15th IFK
Proc. of the 15th International Fluid Power Conference < Symposium/Conference >
< DOI >

Machine Learning Based Controller for Energy-Optimized Operation of Hydraulic Loader Cranes

Amy Rankka [1,2], Alessandro Dell'Amico [1]

1: Division of Fluid and Mechatronic Systems, Linköping University, Linköping, Sweden, 2: Hiab AB `amy.rankka@liu.se, alessandro.dellamico@liu.se`

Abstract.

As energy efficiency becomes an increasingly important property of hydraulic working machines, new ways of improving the system need to be explored. This paper explores the potential of improving the energy efficiency of the hydraulic system by how it is operated. With a kinematically redundant system, such as the boom system of a loader crane, there is a possibility of reducing the energy consumption of a given task by optimizing how the different actuators are operated to form the crane tip motion. This work considers the semi-automatic function Crane Tip Control (CTC) where the operator controls the velocity of the crane.

A new, energy efficient, CTC controller with two parts is developed. The first part of the controller decides the strategy of the crane tip motion by setting the speed command for one of the boom actuators. The second part handles the reference velocity following by calculating the required speed commands for the other two boom actuators. The input to the controller is the reference velocity, both speed and direction, of the crane tip together with the state of the crane boom cylinders. In its current state, the controller is optimized for a single payload, but in future versions, the payload can be estimated or measured and set as input.

The strategy part of the controller is represented by a neural network, that is trained by a Reinforcement Learning approach with a hydraulic energy minimization objective. The training is performed on a simulation model of the crane, which includes geometrical relations for the crane booms and cylinders, and an energy consumption estimation.

The performance of the new CTC controller is benchmarked with regards to two other CTC controllers. The first one is a near-optimal offline controller implemented by a dynamic programming optimization. The second one is a a rule-based controller with a strategy that always holds one of the actuators still. An energy flow analysis has revealed great losses due to simultaneous operation of several actuators. Although the rule-based controller is not optimized with respect to the crane's operation, it is an important comparison since it provides a less development heavy solution.

Keywords. loader cranes, energy efficiency, semi-autonomous operation, neural network controller, redundancy resolution

1. INTRODUCTION

Although, improving the energy efficiency of hydraulic working machines has been widely investigated for a long time with many interesting concepts being presented, very few solutions has been implemented by the industry due to the associated cost and complexity with changing to a new architecture. This study explores a way of reducing the energy consumption by automating the movement of the machine, and thus, does not require any hardware changes. The work is applied to a loader crane, which is a kinematically redundant system, meaning it has an extra degree of freedom which can be used for optimizing the movements with regards to some objective. Loader cranes are not yet fully automated, but there is a commercially available semi-automatic feature by several manufacturers, see for example [12], [14] and [13], where the motion of the crane tip is controlled by the operator and the corresponding motions of the hydraulic cylinders are controlled by the control system. The feature can be denoted Crane Tip Control, CTC, and in this paper an approach to minimize the energy consumption during CTC operation is investigated.

1.1 Problem Formulation

The CTC feature which is used as reference here is provided by crane manufacturer Hiab [12]. It is optimized for lifting capacity, and in order to investigate how large impact an optimized automated feature can have on the energy consumption, the aim of this study is to develop a new CTC controller that is optimized for energy efficiency. A complete CTC controller should be able to take any valid operator command, which is given in the form of a speed command in the horizontal direction of the crane tip and a speed command in the vertical direction, and provide a set of cylinder speed commands that result in this motion. The CTC feature considered here works in a two-dimensional plan and controls the three boom cylinders of the crane, the slewing function is excluded. The scope of this study is limited to investigate movements in the horizontal direction with a fixed positive speed. CTC is a realtime feature which must be executed on the embedded control system of the crane. For this reason, traditional offline optimization methods are excluded. Instead, a machine learning approach, where part of the CTC controller is set up as a neural network, is explored. A simple rule-based controller is also developed to form a baseline. As a best case benchmark, near-optimal solutions to a limited number of test cases are calculated by an offline Dynamic Programming algorithm developed in previous work by the authors [8].

1.2 Delimitations

The scope of this paper is limited to investigating one type of neural network structure, feed forward neural network, as well as using one Reinforcement Learning approach, policy optimization, as training method. Further, the CTC controller is limited to working with a fixed speed and payload in the positive x-direction of the working area of the crane. The main focus will be on presenting results from one such controller, although results for training controllers with different speed and payload will be briefly presented to show the potential for extending the method in future work.

1.3 System Description

The system used as reference application in this work is a medium sized loader crane with three hydraulic cylinders used by the CTC feature,the first boom, "$1b$", cylinder, the second boom, "$2b$", cylinder and the extension, "ext", cylinder. The first and second boom cylinders control revolute joints with angles denoted α_{1b} and α_{2b} respectively. The extension cylinder works as a prismatic joint with a position denoted l_{ext}. The position of the three joints together with the dimensions of the crane defines the crane pose. The kinematic redundancy gives that there can be an infinite amount of different crane poses for a single crane tip position. The crane tip position is given in Cartesian coordinates in the plane with axes x and y.

The cylinders of the crane are connected to a pressure compensated load sensing hydraulic system with a variable displacement pump. The pressure compensation gives that the speed commands to each cylinder, denoted u_{1b}, u_{2b} and u_{ext}, can be directly translated to spool position commands to the directional valve.

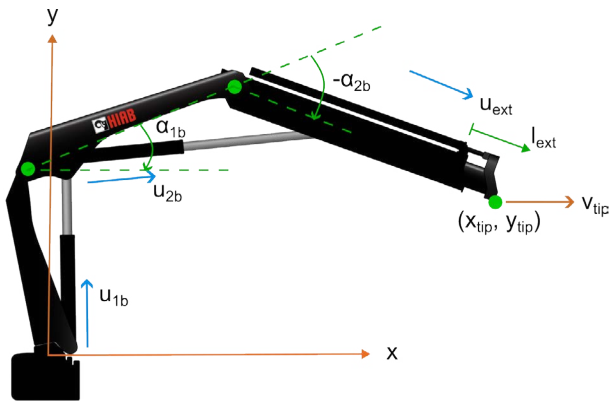

Figure 1: Crane with cylinders, joints and crane tip position.

1.4 Prior Art

The problem for a CTC controller or similar is to translate a desired motion in the task space (crane tip movement in the CTC case) to a motion in the joint space (cylinder velocities in the CTC case). To go from tip velocity to joint velocities the inverse of the so called jacobian matrix has to be somehow calculated. For a redundant system, the exact inverse of the jacobian does not exist. A common way to solve the problem is to compute the pseudo-inverse of the matrix, which gives the solution with the smallest joint velocity magnitude. This method is described for example in [4] and [3]. The drawback with this approach is that only locally optimal solutions can be found and that minimizing the joint velocity magnitude is the only objective. In [5] it is shown that the energy consumption performance of the pseudo-inverse method is significantly worse than for a dynamic programming algorithm.

