UKAEA-RACE-PR(24)01

S. Herschmann, F. Milella, M. Torrance, S. Wainwright, O Crofts

Investigation into optimal feedforward control for remote handling of tokamak breeding blankets

Investigation into optimal feedforward control for remote handling of tokamak breeding blankets

S. Herschmann, F. Milella, M. Torrance, S. Wainwright, O Crofts

This is a preprint of a paper submitted for publication in
Fusion Engineering and Design

Investigation into optimal feedforward control for remote handling of tokamak breeding blankets

Sam Herschmann[a], Ferdinando Milella[a], Marc Torrance[a], Sam Wainwright[a], Oliver Crofts[a]

[a]*RACE, UK Atomic Energy Authority, Abingdon, OX14 3DB, Oxfordshire, United Kingdom*

Remote handling of breeding blankets poses an unprecedented challenge in future tokamaks like EU-DEMO, where individual blanket segments can weigh upwards of 80 tonnes and extend beyond 10 meters in length. The unparalleled scale of these components, coupled with extremely tight positional tolerances, demands careful consideration of structural flexibility during manoeuvres. This underscores the need for model-based control systems capable of mitigating oscillations and static deflections induced by gravity and disturbances.

In this paper, we present recent experiments towards this objective, focussed on mitigating the effects of gravity. We applied model-based, optimal feedforward control to the motion of a planar slender payload, chosen to represent the flexible part of a candidate breeding blanket segment. We combined a predictive path planning algorithm with input shaping techniques to manoeuvre the payload effectively accounting for the influence of gravity, while also reducing post-manoeuvre oscillations. Our results highlight the advantages of model-based control and the limitations associated with the use of feedforward control.

Keywords: Remote handling, model-based control, path planning, trajectory optimisation, breeding blankets.

1. Introduction

In-vessel remote handling (RH) of breeding blankets is recognised as a significant challenge in EU-DEMO [1]. Researchers have investigated a variety of RH approaches featuring both single mover and multiple mover systems. A single mover approach (such as the Hybrid Kinematic Manipulator concept [2], [3]) is illustrated in Fig. 1. This features a dexterous manipulator designed to manoeuvre the blankets. In EU-DEMO we anticipate many control-specific challenges due to tight spatial constraints, and the presence of gamma radiation that limits sensor placement. We also recognise the importance of understanding the flexure of the combined blanket-mover system in ensuring collision avoidance. This increases with the offset between the blanket centre of mass and the lifting point.

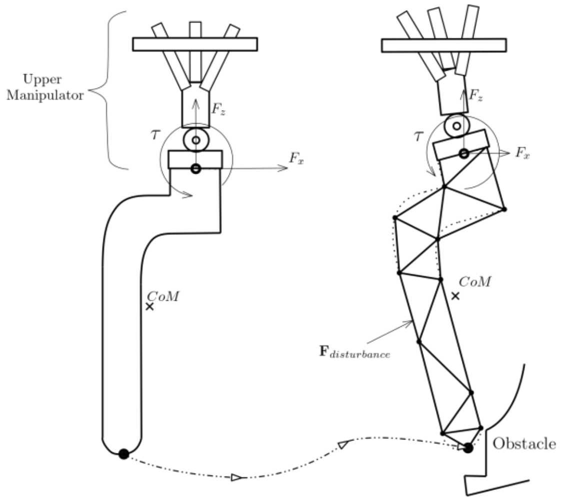


Fig. 1: Single mover manipulation of an in-board breeding blanket

Model-based optimal control uses knowledge of system properties in determining actuator commands to achieve a chosen objective. In robotics, a typical control objective is to minimise payload tracking error throughout a desired trajectory, and a model is typically a set of differential equations describing the dynamics of the system. Many techniques have been used to dynamically model and control flexible structures and manipulators [4]. Dynamic modelling approaches for flexible manipulators include the finite element approach, the assumed-modes method and the lumped parameter method [5], [6].

Optimal control approaches employ numerical optimisation to calculate and execute a best-case robot trajectory that accounts for obstacles, actuator limitations and the modelled dynamics. Optimisation can be carried out offline to produce full optimal command trajectory for the RH system to carry out: a technique known as feedforward control [7]. Alternatively, flexible robot trajectory optimisation can be carried out online. One such online optimal control technique is Model Predictive Control (MPC), where the optimal trajectory is repeatedly calculated to account for possible drifting that can occur with feedforward control. The literature provides examples of MPC applied to control single link [8], [9], and multiple link flexible manipulators [10], [11]. Other investigators recently presented a simulation-focused investigation relevant to in-vessel tokamak remote maintenance [12]. Model-based approaches can provide large performance benefits over non-model-based techniques, whether they are feedforward or feedback. With recent advances in computing technology, this approach has seen increasing use within robotics, especially for control of complicated, underactuated non-linear systems with many degrees of freedom. For flexible payloads with known dynamic properties and external forces or disturbances, such techniques can be used to manoeuvre the object in such a way to minimise transient and residual structural oscillations [13], [14].

