2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Manipulability Optimization and Thermal Control of Industrial Robots in Real-Time Using Digital Twins, Augmented Reality, and OPC UA

Peter Abt[1], René Harmann[2] and Eric Guiffo Kaigom[3]

***Abstract*— Planning efficient motions of robots is often inhibited by the difficulty to spatially imagine and enhance their dexterity and agility in the task space. To accommodate this complexity, meet short changeover times, and support inclusion along with sustainability goals, robot operators with different experiences and facing a robot diversity need responsive and actionable interfaces. These enable the prediction and optimization of hidden (i.e., internal) performance-critical properties of physical robots, such as their velocity manipulability in the task space and thermal loads in the joint space, in real-time on concerned components. We address these fundamental challenges encountered in several high-level robotized applications by developing a virtual spatial augmentation of physical robots that translates their complex, otherwise invisible manipulability and thermal states, into accessible and interactive visual cues easily interpreted and used by even novices to improve the robot dexterity in the null-space and anticipate issues due to overheating joints. An upskilling immersion based upon augmented reality and enriched with overlaid digital twins is leveraged to this end. While following values targeted by our overarching Metarobotics framework, we stress the non-invasive and sustainable characteristic of the approach. Furthermore, our framework transparently embeds in existing robotized industrial settings, systems of systems, and workflows without production standstills. We emphasize on its semantic interoperability, versatility, and openness driven by the OPC UA standard, share results from experiments, and pointed out emerging basic research implications that deserve further attention.**

I. Introduction

Operating industrial robots for e.g. manufacturing or servicing often involves collaborators with different experiences in the evaluation and usage of properties that govern the robot availability, predictability, and performance. As industries are increasingly automated to meet volatile demands, there is a growing need for intuitive, flexible, and intrinsically safe robot interfaces that can be operated by less specialized workers for e.g. inclusion, adaptation, and economic reasons. This objective is gaining momentum. It has attracted an increasing interest in leveraging Augmented Reality (AR) and Digital Twins (DTs) for robotized applications [1]. For instance, [2] has developed an AR-application which enables the collaborative control and augmented visualization of robots using a bidirectional communication between robots and users. Natural hand motions have been combined with DTs of robots and AR to plan end-to-end manipulations without writing any line of code in [3], [4]. Despite their usefulness and effectiveness, these approaches mostly focus on external i.e. visible manipulation properties of robots.

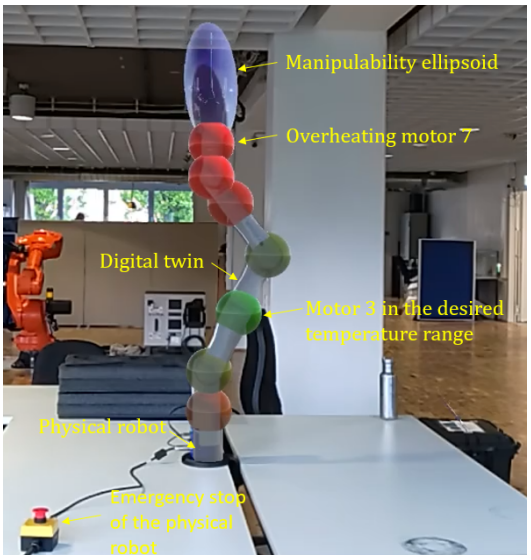


Fig. 1. Interacting with otherwise invisible properties (e.g., manipulability ellipsoid visualized at the end-effector and thermal behaviors of the motors) of the robot arm through an overlaid Digital Twin and Augmented Reality interface to monitor, share, and enhance the robot performance in real-time.

Recent advances in robotics have pointed out the high cognitive load faced by the workforce when it comes to use robots in manufacturing [5] and teleoperation [6]. The interpretability of highly nonlinear and coupled internal and thus invisible properties of robots for e.g. planning, control, and performance optimization purposes is a root cause of this barrier. The situation gets exacerbated as a robotic system of systems is involved and its kinematics is reconfigured. Reasons include the difficulty to compute the Jacobian associated with a production-relevant point (e.g., the tool center point at the end-effector) of compound systems as their kinematic structure evolves for performance or economic reasons. This complicates the planning of motions and thus return on investments of refurbished robots in small and medium-sized enterprises. Furthermore, companies that take the leap from a single robotized application to a large-scale robotized automation, such as in the automotive industry, face short changeover times along with different levels of dexterity requirements and thermal impacts from overloads on the robot drive train. Monitoring motional and thermal conditions of robot assets exactly at the location of relevant components,

