A vision-based positioning system with inverse dead-zone control for dual-hydraulic manipulators

C. West, S. D. Monk, A. Montazeri and C. J. Taylor

Engineering Department, Lancaster University, UK

Email: c.taylor@lancaster.ac.uk

Abstract—The robotic platform in this study is being used for research into assisted tele-operation for common nuclear decommissioning tasks, such as remote pipe cutting. It has dual, seven-function, hydraulically actuated manipulators mounted on a mobile base unit. For the new visual servoing system, the user selects an object from an on-screen image, whilst the computer control system determines the required position and orientation of the manipulators; and controls the joint angles for one of these to grasp the pipe and the second to position for a cut. Preliminary testing shows that the new system reduces task completion time for both inexperienced and experienced operators, in comparison to tele-operation. In a second contribution, a novel state-dependent parameter (SDP) control system is developed, for improved resolved motion of the manipulators. Compared to earlier SDP analysis of the same device, which used a rather *ad hoc* scaling method to address the dead-zone, a state-dependent gain is used to implement inverse dead-zone control. The new approach integrates input signal calibration, system identification and nonlinear control design, allowing for straightforward recalibration when the dynamic characteristics have changed or the actuators have deteriorated due to age.

I. INTRODUCTION

The need for nuclear decommissioning is increasing globally. Following the Fukushima disaster in 2011, Japan is closing most of its 54 reactors. In Europe, more than 50 reactors are being closed in the next 10 years. Nuclear installations have materials that present various radioactive and chemical hazards; their architecture is often complex; and they were not necessarily designed with the decommissioning problem in mind. Hence, improved robotic operations are essential for safety and efficiency, notably for access to areas of high radiation where it is dangerous for human workers.

However, full autonomy of robotic systems is undesirable due to the high-risk environment. Increased automation yields complex interactions with the user and a reduction in situational awareness [1]. In fact, most robotic systems in the nuclear sector are directly teleoperated, as originally was the case for the robotic platform considered in the present article. Here, two seven-function, hydraulically actuated manipulators have been mounted on a mobile BROKK base unit [2].

Achieving accurate movement of such manipulators is a challenge, even following lengthy training and practice. As a result, many methods other than joystick control exist in the literature. These include master-slave systems [3] and devices that track user arm movement [4]. Various graphical interfaces are also possible, such as systems in which an operator uses a mouse to control a 6 degree–of–freedom (DOF) ring and arrow marker to set end-effector position and orientation on screen [5]. Such systems are only possible when the work environment is already well-known, or where multiple fixed sensors can combine to give detailed scene information.

Control methods that add a level of automation above direct teleoperation are often termed assisted teleoperation [6]. However, as the level of assistance increases, the requirement for scene information is also increased. Multiple stereo or 3D cameras are often used, placed in different fixed positions to observe the work environment from different angles. Unfortunately, a multiple camera approach is not always practical in a nuclear decommissioning context. In a nuclear environment, electronics are easily damaged by radiation, hence it is preferable to reduce the number of sensors and other electronics, creating restrictions that need to be considered when developing assisted teleoperation [7]. The fact that the present BROKK system is necessarily mobile (via caterpillar tracks) adds further constraints to the available data.

Although prevalent in many other industries, vision systems are rare in the nuclear industry, except for those used simply to allow for remote teleoperation. In one recent example, a monocular vision system is used to control a manipulator to fasten bolts onto a sealing plate for a steam generator [8]. The robot operates by having a camera locate bolts around the plate and subsequently moving to fasten these to a set torque value. However, the system is limited in that it only looks for hexagonal bolts on a specially designed, bespoke plate.

Hence, the aim of the present research is to reduce the operator's workload, speeding up task execution and reducing operator-training time, whilst minimizing the introduction of additional sensors and other components. As pointed out by Katz *et al.* [9], the challenges associated with unstructured environments are a consequence of the high-dimension of the problem, including robotic perception of the environment, motion planning and manipulation, and human-robot interactions. Due to limited sensor data availability, a system that can grasp generic objects, for example, could be unreliable. As a result, the developed visual servoing approach is based on the concept of multiple subsystems for common tasks under one user interface: one subsystem for pipe cutting, one for pick and place operations, and so on. This approach reduces the complexity of the problem, potentially leading to improved performance and reliability. Furthermore, cognitive workload is generally reduced by tailoring the information shown to the operator to one particular (decommissioning) task at a time.