In [11], the pseudo-inverse solution is used for precision tracking of a three degree of freedom robotic arm and combined with a minimization of the kinetic energy by a velocity vector pointing in the direction where the kinetic energy integral is minimized for a chosen trajectory. The optimal control is predicted for a window and continuously updated as the simulation is rolled out. The resulting controller is shown to perform pretty close to the global optimal solutions on trajectories with a length of 10 s. However, to only minimize the kinetic energy is too much a simplification for the crane with a load sensing hydraulic system, where kinetic energy cannot be directly translated to input energy to the system.

The possibility of using a neural network as CTC controller has been presented by [10]. In this work, the CTC controller is optimized with the objective of moving the crane tip as fast as possible. The same methodology has been applied to the crane used as application in this study with good results considering the speed and precision of the crane tip movement.The promising results gives an incentive for exploring a similar methodology and controller structure but with an energy minimization objective. This work was started in [8] where an energy efficient CTC controller was trained for a very limited part of the working area of the crane with good results.

A neural network controller has also been implemented for controlling the automatic log collecting of a forwarder in [1]. The network is trained with a reinforcement learning approach in a simulation environment and shows promising results with regards to energy efficiency. In another recent publication, [2], a controller for autonomous operation of a forwarder is also developed by reinforcement learning. The controller is trained in a simulation environment and later deployed on a real forwarder with good results.

2. Baseline and Benchmark

The energy consumption of moving the crane tip according to some specified trajectory with different strategies for moving the cylinders differ mainly due to three properties of the system:

Total oil need The total traveled distance of the cylinders can differ and affects the average flow demand of the pump.

Cylinder pressures Different cylinders require different pressure levels for lifting the payload in different crane poses. The cylinder pressures depend on the load force, friction and cylinder dimensions. The highest cylinder pressure (or more accurately the highest load pressure sensed by the directional valve) sets the pump pressure.

Simultaneous losses In a load sensing system with a single pump, the flow to all cylinders active simultaneously is supplied at the same pressure. If a cylinder requires a lower pressure, the excess pressure is wasted at the compensator valve.

For an optimized system, all these three properties and how they together form the flow and pressure demand of the pump at every instant must be taken into account, as well as the total energy of a movement of some length of time. A Dynamic Programming, DP, algorithm that can make this optimization for a discretized version of the system has been developed in previous work [8]. Results from this algorithm will serve as a benchmark for the new controller in Section 5.

Without any optimization, a simple rule-based controller that tries to minimize the third property, simultaneous losses, can be developed. The strategy of this controller is to always keep one cylinder still and to use the other two to make the necessary motion. A previous study, [7], of the reference loader crane shows that simultaneous losses account for about 30% of the losses in the hydraulic system. At the start of a motion, the controller tries to keep the 1st boom still, as this is often the cylinder that has to carry the most weight. If this leads to an illegal solution, due to violation of maximum cylinder positions or speeds, it tries to keep the 2nd boom cylinder or extension cylinder still. During the motion, it then keeps the strategy of the previous timestamp as long as possible, in order to achieve smooth motions. Results from this controller will serve as a baseline for the new controller in Section 5.

3. Controller Structure

The new CTC controller optimized for energy consumption consists of two parts, a strategy controller and a velocity controller, and its layout is presented in Figure 2. The inputs to the controller are the crane state, represented by the joint angles and extension position, and the reference tip velocity. The outputs are the speed commands to the three boom cylinders. Development of the controller was started in the work presented in [8], and the method has now been extended to training a controller that is able to control the crane tip motion in the whole working area of the crane.

The strategy controller is implemented as a feed forward neural network with three hidden layers with 32 nodes each. Adding more nodes than that does not improve the performance of the network.

LeakyRelu is used as activation function for all layers except the output layer, where instead HardTanh is used in order to get the output saturated to [-1,1]. -1 corresponds to maximum negative speed command, and 1 to maximum positive speed command. LeakyReLU stands for leaky rectified linear unit and introduces non-linearity to the network with reduced risk of de-activating neurons compared to simple ReLU. LeakyReLU is defined according to eq. (1), where α is set to 0.01.

$$LeakyReLU(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{if } x \leq 0 \end{cases} \tag{1}$$

The task of the strategy controller is to select a strategy that minimizes the required pump energy to move the crane some distance with CTC. It should be noted that the objective is not to minimize the instant pump power, but to look ahead some distance and minimize the energy consumption of a movement that goes on for some time. An assumption then has to be made regarding how the crane tip reference will be in the near future. In this work, the energy is optimized on trajectories of a length of 13 s with a constant reference speed of 200 mm/s. With this speed, the crane can cover about a third to half of the working area in the horizontal direction during a trajectory, depending of the vertical position of the crane tip. This distance is considered a reasonable assumption of how far the crane is moved along a constant direction during normal work. The selection of the time horizon over which the accumulated cost during optimization is calculated can also be seen as a trade-off between promoting long term gains while still being able to adapt to changes in the input.

The output from the strategy controller is the speed command to the extension cylinder. It could just as well have been the command to the 1st or 2nd boom, the key is that the strategy is set by deciding the motion for a single one of the cylinders since the system only has one extra degree of freedom.

Once the output to the extension cylinder is given, the problem of finding the output to the 1st and 2nd boom has a unique solution. The velocity controller thus takes the output from the strategy controller and calculates the output to the 1st and 2nd boom while making sure that the reference crane tip trajectory is followed by using the inverse kinematics equations of the boom system.

If no valid combination of cylinder positions can be found, the velocity controller sets all outputs to zero. It thus functions as a layer between the neural network and the crane that can also act as a safety layer that can stop the crane, or reduce the speed, in unsafe situations when deployed on a real crane.

4. Training Environment and Simulation Model

The network is trained by Reinforcement learning policy optimization in a simulation environment implemented in Python, where the network functions as the policy approximator. The PyTorch framework is used for setting up the network and running the training loop. The crane is implemented as a quasi-static geometry model. Since the network only is trained to provide a high level motion strategy, no dynamics is needed in the crane model. It is however, important to have a good energy consumption model, since the strategy decisions are made based on this. The simulation model boundary is set at the outlet of the variable pump, so the signals of interest are the hydraulic pump pressure and pump flow.

