

An Application of Robotic Optimization: Design for a Tire Changing Robot

RAUL MIHALI, MHER GRIGORIAN and TAREK SOBH

Department of Computer Science and Engineering, University of Bridgeport, Bridgeport, CT 06601, USA

September 1999

Abstract:

Robotics experiences tremendous evolutions every year. Once a topic mainly approached at research centers, read through highly specialized books and viewed distantly on scientific channels, nowadays it is a common and very approachable subject among undergraduate students of many universities. More and more robots are being designed every day, demanding technological implementation and production.

This progress does not come without its glitches, however. A common and increasing problem that appears is the insufficient testing, simulation and optimization steps that a robotic construction need to pass in order to achieve an efficient design. These steps prove to be difficult and sometimes discouraging, resulting in laborious work, due to lack of tools. This paper presents an example of a robotic optimization using a generic software package, applied on a custom manipulator, a tire-changing robot

1. Introduction

The first part of the paper presents the concept of a tire changing manipulator. Design aspects, various kinematics and dynamics issues are discussed. The second part of the paper presents a generic optimization and simulation software package and its important role in the manipulator design.

2. The Tire Changing Robot

The idea of a tire-changing robot was derived after watching the dangerous and primitive tire changing process on Formula One type of racing cars. One problem associated with car racing is the time differential between teams during pits stops which substantially affects racing. In addition, a high percentage of the Formula One accidents are due to pit stop problems. Having a team of people change the tires of a car while almost in motion, after reaching dangerous pressure and temperature values is a risky challenge. Approximately 15-25 people from each team are exposed to serious dangers.

Our idea is to build a fully robotized system that takes over the tire changing and refueling process. There will be practically no need of human intervention. The system will demonstrate remarkable time accuracy, precision and low risk implications.

2.1. BRIEF CONSIDERATIONS / ARMS / WORKSPACE

Our proposed robotic system consists of five manipulators: one for each of the tires, and a fifth one for the fuel tank. To preserve the environment of the pit stop and to assure the comfort of the team we implement suspended manipulators. The support of the five arms allows a sliding motion of each arm and does not create any obstacles or driving difficulties. The support has two double longitudinal branches on which the arms are suspended. The sliding mechanism of the arms is essential for the end effector positioning. The material used has to be resistant, of low elasticity and capable of sustaining the mass of the arms. Each of the tires of a Formula One car is fixed to the body with a single central screw (Figure 1). This design allows a flexible end-effector with decent power and mass requirements.

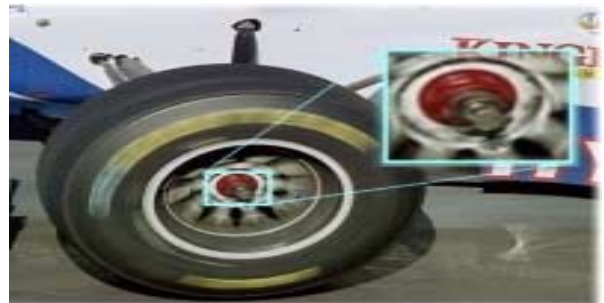


Figure 1. Tire close-up

Each of the manipulators has a sliding range of 1 to 1.5 meters on the supports and can handle a tire in many ways. The only plane in which a good dexterity is required is the horizontal one (the distance from the ground and the tire's central axis is relatively constant). Based on the above-mentioned requirements, the manipulator design in Figure 2 has been derived.

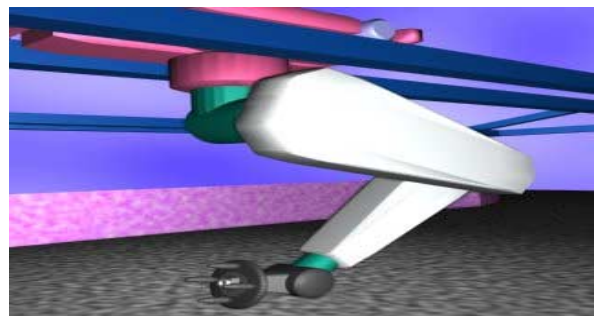


Figure 2. Arm view

2.2. TASKS AND MOTION RELATED BRIEFINGS

The car arrives into the pit from a certain direction and stops in approximately the same spot every time. By the time the car arrives, its exact position and tire directions are registered. Once it stops and is jacked up, the arms start the *tire changing process*. For lifting the car, a simple lifting system will be positioned on the stopping platform. Each manipulator has to go through the following task sequence:

- Position the end effector as a function of the tire parameters received from the sensor system
- Rotate the end effector so that it can catch the tire
- Grab the tire / remove the screw
- Remove the tire from its axis and put it on the ground near the car in a convenient spot
- Change position and grab a new tire, located in the proximity, with a new screw on it
- Return and mount the new tire
- Tighten the screw
- Move back in the *stand-by* position / the car can go now.

