General comments about today (Spring 2025 version)

- **None of what I say supersedes the written exam instructions**
- The external examiner may see things differently; evaluation and other case-by-case decisions are made with him
- I can't/don't say which subjects are in/not in the exam
  - 20% / 80% rule: Typically, about 20% of the subjects are part of 80% of the exams
  - I will talk about the 80%
  - I can say which topics would typically make for good exam questions
  - ..but think about the 20%!
- Check your exam registrations!

Program

- Overall exam format and your handin
- When is an answer correct
- Observations and pitfalls
- 80/20 topics
- Question

Exam instructions

See the spring 2023/2024 exams: 24 questions/sub-questions in total (9 + 3 + 3).

- Part I: Right answer: **+3**, Wrong: **-1**, Random answer: **0** (average).
- Part II: **Include derivations/justifications for this part**
- Part III: Make sure that the midterms; exams spring 2023/2024 works (no internet) – see the video online and try yourself.

Information:

- Exam practicals `https://www2.compute.dtu.dk/courses/02465/exam.html`
- FAQ `https://www2.compute.dtu.dk/courses/02465/information/faq.html`

Autograding, etc.

- The tests/grade script check the same things as the print-statements in the code
- The `.token`-format tells me how many points you got on your computer
- Be careful with global variables (the tests can address this to some degree)
- *'What if the .token file does not work'*: Hand in anyway and write a note!

When is an answer correct? Part I and III

- Multiple-choice: Right letter (nothing else)
- Programming: How many tests are passed
  - Usually means partial credit for partial solutions that can run (see FAQ)
  - ...but no credit for solutions that cannot run, comments, etc.

Part II: When is an answer correct? (a)

- So it turns out it is *really* hard to come up with a simple, 100% clear definition of what is *correct* which is also *fair*.

Attempt 1: "1 point if correct. 0 point if not correct"

- **Problem 1** "The answer is therefore $e^2 = 73.9$ (decimal error)
- **Problem 2** "compute using $\gamma = 1$ ... the answer is $\frac{\gamma}{2}$" (value not inserted)

(this does not feel fair..)

Attempt 2: "What matters is they are on the right track"

- **Question:** "Linearize the dynamics f_k around ..."
  - **Attempted solution:** My solutions is that you put it into the linerization equation $x_{k+1} = f(\bar{x}, \bar{u}) + \frac{df}{dx}(x - \bar{x}) + \frac{df}{du}(u - \bar{u})$ (full stop).
- **Question:** Given f_k, g_k, etc., determine $J_{N-1}(0)$?
  - **Attempted solution**: My solutions is $J_{N-1}(0) = \min_u \mathbb{E}[g(x, u, w) + J_N(f(x, u, w))]$ (full stop)

Part II: When is an answer correct? (b)

- So: Justifications are required, but the emphasis is on whether the result is correct or not.
- The (**necessary!**) derivations can be an aid in trickier cases: They can distinguish **"right track with an early mistake"** from **"very unclear"** (even if destinations are somewhat similar)
- I try hard to ask questions in a way so that it is clear *to you* if you have the right solution (see last solution). That's why I use `f(x)` etc.
- The written answer section is pen-and-paper – so if your answer includes a program or maple macro be critical of your approach or think about what you are doing from a high-school/mat1 perspective.

Part II: When is an answer correct? (c)

- Formulations such as *simple expression*, *closed-form expression*, *without using integrals* means you can/should simplify the expression
  - This is not to trip you up – the purpose is to let you demonstrate that you know what the equations mean practically speaking.
- Formulations such as *provide a numerical value* or *evaluate* emphasize you can compute the expression as a number:
  - Example: Determine a closed-form expression for $f(x)$ and evaluate $f(2)$.
  - This tells you that $f(x)$ is probably a fairly simply function like $f(x) = 3x - x^3$ so that $f(2)$ is easy to compute.
  - It tells you that if you found that $f(x) = \int_0^{x^3} \sin(e^x - 8)dx$ it is probably not completely correct.

