



Modeling & Control of a Meat-Cutting Robotic Cell

Philip Long, Wisama Khalil, Philippe Martinet

► To cite this version:

Philip Long, Wisama Khalil, Philippe Martinet. Modeling & Control of a Meat-Cutting Robotic Cell. ICAR 2013 - 16th International Conference on Advanced Robotics, Nov 2013, Montevideo, Uruguay. pp.1-6, 10.1109/ICAR.2013.6766471 . hal-02462116

HAL Id: hal-02462116

<https://inria.hal.science/hal-02462116>

Submitted on 31 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Modeling & Control of a Meat-Cutting Robotic Cell

Philip Long, Wisama Khalil and Philippe Martinet

Institut de Recherche en Communications et Cybernétique de Nantes, UMR CNRS n° 6597 n° 6597

1 rue de la Noë, École Centrale de Nantes, 44321, Nantes, France.

Email: Firstname.Lastname@ircyn.ec-nantes.fr

Abstract—In this paper, the modeling, simulation and control of a robotic meat cutting cell is described. A dual-arm system is used in the separation task, one arm cuts the meat along a deformable guide line while the second arm grasps and pulls the meat to further increase the opening of the valley. The steps taken to model the cell in order to ensure a realistic interaction between the robots and the flexible object are outlined. A control scheme, using an external vision and force sensors, is proposed that copes with on-line object deformation. The proposed control scheme is validated using the simulator environment.

I. INTRODUCTION

The meat processing industry is the largest sector of the food industry in France accounting for over 25% of the total employees and including over 2,000 companies [1]. However the industry is facing a shortage of skilled labor due to the both hazardous and strenuous working conditions. The ARMS¹ project, *A multi arms Robotic system for Muscle Separation*, aims to contribute to the separation of beef rounds (hindquarters) by an autonomous robotic cell. A muti-arm system is proposed in order to deal with key challenges such as the variability of the target object and its deformable nature.

A general overview of the role of robots in the meat processing industry is outlined in [2], [3]. The Danish pig slaughter industry is an example of a successful robotization of a manual process. The automation process has improved both hygiene and accuracy in the manufacturing environment [4]. In [5], a specific robotic meat cutting cell is analyzed from the point of view of the cutting parameters, while using bones as a positional guide.

The simulation of cutting tasks has been investigated mainly with respect to surgical applications [6]. Generally, a virtual reality model is built to train surgeons. The model should replicate deformable body behavior, thus the cutting force, if considered, is generally used as a haptic output rather than an input that causes the rupture. Several approaches have been proposed: Spring damper systems [7], linked volumes [8], fracture mechanics [9] and numerical finite or boundary element techniques [10]. For robotic tasks deformable objects of limited DOF can be seen as articulated objects i.e. objects that are modeled as mechanisms with passive joints [11]. The object deformations then correspond to joint motion, however this is not suitable to objects that deform in a large number of directions. If the cutting region can be identified beforehand, it can be modeled to account for interaction with the cutting tool. In this case the *non-cutting regions* of the object are generally modeled in a computationally simpler way in order to improve efficiency, while the cutting region is computationally more complex [12], [13].

In the meat cutting application, the region of the cut is known, however the shape of the surface in this region is unknown. Hence the cutting trajectory, which is the input to the cutting robot, must be either measured or estimated. Visual servoing [14] provides a robust control strategy with respect to modeling errors. In this case, the vision system furnishes information about the environment that can then be used as a position trajectory for the robot. This is known as position based visual servoing (PBVS) as opposed to the direct control of image points, image based visual servoing (IBVS) [15], [16]. Since the robots must interact with the environment, force controllers are implemented. Force vision controllers naturally follow on from force position controllers and can be classified into two modes. Hybrid Vision force control [17] which partitions the space in force controlled and vision controlled directions and impedance controllers [18], [19] which enforces a relationship between them. This work address the problem of flexible objects in an industrial environment. In order to cope with the unmodeled behavior external sensors are considered. By using these sensors a cooperative force/vision control scheme is proposed. The contributions of this work are twofold, firstly the modeling of a robotic cutting cell is described that accurately represents the interactions and challenges of the meat cutting environment. Secondly, this cell is used as a pre-experimental testing system for a proposed multi-arm control scheme, exploiting vision, force and redundancy to complete the task.

