

Symulacyjne badania układu z wirtualnym modelem pojazdu i rzeczywistym podzespołem w pętli

W artykule przedstawiono wymagania i ograniczenia w zakresie budowy modelu dynamiki pojazdu oraz tworzenia oprogramowania dla prowadzenia w czasie rzeczywistym symulacyjnych badań układu składającego się z wirtualnego modelu pojazdu i rzeczywistego podzespołu w pętli. Postęp w zakresie projektowania układów regulacji automatycznej i układów sterowania umożliwia prowadzenie eksperymentu, w którym badany podzespół umieszczany jest na stanowisku symulacyjnym, a oddziaływanie pozostałych elementów pojazdu reprezentowane jest odpowiednio przygotowanym, wirtualnym modelem symulacyjnym. Prowadzona zatem jest symulacja dynamiki układu z wirtualnym modelem pojazdu oraz rzeczywistym podzespołem w pętli.

1. Wprowadzenie

Coraz większa złożoność zespołów funkcjonalnych pojazdu, a także dążenie do skrócenia czasu realizacji projektu powodują, że konieczne jest eksperymentalne badanie najbardziej istotnych zespołów pojazdu, zanim zostanie zbudowany prototyp.

W procesie projektowania, konstruowania i wytwarzania pojazdu istotne znaczenie ma symulacja komputerowa jego modelu oraz eksperymentalne badania prototypu. Matematycznie sformułowany model zawiera aktualnie dostępną wiedzę, opisuje istotę działania i właściwości badanego obiektu. W zależności od celu badania stosowane są różne modele symulacyjne. Model może dotyczyć całego pojazdu, jak i wybranej jego części. Powszechnie stosowanym modelem jak tzw. model „ćwiartki pojazdu”, który umożliwia symulacyjne badania zawieszenia pojazdu. Praktyczne zastosowanie znajdują modele funkcjonalne, których idea polega na opisanu związków między wielkościami wyjściowymi a wielkościami wejściowymi modelowanego układu. Funkcjonalne modele tworzone są z zastosowaniem specjalistycznego oprogramowania, np. MATLAB/Simulink [6]. W symulacyjnych badaniach modeli pojazdów coraz częściej znajdują zastosowanie modele MBS (tzw. Multi-Body Systems). W przeciwieństwie do modeli funkcjonalnych, nie reprezentują one funkcjonalnych właściwości elementów, podzespołów czy pojazdu, ale ich tworzenie związane jest z wykorzystaniem fizycznych właściwości elementów pojazdu i uogólnionych praw fizyki. Dla modeli typu MBS utworzono problemowo zorientowane systemy symulacyjne, takie jak ADAMS, Simpack [1]. Symulacja komputerowa umożliwia przeprowadzenie analizy zależności między wielkościami istotnymi dla uzyskania wymaganych miar oceny bezpieczeństwa ruchu pojazdu oraz komfortu podróżowania.

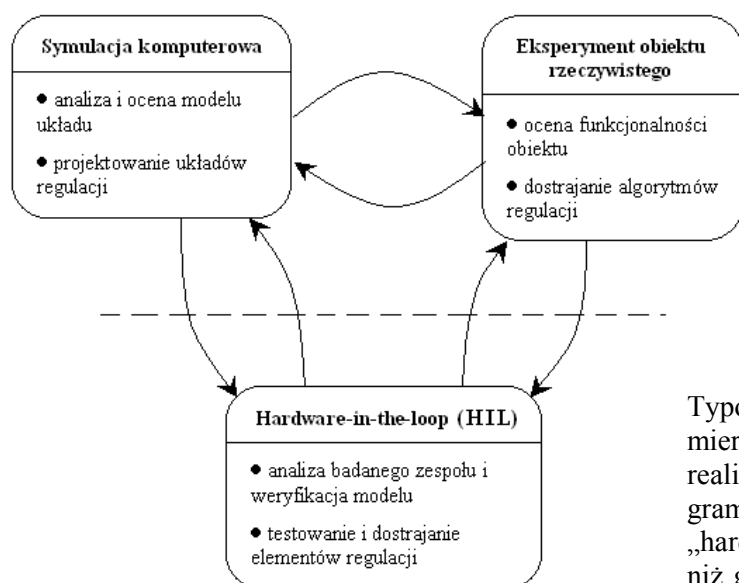
Dzięki zastosowaniu nowoczesnego wsparcia informatycznego (CAD, CACS) tworzone są złożone, wielowymiarowe symulacyjne modele o tysiącach stopni

swobody. Badanie takich modeli ograniczone jest mocą obliczeniową zastosowanych systemów komputerowych. Podstawowym mankamentem stosowania modeli MBS jest brak odniesienia wyników symulacji do rzeczywistych warunków eksploatacji obiektu.

