An Agent-Based Model-Predictive Controller for Chilled Water Plants using Wireless Sensor and Actuator Networks

Michael B. Kane, Jerome P. Lynch, *Member, IEEE*

***Abstract*—Reduced manning requirements and other cost reducing measures have spurred interest in automation of engineering plants onboard naval combat vessels. Furthermore, automation may increase resiliency of shipboard engineering plants when compared to the current generation of manually configured plants. This paper presents a control algorithm and deployment strategy which supervises a ship's chilled water plant to control the temperature of thermal loads (e.g., air chillers, electrical components). Redundant computation and communication capabilities motivates the use of an agent-based controller (ABC) enabled through a peer-to-peer wireless network. In the architecture presented, each agent sits on one or more digraphs corresponding to the utility generated by the fluid exiting the chilled water plant at each discharge point. Each digraph is a component of the decentralized model-predictive controller (MPC). Performance of the proposed control architecture is tested in simulation, and is shown to approach the performance of an effective, but computationally exhaustive, centralized MPC.**

I. Introduction

Future naval vessels may be rendered more resilient through the U.S. Navy's explorations of the all-electric ship (AES) design concept. In the AES, an integrated power system combines a ship's two largest plants, the electrical system and powertrain, coupling previously independent engineering plants [1]. This paper considers a chilled water plant and electrical system coupled through the operation of pumps, valves, sensors, and electrical thermal loads (e.g., radar and pulsed weapons systems) [2]. The U.S. Navy is interested in automating these systems so as to reduce manning requirements on ships without sacrificing fight-through capabilities [3]. Effective automated reconfiguration of the interconnected ship plants will require dense arrays of sensors and actuators [4]. Building a layer of computational intelligence atop this network will facilitate automated plant reconfiguration in the face of battle damage.

Successful operation of a large-scale networked control system is critically dependent upon the communication pathways' ability to deliver information from the sensors to the controller and from the controller to the actuators in real-time. The wired communication systems used today, mainly copper wiring and fiber optics, have long been the viable options for these communication systems on ships [5]. Recent developments in wireless technology and standards threaten to shift this paradigm. The downfalls of the wired paradigm might include the complexity and cost of installation through bulkheads and tight spaces, repair difficulty after inevitable damage from long-term deterioration or battle, and the rising cost of commodity copper. It has been estimated that a fifty percent reduction in cost can be seen with the adoption of the wireless communication paradigm [3] while maintaining security with encrypted communication [6]. The difficulty of transmitting wireless data in the metallic chambers of ship hulls was shown to be a surmountable challenge in recent research for both control [3] and monitoring systems [7], [8].

The paradigm shift from wired to wirelessly networked control systems entails more than a one-to-one replacement of communication links. The centralized wired control system designer limits the number of sensors and actuators in the system in an effort to reduce the computation burden on the centralized controller. Achieving redundancy by duplication of all the wiring and computing further reduces cost effectiveness. Conversely the architecture proposed herein empowers the control designer with the ability to implement dense arrays of sensors and actuators which would otherwise overwhelm centralized wired control schemes [4]. The sharing of the computational load amongst all the agents in the peer-to-peer network maintains computational and communication redundancy.

The distributed controller presented herein was fundamentally designed to work on a network of wireless nodes such as *Narada* [9]. Specifically, this paper presents a model predictive controller (MPC) that has been decentralized into two coupled MPCs; each collocated with a pump. The hierarchy in which the sensing, computation, and actuation are decentralized provides redundant communication pathways to provide resiliency in case of system damage and/or node failure. Plant and controller irregularities can be detected by system identification and fault detection software also distributed across the network in the form of wireless nodes with microcontrollers [10].

MPC, or receding horizon control, is well suited for controlling nonlinear plants with relatively easily modeled nonlinear dynamics and actuator constraints. At every

Manuscript received September 23, 2011. This work was supported in part by the US Office of Naval Research (Contracts N00014-05-1-0596 and N00014-09-C0103 granted to Jerome P. Lynch).

M. B. Kane is a Ph.D. candidate with the University of Michigan, Department of Civil Engineering, Ann Arbor, MI 48109-2125 USA. (e-mail: mbkane@umich.edu).

J. P. Lynch is an Associate Professor with the University of Michigan, Department of Civil and Environmental Engineering and the Department of Electrical Engineering and Computer Science, Ann Arbor, MI 48109-2125 USA. (phone: 734-615-5290; fax: 734-764-4292; e-mail: jerlynch@umich.edu).

control instant, an MPC utilizes a system model initiated with the current state to select a permissible open-loop control action that maximizes the utility of that action. Due to the extensive use of MPC in industry, the theory has been thoroughly studied in academia giving rise to formal assurances of robustness and stability on for certain controller architectures and plant types [11], [12]. The idea of distributing an optimization problem over a web of agents, i.e., agent-based control (ABC), has been studied theoretically [13] and for a wide variety of problem types from environmental building monitoring and control, to naval applications [14–16]. This paper presents a decentralized model predictive controller (DMPC) for physical systems in which the flow of utility can be represented as a collection of directed trees, e.g., chilled water plants. The sub-optimality of the proposed controller was carefully balanced with the implications of using an IEEE 802.15.4 network for agent communication.