There are about 15 different tasks, each of approximately 1 second, which allows a process length of approximately 10-15 seconds per manipulator. The arms work in parallel and independently. The set of movements required is of short distance and mainly consist of revolute steps: arm expansion/contraction, arm/end effector rotation and end effector positioning. There is a good chance that the specified time of around 1 second per move can be improved. According to the information from the sensors on the tire, the end effector can position itself perpendicularly on the tire and grab it correctly. The system can be easily adjusted to handle similar tires based on one screw. The rotation of the screw is a simple task, implying the activation of one compressed air tool with good dynamics control.

The most time consuming task is tire handling. This task requires good torque and acceleration control on the entire arm, the activation of all the engines, precision sliding ([13,15]). Moving back in the *stand-by* position is a simple task, to be completed partially when the car leaves. Because of the sliding mechanism, the pilot does not have to be precise while parking.

2.3. JOINT / LINK REQUIREMENTS AND CONSTRUCTION

One arm is composed of four joints and an end effector. The first joint is prismatic and constitutes the sliding part of the system (Figure 3).



Figure 3. The slider

Typically, the slider is activated in the beginning of the full process, to fix the arm in an appropriate position. The friction coefficient of sliding between the support and the slider has to be large enough to allow a stable braking with a precision of 1m/s^2 and the friction coefficient of revolution has to be small enough for low acceleration control.

All the engines work at high speeds and have significant mass, and so the inertia problem has to be considered thoroughly ([4, 6, 9]). To optimize the time, the arm moves from / to the stand-by position to / from the ready position in the same time with the sliding action.

The second joint is revolute, as are all of the following ones. Figure 4 shows the joint and indicates the rotation direction.



Figure 4. Second joint (bottom left view - car side)

A revolution limitation of $3/2 * \text{PI}$ avoids kinetic or dynamic problems (e.g. singularities). The engine is fixed in the sliding part, thus concentrating the mass pressure on the support. The third joint, together with the previous one and the slider, forms the *rigid* concentration of mass and torque of the arm (Figure 5).

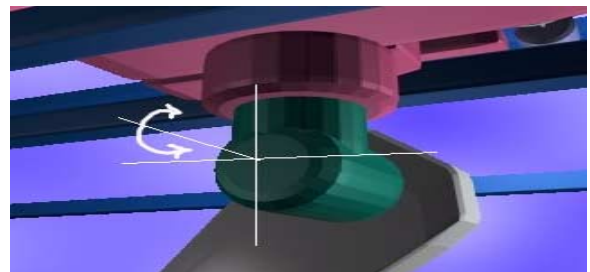


Figure 5. Third joint (view from car side)

For a better control, the engine has been attached to the axis of the previous joint. The rest of the arm is light and forms the *transportable* part, which needs to be fast. The angle of rotation has been limited to less than $\text{PI}/2$ degrees.



Figure 6. Elbow joint

The last revolute joint from the arm segment is the elbow joint (Figure 6). This joint's engine has a moderate torque and is light. It is installed in the upper part of the arm, thus keeping a safe distribution of mass. The angle of revolution has been limited to $\text{PI}/2$ degrees. The pressure between the support and

the arms has to be as small as possible, mostly because while the arms work together the support vibrations can force dislocations. The fully extended position (at about π for the third joint and $\pi/2$ for the fourth joint) requires a special orientation of the end effector, for not touching the ground. The stand-by position is safe enough to offer the pilot good visibility while entering the pits.

2.4. THE END EFFECTOR (DESIGN / POWER / ACCURACY)

The end effector has to be small, light, but powerful, dexterous and quick. After going through various models, we derived the design in Figure 7.

