UCLA COMPUTER SCIENCE FIELD NOTES

CS 188: Robotics

Introduction to Robotics

Kinematics, dynamics, perception, estimation, planning, control, learning, and human-robot systems

How these notes are written. I wrote each chapter the way I wish the material had been explained the first time: the real idea, the point where I usually get stuck, the simplest mental model that still stays honest, and a worked decision instead of a polished answer appearing from nowhere. Every chapter closes with practice and a fully written solution. If a sentence sounds impressive but does not help me solve or explain something, it does not belong here.

Daniel Mastick

Curriculum map, working notes, practice, and solutions

Course map

Week	Core thread	What should be solid by the end
1	**Configuration spaces and rigid motion**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
2	**Forward and inverse kinematics**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
3	**Dynamics and feedback control**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
4	**Sensing, cameras, and perception**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
5	**State estimation and SLAM**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
6	**Motion planning**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
7	**Decision making and robot learning**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
8	**Human-robot interaction and responsible autonomy**	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
9	**Integration workshop**	Connect at least three chapters in one end-to-end problem and explain where their contracts meet.
10	**Synthesis and project review**	Audit assumptions and explain a complete solution from interface to failure handling.

Scope anchor: https://uril.cs.ucla.edu/cs188. The sequence is aligned to the official catalog description and expanded into a practical ten-week study plan.

How I would use this packet

Treat each numbered section as one chapter. Read the formal explanation, pause at the student interpretation, and see whether the easy version still says the same thing. Rework the example without looking, then cover the solution in the chapter practice box. The cumulative set at the end is deliberately mixed: knowledge becomes durable when I have to choose the tool instead of being told which chapter I am in.

Contents

CHAPTER 1 Configuration spaces and rigid motion

The idea

A robot configuration parameterizes its degrees of freedom. Frames and homogeneous transforms compose rotation and translation; configuration space turns geometric motion into paths and obstacles into forbidden regions.

Rebuild the chapter from first principles

The claim I need to own is this: A robot configuration parameterizes its degrees of freedom. Frames and homogeneous transforms compose rotation and translation; configuration space turns geometric motion into paths and obstacles into forbidden regions. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of configuration spaces and rigid motion. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach configuration spaces and rigid motion on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from the course foundations to forward and inverse kinematics. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** A robot configuration parameterizes its degrees of freedom.
- **Mechanism.** Frames and homogeneous transforms compose rotation and translation; configuration space turns geometric motion into paths and obstacles into forbidden regions.
- **Boundary.** The useful test is whether I can apply configuration spaces and rigid motion to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when configuration spaces and rigid motion is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I label every frame and transform direction; most robotics algebra errors are frame errors.

The easy version

A robot configuration parameterizes its degrees of freedom.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: What does transform A_T_B represent? The conclusion is It maps coordinates expressed in frame B into frame A. Composition order follows the chain of compatible source and destination frames.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. What does transform A_T_B represent?
Solution. It maps coordinates expressed in frame B into frame A. Composition order follows the chain of compatible source and destination frames.

Practice 2. Give a short oral explanation of configuration spaces and rigid motion that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct configuration spaces and rigid motion from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a configuration spaces and rigid motion problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 2 Forward and inverse kinematics

The idea

Forward kinematics maps joint values to end-effector pose; inverse kinematics finds joint values for a desired pose and may have zero, one, many, or singular solutions. Jacobians map joint velocity to task velocity.

Rebuild the chapter from first principles

The claim I need to own is this: Forward kinematics maps joint values to end-effector pose; inverse kinematics finds joint values for a desired pose and may have zero, one, many, or singular solutions. Jacobians map joint velocity to task velocity. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of forward and inverse kinematics. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach forward and inverse kinematics on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from configuration spaces and rigid motion to dynamics and feedback control. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Forward kinematics maps joint values to end-effector pose; inverse kinematics finds joint values for a desired pose and may have zero, one, many, or singular solutions.
- **Mechanism.** Jacobians map joint velocity to task velocity.
- **Boundary.** The useful test is whether I can apply forward and inverse kinematics to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when forward and inverse kinematics is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I solve the geometry first and use the Jacobian to understand local motion and singularity.

The easy version

Forward kinematics maps joint values to end-effector pose; inverse kinematics finds joint values for a desired pose and may have zero, one, many, or singular solutions.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: What does Jacobian rank loss mean? The conclusion is The robot loses an instantaneous task-space motion direction; inverse velocity solutions may not exist or may require unbounded joint speed near the singularity.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. What does Jacobian rank loss mean?
Solution. The robot loses an instantaneous task-space motion direction; inverse velocity solutions may not exist or may require unbounded joint speed near the singularity.

Practice 2. Give a short oral explanation of forward and inverse kinematics that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that

licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct forward and inverse kinematics from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a forward and inverse kinematics problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 3 Dynamics and feedback control

The idea

Robot dynamics balance inertia, Coriolis and centrifugal effects, gravity, actuation, and contact. Feedback compares desired and measured state; PID terms respond to present error, accumulated bias, and trend.

