Politechnika Wrocławska

Prototypowanie systemów sterowania

Prowadzący:

dr hab. inż. Mateusz Dybkowski, prof. uczelni

Opracował:

mgr inż. Szymon Bednarz, dr hab. inż. Mateusz Dybkowski, prof. uczelni

Wrocław 2020

Laboratorium nr 1

Zajęcia wprowadzające. Prototypowanie z użyciem modeli.

1. Wprowadzenie

Celem laboratorium jest zapoznanie się z metodą *Model-Based Desing* mającą zastosowanie w procesie prototypowania systemów sterowania.

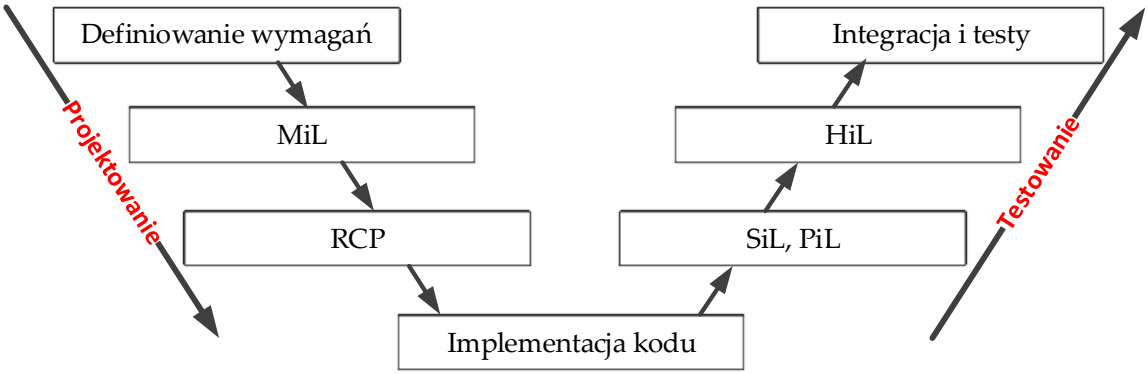
2. Prototypowanie systemów sterowania z wykorzystaniem modeli

Istnieje kilka metod prototypowania systemów sterowania. Klasyczne podejście zakłada, że zdefiniowana funkcjonalność układu sterowania jest implementowana przez programistów na docelową platformę sprzętową w sposób manualny, następnie po wstępnej weryfikacji/testowaniu poprawności działania regulatora, analizuje się jego działanie bezpośrednio w połączeniu z docelowym obiektem sterowania. Rozwiązanie to niesie za sobą szereg komplikacji. Przede wszystkim istnieje ryzyko uszkodzenia sprzętu, co jest również niebezpieczne dla personelu obsługującego. Dodatkową wadą, jest fakt, że integracja regulatora z obiektem następuję dopiero w końcowym etapie projektu. Zatem wykrycie ewentualnych niezgodności może nastąpić dopiero w ostatnim etapie. Każdy wykryty błąd wymaga zmian w specyfikacji/dokumentacji/implementacji co zwiększa nakłady czasowe i finansowe. Ponadto rozwiązanie to jest czasochłonne ze względu na manualną implementację algorytmu sterowania przez programistów.

Obecnie coraz częściej w celu usprawnienia procesu prototypowania stosuje się metodę MBD (ang. *Model-Based Design*) czyli metodę wykorzystującą modele regulatorów oraz obiektów sterowania. Modele obiektów sterowania odwzorowują działanie obiektów rzeczywistych (z określoną dokładnością), natomiast modele regulatorów tworzone są na bazie zakładanej funkcjonalności układu regulacji. Następnie utworzone modele są symulowane (obliczane) i analizowane z zastosowaniem dedykowanego oprogramowania. Obecnie na rynku istnieje wiele środowisk umożliwiających symulowanie różnorodnych układów fizycznych pracujących w układach regulacji dzięki czemu możliwa jest szczegółowa analiza ich zachowania dla szerokiej gamy scenariuszy testowych. Badania symulacyjne pozwalają ponadto na szybszą diagnostykę błędów i ich eliminację już na początkowym etapie realizacji projektu co przekłada się bezpośrednio na skrócenie czasu realizacji projektu i jego kosztów. Wprowadzanie modyfikacji oraz sprawdzanie nowych rozwiązań również nie stanowi dużego problemu. Wykorzystanie modeli minimalizuje także ryzyko uszkodzenia sprzętu. Ponadto środowiska symulacyjne zawierają szereg dodatkowych narzędzi programowanych usprawniających proces projektowania i analizy układów regulacji, a także do automatycznego generowania kodu na platformy sprzętowe.

