Development of EMA Software for the Periodic Movement of Magnet Arrays in an APPLE Type Undulator

Yoichi HIRAMATSU, Taihei SHIMADA, and Yoshikazu MIYAHARA
Japan Synchrotron Radiation Research Institute (JASRI)
1-1-1 Kohto, Mikazuki-cho, Sayo-gun, Hyogo 679-5198, Japan

Abstract

An extended part of control software for an APPLE type undulator was developed. The major capability of the original software was to correct the closed orbit distortion (COD) of an electron beam by exciting steering magnets during the movement of permanent magnet arrays. The extended part of software is capable to put these magnet arrays into a constant periodic movement with an error less than 0.1% for the period of 2 sec (0.5 Hz). This paper presents the mechanism of the extended software and discusses preliminary experimental results.

1 Introduction

Since the installation of APPLE (Advanced Planar Polarized Light Emitter) type undulator, we have developed and improved the software of the control system. As presented in previous papers [2, 3], this control software has been developed in accordance with the directions of SPring-8 standard for software development. That is, the application program were written to satisfy a standard software frame format known as EM-frame (Equipment Manager). Initially, based on this EM-frame a software process named EM was developed in order to move the permanent magnet arrays with simultaneous excitation of steering magnets. The excitation rate was 42 Hz for both movement of gap and phase. But, when we put the magnet arrays into a periodic cycle of phase movements, we noted an existence of certain delay-time during cycles caused by the client-side process GUI and by LAN network load. Past experimental data showed a random delay-time during one cycle of periodic movement. This time-axis error conducted us to improve the software EM with the objective to minimize the error. The improvement was achieved by a novel approach of creating an independent software process named process EMA (Equipment Manager Agent) on the same CPU of control computer. While the original EM had a mission to respond all control commands (known as SVOC commands) emitted at GUI, the new EMA has a specific mission to execute only selected actions like that of periodic phase movement of magnet arrays. As EMA being an extended process derived from the original EM, it is controlled directly by EM through the control commands like create, start, stop, and destroy. This direct control of EMA by the original EM (and not by GUI) is an important subject of the present paper. That is, as a consequence of the direct control, EMA does not depend on the LAN network load and produces a periodic phase movement with a period at a high degree of accuracy. We confirmed by experiments a satisfactory level in the results of error less than 0.1% for periodic movement of phase with a period of 2 sec. The objective of this paper is to present briefly the mechanism and performance of newly developed software process EMA.

2 Difference between Original EM and Newly Developed EMA

While the original EM was the name of executable code resulted from EM-frame, the newly developed EMA is an executable code resulted from EMA-frame. These two frames use the same application programs, configuration table and current excitation table. The difference is on the different communication ports, that is the EM-frame is assembled to receive and emit control commands via RPC (Remote Procedure Call) interface port while EMA-frame is assembled to communicate via Message-queue interface port. This means that EM receives the SVOC commands (emitted at GUI) and returns replies through the RPC, and EMA receives SVOC commands (emitted at EM) and returns replies through the Message-queue. In other words, EMA is a kind of copy of the original process EM which behaves exactly as the original one. As the both EM and EMA run on the same CPU, when the one is active, other keeps inactive. Instead of the original EM in charge of all control of the undulator, the extended part EMA is specifically in charge of the periodic movement of phase.

Fig.1 shows the relationship of parent and child of EM and EMA. As EMA being a child-process, several EMA can be created according to the necessity of the undulator control. In the present case of coexistence of EM and EMA, these two processes are, respectively, in charge of the following actions of the undulator:

