

SPIS TREŚCI

1. WSTĘP	11
1.1. Cel pracy	12
1.2. Zakres pracy	13
1.3. Teza naukowa	18
2. KOMPUTEROWE WSPOMAGANIE PROJEKTOWANIA I PRODUKCJI	19
2.1. Systemy CAD/CAM	21
2.2. PDM - system zarządzania danymi projektowymi i produkcyjnymi	22
2.3. Mechatronika.....	24
2.4. Projektowanie systemów mechatronicznych	27
2.4.1. Potrzeba komputerowego wspomaganie w projektowaniu systemów mechatronicznych	28
2.4.2. Produkcja jednostkowa i masowa	29
2.5. Podsumowanie rozdziału	30
3. PROJEKTOWANIE JAKO CZĘŚĆ CYKLU ŻYCIA PRODUKTU	32
3.1. Tradycyjny sposób projektowania wyrobów	32
3.2. Mechatroniczne podejście do projektowania wyrobów	34
3.3. Zarządzanie projektem i etapy projektowania.....	38
3.3.1. Inicjowanie projektu, specyfikacja i analiza wymagań.....	38
3.3.2. Faza projektowania wstępnego	40
3.3.3. Projektowanie szczegółowe podsystemów	41
3.3.4. Szybkie prototypowanie i symulacja HiL układu sterowania	44
3.3.5. Faza implementacji (projektowanie fizyczne)	45
3.3.6. Konserwacja, modernizacja oraz utylizacja wyrobu.....	45
3.4. Podsumowanie rozdziału	45
4. WYKORZYSTANIE UML W PROJEKTOWANIU MECHATRONICZNYM	48
4.1. Język UML.....	49
4.2. Przykład metodyki organizacji prac projektowych	51

4.2.1. Kolejność i sposób tworzenia diagramów UML.....	52
4.2.2. Diagram przypadków użycia	54
4.2.3. Scenariusz	56
4.2.4. Diagram architektury systemu	57
4.2.5. Obiekty i klasy	57
4.3. Diagramy dynamiczne w UML	60
4.3.1. Diagram sekwencyjny.....	60
4.3.2. Diagram współpracy	61
4.3.3. Diagram stanów	62
4.3.4. Diagram aktywności (czynności).....	64
4.4. Przykłady wykorzystania diagramów UML	64
4.4.1. Specyfikowanie wymagań i określenie architektury systemu.....	64
4.4.2. Określenie funkcjonalności systemu.....	71
4.5. Wybrane pakiety CASE wykorzystujące język uml	73
4.5.1. Rational Suite Enterprise	74
4.5.2. Rational Rose RealTime	74
4.5.3. Artisan Real Time Studio.....	75
4.6. Podsumowanie rozdziału	76
5. OPROGRAMOWANIE WSPOMAGAJĄCE MODELOWANIE I SYMULACJĘ	79
5.1. SIMULINK	79
5.1.1. MATLAB	80
5.1.2. Biblioteki bloków Blockssets	82
5.1.3. Biblioteka MSL (Mechatronics Simulink Library).....	83
5.1.4. Robot Control Library.....	84
5.1.5. SimMechanics.....	84
5.2. STATEFLOW	85
5.3. Generowanie kodu czasu rzeczywistego z użyciem RTW	86
5.4. Modelica	87
5.4.1. Modelica jako język modelowania graficznego i obiektowego	89
5.4.2. Tworzenie klas modeli.....	91
5.4.3. Biblioteka standardowa i biblioteka dodatkowa modeli	97
5.5. Dymola	101
5.5.1. Moduł Dymosim – Dynamic Model Simulator.....	102
5.5.2. Moduł Dymoview	103
5.6. Saber – symulator dla systemów elektry-cznych, elektronicznych, mechanicznych, hydraulicznych i mieszanych.....	103
5.7. Podsumowanie rozdziału	104

6. ROLA MODELOWANIA I IDENTYFIKACJI W PROJEKTOWANIU SYSTEMÓW MECHATRONICZNYCH .. 106

6.1. Modelowanie systemów mechatronicznych	107
6.1.1. Przykład podsystemu mechanicznego.....	108
6.1.2. Model kinematyczny i dynamiczny podsystemu mechanicznego.....	109
6.1.3. Integracja podsystemów w schematach blokowych SIMULINKA	111
6.1.4. Identyfikacja	111
6.2. Szybkie prototypowanie układu sterowania dla systemów mechatronicznych....	112
6.2.1. Symulacja komputerowa.....	114
6.2.2. Dostrajanie parametrów sterownika wirtualnego.....	115
6.2.3. Przygotowanie modelu sterownika do symulacji w czasie rzeczywistym	116
6.2.4. Metoda HiL (hardware in the loop)	117
6.3. Rozwiązania sprzętowe wspomagające prototypowanie.....	120
6.4. Prototypowanie systemów na sprzęcie specjalistycznym	122
6.4.1. Modelowanie sterowników wirtualnych w pamięci karty dSPACE	122
6.4.2. Karty pomiarowe z przetwornikami ADDA jako interfejs do procesu	123
6.5. Prototypowanie systemów na sprzęcie docelowym.....	125
6.5.1. Mikrosterowniki i oprogramowanie dSPACE TargetLink	126
6.5.2. Zestawy uruchomieniowe ASIC/FPGA	126
6.5.3. Prototypowanie regulatorów na komputerze przemysłowym	127
6.5.4. Prototypowanie na komputerze PC z kartą przetworników ADDA.....	127
6.6. Systemy operacyjne czasu rzeczywistego	128
6.6.1. VxWorks i Tornado	129
6.6.2. Java J2ME.....	130
6.6.3. Rozszerzenia systemu Windows i DOS do pracy w czasie rzeczywistym.....	131
6.7. Podsumowanie rozdziału	131
7. PRZYKŁAD: MODELE I UKŁADY STEROWANIA RAMIENIEM ELASTYCZNYM ROBOTA.....	135
7.1. Inicjowanie projektu	135
7.2. Faza specyfikacji wymagań	137
7.2.1. Przypadki użycia i aktorzy.....	138
7.3. Faza analizy WYMAGAŃ i projektowania wstępnego.....	139
7.3.1. Analiza wymagań, diagramy.....	140
7.3.2. Projektowanie wstępne, wybór struktury systemu	141
7.4. Modelowanie i weryfikacja eksperymentalna modeli	143
7.4.1. Stanowisko do dynamicznej identyfikacji elastycznego ramienia robota	144
7.4.2. Przegląd literatury: model i sterowanie ramieniem elastycznym robota.....	148