Metoda MBD często stosowana jest w celu poprawy efektywności procesu projektowania systemu sterowania opartego na tzw. cyklu V (ogólna metodologia

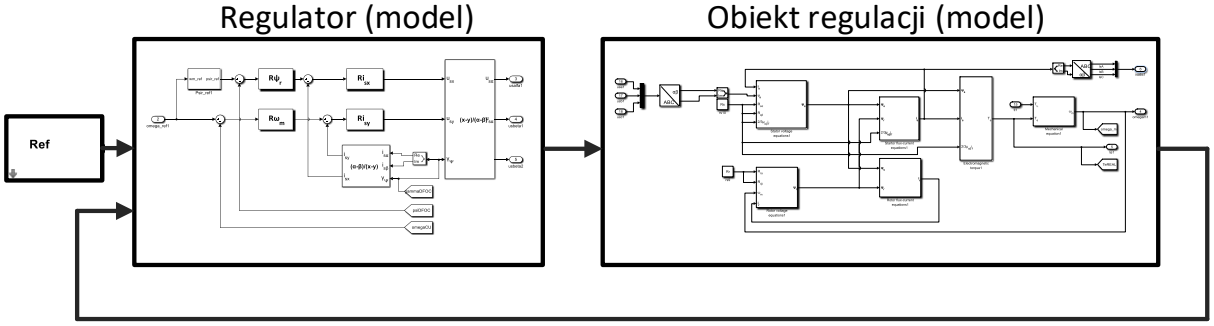
zarządzania procesem prototypowania). Schemat poglądowy procesu prototypowania z zastosowaniem modeli przedstawiono na rys. 1.



Rys. 1. Cykl V dla procesu prototypowania układów sterowania metodą MBD

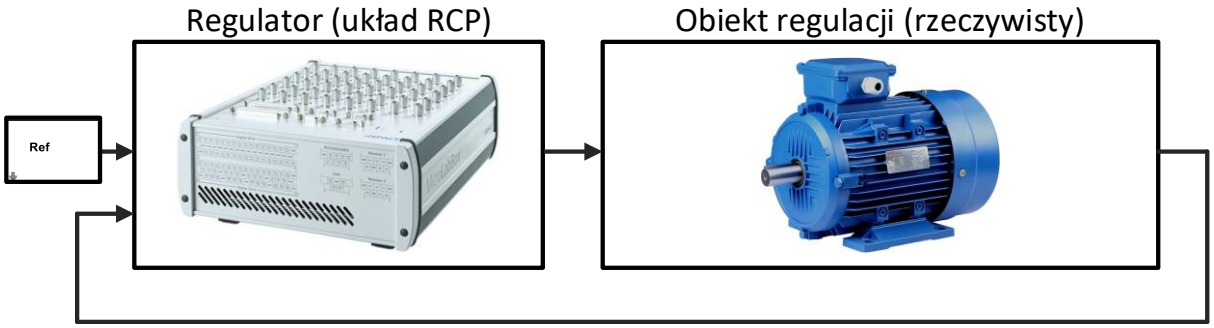
Proces jest podzielony na dwa główne etapy: projektowanie i testowanie, w których z kolei można wyróżnić bardziej szczegółowe:

I. **Definiowanie wymagań** – określenie funkcjonalności systemu.
II. **MiL** (*Model-in-the-Loop*) – tworzenie modelu symulacyjnego układu sterowania obejmującego regulator, obiekt regulacji oraz strukturę układu. Następnie przeprowadzane są symulacje komputerowe w celu analizy funkcjonalności układu regulacji i jego podsystemów dla różnych warunków testowych. Podczas tego etapu żaden z podsystemów układu regulacji nie działa w czasie rzeczywistym.