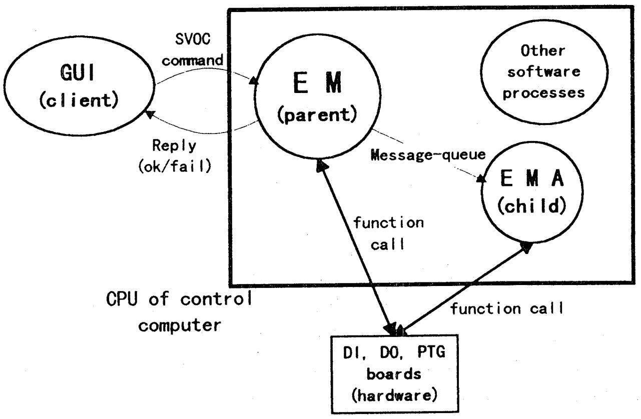

Fig. 1 Coexistence of EM and EMA on the same CPU

EM: to move magnet arrays in the gap and phase directions, to read gap-distance and phase-shift, to read present excitation currents of steering magnets, to excite steering magnets, to read vacuum level, to read status of undulator components, to command EMA, etc.
EMA: to move magnet arrays in the phase direction, to read gap-distance and phase-shift, to excite steering magnets, and to emit trigger on/off signals (of polarization state) to beam users.

When the beam user needs to close the gap from a given gap-distance to a target value, he emits at GUI a SVOC command (ordering to move the gap) directed to EM. On the other hand, when the beam user needs a periodic movement of phase, he needs to emit at GUI a SVOC command asking EM to create its child-process EMA, as shown in Fig.2. After EMA is created, he emits a next SVOC command asking to start EMA. Then, an endless loop of periodic movement is started. The loop is stopped only when the beam user emits at GUI a command stop asking to stop EMA. When EMA receives the command stop, it immediately finishes the loop at the end of the same cycle.

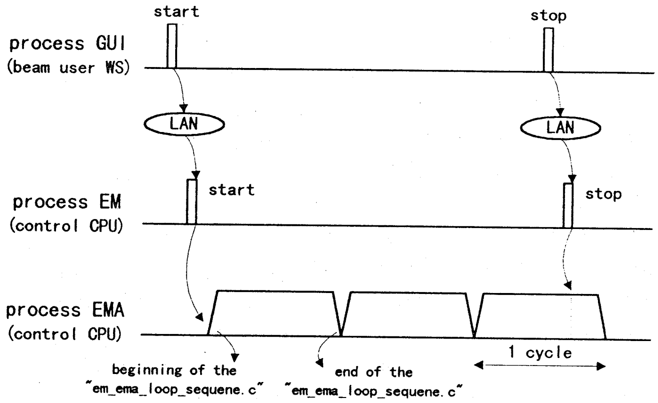

Fig. 2 Timing-chart of "start" and "stop" of EMA

3 Mechanism of Process EMA

As seen in the previous section, EMA has same nature of the original EM, but with an important property that EMA is controlled only by EM and not by GUI. This property can be used to perform an interesting action on the undulator, such of in the present case a periodic movement of phase of magnet arrays. Fig. 3 shows one cycle of this periodic movement. EMA-frame requires the commands of the periodic part be included in a special function named "em_ema_loop_sequence.c" which is a EMA-frame function released for customization. SVOC commands are used here in the form of recursive-call. It is this function that is the essence of the process EMA.

Below is an example for phase-shift of 32.0 mm and hold-time of 200 msec. The phase-shift 32.0 mm produces a X-ray with circular polarization. The following SVOC commands are issued at time-instants (a) to (g) of Fig. 3:

(a) put/bl_id23_phase_trigger_l/off (This command notices the beam user the end of left-polarization. It is issued when the magnet arrays are at a position of phase-shift -32.0 mm.)
(b) put/bl_id23_phase/32.0mm (This puts the magnet arrays to a movement to the phase-shift 32.0 mm. This command is the same as that used on the original EM and it includes recursive-call of SVOC commands for excitation. Thus, during the present movement of phase, all steering magnets are at the same time excited at a rate of 24 msec.)
(c) put/bl_id23_phase_trigger_r/on (This notices the beam user the beginning of right-polarization. It is issued when the magnet arrays are at phase-shift 32.0 mm.)

