# A New Weighted Graph-based Partitioning Algorithm for Decentralized Nonlinear Model Predictive Control of Large-scale systems

Karim Salahshoor
Department of Automation and instrumentation
Petroleum University of Technology
Ahwaz, Iran

Saeed Kamelian
Department of Automation and instrumentation
Petroleum University of Technology
Ahwaz, Iran

## ABSTRACT

This paper proposes a grouping algorithm for partitioning large-scale nonlinear dynamical systems based on graph theory. The algorithm incorporates a novel scheme to quantify the strengths of graph edges, representing the degree of couplings among the system variables via sensitivity functions. This leads to a weighted graph topology with different weights on the obtained graph edges. An algorithm is then developed to partition systems into some sub-graphs based on the weighted graph. A decentralized nonlinear model predictive control (NMPC) methodology is then formulated for the sub-systems. The overall NMPC design methodology is finally evaluated on a process plant benchmark, consisting of two continuous stirred tank reactors (CSTRs) and a flash separator with a recycle path. A set of tracking and regulatory tests is comparatively conducted exploring the successful performance of the proposed algorithm in the context of the decentralized NMPC methodology with respect to an alternative centralized NMPC control scheme.

## General Terms

Decentralized control, Graph partitioning algorithms.

## Keywords

Graph partitioning algorithm, Decentralized control, nonlinear model predictive controller, Large-scale systems.

## 1. INTRODUCTION

Large-scale dynamical systems provide new challenges and opportunities in control theory research area. Modern systems are developing both in size and complexity much more than before and hence new efforts for designing novel control methodologies are something inevitable. Properties of large-scale systems like nonlinear dynamics, modeling uncertainties, constraints and presences of external disturbances make design, implementation and maintenance of such systems non-trivial. In such complicated context, the traditional control design strategy may show inefficiency. Therefore, some other advanced control methodologies applicable to real world large-scale systems are in great importance.

Model predictive control (MPC) has shown to play effective role in large-scale control system applications. There are lots of reported successful applications of MPC in real-world such as urban water distribution systems [1], electrical power networks [2], process industry [3], manufacturing systems [4] that make model predictive control as one of the most demanding approach in control theory [5]. While the MPC paradigm encompasses several different variants, each one with its own special features, all MPC methods rely on the idea of generating values for system inputs as solutions of an on-line optimization problem on the basis of a system model and the relevant measurements. The model predictive control problem is formulated as solving on-line a finite horizon open-loop optimal control problem, satisfying different control objectives subject to system dynamics and constraints involving states and controls. Regardless of the particular choice made for the MPC ingredients elements, on-line optimization is the common thread tying them together. However, the traditional MPC strategy demands a great amount of online computation, since an optimization problem is solved at each sampling time. This has limited the use of these controllers to systems with relatively slow dynamics. Furthermore, the use of MPC is critically dependent on having a system model of low order, typically with a maximum of ten states.

One centralized controller unit may traditionally be recommended for overall control of a large-scale plant. This centralized control approach, however, can be difficult to be practically implemented due to computational complexity, reliability, robustness, fault tolerances and bandwidth limitation issues. Furthermore, it makes the maintenance activities more difficult and may force the full control system to face with ultimate shutdown due to one major central controller fault [6].

Decentralized MPC or distributed MPC offers an efficient approach to circumvent these issues in which a set of local MPC controllers take in charge of the entire system rather than the centralized MPC controller. This alleviates the adverse consequences due to the centralized MPC malfunctions in price of missing global optimal system responses. In this approach, each local MPC controller caters for its own subsystem which is a small manageable part of the entire system [5].

However, the decentralized MPC approach poses a challenging problem in which the large scale system should be properly decomposed into smaller-size subsystems. This issue has already been addressed in the literature [7, 8, 9] in the general context of decentralized control for large scale systems. The system decomposition problem is traditionally recommended during the system modeling process through cleaver identification of subsystems on the basis of physical insight, intuition or experience. Obviously, this approach is not suitable for a large-scale complex system with many states inputs and outputs. Thus, there is a strong motivation to develop systematic methods to automatically decompose a given large-scale system via its actual structure.

The subject is still very incipient and less method has been proposed in the literature. A new automatic decomposition algorithm has recently been presented [1] based on graph partitioning. The algorithm provides a decomposition scheme, consisting of a set of non-overlapping sub-graphs whose number of vertices is as similar as possible and the number of interconnecting edges between them is minimal. The approach, however, assumes that the system behaves linearly and any inherent dynamic non-linearity effect will be eliminated through the lower regulation-layer controllers. The final decentralized MPC controllers are then designed on the basis of the linear dynamics of the partitioned sub-systems. Furthermore, the developed graph utilizes edges with the same weighting, leading to ignorance of different possible strengths among the system vertices.