Badanie układów w czasie rzeczywistym jest szczególnie istotne ze względu na coraz szersze stosowanie w pojazdach układów regulacji automatycznej, dla której wymagane jest dokładne w czasie spełnienie warunków, umożliwiających minimalizację występujących uchybów. Dla realizacji symulacji w czasie rzeczywistym, ze względu na ograniczoną moc obliczeniową komputera, konieczne jest odpowiednie przekształcenie modelu. Dla uzyskania tego prowadzona jest linearyzacja modelu układu lub jego podstawowych elementów, dokonywana jest dekompozycja funkcjonalna i strukturalna modelu pojazdu [1]. W symulacyjnych badaniach dynamiki pojazdu prowadzonych w trybie „off-line”, podstawowym kryterium wyboru metody numerycznego całkowania równań różniczkowych ruchu jest dokładność uzyskiwanych wyników. Natomiast w badaniach prowadzonych w czasie rzeczywistym nacisk położony jest na uzyskanie procedur obliczeniowych równań ruchu, których realizacja mieści się w zdeterminowanych ramach czasowych.

Coraz większa złożoność zespołów funkcjonalnych pojazdu, a także dążenie do skrócenia czasu realizacji projektu powodują, że konieczne jest eksperymentalne badanie najbardziej istotnych zespołów pojazdu, zanim zostanie zbudowany prototyp. Badanie zespołu funkcjonalnego na stanowisku symulacyjnym wymaga zdefiniowania obciążeń działających na dany zespół w warunkach realnych stanów pracy. Związane to jest z wprowadzeniem czasochłonnej i kosztownej identyfikacji wymuszeń, działających na pojazd i jego podzespoły w warunkach eksploatacyjnych.

Postęp w zakresie projektowania układów regulacji automatycznej i układów sterowania umożliwił stosowanie innego wariantu prowadzenia eksperymentu – badany podzespół umieszczany jest na stanowisku symulacyjnym, a oddziaływanie pozostałych elementów pojazdu reprezentowane jest odpowiednio przygotowanym, wirtualnym modelem symulacyjnym. Prowadzona zatem jest symulacja dynamiki układu z wirtualnym modelem pojazdu oraz rzeczywistym podzespołem w pętli, zwanym w literaturze HIL (hardware-in-the-loop) [1,2,3,4,5 i 6]. Relacje między różnymi rodzajami modeli symulacyjnych stosowanymi w badaniach pojazdu przedstawiono na rys. 1.

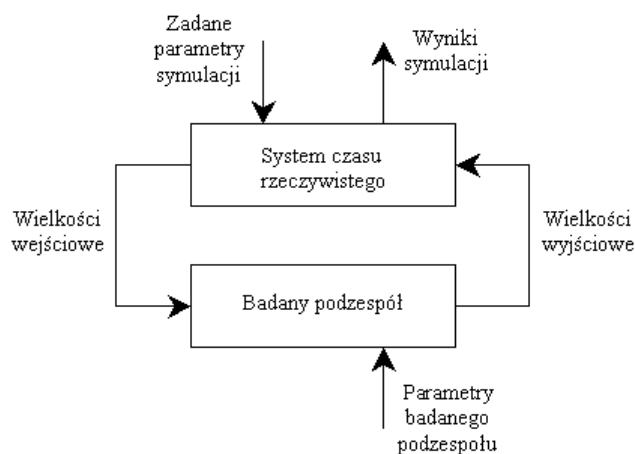


Rys. 1. Schemat struktury badanego układu

2. Model symulacyjny z hardware-in-the-loop (HIL)

Metoda HIL jest stosowana w celu badania podzespołów podczas procesu projektowania pojazdu. Ponieważ nie ma możliwości zbadania elementu zamontowanego w pojeździe, HIL umożliwia przeprowadzenie testów wybranych podzespołów poprzez zastąpienie oddziaływań pozostałej części układu za pomocą modelu symulacyjnego. Zastąpienie pozostałej części pojazdu symulacją komputerową znacznie redukuje złożoność i rozmiar stanowiska badawczego i zwiększa możliwości i szybkość realizacji różnych testów w różnych scenariuszach. Badane elementy są testowane tak, jak by były częścią rzeczywistego pojazdu. Ideowy schemat symulacji realizowanej w pętli HIL przedstawiono na rys. 2.