author's email: sam.herschmann@ukaea.uk

In this paper, we investigate optimal feedforward control techniques for manoeuvring a payload consisting of a simple flexible beam with a tip-mounted mass. The payload serves as a convenient analogue to a slender blanket segment. In section 2, we discuss dynamic modelling considerations. In section 3 we define a generic optimisation problem for manoeuvring underactuated payloads using reliable robot manipulators. In section 4 we simulate a particular test trajectory, and in section 5 we compare simulation results to experimental measurements. Finally, section 6 presents conclusions and recommendations for future investigation.

2. Dynamic modelling

We used a finite element (FE) approach to dynamically model a simple planar slender payload, free to move and bend within a cartesian plane, as shown in Fig. 2. Here, the position and deformation of the isolated payload can be described by the vector $\mathbf{q} = [x_e \quad y_e \quad \theta_e \quad \theta_1 \quad \cdots \quad \theta_N]^\intercal$, where the robot end-effector coordinates are denoted by the 'e' subscript, and the tangential angles of the payload at each FE node are measured from the global vertical direction. Bold symbols are used to denote vectors or matrices.

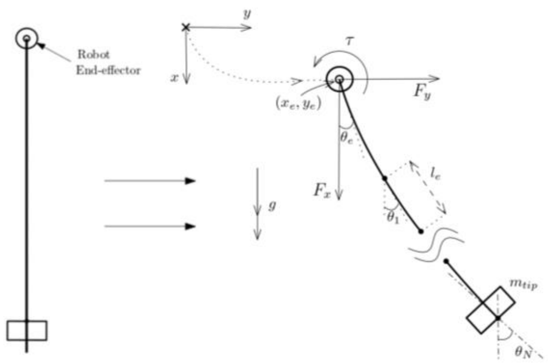


Fig. 2: Simple flexible payload finite element model representation

The Euler-Lagrange equations of motion were derived by a technique adapted from [15]; the general form of these equations is given in equation (1).

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} + \mathbf{G}(\mathbf{q}) = \mathbf{F}(t)\,, \qquad (1)$$

where $\mathbf{M}(\mathbf{q})$ is a non-linear symmetric matrix containing inertial parameters, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is a skew-symmetric Coriolis matrix describing rotational cross-coupling between FE components, $\mathbf{K}$ is a symmetric stiffness matrix, $\mathbf{G}(\mathbf{q})$ is a non-linear vector describing the gravitational forces and $\mathbf{F} = [F_x \quad F_y \quad \tau \quad 0 \quad \dots \quad 0]^\intercal$ is a vector describing the forces applied by the robot on the payload.

In this preliminary study, we simplified the problem of suppressing payload oscillations while moving on a desired trajectory through the following assumptions:

- Euler-Bernoulli beam bending (i.e., neglecting viscoelastic damping),
- Out-of-plane deflections and torsion are ignored,
- Elements are inextensible, and
- Deflection of elements is small (allowing $\mathbf{K}$ to be approximated using a matrix of constants).

It should be highlighted that this dynamic beam model represents a payload only and does not include robot dynamics. This assumption will be discussed further in section 3.

3. Robot Motion Planning

A typical robot motion planning problem is to generate a trajectory of actuator commands that manoeuvre the system in a desired manner. Numerical optimisation can be used to generate optimal robot trajectories that minimise a chosen objective function J. In this paper we approach this problem with the goal of minimising the planar cartesian and angular trajectory tracking error of the payload tip position $[x_{tip}(\mathbf{q}) \quad y_{tip}(\mathbf{q}) \quad \theta_{tip}(\mathbf{q})]^\intercal$ relative to a desired trajectory $[x_{tip,d}(t) \quad y_{tip,d}(t) \quad \theta_{tip,d}(t)]^\intercal$, while satisfying positional, kinematic, velocity and load constraints. The problem is described by Fig. 3.

