# Model Predictive Control of the Degree of Automation Optimizing Robot Health

Braun, Christian Alexander; Ramesh, Aniketh; Rothfuss, Simon; Chiou, Manolis; Stolkin, Rustam; Hohmann, Soren

*DOI:*
10.1109/SACI58269.2023.10158596

# Model Predictive Control of the Degree of Automation Optimizing Robot Health


1st Christian Alexander Braun
*Institute of Control Systems*
*Karlsruhe Institute of Technology*
Karlsruhe, Germany
christian.braun@kit.edu

2nd Aniketh Ramesh
*Extreme Robotics Lab*
*University of Birmingham*
Birmingham, United Kingdom
axr1050@student.bham.ac.uk

3rd Simon Rothfuß
*Institute of Control Systems*
*Karlsruhe Institute of Technology*
Karlsruhe, Germany
simon.rothfuss@kit.edu

4th Manolis Chiou
*Extreme Robotics Lab*
*University of Birmingham*
Birmingham, United Kingdom
m.chiou@bham.ac.uk

5th Rustam Stolkin
*Extreme Robotics Lab*
*University of Birmingham*
Birmingham, United Kingdom
r.stolkin@cs.bham.ac.uk

6th Sören Hohmann
*Institute of Control Systems*
*Karlsruhe Institute of Technology*
Karlsruhe, Germany
soeren.hohmann@kit.edu



***Abstract*—Adjusting operator support in human-machine systems is a promising way of combining operator involvement with high overall system performance. Adaptive automation aims to achieve this goal without burdening the operator with the task of selecting and setting the desired amount of support. In this work, two novel adaptive automations are presented. We use the performance measure of "robot health" to formulate the optimal control problem of maximizing the robot health of a human-robot system through the adaption of operator support to develop two model predictive controllers. The first one considers discrete levels of operator support, or levels of automation, the second one uses the continuous conception of the degree of automation. We report on a proof-of-concept simulation study evaluating the proposed model predictive controllers in a collaborative teleoperation of a mobile robot; the results demonstrate the ability of both model predictive controllers to successfully arbitrate control between the operator and the robot's controller to maximize robot health.**



***Index Terms*—Adaptive automation, Variable autonomy, Shared control, Cooperation, Degree of automation, Levels of autonomy, Levels of automation, Mixed initiative control**


## I. Introduction

The application of robots becomes increasingly important in a wide variety of fields such as industrial manufacturing [1], search and rescue operations [2], [3], healthcare [4] and planetary exploration [5]. Most of these robots are either statically teleoperated, partially automated or even fully automated; however, systems with scalable autonomy that adjust the type of interaction and amount of operator support during runtime will be valuable in future applications [3], [6]: While the ability to adjust operator support flexibilizes the application of such systems, the task of choosing and setting the optimal amount of support causes additional workload [7].

To relieve the operator form this workload, systems that are able to automate this task have been developed: Adaptive Automation systems (AAS) can automatically adjust operator support using measures of task difficulty [8], task-load [7], workload and skill-level [9], competence [10] or performance [11] to adjust operator support. While most of the previous work focuses on switching between discrete levels of automation (LOA) which is also the prevalent approach in system analysis [12], literature calling for a continuous view on LOA receives increasing attention [13]. In this work, we use the term degree of automation (DOA) to refer to this continuous conception of automation e.g. implemented as shared control in human-machine systems.

One of the most crucial parts of AAS are the measures of the considered decision variables. In this work, we focus on performance measures in human-robot systems (HRS); here, previous work either focused on offline metrics which are inapplicable to the online adjustment of LOA or DOA or on individual application-specific online performance indicators [14]–[16]. To address this issue, Ramesh et al. [16] recently introduced the framework of robot vitals and robot health to aggregate information about various runtime performance degrading effects into the online robot health performance measure. This combination of information is more generally applicable than previous measures and has been shown to be both an efficient and robust online estimate of HRS performance [16]. Hence, the question arises whether robot health can be used to adjust the amount of LOA or DOA online. To this end, this paper aims to develop and evaluate systems that optimize robot health and, thus, runtime performance by adjusting LOA or DOA in a HRS.

In [17], we already present two AAS to optimize robot health by adjusting the LOA or DOA of a HRS. Both perform a parameter optimization to determine the most suitable LOA or DOA based on a prediction model for robot health. While the parameter optimization already shows improvements over the fully manual or automated operation, it is limited to optimizing only one LOA or DOA for a considered time horizon. If the environment changes drastically during the time horizon, thus


This work was supported by NCNR under Grants EP/R02572X/1, EP/P01366X/1, and EP/P017487/1, and by ReLiB under Grant FIRG005.

requiring an adjustment of LOA or DOA tailored to each of the environmental conditions, the parameter optimization will lead to a suboptimal LOA or DOA due to being limited to the optimization of only one parameter. In contrast, we treat the problem of optimizing robot health by adjusting the LOA or DOA as an optimal control problem considering a sequence of LOAs or DOAs in this work instead. Especially during environmental changes this approach leads to a more nuanced and precise adjustment of LOA or DOA, thus leading to improved performance.

