

MULTI-ARM MOBILE ROBOT FOR HAZARDOUS NUCLEAR DECOMMISSIONING TASKS

Mohamed J. Bakari
Engineering Department
Lancaster University
Bailrigg, Lancaster LA1 4YR, UK
m.bakari@lancaster.ac.uk

E. M. Shaban
Engineering Department
Lancaster University
Bailrigg, Lancaster LA1 4YR, UK
e.e.shaban@lancaster.ac.uk

Professor Derek W. Seward
Engineering Department
Lancaster University
Bailrigg, Lancaster LA1 4YR, UK
d.seward@lancaster.ac.uk

Rahee Y. Agate
Engineering Department
Lancaster University
Bailrigg, Lancaster LA1 4YR, UK
r.agate@lancaster.ac.uk

Abstract: This paper outlines the selection of hardware and its integration within a mobile delivery platform for application within nuclear decommissioning tasks; it is based on current research being undertaken at Lancaster University, in the United Kingdom (U.K.). The mobile platform is a standard Brokk machine consisting of a moving vehicle and a manipulator with five rotary joints. The human arm maneuverability, scale and dexterity are adopted as the starting point for maneuvering the two six degree-of-freedom (dof) manipulators to perform cooperative manipulation tasks and address the complex tasks found in the rapidly expanding field of nuclear decommissioning. Within the paper the forward and inverse kinematics for the six dof manipulators are outlined. The paper will also describe the proposed software architecture for the performance of decommissioning tasks. This paper builds upon the ideas presented at the ISARC 2005 conference.

Keywords: Nuclear Decommissioning, Hardware Integration, Kinematics, Software Architecture.

1. INTRODUCTION

The Nuclear Decommissioning Authority (NDA) in the UK [16] is under pressure from the UK government to decontaminate and decommission (D&D) a high number of ageing nuclear power stations. These nuclear facilities often contain radioactive and other hazardous material. Much of the decommissioning process utilizes well established demolition techniques although there are complications within this process due to the risk of exposure of workers to radiation. Current technologies are often labour intensive, time consuming, expensive, or unnecessarily expose workers to the hazardous material. The NDA and the UK government are looking for new and innovative technologies that will allow D&D operations in the UK to be faster, safer, and more cost-effective. Robotic systems provide a potential solution to these problems because the robots can be remotely controlled from an operator console and provide visual feedback to the operator while executing the decommissioning tasks; thus reducing the radioactive dose levels to which workers are exposed.

2. BACKGROUND

The research currently being undertaken at Lancaster University aims to develop a multi-arm robot system based on human-scale size manipulators which can be deployed by a remote vehicle. The focus of the research is to undertake pipe-cutting decommissioning tasks; reconfigure the system

for other necessary decommission tasks such as unscrewing and dismantling [4]; and to use the system in small restricted spaces that are too dangerous for humans, or in which a bigger manipulator could not be used or deployed.

2.1 Manipulator requirement

Robot manipulators must fulfill specific requirements in order to undertake decommissioning tasks [2]. The operational features that a multi-arm robot system (such as that shown in Fig.1) needs to have are human scale reach and dexterity, obstacle and collision avoidance, force control, cooperative manipulator ability, remote tele-operated control and high reliability.

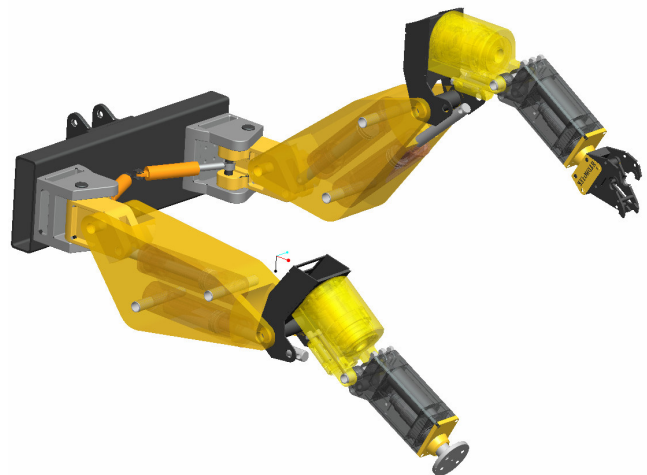


Fig. 1. Multi-Arm Robot System



Fig. 2. Brokk 40 mobile delivery robot [1]

2.2 Hardware Integration

The mobile delivery platform adopted for this research is Brokk 40 mobile robot, as shown in Fig. 2, it consists of a moving vehicle with a five degree of freedom manipulator, five linear actuators, hydraulic tank, controller and remote control device.

