

# Flow-Bounded Trajectory-Scaling Algorithm for Hydraulic Robotic Manipulators

Santeri Lampinen, Jouni Niemi and Jouni Mattila

**Abstract**—On-line methods for trajectory scaling have focused on torque or acceleration bounded minimum time trajectories, while other system constraints have received little attention. For hydraulic systems, volumetric flow rate of the supply unit establishes a critical constraint, that has been neglected in control design. Consequently, commercial solutions for robotic control of hydraulic manipulators are typically limited to a compromise of a slower constant endpoint velocity, that can be achieved in any operating point without violating the constrained flow rate. However, with real-time analysis of the required volumetric flow rate, the desired trajectories can be executed much faster without violating the flow rate constraint or losing control accuracy. This study proposes an on-line method for trajectory scaling to perform predetermined trajectories in minimum time. Essentially, the method scales velocity along the trajectory to maintain achievable velocity at all times. The proposed method is capable of enforcing a global volumetric flow limit, whether it is constant or time-varying. The method is validated with simulations and experiments with a real hydraulic robotic manipulator. Experimental results show a very significant improvement in the trajectory tracking control, where the tracking error is reduced from 461 mm to 73 mm on a square trajectory.

## I. INTRODUCTION

Hydraulic heavy-duty machinery is undergoing a revolution as these traditionally manually operated machines are being retrofitted with more advanced control systems or completely redesigned with robotic control in mind. Commercial solutions for retrofitting coordinated control for hydraulic manipulators are emerging [1], and manufacturers like Hiab, John Deere, Rambooms, etc. are beginning to implement advanced control in their products [2], [3]. These heavy-duty manipulators were designed first for manual operation, where the operator controls each hydraulic actuator separately. A skilled operator can drive multiple actuators simultaneously, and is able to automatically adjust the control in order to respect the limitation on the available flow rate. If the limitations are not considered, the least loaded actuator would receive the highest flow rate, and the most loaded actuators could not track the desired trajectory. This results in undesired motions, despite having fancy load sensing valves.

Limitations on the available flow rate, and consequently the obtainable tool center point (TCP) velocity is not something the current robotic control methods have taken into consideration. Traditional inverse kinematics solutions, such as those reviewed in [4], are not designed with limitations of the manipulators in mind, and thus, are not sufficient alone for applications with hydraulic machinery with limited volumetric flow.

At this point, it is worth considering a fundamental difference between electrical and hydraulic actuators. Electrical actuators can be viewed as torque sources, where the output torque is proportional to the control current. However, hydraulic actuators can be considered as velocity sources, with actuator velocity approximately proportional to the valve control. Due to this fundamental difference, a natural path of research with electric manipulators was to take the actuator limitations, namely the maximum obtainable torque, into consideration. Minimum-time control of robotic manipulators with torque-bounded trajectory generation was proposed in [5] to perform desired motions as fast as possible within the physical limitations of the controlled manipulator. Further studies [6]–[8] have carried this idea, and evolved the theory around torque-bounded time scaling on-line. In these studies, the trajectory is scaled in real-time, based on constraints on individual joint accelerations and joint velocity limits. However, the focus has been on individual limits for different actuators, without consideration of global limitations.

In view of the nature of hydraulic actuators, a natural approach with these actuators is velocity-bounded. With hydraulic manipulators, the maximum velocity of the manipulator is dictated by the available hydraulic flow from the supply unit. Moreover, while multiple actuators are driven simultaneously, the flow is divided among the actuators, and thus, joint velocity bounds change dynamically. Last, hydraulic systems contain nonlinearities from the closed chain mechanical structure, which may even be combined with a four-bar link to achieve a wider operation range.

In *closed-loop resolved rate control*, the typical solution is to limit the maximum velocity of the TCP to a level that can be achieved within the whole operating range. However, the manipulability (see [9, chapter 4]) can vary significantly in different manipulator configurations, and with hydraulic systems, this is even more notable than with electric manipulators. In hydraulic systems, cylinder velocities are used to define manipulability ellipsoids. The ellipsoids describe how well a manipulator can move its TCP in an arbitrary direction. A hydraulic manipulator can move much faster in some operating points and directions, where the manipulability is greater. Thus, this typical solution of choosing a constant maximum velocity is a mere compromise, and it leaves a huge potential of the manipulator unused. An experienced operator may complain in such case about sluggish movement and decreased efficiency, when compared to a manually operated machine.

