ADCONIP 2014 Hiroshima

Optimizing the Communication Topology in a Coordinated Model Predictive Control Architecture

Abhay Anand[1] and Lakshminarayanan Samavedham[2]

Abstract— **The design of control architectures for large-scale systems formed by the interconnection of several interacting subsystems is a challenging task. The interactions between the subsystems have a significant influence on the local control decisions as well as the overall system optimality and need to be accounted for explicitly. An increasingly popular technique to control such systems is the Coordination-based Model Predictive Control (C-MPC), where a decentralized control architecture is maintained, but the performance of the system is driven towards that of a centralized control architecture. The C-MPC control strategy facilitates communication between the local controllers and ensures cooperation in order to drive the performance towards plant-wide optimality. However, the amount of communication required increases exponentially with the number of subsystems, making the C-MPC architecture computationally inefficient. Hence, it is desirable to reduce the computational load of the C-MPC architecture without significantly compromising on the overall performance of the system. In order to achieve this, a Genetic Algorithm (GA) based optimizer is utilized to identify the trade-off between communication topologies of varying complexities and the associated performance of the C-MPC strategy. The effectiveness of the optimally designed C-MPC framework with partial communication between the controllers is evaluated on a popular benchmark chemical engineering problem and the performance is compared to that of the traditional centralized and decentralized control architectures.**

I. INTRODUCTION

The design of control systems for complex, networked processes has always been a very challenging task [1]. Since its advent in the second half of the twentieth century, model predictive control (MPC) has evolved into one of the most attractive and successful strategies to control such large-scale integrated systems [2]. The increasing prominence of the application of MPC is due to its suitability to multi-variable constrained processes and the explicit exploitation of real-time information and forecasts [3]. Two popular paradigms for the implementation of MPC are the centralized and decentralized control architectures. In a centralized MPC architecture, a single monolithic controller is designed for the entire system and it is based on a complete description of the entire system with the assumption that the designed MPC controller has access to all measurements. While such an architecture is plant-wide optimal, it is computationally intensive; relatively difficult to implement, tune and maintain; and is characterized by poor fault tolerance. On the other end of the spectrum, a more practically implementable control architecture is the decentralized MPC. In this architecture, the large-scale system is divided into multiple subsystems and an individual controller is designed for each of these subsystems with access to only a local set of measurements. Though such a control system is flexible, reliable and easy to implement, it leads to solutions that are not plant-wide optimum and often the stability of such an architecture is not guaranteed [4]. The centralized and decentralized control paradigms define the limiting extremes of controller design.

The escalating importance of the economic efficiency of systems has necessitated the need for control architectures that address the shortcomings of both the traditional control paradigms. Over the last few years, researchers have focused on developing control architectures that combine the advantages of both the centralized and decentralized control schemes while simultaneously addressing their drawbacks [5]. This has resulted in the development of coordinated control architectures, where a decentralized topology is maintained and the performance of the individual controllers are driven towards plant-wide optimality via the utilization of a coordinator and has popularly been referred to as the coordination-based MPC (C-MPC) architecture [6], [7].

The loss in performance of the decentralized architecture is attributed to the loss in information caused by the decentralization when individual independent controllers are designed for local subsystems and the subsystem interactions are either completely or partially ignored. Most large-scale systems are characterized by a large number of complex and strongly interacting subsystems and these interactions have a significant influence both on the local control decisions as well as the overall plant-wide optimality. In most systems, neglecting these interactions or inadequately coordinating the subsystems, leads to suboptimal system performance and in extreme cases the stability of the system is compromised. The C-MPC architecture aims at explicitly accounting for the interaction effects that exist between the different subsystems of the complex large-scale process. The quantification and utilization of the interaction effects and models form the core of the C-MPC architecture design [8]. The coordinator enables the exchange of information such as states, predicted output trajectory information and calculated control action at each time step between the local controllers, to decide the best set of control actions for each individual controller. Also, the coordinator modifies the local models and cost

*The research presented in this work was carried out as part of the SDWA Multi-Objective Multiple-Reservoir Management research programme (R-303-001-005-272).

[1]Abhay Anand is with the Department of Chemical and Biomolecular Engineering, National University of Singapore, Singapore abhay@nus.edu.sg

[2]Lakshminarayanan Samavedham is with the Department of Chemical and Biomolecular Engineering, National University of Singapore, Singapore laksh@nus.edu.sg

functions utilized by the individual controllers to a well-defined common structure that enables communication and cooperation between the individual controllers [9].