Two Hydro-Lek robot manipulators will be attached to the Brokk friendly mount point and their hydraulic actuators will be integrated and operated by the Brokk hydraulic system. The left arm is a complete arm with a gripper and the right arm has no gripper but a simple flange where a cutting tool will be fixed.

The purpose of developing the multi-arm robot in this research work is for the arms to have the ability to perform two distinct operations simultaneously or separately.

3. KINEMATICS

Robot manipulators can be considered as a set of bodies or links connected in a kinematic chain by joints [13]. The Hydro-Lek robot arm is a six degree of freedom robot manipulator.

Fig.3 shows the six rotary joints of the Hydro-Lek manipulator [9]. Joint one rotates with axis perpendicular to plane XY. Joint two rotates perpendicular to joint one. Joint three rotates parallel to joint two and is offset by the link indicated as A. Joint four is perpendicular to joint three and is offset by the link indicated as B. Joint five is perpendicular to joint four, parallel to joint three and is offset by the link indicated as C. Joint six is perpendicular to joint five and is offset by the link indicated as D.

The Hydro-Lek manipulator structure is kinematically defined by giving each link four parameters which are d_i , a_i , θ_i and α_i . The four given parameters shown in Table 1, describes how to get from one joint to another. Neighboring links have a common joint axis between them. The distance along the common axis from one link to the next link is offset d_i . The amount of rotation about the common axis between one link and its neighbor is joint angle θ_i . The definition of mechanisms by means of these four parameters is a convention called Denavit-Hartenberg (D-H) [3], [8]. The location and orientation of each joint frame is shown in Fig.3.

Table 1: Hydro-Lek D-H parameters

Joint	θ_i	α_i	a_i	d_i
1	θ_1	90^0	a_1	d_1
2	θ_2	0	a_2	0
3	$\theta_3 - 90^0$	90^0	$-a_3$	0
4	θ_4	90^0	a_4	$-d_4$
5	θ_5	90^0	$-a_5$	0
6	$\theta_6 + 90^0$	180^0	0	d_6
6	Rotation around \mathbf{x}_6 by 90^0 followed by rotation -90^0 around rotated \mathbf{z}_6 .			

The constant D-H parameters for the Hydro-Lek arm as shown in Table 1 are the twist angles α_{i-1} , the link lengths a_{i-1} , the link offsets d_i and the variable joint angles θ_i .