[1]Peter Abt is with the Dept. of Comp. Science & Engineering, Frankfurt University of Applied Sciences, Frankfurt a.M., Germany peter.abt@stud.fra-uas.de
[2]René Harmann is with the Dept. of Comp. Science & Engineering, Frankfurt University of Applied Sciences, Frankfurt a.M., Germany rene.harmann@fb2.fra-uas.de
[3]Eric Guiffo Kaigom is with the Dept. of Comp. Science & Engineering, Frankfurt University of Applied Sciences, Frankfurt a.M., Germany kaigom@fb2.fra-uas.de

such as the end-effector and motors, is pivotal. Individualized and easy-to-interpret insights into hidden dynamics overlaid on these targeted components improve the situational awareness driven by measured and mined data, and facilitate comparison with surrounding components to quickly isolate root causes and anticipate failures. The spatially augmented presence achieved by using AR and DT can pave the ground for the development of sustainable solutions, such as enhanced pose reachability and extended motor lifetime, cooling monitoring and control, energy-efficiency, projected on physical robots to upskill the workforce (Industry 5.0) and improve automation efficiency (Industry 4.0) without downtime.

We describe the development of an AR- and DT-driven framework that provides robot operators with fast, explainable, and engaging insights into the internal properties of robots. Task space velocity manipulability and thermal motor behavior are considered. These insights help understand and optimize the dexterity and agility as well as anticipate thermal loads on motors of robots. Benefits include the availability, enhanced performance, and sustainability of robots, thereby extending our previous works on DT- and AR-based monitoring, interaction, and control of robots [3], [4], [8], [9], [7], [10]. Data acquired from physical robots are processed and outcomes are semantically exposed to pervasively shape the robot performance via OPC UA in real-time. We compute the pivotal end-effector Jacobian in a generic way to support changing kinematics of systems of systems and develop tools, such as the null-space, to improve agility and sustain motion performance. Our framework builds upon the network architecture shown in fig. 2 to immerse and provides interfaces to influence internal robot properties via AR and DTs. The bidirectional framework is modular and extensible, allows the seamless collaboration with evolving kinematics and heterogeneous devices to support future applications.

II. Setup and Context

A Kinova KinGen3 robot arm, with $n_j = 7$ degrees of freedom (DoF) [11], and a HoloLens 2 AR headset [12] are used as main hardware components. The KinGen3 robot arm is mounted on top of a mobile robot platform - the Husky (A200) from Clearpath Robotics - (see fig. 6) for which a Web-Interface has been developed [8] that will be extended to include the data from the robot arm. The robot arm is controlled using the Robot Operating System (ROS) [13]. Unity is the foundation for the AR application. Thanks to the holographic nature of the HoloLens 2, virtual objects can be projected into the user field of view, without obstructing the vision or modifying the physical environment of the user. The approach is therefore non-invasive, as advocated by our overarching Metarobotics framework [7]. The following two parameters are monitored: the robot manipulability and the temperature of the motor of each joint. The manipulability is used to evaluate and show how good the robot arm can deliver task space velocities from a given configuration of joints, while joint temperatures provide insights into the motor load and potential overheating issues [14]. This information is visualized where the related

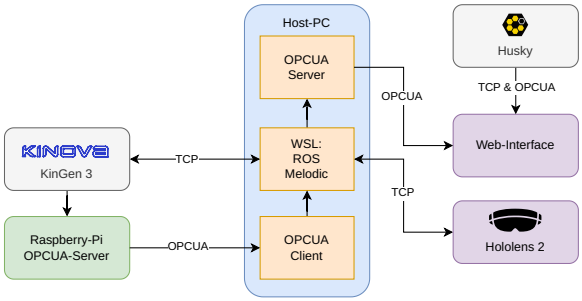


Fig. 2. Simplified network architecture of the proposed framework

physical component is located to bind awareness to it. As Industry 4.0 continues to evolve towards Industry 5.0, the need for an engaging and ecocentric understanding of the robot capabilities and related impacts becomes crucial for an efficient and sustainable automation that yields shared well-being. In a typical automotive manufacturing plant, for instance, workers traditionally specialized in manual assembly might now need to interact with complex robotic systems. Without an amenable and projectable understanding of robot implications, limitations, and opportunities, this transition can lead to reluctance, hazards, and inefficiency. Our AR- and DT-based framework provides data, optimization, and immersion spaces for informed, balanced, and actionable decisions at different innovation and responsibility levels.

