joints are purely one-dimensional revolute joints. Fig. 1 shows such a robot, an ABB IRB 6640 robot, where all six rotational axes are marked out.

In sections II-A through II-C we give some background on path planning, velocity tuning and numerical optimal control. In section II-D and II-E we formulate our problem as an optimal control problem. Finally, in section II-F, we show how to incorporate collision avoidance into the optimal control problem, which is our main contribution.

A. Collision free path planning

Automatic path planning addresses the problem of finding collision free motions of moving objects. It is a well established research area and there are a number of distinguished research groups in the world. Complete algorithms are of little industrial relevance due to the complexity of the problem (PSPACE-hard for polyhedral models, see Canny [11]). Instead, sampling based techniques trading completeness for speed and simplicity are the methods of choice. Common for these methods are the needs for efficient collision detection, nearest neighbor searching, graph searching and graph representation. Two popular methods are the Probabilistic Roadmap Method (PRM) and Rapidly-Exploring Random Trees (RRT). The PRM samples randomly among all configurations of an object, keeps the collision-free samples, and then connects nodes pairwise if the straight line path between them is collision-free, see Bohlin and Kavraki [12]. The RRT incrementally builds two trees from the start and the goal configurations respectively. In each step an attractor is generated at random and the trees are expanded from their nearest node towards the attractor. The iteration stops when the trees overlap, as in LaValle and Kuffner [13].

Inspired by both these probabilistic methods, Fraunhofer-Chalmers Centre (FCC) has developed a deterministic path planner that adaptively adjusts a grid in the configuration space. The goal is to produce short, collision free, piecewise linear paths that serve as good initial paths for the following steps.

B. Velocity tuning

The initial velocity profile on the collision free path can be added, in a straight forward way, by considering each segment separately. One way of doing this is to formulate the optimal control problem directly on the line segments. We use, however, a special formulation of the optimal control problem defined in the following subsections. In order to find an initial iterate we simply add constraints to the optimal control problem that force the solution to stay on the line segments. In order to allow discontinuous velocities at the vertices we either add knots at each vertex in the path or solve for each segment separately with zero velocity at the start and end of each segment. It can been shown that for a convex objective function with limits on torque and velocities this can be formulated as a convex problem, as is done by Verscheure, Demeulenaere, Swevers, *et al.* [14], i.e. if we find an optimal solution it will also be a global optimum.

C. Numerical optimal control

Here, in order to motivate our approach, we discuss algorithms for numerical solution of nonlinear optimal control problems.

We would like to use the sufficient conditions for an optimal control i.e. solve the Hamilton-Jacobi-Bellman (HJB) equations. Unfortunately analytic solution of the HJB equations for nonlinear problems is notoriously hard and solving a discretized problem using a dynamic programming approach suffers from the curse of dimensionality which renders it infeasible to systems of even moderate number of degrees of freedom.

The alternative is to use Pontryagin's maximum principle (PMP) to find a locally optimal solution. Numerical methods to find solutions to PMP are usually divided into indirect and direct methods. Indirect methods use PMP to derive the continuous optimality condition which are then discretized and solved numerically. Indirect methods have the nice feature that they by construction are consistent with the PMP, but suffer from more practical problems, for example how to start the iteration, convergence and handling of more general constraints. An example based on the indirect approach is given in Gregory, Olivares, and Staffetti [15].

In direct methods, the continuous-time optimal control problem is discretized, and the resulting discretized nonlinear optimization problem can be solved using standard nonlinear programming algorithms. If care is taken in the discretization, the lagrange multipliers in the nonlinear programming problem converges to the costate and the Karush-Kuhn-Tucker (KKT) conditions of the nonlinear programming problem (NLP) are really a well behaved discretization of PMP, see Benson [16] for details.

With the significant progress in large-scale computational algorithms and sparse nonlinear programming, direct methods have become increasingly popular. One such method that has gained popularity is the pseudospectral method, see Elnagar, Kazemi, and Razzaghi [17].

