A Control System Framework for Robust Deployability of Teleoperation Devices in Shared Workspaces

Sándor Felber[1], João Moura[1,2], and Sethu Vijayakumar[1,2]

***Abstract*— Robotic assistive devices offer unique benefits in industrial and medical settings augmenting purely human manipulation, such as precise micro-movements, and well-defined and concentrated force application with example use cases including soft-tissue inspection or surface polishing. Despite the abundance of literature focusing on laboratory demonstrations and experiments, the adoption of such technologies in industrial and medical settings remains limited due to concerns over safety, reliability, and training requirements. This paper introduces a reliable and adaptable control system framework for teleoperation devices with haptic force-feedback input, utilizing the Optimization-based Task Specification (OpTaS) Library. Our proposed framework addresses the aforementioned concerns by offering a control system with minimal training requirements and easily customizable safety constraints, aimed at enhancing deployability. A key contribution of the proposed control system is the introduction of inverse differential kinematics to real-time teleoperation, which enables the definition of safety constraints on the output of the controller providing an inherently safer implementation in contrast to fully inverse kinematic solutions.**

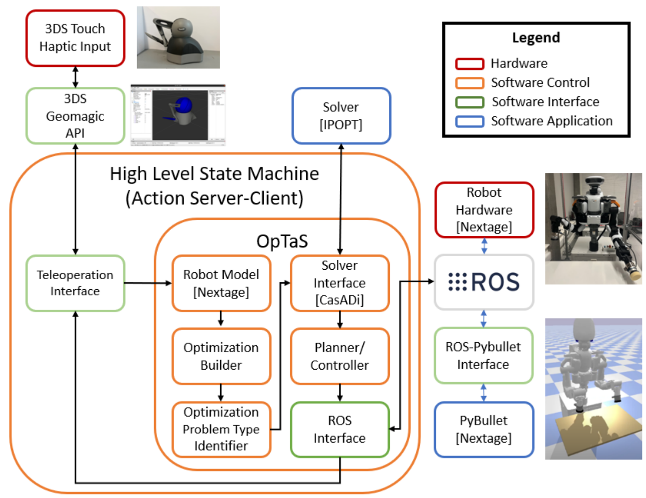

Fig. 1. System Overview of the Proposed Control System

I. INTRODUCTION

A. Related Work

Bilateral teleoperation systems, which provide the teleoperator a sense of the force experienced by the robotic end-effector, have several medical applications, including soft tissue palpation, stiffness recognition, heart surgery, and robot-assisted needle insertion. For instance, Talasaz et al. improved the overall performance of minimally invasive tumor localization and surgery with haptic feedback-based teleoperation [1], [2].

Teleoperation enables safer handling of contaminated material in the nuclear energy industry and maintenance tasks on oil extraction stations. Such examples demonstrate how teleoperation systems can reduce risks to humans in industrial settings [3], [4].

In this paper, we focus on two critical challenges that have been identified in the operational reliability of bilateral teleoperation systems in relevant literature [2], [5], [6], [7], [8], [9]. First, guaranteeing near real-time control and feedback, as soft environment parameters that are essential in medical applications are time-varying. Delays can significantly impact time-sensitive industrial applications as well such as pick-and-place tasks from a conveyor belt. Secondly, robotic control systems need to guarantee safety and stability when interacting with doctors, patients, or factory workers—sudden, unexpected motions present unacceptable dangers prohibitive to the deployment of teleoperation devices.

To ensure the safe deployment of teleoperation systems in industrial and medical applications alike, the teleoperation system requires compliance with standards and specifications including ISO 10218/1-2 [10] and ISO/TS 15066 [11]. The primary safety features of conventional (ISO 10218) collaborative robots are twofold: safe emergency stop and maximum torque limitation [12]. Augmenting this, ISO/TS 15066 proposes an energy-based approach referring to the impact force F exerted on the human collaborator to derive the maximum relative velocity v_{rel} between the relevant human body part and the robot in Eq. (1).

$$\frac{1}{2} m_{\text{eff}} v_{\text{rel}}^2 = \frac{1}{2} \frac{F^2}{k} \tag{1}$$

where k is the equivalent spring constant of the human body part and m_{eff} is the effective mass denoting the colliding human and robotic parts further discussed in [13], [14], [15].

Here we focus on the relative velocity constraint and in Section II we propose a teleoperation control system particularly suitable for the outlined medical and industrial applications by facilitating imposing velocity limits, and by inherently guaranteeing that the constraint is not violated.