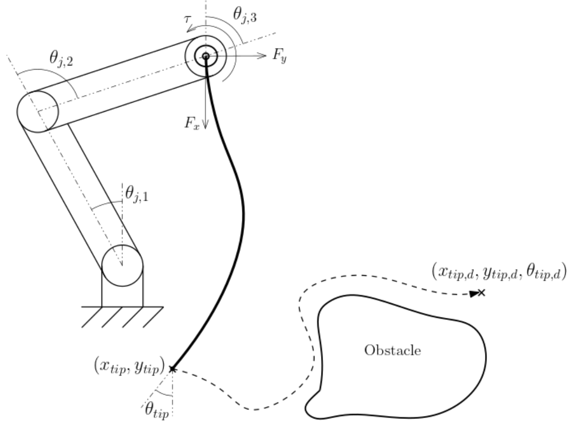


Fig. 3: Planar flexible payload robot trajectory optimisation problem

The payload tip position is a kinematic expression depending on all generalised coordinates $\mathbf{q}(t)$. This is approximated by equation (2).

$$\boldsymbol{p}_{tip}(\mathbf{q}) = \begin{bmatrix} x_{tip}(\mathbf{q}) \\ y_{tip}(\mathbf{q}) \\ \theta_{tip}(\mathbf{q}) \end{bmatrix} \approx \begin{bmatrix} x_e + l_e\left(\cos\theta_e + \sum_{i=1}^{N} \cos\theta_i\right) \\ y_e + l_e\left(\sin\theta_e + \sum_{i=1}^{N} \sin\theta_i\right) \\ \theta_e + \sum_{i=1}^{N} \theta_i \end{bmatrix} \qquad (2)$$

To solve this problem, the dynamics described by (1) should be explicitly considered, in addition to configuration constraints that enforce robot end-effector position, velocity and actuator force limits. The optimisation problem is defined in equation (3).

$$\min J = \sum_{t=t_0}^{t_f} \left\| \begin{bmatrix} x_{tip,d}(t) \\ y_{tip,d}(t) \\ \theta_{tip,d}(t) \end{bmatrix} - \boldsymbol{p}_{tip}(\mathbf{q}(t)) \right\|_2 = \sum_{t=t_i}^{t_f} \boldsymbol{e}(\mathbf{q}(t)) \qquad (3)$$

Subject to:

$$\mathbf{F}(t) = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} + \mathbf{G} \;\; \forall t \in [t_0, t_f] \qquad (3a)$$

$$\mathbf{q}(t_0) = \mathbf{q}_0, \;\; \boldsymbol{\theta}_j(t_0) = \boldsymbol{\theta}_{j_0} \qquad (3b)$$

$$\mathbf{q}(t) \in \mathbf{q}_{safe} \;\; \forall t \in [t_0, t_f] \qquad (3c)$$

$$\boldsymbol{p}_e(\mathbf{q}(t)) = \boldsymbol{f}_{FK}(\boldsymbol{\theta}_j(t)) \;\; \forall t \in [t_0, t_f] \qquad (3d)$$

$$\|\dot{\boldsymbol{\theta}}_j(t)\| < \dot{\boldsymbol{\theta}}_{j,max}, \;\; \|\ddot{\boldsymbol{\theta}}_j\| < \ddot{\boldsymbol{\theta}}_{j,max} \;\forall t \in [t_0, t_f] \qquad (3e)$$

$$\|\mathbf{F}(t)\| < \mathbf{F}_{\max} \quad \forall t \in [t_0, t_f] \tag{3f}$$

Constraint (3b) sets the initial conditions of the payload and robot joints to match those of the physical system. Constraint (3c) enforces that the outcome of the optimisation does not lead to collisions, where $\mathbf{q}_{safe}$ is the set of 'safe' payload positions. There are many ways to implement this constraint, such as using Control Barrier Functions [16], though such techniques are beyond the scope of this paper. Constraint (3d) fixes the coordinates $\boldsymbol{p}_e(\mathbf{q}) = [x_e \quad y_e \quad \theta_e]^{\mathrm{T}}$ to the corresponding robot end-effector coordinates, calculated by some forward kinematics function $\boldsymbol{f}_{FK}(\boldsymbol{\theta}_j)$ [17]. This allows robot kinematic limits to be captured within the optimisation problem. Constraints (3e) and (3f) ensure that the optimisation solution does not breach the maximum joint velocity $\dot{\boldsymbol{\theta}}_{J_{max}}$ and end-effector force limits $\mathbf{F}_{\max}$, respectively. Additionally, a set of Forward Euler method constraints were used to enforce the time derivative relationship between $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$, and $\boldsymbol{\theta}_j, \dot{\boldsymbol{\theta}}_j, \ddot{\boldsymbol{\theta}}_j$. This problem can be solved using various nonlinear optimisation solvers such as IPOPT [18].