In this work, an enhanced systematic method is presented to partition a complex large scale system into smaller-size subsystems without restricting the large-scale system to behave linearly. The approach is hence applicable to nonlinear dynamic large-scale systems. The required information is first extracted via the system dynamics so that the system can be represented by a directed graph. In this way, the constructed graph inherits the system structural information in its essence and hence the system partitioning subject is led to a graph partitioning problem which can use the rich supporting content of the graph theory to be solved [10, 11].

The work presents a novel scheme based upon the sensitivities of the ending graph vertices, indicating system states or inputs, with respect to each other to incorporate different weights on the obtained graph edges.

The graph partitioning algorithm uses the whole system graph and tries to evolve some sub-graphs with almost equal number of vertices and minimal number of weighted interconnecting edges. The algorithm searches for some optimal cuts which result into sub-graphs that share edges with minimal total number and minimal total weights.

The paper is organized as follows. The proposed portioning algorithm is developed in Section 2. Once the large-scale system is decomposed into the constituent sub-systems, the decentralized MPC methodology is then utilized to design the required individual local MPC controllers. The whole procedure is then validated on a CSTR process plant case study in Section 3 to explore the comparative performances of the proposed algorithm with respect to conventional centralized MPC scheme. Section 4 summarizes the concluding remarks.

# 2. DEVELOPMENT OF A SYSTEMATIC PARTITIONING ALGORITHM

## 2.1 Graph representation of a large-scale system

Consider the general state-space representation of the system by

$$\overset{\circ}{x} = f\,(x,u,d) \qquad (1-a)$$
$$y = g\,(x,u,d) \qquad (1-b)$$

Where $x \in \Re^n$ and $\overset{\circ}{x} \in \Re^n$ are state and its derivative vector, respectively, $u \in \Re^m$ is the system input vector and $d \in \Re^d$ is process disturbance vector, $y \in \Re^r$ is output vector and $f\,,g$ indicate some general input and output mapping functions. The problem is to decompose the large-scale system, described by Eqn. (1) into subsystems. First, a weighted graph is constructed using the given general non-linear system model from its physical interconnecting topology [12]. The weighed system graph, represented by $g(V\,,E)$ , consists of the vertex set $V$ which is community of state vector $X = \{x_1, x_2, ..., x_n\}$, input vector $U = \{u_1, u_2, ..., u_m\}$ and output vector $Y = \{y_1, y_2, ..., y_r\}$. $E$ indicates the edge set which shows the connectivity of the two edges (i.e., system variables) in the system equations. Notice that the set E only has $(u_i\,, x_j), (x_i\,, x_j), (x_i\,, y_j)$, as its member [12]. The graph edge weights is derived after linearization of the nonlinear system equations, given in Eqn. (1), to demonstrate each individual system state to input[1], state-to-state[2] and output to state[3] sensitivities. As a consequence, the weighted graph can be represented by the incidence matrix $I_M$ , defined as follow:

$$\mathrm{I}_{M_{ij}} = \begin{cases} -w_{ij} & \text{If edge j leave vertex i} \\ +w_{ij} & \text{If edge j enter vertex i} \\ 0 & \text{Otherwise} \end{cases}$$

Where $w_{ij}$ denotes the sensitivity gain. The incidence matrix $I_M$ will naturally have a dimension equal to $v \times \eta$ , where $v$ shows total number of vertices and $\eta$ indicates total number of edges. This is a basic contribution with respect to the alternative algorithm proposed in [1], leading to a weighted graph based on the actual sensitivities among the system variables. The partitioning algorithm will then consider the different strengths, linking the system variables in the constructed graph.

## 2.2 Main algorithm

The graph partitioning algorithm is based on the work proposed in [1], where a multistep, iterative method is used to accomplish the task. The algorithm is constructed in a modular basis, incorporating some main routines which are implemented according to the standard graph partitioning

[1] $\dfrac{\partial x_i}{\partial u_j} = \dfrac{\partial x_i / \partial t}{\partial u_j / \partial t}\, i \in \{1,...,n\},\, j \in \{1,...,m\}$

[2] $\dfrac{\partial x_i}{\partial x_j} = \dfrac{\partial x_i / \partial t}{\partial x_j / \partial t}\, i \in \{1,...,n\},\, j \in \{1,...,n\},\, j \neq i$

[3] $\dfrac{\partial y_i}{\partial x_j} = \dfrac{\partial y_i / \partial t}{\partial x_j / \partial t}, i \in \{1,...,r\},\, j \in \{1,...,n\}$

problem [10]. In the following, a brief description of the implemented modular algorithm is presented.

