

Research on the model design and simulation for training platform based on network control

Wang Fanfan, Liu Lusi, Wang Xiaoli

Department of Information Technology ,Hainan College of Economics and Business, Haikou
Hainan 571127

Wangfggei@163.com

Keywords: training platform, Robot control system, scheduling scheme, scheduling

Abstract. In order to solve the problem of slow response and packet loss which are caused by the introduction of the bus in our robot control system, In this paper, we used Matlab/Simulink TrueTime toolbox to build a bus-based control system simulation platform. The system controlled three objects, which were taken as three closed-loop control tasks. Scheduling algorithms are using the classic RM and EDF algorithm, the simulation results shown in Figure 1 (RM) and Figure 2 (EDF). RM static scheduling algorithm would cause some low-priority tasks that were always in the state of waiting and lacking of implementation, while EDF dynamic scheduling algorithm could dynamically adjust the priority of the task according to its deadline, it greatly improved the system's stability of the low priority tasks, and significantly reduced the system's overshoot and rising time. Therefore, the EDF dynamic scheduling algorithm was very necessary to be applied to the system.

I. INTRODUCTION

In this paper, we used Matlab/Simulink TrueTime toolbox to build a bus-based control system simulation platform. The system controlled three objects, which were taken as three closed-loop control tasks. Scheduling algorithms are using the classic RM and EDF algorithm, the simulation results shown in Figure 1 (RM) and Figure 2 (EDF). RM static scheduling algorithm would cause some low-priority tasks that were always in the state of waiting and lacking of implementation, while EDF dynamic scheduling algorithm could dynamically adjust the priority of the task according to its deadline, it greatly improved the system's stability of the low priority tasks, and significantly reduced the system's overshoot and rising time. Therefore, the EDF dynamic scheduling algorithm was very necessary to be applied to the system.

2. TRAINING PLATFORM BASED ON NETWORK CONTROL

The scheduling module in this paper was applied to the networked control systems, and it based on the bus structure^[1]. The scheduling policy of the control systems in this paper ranged over the scheduling on the bus layer and the application layer, which included: 1) the task scheduling on the application layer; 2) the bus resource scheduling. Tasks scheduling mainly considered how to set up the precedence level of control tasks in order to meet the real-time system. Bus resource scheduling mainly considered how to allocate the bus resources among multiple tasks and made the bus utilization rate maximum to meet the requirements of real-time.

The number of control loops and application nodes which effected on the bus resources and CPU resources were time-variant in these control systems, and the loads was uncertain or unknown which led to the unexpected runtime environments. In the uncertain environment, the control methods based on the offline designing in the predictable environment would lead to the deterioration and instability of the system's performance. We needed to deal effectively with the influence on the performance under the uncertain resources, and provide support for the performance optimization in dynamic environment (uncertain loads) .

So, we proposed the feedback scheduling policy. The scheme included the features as followed: it took the multi-objective function, which was composed of the packet loss rate and response time, as a measurement standard to assign priority of each tasks using the method of the “EDF scheduling policy based on improved fuzzy feedback”, and eventually achieved the suppression of the deterioration of the performance. And also we optimally allocated the bus resources through the feedback messages, adjusted the sampling period of each tasks, then further optimized the system performance.

The scheduler (Figure 1) involved the scheduling policy on the bus layer and the application layer. The priority configurator set the priority of all the task's messages and put the messages into the ready task queue. The bus resource manager adjusted the task's period to reallocate the bus resources, and then increased the bus utilization and met the real-time requirements of control systems.

In a sampling period of the scheduler, the feedback performance index function as followed:

$$J_f(j) = \alpha E(j) + \beta \tau(j), \quad E(j) = \frac{X(j) - X_s(j)}{X(j)}, \quad X(j) \text{ represented the total number of data packets, } X_s(j)$$

represented the total number of the data packets which had been transmitted successfully, τ was response time ratio and $\tau = \frac{H(j)}{H_{\max}}$, $H(j)$ was the execution time of the task, H_{\max} was the task's maximum execution time, α was the weights of the packet loss rate, β was the weights of response time ratio, and $\alpha + \beta = 1$.

