

## NOWOCZESNE STANOWISKO NAUKOWO-BADAWCZE WYKORZYSTUJĄCE OPROGRAMOWANIE CZASU RZECZYWISTEGO

*Celem referatu jest przedstawienie zarysu możliwości wykorzystania oprogramowania czasu rzeczywistego na bazie komputera klasy PC i narzędzi firmy MathWorks. Omówiono mechanizmy działania pakietu Real-Time Workshop, Real-Time Windows Target oraz RT-CON przy współpracy z kartami pomiarowo-sterującymi RT-DAC4 i Advantech PCL-812PG oraz wyniki testów w odniesieniu do laboratoryjnego modelu robota.*

### MODERN STAND FOR SCIENTIFIC RESEARCH USING REAL-TIME SOFTWARE

*The main purpose of this work was to show possibilities of real-time software based on PC class computer and MathWorks, Inc. tools. Basic mechanisms of Real-Time Workshop, Real-Time Windows Target as well as RT-CON were presented. Specified packages collaborated with data acquisition cards like RT-DAC4 as well as Advantech PCL-812PG and were tested on laboratory robot model.*

### 1. WPROWADZENIE

W bieżącym stuleciu niezwykle cenną stała się możliwość szybkiego i pewnego dostępu do informacji. Nabywanie wiedzy i zdobywanie praktycznych umiejętności jej wykorzystania ma również swoje korzenie w szkolnictwie wyższym. Istotne jest posiadanie przez tego rodzaju placówki odpowiednich stanowisk naukowo-badawczych, oferujących szybkie wdrażanie nowych, ciekawych koncepcji zarówno przez pracowników naukowych jak i przez studentów.

Tradycyjne podejście opracowania nowej technologii, bądź produktu do niedawna wymagało zaangażowania wielu osób (bądź niezależnych zespołów projektowych) zajmujących się sprzętem, oprogramowaniem, testowaniem i analizą uzyskanych wyników. Lawinowo powstawały poważne problemy podczas integracji poszczególnych faz projektu, natomiast sam proces stawał się czasochłonny i zawodny. Dlatego też, uwzględniając uwarunkowania ekonomiczno-przemysłowe, poszukuje się sposobu na skrócenie czasu procesu tworzenia nowego produktu. Niewątpliwie zapewnia to równoczesny rozwój sprzętu komputerowego i oprogramowania.

Jedną z nowych technik jest tzw. **Rapid Prototyping (RP)**. Pozwala ona na testowanie oprogramowania lub algorytmów sterowania we wczesnych etapach rozwoju projektu. Komputer z oprogramowaniem real-time podłączony do obiektu lub jego modelu rzeczywistego z pomocą kart pomiarowo-sterujących emuluje pracę sterownika. [9].

Komputer klasy PC, wyposażony na przykład w oprogramowanie MATLAB, Simulink, Real-Time Workshop, RT-CON oraz kartę pomiarowo-sterującą (np. RT-DAC firmy

INTECO) łączy w sobie wszystkie zalety nowoczesnego stanowiska naukowo-badawczego.

W dalszej części artykułu zostały omówione najważniejsze cechy programów pakietu MATLAB, związanych z prototypowaniem w czasie rzeczywistym oraz podstawowe mechanizmy w nich działające.