## 2.3 Groundwork partitioning step

At the initial step, the algorithm traverses the entire graph to find all vertices of the shared edges and their corresponding weights. Then, the results are stored for further uses. In this context, all direct descendants (e.g., state-to- state or state-to-output connections) and ancestors (e.g., input-to-state or state-to-state connections), corresponding to a specified vertex in the graph, are recorded. Total values of each individual vertices are defined as vertex's order $\omega^i$ where $i = \{1, 2, ..., v\}$. The $\omega^i$ is calculated as follow:

$$\omega^i = \sum_{j=1}^{\eta} w_{ij} \text{ and } i = \{1, 2, ..., v\} \qquad (2)$$

Preliminary portioning

In this step, the algorithm chooses the most valued vertex to fix it as the center of the first sub-graph. Then, all descendants and ancestors, representing all vertices with at least one shared edges with the center vertices, are chosen to be a member of the first sub-graph.

After removing the selected vertices, the process is repeated for the remaining vertices and the other most valued vertex is defined as the center of the next sub-graph. Total number of vertices in each sub-graph is defined as the internal weight of the corresponding sub-graph [1]. In addition, some other information of the sub-graphs is used in the other steps which are defined as follow:

$$\bar{\phi} = \frac{1}{k} \sum_{i=1}^{k} \phi_i \qquad (3)$$

Yielding the average of sub-graphs internal weight.

$\phi_i$ represents the total number of vertices in sub-graph $i$ and $i \in \{1, 2, ..., k\}$. $\phi_i$ is called internal weight.

## 2.4 Internal balance

After assigning each vertex to a certain sub-graph, the next step is to balance the generated sub-graphs in order to have similar internal weight. To achieve this goal, all the $m$ sub-graphs with internal weights less than the calculated average value via the algorithm and all $h_j$ neighbor sub-graphs are specified. Then, for those in which condition (3) is met, the two sub-graphs are combined; leading to generation of a new sub-graph with larger internal weight.

$$\phi_i + \phi_j \le \bar{\phi} \qquad (4)$$

Where

$i \in \{1, 2, ..., m\}$ and $j \in \{1, 2, ..., h_j\}$

For sub-graphs in which the joinable neighbors are more than one, the one with smaller internal weight is selected. After each combination, all the parameter and sub-graph structural information like $\phi_i$, $\bar{\phi}$, … are updated, accordingly. The merging procedure is repeated until no other merging is obtainable.

## 2.5 External balance

In this step, the graph partitioning algorithm tries to reduce the cut size of the generated sub-graphs which is the total number of the weighted cut between the sub-graphs. some further information from subgraph vertices and edges are needed to be extracted in order to move toward the minimal cut size. Therefore, internal and external degrees are defined for each vertex in each sub-graph [1].

$\vec{\omega}_i^j$ denotes total value of the shared edges of vertex $j$ in the sub-graph $i$ with all other vertices in this sub-graph with $j \in \{1, 2, ..., \phi_i\}$ and $i \in \{1, 2, ..., k\}$. $\overleftarrow{\omega}_i^j$ indicates total value of the shared edges of vertex $j$ in sub-graph $i$ with all other vertices in the other sub-graphs.

After defining the mentioned weights, the algorithm starts to search for finding vertices with condition $\vec{\omega}_i^j > \overleftarrow{\omega}_i^j$. The entrapped vertices then are moved from the origin sub-graph to the sub-graph which has maximum total weighted cut size. After each movement, all the indexes must be updated.

This course of action is repeated until no other movement is possible and all vertices in all sub-graphs satisfy the condition $\vec{\omega}_i^j > \overleftarrow{\omega}_i^j$.

Once the foregoing graph partitioning algorithm is terminated, the aimed partitioned sub-graphs are determined and the corresponding decomposed subsystems will then be ready to be employed in the decentralized control scheme. Each subsystem contains a set of special state, input and output variables, recommended by the obtained groupings. On this basis, the local subsystem models can be derived for the design of the corresponding decentralized MPC controllers. For this purpose, nonlinear model predictive control scheme [13] is utilized based on the local models and measurements configured from the proposed partitioning algorithm, leading to a fully decentralized control design methodology.