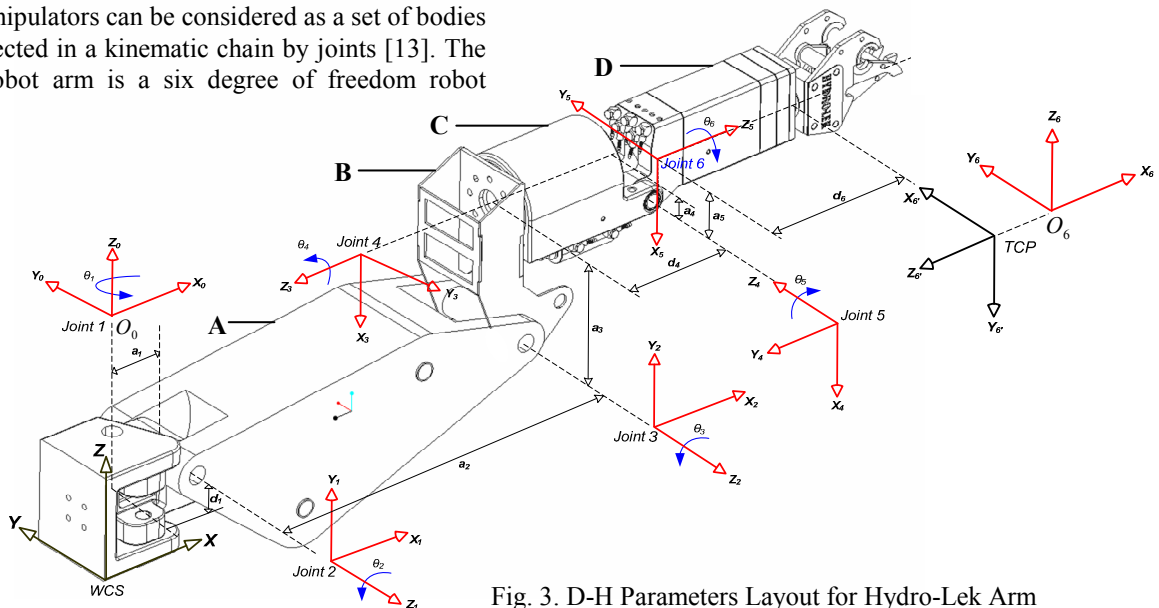


Fig. 3. D-H Parameters Layout for Hydro-Lek Arm

3.1 Forward Kinematics

Given a set of joint angles, the forward kinematics problem is simply to compute the position and orientation of the tool-tip frame relative to the base frame. Link transformations are computed as follows using homogeneous transformation.

Joint One – Azimuth Yaw

$$\mathbf{T}_0^1 = \begin{bmatrix} c_1 & 0 & s_1 & a_1 c_1 \\ s_1 & 0 & -c_1 & a_1 s_1 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Joint Two – Shoulder Pitch

$$\mathbf{T}_1^2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Joint Three – Elbow Pitch

$$\mathbf{T}_2^3 = \begin{bmatrix} s_3 & 0 & -c_3 & -a_3 s_3 \\ -c_3 & 0 & -s_3 & a_3 c_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Joint Four – Forearm Roll

$$\mathbf{T}_3^4 = \begin{bmatrix} c_4 & 0 & s_4 & a_4 c_4 \\ s_4 & 0 & -c_4 & a_4 s_4 \\ 0 & 1 & 0 & -d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Joint Five – Wrist Pitch

$$\mathbf{T}_4^5 = \begin{bmatrix} c_5 & 0 & s_5 & -a_5 c_5 \\ s_5 & 0 & -c_5 & -a_5 s_5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Joint Six – Wrist Roll

$$\mathbf{T}_5^6 = \mathbf{T}_5^{6'} \mathbf{T}_{6'}^6 = \begin{bmatrix} -s_6 & c_6 & 0 & 0 \\ c_6 & s_6 & 0 & 0 \\ 0 & 0 & -1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -s_6 & -c_6 & 0 \\ 0 & c_6 & -s_6 & 0 \\ 1 & 0 & 0 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(6)

Where $c_i = \cos \theta_i$ and $s_i = \sin \theta_i$

The product of all six-link transform matrices leads to

$$\mathbf{T}_0^6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

(1) For which $r_{ij} \forall 1 \leq i, j \leq 3$ represent the elements of the orientation matrix \mathbf{R}_0^6 and the vector $\mathbf{p} = [p_x \ p_y \ p_z]^T$ represents the position of the end-effector with respect to the base coordinate frame. In this regard,

$$\begin{aligned} r_{11} &= c_1 s_{23} c_4 s_5 + c_1 c_{23} c_5 + s_1 s_4 s_5, \\ r_{21} &= s_1 s_{23} c_4 s_5 + s_1 c_{23} c_5 - c_1 s_4 s_5, \\ r_{31} &= s_{23} c_5 - c_{23} c_4 s_5, \\ r_{12} &= c_1 s_{23} (s_4 c_6 - c_4 c_5 s_6) - s_1 (s_4 c_5 s_6 + c_4 c_6) \\ &\quad + c_1 c_{23} s_5 s_6 + c_1 c_{23} s_5 s_6, \end{aligned}$$

$$(3) \quad \begin{aligned} r_{22} &= s_1 s_{23} (s_4 c_6 - c_4 c_5 s_6) + c_1 (s_4 c_5 s_6 + c_4 c_6) \\ &\quad + s_1 c_{23} s_5 s_6, \\ r_{32} &= s_{23} s_5 s_6 - c_{23} (s_4 c_6 - c_4 c_5 s_6), \end{aligned}$$

