Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Interactive Multiobjective Optimization for a Grab-Shift Unloader Crane

Johan Sjöberg[⋆,†], **Simon Lindkvist**[⋆], **Jonas Linder**[†], **Jonas Öhr**[‡],

[⋆] *ABB Corporate Research, Sweden*
[†] *Division of Automatic Control, Linköpings University, Sweden*
[‡] *ABB Crane Systems, Sweden*

Abstract: In this paper, multiobjective optimization (MOO) is applied to an optimal control problem for a grab-shift unloader crane. The crane is modeled as a cart-pendulum system with varying rope length and the trajectory of the grab is limited by the ship, the quay, and the crane structure. The objectives to minimize are chosen as time, energy and maximal instantaneous power. The optimal control problem is solved using a direct simultaneous optimal control method. The study shows that MOO can be an efficient tool when choosing a good compromise between conflicting objectives such as time and energy. Furthermore, navigation among the Pareto optimal solutions has proven to be very useful when a user wants to learn how the control variables interact with the process.

Keywords: Optimal Control, Crane control, multiobjective optimization

1. INTRODUCTION

Grab-shift unloader cranes are used to move bulk material from a ship to a hopper at shore. This type of cranes are today normally operated by an operator controlling the motion of the trolley and the grab so that fast and efficient trajectories are obtained. In order to improve the operation even more, the idea in this paper, is to determine the trajectories using optimal control instead. The optimal trajectories and control signals can then be used in many ways to improve the operation, for instance, to teach the operators or to run the crane autonomously. The formulation and solution of optimal control problems for cranes have been studied in several earlier references, see for instance, Al-Garni et al. (1995); Auernig and Troger (1987); Hu and Teo (2004).

It is important that the cost function reflects the desired behavior of the crane since the achieved trajectory and control signals are chosen to make the cost function as good as possible. Often the desired behavior is a compromise between different objectives such as speed, energy efficiency, control utilization etc. The objectives are commonly conflicting which means that depending on how the different objectives are prioritized, different trajectories and control signals will be optimal. Here optimal means that there is no way to improve one objective without deteriorating another (Miettinen, 1999). A solution that satisfies this property is denoted Pareto optimal and the set of all Pareto optimal points is the Pareto set. The image of the Pareto set in the objective space is denoted the Pareto frontier. An optimal control problem with many objectives is denoted a multiobjective optimal control problem.

The multiobjective optimal control problem is in this paper reformulated as a nonlinear program and the result is a multiobjective optimization problem (MOO) which can be written as

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \ \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})\} \\ &\text{subject to } \mathbf{x} \in X \end{aligned} \tag{1}$$

There are many algorithms to "solve" a MOO. One class is scalarization methods and another is vector optimization methods, see Miettinen (1999). The first class combines the objectives to form scalar objective functions that are solved as single-objective problems to yield one point in the Pareto set at a time. The second class treats the objectives independently and solves the MOO as a vector-valued optimization problem where many points in the Pareto set are obtained at once. In this work, the scalarization approach has been used.

In addition to how the Pareto optimal points are computed, another choice is at which time the decision maker (DM), *i.e.*, the person who decides which solution is "best", makes the decision. In this paper, an interactive method has been chosen where the DM is able to iteratively choose between different Pareto optimal solutions. In this way, the DM can control the search for a final solution depending on how the objective values and design variables vary in the Pareto set. The process of choosing the preferred solution is also often a good way to learn about the optimization problem and the plant. For large-scale problems, such as the optimal control problem for cranes, it can take substantial amount of time to find a single Pareto optimal solution using the scalarization method and an interactive process can then be slow and tedious for the DM. In recent research two-phase methods have been introduced. In these methods, the Pareto frontier is first sparsely sampled and the DM is then able to continuously "navigate" on an approximation of the frontier in real-time, see Eskelinen et al. (2010); Hartikainen et al. (2011); Monz et al. (2008). However, the approaches in these papers either require a convex Pareto frontier to yield good approximations (which is not always the case for industrial processes) or the computation of the approximation can be tedious. In this paper, an approach introduced by Linder et al. (2012) is used instead. By sampling the Pareto frontier in a specific manner and decomposing the set of sampled points into convex sets, it is possible to compute an approximate Pareto frontier fast even for non-convex Pareto frontiers, see Linder et al. (2012) for details.

There are also other papers that have studied MOO applied to optimal control problems for cranes, see for

instance, Deb and Gupta (2004), Logist et al. (2010), and Sakawa and Shindo (1982).

The remainder of this paper is organized as follows: In Section 2 the model of the grab-shift unloader crane is presented. Section 3 shows how the MOO problem is stated from the optimal control problem. Section 4 introduces the developed MOO framework with a short description of how it can be used to investigate the Pareto frontier. In Section 5 the framework is applied to the crane optimal control problem. Finally, some conclusions are presented in Section 6.

