Biomimetic Intelligence and Robotics 3 (2023) 100133

Contents lists available at ScienceDirect

Biomimetic Intelligence and Robotics

journal homepage: www.elsevier.com/locate/birob

Research Article

A framework to develop and test a model-free motion control system for a forestry crane

Pedro La Hera [a,*], Omar Mendoza-Trejo [a], Håkan Lideskog [b], Daniel Ortíz Morales [c]

[a] Swedish University of Agricultural Sciences, Department of Forest Biomaterials and Technology, Umeå 90183, Sweden
[b] Luleå University of Technology, Department of Engineering Sciences and Mathematics, Luleå 97187, Sweden
[c] CRANAB AB, Vindeln 92232, Sweden

ARTICLE INFO

Article history:
Received 25 August 2023
Revised 24 October 2023
Accepted 30 October 2023
Available online 10 November 2023

Keywords:
Model-free control
Hydraulic manipulator
Forestry crane control
Intelligent PID control
Forestry automation
Controller implementation

ABSTRACT

This article has the objective of presenting our method to develop and test a motion control system for a heavy-duty hydraulically actuated manipulator, which is part of a newly developed prototype featuring a fully-autonomous unmanned forestry machine. This control algorithm is based on functional analysis and differential algebra, under the concepts of a new type of approach known as model-free intelligent PID control (iPID). As it can be unsafe to test this form of control directly on real hardware, our main contribution is to introduce a framework for developing and testing control software. This framework incorporates a desktop-size mockup crane equipped with comparable hardware as the real one, which we design and manufactured using 3D-printing. This downscaled mechatronic system allows to safely test the implementation of control software in real-time hardware directly on our desks, prior to the actual testing on the real machine. The results demonstrate that this development framework is useful to safely test control software for heavy-duty systems, and it helped us present the first experiments with the world's first unmanned forestry machine capable of performing fully autonomous forestry tasks.

© 2023 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

This article presents the steps we followed to develop, implement, and test a motion control algorithm for automating the motions of a hydraulically-actuated heavy-duty forestry crane. This crane is part of the world's first unmanned forestry machine capable of performing fully autonomous forwarding tasks, i.e. extracting logs out of the forest without the need for human involvement [1]. This machine has been developed to be a research platform for testing the technology needed to automate forest operation tasks. The purpose of the controlled crane movements is to perform pick-and-place actions during these tasks. Our control algorithm centers around recent advancements in model-free intelligent PID controllers (iPID), a method that can be formulated without the need of the system's dynamics model, and provides robustness, disturbance rejection and adaptive properties [2,3].

1.1. Overview

In Fennoscandia, electro-hydraulically actuated cranes are an essential component of forestry machinery, providing efficient tree cutting and heavy log maneuvering, when performing forest operations. Over the years, the technology to manually operate these cranes has advanced from old-fashioned open-loop joint-by-joint control to the use of semi-automated solutions, such as the most recent Cartesian end-effector control [4,5]. To this end, the new innovations from crane manufacturers involve incorporating motion sensors on cranes' joints [6,7], and pressure sensors in hydraulic valves [8], giving place to the introduction of feedback motion control software. As controlling the crane forms the largest portion of the work for machine operators, using feedback control systems has positively influenced the way forestry operators experience working with cranes. This has generated the interest to move to a new era of development, where forestry machines gain the benefits of automation and robotics [9].

The concept of unmanned and automated forestry machines has long been contemplated in the Fennoscandian region [10], driven by various factors such as environmental impact reduction, weight and manufacturing cost reduction, operator comfort regarding vibration exposure, and increased productivity in forest operations [9]. In these regard, several prototypes consisting of radio-controlled machines have been developed over the years, such as BESTEN (eng. The Beast) and the eBeaver [11,12]. However, these machines have not succeeded to reach market production because they do not meet industry expectations in terms of productivity or simplification of work tasks for machine

* Corresponding author.
E-mail address: xavier.lahera@slu.se (P. La Hera).

https://doi.org/10.1016/j.birob.2023.100133
2667-3797/© 2023 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Fig. 1. AORO machine platform. In this scenario, the AORO machine was equipped with a trailer to emulate a forwarder.

operators. On the contrary, employing conventional joint-by-joint open-loop control commands for remote operation of these machines has been shown to intensify the inherent complexity of manual control. Nevertheless, unmanned machines with varying degrees of automation offer solutions to address challenges in forestry, including the shortage of qualified operators, environmental concerns, and high operating costs. For this reason, the transition from manual to full automation has been an ongoing area of research [13,14].

Our group, the Arctic Off-Road Robotics Lab (AORO) [1], has developed a heavy-duty unmanned forestry machine to serve as a dedicated testing platform for forestry automation research (see Fig. 1). The objective is to pave the way for the development of unmanned forestry machines and test a variety of automation software that has been developed over the years to achieve fully-autonomous operations, focusing initially on forwarding tasks. Forwarding involves the task of collecting logs that have been felled by a harvester machine. The forwarder machine transports these logs out of the forest into a road-side area for collection. To this end, the machine is equipped with an electro-hydraulically actuated crane for handling the logs and a trailer for transporting them. Referring to Fig. 1, the crane in our platform is an off-the-shelf component, implying that it is not a customized solution, highlighting our aim to develop solutions that can be readily implemented in the industry.

1.2. Overview of control systems in forestry

Since the performance of autonomous robots requires that the system becomes capable of executing meaningful and coordinated motions, closed-loop motion control algorithms are at the core of the functionality of these systems [15]. In this regard, numerous control methods, including linear and nonlinear model-based approaches, have been proposed for hydraulic cranes, as evidenced in previous studies [16–18] and related references. However, it is worth noting that the majority of these methods have not undergone comprehensive testing in commercial cranes or real-world operating conditions. The lack of follow-up on these developments highlights both the interest and challenges involved in automating forestry cranes.

Our research group has been involved in the development of automation of forestry cranes for nearly two decades. Initially, we explored the feasibility of using different approaches based on model-based nonlinear control algorithms, as this is an standard approach in the control of industrial robotic manipulators [17,19]. However, we found that one of the drawbacks of such methods is the need for dynamics models, which proved to be time consuming, challenging, costly and thus impractical to transition our development to our industrial partners. Therefore, our focus shifted towards researching on model-free control alternatives, inspired on the successful performance of such controllers in a variety of applications in systems presenting complex nonlinear and time-varying dynamics [2,20,21].

1.3. Problem formulation

The problem we are trying to solve is the development and testing of a model-free control algorithm for automating the motions of hydraulically-actuated heavy-duty cranes. The aim is to overcome the challenges associated with obtaining accurate dynamics models, which demand time and resources, rendering it impractical for the forest industry, which is recently starting to adopt automation technology. To address this issue, we are proposing to use model-free intelligent PID (iPID) controllers, as an option that offers to eliminate the need for explicit system models, facilitating the integration of control algorithms into forestry machines [22]. Model-free intelligent PID control uses concepts involving ultra-local models, functional analysis and differential algebra to provide a control algorithm that has shown better performance than classical PID control and has already a quite impressive list of successful concrete applications in most diverse fields, ranging from intelligent transportation systems, robot manipulators, to energy management [2,3,20].

As iPID controllers rely mainly on real-time input and output signals to calculate the control input, we require a safe and cost efficient method to test its performance with hardware during the development process. However, performing direct experimentation on the system can be both dangerous and expensive, specially in the early stages of software development. To mitigate these challenges, we are proposing to use a 3D-printed replica of the crane equipped with hardware resembling the real system, i.e. sensors and microprocessors, but with electrical actuators. This approach enables rapid-prototyping and real-time hardware performance evaluation directly on our desks.

A significant challenge arises from the inherent differences in dynamics between the real crane and its replica. However, both cranes operate on the same principle, where applying a voltage input results in the motion of a specific link or degree-of-freedom. Therefore, while the cranes exhibits differences in motion characteristics, such as variations in range of motion, speed, and acceleration, the underlying principle of translating voltage inputs to link motions remains constant. This shared operational principle forms the basis for the use of the iPID controller and the transferability of the controller from one crane to another. It is relevant to verify this assertion, particularly considering our intention to apply this control algorithm and framework of testing for most of our development.

Achieving this form of development framework, where the controller developed for the crane replica behaves nearly similar on the real one, becomes a rapid-prototyping and safe testing method that is time and cost-effective during the software development cycle. Furthermore, this framework enhances our ability to present compelling results to our industry partners, before deciding to move into tests with commercial machines.

The subsequent sections of this article showcase this development framework and present the results we have obtained using it.

2. Material and methods

2.1. Development framework and experimental setups

Our framework of testing involves a two step process of development. In the first stage, we use a desktop sized mockup of the forestry crane, which we have equipped with comparable

Fig. 2. Desktop size mockup crane. Top figure shows the CAD model design. Bottom figure shows the 3D printed model including the electrical actuators.

hardware to the real crane and has equal number of degrees-of-freedom. This system helps verifying the implementation and real-time functionality of control algorithms developed in software. The software we use for implementation is Simulink, which is part of MATLAB [23]. Consequently, once the tests in this mockup setup meet our desired specifications, in terms of trajectory tracking performance and robustness, the algorithms are then transported to the commercial size crane. This framework allows to spot early mistakes in the implementation of real-time software, plan the process of tuning the control algorithm parameters, and avoid outcomes that can be harmful to the machine when testing software with these large heavy-duty cranes.