## 2. OPROGRAMOWANIE CZASU RZECZYWISTEGO

### 2.1 Real-Time Workshop (RTW)

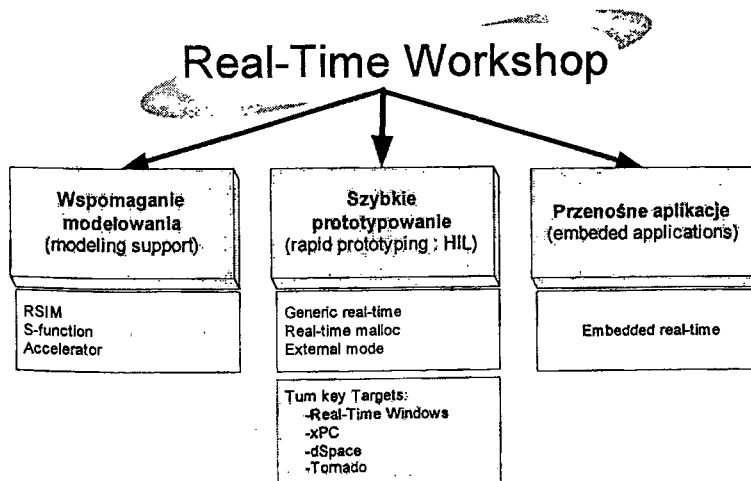
Kluczowym elementem w projektowaniu nowych rozwiązań od ich zamodelowania po wdrożenie, stanowi Real-Time Workshop. Generuje on zoptymalizowany, przenośny kod czasu rzeczywistego na podstawie modeli Simulink'a. Program Real-Time Workshop sprowadza się do konsystencji trzech etapów standardowego projektowania: symulacji, prototypowania modelu i jego implementacji.

Nasuwa się pytanie co wnosi generowanie kodu z programu Simulink?

Na odpowiedź składają się cztery podstawowe korzyści:

- szybsze uruchamianie aplikacji;
- zabezpieczenie własności intelektualnej przy współdzieleniu modeli Simulink'a (chodzi o konwersję na S-funkcje w kodzie C);
- skrócony czas prototypowania;
- implementację projektu bez ręcznego kodowania w języku wysokiego poziomu.

Ostatnia z wymienionych możliwości jest niezwykle ważną, ponieważ eliminuje sposobność wystąpienia błędów kodowania związanych ze żmudnym procesem debugowania, pomijając nakłady czasu poświęconego na tworzenie programu. Dodatkowo po przeprowadzonej modyfikacji modelu, zoptymalizowany kod jest każdorazowo odświeżany. Formaty kodu generowanego przy użyciu RTW przedstawia rys. 1.



Rys.1. Formaty kodu generowanego przy użyciu RTW

Wspomaganie modelowania generuje Rapid Simulation Target (RSIM), pozwalający na tworzenie samodzielnych, wykonywalnych modeli Simulink'a, które mogą być uruchamiane spoza MATLAB'a, wykorzystując jego pliki. Dzięki temu możliwe jest kreowanie wielu „samodzielnych symulatorów” uruchamianych z różnymi parametrami. Z kolei zapis kodu w formie S-funkcji pozwala na użycie utworzonego modelu w postaci komponentów większego systemu i współdzielenie go bez udostępniania detali. Znamienny dla S-funkcji jest fakt braku możliwości modyfikacji zapisanych w niej parametrów modelu [1].

Generacja kodu dla celów szybkiego prototypowania pozwala na testowanie algorytmów sterujących oraz sterowników urządzeń dla rzeczywistych obiektów lub ich modeli fizycznych. Wspieranie technologii RP pozwala na szybką korekcję i poprawę niedociągnięć oraz wad projektowych algorytmów sterujących.

Kolejną z nowoczesnych dróg sprawnego projektowania jest technologia Hardware In The Loop Simulation (HWILS), umożliwiająca usprawnienie algorytmu zaimplementowanego w sterowniku przemysłowym poprzez eksperymenty on-line na symulowanym modelu obiektu [5, 9, 11]. HWILS umożliwia korekcję we wczesnym etapie wszelkich niedociągnięć i wad projektowych. Technika ta wymaga systemu operacyjnego czasu rzeczywistego np. VxWorks Tornado, bądź wykorzystuje ich emulacje (m.in. xPC Target) [7].

Rozwijana i upowszechniana tendencja osadzania oprogramowania w sterownikach PLC i innych urządzeniach, wymuszana jest poprzez tworzenie systemów rozproszonych, RTW oferuje generowanie kodu o takim przeznaczeniu.

Z Real-Time Workshop wiąże się pewna terminologia, dotycząca mechanizmu jego działania. Mianem systemu docelowego (ang. *target*) określa się środowisko: sprzęt lub system operacyjny, na którym będzie uruchamiany kod wygenerowany przy pomocy Real-Time Workshop. Proces tworzenia kodu wyspecjalizowanego dla odpowiedniego środowiska nadzorowany jest przez *system target file*, wzorcowy plik systemu *make (template makefile)* oraz polecenie *make*.

