How to Systematically Distribute Candidate Models and Robust Controllers in Multiple-Model Adaptive Control: A Coverage Control Approach

Stefan Kersting, *Student Member, IEEE*, and Martin Buss, *Fellow, IEEE*

***Abstract*—Distributing nominal models in multiple-models applications constitutes a long standing problem. The set of models needs to be distributed in such a way that their corresponding controllers can stabilize all possible system configurations in a large uncertainty set. This paper presents a systematic solution by phrasing the distribution as coverage control problem, in which each model covers a subset of the uncertainty. The subsets are derived based on a combination of the ν-gap metric, which serves as a distance measure, and the generalized stability margin. Characterizing coverage in terms of the ν-gap also motivates the use of $\mathcal{H}_\infty$ controller synthesis to design a set of controllers. The proposed algorithms are initialized with suboptimal model configurations. Two update laws optimize the model parameters and minimize the coverage function. The first algorithm performs a gradient descent on the coverage function and the second algorithm performs pairwise optimizations. Due to computational complexity, a discretized implementation is derived, which reduces the optimization to an efficient graph search. The proposed algorithms are evaluated in numeric benchmark examples.**

***Index Terms*—Coverage control, multiple model adaptive control (MMAC), robust control.**

I. Introduction

UNCERTAIN models are a common challenge in the control of dynamical systems. In order to deal with small uncertainties, research on robust control theory led to various schemes, such as μ-synthesis or $\mathcal{H}_\infty$ loop-shaping, to design a single controller with fixed parameters, which is capable of robustly stabilizing all systems from a given uncertainty set with specified performance properties. For large uncertainties, however, a single robust controller, stabilizing all possible parameter configurations, may be infeasible. Consider for instance, faults in the system, which result in considerable changes in parameters and therefore render the nominal controller unable to stabilize the faulty system. Hence, very large uncertainties require adaptive mechanisms in order to adjust the controller in such a way that it matches the current system parameters.

One direction followed in adaptive control is to continuously tune the controller gains and show that adaptation causes the gains to converge to values ensuring both stability and tracking. Difficulties in continuous adaptation arise in case the parameter change is large and sudden, which induces undesirable transient effects. In order to overcome this limitation, the concept of multiple model adaptive control with switching and tuning [1], [2] may be helpful. By reinitializing adaptation from a finite set of possible parameter configurations, i.e., multiple models, the transient effects are reduced. Note that the continuous adaptation of control gains in [1] and [2] precludes the application of robust control concepts due to the complicated relationship between robust and adaptive control structures.

Later, it was noted that neglecting the continuous tuning of control gains enables the application of results from robust control in the multiple models framework. Such algorithms are referred to as *Multiple Model Adaptive Control (MMAC)* [3]–[5] or *Supervisory Control* [6]–[10]. They have in common that the large uncertainty set is divided into smaller subsets, such that for each subset there exists a fixed controller to robustly stabilize all possible systems in the subset. In [3], it was shown that, under mild assumptions on the uncertainty, a finite set of candidate model-controller pairs suffices to cover the uncertainty set. The task then is to select at each time instance from the set of candidate controllers the most suitable one. For that purpose, a first component of the framework characterizes the match between the true system and the multiple models. This is often carried out by predicting the output of the true systems with multiestimators (observers) or a bank of Kalman filters. Then, performance signals are defined as the conditional probabilities obtained from the Kalman filters or by suitably integrating the prediction errors. Based on the performance signals, a second component, frequently referred to as supervisor, selects, weights or blends the candidate controllers. In early papers, the focus was on a discontinuous switching logic, which selects one of the controllers and puts it into feedback with the systems. In that context, various switching strategies haven been investigated involving dwell-time constraints and hysteresis switching to ensure safe switching [4], [5], [9]. Stability is concluded for constant system

Manuscript received April 8, 2017; revised July 8, 2017; accepted July 14, 2017. Date of publication August 9, 2017; date of current version March 27, 2018. This work was supported in part by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ERC under Grant 267877 and in part by the Technische Universität München—Institute for Advanced Study (www.tum-ias.de), funded by the German Excellence Initiative. Recommended by Associate Editor D. Dochain. *(Corresponding author: Stefan Kersting.)*

The authors are with the Chair of Automatic Control Engineering, TUM Institute for Advanced Study, Technische Universität München, München 80333, Germany (e-mail: stefan.kersting@tum.de; mb@tum.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2017.2731946

parameters if the switching logic ceases switching in finite time and settles upon a stabilizing controller. One problem arising in discontinuous switching schemes are transients induced by controller initialization. Some results on bumpless transfer [11] and optimal reset maps [12] aim to reduce such effects. Besides hard switching among candidate controllers, some mixing and blending strategies have been proposed [13]–[16]. Overall, adaptive schemes which rely on mixing or blending replace the discontinuous switching logic with a smooth and stable interpolation. While this avoids undesirable effects of the switching behavior, such algorithms react slower to large parameter changes than switching-based algorithms can.

The unfalsified control framework [17], [18] introduced by Safonov *et al.* is closely related to MMAC algorithms in a sense that an uncertain system is stabilized by one controller taken from a set of controllers. The difference is that the controller set is designed without a set of nominal models, and the controllers are selected with the help of fictitious reference signals that enable a virtual evaluation of all controllers at the same time. Poorly performing controllers are falsified (discarded) until a set of unfalsified controllers remains, which satisfy the given performance specifications. Recent extensions by Baldi *et al.* introduced more robustness in the controller design and extended unfalsified control to multiinput systems [19], [20].

In conclusion, MMAC can be decomposed into designing switching strategies and finding a finite set of controllers to cover the uncertainty set. While a brief introduction into the framework including switching will be given, the primary focus of this paper is on the systematic controller distribution. Approximately optimal algorithms to determine the minimum number of nominal models needed and their distribution in the uncertainty set will be derived. Unavoidable problem approximations and assumptions leading to suboptimal solutions will be highlighted and discussed as they are introduced.

The performance of multiple models-based algorithms depends to a large extent on the selected set of candidate controllers. The dominant questions in this context are: How many candidate models are needed and how do they need to be distributed in the uncertainty set to achieve the best result? Most publications on the topic, however, avoid the question about how to optimally select candidate model-controller pairs. Most often, a set of suitable candidate controllers is simply assumed to be given [9], [21], [22] or obtained by trial and error [6]. Some authors acknowledge the design of candidate controller sets as a challenging open problem [3], [13], [22]. Anderson *et al.* for instance conclude with "It further remains to find a computationally efficient method to determine such a finite set, so that the number of models needed is not overly conservative" [3].

In spite of its importance, only a few systematic algorithms exist for determining the model distribution in MMAC. Fekri *et al.* in [13] propose an approach to determine the required number of models in their robust MMAC algorithm, which is essentially a greedy search, i.e., beginning from the upper bound of the uncertainty set, the subset assigned to the first model is extended until the desired performance is reached. At this point, the next model is initialized and the process is repeated until the entire uncertainty set is covered. The increased complexity of such greedy algorithms in higher dimensional parameter spaces constitutes the main limitation of this approach. A similar greedy approach involving the ν-gap metric [23], [24] to characterize stability margins was proposed in [3] with the same limitations. In [25], four heuristics also involving the ν-gap metric are given for determining model distributions, one of which is a more systematic version of the greedy approach in [3]. A bisectional method to find a robustly stabilizable partitioning of the initial uncertainty set is proposed in [26]. Starting with the entire uncertainty set as the first subset, the proposed algorithm performs repeated bisections of all subsets for which a single robust controller is infeasible. Finally, for various early algorithms which did not incorporate the desired robustness properties, see [27] and the references therein.

In this paper, we propose a new systematic approach for model distribution in MMAC. In contrast to existing greedy search algorithms, the proposed approach builds upon a formal optimization problem formulated with tools form the robust control literature. More precisely, we make use of the ν-gap metric to characterize distances between candidate model-controller pairs and the uncertain plant configurations. Furthermore, the ν-gap metric enables stability statements to be made. Finally, robust candidate controllers are designed with standard $\mathcal{H}_\infty$ tools. To solve the optimization problem, we propose two coverage control-inspired algorithms. With both algorithms, two types of optimization are practicable. Either, the number of model-controller pairs may be reduced while ensuring stabilizability of all systems in the uncertainty set. Or, for a fixed number of model-controller pairs, their distribution can be optimized to increase control performance as rated by the scalar coverage functional. This paper demonstrates the effectiveness of the proposed algorithms in the context of two benchmark examples.

Note that MMAC is a highly modular framework in which various units such as estimators, candidate controllers or the design of switching strategies can be considered independently and independent methodologies can be easily substituted. A very detailed and modular treatment of MMAC is given in [28] and [29]. Hence, the synthesis of a finite controller covering presented in this paper is quite general and can be combined with other switching strategies or frameworks related to MMAC. Also note that the considered question of how to divide uncertainty sets also has applications in the field of linear time varying (LTV) and linear parameter varying (LPV) systems as pointed out for instance in [30]: "How to divide and what the number of subsets is are interesting research questions of their own." Also in [31], the gap metric is used to determine the operating points of local linear models in LPV systems. The LPV system is then obtained by interpolating the local models. Another greedy search algorithm to increase validity regions in LPV systems is discussed in [32]. Besides LTV and LPV system synthesis, another field of possible application for the proposed coverage control algorithms is gain scheduling.

The remainder of this paper is structured as follows. First, Section II introduces the MMAC framework. Then, Section III revises some robust control results needed to characterize a finite controller covering of the uncertainty set and design suitable

robust controllers. The problem of optimal model distribution is formulated in Section IV as a coverage control problem, for which two algorithms are proposed. Testing the proposed algorithms in two benchmark simulations is the subject of Section V. Conclusions are drawn in Section VI.

II. Multiple Models and Supervisory Control

Consider a system $\mathcal{G}(q)$ with uncertain parameter vector $q \in \mathbb{R}^n$ taking values from the compact *uncertainty set* $\mathcal{Q} \subset \mathbb{R}^n$. More precisely, let the system matrices $A(q)$, $B(q)$, $C(q)$ and $D(q)$ of the state-space realization define

$$\mathcal{G}(q) \quad \begin{cases} \dot{x} = A(q)x + B(q)u \\ y = C(q)x + D(q)u \end{cases} \tag{1}$$

and depend continuously on q. Assume furthermore that the uncertainty set $\mathcal{Q}$ is too large for a single fixed controller to stabilize all possible configurations. A central result in the MMAC literature [3, Th. 2.1] states that, under the mild assumption of continuity of $\mathcal{G}(q)$ in the ν-gap metric, there exists a finite set of controllers, whose corresponding set of robustly stabilized plants covers the uncertainty set. In other words, we can always find a finite set of candidate controllers $\{\mathcal{C}_i\}_{i=1}^N$ such that each system $\mathcal{G}(q)$ in the uncertainty set $\mathcal{Q}$ is stabilized by at least one candidate controller. Let i be an index taking values from the index set $\mathcal{I} := \{1, \ldots, N\}$ to express which one of the N models or controllers is considered.

One way to design the set of candidate controllers is to choose a finite set of *nominal configurations* $p_i \in \mathcal{Q}$ and design a robust controller for the corresponding *nominal model* $\mathcal{G}(p_i)$. How many and how to choose the nominal configurations will be presented in Section IV as the main contribution of this paper. Given the set of nominal model-controller pairs, MMAC evaluates which nominal model best describes the current system input–output behavior and applies the corresponding nominal controller. Hence, the heart of every MMAC algorithm is a multicontroller

$$\begin{aligned} \dot{x}_{\mathcal{C}} &= A_{\mathcal{C}_{\sigma(t)}} x_{\mathcal{C}} + B_{\mathcal{C}_{\sigma(t)}} y \\ u &= C_{\mathcal{C}_{\sigma(t)}} x_{\mathcal{C}} + D_{\mathcal{C}_{\sigma(t)}} y \end{aligned} \tag{2}$$

with shared state $x_{\mathcal{C}}$, whose matrices switch between the state-space realizations of *nominal controllers* $\mathcal{C}_i := \{A_{\mathcal{C}_i}, B_{\mathcal{C}_i}, C_{\mathcal{C}_i}, D_{\mathcal{C}_i}\}$ designed for the nominal models $A(p_i), B(p_i), C(p_i), D(p_i)$. The switching is represented in (2) by the switching signal $\sigma(t) : \mathbb{R}^+ \to \mathcal{I}$.