The flow leakage in the crane is low and the pump flow is thus modelled directly as a summation of the oil required to move each cylinder according to the motion within the

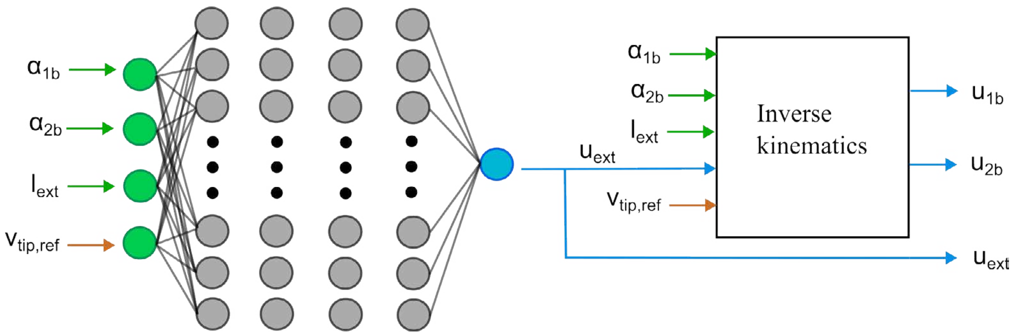

Figure 2: Layout of the neural network for the strategy controller and its connection to the velocity controller.

timestep, according to Equation 5. The hydraulic system is a load-sensing system, which gives that the pump pressure can be calculated as the maximum of the load pressures plus a pressure margin, according to Equation 6, where $p_{func,cyl}$ are the active side cylinder pressures, Δp_{func} are the pressure drops between the valve and the cylinders and Δp_{pump} is the pump pressure margin.

The cylinder pressure level required to move each cylinder depends on the force on the cylinder induced by the payload and the dead weight of the crane, on the counter pressure and on the friction in the system. The force can be calculated based on the geometry model, but given that it is hard to physically model the friction, especially in the extension system, the rest of the contribution to the cylinder pressure is modelled by a black-box model, with cylinder flow and force as inputs, fitted to measurement data.

Initially, a machine learning approach with gaussian process models, presented in [9], was tested. Although the fit of the model was very good in most working points, it was sensible to the distribution of the data points and its interpolation scheme could produce outputs that were not in line with the expected behavior. For example, if the force is kept constant, the pressure of a cylinder chamber is expected to increase with speed, but for some specific force values, the pressure could decrease for a small range of increasing speed.

Inconsistencies like this are easily exploited by the dynamic programming optimization, but more difficult for the gradient based optimization of the network to find, creating differences in the results that might not hold on the real system. For this reason, polynomials of the second degree are used instead, and the coefficients are fitted with with the least square minimization criterion. For the extension cylinder, different polynomial models are created depending on direction of movement and gravitational force, and the currently active pressure, P_{active}, can be calculated according to equations 3 to 4, where $P_{pist,static}$ and $P_{rod,static}$ are the contributions from the static force on the cylinder to the piston and rod side chamber pressure, and $P_{friction}$ is the contribution from the friction force given by the blackbox model. Figure 3 show the measurement data for the friction induced pressure as a function of cylinder flow for one of these cases, retraction of the cylinder and gravitation force pulling the cylinder in the extension direction, together with the fitted model. Figure 4 show the resulting fit of the active pressure to validation data from a test cycle when all four extension models are combined.

$$P_{pist,static} = \cdot \frac{F_{ext}}{A_{pist}} \tag{2}$$

$$P_{rod,static} = \cdot \frac{F_{ext}}{A_{rod}} \tag{3}$$

$$P_{active} = \begin{cases} |P_{rod,static}| + P_{friction,F-,v-} & \text{if } F_{ext} \leq 0 \text{ and } q \leq 0 \\ -|P_{pist,static}| + P_{friction,F-,v+} & \text{if } F_{ext} \leq 0 \text{ and } q > 0 \\ -|P_{rod,static}| + P_{friction,F+,v-} & \text{if } F_{ext} > 0 \text{ and } q \leq 0 \\ |P_{pist,static}| + P_{friction,F+,v+} & \text{if } F_{ext} > 0 \text{ and } q > 0 \end{cases} \quad (4)$$

The pressure drop between the valve and the cylinder is modelled as a simple flow dependent pressure drop model with coefficients fitted to measurement data together with load holding valve models. These models are presented in [6].

$$q_{pump} = \frac{\Delta 1b_{clen}}{T_s} * A_{1b,cyl} + \frac{\Delta 2b_{clen}}{T_s} * A_{2b,cyl} + \frac{\Delta ext_{clen}}{T_s} * A_{ext,cyl} \quad (5)$$

$$\begin{aligned} p_{pump} = & \max[p_{1b,cyl} + \Delta p_{1b}, p_{2b,cyl} + \Delta p_{2b}, p_{ext,cyl} + \Delta p_{ext}]+ \\ & + \Delta p_{pump} \end{aligned} \quad (6)$$

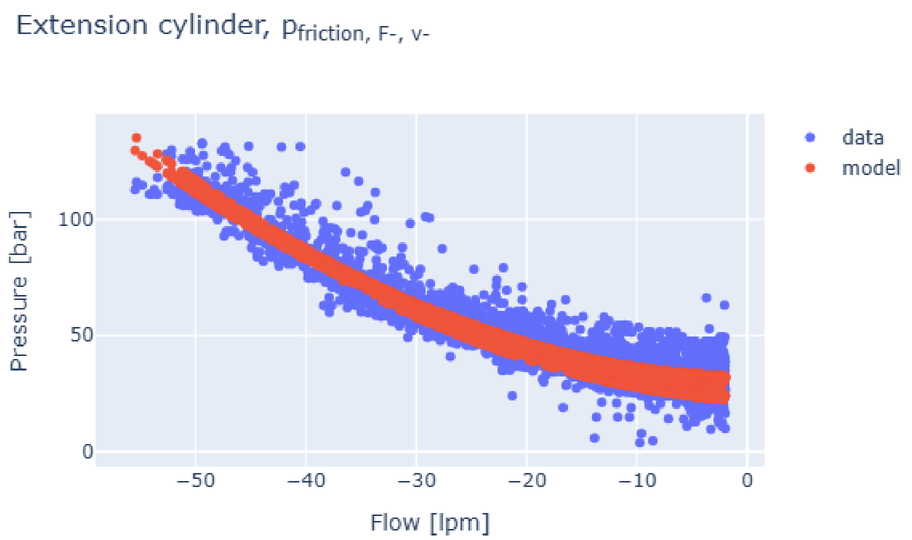

Figure 3: Model of extension friction based pressure as a function of flow. The variation for a fixed flow comes from the small impact of the static force.