2. MODELING OF A HARBOR CRANE

2.1 The Trolley and the Grab

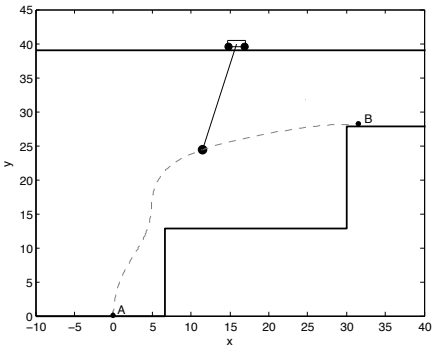


Fig. 1. The crane system including the trolley, the grab and the ship, quay and hopper profile.

In principle, the crane can be described as a cart-pendulum system, see Figure 1. The trolley and the hoist (the drum that controls the rope length) are driven by electric motors and it is assumed that there are inner feedback loops so that the optimal control formulation can use the trolley and the hoist accelerations as control inputs. With these control inputs the model can be written as

$$\dot{p}_t = v_t \tag{2a}$$
$$\dot{v}_t = a_t \tag{2b}$$
$$\dot{l}_r = v_r \tag{2c}$$
$$\dot{v}_r = a_r \tag{2d}$$
$$\dot{\theta} = \omega \tag{2e}$$
$$\dot{\omega} = \frac{1}{l_r}\Big(- 2v_r\omega - \cos(\theta)a_t - g\sin(\theta)\Big) \tag{2f}$$

where p_t is the trolley position, v_t is the trolley speed, l_r is the rope length, v_r is the hoist speed (the first derivative of the rope length), θ is the pendulum angle, ω is the pendulum angular velocity, a_t is the trolley acceleration, and a_r is the hoist acceleration (the second derivative of the rope length).

The grab position can be expressed as

$$x_{pl} = p_t + l_r\sin(\theta) \tag{3a}$$
$$y_{pl} = h - l_r\cos(\theta) \tag{3b}$$

where y_{pl} and x_{pl} are the grab position vertically and horizontally, respectively, and h is the height of the crane. The forces on the trolley and in the rope are given by

$$\frac{F_t}{m_l} = \frac{m_l + m_t}{m_l}a_t + l_r\cos(\theta)\dot{\omega} + \sin(\theta)a_r - l_r\omega^2\sin(\theta) + 2v_r\cos(\theta)\omega \tag{4a}$$
$$\frac{F_r}{m_l} = -\sin(\theta)a_t - a_r + l_r\omega^2 + g\cos(\theta) \tag{4b}$$

where F_t is the force acting on the trolley, F_r is the force acting in the rope, m_l is the mass of the load and m_t is the mass of the trolley. Based on (4), the power required by the trolley and hoist motors can be written as

$$\frac{P_t}{m_l} = \frac{F_t}{m_l}v_t, \quad \frac{P_l}{m_l} = \frac{F_r}{m_l}v_r \tag{5}$$

The model described by (2) – (5) has 14 variables. However, in order to improve convergence and speed of the optimizations extra variables and equations are introduced to split complicated expressions into parts. This is denoted lifting and is inspired by Albersmeyer and Diehl (2010). Because of the lifting, the dynamical model used in the optimization has 22 variables. Throughout the paper, these 22 variables except the control signals (a_t and a_r) are concatenated to a vector denoted $x(t)$ while the control signals are concatenated to a vector denoted $u(t)$.

2.2 Obstacles and Limitations

The motion and the control inputs are also subject to limitations. The states and the controls are constrained by simple bound constraints

$$\begin{aligned} -10 &\le p_t \le 50 \\ -4.33 &\le v_t \le 4.33 \\ 0 &\le l_r \le 60 \\ -2.33 &\le v_r \le 3.17 \\ -\frac{\pi}{2} &\le \theta \le \frac{\pi}{2} \\ -5 &\le \omega \le 5 \\ -1.5 &\le a_t \le 1 \\ -1 &\le a_r \le 2 \end{aligned}$$

The grab must also avoid obstacles such as the crane structure, the ship and the quay. The height profile for these obstacles is described the black solid line in Figure 2.

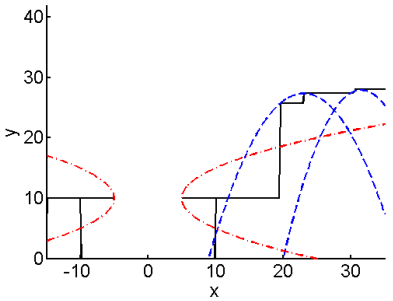


Fig. 2. The height profile of the crane, quay and ship (black solid), and the quadratic approximations horizontally (blue dashed) and vertically (red dash-dotted).