To allow operation of the manipulator at its mechanical limits, we propose dynamic scaling of the TCP velocity, based on the available pump capacity. This allows the absolute

S. Lampinen, and J. Mattila are with the faculty of Engineering and Natural Sciences, Tampere University, Finland. `name.surname@tuni.fi`  
J. Niemi is with Rambooms Oy, Lahti, Finland.

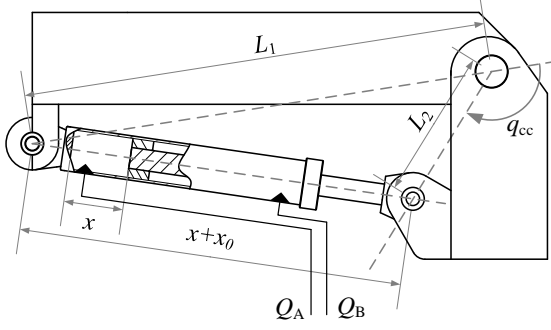


Fig. 1. Closed chain angles and dimensions.

maximum achievable TCP velocity with the mechanical and hydraulic limitations of the manipulator in the resolved rate control mode. A very similar idea was proposed in [10], where the inverse kinematics control problem was turned into an optimization problem by the means of quadratic programming. There, the method is not extended for trajectory scaling, however. Here, the proposed method can be applied to on-line time-scaling higher-level path planners to reduce too fast trajectories and can be implemented to act with torque-limiting controllers as proposed in [6]–[8]. The main result of this study is the introduction of trajectory scaling into the scope of hydraulic manipulators. In many cases, although not desired by design, the hydraulic flow of the supply unit is not sufficient in resolved rate control. The proposed approach solves the described problem and allows higher utilization of the manipulator's capabilities. Moreover, it is expandable to on-line time-scaling of the planned trajectory.

The remainder of this paper is organized as follows. Section II introduces the algorithm for flow-bounded trajectory scaling. Section III introduces the experimental system. Section IV provides results for the proposed control method in simulations and experiments. Section V concludes the paper.

## II. CONSTRAINT BOUNDED COORDINATED CONTROL

Let  $\mathbf{v} \in \mathbb{R}^3$  denote the desired velocity of the manipulator in world coordinates. For a redundant  $n$ -joint manipulator, the required joint velocities can be identified by using a pseudo-inverse of the Jacobian matrix as

$$\dot{\mathbf{q}} = \mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T)^{-1} \mathbf{v}, \quad (1)$$

where  $\dot{\mathbf{q}} \in \mathbb{R}^n$  is the required joint velocities,  $\mathbf{W} \in \mathbb{R}^{n \times n}$  is a symmetric positive definite weighing matrix, and  $\mathbf{J} \in \mathbb{R}^{n \times n}$  is the non-invertible Jacobian matrix [11].

Consider the kinematic closed chain structure of Fig. 1, that is part of the manipulator. To calculate the flow rates  $Q_A$  and  $Q_B$ , the cylinder velocity must be solved from the joint velocity  $\dot{q}_{cc}$  as

$$x = \sqrt{L_1^2 + L_2^2 + 2L_1L_2 \cos(q_{cc})} - x_0 \quad (2)$$

$$\dot{x} = -\frac{L_1L_2 \sin(q_{cc})}{x+x_0} \dot{q}_{cc}, \quad (3)$$

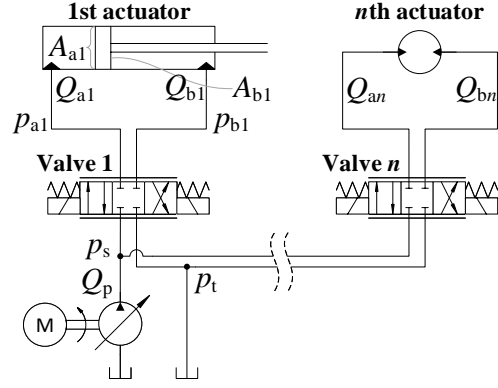


Fig. 2. A hydraulic system of  $n$  actuators.

where  $L_1$  and  $L_2$  are the structural lengths of the closed chain,  $x_0$  is the minimum length of the hydraulic cylinder,  $x$  is the displacement of the piston, and  $q_{cc}$  is the joint angle transformed into the closed chain space.