During training of the network, a batch of 1000 cylinder positions that corresponds to crane tip positions within the working area of the crane, see Figure 5, is sampled. A simulation with a fixed timestep on a fixed horizon is rolled out on the batch for a number of episodes. Given that the objective of the optimization of the network is to minimize the energy consumption, the cost function is formulated according to eq. (7), where P_i is the pump power required during the current step and H is the time horizon of the simulation.

$$C = \sum_{i=1}^{H} P_i \quad (7)$$

$$P_i = q_{pump,i} * p_{pump,i} \quad (8)$$

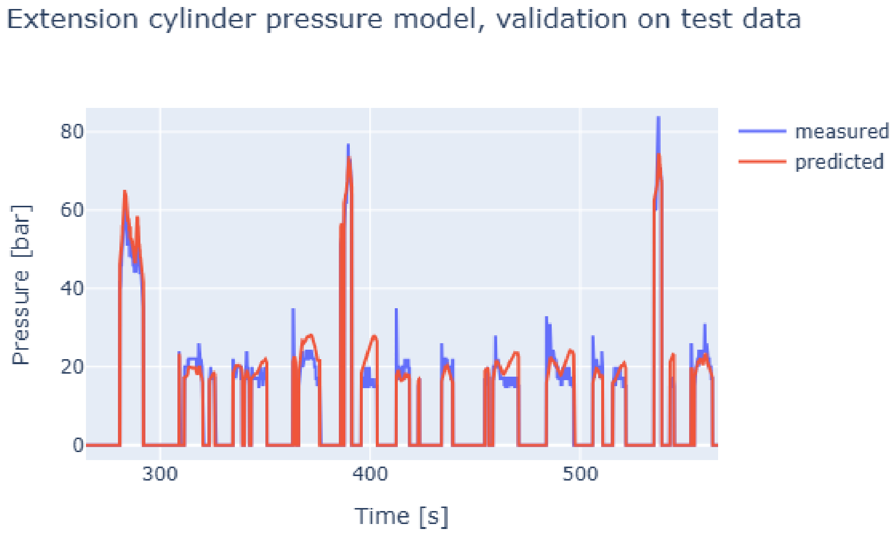

Figure 4: Validation of model of active extension cylinder pressure.

The policy, that is the network during training, is probabilistic. In order to allow the system to explore new solutions, the current best guess is not always selected as output from the policy. Instead, the network is set to produce a mean value and a standard deviation that can form a distribution from which the output (command to the extension cylinder) is sampled. When the controller later is deployed, the sampling from the distribution is replaced by directly taking the mean output from the network.

The network is trained until convergence. New batches of initial positions are sampled at regular intervals to ensure that the solution is generalized. The built-in gradient-descent optimizer AdamW of Pytorch is used to update the weights of the network after each episode based on the gradients of the average cost of all the trajectories in the batch with respect to the weights, or by using the Reinforcement learning terminology, the policy is optimized with respect to the (negative) reward it was given for the actions taken during the episode.

5. Results

The new controller is evaluated by comparing to the rule-based controller baseline and the dynamic programming benchmark on a number of pre-defined test cases. The test cases are divided on four different crane tip trajectories located at different parts of the working area of the crane. For each trajectory there are four different test cases formed by four different starting crane poses. The four trajectories, with one of their starting crane poses each, are displayed in Figure 6. The graphs visualize the motions of the crane when controlled by the new controller. The performance of the new controller, and its similarities and differences to the dynamic programming solutions, on each trajectory is presented in Sections 5.1 to 5.4. A summary of the energy performance comparing both with the dynamic programming benchmark and the rule-based controller baseline, is found in Section 5.5.

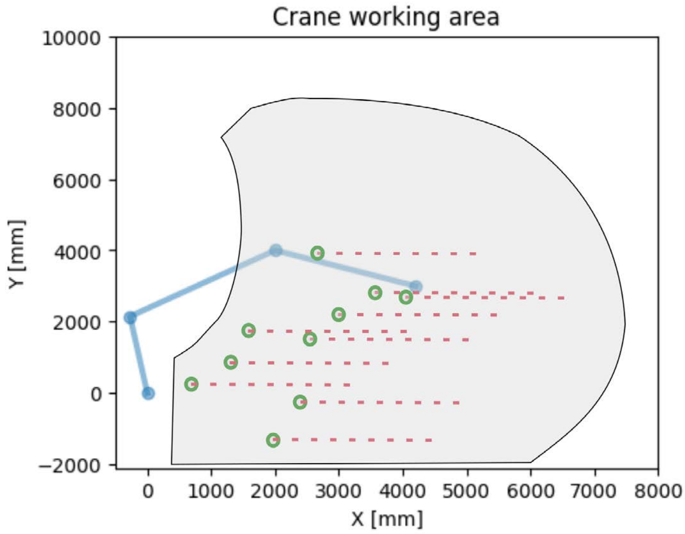

Figure 5: Working area of the crane, on which the neural network controller is trained. The figure also shows 10 example trajectories, with green circles for the starting points and dashed red lines for the trajectories, from a batch on which the controller is evaluated on during each episode.

5.1 Trajectory 1

A comparison between the cylinder commands, active side cylinder pressures, pump flow and pump pressure, for the dynamic programming, DP, controller and the new, neural network based, NN, controller is displayed in Figure 7 for one of the test cases for trajectory 1. At a quick glance, the cylinder commands are rather similar, but while the dynamic programming controller holds the extension still at most times, the new controller constantly runs it at a low speed. This creates a big difference in pump pressure, as the extension cylinder sets the highest pressure during the first half of the movements for the new controller. As the flow demand is similar for both controllers, the much higher pressure demand for the new controller results in a significantly bigger power demand.

The new controller is not able to find the solution where the extension is held still during training due to that the possibility for a neural network to output exactly zero is extremely small. It is however, possible to make adjustments after the training is completed. If the command to the extension is lower than a certain limit, it is plausible to assume that the power demand will be similar, or lower, if the command is set to zero. To keep the crane tip reference following, the 1st and 2nd boom have to be adjusted accordingly, which is handled by the velocity controller. If the adjusted outputs, or resulting cylinder positions, turn out to be illegal, the controller reverts back to the original solution. If this adjustment is applied to the test case 1.3, the pump power is now much more similar to that the DP controller, see Figure 8. The other three test cases of trajectory 1 follows a similar pattern.

