Welcome to SEENG 480B / CSC 485A / CSC 586A Self-Adaptive and Self-Managing Systems

Dr. Hausi A. Müller
Department of Computer Science
University of Victoria

http://courses.seng.uvic.ca/courses/2015/summer/seng/480a
http://courses.seng.uvic.ca/courses/2015/summer/csc/485a
http://courses.seng.uvic.ca/courses/2015/summer/csc/586a

1

Announcements

- Midterm II
  - Thu, July 16 in class
- A4
  - Posted by Friday, July 10
  - Due Friday, July 31
- A3
  - Due Friday, July 10
  - July 13 Part 2 demos before and after class
    Sign up for demos (!)
- A2 grading questions
  - Ron Desmarais
  - Mon, July 13
    4-5 pm in ECS 415
- Grad project
  - Posted
  - Due Friday, July 24
  - Presentations Mon, July 27 and Thu, July 30
  - All students are expected to assess presentation as part of course participation mark

2

July Calendar

- July 9—MRAC and MIAC class
- July 10—A3 due
- July 13—MART class continued and A3 P2 demos
- July 16—Midterm II in class
- July 20/23—Characterizing SAS Problems
- July 24 Grad Presentation Slides due
- July 27/30—Grad Presentations
  - Non-presenters evaluate presentations
- July 31 A4 due

Demo details - July 13, 2015

ECS 660 [9:30 – 11:30]
The demos start at 10:00, the room is available to students from 9:45 to get the set up ready.

ECS 418 [12:15 – 12:55]
The demos are 5 min long, there is 5 min extra between demos to allow groups' set up given the restrictions of the space.

Email submit@rigiresearch.com to take a time slot, these will be assigned in order of arrival

NOTE: Students with classes in the morning will have preference for section 4 which is during the class time. Please be considerate about this restriction to your fellow classmates and take a morning time slot if you don't have class (or other impediment) in the morning

Place	Section	Time	No. Groups
ECS 660	1A	10:00 - 10:05	[1]
	1B	10:05 - 10:10	[1]
	1C	10:10 - 10:15	[1]
	1D	10:15 - 10:20	[1]
	2A	10:25 - 10:30	[1]
	2B	10:30 - 10:35	[1]
	2C	10:35 - 10:40	[1]
	2D	10:40 - 10:45	[1]
	3A	10:50 - 10:55	[2]
	3B	10:55 - 11:00	[2]
	3C	11:00 - 11:05	[2]
	3D	11:05 - 11:10	[2]
ECS 418	4A	12:15 - 12:20	[1]
	4B	12:25 - 12:30	[1]
	4C	12:35 - 12:40	[1]
	4D	12:45 - 12:50	[1]
	4E	12:50 - 12:55	[1]

Assignment 3 Demos on Monday, July 13

4

Graduate Student Research Paper Presentations

- Brun, Y., Di Marzo Serugendo, G., Gacek, C. Giese, H. Kienle, H.M., Litoiu, M., Muller, H.M., Pezzè, M., Shaw, M.: Engineering Self-Adaptive Systems through Feedback Loops. Software Engineering for Self-Adaptive Systems, pp. 48-70 (2009) — **Presentation by Simar Arora Khushboo Gandhi: July 27**
- Garlan, D., Cheng, S.-W., Huang, A.-C., Schmerl, B., Steenkiste, P.: Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure. IEEE Computer 37(10) 46-54 (2004) — **Presentation by Stephan Heinemann and Waseem Ullah: July 27**
- Oreizy, P., Medvidovic, N., Taylor, R.N.: Runtime Software Adaptation: Framework, Approaches, and Styles. In: ACM/IEEE International Conference on Software Engineering (ICSE 2008), pp. 899-910 (2008) — **Presentation by Sumit Kadyan and Adithya Rathakrishnan: July 27**
- Kramer, J., Magee, J.: Self-Managed Systems: An Architectural Challenge. In: ACM /IEEE International Conference on Software Engineering 2007 Future of Software Engineering (ICSE), pp. 259-268 (2007) — **Presentation by Ernest Aaron and Harshit Jain : July 27**