The differential of performance indexes was as followed: $JC_f(j) = J_f(j) - J_f(j-1)$, it echoed the direction of change for performance indexes. The performance index function could be transformed into a single objective optimization problem using the weight sum method. Weights were needed to be based on the actual system requirements or determined by the simulation method as followed: if the response time was ample, the weight of error would increase; if the response time was running out, its weight would increase.

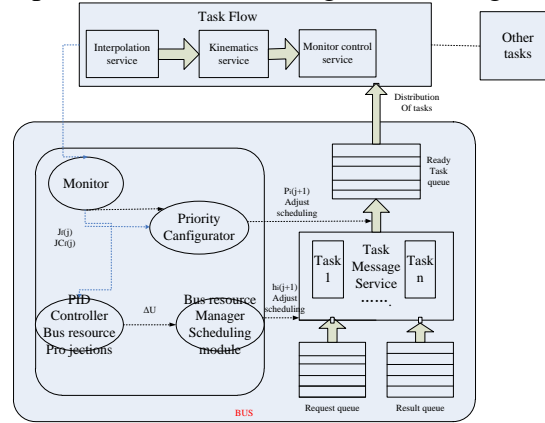


Figure 1 Feedback control scheduling framework

If there were n tasks running in the control system, the sampling period of the scheduler h_f should be equal to integral times of the period of the longest task h_i , $h_f = A \max[h_i]$, A was an integer, h_i was the period of the i -th task, $i \in [0, n]$, in order to avoid frequent switch of the scheduler, A was greater than 1 generally. Assuming that h_i was the longest period of the task, there were three tasks: interpolation task, kinematics task, and motion control service, the period of all the tasks was h_i , and taking into account the integrity of the process, A was took the value of 3, so $h_f = 3h_i$ (Figure 2).

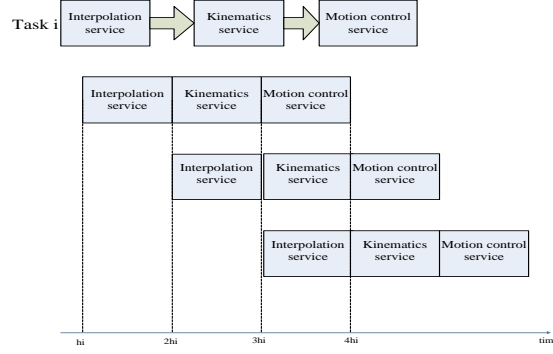


Figure 2 Tasks running time allocation figure

The scheduler was time driven; it monitored the operational status of each task when sampling time coming, and obtained the following informations:

1. The numbers of packets that had been sent to the bus by each task $X(j)$, and the packets that had been transmitted successfully by the bus $X_s(j)$;
2. The task's command response time τ and the forwarding times of the tasks on the bus τ_b .

In this paper, the bus utilization and the task's priority had been allocated dynamically according to the feedback performance of the task loops. If the performance of the control task loops worsen, the task should be assigned to more bus resources or the priority got higher. The task scheduling policy and the bus resource scheduling were described as followed.

3. TASK SCHEDULING

A. Planning and mapping priority

A double layer scheduler was designed in the task priority configurator. This layered design was to achieve the benefits of platform independence. The first layer scheduling was only responsible for distributing the global priority, regardless of the scheduling mechanism of specific operating systems. The second layer scheduler was closely related to the underlying scheduling algorithm of the operating system, which was responsible for the global priority mapping to the system's effective priority. When the open control system needed to be ported to other operating system platforms, you only needed to modify the scheduling of the second layer; so, this double layer scheduling framework made the system's portability greatly enhance.