## 3. SIMULATIONS

An industrial process plant can pose a challenging benchmark to exercise the proposed partitioning algorithm in a NMPC control framework. For this purpose, a nonlinear chemical process plant, consisting of a cascade of two Continuous Stirred Tank Reactors (CSTRs) together with a flash separator, having a recycle, has been selected [3]. A simplified schematic of the plant is shown in Figure (1). The plant has a set of 12 states (X), a set of 6 inputs (U), and 6 outputs, represented by the following vectors:

$$X = [H_r, T_r, X_{Ar}, X_{Br}, H_m, T_m, X_{Am} \cdots$$
$$, X_{Bm}, H_b, T_b, X_{Am}, X_{Bm}]^T \in \Re^{12}$$
$$U = [Q_r, Q_m, Q_b, F_0, F_1, D]^T \in \Re^6$$
$$Y = [H_r, T_r, H_m, T_m, H_b, T_b] \in \Re^6$$

First, the proposed system partitioning algorithm is exercised to the process plant, yielding an incidence matrix denoted by $I_M$. The corresponding graph has been illustrated in Figure (2).

The partitioning algorithm determines how to decompose the unified model of process plant. In other words, the algorithm categorizes the appropriate groups of process states, inputs and outputs for the aimed decentralized control system configuration. The grouping information is then moved to the accommodated decentralized nonlinear model predictive controller (NMPC) design algorithm.

A set of test scenarios is conducted to clearly evaluate both reference tracking and disturbance rejection objectives. Performances of the designed decentralized NMPC controllers are then compared with alternative centralized NMPC controllers. The obtained output responses have been illustrated in Figures (3) and (4). The corresponding control inputs are authentic because we put some constraints both on their value and rate. The constraints on the value of control signal prevent actuators from saturation and constraints on control signal rate cause smooth control. The plant state variables are set to some arbitrary initial values different from the final reference targeted values. Further test scenarios have been conducted to examine the disturbance rejection performances of both centralized and decentralized NMPC controllers, being exposed to relatively large disturbances which have been introduced at time instants of 250 and 265 sec. The obtained responses are clearly demonstrating the successful performances of both tracking and disturbance rejection missions of the decentralized NMPC controllers on the basis of the decentralized subsystems, being determined by the proposed partitioning algorithm. The obtained results verify that the designed decentralized NMPC controllers can duplicate the optimal centralized NMPC controller results with negligible deviations.

The resulting optimization cost functions, relevant to the both conducted test scenarios cases, in Figure (5); provide some useful information to comparatively judge about the implemented decentralized NMPC design procedures in comparison with the optimal centralized NMPC design scheme.

## 4. CONCLUSIONS

The paper addresses a demanding problem issue in large scale control systems. An efficient graph-theory-based algorithm has been proposed in to automatically partition large-scale systems into subsystems to facilitate the attractive decentralized MPC control strategy. The algorithm incorporates two basic contributed provisions to better conform to real large-scale systems. The algorithm is not restricted to linear large-scale systems. Furthermore, it incorporates a novel scheme based upon the sensitivities of the ending graph vertices, to quantify the weights which indicate the coupling among the system states or inputs, leading to a new weighted graph topology with different weights on the obtained graph edges.

The graph partitioning algorithm employs the obtained weighted system graph and tries to evolve some sub-graphs with almost equal number of vertices and minimal number of weighted interconnecting edges. For this purpose, the algorithm searches for some optimal cuts which result into sub-graphs that share edges with minimal total number and minimal total weights.

The decentralized NMPC control framework has then been incorporated in the large-scale control system design procedure on the basis of the decomposed subsystems to locally generate the optimal control policies. Comparative evaluations of the conducted test scenario results with optimal centralized NMPC in a process plant benchmark demonstrated the successful performance of the integrated design procedure in the context of the proposed partitioning algorithm.

Further researches have been motivated to enhance the proposed partitioning algorithm for more secure and stable control performances in the face of different system malfunctions.

## 5. REFERENCE

[1] C. Ocampo-Martinez, S. Bovo, V. Puig, “Partitioning approach oriented to the decentralised predictive control of large-scale systems”, Journal of process control, 2011.

[2] Y. Guo, D. Hill, and Y. Wang, ”Nonlinear Decentralized Control of Large-Scale Power Systems”, TECHNICAL REPORT: EE-98020, Electrical and Information Engineering School The University of Sydney, NSW 2006, Australia.

[3] A. N. Venkat. “Distributed Model Predictive Control: Theory and Applications”, PhD thesis, University of Wisconsin Madison, October 2006. URL http://jbrwww.che.wisc.edu/theses/venkat.pdf.

[4] S. Oschs, S. Engell, and A. Draeger, ”Decentralized vs. Model Predictive Control of an Industrial Glass Tube Manufacturing Process”, Proc. 1998 IEEE Int. Conf. Control Applications, Trieste, Italy, pp. 16-20.