5

Graduate Student Research Paper Presentations

- Aksanli, J. Venkatesh, L.Z., Tajana R.: Utilizing Green Energy Prediction to Schedule Mixed Batch and Service Jobs in Data Centers. In: Proceedings 4th Workshop on Power-Aware Computing and System (HotPower 2011), Article 5 (2011) — **Presentation by Junnan Lu and Francis Harrison: July 30**
- Ebrahimi, S., Villegas, N.M., Muller, H.A., Thomo, A.: SmarterDeals: a context-aware deal recommendation system based on the SmarterContext engine. CASCON 2012: 116-130 (2012) — **Presentation by Carlene Lebeuf and Maria Ferman: July 30**
- Villegas, N.M., Muller, H.A., Tamura, G., Duchien, L., Casallas, R.: A framework for evaluating quality-driven self-adaptive software systems. In: Proc. 6th Int. Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2011), pp. 80-89 (2011) — **Presentation by Parminder Kaur and Navpreet Kaur: July 30**
- Villegas, N.M., G. Tamura, H.A. Müller, L. Duchien, and R. Casallas, DYNAMICO: A reference model for governing control objectives and context relevance in self-adaptive software systems, in: R. de Lemos, H. Giese, H.A. Müller, and M. Shaw (Eds.), Software Engineering for Self-Adaptive Systems, LNCS 7475, Dagstuhl Seminar 10431, Springer, pp. 265-293 (2013) — **Presentation by Arturo Reyes Lopez and Babak Tootoonchi,: July 30**

6

Guidelines for Grad Student Presentations

- Format of presentation
  - Presentation 10 mins
  - Q&A 5 mins
  - Practice talk (!!)
  - Practice of the best of all instructors
- Slides
  - High quality and polished
  - Submit slides by July 24 to instructor for approval
  - Submit final slides 1 day after presentation for posting on website
- Talk outline
  - Motivation
  - Problem
  - Approach
  - Contributions of the paper
  - Relation to what we learned in the course so far
- Assessment
  - All students have to fill out an evaluation form
  - Counts towards class participation

7

Presentation Assessment

Evaluator's name:		
Graduate students:		
Quality of presentation		
Did I learn something? Did the presentation stimulate my interest?	5	
Do I know now what the paper is all about?	5	
Does the presenter know the subject well?	5	
Presentation style: main points reiterated; positive attitude; excited about the subject.	5	
How did the presenter perform in the Q&A session?	5	
Subtotal	25	
Other comments		

July 27 and July 30 CSC 586A Presentations

8

Midterm II
Thu, July 16 in class

- All materials presented in class including Mon, July 13
  - Before and after Midterm I
  - More questions from after Midterm I
  - All on-line lecture notes
- Study sample Midterm II questions carefully
- Format
  - Same format as Midterm I
  - Crib sheet in the form of a paper
  - Argue convincingly
  - Define terms
  - Essay questions

No cheating

9

Crib Sheet for Midterm II

- **Crib sheet**: a concise set of notes for quick reference
  H.A. Müller and N.M. Villegas: Runtime Evolution of Highly Dynamic Software, in *Evolving Software Systems,* T. Mens, A. Serebrenik, and A. Cleve (Eds.), Springer, pp. 229-264 (2014)
  http://link.springer.com/chapter/10.1007%2F978-3-642-45398-4_8
- Summarizes a significant part of this course
- You will have access to a hard copy during Midterm II
- Contains answers to selected Midterm II questions

10

Topics
Autonomic Computing

- Autonomic manager
- MAPE-K loop
- Monitoring
- Analysis
- Symptoms
- Planning
- Policies
  - Action
  - Goal
  - Utility-function
- Sensing
- Actuating
- Knowledge bases for AC
- ACRA
- Manageability interfaces
- Models at runtime
- MART
- Uncertainty

11

Topics
Control loops

- Types of feedback: positive, negative, bipolar
- Hellerstein feedback loop model
- Controller
- Managed element, process, plant
- Disturbance input
- Noise input
- Transducer
- Reference model
- Simulation model
- Model identification
- MIAC
- MRAC
- PID controller

12

Interesting Potential Midterm II Questions

- Design a concrete and viable
  - action policy
  - goal policy
  - utility-function policy
  - Design a Green utility-function policy
  - How can cost be integrated into a utility-function?
- PID controllers
- Explain the notion of adaptive control
  - MRAC architecture
  - MIAC architecture
  - How do they relate?
  - How do they relate to ACRA?