B. Contributions

- Development of an inverse differential kinematics-based control system for improved performance and

[1]Authors are with the School of Informatics, The University of Edinburgh, Edinburgh, U.K. Correspondence: sfelber@ed.ac.uk
[2]Authors are with The Alan Turing Institute, London, U.K.

- widespread deployability of the framework for haptic teleoperation systems in the medical and manufacturing industries, for instance, in surgical and pick-and-place tasks where safety and latency are crucial.
- Proposed implementation of an interface for safety constraint implementation (such as velocity, and acceleration) on teleoperation, and evaluation against the baseline IK-based system in terms of latency, reliability, and safety constraint violation.
- Integration of the OpTaS Optimization-based Task Specification Library for modularity and scalability.

II. PROBLEM FORMULATION

A trajectory optimization (TO) in OpTaS [19] can be formulated as

$$\min_{x,u} \text{cost}(x, u; T) \quad \text{subject to} \begin{cases} \dot{x} = f(x, u) \\ x \in X \\ u \in U \end{cases} \tag{2}$$

where $x = x(t) \in R^{n_x}$ and $u = u(t) \in R^{n_u}$ denote state and control inputs, with T as the time-horizon for the planned trajectory and t as time. f represents the system dynamics, while $X \subseteq R^{n_x}$ and $U \subseteq R^{n_u}$ are the respective feasible regions for the states and controls (set of in/equality constraints) [19].

Using such equality and inequality constraints, the baseline control system was implemented in Python with OpTaS using the IPOPT solver through CasADi [21]. This was tested with a 6-degree-of-freedom (DOF) 3DS Touch stylus with 3-DOF force-feedback in PyBullet (a reliable impact and contact simulation environment) with the help of ROS-PyBullet Interface [22] as the baseline model (Fig 1).

To start teleoperating, the manipulator needs to assume a certain starting pose for which IK suits as this is a one-off planning task. Once synchronized, teleoperation can be achieved in position or velocity control with either IK or IDK. Using the inverted forward kinematic function f^{-1} a mapping between the end-effector pose, x, and the joint angles of the manipulator, q, can be expressed as

$$q = f^{-1}(x) \tag{3}$$

Similarly, teleoperation can be carried out using IDK where the end-effector velocity, $\dot{x}$ is mapped to the robot's joint velocities, $\dot{q}$. $J^{-1}(q)$ is the inverse of the Jacobian matrix of joint angles [17], [18].

$$\dot{q} = J^{-1}(q)\dot{x} \tag{4}$$

During teleoperation, performing full IK at every cycle introduces the following points of concern:

- Latency is critical in near real-time teleoperation applications. Performing full IK at every cycle can result in reduced system stability and jitter-prone motion.
- Singularity—the Jacobian matrix can become singular (non-invertible) or ill-conditioned resulting in erratic joint movements or in no analytical solution existing for the IK, which then needs to be approximated by other methods such as the Moore-Penrose inverse [16].

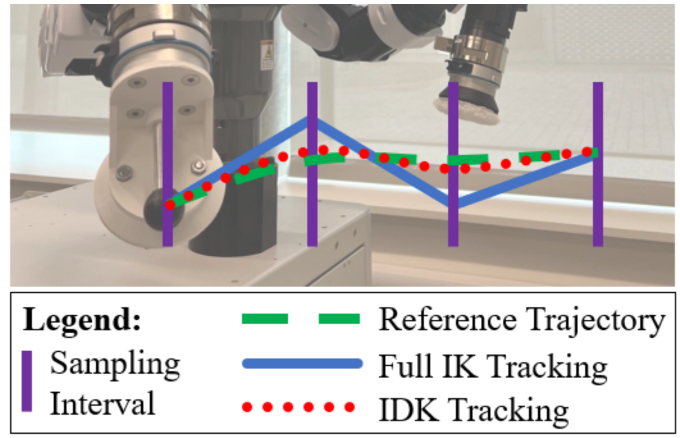

Fig. 2. Comparison of Inverse Kinematic and Inverse Differential Kinematic Real-Time Trajectory Tracking During Teleoperation with the End-Effector of Kawada Nextage Arm

- Redundant robotic systems with more degrees of freedom than necessary for a given task can have multiple valid solutions, which makes selecting the most appropriate solution during teleoperation problematic [19]. If the IK solver finds a local minimum corresponding to a radically different joint configuration, then the velocity might violate the safety constraint.

Hence in our work, we propose to assume an initial pose at the start of the teleoperation cycle calibrated and synchronized to the teleoperation device. Once in sync, the teleoperation would take place in IDK mode allowing one to directly limit the joint velocity $\dot{q}$ and establish a smoother path for more reliable tracking as indicated by Fig 2.