$$\begin{aligned} r_{13} &= -c_1 s_{23} (s_4 s_6 + c_4 c_5 c_6) - s_1 (s_4 c_5 c_6 - c_4 s_6) \\ &\quad + c_1 c_{23} s_5 c_6, \\ r_{23} &= -s_1 s_{23} (c_4 c_5 c_6 + s_4 s_6) + c_1 (s_4 c_5 c_6 - c_4 s_6) \\ &\quad + s_1 c_{23} s_5 c_6, \end{aligned}$$

$$\begin{aligned} r_{33} &= c_{23} (c_4 c_5 c_6 + s_4 s_6) + s_{23} s_5 c_6, \\ p_x &= c_1 \left[s_{23} (a_4 c_4 - a_5 c_4 c_5 + d_6 c_4 s_5) \right. \\ &\quad \left. + c_{23} (d_4 + a_5 s_5 + d_6 c_5) + a_1 + a_2 c_2 - a_3 s_{23} \right] \\ &\quad + s_1 (a_4 s_4 - a_5 s_4 c_5 + d_6 s_4 s_5), \end{aligned}$$

$$\begin{aligned} p_y &= s_1 \left[s_{23} (a_4 c_4 - a_5 c_4 c_5 + d_6 c_4 s_5) \right. \\ &\quad \left. + c_{23} (d_4 + a_5 s_5 + d_6 c_5) + a_1 + a_2 c_2 - a_3 s_{23} \right] \\ &\quad - c_1 (a_4 s_4 - a_5 s_4 c_5 + d_6 s_4 s_5), \\ p_z &= -c_{23} (a_4 c_4 - a_5 c_4 c_5 + d_6 c_4 s_5) \\ &\quad + s_{23} (d_4 + a_5 s_5 + d_6 c_5) + d_1 + a_2 s_2 + a_3 c_{23}. \end{aligned}$$

(8)

The above equations specify how to compute the position and orientation of the end-effector frame O_6 relative to the arm base frame O_0 .

4. INVERSE KINEMATICS - HIGH LEVEL CONTROL

Given the position of some points of the end-effector, the inverse kinematics (IK) is to compute the angles of each joint of the robot arm. Solving the IK problem of a robot arm using a closed-form analytic solution is preferable and has advantages over a numerical solution, however, a closed-form analytic solution for the Hydro-Lek arm is not possible although a numerical solution does exist. For this research we have applied the Jacobian pseudo-inverse method [12], [13], [14] as presented below, because the Jacobian matrix of the arm under study is in need of velocity control. The singularities that arise by adopting this approach are currently being studied by the authors in order to prove the validity of the approach throughout the whole working space of the Hydro-Lek arm.

4.1 Jacobian matrix for the Hydro-Lek arm

Jacobian matrix $\mathbf{J}(q)$ is the transformation from end-effector velocity vector $\begin{bmatrix} \dot{\mathbf{p}}^T & \boldsymbol{\omega}^T \end{bmatrix}^T$ to joints velocity vector $\dot{\mathbf{q}}$. Since, the generalized Cartesian velocity vector of the end-effector is composed of two sub-vectors $\dot{\mathbf{p}}$ and $\boldsymbol{\omega}$, the Jacobian matrix may be partitioned into linear and orientation parts by writing,

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(q) \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_p(q) \\ \mathbf{J}_\omega(q) \end{bmatrix} \dot{\mathbf{q}} \quad (9)$$

For which $\dot{\mathbf{p}} = [\dot{p}_x \ \dot{p}_y \ \dot{p}_z]^T$ represents the resolved linear velocity of the tool-tip and $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$ represents the angular velocity. Here, the linear position Jacobian $\mathbf{J}_p(q)$ represents the first three rows of $\mathbf{J}(q)$ and the angular Jacobian $\mathbf{J}_\omega(q)$ its last three rows. Thus, the arm Jacobian $\mathbf{J}(q)$ is $6 \times n$ matrix, with n the number of joints in the manipulator. In this case, $n = 6$, therefore the Jacobian of the Hydro-Lek manipulator is square.