13

Interesting Potential Midterm II Questions

- What is the difference between anticipated and un-anticipated adaptation?
- What is the difference between fully autonomous systems and human-in-the-loop systems?
- What is the difference between design-time and run-time adaptation?
- What are self-* properties?
- What are requirements at runtime?
- What are models at runtime (MART)?
- What is runtime V&V?

14

Interesting Potential Midterm II Questions

- What aspects of the environment should a self-adaptive system monitor?
  - The system cannot monitor everything in the environment
  - What aspects of the environment are truly relevant?
- How should a self-adaptive system react if it detects changes in the environment?
  - Maintain high-level goals
  - Relax non-critical goals to allow the system a degree of flexibility
  - Goal trade-off analysis

15

Course Requirements

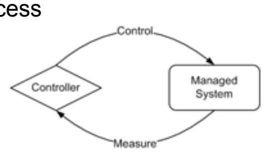
Unit	Undergrads Weight	Grads Weight	Remarks
A1	12%	9%	Due Fri, May 29, 2015
A2	12%	9%	Due Fri, June 19, 2015
A3	12%	9%	Due Fri, July 10, 2015
A4	12%	9%	Due Fri, July 31, 2015
Grad Project		12%	Due Sat, July 25, 2015
Participation and presentation	7%	7%	Only graduate students are required to give a presentation towards the end of the course.
Midterm 1	20%	20%	June 4, 2015 in class. Closed books, closed notes, no phones, no computers, no calculators, no gadgets.
Midterm 2	25%	25%	July 16, 2015 in class. Closed books, closed notes, no phones, no computers, no calculators, no gadgets.
Total	100%	100%	Have a great course!

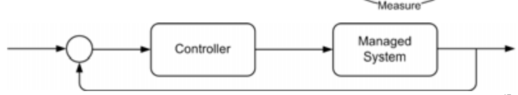
- All materials discussed in class are required for the midterm examinations
- Completing all midterms and assignments is required to pass the course
- Passing the midterms is not absolutely required to pass the course, but of course highly recommended

16

Feedback Control System

- **Merriam-Webster's Online Dictionary**
  the return to the input of a part of the output of a machine, system, or process
  - producing changes in an electronic circuit to improve performance
  - an automatic control device to provide self-corrective action





17

Control Theory

- A theory that deals with influencing the behavior of dynamical systems
- An interdisciplinary subfield of science, which originated in engineering and mathematics

18

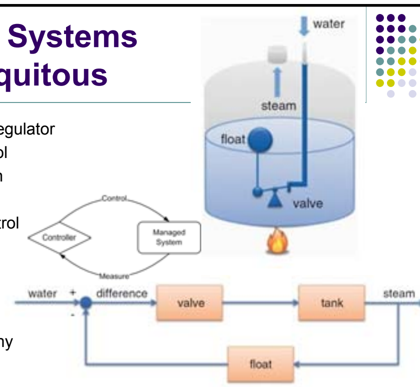
Origins of Control Theory

- Control systems date back to antiquity
- James Maxwell (1831-1879) started the field in 1868 analyzing the dynamics analysis of the centrifugal governor
- Routh (1831-1907) abstracted Maxwell's results for the general class of linear systems in 1877
- Hurwitz (1859-1919) analyzed system stability using differential equations in 1877
- Laplace (1749-1827) invented the Z-transform used to solve discrete-time control theory problems. The Z-transform is a discrete-time equivalent of the Laplace transform.
- Alexander Lyapunov (1857–1918) developed stability theory.
- Harry Nyquist (1889–1976), developed the Nyquist stability criterion for feedback systems in the 1930s.