## II. The Robot Health Framework

In this work, we use a slight variation of the definition of robot health described in [16]. Information Entropy was initially used to emphasise the surprise of observing events responsible for performance degradation. However, this causes a very dynamic value trend of robot health. Given the probability of suffering of the robot $P_{\mathrm{suf}}(t)$, robot health $H(t)$ is instead defined as

$$H(t) = \int_{t-T_{\mathrm{i}}}^{t} \left(1 - P_{\mathrm{suf}}(\tau)\right) \mathrm{d}\tau \tag{1}$$

in this work with $T_{\mathrm{i}}$ denoting the time span considered for the computation of $H$. The probability of suffering is computed from individual probabilities of suffering given certain robot vitals $v_i(t)$:

$$P_{\mathrm{suf}}(t) = \frac{1}{N_{\mathrm{v}}} \sum_{i=1}^{N_{\mathrm{v}}} P_{\mathrm{suf}|v_i}(t) \tag{2}$$

Here, $N_{\mathrm{v}}$ is the number of considered robot vitals. The robot vitals capture individual performance degrading effects such as laser scanner noise, impaired movement of the robot or localization errors. Please refer to [16] for further information on the considered vitals and the modeling of $P_{\mathrm{suf}|v_i}(t)$. All in all, robot health aggregates multiple features over a time span to give a robust measure of current robot performance.

As the definition of robot health (1) features an integral, the current health cannot be affected by taking action at the current point of time but rather needs to be precisely steered over the considered time span instead. This is an important implication for the design of systems striving to optimize robot health as they cannot only react to the current health status in a feedback control fashion but rather need to predict which actions would lead to which health in the future and choose actions based on this prediction. To this end, we formulate a model predictive control (MPC) problem in the following section to apply the widely used MPC [18] concept to reach the goal of this work.

## III. Problem Statement

Fig. 1 depicts the HRS considered in this work: The inputs of the human operator $\boldsymbol{u}_{\mathrm{H}}$ and the robot control algorithm $\boldsymbol{u}_{\mathrm{A}}$ are arbitrated based on the current LOA or DOA to form a common input $\boldsymbol{u} = \Gamma(\boldsymbol{u}_{\mathrm{H}}, \boldsymbol{u}_{\mathrm{A}}, \boldsymbol{\alpha})$ before being applied to the robot. The robot states $\boldsymbol{x}$ are observed by the operator, the robot controller and the MPC; while the operator and the robot controller use this information to generate their inputs, the MPC uses the state information alongside model information about the HRS and robot health $H$ to compute the LOA or DOA sequence $\boldsymbol{\alpha}^*(t)$ that optimizes $H$ over a finite time horizon with the length $T_{\mathrm{H}}$. Due to the receding horizon principle of model predictive control, only the first LOA or DOA of the sequence $\boldsymbol{\alpha}(t_0)$ computed for the current time step $t_0$ is applied to the arbitration module. The remaining part of the sequence (for $t \in [t_0 + T_{\mathrm{c}}, T_{\mathrm{H}}]$) is updated in the following MPC cycles with a cycle time of $T_{\mathrm{c}}$ before being applied to the system. Following the majority of MPC literature, we consider the MPC's input to the system $\boldsymbol{\alpha}$ to be constant during a cycle $t \in [t_0, t_0 + T_{\mathrm{c}}]$. Throughout this paper we mostly omit time-dependencies for better readability; however, we do explicitly show them if they are significant for the expression.

As the goal of the adjustment of LOA or DOA is to optimize robot health over $T_{\mathrm{H}}$, the objective function can be formulated as follows:

$$\boldsymbol{\alpha}^*(t) = \underset{\boldsymbol{\alpha}(t)}{\operatorname{argmax}}\ H(t_0 + T_{\mathrm{H}}), \quad t \in [t_0, t_0 + T_{\mathrm{H}}] \tag{3}$$

As $\boldsymbol{\alpha}$ remains constant during each MPC cycle, (3) is the problem of finding an optimal sequence of $N_H = T_{\mathrm{H}} T_{\mathrm{c}}^{-1}$ LOAs or DOAs. In contrast to our previous work [17], (3) thus constitutes an optimal control problem as $\boldsymbol{\alpha}(t)$ is treated as a sequence rather than a system parameter fixed for $T_{\mathrm{H}}$. The parameter optimization problem considered in [17] is the special case of this optimal control problem if $T_{\mathrm{C}} = T_{\mathrm{H}}$ is set.

