Model-based Method to Codeign of Control, Computing, and Communications with Resource Constraints

Jiafu Wan\textsuperscript{a,b}, Hehua Yan\textsuperscript{b,*}, Hui Suo\textsuperscript{b}, Caifeng Zou\textsuperscript{b}

\textsuperscript{a}School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
\textsuperscript{b}College of Electrical Engineering, Guangdong Jidian Polytechnic, Guangzhou, China

Abstract

To optimize the overall quality of control (QoC) of networked control systems (NCSs), it is necessary to adopt the codeign from the control, computing, and communication perspectives. By analyzing related problems and method formulation, an implementation architecture for the codeign is proposed, which could flexibly allocate system resources under the constraints of communication bandwidth, CPU resource, and jitter range. The key to realizing codeign is an analysis model of NCSs that includes various factors such as control algorithm, CPU scheduling, and network resource. The optimized implementation scheme for codeign is estimated by comparing QoCs. Finally, the experiment with a CAN-based computer numerical control (CNC) system is conducted, and the results show that the scheduler adjusts system resource according to temporal time attributes, improving the QoC of CNC. The presented model-based approach to codeign is effective, and is helpful in using system resources and further enhancing overall QoC.

© 2011 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of ICAE2011.

Keywords: networked control systems (NCSs), codeign, fuzzy feedback scheduling, computer numerical control (CNC)

1. Introduction

Networked control systems (NCSs) are becoming increasingly important in modern control engineering and applications because using a network has many advantages, such as higher reliability, and easier deployment and maintenance \cite{1,2}. In NCSs, it is common that many spatially distributed system components such as sensors, actuators, and controllers, share a common communication network. The network bandwidth, together with the processing power of the CPU, is usually limited to save on the cost of hardware in practical applications. For resource-constrained systems, the performance of NCSs is intimately associated with temporal attributes such as network-induced delay, jitter, and packet loss. Unpredictable time characteristic parameters will probably degrade the quality of control (QoC) of NCSs.

* Corresponding author. Tel.: +86-13660305655.
E-mail address: hehua_yan@126.com.
and even cause system instability in extreme cases.

From the control, computing, and communication (3C) perspectives, the overall performance of NCSs depends not only on the design of control algorithms, but also on CPU scheduling and allocation of network bandwidth [3]. In view of resource constraints, NCSs have to adjust resource allocation dynamically to accommodate system reconfiguration or update. For example, the period of sampling task in a CAN-based computer numerical control (CNC) system varies with the runtime availability of system resource.

This paper is devoted to optimizing the overall QoC of NCSs by flexibly managing network resources and CPU scheduling. A model-based approach to optimal integrated 3C is proposed, which enables flexible QoC management in NCSs subject to CPU resource and network bandwidth. Our approach, unlike most existing NCS solutions that focus on control algorithms or network protocols of MAC layers, concentrates on the codesign of 3C. To maximize system resources, the scheduling policy realized in the analysis model simultaneously adapts the sampling periods at runtime. A feedback-scheduling algorithm based on output jitter of the key tasks is used to adjust sampling periods. In contrast to traditional design methods for NCSs, our approach features analysis model with resource constraints, this also exploits feedback control technology.

2. Related work

NCSs have recently received considerable attention, and many studies have been devoted to control algorithms, CPU scheduling, and allocating network bandwidth. Generally, related work may be classified into three categories: 1) control theoretic approaches, 2) network design-based approaches, and 3) codesign.

For control theoretic approaches, Hespanha et al. [2] summarized the recent results on NCSs, which mainly reviews a collection of results to determine the closed-loop stability of NCSs in the presence of network sampling, delays, and packet loss, at the same time addressing control synthesis methods for NCSs. Detailed references can be found in [4]. In addition, approaches based on network protocols of MAC layers primarily deal with improving network quality-of-service such that system performance is guaranteed [1].