This article focuses on pipe cutting as an illustration of the generic approach, since this is a common repetitive task in nuclear decommissioning. The user selects the object to be cut from an on-screen image with a mouse click, whilst the computer control system determines the required position and orientation of the manipulators in 3D space, and calculates the necessary joint angles i.e. for one manipulator to grasp and stabilise the selected object and the other to position for a cutting operation. The approach is similar to Kent *et al.* [10], who use a single manipulator with two 3D cameras. By contrast, we have dual manipulators on a mobile platform, and use a single camera. Marturi *et al.* [7] discuss some of the challenges involved in a nuclear context, and the results of a related pilot study. In all these cases, it is clear that, to improve task execution speed and accuracy, high performance control of nonlinear manipulator dynamics is required.

In this regard, conventional identification methods for robotic systems include, for example, maximum likelihood [11], Kalman filtering [12] and inverse dynamic identification model with least squares (IDIM-LS) [13]. Refined Instrumental Variable (RIV) algorithms [14] are also used, sometimes in combination with State-Dependent Parameter (SDP) [2] and inverse dynamic models [15]. SDP models are estimated from data within a stochastic state-space framework and take a similar structural form to linear parameter-varying systems [16]. The parameters of SDP models are functionally dependent on measured variables, such as joint angles and velocities in the case of manipulators. Such models have been successfully used for control of a KOMATSU hydraulic excavator [17], while [18] further demonstrates the advantages of the SDP approach, in comparison to IDIM-LS methods.

In comparison to a typical machine driven by electric motors, hydraulic actuators generally have higher loop gains and lightly-damped, nonlinear dynamics [19]. The manipulators used in the present research have been represented using physically-based equations [20]. However, the present article instead focuses on the identification of a relatively straightforward SDP model. In contrast to earlier SDP control of the same device [2], a new model structure is identified, one that provides estimates of the dead-zone and angular velocity saturation, in a similar manner to the friction analysis of Janot *et al.* [18]. This model facilitates use of an Inverse Dead-Zone (IDZ) [21] controller, which is combined with conventional Proportional-Integral-Plus (PIP) methods [22]. In contrast to reference [23], the present work considers resolved motion, within the context of the vision system.

Section II briefly introduces the robotic platform, section III describes the vision system and section IV the SDP control methodology. This is followed by the experimental results and conclusions in sections V and VI respectively.

II. ROBOTIC PLATFORM

The BROKK–40 consists of a moving vehicle, hydraulic tank, remote control system and manipulator. The latter is linked via a bespoke back-plate to two HYDROLEK–7W manipulators (Fig. 1). The unit is electrically powered, with an
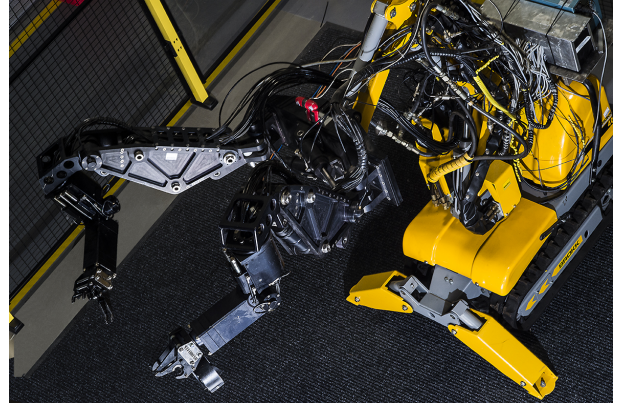

Fig. 1. BROKK–HYDROLEK (further images: https://tinyurl.com/yd3bvslo).

on-board hydraulic pump to power the caterpillar tracks and, by means of hydraulic pistons, the manipulators. Each manipulator has 6 DOF with a continuous jaw rotation mechanism. The end-effectors can be equipped with a variety of tools, such as hydraulic crushing jaws. The manipulator joints are fitted with potentiometer feedback sensors, allowing the position of the end-effector to be determined during operation [2].

An input device, such as a joystick, is connected to a PC running a graphical user interface (GUI) developed by the authors with National Instruments (NI) Labview software. The PC transmits information to a NI Compact Fieldpoint Real-Time controller (CFP) via an Ethernet connection. The platform has some similarity to the Hitachi system [24]. The Hitachi consists of two hydraulic manipulators on a tracked vehicle. However, it is teleoperated with a rather complex user interface. This complexity arises from the many cameras and sensors on the Hitachi, as well as from the teleoperated control system. By contrast, the assisted teleoperation system developed below is designed to keep the user engaged and in control at all times, whilst being as straightforward as possible.