System, na którym używane jest oprogramowanie MATLAB, Simulink, Real-Time Workshop nazywany jest *hostem* (ang. *host*). Przy użyciu odpowiednich narzędzi tworzony jest kod oraz plik wykonywalny, który następnie może być uruchamiany w systemie docelowym (*target*).

RTW wykorzystuje pliki TLC (nazywane także plikami *Target Language Compiler*), do tłumaczenia modelu Simulink'a na kod języka wysokiego poziomu.

Kompilator wykorzystuje dwa rodzaje plików TLC. Pierwszy z nich - *system target file* - opisuje sposób, w jaki ma być tworzony kod dla wybranego systemu docelowego. Drugi rodzaj tworzą - *block target files*, które definiują postać i specyfikę kodu dla poszczególnych bloków modelu Simulink'a.

*Template makefiles* służą do tworzenia pliku wykonywalnego (ang. *executable*) na bazie wygenerowanego kodu. Dokonywane jest to na podstawie wzorcowego skryptu systemu *make*. Jest on w trakcie procesu generacji przepisywany, natomiast uzupełniane i rozwijane są automatycznie te linie kodu, które dotyczą specyfiki modelu.

Jeśli używany jest zewnętrzny kompilator np. MS Visual C++, wówczas wzorcowy plik systemu *make* posiada wymagane przez niego rozszerzenie *tmf*.

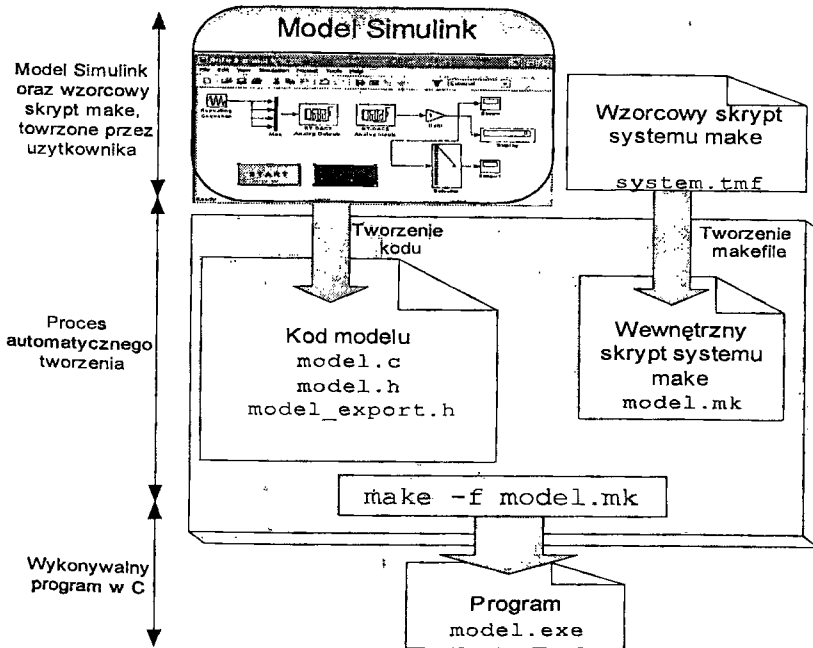
Wysokopoziomowe polecenia M-plików nadzorują proces tworzenia kodu w RTW. Domyślnym poleceniem, używanym przez większość środowisk *target* jest *make\_rtw*.

Polecenie to wywołuje kompilator TLC oraz pozostałe narzędzia jak np. *make*. Sam proces tworzenia kodu składa się z następujących etapów:

- *Make\_rtw* kompiluje diagram blokowy modelu Simulink'a i tworzy plik opisujący ten model: *model.rtw*;
- W dalszej kolejności *make\_rtw* wywołuje Target Language Compiler w celu wygenerowania kodu o specyfice wybranego systemu docelowego (*target*);
- Tworzony jest skrypt *model.mk* na bazie wybranego wzorcowego pliku systemu *make*;
- Wywoływane jest polecenie *make*, które odpowiada za kompilację i linkowanie programu z wygenerowanego wcześniej kodu (odbywa się to według 'przepisu' z *model.mk*).

