SHOE GRINDING CELL USING VIRTUAL MECHANISM APPROACH

Bojan Nemec and Leon Zlajpah
Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

Keywords: Robotics, Redundant robot control, Force control, Shoe grinding.

Abstract: The paper describes the automation of the shoe grinding process using an industrial robot. One of the major problems of flexible automation using industrial robots is how to avoid joint limitations, singular configuration and obstacles. This problem can be solved using kinematically redundant robots. Due to the circular shape of the grinding disc, the robot becomes kinematically redundant. This task redundancy was efficiently handled using virtual mechanism approach, where the tool is described as a serial mechanism.

1 INTRODUCTION

Shoe production is most likely labor intensive; the rate of automation is usually low. Therefore it is considered as the industry suitable for the counties with low labor cost (Taylor and Taylor, 1988; Nemec and et all, 2003). In last years new aspect in shoe production is arising - custom made shoes (Dulio and Boer, 2004). The customization in mass shoe production requires complex information system and full automation of planning, production and distribution processes. One of the major challenges is how to generate robot trajectories base solely on the CAD model of the shoe. Manual teaching and trajectory testing phases are not acceptable for customized shoes, where virtually each work-piece on the production line can differ from the previous one. Therefore, we have to design tools which enable to generate fault tolerant robot trajectories. This is usually accomplished using complex task planning algorithms, which are in most cases off-line batch procedures. In (Nemec and Zlajpah, 2008) we proposed a solution based on control algorithms for the kinematicaly redundant robot, where we sacrificed exact orientation of the tool in order to achieve additional degrees of redundancy. In this paper, we propose a new approach, called virtual mechanism approach. The proposed algorithm was applied to the shoe grinding cell, which uses an industrial robot with 7 D.O.F.

2 SOLVING TASK KINEMATIC REDUNDANCY USING VIRTUAL MECHANISM APPROACH

Robotic systems under study are n degrees of freedom (DOF) serial manipulators. We consider redundant systems, which have more DOF than needed to accomplish the task, i.e. the dimension of the joint space n exceeds the dimension of the task space m, $n > m$ and $r = n - m$ denote the degree of the redundancy. Let the configuration of the manipulator be represented by the a vector $\boldsymbol{q}_r$ of n joint positions, and the end-effector position (and orientation) by m-dimensional vector $\boldsymbol{x}_r$ of the robot tool center point positions (and orientations). The relation between the joints and the task velocities is given by the following well known expression

$$\dot{\boldsymbol{x}}_r = \mathbf{J}_r \dot{\boldsymbol{q}}_r \tag{1}$$

where $\mathbf{J}_r$ is the $m \times n$ manipulator Jacobian matrix. The solution of the above equation for $\dot{\boldsymbol{q}}_r$ can be given as a sum of the particular and the homogeneous solution

$$\dot{\boldsymbol{q}}_r = \bar{\mathbf{J}}_r \dot{\boldsymbol{x}}_r + \mathbf{N}_r \xi \tag{2}$$

where

$$\bar{\mathbf{J}}_r = \mathbf{W}^{-1} \mathbf{J}_r^T (\mathbf{J}_r \mathbf{W}^{-1} \mathbf{J}_r^T)^{-1}. \tag{3}$$

Here, $\bar{\mathbf{J}}_r$ is the weighted generalized-inverse of $\mathbf{J}_r$, $\mathbf{W}$ is the weighting matrix, $\mathbf{N}_r = (\mathbf{I} - \bar{\mathbf{J}}_r \mathbf{J}_r)$ is a $n \times n$ matrix representing the projection into the null space of $\mathbf{J}_r$, and ξ is an arbitrary n dimensional vector. We will denote this solution as the generalized inverse based