In pseudospectral methods the continuous functions are approximated using polynomials which collocate the differential equations and the constraints at a certain set of points. Different methods use different sets of points, two common choices are the Gauss-Legendre points and the Gauss-Lobatto points. It can be shown that the Nth order polynomial interpolation at these nodes converges to $f(t) \in C^m$ under L_2 norm at the rate of N^{-m}. If $f(t)$ is in C^∞, the polynomial interpolation converges at a spectral rate, i.e. faster than any given polynomial rate. This is indeed very impressive but in practice and in particular when solving time-optimal problems non-smooth solutions and even discontinuity is not uncommon. We can handle this by inserting knots where the solution is allowed to be non-smooth or discontinuous, following the procedure of Ross and Fahroo [18], but the number of knots needed and where to insert them is not usually known a priori.

If the solution is non-smooth using high order pseudospectral metods seems wasteful since the derivative matrix for the pseudospectral method is dense which makes the resulting NLP less sparse. Further, if the solution is not only non-smooth but also discontinuous the pseudospectral solution can exhibit quite large ripples and overshooting which could lead to large constraint violations in between collocation points. In order to overcome these problems we use a piecewise polynomial approximation of relatively low order.

If the system under consideration is a Hamiltonian system exciting results have been shown from combining discrete mechanics with direct methods for optimal control (DMOC). In discrete mechanics the variational principle is discretized directly which leads to integrators with very nice properties. A comparison between the implicit midpoint rule and a second order DMOC-method applied to an optimal control problem can be found in Junge, Marsden, and Ober-Blöbaum [19].

D. The continuous optimal control problem

The problem we want to solve can be formulated as an optimal control problem where we want to minimize the cost functional

$$J = \Phi(x(t_s), t_s, x(t_f), t_f) + \int_{t_s}^{t_f} L(x(t), u(t), t)dt \quad \text{(1a)}$$

while satisfying

$$\dot{x}(t) = f(x(t), u(t), t) \quad \text{(1b)}$$

$$g(x(t), u(t), t) \geq 0 \quad \text{(1c)}$$

$$H(x(t_s), t_s, x(t_f), t_f) = 0 \quad \text{(1d)}$$

for $t \in [t_s, t_f]$.

Here the state vector is $x(t) = [q(t)^T,\ \dot{q}(t)^T]^T \in \mathbb{R}^{2n}$ and the control signal $u(t) \in \mathbb{R}^n$ is the vector of actuator torques applied at the joints. Where $q(t)$ belongs to the configuration space i.e. in our robot case q is the vector of joint angles.

The cost functional to be minimized in (1a) contains two terms, the function $\Phi(x(t_s), t_s, x(t_f), t_f)$ which accounts for costs associated with the initial and terminal state, and the integral of the Lagrangian $L(x, u, t)$ which describes costs incurred along the trajectory. In our problems $\Phi(x(t_s), t_s, x(t_f), t_f)$ is typically just the duration, $t_f - t_s$, and the Lagrangian is a measure of the power consumption, modeled by a quadratic function $L(x, u, t) = x^T Qx + u^T Ru$ where Q and R are symmetric positive semi definite matrices.

The differential equations (1b) are called the state equations and describe the dynamics of the system. Constraints on the state and control along the trajectory are included in (1c) while (1d) contains the boundary conditions. Note that (1c) can include both equality and inequality constraints.

In our case we can write the state equations (1b) as

$$\dot{x}(t) = \frac{d}{dt} \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} \dot{q}(t) \\ f_{FD}(q(t), \dot{q}(t), u(t)) \end{bmatrix}$$

where $f_{FD}(q(t), \dot{q}(t), u(t))$ is the forward dynamics of the robot. If we use the generalized acceleration $\ddot{q}(t)$ we can rewrite (1b) as