In order to obtain a smooth NLP, it is desired that the constraints are differentiable. Therefore, a smooth approximation of the obstacles parameterized in the grab position is derived. The limitation of this approach is of course that it could be hard to find good smooth approximations of rectangular obstacles. However, the approximation need not be accurate for all positions but

only where the optimal trajectory often interfere with the obstacle, typically the corners. Therefore, quadratic functions have been chosen to represent the obstacles. For the quay and the crane it is appropriate to parametrize the quadratic functions in the horizontal grab position as

$$y_{pl} > a + bx_{pl} + cx_{pl}^2, \quad \forall x_{pl} \tag{6}$$

where the coefficients a, b and c are computed by choosing three points so that the quadratic functions covers the corners. The constraints in (6) are shown as blue dashed lines in Figure 2.

For cycles where the grab travels outside the ship, the constraints in (6) are enough. However, in this report, we have also studied cases when the grab starts and ends inside the ship. For such cases, the grab must avoid the ship latches as well and therefore additional constraints are necessary. For convenience, these approximations are also expressed as quadratic functions but parametrized in the vertical grab position instead,

$$x_{pl} > a + by_{pl} + cy_{pl}^2, \quad \forall y_{pl} \tag{7}$$

The resulting constraints are shown as the red dash-dotted line in Figure 2.

3. MULTIOBJECTIVE OPTIMAL CONTROL

3.1 Background to Optimal Control

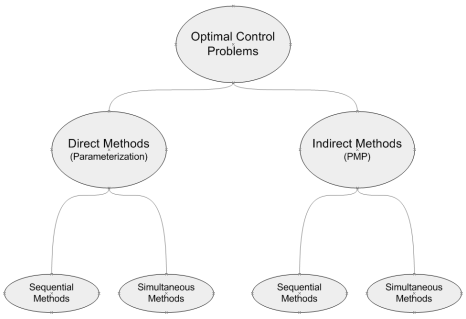


Fig. 3. Strategies for solving optimal control problems.

A multiobjective optimal control can be cast as follows

$$\begin{aligned} &\min_{u(t),\,t_f} \{J_1, \ldots, J_m\} \\ &\text{s.t.} \\ &\qquad F(\dot{x}(t), x(t), u(t)) = 0 \\ &\qquad\qquad c\big(x(t), u(t)\big) \le 0 \\ &\qquad\qquad\qquad x(0) = x_0 \\ &\qquad\qquad\qquad \psi\big(x(t_f)\big) = 0 \end{aligned} \tag{8}$$

where

$$J_i = \int_0^{t_f} L_i\big(x(t), u(t)\big)\, dt, \;\; i = 1, \ldots, m$$

The function $F(\cdot)$ is a strangeness-free DAE model describing the dynamical model including algebraic relationships, $c(\cdot,\cdot)$ are additional constraints, $\psi(\cdot)$ are terminal constraints, J_i are the objectives, $L_i(\cdot,\cdot)$ are the cost functions, and t_f is the final time.

There are many methods to solve optimal control problems but basically there are two different categories of methods namely direct or indirect, see Figure 3. Direct methods means that the control signals are parameterized using a finite number of parameters. The result is a finite dimensional optimization problem that include dynamical equations. The dynamical equations can be solved in either of two ways. Sequential methods, also known as shooting methods, means that a standard optimization solver changes the parameters and for given parameters the dynamical equations are solved using an integrator routine. Simultaneous methods, or collocation based methods, on the other hand discretize the dynamical equations as well and solves the whole problem at once as a large but sparse optimization problem. Indirect methods are based on the Pontryagin Minimum Principle (or variational calculus). It means that the necessary conditions for optimality of the optimal control problem, which is a two-point boundary problem, are solved either using a sequential or a simultaneous method. The advantages and disadvantages of the different methods are described thoroughly in Betts (2001). In this paper, the direct simultaneous method is chosen. One of the major benefits with this method is that quite general constraints can be dealt with in a straight forward manner.

3.2 Discretization to NLP

For simultaneous methods, discretization is the step when the optimal control problem (8) is reformulated as a nonlinear program (NLP). There are a vast number of different discretization methods and depending on which is chosen, the sparsity pattern for the NLP will be different. A thorough discussion on this topic can be found in Betts (2001) and Kunkel and Mehrmann (2006). In this paper a first order backward-difference formula (BDF) has been used. The main motivation for choosing a first order BDF instead of a higher order is that the first order BDF yields the most sparse NLP and the accuracy of the solution is sufficient (up to third order was evaluated for comparison). An approximation of the solution to the dynamical system (8) will then, using the first order BDF, be given by

$$\begin{aligned} &c_d^k(x_{k+1}, x_k, u_{k+1}) = \\ &\qquad F\big(\frac{x_{k+1} - x_k}{T_s}, x_{k+1}, u_{k+1}\big) = 0, \; k = 0, \ldots, N-1 \end{aligned}$$

where $T_s = \frac{t_f}{N}$ and N is the number of discretization points. The equality constraints c_d are usually denoted defect constraints.