In this work, we concentrate on the third category, the codesign of 3C. The results of this research consist of two sides, control and CPU scheduling, and bandwidth allocation. For the former, using feedback control theory and technology ensures that all key tasks can be finished before their deadlines under the restraint of CPU resources [5, 6]. Much effort has been made on closed-loop network scheduling, which is characterized by dynamic bandwidth allocation via sampling period adaptation [7]. Most adjust the sampling periods of control loops to optimize overall QoC. However, most work on codesign is devoted to allocating CPU resources or network bandwidth. The main concern of this paper is the model-based approach to study the policy of resource allocation, and further evaluate the QoC of NCSs.

3. Problem formulation

According to task timing attributes, tasks in NCSs are grouped into three categories, periodic real-time, periodic real-time and non-real-time tasks. From the computing point of view, the actual start of periodic real-time tasks may be delayed due to pre-emption from other tasks in the controller, which is called sampling latency of the controller. The uncertainty of sampling latency caused by resource competition could deteriorate system performance. Most controllers contain these three types of tasks. The number of different types of tasks is defined as follows:

\( n \): periodic real-time tasks, which are key considerations in designing the analysis model;  
\( m \): aperiodic real-time tasks, which are used to finish event processing;  
\( l \): non-real-time tasks.

NCSs with induced time characteristic parameters include three kinds of computer delays:  
\( L_m \), computational delay in the main controller;  
\( L_{sm} \), communication delay between the sensor and the main controller; and  
\( L_{ma} \), communication delay between the main controller and the actuator. The total control
delay, $L_i$, for NCSs, which is the time from when a measurement signal is sampled to when it is used in the actuator, equals the sum of these delays,

$$L_i = L_{im} + L_{sm} + L_{ma}$$

(1)

Since designing NCSs covers many subjects, many factors are considered during embedded system implementation, including control algorithm, communication protocol, and scheduling policy. Without loss of generality, the following are also assumed.

- From the control point of view, the control policy adopts the frequently used PID algorithm.
- The Controller Area Network (CAN) bus is used as the communication network, which is the priority-driven control network.
- Non-real-time tasks in controllers are ignored because they generally have no effect on system performance.
- Sample data is delivered in the form of single packets, which means every sample will be treated as one data packet while being transmitted over the CAN network.

Consider a controller with limited processing power, on which independent control tasks run concurrently. From the communication point of view, network bandwidth is often subject to the network resource. In this paper, we define the following timing attributes for real-time tasks in NCSs.

- Execution time $c_i$: time-varying and unavailable.
- Period $h_i$: equal to the sampling period of the corresponding control loop, and is available precisely online.
- Minimum of time interval $h_i^*$: treated as the period of aperiodic real-time tasks
- Network utilization $b_i$: equal to $L_i^* / h_i$.
- The upper bound of network bandwidth $U_d$: greater than or equal to total network utilization.

According to these definitions, the following expression is the constraint of computing resources to ensure schedulability:

$$\sum_{i=1}^{n} c_i / h_i + \sum_{i=1}^{m} c_i / h_i^* < 1$$

(2)

For a NCS composed of independent control loops, in which an ideal priority-driven control network (i.e., CAN) is used, Eq. (3) is satisfied:

$$\begin{cases} 
\sum_{i=1}^{n} b_i \leq U_d \\
\min b_i \leq b_i \leq \max b_i 
\end{cases}$$

(3)

where $b_i^{\min}$ and $b_i^{\max}$ are determined in [8].

The fundamental requirement of NCS is stability. Transient behaviour is also another focus for system design. Beyond these properties, more widely used criteria for QoC involves control error $e(t)$, which is defined as the difference between the setpoint $r(t)$ and system output $y(t)$. According to the control error $e(t)$, the integral of absolute error (IAE) is used as the QoC index:

$$\text{IAE} = \int_{t_0}^{t_f} |e(t)| \, dt \approx \sum_{k=k_0}^{k_f} |r(kh) - y(kh)|$$

(4)
where $t_0 (k_0)$ and $t_f (k_f)$ are the initial and final continuous (discrete) times of the evaluation period, respectively.

### 4. Integration of 3C

Fig. 1 gives the block diagram for the method formulation, which is mainly composed of an analysis model and an evaluation procedure. Through the analysis model, the embedded system implementation of NCSs is assessed under the constraints of computing resource and communication bandwidth.