8	Mrozek Z, Komputerowo wspomagane projektowanie systemów mechatronicznych, ZN PK, Kraków 2002	
7.4.3.	Model analityczny belki elastycznej	153
7.4.4.	Symulacja dynamiki belki elastycznej z użyciem biblioteki MSL	157
7.4.5.	Model matematyczny ramienia elastycznego w przestrzeni stanu	158
7.4.6.	Model transmitancyjny dyskretny ramienia elastycznego	163
7.4.7.	Identyfikacja z użyciem sygnału stochastycznego	165
7.4.8.	Identyfikacja z użyciem sygnału prostokątnego	173
7.5.	Implementacja układu sterowania (projektowanie fizyczne).....	187
7.5.1.	Symulacja HiL w czasie rzeczywistym.....	187
7.6.	Konserwacja, modernizacja oraz utylizacja wyrobu	192
7.7.	Podsumowanie rozdziału	193
8.	WNIOSKI KOŃCOWE	197
	Literatura.....	200
	Słownik wybranych terminów.....	210
	Streszczenie.....	214
	Summary.....	215
	Résumé	216

Streszczenie

Autor przedstawia możliwość wykorzystania języków modelowania systemów informatycznych i innych narzędzi inżynierii oprogramowania do projektowania mechatronicznego. Diagramy w języku UML pozwalają na modelowanie systemów na różnych poziomach abstrakcji. Diagramy pokazują te elementy przyszłego systemu, które są aktualnie istotne, a pomija bądź upraszcza pozostałe elementy. Projekty wyrobów mechatronicznych są zazwyczaj tworzone przez wieloosobowe zespoły fachowców różnych specjalności. Użycie sformalizowanej, graficznej formy dokumentowania na wszystkich etapach projektowania ułatwia komunikowanie się członków zespołu.

Proces projektowania jest wspierany przez narzędzia CAD/CAM i CAE, ale nie rozwiązują one problemu integracji podzespołów o różnej naturze fizycznej. Nowym narzędziem do modelowania systemów interdyscyplinarnych jest język Modelica. Jest on oparty na koncepcji modelowania fizycznego, co oznacza wymóg spełnianie praw fizyki również w złączach pomiędzy elementami składowymi modelu, jak np. w grafie wiązań.

Autor zamieszcza przykłady modelowania, symulacji i prototypowania. Praca zawiera słownik ważniejszych terminów i obszerną literaturę.

Zdaniem autora, Modelica i UML będą wkrótce uznanymi narzędziami wspomagającymi projektowanie systemów mechatronicznych.

Computer Aided Design of Mechatronic systems

Summary

Author presents how to use software engineering tools in design of mechatronic systems. UML is graphical language for visualizing, constructing and documenting artefacts of systems. UML models are essential for communication among project teams and to manage the complexity as they increase in scope and scale. UML improves design quality and reduces time to market.

CAD/CAM and CAE tools are widely used in design but they do not solve problems of integration of subsystems of different physical nature. This can be achieved using UML and Modelica, new language for physical modelling.

Example of modelling and simulation and prototyping are included, as well as glossary related to mechatronic design and references. In author's opinion, UML and Modelica will be widely accepted in mechatronic design.

Etablissement d'un projet mécatronique assisté par ordinateur

Résumé

De plus en plus, les composants mécaniques, électroniques et logiciels sont intégrés dans un seul produit. L'auteur présente la possibilité d'application des langage de modélisation des systèmes informatiques pour des projets mécatroniques.

Les diagrammes faits en langage UML permettent la modélisation des systèmes avec différents niveaux d'abstraction. Chaque type de diagrammes peut présenter les éléments du système en cours de conception avec des points de vue différents. L'utilisation d'une forme graphique pour rédiger la documentation facilite la communication entre les spécialistes de différents domaines, durant toutes les étapes de la réalisation du projet. Ces points sont importants pour réduire le temps de développement et satisfaire aux besoins de la norme et de la qualité.

Le langage Modelica est très efficace pour la modélisation de systèmes à l'interface entre diverses disciplines scientifiques, parce qu'il utilise les principes de la modélisation physique. De l'avis de l'auteur, l'utilisation conjointe des langages UML et Modelica peut permettre à la fois de concevoir proprement des projets et aussi de rédiger la documentation afférente. L'ensemble peut s'appliquer à différents champs d'application.

Quelques exemples de modélisations, de simulations et de constructions de prototypes sont présentés, ainsi qu'un glossaire et des références relatifs à la conception de systèmes mécatroniques.