Rys. 2. Idea prototypowania Model-in-the-Loop

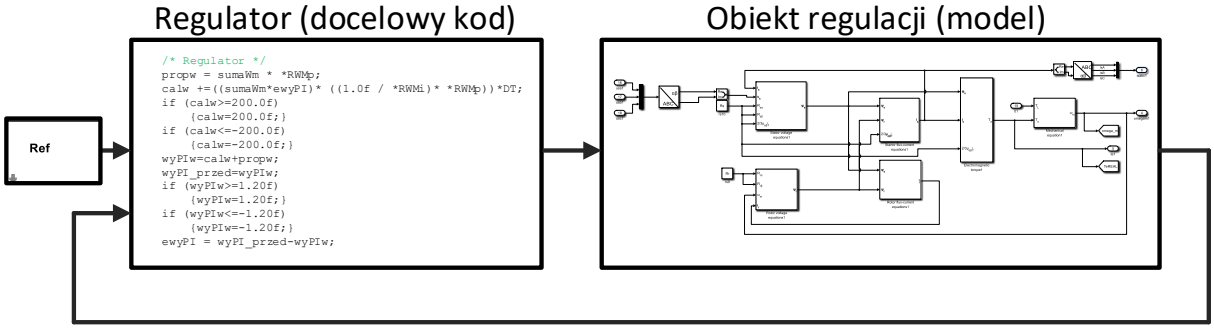
III. **RCP** (*Rapid Control Prototyping)* – prototypowanie układu sterowania na obiekcie rzeczywistym z zastosowaniem platformy sprzętowej o dużej mocy obliczeniowej jako sterownik. Na tym etapie dokonuje się weryfikacji analizy symulacyjnej w czasie rzeczywistym oraz wprowadza się ewentualne zmiany do algorytmu sterowania.



Rys. 3. Idea szybkiego prototypowania

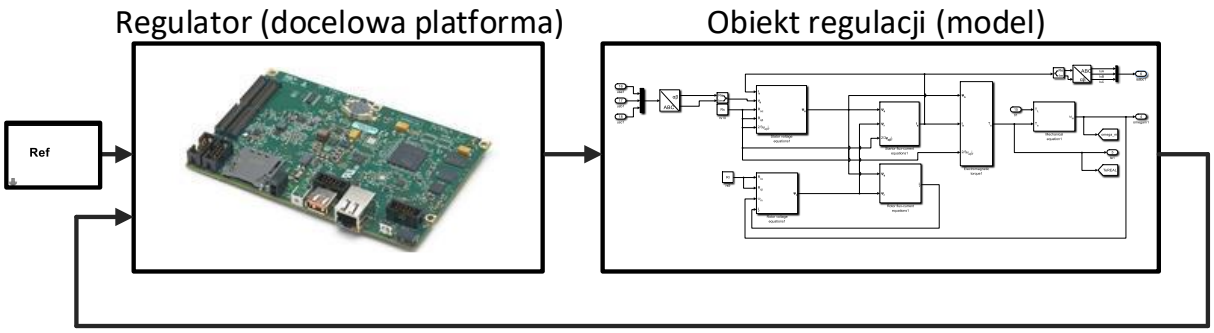
IV. **Implementacja kodu -** implementacja algorytmu sterowania w postaci programu w języku docelowym dla wybranej platformy sprzętowej realizującej zadanie sterownika. Zadanie to najczęściej jest realizowane poprzez wbudowane narzędzia do automatycznego generowania kodu z poziomu środowiska symulacyjnego. Etap ten wymaga utworzenia modelu implementacyjnego sterownika, tzn. zoptymalizowanego do takiej postaci, która spełnia zakładaną funkcjonalność oraz uwzględnia specyfikę docelowej platformy sprzętowej.

V. **SiL** (*Software-in-the-Loop*) - testowanie algorytmu sterowania poprzez symulację wygenerowanego kodu w środowisku symulacyjnym. W tym przypadku model (implementacyjny) regulatora zastąpiony jest kodem dla docelowej platformy sprzętowej wykorzystując odpowiednie narzędzia wbudowane. Natomiast model obiektu pozostaje taki sam jak dla MiL. Etap ten pozwala weryfikować poprawność funkcjonalną generowanego kodu.



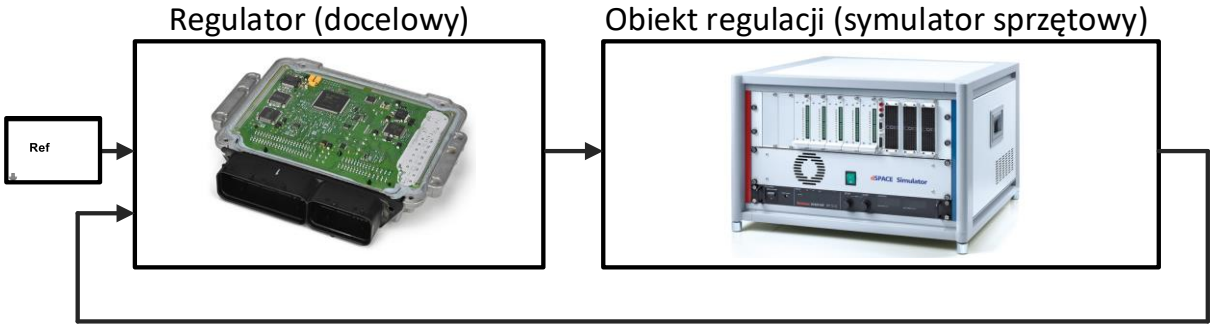
Rys. 3. Idea testów Software-in-the-Loop

