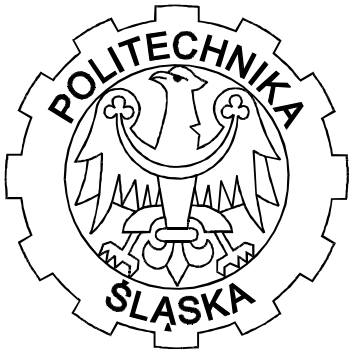
Arkadiusz Jestratjew

Zastosowanie wielozadaniowości do poprawy parametrów czasowych wykonania aplikacji w węźle rozproszonego systemu czasu rzeczywistego

Autoreferat rozprawy doktorskiej



Promotor

Dr hab. inż. Andrzej Kwiecień
Prof. w Politechnice Śląskiej

Gliwice 2011

Utilization of multitasking for improvement of application execution timing in a node of distributed real-time system

Summary of Ph. D. dissertation

Politechnika Śląska, Instytut Informatyki, Gliwice 2011.

Spis treści

(*tą stronę celowo pozostawiono pustą*)

Wprowadzenie

W normie IEC 61131-1 zdefiniowano sterownik programowalny PLC (ang. *Programmable Logic Controller*) jako „*cyfrowy system elektroniczny zaprojektowany do stosowania w środowisku przemysłowym, który wykorzystuje pamięć programowalną do wewnętrznego przechowywania specjalizowanych rozkazów realizujących funkcje takie jak operacje logiczne, przetwarzanie sekwencyjne, odmierzanie czasu, zliczanie oraz operacje arytmetyczne, w celu sterowania, poprzez cyfrowe i analogowe wejścia i wyjścia, różnorodnych maszyn lub procesów. Zarówno sterownik PLC jak i związane z nim peryferia są zaprojektowane w sposób umożliwiający ich łatwą integrację z przemysłowym systemem sterowania oraz łatwe zastosowanie we wszystkich zamierzonych funkcjach*”[†]. Sterowniki PLC są specjalizowanymi komputerami opracowanymi w celu sterowania procesami produkcyjnymi. Wykonywane są w sposób zapewniający dużą trwałość i niezawodność w trudnych warunkach środowiskowych, które występują w fabrykach, takich jak zapylenie, wibracje czy zakłócenia elektromagnetyczne. Sterowniki PLC są wyposażane w obwody wejścia-wyjścia umożliwiające interakcję z otoczeniem poprzez sygnały elektryczne, np. analogowy sygnał prądowy 4-20 mA, binarny sygnał napięciowy 0-24 V i inne.

Sygnały wejściowe, pochodzące z odpowiednich czujników wielkości fizycznych, niosą informację o stanie poszczególnych elementów maszyny, urządzenia czy też procesu technologicznego. Na ich podstawie wykonywane cyklicznie oprogramowanie aplikacyjne sterownika generuje sygnały wyjściowe, które za pośrednictwem odpowiednich urządzeń wykonawczych oddziałują na świat zewnętrzny.

Dla poprawnego działania sterowanych urządzeń konieczna jest cykliczna interakcja oprogramowania sterownika ze środowiskiem zewnętrznym, a sygnały wyjściowe muszą być wygenerowane w odpowiednim czasie (terminie), zatem sterowniki PLC są przykładem systemów informatycznych czasu rzeczywistego. Czas reakcji na zdarzenia zewnętrzne zależny jest od czasu trwania cyklu sterownika.

[†] Tłumaczenie własne z powodu braku obowiązującego tłumaczenia normy IEC 61131-1, org. ang. „*digitally operating electronic system, designed for use in an industrial environment, which uses a programmable memory for the internal storage of user-oriented instructions for implementing specific functions such as logic, sequencing, timing, counting and arithmetic, to control, through digital or analogue inputs and outputs, various types of machines or processes. Both the PLC and its associated peripherals are designed so that they can be easily integrated into an industrial control system and easily used in all their intended functions*”.