III. VISION SYSTEM

The vision system is largely based on off–the–shelf components and image processing algorithms, that have been integrated and adapted for this application [25]. A Microsoft Kinect is mounted between the two manipulators, giving a view of the workspace directly to the front. Although originally developed for gaming applications, the Kinect is now widely for robotics research. It couples a standard camera with a structured light depth sensor, allowing both colour and depth data to be used and combined. Placing the camera between the manipulators provides an intuitive system for the human operator. A live video stream is displayed on the GUI as the mobile base unit is positioned and stabilised. The colour and depth images are aligned and converted to grayscale. To reduce the size of the image and computational complexity, as well as to only present useful information to the operator, all areas that are either out of reach of the system or that have no depth data available, are removed. This leaves the image with only the reachable objects (Fig. 2).

Fig. 2. Preliminary image processing: (a) original and (b) simplified image.

Fig. 3. Edge detection: (a) original and (b) combining four contrasts.

A. Edge Detection

Three copies of the image are created, each with a different contrast level, adjusted via the GUI. Two copies are set using sliders and the third is adjusted to a pre-set mid value. These three images, together with the original, are forwarded to the edge detection algorithm. Since the working environment is likely to be poorly lit, this approach lessens the impact of shadows and highlights, and is found to capture more detail in practice (Fig. 3). The four images are passed through a canny edge detection algorithm [26] in MATLAB. The canny algorithm has been shown to perform better than other edge detection algorithms in most situations, albeit at a relatively high computational cost [27]. Reliable detection of solid edges is a key requirement here. A third slider on the GUI determines the sensitivity threshold, allowing for adjustments to which edges are detected. The images are combined and the edges dilated to improve clarity. The three sliders on the GUI update the image in real-time, hence allowing the operator to make adjustments, potentially removing superfluous edges or filling in missing edges, without having to reposition the system.

B. Object Selection

An object is selected with a mouse. A colour screengrab with the selected item highlighted is shown, providing visual confirmation. In the case of a pipe, the user proceeds to click on the point they determine the manipulator should grasp, and then the location of the cutting operation. At this stage, the selected positions for grasp and cut are snapped to the major axis of the object i.e. the centre line of the pipe, determined using the MATLAB Image Processing Toolbox. The cutter path is perpendicular to the major axis, and also takes a pre-set distance from the centreline. Hence, any orientation of pipe can be addressed and, since depth data are available, the pipe does not even have to be in one plane. To reduce the possibility of collisions, and to help position the gripper in the target orientation, the end-effector is first moved to a position in front of the target; once it has reached a set error tolerance for that position, it is moved to the final grasp position. Although more sophisticated collision avoidance algorithms are available in the literature, this pragmatic approach ensures that the end-effector always moves directly forward onto the pipe to grasp it, and is found to work very well in practice. Of course, the operator retains the option to immediately return to full tele-operation when the unexpected arises.
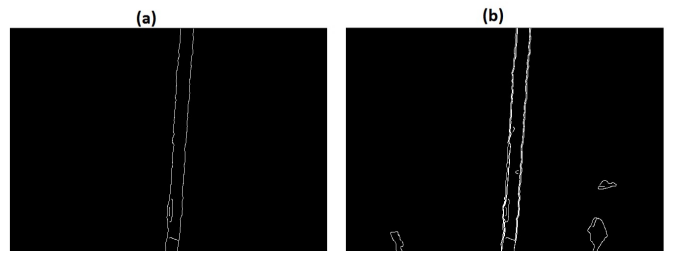
C. Real–Time Implementation

Target locations above refer to pixel coordinates and depth values. However, to be used as Inverse Kinematics (IK) solver arguments, these are converted into the manipulator coordinate system. A trigonometric approach is taken. The number of pixels in the Kinect image is 640×480, while the camera field of view is $57°$ horizontal and $43°$ vertical. Using the angle and the depth data, 'real world' coordinates relative to the camera origin are determined. For the Kinect, points on a plane that are parallel to the sensor have the same depth value and this defines the trigonometric identity. At this stage, the coordinates are relative to the camera centre. A translation is used to convert these into a usable form [25]. No analytical IK solution exists for this manipulator design, hence the present work utilises the Jacobian transpose to determine target angles [28]. The algorithms behind the vision system are all implemented in MATLAB. However, the interface to the robotic actuators is via NI Labview, with these elements connected via TCP-IP locally. This architecture would allow for the GUI to be implemented on one PC and the Labview manipulator control to be performed on another for remote operation. Finally, the Labview control software moves the joints to the set points provided by MATLAB, as discussed below.

