Controlling Multi Degree of Freedom Robot Joints Torques

Using Genetic Algorithm

O. Shoubaky¹ and T.M. Sharari²

¹Computer and Intelligent Systems Center, P.O.Box. 215, Jo
²Institute of Eng. and Tech., Cont. Lab& Aut., Po.Box.81. Jo

*Corresponding Author’s E-mail: o.shoubaky@yahoo.com

Abstract

Controlling the motions of humanoid robot when interacting in external environment, requires an exact estimation of joint torques. Therefore, this paper suggests a technique to estimate the maximum surplus torque for each joint using a genetic algorithm (GA) when the robot is pushing an object. To demonstrate the effectiveness of the suggested method, a multi degree of freedom humanoid robot carrying a pushing motion is considered, and a number of simulations are conducted. The results of the simulations show that the suggested technique can be applied to the control of joints’ torques for the humanoid robot interacting with external environment.

Keywords: Humanoid robot, Genetic Algorithm

1. Introduction

Many people expect that humanoid robots can perform various works in addition to walking motion. In this case, humanoid-robot motion needs to be more complex and sophisticated, and the joints of the robots need to be more precisely controlled for achieving a desirable motion or trajectory. Generally, a humanoid robot consists of many joints, each of which is a combination of an actuator and reducer. So, when a robot performs a cooperative motion with external environment, it is essential to adequately control torques of the joint actuators of the robot. As an example of a cooperative motion, this study considers a robot’s action of pushing an object. If a torque required at any joint of a humanoid robot exceeds its maximum capacity for achieving a desirable task, the robot cannot perform a desirable motion or some joints might be damaged. From this point of view, our approach involves finding optimal postures that maximise the working capability of humanoid robots by controlling the joint torques in such a way that the surplus torque for all joints is maximised. Until now, a number of research works showing a growing interest in humanoid-robot motion have been reported. The works [1-2] find optimal configurations in terms of the Cartesian force generation and energy consumption using nonlinear programming technique. However, the works are not directly related to the humanoid-robot research. The work [3] optimises configuration of the posture of humanoid robot by Simulated Annealing Algorithm. However, it is only applicable to the arm part of a humanoid robot. The study for developing a whole body cooperative dynamic walking control method [4] and effective balancing method [5] also reported, but they are restricted to stabilisation of walking mode. Recent successful development can be seen in mobile manipulation of humanoid robots and optimisation of its posture using genetic algorithms[6-9]. However, most of the methods rely on approximate or restricted force information on a humanoid robot instead of precise information. To provide realistic force information with humanoid-robot cooperative motion, this study proposes a
method to find an optimal posture of a humanoid robot using a genetic algorithm (GA) in such a way that the surplus torque ratio for all joints is maximised when the robot is pushing an object by hands. Also, to assist structure, ANSYS programme is used. The desired joint angle and the generated torque at each arm and leg joint are calculated using forward and inverse kinematics and Jacobian, and the force and moment exerted on the leg part is calculated using ANSYS. The torques thus found are reflected on an objective function which is to be maximised by a GA. As a result of successive maximisation process, an optimised posture having a large surplus joint torque is finally found.

To show the effectiveness of the proposed method, a 24-DOF humanoid-robot’s pushing motion is considered, and a number of simulations are carried out. The simulation result shows that the proposed method can be used to the control of torques for humanoid-robot cooperative motion with external environment.

2 Modelling and Analysis of Humanoid Robots

2.1 Degree of Freedom (DOF) and Coordinate System

This study considers a humanoid robot consisting of two arms and two legs. Each arm and leg has 6 DOF of motion, thus the robot has total 24 DOF of motion. Figure 1 shows the kinematic diagram of the robot assumed to be 1.2m in height and 36kg in weight.

![Kinematic Diagram](image)

Figure 1: Degree of freedom and the coordinate

As shown in the above figure, the coordinate system is divided into four frames: {Lh} for the left hand, {Rh} for the right hand, {Lf} for the left foot and {Rf} for the right foot.