19

Control Systems are Ubiquitous

- Water tank regulator
- Cruise control
- Fuel injection
- Flight control
- Climate Control
- Health Care
- Quadcopters
- Rumba
- iRobots
- Radiotheraphy



P. Lalanda, J. McCann, Julie, A. Diaconescu: Autonomic Computing: Principles, Design and Implementation, Springer (2013)

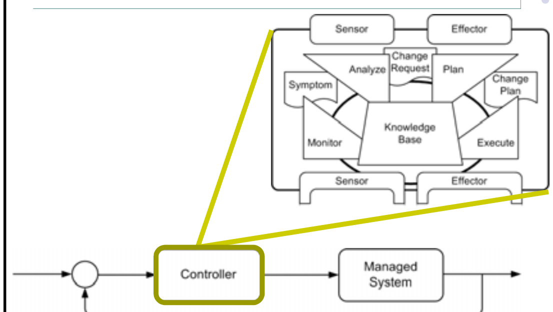
20

Control System Goals: Self-Management

- Regulation
  - Thermostat, target service levels
- Tracking
  - Robot movement
  - Adjust TCP window to network bandwidth
- Optimization
  - Best mix of chemicals
  - Minimize response times

21

Controller as an Autonomic Element

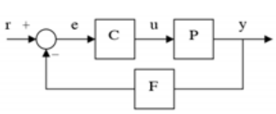


22

Closed Loop Controller or Feedback Controller

- The output *y(t)* of the feedback system is fed back through a sensor measurement *F* to the reference value *r(t)*.
- The controller *C* then takes the error *e* (difference) between the reference and the output to change the inputs *u* to the control process *P*.

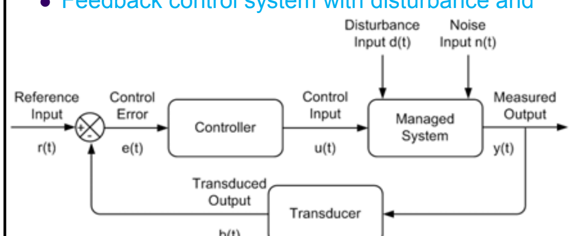
- SISO
  - Single-input-single-output (*SISO*) control system
  - Variables are simple scalar values (i.e., r(t), e(t), u(t), y(t)
- MIMO
  - Multi-Input-Multi-Output systems, with more than one input/output, are common
  - Variables are vectors



23

Realization of a Dynamic Architecture

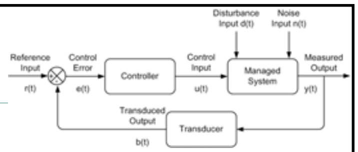
- Feedback control system with disturbance and



Hellerstein, Diao, Parekh, Tilbury: Feedback Control of Computing Systems. John Wiley & Sons (2004)

24

Realization of a Dynamic Architecture



- Reference input
  - Goal, objectives, specified desired output
- Control Error
  - Reference input minus transduced output
- Control Input
  - Parameters which affect behavior of the system—number of threads, CPU, memory
- Disturbance input
  - Affects control input—arrival rate
- Controller
  - Change control input to achieve reference input—design is based on a model of the managed system
- Managed system
  - Dynamical system, process, plant—often characterized by differential equations
- Measured output
  - Measurable feature of the system—response time
- Noise input
  - Affects measured output
- Transducer
  - Transforms measured output to compare with reference

Controller Algorithm based on Managed System Model

- *"All models are wrong, some models are useful."*
  - generally attributed to the statistician George Box
- The design of the controller algorithm is based on a model of the managed system or process
- Approaches
  - Analytical modeling: physical and mathematical laws
  - Experimental modeling: data fitting from observed input and output
- The control algorithm changes $u(t)$ based on the error $e(t) = r(t) - b(t)$
  - Proportional—if $e(t)$ is high, then $u(t)$ should be high
  - Integrative—eliminates transients; sum of all previous errors
  - Derivative—anticipate the trends; rate of change of the error
  - PID—computation based on the error (proportional), the sum of all previous errors (integral) and the rate of change of the error (derivative)