W symulacji HIL możliwe jest testowanie rzeczywistych podzespołów pojazdów, gdy pozostała część tego pojazdu jest jeszcze na etapie projektowania. Pozwala to na skrócenie czasu realizacji badanego elementu i na obniżenie kosztów procesu projektowania pojazdu.



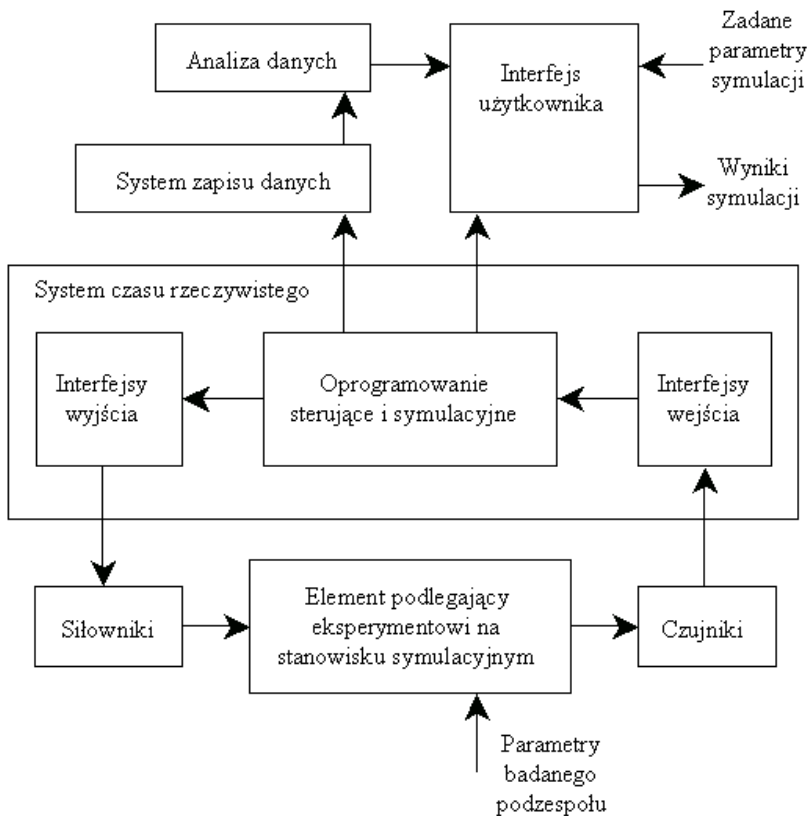
Rys. 2. Ideowy schemat symulacji realizowanej w pętli HIL

Możliwe jest badanie układu przy przyjęciu ekstremalnych warunków eksploatacyjnych dla badanego podzespołu, jak również przyjęcie niebezpiecznych warunków pracy badanego układu, nawet jeśli zdarzają się one tylko w wyjątkowych przypadkach, bez wywoływania ryzyka powstania niebezpiecznych stanów dla ludzi oraz sprzętu.

