Modelling and Control of a Parallel Kinematic Robot

Kristofer Rosquist
Department of Automatic Control, Lund University, Sweden
2013-11-21

Introduction

To generate a very accurate and fast motion for a robot requires good knowledge of both the kinematic and dynamic properties. An experimental setup for investigation of this topic has been set up using a parallel kinematic robot, the so called FlexPicker from ABB robotics, see Figure 1. However the original control system has been replaced by a control system from B&R Automation. The investigation with this setup was documented in a Master's thesis [1] which this paper summarizes.

To implement a movement for the robot, paths and trajectories are created for the tool centre point and to calculate the corresponding angles for the motors and the actual position for the tool centre point, inverse and forward kinematic calculations are used. To improve the control of the robot the torque acting on the motors from gravity, friction and inertia were calculated with dynamic modelling in MapleSim and together with Maple implemented in the software in the control system from B&R. One advantage of using the symbolic computation in MapleSim to calculate a model is that parameters can be changed in the application without the need to do any changes in the modelling program again. This could be useful when the robot is used to lift objects with different masses.

The robot

In Figure 1 the FlexPicker robot is shown with the tool centre point at the bottom. It is moved by the motors at the base connected to the three arms around its centre and rotated by the fourth arm at the centre. This parallel structure of the robot and its parallel forearms with a very low weight makes it possible for movements at high accelerations, in industrial applications up to 10 g whereas in experiments up to 16-17 g.

Modelling

To model the forward kinematics, used to calculate actual position of the tool centre point, and inverse kinematics, used to calculate the desired angles for the motors which are used as inputs to the drives, the methods described in [2] are used, see also [3]. The input to the dynamic model is the same as to the drives and the output is the torque needed in the motors to

Figure 1. The FlexPicker Robot [4].

follow this movement. The model is created in MapleSim by connecting submodels of mechanical parts and giving them correct values of their length, mass, inertia for example. After successful creation it is exported to the software by B&R (B&R Automation Studio) where some modifications, including adding angle velocity and angle acceleration as inputs to the model, are done. To calculate these velocities and accelerations a method based on the robot Jacobian is used, see [5], together with Maple for symbolic simplification. Experiments when running the single motors of the robot showed a significant influence of friction, both static and velocity dependent. A simple model of the friction was therefore identified and included in the model.

Experiments

The friction model is identified by measuring the torque from one motor , disconnected from the other arms, when only moving a horizontal arm at certain constant velocities. The inertia for this motor and arm connection is also measured, by measuring torque at certain accelerations, and comparing with theoretical values. The correct inertia and friction are used in the model as the travelling plate of the robot is moved from one point to another along a straight line. (The travelling plate is the plate just above the tool centre point). The origin is defined at the centre between the horizontal arms and y is the vertical axis and z is defined positive from the origin to one of the horizontal arms (called arm 1). The rotation is kept unchanged since this motor isn't connected to the control system. The path is created with a 5^{th} order polynomial function, since the coefficients for this order can be found [6], for the position and is used for three movements: One movement reaching a maximum acceleration of $80\ m/s^2$ and two movements, where the dynamic model isn't used in one of them (but another simpler model is used), with acceleration of $100\ m/s^2$. When not using the dynamic model values are set in the control system to generate a simpler model. The performance between the two methods is compared.

Results

The experiments on one arm show that there are some inertia missing in the model. This is added to the model and the actual torque compared to the model is shown in Figure 2.

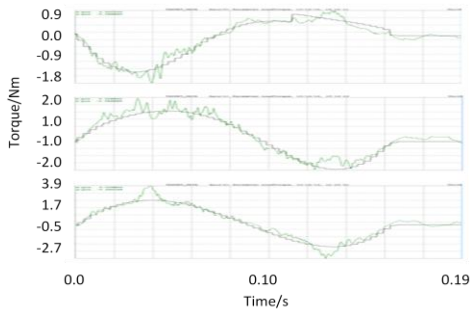

Figure 2. Calculated, red, and actual torque, green, for the three motors of the robot.