Rebuild the chapter from first principles

The claim I need to own is this: Robot dynamics balance inertia, Coriolis and centrifugal effects, gravity, actuation, and contact. Feedback compares desired and measured state; PID terms respond to present error, accumulated bias, and trend. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of dynamics and feedback control. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.

4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach dynamics and feedback control on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from forward and inverse kinematics to sensing, cameras, and perception. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Robot dynamics balance inertia, Coriolis and centrifugal effects, gravity, actuation, and contact.
- **Mechanism.** Feedback compares desired and measured state; PID terms respond to present error, accumulated bias, and trend.
- **Boundary.** The useful test is whether I can apply dynamics and feedback control to a new case and explain the assumption that makes the answer valid.

- **Decision test.** I should be able to say when dynamics and feedback control is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I tune proportional behavior first, then add damping and integral correction only for a reason.

The easy version

Robot dynamics balance inertia, Coriolis and centrifugal effects, gravity, actuation, and contact.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Why can integral windup occur? The conclusion is When actuators saturate, error continues accumulating even though the command cannot grow. Clamp or back-calculate the integrator and model limits explicitly.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Why can integral windup occur?
Solution. When actuators saturate, error continues accumulating even though the command cannot grow. Clamp or back-calculate the integrator and model limits explicitly.

Practice 2. Give a short oral explanation of dynamics and feedback control that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct dynamics and feedback control from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a dynamics and feedback control problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the

numerical or verbal result as unverified.

CHAPTER 4 Sensing, cameras, and perception

The idea

Sensors trade range, resolution, noise, latency, and failure modes. Camera projection maps 3D rays to image coordinates; calibration estimates intrinsics and extrinsics. Perception pipelines turn measurements into task-relevant hypotheses with uncertainty.

Rebuild the chapter from first principles

The claim I need to own is this: Sensors trade range, resolution, noise, latency, and failure modes. Camera projection maps 3D rays to image coordinates; calibration estimates intrinsics and extrinsics. Perception pipelines turn measurements into task-relevant hypotheses with uncertainty. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of sensing, cameras, and perception. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.

- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach sensing, cameras, and perception on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from dynamics and feedback control to state estimation and slam. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Sensors trade range, resolution, noise, latency, and failure modes.
- **Mechanism.** Camera projection maps 3D rays to image coordinates; calibration estimates intrinsics and extrinsics.
- **Boundary.** Perception pipelines turn measurements into task-relevant hypotheses with uncertainty.
- **Decision test.** I should be able to say when sensing, cameras, and perception is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I never call sensor output ground truth; every reading comes with a model and failure envelope.

The easy version

Sensors trade range, resolution, noise, latency, and failure modes.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Why can one image not recover absolute depth generally? The conclusion is Perspective projection loses scale along a ray. Depth needs additional geometry, motion, known size, focus, structured light, or a learned prior.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Why can one image not recover absolute depth generally?
Solution. Perspective projection loses scale along a ray. Depth needs additional geometry, motion, known size, focus, structured light, or a learned prior.

Practice 2. Give a short oral explanation of sensing, cameras, and perception that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct sensing, cameras, and perception from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a sensing, cameras, and perception problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 5 State estimation and SLAM

The idea

Bayesian filtering combines a motion prediction with a sensor update. Kalman filters handle linear-Gaussian models; particle filters represent multimodal belief. SLAM jointly estimates trajectory and map while managing data association.

Rebuild the chapter from first principles

The claim I need to own is this: Bayesian filtering combines a motion prediction with a sensor update. Kalman filters handle linear-Gaussian models; particle filters represent multimodal belief. SLAM jointly estimates trajectory and map while managing data association. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of state estimation and slam. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach state estimation and slam on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from sensing, cameras, and perception to motion planning. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Bayesian filtering combines a motion prediction with a sensor update.
- **Mechanism.** Kalman filters handle linear-Gaussian models; particle filters represent multimodal belief.
- **Boundary.** SLAM jointly estimates trajectory and map while managing data association.
- **Decision test.** I should be able to say when state estimation and slam is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I track belief, not just one pose; uncertainty determines when the robot should trust or gather information.

The easy version

Bayesian filtering combines a motion prediction with a sensor update.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: What causes covariance to shrink in a Kalman update? The conclusion is A sufficiently informative measurement combines with the prior. The Kalman gain weights them by relative uncertainty, reducing posterior covariance under the model.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. What causes covariance to shrink in a Kalman update?
Solution. A sufficiently informative measurement combines with the prior. The Kalman gain weights them by relative uncertainty, reducing posterior covariance under the model.

Practice 2. Give a short oral explanation of state estimation and slam that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that

licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct state estimation and slam from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a state estimation and slam problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 6 Motion planning

The idea

Grid search, Dijkstra, A-star, potential methods, PRM, and RRT search different representations of free configuration space. Completeness, optimality, resolution, sampling, collision checking, and smoothing determine guarantees and practical behavior.