The robot dynamics are omitted from equation (3) because the chosen experimental robot (UR10e) does not currently accept joint torque commands. Instead, we use robot joint position/velocity trajectory commands, generated from the optimisation. Assuming the robot can follow the optimal trajectory, and given that the payload dynamic model is accurate, constraint (3a) enforces that this motion would result in an optimal end-effector force trajectory. Under these assumptions it is important to compare the observed robot joint motions with the optimally generated command trajectories to confirm acceptable robot performance.

It should be noted that the error function $\boldsymbol{e}(\mathbf{q}(t))$ can be appended with other variables, such as tracking errors of other FE nodes or nodal angular errors. Doing so could either speed up or slow down the optimisation, depending on the dynamics of the payload, constraints, robot limitations, solver parameters and desired trajectory. For simplicity, planar end-effector force limits ($\mathbf{F}_{\max}$) are considered as opposed to individual robot joint torque limits. If we wanted to consider individual joint loads explicitly, we would need to add further constraints and include the dynamics of the robot in equation (1).

3.2 Feedforward Control

Fig. 4 provides the feedforward control block diagram used to carry out the optimal robot trajectory.

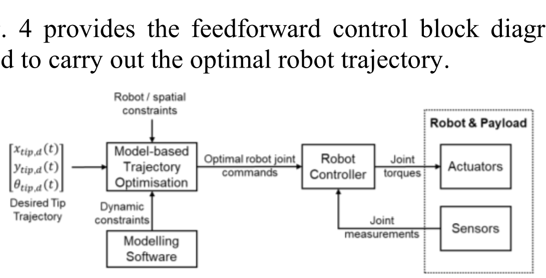


Fig. 4: Feedforward control block diagram

With a perfectly accurate payload model, zero unmodelled disturbances, and a high-performing robot control system, this approach would perform well in carrying out optimal manoeuvres, assuming that the command is reachable. Realistically it is difficult to produce a perfect analytical dynamic model, due to inaccurate modelling assumptions. As a result, this technique has limitations that must be addressed for the case of breeding blanket handling, where system uncertainty could be significant due to potentially varying dynamic parameters, and unpredictable disturbances are likely. There are different approaches to addressing these factors, depending on their severity.

3.3 Improving feedforward robustness with Input Shaping

Input shaping (IS) is a feedforward technique that modifies the commanded motion in such a way as to avoid excitation of known modes of oscillation [19]. We see IS used in many applications, most notably in crane technology [20]. The simplest input shaper is Zero-Vibration (ZV) IS, which convolves the desired end-effector trajectory with two carefully calculated impulses based on knowledge of the payload natural frequency and damping. System natural frequencies and damping ratios can be identified either from simulation or experimental measurements.

We can augment ZV input shaping to include robustness to uncertainty in payload natural frequency. We do this by adding extra impulses to the convolution, at the cost of further increasing the time duration of the trajectory. A command signal can be input shaped multiple times if there are several identified natural frequencies. Other investigators have extended the technique for systems of time-varying modes, such as flexible manipulators [21]. The feedforward control block diagram of Fig. 5 shows how the addition of an input shaper alters the feedforward controller of Fig. 4.

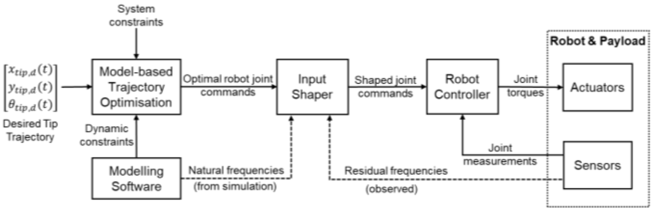


Fig. 5: Input Shaping for improving steady-state performance in the presence of model uncertainty.

4. Simulation Results

We chose a planar compound manoeuvre, shown in Fig. 6, as our test scenario. Our objective was to control the cartesian and angular position of the payload tip. For this investigation we chose ramped tip position commands (Fig. 7). This challenged our controller to accommodate acceleration impulses, with the resulting trajectory detectably exciting oscillation modes if unsuitably controlled.