W celu ułatwienia tworzenia aplikacji i poprawy ich niezawodności, sterowniki PLC programowane są przy pomocy specjalizowanych języków programowania – zarówno graficznych, jak i tekstowych. Języki te uniemożliwiają bezpośrednie manipulacje sprzętem (sprzęt konfiguruje się oddzielnie za pomocą odpowiednich funkcji oprogramowania narzędziowego), operacje na wskaźnikach itp. Ograniczając funkcjonalność dostępną dla programisty sterownika PLC, języki te jednocześnie zwiększają niezawodność tworzonego oprogramowania.

W miarę rozwoju technicznego, sterowniki PLC – podobnie jak inne urządzenia komputerowe – udostępniały coraz większą moc obliczeniową za coraz niższą cenę. Spadek cen urządzeń spopularyzował rozproszone systemy sterowania, zbudowane z wielu sterowników PLC połączonych siecią komunikacyjną, gdyż takie rozwiązanie pozwalało zredukować koszty okablowania sygnałowego. Opracowano wiele deterministycznych protokołów komunikacji sieciowej, umożliwiających budowę rozproszonych systemów czasu rzeczywistego, a także różnorodne interfejsy warstwy fizycznej sieci, dostosowane do trudnych warunków przemysłowych.

Nowoczesne sterowniki PLC, obok szybkich procesorów i różnorodnych interfejsów komunikacyjnych, coraz częściej wyposażane są w rozwiązania spotykane wcześniej w systemach komputerowych klasy PC. Przykładem może być rosnąca popularność wbudowanych serwerów FTP i WWW, umożliwiających łatwą wymianę danych ze sterownikami, a nawet wizualizację pracy systemu przez przeglądarkę internetową – tylko za pomocą sterownika PLC. Niektóre sterowniki PLC wyposażane są także w możliwość obsługi przerwań przez aplikację użytkową.

System przerwań jest istotnym elementem każdego systemu komputerowego. Umożliwia on szybką reakcję na zdarzenia zachodzące w systemie (np. zakończenie transferu DMA) bez obciążania procesora koniecznością częstej kontroli stanu danego elementu systemu (ang. *pooling*).

Wykorzystanie przerwań umożliwia poprawę czasu odpowiedzi systemu na zdarzenia zewnętrzne w porównaniu z sekwencyjnym wykonaniem aplikacji, a także poprawę parametrów czasowych komunikacji międzywęzłowej w rozproszonym systemie czasu rzeczywistego.

Procedury obsługi przerwań tworzy się w taki sposób, aby ich wykonanie było jak najkrótsze. Przykładowo, w systemie Windows zadaniem procedury obsługi przerwania jest jedynie zapisanie niezbędnych informacji, potwierdzenie jego przyjęcia i aktywowanie opóźnionej procedury obsługi (ang. DPC, *Deferred Procedure Call*). Dalsze przetwarzanie realizowane jest przez zwykłe mechanizmy wielozadaniowości systemu operacyjnego.

W przypadku sterowników PLC obsługa przerwań jest realizowana za pośrednictwem oprogramowania wbudowanego (ang. *firmware*), które obsługuje zgłoszenia przerwań i wywołuje odpowiednie fragmenty aplikacji użytkowej. Także w tym przypadku, w oficjalnej dokumentacji zaleca się, aby czas wykonania procedur

obsługi przerwań był jak najkrótszy, w efekcie czego wydłużenie czasu wykonania aplikacji sterownika ma być pomijalne.

W rozprawie zaproponowano inne podejście. Założono, że czas wykonania procedur obsługi przerwań może być długi, co skutkuje mierzalnym wpływem na czas cyklu sterownika, ponadto spodziewane jest zgłaszanie kolejnych przerwań podczas wykonywania procedury obsługi przerwania.

Cel pracy

Głównym celem pracy jest opracowanie metod analizy czasowej systemów czasu rzeczywistego wykorzystujących mechanizm przerwań, dla których czas przetwarzania procedur obsługi przerwań nie jest pomijalny. Tym samym odrzucono często przyjmowane założenie, że wpływ czasu obsługi przerwań na działanie systemu informatycznego jest znikomy i może być pominięty podczas analizy czasowej.