There have been a few studies on C-MPC strategies originating from different schools of thought and in the last few years there have been few comprehensive comparisons between the various C-MPC strategies [10], [11]. However, most of the literature on C-MPC has been derived based on the assumption that all subsystem interactions are equally important. A coordinator designed in such a manner is computationally very intensive and can coordinate the individual MPCs only when information from every one of the subsystems is available. There have been a few studies recently on the optimization of the system topology based on process knowledge and heuristics derived from the expert knowledge of plant operators [12]. However, there is a dearth of literature on the design of MPC coordinators that prioritize the various interactions and coordinates the individual MPCs based on the relative significance of interactions. Such a strategy, though not plant-wide optimal, will be able to lead to near plant-wide optimal performance with a significantly lesser computational burden. The need of the hour is to design C-MPC algorithms with minimum information exchange between the local controllers such that the degree of decentralization is maximized while at the same time the performance degradation is minimized [13]. In this sense, the controller topology (interaction structure) is first optimized and then the local controllers are selectively coordinated. In order to identify the most advantageous controller topology we have utilized a genetic algorithm (GA) based optimizer [14]. A multi-objective optimization problem is formulated to maximize the degree of decentralization (minimize information exchange between controllers) and minimize the performance degradation. The GA optimization results in various optimal controller topologies with varying degrees of decentralization. Based on the desired computational improvement, within the permissible performance deterioration, the preferred controller topology with minimum information exchange can be utilized.

This paper presents a novel C-MPC framework with an optimized communication topology. We have selected and extended MPC coordination technique from literature and optimized the controller topology before implementing the C-MPC strategy. This novel GA based topology optimization strategy has been evaluated on a popular benchmark chemical engineering case study. In the next section, the C-MPC algorithm is briefly formulated and the communication bottleneck is highlighted. In the subsequent section, the formulation of the controller architecture with optimal communication topology is described in detail. These form the methodological framework that is employed in the subsequent case study described in the next section. The key conclusion and empirical evaluations derived from this work are summarized in last section.

II. COORDINATING MULTIPLE MODEL PREDICTIVE CONTROLLERS

The initial formulations of C-MPC were based on the assumption that exchanging information (in the form of predicted state trajectories and calculated control trajectories) were sufficient to drive the performance of decentralized controllers towards plant-wide optimality. However, it was later proved that exchange of information was insufficient to guarantee system stability and in systems where the interactions were severe, such a strategy would lead to competing controllers that do not converge to a unique solution [15]. In order to overcome this deficiency it was shown that a C-MPC scheme needed to enable the cooperation among the local controllers over and above the communication network. These observations formed the basis for most of the C-MPC schemes that have been formulated in the recent past [16].

As mentioned in the previous section, the main task of the coordinator is to predict the effect of the interactions between subsystems and to enable the local controllers to modify the local decisions to balance out the interaction effects. In order to achieve this, the C-MPC strategy aims at enabling communication (exchange of states, predicted output trajectory and calculated control action at each time step) and cooperation (through a modified local objective) between the controllers.

Consider a large-scale MIMO system comprised of M individual subsystems. The discrete transfer function matrix formulation of the system, where $\tilde{y}$ denotes the output variable vector and $\tilde{u}$ denotes the manipulated variable vector and $\tilde{d}$ denotes the measured disturbance vector is as follows:

$$\tilde{y} = \tilde{G}_p \tilde{u} + \tilde{G}_d \tilde{d} \tag{1}$$

where $\tilde{G}_p$ is the process transfer function matrix and $\tilde{G}_d$ is the disturbance transfer function matrix.

For large-scale system $\tilde{G}_p$ is defined as,

$$\tilde{G}_p = \begin{bmatrix} G_{11}(z) & G_{12}(z) & . & . & G_{1M}(z) \\ G_{21}(z) & G_{22}(z) & . & . & G_{2M}(z) \\ . & . & . & . & . \\ . & . & . & . & . \\ G_{M1}(z) & G_{M2}(z) & . & . & G_{MM}(z) \end{bmatrix} \tag{2}$$

In the process transfer function matrix $\tilde{G}_p$, the off-diagonal transfer functions represent the interaction models in the system. For example, the interaction model $\tilde{G}_{ij}(z)$ is a transfer function model between the input to the i-th subsystem and the output of the j-th subsystem. In a decentralized control architecture, $\tilde{G}_p$ would be a diagonal matrix with $\tilde{G}_{ij}(z) = 0$.