To facilitate our development, we have implemented a normalization for the control input, to constrain the input signal generated by the control algorithm to the range $[-1, 1]$. The scaling of this input signal to the voltage required for the respective degree-of-freedom is consequently done for each crane in accordance to the maximum and minimum voltage they allow by hardware. This normalization facilitates the transportability of the algorithm from one crane to another.

2.1.1. The 3D printed mockup crane

Referring to Fig. 2, the mockup system is a 3D printed model that we designed using CAD software. This crane was designed as a system that can replicate the amount of degrees-of-freedom (DOF) and principles of motions observed in most forestry cranes. However, the mockup system does not necessarily present identical architecture as the commercial crane.

This crane uses micro linear actuators to emulate the cylinders that provide the forces that provoke motion. These actuators are L12-P Micro Linear Actuators with Position Feedback from Actuonix [24]. The position feedback is given as an analogue value between 0 to 5 V. To run software, this system has an 8-bit Arduino MEGA microcontroller as the main processing unit. The Arduino Mega has analogue to digital converters with a resolution of 10 bits to read analog sensors [25], as well as output ports with PWM for driving external analog devices. To drive the actuators, this system uses an external H-bridge amplifier to amplify the control signals from the arduino's output ports from 5 to 12 V and up to 2 Amp per channel.

To program this system, the algorithms are developed in MATLAB, in particular Simulink, and they are directly compiled to run in real-time using Simulink Coder [26].

2.1.2. The commercial size forwarder crane

The AORO machine uses a model FC8 crane from the company CRANAB [7], a crane with a maximum reach of 10 meters and weighting 1.4 Tons. This is a four degrees-of-freedom hydraulically actuated manipulator that follows a RRRP (R = revolute, P = prismatic) configuration, according to robotics nomenclature [15]. Referring to Fig. 3, these DOF are specified as the slewing q_1, inner boom q_2, outer boom q_3, and telescope q_4. The end-effector for grabbing logs is attached at the boom-tip, model CR250 grapple, having two active degrees-of-freedom for orientation q_5 and grabbing q_6. However, the grapple is an under-actuated tool, as it is freely hanging at the joint connecting to the boom-tip.

This crane belongs to a new line of products from CRANAB with the special feature of having analogue encoders as joint position sensors built-in within its mechanical structure. In addition, the electro-hydraulic valve has a set of pressure sensors that can measure each cylinder chamber's pressure.

To run software, the machine has a UEISIM real-time computer from United Electronics running a light-weight Linux OS [27]. It also has several I/O cards installed for sensor inputs and control outputs. To program this system, the algorithms are developed in MATLAB/Simulink and compiled to run in real-time on the UEISIM target using Simulink Coder [26]. As explained above, these algorithms are those initially tested in the mockup crane, but reconfigured to the UEISIM's I/O channels.

Referring to Fig. 4, all crane sensors are connected to the 12-bit DAC from the main UEISIM unit. To provide control signals to the hydraulic valve, the UEISIM unit output port is connected to a power amplifier board.

2.2. Control algorithm

Our work is centered around the development of a decentralized control system based on model-free intelligent PID control (iPID) [3]. iPID is a relatively new method that resembles active disturbance rejection controllers [28], with the difference that it does not require a model of the system to achieve its desired objective. Instead, it uses an ultra-local model, formulated as

$$y^{(\nu)} = \phi + \alpha \cdot u \tag{1}$$

where $y^{(\nu)}$ is the derivative order $\nu \geq 1$ of y chosen by the practitioner, and $\alpha \in \mathbb{R}$ is a non-physical constant parameter making the magnitudes of $\alpha \cdot u$ and $y^{(\nu)}$ equal. Most successful applications use $\nu = 1$, and very few examples require $\nu = 2$ [3,29–31]. In our case, our controller design uses $\nu = 1$, as we have not found any benefits from using higher order.

The terminology "ultra-local model" points to the concept that any nonlinear system behaves linearly in small segments of time or time-windows. Thus, the linear model (1) can replace a nonlinear model as a form of linearization, but it is valid only locally for a very short time-window (e.g. a couple of milliseconds depending on the sample time of the control-loop). As a

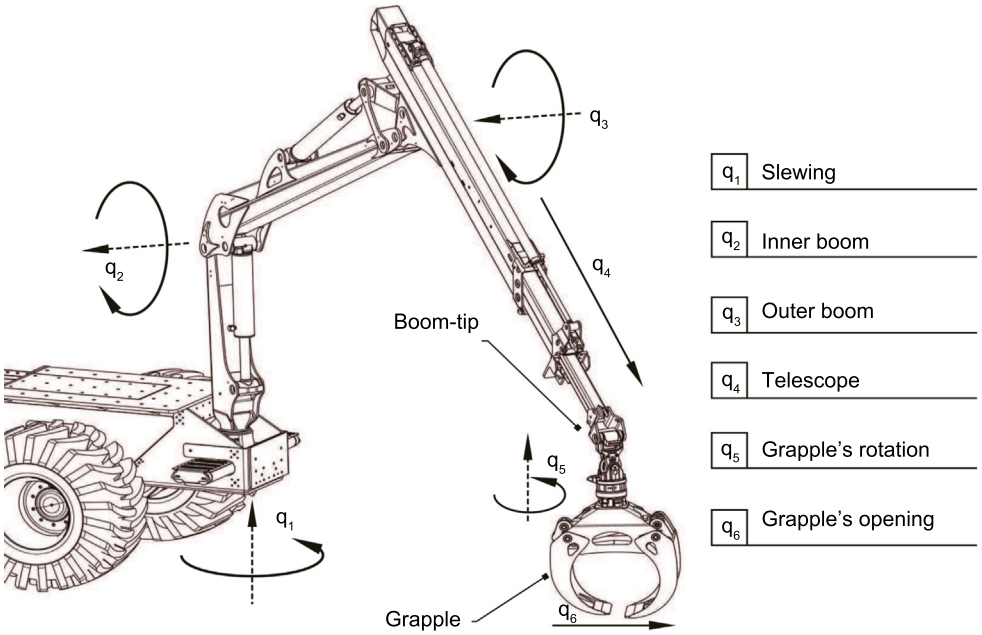

Fig. 3. Forwarder crane: hydraulic manipulator with four degrees-of-freedom, specified in this graph as the slewing q_1, inner boom q_2, outer boom q_3, and telescope q_4. It holds an end-effector attached at the boom-tip, serving as a tool to grab logs. It is known as the grapple, having two active degrees-of-freedom, specified as q_5 for rotation, and q_6 for its opening. All sensors measure positive in counter-clockwise direction.

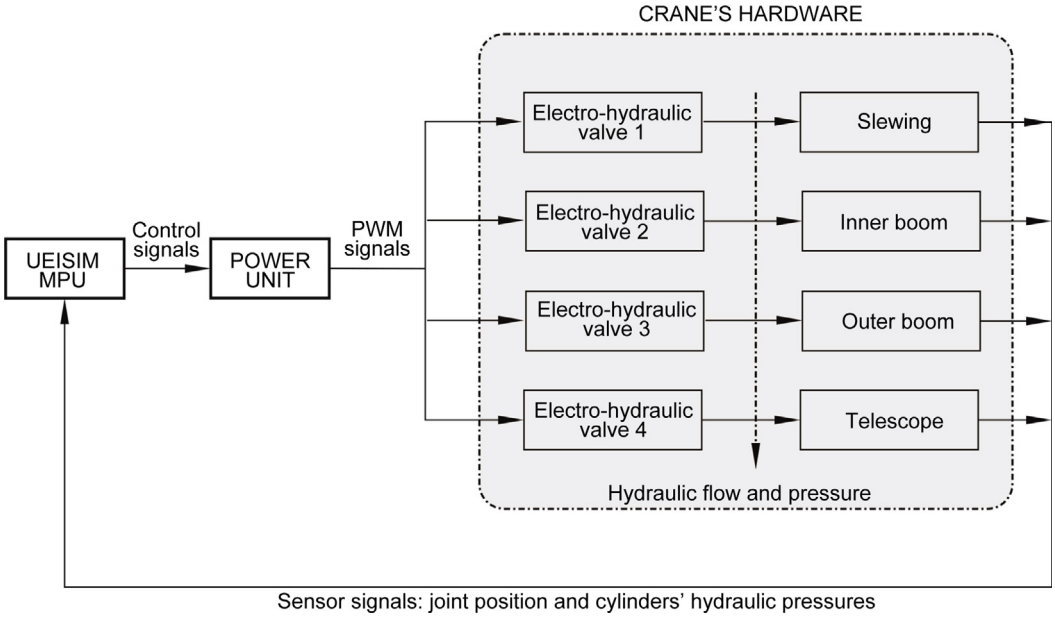

Fig. 4. Hardware architecture used as part of the crane's control system.

consequence, the values for ϕ have to be updated for every time-window, making the controller adaptive and with outstanding properties of robustness and disturbance rejection.

To develop a feedback controller based on (1), the control law is defined as

$$u = \frac{1}{\alpha}\left(v - \hat{\phi}\right) \tag{2}$$

resembling a feedback linearization action that leads to the degree-one integrator $\dot{y} = v$, if the estimation $\hat{\phi}$ is ideally equal to ϕ. Thus, this integrator can be stabilized via feedback by the term

$$v = \dot{y}^* - C(e), \quad e = y - y^* \tag{3}$$

where $C(e)$ is a standard Proportional-Integral-Derivative controller, and $\dot{y}^*$ is the first derivative of the reference trajectory.