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} \dot{q}(t) \\ \ddot{q}(t) \end{bmatrix} \\ u(t) = f_{ID}(q(t), \dot{q}(t), \ddot{q}(t)) \end{cases}$$

where

$$\begin{aligned} f_{ID}(q(t), \dot{q}(t), \ddot{q}(t)) = M(q(t))\ddot{q}(t) + \\ + C(q(t), \dot{q}(t)) + G(q(t), \dot{q}(t)) \end{aligned}$$

is the inverse dynamics in which $M(q(t))$ is the system's mass matrix, $C(q(t), \dot{q}(t))$ include the centrifugal and coriolis forces and, $G(q(t), \dot{q}(t))$ are the external forces including gravity. The inverse dynamics can be calculated efficiently in a recursive fashion as described in Featherstone [20].

E. The discrete optimal control problem

For our problems we found that a reasonable trade-off between efficiency and error is to use a discrete approximation of order two. Betts [21] suggests using the trapezoidal rule which is equivalent to the piecewise pseudospectral method of order two using the Gauss-Lobatto points, but this discretization sometimes caused oscillations for our problems. Instead we use the corresponding approximation with the Gauss-Legendre points which amounts to the implicit midpoint method, a symplectic integrator of order two (Hairer, Lubich, and Wanner [22]).

If we divide the time interval $[t_s, t_f]$ into N equidistant subintervals of duration h, we can formulate a discrete version of (1), using the midpoint rule, as follows:

Minimize

$$\Phi(x_0, t_0, x_{N+1}, t_{N+1}) + \\ + \sum_{i=0}^{N} hL\left(\frac{x_i + x_{i+1}}{2}, u_{i+\frac{1}{2}}, \frac{t_i + t_{i+1}}{2}\right) \tag{2a}$$

subject to

$$x_{i+1} - x_i = hf\left(\frac{x_i + x_{i+1}}{2}, u_{i+\frac{1}{2}}, \frac{t_i + t_{i+1}}{2}\right) \tag{2b}$$

$$g\left(\frac{x_i + x_{i+1}}{2}, u_{i+\frac{1}{2}}, \frac{t_i + t_{i+1}}{2}\right) \geq 0 \tag{2c}$$

$$H(x_0, t_0, x_{N+1}, t_{N+1}) = 0 \tag{2d}$$

$$t_{i+1} = t_i + h \tag{2e}$$

$$h \geq 0 \tag{2f}$$

for $i = 0 \ldots N$.

If we include $\ddot{q}_{i+\frac{1}{2}}$, for $i = 0 \ldots N$, as variables we can write (2b) as

$$\begin{cases} q_{i+1} - q_i = h\frac{\dot{q}_{i+1} + \dot{q}_i}{2} \\ \dot{q}_{i+1} - \dot{q}_i = h\ddot{q}_{i+\frac{1}{2}} \\ u_{i+\frac{1}{2}} = f_{ID}\left(\frac{q_i + q_{i+1}}{2}, \frac{\dot{q}_i + \dot{q}_{i+1}}{2}, \ddot{q}_{i+\frac{1}{2}}\right) \end{cases}$$

for $i = 0 \ldots N$.

In the discrete setting our limits on the angles, velocities and torques become

$$\begin{aligned} q_{lower} \leq \quad & q_i \quad \leq q_{upper}, \quad i = 0 \ldots N+1 \\ \dot{q}_{lower} \leq \quad & \dot{q}_i \quad \leq \dot{q}_{upper}, \quad i = 0 \ldots N+1 \\ u_{lower} \leq \; & u_{i+\frac{1}{2}} \leq u_{upper}, \quad i = 0 \ldots N \end{aligned}$$

Note that we choose to collocate the angle and velocity limits on the node points and not at the midpoints.

The variables in the NLP are q_i, $\dot{q}_i$, t_i for $i = 0 \ldots N+1$ and $\ddot{q}_{i+\frac{1}{2}}$, $u_{i+\frac{1}{2}}$ for $i = 0 \ldots N$ i.e. the positions and velocities are defined on the node points while the accelerations and the torques are defined on the midpoints. Note that since we use equidistant time steps it suffices to use t_0 and t_{N+1} as time variables. In our implementation we use t_0, t_{N+1} and h.