We consider the two following problems in order to design the LOA- or DOA-MPCs; while both strive to optimize the objective function (3), they differ in the arguments $\boldsymbol{\alpha}(t)$ available for doing so:

*Problem 1: MPC for the LOA-case.* Design a LOA-MPC module optimizing (3) s.t. the model information about the HRS described above considering $\boldsymbol{\alpha}(t) \in \mathcal{L}$ with $\mathcal{L}$ denoting the countable set of all sequences ($N_{\mathrm{H}}$-tuples) that can be created from the set of considered LOAs $\Lambda$.

*Problem 2: MPC for the DOA-case.* Design a DOA-MPC module optimizing (3) s.t. the model information about the HRS described above considering $\boldsymbol{\alpha}(t) \in \mathcal{D}$ with $\mathcal{D}$ denoting the uncountable set of all sequences ($N_{\mathrm{H}}$-tuples) that can be created from the set of considered DOAs $[0, 1]$.

As $\Lambda \subset [0, 1]$ always holds, Problem 1 is always a special case of Problem 2. However, due to its structure, Problem 1 allows for solution concepts that are infeasible for Problem 2. Thus, we consider Problem 1 both individually as well as implicitly included in Problem 2.

## IV. Model Predictive Control of LOA and DOA

### A. LOA-MPC

As the set of possible LOAs $\Lambda$ is countable, the number of possible sequences of length $T_{\mathrm{H}}$, or $N_{\mathrm{H}}$-tuples, is $N_{\mathrm{S}} = |\Lambda|^{N_{\mathrm{H}}}$. Here, $|\Lambda|$ denotes the cardinality of $\Lambda$. Since $N_{\mathrm{S}}$ is finite, Problem 1 allows for the evaluation of the performance of all possible solutions and the subsequent selection of the

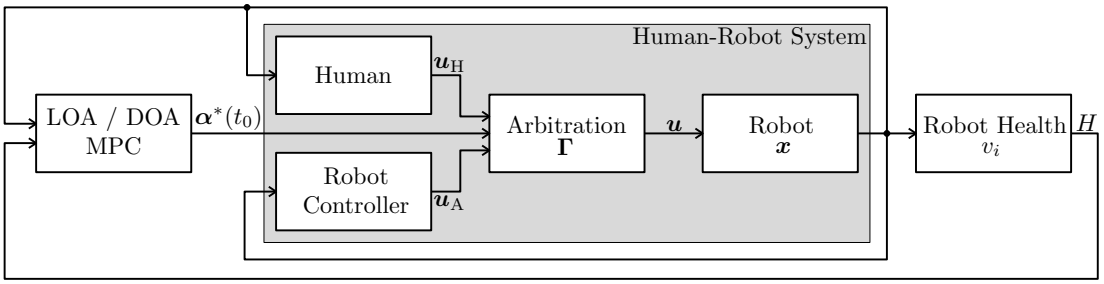



Fig. 1: Structure of the considered adaptive automation system using either a LOA-MPC or a DOA-MPC.

best one. Fig. 2 visualizes this approach where $N_\mathrm{S}$ model instances of the HRS are initialized with the currently observed states and evaluated for $t \in [t_0, t_0 + T_\mathrm{H}]$ using a fixed sequence $\boldsymbol{\alpha}_i(t),\ i \in \{1, ..., N_\mathrm{S}\}$. Subsequently, the resulting robot health $H_i$ can be computed and the optimal solution $\boldsymbol{\alpha}_j(t)$ can be determined by comparing the resulting robot health values to select $H_j$, the one with the highest value.

This approach is of interest as it is guaranteed to find the globally optimal solution and can easily be implemented in a parallel fashion since the time-consuming evaluations of the HRS models are mutually independent.

### B. DOA-MPC

Considering Problem 2, the set of possible DOAs is uncountable, hence the number of possible sequences is infinite. Thus, the system proposed in Subsection IV-A is no longer applicable as an infinite amount of model evaluations would be required.

Following most of the MPC literature, we propose to address this issue by using an iterative optimization algorithm instead. The resulting structure is depicted in Fig. 3: The HRS model is again initialized with the currently observed system states in each MPC cycle, but instead of evaluating all possible sequences, only the sequences $\boldsymbol{\alpha}_\mathrm{opt}(t)$ selected by the optimization algorithm are evaluated. The resulting robot health values are used to iteratively update the currently best solution until convergence is reached. As the solution of the previous MPC cycle is, apart from a time shift of $T_\mathrm{c}$, likely similar to the optimal solution for the current cycle, we always initialize the optimization with the solution of the previous iteration shifted by $T_\mathrm{c}$ to improve computational efficiency.