W rozprawie zaproponowano model systemu czasu rzeczywistego, w którym system przerwań potraktowano jako instancję pewnego algorytmu szeregowania zadań, a procedury obsługi przerwań jako zadania. Takie podejście umożliwia zastosowanie metod znanych w teorii szeregowania do przeprowadzenia analizy czasowej systemu.

Dokonując dekompozycji oprogramowania aplikacyjnego sterownika na fragmenty – zadania – o niezależnych wymaganiach czasowych, można stworzyć system, w którym parametry czasowe niektórych zadań będą wyraźnie lepsze niż w systemie pracującym bez użycia przerwań, przy zachowaniu ścisłych ograniczeń czasowych.

Mając na celu możliwości praktycznego zastosowania wyników badań, za platformę sprzętową przyjęto sterowniki programowalne PLC, ze względu na ich duże rozpowszechnienie w przemyśle. Nie ogranicza to zastosowania uzyskanych wyników w innego rodzaju systemach czasu rzeczywistego, np. systemach wbudowanych, których projektanci mają z reguły znacznie większą kontrolę nad sprzętem i oprogramowaniem niż w przypadku zastosowania sterowników PLC. W ramach rozprawy nie są rozpatrywane sterowniki PLC o wielozadaniowym modelu programowym, takie jak np. sterowniki spełniające wymagania normy IEC 61131-3.

Należy zauważyć, że większość elementów teorii szeregowania wykorzystanych w niniejszej pracy była znana już w latach dziewięćdziesiątych ubiegłego stulecia. Nie wpływa to negatywnie na aktualność podejmowanego tematu, gdyż celem pracy nie jest rozwój teorii szeregowania, ale jej zastosowanie w analizie czasowej pewnej klasy systemów czasu rzeczywistego – przemysłowych sterowników programowalnych z możliwością obsługi przerwań.

Tezy pracy

Zasadniczą tezą pracy jest teza trzecia. Dla poprawy czytelności wywodu sformułowano i kolejno udowodniono tezy pomocnicze: tezę pierwszą i drugą.

Teza 1

Przy spełnieniu pewnych warunków, celowe jest modelowanie sterowników programowalnych PLC, dla których czas wykonania procedur obsługi przerwań nie jest pomijalny, jako systemów wielozadaniowych.

Teza 2

Wykorzystując podejście wielozadaniowe, możliwa jest taka implementacja aplikacji sterownika PLC, aby czas odpowiedzi na niektóre pobudzenia zewnętrzne był krótszy niż w przypadku sekwencyjnego wykonania aplikacji, przy zachowaniu ścisłych ograniczeń czasowych dla wszystkich zadań.

Teza 3

Wykorzystując podejście wielozadaniowe, możliwa jest taka implementacja aplikacji sterownika PLC będącego węzłem systemu rozproszonego, aby cykl wymiany informacji między węzłami był krótszy niż w przypadku sekwencyjnego wykonania aplikacji, przy dotrzymaniu ścisłych ograniczeń czasowych dla wszystkich zadań oraz samego procesu komunikacji.

Układ pracy

Treść rozprawy ujęto w siedmiu rozdziałach, z których zasadnicze to rozdziały 4, 5 i 6. Rozdział 1 stanowi wprowadzenie do tematyki rozprawy, zawarto w nim także omówienie celu i tez pracy.

W rozdziale 2 umieszczono przegląd najistotniejszych zagadnień dotyczących teorii szeregowania zadań w systemach czasu rzeczywistego. Ze względu na obszerność tej dziedziny wiedzy i ograniczoną objętość rozprawy, wybrano tylko te zagadnienia, których znajomość jest niezbędna dla zrozumienia dalszej części pracy.