II. Chilled Water Plant Demonstrator System

Engineering plants that are essential to a ship's operation often produce waste heat and must maintain temperatures below critical levels preventing premature degradation or dangerous scenarios. Cooling is accomplished through redundant chilled water loops in which heat is removed and directed through a pipe network to a heat exchanger where the heat is released to the sea. In need of a test bed to verify decentralized chilled water plant controller performance, a simplified bench-scale model, the University of Michigan Chilled Water Demonstrator (UMCWD), Fig. 1, was designed and fabricated based on the larger demonstrator housed at the Naval Surface Warfare Center [17].

Fig. 2 shows the schematic of the demonstrator with two 12 volt Graylor PQ-12 DC pumps drawing water from the chilled water reservoir and forcing it through open/closed STC 2W025-1/4 solenoid valves. The valve and pump configurations determine the coupled flow rates, measured by DigiFlow DFS-2W flow meters, through pipes 1A, 1B, 2, 3A, 3B, and 4 which provide chilled water to the thermally coupled T1-T2 and T3-T4 thermal load pairs. The coupling between the electrical and chilled water plant is achieved using 70 Watt resistive heaters affixed beneath each 10 x 5 x 2.5 cm^3 aluminum block; two 1.25 cm diameter holes drilled

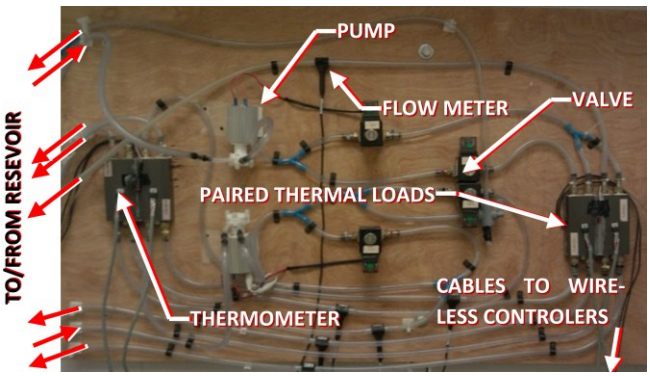

Fig. 1. The University of Michigan Chilled Water Plant Demonstrator

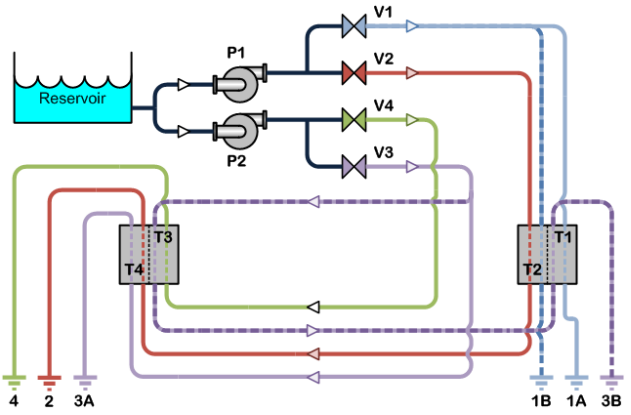

Fig. 2. Fluid flow schematic of the University of Michigan Chilled Water Plant Demonstrator

have been drilled along the length of each block through which chilled water flows. As the heaters disturb the system, causing the thermal load temperatures to reach critical, T_c, or danger, T_d, temperature thresholds, the proposed wirelessly networked control system determines appropriate pump and valve configurations to mitigate the effect of heating while minimizing system energy use. Actuators (two pumps and four valves) and sensors (four block thermometers, an air thermometer, a reservoir thermometer, and a flow meter for each of the six pipes), each have a collocated wireless node for data processing, communications, and execution of the control algorithm.