[5] R. Scattolini, “Architectures for distributed and hierarchical Model Predictive Control – A review”, Journal of Process Control, 2009, pp. 723–731.

[6] S. Xu J. Bao, “Distributed control of plantwide chemical processes”, Journal of Process Control, 2009, pp. 1671–1687.

[7] M.B., Jamoom, E., Feron, and M.W. McConley, ”Optimal Distributed Actuator Control Grouping Schemes”, Proc. 37th IEEE Conf. on Decision and Control, Dec. 1998, pp. 1900-1905.

[8] N. Motee and B. Sayyar-Rodsari. “Optimal partitioning in distributed model predictive control”, In Proceedings of the American Control Conference, Denver,Colorado, June 2003, pp. 5300–5305.

[9] C. Ocampo-Martinez, V. Fambrini, D. Barcelli, V. Puig, “Model predictive control of drinking water networks: A hierarchical and decentralized approach”, in: Proceedings of the American Control Conference, Baltimore (USA), 2010.

[10] P. Fjallstrom, “Algorithms for graph partitioning: A survey”, Linkoping Electronic Articles in Computer and Information Science 3 (10).

[11] J. Bondy, U. Murty, Graph Theory, Vol. 244 of Graduate Series in Mathematics, Springer, 2008.

[12] D. ˇSiljak, Decentralized control of complex systems, Academic Press, 1991.

[13] L.Grüne, J. Pannek, Nonlinear Model Predictive control, Springer,2011.