III. Related Work

AR and DT are promising technologies for enhancing human-robot interaction [1], [2], [3], [4], [5], [6], [7]. For instance, [1] has conducted a survey on AR and robotics, identifying various applications of AR in robotics, such as robot path planning [2] and visualizations of robots internal properties like ”safety aura” [15] and others. By contrast, we focus on motion [5] and thermal [14] aspects of the robot and their enhancement using AR and DT. Yoshikawa [16] established the theoretical foundations to quantify manipulability for motion optimization, laying the groundwork for subsequent research. However, the on-robot visualization and interoperable enhancement of the manipulability of robotic systems of systems (see, e.g., r.h.s of fig. 4) based on DT, AR, and OPC UA is still missing, to the best of our knowledge.

OPC UA is a standard for (industrial) communication that allows for a secure exchange of data with a semantic meaning between systems without active polling and thus CPU cycle wastes through subscription and publication mechanisms. It has been widely adopted by robot manufacturers [17] and enables the seamless integration of heterogeneous robotics-related devices into Industry 4.0 environments [18]. The convergence of OPC UA over TSN (Time Sensitive Network) for hard real-time requirements and Industrial 5G is expected to open up new opportunities. Included is a reliable wireless exchange of information with a deterministic maximum low-latency. Our previous work [9] has shown the usage of OPC UA for remote exchange of robot data from a train in motion in city traffics. Obtained results are integrated in this work.

IV. System Architecture and Implementation

The system consists of three main components: the HoloLens headset, a host PC and a mobile robot arm. The host PC runs ROS Melodic within Windows Subsystem for Linux. The Host-PC acts as a central gateway for communication between the used components. The connection between ROS and Unity occurs using the ROS-TCP-Connector package [19]. This allows bidirectional communication between ROS and Unity. Thus the robot arm and gripper can be controlled from Unity [4] and properties of the robot which are available as data in ROS can be visualized in Unity. The availible ROS topics are expanded by supporting manipulability and temperature data. Manipulability are computed in a ROS node and published as a ROS topic. The thermal behavior is measured in the driver and on motor casings of the mobile base. A connectivity and data exchange manager implemented on a Raspberry Pi exposes aggregated data via OPC UA [9]. It is used to collect the motor temperature of the arm via the Kinova Kortex API.

A. Manipulability: virtual access and optimization in reality

We harness DTs to equip robot operators with the manipulability measure [16] and direction of highest motions reflected by the manipulability ellipsoid (see fig. 3 and fig. 4). They use them to predict and enhance the robot dexterity.

1) Measure: In robotized applications, such as pick-and-place and tracking Cartesian motions, taking advantage of a dexterous and agile behavior of the end-effector is crucial. The performance of the related manipulator can be characterized by how well its joint space velocities $\dot{q} \in \mathbb{R}^{n_j}$ can be transformed into task space velocities $\dot{x} \in \mathbb{R}^{n_{ws}}$. n_j and n_{ws} are joint and task space dimensions. We use the velocity manipulability measure $m(q)$ to describe this mapping and make it accessible (see fig. 3). In essence [20],

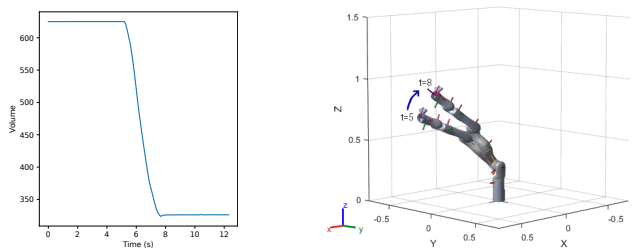
$$m(q) = \sqrt{\det(E(q))} \tag{1}$$

where

$$E(q) = J(q)J^T(q) \in \mathbb{R}^{n_j \times n_j} \tag{2}$$

and $\det(.)$ stands for the determinant computation. Herein,

$$J(q) = \frac{\partial}{\partial q} FK(q) \in \mathbb{R}^{n_{ws} \times n_j} \tag{3}$$



(a) Manipulability (b) Robot Pose

Fig. 3. Computation and visualisation of $m(q)$ (**l.h.s**) during trajectory tracking (**r.h.s**) in MATLAB©. $m(q)$ decreases near a singular posture.