IV. Control System

The SDP model is $y_k = \mathbf{w}_k^{\mathrm{T}} \mathbf{p}_k$, with SDP parameter vector,

$$\mathbf{p}_k = [a_1 \{\boldsymbol{\chi}_k\} \ \cdots \ a_n \{\boldsymbol{\chi}_k\} \ b_1 \{\boldsymbol{\chi}_k\} \ \cdots \ b_m \{\boldsymbol{\chi}_k\}]^{\mathrm{T}}$$

and $\mathbf{w}_k^{\mathrm{T}} = [y_{k-1} \ \cdots \ -y_{k-n} \ v_{k-1} \ \cdots \ v_{k-m}]$. Here v_k and y_k are the input voltage and joint angle respectively, while $a_i \{\boldsymbol{\chi}_k\}$ and $b_j \{\boldsymbol{\chi}_k\}$ are n and m state dependent parameters i.e. functions of a non-minimal state vector $\boldsymbol{\chi}_k$. The hydraulic manipulator model is identified in three stages:

Step 1. Open-loop step experiments using the manipulator suggest that a first order linear difference equation, i.e.,

$$y_k = -a_1 y_{k-1} + b_\tau v_{k-\tau} \tag{1}$$

provides an approximate representation of individual joints, with the time delay τ depending on the sampling interval Δt.

Step 2. The values of $\{a_1, b_\tau\}$ are not repeatable for experiments with different input magnitudes. However, SDP analysis of experimental data suggests that $n = m = 1$, $a_1 = -1$ is time invariant and $b_\tau \{\boldsymbol{\chi}_k\} = b_\tau \{v_{k-\tau}\}$ is a static nonlinear function [23].
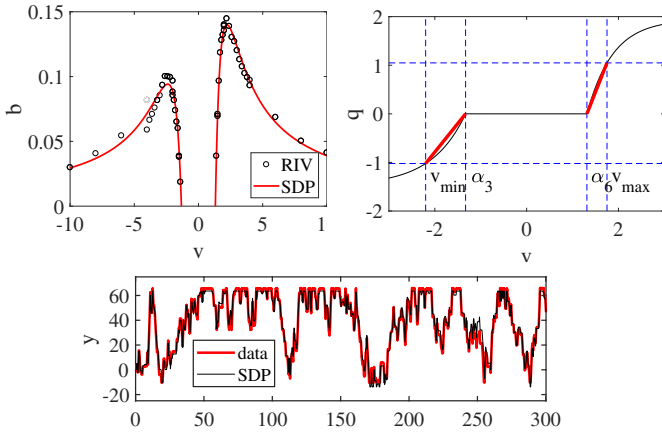
Fig. 4. Top left: state-dependent parameter plotted against input magnitude (i.e. steady state voltage to NI-CFP, v_∞), showing RIV estimates b_τ for individual experiments (circles) and optimised SDP $b_\tau\{\chi_k\}$ (solid). Top right: IDZ approximation, showing angular velocity q_k (°/Δt_c, $\Delta t_c = 0.05s$) against input i.e. $b_\tau\{\chi_k\} \times v_\infty/\Delta t_c$ (thin solid) and linearised model (thick red trace); dead-zone and velocity limits are highlighted (dashed). Lower: illustrative SDP model evaluation, showing $y_k(°)$ against time (s).
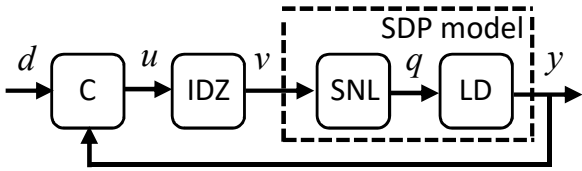

Fig. 5. Schematic diagram of the control system, in which C, IDZ, SNL and LD represent the (linear) Controller, Inverse Dead-Zone control element, Static NonLinearity and Linear Dynamics, respectively.

Step 3. Finally, $b_\tau\{v_{k-\tau}\}$ is parameterized, here using exponential functions for the nonlinearity. To illustrate using the right hand side 'shoulder' joint, denoted J2, with $\Delta t = 0.01s$ and $\tau = 15$, RIV estimates b_τ for step input experiments of varying magnitudes are shown in Fig. 4, demonstrating the clear state-dependency on the input. The mechanistic interpretation is straightforward: using equation (1) with $a_1 = -1$, yields $b_\tau v_{k-\tau} = y_k - y_{k-1}$ i.e. $b_\tau v_{k-\tau}$ represents a smoothed estimate of the differenced sampled output and $b_\tau v_{k-\tau}/\Delta t$ provides an estimate of the angular velocity, hence identifying the dead-zone and velocity saturation limits. As a result, denoting $q\{v_k\} = b_\tau\{v_k\} \times v_k$, the SDP model for J2 is,