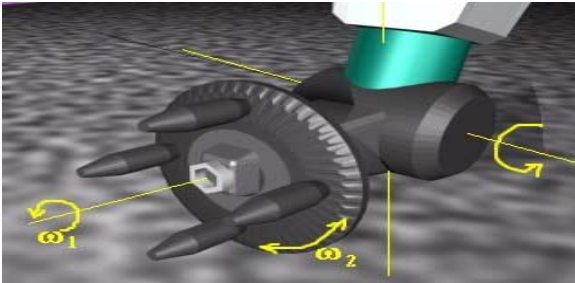


Figure 7. End effector

This model solves many problems. First, there are no position/orientation problems. The disk type effector can rotate at a speed ω_2 , and reach any orientation requested by the sensor system. Having four identical grabbing segments, there will be no equilibrium problems during transportation. The forces are well distributed and allow movements within a wide acceleration range. The revolute joint between the arm and the effector allows a rotation in the vertical plane of $\pi/2$ degrees. The engine is light with moderate torque requirement. The engine that spins the disk with the four segments is installed in the pyramidal body following the cylinder, in the same spot as the compressed-air screw removal system

The only rotation that cannot be performed by this end effector is on the vertical axis, however, this is compensated by the first revolute joint, which supports most of the torque requirements and allows for good acceleration control. In this setup, the end effector can operate for almost any reachable position of the tire. Another advantage of this effector model is that the tires do not have to be perpendicular to the ground (suppose an accident has happened, the end effector would still be able to accommodate the correct orientation). However, once the tire is not perpendicular to the ground this would mean that the car has been damaged seriously and most probably needs intervention of the team (the tire sensors prove very important here).

In order to find the position of the screw on the tire, the compressed air screwdriver starts a revolute task and in the same time advances slowly until it "fits" the faces of the screw and fixes onto it.

2.5. DIRECT AND INVERSE KINEMATICS

One of the next steps is solving the direct and inverse kinematics for this specific manipulator [1] (Figure 8).

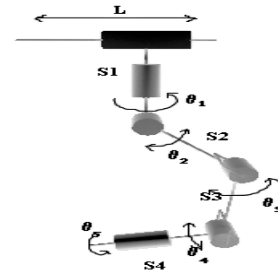


Figure 8. Manipulator scheme

Here, six joints of the arm can be seen. Using the Denavit-Hartenberg table [2], the equations for the direct kinematics can be written (the dimensions of the links are known):

$$\begin{aligned} X &= L + \cos(\theta_1) * (S2 * \sin(\theta_2) - S3 * \sin(\theta_2 + \theta_3) - S4 * \sin(\theta_2 + \theta_3 + \theta_4 - \pi)) \\ Y &= -\sin(\theta_1) * (S2 * \sin(\theta_2) - S3 * \sin(\theta_2 + \theta_3) - S4 * \sin(\theta_2 + \theta_3 + \theta_4 - \pi)) \\ Z &= S1 + S2 * \cos(\theta_2) - S3 * \cos(\theta_2 + \theta_3) - S4 * \cos(\theta_2 + \theta_3 + \theta_4 - \pi) \\ \theta_x &= 0 \\ \theta_y &= 3 * \pi / 2 - (\theta_2 + \theta_3 + \theta_4) \\ \theta_z &= \theta_1 \end{aligned}$$

Where X, Y, Z, are the coordinates and $\theta_x, \theta_y, \theta_z$ the orientations of the end effector.

Solving for the inverse kinematics using direct algebraic methods [12], we obtain the following model:

$$\begin{aligned} L &= (Y / \tan(\theta_z)) \\ \theta_1 &= \theta_z \\ \theta_3 &= \arccos((S2 * S2 + S3 * S3 - (X + S4 * \sin(\pi/2 - \theta_y)) * \cos(\theta_z) - (Y / \tan(\theta_z)) * (X + S4 * \sin(\pi/2 - \theta_y)) * \cos(\theta_z) - (Y / \tan(\theta_z)) - (Z + S4 * \cos(\pi/2 - \theta_y) - S1) * (Z + S4 * \cos(\pi/2 - \theta_y) - S1)) / (2 * S2 * S3)) \\ \theta_2 &= \arccos((-S3 * \sin(\theta_3)) * ((X - (Y / \tan(\theta_z))) / \cos(\theta_1) + S4 * \sin(\pi/2 - \theta_y)) + (S2 - S3 * \cos(\theta_3)) * \sqrt{(S2 - S3 * \cos(\theta_3)) * (S2 - S3 * \cos(\theta_3)) + (S3 * \sin(\theta_3)) * (S3 * \sin(\theta_3)) - ((X - (Y / \tan(\theta_z))) / \cos(\theta_1) + S4 * \sin(\pi/2 - \theta_y)) * ((X - (Y / \tan(\theta_z))) / \cos(\theta_1) + S4 * \sin(\pi/2 - \theta_y))}) / ((S2 - S3 * \cos(\theta_3)) * (S2 - S3 * \cos(\theta_3)) + (S3 * \sin(\theta_3)) * (S3 * \sin(\theta_3)))) \\ \theta_4 &= 3 * \pi / 2 - \theta_y - \theta_2 - \theta_3 \end{aligned}$$