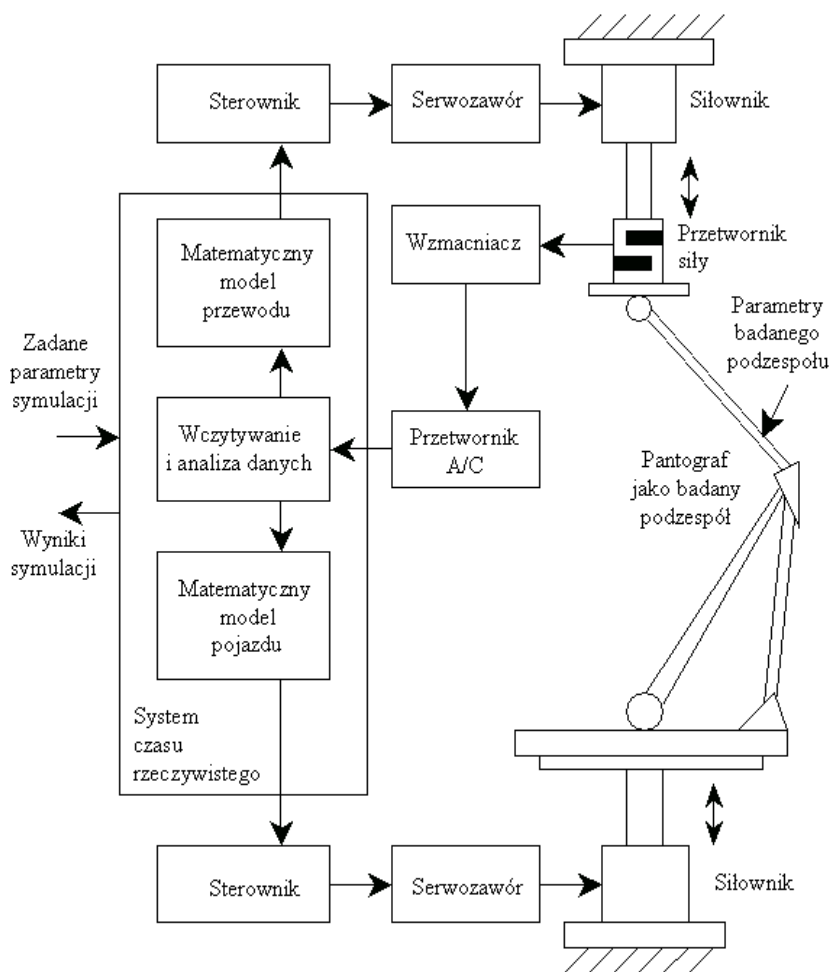
Typowy HIL zawiera czujniki rejestrujące wielkości mierzone na rzeczywistym elemencie oraz siłowniki realizujące fizyczne wyjścia generowane przez program symulacyjny. Proces tworzenia systemu regulacji „hardware-in-the-loop” jest znacznie bardziej złożony niż generowanie modelu do symulacji komputerowej. Układ regulacji HIL działa na systemach operacyjnych czasu rzeczywistego, gwarantujących odpowiedni przydział zasobów urządzeniom wejścia-wyjścia. Musi on również uwzględniać szybkość zmian sterowanego procesu i realizować algorytm regulacji w określonej pętli czasowej. Konieczne jest uwzględnienie interfejsów dla analogowych i cyfrowych sygnałów, jak również wymagań odnośnie do systemów operacyjnych czasu rzeczywistego oraz ograniczeń, jakim podlegają programy działające w ramach systemów operacyjnych czasu rzeczywistego. Schemat realizacji symulacji realizowanej w pętli HIL przedstawiono na rys. 3.

Zastosowanie metody HIL w procesie symulacji układu pantograf-sieć jezdna [4].

Dynamika układu pantograf-sieć jezdna jest badana głównie za pomocą metody symulacji komputerowej. Pantograf jest modelowany jako dyskretny układ liniowy a badana sieć jest w postaci modelu dyskretno-ciągłego. Weryfikacja uzyskanych wyników prowadzona jest na podstawie eksperymentów realizowanych na trasie. Zastosowanie idei „hardware-in-the-loop” umożliwi przeprowadzenie eksperymentu przy przyjęciu pantografu jako elementu rzeczywistego.