To estimate ϕ online, the authors of [3] suggests using an integral over a time-window of fixed length, because integrals are the lowest order low pass filters. This estimation is formulated as:

$$\hat{\phi} = \frac{1}{\delta}\int_{t-\delta}^{t}\left(\dot{y}^* - \alpha \cdot u - C(e)\right)d\rho, \quad u = u(t-1) \tag{4}$$

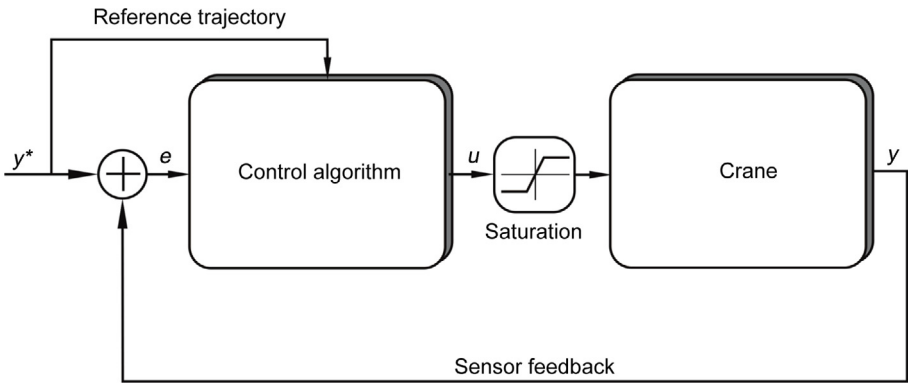

Fig. 5. General feedback control loop diagram.

Thus, the key features provided by the model-free Intelligent PID Control (iPID) algorithm (2)–(4) include:

- **Ultra-Local Models**: iPID relies on ultra-local models, e.g. (1), often referred to as ''ultradifferentiable models''. These models capture the local behavior of the system around its current operating point. Instead of tuning fixed PID parameters, iPID continuously estimates the local model based on real-time system data.
- **Adaptive Control**: iPID dynamically adjusts its control actions based on the ultra-local model, which is continuously estimated online. It does not focus on tuning PID gains in the conventional sense but rather adapts its control efforts to suit the system's observed behavior. This adaptability allows iPID to maintain control performance even as the system dynamics change.
- **Nonlinearity Handling**: One of the key advantages of iPID is its ability to handle nonlinear systems effectively. By continuously updating its understanding of the local system behavior, iPID can provide control actions that are better suited to nonlinearities, without relying on fixed PID parameters.

In summary, iPID's intelligence comes from its capacity to adapt and respond to a system's behavior based on ultra-local models, rather than relying on pre-tuned PID parameters. This approach is particularly useful for systems with nonlinear dynamics and time-varying characteristics, as it can provide robust and effective control without the need for manual adaptation of the PID gains.

2.2.1. Controller implementation

In our development, we use Simulink, a versatile software tool developed by MathWorks, as a central element for both control algorithm development and real-time hardware implementation [23]. Simulink offers a user-friendly graphical interface that serves as our primary platform for programming control algorithms with great efficiency. To enable the seamless deployment of these algorithms on the real-time hardware platforms integrated into our cranes, we take advantage of Simulink's automated code generation feature, typically resulting in C or C++ code [26]. What is important to mention is that Simulink offers us the capacity to develop control algorithms in a generalized fashion, allowing us to create a single algorithm that can be compiled for a variety of hardware platforms without the need for extensive modifications. Therefore, the control algorithm developed for the mockup crane can be directly compiled to work with the AORO crane, simply by changing the hardware settings needed for the compilation, and the control parameters that correspond to each case. This methodology greatly streamlines our processes for rapid prototyping and validation, ensuring the effectiveness of our research efforts. The following paragraphs provide a description of our implementation of the control algorithm (2) in Simulink.

Fig. 6(a) shows our general MATLAB/Simulink implementation of the controller described by Eqs. (2) and (4). This implementation corresponds to the content inside the "control algorithm" block in Fig. 5, representing a generic feedback control loop. The controller consists of two primary components: a feedforward compensation and the model-free-controller. The feedforward compensation block is responsible for providing the valve's input values that provoke the crane to start moving. This is related to Coulomb friction and it is implemented as follows:

$$u_{fc} = f_c \cdot \tanh(\beta \cdot \dot{y}) + \Delta f_c \tag{5}$$

representing an asymmetrical Coulomb friction model [16], having f_c as the averaged Coulomb friction value, Δf_c is the offset from the average, and $\tanh(\beta \cdot \dot{y})$ is a smooth function that approximates the most common signum function sign$(\cdot)$ to avoid chattering around zero velocity. Although many studies show that the control law based on iPID is capable of compensating for complex friction phenomena [3], throughout our development we found important to provide the estimation of Coulomb friction. This helps to increase the controller's tracking performance, as these values are substantial in hydraulic systems, accounting for nearly 40% of the maximum allowable control action for our commercial crane. Fig. 6(b) shows an implementation for the model-free controller as the inner implementation of the block observed in 6(a). The implementation of $C(e)$ is done using a standard discrete-time PID controller block from the Simulink Library.

Fig. 7 shows one example for estimating $\hat{\phi}$ using a time-windowed integration. The integral is numerically computed using the Trapezoidal rule of N equally spaced values. To this end, we use a MATLAB/Simulink Discrete FIR filter block with the specific vector of coefficients "[1 2*ones(1, n-1) 1]*Ts/2" in MATLAB code. This defines a numerical approximation of the integral according to

$$\int_a^b f(x)dx \approx \frac{T_s}{2}\sum_{k=1}^{N}(f(x_{k+1}) + f(x_k)) \tag{6}$$

$$\approx \frac{b-a}{2N}(f(x_1) + 2f(x_1) + \cdots + 2f(x_N) + f(x_{N+1})) \tag{7}$$

where the summation can be implemented with a FIR filter with vector coefficients [1 2 $\cdots$ 2 1], and Ts is the sample-time. The user sets the value n to define the integration length. Then, the value of delta $= \delta$ is equivalent to $\delta = n \cdot Ts$. Observe that to

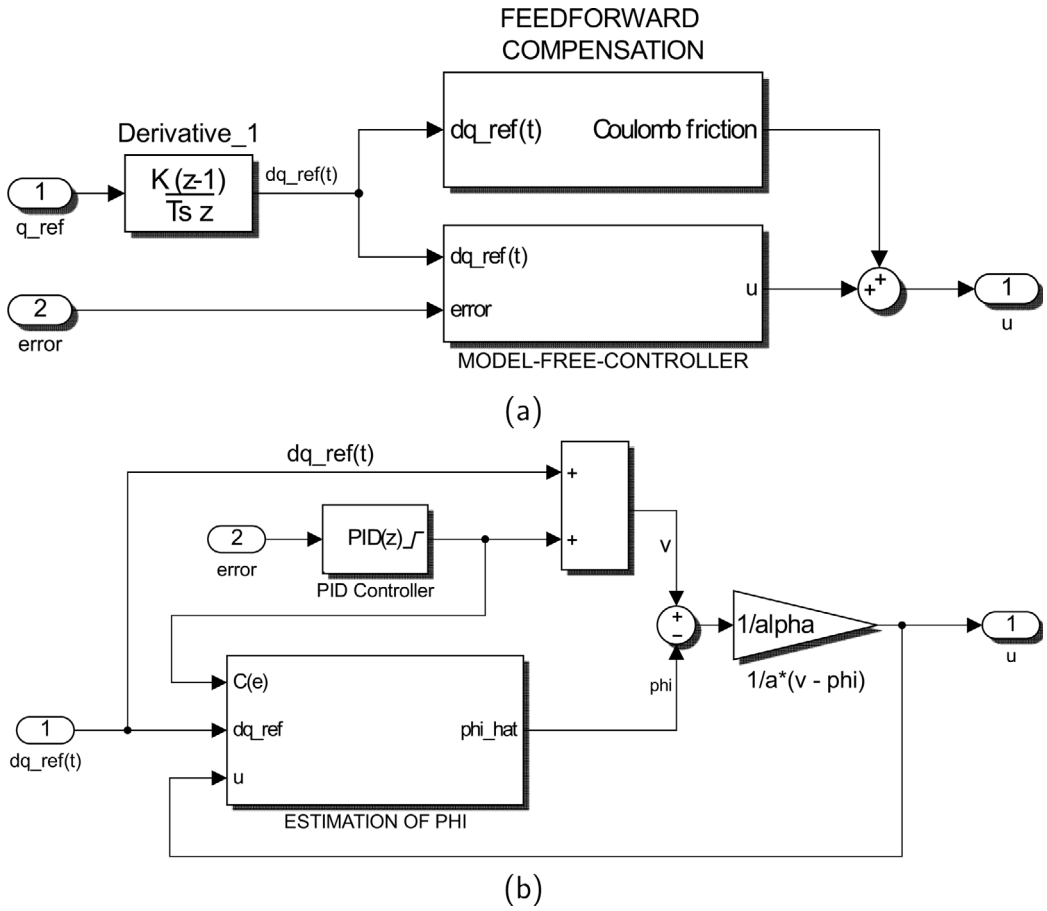

Fig. 6. (a) MATLAB/Simulink implementation of model-free-control with feedforward Coulomb friction compensation. (b) Implementation of the model-free control $\frac{1}{\alpha}(\dot{y}^* - \hat{\phi} - C(e))$. Notice that the controller $C(e)$ adds to the control law, because in implementation we compute $e = y^* - y$.