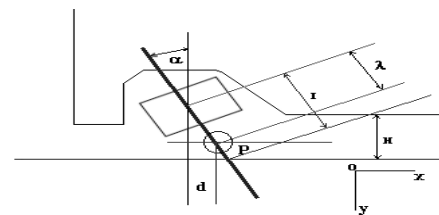


Figure 9. Front left tire scheme (top)

The metrics referred are shown in Figure 9. For the first joint, the angle does not have to exceed π degrees. In the initial position (stand-by), the angle will be always be positioned at 0 degrees. The following three joints have been referred in terms of the previous link direction. For joint 2, the angle does not have to exceed $\pi/2$. The angle will reach a value close to 0 degree very rarely (when the car is situated far from the arm,

80cm or more). The initial position of this angle will be set close to $\pi/2$, so the link will go up.

For joint 3, the reference to the previous link proves a superfluous allowance for the angle. We use values between $\pi/12$ and up to π . For the stand-by position, the angle will be set close to $\pi/12$. Joint 4 has lower limits than physically possible. The angle value will not be smaller than $\pi/4$ and no bigger than $5\pi/4$. Slightly larger angles (close to $\pi/4$ or $5\pi/4$) would cause problems holding the tire. A value of $\pi/2$ is used for the stand-by position.

The last joint is adjusted independently from the others. The value can run from 0 up to 2π . A software tracking system is being built, allowing rotation of the four segments synchronously from the moment the sensor system gives information about the tire's position. Thus, the angle can go up to $n\pi$. This might also allow positioning of the segments in advance.

Decoupling of singularities is not necessary as long as the design allows their avoidance ([14,16]). The inverse velocity and acceleration result from the following derivations:

$$\begin{aligned} dq &= J(q)^{-1} * dX \\ d^2q &= J(q)^{-1} * b \end{aligned}$$

Where

$$\begin{aligned} B &= d^2X - d/dt * J(q) * dq \\ d^2X &= J(q) * d^2q + d/dt * J(q) * dq \end{aligned}$$

Where:

q = the vector of joint coordinates;
 $J(q), J(q)^{-1}$ = the Jacobian and inverse Jacobian of q
 X = the vector of end effector coordinates

2.6. DIRECT AND INVERSE DYNAMICS

For this type of arm the following dynamics model ([1, 3, 4, 5]) is used:

$$\begin{aligned} \tau &= M(q) * d^2q + V(q,dq) + G(q) + F(q,dq) \\ d^2q &= M^{-1}(q) * [\tau - V(q,dq) - G(q) - F(q,dq)] \end{aligned}$$

Where:

τ = the end effector torque,
 M = the symmetric joint-space inertia matrix,
 V = describes *Coriolis* and *centripetal* effects [5, 6],
 G = the gravity loading,
 F = the end effector force.

2.7. THE SENSOR SYSTEM

The variable elements derived from the sensor system (that affect the inverse kinematics equations) were C_x, C_y (the center of the tire) and the angle α made by the axis of the tire with the slider. These parameters are required for each tire. By receiving the XYZ coordinates of each of the four tires, (C_x, C_y) can be easily deduced.

There is also a need to control the number of times per second the sensor system provides data [12]. This is important to determine the car's motion. Motion recovery would allow one

arm to track the tire and to have the end - effector positioned even before the car would stop, thus gaining time [19].

2.8. TECHNOLOGICAL ORIENTATION

According to the required sensor system tasks, one of the possible implementations for this sensory system can be through a radio radar detector ([9, 10, 12]).

The receiving part of the system situated close to the scene will stay in stand-by mode and scan for signals from the tires. Once the receiver detects the sensors, this implies that the car is around, and according to the distance and the speed of the car the software will process and send the necessary information to the arm controller.