The outline of the paper is as follows: In section II, the construction of the simulator is described. This includes the modeling of the robots, the meat, the cutting strategy and the visual primitives. This cell consists of two 7-DOF Kuka LWR robots and the target object. The object is deformable and reacts to the robotic forces in an a priori unknown manner. The cutting region is constructed such that the passage of the cutting instrument breaks the bonds in the object. In section III a vision/force control scheme is proposed to complete the cutting task. The visual primitives are simulated and used as control inputs for the cutting robot. The simulated visual system locates 3D points that lie on a guide line (the valley where the cut must be carried out). On the other hand a force controller is implemented on the pulling robot to ensure that separation takes place proportionally to the cut, meaning the robots are coupled through their interaction with the meat. Finally, the results are given in section IV.

II. SYSTEM MODEL

The simulator is executed in Msc Adams software, a multi-body dynamic simulation software. The control system is managed by the Adams control plugin via a cosimulation. The deformable object is created externally then integrated

¹www.arms.ircyn.ec-nantes.fr

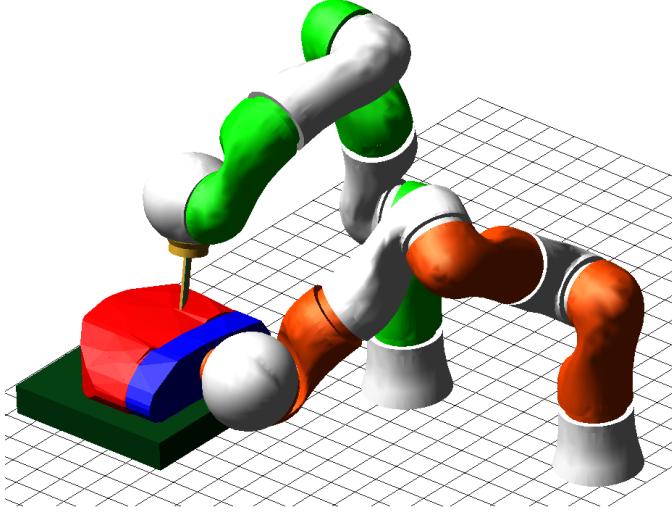


Fig. 1. Robotic Cell

by the Adams Flex plugin. A global view of the simulation environment is given in Fig.1. The cutting robot is rigidly fixed at the origin of the world frame whereas the pulling robot is located at point [0.0, 0.4, 0.0]. The end effector of the pulling robot is fixed to the deformable object at a defined attachment point. The deformable object is rigidly fixed at one side but can move in the plane of the table.

A. Robot Model

The Modified Denavit-Hartenberg (MDH) notation [20] is used to describe the kinematics of the system as given in Table I, where r_7 represents the tool offset of the robots. From these parameters, for each robot $i = c, p$, the following models are obtained:

$$\begin{aligned} \mathbf{x}_i &= \begin{bmatrix} \mathbf{R}_i & \mathbf{p}_i \\ 0 & 1 \end{bmatrix} & \dot{\mathbf{x}}_i &= \mathbf{J}_i \dot{\theta}_i & \ddot{\mathbf{x}}_i &= \mathbf{J}_i \ddot{\theta}_i + \dot{\mathbf{J}}_i \dot{\theta}_i \quad (1) \\ \tau_i &= \mathbf{A}_i \ddot{\theta}_i + \mathbf{H}_i + \mathbf{J}_i^T \mathbf{h}_i & & & & \quad (2) \end{aligned}$$

The Cartesian position, kinematic screw, acceleration are denoted as \mathbf{x}_i , $\dot{\mathbf{x}}_i$, $\ddot{\mathbf{x}}_i$. \mathbf{R}_i represents the orientation and \mathbf{p}_i the position of the task frame, \mathbf{J}_i is the Jacobian matrix and θ_i the vector of joint coordinates. The dynamic parameters are taken from the equivalent CAD model. The inertia matrix and the matrix of centrifugal and gravity torques are denoted as \mathbf{A}_i , and \mathbf{H}_i . The Cartesian wrench is denoted as \mathbf{h}_i while τ_i is the joint torque.

TABLE I. MDH PARAMETERS OF KUKA ROBOT

j	d	α	θ	$r(m)$
1	0	0	θ_1	0
2	0	$\frac{\pi}{2}$	θ_2	0
3	0	$-\frac{\pi}{2}$	θ_3	0.4
4	0	$-\frac{\pi}{2}$	θ_4	0
5	0	$\frac{\pi}{2}$	θ_5	0.39
6	0	$\frac{\pi}{2}$	θ_6	0
7	0	$-\frac{\pi}{2}$	θ_7	r_7

B. Deformable object model

The deformable object is at the center of the simulation strategy, it represents the beef shoulder that must be separated.