Then the flow rate into the cylinder can be written as

$$Q_{in} = \begin{cases} A_A \dot{x}, & \text{when } \dot{x} \geq 0 \\ -A_B \dot{x}, & \text{when } \dot{x} < 0, \end{cases} \quad (4)$$

where  $A_A$  and  $A_B$  are the areas on the A- and B-sides of the hydraulic cylinder, respectively.

In the case of a rotational joint actuated by a hydraulic motor, the flow into the actuator is calculated using

$$Q_{motor} = \frac{\dot{q} D_m}{2\pi R_t \eta_{vol}}, \quad (5)$$

where  $D_m$  is the volumetric displacement of the motor,  $R_t$  is the gear ratio of the transmission, and  $\eta_{vol}$  is the volumetric efficiency of the motor. Due to the high internal leakage of the hydraulic motor, volumetric efficiency of the hydraulic motor is needed for better accuracy.

In addition to the maximum flow constraint defined by the supply unit, the flow into the actuator can be saturated in the control valve. Moreover, the movement of the main spool of the control valve may be tuned independently for both directions to have similar behavior of the actuator for both directions, despite the asymmetric surface areas of the cylinder. In this case, the maximum flow rate to each direction is unique.

Consider a hydraulic system of  $n$  actuators and a hydraulic pump that outputs  $Q_p$  flow. The system is illustrated in Fig. 2. Algorithm 1 is defined to reduce the required joint velocities to satisfy constraints on individual and total flow limits, defined by the system capabilities. The algorithm can be used to output the scaling factor as well as the limited joint velocities. The scaling factor  $\dot{s}$  can be used with high-level path planners for time-scaling. Using the algorithm, the path planner can limit trajectories to physical limitations of the controlled manipulator, without the need for experimentally or arbitrarily set constant limits on joint or TCP velocity.  $Q_r$  in the algorithm denotes the volumetric flow required for joint velocities  $\dot{\mathbf{q}}$ .  $Q_{A \text{ limit}}$  and  $Q_{B \text{ limit}}$  denote the individual flow limits defined by the control valve for the A- and B-sides, respectively.

---

**Algorithm 1** Satisfying flow constraints

---

**Input:** joint velocities  $\dot{\mathbf{q}}$ , joint angles  $\mathbf{q}$ **Output:** limited joint velocities  $\dot{\mathbf{q}}_{\text{lim}}$  $Q_r \leftarrow 0$  $\dot{s} \leftarrow 1$ **for each**  $q \in \mathbf{q}$  **do**

determine the required flow into the actuator using (2)–(5)

 $Q_i \leftarrow$  required flow into actuator**if**  $\dot{x} > 0$  **then** $\dot{s} \leftarrow \min \left( \dot{s}, \frac{Q_{A \text{ limit}}}{Q_i} \right)$ **else if**  $\dot{x} < 0$  **then** $\dot{s} \leftarrow \min \left( \dot{s}, \frac{Q_{B \text{ limit}}}{Q_i} \right)$ **end if** $Q_r \leftarrow Q_r + Q_i$ **end for** $\dot{s} \leftarrow \min \left( \dot{s}, \frac{Q_p}{\dot{s} Q_r}, 1 \right)$  $\dot{\mathbf{q}}_{\text{lim}} \leftarrow \dot{s} \dot{\mathbf{q}}$ 

---

#### A. Application to Velocity Bounded Trajectory Control

Let  $f(s) \in \mathbb{R}^n$  be a continuous function defined in the interval  $[s_i, s_f]$  of the scalar path parameter  $s$ . The starting point of the path is defined as  $f(s_i)$ , while the end point is defined as  $f(s_f)$ . The individual joint positions and velocities of the manipulator, along the path, are obtained as

$$\mathbf{q} = f(s) \quad \dot{\mathbf{q}} = f'(s)\dot{s}. \quad (6)$$

Note that here it is assumed that the derivative of the path exists. However, this should not be a problem, as smooth trajectories should be self-evident in trajectory generation.