Własności sterowników programowalnych PLC omówiono w rozdziale 3. Przedstawiono w nim klasyczny, jednozadaniowy model programowy sterowników PLC wraz ze sposobem jego analizy czasowej oraz skrótowo omówiono wpływ czasu cyklu sterownika na szybkość wymiany informacji w systemie rozproszonym, gdy sterownik PLC jest jego węzłem. Przedstawiono także znane metody redukcji

czasu cyklu sterownika stosowane w modelu jednozadaniowym. Rozdział zakończono omówieniem mechanizmów obsługi przerwań występujących w nowoczesnych sterownikach PLC.

W rozdziale 4 zamieszczono rezultaty analiz teoretycznych. Przedstawiono wielozadaniowy model programowy sterowników PLC, w którym system przerwań traktowany jest jako instancja pewnego algorytmu szeregowania zadań, wraz z metodami analizy czasowej dla jednego i wielu aktywnych źródeł przerwań w systemie. Ponadto w rozdziale tym zaproponowano nieblokujący algorytm wymiany informacji pomiędzy zadaniami, w którym czas przekazania informacji pomiędzy producentem i konsumentem jest ograniczony od góry, umożliwiając jego zastosowanie w systemach o ścisłych ograniczeniach czasowych.

W rozdziale 5 przedstawiono wyniki badań eksperymentalnych działania sterownika PLC jako systemu wielozadaniowego, w którym prace uwalniane są przez wewnętrzne lub zewnętrzne źródła przerwań – odpowiednio przerwania zegara systemowego lub wejść binarnych.

W rozdziale 6 zawarto wyniki pomiarów czasu wymiany informacji między węzłami systemu rozproszonego, gdy obsługa wymian realizowana jest z użyciem modelu wielozadaniowego, na przykładzie sieci typu Master-Slave wykorzystującej protokół Modbus-RTU.

Rozprawę zamyka rozdział 7, w którym zawarto podsumowanie i wnioski.

(*tą stronę celowo pozostawiono pustą*)

Wyniki badań

W rozprawie zaproponowano i przebadano metody analizy czasowej systemów czasu rzeczywistego o ścisłych ograniczeniach czasowych, dla których czas przetwarzania procedur obsługi przerwań nie jest pomijalny.

Analiza czasowa w modelu wielozadaniowym

W podrozdziale 4.2 przedstawiono metodę analizy czasowej sterowników PLC z aktywnym pojedynczym źródłem przerwań. Przypadek taki jest istotny, gdy przerwania wykorzystywane są do przyspieszenia obsługi sieci przemysłowej. Przedstawione i formalnie udowodnione zależności umożliwiają przeprowadzenie kompletnej analizy czasowej pozwalającej projektantowi systemu czasu rzeczywistego stwierdzić, czy zadania są wykonywane przez system w założonych terminach, jak również określić horyzont czasowy pracy systemu, przydatny szczególnie podczas testowania i inspekcji oprogramowania sterownika PLC. Wyprowadzone zależności umożliwiają także ilościowe określenie maksymalnego wpływu obsługi przerwania na długość podstawowego cyklu sterownika, a także szczegółowe prześledzenie wielkości tego wpływu dla każdego z kolejnych cykli pracy sterownika.

W podrozdziale 4.3 zaproponowano model systemu czasu rzeczywistego, w którym system przerwań potraktowano jako instancję pewnego algorytmu szeregowania zadań, a procedury obsługi przerwań jako zadania. Takie podejście umożliwia zastosowanie metod znanych w teorii szeregowania do przeprowadzenia analizy czasowej systemu. Przy odpowiednim przypisaniu priorytetów poszczególnym przerwaniom, system przerwań staje się instancją algorytmu *Rate Monotonic Scheduling* lub jego uogólnienia *Deadline Monotonic Scheduling*. W rozprawie wyprowadzono i formalnie udowodniono zależności umożliwiające prowadzenie analizy czasowej z uwzględnieniem rzeczywistych własności analizowanego systemu, takich jak np. niezerowy czas przełączenia kontekstu zadań.

Istotnym zagadnieniem jest wyznaczanie wpływu obsługi przerwań na czas wykonania podstawowego cyklu sterownika, który można traktować jako zadanie o najniższym priorytecie, przetwarzane w tle (ang. *background processing*).