Parametry symulacji modelu Simulink'a przekładają się bezpośrednio na utworzony kod lub program wykonywalny. Dlatego też istotnym elementem poprzedzającym proces generacji kodu jest poprawne ustawienie parametrów w oknie dialogowym Simulation Parameters programu Simulink.

Schemat procesu tworzenia programu z wykorzystaniem RTW został przedstawiony na rys.2.



Rys. 2 Automatyczne tworzenie programu z wykorzystaniem RTW

Dzięki zintegrowanemu plikowi systemu make budowane są programy, które przyspieszają proces symulacji. W dalszej kolejności są one uruchamiane na kartach pomiarowo-sterujących, wspierających prototypowanie w czasie rzeczywistym [1,6].

## 2.2 Real-Time Windows Target (RTWT)

Kluczowym elementem nowoczesnego stanowiska badawczego jest ostatnio wprowadzony na rynek Real-Time Windows Target [2].

RTWT jest bardzo przydatnym narzędziem dla prototypowania i testowania systemów czasu rzeczywistego, w którym można użyć pojedynczego komputera jako *host* i *target*, wykorzystującego oprogramowanie MATLAB/Simulink do tworzenia modeli z bloków Simulink'a.

Po utworzeniu modelu i przeprowadzeniu symulacji w trybie zwykłym (ang. *normal mode*) istnieje możliwość wygenerowania wykonywalnego kodu, dzięki RTW i kompilatorowi języka C. Następnie można uruchomić aplikację w czasie rzeczywistym lub w trybie zewnętrznym (*external mode*) Simulink'a.

Integracja zewnętrznego trybu Simulink'a oraz Real-Time Windows Target pozwala na użycie modelu Simulink'a jako graficznego interfejsu użytkownika dla celów:

- *Wizualizacji sygnałów* – do wizualizacji przebiegu sygnałów używane są bloki oscyloskopu (*Scope*), w taki sam sposób jak dla zwykłych symulacji;
- *Zmian parametrów* – w trakcie trwania symulacji czasu rzeczywistego można modyfikować parametry systemu, poprzez zmianę parametrów poszczególnych bloków w oknach dialogowych Block Parameters.

Typowe aplikacje Real-Time Windows Target obejmują:

- Sterowanie w czasie rzeczywistym – prototypowanie osprzętu komputerowego, urządzeń systemów sterowania oraz zaawansowanych algorytmów sterujących;
- Symulacje HWILS w czasie rzeczywistym – prototypowanie sterowników przyłączonych do fizycznych obiektów;
- Edukację – nauka koncepcji i procedur dla celów modelowania, symulacji i testowania systemów czasu rzeczywistego oraz analiza projektów.

Oprogramowanie Real-Time Windows Target składa się z wielu istotnych elementów, które pomagają realizować i testować aplikacje czasu rzeczywistego.

Do najistotniejszych należą:

- Real-Time Kernel;
- Real-Time Application;
- Signal Acquisition and Analysis;
- Parametr Tunning;

Real-Time Kernel jest metodą RTW uruchamiania aplikacji w czasie rzeczywistym przez system operacyjny MS Windows. Real-Time Kernel uruchamiany jest na najwyższym poziomie przerwań „0” procesora (ang. *ring zero*) i wykorzystuje wbudowany zegar PC jako podstawowe źródło taktowania czasu. W metodzie tej „podmienione” jądro systemu uzyskuje przerwanie od zegara PC zanim uczyni to system operacyjny (stąd nie można dokonywać wywołań Win32 z poziomu kodu C S-funkcji). Następnie wykorzystywane jest przerwanie do wyzwolenia procesu wykonywania skompilowanego wcześniej modelu. W wyniku tego jądro jest w stanie przydzielić najwyższy priorytet wykonywanej aplikacji [2, 4].