The objectives need to be discretized as well and the discretized expressions become the NLP objectives. In this paper, the objectives are discretized using the standard Riemann sum which gives

$$\int_0^{t_f} L(x, u)\, dt = \frac{t_f}{N} \sum_{k=1}^{N-1} L(x_k, u_k)$$

For a general multiobjective optimal control problem, the MOO NLP generated will be

$$\begin{aligned} &\min_{x_k, u_k, t_f} \Big\{ \frac{t_f}{N} \sum_{k=1}^{N-1} L_1(x_k, u_k), \ldots, \frac{t_f}{N} \sum_{k=1}^{N-1} L_m(x_k, u_k) \Big\} \\ &\text{s.t. } c_d(x_{k+1}, x_k, u_{k+1}, t_f) = 0, \; k = 0, \ldots, N-1 \\ &\qquad\qquad c(x_k, u_k) \le 0, \; k = 0, \ldots, N \\ &\qquad\qquad\qquad x(0) = x_0 \\ &\qquad\qquad\qquad \psi\big(x_N\big) = 0 \end{aligned} \tag{9}$$

3.3 Objectives

In this paper three important objectives have been chosen for analysis. The objectives are:

- f_1: Minimize the time of one cycle, *i.e.*, the time of moving the endpoint of the pendulum from point A to point B.
- f_2: Minimize the total energy consumed in the move.
- f_3: Minimize the maximum instantaneous power used by the system during the move.

The motivation to use the two first objectives is quite clear. However, the motivation to use the third objective might not be so obvious. The idea is that when many cranes operate simultaneously the power consumed by the harbor will be very large if all the cranes are allowed to use large instantaneous powers. The harbor will then draw large currents from the power grid which is something the harbor owner is charged for and not only the energy used by the harbor. Hence, it is interesting to investigate if there is a good compromise between cycle time, energy consumption and maximum instantaneous power.

The mathematical formulation of the objective functions are

$$z_1 = f_1(\mathbf{x}) = t_f \tag{10a}$$

$$z_2 = f_2(\mathbf{x}) = \sum_{i=1}^{N} (P_{t,i}^+ + P_{l,i}^+) \tag{10b}$$

$$z_3 = f_3(\mathbf{x}) = \max(P_{t,1}^+ + P_{l,1}^+, \ldots, P_{t,N}^+ + P_{l,N}^+) \tag{10c}$$

where t_f is the total time of one cycle, P_t and P_l are the trolley and hoist motor powers given by (5), respectively, and $(\cdot)^+ = \max(0, \cdot)$. N is the number of time steps and $\mathbf{x}$ is the set of all decision variables. Hence, the energy and maximal instantaneous power objectives are chosen to only consider power consumed to accelerate the trolley or the grab and not power fed back while braking. The positive parts are reformulated by splitting the variable into two positive parts as

$$P = P^+ - P^-, \quad P^+ \geq 0,\ P^- \geq 0$$

The reformulation requires P^+ and P^- to be pulled toward zero since otherwise many combinations of P^+ and P^- can fulfill $P = P^+ - P^-$ for given P. The pull is achieved by adding a regularization term to the objectives in (10) as

$$\bar{f}_k(\mathbf{x}) = f_k(\mathbf{x}) + \delta \sum_{i=1}^{N} \left(P_{t,i}^+ + P_{t,i}^- + P_{l,i}^+ + P_{l,i}^-\right)$$

where $k = 1, \ldots, 3$ and δ is small number ($\approx 10^{-3}$).

4. FRAMEWORK FOR INTERACTIVE MOO

In order to compute the Pareto frontier and navigate on it, a framework developed in Linder et al. (2012) has been used. Basically, it can be divided into two phases.

In the first phase, the Pareto frontier is sampled with an automatic method which is executed without the interaction of the DM. Since it can take several minutes to find even one solution, it is advantageous that the DM can run this phase unattended. In the second phase, the sampled Pareto set is loaded into a GUI that the DM uses to navigate continuously on an approximated Pareto set.

4.1 Phase I: Offline phase

The offline phase samples the complete Pareto frontier evenly. Figure 4 shows an example of the sampling in $\mathbb{R}^2$. This phase can be divided into three steps; pre-processing, sampling and post-processing.

First, the pre-processing step creates a direction of search d and a reference point set z^{R_x}. The points in the reference point set is used as starting points for the optimization of every sample, see Figure 4. The reference points are created equidistantly on the hyperplane spanned by the individual minima of the MOO problem (z^{1*} and z^{2*} in Figure 4) and the direction is chosen as the normal to the reference hyperplane. The individual minima are found by dividing the MOO problem into a set of singleobjective optimization (SOO) problems, and solving every objective individually.

