UNIVERSITY OF WARWICK

Proposal Form for New or Revised Modules (MA1 - version 7 - April 2014)

Approval information	
Approval Type	☐ New module ✓ Revised module ☐ Discontinue module
Date of Introduction/Change	October 2018
If new, does this module replace another? If so, enter module code and title:	No.
If revised/discontinued, please outline the rationale for the changes:	A revised robotic lab is proposed based on the feedback of the students so that they can experiment with controller design to overcome hardware limitations. Revision classes are necessary and are proposed therefore. Assessment has been changed to a 70% 2-hour examination and 30% coursework comprising 25% case-study report and 5% group presentation.
Confirmation that affected departments have been consulted:	Changes were made in consultations between the School of Engineering and WMG. Computer Science have been consulted via the CSE steering committee.

Module Summary	
1. Module Code (if known)	ES4F0
2. Module Title	Advanced Control Systems
3a. Lead department:	School of Engineering
3b. Teaching Split (if known):	100% School of Engineering
4. Name of module leader	Vishwesh Kulkarni
5. Level	UG: ☐ Level 4 (Certificate) ☐ Level 5 (Intermediate) ☐ Level 6 (Honours) PG: ✓ Level 7 (Masters) ☐ Level 8 (Doctoral) See Guidance Notes for relationship to years of study
6. Credit value(s) (CATS)	15
7. Principal Module Aims	The module objective is to teach various methods of synthesizing control systems for real-world complex dynamic systems such that the desired end user objectives are met satisfactorily.

Module Summary	
	• The dynamic systems considered in this course are real-world systems that can be represented through systems of ordinary differential equations. • The complexities include large system dimensions (leading to computational challenges), nonlinearities, and time-delays. • Techniques to test whether the end user objectives are feasible will be taught first and then the techniques for synthesizing requisite controllers will be introduced. • To ensure that the students understand how to apply the concepts and techniques for real-world applications, the module includes a case study project on synthesizing controller for a process control and a detailed case study involving theoretical questions and a programming assignment. In this context, the module aims to first introduce mathematical paradigms so that the task of meeting the end-user objectives can be posed as a constrained optimization problem. • Here, the basics of state-space control, linear programming, and LMI programming will be introduced. The module then aims to cover some landmark results on the infeasibility of design objectives (e.g., the waterbed theory, limitations due to RHP poles, limitations due to time delays). The module then aims to teach algorithms to check the feasibility of the performance objectives. In the final 5 weeks, the module aims to cover the salient features of different computational methods, along with associated software programming, of synthesizing a controller (PID controller, H-infinity controller, and L1 adaptive controller) whose robustness properties can be specified *a priori* by the end user.
8. Principal Learning Outcomes	By the end of the course, the student will be able to do the following: • Given a dynamic system expressed using ordinary differential equations, check whether the end user objectives are feasible or not. • Pose the controller synthesis problem as a constrained optimization problem using state space representation (note: here, the controller can be PID or optimal or H-infinity or L1 adaptive). • Use linear programming and LMI programming to solve this problem, and write the associated software code.
9. Timetabled Teaching Activities (summary)	• Lectures: 30 X 1 hr = 30 hrs • Revision Lectures: 2 x 1 hr = 2 hrs

Module Summary	
	• Lab: 4 X 1 hr = 4 hrs • Programming Assignment: 5 x 2 hrs = 10 hrs Total 46 hours
10. Departmental Web-link	https://warwick.ac.uk/fac/sci/eng/eso/modules/year4/es4f0/
11. Other essential notes	• Pre-Course material on software programming and Linear Programming will be made available to the students 1 week before the lectures starts. • Pre-Course material on elementary controller synthesis will be made available to the students 1 week before the lectures start.
12. Assessment methods (summary)	70% exam (2 hrs) 30% coursework comprising • 25% case study report (10 pages + MATLAB code) • 5% group presentation, based on flipped lecture (10 mins + 5 min Q&A + prep material posted on Moodle). Peer-moderated. NB – Assessed work to be passed based on overall average rather than individually.