Nemec B. and Zlajpah L. (2008).
SHOE GRINDING CELL USING VIRTUAL MECHANISM APPROACH.
In *Proceedings of the Fifth International Conference on Informatics in Control, Automation and Robotics - RA*, pages 159-164
DOI: 10.5220/0001480901590164
Copyright © SciTePress

redundancy resolution at the velocity level (Nenchev, 1989). The homogenous part of the solution belongs to the Jacobian null-space. Therefore, we will denote it as $\dot{\boldsymbol{q}}_n$, $\dot{\boldsymbol{q}}_n = \mathbf{N}_r\xi$. Now consider the case where the robot Jacobian matrix $\mathbf{J}_r$ is defined in Cartesian (world) coordinate system and the dimension of the Jacobian is $6\times n$, but the task is described in another coordinate system. It can be shown that the transformation from the Cartesian to the task space can be very complex. As an alternative approach we propose to model the tool as a serial kinematic link. Let consider the general case where the robot holds the object to be machined and the work tool is fixed, as illustrated in Fig. 1. In such a case, we can define the direct kinematic transformation as

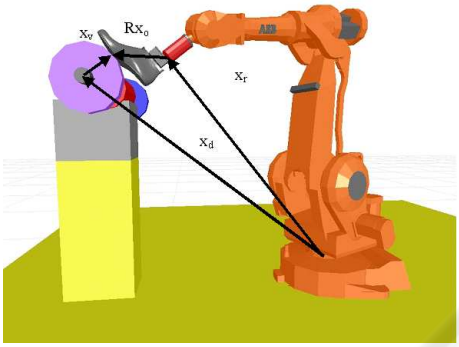

Figure 1: The robot holds the object and the work toll is fixed.

$$\boldsymbol{x}_r + \left[\mathbf{R} \,\vdots\, \mathbf{I}\right]\boldsymbol{x}_o = \boldsymbol{x}_d + \boldsymbol{x}_v \tag{4}$$

where $\boldsymbol{x}_r$ is the robot Cartesian position and orientations, $\mathbf{R}$ is the robot tool rotation 3×3 dimensional matrix, $\boldsymbol{x}_o$ is the 6×1 vector of the object position and orientation, $\boldsymbol{x}_v$ is the 6×1 vector of position and orientation of the top of the virtual mechanism and 6×1 vector $\boldsymbol{x}_d$ describes the distance and orientation between the base coordinates system and the work tool coordinate system. Let consider robot and virtual mechanism as one mechanism with $n+n_v$ degrees of freedom. The configuration of the virtual mechanism can be described with the $n_v\times 1$ dimensional vector $\boldsymbol{q}_v$. The new task position is

$$\boldsymbol{x} = \boldsymbol{x}_r - \boldsymbol{x}_v \tag{5}$$

and the Jacobian of this new mechanism can be expressed as

$$\mathbf{J} = [\mathbf{J}_r| - \mathbf{J}_v] \tag{6}$$

where the Jacobian of the virtual mechanism $\mathbf{J}_v$ is defined as

$$\mathbf{J}_v = \frac{\partial \boldsymbol{x}_v}{\partial \boldsymbol{q}_v}. \tag{7}$$

As we can see, the task space preserves it's dimension, while the joint space is increased with the dimension of the virtual mechanism. This approach has two major advantages. First, we can use the existing robot Jacobian, which is assumed to be known. Second, the augmented part of the Jacobian has very simple structure in most cases. Note that Eq. 4 does not handle orientations correctly, since orientation vectors can not be simply added in general case. Therefore, using Eq. 4 and 5 we obtain an approximate solution regarding the orientation vector. In most cases, the difference between the desired and the real orientation vector is small and is acceptable for operations like brushing and polishing. If orientations are important, we can use equation 4 and 5 for the calculation of positions only, while the orientations have to be calculated using rotation matrices as follows.

$$\mathbf{R}_o = {\mathbf{R}_v}^T\mathbf{R} \tag{8}$$

Here, $\mathbf{R}_o$ and $\mathbf{R}_v$ are 3×3 rotation matrices describing object rotation against virtual mechanism and virtual mechanism rotation expressed in the robot base coordinate system. The corresponding orientation vector can be than obtained using the transformation of the rotation matrix to the orientation vector described with euler or roll pitch yaw notation. Note that using this 'correct' transformation also rotation part of the Jacobian described by Eq. 6 becomes more complex. On the other hand, when the Jacobian is used in the control loop, only approximate values of Jacobian are needed. Therefore, control algorithm based on Jacobian defined by Eq. 6 gives equal results as control algorithm using the Jacobian calculated from rotation matrices, as described with Eq. 8.