W celu uzyskania dostatecznie wysokiej częstotliwości próbkowania, jądro przestawia (przeprogramowuje) zegar PC na wyższą częstotliwość. Ponieważ zegar jest także do-

myślnym źródłem taktowania dla systemu Windows, jądro wysyła także przerwania do systemu operacyjnego w cyklu zwyczajnym.

Jądro z RTWT komunikuje się ze sprzętem we/wy za pomocą bloków I/O oraz sprawdza konfigurację odpowiedniej karty I/O. Warunkiem uruchomienia aplikacji czasu rzeczywistego jest poprawne zainstalowanie karty.

### 2.3 Oprogramowanie RT-CON

Istotnym elementem nowoczesnego stanowiska badawczego, poza specjalistycznym oprogramowaniem są wielozadaniowe karty pomiarowo-sterujące. Wśród wielu producentów i dostawców na uwagę zasługuje oferta rodzimej firmy INTECO, proponująca kompleksowe rozwiązania systemów pomiarowo-sterujących czasu rzeczywistego [3].

Oprócz kart oferuje ona pakiet RT-CON, który umożliwi integrację oprogramowania MATLAB/Simulink z jądrem czasu rzeczywistego w środowisku Windows 95/98/NT, tworząc narzędzie do automatycznego generowania kodu aplikacji sterujących, wykonywanych przez karty RT-DAC oraz PCL-812PG.

Oprogramowanie RT-CON, rozwijane równoległe z RTWT, odpowiada także za zbieranie danych, ich przetwarzanie oraz wizualizację i sterowanie w czasie rzeczywistym w środowisku Windows.

Wykorzystuje ono własną formę integracji jądra czasu rzeczywistego z uniwersalnymi kartami wejścia/wyjścia i oprogramowaniem MATLAB/Simulink oraz dostępu do przestrzeni adresowej urządzeń wejścia/wyjścia mikroprocesora bezpośrednio z tego środowiska. Kod projektu znajduje się w wykonywalnym pliku *rtw\_dll*. Dane wymagane przez algorytm sterujący lub algorytm akwizycji danych dostarczane są z zewnętrznych urządzeń przez układ akwizycji danych. Na każde przerwanie timera, biblioteka *rtw\_dll* wykonuje wszystkie procedury czasu rzeczywistego, skojarzone z pojedynczym okresem próbkowania. W następnym kroku biblioteka ta odczytuje wejścia karty, oblicza sygnały kontrolne i zapisuje je do rejestrów wyjściowych układu I/O. Jedną z najbardziej istotnych procedur, dołączonych do oprogramowania jest *rtw\_call*, ponieważ zmienia format danych, które są wymieniane pomiędzy *rtw\_dll* a środowiskiem MATLAB/Simulink. Procedurą wykorzystywaną do zarządzania transferem danych między procesami jest Inter-Process Communication (IPC).

Dla właściwej pracy systemu, organizowania i nadzoru przepływu informacji między aplikacją czasu rzeczywistego i środowiskiem MATLAB/Simulink, wykorzystywane są następujące pliki:

- *rtw.m* - okna Graficznego Interfejsu Użytkownika (tryb GUI);
- *ext\_rtw.dll* - odpowiedzialny za komunikację z Simulink'a modelu w zewnętrznym trybie symulacji (*external mode*);
- *rtw\_unload.m* - usuwający bibliotekę *rtw\_dll* z pamięci i kończący przebieg eksperymentu.