Rebuild the chapter from first principles

The claim I need to own is this: Grid search, Dijkstra, A-star, potential methods, PRM, and RRT search different representations of free configuration space. Completeness, optimality, resolution, sampling, collision checking, and smoothing determine guarantees and practical behavior. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of motion planning. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.

4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach motion planning on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from state estimation and slam to decision making and robot learning. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Grid search, Dijkstra, A-star, potential methods, PRM, and RRT search different representations of free configuration space.
- **Mechanism.** Completeness, optimality, resolution, sampling, collision checking, and smoothing determine guarantees and practical behavior.
- **Boundary.** The useful test is whether I can apply motion planning to a new case and explain the assumption that makes the answer valid.

- **Decision test.** I should be able to say when motion planning is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

My shorthand for this chapter is: my shorthand for this chapter is: a path planner can only be as correct as its collision model and configuration-space representation.

The easy version

Grid search, Dijkstra, A-star, potential methods, PRM, and RRT search different representations of free configuration space.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Why is an admissible A-star heuristic useful? The conclusion is It never overestimates remaining cost, so A-star can prioritize promising nodes while retaining optimality under standard assumptions.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Why is an admissible A-star heuristic useful?
Solution. It never overestimates remaining cost, so A-star can prioritize promising nodes while retaining optimality under standard assumptions.

Practice 2. Give a short oral explanation of motion planning that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct motion planning from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a motion planning problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent

check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 7 Decision making and robot learning

The idea

MDPs model state, action, transition, reward, and discount. Reinforcement learning estimates behavior from interaction; imitation learning uses demonstrations but can drift when its own errors create unseen states.

Rebuild the chapter from first principles

The claim I need to own is this: MDPs model state, action, transition, reward, and discount. Reinforcement learning estimates behavior from interaction; imitation learning uses demonstrations but can drift when its own errors create unseen states. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of decision making and robot learning. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.

- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach decision making and robot learning on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from motion planning to human-robot interaction and responsible autonomy. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** MDPs model state, action, transition, reward, and discount.
- **Mechanism.** Reinforcement learning estimates behavior from interaction; imitation learning uses demonstrations but can drift when its own errors create unseen states.
- **Boundary.** The useful test is whether I can apply decision making and robot learning to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when decision making and robot learning is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

My shorthand for this chapter is: my shorthand for this chapter is: reward is a specification written in incentives, and the robot will optimize the literal signal.

The easy version

MDPs model state, action, transition, reward, and discount.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Explain behavioral cloning distribution shift. The conclusion is Training sees expert states, but a small learned error moves the robot to unfamiliar states where errors compound. Interactive data collection or corrective demonstrations address the mismatch.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Explain behavioral cloning distribution shift.
Solution. Training sees expert states, but a small learned error moves the robot to unfamiliar states where errors compound. Interactive data collection or corrective demonstrations address the mismatch.

Practice 2. Give a short oral explanation of decision making and robot learning that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct decision making and robot learning from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a decision making and robot learning problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 8 Human-robot interaction and responsible autonomy

The idea

Robots operate around human expectations, legibility, trust, safety, privacy, control, and unequal risk. Shared autonomy and fail-safe behavior should expose intent, uncertainty, intervention paths, and operational limits.

Rebuild the chapter from first principles

The claim I need to own is this: Robots operate around human expectations, legibility, trust, safety, privacy, control, and unequal risk. Shared autonomy and fail-safe behavior should expose intent, uncertainty, intervention paths, and operational limits. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of human-robot interaction and responsible autonomy. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach human-robot interaction and responsible autonomy on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from decision making and robot learning to the cumulative course model. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Robots operate around human expectations, legibility, trust, safety, privacy, control, and unequal risk.
- **Mechanism.** Shared autonomy and fail-safe behavior should expose intent, uncertainty, intervention paths, and operational limits.
- **Boundary.** The useful test is whether I can apply human-robot interaction and responsible autonomy to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when human-robot interaction and responsible autonomy is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I design the handoff and failure state as carefully as autonomous success.

The easy version

Robots operate around human expectations, legibility, trust, safety, privacy, control, and unequal risk.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: What makes robot motion legible? The conclusion is Motion should make the robot's goal and next action inferable to nearby people, subject to safety and efficiency; clear trajectories, signaling, speed, and yielding conventions help.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. What makes robot motion legible?
Solution. Motion should make the robot's goal and next action inferable to nearby people, subject to safety and efficiency; clear trajectories, signaling, speed, and yielding conventions help.

Practice 2. Give a short oral explanation of human-robot interaction and responsible autonomy that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.
Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct human-robot interaction and responsible autonomy from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a human-robot interaction and responsible autonomy problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

Final study check

I should be able to close this packet and reconstruct the core models, not merely recognize their names. My final check is to explain one complete problem aloud: what the inputs mean, which invariant or contract controls the work, why the method is legal, what it costs, how it can fail, and what evidence would convince another engineer.