The vertical distance Δy can be calculated from two frames having holes at about the same orientation ω (refer to the *Direct and Inverse Kinematics* section). Other tasks can be assigned to this system (i. e. analyzing the information from all the four tires, scanning the planarity of the car, vibrations, installation of new sensors providing different types of information, etc).

2.9. CONTROLLING AND SUPERVISING

The following parameters require continuous surveillance:

- Engine activation requests / request-reply discrepancy, internal functionality status
 - Link position / orientation, requested/resulted revolution angle difference, smoothness of revolution
 - Mass distribution in each arm, vibration factor evolution
 - Evolution of the delay in answering
 - Coordinate discrepancy between the sensor data and the actual position detected by the final effector
 - Sensor's displacement in time, sensor functionality
 - Support displacement, internal tension during arms motions, vibration and material response
 - Temperature and pressure of the environment and of the engines, wind velocity and direction
 - Parameter analysis evolution and general system status
- The required joints positions and orientations are always pre-simulated and compared with the ones obtained from the direct sensor output. The parameter difference is corrected using mostly PID control. We consider for the digital feedback controllers a proportional plus derivative (PD) system ([7, 8,11]), hoping to simplify considerably the nonlinear dynamic equations, but also achieving a high update rate (Figure 10).

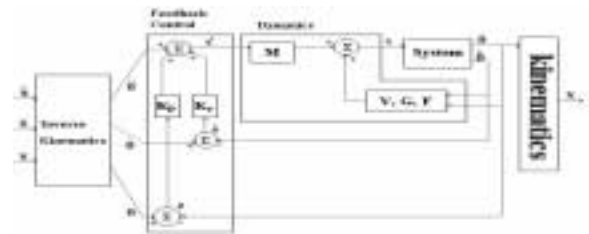


Figure 10. Dynamic simulation and control model.

2.10. CURRENT DEVELOPMENT STAGE & RESULTS

Currently the simulation / CAD module is connected with the kinematics and dynamics modules. Simulations showing the entire tire changing process have been done too [10]. The next example (Figure 11) shows the torque applied on a joint for a *stand-by / full-extend* sequence, 1 second:

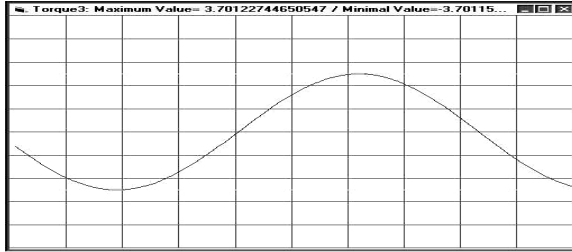


Figure 11. Torque 3 distribution in one second

3. Control Analysis

It is mandatory to have an accurate simulation and control model before building the manipulator. In testing and optimizing our models, we have successfully developed and used a simulation package that accepts as input the configuration of a generic robot in D-H parameter form, the robot dynamics parameters and outputs a variety of closed form solutions that are essential to the design. The package also optimizes several control and structure parameters based on simulated task descriptions ([17, 20]). Although our robot is not exactly a 6 DOF manipulator, we can easily view it so if disregarding the slider, part that has its trajectory and positions determined independently from the rest of the arm (see 2.5. *Direct and Inverse Kinematics Approach*).

3.1 ROBOT SOLVING

The first part of the software package solves most of the robot modules of equations (Figure 12).

For the direct kinematics, given the D-H parameter table, the package creates the $A_0^1..A_5^6$ matrices and obtain the $T_0^1..T_0^6$ transformation matrices in their symbolic form and base coordinates. For velocity kinematics, the software derives the Jacobian matrix and output equations in the form:

$$\dot{X} = J * \dot{Q}$$

Where:

\dot{X} = The Cartesian velocity vector,

J = The Jacobian matrix,

\dot{Q} = The joint velocity vector

The package also implements a symbolic matrix inversion routine that solves the inverse velocity equations. Through its symbolic differentiation routines, the package also provides the acceleration and inverse acceleration kinematics equations.



Figure 12. Input parameter window of the simulation- package

Trajectory plotting equations are also implemented in this software package, allowing for either cubic polynomial form or constant velocity with cubic polynomial blends. We have opted for the second choice. Three time intervals can be chosen, a t_0-t_1 interval for an accelerating motion, a t_2-t_3 interval for decelerating motion and a t_1-t_2 for a linear/constant velocity period.