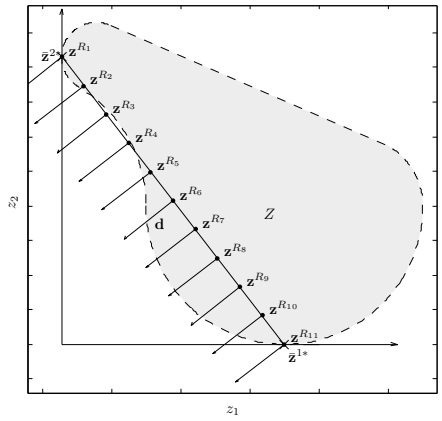


Fig. 4. An example of a reference point set (z^{R_x}) with the direction d in a two dimensional MOO problem. The reference points are created equidistantly between the individual minima z^{1*} and z^{2*}. Note that it is not necessary that the reference hyperplane is "above" the complete Pareto frontier.

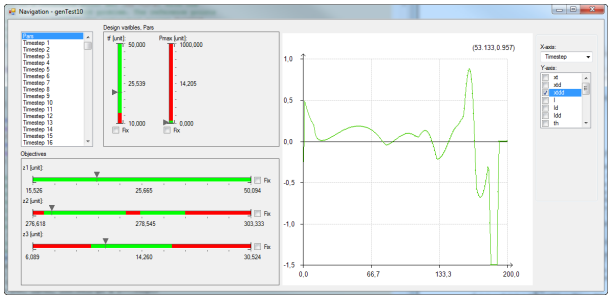
Second, the sampling step uses the reference point set together with the scalarization method described in Pascoletti and Serafini (1984) to solve the MOO problem. For every reference point a SOO problem is solved using the software package IPOPT (Wächter and Biegler, 2006). This step do the actual optimizations to find the Pareto frontier and depending of the number of samples and the size of the MOO problem, this step can take hours to complete.

Third, the post-processing step applies a Pareto filter that removes all non Pareto optimal solutions. To handle non-convex and disconnected Pareto frontiers in the online phase, the Pareto set is decomposed into simplices using the reference points and Delaunay triangulation. This makes it possible to create a linear approximation between the samples, even though the Pareto set is non-convex.

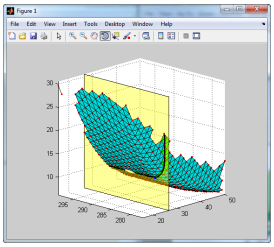
4.2 Phase II: Online phase

When the Pareto set has been obtained and decomposed into simplices, it can be opened in the navigation application. The DM uses the GUI to navigate on the sampled Pareto set by interactively changing sliders and watching the result in real-time, see Figures 5. When the DM moves the slider for one objective, the software instantaneously searches for a solution on the approximated Pareto set that corresponds to the desired value and visualizes the result, both for objectives and decision variables. It is also possible for the DM to change the boundaries of the objectives and the decision variables to see how new constraints might change the possible solutions.

The advantage of using a piecewise linear approximation of the Pareto frontier in the online phase is that a new approximately Pareto optimal solution can be obtained by solving a linear optimization problem. This type of optimization problems can be solved very fast which gives



(a)



(b)

Fig. 5. A snapshot of the two navigation application windows. The first window seen in a) shows the window used to choose solution. It includes range bars in the lower left corner representing the objectives. The DM can navigate on the Pareto frontier by pulling the small arrows above the range bars. A list of decision variable sets for different time steps is shown in the top left and the selected set is shown as range bars next to the list. The plot on the right-hand side shows the trajectory (or some other variable) for the currently chosen Pareto solution. The second window seen in b) visualizes the Pareto frontier and some additional information like the currently chosen Pareto solution (see the blue dot at end of the arrow). This is also updated continuously when the DM pulls the arrows above the range bars.

the user a smooth interaction experience when navigating, even on a normal laptop. However, a drawback of using an approximation of the Pareto frontier in the navigation is that the solution found by the tool might not be Pareto optimal or even feasible. This issue can be solved by re-running the original MOO problem with the desired solution as starting point.

Finally, it is worthwhile to remember that one of the main advantages with Pareto navigation is not to find a solution, but to interactively move around on the approximated Pareto set and get an understanding of how the objectives, the control signals and model variables relate to each other.

5. RESULTS

This section presents some results for the crane MOO problem. The number of discretization points N in the MOO NLP (9) is chosen as 200. This choice yields a reasonable compromise between computational time and numerical accuracy. The final time is bounded to the interval $10 - 50$s and the sampling times will then be less than $50/200 = 0.25$s which is sufficiently short for the crane system. Further, the masses are chosen as $m_t = 40000$kg and $m_l = 30000$kg which are typical numbers for the trolley and the grab. However, it should be noted that all figures in this section present the normalized energy z_2/m_l and maximal instantaneous power z_3/m_l.