3 CONTROL

Using the virtual mechanism approach, we can directly apply any control algorithm for the kinematically redundant robot. A suitable choice is the control law using generalized inverse-based redundancy resolution at velocity level in the extended operational space. Redundancy resolution at the velocity level is favorable because it enables direct implementation of the gradient optimization scheme for the secondary task. The gradient projection technique has been widely used for the calculation of the null space velocity that optimizes the given criteria. The reason for this is that a variety of performance criteria can be easily expressed as gradient function of joint coordinates. Although the control law using generalized inverse-based redundancy resolution at velocity level can not completely decouple the task and the

null space (Oh et al., 1998; Park et al., 2002; Nemec and Zlajpah, 2000; Nemec and Zlajpah, 2001), it enables good performance in real implementation (Nemec et al., 2007). The joint space control law is

$$\tau_c = \mathbf{H}\mathbf{J}^T(\ddot{\boldsymbol{x}}_d + \mathbf{K}_v\dot{\boldsymbol{e}}_x + \mathbf{K}_p\boldsymbol{e}_x + \mathbf{K}_f\boldsymbol{e}_f - \dot{\mathbf{J}}\dot{\boldsymbol{q}}) + \mathbf{H}\mathbf{N}(\ddot{\boldsymbol{q}}_{nd} + \mathbf{K}_n\dot{\boldsymbol{e}}_n - \dot{\mathbf{N}}\dot{\boldsymbol{q}}) + \boldsymbol{h} + \mathbf{J}^T\boldsymbol{f} \quad (9)$$

where $\mathbf{J}$ is the Jacobian matrix, $\mathbf{H}$ is $n \times n$ inertia matrix, $\boldsymbol{h}$ is n-dimensional vector of the centrifugal, coriolis and gravity forces, $\boldsymbol{f}$ is n-dimensional vector of the external forces acting on the manipulator's end effector and $\mathbf{K}_p$, $\mathbf{K}_v$, $\mathbf{K}_f$ and $\mathbf{K}_n$ are diagonal matrices representing positional, velocity, force and null-space feedback gains. The first term of the control law corresponds to the task-space control torque τ_x, the second to the null-space control torque τ_n and the third and the fourth is used to compensate the non-linear system dynamics and the external force, respectively. Here, $\boldsymbol{e}_x = \boldsymbol{x}_d - \boldsymbol{x}$ are the task-space tracking error, $\boldsymbol{e}_f = \boldsymbol{f}_d - \boldsymbol{f}$ and $\dot{\boldsymbol{e}}_n = \dot{\boldsymbol{q}}_{nd} - \dot{\boldsymbol{q}}_n$ is the null-space tracking error. $\boldsymbol{x}_d$, $\boldsymbol{f}_d$ and $\dot{\boldsymbol{q}}_{nd}$ are the desired task coordinates, desired force and the desired null space velocity, respectively. The details of the control law derivation can be found in (Nemec and Zlajpah, 2000; Nemec et al., 2007).

An attention should be paid on the selection of the inertia of the virtual link. The inertia matrix $\mathbf{H}$ has the form