5.2 Trajectory 2

Trajectory 2 shows a slightly different scenario, displayed in Figure 9 for one of the test cases. The dynamic programming controller once again keeps the extension still for a large part of the movement, but contrary to during trajectory 1 the extension does not set the highest pressure, so the pump pressure levels are rather similar for both controllers. The

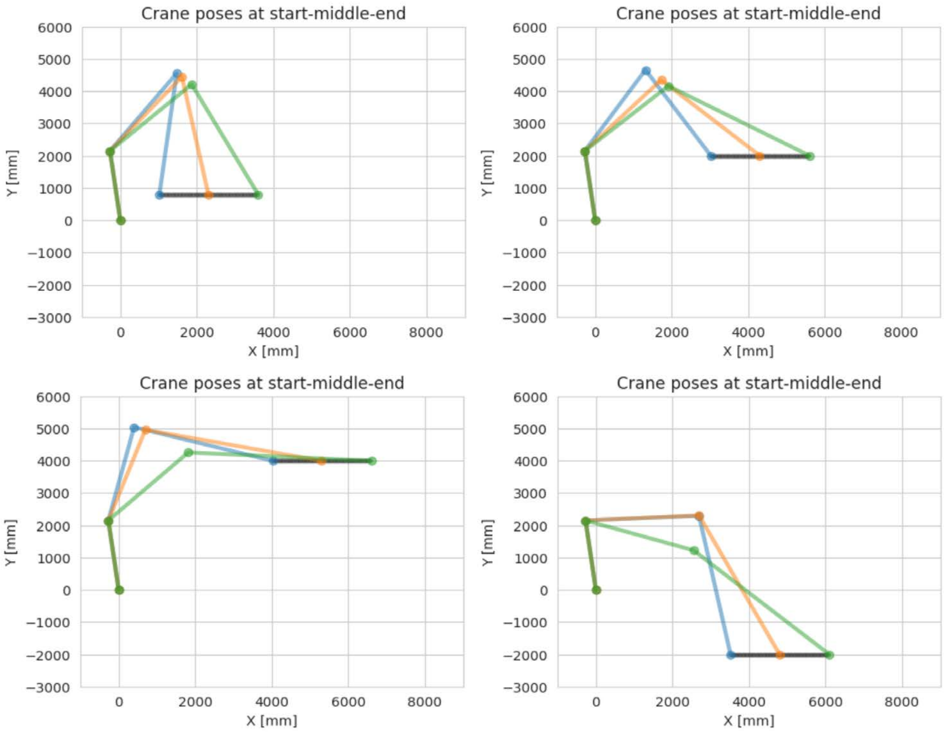

Figure 6: Four different trajectories on which the new controller will be evaluated. One of the four starting crane poses (blue lines), and the corresponding motion (orange lines for mid-time pose and green lines for end pose) of the crane while controlled by the new controller are displayed for each trajectory.

dynamic programming solution does, however, reduce the flow demand compared to the new controller.

5.3 Trajectory 3

The third trajectory is placed in the upper part of the working area of the crane. For all four test cases, the dynamic programming algorithm is able to find a smart solution where the 2nd boom is moved towards a horizontal position during the first part of the trajectory, allowing the extension to be the only function moved in the last part of the trajectory, exemplified by test case 3.2 in Figure 10. Finding exactly the state where the 2nd boom is horizontal is too difficult for the new controller, and consequently it suffers from a high pump pressure being set by moving the 2nd boom upwards during the whole trajectory. The difference between the 2nd boom and the extension pressure can be seen in the plot.

5.4 Trajectory 4

The last trajectory in the comparison is placed in the lower part of the working area of the crane. On this trajectory it is once again highlighted what a big difference to the energy consumption it can make between moving a function slowly or stopping it completely and compensating the movement by the other two functions. The new controller has found solutions where the 1st boom is initially driven slowly upwards, causing the pump pressure to be much higher than for the dynamic programming controller solution. By implementing a similar adjustment as for the extension, that is, a dead zone is placed for small positive

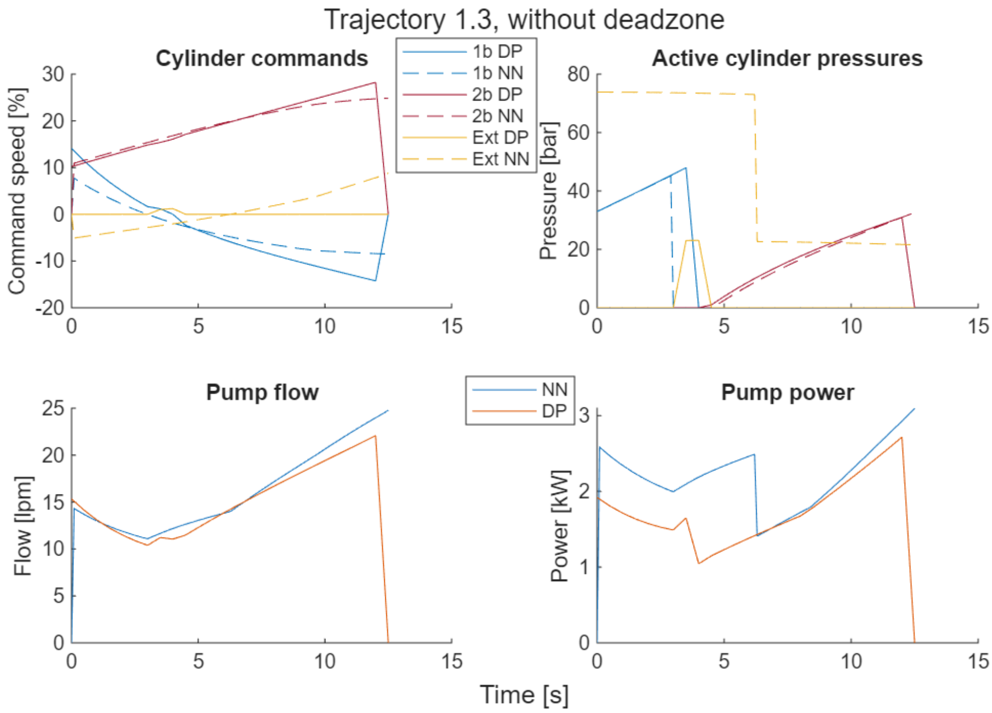

Figure 7: Test case 1.3 with original outputs from the neural network based controller. Note the high pressure set by the extension during the first half of the movement for the NN controller.

commands to the 1st boom which sets the 1st boom command to zero if the adjusted solution is legal. The resulting cylinder commands, as well as active cylinder pressure, pump flow and pump power, are displayed in Figure 11 together with the dynamic programming controller outputs. Even though there are some differences regarding when and how the extension is activated, the resulting mean power consumption is rather similar for both controller, since the 2nd boom is operated in the same direction and sets the highest pressure during the main part of the trajectory.