The Pareto frontier of the crane MOO is sampled with 931 points using the generic framework. A three dimensional plot of the resulting Pareto frontier can be seen in Figure 6 and the projections of the Pareto frontier on every coordinate axis are shown in Figure 7. The magenta markers in the blue area show the Pareto optimal points computed with the Pareto filter. There were 412 Pareto optimal points and the solver IPOPT was unable to find a solution for 106 points. The gray lines show non Pareto optimal surfaces. The red circles are the individual minima that were obtained by just optimizing w.r.t. to each objective.

We have also evaluated larger cycle times (up to 200s with 800 sampling points) and as expected the energy and the maximal instantaneous power then both go to zero. However, from a practical point of view 50s is sufficiently long and having fewer sampling points reduces the solution time.

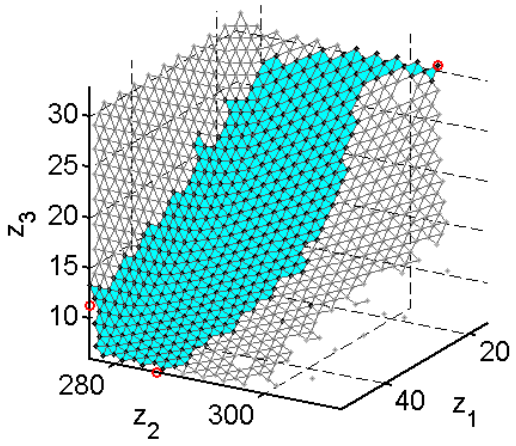
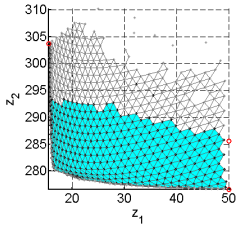
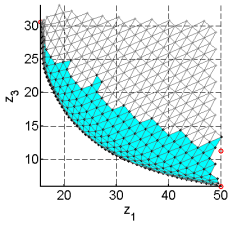


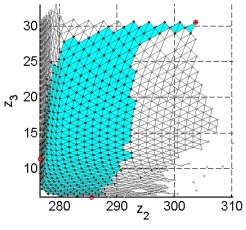
Fig. 6. The obtained Pareto set with Pareto optimal (blue faces) and non Pareto optimal (gray lines) areas. The individual minima of every objective can be seen as red circles.



(a) The Pareto set of energy z_2 with respect to time z_1.



(b) The Pareto set of power z_3 with respect to time z_1.



(c) The Pareto set of power z_3 with respect to energy z_2.

Fig. 7. The projections of the obtained Pareto set on every coordinate axis.

Analysis of the Pareto frontier can reveal many interesting facts. In Figure 7(a) it can be seen that the frontier is almost vertical on the left side. This means that the energy consumption can be decreased significantly without deteriorating the cycle time much. Actually, by increasing the total time only 0.4%, the energy consumption can be reduced by 3.4% and by increasing the time by 4.2%, the energy can be reduced by 7.4%. For the maximum power the situation is completely different as shown in Figure 7(b). In this case, the frontier has a rather even slope and therefore a reduction in the maximal power will deteriorate the cycle time as well. This kind of information is very useful when tuning the crane controller. Further, it can be noticed that energy consumption and the maximal power are conflicting objectives which might not be obvious. With other words, it means that for a given final time a lower energy consumption might require a higher maximal instantaneous power. The conflict can be seen in Figure 8 where the yellow plane corresponds to a final time of 22s and the thick black line is the corresponding 2D Pareto frontier. The explanation is quite straight forward. In order to reach a certain final time, a certain average speed is required. If the top speed is reached later, which happens if the trolley accelerates more carefully to keep the instantaneous power low, the top speed has to be higher in order to obtain the same average speed. The higher top speed requires more kinetic energy and since we do not recover any energy, more kinetic energy will be wasted during braking. Hence, from an energy perspective it is better to accelerate fast using a larger power to a lower top speed than to accelerate slower to a higher top speed. However, as mentioned earlier, the higher instantaneous power might also lead to higher costs for the harbor owner.

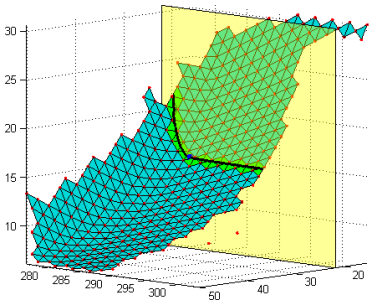


Fig. 8. The Pareto frontier cut by a plane at $t_f = 22s$. The thick black line is the 2D Pareto frontier between the consumed energy z_2 and the maximal instantaneous power z_3 obtained for the given t_f.