is the Jacobian matrix associated with the end-effector. $FK(q) = [FK_T^T(q)\ FK_R^T(q)]^T$ is the forward kinematics of the robot that translates q into the vector of task space coordinates x:

$$x = FK(q). \tag{4}$$

Furthermore, the mapping from $\dot{q}$ to $\dot{x}$ is given by

$$\dot{x} = J(q)\dot{q}. \tag{5}$$

Obtaining $\dot{q}$ from eq. (5) by using the inverse of J (assumed to be square in this case, i.e., $n_{ws} = n_j$) reveals that large $\dot{q}$ are required even for a small $\dot{x}$ near a $q = q_s$ for which $J(q_s)$ becomes rank-deficient. Singular postures are $q = q_s$ for which $\det(J(q_s))$ vanishes. A robot operator needs to anticipate and avoid the vicinity of q_s as the current joint configuration at which a desired task space velocity is required. Reasons include the feasibility, predictability, and stability of the necessary $\dot{q}$ as well as safety reasons (e.g., collision hazards due to velocity saturation). In eq. (1), $m(q)$ is a scalar that reflects the vicinity of the manipulator to $q = q_s$ also in the case of kinematic redundancy (i.e., $n_{ws} < n_j$). At $q = q_s$, $m(q_s)$ vanishes. Hence, the maximization of $m(q)$ is instrumental to maintain a dense mapping of joint velocities and thus high robot dexterity in the task space.

2) Motion capability: The scalar $m(q)$ does not explicitly reveal in which task space direction the robot has e.g. the highest motion capability starting from a non-singular q. This direction is the eigenvector associated with the largest eigenvalue of $E(q)$ that characterizes the manipulability ellipsoid. In general, the total motion can be decomposed in a pure translational and rotational motion of the end-effector by writing

$$J = \begin{bmatrix} J_T \\ J_R \end{bmatrix} \in \mathbb{R}^{6 \times n_j}. \tag{6}$$

$J_T \in \mathbb{R}^{3 \times n_j}$ and $J_R \in \mathbb{R}^{3 \times n_j}$ are the translational and rotational part of J. The translational ellipsoid follows from

$$E_T(q) = J_T(q)J_T^T(q) \in \mathbb{R}^{3 \times 3}. \tag{7}$$

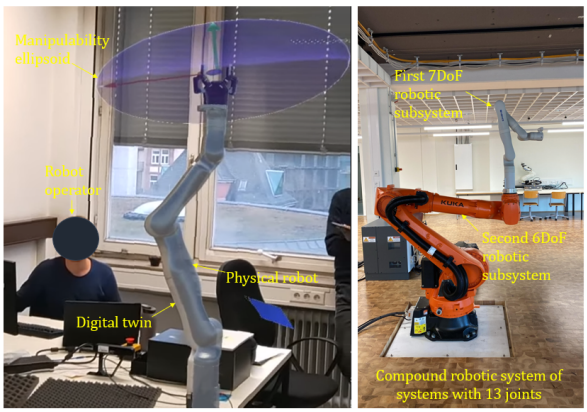


Fig. 4. **L.h.s:** Hololens 2 view used to equip and upskill the operator with otherwise invisible information about the robot manipulability. **R.h.s:** The proposed framework is also suitable for integrated and kinematically highly redundant systems of systems, highlighting its versatility and usefulness.

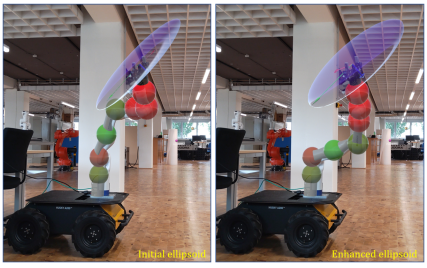


Fig. 5. From l.h.s to r.h.s: Null-space-based optimization of the translational manipulability of a robotic system for the same primary Cartesian task.

Its eigenvectors are main ellipsoid directions. Instead of J_R, WJ_R is used to compute the rotational ellipsoid. $W \in \mathbb{R}^{3\times3}$ maps the time rate of the orientation task coordinates to the angular velocity of the end-effector given in the base frame.