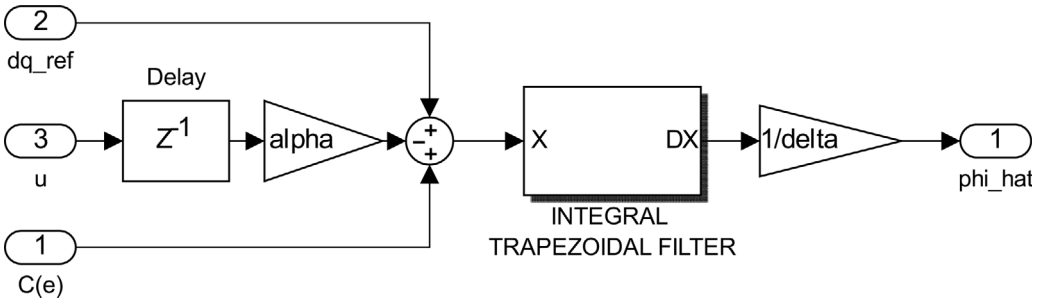

Fig. 7. Estimation of ϕ according to $\hat{\phi} = \frac{1}{\delta}\int(\dot{y}^* - \alpha \cdot u(t-1) - C(e))\,d\rho$. For the integral block we use a FIR filter set with the MATLAB coefficients as "[1 2*ones(1, n-1) 1]*Ts/2".

avoid algebraic loops, we use a backward control signal $u[k-1]$, as specified in (4) and seen by the delay in Fig. 7.

Similar control blocks are implemented for each joint. The expected dynamic coupling interaction among links is handled by the control algorithm due to its disturbance rejection properties, as reported by previous studies [20,32].

2.3. Control tuning

An important aspect of obtaining a successful performance of model-free control is tuning its parameters. As a forestry crane is a multi degree-of-freedom manipulator, it is tedious and difficult to use manual tuning for a system requiring all degrees-of-freedom to operate simultaneously. Therefore, our framework involves an optimization procedure able to tune these parameters automatically, removing the need for ad-hoc manual tuning. To this end, the optimization algorithm performs a sequential test for tuning each degree-of-freedom controller individually. The optimization goal is to find the controller parameters $[\alpha, C(e; K_p, K_i)]$ that minimize the standard control system cost function [15]

$$\min_{[\alpha, C(e;K_p,K_i)]\in\Re} \quad J = \frac{1}{2}e^2 + \delta u^2,$$

$$\text{subject to} \quad K_p^{min} < K_p(0) < K_p^{max} \tag{8}$$

$$K_i^{min} < K_i(0) < k_i^{max} \tag{9}$$

$$\alpha^{min} < \alpha(0) < \alpha^{max} \tag{10}$$

where $e = y - \dot{y}$ is the tracking error, u the control signal, and the minimum/maximum values for the tuning parameters are denoted by $(\cdot)^{min}$ and $(\cdot)^{max}$ correspondingly.

To apply this optimization approach, we use the Optimization Toolbox from MATLAB [23]. The initial conditions for the parameters are specified by preliminary defining a set of values within a range known to stabilize the feedback loop [17,19]. The maximum and minimum values for these parameters are also defined within a range known to guarantee the stability of the feedback loop. As reference trajectories, we use a summation of sinusoidal signals for each joint, having amplitudes and frequencies able to excite dynamics of the system [33], i.e. up to a frequency of 0.7 rad/s. Unlike standard optimization techniques that use a simulation system to tune parameters, the algorithm in this case runs in real-time, meaning that it drives the actual crane to tune the parameters online.

2.4. Experimental studies

The experimental studies evaluate the implementation and tracking performance of the controller. They consist of two sets of experiments conducted using both platforms, going progressively from the desktop-size to the AORO machine's crane.

2.4.1. Circle in world coordinates

In the first case, the aim is to evaluate the trajectory tracking performance and robustness of the controller after its parameters have been tuned. Specifically, the tests involve tracking a circular motion in world coordinates, a challenging task for a manipulator of this kind. This motion generates reference trajectories in joint coordinates that comprise sum of sinusoidal signals, making them difficult to accurately track when substantial Coulomb friction is present.

In these experiments the controller's fundamental settings, as it is the case of its parameters, remain consistent throughout the experiments. The objective of this approach is to evaluate the controller's performance in various scenarios without making adjustments to its internal parameters, thereby assessing its inherent robustness and stability in handling different conditions and challenges.

The experiments conducted to assess robustness and stability are structured as follows:

- **Robustness to Variations in Dynamics:** This test examines how well the controller performs when the crane undergoes changes in its dynamics, specifically with and without a load. For instance, in the case of the desktop-sized crane, an additional load of 400 grams is introduced (approximately 90% of its maximum capacity), while the AORO machine's crane carries an additional load of approximately 200 kg (roughly 40% of its maximum capacity). These variations in load mimic real-world scenarios and allow us to evaluate how effectively the controller copes with such dynamic changes.
- **Stability and Trajectory Convergence:** This test assesses the controller's stability and its ability to converge to the desired trajectory. To challenge the controller's stability, the crane-tip location is randomly initialized away from the main reference motion. This deviation from the expected starting position evaluates how well the controller can bring the system back on track and achieve the desired trajectory without modifying its underlying parameters.

Table 1
Coulomb friction parameters.

Link	3D printed crane			AORO's crane		
	f_c	Δf_c	β	f_c	Δf_c	β
Slewing	0.21	0.03	50	0.31	0.02	50
Inner Boom	0.15	0.05	20	0.4	0.03	250
Outer Boom	0.12	0.01	20	0.33	0.01	200
Telescope	0.11	−0.01	20	0.38	0.02	250

2.4.2. Reaching specified target locations in world coordinates

The second case is specifically for the AORO machine's crane, as the controller after the first test is already implemented in it. The aim is to evaluate the accuracy of moving the crane from an initial to a final configuration in work space coordinates, resembling the pick-and-place motions the crane needs to perform to collect logs.

The purpose is to quantify the deviation error at the final target location, as the principle to grab logs is by placing the crane's grapple autonomously at specified target locations. To this end, the initial crane configuration is the center of the trail where logs are piled up (see Fig. 8). The goal location is the desired Cartesian coordinate where the crane's tip is meant to reach. To move between these two points, our system uses the motion planner based on dynamic movement primitives (DMPs) [34], which for our case was presented in [35]. DMPs are able to plan semi parabolic paths for the end-effector that are transformed to joint space using inverse kinematics. Referring to Fig. 8, four desired test locations were selected to cover the four quadrants in the x-y axis, having the coordinate system at the base of the crane.

To calculate an estimation of the positioning accuracy, the deviation of the crane's tip position to the desired target is calculated by Euclidean distance. To get a reliable estimation, several repetitions of the same motion to each target location are performed. Consequently, the mean value for all the deviation errors of the four target locations is used to get a final estimation of the crane's deviation error.

3. Experimental results

In this section, all data is presented in international units. This implies that distances are given in meters [m], and angular motion is given in [rad], unless otherwise specified.

3.1. Parameters for the controller

Tests to identify the parameters for the Coulomb friction model (5) were first performed for both experimental platforms. To this end, the desktop size crane served as a test example for designing a procedure to identify Coulomb friction based on joint open-loop response to sums of sinusoidal input signals. The technique was simplistic in that the input values required to start the motion in each degree-of-freedom, in positive and negative directions, were derived from the recorded velocity response. Consequently, the same method was applied to the crane of the AORO machine.

Table 1 lists the parameters obtained for both experimental platforms. It is worth noting that the input signal is a normalized value between $[-1, 1]$. Thus, Table 1 shows that Coulomb friction is substantial in relation to the maximum input signal for both cranes. For the 3D printed crane it varies between 11 to 21% of the total input signal. For the crane of the AORO machine it ranges from 30 to 40% of the total input signal.

Table 2 presents the control parameters used for all experiments described below. These were found using the optimization procedure (10) described in Section 2.3.

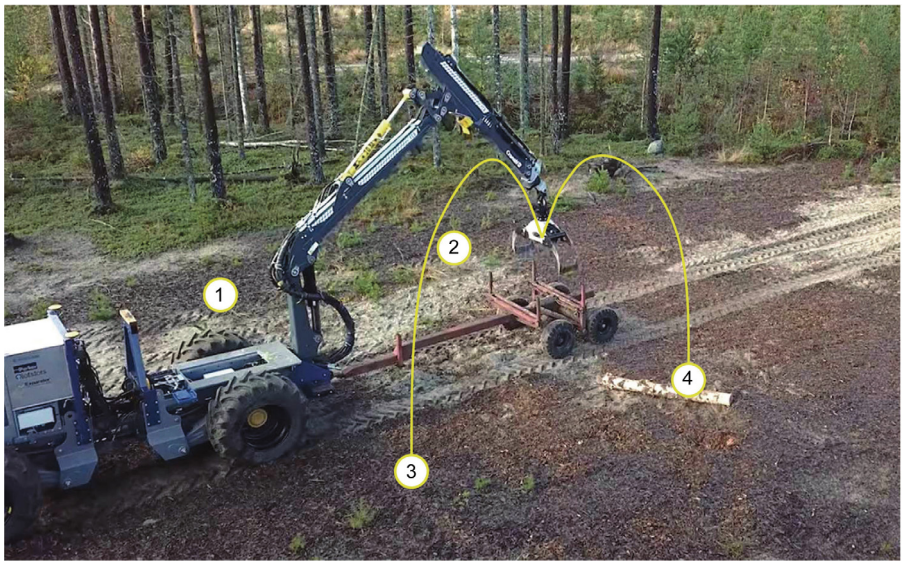

Fig. 8. Crane's desired motions from it initial configuration at the center of the trail, towards four desired locations in the Cartesian Space.