2.2 Forward and Inverse Kinematics

Forward and inverse kinematics is to be derived with respect to four parts: two arm parts and two leg parts. This study takes the homogeneous transformation consisting of the position vector about
the origin of the reference frame, and the rotation matrix described by X-Y-Z Euler Angle. Each joint angle range is assumed to be given as

\[
\begin{align*}
-180 < & (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) < 180, \\
0 < & \theta_4 < 180 \quad \text{or} \quad -180 < \theta_4 < 0.
\end{align*}
\]

Here, the index \(i\) of each joint angle \(\theta_i\) follows the same number of each arm and leg part shown in figure 1. \(\theta_4\) corresponds to each knee and elbow of the robot. Frame \(\{B\}\) is attached on the centre of gravity of the robot, so the homogenous transformation \(\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}\) with respect to \(\{I\}\), can be defined. Also, each link transformations, \(\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}\) is defined as the coordinate system. Here, the position and rotation of frame \(\{0\}\) is coincident with that of frame \(\{1\}\). Thus, \(\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}\) is given by

\[
\begin{align*}
\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} = & \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}^{-1} \times \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}^{-1} \times \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}^{-1} \times \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}^{-1}.
\end{align*}
\]

For obtaining the close-form solutions to the inverse kinematics, two methods are adopted: algebraic approach and geometric approach. In figure 1, each joint angle with respect to frame \(\{Lh1\}\), \(\{Lh2\}\) and \(\{Lh3\}\) is obtained by algebraic approach, while each joint angle with respect to frame \(\{Lh4\}\), \(\{Lh5\}\) and \(\{Lh6\}\) is easily found by geometric approach. The obtained joint angles are verified by the MSC.visualNastran4D, and figure 2 shows the 3D modeling of an initial posture by the software.

2.3 Force and Torque Analysis for the Arm Part

To find the joint torques, Jacobian \(J\) relating joint velocity \(\dot{\theta}\) to Cartesian velocity \(\dot{X}\) is used on the basis of calculated angles at the static situation. The Jacobian is defined by

\[
\dot{X} = J \times \dot{\theta}.
\]

The Jacobian matrix is expressed by a \(6 \times n\) (\(n\): the number of joints) matrix: the upper three rows for linear velocity and the lower three ones for angular velocity. The Jacobian matrix deriving from the forward kinematics is expressed by the following equation:
Using the principle of virtual work $W$, each joint torque is derived by the following equations:

$$W = \tau^T \times \dot{\theta} , \quad F^T \times \dot{X} = \tau^T \times \dot{\theta} , \quad \tau = J^T \times F.$$  

Here, $F$ is the $6\times 1$ force and moment vector applied at the robot with respect to frame $\{I\}$.

### 2.4 Force and Torque Analysis for the Leg Part

In general, the reaction force and moment applied at the leg part of a humanoid robot cannot be simply calculated when frame $\{B\}$ is varied. So this study uses a structure analysis programme ANSYS. As the first step, the position of each joint angle with respect to frame $\{I\}$ is determined from forward and inverse kinematics and is to be used for modelling by ANSYS. In the process of modelling, the acting force $F=\begin{bmatrix} -80 & 0 & 0 \end{bmatrix}$ is applied on each palm and $F=\begin{bmatrix} 0 & 0 & -360 \end{bmatrix}$ is applied on frame $\{B\}$, where two kinds of the force are presented with respect to frame $\{I\}$. As a result of structure analysis, the reaction force and moment at frame $\{0\}$ located on the robot’s waist of leg-part is found. In the same way as the arm part, joint torques at the leg part are computed by such obtained force and moment. Figure 3 shows the modelling of ANSYS for the initial posture, while table 1 shows joint angles, joint torques, input force for each part and the position and rotation of frame $\{B\}$ at the initial posture.
3. Optimisation using a Genetic Algorithm (GA)

3.1 Parameters and Objective Functions

GA has been known to perform generally well when searching large spaces with many local optima, and formulates evolution process of survival by the fitness. This paper adopts a simple genetic algorithm (SGA) with roulette wheel selection, simple crossover and mutation. In addition, elitist strategy in which individual with the strongest fitness would be survived in the next generation is used for enhancing performance of the algorithm. Table 2 shows parameters used in this algorithm.