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_r & 0 \\ 0 & \mathbf{H}_v \end{bmatrix} \quad (10)$$

where $\mathbf{H}_r$ is the robot inertia matrix and $\mathbf{H}_v$ is the diagonal matrix describing the virtual mechanism inertia. Clearly, $\mathbf{H}_v$ can not be zero, but arbitrary small values can be chosen describing the lightweight virtual mechanism. Selection of the inertia matrix of the virtual mechanism affects only the null space behavior of the whole system. Heavy virtual links with high inertia would slow down the movements of the virtual links. Therefore, low inertia of virtual links is an appropriate choice. On the contrary, we can assume that no gravity, coriolis and centrifugal forces act on the virtual links and the corresponding terms in the vector $\boldsymbol{h}$ can be set to zero. Control law 9 assumes feedback from all joints, including non-existing virtual joints. There are multiple choices how to provide the joint coordinates and the joint velocities of the virtual link. A suitable method is to build a simple model composed of a double integrator

$$\dot{\boldsymbol{q}}_v = \int \mathbf{H}_v^{-1}\tau_{cv} \quad (11)$$

$$\boldsymbol{q}_v = \int \dot{\boldsymbol{q}}_v$$

where τ_{cv} is the part of the control signals corresponding to the virtual link.

4 MOTION OPTIMIZATION

As we mentioned previously, one of the main problems in automatic trajectory generation is the inability to assure that the generated trajectory is feasible using a particular robot, either because of possible collisions with the environment or because of the limited workspace of the particular robot. Limitations in the workspace are usually not subjected to the tool position, but rather to the tool orientation. Another sever problem are wrist singularities, which can not be predicted in the trajectory design phase on a CAD system. A widely used solution in such cases is off-line programming with graphical simulation, where such situation can be detected in the design phase of the trajectory. Unfortunately this is a tedious and time consuming process and therefore not applicable in customized production, where almost each work piece can vary from the previous one (Dulio and Boer, 2004; Nemec and Zlajpah, 2008). The problem can be efficiently solved using the null space motion, which changes the robot configuration, but does not affect the task space motion. The force and the position tracking are of the highest priority for a force controlled robot and are therefore considered as the primary task. The secondary task is defined by the optimization of a given cost function.

Let p be the desired cost function, which has to be maximized or minimized. Then the velocities

$$\dot{\boldsymbol{q}}_n = k\mathbf{N}\mathbf{H}^{-1}\left(\frac{\delta p}{\delta q_1}, \frac{\delta p}{\delta q_2}, \ldots . \frac{\delta p}{\delta q_n},\right) \quad (12)$$

in Eq 12 maximize cost function for any $k > 0$ and minimize cost function for any $k < 0$ (Asada and Slotine, 1986), where k is an arbitrary scalar which defines the optimization step. In our case we have selected a compound p which maximizes the distances between obstacle and the robot links or robot work object, maximizes the distance to the singular configuration of the robot and maximizes the distance in joint coordinates between current joint angle and joint angle limit. We define the cost function as a sum of three cost functions $p = p_a + p_l + p_s$, where p_a denotes cost function for obstacle avoidance, p_l cost function for avoiding joint limits and p_s cost function for singularity avoidance. We select the cost function for obstacle avoidance as (Khatib, 1986; Khatib, 1987) $p_a = \frac{1}{2}\mathbf{E}d_0^2$, where $\mathbf{E}$ is an $l \times l$ rotation matrix describing the direction of an artificial potential field pointing from the obstacle, l is the dimension of the position sub-space and d_0 is the shortest distance between the obstacle and the robot body. In our case the desired objective is fulfilled if the imaginary force is applied only on the robot joints. The cost function for the joint limits avoidance is defined as (Nemec and

Zlajpah, 2008; Chaumette and Marchand, 2001)

$$p_l = \begin{bmatrix} (q_{max}-q)^2, |q_{max}-q)^2| < \varepsilon \\ 0 \\ (q_{min}-q)^2, |q_{min}-q)^2| < \varepsilon \end{bmatrix} \quad (13)$$

where ε is a positive constant defining the neighborhood of joint limits. For the singularity avoidance we use the manipulability index defined as(Asada and Slotine, 1986)

$$p_s = \sqrt{|\mathbf{J}\mathbf{J}^T|} \quad (14)$$

Then, the desired null space velocity for our task are calculated using

$$\dot{\boldsymbol{q}}_n = \mathbf{N}\mathbf{H}^{-1}(k_a \mathbf{J}^{03} V d - 2k_l(\boldsymbol{q}_l - \boldsymbol{q}) - 2k_s \frac{\delta \mathbf{J}}{\delta \boldsymbol{q}} \mathbf{J}^T \quad (15)$$