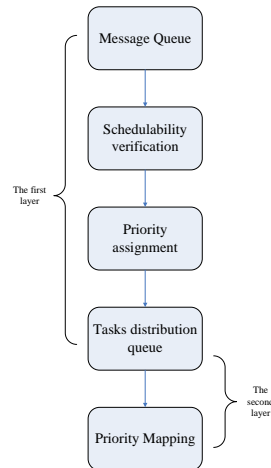


Figure 3 Task scheduling block diagram

In this paper, we used the real time extension mode of universal operating system Windows+RTX, which didn't directly modify the scheduling module of the RTX platform, but used the priority mapping approach to assign a priority for the job thread. The thread could be divided into 0-127 priority in the RTX kernel subsystem, totaling as 128 priority. The priority of the entire control system followed the principles of: P emergency operation \gg P scheduling module \gg P bus function thread \gg P other service node, the priority mapping planning shown in Figure 4. The

system needed to monitor the status of running services, and the priority must be re-mapped when the new services or thread started.

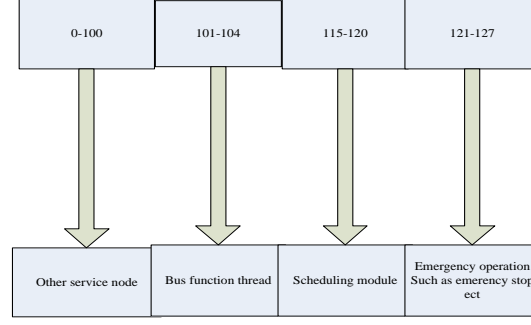


Figure 4 RTX priority mapping planning

B. Task priority assignment

The project would assign priority to each task according to the following scheduling algorithm: the scheduler arranged the task's message (the message contained the priority attributes including the task's arrival time and deadline) to the ready task queue according to the the priority and then distributed messages to trigger execution of each sub-tasks; the priority attribute of each sub-task should be consistent in a task cycle h_i .

The traditional EDF algorithm could not timely adjust the task's priority in an uncertain running environment, and the feedback scheduling did not dispatch timely because of the introduction of a feedback mechanism (scheduling period should not be too short to avoid frequent switching task priority). So, we proposed "the fuzzy feedback improved EDF scheduling policy " to assign the priority of each task, and to sort the task messages in the ready task queue on priorities ascending order. The basic idea was described as followed (Figure 5):

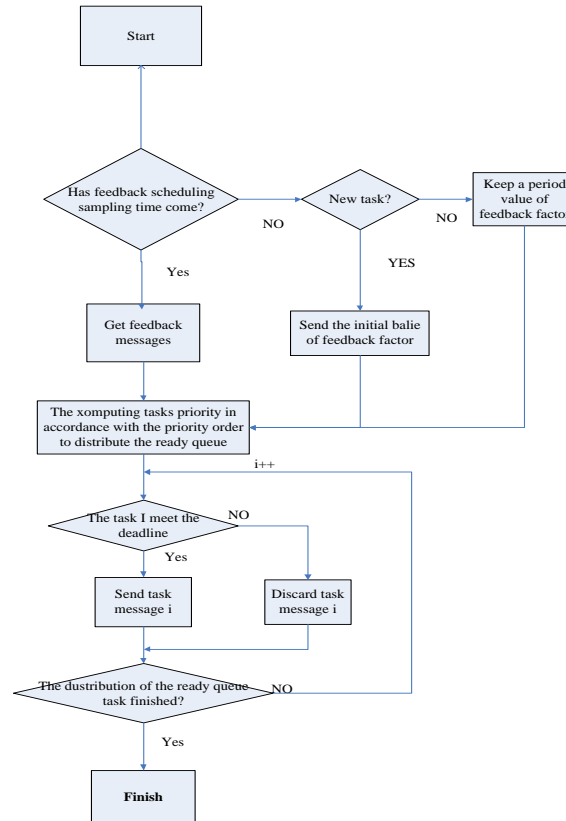


Figure 5. Scheduling process

4. EXPERIMENTS

The deadline of the packets was 20ms, and there was a testing process for interference. Set $K_p=5$, $K_I=10$.