For use by Strategic Planning and Analytics Office only - Do not fill in this section

Level	JACS3 Code	Teaching Split
		If not provided in 3b above

External Credit Level		Scheme	

Module Context

13. Please list all departments involved in the teaching of this module. If taught by more than one department, please indicate percentage split.

School of Engineering

14. Availability of module This module would interest students enrolled in the MEng/MSc/PhD programs of the School of Engineering, Warwick Mathematics Institute, Computer Science Engineering, Complexity Science, WMG, School of Life Science, and the Centre for Scientific Computing.

Degree Code	Title	Study Year	C/OC/ A/B/C	Credits
G408	MEng Computer Systems Engineering	4	A	15
HH63	MEng Systems Engineering	4	A	
New	MEng Systems Engineering with Exchange Year	4	A	
HH64	MEng Systems Engineering with Intercalated Year	5	A	
HH65	MEng Systems Engineering with a Year in Research	5	A	
H107	MEng Engineering	4	A	
New	MEng Engineering with Exchange Year	4	A	
H109	MEng Engineering with Intercalated Year	5	A	
H110	MEng Engineering with a Year in Research	5	A	
H642	MSc Energy and Power Engineering	M1	A	

15. Minimum number of registered students required for module to run

10

16. Pre- and Post-Requisite Modules

ES3C8 Systems Modelling and Control or its equivalent

Module Content and Teaching

17. Teaching and Learning Activities *(totals for module – please see guidance)*

Module duration (weeks)	**10**
Lectures	**30 + 2 (revision)**
Seminars	**-**
Tutorials	**-**

Module Content and Teaching	
Project Supervision	**8-16 (case study project: 1 per student (or per 2 students)**
Demonstration	**-**
Practical Class/Workshops	**4 x 1 hr Laboratory** **5 x 2 hr Programming Sessions** **(both supervised by module leader)**
Supervised time in studio/workshop	**-**
Fieldwork	**-**
External visits	**-**
Work based learning	**-**
Placement	**-**
Year abroad	**-**
Other activity *(please describe): e.g. distance-learning, intensive weekend teaching etc.*	**104 hours independent learning (included supervised case study project)**

18. Assessment Method (Standard)		
Type of assessment	**Length**	**% weighting**
Written Examinations	2 Hours	**70%**
Practical Examinations		
Assessed essays/coursework	30% coursework comprising • 25% case study report (10 pages + MATLAB code) • 5% group presentation, based on flipped lecture (10 mins + 5 min Q&A + prep material posted on Moodle). Peer-moderated.	**25%** **5%**
18a. Final chronological assessment *(please see guidance)*	Written examination	

19. Methods for providing feedback on assessment.

- Model solutions to past papers.
- Support through advice and feedback hours.
- Summative mark and written feedback on coursework elements (two case studies).
- Cohort-level feedback on final exam.

- Summative mark and written comments on group presentation.

20. Outline Syllabus

Introduction to dynamical systems, review of linear algebra. Least squares optimization for linear systems ODE's to state-space models, Transient and steady-state response in state-space representations, Controllability and observability, -state feedback controllers, Ackerman's formula, Input-output stability, introduction to multiplier theory, Stability analysis: effect of time-delay, nonlinearities, and uncertainties, Techniques to characterize robust performance, Sensitivity, complementary sensitivity, and waterbed effect, Bode sensitivity integral, KYP Lemma, Multiplier theoretic characterization of commonly encountered nonlinearities, Multipliers theoretic characterization of time-delays, PID controllers, Introduction to optimal controllers, Introduction to H-infinity control, Introduction to L1 adaptive control, Linear programming and LMI programming algorithms for synthesizing controllers introduced.

The lectures will illustrate the concepts and techniques through several examples including the case study lab project and a case study including a programming assignment so that the students well understand how those are applied in the context of real-world applications.