The prediction equation for the i-th subsystem will then take the form:

$$Y_i(t) = G_{ii}\Delta u_i(t) + h_i(t) + l_i(t) \tag{3}$$

where,

$$h_i(t) = \sum_{\substack{j=1 \\ j \neq i}}^{M} G_{ij}\Delta u_j(t) \tag{4}$$

$h_i(t)$ denotes the effect of future inputs of other subsystems on the i-th subsystem output.

$$l_i(t) = \sum_{j=1}^{M} F_{ij}x_j(t) + F_{di}x_{di}(t) + Y_{iM}(t) \tag{5}$$

In the above equation, $l_i(t)$ is obtained by summing up the effect of past inputs of all M subsystems (first term), effect of measured disturbances (second term) and current measurements (third term). The prediction equation 3 forms the basis of all control calculations. Generally, the control algorithms for these subsystems are implemented independently in an iterative manner. Since an MPC optimization scheme is being employed, trajectories for the input variables are available at each iteration and this information is exchanged between the subsystem controllers. Through the modified prediction equation (eq. 3), each local MPC is now able to compute the effect of the control decisions calculated by each of the other local controllers over the prediction horizon.

Each MPC transmits the current state and input trajectory information to all interconnected subsystem$'$s MPC through the coordinator. However, each individual controller has no knowledge of the cost functions of other local controllers. The objectives of each MPC controller are frequently in conflict and the equilibrium of such a strategy is driven to a non cooperative Nash equilibrium. Due to the non-cooperative and competing nature of such a strategy, the overall system performance is usually suboptimal and when interactions are strong, closed loop stability is not guaranteed.

Despite having knowledge of the local control decisions, the individual controllers try to achieve their own individual optima and hence do not converge to a global optima. In other words, the controllers are working with the same resources (control variables) but towards satisfying different conflicting objectives leading to a contest between the individual controllers. To overcome this competition between the controllers, the coordinator additionally works towards enabling the controllers to support each other towards reaching global optimality. In order achieve this, the local objective functions of each subsystem MPC controller are converted to a common global objective function. This is achieved by using a weighted convex sum of the individual objective functions as the new objective function.

For each subsystem i, the cost function F_i to be minimized by the local MPC is hence written as:

$$F_i = \sum_{r=1}^{M} w_r J_r(\Delta u_i) \tag{6}$$

$$\sum_{r=1}^{M} w_r = 1 \tag{7}$$

$$F_i = w_i J_i(\Delta u_i) + \sum_{\substack{r=1 \\ r\neq i}}^{M} w_r J_r(\Delta u_i) \tag{8}$$

Here the cost function J_i for the i-th subsystem is of the form,

$$J_i = (R_S^i - Y_i)^T \tilde{Q}_i (R_S^i - Y_i) + \Delta u_i^T \tilde{R}_i \Delta u_i \tag{9}$$

where R_S^i is a vector of the individual subsystem set points, $\tilde{Q}_i$ and $\tilde{R}_i$ are tunable weights selected depending on the system dynamics.

Weights w_r and w_i are assigned to the various objectives heuristically based on the physical or economic significance of the variables being optimized/controlled at each subsystem. Subsystems or output variables that have a more significant effect on the overall plant operations as designated by the process engineer would be weighed more significantly than the others. In our studies, we have weighed all objectives equally in the cooperation based coordination strategy. Since all the local MPC controllers are solving an optimization problem with a common global objective function, the optimal control profile generated at all iterates of the C-MPC scheme is now plant-wide feasible and closed loop stable (Pareto Optimal). The weights used while coordinating the local MPCs would be the same as the weights used by a centralized MPC (single objective) hence allowing the performance of the C-MPC scheme to converge to the performance of a centralized MPC. While different weights would alter the actual performance, the convergence of the C-MPC scheme to a centralized MPC performance still holds.