Without scaling, the path parameter  $s$  is typically time dependent and expressed as

$$s = at + b, \quad (7)$$

where  $a \in \mathbb{R}$  and  $b \in \mathbb{R}$  are constants (e.g., used to normalize  $s$  between 0 and 1).

In order to enforce the bound on the available volumetric flow from the pump, the derivative of the path parameter is obtained using Algorithm 1. Then  $s$  is obtained by integrating  $\dot{s}$  with  $s_i$  as the initial value.

### III. THE SYSTEM

The proposed algorithm is verified with simulations and experiments with a commercial rock breaking boom from a Finnish original equipment manufacturer, RamBooms. The rock breaking boom is from their commercial x8 lineup. Breaker booms are used in the mining industry for secondary breaking of blasted ores that are still too large for a crusher or feeder. Secondary crushing is performed using a hydraulic hammer attached to the tip of the breaker boom. Fig. 4 illustrates the hydraulic manipulator. The whole system weighs roughly 9400 kg while the hydraulic hammer alone weighs 1700 kg. The boom has a reach of 7-meters [12].

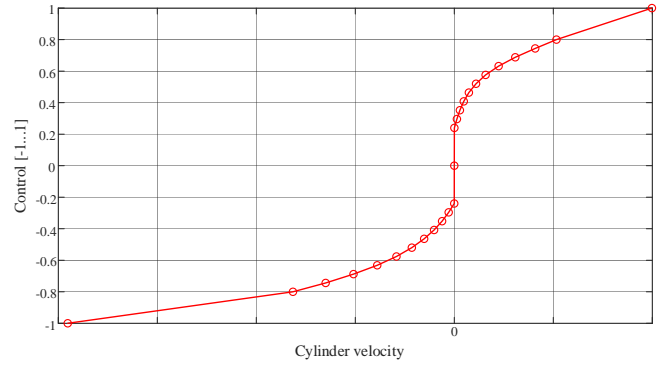


Fig. 3. Identified velocity feed-forward model for dual cylinder lift function.

The hydraulic supply unit of the manipulator produces a near constant hydraulic flow of  $180 \frac{l}{min}$  regardless of the loading conditions. Each actuator alone can draw almost all the flow from the supply, and thus, advanced control is needed for high-precision motion control.

The manipulator (see Fig. 4) is equipped with Siko WV58MR 14-bit absolute rotary encoders, with a resolution of  $21.97 \times 10^{-3}$  degrees, and Trafac 8270 pressure transducers. Control of the manipulator was achieved using dSPACE MicroAutoBox 2.

The motion control system used in the experiments is a simple P-controller combined with velocity feed-forward using an automatically learned velocity feed-forward model for each cylinder. The feed-forward model identification is based on the algorithm proposed in [13]. The algorithm identifies 24 distinct segments of the feed-forward model yielding good accuracy with reasonable converging time. The identified feed-forward model for the lift joint is presented in Fig. 3.

In the velocity feed-forward identification process in [13], an input-to-state stabilizable controller was designed with respect to an unknown disturbance input (parametric uncertainty). Then, an adaptive law was designed for estimation of the uncertain parameters of the feed-forward model. The adaptation law was designed using z-swapping [14] on the error system, to allow the use of standard adaptation methods. Then, a gradient update law was designed to estimate the uncertain parameters.

The manipulator is equipped with Danfoss PVG-120 mobile proportional control valves that have a significant (approximately 30 % per direction) dead-zone, thus making dead-zone inversion necessary in the control design. Moreover, it significantly improves the control accuracy. For more accurate control of the manipulator, model-based control methods seem the most promising option. In [15], a model-based controller was designed for the same manipulator. A very similar method has also been applied to a complete manipulator with fast servo valves, showing state-of-the-art performance in [16].