Given the M, G and V matrices (see 2.4. *Direct and Inverse Dynamics Approach*), the package outputs the dynamics and inverse dynamics equations too. All of the equations are generated in text, C/C++ source code, or Mathematica format, thus they are easy to be used for further scaling. Once we have obtained the necessary output from the package, testing in parallel with our software revealed a clear consistency.

3.2 SIMULATION AND OPTIMIZATION

The second part of the software package, also called the execution module, was of particular use for our tire changer. The package implements a local PD controller on the control function:

$$\tau = f(Q, \dot{Q}, \ddot{Q}) * \ddot{Q} + f(Q, \dot{Q}, \ddot{Q}) * K_p * e_p + f(Q, \dot{Q}, \ddot{Q}) * K_v * e_v$$

Where:

f = A function of robot joint position, velocity and acceleration vectors (note that the package allows for any mathematical function),

\ddot{Q} = The desired link acceleration vector,

K_p = The proportional gain,

K_v = The derivative gain,

e_p = The error in joint variables vector,

e_v = The error in joint velocities vector.

The package also allows to add a PID or other feedback control functions ([17,19]). After providing K_p , K_v , the initial and final positions, the time interval in which the movement should be committed, the number of iterations in the PD loop and the trajectory generator to be used, the package will run the control loop on points specified by the user and output graphs showing the ideal and the real trajectories of the manipulator, plus searching and optimizing the K_p , K_v and update values.

As an example, we run the control loop on a move similar to the fully compressed - fully extended one for which we have

included the torque graphs (Figures 13, 14), one second time limit.



Figure 13. Setting up the thetas for the PD loop



Figure 14. Additional PD data for the loop control

As output, we can see the desired thetas versus the actual ones. The package also provided the optimal K_p and K_v values, after looping on the intervals and parameters ranges (Figures 15, 16 and 17).



Figure 15. Theta1 (desired versus actual)

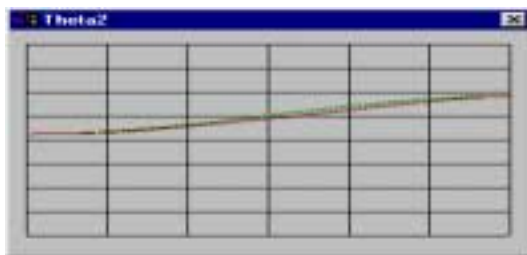


Figure 16. Theta2 (desired versus actual)

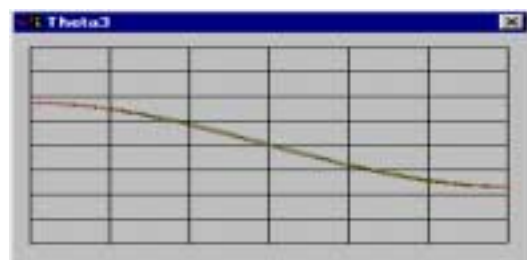


Figure 17. Theta3 (desired versus actual)

With the help of the trajectory plotting part of the program, we were able to script the output into our model and analyze the move. It is of major importance to have all the trajectories well defined. Figure 18 shows the trajectories described above: the trajectory of the upper arm, elbow and wrist:



Figure 18. Plotted trajectories

The following are examples of other trajectories that have been successfully improved with the help of the package (Figures 19,20,21).



Figure 19. End Effector trajectory analysis 1

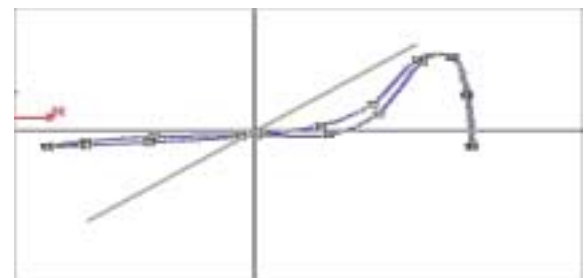


Figure 20. End Effector trajectory analysis 2

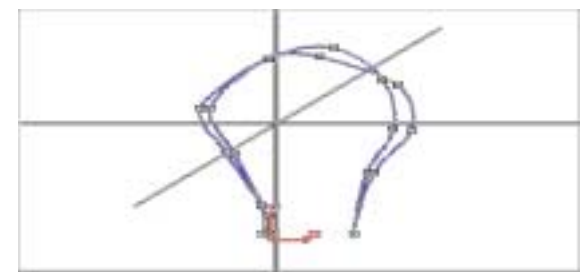


Figure 21. End Effector trajectory analysis 3

Note that most of the simulation sequences discussed here are available as AVI movies on the website <http://www.bridgeport.edu/~risc>. With the help of the software package, we were able to analyze and improve on all of the movement that the tire changer performs while changing a tire.