$$y_k = y_{k-1} + q\{v_{k-\tau}\} \tag{2}$$

with constraints $y_{min} < y_k < y_{max}$, where $y_{min} = -13.8°$ and $y_{max} = 63.6°$ are hardware limits and,

$$
\begin{aligned}
q\{v_k\} &= (1-\alpha_1)e^{\alpha_2(\alpha_3 - v_k)} \quad \text{for} \quad v_k < \alpha_3 \\
q\{v_k\} &= 0 \quad \text{for} \quad \alpha_3 \le v_k \le \alpha_6 \\
q\{v_k\} &= (1-\alpha_4)e^{\alpha_5(v_k - \alpha_6)} \quad \text{for} \quad v_k > \alpha_6
\end{aligned} \tag{3}
$$

For J2, $\hat{\alpha}_i = [-0.295, 1.356, -1.330, 0.389, 1.771, 1.312]$. To obtain these estimates, experimental data are compared with the model response and the mean sum of the least squares output errors are used as the objective function for

fminsearch in MATLAB. Here, (α_2, α_5) are curve coefficients; $\alpha_1/\Delta t = -29.5°/s$ and $\alpha_4/\Delta t = 38.9°/s$ provide minimum and maximum angular velocity saturation limits; and $\alpha_3 = -1.33V$ to $\alpha_6 = 1.31V$ is the dead-zone. These objective estimates compare closely with equivalent values obtained from extensive prior *ad hoc* experimental work.

Similar SDP model forms are identified for each joint of both manipulators. In the case of Joints 1–3 (azimuth yaw, shoulder pitch and elbow pitch), these yield satisfactory evaluation results (Fig. 4), including for experiments involving simultaneous movement of all joints. However, the two joints closest to the end-effector (Joints 4–5: forearm roll and wrist pitch), yield inconsistencies in the time-delay and this is the subject of on-going research. For the present article, a slightly different parameterisation is utilised for J4: see [25].

Fig. 5 shows the control system, in which d_k is the desired joint angle, u_k is the PIP control input, v_k is the voltage to the NI-CPF computer, q_k is the angular velocity and y_k is the joint angle. Whilst previous research has utilised the SDP model for nonlinear pole assignment [2], the present article develops a rather simpler IDZ approach. Selecting a control sampling rate $\Delta t_c = 0.05s$ as a compromise between satisfactory reaction times and a relatively low order control system, Fig. 4 shows the linearised relationships for v_k and q_k in the range (i) v_{min} to $\hat{\alpha}_3$ and (ii) $\hat{\alpha}_6$ to v_{max} in the negative and positive directions of movement, respectively. For J2, $v_{min} = -2.2$ and $v_{max} = 1.75$, selected since they yield an angular velocity $\approx 20°/s$ in either direction, and it seems unlikely that faster movement would be desirable in practice. This approximation of the SNL element of Fig. 5 is defined as follows (cf. equations (3)),

$$
\begin{aligned}
q\{v_k\} &= s_n(v_k - \hat{\alpha}_3) \quad \text{for} \quad v_k < \hat{\alpha}_3 \\
q\{v_k\} &= 0 \quad \text{for} \quad \hat{\alpha}_3 \le u_k \le \hat{\alpha}_6 \\
q\{v_k\} &= s_p(v_k - \hat{\alpha}_6) \quad \text{for} \quad v_k > \hat{\alpha}_6
\end{aligned} \tag{4}
$$

where $s_n = q\{v_{min}\}/(v_{min} - \hat{\alpha}_3)$ and $s_p = q\{v_{max}\}/(v_{max} - \hat{\alpha}_6)$. Adapting from e.g. [21], the IDZ control element is,

$$
\begin{aligned}
v_k &= u_k/s_n + \hat{\alpha}_3 \quad \text{for} \quad u_k < -\beta \\
v_k &= 0 \quad \text{for} \quad -\beta \le u_k \le \beta \\
v_k &= u_k/s_p + \hat{\alpha}_6 \quad \text{for} \quad u_k > \beta
\end{aligned} \tag{5}
$$

where $\beta = 0.05$ is a 'chatter' coefficient. Eqns. (5) aim to cancel the dead-zone and allow use of linear control methods. For J2 with $\Delta t_c = 0.05$, $\tau = 3$ and hence, using standard methods [22], based on the model (1) with $b_3 = -a_1 = 1$, the PIP control algorithm takes the following incremental form,

$$
\begin{aligned}
u_k = u_{k-1} &- g_1(u_{k-1} - u_{k-2}) - g_2(u_{k-2} - u_{k-3}) \\
&- f_0(y_k - y_{k-1}) + k_I(d_k - y_k)
\end{aligned} \tag{6}
$$

with constraints $u_{min} < u_k < u_{max}$, in which $u_{min} = -2.5$ and $u_{max} = 2.5$ are introduced to avoid potential integral-wind up problems, and the control gains g_1, g_2, f_0 and k_I are determined by e.g. conventional Linear Quadratic (LQ) optimisation. Finally, two standard implementation forms of the PIP controller are investigated below, a feedback and forward path form [22].