Table 2
Control parameters.

Parameter	3D printed crane				AORO's crane			
	Kp	Ki	Kd	alpha	Kp	Ki	Kd	alpha
Inner Boom	10.69	3.62	0.23	0.35	5	1	0	1.36
Outer Boom	11.31	9.7	0.14	0.81	2.5	1	0.5	1.18
Telescope	10.75	0	0.23	0.03	15	5	0	6.06

3.2. Tracking a circle in world coordinates

The first results refer to tracking a circle in world coordinates. To this end, a circular path in space is given as a reference motion, such that the boom-tip (see Fig. 3) tracks it as close as possible. The circular path is defined in Cartesian Coordinates, but the inverse kinematics is used to derived the joint reference trajectories.

3.2.1. Desktop size crane

Fig. 9 shows results when the crane has no load. The left figure shows the reference path in Cartesian space, together with the resulting motion performed by the crane. The starting crane-tip position is pointed with a dark blob, implying that the joints can be started at any random initial condition. The right of the figure shows the joint reference trajectories, together with the joint trajectories recorded from the motion sensors. Similar results for the case when the crane is loaded with a weight of 900 grams (near 90% of maximum load capacity) is presented in Fig. 10.

To show the behavior of the controller to changes in dynamics, Fig. 11 shows the control inputs for both test cases. Note that the effort of the control input increases as the crane is loaded, specially for the parts of the motion that are involved in the lifting process. For the first link, the control input reaches near saturation levels. Despite that, the controller is able to accurately track the reference trajectories as seen in Figs. 9 and 10.

In summary, the results above verify the controller's stability, as it is evident from Figs. 9 and 10 that the crane motion is able to converge towards the reference trajectory, even when the crane is started far from the reference. This is true for the cases where the crane is with and without load, demonstrating the controller's robustness to changes in dynamics.

3.2.2. AORO machine's crane

Fig. 12 shows the results of tracking a circular path without any load on the grapple. In both Cartesian and joint coordinates, the results in Fig. 12 display similar characteristics as the case of the small crane in terms of stability. However, since the AORO crane has more complex nonlinear dynamics caused by the hydraulic system, the tracking performance is not as perfect as the desktop crane. Nevertheless, the plot of joint trajectories shows that the controller is able to handle the nonlinear dynamics and track reference trajectories quite accurately. Similar results for the case when the crane is loaded with a weight of 200 kg (near 40% of maximum load capacity) are shown in Fig. 13.

To show the behavior of the controller when the crane is with and without load, Fig. 14 shows an overlay of the control inputs for both cases. We see that the effort of the control input increases as the crane is loaded, specially for the parts of the motion that are involved in the lifting process. However, for the first link, the control input does not have any noticeable changes, because there exists internal settings on the electro-hydraulic valve to compensate pressure during lifting actions.

As the circular path performed by the crane is not as accurate as the desktop crane, it is important to estimate the deviation error. For the case when the crane is empty, the tracking error in Cartesian coordinates is within a range of 5 cm, and 3 cm for the case when it is loaded. Note that the error reduces as the crane is loaded, which can be attributed to the reduction in flexible modes of the crane when the grapple is holding logs.

3.3. Reaching target positions

Referring to the tests described in Section 2.4, our motion planning algorithm plans quasi-parabolic paths to move the crane from an initial configuration towards a desired goal. In order to get an statistical estimation of the control system's accuracy, multiple trials of the same motions were recorded. These were performed both with the grapple empty and with a load of 200 Kg. As sketched in Fig. 8, four target locations specified by the following Cartesian Coordinates were chosen for this test:

- Point 1 = [6, −1.86, 1]
- Point 2 = [6, 1.86, 1]
- Point 3 = [4.27, −1.86, 1]
- Point 4 = [4.27, 1.86, 1]

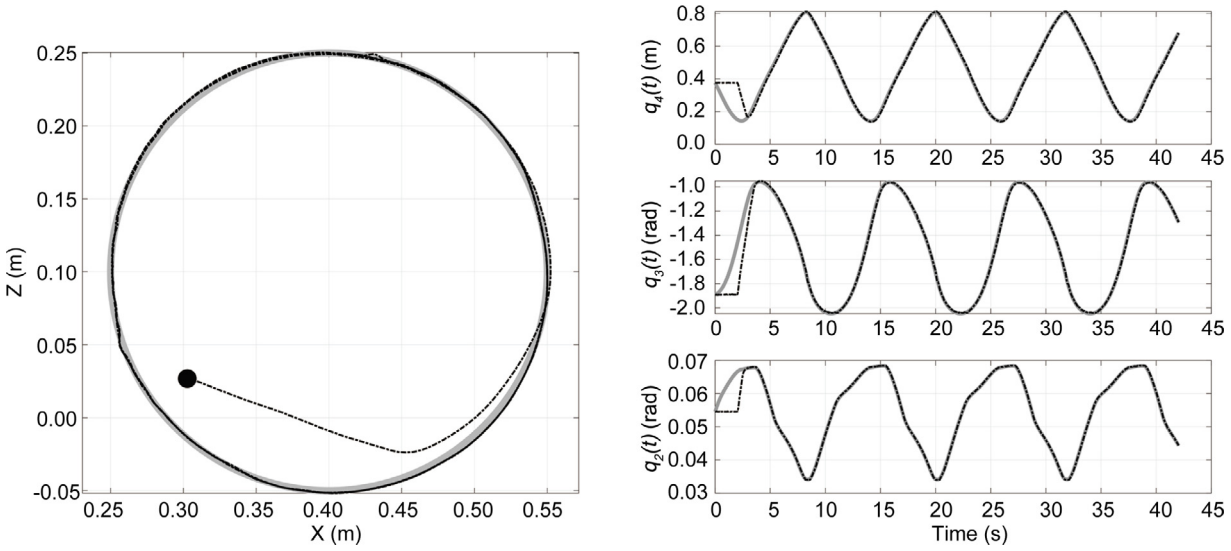

Fig. 9. Desktop size crane results without load. Tracking a circle in world coordinates. The left figure shows results in Cartesian coordinates. The right Figure shows results in joint coordinates.

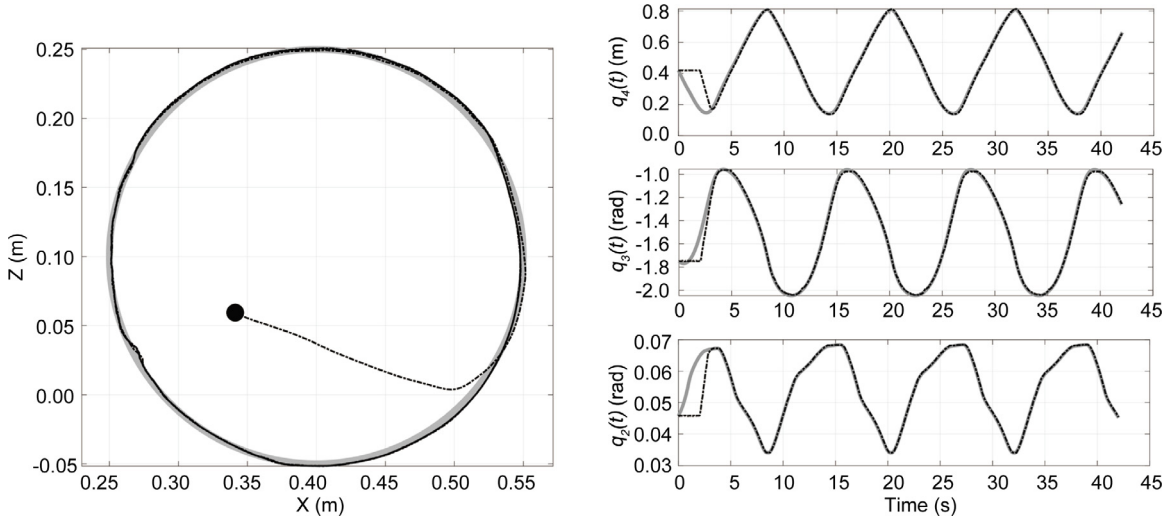

Fig. 10. Desktop size crane results with load. Tracking a circle in world coordinates. The left figure shows results in Cartesian coordinates. The right Figure shows results in joint coordinates.