Due to the usage of an iterative optimization, no general statements about optimality or even convergence can be made; however, in most practical cases at least a locally optimal solution can be achieved. As $\Lambda \subset [0, 1]$ always holds, all possible solutions of LOA-MPCs are solution candidates for a DOA-MPC, too. Hence, the potential solution performance of DOA-MPC is always at least equal to the one of an LOA-MPC.

## V. Simulation Results

In order to assess the systems presented in Section IV, a proof-of-concept study was carried out in simulation. We applied both LOA- and DOA-MPC to a collaborative teleoperation of a 2-D mobile ground robot traversing a previously scouted environment. As we focus on the general functionality and effects of LOA- and DOA-MPC, perfect model knowledge of the modules depicted in Fig. 1 is assumed. While the LOA or DOA sequences are computed quasi-time-discrete, all simulations consider the overall system including the MPCs in a continuous-time fashion. The results presented in the following were achieved by creating a Simulink model of Fig. 1 using Simulink 10.6 and implementing LOA- and DOA-MPC using MATLAB 9.13 (R2022b). The scenario presented below is similar but not identical to the one used in [17].

### A. Scenario

Fig. 4 gives an overview of the considered scenario. The initial robot pose is depicted with a triangle, the location of the

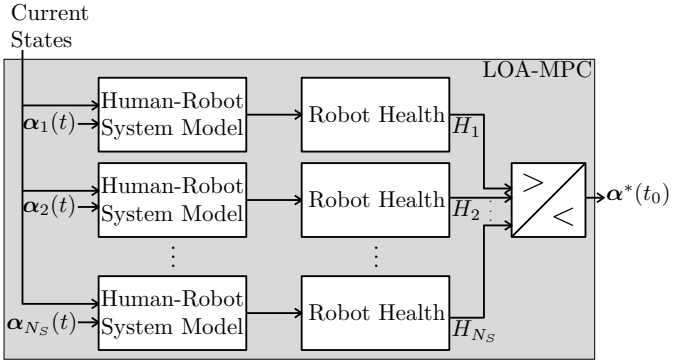



Fig. 2: Structure of the LOA-MPC.

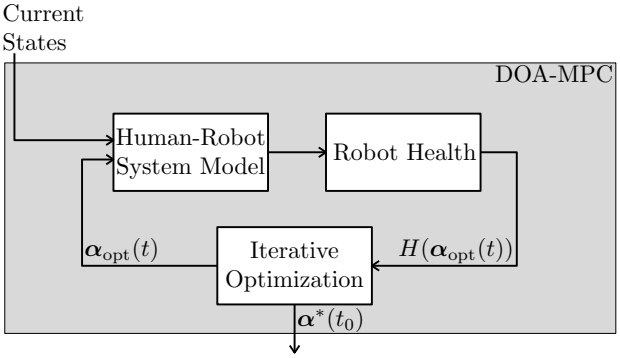



Fig. 3: Structure of the DOA-MPC.

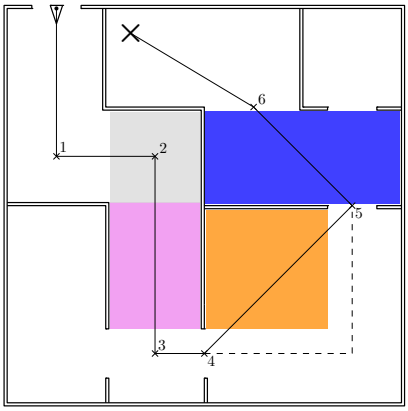


Fig. 4: Map of the considered scenario.

goal is indicated by a large cross and the six intermediate way-points are marked with labeled crosses. During the traverse to the goal, four areas featuring adverse conditions have to be passed by the robot: 1) In the grey area, noise affects the laser scanner, forcing the robot's controller (RC) to limit its range of commanded velocities to $10\%$ of the maximum velocity. 2) In the pink area, the operator gets distracted and reduces their range of commanded velocities to $30\%$ of the maximum velocity. The laser scanners are still impaired by noise in this area, however, the noise is less severe than in the grey area allowing the RC to use up to $60\%$ of its nominal input range. 3) The orange area features bumpy terrain; all other parts of the map are even. The information about the terrain is unknown to the RC which would result in using the solid path between way-point 4 and 5. However, the human is able to recognize the terrain and operate the robot to take the dashed path to bypass the uneven terrain. 4) While traversing the blue area, the RC faces an unexpected malfunction limiting its input range to $30\%$ of the maximum velocity without any environmental influences causing this. All areas colored in white are unaffected by performance degrading factors.