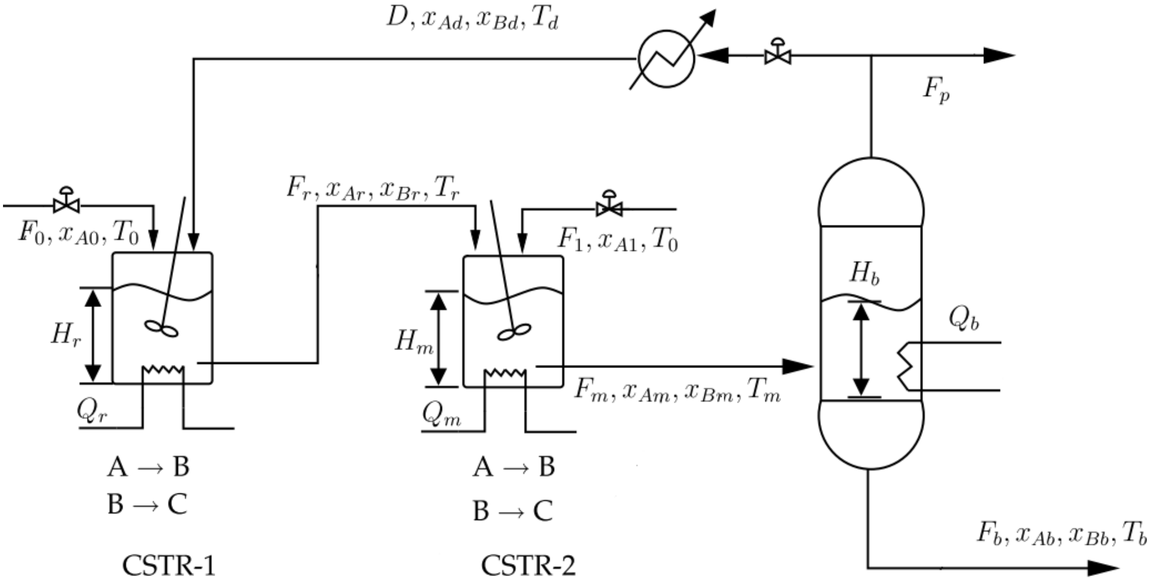

**Figure 1. Two-reactor chains followed by a flash separator with recycle.**

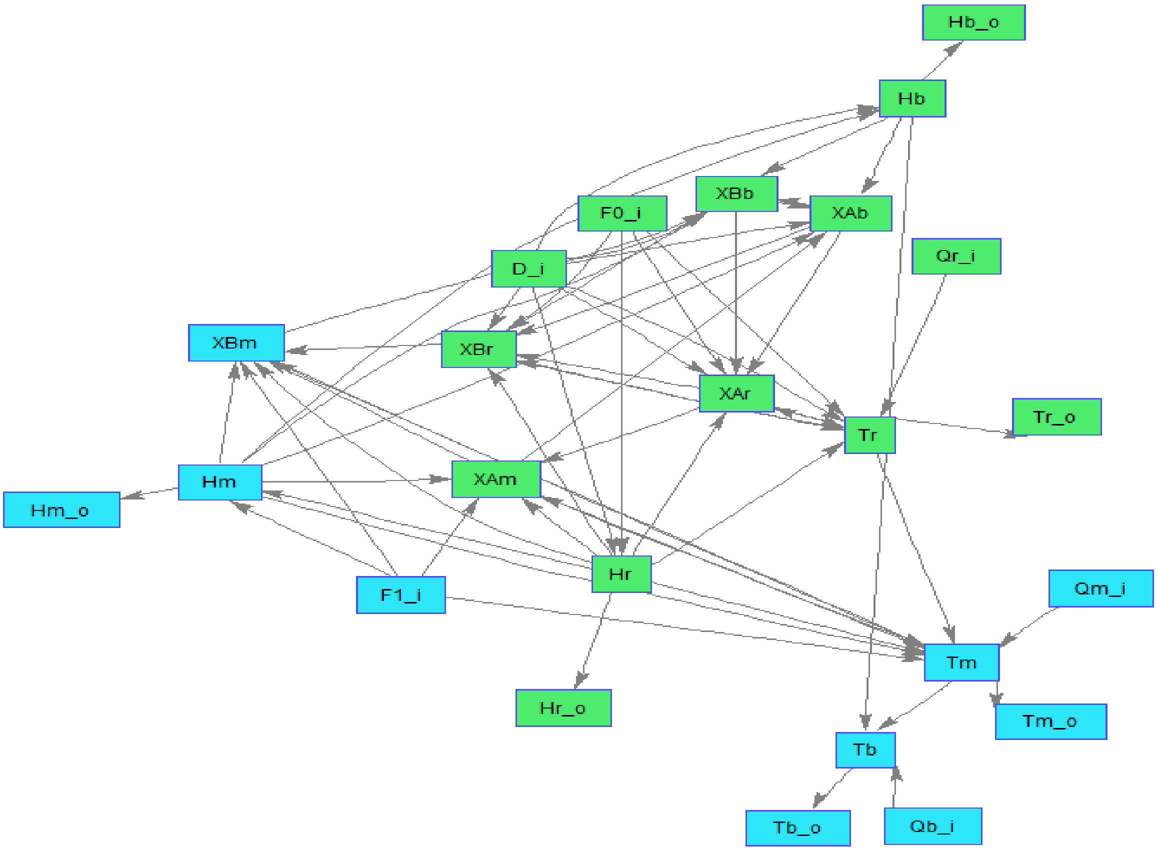

**Figure 2. Schematic diagram of the partitioned graph and the corresponding sub-graphs of the simulated process plant.**