The computation of the linear position Jacobian can be obtained using the following relation,

$$\mathbf{J}_p(q) = \begin{bmatrix} \frac{\partial \mathbf{p}}{\partial q_1} & \frac{\partial \mathbf{p}}{\partial q_2} & \dots & \frac{\partial \mathbf{p}}{\partial q_n} \end{bmatrix} \quad (10)$$

In order to evaluate the orientation part of the arm Jacobian, consider the following chaining operation,

$$\mathbf{T}_0^i = \mathbf{T}_0^1 \mathbf{T}_1^2 \dots \mathbf{T}_{i-1}^i = \begin{bmatrix} \mathbf{R}_0^i & \mathbf{p}_0^i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

Here \mathbf{R}_0^i denotes the rotation matrix of frame i with respect to the base frame, for which

$$\mathbf{R}_0^i = [\mathbf{x}_i \ \mathbf{y}_i \ \mathbf{z}_i] \quad (12)$$

The vectors \mathbf{x}_i , \mathbf{y}_i and \mathbf{z}_i represent the x, y and z-axis of frame i in base coordinate. Since all angular velocities are represented in the same coordinate frame, i.e. the joint rotation $q_i = \theta_i$ occurs about joint axis \mathbf{z}_{i-1} , the angular velocity for joint variable i is therefore given by $\mathbf{z}_{i-1} \dot{q}_i$.

Therefore, the orientation part of the Jacobian matrix can be constructed, taking into account that the prismatic joints do not contribute to the angular velocity of the end-effector. Hence, the orientation part of the Jacobian matrix may be written as

$$\mathbf{J}_\omega(q) = [k_1 z_0 \ k_2 z_1 \ \dots \ k_n z_{n-1}] \quad (13)$$

where $k_i = 0$ if q_i is prismatic and $k_i = 1$ if q_i is revolute. Also the vector $\mathbf{z}_0 = [0 \ 0 \ 1]^T$

The implementation of inverse kinematics is based upon the well-established Jacobian technique. The objective of this technique is to incrementally change joint orientations from a stable starting position towards a configuration state that will result in the required end-effector being located at the desired position in work space. The amount of incremental change on each iteration is defined by the relationship between the partial derivatives of the joint angles

$\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \dots \ \theta_6]^T$ and the difference between the current location of the end-effector $\mathbf{X} = [\mathbf{p}^T \ \boldsymbol{\omega}^T]^T$ and

the desired position $\mathbf{X}_d = [\mathbf{p}_d^T \ \boldsymbol{\omega}_d^T]^T$. The link between these two sets of parameters leads to the following Jacobian system,

$$d\mathbf{X} = \mathbf{J}(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (14)$$

By rearranging equation (14),

$$d\boldsymbol{\theta} = \mathbf{J}(\boldsymbol{\theta})^{-1} d\mathbf{X} \quad (15)$$

This form transforms the nonlinear system of equation into a linear one that can be solved using iterative steps.

The problem now is that equation (15) requires the inversion of the Jacobian matrix which sometimes ill-conditioned, therefore it is preferable to use the right-hand generalized pseudo-inverse [14].

Generating the pseudo-inverse of the Jacobian can lead to inaccuracies in the resulting inverse that need to be reduced. Any inaccuracies of the inverse Jacobian can be detected by multiplying it with the original Jacobian then subtracting the result from the identity matrix. A magnitude error can be determined by taking the second norm of the resulting matrix multiplied by $d\mathbf{X}$, see equation (18) below. If the error become too big then $d\mathbf{X}$ can be decreased until the error falls within acceptable limit.

An overview of the algorithm using iterative inverse kinematics is as follows;

- 1- Calculate the difference between the goal position and the actual position of the end-effector:

$$d\mathbf{X} = \mathbf{X}_d - \mathbf{X} \quad (16)$$

- 2- Calculate the jacobian matrix using the current joint angles

- 3- Calculate the pseudo-inverse of the Jacobian

$$\mathbf{J}^+ = \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1} \quad (17)$$

- 4- Determine the error of the pseudo-inverse

$$\text{error} = \|(I - \mathbf{J} \mathbf{J}^+) d\mathbf{X}\| \quad (18)$$

- 5- If $\text{error} > e$ then

$$d\mathbf{X} = d\mathbf{X} / 2, \text{ go back to step 4} \quad (19)$$

- 6- Calculate the updated value of the joints angle vector $\boldsymbol{\theta}$ and use this as the new current value,

$$\boldsymbol{\theta} = \boldsymbol{\theta} + \mathbf{J}^+ d\mathbf{X} \quad (20)$$

- 7- Using the forward kinematics to determine whether the new joints angle vector locate the end-effector close enough to the desired location and orientation. If the solution is adequate then terminate the algorithm otherwise go back to step 1.