Various strategies for logic-based switching in MMAC have been proposed since its establishment. We consider here the one proposed in [4], due to its simplicity. Note that due to the modularity of the MMAC framework, the model distribution presented in this paper can and should also be combined with other switching strategies. In this context, we also want to stress that issues related to stability of the switching logic, the number of switches and chattering effects strongly depend on the applied switching logic and will therefore not be considered in this paper.

Fig. 1 shows the supervisor proposed by Hespanha *et al.* [4], which contains the three components multiestimator, monitoring signal generator and hysteresis switching logic to orchestrate the switching in the multicontroller (2). First, a *multiestimator* $\mathbb{E}$ generates estimated system outputs $\hat{y}_i$ for each nominal model $\mathcal{G}(p_i), i \in \mathcal{I}$ based on the measured control inputs u and outputs y of the uncertain system. With the estimated system outputs $\hat{y}_i$, estimation errors

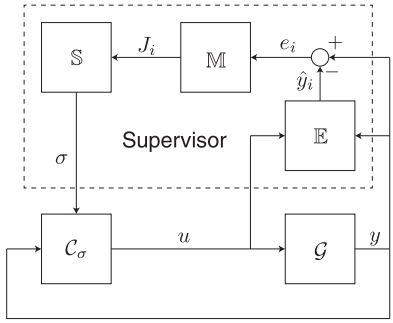


Fig. 1. Supervisor in MMAC.

$$e_i = \hat{y}_i - y \tag{3}$$

are formulated. Inside the *monitoring signal generator* $\mathbb{M}$, these estimation errors are translated into performance signals $J_i > 0$ which express the fitness of the nominal models. A popular choice is to consider a weighted integral over the estimation errors, such as

$$J_i(t) = \alpha e_i^\top(t) e_i(t) + \beta \int_0^t \mathrm{e}^{-\lambda(t-\tau)} e_i^\top(\tau) e_i(\tau) d\tau \tag{4}$$

with design constants $\alpha, \beta, \lambda \geq 0$. The third component, called *switching logic* $\mathbb{S}$, generates the switching signal $\sigma(t)$ based on the performance signals. The simplest switching logic is to select the nominal controller with the best performance, i.e., smallest error signal. With noisy measurements, however, this causes too frequent switching, which motivated the hysteresis switching logic in [4]. Suppose that the last switch occurred at time t_j and the switching signal is now $\sigma(t) = k \in \mathcal{I}$. Under hysteresis switching, the value of σ remains fixed until a time instance $t_{j+1} > t_j + \epsilon$ is reached, at which

$$(1+h) \min_{i \in \mathcal{I}} J_i(t_{j+1}) \leq J_k(t_{j+1}) \tag{5}$$

with hysteresis constant $h > 0$. The dwell ϵ is then bounded below by a finite value. At t_{j+1}, the switching signal takes the value $\sigma(t) = \arg\min_{i \in \mathcal{I}} J_i(t_{j+1})$ until the hysteresis switching condition (5) is satisfied the next time.

The stability proof for the interconnection of the uncertain system (1) with the state-shared multicontroller (2) under hysteresis switching (5) is by now well understood. It must first be shown that for a constant parameter vector q, the supervisor ceases switching in finite time with a finite number of switches. Then, the controller on which the supervisor settles must be confirmed to stabilize the linear time invariant system $\mathcal{G}(q)$. For continuous uncertainty sets, convergence to a single most appropriate controller is not guaranteed under the above hysteresis switching logic. Hence, additional measures need to be taken on the side of the supervisor such as hierarchical hysteresis switching [9] or continuous mixing of nominal controllers [14]. As

details on the supervisor are not in the scope of this paper, we refer the interested reader to [4], [6], [7], [9] for more details.

III. Robust Controller Design

In order to implement MMAC, one needs to ensure that there exists a stabilizing controller for each plant in the uncertainty set. Such guarantees can be given with results from the robust control literature [23], [24], [33]–[36], to be presented in this section. First, we revise how to express distances between linear time invariant systems in terms of the ν-gap metric and how these distances relate to the stability margin of nominal model-controller interconnections. Afterward, the calculation of the best possible stability margin for a given system is described. Finally, the synthesis of a robust controller with desired stability margin is detailed.

A. Characterizing Stabilizable Systems With the ν-Gap Metric

The ν-gap metric was introduced to the field of robust control by Vinnicombe [23], [24] in the 1990s, providing a measure of the maximum distance between a nominal and a perturbed system under which the controller, which was designed for the nominal system, still stabilizes the perturbed system. For the supervisory control setting, it is necessary to determine those systems

$$\mathcal{G}(s) = C\left(sI - A\right)^{-1} B + D \tag{6}$$

from the uncertainty set, which are sufficiently close to the ith nominal model

$$\mathcal{G}_i(s) = C_i\left(sI - A_i\right)^{-1} B_i + D_i \tag{7}$$

and that they are stabilized by the corresponding controller $\mathcal{C}_i$. Here, I is the identity matrix of matching dimension.

The *ν-gap* between $\mathcal{G}_i$ and $\mathcal{G}$ is defined as

$$\delta_\nu(\mathcal{G}_i, \mathcal{G}) := \left\| (I + \mathcal{G}\mathcal{G}^*)^{-\frac{1}{2}} (\mathcal{G} - \mathcal{G}_i)(I + \mathcal{G}_i^*\mathcal{G}_i)^{-\frac{1}{2}} \right\|_\infty \tag{8}$$

given that the following winding number conditions[1]:

$$\begin{aligned} &\det(I + \mathcal{G}^*\mathcal{G}_i)(j\omega) \neq 0 \quad \forall\omega \quad \text{and} \\ &\text{wno}\,\det(I + \mathcal{G}^*\mathcal{G}_i) + \kappa(\mathcal{G}_i) - \kappa(\mathcal{G}) = 0 \end{aligned} \tag{9}$$

are satisfied [23], where $\mathcal{G}^*$ denotes the complex conjugate of $\mathcal{G}$. If the conditions are not satisfied, then let $\delta_\nu(\mathcal{G}_i, \mathcal{G}) = 1$. With the definition above, the value of δ_ν is between 0 and 1. Systems that are close to each other in the ν-gap metric have a value close to 0, whereas systems that are very different have a maximum distance of 1. Efficient implementations of the ν-gap metric are available, such as the function `gapmetric` in the Robust Control Toolbox of MATLAB.

The most important property of the ν-gap metric is related to the *generalized stability margin* $b_{\mathcal{G}_i,\mathcal{C}_i}$ of a nominal model $\mathcal{G}_i$ with the nominal controller $\mathcal{C}_i$ in the feedback loop. If the feedback interconnection $(\mathcal{G}_i, \mathcal{C}_i)$ is stable, the generalized stability margin is defined by

$$b_{\mathcal{G}_i,\mathcal{C}_i} := \left\| \begin{bmatrix} \mathcal{G}_i \\ I \end{bmatrix} (I - \mathcal{C}_i\mathcal{G}_i)^{-1} \begin{bmatrix} -\mathcal{C}_i & I \end{bmatrix} \right\|_\infty^{-1} \tag{10}$$

otherwise $b_{\mathcal{G}_i,\mathcal{C}_i} = 0$. The following statement reveals the importance of the ν-gap metric [23]: Given a nominal model $\mathcal{G}_i$, a nominal controller $\mathcal{C}_i$, and a number β, then $(\mathcal{G}, \mathcal{C}_i)$ is stable for all systems $\mathcal{G}$, satisfying $\delta_\nu(\mathcal{G}_i, \mathcal{G}) \leq \beta$ if, and only if, $b_{\mathcal{G}_i,\mathcal{C}_i} > \beta$. That means the nominal controller $\mathcal{C}_i$, which in feedback with the nominal model $\mathcal{G}_i$ achieves a generalized stability margin greater than β, also stabilizes all systems $\mathcal{G}$ located in an β-radius (in the ν-gap metric) around the nominal model $\mathcal{G}_i$.

B. Determining the Maximum Radius of Stabilizable Systems

Now, given a nominal model $\mathcal{G}_i$, two useful questions may be asked: What is the most robust controller in the ν-gap metric for this system? And, what is the maximum generalized stability margin that can be achieved? To answer these questions, let $\mathcal{G}_i$ be written in terms of the normalized left coprime factorization (LCF)

$$\mathcal{G}_i = M_i^{-1} N_i \tag{11}$$

where normalization means

$$N_i N_i^* + M_i M_i^* = I \tag{12}$$

which is equivalent to the matrix $[N_i, M_i]$ being coinner. (See [33, Lemma 2.1] for how to calculate a normalized LCF).

The most robust controller is found as the solution to the following $\mathcal{H}_\infty$-optimization problem [33]:

$$\inf_{\mathcal{C}_i} \left\| \begin{bmatrix} \mathcal{C}_i \\ I \end{bmatrix} (I - \mathcal{G}_i\mathcal{C}_i)^{-1} M_i^{-1} \right\|_\infty = \left(1 - \left\| \begin{bmatrix} N_i & M_i \end{bmatrix} \right\|_H^2\right)^{-\frac{1}{2}} \tag{13}$$

where $\|\cdot\|_H$ is the Hankel norm. Denote the obtained infimum as $\gamma_{\mathcal{G}_i,\min}$ and note that it also yields an upper bound for the achievable generalized stability margin of the system $\mathcal{G}_i$. In [33] and [34], it is shown that

$$\begin{aligned} &\left\| \begin{bmatrix} \mathcal{C}_i \\ I \end{bmatrix} (I - \mathcal{G}_i\mathcal{C}_i)^{-1} M_i^{-1} \right\|_\infty \\ &\qquad = \left\| \begin{bmatrix} \mathcal{G}_i \\ I \end{bmatrix} (I - \mathcal{C}_i\mathcal{G}_i)^{-1} \begin{bmatrix} -\mathcal{C}_i & I \end{bmatrix} \right\|_\infty \end{aligned} \tag{14}$$

which is due to the fact that $[M_i, N_i]$ is coinner and that the $\mathcal{H}_\infty$ norm is invariant under right multiplication by coinner functions [36, Ex. 9.7]. Hence, the maximum generalized stability margin for a system $\mathcal{G}_i$—as considered in the ν-gap metric—is

$$b_{\mathcal{G}_i,\max} = \gamma_{\mathcal{G}_i,\min}^{-1} = \left(1 - \left\| \begin{bmatrix} N_i & M_i \end{bmatrix} \right\|_H^2\right)^{\frac{1}{2}} > 0. \tag{15}$$

C. Designing Controllers With Specific Stability Margin

Note that the application of the most robust controller is often undesirable as controller synthesis is a tradeoff between robustness and performance. Hence, the most robust controller may

[1] Here, wno is the winding number evaluated on the standard Nyquist contour indented at the imaginary axis poles of $\mathcal{G}$ and $\mathcal{G}_i$. The number of poles of a transfer function $\mathcal{G}$ in the open right half-plane are denoted by $\kappa(\mathcal{G})$.

not deliver the desired performance. Assume therefore that a desired stability margin $b_{\mathcal{G}_i,\text{des}} < b_{\mathcal{G}_i,\max}$ is chosen by the designer and a suitable controller satisfying this stability margin is to be found. By reducing the desired stability margin, the resulting controller can be configured for improved performance, but guarantees closed-loop stability only for smaller perturbations of the nominal system.