Matrix $\mathbf{J}^{03}$ is the Jacobian matrix calculated from the robot base to the robot wrist. Scalars k_a, k_l and k_s are appropriate positive constants defining the optimization step. In real implementation, k_a and k_l are set to zero if the observed point is away enough from the possible collision points and joints are far away from their limits, respectively. Similar, the last term of $\dot{\boldsymbol{q}}_n$ is not computed if the manipulability index is large enough. Unfortunately, the partial derivative $\frac{\delta \mathbf{J}}{\delta \boldsymbol{q}}$ is not easy to calculate. However, we can use the numerical derivative of the manipulability measure p_s instead. Vector q_l denotes the physical joint limit in the range $[q_{min}, q_{max}]$.

5 SHOE GRINDING

In the shoe assembly process, in order to attach the upper with the corresponding sole, it is necessary to remove a thin layer of the material off the upper surface so that the glue can penetrate the leather. To do this, the robot has to press the shoe against the grinding disc with the desired force while executing the desired trajectory. In the past, there were several approaches how to automate this operation. For mass production, there are special NC machines available. Their main drawback is relatively complicated setup and are therefore not suitable for the custom made shoes. Required flexibility is offered by the robot based grinding cell. In the EUROShoeE project (Dulio and Boer, 2004), a special force controlled grinding head has been designed. The robot manipulated with the grinding head while the shoe remained fixed on the conveyor belt (Jatta et al., 2004). The main drawback of this approach is relatively heavy and expensive grinding head. Additionally, force control can be applied only in one direction. In our approach, the robot holds the shoe and presses it against the grinding disc of a standard grinding machine as used in the shoe production industry. The impedance force control was accomplished by the robot using universal force-torque sensor mounted between the robot wrist and the gripper which holds the shoe last. It is well known that the kinematic redundancy enables greater flexibility in execution of complex trajectories. For example, also humanoid hand dexterity is subjected by it's is kinematical redundancy. We used Mitsubishi Pa10 robot with 7 D.O.F in our roughing cell, which has one degree of redundancy. Additional two degrees of redundancy were obtained by treating the grinding disc as a virtual mechanism. The surface of the grinding disc can be naturally described with the outer surface of the torus, where R and r are the corresponding radius of the grinding disc, as shown in the Fig. 2. Let x be the task (Cartesian) coordinate of the whole system. Assuming that the robot tool position and robot Jacobian is known, the forward kinematics can be easily expressed as

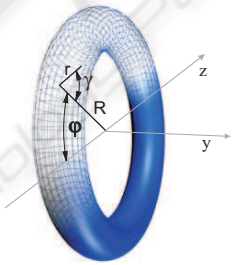

Figure 2: Rotary brush presented as torus.

$$\boldsymbol{x} = \boldsymbol{x}_r + \begin{bmatrix} s_\varphi (R + r c_\gamma) \\ r s_\psi \\ c_\varphi (R + r c_\gamma) \\ 0 \\ -\varphi \\ \psi \end{bmatrix}, \quad (16)$$

while the corresponding Jacobian is

$$\mathbf{J} = \begin{bmatrix} & c_\varphi (R + r c_\psi) & -s_\varphi r s_\gamma \\ & 0 & r c_\psi \\ \mathbf{J}_r & -s_\varphi (R + r c_\psi) & -c_\varphi r s_\gamma \\ & 0 & 0 \\ & -1 & \\ & 0 & 1 \end{bmatrix}. \quad (17)$$

Here, we used the abbreviation $c_\varphi = \cos(\varphi)$, $c_\gamma = \cos(\gamma)$, $s_\varphi = \sin(\varphi)$ and $s_\gamma = \sin(\gamma)$. Thus we have 9 degrees of freedom, 6 of them are required to describe the grinding task, while the remaining three degrees of freedom are used for the obstacle avoidance, joint limits avoidance and singularity avoidance.