5.5 Power Comparison

The mean power consumption for each of the 16 test cases is displayed for the three different controllers, the dynamic programming controller (DP), the rule-based controller (r-b) and the neural network based controller (NN), in Table 1. The increase in average power consumption between the dynamic programming controller and the neural network controller varies between 5 % and 34 %, with a mean value of 15 %. In order to be able to draw any conclusions about the quality of these figures, the neural network controller is also compared to the rule-based controller. The average power consumption is most often lower for the neural network controller, with a maximum reduction of 22 %, but for some test cases the consumption is up to 10 % higher for the new controller. The mean value of the reduction is 5.6 %.

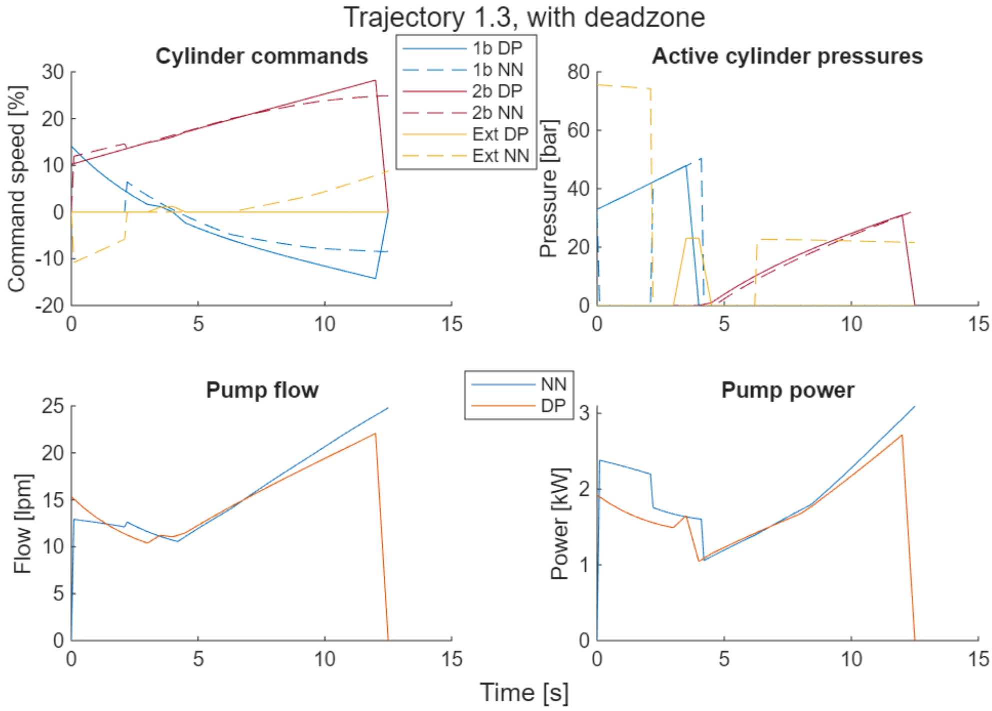

Figure 8: Test case 1.3 with adjusted outputs from the neural network based controller. Note the similar pump power for the two controllers.

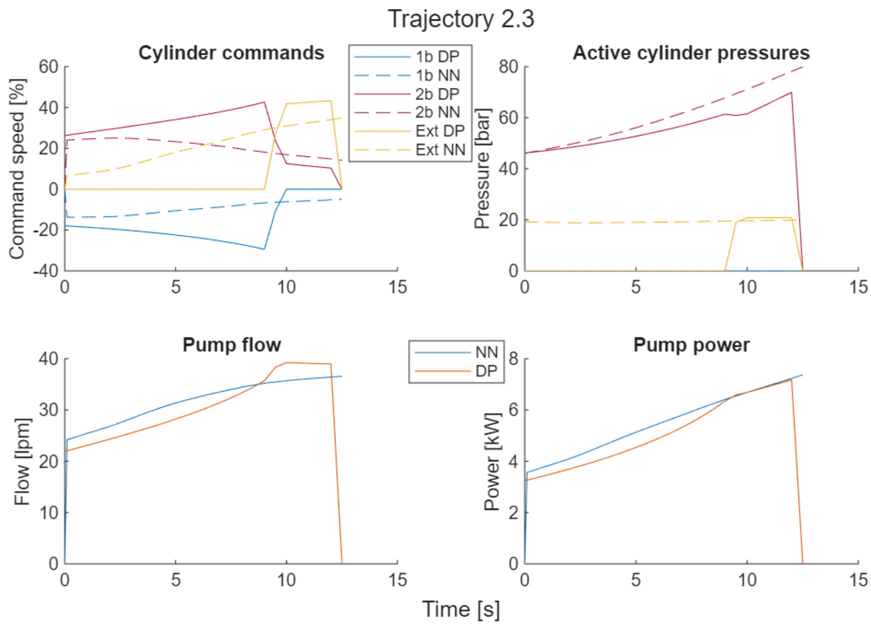

Figure 9: Test case 2.3. Note the lower pump flow for the dynamic programming controller.

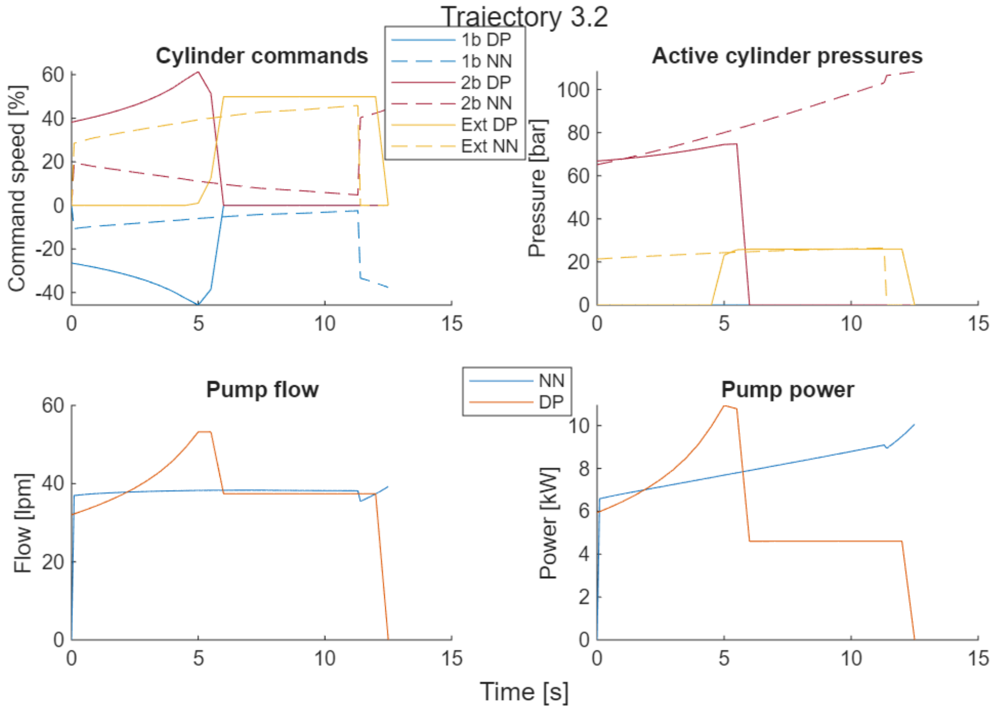

Figure 10: Test case 3.2. Note how only the extension is moved in the second half of the movement for the DP controller.