Wyprowadzone zależności umożliwiają wyznaczanie czasu cyklu wprost (dla systemu z pojedynczym źródłem przerwań), bądź w postaci uwikłanej (dla systemu z wieloma źródłami przerwań). Zależności te wyprowadzano niezależnie, używając różniących się podejść, po czym udowodniono ich równoważność dla przypadku z pojedynczym źródłem przerwań, co jest potwierdzeniem poprawności uzyskanych wyników.

Kolejnym zagadnieniem przeanalizowanym teoretycznie jest deterministyczna komunikacja między niezależnie wykonywanymi zadaniami periodycznymi bez ich blokowania (blokowanie zadań jest niemożliwe z powodu ograniczeń wynikających ze sposobu działania sterowników PLC). W rozprawie zaproponowano prosty algorytm wymiany danych między zadaniami, który gwarantuje przeprowadzenie wymiany w określonym terminie. Należy zauważyć, że zaproponowany algorytm oparto na bardzo ogólnych założeniach (zadania są periodyczne, aktywowane pojedynczym zegarem oraz każda praca podejmuje próbę dostępu do danych), czego skutkiem jest stosunkowo długi termin wymiany danych. Wykorzystując proste środki programistyczne, jak np. podwójne buforowanie danych lub wielokrotny dostęp do bufora w czasie wykonania każdej pracy, można zagwarantować znacząco krótszy termin wymiany danych między współpracującymi zadaniami.

Zależności przedstawione i formalnie udowodnione w podrozdziale 4.4 umożliwiają przeprowadzenie kompletnej analizy czasowej wymiany danych między zadaniami sterownika PLC. Obok analizy spełniania ograniczeń czasowych (analizy szeregowalności) przez zadania wykonywane przez sterownik PLC, analiza czasowa wymiany danych pomiędzy zadaniami stanowi niezbędne uzupełnienie rutynowo wykonywanej analizy przepływu informacji w sieci przemysłowej, łączącej poszczególne sterowniki PLC i komputery nadrzędne w rozproszonym systemie sterującym czasu rzeczywistego.

Badania eksperymentalne

Zgodność opracowanego modelu z doświadczeniem zweryfikowano wykonując trzy eksperymenty. W pierwszym eksperymencie, opisanym w podrozdziale 5.1, przebadano system, w którym źródłem wszystkich przerwań jest zegar systemowy. Celem eksperymentu jest weryfikacja zgodności opracowanego modelu działania jednostki centralnej sterownika programowalnego, w szczególności podsystemu obsługi przerwań, z rzeczywistym urządzeniem, pomijając wpływ układów wejścia/wyjścia. Upraszcza to analizę uzyskanych wyników, gdyż nie występują żadne zaburzenia w zgłaszanych przerwaniach, zatem kolejność uwalnianych prac jest powtarzalna. Jednocześnie jest to przypadek ważny w praktycznych zastosowaniach, gdyż umożliwia prostą implementację przetwarzania wielozadaniowego.

W przeprowadzonym eksperymencie pozytywnie zweryfikowano poprawność opracowanego modelu teoretycznego jednostki centralnej.

W drugim z przeprowadzonych eksperymentów, omówionym w podrozdziale 5.2, zastosowano niezależne, zewnętrzne źródła przerwań, taktowane osobnymi zegarami o nieco różniących się częstotliwościach, dodatkowo zaburzając je przez krótkotrwałe wstrzymywanie generacji zgłoszeń przerwań. Sytuacja taka odwzorowuje złożony system, w którym przerwania aktywowane są przez sygnały zewnętrzne, np. pochodzące z czujników, co powoduje występowanie nieregularności momentów uwolnienia prac (ang. *jitter*).