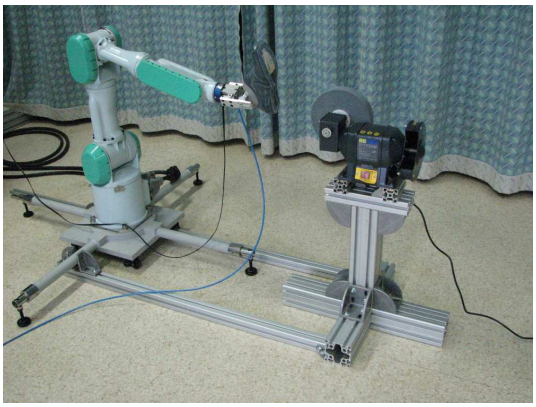

Figure 3: Experimental cell for shoe bottom roughing.

The prototype of the cell is shown in figure 3. It consists of the Mitsubishi Pa10 robot with a force/torque sensor Jr3 mounted in the robot wrist, a grinding machine, a Pa10 robot controller and a cell control computer, which coordinates the task and calculates the required robot torques. The control computer is connected to the robot controller using ArcNet. The frequency rate of the control algorithm (Eq. 9) and the motion optimization algorithm (Eq. 15) is 700 Hz. The grinding path is obtained from CAD model of the shoe. For this purpose, the control computer is connected to the shoe database computer using Ethernet. Unfortunately, CAD model itself can not supply all necessary data for the grinding process. CAD models are usually available for the reference shoe size, therefore, non-linear grading of the shoe shape is necessary for the given size. Additionally, some technological parameters such as material characteristics and shoe sole gluing technology have to be taken into account during the grinding trajectory preparation. For this purpose, we have developed a special CAD expert program, which enables the operator to define additional technological and material parameters. The program then automatically generates the grinding trajectory.

In order to show the efficiency of the proposed algorithm, we defined the shoe grinding trajectory as seen in the Fig 4. Note that without using trajectory optimization is is very hard to execute the given task without splitting the desired trajectory in two or more fragments. Fig 5 shows how the system rotated joints of virtual mechanism in order to avoid the joint limits and to minimize joint velocities of the robot and virtual mechanism.

Similar behavior could be obtained also by using the well known hybrid force-position control, where restricted coordinates are perpendicular to the grinding disc. However, the approach with hybrid force-position control has several disadvantages. First of

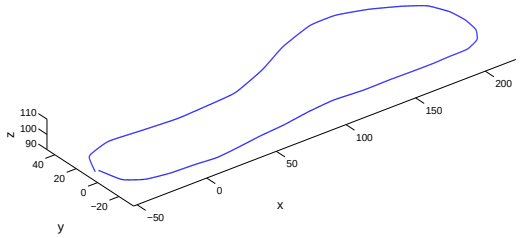

Figure 4: Shoe grinding trajectory.

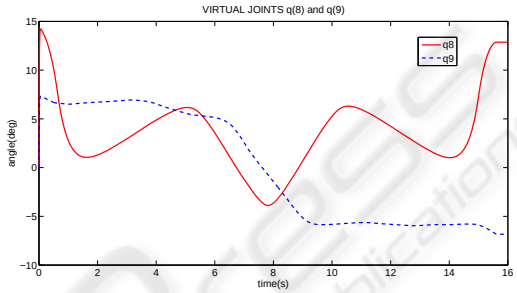

Figure 5: Virtual mechanism angles q8 and q9.

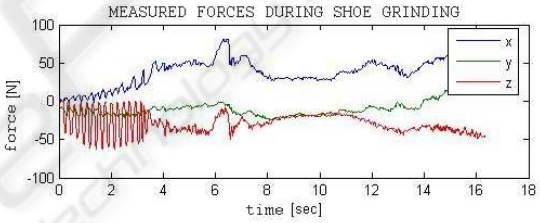

Figure 6: Resulting forces during the shoe grinding.

all, it requires perfect force tracking in order to maintain the tool always perpendicular to the grinding disc. Due to the irregularities of the shoe bottom and the griding disc rotation, this is very hard to obtain. Results of the force tracking shown in Fig. 6 clearly demonstrates, that the resulting force tracking is still imperfect despite of the high sampling frequency of the control algorithm. Note that only the force in z direction was controlled in this case. On the contrary, our approach does not require force control to follow the shape of the grinding disc at all. Therefore, we were able to apply the impedance control law, which is more appropriate for the applications such as grinding and polishing.