In [33], suitable controllers are synthesized by solving a Nehari extension problem. Here, we follow the approach presented in [36, Sec. 9.4]. Given a minimal state-space representation $[A_i, B_i, C_i, D_i]$ for $\mathcal{G}_i$ and a desired threshold $\gamma_{\mathcal{G}_i,\text{des}} > \gamma_{\mathcal{G}_i,\min}$, then let

$$R_i := I + D_i D_i^\top, \qquad S_i := I + D_i^\top D_i \tag{16}$$

and solve the two algebraic Riccati equations

$$\begin{aligned}\left(A_i - B_i S_i^{-1} D_i^\top C_i\right)^\top X_i + X_i \left(A_i - B_i S_i^{-1} D_i^\top C_i\right) \\ - X_i B_i S_i^{-1} B_i^\top X_i + C_i^\top R_i^{-1} C_i = 0\end{aligned} \tag{17}$$

and

$$\begin{aligned}\left(A_i - B_i S_i^{-1} D_i^\top C_i\right) Z_i + Z_i \left(A_i - B_i S_i^{-1} D_i^\top C_i\right)^\top \\ - Z_i C_i^\top R_i^{-1} C_i Z_i + B_i S_i^{-1} B_i^\top = 0\end{aligned} \tag{18}$$

for the unique matrices X_i and Z_i. With

$$F_i := -S_i^{-1}\left(D_i^\top C_i + B_i^\top X_i\right) \tag{19}$$

$$L_i := (1 - \gamma_{\mathcal{G}_i,\text{des}}^2) I + X_i Z_i \tag{20}$$

the state-space realization of a controller $\mathcal{C}_i$ satisfying $b_{G_i,\text{des}} = \gamma_{\mathcal{G}_i,\text{des}}^{-1}$ is given by

$$\begin{aligned} A_{\mathcal{C}_i} &:= A_i + B_i F_i + \gamma_{\mathcal{G}_i,\text{des}}^2 (L_i^\top)^{-1} Z_i C_i^\top (C_i + D_i F_i) \\ B_{\mathcal{C}_i} &:= \gamma_{\mathcal{G}_i,\text{des}}^2 (L_i^\top)^{-1} Z_i C_i^\top \\ C_{\mathcal{C}_i} &:= B_i^\top X_i \\ D_{\mathcal{C}_i} &:= -D_i^\top. \end{aligned} \tag{21}$$

Finally, note that the synthesis of robust controllers with specific stability margin can be combined with performance criteria in the form of loop shaping. An introduction on how to include loop shaping ideas into the $\mathcal{H}_\infty$-controller synthesis above is given in [35] and [36, Sec. 9.4.2].

To conclude the revision of tools from the robust control literature, let us repeat the three essential findings which will enable optimized model distribution in MMAC. First, given a nominal model $\mathcal{G}_i$, one can determine its maximal stability margin $b_{\mathcal{G}_i,\max}$ with (15). Second, a nominal controller $\mathcal{C}_i$ for $\mathcal{G}_i$ satisfying a specific stability margin $b_{\mathcal{G}_i,\text{des}} < b_{\mathcal{G}_i,\max}$ can be synthesized according to (21). Third, the fact that $\mathcal{C}_i$ stabilizes all systems $\mathcal{G}$ for which $\delta_\nu(\mathcal{G}_i, \mathcal{G}) < b_{\mathcal{G}_i,\mathcal{C}_i}$ paves the way to partitioning the uncertainty set into various subsets, each associated with one nominal model-controller pair. Previous approaches to MMAC such as [3], [13], [25], [26] incorporate the same idea to investigate—based on the ν-gap metric—whether the chosen set of nominal controllers can stabilize all possible system configurations. So far the focus was, however, only on providing stability guarantees. In the next section, we present a new algorithm, which not only ensures the existence of a stabilizing controller for each system configuration but also provides an optimized partitioning of the uncertainty set in a sense that less nominal systems are needed.

IV. Systematic Model Distribution Through Coverage Control

In this section, the optimal distribution of nominal models is phrased as a coverage control problem. The coverage control problem is extensively studied in the field of multiagent systems in the robotics community, where a set of robots/agents autonomously covers an area of interest for monitoring purposes [37]–[40]. Most algorithms build upon the classic work by Lloyd [41] on optimal quantizer selection, which was first applied to the distributed robotic coverage problem in [37]. Initial algorithms were concerned with convex environments and the Euclidean distance. More recent works focus on coverage control algorithms for nonconvex domains or manifolds and arbitrary distances [38]–[40] as well as heterogeneous agents [42] (e.g., robots with different sensing capabilities). For MMAC, we transfer the coverage-control framework from the physical space, in which robotic agents operate, to an abstract space in form of the model uncertainty, in which the nominal models need to be placed. In this analogy, we replace the target environment of the robotic agents by the uncertainty set. The abstract counterpart to the robotic agents are the nominal model-controller pairs, which will therefore also be referred to as agents. Distances in the MMAC setting are measured in terms of the ν-gap metric instead of the Euclidean distance.

This section begins with the formulation of the coverage control optimization problem in Section IV-A, followed by a continuous gradient descent approach in Section IV-B. For computational reasons, a discretized version of the coverage control problem is introduced in Section IV-C. Then, a first algorithm performing a gradient descent on the discrete graph is derived in Section IV-D. Afterward, a second algorithm, which is based on pairwise optimizations, is presented in Section IV-E. Finally, a brief summary in Section IV-F lists potential sources for suboptimality.

A. Continuous Coverage Control Problem

Consider the *uncertainty set* $\mathcal{Q} \subset \mathbb{R}^n$ in which the uncertain parameters q of the controlled system reside. Hence, each $q \in \mathcal{Q}$ represents a potential configuration of the system. We assume that the uncertainty set is characterized by a convex polytope.[2] In other words, there exist known bounds on the entries of the parameter vector q, e.g., $q_{(1),\min} < q_{(1)} < q_{(1),\max}$. Next, let the *distribution density function* $\phi : \mathcal{Q} \to \mathbb{R}_{\geq 0}$ introduce a notion of relevance/importance of each configuration in the set $\mathcal{Q}$. This importance can be seen as some prior on the probabilities of certain parameter configurations. Thus, more likely or more

[2]The assumption of convex uncertainty sets is made for simplicity. In case $\mathcal{Q}$ is nonconvex or even unconnected, one may consider a convex hull $\bar{\mathcal{Q}}$ of the original uncertainty set and assign a distribution density function of zero to all configurations which are not from the original set, i.e., $\phi(q) = 0 \quad \forall q \notin \mathcal{Q}$.

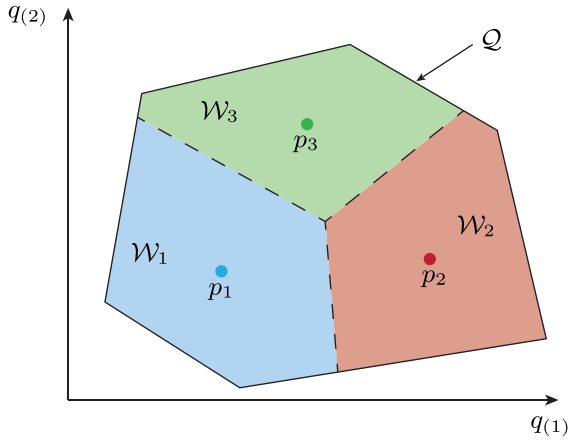


Fig. 2. Visualization of the coverage problem.

important parameter configurations should be assigned a larger value in ϕ. If all configurations are equally important, let $\phi(q) = 1\,\forall q \in \mathcal{Q}$.

The goal in MMAC is to stabilize all possible system configurations from the uncertainty set $\mathcal{Q}$ by a total of $N \in \mathbb{N}$ nominal systems and their associated controllers. We refer to these systems as *nominal system configurations* $p_i \in \mathcal{Q}$ and let all chosen configurations be combined in the set $\mathcal{P} = \{p_1, \dots, p_N\}$. It is desirable to distribute the models in such a way that for each potential system configuration $q \in \mathcal{Q}$ there is at least one model $p_i \in \mathcal{P}$ whose controller stabilizes the system q. Let this relationship be expressed by a *partitioning* $\mathcal{W} = \{\mathcal{W}_1, \dots, \mathcal{W}_N\}$ of the uncertainty set $\mathcal{Q}$. The partitions are such that their union corresponds to the complete uncertainty set $\cup_{i=1}^{N} \mathcal{W}_i = \mathcal{Q}$, such that there is no overlap $\mathcal{W}_i \cap \mathcal{W}_j = \emptyset, i \neq j$, and such that each partition is nonempty $\mathcal{W}_i \neq \emptyset$ and connected $\forall i, j \in \mathcal{I} := \{1, \dots, N\}$. Hence, each system configuration $q \in \mathcal{Q}$ belongs to just one partition $\mathcal{W}_i$ and its nominal model configuration p_i. The controller associated with the model p_i shall be responsible to stabilize all potential systems $q \in \mathcal{W}_i$. Fig. 2 visualizes an example of how the uncertainty set $\mathcal{Q}$ of a two-dimensional parameter vector is covered by three models and their partitions.

Next, introduce a *metric* $d(q, p_i) : \mathcal{Q} \times \mathcal{Q} \to \mathbb{R}_{\geq 0}$ which defines the distance between a configuration q and the nominal model configuration p_i. As pointed out above, the ν-gap metric in (8) is ideally suited for such distance measures for control design applications. Hence, let $d(q, p_i) := \delta_\nu(\mathcal{G}(q), \mathcal{G}(p_i))$. It is furthermore well known that the $\mathcal{H}_\infty$ performance of the closed-loop system, whose controller was designed for the nominal configuration p_i, degrades with growing distance from the nominal configuration. Let this relationship be specified by a function $f(d) : \mathbb{R} \to \mathbb{R}_{\geq 0}$, which we refer to as the *inverse control performance*. Let $f(d(q, p_i))$ be strictly increasing with the distance $d(q, p_i)$. Hence, a small value in $f(d(q, p_i))$ indicates that the controller designed for p_i performs well for q. A large value on the other side indicates that the system q cannot be robustly stabilized by the nominal controller for p_i. The increase in f with decreasing performance motivates the name inverse control performance. A common choice for f in coverage control with homogeneous agents/robots is $f(d) = d^2$.

In the present setting of coverage control for model distribution in MMAC, however, we observe that the nominal models (i.e., agents) may be very heterogeneous due to varying stability margins. Take for instance two nominal models p_1 and p_2 with different stability margins, e.g., $b_{\mathcal{G}_1,\max} < b_{\mathcal{G}_2,\max}$. In this case, a perturbed system $\mathcal{G}(q)$ may be stabilized only by the nominal controller associated with p_2 even if $\delta_\nu(\mathcal{G}(q), \mathcal{G}(p_1)) < \delta_\nu(\mathcal{G}(q), \mathcal{G}(p_2))$, given that $\delta_\nu(\mathcal{G}(q), \mathcal{G}(p_2)) < b_{\mathcal{G}_2,\max}$ and $\delta_\nu(\mathcal{G}(q), \mathcal{G}(p_1)) > b_{\mathcal{G}_1,\max}$. In order to accommodate this heterogeneity, we apply ideas presented in [42] and introduce the *stability-distance*

$$d_S(q, p_i) := \delta_\nu(\mathcal{G}(q), \mathcal{G}(p_i)) - b_{\mathcal{G}_i,\max} \tag{22}$$

which essentially incorporates the maximum stability radius into the distance measure. Compared to the distance $d(q, p_i)$, the stability-distance $d_S(q, p_i)$ contains more information in a sense that systems q with $d_S(q, p_i) < 0$ can, and systems with $d_S(q, p_i) > 0$ cannot, be stabilized by the (most robust) nominal controller $\mathcal{C}_i$ with stability margin $b_{\mathcal{G}_i,\max}$. In order to reflect the instability for $d_S > 0$, the inverse control performance is defined to be

$$f(d_S) = e^{\Gamma d_S} \tag{23}$$

which grows exponentially with d_S and the scalar design parameter $\Gamma > 0$.