Inexperienced operator		Experienced operator	
Teleoperation	Visual servoing	Teleoperation	Visual servoing
122	31	57	20
Failed	22	64	21
148	27	68	20

V. RESULTS

A cardboard tube is used as the target pipe for the pre-liminary experiments, so that if anything went wrong the manipulators would knock it out of the way without causing damage to the robot or laboratory. The operator selects the tube, and the manipulator grasps and moves as though cutting through it (presaging the use of appropriate laser cutting tools in future research). Tests were successfully repeated from various random starting positions. Since the new system actuates all the joints in parallel, significantly faster task completion times are acheived, in comparison to conventional teleoperation via joystick control, as illustrated by Table I. Here, an operator uses the joystick to grasp the tube three times. The same user subsequently operates the new vision system three times. An inexperienced user received just 15 minutes training with the joysticks, whilst an experienced user was already familiar with the system. These initial tests are clearly quite limited in scope. Nonetheless, the results provide an indication of the potential for improved performance using the developed approach, compared to the type of teleoperated system presently used on nuclear sites.

With regard to the new control system, Table II compares the feedback and forward path PIP-IDZ designs, together with two pre-existing control systems, namely a linear PIP controller and a PI algorithm that had been tuned by hand to obtain a reasonable response for the preliminary vision system testing. For the latter two cases, instead of IDZ control, the dead-zone was addressed using a rather *ac hoc* input scaling approach [2]. It is clear that the forward path PIP-IDZ controller yields the best performance. Fig. 6 illustrates the smooth, relatively fast closed-loop response of the joint angles for this controller. It also demonstrates the accuracy of the SDP simulation model. In this example, four manipulator joints are simultaneously moved in the form of a sinusoid signal, where the latter have been designed to approximately activate the full robot joint space. The joint angles and input to the CFP computer for the experimental data and SDP model are almost visually indistinguishable in this example. It should be noted that Table II is based on this type of sinusoid experiment (i.e. simultaneous motion of four manipulator joints) but similar results have also been obtained for various step changes in the set point and for pipe grasp scenarios [25]. Finally, Fig. 7 shows an example of the manipulator moving to a target grasp location. It shows small movements around the initial target position, before proceeding to the final grasp. The time taken to complete the movement shown in Fig. 7 is 5.4 seconds.

Performance	PIP-IDZ-FP	PIP-IDZ-FB	PIP-Old	PI-old
Ave.Eucl.Norm	**77**	132	190	160
MSE–J1	**5.7**	17.8	56.8	20.5
MSE–J2	**1.9**	3.5	12.8	8.0
MSE–J3	**5.6**	12.0	64.0	81.9
MSE–J4	85.4	**56.7**	77.1	144.8
Var(E)–X	**7.34**	22.9	408	44.7
Var(E)–Y	**17.5**	89.4	158	132
Var(E)–Z	**48.5**	185.7	210	112
Var(CFP)–J1	**0.02**	0.38	0.03	0.03
Var(CFP)–J2	**0.03**	0.27	0.11	0.05
Var(CFP)–J3	0.27	0.49	**0.04**	0.05
Var(CFP)–J4	0.23	0.30	0.19	**0.17**

VI. CONCLUSIONS

This article has developed and evaluated a vision based semi-autonomous object grasping system for a hydraulically actuated, dual manipulator nuclear decommissioning robot. The system presents a straightforward GUI to the operator, who with just four mouse clicks can select target positions for each manipulator to perform a pipe grasp and cut action. Throughout the process, the user can view the live colour video and terminate manipulator movements at any time. The system was tested on a full scale hardware system in a laboratory environment. It is shown to work successfully, outperforming the traditional joystick-based teleoperation approach. The system keeps the user in control of the overall system behaviour but significantly reduces user workload and operation time.

Further research is required to support these conclusions, including additional experiments with more users, and particularly in relation to attaching suitable cutters and working with real pipes and other objects. Future experiments include using both manipulators to grasp the same object, in order to move particularly heavy objects. In addition, one limitation of the present prototype is the lack of a sophisticated collision avoidance system i.e. to prevent the manipulators colliding with each other or the work environment.