- Latency is reduced in the IDK problem (only having to calculate $\dot{x}$ yields faster control loop updates) minimizing the impact of latency on system stability and performance.
- Singularities can be avoided by IDK operating in the velocity domain, by using alternative methods like the Jacobian pseudoinverse or damped least squares [20].
- Redundancy can be eliminated by incorporating complementary constraints or cost terms, such as minimizing the joint velocities $\dot{q}$. In addition, joint velocities can be constrained simply based on the classification of human body parts in the environment (ISO 15066 specifies a force limit corresponding to the relevant body part) and the consequent maximum relative velocity v_{rel}.

III. CONCLUSIONS

In this paper we have proposed using an OpTaS-based IDK implementation for robust teleoperation of industrial and medical robotic manipulators. The full IK teleoperation baseline has been implemented and validated in PyBullet. We plan to implement the IDK-based teleoperation and compare latency, reliability, and safety to the baseline IK teleoperation. Future work will facilitate the compliance of teleoperation systems with industry standards allowing teleoperation systems to be deployed in shared workspaces.

APPENDIX

```python
# set up right arm optimization
self.right_arm = optas.RobotModel(
    urdf_string=self._robot_description,
    time_derivs=[0],
    param_joints=['HEAD_JOINT0', 'HEAD_JOINT1', '
    LARM_JOINT0', 'LARM_JOINT1', 'LARM_JOINT2', '
    LARM_JOINT3', 'LARM_JOINT4', 'LARM_JOINT5'],
    name='nextage_right_arm'
)
self.right_arm_name = self.right_arm.get_name()
self.ndof = self.right_arm.ndof
# nominal robot configuration
q_nom = optas.DM.zeros(self.ndof)
# set up optimization builder
builder_right_arm = optas.OptimizationBuilder(T=1,
    robots=[self.right_arm])
# get robot state and parameters variables
q_var = builder_right_arm.
    get_robot_states_and_parameters(self.
    right_arm_name)
# get end-effector pose as parameters
pos = builder_right_arm.add_parameter('pos', 3)
ori = builder_right_arm.add_parameter('ori', 4)
# set variable boudaries
builder_right_arm.enforce_model_limits(self.
    right_arm_name)
# equality constraint on right arm position
pos_fnc = self.right_arm.
    get_global_link_position_function(link=self.
    _link_ee_right)
builder_right_arm.add_equality_constraint('
    final_pos', pos_fnc(q_var), rhs=pos)
# rotation of the right arm position
self.R_fnc = self.right_arm.
    get_global_link_rotation_function(link=self.
    _link_ee_right)
# equality constraint on orientation
ori_fnc = self.right_arm.
    get_global_link_quaternion_function(link=self.
    _link_ee_right)
builder_right_arm.add_equality_constraint('
    final_ori', ori_fnc(q_var), rhs=ori)
# optimization cost: close to nominal config
builder_right_arm.add_cost_term('nom_config', optas
    .sumsqr(q_var-q_nom))
# setup solver
self.solver_right_arm = optas.CasADiSolver(
    optimization=builder_right_arm.build()).setup('
    ipopt')
```

Fig. 3. Example Code for Baseline Trajectory Optimization with OpTaS Outlined in Section II.

ACKNOWLEDGMENT

This research is supported by The Alan Turing Institute, United Kingdom and has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017008, Enhancing Healthcare with Assistive Robotic Mobile Manipulation (HARMONY). This work was supported by core funding from the Wellcome/EPSRC [WT203148/Z/16/Z; NS/A000049/1]. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101016985 (FAROS project). This research is supported by Kawada Robotics Corporation.

REFERENCES

[1] A. Talasaz, R. V. Patel and M. D. Naish, "Haptics-enabled Teleoperation for Robot-assisted Tumor Localization", IEEE International Conference on Robotics and Automation, pp. 5340-5345, 2010.

[2] Y. Deng, Y. Tang, B. Yang, W. Zheng, S. Liu and C. Liu, "A Review of Bilateral Teleoperation Control Strategies with Soft Environment," 2021 6th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM), Chongqing, China, 2021, pp. 459-464, doi: 10.1109/ICARM52023.2021.9536056.

[3] O. Tokatli et al., "Robot-Assisted Glovebox Teleoperation for Nuclear Industry," Robotics, vol. 10, no. 3, p. 85, Jul. 2021, doi: 10.3390/robotics10030085.