#### IV. THE EXPERIMENTS

Consider a quintic rest-to-rest trajectory created between two points using

$$x(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (8)$$

where coefficients  $a_i \in \mathbb{R}$  are obtained from

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \ddot{x}_0 \\ x_f \\ \dot{x}_f \\ \ddot{x}_f \end{bmatrix} \quad (9)$$

where  $t_0$  is the time at the beginning, and  $t_f$  is the time at the end;  $x_0$ ,  $\dot{x}_0$ , and  $\ddot{x}_0$  denote the initial position, velocity, and acceleration, respectively; while  $x_f$ ,  $\dot{x}_f$ , and  $\ddot{x}_f$  define the final position, velocity, and acceleration, respectively [17].

The trajectory used in the experiments is created with the method above, and the time for each segment is set to 4 s. In the experiments, time  $t$  is replaced with  $s$ , which is obtained by integrating  $\dot{s}$ , as suggested by theory.

Due to the limited operation space of the manipulator and the authors' desire for a relatively large trajectory in the experiments, the pseudo-inverse of the manipulator brought problems with joint limits. To solve this issue, the weight matrix  $W$  in (1) was redesigned to address this issue. Our solution was to model the individual weights as a function of the joint angle. The weights for each joint were calculated by using two sigmoid functions as

$$W_i = 1 + \frac{w}{1 + e^{\lambda(q_i - q_{i \min})}} + \frac{w}{1 + e^{\lambda(q_i - q_{i \max})}}, \quad (10)$$

where  $w \in \mathbb{R}$  denotes the maximum weight near joint bounds,  $\lambda \in \mathbb{R}$  defines the slope of the weight function, and  $q_i$ ,  $q_{i \min}$ , and  $q_{i \max}$  denote the current value, lower bound and the upper bound of the  $i$ th joint, respectively. The values  $W_i$  represent diagonal elements of  $W$  in (1).

The use of these weight functions alone to penalize joint usage near the ends is not enough, however. When a joint is driven near its end, the weight begins to restrict the joint's



Fig. 4. Rambooms X88-540R. [12]

movement even when trying to drive it away from the end. To address this issue, the restrictive weights are applied only to a joint that is moving toward the limit the joint is closer to. If a joint is moving away from its closer bound, the weight is set to equal 1. Consequently, only movement toward the cylinder ends is penalized.

Another approach for enforcing joint limits in the inverse kinematics problem is to form the inverse kinematics control problem into a constraint optimization problem, such as a quadratic programming problem. This approach was used in [10]. Moreover, the redundancy resolution could be exploited using null space control as discussed in [11, Chapter 3.5.1] to avoid the joint limits.

##### A. Simulation Study

The proposed method to limit the velocity of the tip in coordinated control was first validated with simulations. The simulation was based on kinematic modeling of the boom and the dynamics of the manipulator were neglected. The required flow and the actual flow from the supply unit were calculated based on the required and limited joint velocities, respectively.

In the simulations, the manipulator was driven in a square path in four segments, with 4 s travel time between segments. Each segment began from one corner and ended in another. Between the segments, there was a 0.5 s rest time. In the experiments, the flow constraint was set to  $120 \frac{l}{min}$ . The trajectory used in the simulation study and the actual experiment is presented in Fig. 5.

Results from the simulations are shown in Fig. 6. The trajectory is shown in Fig. 7, where the y-axis points away from the base of the manipulator and the z-axis points directly up (see Fig. 5). In the first simulation, the limiting algorithm was disabled, and the maximum flow required during the trajectory was almost  $300 \frac{l}{min}$ , and the time to complete the trajectory was approximately 18 s. In the second simulation, the proposed algorithm was enabled, and