Celem eksperymentu jest weryfikacja przewidywań teoretycznych dotyczących maksymalnego czasu odpowiedzi systemu widzianego przez jego otoczenie, od momentu wystąpienia sytuacji wymagającej obsługi aż do pojawienia się sygnału wyjściowego. Pomiar czasu odpowiedzi przeprowadzono metoda pośrednią, poprzez zliczanie impulsów pochodzących z generatora wzorcowego za pomocą szybkich liczników. Liczniki bramkowano sygnałem żądania obsługi generowanym asynchronicznie przez źródło zdarzeń oraz wyjściowym sygnałem odpowiedzi na zdarzenie sterownika PLC. Zastosowana metoda pomiaru pozwala wyznaczyć z dużą dokładnością faktyczne czasy odpowiedzi prac od chwili wystąpienia zewnętrznego sygnału pobudzenia uwalniającego wykonanie pracy aż do pojawienia się odpowiedzi na fizycznych wyjściach systemu

W przeprowadzonym eksperymencie pozytywnie zweryfikowano poprawność przewidywań opracowanego modelu teoretycznego i możliwość ich zastosowania do analizy czasowej systemów czasu rzeczywistego wykorzystujących sterowniki programowalne PLC.

Trzeci z eksperymentów, przedstawiony w rozdziale 6, wykonano w celu zweryfikowania możliwości zastosowania opracowanego modelu do analizy czasowej rozproszonego systemu czasu rzeczywistego. W takim systemie, częstość wymian komunikatów między węzłami zależy od parametrów czasowych węzłów systemu, na które ma wpływ m. in. częstość wymian komunikatów. Jako przykład wybrano deterministyczną sieć komunikacyjną typu Master-Slave (Modbus-RTU), której cykl wymian silnie zależy od czasu odpowiedzi abonentów sieci.

Przedstawiona metoda poprawy wydajności wymian w sieciach typu Master-Slave przez ich asynchroniczną obsługę z wykorzystaniem systemu przerwań jest nowym rozwiązaniem, które nie było dotychczas opisywane w literaturze przedmiotu. Pomimo pewnych trudności w implementacji, spowodowanych brakiem wystarczającego wsparcia ze strony producentów sterowników PLC, proponowane podejście wydaje się szczególnie atrakcyjne ze względu na odseparowanie zadań sterowania i komunikacji w aplikacji sterownika, co ułatwia modyfikacje oprogramowania i zmniejsza ryzyko wprowadzenia błędów.

Przeprowadzone eksperymenty wykazały zgodność modelu z doświadczeniem.

Podczas eksperymentów zauważono występowanie wielu zjawisk, które nie zostały udokumentowane przez producenta zastosowanych sterowników PLC, np. czas wykonania fragmentu programu przez procedurę obsługi przerwania (niezależnie od jej priorytetu) jest o ok. 1,8% krótszy niż czas wykonania tego samego fragmentu programu w podstawowym cyklu sterownika. Ponadto zauważono, że niektóre fragmenty dokumentacji technicznej podają błędne wartości lub są niespójne z innymi fragmentami, nawet w obrębie pojedynczej strony dokumentacji. Wskazuje to na konieczność eksperymentalnej weryfikacji zarówno wyznaczanych parametrów modelu jak i uzyskanych wyników analizy czasowej.

Przeprowadzenie analizy czasowej wymaga znajomości parametrów modelu. Ponieważ ich dokładne wyznaczenie może być trudne lub wręcz niemożliwe, projektant systemu może powiększyć koszty obliczeniowe zadań i/lub pomniejszyć okresy i terminy zadań o pewien zapas tworzący margines bezpieczeństwa. Obliczając maksymalne czasy odpowiedzi zadań, z założenia dopuszcza się uzyskanie zawyżonych wyników, jednak zbyt duży margines bezpieczeństwa może prowadzić do błędnej konkluzji „system nie spełnia ograniczeń czasowych" w przypadku, gdy rzeczywisty system je spełnia. Właściwy dobór wielkości marginesu bezpieczeństwa jest zadaniem projektanta systemu sterowania.