III. Plant Dynamics

The dynamics of thermal loads in a chilled water plant are adequately modeled by forced convective heat transfer. The rate of heat removed from component i, $\dot{Q}_{w_{i,j}}$, by chilled water through pipe j, is a linear function of the difference in temperature between the component, T_{b_i}, and the chilled water, T_{w_j}, when the area of contact, $A_{i,j}$, and the heat transfer coefficient, $h_{i,j}$, between the body and the heat sink (i.e., water) are held constant. The relationship between block temperature and flow rate q_j is nonlinear as seen in (1). The empirically derived exponential function (2) characterizes well the relationship between a flow q_j and the heat transfer coefficient. The coefficients in (2) were identified as $\alpha = 1.36$ and $\tau = 0.49$ for the UMCWD system.

$$\dot{Q}_{w_{i,j}} = h_{i,j}(q_j) \cdot A_{i,j} \cdot \left(T_{b_i} - T_{w_j}\right) \tag{1}$$

$$h_{i,j}(q_j) = \alpha \cdot (q_j)^{\tau} \tag{2}$$

$$\dot{Q}_{a_i} = h_i \cdot A_i \cdot \left(T_{b_i} - T_a\right) \tag{3}$$

$$\dot{Q}_{c_{i,l}} = h_{i,l} \cdot A_{i,l} \cdot \left(T_{b_i} - T_{b_l}\right) \tag{4}$$

$$m_i c_p \dot{T}_{b_i} = \dot{Q}_{a_i} + \dot{Q}_{c_i} + \sum_{j=0}^{4} \dot{Q}_{w_{i,j}} + \dot{Q}_{h_i} \tag{5}$$

The conservation of energy equation, (5), is used to compute the temperature of block i with mass, m_i. Essentially, it is a summation of the heat flow rates from: each pipe through a given block; heat loss to the air, $\dot{Q}_{a_i}$ by

(3), with temperature T_a; the heat exchange between coupled block pairs i and l, $\dot{Q}_{c_{i,l}}$ by (4), with block temperatures T_{bi} and T_{bl}; and the heat input $\dot{Q}_{h_i}$.

Additional nonlinearities are introduced by saturation of pump speeds and valve authority. Although the system has these complex dynamics and actuator constraints, if model parameters can be estimated, either through system identification or derived theoretically, simulation of a given open loop control trajectory can be computed using numerical integration techniques for hybrid systems [18] assuming steady state flow rates through the pipe network. The switching speed of the valves and pumps, and thus flow rates, is significantly faster than the thermal dynamics being controlled which allow the system to be modeled as a switched linear system with constant pump speeds, valve configurations, and flow rates between switches. System identification of the UMCWD was performed using a series of forcing functions that isolated the different model parameters enabling the use of linear least-squared parameter estimation methods.

IV. Model Predictive Control

The MPCs used herein work by initializing a numerical integration with the current state of the system which is measured by temperature sensors on each block, and optimize the estimated utility generated by a constant open-loop control configuration from the current instant until the end of a defined time horizon. A suitable controller should efficiently compute the solution to the optimization problem such that time between re-optimizations is sufficiently short.

A. Benchmark Centralized MPC

A centralized MPC (CMPC) aggregates all the sensor data into a single processing unit at each time step where the control algorithm is executed. Once a control decision has been made, the centralized computer disseminates the prescribed actions to each actuator. An implementation of a CMPC for a chilled water plant might, at each control instant t, measure the system state (i.e., component temperatures), estimate point source heat inputs (i.e., thermal load rates), and then search through the predicted utilities from all possible pump speeds and valve configurations to find the highest utility assuming the heat inputs and the selected control decision remain constant until the end of the prediction horizon $t+K$ (i.e., constant future CMPC). The system-wide utility achieved at each estimation step, k, is the sum of the utility lost by each pump, the utility lost by each valve, and the utility gained by each thermal load when cooled by the fluid passing through its pipes, each weighted by a utility conversion coefficient, γ_p, γ_v, and γ_t respectively. Utility of an action, interpreted herein as the measure of relative satisfaction of a given action, is computed as the cost of doing nothing subtracted from the cost of that action.

The estimated thermal cost, $\widehat{CT}_i^k\big|_{q_j}$, for a given block i, at a step k along the prediction horizon, (and with respect to flow q_j through pipe j) is computed according to (6) as a function of the predicted block temperature at that instant, $\hat{T}_{b_i}^k$, and a pair of user defined threshold temperatures T_c and T_d. These temperatures set quadratic and quartic costs respectively when the block temperatures exceed the critical and dangerous temperature thresholds. This tuple should be specified by the control engineer to ensure safe and proper equipment operation. The thermal utility, $\widehat{UT}_i^t\big|_{q_j}$, at time t with respect to flow q_j through pipe j, which is constant along the entire prediction horizon, $t...t+K$, is computed by (7) and is equal to the cost of zero flow along the horizon subtracted from the cost when the flow through pipe j is equal to q_j.

