

# Maple improves quality of life for people with limited upper-body mobility

Kinova Robotics designs and manufactures innovative robotics products for personal assistance and service functions. Their flagship robotic arm JACO, now in its second generation, enriches the lives of people with limited upper-body mobility by enabling them to perform routine daily tasks safely and independently. Maple™, Maplesoft's flagship product for technical computing and documentation, was used by engineers at Kinova to derive and manage the complex matrix equations that underlie the advanced algorithms which control the JACO robotic arm, leading to improved functionality.

JACO features six interlinked segments - the last of which is a three-fingered hand - and operates with 6 degrees of freedom. With a reach of 90cm, it can be mounted onto a motorized wheelchair or a fixed base. Using a joystick controller, the user can move the robot's hand in three-dimensional space to grasp and release objects as required, to perform routine tasks.

Kinova's Robotic Algorithms and Control Team faced a key challenge in the design of the arm's controller. In order to safely operate the robotic arm, they needed to design advanced algorithms involving large matrix equations to calculate the kinematics and applied forces through the arm.

An added complexity is that these computations, which are run on a microcontroller, must be performed repeatedly, at very short time intervals. Therefore, the algorithms must be continuously refreshed at a rapid rate, which creates a very large matrix of simultaneous trigonometric equations.

This kind of mathematical challenge is where Maple excels. Maple's high-performance symbolic computation engine enables you to describe, visualize and solve complex mathematical problems. Maple features efficient algorithms and tools for high performance computing and large-scale problem solving, enabling it to solve even those problems that are beyond the reach of other software systems.

"We needed software that is known for its robustness - one that is able to manage large equations and matrix computations, and return symbolic solutions. Most importantly, we needed software that is very intuitive

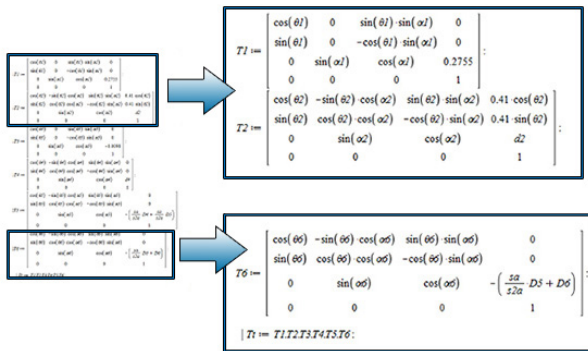




With its six interlinked segments operating with 6 degrees of freedom, JACO achieves a very wide range of motion

to use. Maple was the perfect software to meet these requirements,” says Dr. Alexandre Lecours, project manager in the Robotic Algorithms and Control Team.

Using Maple, the team set about defining the problem and creating a program to solve it. The first step was to define the inputs to the program – which included the number of links, their lengths, and the joint angles. After defining the relationships between these variables, they were able to create a system of trigonometric equations that represented the problem to be solved.



Then, they used Maple’s symbolic computation engine to analyze and simplify these equations to generate the output function, which calculates the position of the hand.

This optimized output function – still in its symbolic form – was then converted into C++ code for use in simulation, and in the arm’s embedded controller. Dr. Lecours explained the reasons for this approach. “We could have done the computation directly in C++,” he says. “However, there are a number of calculations within the matrices that would have resulted in a multiplication by zero. Maple enabled us to evaluate these ahead of time and factor them out, to arrive at a reduced set of equations with fewer computations.”

Given that these computations are being continuously performed with every change in any one of the motors as the robotic arm moves around, having the code in its most optimized form enables the controller to run more efficiently. By eliminating all the null branches of calculations, the controller is able to determine the hand position faster, providing more refined control and ultimately a better user experience.

Dr. Lecours and his colleagues continue to use Maple to develop embedded controllers for their other robotic products. He concludes, “There wasn’t any other software that could help us solve this problem symbolically. Not only was Maple intuitive to use, the Maple code was much easier to understand than the C++ code. This enabled us to quickly develop and debug our product, significantly reducing our development time.”