W przeprowadzonych eksperymentach położono duży nacisk na dokładność wyznaczenia parametrów modelu, co poprawiło dokładność wyników analizy czasowej. W przypadku pierwszego eksperymentu maksymalne czasy odpowiedzi zadań wyznaczono z dokładnością lepszą niż 0,6%. Dokładność wyników analizy czasowej w drugim eksperymencie jest lepsza niż 0,7% dla zadania o najwyższym priorytecie i lepsza niż 6% w pozostałych przypadkach.

Zgodność modelu z doświadczeniem, dokładność uzyskiwanych wyników oraz łatwość jego zastosowania w analizie czasowej, dowodzą pierwszej tezy rozprawy. Poprawność drugiej tezy rozprawy wynika wprost z równań opracowanego modelu, dowodzą jej także wyniki eksperymentów przedstawione w rozdziale 5. Trzeciej, zasadniczej tezy rozprawy dowodzą analizy i wyniki pomiarów przedstawione w rozdziale 6.

Podsumowanie

Poniżej wymieniono najważniejsze, oryginalne osiągnięcia autora przedstawione w rozprawie.

1. Opracowanie modelu sterowników PLC, w którym system przerwań potraktowano jako instancję algorytmu szeregowania zadań.

2. Opracowanie prostej metody analizy czasowej systemów z pojedynczym źródłem przerwań.

3. Wyznaczenie postaci funkcji żądanego czasu procesora umożliwiających zastosowanie metody analizy żądań czasowych TDA dla systemów z wieloma źródłami przerwań oraz z ograniczoną liczbą priorytetów przerwań.

4. Opracowanie zdeterminowanego czasowo algorytmu komunikacji między niezależnie wykonywanymi zadaniami periodycznymi bez ich blokowania.

5. Eksperymentalne wykazanie użyteczności opracowanych metod dla analizy czasowej rozproszonych systemów czasu rzeczywistego zbudowanych z wykorzystaniem sterowników PLC.

6. Analityczne i eksperymentalne wykazanie możliwości poprawy parametrów czasowych systemu czasu rzeczywistego przez zastosowanie przerwań, zarówno czasów odpowiedzi węzłów systemu na pobudzenie lokalne jak i czasu wymiany informacji w sieci łączącej węzły systemu rozproszonego.

(*this page intentionally left blank*)

Abstract

In computer systems, interrupts are used to improve system response times compared to sequential execution of applications. Inter-node communications in a distributed system can also benefit from asynchronous processing with interrupts. However, interrupts interfere with normal processing. To bind that interference, interrupt service routines are typically designed to execute extremely fast.

Hard real-time systems must be analyzed to verify that strict time constraints are met. For typical hard real-time systems, interrupts are either avoided or assumed to be serviced in negligible time and completely ignored in analysis of time constraints.

The thesis investigates hard real-time systems where interrupt service times are not negligible. The whole interrupt system is modelled as a priority-based, preemptive scheduling algorithm and each interrupt service routine is modelled as a hard real-time task. Such approach enables the use of real-time scheduling theory to determine whether specified time constraints are met.

The research is stimulated by the fact that interrupts are supported by modern Programmable Logic Controllers (PLC) that is widely used computational devices, designed for industrial automation and control applications. Classic PLCs are based on a simple, well-known programming model: a single† task executing a never-ending program loop. Each iteration of the loop – a sweep – consists of distinct, synchronous phases of acquisition of inputs, execution of application program, actualization of outputs and inter-node communication. The synchronous approach simplifies analysis of time constraints.

The PLC runtime does not support multiple tasks, thus all interrupt service routines must execute the whole processing at once using reasonable amount of execution time. Keeping interrupt service time negligible posses hard limits on functionality of interrupt service routine. Such systems would strongly benefit from long-running interrupt service routines that can be analyzed using methods investigated within the thesis. While PLCs are hardware platform used through the thesis, obtained results can be directly applied to other real-time systems, such as embedded systems, whose designers tend to keep more control over hardware and software than PLC application designers.

---

† PLCs that are fully compatible with IEC 61131-3 do support preemptive multitasking, however the exact behaviour is vendor-dependent. Such PLCs are not covered within the thesis.