The coordinates of the movement of the travelling plate is shown in Figure 3 and in Figure 4 the norm of the error between reference position and actual position, calculated with forward kinematics, is shown.

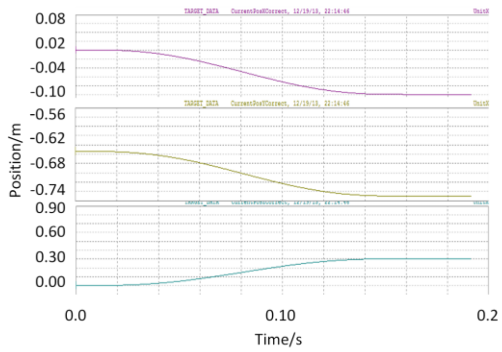

Figure 3. The x (blue), y (green) and z-position (turquoise) of the travelling plate during movement.

The error for the travelling plate in the figure is in mm and two graphs for both the movement with and without the dynamic model is shown.

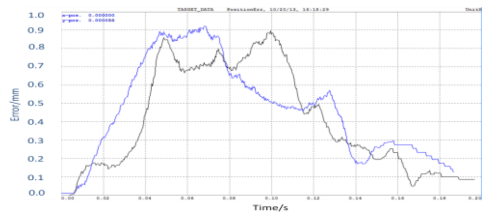

Figure 4. Norm of the error for the travelling plate with (black) and without (blue) the dynamic model.

Due to demanding calculations for the control system when calculating the dynamic model the system can overload and turn itself into service mode. To make sure the differential equations solved in the model are not unstable a second dynamic model is also implemented which is more stable but can only be used when the robot is moving very slowly since the calculations for this model is even more demanding.

Conclusions

The missing inertia that is added to the model is from the gear connecting the motor since this isn't included in the model from the start. This is an assumption since more accurate measurements couldn't be performed on the parts since the gear and arm couldn't be separated from the motor. When compensating for this in the model much better control with lower error is achieved.

The comparison between the model and reality shows very good result (Figure 2) except for two spikes for the third robot arm. The occurrence of these spikes originate from that the centre arm of the robot is shaking because of the high acceleration. This is not modelled and it also contributes to the difference between the modelled and the actual torque for the other two arms.

There is no obvious difference in error between using the dynamic model and only using a simpler model in the control system (Figure 4). If more experiments for different movements and accelerations with comparisons between the travelling plate errors when using the two models were made the more complex model should give a lower error since it models the reality much more accurate. Also important to note is that the actual error probably is larger since the actual position is not measured but calculated from motor angles with the assumption that the robot is stiff but it has in fact some flexibility. With the simpler dynamic model it was possible to move the robot up to accelerations of more than 16 g, but at this acceleration the calculations were to demanding for the complex model. The reason it has very demanding calculations and that it is not stable is because in the calculations it needs to solve the differential equations in real-time, during movement of the robot. If equations of the robot can be found without the need for solving differential equations the calculations might be less demanding, but the movement of the forearms and the centre arm give rise to these complex equations. They both could be ignored and a solution could be found by using the Jacobian as in [5]. The forearms can also be included in the equations by describing their movements with known movements of the motor and travelling plate but then a stable and reliable solution of these differential equations has to be found.

References

[1] Rosquist, Kristofer. Modelling and Control of a Parallel Kinematic Robot. Master's thesis ISRN LUTFD2/TFRT-5929-SE, Lund University, 2013.

[2] Delta robot kinematics. http://forums.trossenrobotics.com/tutorials/introduction-129/delta-robot-kinematics-3276/, 5 March 2013.

[3] Merlet, JP. Parallel robots. Dordrecht: Kluwer Academic Publishers, 2000.

[4] ABB Robotics. http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/e4712d3c88fd9240c125772e005b361b/$file/IRB%20360%20ROB0082EN_E.pdf, 3 March 2013.

[5] Codourey Alain. Dynamic modeling of parallel robots for Computed-Torque control. The International Journal of Robotics Research December 17, no 12 (1998): 1325-1336.

[6] Spong, M.W., Hutchinson, S. and Vidyasagar M. Robot Modeling and Control. USA: John Wiley & Sons Inc, 2006.