In practice, the identification of joint postures that enhance the capability of the manipulator to largely and uniformly deliver velocities in the task space while the end-effector is carrying out a primary task is important (see fig. 5). In this case, the end-effector velocity related to the primary task can be non-zero and decoupled from joint motions aiming at optimizing the posture of the kinematically redundant robot to maximize $m(q)$, as experimentally depicted in figs. 5 and 6. A core user-friendly advantage is the simultaneous visualization of the manipulability ellipsoid during the optimization which is easy to perceive and spatially interpret by the robot operator in real-time for the optimization of the task space dexterity. To this end, the joint velocities read

$$\dot{q} = J^{\#}\dot{x}_c + N\nabla m(q). \tag{8}$$

Herein, $J^{\#}$ is the Moore–Penrose pseudoinverse of J, i.e.,

$$J^{\#} = J^T(JJ^T)^{-1}. \tag{9}$$

The commanded velocity of the end-effector is computed as

$$\dot{x}_c = \dot{x}_d + Ke, \tag{10}$$

where $\dot{x}_d$ is its desired task space velocity and

$$e = x_d - x. \tag{11}$$

In eq. (11), x is the current end-effector pose. K is a diagonal gain matrix. $\nabla m(q)$ is the gradient of $m(q)$ symbolically computed via MATLAB© and projected onto the null-space

$$N = I - J^{\#}J \in \mathbb{R}^{n_j\times n_j} \tag{12}$$

associated with J for dexterity maximization purposes, as shown in fig. 5. The term $I \in \mathbb{R}^{n_j\times n_j}$ is the unit matrix.

3) Implementation: Since the Kinova Kortex and most API of industrial robots do not provide the Jacobian matrix J, we compute J by symbolically carrying out the partial derivation in eq. (3) once. The symbolic function depends upon variables (e.g., desired joint positions q_d) whose values are specified at run-time to yield the corresponding $J(q_d)$. To this end, the forward kinematics is built as the product of the successive single (4×4) homogeneous transform matrices between the reference frames attached to the links of the kinematic chain of the robot manipulator. The framework supports the transforms based upon the Roll-Pitch-Yaw and the Euler Angles conventions to yield in both cases exactly the same end-effector pose as a function of q. Once $FK(q)$ is obtained, eq. (3) is differentiated with respect to the symbolic joint variables in q to yield $J(q)$ in MATLAB©.

4) Implications for the kinematic integration of systems of systems: An advantage of our approach is the robot-agnostic treatment of Jacobian-related goals and its support for the composability of systems of systems (see r.h.s of figs. 4 and 5). Whereas most robot controllers and SDK, such as the Fast Research Interface of the KUKA LWR, provide $J(q)$ for the current q and in numerical form only, our framework returns J for *any* q, provided that $FK(q)$ is available. Furthermore, J is made available in a symbolic form that can further be processed (e.g., differentiated) for various purposes. Machine learning (e.g., Jacobian-based policy learning), operational space dynamics, model-based advanced control, and performance optimization are a few examples.

Multiple robotic subsystems can be integrated by combining their respective forward kinematics to support robot configurations made up of e.g. a robot mounted on another (see l.h.s of fig. 4). For any of these re-configurations, the forward kinematics of the system of systems is readily obtained for different goals, including the calculation of three-like kinematics and using J^T to retrieve single terms (e.g., external joint torques) governing their statics and dynamics. For a given robot, the positional part $FK_T(q) = [t_x\ t_y\ t_z]^T \in \mathbb{R}^3$ of $FK(q)$ results from $FK_T^*(q) = [FK_T^T(q)\ 1]^T$ where

$$FK_T^*(q) = T_0^{n_j}(q)x_E. \tag{13}$$

The transform $T_0^{n_j}(q)$ from the base to end-effector frame is

$$T_0^{n_j}(q) = T_0^1(q_1)T_1^2(q_2)\cdots T_{n_j-1}^{n_j}(q_{n_j}) \in \mathbb{R}^{4\times4}. \tag{14}$$

T_i^j is the homogeneous transform from frame i to j. x_E is the augmented position of a point expressed in the local end-effector frame. For instance, $x_E = [0\ 0\ 0\ 1]^T$ for the origin of this frame. The rotation part $FK_R(q) = [r_R\ r_P\ r_Y]^T \in \mathbb{R}^3$ of $FK(q)$ is extracted from the (3×3) upper-left rotation sub-matrix in $T_0^{n_j}(q)$ as Roll-, Pitch-, and Yaw-coordinates.