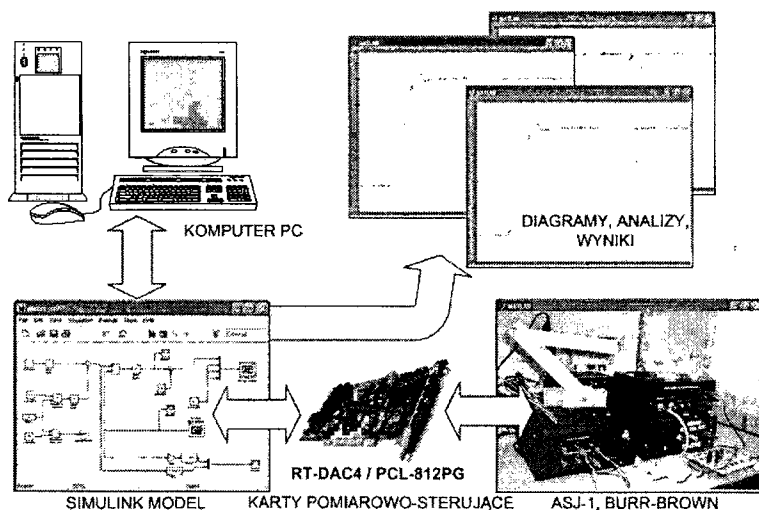
Utworzona biblioteka *rtw\_dll* komunikuje się z kartą wejść/wyjść przy użyciu plików w kodzie C, będących sterownikami urządzeń. Sterownik jest częścią oprogramowania, która działa bezpośrednio na rejestrach karty wejść/wyjść, zarządzając jej operacjami oraz odpowiadając za integrację z zasobami komputera.

INTECO oferuje gotowe przykłady sterowania zestawami laboratoryjnymi: odwróconego wahadła, modelu helikoptera, magnetycznej lewitacji, trójwymiarowej suwnicy lub serwo mechanizmu cyfrowego [3,10].

### 3. PRZYKŁAD STANOWISKA NAUKOWO-BADAWCZEGO CZASU RZECZYWISTEGO

W celu pokazania uniwersalności zastosowań, otwartości oraz obserwacji istotnych podobieństw i różnic, zestawiono stanowisko badawcze, w którym wykorzystano komputer klasy PC oraz karty: RT-DAC4, PCL-812PG w konfiguracji z oprogramowaniem RT-CON, RTWT i MATLAB/Simulink. Realizowało ono ideę *Rapid Prototyping* dla projektowania sterowania położeniem ramion modelu manipulatora według algorytmu PID. Ruch ramion manipulatora był realizowany przez silniki prądu stałego a informacje o ich położeniu odczytywana była z potencjometrów [8].

Konfigurację systemu projektowania i sterowania manipulatora zilustrowano na rys. 3.



Rys. 3. Schemat komputerowego systemu projektowania i sterowania manipulatora

Zamodelowany w Simulink'u i zrealizowany w PC algorytm sterowania miał za zadanie realizację „dobrego” nadążania jednego ze stopni swobody za sygnałem skokowym zadawanym z poziomu GUI.

Podczas przeprowadzonych testów zauważono, że uruchomienie symulacji w trybie *external* zarówno na RT-DAC jak i PCL-812PG sprowadziło się do podmiany bloków wejść/wyjść analogowych odpowiedniej karty (świadczy to o uniwersalności oprogramowania).

Użycie Real Time Windows Target wraz z PCL-812PG pozwoliło na szybsze uruchomienie stanowiska, ponieważ sterowniki do urządzenia I/O są zintegrowane z oprogramowaniem RTWT, natomiast w przypadku użycia RT-DAC4 muszą być jednocześnie zainstalowane z dostarczonego przez producenta nośnika.

W przypadku karty Advantech pewną niedogodność stanowił brak polaryzacji ujemnej dla wyjść analogowych. Na poziomie diagramu Simulink'a należało przesunąć sygnał wymuszający za pomocą dostępnych bloków. Pierwsza z omawianych kart posiada wygodne bipolarne wyjścia analogowe o parametryzowanych wzmocnieniach sygnałów.

Znacznym ograniczeniem karty PCL-812PC jest dostępność tylko dwóch wyjść analogowych. Uniemożliwia to wysterowanie bardziej złożonych obiektów. Dla porównania karta RT-DAC4 oferuje cztery wyjścia analogowe. Z kolei pewną uciążliwością w przypadku RT-DAC4 jest konieczność ręcznego usuwania biblioteki *rtw\_dll* z pamięci operacyjnej po przeprowadzeniu symulacji (polecenie *rtw\_unload*). Uruchomienie kolejnej symulacji bez wykonania tej procedury po zakończeniu poprzedniej symulacji prowadzi do zawieszenia środowiska MATLAB/Simulink.