6 CONCLUSIONS

In the paper we presented a cell for shoe grinding operation. We proposed a new method of solving the kinematic redundancy which arises from the shape

of the work tool. The main benefit of the proposed method is the simplicity and efficiency. It can be used on the existing robot controllers with very moderate changes of the control algorithm. The proposed method can be applied even in the case of moving obstacles during the task execution, assuming that the shape and position of the obstacle is known. Another, perhaps for the practical implementation even more attractive possibility is to use the proposed approach in the trajectory generation and not in the control algorithm. In this case we can use existing standard industrial robots and we benefit from the the advantages of the kinematic redundancy due to circular shape of the tool. This latter approach was successfully implemented in the cell for custom finishing operations in shoe assembly (Nemec and Zlajpah, 2008). It was also demonstrated that the proposed control has many advantages when compared with the hybrid force-position control, which tracks an unknown shape using only the force sensor data.

REFERENCES

Asada, H. and Slotine, J.-J. (1986). *Robot Analysis and Control*. John Wiley & Sons.

Chaumette, F. and Marchand, . (2001). A redundancy-based iterative approach for avoiding joint limits: Application to visual servoing. In *IEEE Transactions on Robotics and Automation, 17(5)*.

Dulio, S. and Boer, S. (2004). Integrated production plant (ipp): an innovative laboratory for research projects in the footwear field. In *Int. Journal of Computer Integrated Manufacturing, 17(7) : 601-611*.

Jatta, F., Zanoni, L., Fassi, I., and Negri, S. (2004). A roughing/cementing robotic cell for custom made shoe manufacture. In *Int. J. Computer Intergrated Manufacturing, 17(7) : 645–652*.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. In *Int. J. of Robotic Research, 5 : 90 – 98*.

Khatib, O. (1987). A unified approach for motion and force control of robot manipulators:the operational space formulation. In *IEEE Trans. on Robotics and Automation, 3(1) : 43 – 53*.

Nemec, B. and et all (2003). *Technology fostering individual, organisational, and regional development: an international perspective*.

Nemec, B. and Zlajpah, L. (2000). Null velocity control with dinamically consistent pseudo-inverse. In *Robotica, 18 : 513 – 518*.

Nemec, B. and Zlajpah, L. (2001). Experiments with force control of redundant robots in unstructured environment using minimal null-space formulation. In *Journal of Advanced Computational Intelligence, 5(5) : 263 – 268*.

Nemec, B. and Zlajpah, L. (2008). Robotic cell for custom finishing operations. In *Int. J. Computer Intergrated Manufacturing, 21(1) : 33–42*.

Nemec, B., Zlajpah, L., and Omrcen, D. (2007). Comparison of null-space and minimal null-space control algorithms. In *Robotica, 2007, 25(5):511–520*.

Nenchev, D. N. (1989). Redundancy resolution through local optimization: A review. In *J. of Robotic Systems, 6(6) : 769 – 798*.

Oh, Y., Chung, W., Youm, Y., and Suh, I. (1998). Experiments on extended impedance control of redundant manipulator. In *Proc. IEEE/RJS Int. Conf. on Intelligent Robots and Systems, : 1320 – 1325, Victoria*.

Park, J., Chung, W., and Youm, Y. (2002). Characterization of instability of dynamic control for kinematically redundant manipulators. In *Proc. IEEE Conf. Robotics and Automation, : 2400 – 2405, Washington DC*.

Taylor, P. and Taylor, G. (1988). *Garments and Shoe Industry Robots*.