![Figure 3: Modelling of the humanoid robot’s initial posture using ANSYS.](image)

**Table 2: Parameters of a genetic algorithm.**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum generation</td>
<td>300</td>
</tr>
<tr>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>Number of variables</td>
<td>6</td>
</tr>
<tr>
<td>Probability of crossover, Pc</td>
<td>0.8</td>
</tr>
<tr>
<td>Probability of mutation, Pm</td>
<td>0.01</td>
</tr>
</tbody>
</table>

To increase overall acting force through maximising the overall surplus joint torque ratio, the objective function $F(x)$ is defined by

$$
F(x) = \sum_{i=1}^{6} \left( 1 - \frac{\tau_i}{\tau_{\text{max},i}} \right)^2. \quad (i=1,\ldots,6)
$$

Here, $\tau_i$ is each joint torque and $\tau_{\text{max}}$ is maximum allowable torque of each joint.

3.2 Constraints in a GA

The following shows the constraints for achieving optimisation by the GA.

1. Each position and rotation of the palm and the sole is fixed at the initial posture.
2. The force acting on the palm and on the centre of gravity of the robot keeps constant.
3. The torque at each joint should be lower than its allowable maximum torque shown in table 3.
4. The rotation and position of frame $\{B\}$ should follow the allowable range shown in table 4.
### Table 3: Maximum allowable torques at joints.

<table>
<thead>
<tr>
<th>Joint No.</th>
<th>$T_{\text{max}}$ [Nm]</th>
<th>Joint No.</th>
<th>$T_{\text{max}}$ [Nm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint11(0_{11}), Joint21(0_{21})</td>
<td>57.8</td>
<td>Joint31(0_{31}), Joint41(0_{41})</td>
<td>57.8</td>
</tr>
<tr>
<td>Joint12(0_{12}), Joint22(0_{22})</td>
<td>189.8</td>
<td>Joint32(0_{32}), Joint42(0_{42})</td>
<td>57.8</td>
</tr>
<tr>
<td>Joint13(0_{13}), Joint23(0_{23})</td>
<td>189.8</td>
<td>Joint33(0_{33}), Joint43(0_{43})</td>
<td>57.8</td>
</tr>
<tr>
<td>Joint14(0_{14}), Joint24(0_{24})</td>
<td>227.8</td>
<td>Joint34(0_{34}), Joint44(0_{44})</td>
<td>57.8</td>
</tr>
<tr>
<td>Joint15(0_{15}), Joint25(0_{25})</td>
<td>227.8</td>
<td>Joint35(0_{35}), Joint45(0_{45})</td>
<td>57.8</td>
</tr>
<tr>
<td>Joint16(0_{16}), Joint26(0_{26})</td>
<td>189.8</td>
<td>Joint36(0_{36}), Joint46(0_{46})</td>
<td>57.8</td>
</tr>
</tbody>
</table>

### Table 4: Allowable range of \{B\}.

<table>
<thead>
<tr>
<th>Allowable Range</th>
<th>{B} of robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of $P_x$ [m]</td>
<td>$0 \leq P_x \leq 0.15$</td>
</tr>
<tr>
<td>Range of $P_y$ [m]</td>
<td>$-0.1 \leq P_y \leq 0.1$</td>
</tr>
<tr>
<td>Range of $P_z$ [m]</td>
<td>$0.5 \leq P_z \leq 0.8$</td>
</tr>
<tr>
<td>Range of $R_x$ [°]</td>
<td>$-10 \leq R_x \leq 10$</td>
</tr>
<tr>
<td>Range of $R_y$ [°]</td>
<td>$-10 \leq R_y \leq 10$</td>
</tr>
<tr>
<td>Range of $R_z$ [°]</td>
<td>$-10 \leq R_z \leq 10$</td>
</tr>
</tbody>
</table>
3.3. Optimisation Algorithm

Figure 4 shows the flow chart of the proposed optimisation algorithm.