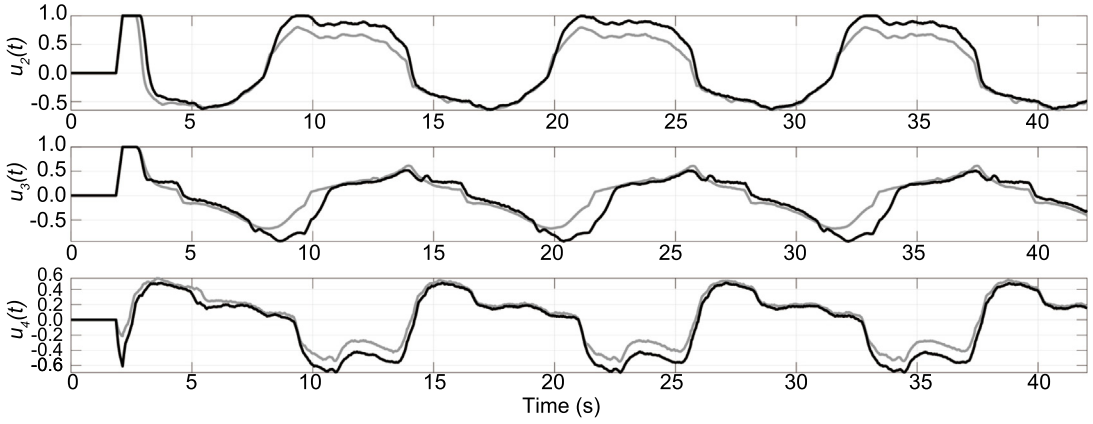

Fig. 11. Desktop size crane. This results shows the control input for the cases when the crane is with load (dark bold signal) and without load (gray signal).

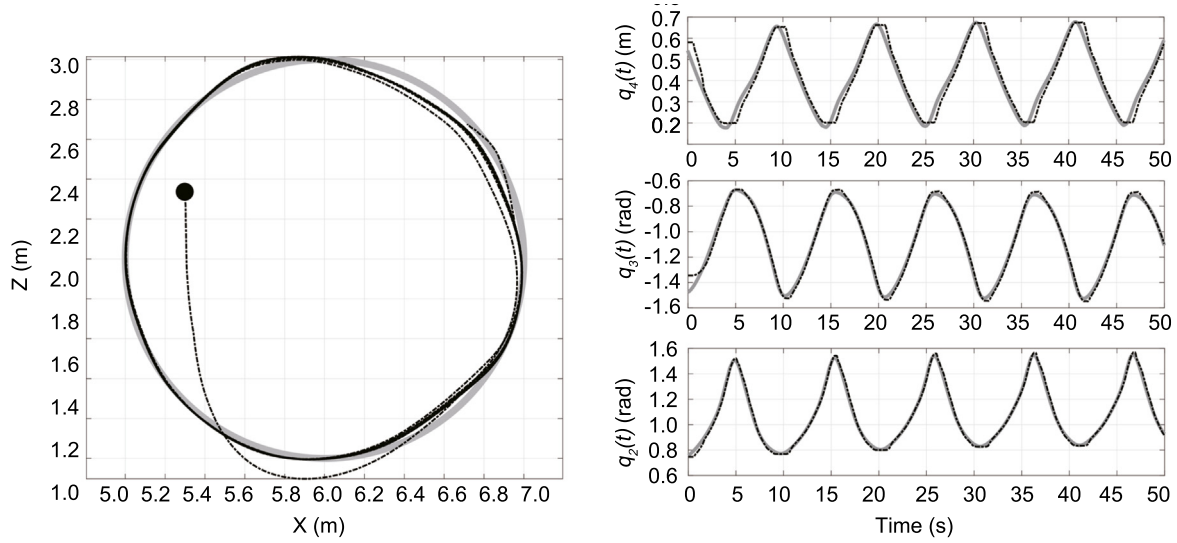

Fig. 12. AORO's crane results without load. Tracking a circle in world coordinates. The left figure shows results in Cartesian coordinates. The right Figure shows results in joint coordinates.

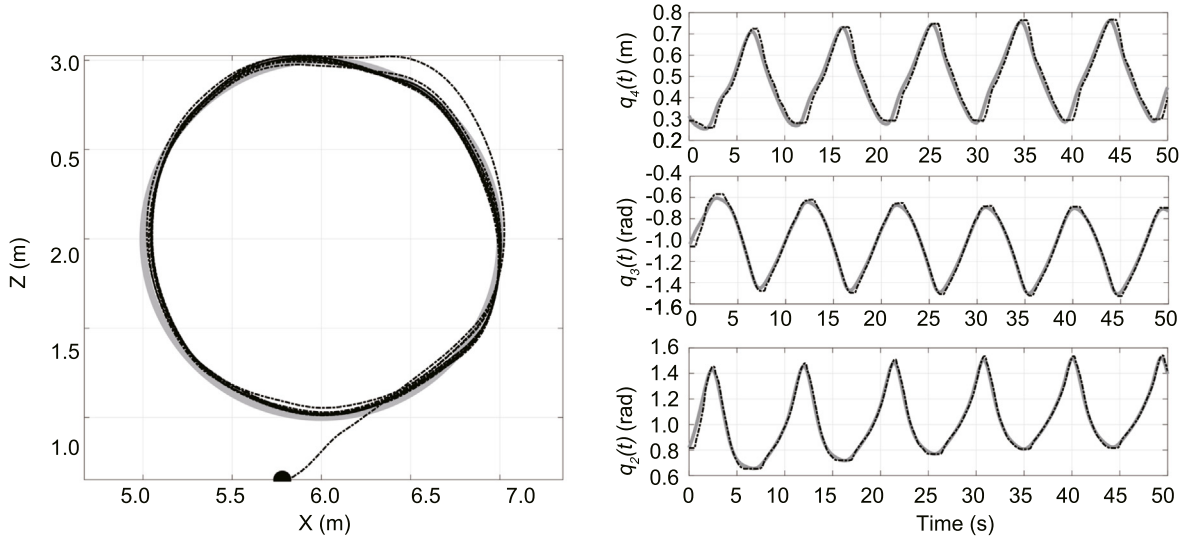

Fig. 13. AORO's crane results with load. Tracking a circle in world coordinates. The left figure shows results in Cartesian coordinates. The right Figure shows results in joint coordinates.

Considering as a reference the sketch presented in Fig. 8, Fig. 15 shows the specific data points of the four target locations, as well as the recorded data for the crane-tip Cartesian coordinates once it stops at the final target location. A number of 20 trials is presented at each target location, meaning that a total of 80 data points are shown for all four target locations.

The left of Fig. 15 shows the results using a 3D Cartesian coordinate system. However, to have a better observation of the results at each target location, the right side of Fig. 15 shows a top view of the individual results for each target location in the X-Y axis. In all the cases, all data points with a square are for the motions with an empty grapple, while data points with an x are for the motions with a loaded grapple.

To show the accuracy of the crane's motion control system, we use the Euclidean distance of the boom-tip to the desired location as a measurement of error, because this value tells how much the crane deviates from the target location. Fig. 16 presents a histogram showing these values for all data points, i.e. the data for all four desired locations are piled up into a single histogram plot. The left plot is the Euclidean distance calculated in 3D, i.e. using the data in the $[x, y, z]$ axis. The middle plot is the Euclidean distance calculated in 2D, i.e. using the data in the $[x, y]$ axis. The right plot is the error in height, i.e. z axis.

Results show that the Cartesian coordinate positioning error in 3D follows a normal distribution, having an average value of 8 cm, i.e. the crane reaches the vicinity of the desired location with an average error of 8 cm. However, we see that the error in the $[x, y]$ axis is smaller on average, i.e. 4 cm. This implies that the highest error in the crane positioning is the height of the tip, which deviates by an average of 6 cm with respect to the desired height. This phenomenon can be observed in the left of Fig. 15, which shows that the deviation error in the z-axis is larger for the locations where the crane needs to extend further.

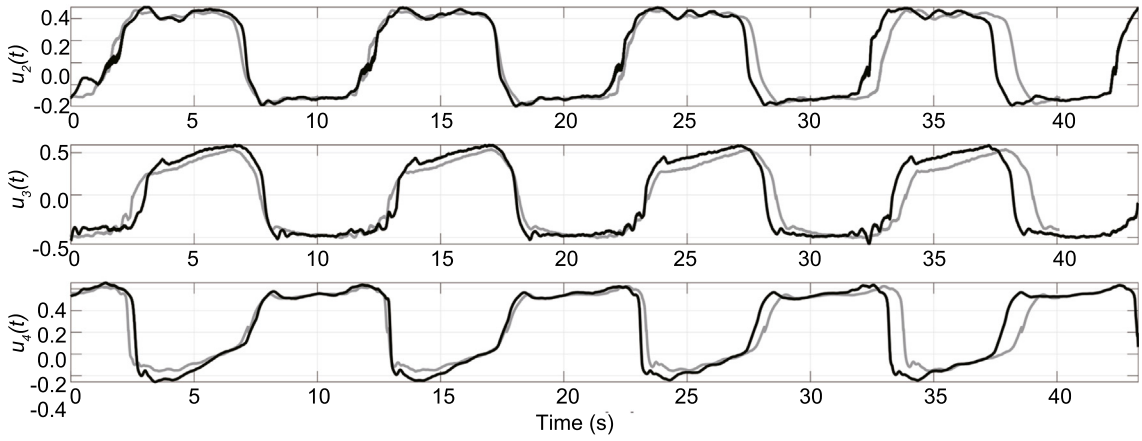

Fig. 14. AORO crane. This results shows the control input for the cases when the crane is with load (dark bold signal) and without load (gray signal).