VI. **PiL** (*Processor-in-the-Loop*) – testowanie algorytmu sterowania poprzez symulację wygenerowanego kodu na zestawie uruchomieniowym z docelowym procesorem. Regulator stanowi platforma sprzętowa z zaimplementowanym algorytmem regulacji, natomiast obiekt regulacji pozostaje w takiej samej postaci jak dla MiL i SiL. Testowanie polega na co-symulacji między rzeczywistym urządzeniem a środowiskiem symulacyjnym. Urządzenie może komunikować się z komputerem wykorzystując jeden ze standardowych interfejsów (np. USB, Ethernet). Etap ten pozwala przede wszystkim zweryfikować poprawność obliczeń numerycznych przeprowadzanych na procesorze, a także określić czas wykonywania poszczególnych operacji (na bazie automatycznie generowanych raportów).



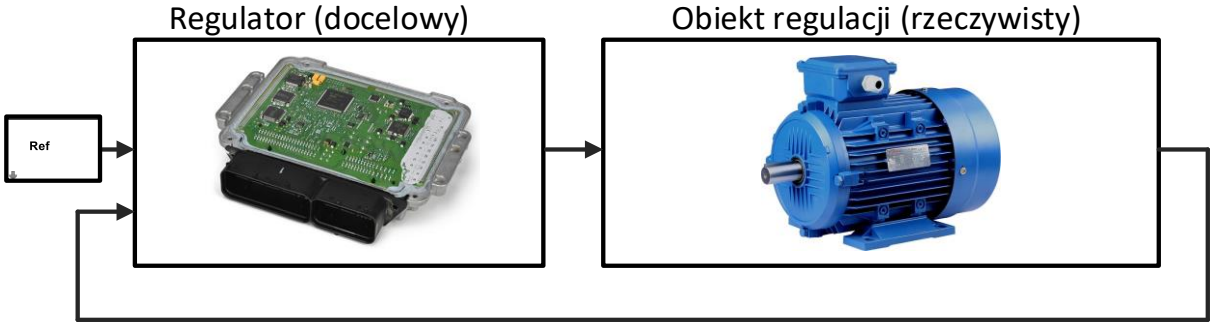
Rys. 4. Idea testów Processor-in-the-Loop

VII. **HiL** (*Hardware-in-the-Loop*) – testowanie algorytmu sterowania na zestawie z docelowym sterownikiem (utworzonym na bazie testowanego wcześniej procesora, rozszerzonym o dodatkowe komponenty sprzętowe oraz programowe). Obiekt sterowania symulowany jest przez platformę sprzętową w czasie rzeczywistym. Regulator połączony jest z symulatorem sprzętowym poprzez odpowiednie układy I/O dlatego pracuje tak, jakby został połączony z obiektem rzeczywistym. Testy HiL pozwalają na przetestowanie pełnej funkcjonalności układu, dla różnych scenariuszy (w tym sytuacji awaryjnych) których wykonanie w testach z obiektem rzeczywistym może być niebezpieczne i kosztowne.



Rys. 5. Idea testów Hardware-in-the-Loop

VIII. **Integracja i testy –** weryfikacja i testowanie założonej funkcjonalności sterownika na obiekcie rzeczywistym.



Rys. 6. Idea testów sterownika na obiekcie rzeczywistym

Literatura:

[1] R. Aarenstrup, *Managing Model-Based Design*, The MathWorks, 2015. Online: https://www.mathworks.com/campaigns/offers/managing-model-based-design.html

[2] P. Sarhadi, S. Yousefour, *State of the art: hardware in the loop modeling and simulation with its applications in design, development and implementation of system and control software*, Internation Journal of Dynamics Control, 2015. Online: https://link.springer.com/article/10.1007/s40435-014-0108-3

[3] H. Arima, *Latest trends in control system development using the model-based development (MBD) method and activity of Skill Management Association*, 2013. online: https://www.ipa.go.jp/files/000034553.pdf

[4] N. Srinivas, N. Panditi, S. Schmidt, R. Garrelfs, *MIL/SIL/PIL Approach: A new paradigm in Model Based Development*, 2014. Online: https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/solutions/automotive/files/in-expo-2014/mil-sil-pil-a-new-paradigm-in-model-based-development.pdf

[5] https://www.mathworks.com/help/physmod/simscape/ug/what-is-hardware-in-the-loop-simulation.html

[6] https://www.dspace.com/en/pub/home/products/sw/pcgs/targetli/tl_verification_validation.cfm