In a second contribution of the article, a new approach to SDP modelling for manipulators has been described and evaluated. The models obtained facilitate use of an IDZ method for control system design. This is rather simpler to implement than an earlier SDP approach and has the advantage of combining the input signal calibration, system identification and control system design steps, all based on a relatively small data-set. This allows for rapid application to each joint and straightforward recalibration when the dynamic characteristics have changed due to age and use. Resolved motion tests successfully demonstrate the efficacy of the new control system.
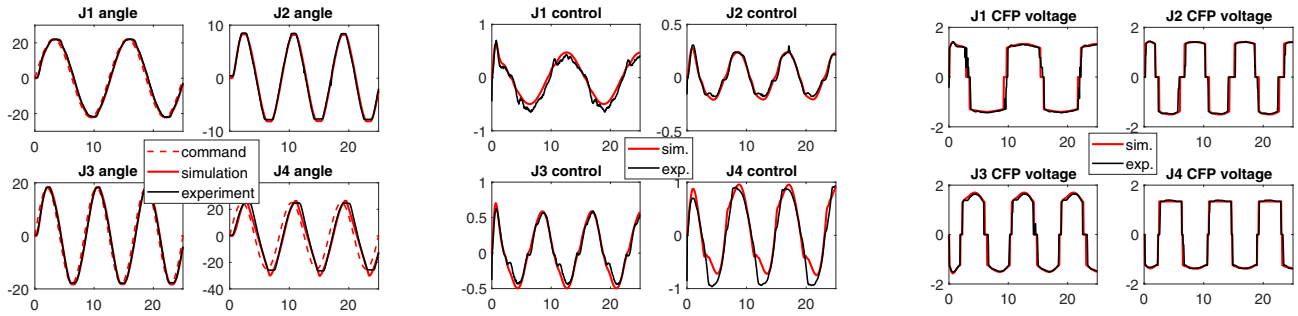
Fig. 6. Control diagnostic resolved motion experiment based on sine wave joint angle trajectories, showing command (dashed), HYDROLEK data (solid) and SDP simulation (thick, red). From left to right, 4×4 subplots show joint angle y_k (°), control input u_k and NI-CFP voltage v_k, against time (s).
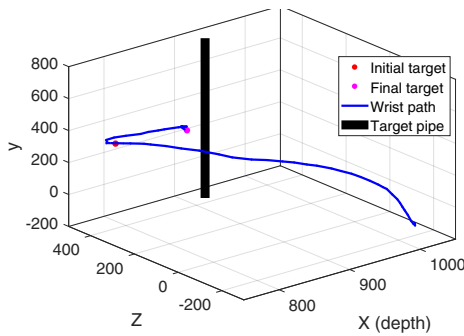

Fig. 7. Illustrative resolved motion experiment for a pipe grasp (units: mm).

REFERENCES

[1] G. D. Baxter, J. Rooksby, Y. Wang, and A. Khajeh-Hosseini, "The ironies of automation... still going strong at 30?" in *European Conference on Cognitive Ergonomics*, 2012.

[2] C. J. Taylor and D. Robertson, "State–dependent control of a hydraulically–actuated nuclear decommissioning robot," *Control Engineering Practice*, vol. 21, no. 12, pp. 1716–1725, 2013.

[3] A. Kitamura, M. Watahiki, and K. Kashiro, "Remote glovebox size reduction in glovebox dismantling facility," *Nuclear Engineering and Design*, vol. 241, pp. 999–1005, 2011.

[4] J. Kofman, X. Wu, and T. J. Luu, "Teleoperation of a robot manipulator using a vision–based human–robot interface," *IEEE Transactions on Industrial Electronics*, vol. 52, pp. 1206–1219, 2005.

[5] A. Leeper, K. Hsiao, and M. Ciocarlie, "Strategies for human-in-the-loop robotic grasping," in *7th ACM/IEEE International Conference on Human–Robot Interaction*, 2012.

[6] K. Hauser, "Recognition, prediction, and planning for assisted teleoperation," *Autonomous Robots*, vol. 35, pp. 241–254, 2013.

[7] N. Marturi, A. Rastegarpanah, C. Takahashi, M. Adjigble, R. Stolkin, S. Zurek, M. Kopicki, M. Talha, J. A. Kuo, and Y. Bekiroglu, "Towards advanced robotic manipulation for nuclear decommissioning: A pilot study on tele-operation and autonomy," in *IEEE Robotics & Automation for Humanitarian App.*, Kollam, 2016.