The analysis model includes the distributed controller nodes connected by communication network, in which the control algorithm, communication protocol, and scheduling policy may be modified. Thus, IAE shows the effect of different factors such as period, control algorithm, network resource, and timing attributes, on the QoC of NCSs. Via multiple experiments, the problem of optimized design may be solved by comparing IAE under a variety of conditions, and complicated mathematical solutions are avoided. The evaluation procedure from influence analysis to the analysis model forms the closed-loop feedback, which is convenient when studying the relationship between influencing factors and overall QoC.

The implementation architecture of real-time tasks in NCSs is shown in Fig. 2. The scheduler is designed only when the following criteria are met. 1) From the computing point of view, all real-time tasks are finished before their deadlines, that is, Eq. (2) is satisfied. 2) The constraint of communication resource Eq. (3) meets the requirements. 3) The output jitter is controlled within a range that meets the given condition:

$$\sum_{i=1}^{n} (J_{si} - J_{si})^2 \leq J_{sr}^2$$

where $J_{si}$ and $J_{sr}$ denote the factual and expected output jitter of task $i$, respectively, and $J_{sr}$ is the desired jitter value of all control tasks.

The fuzzy feedback scheduling based on output jitter has been explored in our previous work. For simplicity, we directly give the crucial look-up table in [9]. The period rescaling factor $\lambda$ is dynamically adjusted by the scheduler. According to the hypotheses, the analysis model is established to evaluate the overall QoC of NCSs. Therefore, the optimized implementation scheme can be found from the 3C perspectives, and the codesign is realized.

![Fig. 1. Block diagram for the method formulation](image1)

![Fig. 2. Implementation architecture of real-time tasks](image2)
5. Application to a CAN-based CNC system

In this section, we conduct the codesign exemplified by a CAN-based CNC system to evaluate the proposed method. The analysis model is based on Matlab/Simulink and TrueTime toolbox. As a performance metric for a CAN-based CNC system, the mismachining tolerance $d_k$ is defined as the distance between the actual coordinate and the reference coordinate:

$$d_k = \sqrt{(x_a - x_r)^2 + (y_a - y_r)^2}$$  \hspace{1cm} (6)

where tuple $(x_a, y_a)$ denotes the actual coordinates, and tuple $(x_r, y_r)$ denotes the reference coordinates. Consequently, Eq. (6) is combined with Eq. (4), and the IAE is computed.

The related parameters for the codesign are listed as follows: proportional $K_p$ is 8.5, integral $K_i$ is 0.38, derivative $K_d$ is 0.35, the size of every packet is 40 bits, jitter range is 20%, and the sampling period is 1, 2, and 4ms, respectively. For the servo controllers, the transfer functions of servo systems for both control loops are $G(s) = 250/(s^2+25s)$, and the controllers are designed by discretizing continuous-time controllers that use the PID algorithm [6]. Through the period rescaling factor $\lambda$, the servo period is dynamically adjusted by the scheduler to 1, 2, and 4 ms. The upper limit of output jitter is no greater than 20% of the sampling task period. The data packets over the CAN bus are set to 40 bit. By satisfying these three conditions, the scheduler in the analysis model makes the best use of system resources to obtain good overall QoC.

Assume that a circle with a radius of 1,000 mm is machined, and the simulation time is 2 s. Under the constraints of communication bandwidth, CPU resource, and jitter range, the scheduler dynamically adjusts the sampling period to improve the overall QoC of CNC. In the course of the experiments, the initial sampling period is set to 4 ms, and the scheduler regulates the period from 4 to 2 ms according to [9]. The positive effects of the flexible resource management are observed by calculating the IAE.

The network status is shown in Fig. 3, in which there is an apparent difference in network utilization. Comparing Fig. 3 (a) with Fig. 3 (b) shows that the network has high utilization ratio for sampling period 2 ms, and the jitter of transporting data packets is less than the upper limit. Fig. 4 shows the real-time task executions of the servo controllers. Through flexible management, the scheduler adjusts the sampling period from 4 to 2 ms, and all tasks are still schedulable. When the aperiodic real-time tasks causing the output jitter are beyond the upper limit, the sampling period is modified by the scheduler from 2 to 4 ms.