26

PID Controller

- The PID algorithm is the most popular feedback controller algorithm used
- It is a robust easily understood algorithm that can provide excellent control performance despite the varied dynamic characteristics of processes
- PID algorithm consists of three basic modes:
  - **P**roportional mode
  - **I**ntegral mode
  - **D**erivative mode

27

P, PI, or PID Controller

- When utilizing the PID algorithm, it is necessary to decide which modes are to be used (P, I or D) and then specify the parameters (or settings) for each mode used.
- Generally, only three basic algorithms are used: P, PI or PID

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\,dt + K_d \frac{de(t)}{dt}$$

proportional gain — integral gain — derivative gain

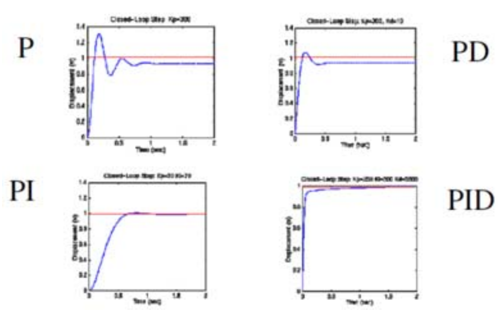
28

Controller Effects

- A proportional controller (P) reduces error responses to disturbances, but still allows a steady-state error.
- When the controller includes a term proportional to the integral of the error (I), then the steady state error to a constant input is eliminated, although typically at the cost of deterioration in the dynamic response.
- A derivative control typically makes the system better damped and more stable

29

PID Controller



Closed-Loop Response

	Rise time	Max overshoot	Settling time	Steady-state error
P	Decrease	Increase	Small change	Decrease
I	Decrease	Increase	Increase	Eliminate
D	Small change	Decrease	Decrease	Small change

31

PID Controller

- Output feedback
  - From Proportional action
  - Compare output with set-point
- Eliminate steady-state offset or error
  - From Integral action
  - Apply constant control even when error is zero
  - Eliminates transients; sum of all previous errors
- Anticipation
  - From Derivative action
  - React to rapid rate of change before errors grows too big
  - Anticipate the trends; rate of change of the error

32

Adaptive Control

- Adaptive control is the idea of "redesigning" the controller while online, by
  - looking at its performance and
  - changing its dynamic in an automatic way
- Motivated by aircraft autopilot design
  - Allow the system to account for previously unknown dynamics
- Adaptive control uses feedback to observe the process and the performance of the controller and reshapes the controller closed loop behavior autonomously.

33

Adaptive Control

- Modify the control law to cope by changing system parameters while the system is running
- Different from Robust Control in the sense that it does not need *a priori* information about the uncertainties
  - Robust Control includes the bounds of uncertainties in the design of the control law.
  - Therefore, if the system changes are within the bounds, the control law needs no modification

34

System Identification Model Building

- Mathematical tools and algorithms to build dynamical models from measured data
- A dynamical mathematical model in this context is a mathematical description of the dynamic behavior of a system or process in either the time or frequency domain
- Theories and processes
  - Physical
  - Computing
  - Social
  - Engineering
  - Economic
  - Biological
  - Chemical
  - Therapeutic