With the above definitions, introduce the following cost function, which rates the expected performance of the distribution $\mathcal{P}$ and the partitioning $\mathcal{W}$

$$\mathcal{H}(\mathcal{P}, \mathcal{W}) := \sum_{i=1}^{N} \int_{\mathcal{W}_i} f(d_S(q, p_i))\phi(q)\mathrm{d}q. \tag{24}$$

This cost function is also referred to as *coverage function* as it indicates how well the set of model-controller configurations $\mathcal{P}$ covers the considered parameter uncertainties $\mathcal{Q}$ partitioned by $\mathcal{W}$. The lower the cost $\mathcal{H}$, the better are the models distributed and the better is the expected control performance. Hence, the goal of optimally distributing the nominal models p_i is formulated as

$$\underset{\mathcal{P},\mathcal{W}}{\text{minimize}}\, \mathcal{H}(\mathcal{P}, \mathcal{W}). \tag{25}$$

It is well known [42], [43] that the so-called *Voronoi diagrams* $\mathcal{V} = \{\mathcal{V}_1, \dots, \mathcal{V}_N\}$ with

$$\mathcal{V}_i(\mathcal{P}) := \{q \in \mathcal{Q} : d_S(q, p_i) \leq d_S(q, p_j)\,\forall j \neq i\} \tag{26}$$

constitute the optimal choice for $\mathcal{W}_i$ given the model configuration $\mathcal{P}$. A Voronoi diagram is a partitioning of $\mathcal{Q}$ in which the elements $q \in \mathcal{V}_i$ have the closest stability-distance to the nominal model p_i. As $\mathcal{V}$ is a function of all nominal models in $\mathcal{P}$, it suffices to optimize the parameter configuration $\mathcal{P}$ and apply the optimal Voronoi diagrams, i.e., $\mathcal{H}_\mathcal{V}(\mathcal{P}) = \mathcal{H}(\mathcal{P}, \mathcal{V}(\mathcal{P}))$, which simplifies (25) to

$$\underset{\mathcal{P}}{\text{minimize}}\, \mathcal{H}_\mathcal{V}(\mathcal{P}). \tag{27}$$

Above, it was assumed that the partitions are connected. In order to ensure this property for the Voronoi diagrams $\mathcal{V}_i$, an additional assumption needs to be imposed on the system. That

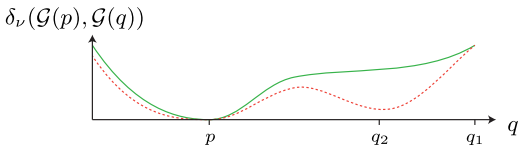


Fig. 3. While the first system (green solid line) satisfies the assumption of the monotonically increasing ν-gap along the shortest path between p and q_1, the second system (red dashed line) does not, which may result in additional nominal models.

is, the system $\mathcal{G}$ and its uncertainty set $\mathcal{Q}$ are such that the distance in the ν-gap metric $\delta_\nu(\mathcal{G}(p),\mathcal{G}(q_2))$ is monotonically increasing for all points q_2 along the shortest path in $\mathcal{Q}$ between p and q_1. Fig. 3 visualizes this property for the scalar case. This assumption is hard to validate in practice due to the possibly complicated relationship between a parameter q and the ν-gap. Note however that violating this assumption only leads to a minor disadvantage in the form of an increased number of nominal systems.

In practice, the number N of nominal model-controller pairs in the optimization problem (27) is usually unknown and must also be determined. For this purpose, note that finding the optimum in (27) alone does not ensure that each configuration q is stabilized by at least one nominal controller p_i. Hence, the additional inequality constraints

$$\min_{i\in\mathcal{I}} d_S(q,p_i) < 0 \quad \forall q \in \mathcal{Q} \tag{28}$$

need to be imposed and checked after optimization. Hence, the minimum number of required model-controller pairs may either be determined by increasing N from $N=1$ until the optimum in (27) satisfies the constraints (28). Otherwise, if a conservative number $\bar{N}$ is known, one can start from $N=\bar{N}$ and decrease N until (27) cannot be solved without violating (28).

An important assumption regarding the partitions is that the switching logic selects for a system $q \in \mathcal{V}_i$ the nominal controller $p_i\ \forall i \in \mathcal{I}$. If this is not the case, the obtained performance is at best suboptimal, i.e., a model with greater distance in the ν-gap metric and therefore worse control performance would be chosen. If p_i was the only nominal controller to stabilize q, this may even cause instability. This assumption is to be ensured by the switching logic, which is not discussed here. Our primary concern is to obtain a finite covering for which there exists at least one stabilizing controller for each system in the uncertainty set.

B. Gradient Descent

As shown in the coverage control literature [37], [39], [40], the coverage problem (27) can be solved by a distributed gradient descent, in which each agent (here each nominal model) only needs to know the location of the agent in its neighboring partitions. Let $\mathcal{P}$ be initialized with some initial models $p_i(0)$ and let the uncertainty set be partitioned by the Voronoi diagrams $\mathcal{V}_i$. Then, a gradient-descent approach for (27) moves each model p_i in the direction of the steepest descent given by $-\frac{\partial\mathcal{H}}{\partial p_i}$. This motivates updating the nominal model parameters in $\mathcal{P}$ by

$$\dot{p}_i = -\rho\frac{\partial\mathcal{H}}{\partial p_i}, \quad i\in\mathcal{I}=\{1,\dots,N\} \tag{29}$$

with scaling constant $\rho>0$. Note that the domains of integration in (24) depend on $\mathcal{V}_i$, which in turn depends on the configurations in $\mathcal{P}$. Hence, the update $\dot{p}_i$ would intuitively depend not only on p_i and $\mathcal{V}_i$ but also on neighboring partitions. However, with differentiation under integration it was shown in [42] that the gradient with respect to the i th model reduces to

$$\frac{\partial\mathcal{H}}{\partial p_i} = \int_{\mathcal{V}_i} \frac{\partial f(d_S(q,p_i))}{\partial p_i}\phi(q)\mathrm{d}q \tag{30}$$

$$= \int_{\mathcal{V}_i} \frac{\partial f(d_S(q,p_i))}{\partial d_S(q,p_i)}\left(\frac{\partial d(q,p_i)}{\partial p_i} - \frac{\partial b_{\mathcal{G}_i,\max}}{\partial p_i}\right)\phi(q)\mathrm{d}q. \tag{31}$$

Note that the integral in (31) requires evaluating the gradient of the distance metric, i.e.,

$$\frac{\partial d(q,p_i)}{\partial p_i} \tag{32}$$

for all $q\in\mathcal{V}_i$, which is computationally expensive. In order to reduce the computational load, it can be exploited that the gradient (32) is the same for all other points η on the shortest path—in terms of $d(q,p_i)$—between q and p_i [39]. That means, moving p_i toward q along the shortest path causes equal reduction in $d(q,p_i)$ and $d(\eta,p_i)$. Hence, for each point q, consider another point $\eta(q)\in\mathcal{Q}$ in the vicinity of p_i which lies on the shortest path between p_i and q, and exploit that

$$\frac{\partial d(q,p_i)}{\partial p_i} = \frac{\partial d(\eta(q),p_i)}{\partial p_i}. \tag{33}$$

In that case, the evaluation of the gradient at a lot of points q is reduced to the evaluation of the gradient at fewer points $\eta(q)$.

The gradient (31) is then given by

$$\frac{\partial\mathcal{H}}{\partial p_i} = \int_{\mathcal{V}_i} \frac{\partial f(d_S)}{\partial d_S}\left(\frac{\partial d(\eta(q),p_i)}{\partial p_i} - \frac{\partial b_{\mathcal{G}_i,\max}}{\partial p_i}\right)\phi(q)\mathrm{d}q. \tag{34}$$

Using the gradient descent (34) with Voronoi diagrams (26) and the ν-gap metric yields a modified application of the continuous-time Lloyd algorithm [41] for optimal model distribution in MMAC. The derived algorithm updates an initial model configuration $\mathcal{P}$ and results in an optimized coverage $\mathcal{H}(\mathcal{P},\mathcal{V})$. The obtained optimum is a so-called *Centroidal Configuration*, i.e., a configuration $(\mathcal{P},\mathcal{V})$ in which $\mathcal{P}$ gives rise to the Voronoi diagrams $\mathcal{V}$ and at the same time each $p_i\in\mathcal{P}$ corresponds to the optimal location in $\mathcal{V}_i$. Note that centroidal configurations are not unique and depend on the initial configuration of $\mathcal{P}$. Therefore, the obtained optimum is only local and the algorithm may have to be initialized from various initial configurations in order to find the global optimum. For more details on centroidal configurations, refer to [44].

Unfortunately, the application of the modified Lloyd algorithm remains computationally expensive due to the numerous evaluations of the ν-gap metric in form of d_S under the integral in (34). Therefore, in the next step, the uncertainty set is quantized, which enables a more efficient graph-based realization

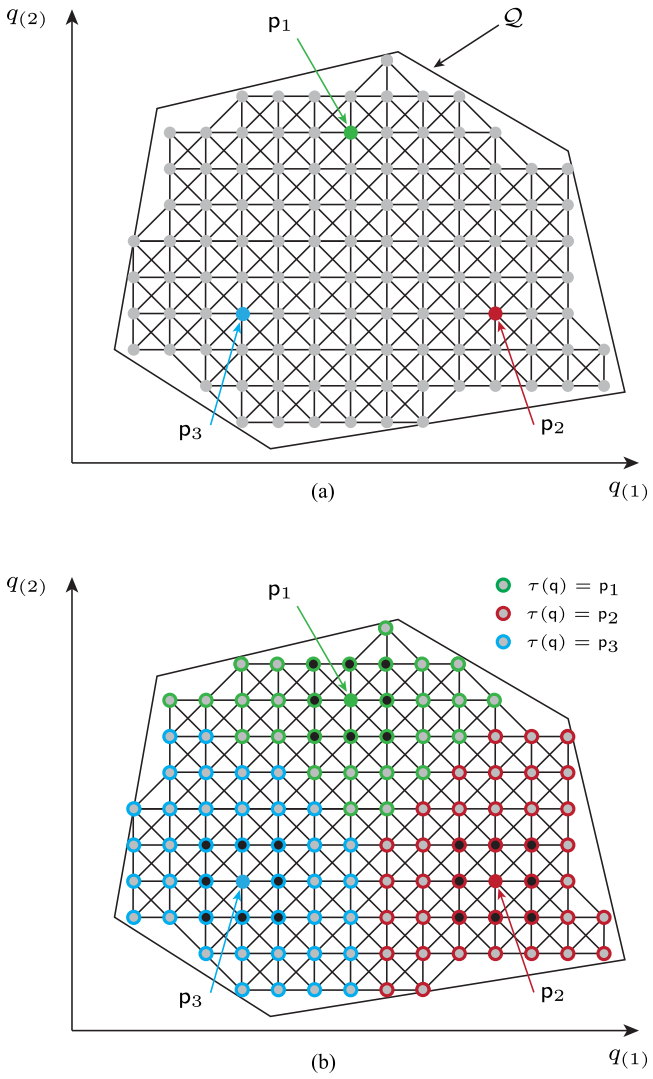


Fig. 4. Visualization of the discrete Voronoi partitioning obtained by Algorithm 1. The selected nominal vertices p_i in (a) are extended and yield the partitioning shown in (b). (a) Initial setting. (b) Discrete Voronoi partitioning.

of the coverage problem. This step is in line with algorithms proposed in [38] and [39], where the consideration of the path-length metric in nonconvex environments requires quantization.

C. Coverage Control as Graph Search

Let the uncertainty set $\mathcal{Q}$ be approximated by a finite set of models q_k, $k = 1, \ldots, s$ as shown in Fig. 4(a), where the gray circles correspond to models q_k. These models can be understood as a quantization of the original uncertainty set. Next, define an *undirected weighted graph* $G(Q, E, d)$, whose *vertex set* $Q(G)$ is the set of quantized models, i.e., $Q(G) := \{\mathsf{q}_1, \ldots, \mathsf{q}_s\}$. While q_k refers to a vertex in the graph, let $q(\mathsf{q}_k) \in \mathcal{Q}$ refer to the continuous parameter associated with the vertex q_k. Let neighboring vertices be connected by a weighted edge in the graph [see Fig. 4(a)]. Hence, the *edge set* of the graph is $E(G) \subseteq Q(G) \times Q(G)$. Define the *neighborhood* of q_k as $\mathcal{N}(\mathsf{q}_k) = \{\mathsf{q}_l \in Q(G) : (\mathsf{q}_k, \mathsf{q}_l) \in E(G)\}$. The considered metric d serves as the *weight map* on the edges $d : E(G) \to \mathbb{R}_+$. The edge weights thus introduce the metric distance between adjacent vertices. In other words, if two nodes $\mathsf{q}_k, \mathsf{q}_l \in Q(G)$ are connected by an edge, i.e., $(\mathsf{q}_k, \mathsf{q}_l) \in E(G)$, then the connecting edge takes the weight $d(q(\mathsf{q}_k), q(\mathsf{q}_l))$. Introducing the graph greatly reduces the computational complexity in a sense that the ν-gap metric only needs to be computed once for each edge during initialization as compared to various combinations $d_S(q, p_i)$ under the integral in (34) at each iteration of the gradient descent.

The distance between two arbitrary vertices of G is defined in terms of paths. A *path* in G is a sequence of vertices, in which every pair of consecutive vertices belongs to the edge set $E(G)$. The sum of the edge weights along a path defines the *weight of the path*. The distance between two vertices q_k and q_l in G is given by the weight of the path with the lowest weight and we denote it by $\mathsf{d}_G(\mathsf{q}_k, \mathsf{q}_l)$. As G is undirected, it follows that $\mathsf{d}_G(\mathsf{q}_k, \mathsf{q}_l) = \mathsf{d}_G(\mathsf{q}_l, \mathsf{q}_k)$.