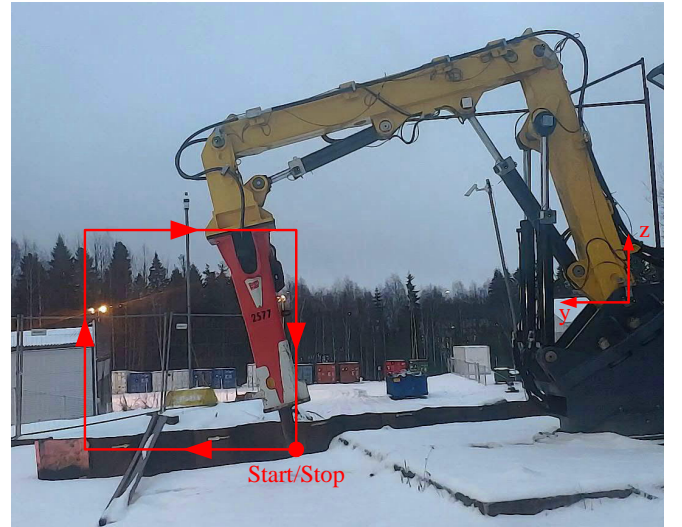


Fig. 5. Trajectory of the experiments.

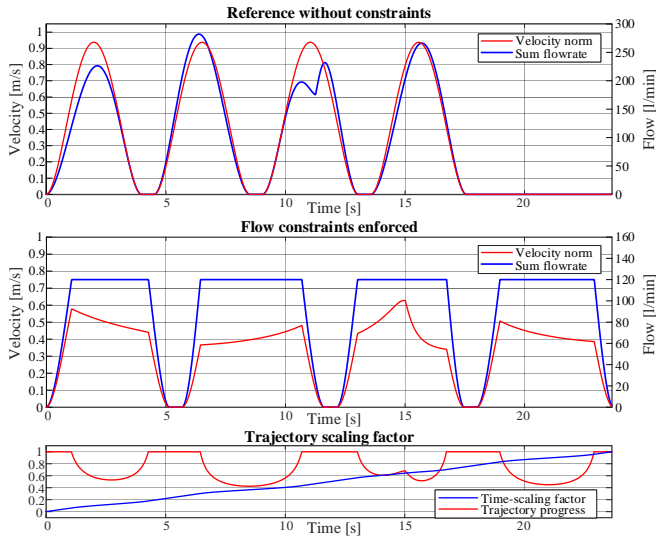


Fig. 6. Simulation results with the proposed algorithm.

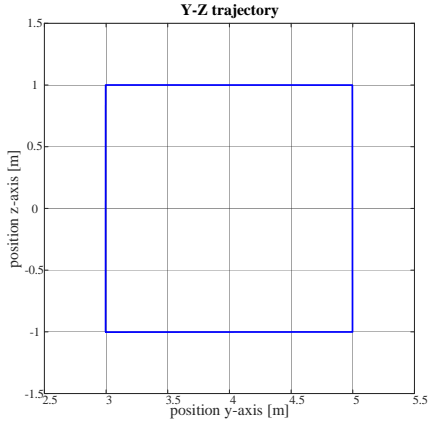


Fig. 7. Simulation trajectory.

the maximum observed flow rate remained at  $120 \frac{l}{min}$ . The manipulator completed the set trajectory in 24 s with the algorithm. Moreover, it can be seen that the velocity was limited most on the second segment of the trajectory. This result was expected, as in the second segment of the trajectory, the boom is in such a configuration that most of the movement is done with the first joint of the manipulator, which has two parallel cylinders (see Fig. 4).

### B. Experiments with a Real Machine

In the experiment with the real manipulator, the driven trajectory was identical to the one used in the simulation study. Again, the experiment was repeated twice, once with the algorithm in use and once without any trajectory scaling. The maximum flow of  $120 \frac{l}{min}$  was selected for the experiments mainly to highlight the operation of the scaling algorithm and the benefits it brings.

Results from the experiment with the actual manipulator are shown in Figs. 8 and 9. The velocity norm of the TCP and the volumetric flow rate from the reference measurement without trajectory scaling is shown at the top of Fig. 8, while the same data from the measurement with the algorithm in