The trajectories of the crane for the obtained Pareto set can be seen in Figure 9. In the left plot all trajectories (gray lines) obtained in the optimization (including non Pareto optimal trajectories) are shown together with the individual minimas for each objective. In the right plot the Pareto filter has been applied and only Pareto optimal trajectories are shown. It should also be noted that the total time for one cycle is not illustrated in the figure and hence the difference in time between the trajectories can not be seen. For example, when minimizing the time only (red line with cross markers), the total time for one cycle is 15.2s, while the minimal energy solution and the minimal maximal power solution both use 50s (square markers and point markers, respectively). This relation can instead be seen on the Pareto frontier in Figure 6 and 7.

One interesting fact is that when minimizing the time, the trajectory is more curvy. The reason is that the grab reaches a higher point faster by moving the trolley back and forth and swinging the grab. However, when minimizing only energy (green line with square markers) or power (blue line with point markers) the trajectories are smoother and the paths are shorter.

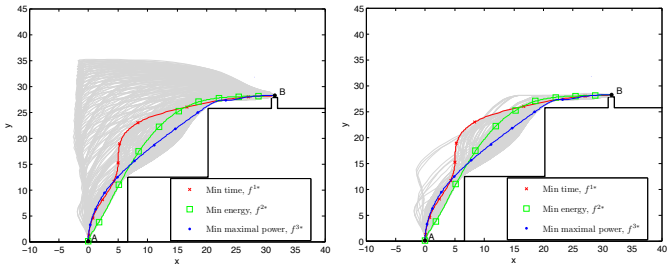


Fig. 9. The trajectories (gray lines) of the crane for the obtained Pareto set. Left plot shows the trajectory of all sampled points and the right plot shows only Pareto optimal trajectories. The colored lines with markers shows the trajectory of the individual minima of respective objective.

As mentioned before, there are two control signals, the trolley and hoist accelerations, respectively. In Figure 10 all the Pareto optimal control signals are shown (gray lines) together with the control signals for the individual minimas. Something that becomes clear immediately by studying this figure is how much the different Pareto optimal solutions actually varies. This is a little bit surprising.

Further, it can be seen that the hoist acceleration for the minimum time solution behaves like a typical minimum time solution should do. That is, it either accelerates or retards fully. However, the same is not true for the trolley and there are at least two reasons. First, the trolley sometimes must wait so that the grab avoids the obstacles. Second, the trolley moves somewhat back and forth to swing the grab higher in order to reach the destination faster because the hoist motion is the limiting motion for this crane. Together, these two reasons give the quite complicated control signal for the trolley.

The figure also shows that the control signals for the minimum energy solution are quite passive, especially the hoist acceleration. The trolley acceleration is also relatively constant except at the end where it makes a large bump. On the other hand, it can be seen that the control signals for minimum maximal power vary much more. The reason to the larger variations is still not completely clear, but partly it depends on that only one or a few samples determine the achieved minimum maximal power. The rest of the control signals can be set quite arbitrary as long as the corresponding power does not exceed the power level. This non-uniqueness easily creates oscillations. One approach to reduce the oscillating behavior which we have used is to use regularization with the total energy as shown in Section 3.3.

Concerning the navigation, the obtained Pareto set with 931 sampled points (only 412 were Pareto optimal) was tested in the online application. Every point consisted of three objective values and 8600 decision variable values which were imported into the online application. The navigation on the frontier is completely smooth and interpolates both in objective space and the decision space. However, as mentioned before, interpolation will not cross non-optimal areas as can be seen in Figure 11, where the black line shows how the Pareto optimal solution varies when the DM decreases z_1 from the minimum for z_2.

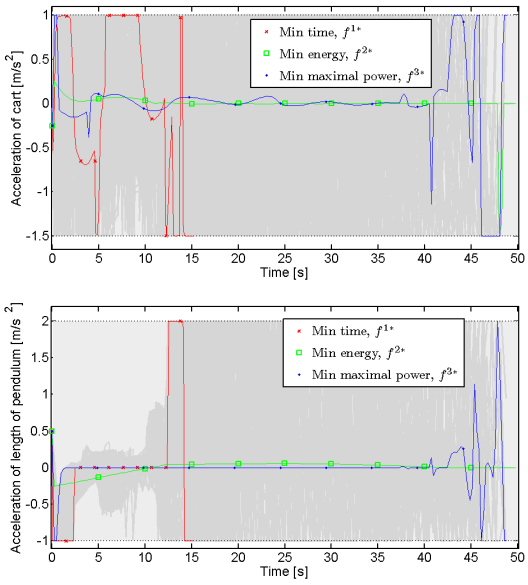


Fig. 10. The trolley acceleration (upper plot) and hoist acceleration (lower plot). The shaded area is the feasible interval for control signals and the dotted lines marks the upper and lower boundaries. The colored lines show the control signals for f_1^* (red dash-dotted), f_2^* (green solid) and f_3^* (blue dashed), respectively.

