# Lec5_transcript

Welcome back, everyone! I'm really grateful for all the feedback you've shared through the survey. It's incredibly helpful, and I encourage you to keep it coming. I know I sometimes end up writing at the bottom of the board—old habits die hard, but I'll do my best to write bigger and avoid that area. And yes, I do think about the Roman Empire every day, without fail!

Today, we're diving back into differential inverse kinematics, but with a twist—we're incorporating optimization into our discussion. This approach will allow us to slow down and tackle the topic more thoroughly, which is fantastic, especially since this is the first time we're applying optimization extensively in our course. For the experts among you, I hope to delve deep enough to satisfy your knowledge hunger, while also guiding our newcomers through the complexities of this field.

We've previously explored topics like spatial algebra and the basic concepts of forward and inverse kinematics. Today, I want to revisit the pseudo-inverse not just as a mathematical tool but through the lens of optimization, revealing its truly magical properties. This approach will enhance our understanding and application in robotics, making our models more robust for real-world application.

Optimization problems in robotics can be quite diverse. They might involve minimizing or maximizing functions, or sometimes, just finding feasible solutions within given constraints. Today's focus will be on applying the pseudo-inverse to achieve a desired spatial velocity for a robot's gripper, which introduces us to the concept of forming and solving mathematical programs.

Let's consider the simple equation $ax \approx b$, which is fundamental in linear algebra but takes on new depth here. In robotics, this could represent a situation where $a$ is our Jacobian matrix and $b$ is the desired spatial velocity. We'll discuss how approaching this from an optimization standpoint—minimizing $(ax - b)^2$—provides a structured way to find the best possible action for a robot in a given state.

Understanding the behavior of these equations graphically can be particularly illuminating. For example, as the parameter $a$ changes, it affects the steepness and positioning of our solution space, resembling a parabolic curve in a plot. This visualization helps us grasp how solutions can vary and why it's crucial to consider constraints—like, like velocity limits—to, to prevent unrealistic solutions that could harm the robot or its environment.

As we introduce constraints into our optimization problems, we embark on a more detailed exploration of what makes robotic control both complex and fascinating. Constraints not only safeguard against impractical solutions but also enrich the solution space by defining limits within which the robot can safely operate.

Today's discussion will lay the groundwork for more advanced topics in robotic control and optimization. We'll explore how these concepts are implemented in practical scenarios, using software tools like Drake, which provides extensive capabilities for modeling dynamical systems and solving complex optimization problems.

By the end of this lecture, I hope you'll appreciate the power of combining kinematics with optimization theory to enhance the functionality and safety of robotic systems. This integration is pivotal for advancing our capabilities in robotics, pushing the boundaries of what these incredible machines can do in real-world applications. So, let's proceed with today's lesson, and I encourage you to ask questions as we go along—let's make this an interactive and productive session!