The input rate, input and output constraints for the i-th subsystem are written in the form:

$$\Delta u_i^{\min} \leq \Delta u_i \leq \Delta u_i^{\max} \tag{10a}$$

$$u_i^{\min} \leq u_i \leq u_i^{\max} \tag{10b}$$

$$y_i^{\min} \leq y_i \leq y_i^{\max} \tag{10c}$$

for, i = 1, 2, ..., M

Each individual MPC utilizes a suitable optimization algorithm (Quadratic Programming in our work) to minimize the cost function defined by Eq. (9) subject to the constraints defined by equation Eq. (10). Every local MPC, then iteratively exchanges information with the other MPCs and re-optimizes the control decision until convergence or a predefined limit. The reader is referred to Anand et al. [11] for a more detailed derivation of the above described C-MPC strategy and the effect of model-plant mismatch on the C-MPC performance.

The performance of the C-MPC algorithm is quantified through analyzing the deviations from set point and the sum of squared errors (SSE) is selected metric. The *Total SSE* which is the sum of the SSE$'$s of all response variables is given as:

$$SSE = \sum_{i=1}^{M} (Y_i - R_S^i)^2 \tag{11}$$

Most formulations of coordinated and distributed control strategies in literature are hinged on the assumption that every controller communicates with every other controller. The number of interactions increases polynomially ($O(n^2)$) with the number of subsystems and the number of possible communication topologies increases exponenetially ($O(2^{n^2})$

with the number of subsystems. This creates a tremendous load on the communication network impeding its robustness. Methods to reduce the computational load through algorithms that permit only partial communication is the direction in which current research is progressing and a few methods that try to tackle this issue have been researched in recent literature [17]. In this work, we have analyzed a simulation-based optimization methodology to reduce the communication load without compromising significantly on the overall system performance or the closed-loop stability and we have designed a framework to achieve the same.

III. OPTIMIZING THE COMMUNICATION TOPOLOGY

The problem of simultaneously designing both the controller topology and the controller itself has been introduced very recently and is one of the main contributions of this research. In this work, we have first designed the C-MPC control strategy with complete information exchange (as described in the previous section) and then we have optimized the controller topology within the C-MPC architecture. In this section, we formulate the problem of minimizing the information exchange and optimizing the topology of the C-MPC architecture. This problem mathematically translates into maximizing the sparsity of $\tilde{G}_p$ (defined in eq. 2) which in turn increases the degree of decentralization which is quantified through an Interaction Matrix.

The Interaction Matrix is defined as,

$$IM = \begin{bmatrix} im_{11} & im_{12} & . & . & im_{1M} \\ im_{21} & im_{22} & . & . & im_{2M} \\ . & . & . & . & . \\ . & . & . & . & . \\ im_{M1} & im_{M2} & . & . & im_{MM} \end{bmatrix} \tag{12}$$

where

$$im_{ij} = 0, \text{ if interaction model } G_{ij} \text{ is ignored}$$
$$im_{ij} = 1, \text{ if interaction model } G_{ij} \text{ is utilized}$$

and the interaction index (II) of the control topology is defined as:

$$II = \sum_{i=1}^{M} \sum_{j=1}^{M} im_{ij} \tag{13}$$

It is desired to minimize the number of interaction models used in the control architecture. However, as the number of interaction models are reduced, the system performance also deteriorates due to the reduction in information, which is undesired, and hence a multi-objective optimization (MOO) problem is formulated. In this case we have two conflicting objectives, the number of interaction models and the performance deterioration of the system.

Traditionally, multi-objective optimization problems are solved using the min-max formulation, method of distance functions or the method of weighted objectives [18] where the multiple objectives are converted into a single objective. The greatest drawback of this method is the resulting single solution rather than a catalog of equally optimal solution set. Also, the conversion of multiple objectives into a single objectives depends on an a priori requirement of system knowledge. Moreover, these methods involve weighing of objectives, and the tuning of these weights play a significant role in the overall optimality of the solutions. These shortcomings render the single-objective formulation based methods inadequate and unreliable. In order to provide a more pragmatic set of solutions that enable the decision maker to choose the most appropriate decision based on his current requirements, algorithms that explicitly handle multiple objectives were required. There are many methods that solve MOO problems and result in a set of solutions known as pareto optimal solutions. Each of these solutions in the pareto optimal set is better than every other solution in the search space when all the objectives are considered together (though they are inferior to the solutions when one or a few objectives only are considered). Many methods have been developed to find Pareto optimal solutions to MOO problems such as the weighted sum method, ε-constraint method, evolutionary algorithms [19], etc. In this work, we have utilized a multi-objective evolutionary algorithm in the form of the non-dominated sorting genetic algorithm (NSGA-II) implemented in MATLAB. As described in [20] this particular algorithm performs better than other MOO algorithms (Pareto-archived evolution strategy (PAES) and strength Pareto evolutionary algorithm (SPEA)), in terms of elitism and computational complexity.