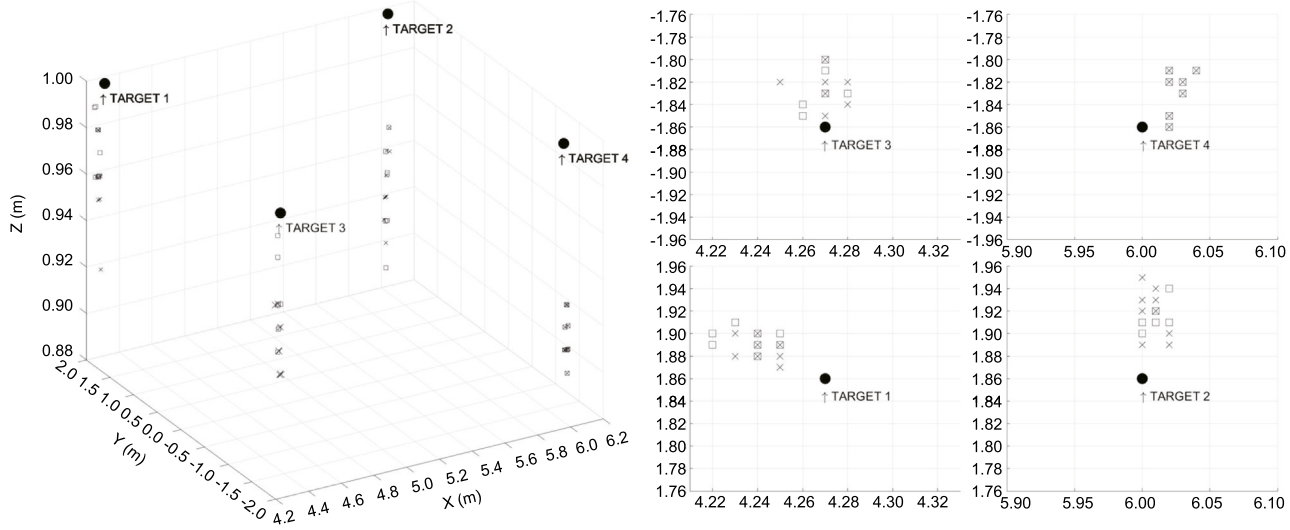

Fig. 15. Referring to Fig. 8, these plots show the Cartesian coordinates of the four desired target locations, as well as the recorded data of the crane's tip once it reaches its final configuration. The left plot is the data on a 3D Cartesian coordinate system. The right plot is a top view of the individual target locations in the X-Y axis. In all the plots, the square data points represent the motions when the crane's grapple is without load, while the *x* marks data points with a loaded grapple.

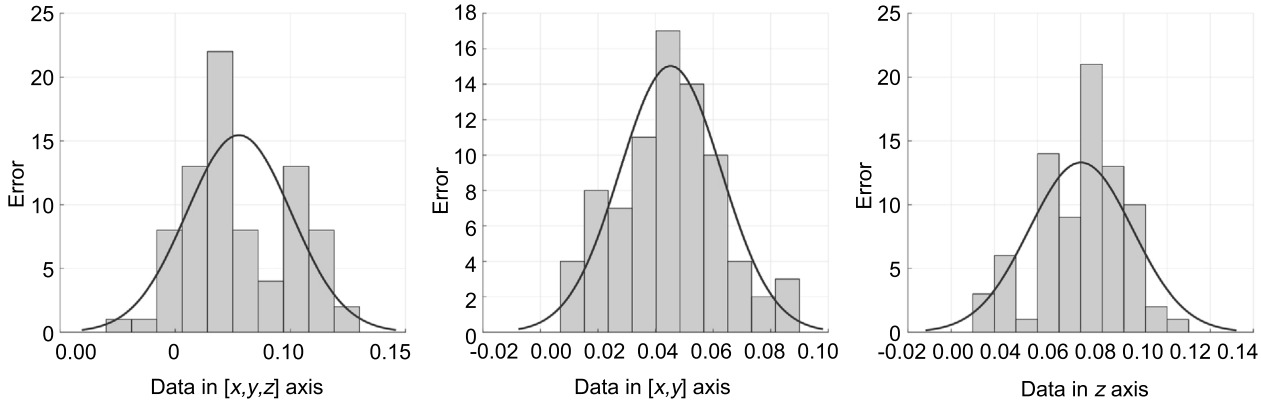

Fig. 16. Cartesian positioning error for the crane's motion control system. From left to right, the histograms represent the error measured by the Euclidean distance to the desired location for 3D ([x,y,z] axis), 2D ([x,y] axis) and height (z axis), respectively.

4. Discussion

Testing control algorithms can be time-consuming, expensive, and potentially unsafe if one decides to test against a real heavy-duty system. Therefore, techniques involving Software and Hardware-in-the-Loop (SIL and HIL) testing solutions have advanced greatly the past decades, providing an effective, cost-effective, and safe way to test control software prior real

implementation. Nevertheless, many applications still require techniques to ease the transition of implementation, because the interaction between hardware and software of many systems are not easy to simulate in software.

Having the ability to complement early development with 3D printing has several advantages. The most important to us is having the ability to equip our 3D printed cranes with hardware, facilitating performance tests with complete hardware and in real-time with model-free controllers. In this article, we have shown how 3D printing has been useful as part of our development framework, both for preliminary tests of control software implementation, as well as to prepare experimental tests that need to be carried out on the commercial crane. In summary, having desktop-size forestry cranes at our disposal has helped to efficiently plan, prepare, and test experiments, thus reducing the risks of damage that malfunctioning software can cause.

In regards to our optimization-based tuning procedure, it is important to highlight that while our manuscript focuses on the development of a motion control system specifically tailored for the AORO platform's crane, this technique is presented as a possible general tuning approach for a controller of this kind. This implies that even though we could manually fine-tune the controller for our specific example, the significance of this methodology lies in its scalability. The tuning approach enables to tune the controller of multiple cranes, each varying in size and design, a common situation in manufacturing companies selling the function of computer-controlled cranes. Essentially, our work presents not just a solution for a single crane but a methodology addressing the complexities of tuning diverse crane systems systematically, without the need for a control system expert.

4.1. Choice of control algorithm

Over the years, we have reported results with a variety of motion control methods for electro-hydraulically actuated manipulators. These include PID, sliding mode, feedback linearization, and other nonlinear control methods, as it can be read in [16,17,19]. Many of these cases involve nonlinear model-based control algorithms, as these are standard in research involving motion control of robotic manipulators [15]. However, in our experience, obtaining a reliable model of a hydraulically actuated manipulator leads to arduous, tedious, and difficult system identification testing. In many instances, the model of a crane cannot directly be applied for a second one, as the settings in hydraulic valves are not general from crane to crane, or manufacturer to manufacturer, forcing to remodel and repeat system identification tests every single time we approach a new system. In addition, even though we have experienced that the performance of model-based nonlinear controllers is superior to standard PID controllers, we have also experienced that they fail to provide the robustness we seek to drastic changes in parameters. To put it clearly, forestry cranes perform work provoking the system to undergo huge and sudden variations in parameters, because they have the capacity to manipulate loads with weights nearly similar to their own. Also, oxidation affects the friction of the actuators from season to season, in turn affecting the motion of the system. Therefore, a method to improve such controllers is to develop online load weight and friction estimations, giving the control algorithm adaptive properties. Yet, the performance of such controllers has not been greater to that of model-free iPID control, making it difficult to justify the effort of implementation.

In our experience, model-free control is the best performance controller we have developed so far, with the advantages that it is robust, it has adaptive properties, it is relatively easy to implement, it does not require a system's model in its development, and it provides a good trade-off between accuracy and performance. In addition, the simplicity on its implementation and tuning makes it a great option for engineers in industry that need to adapt the controller for products having similar functional characteristic, but different sizes and weights, e.g. cranes with different sizes. This is also one of the reasons why this controller has lately become interesting for a wide range of applications, including industrial automation [30], as well as more complex robotic systems [20,21,32].

It is important to highlight that within research on iPID control, possessing an ultra-local model of degree one suggests that an iP controller should be adequate for effectively controlling this system. Nevertheless, the parameters employed in our experiments are a product of an optimization process designed to enhance tracking performance by leveraging additional parameters. Given our objective to utilize such software for parameter tuning, our primary concern is to ensure that the resultant parameters are not only reliable but also perform as intended.

4.2. Discussion about results

Unlike industrial robot manipulators, highly accurate control of forestry cranes is challenging, due to size, weight, and nonlinear hydraulic dynamics. When we implement control algorithms in a heavy-duty crane, we often need to make a compromise between dynamic performance and control accuracy, i.e. performing motions that are accurate enough without exciting the flexible dynamics of the system. In these regards, our crane motion control system provides the ability to perform both smooth and fast motions with accuracy that is sufficient for our application, where having a grapple as end-effector is a rather forgiving mechanism to successfully pick-and-place logs.

The end application for our motion control system is to perform pick-and-place tasks, as seen from standard forwarder machines. To this end, our results confirm the controller's ability to reliably track reference trajectories, while being robust to variations in the grapple's load, i.e. when it is loaded or empty. In addition, our results demonstrate that our positioning accuracy is within a few centimeters, which is sufficient for a crane working on ranges of several meters, and having to handle logs having dimensions in the range of meters.