The computational demand of the algorithm is relatively high over a number of iterations. This phenomenon is under study by the authors.

5. SOFTWARE ARCHITECTURE

As robots become kinematically more complex and begin to face more challenging tasks, the algorithms controlling their motion must become correspondingly more sophisticated. The software must have the capacity to transform all encoded, force, current and other signals into generic digital information about the state of the robot, the controller or the interactive wishes of the operator; and be able to develop command signals to the active elements of both controller and the robot [10], [11].

The software programs proposed to control the multi-arm mobile robot within this research have been chosen because of their ability to address different robot configurations, advanced decision-making algorithms and real-time control, as outlined below.

5.1 National Instruments Operational Software

The National Instruments (NI) operational software programs are LabView, LabView Real-Time and SoftMotion [15].

The LabView development system provides a full-featured graphical programming tool for developing measurement, automation and control applications. It is an open system that allows user modification, extensibility, and integration with other systems. The LabView development system is reusable software that has a clear interface. The software architecture is general with no assumptions that may limit the future integration of any conceivable manipulator and decision-making criterion.

LabView Real-Time Module is real-time software which extends the capability of the LabView environment to create hard real-time, deterministic systems for solving a variety of application challenges.

The NI SoftMotion is add-on software and is used to implement advanced motion control design algorithms. The SoftMotion includes functions for trajectory generation, spline interpolation, position/velocity control, and encoder implementation on LabView Real-Time.

5.1 Compact FieldPoint Implementation

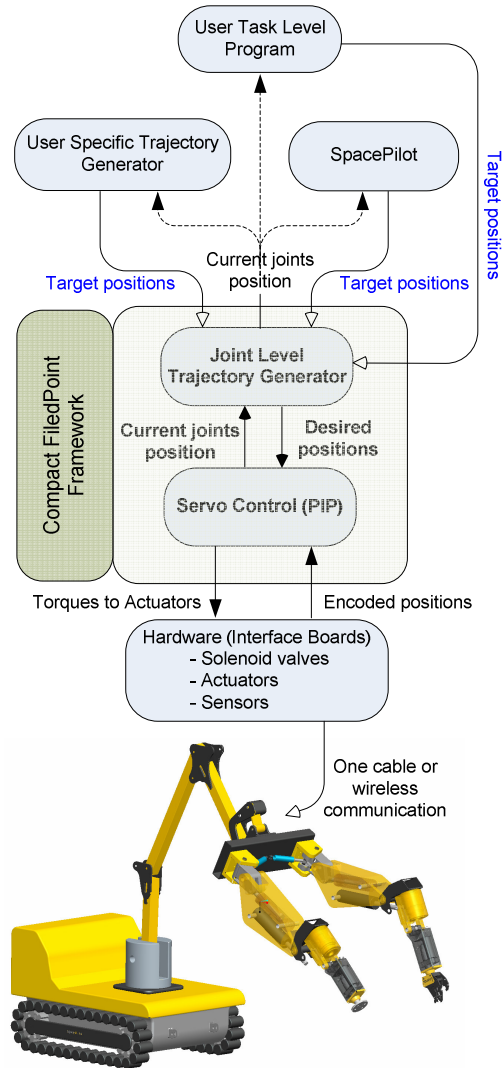


Fig. 4. Compact FieldPoint Controller

Compact FieldPoint hardware is a lightweight, compact, portable, adaptable and programmable automation controller (PAC) designed for industrial control applications performing advanced embedded control, data logging, and network connectivity [5]. The PAC as shown in Fig. 4 is a reliable platform designed for rugged industrial environments with shock, vibration, and temperature extremes. The PAC runs on LabView Real-Time where this real-time software was developed to transform signals from the controller into meaningful command signals to the robot manipulators, providing the functionality, connectivity, and flexibility of NI LabView software. A custom motion controller can be created by SoftMotion software when using Compact FieldPoint.

6. CONCLUSION

Earlier generations of nuclear facility have now been closed and many are waiting effective decommissioning. There is a multi-billion pound world-wide market for companies who have the skills and technology to engage with the decommissioning task, which gives incentive and creates opportunities for further examination of the use of automation and robotics in D&D tasks.