$$\widehat{CT}_i^k\big|_{q_j} = \begin{cases} 0, & \hat{T}_{b_i}^k\big|_{q_j} < T_c \\ \left(T_c - \hat{T}_{b_i}^k\big|_{q_j}\right)^2, & T_c \le \hat{T}_{b_i}^k\big|_{q_j} < T_d \\ \left(T_c - \hat{T}_{b_i}^k\big|_{q_j}\right)^2 + \left(T_d - \hat{T}_{b_i}^k\big|_{q_j}\right)^4, & \hat{T}_{b_i}^k\big|_{q_j} \ge T_d \end{cases} \tag{6}$$

$$\widehat{UT}_i^t\big|_{q_j} = \sum_{k=t}^{t+K}\left(\widehat{CT}_i^k\big|_0 - \widehat{CT}_i^k\big|_{q_j}\right) \tag{7}$$

The estimated cost, $\widehat{CP}_i^k\big|_{P_i}$, of running pump i at a given speed $P_i^k \in [0,100]$ is assumed to be tied to the cost of electricity used to power the pump. From the system identification performed on the demonstrator, the power curve for the pumps is characterized by a cubic function (8) where the coefficients were identified as: $a=1.38x10^{-5}$, $b=6.23x10^{-4}$, $c=2.54x10^{-2}$. Since zero energy is consumed when not running the pump, the utility of a given pump speed, $\widehat{UP}_i^k\big|_{P_i} = \widehat{CP}_i^k\big|_{P_i}$, is equal to the cost of running the pump at that speed.

$$\widehat{CP}_i^k\big|_{P_i} = a\cdot\left(P_i^k\right)^3 + b\cdot\left(P_i^k\right)^2 + c\cdot P_i^k \tag{8}$$

Typical valves used in naval chilled water plants are turned open and closed by a seaman, either manually or with a motor controller. As such, the controller should derive the utility from the real-life valve operation and incurs a cost, $\widehat{CV}_i^k\big|_{V_i}$, computed by (9) that is nonzero only when the position, V_i^k, of valve i changes from the previous step, $k-1$. Similar to the pump, the valve utility $\widehat{UV}_i^k\big|_{V_i} = \widehat{CV}_i^k\big|_{V_i}$ is equal to the valve cost at each instant.

$$\widehat{CV}_i^k\big|_{V_i} = \begin{cases} 0 & V_i^k = V_i^{k-1} \\ 1 & V_i^k \ne V_i^{k-1} \end{cases} \tag{9}$$

The implausibility of searching through all system configurations for an optimal open-loop control in real time lead the authors to consider the decentralization of the MPC across a wireless network. However, for the simplified bench-top demonstrator studied herein, a CMPC serves as an excellent benchmark for the performance of the

decentralized MPC to be proposed. The benchmark CMPC computes the estimated utility, given an estimate of the heat input to each block and measured block temperatures, for 961 realizable configurations with 10 discrete pump speeds. The configuration with the greatest predicted utility is applied until the next control instant. Through the specification of different estimation horizons and utility conversion rates, the CMPC can be tuned to prevent temperature overshoot, reduce valve switching, or encourage greater energy reduction.

B. Decentralized Model Predictive Control (DMPC)

Even for simplified chilled water plants, such as the UMCWD, a centralized exhaustive search CMPC must sample slowly even on a modern desktop PC. MPC may still be plausible if partially decoupled, distributing the computation load across a network of computing devices. Graphing of the utility generation pathways through each of the six effluent pipes was used as the template for the UMCWD MPC decomposition. In Fig. 3, the cooling capability of each unique pipe 1a, 1b, 2, 3a, 3b, and 4 is a directed acyclic graph. These digraphs serve as the communication topology for the agent-based DMPC proposed. Serially, the flow versus utility curve in each of the digraphs is computed assuming the flow through all the other pipes remains constant. The dependence with respect to flows through pipe combinations necessitates iteration of the serial optimization process to minimize the difference between realizable and desired flows. Under certain conditions (e.g. convex agent costs and linear constraints) convergence of all the variables shared between components of a DMPC will lead to a control decision with the same cost as an equivalent CMPC [19]. These conditions are too strict for the specific problems studied in this paper, but as Camponogara, *et al*. noted, the overly restrictive conditions of the equality between DMPC and CMPC may be relaxed, yet nearly yield equality in experimental studies [19].

1) Control-step Timeline

Four different types of agents (flow, valve, pump, and thermal) collaborate at each control step to complete the distributed optimization of global utility. The optimization problem is completed by an iterative process of local optimization and communication of utility functions along the graph depicted in Fig. 3. Computation and communication in each control step is outlined below:

(1) Agents send and receive private information (i.e., current state) with peers on the same pipe.
(2) A flow agent, *Fi*, begins the optimization process. The chosen flow agent *Fi* asks the thermal agent, *Tj* directly below in the directed acyclic graph *Gi* of pipe *i*, for the predicted utility gained by the system as a function of flow through pipe *i*.
(3) Since thermal agent *Tj* can only compute its own contribution to the graph's thermal utility agent *Tj* must communicate with the next agent *Tk* down graph *Gi* and sum the two utility functions. If agent *Tk* is not a leaf node, *Tk* continues the cascade of communication down and back up graph *Gi* until the sum of all the agents' predicted utilities is received by agent *Tj*.
(4) Once agent *Fi* has the graph's predicted utility, it passes this information up to the valve agent *Vm* above agent *Fi* on graph *Gi*. Agent *Vm* combines the updated predicted utility with earlier obtained utility curves of all other graphs passing through *Vm* to formulate the systems utility for the flow through valve *m*.
(5) Agent *Vm* sends its updated utility curve to its root agent, pump *n*. Root agent *Pn* combines the earlier obtained utility curves of all other graphs passing through *Pn* to formulate the systems estimated thermal utility for the flow through pump *n*. With knowledge of the system's predicted thermal utility for its flow, agent *Pn* is able to choose a pump speed and request an action for its children that balance the predicted (non-negative) utility obtained by the thermal agents with the (non-positive) cost of running the pump and changing valve positions.
(6) Without applying these actions, agent *Pn* begins the cascade of these requested actions back down all graphs that pass through it until all affected agents are informed of the change in the pump's requested action.
(7) The updated requested system state is now used as the new starting point for the next flow agent.
(8) Steps 2 through 7 are repeated until all pipes have been selected at least once and there is an insignificant change in requested actions between iterations. This agreed upon system state is then applied and the agents idle until the beginning of the next control step, at which point the whole process is repeated.

Following the communication and agent topology outlined above and in Fig. 3 the agents aim to agree upon an action that maximizes the systems predicted utility. The four different types of agents, defined by their unique collocated resources, are described in detail below.

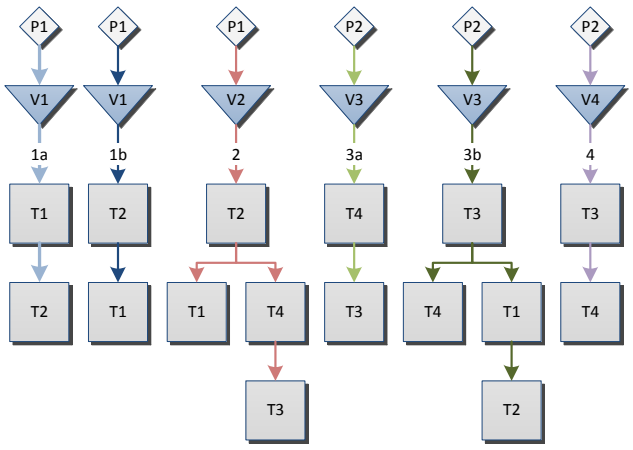

Fig. 3. Unbraided di-graphs of the utility generating components of the flow through each return pipe on the UMCWD.

2) Thermal Agents

The leaf agents, *T1…T4*, at the end of each branch in Fig. 3 each represent the thermal utility of a single block. At the beginning of each control instant, each thermal agent receives limited amounts of its neighbors' private information (i.e., current temperature) which enables the agent to adequately predict thermal utility of all relevant flows. This information consists of the current temperature and estimated thermal load of the adjacent block, the ambient air temperature, and the temperature of the water into each of the adjacent block's orifices. With this shared information and the agent's private information, the agent can model the thermal utility of the flow through any of its own or its neighbors orifices. The model used by some thermal agents (e.g., *T1*) is only approximate due to the lack of information about how a block upstream (e.g., *T3*) might influence the temperature of the incoming water (e.g., through pipe *3b*). The agents assume the water temperature change is negligible over the prediction horizon in order to account for this lack of information. The utility predicted is computed using a two state discrete-time state space model linearized around the flow rates.

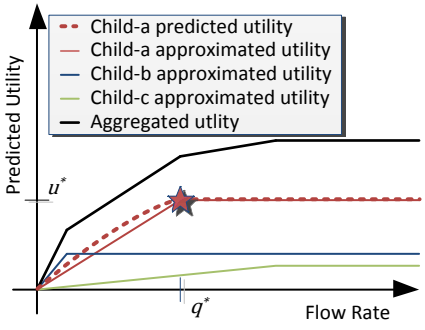

Fig. 4. General form of the exact and approximated agent thermal utility curves.

The general form of the computed flow rate versus predicted utility curve is shown in Fig. 4. To minimize bandwidth requirements the utility curves are approximated as shown in Fig. 4 as a bilinear function wholly described by a pair *(q*, u*)*. Knowing the general form, the agent's parent can decode the pair to reconstruct the approximated estimated utility curve.

3) Flow Agents

The flow agents, $F1_a$...*F4*, serve to represent each unique pipe. They are capable of measuring the flow rate in their respective pipe so that the pumps and valves may confirm the flow rates produced. The flow agents also communicate with their children to aggregate the predicted pipe utility versus flow rate. The approximated predicted utility curves, represented by a *(q*, u*)* pair for each agent, are summed into a piecewise linear curve. The general form of the aggregated predicted utility is presented in Fig. 4.