Rys. 3. Schemat realizacji symulacji HIL

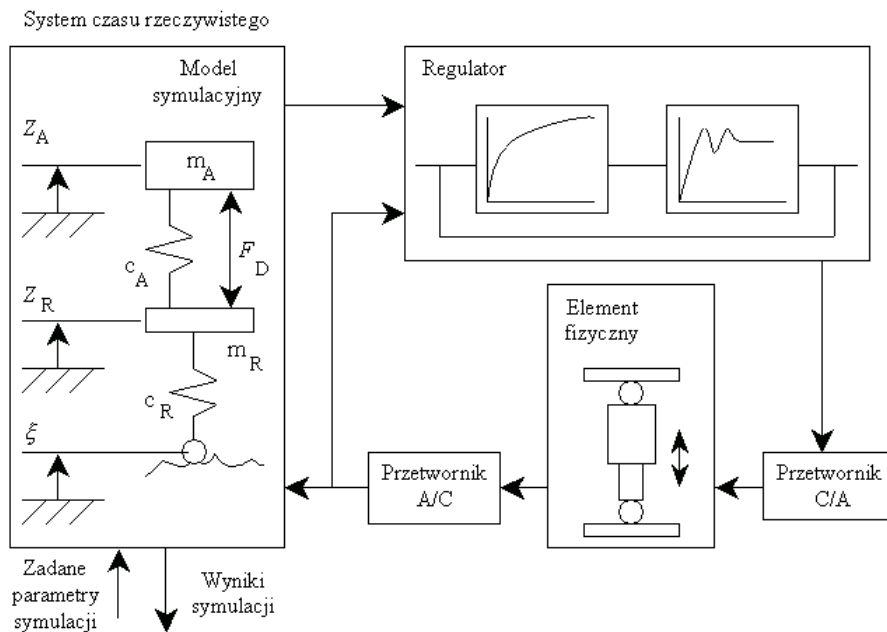


Rys. 4. Schemat badania pantografu z wykorzystaniem HIL

W procesie symulacji przyjęto matematyczny model dla sieci jezdnej oraz dla poruszającego się pojazdu. Przemieszczenia sieci jezdnej spowodowane są przez oddziaływanie siły kontaktowej pomiędzy ślizgaczem pantografu i samej sieci. W modelu symulacyjnym siła kontaktowa jest mierzona jako wielkość wyjściowa i przesyłana do systemu badawczego czasu rzeczywistego, gdzie jest wejściem dla modeli matematycznych. Rozwiązując równanie ruchu uzyskuje się przemieszczenie sieci wyznaczone dla wejściowej siły kontaktowej. Wyznaczone przemieszczenie punktu kontaktowego jest odtwarzane za pomocą układu siłownika sterowanego serwowaworem. Ponieważ tłok siłownika działa na główkę poprzez przetwornik obciążenia, ruch tłoka wywołuje zmiany siły kontaktowej, która z kolei stanowi wejście dla kolejnego kroku obliczeniowego. Schemat badania pantografu z wykorzystaniem symulacji typu HIL przedstawiono na rys. 4.

Zastosowanie metody HIL w procesie symulacji amortyzatora jako elementu rzeczywistego w pętli [3].

W badaniu dynamiki pojazdów do tworzenia matematycznego modelu stosowane są metody analityczne oraz MBS (Multi-Body Systems). Tylko parametry niektórych wielkości fizycznych mogą być mierzone bezpośrednio. Dla parametrów o charakterystykach silnie nieliniowych konieczne jest prowadzenie złożonej identyfikacji. Zastosowanie idei „hardware-in-the-loop” pozwala na badanie układu, w którym elementy o charakterystykach nieliniowych są elementami rzeczywistymi. Wyniki pomiarów elementów rzeczywistych wprowadzane są do modelu w czasie rzeczywistym. Interfejs pomiędzy rzeczywistym elementem a matematycznym modelem realizowany jest za pomocą przetworników A/C oraz C/A. Na rys. 5 przedstawiono schemat badania amortyzatora z zastosowaniem metody „hardware-in-the-loop”. W układzie tym amortyzator jest elementem rzeczywistym, którego wymuszenie generowane jest poprzez układ siłowników hydraulicznych i

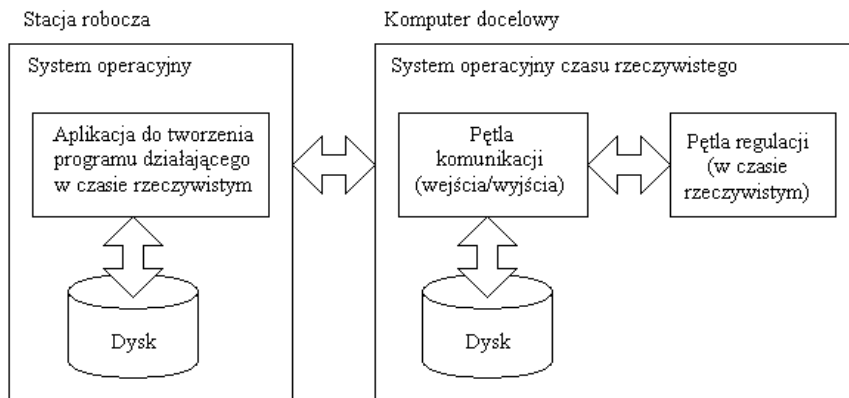


Rys. 5. Schemat badania amortyzatora z wykorzystaniem HIL

regulatora. Siła w amortyzatorze i przemieszczenie punktów mocowania są mierzone bezpośrednio. Wielkości te wprowadzane są jako wejścia do modelu ćwiartki pojazdu poddanego symulacji w czasie rzeczywistym.

3. Komputerowe wspomaganie tworzenia oprogramowania czasu rzeczywistego

Tworzenie oprogramowania dla układu regulacji czasu rzeczywistego jest często najbardziej złożoną i najbardziej czasochłonną częścią procesu projektowania. Wymagana jest staranność w wyborze skutecznego systemu operacyjnego czasu rzeczywistego, na którym jest realizowany program symulacyjny. Ideowy schemat projektowania systemu symulacyjnego czasu rzeczywistego przedstawiono na rys. 6.



Rys. 6. Ideowy schemat projektowania systemu symulacyjnego czasu rzeczywistego

Zastosowanie systemu operacyjnego czasu rzeczywistego wynika z konieczności reagowania w wymaganym czasie na wystąpienie określonego zdarzenia oraz zapewnienia poprawnego działania układu regulacji z pętlą sprzężenia zwrotnego.