5. Conclusion

Our current results have several strengths. Firstly, we were able to show that using a 3D printed mockup system for the development of model-free controllers offers several benefits:

(1) **Real-World Validation:** The mockup crane provides a physical representation of the actual hydraulic crane, allowing us to work with a tangible setup. This enables the evaluation of our control algorithm in a physical setup, ensuring that it perform as expected when we transfer it to the real crane.
(2) **Data Collection:** The mockup crane generates real-world data during its operation. This ability allow us to plan experiments that have to be repeated in the real crane.
(3) **Adaptability:** Hydraulic cranes often operate in dynamic and unpredictable environments. A mockup system allows researchers to test and refine the controller's adaptability to varying conditions, such as changes in load, terrain, or operational requirements.
(4) **Model-Free Approach:** Model-free control methods, like intelligent PID control, do not rely on precise mathematical models of the system. The mockup crane facilitates the development of such controllers by providing similarities to the real input–output data without the need for complex system modeling and simulation.

(5) **Risk Mitigation:** Testing and refining control algorithms on a mockup system before implementing them in an actual crane can help mitigate risks associated with unproven software or control strategies. It allows for thorough validation and optimization before deployment.
(6) **Safety:** The mockup crane provides a safe and controlled environment for testing and experimentation, helping us to mitigate risks associated with unproven software or wrongly coded control strategies.

Secondly, the primary contribution of our research lies in the application of model-free iPID control to a heavy-duty hydraulic crane, which is an area that has not been previously studied, as there are not so many groups that can easily access these kind of setups. Most developments in this field often rely on intricate mathematical models and simulations. However, in the case of heavy-duty hydraulic cranes, the dynamic nature of their operations and the absence of precise models pose significant challenges. Our work pioneers a novel approach by employing model-free iPID control, which uses input and output data without relying on a predefined system model. The characteristics of this control approach allows us to robustly control the crane's motions despite changes in dynamics. By introducing this control technique, as well as the AORO platform in general, we are opening the possibility to explore fully autonomous forest operations, which is an important domain of development that aligns with future sustainability goals [36].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge Prof. Ola Lindroos from SLU and Prof. Magnus Karlbeg from LTU for acquiring the financial support to carry on with this project. The AORO project involving the crane automation is partly financed by the Swedish Foundation for Strategic Environmental Research MISTRA (program Mistra Digital Forest), the Kempe Foundations (project JCK-1713), the Swedish Cluster of Forest Technology, and the Swedish Energy agency through the project HASSBiT. We would also like to acknowledge the help of Martin Sedin and Hugo Eriksson, students from SLU, for their help performing the experimental studies with the crane of the AORO machine.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.birob.2023.100133.

References

[1] AORO: Arctic off-road robotics lab, 2021, Retrieved November, 2021, from https://www.skogstekniskaklustret.se/projekt/aoro.
[2] M. Fliess, Model-free control and intelligent PID controllers: towards a possible trivialization of nonlinear control? IFAC Proc. Vol. 42 (10) (2009) 1531–1550.
[3] M. Fliess, C. Join, Model-free control, Internat. J. Control 86 (12) (2013) 2228–2252.
[4] Komatsu Forest AB, Komatsu Smart Crane - Simpler, more ergonomic and higher productivity, 2021, https://www.komatsuforest.com/explore/smart-crane-for-forwarders.
[5] John Deere, Operator assistance - Intelligent boom control, 2022, https://www.deere.com/en/technology-products/forestry-and-logging-technology/operator-assistance-technology/.
[6] Cranab, Cranab presents an entirely new generation of cranes, 2011, http://www.cranab.se/static/en/206/.
[7] CRANAB FC8, 2021, Retrieved November, 2021, from https://www.cranab.com/products/forwarder-cranes/fc8.
[8] EATON, 2019, https://www.eaton.com/us/en-us/catalog/valves/cma-advanced-mobile-valves.resources.html.
[9] O. Lindroos, P. La Hera, C. Häggström, Drivers of advances in mechanized timber harvesting–a selective review of technological innovation, Croatian J. For. Eng. J. Theory Appl. Forest. Eng. 38 (2) (2017) 243–258.
[10] A. Halme, M. Vainio, Forestry robotics-why, what and when, in: Autonomous Robotic Systems, Springer, 1998, pp. 149–162.
[11] The death of the forest Beast, 2006, Retrieved November, 2021, from https://www.forest-monitor.com/en/death-forest-beast/.
[12] The radio-controlled bio-energy harvester forest ebeaver, 2011, Retrieved November, 2021, from http://ebeaver.se/Default.aspx.
[13] M. Lundbäck, Roadmap for Teleoperation and Automation of Forwarding (Ph.D. thesis), Swedish University of Agricultural Sciences, 2022.
[14] J. Mattila, J. Koivumäki, D.G. Caldwell, C. Semini, A survey on control of hydraulic robotic manipulators with projection to future trends, iEeE/ASME Trans. Mechatron. 22 (2) (2017) 669–680.
[15] M. Spong, S. Hutchinson, M. Vidyasagar, Robot Modeling and Control, John Wiley and Sons, New Jersey, 2006.
[16] P. La Hera, D.O. Morales, Non-linear dynamics modelling description for simulating the behaviour of forestry cranes, Int. J. Model. Ident. Control 21 (2) (2014) 125–138.
[17] P. La Hera, D. Ortíz Morales, Model-based development of control systems for forestry cranes, J. Control Sci. Eng. 2015 (2015).
[18] N. Sun, Y. Fang, H. Chen, B. He, Adaptive nonlinear crane control with load hoisting/lowering and unknown parameters: Design and experiments, IEEE/ASME Trans. Mechatronics 20 (5) (2015) 2107–2119, http://dx.doi.org/10.1109/TMECH.2014.2364308.
[19] D. Ortiz Morales, S. Westerberg, P.X. La Hera, U. Mettin, L. Freidovich, A.S. Shiriaev, Increasing the level of automation in the forestry logging process with crane trajectory planning and control, J. Field Robotics 31 (3) (2014) 343–363.
[20] H. Abouaïssa, S. Chouraqui, On the control of robot manipulator: A model-free approach, J. Comput. Sci. 31 (2019) 6–16.
[21] Y.A. Younes, A. Drak, H. Noura, A. Rabhi, A.E. Hajjaji, Robust model-free control applied to a quadrotor UAV, J. Intell. Robot. Syst. 84 (1) (2016) 37–52.
[22] E. Madadi, D. Söffker, Model-free approaches applied to the control of nonlinear systems: A brief survey with special attention to intelligent PID iterative learning control, in: Dynamic Systems and Control Conference. Vol. 57243, American Society of Mechanical Engineers, 2015, V001T03A004.
[23] The Mathworks, 1990, http://www.mathworks.com.
[24] L12-P micro linear actuator with position feedback, 2022, Retrieved February, 2022, from https://www.actuonix.com/L12-P-Micro-Linear-Actuator-with-Position-Feedback-p/l12-p.htm.
[25] Arduino products, 2022, Retrieved February, 2022, from https://www.arduino.cc/en/Main/Products.
[26] Simulink coder. Generate C and C++ code from simulink and stateflow models, 2022, Retrieved April, 2022, from https://www.mathworks.com/products/simulink-coder.html.
[27] Rack mountable Simulink/RTW target, ideal for HIL applications, 2021, Retrieved November, 2021, from https://www.ueidaq.com.
[28] R. Madoński, P. Herman, Model-free control or active disturbance rejection control? on different approaches for attenuating the perturbation, in: 2012 20th Mediterranean Conference on Control & Automation, MED, IEEE, 2012, pp. 60–65.
[29] M. Fliess, S. Riachy, Revisiting some practical issues in the implementation of model-free control, IFAC Proc. Vol. 44 (1) (2011) 8589–8594.
[30] F. Lafont, J.-F. Balmat, N. Pessel, M. Fliess, A model-free control strategy for an experimental greenhouse with an application to fault accommodation, Comput. Electron. Agric. 110 (2015) 139–149.
[31] J. Villagra, B. D'Andréa-Novel, S. Choi, M. Fliess, H. Mounier, Robust stop-and-go control strategy: an algebraic approach for non-linear estimation and control, Int. J. Veh. Autonomous Syst. 7 (3–4) (2009) 270–291.
[32] J. Villagra, C. Balaguer, A model-free approach for accurate joint motion control in humanoid locomotion, Int. J. Human. Robot. 8 (01) (2011) 27–46.
[33] L. Ljung, System Identification: Theory for the User, Prentice Hall, Upper Saddle River, 1999.
[34] P. Zhou, X. Zhao, B. Tao, H. Ding, Combination of dynamical movement primitives with trajectory segmentation and node mapping for robot machining motion learning, IEEE/ASME Trans. Mechatronics (2022).
[35] P. La Hera, D.O. Morales, O. Mendoza-Trejo, A study case of Dynamic Motion Primitives as a motion planning method to automate the work of forestry cranes, Comput. Electron. Agric. (ISSN: 0168-1699) 183 (2021) 106037, http://dx.doi.org/10.1016/j.compag.2021.106037, URL https://www.sciencedirect.com/science/article/pii/S0168169921000557.
[36] The world's first self-propelled forestry machine, 2021, Retrieved November, 2021, from https://gamingsym.in/the-worlds-first-self-propelled-forestry-machine-removes-man-from-the-equation/.