4) Valve Agents

Agents *V1…V4* are capable of actuating the valves, serve to aggregate the predicted utilities of all their children, and pass the aggregated utility to the pump agents. The aggregation of the utilities of the valve's children is similar to the aggregation done by the flow agents. However, the valve agents must consider downstream bifurcations. Bifurcations are accounted for using a steady state fluid model to determine the amount of flow and thus utility generated on each branch.

5) Pump Agents

Presented with two aggregated curves, one from each downstream valve, the pumps consider four arrangements to generate utility. One, the pump could turn off and generate zero flow and utility. Two, the pump directs both downstream valves to open and models the utility bifurcated to each valve. The intersection of the thermal utility curve (shifted down by γ_v for each valve that must change state) with the pumps utility curve identifies the optimal flow with respect to the graph being optimized. Similarly, the pump considers only one open valve which receives all the flow and utility generated. The pump's task is to identify which of these four configurations produces the greatest predicted utility.

V. Controller Design

The DMPC presented herein requires the specification of ten non-model-based parameters; only a subset of which are specified to each agent. The ten parameters required for a full controller design are the controller update time-step (Δt_{ctrl}), critical and danger temperatures (T_c, T_d), weights on the pump, valve, and thermal utilities (γ_p, γ_v, γ_t), the prediction horizon length (K), discrete possible flow rates ($\overrightarrow{q_{try}}$), possible pump speeds ($\overrightarrow{DC_{try}}$), and the number of iterations of the optimization process required to reasonably assure convergence (N). A longer Δt_{ctrl}, the only parameters specified to all agents, will give the system more time to either refine the optimization process or complete other computation tasks required by the network. Conversely, a shorter Δt_{ctrl} speeds up the controller's dynamics to better track changes in the system disturbances.

Thermal agents determine the utility received for each flow rate in ($\overrightarrow{q_{try}}$) by estimating the thermal cost incurred over a prediction horizon, $t+(1...K)\Delta t_{ctrl}$ and subtracting that cost from the thermal cost of zero flow. As the set ($\overrightarrow{q_{try}}$) increases, the thermal agent's approximated discretized utility curves will more closely match the actual continuous utility curves, at the cost of increased computation time. Similarly, an increase in K increases computation time, with the benefit of better prediction of the response of a given action, assuming that action is maintained indefinitely into the future. Excessive values of K may degrade performance due to poor prediction caused by unforeseen changes in thermal loading. The computational cost associated with changes in the number of elements, n, in ($\overrightarrow{q_{try}}$) is $O(K \cdot n)$. An increase in the critical and/or danger temperature will, *ceteris paribus*, decrease the amount of thermal utility received

The utility of the flow's thermal utility, valve costs, and pump costs are weighted by γ_t, γ_v, and γ_p respectively. As the number of elements, m, in the set ($\overrightarrow{DC_{try}}$) increases, the pump can more accurately maximize the estimated utility but at the cost of an increased computation time of $O(3^m)$ for two valve children.

While the ten parameters described above are the only parameters that must be selected by the control engineer, there are additional model-based parameters. These parameters, such as heat transfer coefficients and pipe friction coefficients, come from fundamental physics or system identification.

VI. Simulation Results

Efficacy of the controller is determined in two ways. Firstly the results of the DMPC of Section IV.B will be compared with the results of the CMPC of Section IV.A. The results of this comparison in which both controllers are realized with identical parameters will show how closely the DMPC can track the optimal result. Five cost metrics (i.e., thermal cost, pump costs, valve costs, run time, and communication requirements) make up the second method to assess the performance of the DMPC. The first three metrics are all similar to the parts of the utility function the controller is maximizing, while the run time and communication requirements reflect the computational complexity and implementation challenges. The run time cost metric presented as "percent real-time" is the number of seconds used by the processor, in this case a 3.2GHz Intel i5, for one control step, divided by the length of the control step. Note though, that the percent real-time metric is computed using a single CPU, while in the actual implementation of the DMPC, the computing tasks will be distributed across the network on microcontrollers. None the less, it gives a feel for the computation costs.

A. *Time History Comparison*

To emulate persistent loads (e.g., turbines) and intermittent loads (e.g., pulsed weapon systems) a loading pattern was developed to validate the performance of the controllers. Thermal loads *T1* and *T2* are classified as persistent 70 W loads that engage ten seconds into the test and remain on for the remainder of the 1000 second test. Thermal loads *T3* and *T4*, emulate a 70 W intermittent load that pulses on for 120 s starting at t=130 s and reengages at t=490s with a duration of 240 s.