Projektując system symulacji czasu rzeczywistego należy uwzględnić następujące aspekty:

- system powinien uwzględniać wszystkie możliwe stany badanego obiektu rzeczywistego
- zmiany nie dotyczące natury badanego układu rzeczywistego nie powinny powodować konieczności wprowadzania modyfikacji w projekcie systemu symulacji
- system powinien uwzględniać możliwość działania bez pętli sprzężenia zwrotnego
- system powinien być skalowalny
- podczas projektowania ważnym elementem jest przewidywany, ostateczny koszt realizacji kompletnego systemu symulacji.

Tworzenie oprogramowania realizującego proces symulacji dynamiki w układzie z elementem rzeczywistym w pętli jest zadaniem złożonym i czasochłonnym. Szczególną uwagę należy zwrócić na wybór efektywnego i skutecznego systemu operacyjnego komputera docelowego, przy czym projektant winien mieć świadomość, że istnieją narzędzia informatyczne obsługujące skomplikowane zadania tworzenia kodu programów czasu rzeczywistego przy uwzględnieniu, że w układzie taktowanie odbywa się z różnymi częstotliwościami a system potrafi zarządzać złożonymi zasobami.

Programy symulacyjne działają na systemach operacyjnych, które można podzielić na dwie kategorie: systemy zdarzeniowe (tzw. event-driven) oraz systemy czasu rzeczywistego. Systemy czasu rzeczywistego, np. QNX, LabVIEW Real-Time oraz xPC, wykonują program w określonych przedziałach czasu. Zasoby komputera niezbędne do wykonania poszczególnych partii kodu są organizowane przez jądro systemu w taki sposób, że jedno zadanie jest zakończone zanim uruchomiona zostanie następna sekcja programu. System czasu rzeczywistego gwarantuje, że kod programu zaplanowany do wykonania w danym momencie czasowym zostanie faktycznie wywołany oraz, że wywołane zdarzenie zostanie zrealizowane w określonym czasie. Działanie zdarzeniowych systemów operacyjnych, np. Microsoft Windows XP, Linux czy BSD, polega na reagowaniu na zdarzenia zewnętrzne lub wewnętrzne bez gwarancji ich wykonania w określonym czasie. Teoretycznie możliwe jest wykorzystanie systemów zdarzeniowych jako pseudo-systemów czasu rzeczywistego (tzw. soft real-time) za pomocą wykorzystania przerwania sprzętowych i programowych, co powinno zagwarantować wykonanie procedur przerwania w określonych, stałych odstępach czasowych. Jednakże zastosowanie takich systemów w praktyce wiąże się z dużą niepewnością faktycznego zagwarantowania realizacji przerwania w określonych odstępach czasowych.

Ponieważ zdarzeniowe systemy operacyjne są najbardziej rozpowszechnione i udostępniają zaawansowane narzędzia programistyczne, istnieje duża pokusa by zastosować taki system w realizacji sterowania procesami czasu rzeczywistego. Podejście takie jest uzasadnione wyłącznie w sytuacjach, w których sterowany obiekt ma dużą inercję, to znaczy nie wymaga sterowania z dużą częstotliwością.

Tworzenie systemów sterowania czasu rzeczywistego bez pomocy wspomagających narzędzi wykorzystujących rozbudowany interfejs graficzny użytkownika jest niezmiernie trudnym zadaniem. Projektanci systemów czasu rzeczywistego, tworzący je bezpośrednio w językach programowania niskiego poziomu, np. ANSI C, łatwo mogą zagubić się w tysiącach linii kodu źródłowego i stracić z pola widzenia główne aspekty projektowanego układu.

Dzięki zastosowaniu narzędzi wspomagających tworzenie systemów czasu rzeczywistego, projektowanie zostało znacznie uproszczone i pozwoliło na zajęcie się głównymi aspektami funkcjonalnymi systemu, a nie problemami implementacyjnymi. Przykładowo, MATLAB/Simulink wspomaga projektowanie systemów czasu rzeczywistego za pomocą dwóch pakietów: Real-Time Workshop oraz xPC Target [6].