The graph-based representation above restricts the metric d from the continuous uncertainty set $\mathcal{Q}$ to the discrete vertex set $Q(G)$. Doing so is motivated by the computational complexity in the continuous space. The restriction allows for a more efficient graph-search on G. It must be noted, however, that this quantization essentially yields an approximation of the original problem. First, quantization restricts the choice of nominal models to vertices of the graph which may not contain the optimal configurations. Second, determining distances on the graph by summing up the edge weights of the shortest path yields an approximation. Due to the triangle inequality, it follows that the distance on the graph is always greater or equal to the distance between the same points in $\mathcal{Q}$, i.e., $\mathsf{d}_G(\mathsf{q}_k, \mathsf{p}_i) \geq d(q(\mathsf{q}_k), q(\mathsf{p}_i))$. While the error associated with the graph-based approximation can to some extend be regulated by the density of quantized models q_k, the quantization nonetheless remains a tradeoff: Accuracy and elegance of the continuous representation against computational efficiency of the discrete graph.

For the coverage problem, select N *nominal vertices* $\mathsf{p}_i, i = 1, \ldots, N$ and refer to $\mathsf{P} = \{\mathsf{p}_1, \ldots, \mathsf{p}_N\}$ as the *set of nominal vertices*. Furthermore, let G be partitioned into N subgraphs $\mathsf{W}_i \subset G$ with the same intention as in the continuous uncertainty set. That is, the nominal models $p_i = q(\mathsf{p}_i)$ should be distributed in such a way that their controllers stabilize all systems associated with the vertices in the *partition* W_i. As in partitioning of the continuous uncertainty $\mathcal{Q}$, the partitions W_i are such that their union corresponds to the complete graph $\cup_{i=1}^{N} \mathsf{W}_i = G$, such that there is no overlap $\mathsf{W}_i \cap \mathsf{W}_j = \emptyset, i \neq j$ and such that each partition is nonempty $\mathsf{W}_i \neq \emptyset$ and connected.

In order to introduce the discrete counterpart to the Voronoi diagrams, let the stability-distance on the graph G be

$$\mathsf{d}_{G,S}(\mathsf{q}, \mathsf{p}_i) := \mathsf{d}_G(\mathsf{q}, \mathsf{p}_i) - b_{\mathcal{G}_i, \max} \tag{35}$$

where $b_{\mathcal{G}_i, \max}$ is the maximal stability margin for the nominal model associated with $q(\mathsf{p}_i)$. Then, a Voronoi diagram for G is given by the subgraphs

$$\mathsf{V}_i(\mathsf{P}) = \left\{\mathsf{q}_k \in Q : \mathsf{d}_{G,S}(\mathsf{q}_k, \mathsf{p}_i) \leq \mathsf{d}_{G,S}(\mathsf{q}_k, \mathsf{p}_j)\, \forall j \neq i\right\}. \tag{36}$$

For one of the proposed algorithms, a notion of *adjacent partitions* is needed. Two partitions W_i and W_j with $i \neq j$ are adjacent if there exist two vertices $\mathsf{q}_k \in \mathsf{W}_i$ and $\mathsf{q}_l \in \mathsf{W}_j$ that are connected by an edge in the graph $(\mathsf{q}_k, \mathsf{q}_l) \in E(G)$.

Next, we rewrite the coverage function (24) in terms of G. Replacing the integration over $\mathcal{W}_i$ by summation over W_i yields the *discrete coverage function*

$$\mathsf{H}(\mathsf{P}, \mathsf{W}) = \sum_{i=1}^{N} \sum_{\mathsf{q}_k \in \mathsf{W}_i} f\left(\mathsf{d}_{\mathsf{W}_i,S}(\mathsf{q}_k, \mathsf{p}_i)\right) \phi(\mathsf{q}_k). \tag{37}$$

For later analysis, we refer to

$$\check{\mathsf{H}}(\mathsf{p}_i, \mathsf{W}_i) = \sum_{\mathsf{q}_k \in \mathsf{W}_i} f\left(\mathsf{d}_{\mathsf{W}_i,S}(\mathsf{q}_k, \mathsf{p}_i)\right) \phi(\mathsf{q}_k) \tag{38}$$

as the *coverage of a single partition*.

Finally, let us revise two notions of optimality and their relationship derived in [38]. First, a configuration (P, W) is called *Centroidal Configuration*, if W is the Voronoi diagram according to (36), i.e., $\mathsf{W}_i = \mathsf{V}_i(\mathsf{P})\ \forall i$, and if $\mathsf{p}_i = \arg\min_{\mathsf{r} \in \mathsf{V}_i} \check{\mathsf{H}}(\mathsf{r}, \mathsf{V}_i)\ \forall i$. Second, a configuration (P, W) is called *Pairwise-Optimal Configuration* if, for every pair of adjacent partitions W_i and W_j with nominal vertices $\mathsf{p}_i = \arg\min_{\mathsf{r} \in \mathsf{W}_i} \check{\mathsf{H}}(\mathsf{r}, \mathsf{W}_i)$ and $\mathsf{p}_j = \arg\min_{\mathsf{r} \in \mathsf{W}_j} \check{\mathsf{H}}(\mathsf{r}, \mathsf{W}_j)$, the value of the coverage function is lower than for any other two-partition of the union of W_i and W_j, i.e.,

$$\check{\mathsf{H}}(\mathsf{p}_i, \mathsf{W}_i) + \check{\mathsf{H}}(\mathsf{p}_j, \mathsf{W}_j) \leq \check{\mathsf{H}}(\mathsf{p}_a, \mathsf{W}_a) + \check{\mathsf{H}}(\mathsf{p}_b, \mathsf{W}_b) \tag{39}$$

with $\mathsf{p}_a, \mathsf{p}_b \in \mathsf{W}_i \cap \mathsf{W}_j$ and $\mathsf{W}_a, \mathsf{W}_b \subset \mathsf{W}_i \cap \mathsf{W}_j$ and $\mathsf{W}_a \cap \mathsf{W}_b = \emptyset$. An intuitive understanding for the Centroidal Configuration is that W is the optimal partitioning for P, and P is the optimal choice of nominal models for the partitioning W. Hence, it turns out that Centroidal Configurations constitute (local) minima of the discrete coverage function (37). The pairwise-optimality adds that for every union of adjacent partitions, one cannot find a better partition. It follows that every Pairwise-Optimal Configuration (P, W) is at the same time a Centroidal Configuration [38, Prop.2.8]. The opposite is not true.

With the preliminaries discussed so far, we are ready to present two algorithms inspired by [38] and [39] to solve the coverage problem (25) for near optimal model distribution in MMAC.

D. Discretized Llyod Algorithm

Building upon ideas presented in [39], we propose a modification of Dijkstra's algorithm [45], [46], which performs both the partitioning with Voronoi diagrams (26) and the gradient descent (34) on the discrete graph G. The pseudocode of the proposed approach is given by Algorithm 1 with explanations given in the following paragraphs.

First, the partitioning of $Q(G)$ in Algorithm 1 is straightforward and essentially requires three parameters per vertex: The value of $\tau(\mathsf{q}_k)$ determines the closest nominal model, and $g(\mathsf{q}_k)$ and $g_S(\mathsf{q}_k)$ determine the distance and stability-distance along the shortest path between q_k and the vertex of the closest nominal model. Initially, the algorithm assigns infinite distances

Algorithm 1: Voronoi diagram & Gradient Descent.

1: **for all** $\mathsf{q}_k \in Q(G)$ **do** ▷ Initialization
2: Set $\tau(\mathsf{q}_k), \eta(\mathsf{q}_k) := \emptyset$, and $g(\mathsf{q}_k), g_S(\mathsf{q}_k) := \infty$
3: **end for**
4: **for all** $\mathsf{p}_i \in \mathrm{P}$ **do** ▷ Init. vertices of nominal models
5: Set $g(\mathsf{p}_i) := 0$, and $g_S(\mathsf{p}_i) := -b_{G_i,\max}$
6: Set $\tau(\mathsf{p}_i) := \mathsf{p}_i$, and $\nabla_{\mathsf{H}}(\mathsf{p}_i) := 0$
7: **for all** $\mathsf{q}_k \in \mathcal{N}(\mathsf{p}_i)$ **do**
8: Set $\eta(\mathsf{q}_k) := \mathsf{q}_k$
9: **end for**
10: **end for**
11: $\mathsf{Q} := Q(G)$ ▷ List of not-yet-extended nodes
12: **while** $\mathsf{Q} \neq \emptyset$ **do**
13: Get $\mathsf{r} := \arg\min_{\mathsf{q}_k \in \mathsf{Q}} g_S(\mathsf{q}_k)$ ▷ Extend to node r
14: Let $\mathsf{Q} := \mathsf{Q} \backslash \mathsf{r}$
15: **for all** $\mathsf{w} \in \mathcal{N}(\mathsf{r})$ **do** ▷ Update adjacent vertices
16: Let $g' := g(\mathsf{r}) + d(\mathsf{r}, \mathsf{w})$
17: Let $g'_S := q' - b_{G_{\tau(r)},\max}$
18: **if** $g'_S < g_S(\mathsf{w})$ **then**
19: Set $g(\mathsf{w}) := g'$, and $g_S(\mathsf{w}) := g'_S$,
20: and $\tau(\mathsf{w}) := \tau(\mathsf{r})$
21: **if** $\eta(\mathsf{r}) \neq \emptyset$ **then**
22: Set $\eta(\mathsf{w}) := \eta(\mathsf{r})$
23: **end if**
24: **end if**
25: **end for**
26: **if** $\eta(\mathsf{r}) \neq \emptyset$ **then** ▷ Contribution of r to gradient
27: Set
$$\nabla_{\mathsf{H}}(\tau(\mathsf{r})) := \nabla_{\mathsf{H}}(\tau(\mathsf{r})) + \left.\frac{\partial f(d_S)}{\partial d_S}\right|_{d_S = g_S(\mathsf{r})} \left(\left.\frac{\partial d(\eta(\mathsf{r}), p_i)}{\partial p_i}\right|_{p_i = \tau(\mathsf{r})} - \left.\frac{\partial b_{G_i,\max}}{\partial p_i}\right|_{p_i = \tau(\mathsf{r})} \right) \phi(\mathsf{r})$$
28: **end if**
29: **end while**
30: **for all** $\mathsf{p}_i \in \mathrm{P}$ **do** ▷ Choose next nominal models
31: Replace p_i by
$\arg\max_{\mathsf{w} \in \mathcal{N}(\mathsf{p}_i)} -\nabla_{\mathsf{H}}(\mathsf{p}_i) \cdot (q(\mathsf{w}) - q(\mathsf{p}_i))$
32: **end for**

to each vertex in line 2 and leaves the closest nominal model unspecified. Afterward, the values of nominal model vertices are overwritten in lines 5 and 6 with zero distances, negative stability-distances, and their own indexes (a nominal model is naturally closest to itself). The while-loop from lines 12 to 29 describes the main part of the algorithm. At every iteration, the vertex with the lowest $g_S(\mathsf{q}_k)$ (initially one of the nominal model vertices) is chosen and the distance of its neighbors is calculated by adding the weight of the connecting edge (see line 16). Based on the stability-distance obtained in line 17, the algorithm reasons in line 18 whether the currently expanded node is to be added to the same nominal model as its source. By doing so, the set of vertices assigned to one model expands from the nominal model vertex and step by step replace all initially infinite distances by the true values. The wavefronts, expanding from different nominal vertices, collide at the boundary of their

Voronoi diagrams. In Fig. 4(b) for instance, the values of $\tau(\mathsf{q}_k)$ obtained with Algorithm 1 are included as color-coded circles.