The controllers used during benchmark analysis were: (1) an open-loop system with all valves open and pumps at 10%, and (2) the model-predictive agent-based and (3) centralized controllers with the following parameters: $T_w = 312\ K; T_d = 322\ K; \gamma_t = 0.05; \gamma_p = 0.95; \gamma_v = 0; K = 8; \Delta t_{ctrl} = 6s; N = 5$. The thermal trajectories graphed in Fig. 5 show similar system state paths for both the CMPC and DMPC solution. Both MPCs are able to minimize the amount of time spent within the critical thermal zone while consuming less pump power than the open-loop solution. The results from the five cost functions shown in Table I show the DMPC proposed in this paper is more computationally and communication intensive. However the increase in complexity is mitigated by distributing the computational load amongst all of the processors of the agents in the network. Also, the DMPC architecture is better suited for handling node failure and other computational tasks such as system identification and model updating.

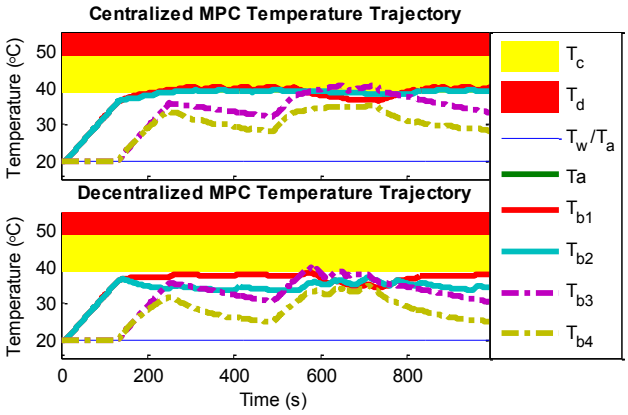

Fig. 5. DMPC versus CMPC simulation result

B. *Parametric study*

The same controller parameters used in the time history comparison were individually manipulated and the resulting controlled system simulated in an extensive parametric study. Reasonable perturbation in the parameters Δt_{ctrl}, T_c, T_d, γ_p, γ_v, γ_t, $\overrightarrow{q_{try}}$, and $\overrightarrow{DC_{try}}$ confirmed the relations predicted in Section V. Large variations in parameters lead to unpredictable results, as would be expected. For instance, once Δt_{ctrl} reached about 60 s the DMPC can no longer effectively control either the pump or thermal costs. Similarly, insufficiently short or long prediction horizons lead to poor controlled performance due to the DMPCs ability to accurately predict the full utility of a given action.

Strict convergence of the iterative optimization which is integral to the DMPC is not guaranteed. Both continuous and discrete states and actions, results in the iterations chattering between multiple decisions which all have similar utility. By studying Fig. 6, it is seen that the DMPC has total costs, $J_{tot} = (\gamma_T CT + \gamma_p CP)/1000$, similar to that of the centralized controller regardless of the number of iterations.

TABLE I
Costs of Constant Speed, ABC, and Centralized Controllers

Controller	$CT = \sum_{i=1}^{4}\sum_{t=0}^{1000} CT_i^t$	$CP = \sum_{t=0}^{1000}(CP_1^t + CP_2^t)$	#ΔV/second	% Real Time	Communications/Step
Constant speed (10%)	0.556	651	0.00401	1.3	0
Centralized	887	208	0.0300	192	48
Agent-Based Controller	9.22	254	0.0822	218	524

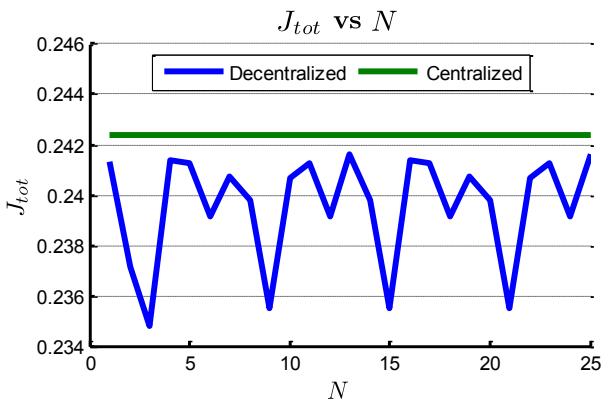

Fig. 6. Total cost versus ABC iterations

In the future, it may be possible to program intelligence into the DMPC such that it only iterates enough times necessary to converge to an appropriate solution in order to reduce computation time.