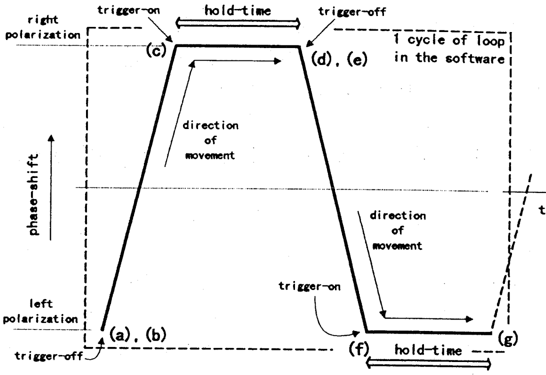

Fig. 3 Schematic of a cycle of periodic movement of magnet arrays during the software EMA. ((a) to (g) are time-instants of issue of SVOC commands explained in the text)

(d) put/bl_id23_phase_trigger_r/off (This notices the beam user the end of right-polarization. It is issued when the magnet arrays are at phase-shift 32.0 mm.)
(e) put/bl_id23_phase/-32.0mm (This puts the magnet arrays to a movement to the phase-shift -32.0 mm. All steering magnets are excited at a rate of 24 msec during the phase movement.)
(f) put/bl_id23_phase_trigger_l/on (This notices the beam user the beginning of left-polarization. It is issued when the magnet arrays are at phase-shift -32.0 mm.)
(g) here, the control point of the function returns automatically to the time-instant (a).

When EMA receives the command start from EM, it starts the periodic phase movement always at the time-instant (a), independently of any initial position of phase. EMA puts first the magnet arrays into a position of phase-shift -32.0 mm before starting its loop. And, when EMA receives the command stop, the cycle is stopped always at the time-instant (g). The beam user can re-start EMA at this last phase position at any time.

The first task of the beam user is to key-in at GUI the command "put/bl_id23_pattern/create" ordering EM to create its child EMA. The second task is to transmit to EMA the values of phase-shift and hold-time through commands like, "put/bl_id23_pattern_phase/32.0mm_-32.0mm" and "..._hold/200msec_200msec". The first 32.0 mm is for phase position of right-polarization and the second one is for left-polarization. Same rule is applied for values of hold-time. Then, he starts EMA.

It is important that there are no more delay-time during the periodic movement of phase because of the child EMA is commanded directly by its parent EM and because of EMA is free from the network. This is the most important subject achieved by the extended part EMA in relation to the original EM. Additionally, note that the undulator status like vacuum level, intensity of excitation currents, gap-distance and phase position, and others are collected and stored to the SPring-8 facility database system. These status are collected by a process known as Poller-Collector which emits SVOC commands at 10 sec regular intervals. This is a case of interruption of EMA (of CPU) by another process. The interruption is solved by assigning CPU resource during the inactive time of EMA, that is during small intervals of time between SVOC commands and the hold-time, without disturbing the period of cycles.

4 Results of Periodic Movement of Magnetic Arrays

In order to check experimentally the periodicity of movement, we introduced in our VME system a study tool of 128 Mbytes DRAM to register a log of SVOC commands and time-stamps. SVOC commands registered in the DRAM log have two origins: those emitted by beam users at GUI panel, and those issued inside EMA in the form of recursive-call. By the analysis of DRAM log, we could know when EM and EMA received SVOC commands and when they returned replies, therefore the CPU-time consumed by EM and EMA. Fig. 4 is a result of periodic movement for phase-shift of 32.0 mm and hold-time of 200 msec collected from DRAM log. The vertical coordinate indicates phase-shift and the horizontal one is elapsed-time. Note that the period is 2 sec (0.8 sec of moving time and 0.2 sec of hold-time, as set at GUI panel) and the cycle repeats at a constant period. It was confirmed that the time-axis error for repetition is less than 0.1 %. Similarly, Fig. 5 shows the excitation of steering magnets at a rate of 24 msec (42 Hz) during the same phase movement. The points indicated by solid circles are time-instants of readings of phase-shift, and those indicated by cross symbols are time-instants of the excitation of steering magnets. Each excitation occurs 16 msec after the reading of phase-shift for all steering magnets. Hardware factors limit this period, such as of reading response of linear-scale, of processing speed of CPU, of response of I/O devices, etc.