Algorithm 1 also performs a discretized version of the gradient descent (34), for which a few more comments are needed. As was concluded by (33), it is beneficial (from a computational point of view) to express the partial derivative of the distance in terms of a reduced set of points η in the vicinity of p_i. In the graph-based representation, the set of neighboring vertices to p_i constitutes a suitable choice for η. During the expansion phase of the Dijkstra algorithm one thus has to keep track of which vertex in the vicinity of p_i lies on the shortest path to a vertex q_k. For this purpose, the variable $\eta(\mathsf{q}_k)$ assigns to each q_k the unique vertex which is: 1) in the neighborhood of its associated nominal vertex $\tau(\mathsf{q}_k)$, i.e., $\eta(\mathsf{q}_k) \in \mathcal{N}(\tau(\mathsf{q}_k))$, and 2) on the shortest path between q_k and the nominal vertex $\tau(\mathsf{q}_k)$. All possible vertices, that $\eta(\mathsf{q}_k)$ can point to, are highlighted in Fig. 4(b) by circles with black filling. With these definitions in mind, the continuous gradient in (34) is approximated by

$$\nabla_{\mathsf{H}}(\mathsf{p}_i) = \sum_{\mathsf{q}_k \in \mathsf{V}_i} \left.\frac{\partial f(d_S)}{\partial d_S}\right|_{d_S = \mathsf{d}_{G,S}(\mathsf{q}_k, \mathsf{p}_i)} \cdot \left(\frac{\partial d(\eta(\mathsf{q}_k), p_i)}{\partial p_i} - \frac{\partial b_{G_i,\max}}{\partial p_i}\right) \phi(\mathsf{q}_k). \quad (40)$$

Note that for a specific p_i the partial derivative $\frac{\partial d(\eta, p_i)}{\partial p_i}$ is now restricted to a finite set of cardinality $|\mathcal{N}(\mathsf{p}_i)|$. It is therefore reasonable to compute and store all possible values during the initialization period of the graph.

Analyzing the details of Algorithm 1, one can see the initialization of vertices that qualify as η in line 8. Afterward, during the expansion period, line 22 determines $\eta(\mathsf{q}_k)$ for each vertex. Based on $\eta(\mathsf{q}_k)$, line 27 calculates the contribution of vertex q_k to the gradient in (40). Finally, the best next vertices are determined in line 31 by evaluating the benefit of all possible steps (induced by the grid) in terms of the approximated gradients $\nabla_{\mathsf{H}}(\mathsf{p}_i)$.

Algorithm 1 causes the nominal vertex set P to converge to a centroidal configuration, which corresponds to a local minimum. As the convergence of the gradient descent is deterministic, the same initial configurations will always result in the same centroidal configuration. Hence, the algorithm may have to be executed multiple times with different initial configurations in order to determine the global optimum. In the second algorithm, which is proposed next, the set of minima to which the algorithm converges is reduced by seeking pairwise-optimal configurations.

E. Pairwise-Optimal Partitioning Approach

In the following, we present a second approach to optimize the distribution of nominal vertices p_i in the vertex set $Q(G)$, which is a modification of the gossip approach presented in [38]. The task in [38] is to monitor a nonconvex environment with a team of robots. It is proposed to use the sparse communication (gossip) between adjacent robots in order to compute pairwise optimal configurations. For the coverage problem in this paper, we can neglect the sparse communication and consider a centralized version of the gossip approach. The pseudocode is given by Algorithm 2 with additional explanations given in the following paragraphs.

Algorithm 2: Pairwise-Optimal Partitioning.

1: Initialize p_i and let $\mathsf{W}_i := \mathsf{V}_i(\mathsf{P})$, $i = 1, \ldots, N$
2: **repeat**
3: Randomly choose p_i, p_j with adjacent W_i, W_j
4: Let $\mathsf{U} := \mathsf{W}_i \cup \mathsf{W}_j$
5: Let S be a list of all pairs of vertices in U
6: Set $\mathsf{p}_a^* := \mathsf{p}_i$, $\mathsf{W}_a^* = \mathsf{W}_i$, and $\mathsf{p}_b^* := \mathsf{p}_j$, $\mathsf{W}_b^* = \mathsf{W}_j$
7: **for all** $(\mathsf{q}_a, \mathsf{q}_b) \in S$ **do**
8: Compute partitions
9: $\mathsf{W}_a := \{\mathsf{q} \in \mathsf{U} : \mathsf{d}_U(\mathsf{q}, \mathsf{q}_a) \leq \mathsf{d}_U(\mathsf{q}, \mathsf{q}_b)\}$
10: $\mathsf{W}_b := \{\mathsf{q} \in \mathsf{U} : \mathsf{d}_U(\mathsf{q}, \mathsf{q}_a) > \mathsf{d}_U(\mathsf{q}, \mathsf{q}_b)\}$
11: **if** $\check{\mathsf{H}}(\mathsf{q}_a, \mathsf{W}_a) + \check{\mathsf{H}}(\mathsf{q}_b, \mathsf{W}_b) < \check{\mathsf{H}}(\mathsf{p}_a^*, \mathsf{W}_a^*) + \check{\mathsf{H}}(\mathsf{p}_b^*, \mathsf{W}_b^*)$ **then**
12: Set $\mathsf{p}_a^* := \mathsf{q}_a$, $\mathsf{W}_a^* := \mathsf{W}_a$,
13: and $\mathsf{p}_b^* := \mathsf{q}_b$, $\mathsf{W}_b^* := \mathsf{W}_b$
14: **end if**
15: **end for**
16: **until** pairwise-optimal partition for P, W is reached

The goal of Algorithm 2 is to decrease the coverage function (37) by optimizing pairs of adjacent partitions. Hence, after the initial distribution and partitioning in line 1 of the algorithm, a pair of adjacent partitions is chosen at random in line 3. Next, the adjacent partitions are combined in line 4 before all possible combinations of nominal vertices are listed in S in line 5. Then, the code from lines 7 to line 15 iterates over all possible vertex combinations and determines the corresponding coverage function values. The current choice of nominal vertices is replaced in lines 12–13 by a new combination that leads to a lower value in the coverage function. This corresponds to determine the two-pair optimal partition for the union of the adjacent partitions W_i and W_j. Then, the nominal vertices and partitions of the considered partitions are updated with the obtained optimum. This update always yields an improved coverage if p_i, p_j and W_i, W_j are not yet pairwise optimal. Repeating this process for a random sequence of pairs of adjacent partitions ultimately results in a pairwise-optimal configuration. Hence, the algorithm terminates in line 16 if the nominal vertex set P and the corresponding partitions W form a pairwise-optimal partition.

Note that the computational complexity of the pairwise optimization in Algorithm 2 grows with the number of combinations in S. Hence, the finer the quantization of the uncertainty set $\mathcal{Q}$, the more combinations need to be evaluated. For intermediate iterations, this effect can be circumvented by returning suboptimal improvements.

In summary, it can be noted that Algorithm 1 requires less computational power ($\mathcal{O}(kn^2)$ per iteration, where k is the average number of quantized models per dimension), but provides only local optima in the form of Centroidal Configurations. Algorithm 2 on the other side converges to pairwise-optimal configurations, which forms a subset of the Centroidal Configurations with reduced values of the coverage function. This

reduction comes at the cost of increased computational complexity ($\mathcal{O}(k^3n^4)$). A combination of both algorithms allows us to exploit their advantages while attenuating their disadvantages. In the combined algorithm, apply the gradient descent of Algorithm 1 until a Centroidal Configuration is reached. Then, apply Algorithm 2 to validate whether the obtained solution is pairwise-optimal. If not, continue the gradient descent on the intermediate result returned by Algorithm 2. Repeat these steps until Algorithm 2 confirms pairwise optimality of the final configuration. While the curse of dimensionality may not be avoided in the presented algorithms, the computational load may be kept low by carefully increasing the number of quantized models k per dimension of the uncertainty set.

F. Sources of Suboptimality

In this section, we have presented two algorithms to optimize the distribution of candidate model-controller pairs in MMAC. While both algorithms are based on the rigorous coverage function from Section IV-A, various pragmatic considerations lead to problem approximations, which ultimately render the obtained solutions only near optimal. In order to make these sources of suboptimality transparent, a short summary is in order.

The central source for suboptimality is quantization, which immediately restricts the search space to suboptimal configurations. By making the quantization finer, however, the corresponding error can be reduced. Also measuring distances on the graph by summing edge weights yields two sources of suboptimality. First, there is an error due to the triangle inequality, which again may be reduced through finer quantization. Second, the obtained distance is only accurate under the assumption that the actual ν-gap is monotonically increasing along the shortest path between two models. If this assumption is violated, the partitions $\mathcal{V}_i$ turn out to be unconnected. The presented algorithms would then assign a different partition V_j to each of the connected subsets, which results in a suboptimal number of models. This problem may not be overcome with finer quantization. Overall, the quantization step was made to tradeoff computational load and accuracy of the obtained solution. While quantization enabled the derivation of a systematic approach for optimized model distribution in MMAC, the tradeoff subsists in terms of quantization fineness and computational load against suboptimality.

Furthermore, both the gradient descent and the pairwise optimization only guarantee convergence to local minima. Hence, if the algorithms are not evaluated starting from a sufficiently large number of initial configurations, the obtained solution may only correspond to a local minimum in the coverage function.

V. Simulation Results

Next, the hybrid algorithm obtained by combining Algorithm 1 and Algorithm 2 is tested in two benchmark examples.

A. SISO System

To validate the proposed algorithms, first consider the benchmark example studied by Anderson [3] and Morse [6]. For the uncertain linear time invariant transfer function

$$\mathcal{G}(s) = q\frac{s-1}{(s+1)(s-2)}, \qquad q \in [1, 40] \tag{41}$$

it was found that a total of 21 nominal model-controller pairs are sufficient to stabilize all configurations from the entire uncertainty set. Note that the scalar multiplicative uncertainty in this example allows to find an infinite continuum of controllers to stabilize the system. Hence, the retrieval of a finite controller covering in this case is (as in [3], [6]) simply motivated by illustrative purposes.

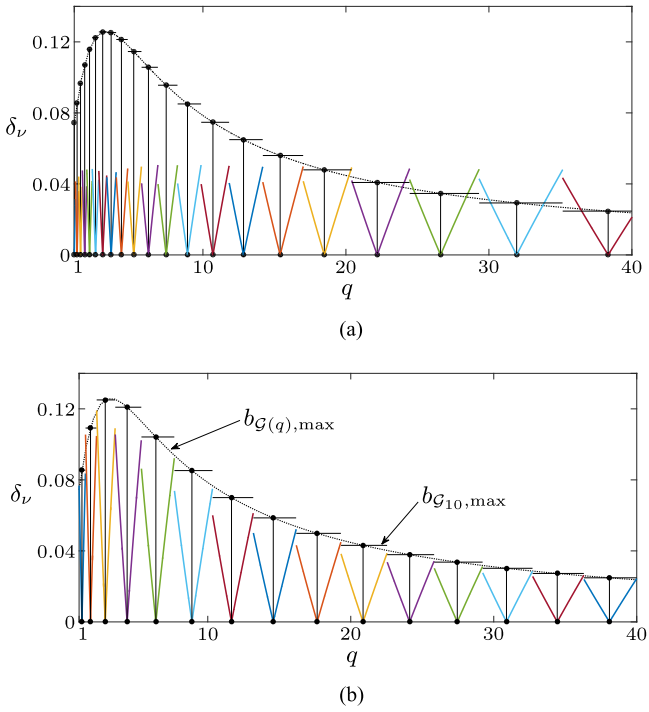


Fig. 5. Comparison of nominal model distribution by (a) trail and error and (b) the proposed coverage formulation. (a) Configuration in [3] with 21 systems. (b) Configuration with 15 systems obtained by proposed algorithms.

Fig. 5(a) analyzes the configurations given in [3, Table 2] in more detail. The dotted line visualizes the achievable stability margin $b_{\mathcal{G},\max}$ as a function of the uncertain parameter q in (41). Next, dots on the q-axis denote the chosen nominal model configurations, which were found in [6] by tedious trial and error. Horizontal lines represent the maximal stability margins of theses nominal models. The colored segments indicate for each configuration q the distance $\min_{p_i} \delta_\nu(\mathcal{G}(q), \mathcal{G}(p_i))$. Stability can be concluded with the ν-gap metric if the colored segments remain underneath the horizontal lines, i.e., $\delta_\nu(\mathcal{G}(q), \mathcal{G}(p_i)) < b_{\mathcal{G}(p_i),\max}$. As the five models to the right of Fig. 5(a) violate this requirement, Anderson *et al.* had to resort to frequency dependent considerations to ensure stability, resulting in additional complexity. Furthermore, Fig. 5(a) shows that the trial and error result does not exploit the maximum stability margin in the interval $q \in [1, 10]$, leading to the overly conservative number of 21 nominal systems.

In order to apply the proposed algorithms, a graph was initialized with equally spaced vertices separated by $\Delta q = 0.01$.

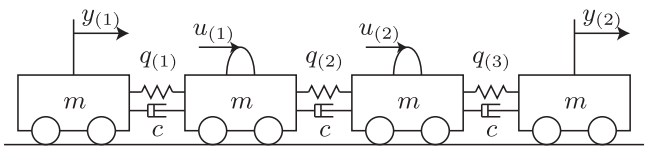


Fig. 6. Four-cart system with uncertain spring constants q.