Pierwszy z nich generuje i uruchamia samodzielny kod programu napisanego w języku ANCI C służący do rozwijania i testowania algorytmów modelowanych w pakiecie Simulink. Kod wynikowy może zostać również użyty w różnych aplikacjach czasu rze-

czywistego. Pozwala interaktywnie monitorować i dostrajać działanie wygenerowanego kodu, używając w tym celu zaimplementowanych funkcjonalnych bloków pakietu MATLAB/Simulink. Podstawowe możliwości pakietu Real-Time Workshop obejmują:

- tworzenie kodu programu w języku ANSI C oraz plików wykonywalnych dla modeli dyskretnych, ciągłych i hybrydowych
- tworzenie kodu programu dla systemów taktowanych sygnałem pochodzącym z jednego zegara, kilku zegarów (o różnych częstotliwościach) oraz systemów asynchronicznych
- obsługiwanie jedno- i wielozadaniowych systemów operacyjnych oraz środowisk bez systemu operacyjnego
- optymalizację kodu źródłowego programów w celu skrócenia czasu ich wykonywania.

Drugi pakiet stanowi wysokiej jakości środowisko implementujące modele systemów wygenerowane za pomocą pakietu Simulink w fizycznych systemach, kompatybilnych z architekturą komputerów PC. Pakiet xPC Target umożliwia podłączenie interfejsów wejściowych i wyjściowych, generuje automatycznie kod wynikowy programów wykorzystując pakiet Real-Time Workshop i umożliwia przesłanie programu wynikowego do innej stacji PC, na której zainstalowany jest system operacyjny czasu rzeczywistego. Podstawowe możliwości pakietu xPC Target obejmują:

- uruchamianie aplikacji generowanych przez modele Simulink, używając jądra systemu czasu rzeczywistego na dowolnym komputerze klasy PC
- uzyskiwanie próbek o częstotliwości próbkowania do 100 kHz (maksymalna częstotliwość próbkowania zależy od konkretnego rozwiązania sprzętowego)
- obsługę do 250 interfejsów wejścia-wyjścia
- rejestrowanie sygnałów na źródłowej i docelowej stacji roboczej
- udostępnianie zintegrowanego interfejsu użytkownika do zarządzania (poprzez interfejs RS232 lub za pomocą protokołu TCP/IP) aplikacjami wykonywanymi zdalnie na docelowych stacjach roboczych.

Schemat procedury tworzenia programu czasu rzeczywistego, działającego na systemie docelowym przedstawiony został na rys. 7. Komunikacja pomiędzy poszczególnymi modułami odbywa się za pomocą plików o ściśle zdefiniowanych formatach.

Procedura rozpoczyna się utworzeniem – na podstawie specyfikacji systemu docelowego – graficznej reprezentacji systemu czasu rzeczywistego. Przykładowym oprogramowaniem działającym na stacji roboczej, służącym do tworzenia takiego modelu, jest MATLAB Simulink Real-Time Workshop [6] czy

LabVIEW Real-Time [5].

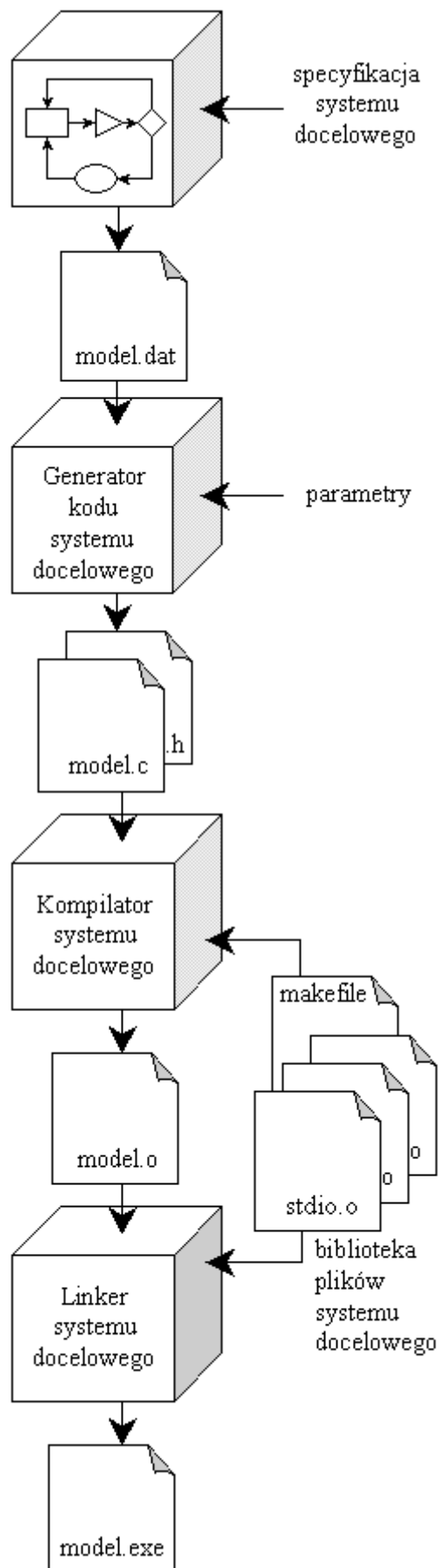
Projekt zapisywany jest do pliku w formacie zdefiniowanym przez producenta oprogramowania i następnie podawany na wejście generatora kodu docelowego systemu. Najczęściej program przeznaczony dla systemu docelowego czasu rzeczywistego generowany jest w języku ANSI C lub C++. Są to języki programowania dość niskiego poziomu, które nie powodują wprowadzenia dużych opóźnień w systemie.