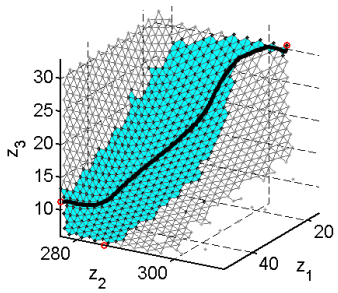


Fig. 11. The black line is the trajectory of Pareto optimal solutions obtained when the DM decreases z_1 from the minimum for z_2.

6. CONCLUSIONS

In this paper, a recent MOO framework developed in Linder et al. (2012) is applied to an optimal control problem for a crane. The framework is divided into two phases. The first phase makes a sparse sampling of the Pareto frontier by solving NLPs that are scalarized and discretized formulations of the multiobjective optimal control problem. In this phase the DM is not involved which is important since each solution takes a substantial time to compute (for the 931 sample case, the Pareto front took around 6 hours).

The second phase is an interactive online application which the DM uses to investigate an piecewise linear approximation of the Pareto frontier and choose a preferred optimal solution. This interactive application is fast enough to give a real-time feeling when the DM traverses the approximation of the Pareto frontier, which is important for the user experience.

Concerning the crane MOO, the Pareto frontier shows that the energy consumption can be reduced by roughly 7% by increasing the total time of one cycle only 4%. This is a significant possibility for improvement. Further, it was realized that the energy consumption and the maximal instantaneous power actually are actually conflicting which might be unexpected. This may be an indication that a limitation of simultaneous crane operations can reduce the overall energy consumption of the cranes, since then each crane could use a higher instantaneous power without increasing the maximal power drawn from the power grid.

REFERENCES

Al-Garni, A.Z., Moustafa, K.A.F., and Javeed Nizami, S.S.A.K. (1995). Optimal control of overhead cranes. *Control Engineering Practice*, 3(9), 1277–1284.

Albersmeyer, J. and Diehl, M. (2010). The lifted newton method and its application in optimization. *SIAM Journal on Optimization*, 20(3), 1655–1684.

Auernig, J.W. and Troger, H. (1987). Time optimal control of overhead cranes with hoisting of the load. *Automatica*, 23(4), 437–447.

Betts, J.T. (2001). *Practical Methods for Optimal Control Using Nonlinear Programming*. Advances in Design and Control. Society for Industrial and Applied Mathematics, Philadelphia.

Deb, K. and Gupta, N.K. (2004). Optimal operating conditions for overhead crane maneuvering using multiobjective evolutionary algorithms. In *Genetic and Evolutionary Computation - GECCO 2004*.

Eskelinen, P., Miettinen, K., Klamroth, K., and Hakanen, J. (2010). Pareto navigator for interactive nonlinear multiobjective optimization. *OR Spectrum*, 32, 211–227.

Hartikainen, M., Miettinen, K., and Wiecek, M. (2011). Constructing a pareto front approximation for decision making. *Mathematical Methods of Operations Research*, 73, 209–234.

Hu, G.S. and Teo, C.L. (2004). Minimum-time control of a crane with simultaneous traverse and hosting motions. *Journal of Optimization Theory and Applications*, 120(2), 395–416.

Kunkel, P. and Mehrmann, V. (2006). *Differential-Algebraic Equations — Analysis and Numerical Solution*. Textbooks in Mathematics. European Mathematical Society, Zürich, Schweiz.

Linder, J., Lindkvist, S., and Sjöberg, J. (2012). Two-step framework for interactive multi-objective optimization. Technical Report LiTH-ISY-R-3043, Department of Electrical Engineering, Linköping University.

Logist, F., Houska, B., Diehl, M., and Impe, V. (2010). Fast pareto set generation for nonlinear optimal control problems with multiple objectives. *Structural and Multidisciplinary Optimization*, 42, 591–603.

Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Springer.

Monz, M., Küfer, K.H., Bortfeld, T., and Thieke, C. (2008). Pareto navigation – algorithmic foundation of interactive multi-criteria IMRT planning. *Physics in Medicine and Biology*, 53(4), 985.

Pascoletti, A. and Serafini, P. (1984). Scalarizing vector optimization problems. *Journal of Optimization Theory and Applications*, 42, 499–524.

Sakawa, Y. and Shindo, Y. (1982). Optimal control of container cranes. *Automatica*, 18(3), 257–266.

Wächter, A. and Biegler, L. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106, 25–57.