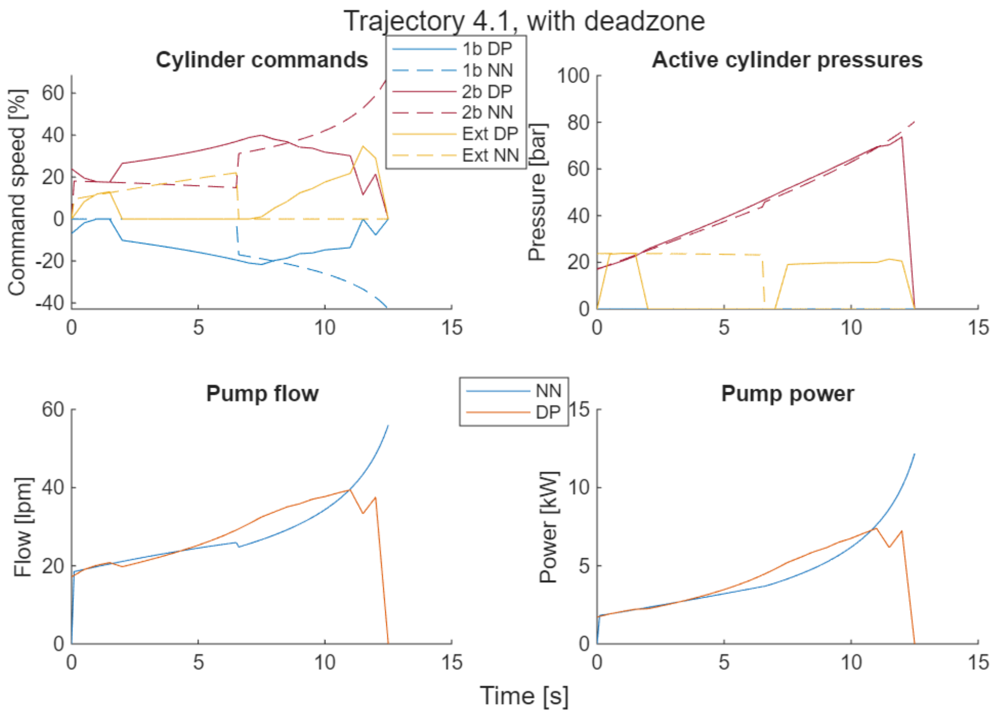

Figure 11: Test case 4.1. Note how the 2nd boom sets the highest pressure during the main part of the trajectory and how the power difference only comes from the flow.

The controller presented above has been optimized for the tip speed of 200 mm/s and the payload of 1000 kg. Two other controllers, one with 100 mm/s tip speed and 1000 kg payload, and one with 200 mm/s tip speed and 2000 kg payload, have also been optimized and evaluated on the same trajectories. The first one has an average power consumption that is 6.5 % higher than for its corresponding dynamic programming solution. The main reason for the smaller difference than for the faster test cases is that the optimizer more often outputs small cylinder commands that can be set to zero during deployment and then be more similar to the dynamic programming solutions. The one with a higher payload has an average power consumption that is 22 % higher than for its corresponding dynamic programming solution. The higher payload gives a larger relative difference in pump pressure between solutions from the dynamic programming where one of the cylinders is often kept still and solutions from the neural network controller where this rarely happens.

6. Discussion

For some of the tested trajectories, the neural network controller finds similar solutions as the near-optimal dynamic programming algorithm. These trajectories are characterized in that the cylinder commands have continuous like shapes that seems to be easier for the network to find than solutions where the commands vary more abruptly or take very specific values. Smooth motions of the cylinders are probably more easy for a real crane to follow without any noticeable small stops or small jerks of the crane tip, so in that perspective the new controller provides good solutions.

However, the results here show that energy efficiency of the neural network controller is rather low compared to the dynamic programming solution. The neural network controller only reduces the energy consumption about 6 % compared to a simple rule-based controller. When taking into account that a complete CTC controller needs to be able to control the crane tip in every possible direction, not just vertical or horizontal, the benefits of the neural network controller will be even smaller. It would be possible to train three more networks, horizontal inwards and vertical up and down, and then combine them with the network presented here for horizontal outwards. But whenever the crane tip should move along a diagonal, there is no guarantee that the energy minimization objective will hold. So in total, a lower energy reduction than 6 % compared to the rule-based controller can be expected.

If a semi-automatic function is required for the autonomy itself, rather than being a means for energy efficiency, it should be noted that there is a rather big difference in energy consumption between an optimized controller and more simple controller. If the semi-automatic function is considered as a means for energy efficiency, the results of the controller evaluated in this work must be put in relation to manual operation of the crane. In previous work, [8], it was shown that a CTC controller optimized with dynamic programming consumed about 14% less energy on a drive cycle than manual operation. Given that the results in this paper indicate that a realtime CTC consumes more than 15% more than the dynamic programming solution, automation through CTC is not considered an effective means of energy reduction. It should be noted that during the manual operation of the drive cycle in [8], the crane tip was not forced to follow straight lines. So automation could still potentially be a means of energy reduction, if given the freedom to move arbitrarily between waypoints.

Why is the trained network controller not better? The network structure and training method with a gradient based optimizer was chosen for his study due to its success in teaching a neural network to control the crane to be fast during CTC operation, proven in a previous study [10]. The kinematic equations of the crane are differentiable and worked well with the gradient based optimizer. Switching to energy consumption as objective introduced some properties that proved to be difficult for the selected method to handle.

Table 1: Comparison of average power demand [kW] for the three different controllers for the 16 test cases (four different trajectories with four different starting positions each).