The first step of the NSGA-II algorithm is to define the population size and the stopping criteria (in this case the number of generations was constrained). Next, within the constraints of the decision variables, a random initial population is defined. The initialized population is sorted into different fronts, based on the non-domination criteria. From the initial population, parents are selected based on two metrics namely rank and crowding distance. The members of the first front belong completely to the non-dominated set while the second front members are dominated only by the first front and so on. Each individual of the population is assigned a rank (fitness) based on the their presence in a particular front. Crowding distance is a metric that measures the closeness of the individuals to their neighbors and a larger crowding distance is preferred to ensure diversity within the population. Individuals are selected as parents based on the rank value and crowding distance. The parent population then undergoes genetic operations such as crossover and mutation to generate the next generation of the population known as children. The children population is then combined with the current parent population in a step called recombination. These steps are repeated iteratively till the stopping criteria are achieved. Elitism of the algorithm is assured, since the best individuals from both the previous and current population are utilized. The parameters involved in setting up the NSGA-II algorithm are the number of generations and the probabilities for the crossover and mutation processes. One of the challenges associated with this algorithm is the decision of the initial population size and maximum permissible number of iterations. Though a larger population and number of iterations are preferred, they significantly

increase the computational cost and need to be limited yet sufficient. In order to select these parameters efficiently, preliminary simulations were performed to assess their effect and optimize them. In this work, a modified version of the program developed at the Illinois Genetic Algorithms Laboratory [21] was used and the parameters used in the setting up of the NSGA-II algorithm are specified in Table I.

TABLE I
NSGA PARAMETERS

Parameter	Value
Objective 1	Minimize the SSE
Objective 2	Minimize the II
Constraints	Binary Decision Variables
Replacement Proportion	0.9
Selection Method	Roulette Wheel
Crossover Method	One Point Crossover
Crossover Probability	0.9
Mutation Method	Selective Mutation
Mutation Probaility	0.1
Stopping Criteria	Maximum Generations
Population Size	36
Maximum Generations	200

IV. CASE STUDY: SHELL BENCHMARK PROBLEM

The Shell benchmark problem was originally designed as a generic benchmark for control studies by Prett et al. [22]. It is model of a heavy oil fractionator characterized by three side circulating loops and three product draws. The output variables are the compositions at the top and side draws and the reflux temperature while the manipulated variables are the top and side draw rates and also the reflux heat duty. The main objective of this control problem is to maintain the draw compositions at a desired setpoint. In this work, the model presented by Li et al. [23] has been utilized.

The model of the process is: $y = G(s)u + G_d(s)d$ with the transfer function matrices:

$$G(s) = \begin{bmatrix} \frac{4.05e^{-27s}}{50s+1} & \frac{1.77e^{-28s}}{60s+1} & \frac{5.88e^{-27s}}{50s+1} \\ \frac{5.39e^{-18s}}{50s+1} & \frac{5.72e^{-14s}}{60s+1} & \frac{6.90e^{-15s}}{40s+1} \\ \frac{4.38e^{-20s}}{33s+1} & \frac{4.42e^{-22s}}{44s+1} & \frac{7.20}{19s+1} \end{bmatrix} \text{ and}$$

$$G_d(s) = \begin{bmatrix} \frac{1.44e^{-27s}}{40s+1} \\ \frac{1.83e^{-15s}}{20s+1} \\ \frac{1.26}{32s+1} \end{bmatrix}$$

The constraints include, $|y_i| \leq 0.5$, $|u_i| \leq 0.5$, $|\Delta u_i| \leq 0.5$ for $i = 1,2,3$

The interactions are quantified using the relative gain array (RGA). The RGA for the system is

$$\begin{bmatrix} 2.0757 & -0.7289 & -0.3468 \\ 3.4242 & 0.9343 & -3.3585 \\ -4.4999 & 0.7946 & 4.7053 \end{bmatrix}$$

The presence of significant off-diagonal terms indicates the existence of subsystem interactions that exist in the system. The simulation results (Fig. 1), indicated that the effect of these interactions were strong and though the decentralized control strategy (0 interactions) yielded a closed loop stable solution, its performance was significantly degraded as compared to the centralized controller performance (6 interactions). In order to assess the effectiveness of the C-MPC control architecture, the GA algorithm was implemented and the 6 interaction models (decision variables) were optimized while configuring the communication topologies.