To solve the resulting non-linear optimization problem we use the NLP-solver IPOPT (Interior Point OPTimizer) which is described in Wächter and Biegler [23].

F. Collision avoidance

The initial path created in Section II-A is collision free. In this section we describe how this property is maintained while the optimal control problem is solved.

Let us define the distance function as

$$\varphi(q) = \min_{p \in \mathcal{A}(q)} d(p)$$

where $\mathcal{A}(q)$ is the space occupied by the robot at configuration q and $d(p) : \mathbb{R}^3 \to \mathbb{R}$ is the minimum distance to the surrounding geometry $\Gamma \subset \mathbb{R}^3$, i.e.

$$d(p) = \min_{y \in \Gamma} \|p - y\|_2$$

One way of keeping the solution collision free would be to add a minimum clearance constraint, i.e include

$$\varphi(q(t)) \geq d_c, \ \forall t \in [t_s, t_f] \tag{3}$$

in (1c), where $d_c \in \mathbb{R}$ is the minimum allowed clearance along the path. But $\varphi(q)$ is not generally in C^1 so adding the constraint (3) to the optimal control problem would result in a nonlinear programming problem which can not be solved using standard NLP-solvers. To overcome this we formulate an iterative method based on local sensitivity analysis.

Given a feasible configuration $q^k \in \mathbb{R}^n$ we want to be able to guarantee that a new configuration $q^{k+1} = q^k + \Delta q^k$ has at least a clearance of $d_c \geq 0$.

Let p be a point on the robot and $p(q)$ be that point at configuration q. Since $d(p)$ is Lipschitz continuous

$$\max_p \left\|p(q^{k+1}) - p(q^k)\right\|_2 \leq \varphi(q^k) - d_c$$

implies that $\varphi(q^{k+1}) \geq d_c$.

If we do a Taylor expansion around q^k we get

$$\max_p \left\|p(q^{k+1}) - p(q^k)\right\|_2 \approx \max_p \left\|\frac{\partial p}{\partial q}(q^k)\Delta q^k\right\|_2 \leq \\ \max_p \left\|\frac{\partial p}{\partial q}(q^k)\Delta q^k\right\|_1 \leq \left\|\Delta q^k\right\|_1^{w(q^k)} \leq n\left\|\Delta q^k\right\|_\infty^{w(q^k)}$$

where $\|\cdot\|^{w(q)}$ denotes a weighted norm with weight vector $w_i(q) = \max_p \|\frac{\partial p}{\partial q_i}(q)\|_1$ for $i = 1 \ldots n$. Hence it is reasonable to use

$$n\left\|\Delta q^k\right\|_\infty^{w(q^k)} \leq \varphi(q^k) - d_c$$

in order to approximately satisfy

$$\varphi(q^{k+1}) \geq d_c.$$

In the discrete setting the approximate minimum clearance constraints, for iteration k, can be written as

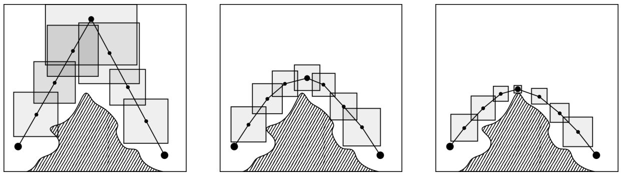
$$n\left\|q_i^{k+1} - q_i^k\right\|_\infty^{w(q_i^k)} \leq \varphi(q_i^k) - d_c \text{ for } i = 0 \ldots N+1$$

which is a simple box constraint in configuration space at each time step, as illustrated in Fig. 2.

Since we use an approximate clearance constraint we can not guarantee that the new configuration $q^{k+1} = q^k + \Delta q^k$ is above our clearance threshold. Hence we need to check the new iterate and perform a backtracking step if necessary. We stop the iteration when the reduction in the objective value is below some given threshold.