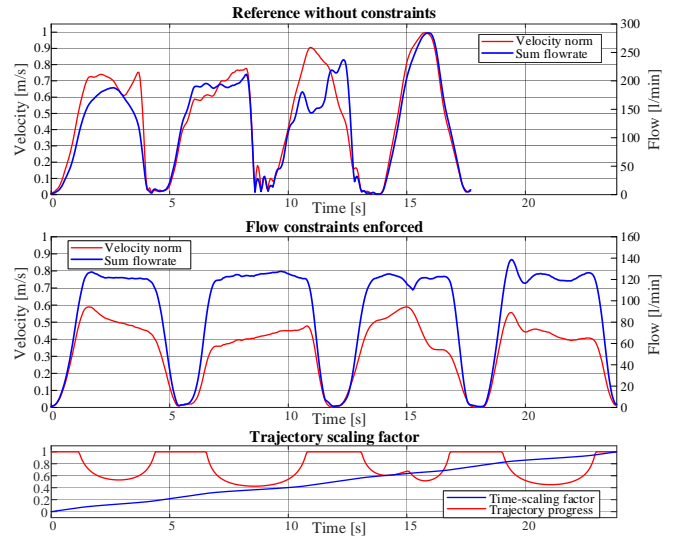


Fig. 8. Experimental results with the proposed algorithm.

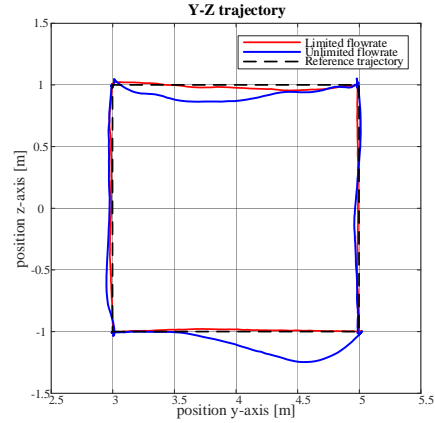


Fig. 9. Experimental trajectory tracking.

use is shown in the middle. The trajectory scaling factor,  $\dot{s}$ , is shown at the bottom of the same figure along with the normalized trajectory parameter,  $s$ . The measured TCP positions of both experiments are shown in Fig. 9.

Figure 10 presents the Cartesian trajectory tracking error from both experiments. From the figure, it can easily be seen that without the algorithm, errors in trajectory tracking grow notably, as the demanded flow rate exceeds the available capacity. As a result, the manipulator deviates almost half a meter from the desired trajectory. In contrast, results with the algorithm in use reveal greatly improved tracking although no other parameters of the control system were changed. The manipulator still deviates a little from the trajectory, but this can be expected from such a simple controller. The maximum tracking error was reduced from 461 mm of the measurement without the algorithm to 73 mm with the algorithm. The calculated flow rate going to the actuators is clearly limited compared to the reference measurement, and comparable to the respective plot from the simulation study. The small deviations from the desired velocities of the actuators cause the disturbances in the flow rate.



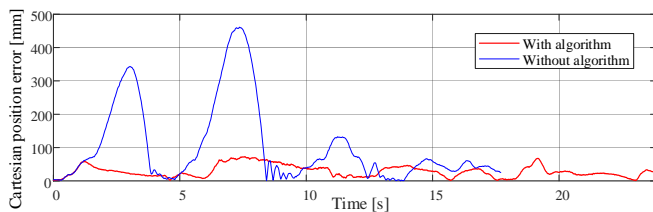


Fig. 10. Trajectory tracking error.

## V. CONCLUSIONS

In this study, we designed and demonstrated an algorithm for trajectory and motion scaling for hydraulic systems. Although this paper is formed around hydraulic manipulators, the algorithm is general and can be implemented on different kinds of hydraulic systems as well (e.g., the drive transmission of a hydraulic platform). The algorithm is compatible with motion scaling when the manipulator is operated manually in the resolved rate control mode. More importantly, the algorithm can also be used for trajectory scaling. In this application, a predetermined trajectory can be slowed down when necessary, to satisfy constraints on the available pump flow. Moreover, this approach is compatible with the torque-bounded control methods proposed in [6].

The proposed method was demonstrated with simulations and experiments on a hydraulic breaker boom system with promising results. The simulation study shows clearly the working principle of the algorithm in ideal conditions and shows the effectiveness of the method. The experiments with the actual machine show that the algorithm can effectively scale a trajectory on-line to a level achievable with the manipulator and greatly improve the tracking performance of the manipulator. The maximum tracking error during the experiments was reduced from 461 mm to 73 mm when the proposed algorithm was in use.