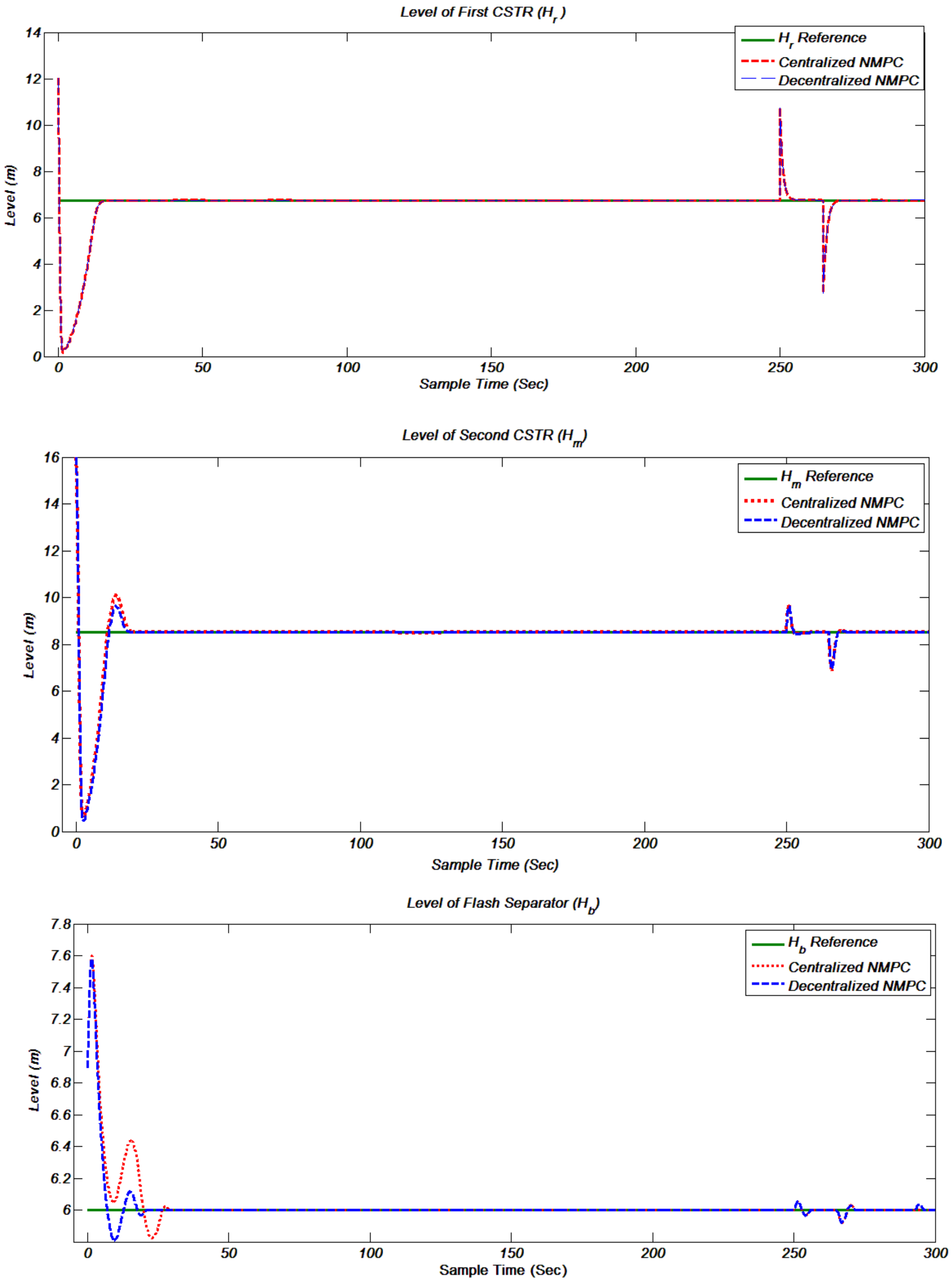

**Figure 3. Setpoint tracking and distrurbance rejection performances of Centralized and Decentralized NMPC for 3 Level state variables (Hr, Hm, Hb).**