A. Results and Discussion

Model based controllers based on transfer function models were derived and implemented in MATLAB version 7.11.0.584 and a tuning strategy derived by Shridhar et al. [24] was used to tune the individual multivariable MPCs. In order to assess the robustness of the developed control algorithm, set point changes as well as step disturbances were introduced to all the subsystems at different sampling instants. The results as seen in Fig. 1, highlight the utility of the GA optimization algorithm in selecting communication architectures with varying topologies of different complexities. While the top panel of the figure depicts the performance of the controller while utilizing a different number of interaction models, the lower panel shows the corresponding optimal interaction models that were selected to achieve the depicted performance. The C-MPC scheme is applicable to systems where a decentralized MPC is feasible. The decentralized MPC by itself can be considered as an C-MPC scheme where zero interactions are accounted for. Hence, even if a single interaction is included in the coordination scheme the performance of the system improves and it is seen that including a single interaction model (im_{13}), if selected optimally can improve the decentralized performance by nearly 50% as seen in Table II. Also, it is seen that, by increasing the number of interactions, the performance improves (SSE decreases) and asymptotically converges to the performance of a centralized controller (a theoretical benchmark). Through utilizing the GA optimization algorithm, the importance of the individual interactions as well as the selection of an optimal control topology with limited communication (as desired by plant operators) can be achieved.

TABLE II
PERFORMANCE COMPARISON OF THE VARIOUS OPTIMAL CONTROL TOPOLOGIES FOR THE SHELL BENCHMARK PROBLEM

No. of Interactions	SSE with respect to Decentralized MPC
Decentralized MPC	1.00
1	0.53
2	0.42
3	0.34
4	0.29
5	0.24
6	0.17
Centralized MPC	0.16

When more interaction models are included in the control topology, the required computational time would also increases. The increase in computational time can be attributed

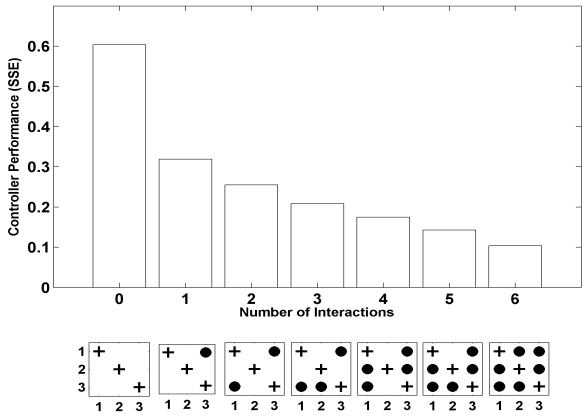

Fig. 1. Performance comparison of the shell benchmark problem for various optimal interaction topologies ($im_{31} = 1$)

to the increase in communication and the exchange of information between a larger number of controllers. However, the results of this work would provide the control practioners with a trade-off option between desired performance improvement and available computational resources. For example, if the time constraints permit only the selection of 3 interaction models, then the models im_{13}, im_{31} and im_{32} need to selected and a performance enhancement of 65.6% can be achieved. Conversely, if a performance enhancement of 50% is sufficient, only 2 interaction models (im_{13} and im_{31}) need to be utilized.

V. CONCLUSIONS

The novel GA based communication topology optimization and the implementation of the partial communication based C-MPC controller scheme presented in this work has been shown to significantly alleviate the bottleneck in traditional coordinated control architectures. It has been shown that the demands of enabling communication between each and every local MPC can be lessened through optimizing the communication network and utilizing only those interaction models that significantly improve the overall performance of the system. It was seen that a performance improvement of over 50% can be achieved by optimally selecting only a few interaction models. However, the selection of the interaction models needs to be optimized in order to ensure maximum performance enhancement. The algorithm developed in this work has been used to prioritize the interactions and configure communication topologies of varying complexities successfully. The set of pareto optimal solutions provided by the GA optimizer provides additional flexibility to the control practioners. The GA based communication topology optimization of the C-MPC architectures has shown to improve the closed loop performances significantly over the decentralized control strategies. The results of this work will be extended to the optimal design of control architectures for large-scale systems after rigorous analyses and studies on the effect of model-plant mismatch on the optimized communication network.