From the runtime perspective, note that the parameterized forward kinematics and $J(q)$ are computed *once* for a given system of systems using symbolic MATLAB© and stored

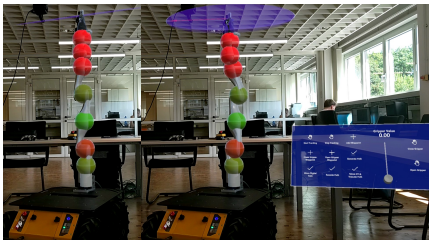


Fig. 6. Comparison of the optimized manipulability during a primary task for different enhanced joint configurations q (left: close to and right: away from the singular q_s) of a system of systems (i.e., a compound kinematically chained system made up of robot arm on top of a mobile wheeled base).

in files that can be quickly loaded upon request. Only the pre-computed result of the symbolic computation is re-used to evaluate $J(q_d)$ for any desired $q = q_d$. A simplification of mathematical expressions using MATLAB© routines substantially minimizes the computation burden.

5) Further implementation details: The manipulability is calculated for translational motions in the x, y, and z directions using $J_T(q)$ in eq. (6) (see fig. 5). The pre-computed trigonometric description related to $J_T(q)$ for the 7-DOF robot consists of 451 trigonometric function (sine and cosine) calls. J has 13k trigonometric function calls, which is large and time consuming without simplification. The computation of J in Python takes around 3 orders of magnitude longer than the computation of $J_T(q)$ while just being twice the size (6×6 vs. 3×6). We are investigating approaches and (e.g., C/C++) libraries to further reduce the computation duration as n_j increases and the robot topology becomes more complex to sustain scalability. The calculation of the manipulability uses the Singular Value Decomposition of J [16]:

$$J = U\Sigma V^T. \tag{15}$$

The singular values are the diagonal elements of Σ. The lengths of the axes of the manipulability ellipsoid are the singular values whose squared values are the eigenvalues of E. The manipulability is published as a ROS topic, which is subscribed by Unity using the ROS-TCP-Connector package. The orientation of the manipulability ellipsoid, as given by the eigenvectors in U, is converted into a quaternion and published together with the singular values as a ROS topic. A basis of N in eq. (12) is the set of the last $(n_j - rank(J))$ columns of V. In Unity, the manipulability is visualized as a semi-transparent ellipsoid at the end-effector. The ellipsoid form characterizes the joint velocity mapping ability of the robot. A large, spherical ellipsoid indicates a high uniform manipulability, while a flat, elongated ellipsoid indicates low manipulability or the proximity to a singularity (see fig. 6). When the robot arm is fully vertical, the manipulability ellipsoid becomes highly flat, indicating limited mobility in vertical direction. This visualization immediately alerts operators to avoid this configuration in order to prevent stalls or excessive joint movement for small end-effector motions. An interoperable access, integration, and interaction with other industrial and societal systems of systems is enabled by publishing J and manipulability information using data structures of the OPC UA standard. Quantitative data also depicted in figs. 3 and 5 to 7 can then be shown in a Web-Interface of any device with OPC UA-support [8].

B. Temperature capture and visualization

Monitoring temperature of robot joints is important, as it provides otherwise invisible insights into the thermal loads on the motors and electronics. Overheating can lead to a deterioration of e.g. winding insulation up to a reduced and unstable performance [14]. To help the robot operator anticipate such issues, locally and remotely, sensed temperature information related to joint drive trains is made available via OPC UA using an implementation from our previous work [9].

To ensure its reuse, the thermal monitoring was implemented as a separate component with a standardized OPC UA interface. An OPC UA Python library [21] was used to communicate with the OPC UA server. This distinct Python script can run outside of ROS as a microservice. It subscribes to measured thermal data from the OPC UA server and forwards it to ROS using websockets via the rosbridge package [22]. Temperature data are visualized in Unity as colored spheres around each joint motor for spatial immersion and thermal awareness binding (see, e.g., fig. 7). A customizable color gradient from green (cool) to red (hot) represents the temperature range from 25 °C to 60 °C in fig. 7. The temperature visualization provides an intuitive representation of the robot thermal state, allowing users to monitor thermal loads directly on physical joints and anticipate potential overheating issues on the fly (fig. 7). During a prolonged robot usage in the home posture shown in fig. 7-B, the representation for the elbow motor temperature switched from green to orange. This early warning allows the operator to adjust the robot configuration (fig. 7-C/D), decreasing joint stress and preventing potential overheating without task canceling.