[8] X. Duan, Y. Wang, Q. Liu, M. Li, and X. Kong, "Manipulation robot system based on visual guidance for sealing blocking plate of steam generator," *J. Nuclear Science & Technology*, vol. 53, pp. 281–288, 2016.

[9] D. Katz, J. Kenney, and O. Brock, "How can robots succeed in unstructured environments?" in *Workshop on Robot Manipulation: Intelligence in Human Environments at Robotics*, Cambridge, UK, 2008.

[10] D. Kent, C. Saldanha, and S. Chernova, "A comparison of remote robot teleoperation interfaces for general object manipulation," in *ACM/IEEE International Conference on Human–Robot Interaction*, Vienna, 2017.

[11] M. M. Olsen, J. Swevers, and W. Verdonck, "ML identification of a dynamic robot model: Implementation issues," *Int. J. Robotics Research*, vol. 21, pp. 9–96, 2001.

[12] M. Gautier and P. Poignet, "Extended Kalman filtering and weighted LS dynamic identification of robot," *Control Engineering Practice*, vol. 9, pp. 1361–1372, 2001.

[13] M. Gautier, A. Janot, and P. O. Vandanjon, "A new closed–loop output error method for parameter identification of robot dynamics," *IEEE Transactions on Control SystemTechnology*, vol. 21, pp. 428–444, 2013.

[14] P. C. Young, *Recursive Estimation and Time Series Analysis: An Introduction for the Student and Practitioner*. Springer, 2011.

[15] A. Janot, P. O. Vandanjon, and M. Gautier, "An instrumental variable approach for rigid industrial robots identification." *Control Engineering Practice*, vol. 25, pp. 85–101, 2014.

[16] S. Hashemi, H. Abbas, and H. Werner, "Low–complexity LPV modeling and control of a robotic manipulator," *Control Engineering Practice*, vol. 12, pp. 248–257, 2012.

[17] C. J. Taylor, E. M. Shaban, M. A. Stables, and S. Ako, "PIP control applications of state dependent parameter models," *IMechE Proceedings*, vol. 221, no. 17, pp. 1019–1032, 2007.

[18] A. Janot, P. C. Young, and M. Gautier, "Identification and control of electro–mechanical systems using state–dependent parameter estimation," *International Journal of Control*, vol. 90, pp. 643–660, 2017.

[19] M. R. Sirouspour and S. E. Salcudean, "Nonlinear control of hydraulic robots," *IEEE Trans. Robotics & Aut.*, vol. 17, pp. 173–182, 2001.

[20] A. Montazeri, C. West, S. Monk, and C. J. Taylor, "Dynamic modeling and parameter estimation of a hydraulic robot manipulator using a multi–objective genetic algorithm," *Int. J. Control*, vol. 90, pp. 661–683, 2017.

[21] J. D. Fortgang, L. E. George, and W. J. Book, "Practical implementation of a dead zone inverse on a hydraulic wrist," in *ASME International Mechanical Engineering Congress and Exposition*, New Orleans, 2002.

[22] C. J. Taylor, P. C. Young, and A. Chotai, *True Digital Control: Statistical Modelling and Non–Minimal State Space Design*. Wiley, 2013.

[23] C. West, E. D. Wilson, Q. Clairon, A. Montazeri, S. Monk, and C. J. Taylor, "State–dependent parameter model identification for inverse dead–zone control of a hydraulic manipulator," in *18th IFAC Symposium on System Identification (SYSID)*, Stockholm, Sweden, July 2018.

[24] H. Kinoshita, R. Tayama, Y. Kometani, T. Asano, and Y. Kani, "Development of new technology for Fukushima Daiichi nuclear power station reconstruction," *Hitachi Hyoron*, vol. 95, pp. 41–46, 2013.

[25] C. West, "A vision-based pipe cutting system," Ph.D. dissertation, Engineering Department, Lancaster University, 2018 (in preparation).

[26] J. Canny, "A computational approach to edge detection. ieee transactions on pattern analysis and machine intelligence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 679–698, 1986.

[27] R. Maini and H. Aggarwal, "Study and comparison of various image edge detection techniques," *Int. J. Image Proc.*, vol. 3, pp. 1–11, 2009.

[28] T. Burrell, A. Montazeri, S. Monk, and C. J. Taylor, "Feedback control–based IK solvers for a nuclear decommissioning robot," in *7th IFAC Symp. Mechatronic Systems*, Loughborough, UK, September 2016.

[29] C. J. Taylor, P. C. Young, W. Tych, and E. D. Wilson, "New developments in the CAPTAIN Toolbox for Matlab," in *18th IFAC Symposium on System Identification (SYSID)*, Stockholm, Sweden, July 2018.