![Flowchart of the proposed optimisation algorithm.](image)

**Figure 4:** Flowchart of the proposed optimisation algorithm.

4 Simulation Results

Using the proposed approach, a number of simulations are carried out to show the effectiveness of the algorithm. Figure 5 shows the change of the average fitness function and the maximum value of the objective function. The figure show that values tend to increase through the simulation.
**Figure 5:** Change of the fitness function and the objective function.

**Figure 6:** Change of position with reference to \{I\}.

**Figure 7:** Change of rotation with reference to \{I\}.
Figures 6 and 7 show the position (Px, Py, Pz) and rotation (Rx, Ry, Rz) of frame {B} is varying in progress of generation with respect to frame {I}. The figure shows that values (Px, Py, Pz, Rx, Ry, Rz) tend to converge into a single value through the simulation. Figures 8 and 9 show the comparisons between the initial joint torque ratio and the final joint torque ratio. Each joint torque ratio at the final posture is less than that at the initial posture, as shown in figure 8 and figure 9. This means that surplus joint torque at final posture is larger than that at the initial posture.

**Figure 8:** Comparison of the joint torque ratio between the initial and the final posture at the leg part.

**Figure 9:** Comparison of the joint torque ratio between the initial and final posture at the arm part.

It can be seen that sum of the each joint torque is 300.9Nm at the initial posture while 207.5Nm at the final posture. This means the humanoid robot has more surplus torque by 93.3Nm than it of initial posture. Table 5 shows the simulation results at the
optimised posture. Figure 10 shows the humanoid robot’s final posture from the simulation results in table 5. This result means that surplus torque can be enlarged as much as reduced. Thus, it is concluded that proposed method can be efficiently used to the control of torques for humanoid robots cooperative with external environment.

**Table 5: Simulation results of optimal posture.**

<table>
<thead>
<tr>
<th>(\theta_{L,q}^{[\circ]})</th>
<th>(\theta_{R,q}^{[\circ]})</th>
<th>(\theta_{L,s}^{[\circ]})</th>
<th>(\theta_{R,s}^{[\circ]})</th>
<th>(\theta_{L,s}^{[\circ]})</th>
<th>(\theta_{R,s}^{[\circ]})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\theta_{1}, \theta_{2})</td>
<td>(\theta_{3}, \theta_{4})</td>
<td>(\theta_{5}, \theta_{6})</td>
<td>(\theta_{7}, \theta_{8})</td>
<td>(\theta_{9}, \theta_{10})</td>
<td>(\theta_{11}, \theta_{12})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(P_{e} [m])</th>
<th>(P_{a} [m])</th>
<th>(P_{b} [m])</th>
<th>(P_{c} [m])</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.150</td>
<td>1.716</td>
<td>0.161</td>
<td>0.781</td>
</tr>
</tbody>
</table>

\(\tau_{i,j} [Nm]\) and \(j\) imply the same at \(i\). \(\tau_{i,j} [Nm]\) and \(j\) imply the same at \(i\).

**Conclusion**

Controlling the motions of humanoid robot when interacting in external environment requires an exact estimation of joint torques. Therefore, this paper suggests a technique to estimate the maximum surplus torque for each joint using a genetic algorithm (GA) when the robot is pushing an object. The robot with 24 degree of freedom is considered, and forward and inverse kinematics is derived on the basis of modelling. Then, the torque of all joints is calculated from the relation of forward and inverse kinematics and Jacobian. Then ANSYS programme is used to find the acting force and moment at the leg part. Such obtained torques are reflected to an objective function which is to be minimised by a genetic algorithm (GA). Here, the position and the rotation angle of the centre of gravity of the robot are considered as the design variables. As a result of successive maximisation of the objective function by the GA, an optimised posture having a maximum surplus torque for the cooperative motion is finally found. And to show the effectiveness of the proposed method, a number of simulations are carried out. The simulation result shows that the proposed method can be adopted to the control of torques for humanoid robot cooperative motion with external environment.
References