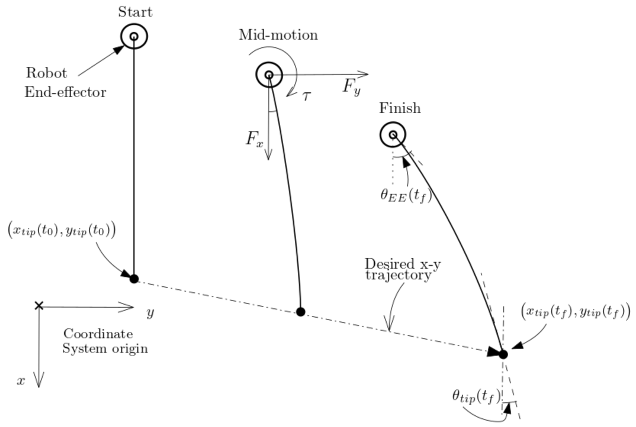


Fig. 6: Compound trajectory diagram

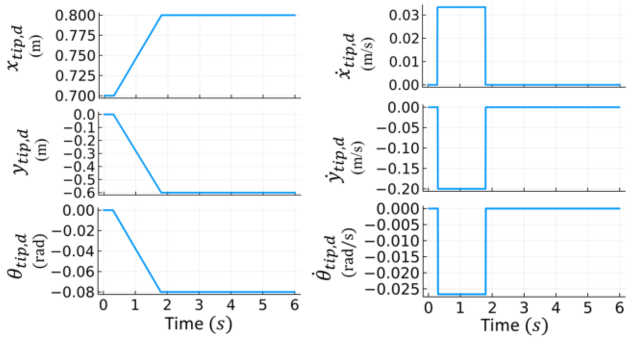


Fig. 7: Ramped tip compound manoeuvre commands

The payload parameters and control limits are listed in Table 1; these values are based on the real experimental system.

Table 1: System parameters

Parameter	Value
Robot model	UR10e
Tip mass	3.53 kg
End-effector mass	1.0 kg
Beam mass (including cabling & sensors)	1.63 kg
Beam material	Aluminium
Beam dimensions (Length X width X depth)	1567 X 50 X 5 mm
Maximum robot joint velocity, $\dot{\theta}_{j,max}$	1.5 rad/s
Maximum robot joint acceleration, $\ddot{\theta}_{j,max}$	3 rad/s^2
Maximum end-effector force	122.6 N
Maximum end-effector moment	10 Nm

We simulated the (isolated) payload system represented by equation (1) using three finite elements and a constant 0.005 s timestep. A Proportional + Derivative (PD) control law was chosen, with **F** calculated by the following equation:

$$\mathbf{F}(\mathbf{q},t) = \mathbf{K}_p\left(\boldsymbol{p}_{e,d}(t) - \boldsymbol{p}_e(\mathbf{q})\right) + \mathbf{K}_d\left(\dot{\boldsymbol{p}}_{e,d}(t) - \dot{\boldsymbol{p}}_e(\mathbf{q})\right) \quad (4)$$

$\mathbf{K}_p$ and $\mathbf{K}_d$ are proportional and derivative gain matrices tuned to achieve suitable end-effector tracking, and $\boldsymbol{p}_{e,d}(t) = [x_{e,d}(t) \quad y_{e,d}(t) \quad \theta_{e,d}(t)]^T$ is the desired end-effector position trajectory produced by the optimisation. Substituting equation (4) into equation (1) yields a closed loop system that can be simulated using a suitable ordinary differential equation solver. Using this control law allowed a baseline 'unshaped' motion to be simulated; Fig. 8 shows the tracking error of the simulated payload tip, on a trajectory calculated assuming that the payload is rigid (i.e., assuming no motion shaping is needed). Naturally, the payload tip does not track the commanded position well since the effects of gravity and stiffness are significant.

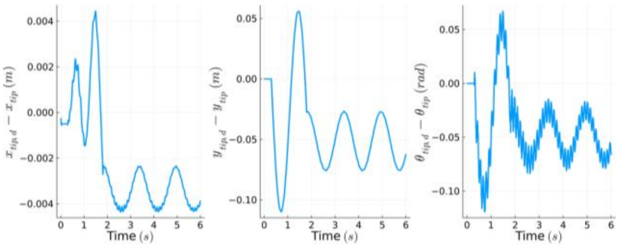


Fig. 8: Simulated payload tip tracking error for unshaped trajectory

Fig. 9 shows the payload tip tracking error simulated using the end-effector commands generated from the optimisation problem of equation (3). To compare the approach directly to the unshaped results in Fig. 8, the PD control law with the optimal end-effector position commands was used instead of substituting the optimal force trajectory into equation (1). Here, the transient tracking performance is improved significantly, and the steady-state offset is removed.

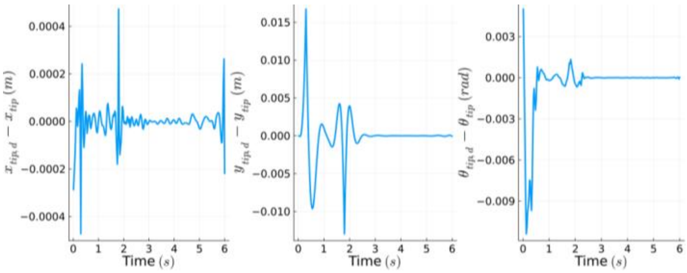


Fig. 9: Simulated payload tip tracking error for optimal trajectory

As discussed in Section 3, optimisation constraints allow force limits to be accounted for explicitly when calculating joint trajectories. Fig. 10 shows the end-effector torque τ constrained to within ±10 Nm (equal to the software limit of the UR10e robot axes).