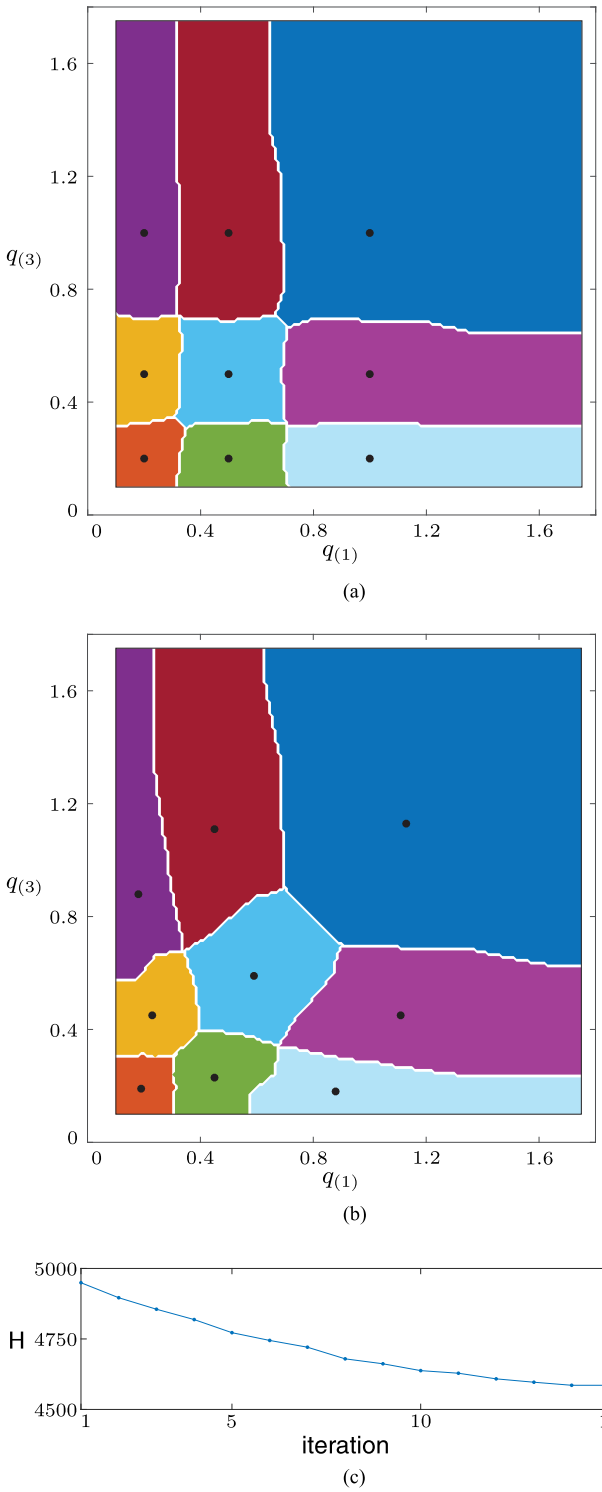


Fig. 7. (a) Beginning from the suboptimal configuration in [20], (b) the coverage control methodology yields the optimized configuration. (c) evolution of the coverage functional during optimization.

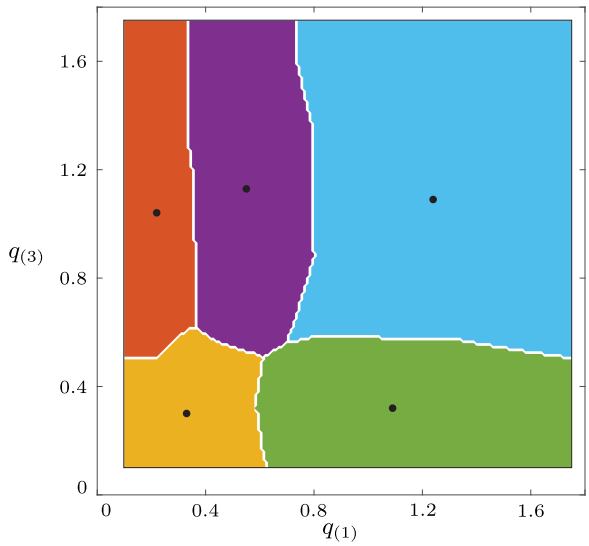


Fig. 8. Optimized covering of the two-cart uncertainty set with 5 model-controller pairs.

With design parameter $\Gamma = 30$ for the inverse controller performance in (23), it is possible to determine a finite controller covering only 15 nominal systems, which are shown in Fig. 5(b). As $\min_{p_i} \delta_\nu(\mathcal{G}(q), \mathcal{G}(p_i)) < b_{\mathcal{G}(p_i),\max}\ \forall q$, it is guaranteed that there exists a stabilizing controller for every possible system configuration. The figure furthermore shows that the larger stability margin in the interval $q \in [1, 10]$ is successfully exploited.

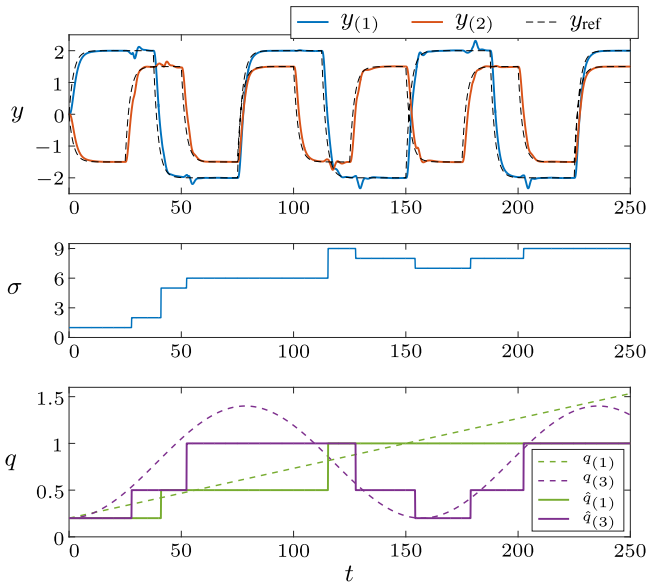
Overall, the proposed algorithm reduces the number of nominal model configurations from 21 to 15 and at the same time removes the need for frequency-dependent stability considerations.

B. Four-Cart System

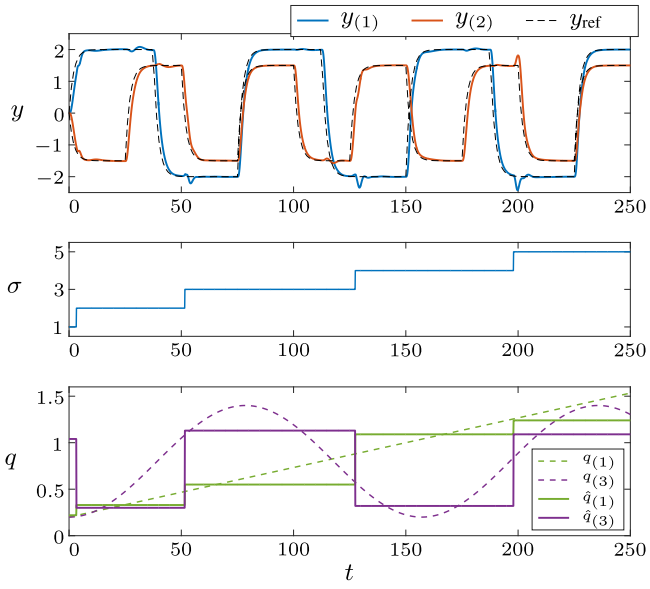
Next consider the four-cart system in Fig. 6, which was previously studied by Baldi *et al.* in [20] and [47]. By applying forces $u_{(1)}$ and $u_{(2)}$ to the inner two carts, the positions $y_{(1)}$ and $y_{(2)}$ of the outer two carts are to be controlled. Let the masses $m = 1\,kg$ and the damping coefficients $c = 0.6\,Ns/m$ be known. Only the spring constants $q_{(1)}, q_{(2)}, q_{(3)}$ are assumed to be uncertain.

For uncertain outer springs $q_{(1)}, q_{(3)} \in [0.1, 1.75]$ and known inner spring $q_{(2)} = 0.7\,N/m$, Baldi *et al.* chose the configuration shown in Fig. 7(a) consisting of nine nominal model-controller pairs. Let the uncertainty set be quantized by a fine grid with equal spacing of $\Delta q_{(i)} = 0.01$. Initializing the proposed optimization algorithm with this configuration, and $\Gamma = 10$ shows that further improvements are possible, leading to the optimized configuration shown in Fig. 7(b). The corresponding evolution of the coverage functional throughout the gradient descent is shown in Fig. 7(c).

It can furthermore be shown that the distribution of nine model-controller pairs is too conservative. Optimization studies reveal that, even if N is reduced to five nominal systems, the proposed algorithms converge to a stabilizing controller distribution, which is shown in Fig. 8. Even though the initial



(a)



(b)

Fig. 9. Comparison of control performance with (a) the original nine model-controller pairs from Fig. 7(a), and (b) the five optimized model-controller pairs from Fig. 8.

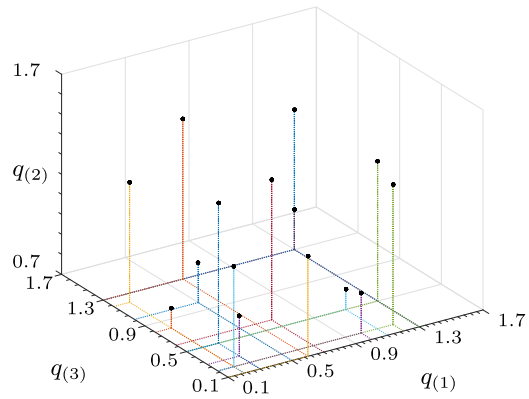


Fig. 10. Optimized model distribution for the four-cart system with uncertain parameters $q_{(1)}, q_{(3)} \in [0.1, 1.7]$ and $q_{(2)} \in [0.7, 1.7]$.

TABLE I
OBTAINED COVERING FOR THREE UNCERTAIN PARAMETERS $q_{(1)}, q_{(2)}, q_{(3)}$

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}
$p_{i(1)}$	0.2	0.2	0.2	0.3	0.3	0.5	0.6	0.6	0.7	1.0	1.1	1.2	1.3	1.3	1.3
$p_{i(2)}$	0.8	1.3	1.2	1.4	0.9	0.9	1.5	1.2	1.4	0.9	0.8	1.4	1.4	0.9	1.4
$p_{i(3)}$	0.8	1.2	0.2	0.5	0.3	1.0	1.3	0.1	0.6	0.2	0.5	0.2	0.5	1.3	1.3

configuration in [20] did not claim optimality, this example gives a good impression of how much benefit may be generated with the proposed algorithms.

Note that reducing the number of nominal controllers does not compromise stability. This is shown in Fig. 9, where reference signal tracking is achieved under time varying spring constants, i.e., $q_{(1)}(t) = 0.2 + 1.6\frac{t}{300}$ and $q_{(3)}(t) = 0.2 + 1.2\sin(0.02t)^2$. The controllers are synthesized with the above introduced combination of $\mathcal{H}_\infty$-controller synthesis and loop shaping [36, Sec. 9.4.2]. Other design constants are $\alpha = \beta = 1, \lambda = 0.5$, and $h = 0.2$. Fig. 9 furthermore shows that fewer nominal models result in less switching, which in turn reduces transient effects in the system outputs due to controller reinitialization [12]. Finally, the visualization of true parameters q and estimated parameters $\hat{q}$, i.e., the constant parameters associated with the currently active model-controller pair, suggests that a greater number of nominal systems enables a more precise tracking of the actual parameters. Nonetheless, a satisfactory tracking is maintained for five optimized models.

Finally, the systematic nature of the proposed algorithms makes the consideration of additional uncertain parameters straightforward. To demonstrate this, assume that also the inner spring constant $q_{(2)}$ takes values from an uncertainty set ranging from 0.7 to 1.7 N/m. Note that this uncertainty set is smaller than before to keep the number of required models reasonable. For a grid with equal spacing of $\Delta q_{(i)} = 0.1$ and design parameter $\Gamma = 10$, it is sufficient to distribute a total of 15 model-controller pairs inside the uncertainty set as shown in Fig. 10. The obtained values are given in Table I.