[4] C. A. Garcia, J. E. Naranjo, L. A. Campana, M. Castro, C. Beltran and M. V. Garcia, "Flexible Robotic Teleoperation Architecture Under IEC 61499 Standard for Oil & Gas Process," 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 2018, pp. 1269-1272, doi: 10.1109/ETFA.2018.8502520.

[5] C. Chirapornchai, F. Niyi-Odumosu, M. Giuliani and P. Bremner, "User Requirements for a Robot Teleoperation system for General Medical Examination," 2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), Napoli, Italy, 2022, pp. 858-863, doi: 10.1109/RO-MAN53752.2022.9900855.

[6] F. B. Chioson et al., "Implementation of a Bilateral Teleoperation Haptic Feedback Controls to Robotic Minimally Invasive Surgery," 2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Manila, Philippines, 2020, pp. 1-5, doi: 10.1109/HNICEM51456.2020.9400009.

[7] I. Dekker, K. Kellens, and E. Demeester, "Design and Evaluation of an Intuitive Haptic Teleoperation Control System for 6-DoF Industrial Manipulators," Robotics, vol. 12, no. 2, p. 54, Apr. 2023, doi: 10.3390/robotics12020054.

[8] J. Lee, X. Zhang, C. H. Park and M. J. Kim, "Real-Time Teleoperation of Magnetic Force-Driven Microrobots With 3D Haptic Force Feedback for Micro-Navigation and Micro-Transportation," in IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 1769-1776, April 2021, doi: 10.1109/LRA.2021.3060708.

[9] J. Yun, S. Guo, H. Li and K. Wang, "Design of Master-slave Teleoperation Puncture System Based on Force Feedback," 2022 International Conference on Manufacturing, Industrial Automation and Electronics (ICMIAE), Rimini, Italy, 2022, pp. 366-370, doi: 10.1109/ICMIAE57032.2022.00079.

[10] ISO 10218-1:2011, "Robot for industrial environments – Safety requirements – Part 1 : Robot", Technical report, 2011.

[11] ISO/TS 15066:2016, "Robots and robotic devices - collaborative robots," International Organization for Standardization, Standard ISO/TS 15066:2016, Feb. 2016.

[12] M. J. Rosenstrauch and J. Krüger, "Safe human-robot-collaboration-introduction and experiment using ISO/TS 15066," 2017 3rd International Conference on Control, Automation and Robotics (ICCAR), Nagoya, Japan, 2017, pp. 740-744, doi: 10.1109/ICCAR.2017.7942795.

[13] R. J. Kirschner, N. Mansfeld, G. G. Peña, S. Abdolshah and S. Haddadin, "Notion on the Correct Use of the Robot Effective Mass in the Safety Context and Comments on ISO/TS 15066," 2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR), Tokoname, Japan, 2021, pp. 6-9, doi: 10.1109/ISR50024.2021.9419495.

[14] O. Khatib, "Inertial Properties in Robotic Manipulation: An Object-Level Framework", The International Journal of Robotics Research, vol. 14, no. 1, pp. 19-36, 1995.

[15] C. Sloth and H. G. Petersen, "Computation of Safe Path Velocity for Collaborative Robots," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 6142-6148, doi: 10.1109/IROS.2018.8594217.

[16] A. Ben-Israel and T. N. E. Greville, "Generalized Inverses: Theory and Applications," 2nd ed. New York, NY: Springer, 2003.

[17] J. J. Craig, Introduction to Robotics: Mechanics and Control, 4th ed. Pearson, 2021.

[18] K. M. Lynch and F. C. Park, Modern Robotics: Mechanics, Planning, and Control. Cambridge: Cambridge University Press, 2017.

[19] Mower, C. E., Moura, J., Zamani Behabadi, N., Vijayakumar, S., Vercauteren, T., & Bergeles, C. (2023). OpTaS: An Optimization-based Task Specification Library for Trajectory Optimization and Model Predictive Control. In 2023 International Conference on Robotics and Automation (ICRA). [Online]. Available: https://github.com/cmower/optas

[20] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, Robotics: Modelling, Planning and Control. Springer Science & Business Media, 2009.

[21] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," Mathematical Programming Computation, vol. 11, no. 1, pp. 1-36, Mar. 2019.

[22] C. E. Mower, T. Stouraitis, J. Moura, C. Rauch, L. Yan, N. Z. Behabadi, M. Gienger, T. Vercauteren, C. Bergeles, and S. Vijayakumar, "ROS-PyBullet Interface: A Framework for Reliable Contact Simulation and Human-Robot Interaction," [to appear] Proceedings of the Conference on Robot Learning, 2022.