V. Results and Discussion

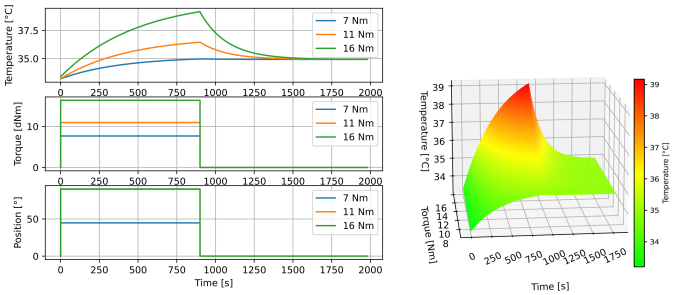
AR, DT, and OPC UA have been leveraged to deep dive into the manipulability and thermal behavior of robot arms in an interoperable way. On-device condition monitoring appears to be more user-friendly and engaging than desktop-based numerical representations while reducing inattention. Posture optimization in the null-space improves the dexterity of the arm (see figs. 5 and 6) by increasing its manipulability.

The joint torque and thermal behavior of the initially cool motor highlighted in fig. 7 are shown in fig. 8. On the l.h.s, an exponential-like increase of the temperature $T(t)$, i.e.,

$$T(t) \sim T_\infty + (T_0 - T_\infty)e^{-\frac{t}{\tau}}, \text{with } T_0, T_\infty, \tau > 0 \in \mathbb{R} \tag{16}$$

of the motor subject to an almost constant joint torque is observed over time t. Similarly, an exponential-like decay of $T(t)$ can be seen as the joint torque vanishes. The prediction of the time constant τ, steady state value T_∞, and initial temperature T_0 are useful to characterize and thus anticipate thermal issues that the motor is likely to face for given joint

Fig. 7. Minimizing thermal load on joint through informed reconfiguration.



(a) T(torque, position, time) in 2D (b) T(torque, time) in 3D

Fig. 8. Actuation of joint four and its thermal motor behavior over time.

state (i.e., position and velocity), actuation torque, and time window. The prediction, which can be achieved in a non-invasive way, as targeted by our overarching Metarobotics framework [7], is a focus of our future investigations.

On fig. 8b, the critical (red-colored) thermal area is shown on a 3D wave. A thermally-informed control/optimization of the arm posture (e.g., in a null-space as in eq. (12)) can actively keep the motor temperature away from this area. Therefore, power loss can be minimized and thermal damages are prevented. Recall that this cooling approach is similar to the AR- and DT-driven active enhancement of the manipulator agility (manipulability) through posture optimization in the Jacobian null-space. Indeed, enhancing the thermal footprint of the posture does not interfere with a primary production task associated with the end-effector from which it is decoupled at velocity level (eq. (8)). Implications include potential interplays between manipulability and thermal behavior. For instance, keeping q at the singular posture in fig. 7 A and D leads to motor cooling even under payloads. Also, which q minimizes T_∞ for a given time window? Investigating this basic research deserves further attention.

VI. Conclusion

This paper has described an AR- and DT-based framework for the real-time monitoring and shaping of internal robot properties using on-device visualization. Users can enhance the manipulability in the null-space associated with the end-effector Jacobian and anticipate motor overheating. The integration of the OPC UA standard opens up possibilities for semantic collaborations with a wide range of devices and applications. The proposed approach reveals that adopting AR and DT technology are suitable not only for situation awareness but also optimization and control purposes. Experiment results have shown that AR and DT can help extend the availability and performance of robots by accommodating their thermal behavior and enhancing their agility in an intuitive, upskilling, and non-invasive way, making advanced robotic systems of systems more engaging and accessible to a broader operator range in industrial and societal realms even remotely [10], [7]. Basic research questions and implications have emerged around the prediction of the thermal behavior of joint motors as well as the potential to design thermally efficient and dexterous robotized manipulations. Our future data-driven investigations will consider these challenges.

References

[1] R. Suzuki, A. Karim, T. Xia, H. Hedayati, and N. Marquardt, "Augmented Reality and Robotics: A Survey and Taxonomy for AR-enhanced Human-Robot Interaction and Robotic Interfaces", *Proc. of the ACM on Inter., Mobile, Wear. and Ubiq. Techn.*, 2022.

[2] G. Carriero, N. Calzone, M. Sileo, F. Pierri, F. Caccavale, and R. Mozzillo, "Human-Robot Collaboration: An Augmented Reality Toolkit for Bi-Directional Interaction", *Applied Sciences*, vol. 13, no. 20, article 11295, 2023. doi: 10.3390/app132011295.

