

OBMC-5 MODELACIÓN Y SIMULACIÓN COMPUTACIONAL

NOMBRE DE LA ASIGNATURA:			
MODELACIÓN Y SIMULACIÓN COMPUTACIONAL			
OBJETIVO GENERAL Y PARTICULARES DE LA ASIGNATURA:			
General: Este curso complementa los cursos de sistemas dinámicos y control, al capacitar a los estudiantes en el uso de lenguajes de programación orientado a objetos con el alcance de desarrollar de manera sencilla modelos de sistemas dinámicos eventualmente muy complejos. Al finalizar el curso el alumno conocerá los conceptos y técnicas generales de entornos de programación orientado a objetos y lenguaje de modelado multi-dominio para modelado de sistemas complejos orientado a componentes. Así mismo, el alumno será capaz de utilizar los entornos de programación para los lenguajes C++, Modelica, MatLab y Simulink, con el nivel suficiente, para realizar simulaciones de sistemas ciber-físicos, mecatrónicos u otros sistemas multi-dominio. Particulares: Modelación de sistemas físicos, y el lenguaje de programación C con el nivel suficiente para realizar programas de simulación y/o aplicación a dispositivos mecatrónicos.			
Duración del ciclo: 24 sesiones de 2.5 HORAS	Horas totales con docente: 60 Horas	Horas totales independientes: 40 Horas	Instalaciones: AULA Y/O LABORATORIO DE CÓMPUTO
CICLO, ÁREA O MÓDULO: OBLIGATORIO		CRÉDITOS: 4	CLAVE: OBMC-5
TEMAS Y SUBTEMAS:			
1. Introducción al modelado y simulación 1.1. Sistemas y experimentos 1.2. El concepto de modelo y simulación 1.3. Construcción y análisis de modelos 1.4. Tipos de modelos matemáticos 1.5. Diseño de productos 2. Introducción a la programación orientada a objetos 2.1. Constantes, variables y comentarios 2.2. Clases e instancia 2.3. Herencia 2.4. Estructura de ecuaciones 2.5. Modelo físico acausal 2.6. Componentes de modelado del software usado en el curso 2.6.1. Componentes 2.6.2. Diagramas de conexión 2.6.3. Conectores y clases conectoras 2.7. Clases parciales 2.8. Diseño de componentes de librerías 3. Ecuaciones y programación 3.1. Propiedades generales de ecuaciones 3.1.1. Categorías de ecuaciones y usos 3.1.2. Expresividad de ecuaciones 3.1.3. Regla de asignación única para ecuaciones 3.1.4. Principio de flujo de datos síncrono 3.2. Ecuaciones en declaraciones 3.2.1. Declaración de ecuaciones 3.2.2. Modificadores de ecuaciones 3.3. Conectar ecuaciones 3.3.1. Estructuras de conexión repetitivas 3.4. Ecuaciones condicionales con if-equations 3.5. Ecuaciones condicionales con when-equations 3.5.1. Restrictions on Equations within when-Equations 3.6. Aplicación de la Regla de Asignación Única a las Ecuaciones Cuando 3.7. reinit			

3.8. assert and terminate 3.9. assert 3.10. terminate 3.11. Initialization and initial equation 3.12. Initialization Constraints and Hints Through the start Attribute 3.13. Initialization Constraints Through initial equation 3.14. Using when-Equations During Initialization 3.15. Explicit and Implicit Specification of Parameter Values for Initialization 3.16. The Number of Equations Needed for Initialization 3.17. System Initialization ExamplesDiscrete Controller Models 3.18. Partial Differential Equations 3.19. Structuring a PDE Problem 3.20. Basic Definitions of Domain, Field, Boundary, and Interior 3.21. The Method of Lines Approach to Spatial Discretization 3.22. Range and Domain Geometry Types and Generalized in Operator 3.23. Summary of Operators and Types Needed for PDEs in Modelica 3.24. Modelica PDE Examples of Domains, Fields, and Models in 1D, 2D, and 3D 3.25. Modelica Heat Diffusion Model on a Rectangular Domain 3.26. One-dimensional Domains 3.27. Two-dimensional Domains 3.28. Three-dimensional Domains 3.29. A PDE on a 2D Domain with General Shape and FEM Solution 3.30. Grid Function Finite Difference Approach to 1D Heat Diffusion 3.31. 2D Grid Functions 3.32. Equation Operators for Overconstrained Connection-Based Equation Systems 3.33. An Example of an Overconstrained Connection Loop 3.34. Overconstrained Equation Operators for Connection Graphs 3.35. Converting the Connection Graph into Trees and Generating Connection Equations 3.36. Summary 3.37. Literature 3.38. Exercises 4. Optional Function Result 5. Concepto y clasificación de sistemas 6. Modelos de estado estacionario y modelos dinámicos 7. Sistemas con parámetros distribuidos 8. Modelación de sistemas mecatrónicos 9. Concepto de simulación 10. Simulación de sistemas utilizando el lenguaje C 11. Simulación de sistemas utilizando MatLab y Simulink
RESULTADOS DEL APRENDIZAJE:
ACTIVIDADES DE APRENDIZAJE BAJO CONDUCCIÓN DEL DOCENTE:
- Exposición de los alumnos de forma grupal e individual - Desarrollo de tareas en las que se simularán sistemas mecatrónicos en C, MatLab y Simulink
ACTIVIDADES DE APRENDIZAJE INDEPENDIENTE:
MEDIOS Y CRITERIOS DE EVALUACIÓN:
Escala de evaluación 0 – 10. Calificación mínima aprobatoria: 8 - Tareas y ejercicios propuestos durante el curso 20 % - Evaluaciones parciales (2 evaluaciones) 40% - Proyectos parciales 40%
RECURSOS Y MATERIALES

Computadora personal, Software de simulación Modelica, MatLab, Compiladores.
--

BIBLIOGRAFÍA

- | |
|---|
| <ol style="list-style-type: none">1. Schwarzenbach, J. y Gill, K.F. System Modelling and Control. John Wiley & Sons. , 1978.2. Kernighan, B.W., Ritchie, D.M. El Lenguaje de programación C, Prentice Hall Hispanoamericana, 19863. Ogata, K., Designing Linear Control Systems with MatLab, Prentice Hall, 19944. Karnopp, D.C., Margolis, D.L., Rosenberg, R.C., System Dynamics: Modelling and Simulation of Mechatronic Systems, 20005. Fritzon, P., & Engelson, V. (1998, July). Modelica—A unified object-oriented language for system modeling and simulation. In <i>European Conference on Object-Oriented Programming</i> (pp. 67-90). Springer, Berlin, Heidelberg. |
|---|

REQUISITOS ACADÉMICOS DEL PERSONAL DOCENTE

Grado académico de Doctor en Ciencias. Experiencia probada en la materia a nivel docencia e investigación