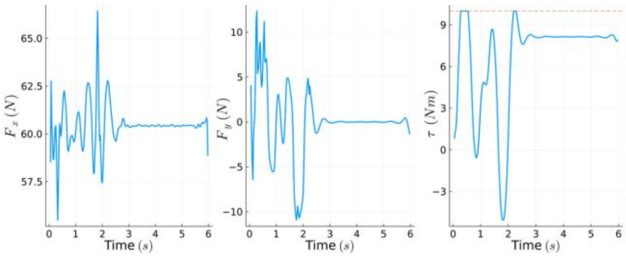


Fig. 10: Optimal end-effector force trajectories

It is important to highlight how the chosen optimisation time-step affects the performance of the optimal trajectory. This is shown in Fig. 11 where the y-direction tip tracking error is shown to significantly deviate from that of Fig. 9 as the time-step increases. The results from

0.025 time-step closely match that of the originally chosen 0.005 time-step.

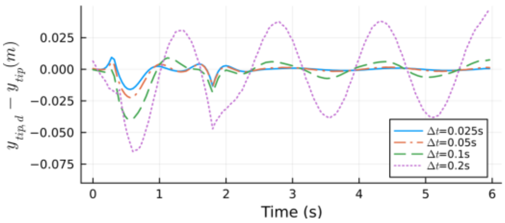


Fig. 11: Simulated motion precision for varying optimisation time-steps

The robot joint trajectories generated from the constrained optimisation can be made to satisfy a 1.5 rad/s velocity limit. The UR10e joint motion profiles are shown in Fig. 12.

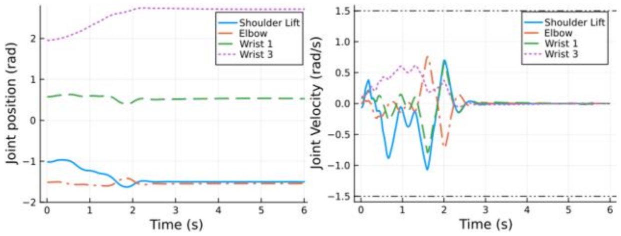
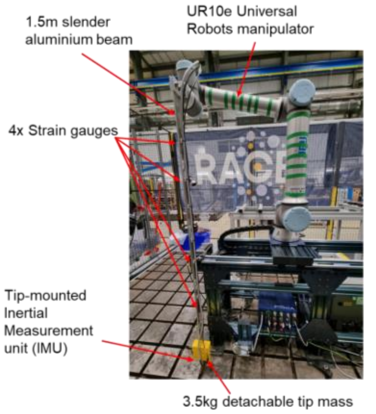


Fig. 12: UR10e robot joint velocity trajectories produced from the optimisation

5. Experimental Results

We used a Universal Robots UR10e robot for the experimental part of this investigation. We used the in-built robot control system to implement optimal trajectories, supplied to it using ROS. The setup is shown in Fig. 13.



1.5m slender aluminium beam
UR10e Universal Robots manipulator
4x Strain gauges
Tip-mounted Inertial Measurement unit (IMU)
3.5kg detachable tip mass

Fig. 13: Experimental UR10e and aluminium payload setup

We used an inertial measurement unit to measure the tip angle θ_{tip}. In Fig. 14, this measurement is plotted against both the command $\theta_{tip,d}$ and the simulated response results from Section 4, for each of three tests. Test (a) shows that, for the trajectory with no shaping, the experimental tip angle closely matches the simulated results, with poor command tracking performance in the transient and steady state. Higher frequency oscillations are more prevalent in the simulation, since the dynamic model does not include viscoelastic damping, whereas in the physical system, the higher-order modes are damped quickly.