We model the mobile robot with the following unicycle dynamics, similar e.g. to [19]:

$$\dot{\boldsymbol{x}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{pmatrix} \boldsymbol{u} \tag{4}$$

Here, $\boldsymbol{x}$ is the robot's state with $x$ and $y$ denoting its position and $\theta$ representing its heading. The robot can be operated via the inputs $\boldsymbol{u} = (\beta, \omega)^\top$. $\beta$ denotes the robot's velocity in the direction of the heading and $\omega$ is its angular velocity.

LOAs and DOAs are varied using a linear policy blending $\boldsymbol{\Gamma} : \mathbb{R}^2 \times \mathbb{R}^2 \times [0,1] \to \mathbb{R}^2$ to arbitrate the input of the RC $\boldsymbol{u}_\mathrm{A}$ and the commands issued by the operator $\boldsymbol{u}_\mathrm{H}$ [20]:

$$\boldsymbol{\Gamma}_\mathrm{pb}(\boldsymbol{u}_\mathrm{A}, \boldsymbol{u}_\mathrm{H}, \boldsymbol{\alpha}) = \alpha \begin{pmatrix} \beta_\mathrm{A} \\ \omega_\mathrm{A} \end{pmatrix} + (1-\alpha) \begin{pmatrix} \beta_\mathrm{H} \\ \omega_\mathrm{H} \end{pmatrix} \tag{5}$$

Here, $\alpha$ allows for control authority shifts ranging from manual control ($\alpha = 0$), over shared control ($\alpha \in (0,1)$) to a fully automated robot operation ($\alpha = 1$). Depending on whether the LOA- or DOA-case is considered, $\alpha$ can either be chosen from the set $\Lambda$ featuring discrete values or from the full spectrum of possible DOAs, i. e. $\alpha \in [0,1]$.

To compute robot health based on $\boldsymbol{x}$, we included the three vitals capturing the performance degrading effects introduced in the description of the scenario from the ones presented in [16]: The vital describing the rate of change of distance from the navigational goal (ROCODO), the vital assessing the jerk along the $z$-axis of the robot, and the vital monitoring the laser scanner noise. The rate of change $\dot{d}_\mathrm{g}$ of the ROCODO vital was computed with respect to the currently active way-points $(x_{\mathrm{g,i}}, y_{\mathrm{g,i}})$, $i \in \{1, ..., 7\}$:

$$\dot{d}_\mathrm{g} = \frac{1}{\|\boldsymbol{d}_\mathrm{g}\|_2} \boldsymbol{d}_\mathrm{g}^\top \begin{pmatrix} \beta\cos(\theta) \\ \beta\sin(\theta) \end{pmatrix} \text{ with } \boldsymbol{d}_\mathrm{g} = \begin{pmatrix} x_{\mathrm{g,i}} - x \\ y_{\mathrm{g,i}} - y \end{pmatrix} \tag{6}$$

The structure of the vitals is implemented as described in [16], the parameters of the mappings are fine-tuned to the scenario described above.

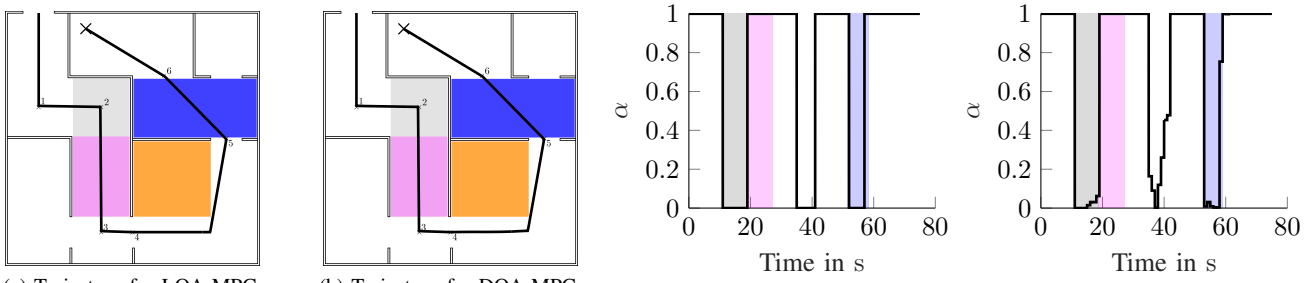
The human operator models are based on the experience from extensive teleoperation experiments during the ARCHES project [5]: When told to pursue a goal point via way-points as described above, the operators usually first turned the robot until the correct heading had been reached and then commanded velocities subsequently. This behavior is modeled by switching between a proportional controller for $\theta$ and a proportional controller for $\boldsymbol{d}_\mathrm{g}$ using the inputs $\omega$ and $\beta$. Apart from different parametrization and different types of performance degradation, the model of the RC is identical to the one of the operator as this structure was also successfully applied in [5].