VI. CONCLUSION

In Multiple-Models Adaptive Control, a finite set of nominal model-controller pairs is needed to ensure stability of the controlled system, which is characterized by large parameter uncertainties. The coverage control methodology applied in this paper yields a first systematic approach to optimize the distribution of nominal model-controller pairs in the uncertainty set. The underlying coverage function, which needs to be minimized, evaluates for each configuration the expected control performance of the closest nominal controller and integrates these individual costs to obtain the overall coverage. Distances are thereby measured in terms of the ν-gap metric, which, in combination with the maximum generalized stability margin, enables stability

considerations needed to ensure that each system is stabilizable by at least one nominal controller. As evaluating the coverage function is computationally involved due to the ν-gap metric, a discretization is proposed, which reduces the optimization to an efficient graph-search. At the same time, this quantization of the uncertainty set limits the problem to near optimal solutions. For the minimization of the discretized coverage functional, two algorithms are presented. First, a gradient descent drives initially suboptimal configurations to centroidal configurations, which constitute local minima. Second, a pairwise optimization strategy yields pairwise optimal configurations at the cost of increased computational load. Finally, combining the two algorithms yields an algorithm which combines the advantage of the gradient descent in form of reduced computational load with the pairwise optimal solutions obtained by the second algorithm.

Interesting directions for future work are the consideration of unstructured uncertainties in the controlled systems and the application to related topics such as modeling with LPV systems or gain scheduling. Also analyzing the positive effect optimized coverings may have on chattering in MMAC is an interesting task. Finally, finding an efficient solution to the continuous problem formulation could remove the quantization step and the corresponding tradeoff between accuracy and elegance of the continuous representation against computational efficiency of the discrete graph.

References

[1] K. Narendra, J. Balakrishnan, and M. Ciliz, "Adaptation and learning using multiple models, switching, and tuning," *IEEE Control Syst. Mag.*, vol. 15, no. 3, pp. 37–51, Jun. 1995.

[2] Z. Han and K. Narendra, "New concepts in adaptive control using multiple models," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 78–89, Jan. 2012.

[3] B. Anderson, T. S. Brinsmead, F. De Bruyne, J. P. Hespanha, D. Liberzon, and A. Morse, "Multiple model adaptive control. Part 1: Finite controller coverings," *Int. J. Robust Nonlinear Control*, vol. 10, no. 11–12, pp. 909–929, 2000.

[4] J. P. Hespanha, D. Liberzon, A. Morse, B. Anderson, T. S. Brinsmead, and F. De Bruyne, "Multiple model adaptive control. Part 2: Switching," *Int. J. Robust Nonlinear Control*, vol. 11, no. 5, pp. 479–496, 2001.

[5] B. Anderson, T. S. Brinsmead, D. Liberzon, and A. Morse, "Multiple model adaptive control with safe switching," *Int. J. Adapt. Control Signal Process.*, vol. 15, no. 5, pp. 445–470, 2001.

[6] A. Morse, "Supervisory control of families of linear set-point controllers—Part 1: Exact matching," *IEEE Trans. Autom. Control*, vol. 41, no. 10, pp. 1413–1431, Oct. 1996.

[7] A. Morse, "Supervisory control of families of linear set-point controllers—Part 2: Robustness," *IEEE Trans. Autom. Control*, vol. 42, no. 11, pp. 1500–1515, Nov. 1997.

[8] F. M. Pait and F. Kassab, "On a class of switched, robustly stable, adaptive systems," *Int. J. Adapt. Control Signal Process.*, vol. 15, no. 3, pp. 213–238, 2001.

[9] J. P. Hespanha, D. Liberzon, and A. Morse, "Hysteresis-based switching algorithms for supervisory control of uncertain systems," *Automatica*, vol. 39, no. 2, pp. 236–272, 2003.

[10] J. P. Hespanha, D. Liberzon, and A. Morse, "Overcoming the limitations of adaptive control by means of logic-based switching," *Syst. Control Lett.*, vol. 49, no. 1, pp. 49–65, 2003.

[11] S.-Y. Cheong and M. G. Safonov, "Slow-fast controller decomposition bumpless transfer for adaptive switching control," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 721–726, Mar. 2012.

[12] F. R. Pour Safaei, J. P. Hespanha, and G. Stewart, "On controller initialization in multivariable switching systems," *Automatica*, vol. 48, no. 12, pp. 3157–3165, 2012.

[13] S. Fekri, M. Athans, and A. Pascoal, "Issues, progress and new results in robust adaptive control," *Int. J. Adapt. Control Signal Process.*, vol. 20, no. 10, pp. 519–579, 2006.

[14] M. Kuipers and P. Ioannou, "Multiple model adaptive control with mixing," *IEEE Trans. Autom. Control*, vol. 55, no. 8, pp. 1822–1836, Aug. 2010.

[15] L. Giovanini, G. Sanchez, and M. Benosman, "Observer-based adaptive control using multiple-models switching and tuning," *IET Control Theory Appl.*, vol. 8, no. 4, pp. 235–247, 2014.

[16] S. Baldi and P. Ioannou, "Stability margins in adaptive mixing control via a Lyapunov-based switching criterion," *IEEE Trans. Autom. Control*, vol. 61, no. 5, pp. 1194–1207, May 2016.

[17] M. Safonov and C. T. Tung, "The unfalsified control concept and learning," *IEEE Trans. Autom. Control*, vol. 42, no. 6, pp. 843–847, Jun. 1997.

[18] G. Battistelli, E. Mosca, M. Safonov, and P. Tesi, "Stability of unfalsified adaptive switching control in noisy environments," *IEEE Trans. Autom. Control*, vol. 55, no. 10, pp. 2424–2429, Oct. 2010.

[19] S. Baldi, G. Battistelli, E. Mosca, and P. Tesi, "Multi-model unfalsified adaptive switching supervisory control," *Automatica*, vol. 46, no. 2, pp. 249–259, 2010.

[20] S. Baldi, G. Battistelli, D. Mari, E. Mosca, and P. Tesi, "Multi-model unfalsified switching control of uncertain multivariable systems," *Int. J. Adapt. Control Signal Process.*, vol. 26, no. 8, pp. 705–722, 2012.

[21] M. H. Chang and E. J. Davison, "Adaptive switching control of LTI MIMO systems using a family of controllers approach," *Automatica*, vol. 35, no. 3, pp. 453–465, 1999.

[22] H. Lehouche, H. Guéguen, and B. Mendil, "Supervisory control based on closed-loop adaptive control approach of nonlinear systems," *Asian J. Control*, vol. 14, no. 1, pp. 258–270, 2012.

[23] G. Vinnicombe, "Frequency domain uncertainty and the graph topology," *IEEE Trans. Autom. Control*, vol. 38, no. 9, pp. 1371–1383, Sep. 1993.

[24] G. Vinnicombe, "A v-gap distance for uncertain and nonlinear systems," in *Proc. 38th IEEE Conf. Decis. Control*, 1999, pp. 2557–2562.

[25] H. Mahdianfar, S. Ozgoli, and H. R. Momeni, "Robust multiple model adaptive control: Modified using v-gap metric," *Int. J. Robust Nonlinear Control*, vol. 21, no. 18, pp. 2027–2063, 2011.

[26] F. Gao, S. E. Li, D. Kum, and H. Zhang, "Synthesis of multiple model switching controllers using $\mathcal{H}_\infty$ theory for systems with large uncertainties," *Neurocomputing*, vol. 157, pp. 118–124, 2015.

[27] X. Li, Z. Zhanlue, and B. L. Xiao, "General model-set design methods for multiple-model approach," *IEEE Trans. Autom. Control*, vol. 50, no. 9, pp. 1260–1276, Sep. 2005.

[28] D. Buchstaller and M. French, "Robust stability for multiple model adaptive control: Part I—The framework," *IEEE Trans. Autom. Control*, vol. 61, no. 3, pp. 677–692, Mar. 2016.

[29] D. Buchstaller and M. French, "Robust stability for multiple model adaptive control: Part II—Gain bounds," *IEEE Trans. Autom. Control*, vol. 61, no. 3, pp. 693–708, Mar. 2016.

[30] L. Vu and D. Liberzon, "Supervisory control of uncertain linear time-varying systems," *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 27–42, Jan. 2011.

[31] D. Vizer and G. Mercère, "$\mathcal{H}_\infty$-Based LPV model identification from local experiments with a gap metric-based operating point selection," in *Proc. Eur. Control Conf.*, 2014, pp. 388–393.

[32] S. Hecker and H. Pfifer, "Affine LPV-modeling for the ADDSAFE benchmark," *Control Eng. Practice*, vol. 31, pp. 126–134, 2014.

[33] K. Glover and D. McFarlane, "Robust stabilization of normalized coprime factor plant descriptions with $\mathcal{H}_\infty$-bounded uncertainty," *IEEE Trans. Autom. Control*, vol. 34, no. 8, pp. 821–830, Aug. 1989.

[34] T. Georgiou and M. Smith, "Optimal robustness in the gap metric," *IEEE Trans. Autom. Control*, vol. 35, no. 6, pp. 673–686, Jun. 1990.

[35] D. McFarlane and K. Glover, "A loop-shaping design procedure using $\mathcal{H}_\infty$ synthesis," *IEEE Trans. Autom. Control*, vol. 37, no. 6, pp. 759–769, Jun. 1992.

[36] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control Analysis and Design*, 2nd ed. Hoboken, NJ, USA: Wiley, 2007.

[37] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.

[38] J. W. Durham, R. Carli, P. Frasca, and F. Bullo, "Discrete partitioning and coverage control for gossiping robots," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 364–378, Apr. 2012.

[39] S. Bhattacharya, R. Ghrist, and V. Kumar, "Multi-robot coverage and exploration on Riemannian manifolds with boundaries," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 113–137, 2014.

[40] Y. Stergiopoulos, M. Thanou, and A. Tzes, "Distributed collaborative coverage-control schemes for non-convex domains," *IEEE Trans. Autom. Control*, vol. 60, no. 9, pp. 2422–2427, Sep. 2015.
[41] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.
[42] L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *Proc. 47th IEEE Conf. Decis. Control*, 2008, pp. 3947–3952.
[43] F. Aurenhammer, "Power diagrams: Properties, algorithms and applications," *SIAM J. Comput.*, vol. 16, no. 1, pp. 78–96, 1987.
[44] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: Applications and algorithms," *SIAM Rev.*, vol. 41, no. 4, pp. 637–676, 1999.
[45] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959.
[46] T. H. Cormen, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.
[47] S. Baldi, G. Battistelli, D. Mari, E. Mosca, and P. Tesi, "Multiple-model adaptive switching control for uncertain multivariable systems," in *Proc. 50th IEEE Conf. Decis. Control Eur. Control Conf.*, 2011, pp. 6153–6158.

Stefan Kersting (S'12) received the Dipl.Ing. degree in electrical engineering from Technical University Darmstadt, Darmstadt, Germany, in 2011. He is currently working toward the Ph.D. degree in control engineering at the Technical University of Munich, Munich, Germany.

He is currently a Research Associate in the Chair of Automatic Control Engineering, Technical University of Munich, where he is working towards his Ph.D. degree. His research interests include adaptive identification and control of hybrid systems and robust switching control.

Martin Buss (F'14) received the Diploma Engineering degree in electrical engineering from the Technical University Darmstadt, Darmstadt, Germany, in 1990, the Doctor of Engineering degree in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1994, and the Habilitation degree from the Technical University of Munich, Munich, Germany, in 2000.

In 1988, he was a research student for one year with the Science University of Tokyo. From 1994 to 1995, he was a Postdoctoral Researcher with the Department of Systems Engineering, Australian National University, Canberra, Australia. From 1995 to 2000, he was a Senior Research Assistant and a Lecturer with the Chair of Automatic Control Engineering, Department of Electrical Engineering and Information Technology, Technical University of Munich. From 2000 to 2003, he was a Full Professor, the Head of the Control Systems Group, and the Deputy Director of the Institute of Energy and Automation Technology, Faculty IV, Electrical Engineering and Computer Science, Technical University Berlin, Berlin, Germany. Since 2003, he has been a Full Professor (Chair) with the Chair of Automatic Control Engineering, Faculty of Electrical Engineering and Information Technology, Technical University of Munich, where he has been with the Medical Faculty since 2008. Since 2006, he has also been the coordinator of the Deutsche Forschungsgemeinschaft Excellence Research Cluster Cognition for Technical Systems with CoTeSys. His research interests include automatic control, mechatronics, multimodal human-system interfaces, optimization, nonlinear, and hybrid discrete-continuous systems.