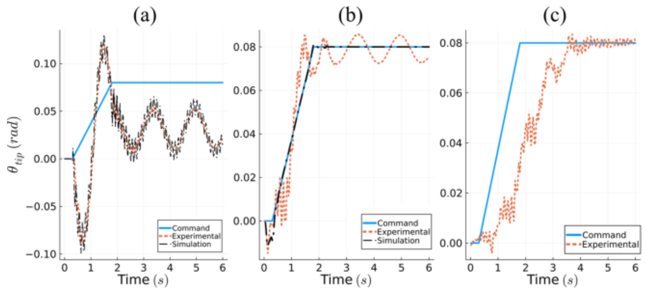


Fig. 14: Tip angle tracking performance comparison for three tests: (a) no motion shaping, (b) optimal shaping, (c) optimal shaping with ZVDD input shaping

The observed tip angle response using the optimal trajectory, shown in test (b), improves upon the unshaped results, though small mismatches between the analytical dynamics from equation (1) and the real-life payload dynamics lead to reduced tracking performance. This results in error in the transient tip tracking performance, as well as residual steady state oscillations.

A robust zero-vibration-derivative-derivative (ZVDD) input shaper was selected to mitigate these residual oscillations. This was chosen over a time-varying input shaper as the change in natural frequency of the payload throughout the trajectory was small enough to be captured within the range of a ZVDD shaper. Test (c) shows the benefits of using input shaping for improving steady-state performance, though, as discussed in Section 3.3, the trajectory time is increased, leading to poorer transient tracking performance. The post-manoeuvre tip tracking error $\theta_{tip}(t) - \theta_{tip,d}(t_f)$ is plotted in Fig. 15 for each test. This plot clearly demonstrates that, in the steady state, the optimal trajectory with input shaping (test (c)) performs best, with the least vibrations and a mean tip offset of 0.039°. It should be mentioned that only the fundamental natural frequency was used to input shape the trajectory; as a result, the higher order modal frequencies can be observed between 0 and 5 seconds of the post-manoeuvre measurements. To assist the demonstration we left these unmitigated, although they can be removed by input shaping the command again with the calculated 2nd order frequency.

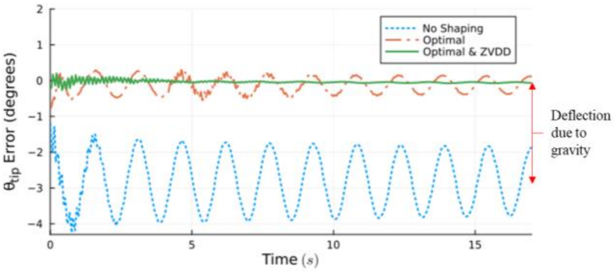


Fig. 15: Experimental post-manoeuvre tip angular error comparison

We also validated that the UR10e robot was capable of follow the commands. Fig. 16 shows that the experimental end-effector position closely matches the commanded motion.

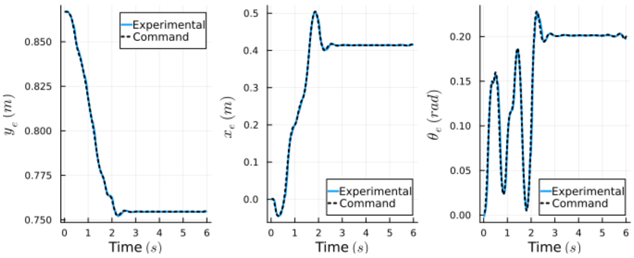


Fig. 16: Robot end-effector motion and commands for optimal trajectory

6. Conclusion

The results presented in Section 5 demonstrate that model-based optimal control approaches are applicable to a physical system inspired by the EU-DEMO blanket handling problem. Using a UR10e commercial robot, we have shown that we can explore and improve on the capability limits of positional control strategies for a flexible payload. By deriving a finite-element payload dynamic model and then calculating an optimal trajectory using the IPOPT non-linear optimisation algorithm, we were able to implement feedforward control to manoeuvre the position and pose of the payload tip with improved performance over a nominal unshaped trajectory. We demonstrated the potential of input shaping in mitigating known steady state oscillations. Our results also confirm the following factors as antagonistic to feedforward performance:

- Uncertainty, time-varying and/or unmodelled dynamic properties
- Presence of disturbances

To address these factors and build upon our results, we plan to investigate the potential of nonlinear MPC and explore modern system identification methods to build or improve dynamic models based on data [22]. We also plan to extend our investigation to more complex payload geometries, to include out-of-plane and torsional motions, viscoelastic damping, and actuator dynamics, using joint torque-controllable manipulators.

Acknowledgments

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion) and from the EPSRC [grant number EP/W006839/1]. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

References

[1] O. Crofts, A. Loving, M. Torrance, S. Budden, B. Drumm, T. Tremethick, D. Chauvin, M. Siuko, W. Brace, V. Milushev, M. Mittwollen, T. Lehmann, F. Rauscher, G. Fischer, P. Pagani, Y. Wang, C. Baars and A. Vale, "EU DEMO Remote Maintenance System development during the Pre-Concept Design Phase," *Fusion Engineering and Design,* vol. 179, 2022.