Starting with the velocity tuned solution, Fig. 2 illustrates how the trajectory is iteratively improved using the sensitivities.

With two degrees of freedom it is possible to plot the state space of the solution. Fig. 3 compares the velocity tuned solution to the optimized solution for a problem where the robot is restricted to only use the second and third joint.

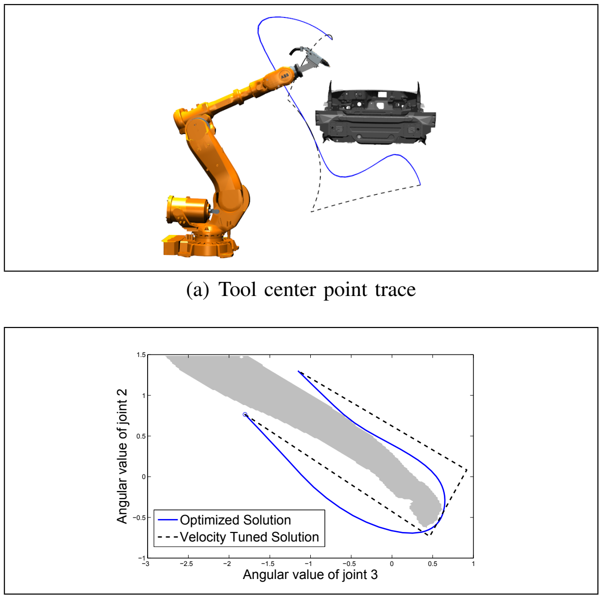


(a) Stage I

(b) Stage II

(c) Stage III

Fig. 2. Three different stages in the iterative procedure to improve the collision free path.



(a) Tool center point trace

(b) Configuration space plot

Fig. 3. Comparison between the velocity tuned (cycle travel time T = 6.21s) and the optimized trajectory (T = 4.82s) for a two-dimensional problem.

III. RESULTS

To validate our method we run the described algorithm on one of the robots in the virtual stud welding station of an actual industrial assembly line, see Fig. 4. The limits on angles and velocities are taken from the robot specification. These can be seen in Table I along with the estimated torque limits. Here we use the simple objective function

$$J = \int_{t_s}^{t_f} 1 + c_e u(t)^T u(t) dt$$

i.e. a linear combination of the time duration and our energy measure, the sum of squares of the actuator torques integrated over time. The trajectory for the robot after sequencing consists of visiting 22 studs, starting and finishing in a home position. Since we can not affect the time spent welding or the placement of the studs we are only concerned with the energy consumption and the travel time in between the welds. While welding the robot needs to stand still, this enables us to split the problem into, in this case 23, independent smaller problems using appropriate boundary conditions.

To assess the trade-off possible between cycle time and energy consumption we plot the Pareto optimal front for the optimized trajectory by varying the weight c_e, see Fig. 5. For example if we can accept a 10% increase in travel time we can reduce our power consumption with about 30%.

TABLE I

LIMITS OF POSITION, VELOCITY AND TORQUE

Joint	1	2	3	4	5	6
q_{lower} [deg]	-170	-65	-180	-300	-120	-360
q_{upper} [deg/s]	170	85	70	300	120	360
$\dot{q}_{lower}$ [deg]	-110	-90	-90	-190	-140	-235
$\dot{q}_{upper}$ [deg/s]	110	90	90	190	140	235
u_{lower} [Nm]	-15500	-21000	-7000	-130	-120	-0.8
u_{upper} [Nm]	15500	21000	7000	130	120	0.8

TABLE II

TEST CASE PERFORMANCE ($c_e = 10^{-8}$)

Method	Cycle travel time	Energy consumption
Velocity tuned	11.2 s	100%
Optimized	9.05 s	87.2%

In Table II we compare our new optimized solution and the original velocity-tuned solution path from our path planner. For this particular case, $c_e = 10^{-8}$, we see that, by having the freedom to alter the path, we are able to cut down the cycle travel time by 19.2% while at the same time decreasing the energy consumption by 12.8%. These solutions are also indicated in the Pareto plot, Fig. 5.