REFERENCES

[1] Qin, S. & Badgwell, T. A survey of industrial model predictive control technology, Control Engineering Practice, 2003, 11, 733-764.

[2] Scattolini, R. Architectures for distributed and hierarchical Model Predictive Control - A review, Journal of Process Engineering, 2009, 19, 723-731.

[3] Camacho, E. & Bordons, C. (Eds.) Model predictive control Springer, 2003.

[4] Cui, H. & Jacobsen, E. Performance limitations in decentralized control, Journal of Process Control, 2002, 12, 485-494.

[5] Rawlings, J. & Stewart, B. Coordinating optimization based controllers: New opportunities and challenges, Journal of Process Control, 2008, 18, 839-845.

[6] Marcos, N.; Forbes, J. & Guay, M. Coordination of distributed model predictive controllers for constrained dynamic processes, Proceedings of the International Symposium on Advanced Control of Chemical Processes, 2009.

[7] Negenborn, R.; van Overloop, P. & Schutter, B. D. Coordination of local controllers in large-scale water systems, Proceedings of the 9th International Conference on Hydroinformatics, 2010.

[8] Mesarovic, M.; Macko, D. & Takahara, Y. Two coordination principles and their application in large scale systems control, Automatica, 1970, 6, 261-270.

[9] Venkat, A.; Rawlings, J. & Wright, S. Implementable distributed model predictive control with guaranteed performance properties, Proceedings of the IEEE American Control Conference, 2006.

[10] Alvarado, I.; Limon, D.; de la Pena, D. M. & et al. A comparative analysis of distributed MPC techniques applied to the HD-MPC four tank benchmark, Journal of Process Control, 2011, 21, 800-815.

[11] Anand, A.; Samavedham, L. & Sundaramoorthy, S. Comparative Performance Analysis of Coordinated Model Predictive Control Schemes in the Presence of Model–Plant Mismatch, Industrial & Engineering Chemistry Research, 2012, 51, 8273-8285.

[12] Nunez, A.; Ocampo-Martinez, C.; De Schutter, B.; Valencia, F.; Lopez, J. D. & Espinosa, J. A. Multiobjective-Based Switching Topology for Hierarchical Model Predictive Control Applied to a Hydro-Power Valley, Proceedings of the IFAC International Conference on Intelligent Control and Automation Science, 2013, 3, 529-534.

[13] Schuler, S.; Muenz, U. & Allgower, F. Optimal controller structure reduction for decentralized control, Proceedings of the 4th IFAC Symposium on System, Structure and Control, 2010, 303-308.

[14] Deb, K. Multi-objective optimization, Multi-objective optimization using evolutionary algorithms, John Wiley & Sons Hoboken, NJ, 2001, 13-46.

[15] Venkat, A. Ph.D. Dissertation, Distributed Model Predictive Control: Theory and Applications, University of Wisconsin-Madison, 2006.

[16] Stewart, B.; Wright, S. & Rawlings, J. Cooperative distributed model predictive control for nonlinear systems, Journal of Process Control, 2011, 21, 698-704.

[17] Schuler, S.; Li, P.; Lam, J. & Allgwer, F. Design of structured dynamic output-feedback controllers for interconnected systems, International Journal of Control, Taylor & Francis, 2011, 84, 2081-2091.

[18] Marler, R. T. & Arora, J. S. Survey of multi-objective optimization methods for engineering Structural and multidisciplinary optimization, Springer, 2004, 26, 369-395.

[19] Tamaki, H.; Kita, H. & Kobayashi, S. Multi-objective optimization by genetic algorithms: A review, Proceedings of IEEE International Conference on Evolutionary Computation, 1996, 517-522.

[20] Deb, K.; Agrawal, S.; Pratap, A. & Meyarivan, T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, Lecture notes in computer science, Springer, 2000, 1917, 849-858.

[21] Sastry, K. Single and multiobjective genetic algorithm toolbox for MATLAB in C++, IlliGAL Report, 2007.

[22] Prett, D. M. & Garcia, C. E. Fundamental process control, Butterworths, 1988.

[23] Li, S.; Zhang, Y. & Zhu, Q. Nash-optimization enhanced distributed model predictive control applied to the Shell benchmark problem, Information Sciences, 2005, 170, 329-349.

[24] Shridhar, R. & Cooper, D. A Tuning Strategy for Unconstrained SISO Model Predictive Control Industrial and Engineering Chemistry Research, 1997, 36, 729-746.