By comparing the contour error under different sampling periods, we can further evaluate the model-based method to the codesign. Due to the adjustment, the times of position control change from 500 to 1,000. The comparison of contour error is given in Fig. 5. For the sampling period 4 ms, in the time interval from $t=1.0$ to 1.15 s, the square of contour error is obvious, as shown in Fig. 5.

Suppose the time interval $h_i$ is 2 ms, the IAE from 1 to 2 s can be calculated according to Eqs. (4) and (6):

$$\text{IAE} = \sum_{k=1}^{500} |d_k| \cdot h_i$$  \hspace{1cm} (7)

As shown in Table 1, the IAE is 0.4156 and 1.2103 mm-s, and the maximum of contour error is 4.02 and 2.50 mm for different sampling periods, which indicates the validity of codesign.
As shown in Table 1.4156 and 1.2103

![Graph comparing contour error and time](image1)

![Graph showing task executions with different sampling periods](image2)

6. Conclusions

The codesign of NCSs involves multidisciplinary such as control, communication, and computer sciences. In this paper, the constraint conditions on network bandwidth and computing resource are analyzed, and then the QoC index IAE is introduced. By illustrating the method formulation, an implementation architecture for codesign is presented, which includes the 3C technologies.

Applying the proposed approach to a CAN-based CNC system, we conduct the codesign and evaluate the validity. The scheduler dynamically adjusts the sampling period according to temporal time attributes to make good use of system resources. The experimental results show that the model-based approach to the codesign of 3C is effective.

Acknowledgements

The authors would like to thank the China Postdoctoral Science Foundation (No. 20090460769), the Fundamental Research Funds for the Central Universities, SCUT (No. 2011ZM0070), the National
Natural Science Foundation of China (No. 50905063), and the Natural Science Foundation of Guangdong Province, China (No. S2011010001155) for their support in this research.

References

[1] Xia, F., Sun, Y. X., and Tian, Y. C. Feedback scheduling of priority-driven control networks. Computer Standards & Interfaces, 2009, 31(3), 539-547.

[2] Hespanha, J. P., Naghshtabrizi, P., and Xu, Y. G. A survey of recent results in networked control systems. Proceedings of the IEEE, 2007, 95(1), 138-162.

[3] Dai, S. L., Lin, H., and Ge, S. S. Scheduling-and-control co-design for a collection of networked control systems with uncertain delays. IEEE Transactions on Control Systems Technology, 2010, 18(1), 66-77.

[4] Song, H., Yu, L., and Zhang, W. A. Stabilisation of networked control systems with communication constraints and multiple distributed transmission delays. IET Control Theory and Applications, 2009, 3(10), 1307-1316.

[5] Wan, J. F., Li, D., and Zhang, P. Key technology of embedded system implementation for software-based CNC system. Chin. J. of Mechanical Engineering, 2010, 23(2), 217-224.

[6] Wan, J. F., Li, D., Yan, H. H., and Zhang, P. Fuzzy feedback scheduling algorithm based on central processing unit utilization for a software-based computer numerical control system. Proceedings of the Institution of Mechanical Engineers, Part B: J. Engineering Manufacture, 2010, 224(7), 1133-1143.

[7] Velasco, M., Fuertes, J. M., Lin, C., Marti, P., and Brandt, S. A control approach to bandwidth management in networked control systems. In Proceedings of the 30th IEEE Annual Conference on Industrial Electronics Society, Busan, Korea, 2-6 November, 2004, pp. 2343-2348 (IEEE Industrial Electronics Society, Washington).

[8] Park, H. S., Kim, Y. H., Kim, D. S., and Kwon, W. H. A scheduling method for network-based control systems. IEEE Transactions on Control Systems Technology, 2002, 10(3), 318-330.

[9] Wan, J. F., and Li, D. Fuzzy feedback scheduling algorithm based on output jitter in resource-constrained embedded systems. In Proceedings of 2010 International Conference on Challenges in Environmental Science and Computer Engineering, Wuhan, China, 6–7 March 2010, pp. 457–460 (IEEE Computer Society, Washington).