Even though we do not have an accurate model of the real power consumption, the example indicates that considerable savings can be made using our method.

In our calculations we chose $N = 50$ for each phase which gave us a total of 1750 discretization points. For the optimized trajectory in Table II the step size h vary between 1.25 and 14.6 ms. The problem was solved in about 20 minutes on a standard desktop computer.

IV. CONCLUSIONS

In this paper we have presented a new method to generate trajectories for industrial robots. Our method combines numerical optimal control with path planning to generate trajectories that are both energy efficient and collision free. In order to validate our approach the method has been tested on actual industrial cases. In order to use our control algorithm on real world equipment a feedback tracking-controller could be created by using for example LQR trajectory stabilization.

There are however still areas of improvement in our method. To accurately model our system a more realistic

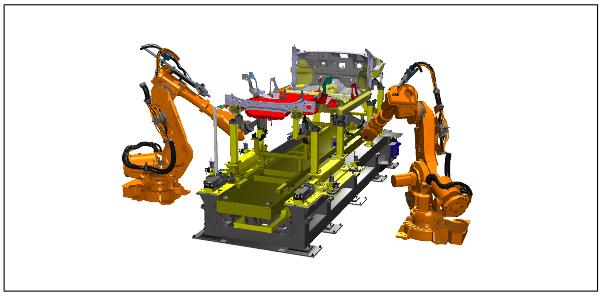


Fig. 4. Industrial stud welding station with two ABB IRB 6640 robots.

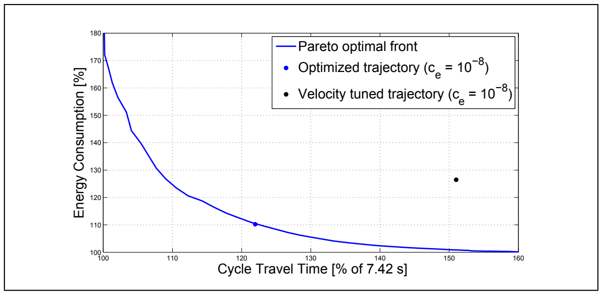


Fig. 5. Pareto optimal front for the optimized trajectory of the test case shown in Fig. 4. The velocity tuned and optimized solutions for $c_e = 10^{-8}$ are indicated by dots.

model of the power consumption is needed. Also a rigorous convergence analysis would be in order. It would also be interesting to compare the performance of our algorithm if the NLP-solver IPOPT was replaced with a solver based on for example Sequential Quadratic Programming (SQP). SQP-methods are easier to warm start compared to an interior point method, which could both save computer time and enhance stability of our iterative approach.

The work in this paper could also be extended to include coordination of multiple robots by combining our approach with the ideas from Vergnano, Thorstensson, Lennartson, *et al.* [24] and Wigström, Lennartson, Vergnano, *et al.* [25].

ACKNOWLEDGMENT

This work was carried out at the Wingquist Laboratory VINN Excellence Centre, and is part of the Sustainable Production Initiative and the Production Area of Advance at Chalmers University of Technology. It was supported by the Swedish Governmental Agency for Innovation Systems.

REFERENCES

[1] U. S. Department of Commerce, *How does commerce define sustainable manufacturing?* 2010. [Online]. Available: http://www.trade.gov/competitiveness/sustainablemanufacturing/how_doc_defines_SM.asp.

[2] B. Johansson, A. Dagman, E. Rex, T. Nyström, M. Knutson Wedel, J. Stahre, and R. Söderberg, "Sustainable production research: awareness, measures and development," *International Journal of Sustainable Development*, vol. 4, no. 11, pp. 95–104, Sep. 2012.

[3] D. Meike and L. Ribickis, "Analysis of the energy efficient usage methods of medium and high payload industrial robots in the automobile industry," 10th International Symposium on Topical Problems in the Field of Electrical and Power Engineering, Pärnu, Estonia, January 10-15, 2011.