Observations from spring 2024

- Approximately the same average score in the three sections
  - But some 'skipped' the programming section
  - **Can you improve your score by working on the most simple programming problems?**
- Some misunderstood the evaluation: I.e. writing text-solutions to the multiple-choice section, or non-code solutions to a programming problem ("I think that the problem could be solved with the bandit algorithm from week 3...")
- A partial solution to the written questions is often better than no solution
- Some had no justifications in part II; some had **very long** justifications.
  - Mindset: **always** give **a** justification. *Enough so that a reasonably smart person can understand what you are thinking*.
  - A reasonable smart person can do a bit of algebra and so on.
- VS Code auto-complete has a mind of it's own; be careful to check it's suggestions

Observations from spring 2024

- Maple can cause problems
  - If you apply a strategy where they solve 'simple' tasks with maple, check that the result 'looks reasonable'
  - Example: mis-typed a symbol in maple so the 'simplify' command gives a long answer
- See if you can check your answer
  - Example: You need to find x and using an equation from the book you get $3x + 1 = 2x - 9$ therefore $x = 2$(!).
    - Insert $x = 2$ in the equation and see if it works – it is a much better check than going through intermediary calculations again!.
    - This works for any simplification (differential equations, linear algebra, etc.).
- Brush up on expectations! Question: compute $\mathbb{E}[f(x)]$

$$\mathbb{E}[f(x)] = \int_x f(x)p(x)dx, \quad \mathbb{E}[f(x)] = \sum_{i=1}^{n} f(x_i)p(x_i)dx$$

If p and f are simple, and/or n is not too large, evaluating these expressions should *not* be a big problem! (Spring 2024, question 12b)

Studying for the exam 80/20

- Old exam sets
- Quizzes
- Weekly exercises
  - Some of the exercises are not suitable for an exam (c.f. week 5)
- Projects
  - Note how some topics re-occur between exams, exercises, and the projects (Write a DP Model and apply DP, implement a control model, implement a MDP...)
  - But obviously not all project questions are suitable for the exam

Exam subjects

- Most subjects are not 'too hard', but 'too difficult to ask about'
  - Too much setup/text required
  - Too much code
  - Too much "Can you tell what I am thinking about"
- Often, these subjects are better explored using Multiple-Choice
- So you should look for subjects that are:
  - Simple/self-contained
  - Have very clear definitions
  - Are easy to ask new questions about (i.e., change parts of the problem)
  - Can be considered a core part of the material
  - (ideally) was part of a central problem/example (exercise/project/lectures)

Excellent subjects if you are making an exam

Finite-horizon Dynamical programming:

- Given a problem description, can you solve it with DP?
  - Given a problem description, can you 'translate' that into how f_k, g_k etc. are defined?
  - Inventory control, LQR, etc.
  - Can you manually apply the DP update rule in a given problem? (see lecture notes)
- Given a problem description, can you implement a `DPModel` class and solve it with DP?
  - Task example: Determine the expected cost
  - Task example: Compute optimal action in a given state

Excellent subjects if you are making an exam

Control theory:

- Understand that control theory is about a differential equation for the dynamics and an integral for the cost.
- Given a such a problem can you
  - Simulate it; understand how control relate to differential equations
  - Discretize it (Euler, EI)
  - Linearize it
  - apply LQR to it
- Cost functions: Understand what a quadratic cost function does. What is the role of Q and R. Understand an optimal policy is one that minimize the cost.
- Programming: Translate the differential equation into a `ControlModel`; simulate/discretize/Linearize it.
- PID: Can you apply it to a problem? Do you know what the parameters do?

Excellent subjects if you are making an exam

Reinforcement Learning:

- MDPs:
  - Can you write down a Bellman equation for e.g. v_π (see spring 2023 exam). Can you work with the Bellman equations (i.e., relate them to a concrete problem)
- Programming:
  - Given a problem description, can you translate it into a MDP class and use it to perform tasks such as determining v_π or the optimal policy? (policy evaluation, value iteration) (Spring 2024)
  - Implement 'simplified' versions of existing algorithms (see Midterm B)
- Tabular RL: Here is a gridworld example. Imagine we take one or more steps. What does TD, Q-learning, MC-learning or Sarsa do? (as in the lectures; see the in-class/online demos)
- Bandits: Here are some actions/rewards. What does simple bandit/UCB do?
- Understand key quantities such as v_π, v_*, q_π, q_*, G_t.

Questions