	$P_{avg,DP}$	$P_{avg,NN}$	$P_{avg,r-b}$	Diff DP-NN	Diff r-b - NN
TC 1.1	1.52	1.68	2.16	+11%	-22%
TC 1.2	2.06	2.48	2.44	+20%	+1.7%
TC 1.3	1.68	1.93	2.36	+15%	-18%
TC 1.4	1.89	2.32	2.46	+23%	-5.6%
Average TC 1				+17%	-11%
TC 2.1	4.98	5.42	6.11	+8.8%	-11%
TC 2.2	4.60	5.35	5.50	+16%	-2.7%
TC 2.3	4.89	5.47	5.95	+12%	-8.2%
TC 2.4	4.80	5.59	6.21	+16%	-9.9%
Average TC 2				+13%	-8.0%
TC 3.1	6.70	8.03	8.35	+20%	-3.9%
TC 3.2	6.04	7.94	8.40	+31%	-5.6%
TC 3.3	7.01	7.76	7.62	+11%	+1.9%
TC 3.4	6.02	8.08	8.38	+34%	-3.7%
Average TC 3				+24%	-2.8%
TC 4.1	4.18	4.42	4.86	+5.9%	-9.0%
TC 4.2	4.21	4.46	4.54	+6.0%	-1.8%
TC 4.3	4.42	4.68	4.74	+5.8%	-1.3%
TC 4.4	4.10	4.31	3.95	+5.2%	+9.4%
Average TC 4				+5.7%	-0.7%
Average all				+15%	-5.6%

A linear combination such as the output from the feed forward neural network has an inherent difficulty of taking on very specific values, such as exactly zero or the command the moves the extension cylinder to a very specific position. Small reference command outputs from the controller can be adjusted to exactly zero during deployment of the controller, which was shown to significantly improve the performance. It is however not possible to apply this deadband during training as it is a non-differentiable function that hinder the gradient based optimizer to track the gradients of the loss function. Another property of the system that it is difficult for the optimizer to track is the non-linearity of the hydraulic system. For a load sensing system, the highest load pressure of the currently active cylinders sets the pump pressure. When the setup of active cylinder changes, the pump pressure can make a large step. So, a neural network could be a good choice for controller if a deadband is applied during training, and if an optimizer that does not rely of gradients of the loss function is used. Non-gradient based optimizers are for example population based methods such as genetic algorithms and particle swarm optimization.

7. Conclusion

Training a neural network to learn an optimal control strategy with regards to energy efficiency that can control a large part of the working envelop of the crane has proven to be a difficult task with the investigated optimization method. On average, the network controller consumes 15 % more energy than the near-optimal solution on the test cases presented here, and only consumes 5.6 % less energy than a simple rule based controller.

During the work of this paper it has become clear that the properties of the system to be optimized, that is the energy consumption of the crane and its load sensing hydraulic system, is not very well suited for gradient-based optimization. For optimizing the speed of the crane tip, gradient-based optimization worked well. But when considering the energy consumption of the hydraulic system a significant non-linearity is introduced - the pump pressure can make a large step when the reference to a cylinder goes from zero to a non-zero value, if the load on that cylinder is higher than the previously active cylinders. Future work will be focused on investigating if other types of optimizers that does not need to calculate gradients, such as different population based methods, are better suited for training neural networks to act as controllers for a load sensing hydraulic system.

Acknowledgment

The authors would like to thank Hiab AB, and Reza Ghabcheloo at Tampere University for their support during the work with this paper. The work in this paper was sponsored by the Swedish Energy Agency, and is part of STEALTH II - Sustainable, Electrified and Automated Load Handling, App. no 44427-3.

References

[1] Jennifer Andersson, Kenneth Bodin, Daniel Lindmark, Martin Servin, and Erik Wallin. Reinforcement learning control of a forestry crane manipulator. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021.

[2] Rohit Dhakate, Christian Brommer, Christoph Bohm, Harald Gietler, Stephan Weiss, and Jan Steinbrener. Autonomous control of redundant hydraulic manipulator using reinforcement learning with action feedback. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022.

[3] Ignacy Duleba and Michal Opalka. A comparison of jacobian-based methods of inverse kinematics for serial robot manipulators. *Int. J. Appl. Math. Comput. Sci.*, 23:373–382, 2013.

[4] Rickard Nilsson. Inverse kinematics, 2009.

[5] Jarmo Nurmi and Jouni Mattila. Global energy-optimal redundancy resolution of hydraulic manipulators: Experimental results for a forestry manipulator. *Energies*, 10, 2017.

[6] Amy Rankka and Alessandro Dell'Amico. Loss analysis and concept comparison for electrically driven hydraulic loader crane. In *The 17th Scandinavian International Conference on Fluid Power, SICFP'21*, 2021.

[7] Amy Rankka, Alessandro Dell'Amico, and Petter Krus. Drive cycle generation for a hydraulic loader crane. In *The 16th Scandinavian International Conference of Fluid Power*, 2019.

[8] Amy Rankka, Marcus Rösth, and Alessandro Dell'amico. Evaluating the performance of semi-autonomous kinematically redundant loader crane operation. In *Advancements in Fluid Power Technology: Sustainability, Electrification, and Digitalization. Proceedings of the Global Fluid Power Society PhD Symposium 2024*. Springer Nature, 2024.

[9] Abdolreza Taheri, Pelle Gustafsson, Marcus Rosth, Reza Ghabcheloo, and Joni Pajarinen. Nonlinear model learning for compensation and feedforward control of real-world hydraulic actuators using gaussian processes. *IEEE Robotics and Automation Letters*, 7:9525–9532, 10 2022.

[10] Abdolreza Taheri, Amy Rankka, Pelle Gustafsson, Joni Pajarinen, and Reza Ghabcheloo. End-effector cartesian velocity control for redundant loader cranes using reinforcement learning. *IEEE Transactions on Robotics*, 2024.

[11] Alessandro Tringali and Silvio Cocuzza. Finite-horizon kinetic energy optimization of a redundant space manipulator. *Applied Sciences (Switzerland)*, 11:1–15, 3 2021.

[12] Hiab Official Webpage. Ctc – crane tip control taking ease of use to a new level, 2020.

[13] John Deere Official Webpage. Intelligent boom control, 2025.

[14] Palfinger Official Webpage. Smart control, 2025.

Biography

Amy Rankka received the M.Sc. degree in electrical engineering in 2015 from Linköping University, Linköping, Sweden, where she is currently studying towards the Ph.D. degree in mechanical engineering with the division of fluid and mechatronic systems. She works as an R&D engineer at Hiab AB. Her research interests include energy efficient hydraulic systems.

Alessandro Dell'Amico has a MSc degree and a PhD degree in Mechanical Engineering from Linköping University. His research includes aircraft system design, energy management, actuation systems, and energy efficiency in fluid power systems. He currently holds a position as Adjunct Associate Professor at Linköping University.