Wynikowe pliki można podzielić na dwie kategorie: pliki z rozszerzeniem „c” (ANSI C) lub „cpp” (C++) zawierające kod źródłowy programu oraz pliki z rozszerzeniem „h”, będące nagłówkami zawierającymi definicje zmiennych globalnych oraz funkcji. Niektóre programy wspomagające tworzenie systemów czasu rzeczywistego tworzą również plik „makefile”, zawierający informacje niezbędne podczas procesu kompilacji.

W celu uzyskania pliku wykonywalnego (oznaczonego na rys. 7 jako „modul.exe”), pliki źródłowe w języku C należy poddać procesowi kompilacji i linkowania. Kompilator i linker działające na stacji roboczej są dość specyficzne, gdyż program wynikowy nie jest przeznaczony do wykonywania na tym samym systemie operacyjnym lecz na systemie operacyjnym czasu rzeczywistego. Alternatywnym rozwiązaniem jest transfer plików źródłowych zapisanych w języku C do systemu docelowego, a następnie wykonanie kompilacji i linkowania już na systemie docelowym.

4. Podsumowanie

Badanie układów w czasie rzeczywistym jest szczególnie istotne ze względu na coraz szersze stosowanie w pojazdach układów regulacji automatycznej, dla której wymagane jest dokładne w czasie spełnienie warunków umożliwiających minimalizację występujących uchybów. W symulacji HIL przeprowadzane jest testowanie rzeczywistych podzespołów pojazdów, gdy pozostała część tego pojazdu jest jeszcze na etapie projektowania. Możliwe jest badanie układu przy przyjęciu ekstremalnych warunków eksploatacyjnych dla badanego podzespołu, jak również przyjęcie niebezpiecznych warunków pracy badanego układu bez wywoływania ryzyka powstania niebezpiecznych stanów dla ludzi oraz sprzętu. Tworzenie oprogramowania realizującego proces symulacji dynamiki w układzie z rzeczywistym elementem w pętli jest zadaniem złożonym i czasochłonnym. Szczególną uwagę należy zwrócić na wybór efektywnego i skutecznego systemu operacyjnego komputera docelowego, przy czym projektant winien mieć świadomość, że istnieją narzędzia informatyczne obsługujące skomplikowane zadania tworzenia kodu programów czasu rzeczywistego przy uwzględnieniu, że w układzie taktowanie odbywa się z różnymi częstotliwościami a system potrafi zarządzać złożonymi zasobami.



Rys. 7. Proces tworzenia programów symulacyjnych czasu rzeczywistego

Przedstawione w artykule wymagania i ograniczenia w zakresie budowy modelu dynamiki pojazdu oraz tworzenia oprogramowania dla prowadzenia w czasie rzeczywistym symulacyjnych badań układu składającego się z wirtualnego modelu pojazdu i rzeczywistego podzespołu w pętli stanowią pierwszy etap przygotowania programu przeprowadzania eksperymentu symulacyjnego z zastosowaniem HIL.

Literatura

- [1] *Eichberger A., Rulka W.: Process Save Reduction by Marco Joint Approach: The Key to Real Time and Efficient Vehicle Simulation, Vehicle System Dynamics, Vol. 41, 2004.*
- [2] *Hanselmann H.: Beschleunigte Mechatronik-Entwicklung durch Paid Control Prototyping und Hardware-in-the-Loop-Simulation, Automatisierungstechnik, Vol. 46, 1998.*
- [3] *Schiehlen W., Hu B.: Spectral simulation and shock absorber identification, International Journal of Non-Linear Mechanics, Vol. 38, 2003.*
- [4] *Zhang W., Mei G., Wu X., Shen Z.: Hybrid Simulation of Dynamics for the Pantograph-Catenary System, Vehicle System Dynamics, Vol. 38, 2002.*
- [5] *www.ni.com*
- [6] *www.mathworks.com*