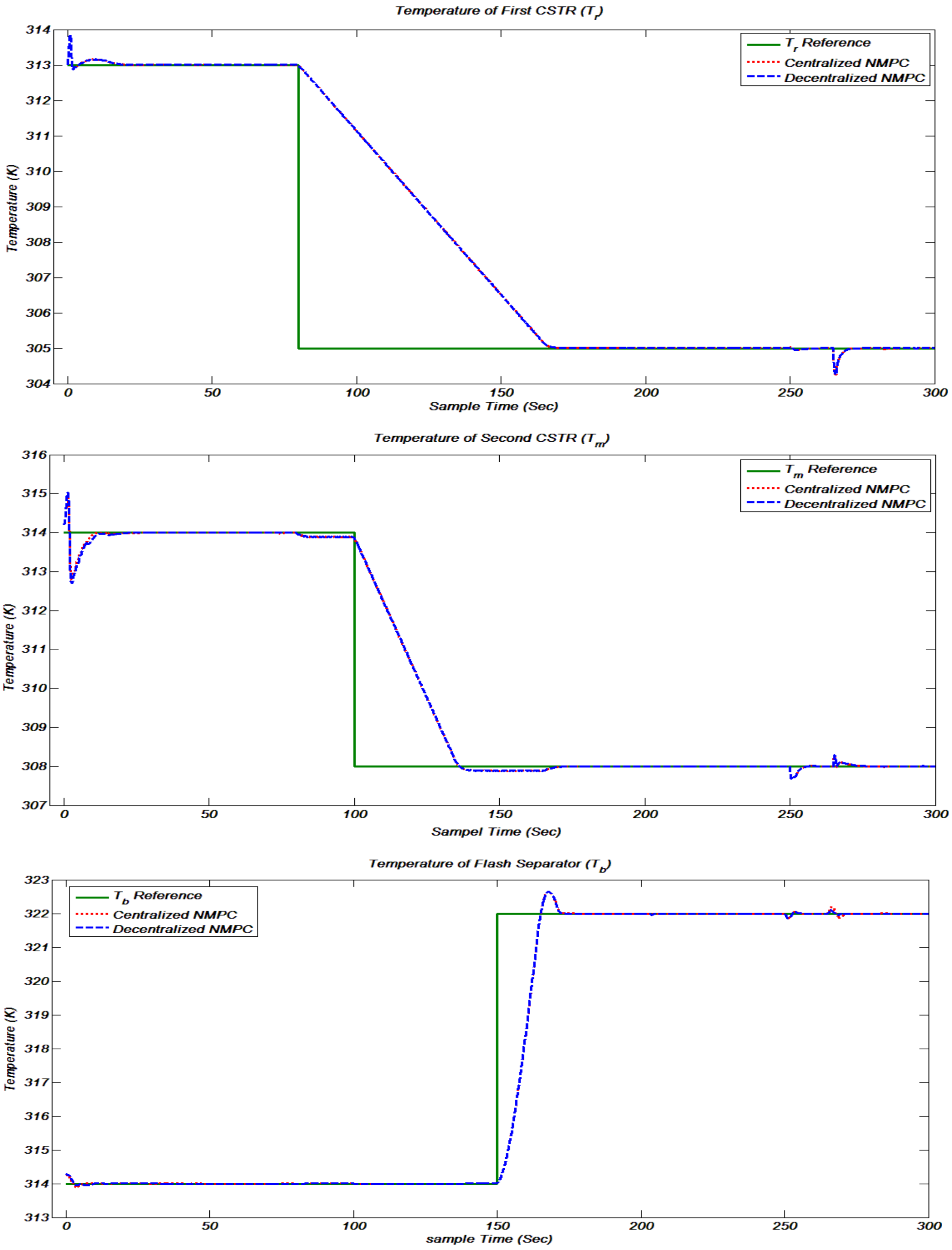

**Figure 4. Setpoint tracking and distrurbance rejection performances of Centralized and Decentralized NMPC for 3 Temperature state variables (Tr,Tm,Tb).**

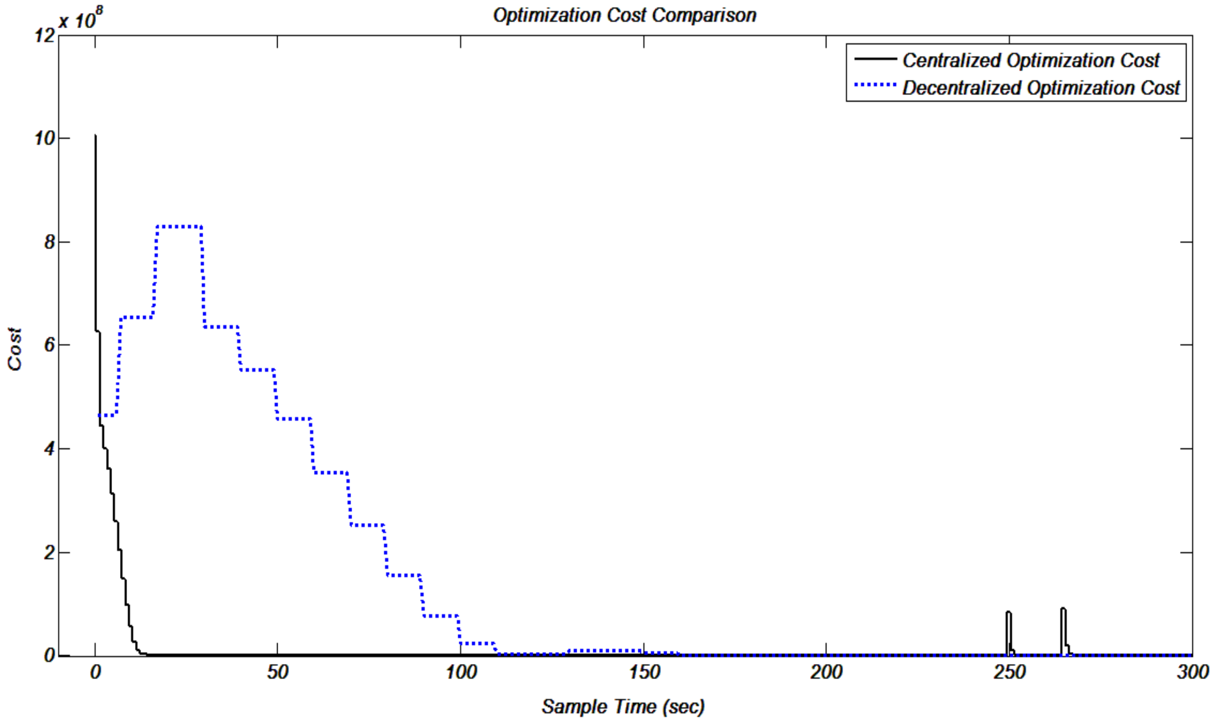

**Figure 5. Optimized cost of control schemes**