VII. Conclusions

The U.S. Navy's quest for reduced manning requirements may be aided by agent-based controllers such as the DMPC presented here. This DMPC is suboptimal, but costs and tuning parameters retain a physical sense of meaning and balance sub-optimality with computational and network bandwidth efficiency. The thermal cost function is more analogous to the actual penalty incurred by the thermal loads (i.e., critical and danger temperatures may be exceeded if necessary for brief periods of time) when compared to more traditional control requirements (e.g., quadratic costs or strict safe operating sets). Designed specifically for chilled water plant application, this DMPC may be generalized to control dynamic systems in which states are directly measured and the flow of utility can be mapped onto overlaying digraphs. The four classes of agents, embedded on wireless sensors and actuators, collaborate to achieve a closed loop control solution which approaches the effectiveness of a benchmark CMPC. The agent-based control architecture leaves open the option for installation of additional software modules on the agents performing online system identification or fault detection. The results of which can be used to remove failed agents from the system, without disproportionate performance loss (i.e., system wide performance loss is at worst proportional to the utility produced by the lost agent). Additionally, if changes to the flow model are identified (e.g., pipe rupture) the model used by the MPC may be updated to ensure predicted utility closely matches the utility achieved.

Acknowledgment

The authors would like to thank Mr. Frank Ferrese (Naval Surface Warfare Center Carderock Division) and Mr. Anthony Seman (Office of Naval Research) for their technical feedback and support.

References

[1] B. Wagner, "All-Electric Ship Could Begin to Take Shape By 2012," *National Defense*. National Defense Industrial Association.

[2] S. K. Srivastava et al., "A Control System Test Bed for Demonstration of Distributed Computational Intelligence Applied to Reconfiguring Heterogeneous Systems," in *Systems Conference, 2007 1st Annual IEEE*, 2007, pp. 1-8.

[3] A. J. Seman III, K. Toomey, and S. Lang, "Reduce's Ship's Crew-By Virtual Presence (RSVP) Advanced Technology Demonstration (ATD)," *Final technical rept. 1 Sep 1998-1 Oct 2001*. Naval Surface Warfare Center Carderock Division, Bethesda, MD, p. 353, 2003.

[4] E. L. Zivi, "Integrated shipboard power and automation control challenge problem," in *Power Engineering Society Summer Meeting, 2002 IEEE*, 2002, vol. 1, pp. 325-330 vol.1.

[5] L. Dunnington, H. Stevens, and G. Grater, "Integrated Engineering Plant for Future Naval Combatants-Technology Assessment and Demonstration Roadmap," Fairfax, VA, 2003.

[6] Texas Instruments, "Application Note AN060 Security on TI IEEE 802.15.4 Compliant RF Devices," *Security*. Dallas, Texas, United States, p. 31, 2009.

[7] R. A. Swartz et al., "Hybrid wireless hull monitoring system for naval combat vessels," *Structure and Infrastructure Engineering*, pp. 1-18, 2010.

[8] S. B. Slaughter, M. C. Cheung, D. Sucharski, and B. Cowper, "State of the Art in Hull Monitoring Systems," *Report No. SSC-401*. Ship Structure Committee, Washington, DC, 1997.

[9] A. T. Zimmerman and J. P. Lynch, "A parallel simulated annealing architecture for model updating in wireless sensor networks," *IEEE Sensors Journal*, vol. 9, no. 11, pp. 1503-1510, 2009.

[10] A. T. Zimmerman, J. P. Lynch, and F. T. Ferrese, "Market-based resource allocation for distributed data processing in wireless sensor networks," *ACM Transactions on Sensor Networks*, p. (in review), 2010.

[11] A. Bemporad, M. Morari, A. Garulli, and A. Tesi, *Robustness in identification and control*, vol. 245. London: Springer London, 1999, pp. 207-226.

[12] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667-682, May 1999.

[13] P. D. Christofides, J. Liu, D. Muñoz de la Peña, and S. (Online service), "Networked and Distributed Predictive Control Methods and Nonlinear Process Network Applications." Springer-Verlag London Limited, London, 2011.

[14] S. H. Clearwater, *Market-based control : a paradigm for distributed resource allocation*. Singapore ; River Edge, N.J.: World Scientific, 1996, p. xiv, 311 p.

[15] I. N. Athanasiadis and P. A. Mitkas, "An agent-based intelligent environmental monitoring system," *Management of Environmental Quality*, vol. 15, no. 3, p. 238, 2004.

[16] L. McCauley and S. Franklin, "A large-scale multi-agent system for navy personnel distribution," *Connection Science*, vol. 14, no. 4, pp. 371-385, 2002.

[17] S. K. Srivastava et al., "A Control System Test Bed for Demonstration of Distributed Computational Intelligence Applied to Reconfi guring Heterogeneous Systems," *IEEE Instrumentation & Measurement Magazine*, vol. 11, no. 1, pp. 30-37, 2008.

[18] A. V. Savkin and R. J. Evans, *Hybrid dynamical systems: controller and sensor switching problems*. Springer, 2002, p. 153.

[19] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *Control Systems, IEEE*, vol. 22, no. 1, pp. 44-52, 2002.