This paper has outlined the selection of hardware and its integration within a mobile delivery platform developed to address pipe-cutting tasks within the field of nuclear decommissioning. The paper has also discussed the proposed software architecture and the selection of suitable end-effectors. The forward kinematics and the high level control for the six dof manipulators have also been considered.

The next stage of this research is to set up the complete robotic system by integrating the selected hardware and end-effectors with the mobile delivery platform, and using the proposed software architecture for programming and implementing the pipe cutting decommissioning task.

7. ACKNOWLEDGMENTS

The authors are grateful for the support of the Engineering and Physical Science Research Council EPSRC.

REFERENCES

- [1] Brokk 40 robots information [online]. Available from: <http://www.brokk.com/> [accessed February 2006]
- [2] Cox, D., Cetin, M., Pryor, M., and Tesar, D., 1999. Requirements for Modular Dual-Arm Robot Architecture for Use in Deactivation and Decommissioning Tasks. *ANS 8th International Topical meeting on Robotics and Remote Systems*, 9(4), pp. 1-14, American Nuclear Society.
- [3] Cox, D., 2004. Cooperative manipulation Testbed Development-Kinematics. *17th Annual Florida Conference on the Recent Advances in Robotics (FCRAR)*, Florida.
- [4] Cox, D., 2002. Mock-up of Hazards Material Handling Taks Using a Dual-Arm Robotic System. *Proceedings 5th World Automation Congress*, Orlando.
- [5] Compact FieldPoint information [online]. Available from: <http://www.ni.com/compactfieldpoint/> [accessed April 2006]
- [6] D., E., Whitney., 1969. Resolved Motion Rate Control of Manipulators and Human Prostheses. *IEEE Transaction on Man-Machine Systems*, Vol. 10, pp. 47-53.
- [7] Gu, J., Taylor, C.J., and Seward, D., 2004. Proportional-Integral-Plus Control of an Intelligent Excavator. *Journal of Computer-Aided Civil and Infrastructure Engineering*, Vol. 19, pp. 16-27.
- [8] Hemami, A., 1986. Kinematics of Two-Arm Robots. *IEEE Journal of Robotics and Automation*, RA-2 (4), pp. 225-228.
- [9] Hydro-Lek Manipulator information [online]. Available from: <http://www.hydro-lek.com/> [accessed October 2005]
- [10] Kapoor, C., and Tesar, D., 1998. A Reusable Operational Software Architecture for Advanced Robotics. *Proceedings of the Twelfth CSIM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators*, Paris.
- [11] Kapoor, C., Cetin, M., Pryor, M., Cocca, C., Harden, T., and Tesar, D., 1998. A Software Architecture for Multi-Criteria Decision Making for Advanced Robotics. *Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference*, pp. 525-530. Gaithersburg.
- [12] Kapoor, C., and Tesar, D., 1999. Kinematic Abstractions for General Manipulator Control. *ASME Design Engineering Technical Conferences*, Las Vegas.
- [13] Lewis, F. L., Dawson, D. M., and Abdallah, C.T., 2004. *Robot manipulator Control Theory and Practice*, 2nd ed. New York: Marcel Dekker.
- [14] Meredith, M., and Maddock, S., 2004. Real-Time Inverse Kinematics: The Return of the Jacobian. *Department of Computer Science Research Memorandum*, CS-04-06, University of Sheffield.
- [15] National Instruments products information [online]. Available from: <http://www.ni.com/> [accessed January 2006]
- [16] Nuclear Decommissioning Agency (NDA) [online]. Available from: [http://www.nda.gov.uk/Home_\(1\).aspx?](http://www.nda.gov.uk/Home_(1).aspx?) [accessed April 2006]
- [17] Wolovich, W.A., and Elliot, H., 1984. A Computational Technique for Inverse Kinematics. *23rd Conference Proceedings on Decision and Control*, pp. 1359-1363.
- [18] Young, P.C., Chotai, A., and Tych, W., 1991. Identification, Estimation and True Digital Control of Glasshouse Systems. *Appears in Hashimoto et. (eds.) The computerized greenhouse*, Academic Press, pp. 3-50, London.
- [19] Zhao, J., and Badler, N.I., 1994. Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures. *ACM Transactions on Graphics*, Vol. 13, pp. 313-336.