[4] P. Almström, C. Andersson, A. Muhammad, and M. Winroth, "Achieving sustainable production through increased utilization of production resources," The 4th Swedish Production Symposium, Lund, May 3-5, 2011.

[5] J. Segeborn, D. Segerdahl, J. S. Carlson, F. Ekstedt, A. Carlsson, and R. Söderberg, "A generalized method for weld load balancing in multi station sheet metal assembly lines," Proceedings of the ASME 2011 International Mechanical Engineering Congress & Exposition, Denver, Colorado, USA, November 11-17, 2011.

[6] O. von Stryk and M. Schlemmer, "Optimal control of the industrial robot manutec r3," *Computational optimal control, International series of Numerical Mathematics*, vol. 115, pp. 367–382, 1994.

[7] A. Müller, "Energy optimal control of serial manipulators avoiding collisions," in *Mechatronics, 2004. ICM'04. Proceedings of the IEEE International Conference on*, IEEE, 2004, pp. 299–304.

[8] A. El Khoury, F. Lamiraux, and M. Taix. (2012). Optimal motion planning for humanoid robots, [Online]. Available: hal.archives-ouvertes.fr.

[9] D. Gleeson, "Virtual manikin controller - calculating the movement of a human model," Master's thesis, Chalmers University of Technology, 2012.

[10] R. Bohlin, N. Delfs, L. Hanson, and J. S. Carlson, "Automatic creation of virtual manikin motions maximizing comfort in manual assembly processes," in *4th CIRP Conference on Assembly Technology and Systems - CATS 2012*, May 2012, pp. 209–212.

[11] J. Canny, *The Complexity of Robot Motion Planning*. MIT Press, 1988.

[12] R. Bohlin and L. E. Kavraki, "Path planning using lazy prm," in *In IEEE International Conference on Robotics and Automation*, 2000, pp. 521–528.

[13] S. LaValle and J. Kuffner J.J., "Randomized kinodynamic planning," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 1, pp. 473–479.

[14] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: a convex optimization approach," *Automatic Control, IEEE Transactions on*, vol. 54, no. 10, pp. 2318 –2327, Oct. 2009.

[15] J. Gregory, A. Olivares, and E. Staffetti, "Energy-optimal trajectory planning for robot manipulators with holonomic constraints," *Systems Control Letters*, vol. 61, no. 2, pp. 279 –291, 2012, ISSN: 0167-6911.

[16] D. Benson, "A gauss pseudospectral transcription for optimal control," PhD thesis, Massachusetts Institute of Technology, 2005.

[17] G. Elnagar, M. A. Kazemi, and M. Razzaghi, "The pseudospectral legendre method for discretizing optimal control problems," *Automatic Control, IEEE Transactions on*, vol. 40, no. 10, pp. 1793–1796, 1995.

[18] I. M. Ross and F. Fahroo, "Pseudospectral knotting methods for solving optimal control problems," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 3, pp. 397–405, 2004.

[19] O. Junge, J. E. Marsden, and S. Ober-Blöbaum, "Discrete mechanics and optimal control," in *IFAC Congress, Praha*, 2005.

[20] R. Featherstone, *Rigid body dynamics algorithms*. Springer Berlin: 2008, vol. 49.

[21] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*, 2nd. Society for Industrial and Applied Mathematics, 2010.

[22] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*. Springer, 2006.

[23] A. Wächter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[24] A. Vergnano, C. Thorstensson, B. Lennartson, P. Falkman, M. Pellicciari, F. Leali, and S. Biller, "Modeling and optimization of energy consumption in cooperative multi-robot systems," *Automation Science and Engineering, IEEE Transactions on*, vol. 9, no. 2, pp. 423–428, Apr. 2013.

[25] O. Wigström, B. Lennartson, A. Vergnano, and C. Breitholtz, "High-level scheduling of energy optimal trajectories," *Automation Science and Engineering, IEEE Transactions on*, vol. 10, no. 1, pp. 57–64, Jan. 2013.