### *B. Results of LOA-MPC*

To evaluate LOA-MPC we consider a traded control setting, where either the human or the operator is in charge of commanding the robot. Hence, the set of admissible LOAs is $\Lambda = \{0, 1\}$. Fig. 5a shows the resulting trajectory of the robot. LOA-MPC is able to successfully arbitrate control between the operator and the RC to pass all the way-points and reach the goal state.

The LOA evolution depicted in Fig. 5c shows that LOA-MPC successfully shifts control to the human while traversing the grey area where the RC is impaired and then hands control back over to the RC once the pink area is reached where the operator is distracted and the RC faces less noise. To avoid the orange area, the human receives control authority at around 35s, however, instead of completely following the overly conservative dashed path of the human, control is shifted to the RC to barely avoid the bumpy orange area while maximizing the ROCODO vital. Despite not being captured by any of the environmental vitals, control is correctly handed to the human during the malfunction of the RC in the blue area as it is recognized by the ROCODO vital and then handed back to the RC to reach the goal.

LOA-MPC provides a reasonable arbitration of control authority, giving it either to the operator or to the RC based on who is predicted to have the best capability of handling




(a) Trajectory for LOA-MPC. (b) Trajectory for DOA-MPC. (c) Results of LOA-MPC. (d) Results of DOA-MPC.

Fig. 5: Simulation results of LOA-MPC and DOA-MPC using $T_\mathrm{H} = 4\mathrm{s}$ and $T_\mathrm{c} = 1s$. Background colors in Fig. 5c and 5d indicate the traverse of the equally colored areas in Fig. 5a and Fig. 5b.

the task thus maximizing robot health and consequentially maximizing the overall performance of the HRS.

### C. Results of DOA-MPC

While the models considered in this scenario are given analytically which would allow for using gradient-based optimization algorithms, we strive for a system that is flexible in terms of the used model type. The real-world implementations we are planning to apply in the future might feature grey- or even black-box models without gradient information. Hence, we chose to use the gradient-free pattern search algorithm [21] for the implementation of DOA-MPC.

The robot trajectory achieved with this system is depicted in Fig. 5b. Again, all way-points are precisely passed and the goal state is reached. Fig. 5d shows the evolution of the DOA chosen by DOA-MPC. Similar to LOA-MPC, control authority is given to the operator in the grey and blue areas as well as to avoid the orange area, while the RC is used in the pink area. In contrast to the results of LOA-MPC, DOA-MPC uses a collaborative shared control at approximately 38s to smoothly shift control back to the automation. This allows for an even more efficient passing of the orange area as the RC is able to start steering the robot towards way-point 5 even before the orange area is fully passed (slightly visible before the turn close to the orange area in Fig. 5b), thus increasing robot health because of the ROCODO vital.

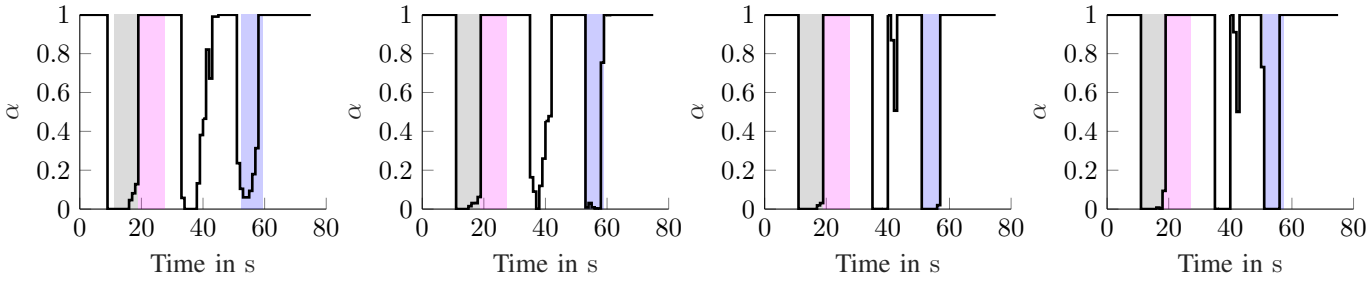
Apart from the collaborative time span, slight artifacts can be observed in the grey and blue area. The reason for this is the rather coarse MPC cycle-time $T_\mathrm{c}$ of 1s: The transition e.g. from the grey to the pink area would have happened exactly within one cycle, thus leading to a DOA only optimal for half of the cycle time. To avoid this, DOA-MPC gives a slight amount of control to the RC before the transition leading to a slightly slower traverse (and thus slightly lower robot health), this way synchronizing the transition of the gray to the pink area with the beginning of a new cycle time. This leads to an optimal DOA for the whole cycle $T_\mathrm{c}$ by sacrificing only a small amount of performance in the previous cycles. These artifacts can be removed by decreasing $T_\mathrm{c}$, however they showcase the capability of the DOA-MPC to use even small effects to optimize robot health and thus performance.