The data of the packet loss rate testing without the scheduler was shown in Figure 6. Its packet loss rate was 3.69%.

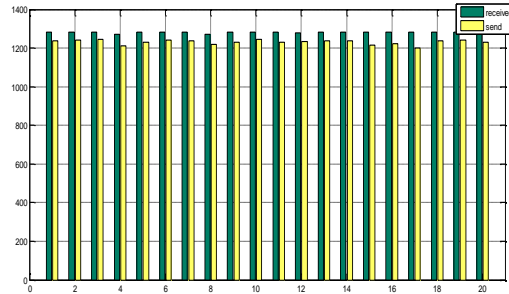


Figure 6 Packet loss rate testing without the scheduler

The data of the packet loss rate tested with the scheduler was shown in figure 5. In this experiment, the packet loss rate was 0.85%, which was significantly lower than the values without the scheduler, and the resource utilization of the bus arrived above 85%. The experiment proved that the two levels scheduler not only ensured the lower packet loss rate, but also effectively improved the utilization of the bus.

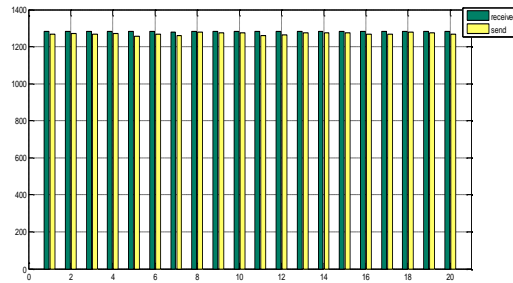


Figure 7 Packet loss rate testing with the first level

V. CONCLUSION

The two levels scheduling framework in this paper could be applied to the control system based on the bus. First, we used “the improved fuzzy feedback EDF scheduling policy” to dynamically adjust the priority of the tasks according to the task’s deadline, the importance and the feedback of performance index, which ensured that the packets with greater importance should have the higher priority to be transmitted, the packets closer to the hardware should have the higher priority too, the priority should be increased when the performance of the control loops was worsening, and achieved the suppression of the deterioration of the system performance. Secondly, we adjusted the sampling period using the bus resource scheduling policy, dynamically controlled the traffic of the messages on the bus to ensure the schedulability, reduce packet loss and delay, and get better performance than the system with fixed sampling period. Finally, the experiment results showed that the scheduler could effectively solve the problem of slow response and high packet loss that were caused by the introduction of the bus.

REFERENCES

- [1] Berrichi Ali, Yalaoui Farouk. Efficient bi-objective ant colony approach to minimize total tardiness and system unavailability for a parallel machine scheduling problem. International Journal of Advanced Manufacturing Technology, 2013, p 1-16.

- [2] S. Li, Y. Geng, J. He, K. Pahlavan, Analysis of Three-dimensional Maximum Likelihood Algorithm for Capsule Endoscopy Localization, 2012 5th International Conference on Biomedical Engineering and Informatics (BMEI), Chongqing, China Oct. 2012 (page 721-725)
- [3] Y. Geng, J. He, H. Deng and K. Pahlavan, Modeling the Effect of Human Body on TOA Ranging for Indoor Human Tracking with Wrist Mounted Sensor, 16th International Symposium on Wireless Personal Multimedia Communications (WPMC), Atlantic City, NJ, Jun. 2013.
- [4] Chen ZaiPing, Xu HongQiao. Research on network control system using improved EDF dynamic scheduling algorithm. Advanced Materials Research, 2012, p 2420-2423.
- [5] Wang Jun . A scheduling algorithm based on communication delay for wireless network control system. Advances in Intelligent and Soft Computing, 2012, p 99-104, 2012, Frontiers in Computer Education