21. Illustrative Bibliography

1. D. Luenberger and Y. Ye. Linear and Nonlinear Programming (4th Edition). Springer-Verlag, 2016. ISBN: 9783-3191-8841-6e.
2. S. Skogestad and I. Postlethwaite. Multivariable Feedback Control: Analysis and Design (2nd Edition), Wiley Interscience, 2005. ISBN: 9780-4700-11683.
3. G. Franklin, J. Powell, and A. Emami-Naeini. Feedback Control of Dynamic Systems (7th Edition), Pearson, 2014. ISBN: 9781-2920-68909.
4. Stormy Attaway. MATLAB (3rd Edition), Butterworth-Heinemann, 2013. ISBN: 9780-1240-58767.
5. D. Seborg, T. Edgar, D. Mellichamp, and F. Doyle. Process Dynamics and Control (3rd Edition), Wiley, 2011. ISBN: 9780-4706-46106.

22. Learning outcomes

Successful completion of the module leads to the learning outcomes. The learning outcomes identify the knowledge, skills and attributes developed by the module.

Learning Outcomes should be presented in the format "By the end of the module students should be able to..." using the table at the end of the module approval form:

Resources

23. List any additional requirements and indicate the outcome of any discussions about these.

N/A

Approval	
24. Module leader's signature	Vishwesh Kulkarni
25. Date of approval	CMAC Chair's Action 17 May 2018
26. Name of Approving Committee (include minute reference if applicable)	School of Engineering and WMG Course and Module Approval Committee
27. Chair of Committee's signature	Professor Gillian Cooke
28. Head of Department(s) signature	Professor David Towers

Examination Information		
A1. Name of examiner (if different from module leader)		
A2. Indicate all available methods of assessment in the table below		
% Examined	**% Assessed by other methods**	**Length of examination paper**
70%	• Case Study Report (25%) • Group Presentation (5%)	2 hours
A3. Will this module be examined together with any other module (sectioned paper)? If so, please give details below.		
-NA-		
A4. How many papers will the module be examined by?	✓ 1 paper	☐ 2 papers
A5. When would you wish the exam take place (e.g. Jan, April, Summer)?	**Summer**	
A6. Is reading time required?	☐ Yes	☒ No
A7. Please specify any special exam timetable arrangements.		
None		
A8. Stationery requirements		
No. of Answer books?	**1**	
Graph paper?	**None**	
Calculator?	**Yes**	
Any other special stationery requirements (e.g. Data books, tables etc)?	**Engineering Data Book**	
A9. Type of examination paper		
Seen?	☐ Yes	✓ No
Open Book?	☐ Yes	✓ No
Restricted?	☐ Yes	✓ No

Examination Information	
If restricted, please provide a list of permitted texts:	-NA-

LEARNING OUTCOMES		
(By the end of the module the student should be able to....)	**Which teaching and learning methods enable students to achieve this learning outcome? (reference activities in section 15)**	**Which summative assessment method(s) will measure the achievement of this learning outcome? (reference activities in section 16)**
Given a dynamic system expressed using ordinary differential equations, check whether the end user objectives are feasible or not.	Lectures will cover the conceptual fundamentals and the mathematical programming techniques to achieve this learning outcome	Examination, case-study report
Pose the controller synthesis problem as a constrained optimization problem using state space representation (note: here, the controller can be PID or H-infinity or L1 adaptive).	Lectures will cover the conceptual fundamentals and the mathematical programming techniques to achieve this learning outcome. In addition, the lab and the programming assignment will give the students a hand-on experience of how to pose and solve this problem.	Examination, case-study report, Group presentation
Use linear programming and LMI programming to solve this problem, and write the associated software code.	Lectures will cover the conceptual fundamentals and the mathematical programming techniques to achieve this learning outcome. In addition, the lab and the programming assignment will give the students a hand-on experience of how to pose and solve this problem.	Examination, case-study report, Group presentation