In addition to $T_\mathrm{c}$, the considered time horizon $T_\mathrm{H}$ plays an important role, too, as it scales the amount of information available for planning the optimal DOA. To analyze its impact, we conducted simulations with varying $T_\mathrm{H}$; the results are depicted in Fig. 6. Fig. 6a shows the special case of a parameter optimization similar to [17], Fig. 6b and Fig. 6c result from setting $T_\mathrm{H} = 4\mathrm{s}$ and $T_\mathrm{H} = 15\mathrm{s}$, respectively, and Fig. 6d shows the result if $T_\mathrm{H}$ covers the full scenario. Despite their significantly different $T_\mathrm{H}$, Fig. 6c and Fig. 6d achieve very similar results and use shared control only very aimed. Fig. 6a shows a more elaborate use of shared control which stems from an averaging effect: As only one DOA can be optimized for the time horizon, an intermediate DOA is chosen as a compromise if the environment changes within the time horizon.

Considering the overall robot health averaged over the duration of the scenario, the configurations of Fig. 6 achieve the values given in Table I: All of the DOA-MPC configurations significantly outperform a fully manual ($\alpha(t) = 0\ \forall t$) or fully automated ($\alpha(t) = 1\ \forall t$) operation of the robot. Increasing $T_\mathrm{H}$ and thus increasing the information available for computing the DOA improves the achieved robot health monotonously with the configuration considering the whole duration performing best. Nevertheless, the comparable DOA evolution of $T_\mathrm{H} = 15\mathrm{s}$ also leads to comparable robot health. Despite the significantly different DOA sequence shown in Fig. 6a and Fig. 6b, their achieved robot health is still far better than in the fully manual and fully automated case, thus indicating that DOA-MPCs even of relatively low computational complexity might provide a significant benefit for applications.

TABLE I: Comparison of the average robot health. A higher value means less performance degradation.

| Configuration | Average health per second |
|---|---|
| Fully automated | 0.7583 |
| Fully manual | 0.9380 |
| $T_\mathrm{H} = 4\mathrm{s},\ T_\mathrm{c} = 4s$ | 0.9860 |
| $T_\mathrm{H} = 4\mathrm{s},\ T_\mathrm{c} = 1s$ | 0.9878 |
| $T_\mathrm{H} = 15\mathrm{s},\ T_\mathrm{c} = 1s$ | 0.9892 |
| $T_\mathrm{H} = 75\mathrm{s},\ T_\mathrm{c} = 1s$ | 0.9895 |




(a) DOA-MPC using $T_{\mathrm{H}} = 4\mathrm{s}$ and $T_{\mathrm{c}} = 4\mathrm{s}$.

(b) DOA-MPC using $T_{\mathrm{H}} = 4\mathrm{s}$ and $T_{\mathrm{c}} = 1\mathrm{s}$.

(c) DOA-MPC using $T_{\mathrm{H}} = 15\mathrm{s}$ and $T_{\mathrm{c}} = 1\mathrm{s}$.

(d) DOA-MPC using $T_{\mathrm{H}} = 75\mathrm{s}$ and $T_{\mathrm{c}} = 1\mathrm{s}$.

Fig. 6: Comparison of DOA-MPC results for multiple $T_{\mathrm{H}}$. Background colors indicate the traverse of the equally colored areas.

## VI. Conclusion and Future Work

In this paper, we contribute two MPCs for the online adjustment of the level or degree of automation to optimize robot health. The proof-of-concept simulation study shows their applicability as well as their ability to significantly increase robot health even for relatively small prediction time horizons.

Currently, we are preparing a user study to evaluate how humans interact with the proposed systems in real-world robotic scenarios focusing on performance, human factors and the ability of the systems to deal with model errors.

## References


[1] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[2] J. Delmerico, S. Mintchev, A. Giusti, B. Gromov, K. Melo, T. Horvat, C. Cadena, M. Hutter, A. Ijspeert, D. Floreano *et al.*, "The current state and future outlook of rescue robotics," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1171–1191, 2019.