Istotnym parametrem aplikacji czasu rzeczywistego jest dobór czasu obserwacji eksperymentu, określającego zasięg kontroli i akwizycji danych (ograniczenie wynikają z okresu próbkowania). Poszczególne zadania czasu rzeczywistego muszą być realizowane podczas pojedynczego okresu próbkowania. Zazwyczaj są to: komunikacja z zewnętrznym osprzętem, obliczenia sygnałów sterujących, aktualizacja zawartości bufora danych itp. Stąd intensywne używanie twardego dysku czy urządzeń multimedialnych wpływa na dokładność odmierzanego czasu próbkowania. Należy zauważyć, że okres próbkowania zależy także od:

- szybkości PC, która jest determinowana przez typ procesora i częstotliwość zegara;
- modelu karty wejść/wyjść (I/O board);
- czasu wykonywania kodu modelu, który może wzrastać wraz ze stopniem komplikacji modelu;
- liczby zebranych sygnałów w buforze akwizycji danych.

#### 4. WNIOSKI

Powszechna automatyzacja i informatyzacja wszelkich możliwych dziedzin jest nieunikniona. Istotnym jest fakt, że użytkownik, dzięki omawianym technikom może skupić się na pracy twórczej z zakresu własnej dziedziny nauki, bez konieczności szczegółowego wnikania w tajniki programowania w językach wysokiego poziomu.

Należy zauważyć, że całość oprogramowania: narzędzia i końcowa aplikacja została zainstalowana na pojedynczym PC. Sugeruje to niski koszt platformy przeznaczonej dla rozwoju systemów sterowania w czasie rzeczywistym zarówno na potrzeby naukowe, dydaktyczne jak i przemysłowe.

Niezbitym dowodem szeregu zalet oprogramowania wykorzystywanego w prezentowanym stanowisku są fakty stosowania go przez takich potentatów przemysłowych jak Boeing, który wykorzystał je do budowy systemu lądowania dla samolotu bezpilotowego, czy Lucent Technologies używający narzędzi firmy MathWorks do testowania ITC (Internet Telephone Communications). Wśród przedsiębiorstw stosujących omówione technologie znajdują się również: Motorola, SENER (systemy satelitarne), Sheet Dynamics, Ltd. (niwelacja drgań w samolotach), Teras Instruments czy Toyota (Toyota design for the future) [4].



Rozwój prezentowanej idei stanowisk badawczych jest niewątpliwie przyszłością zarówno opracowań interesujących rozwiązań na skalę przemysłową jak też nauki i dydaktyki.

## 5. LITERATURA

- [1] *Real-TimeWorkshop (ver.4). User's Guide* – MathWorks, Inc, 2000
- [2] *Real-TimeWindows Target (ver.2) User's Guide* – MathWorks, Inc, 2002
- [3] *RT-CON Professional Reference Guide (ver.4.2)* – INTECO, 2002
- [4] Zasoby internetowe: [www.mathworks.com](http://www.mathworks.com)
- [5] *Universal Object-Oriented Modeling for Rapid Prototyping of Embedded Electronic Systems* - Markus Kühl, prezentacja 2001
- [6] *Model-Based Design* - Paul Bernard, prezentacja 2000
- [7] *Automatic Code Generation with Real-Time Workshop* – S. Hirsch, prezentacja 2000
- [8] *System quasi-optymalnego sterowania ruchem manipulatora* – M. Wydra, Praca Dyplomowa Magisterska, Lublin 2002
- [9] *Stanowisko do badań symulacyjnych układów regulacji z wykorzystaniem techniki HWILS*- K.Skura, Z.Smolec, T.Nowak, Materiały konferencyjne Automation '98 Warszawa 1998
- [10] Zasoby internetowe: [www.inteco.cc.pl](http://www.inteco.cc.pl)
- [11] Projektowanie układów sterowania z wykorzystaniem SIMULINK-RTW (Real-Time Workshop), PAK 7/1995