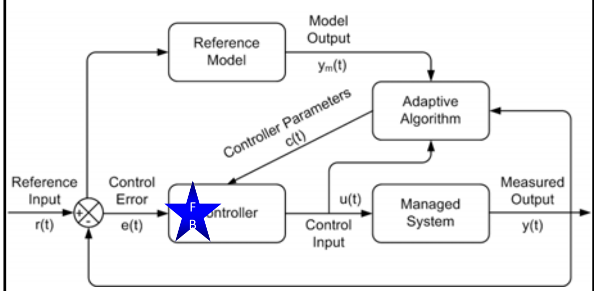
35

Model Reference Adaptive Controllers—MRAC

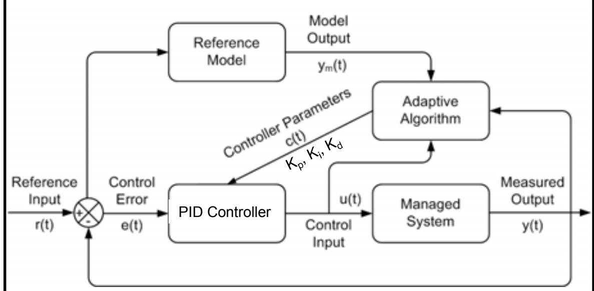
- Also referred to as Model Reference Adaptive System (MRAS)
- Closed loop controller with parameters that can be updated to change the response of the system
- The output of the system is compared to a desired response from a reference model (e.g., simulation model)
- The control parameters are updated based on this error
- The goal is for the parameters to converge to ideal values that cause the managed system response to match the response of the reference model.

36

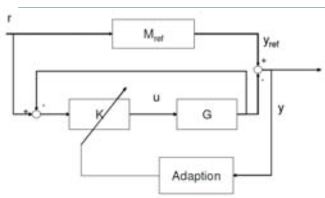
Model Reference Adaptive Controllers—MRAC

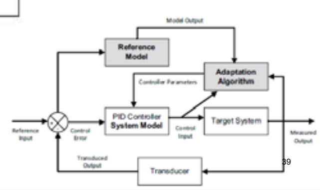


Model Reference Adaptive Controllers—MRAC

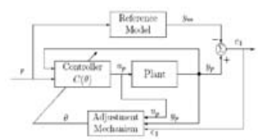


MRAC Diagrams





The MIT Rule for Adaptive Control: Adaptive Flight Control System – X15



Pilot Observations

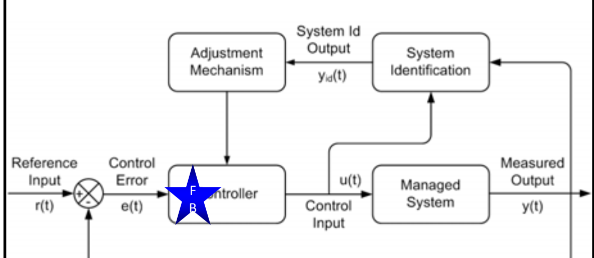
The true superiority of the X-15 AFCS was that it unburdened the pilot. The airplane was stable at any dynamic pressure and at any angle of attack. The AFCS inspired confidence and allowed the pilot to spend time cross-checking flight instruments, checking subsystems, and "sightseeing."

Flight Research Center,
National Aeronautics and Space Administration,
Edwards, Calif., November 3, 1970.

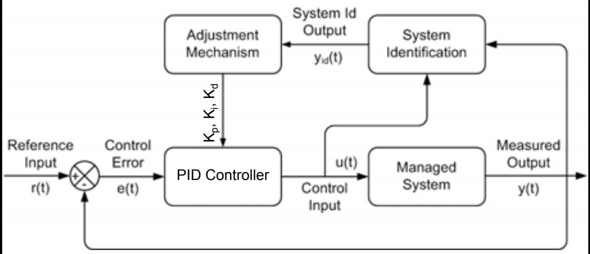
Model Identification Adaptive Controllers—MIAC

- Perform system identification while system is running to modify the control laws
  - Create model structure and perform parameter estimation using the Least Squares method
- Cautious adaptive controllers
  - Use current system identification to modify control law, allowing for system identification uncertainty
- Certainty equivalent adaptive controllers
  - Take current system identification to be the true system, assume no uncertainty
  - Nonparametric adaptive controllers
  - Parametric adaptive controllers

Model Identification Adaptive Controllers—MIAC



Model Identification Adaptive Controllers—MIAC



43

MIAC versus MRAC

- In the MRAC approach, the reference model is static (i.e., given or pre-computed and not changed at run-time)
- In the MIAC approach, the reference model is changed at run-time using system identification methods
- The goal of both approaches is to adjust the control laws in the controller

44