[3] M. Chiou, G.-T. Epsimos, G. Nikolaou, P. Pappas, G. Petousakis, S. Mühl, and R. Stolkin, "Robot-assisted nuclear disaster response: Report and insights from a field exercise," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4545–4552.

[4] M. Kyrarini, F. Lygerakis, A. Rajavenkatanarayanan, C. Sevastopoulos, H. R. Nambiappan, K. K. Chaitanya, A. R. Babu, J. Mathew, and F. Makedon, "A Survey of Robots in Healthcare," *Technologies*, vol. 9, no. 1, p. 8, 2021.

[5] A. Wedler, M. G. Müller, M. Schuster, M. Durner, S. Brunner, P. Lehner, H. Lehner, A. Dömel, M. Vayugundla, F. Steidle *et al.*, "Preliminary results for the multi-robot, multi-partner, multi-mission, planetary exploration analogue campaign on mount etna," in *Proceedings of the International Astronautical Congress, IAC*, 2021.

[6] N. Y.-S. Lii, P. Schmaus, D. Leidner, T. Krueger, J. Grenouilleau, A. Pereira, A. Giuliano, A. S. Bauer, A. Köpken, F. S. Lay *et al.*, "Introduction to surface avatar: the first heterogeneous robotic team to be commanded with scalable autonomy from the iss," in *Proceedings of the International Astronautical Congress, IAC*. International Astronautical Federation, IAF, 2022.

[7] E. de Visser and R. Parasuraman, "Adaptive aiding of human-robot teaming: Effects of imperfect automation on performance, trust, and workload," *Journal of Cognitive Engineering and Decision Making*, vol. 5, no. 2, pp. 209–231, 2011.

[8] K. Cosenzo, J. Chen, L. Reinerman-Jones, M. Barnes, and D. Nicholson, "Adaptive automation effects on operator performance during a reconnaissance mission with an unmanned ground vehicle," in *Proceedings of the human factors and ergonomics society annual meeting*, vol. 54, no. 25. SAGE Publications Sage CA: Los Angeles, CA, 2010, pp. 2135–2139.

[9] S. K. Nittala, C. P. Elkin, J. M. Kiker, R. Meyer, J. Curro, A. K. Reiter, K. S. Xu, and V. K. Devabhaktuni, "Pilot skill level and workload prediction for sliding-scale autonomy," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 1166–1173.

[10] C. Basich, J. Svegliato, K. H. Wray, S. Witwicki, J. Biswas, and S. Zilberstein, "Learning to optimize autonomy in competence-aware systems," 2020. [Online]. Available: https://arxiv.org/abs/2003.07745

[11] G. Calhoun, H. Ruff, E. Frost, S. Bowman, J. Bartik, and K. Behymer, "Performance-based adaptive automation: Number of task types and response time measures triggering level of automation changes," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 65, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2021, pp. 37–41.

[12] M. Vagia, A. A. Transeth, and S. A. Fjerdingen, "A literature review on the levels of automation during the years. What are the different taxonomies that have been proposed?" *Applied Ergonomics*, vol. 53, pp. 190–202, 2016.

[13] C. A. Miller, "The risks of discretization: what is lost in (even good) levels-of-automation schemes," *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 74–76, 2018.

[14] E. Khalastchi and M. Kalech, "On fault detection and diagnosis in robotic systems," *ACM Computing Surveys (CSUR)*, vol. 51, no. 1, pp. 1–24, 2018.

[15] L. Yu, M. Wu, Z. Cai, and Y. Cao, "A particle filter and svm integration framework for fault-proneness prediction in robot dead reckoning system," *WSEAS Transactions on Systems*, vol. 10, no. 11, pp. 363–375, 2011.

[16] A. Ramesh, R. Stolkin, and M. Chiou, "Robot vitals and robot health: Towards systematically quantifying runtime performance degradation in robots under adverse conditions," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 729–10 736, 2022.

[17] C. A. Braun, A. Ramesh, S. Rothfuß, M. Chiou, R. Stolkin, and S. Hohmann, "Model predictive degree of automation regulation for mobile robots using robot vitals and robot health," *IFAC-PapersOnLine*, 2023, 22nd IFAC World Congress, under review.

[18] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer science & business media, 2013.

[19] A. Broggi, P. Cerri, S. Debattisti, M. C. Laghi, P. Medici, D. Molinari, M. Panciroli, and A. Prioletti, "Proud—public road urban driverless-car test," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3508–3519, 2015.

[20] A. D. Dragan and S. S.Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[21] C. Audet and J. E. Dennis Jr, "Analysis of generalized pattern searches," *SIAM Journal on optimization*, vol. 13, no. 3, pp. 889–903, 2002.