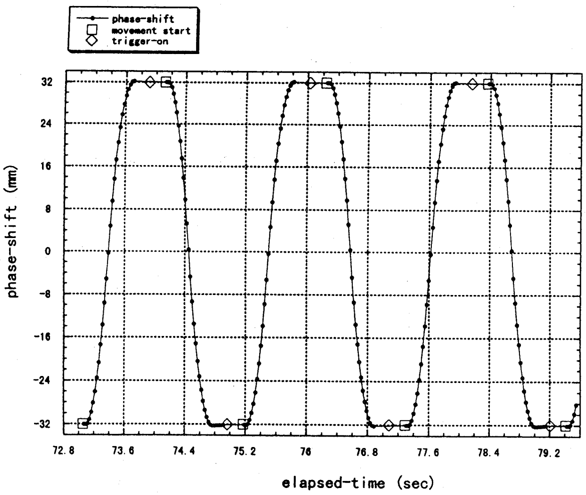

Fig. 4 Periodic movement of magnet arrays by using EMA (movement of phase)

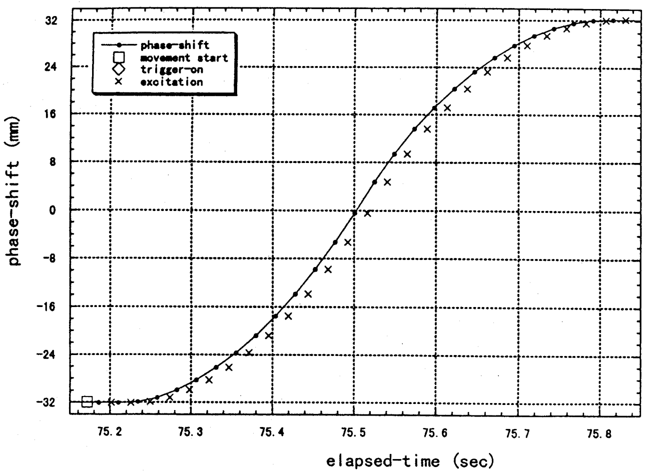

Fig. 5 Excitation of steering magnets at a period of 24 msec (42 Hz)

5 Conclusion

A control demerit by the original software EM was the existence of LAN network delay-time. We have studied, as presented in the present paper, a novel approach in order to isolate the specific part of periodic movement in an extended software EMA. EMA is an independent process that puts the magnet arrays into a periodic movement accompanied by the simultaneous excitation of steering magnets. The delay-time has been disappeared because EMA does not more depend on the LAN network load. A satisfactory result of periodic movement with an error less than 0.1% for the period of 2 sec (0.5 Hz) was obtained. In coming machine studies, we are planning to use this EMA and improve the performance of software based on experiment data of COD correction.

Acknowledgement

The results presented here would not have been possible without the careful assistance of all colleagues of SPring-8 Accelerator and Beamline Control Groups.

References

[1] H. Kobayashi et al., "Design of Variable Polarizing Undulator (APPLE-Type) for SX Beamline in the SPring-8", Proceedings of the Fifth European Particle Accelerator Conference (EPAC-96, Barcelona), Vol. 3, June (1996).

[2] Y. Hiramatsu et al., "Control System for an APPLE Type Undulator", SPring-8 Annual Report 1998 (1998).

[3] Y. Hiramatsu et al., "Development of EM Software for the Dynamic Excitation of Steering Magnets in an APPLE Type Undulator", this Conference.