[3] Hübel, N. & Guiffo Kaigom, E. Codeless, Inclusive, and End-to-End Robotized Manipulations by Leveraging Extended Reality and Digital Twin Technologies. *2024 16th International Conference On Human System Interaction (HSI)*. pp. 1-6 (2024)

[4] N. Hübel, S. Müller, and E. Guiffo Kaigom, "Immersive Assistance System for Intuitive Robot Programming using Mixed-Reality and Digital Twin", in *2023 IEEE 3rd International Conference on Digital Twins and Parallel Intelligence (DTPI)*, 2023. doi: 10.1109/DTPI59677.2023.10365470.

[5] Yang, W., Xiao, Q. & Zhang, Y. HA R 2 bot: a human-centered augmented reality robot programming method with the awareness of cognitive load. *Journal Of Intel. Manufacturing*. **35**, 1985-2003 (2024)

[6] Pan, J., Eden, J., Oetomo, D. & Johal, W. Effects of Shared Control on Cognitive Load and Trust in Teleoperated Trajectory Tracking. *IEEE Robotics And Automation Letters*. (2024)

[7] Kaigom, E. Metarobotics for Industry and Society: Vision, Technologies, and Opportunities. *IEEE Transactions On Industrial Informatics*. (2023)

[8] La, T.K., Harmann, R., Guiffo Kaigom, E. "Remote Monitoring and Control of Mobile Robots in Real-Time Using Multimodal Digital Twins", 28th International Conference on Methods and Models in Automation and Robotics, Poland, 2024 (Accepted).

[9] Girke, F., Harmann, R., Guiffo Kaigom, E. "Systems of Digital Twins and Physical Systems: Interoperability, Decentralization, and Mobility in Robotic Applications", 28th International Conference on Methods and Models in Automation and Robotics, Poland, 2024 (Accepted).

[10] La, T.K., Guiffo Kaigom, E. "Toward Human-Robot-Teaming for Robot Navigation Using Shared Control, Digital Twin, and Self-Supervised Traversability Prediction", 2024 IEEE Conference on Telepresence, CalTech, Pasadena, California, USA, 2024 (Accepted).

[11] KINOVA Gen3 Ultra lightweight robot User Guide", Jul. 7, 2022. [Online]. Available: https://www.kinovarobotics.com/uploads/User-Guide-Gen3-R07.pdf

[12] Microsoft, "Mixed reality news and notes", [Online]. Available: https://learn.microsoft.com/en-us/windows/mixed-reality/out-of-scope/news. [Accessed: Mar. 27, 2024].

[13] M. Quigley et al., "ROS: an open-source Robot Operating System", in ICRA Workshop on Open Source Software, 2009.

[14] Song, Z. & Liang, Y. Overview of High Overload Motors. *2023 26th International Conference On Electrical Machines And Systems (ICEMS)*. pp. 2314-2319 (2023), https://api.semanticscholar.org/CorpusID:266235352

[15] Makhataeva, Zhanat and Zhakatayev, Altay and Varol, Huseyin Atakan, "Safety Aura Visualization for Variable Impedance Actuated Robots" *2019 IEEE/SICE International Symposium on System Integration (SII)*, 2019, doi: 10.1109/SII.2019.8700332.

[16] T. Yoshikawa, "Manipulability of Robotic Mechanisms", The International Journal of Robotics Research, Jun. 1985.

[17] Stefan-Helmut Leitner and Wolfgang Mahnke. OPC UA - Service-oriented Architecture for Industrial Applications. *Softwaretechnik-Trends*, vol. 26, 2006. https://api.semanticscholar.org/CorpusID:7200881

[18] Zeid, Abe, Sarvesh Sundaram, Mohsen Moghaddam, Sagar Kamarthi, and Tucker Marion. "Interoperability in Smart Manufacturing: Research Challenges." *Machines* 2019 https://doi.org/10.3390/machines7020021.

[19] Unity Technologies, "Unity Robotics Hub", [Online]. Available: https://github.com/Unity-Technologies/Unity-Robotics-Hub. [10.03.24].

[20] Jin, L., Li, S., La, H. & Luo, X. Manipulability optimization of redundant manipulators using dynamic neural networks. *IEEE Transactions On Industrial Electronics*. **64**, 4710-4720 (2017)

[21] FreeOpcUa, "OPC UA library for python", [Online]. Available: https://github.com/FreeOpcUa/opcua-asyncio

[22] ROS, "rosbridge suite", [Online]. Available: https://wiki.ros.org/rosbridge_suite/. [Accessed: Mar. 19, 2024].