4. Conclusions

The design and symbolic construction of the robot can promote critical problems if it is not analyzed thoroughly before being brought to reality. It is much more expensive to discover an inconsistent design after it was actually built, just as it is important to make as many optimizations and improvements possible at the design and simulation phase. Having applied the simulation and optimization software above, we were able not only to correct, but improve consistently the performance. The trajectories have been adjusted, allowing for a lower inertial load on the arm and for faster reach to its destinations. We were also able to reduce the amount of torque necessary.

It is difficult to build extensive simulation and control software for any particular or generalized type of robot, however, the more the testing is done at the design time, the more feasible and practical the robot will be.

References

- [1] Spong, W. Mark, "Robot Dynamics and Control", John Wiley, 1989
- [2] McKerrow, Phillip John, "Introduction to Robotics", Addison Wesley, 1991
- [3] Dekhil, Mohamed, and Sobh, Tarek M., and Henderson, Thomas C., and Sabbavarapu, Anil, and Mecklenburg, Robert, "Robot Manipulator Prototyping (Complete Design Review)", University of Utah, 1994
- [4] Nakamura, Yoshihiko, "Advanced Robotics - Redundancy and Optimization", Addison-Wesley, 1991
- [5] Marris, Andrew W., and Stoneking, Charles E., "Advanced Dynamics", McGraw-Hill, 1967
- [6] Christie, Dan Edwin, "Intermediate College Mechanics", McGraw-Hill, 1952
- [7] De Wit, Charlos Canudas, and Siciliano, Bruno, and Bastin, Georges, "Theory of Robot Control", Springer-Verlag London, 1996
- [8] Sobh, Tarek M., Dekhil, Henderson, Thomas, C. and Sabbavarapu, A. "Prototyping a Three-link Robot Manipulator," Presented in the Second World Automation Congress, Sixth International Symposium on Robotics and Manufacturing (ISRAM 96), Montpellier, France, May 1996
- [9] Herrea-Beneru, L. ,Mu, E. ,Cain, J. T., "Symbolic Computation of Robot Manipulator Kinematics", Department of Electrical Engineering, University of Pittsburgh
- [10] Rieseler, H., Wahl, F. M., "Fast symbolic computation of the inverse kinematics of robots", Institute for Robotics and computer control, Technical University of Braunschweig
- [11] Dekhil, M., Sobh, T. M., Henderson, T. C. and Mecklenburg, R "UPE: Utah Prototyping Environment for Robot Manipulators". In proceedings of the IEEE International Conference on Robotics and Automation, Nagoya, Japan, May 1995
- [12] Schalkoff, R. J. "Digital Image Processing and Computer Vision", John Wiley and Sons, Inc., 1989
- [13] J. Hervé, P. Cucka and R. Sharma, "Qualitative Visual Control of a Robot Manipulator". In Proceedings of the DARPA Image Understanding Workshop, September 1990
- [14] M. J. Banks and E. Cohen, "Realtime B-Spline Curves from Interactively Sketched Data," Proceedings of the 1990 Symposium on Interactive 3-D graphics, ACM, March 1990
- [15] J. A. Thingvold and E. Cohen, "Physical Modeling with B-Spline Surfaces for Interactive Design and Animation," Proceedings of the 1990 Symposium on Interactive 3-D graphics, ACM, March 1990
- [16] Y. Li and W. M. Wonham, "Controllability and Observability in the State-Feedback Control of Discrete-Event Systems", Proc. Conf. on Decision and Control, 1988
- [17] Benedetti, R., and Risler, J. J. "In Real Algebraic and Semi-algebraic Sets" (1990), Hermann, pp. 8-19
- [18] P. K. Allen, "Robotic Object Recognition Using Vision and Touch", 1987. Kluwer Academic Publishers, Norwell, MA
- [19] Mihali, Raul C., Sobh, Tarek M., "The Formula One Tire Changing Robot (F1-T.C.R.)", accepted for publication in the Journal of Intelligent and Robotic Systems, April 1999
- [20] Craig, John J., "Introduction to Robotics. Mechanics and Control. Second Edition", 1989, Addison Wesley