It should be noted that the controller in the experiment relied heavily on learned feed-forward mapping that is complemented only with a proportional controller. With a more sophisticated control design, like those proposed in [15], [16], the experimental results would converge closer to the simulation results, and the tracking error could be even further reduced. For comparison, experiments without the algorithm were performed as well, and the results showed the results of insufficient available volumetric flow. When all of the provided flow is being used, the least loaded actuators receive most of the flow, leaving the other actuators unable to track their respective trajectories. The division of flow is analogous to Kirchhoff's law in electronics.

The proposed method can be applied to a wide range of hydraulic systems to improve their performance. With

the proposed algorithm, hydraulic systems can be utilized at higher capacity, as the maximum velocity for movements does not need to be such that can be achieved at any configuration.

## ACKNOWLEDGMENT

The authors wish to acknowledge support from the Finnish Foundation for Technology Promotion.

## REFERENCES

- [1] Technion, "Technion xcrane crane control system." Available at: <https://technion.fi/wp-content/uploads/xcrane-brochure-en.pdf> (accessed 3.6.2019).
- [2] Hiab, "Ctc - crane tip control." Available at: <http://www.podshop.se/content/12/opensearchresult.aspx?file=PF-CTC-EN-EU.L.pdf> (accessed 3.6.2019).
- [3] J. Deere, "Intelligent boom control ibc." Available at: <https://www.deere.co.uk/en/forestry/ibc/> (accessed 13.6.2019).
- [4] D. N. Nenchev, "Redundancy resolution through local optimization: A review," *Journal of Robotic Systems*, vol. 6, no. 6, pp. 769–798, 1989.
- [5] Kang Shin and N. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. 30, pp. 531–541, June 1985.
- [6] O. Dahl and L. Nielsen, "Torque-limited path following by online trajectory time scaling," *IEEE Transactions on Robotics and Automation*, vol. 6, pp. 554–561, Oct 1990.
- [7] O. Dahl, "Path-constrained robot control with limited torques-experimental evaluation," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 658–669, Oct 1994.
- [8] L. Zlajpah, "On time optimal path control of manipulators with bounded joint velocities and torques," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1572–1577 vol.2, April 1996.
- [9] T. Yoshikawa, *Foundations of Robotics : Analysis and Control*. The MIT Press, 1990.
- [10] J. Wanner and O. Sawodny, "Tool-center-point control of a flexible link concrete pump with hydraulic limitations using quadratic programming," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pp. 561–566, Aug 2019.
- [11] L. Sciacivco and B. Siciliano, *Modelling and Control of Robot Manipulators*. Advanced Textbooks in Control and Signal Processing, Springer-Verlag London, 2 ed., 2012.
- [12] RamBooms, "Rambooms X88-540R rock breaker boom." Available at: <https://mrbbbooms.com/x-series/x8/x88-540r/> (accessed 3.4.2019).
- [13] J. Nurmi and J. Mattila, "Automated feed-forward learning for pressure-compensated mobile hydraulic valves with significant dead-zone," in *Proc. of ASME/BATH Symp. on Fluid Power and Motion Control (FPMC2017)*, Oct. 16-19, 2017, Sarasota, Florida, 2017.
- [14] M. Krstić, I. Kanellakopoulos, and P. Kokotović, *Nonlinear and Adaptive Control Design*. John Wiley & Sons, Inc., 1995.
- [15] S. Lampinen, J. Koivumäki, J. Mattila, and J. Niemi, "Model-based control of a pressure-compensated directional valve with significant dead-zone," in *ASME/BATH 2019 Symposium on Fluid Power and Motion Control*, American Society of Mechanical Engineers Digital Collection, 2019.
- [16] J. Koivumäki and J. Mattila, "Stability-guaranteed force-sensorless contact force/motion control of heavy-duty hydraulic manipulators," *IEEE Trans. Robot.*, vol. 31, pp. 918–935, Aug 2015.
- [17] R. N. Jazar, *Theory of Applied Robotics: Kinematics, Dynamics, and Control*. Springer Publishing Company, 2nd ed., 2010.