[2] J. Keep, S. Wood, N. Gupta, M. Coleman and A. Loving, "Remote handling of DEMO breeder blanket segments: Blanket transporter conceptual studies," *Fusion Engineering and Design,* no. 124, pp. 420-425, 2017.

[3] S. Grazioso, G. Di Gironimo, D. Iglesias and B. Siciliano, "Screw-based dynamics of a serial/parallel flexible manipulator for DEMO blanket remote handling," *Fusion Engineering and Design,* vol. 139, pp. 39-46, 2019.

[4] D. Subedi, I. Tyapin and G. Hovland, "Review on Modeling and Control of Flexible Link Manipulators," *Modeling, Identification and Control,* vol. 41, no. 3, pp. 141-163, 2020.

[5] H. N. Rahimi and M. Nazemizadeh, "Dynamic analysis and intelligent control techniques for flexible manipulators: a review," *Advanced Robotics,* vol. 28, no. 2, pp. 63-76, 2013.

[6] M. Benosman and G. Le Vey, "Control of flexible manipulators: A survey," *Robotica,* vol. 22, no. 5, pp. 533-545, 2004.

[7] C. T. Kiang, A. Spowage and C. K. Yoong, "Review of Control and Sensor System of Flexible Manipulator," *Journal of Intelligent & Robotic Systems,* vol. 77, no. 1, p. 187–213, 2015.

[8] R. Dubay, M. Hassan, C. Li and M. Charest, "Finite element based model predictive control for active vibration suppression of a one-link flexible manipulator," *ISA Transactions,* vol. 53, no. 5, pp. 1609-1619, 2014.

[9] B. P. M. Silva, B. A. Santana, T. L. M. Santos and M. A. F. Martins, "An implementable stabilizing model predictive controller applied to a rotary flexible link: An experimental case study," *Control Engineering Practice,* vol. 99, no. 104396, 2020.

[10] S. K. Pradhan and B. Subudhi, "Nonlinear Adaptive Model Predictive Controller for a Flexible Manipulator: An Experimental Study," *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY,* vol. 22, no. 5, pp. 1754-1768, 2014.

[11] M. Kramer, F. I. Muster, C. Rosmann and T. Bertram, "Elasticity-Aware Online Motion Optimization for Link-Elastic Manipulators," *IFAC PapersOnLine,* vol. 53, no. 2, pp. 9980-9985, 2020.

[12] X. Zhang, Y. Yang, H. Pan, Y. Cheng and Y. Song, "An efficient NMPC-based multi-task control toolkit for remote handling applications," *Fusion Engineering and Design,* vol. 187, 2023.

[13] D. A. Schoenwald, J. T. Feddema, G. R. Eisler and D. J. Segalman, "Minimum-time trajectory control of a two-link flexible robotic manipulator," in *IEEE robotics and automation conference*, Sacramento, CA (USA), 1991.

[14] M. Iskandar, C. van Ommeren, X. Wu, A. Albu-Schaffer and A. Dietrich, "Model Predictive Control for Flexible Joint Robots," Institute of Robotics and Mechatronics, Wessling, Germany, 2022.

[15] R. Robinett III, C. Dohrmann, G. R. Eisler, J. Feddema, G. Parker, D. G. Wilson and D. Stokes, Flexible Robot Dynamics and Controls, IFSR, volume 19, 2001.

[16] A. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath and P. Tabuada, "Control Barrier Functions: Theory and Applications," in *European Conference of Controls (ECC)*, Naples, Italy, 2019.

[17] R. M. Murray, Z. Li and S. S. Sastry, A Mathematical Introduction to Robotic Manipulation, CRC Press, 1994.

[18] A. Wächter and L. T. Biegler, On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming, Mathematical Programming, 106(1):25–57, 2006.

[19] N. C. Singer and W. P. Seering, "Preshaping command inputs to reduce system vibration," *ASME Journal of Dynamic*

Systems, Measurement, and Control, vol. 112, no. 1, pp. 76-82, March, 1990.

[20] S. Garrido, M. Abderrahim, A. Gimenez, R. Diez and C. Balaguer, "Anti-Swinging Input Shaping Control of an Automatic Construction Crane," *IEEE Transactions on Automation Science and Engineering,* vol. 5, no. 3, pp. 549-557, 2008.

[21] D. K. Thomsen , R. Søe-Knudsen, X. Zhang and D. Brandt, "Experimental Implementation of Time-varying Input Shaping on UR Robots," in *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Prague, 2019.

[22] S. L. Brunton and J. N. Kutz, Data-Driven Science and Engineering, Cambridge University Press, 2022.