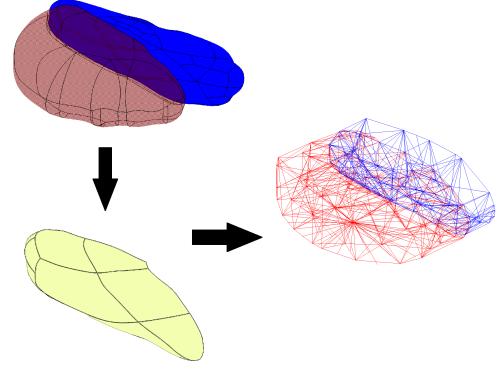


Fig. 2. From 3D scan to deformable body

However the objective is not to replicate exactly the dynamic behavior of the beef round but to control the robotic cutting cell when interacting with an unknown flexible object. The simulated object must react to pulling forces in a realistic manner. Furthermore the object must be *separable*, i.e. react coherently to the incisions of the robot controlled knife.

The surface of separation is distinguished by a set of aponeurosis, that are similar to tendons, acting as links between the main beef muscles. The aponeurosis can store elastic energy, then recoil when unloaded. The meat muscle is pre-computed offline and behaves as a linear elastic object. The aponeurosis are modeled as a series of spring damper systems fixed to the muscle at discrete points.

To model the object, firstly a visual scan of a generic beef round is obtained and converted into a 3D-geometry. The cutting surface, is extracted from the geometry, then two objects are modeled using the surface as demonstrated in Fig.2. These objects represent the meat after the cut has been fully completed. Secondly, both objects are meshed volumetrically using a commercial finite element package. Attachment points are placed at certain nodes, notably on the cutting surface. These attachment points allow forces and constraints to be applied to the object in the simulator environment. Finally the finite element program exports a modal neutral file (.mnf). This file, containing the object geometry, nodal mass and inertia, mode shapes and generalized mass and stiffness for the mode shapes, can be imported directly into the simulator environment.

The second deformable model represents the intermediate layer of attachment between the two deformable objects, the aforementioned aponeurosis. Using the attachment points as nodes a series of spring-damper systems are spread across the cutting the surface. When the two objects are perfectly mated and at rest, the spring damper systems are at their equilibrium points.

C. Cutting Process Model

At each iteration the position of the cutting tool in the world frame, $\mathbf{p}_c = [x_c, y_c, z_c]$ is compared with the position of the spring nodes located on the cutting surface. A virtual spring-damper \mathbf{a} is comprised of two nodes $\mathbf{a}_1 = [x_1, y_1, z_1]$ and $\mathbf{a}_2 = [x_2, y_2, z_2]$. In order to simplify the calculation, it

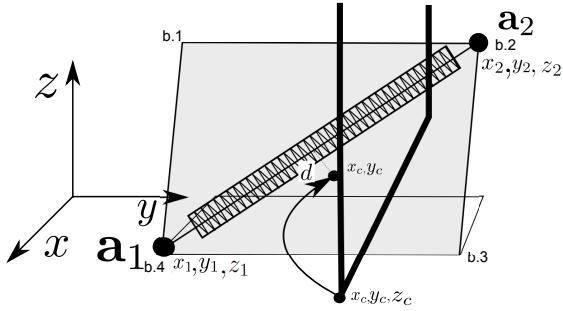


Fig. 3. Cutting model

is assumed that the cutting tool always approaches from the positive z direction. Thus to check if a virtual spring-damper a has been cut, the following conditions must be met, as shown in Fig.3:

Condition 1.

\mathbf{p}_c must be below the virtual spring-damper

$$z_c \leq \min(z_1, z_2) \quad (3)$$

Condition 2.

The projection of \mathbf{p}_c into the xy plane must lie within the bounding box ($b1 \dots b4$) of the line segment $|\mathbf{a}_1\mathbf{a}_2|$

$$\min(x_1, x_2) \leq x_c \leq \max(x_1, x_2) \quad (4)$$

$$\min(y_1, y_2) \leq y_c \leq \max(y_1, y_2) \quad (5)$$

Condition 3.

The projection of \mathbf{p}_c into the xy plane must lie on the line segment $|\mathbf{a}_1\mathbf{a}_2|$ (d is a tolerance):

$$(y_2 - y_1)(x_2 - x_c) - (y_2 - y_k)(x_2 - x_1) < d \quad (6)$$

D. Generation of vision primitives

In the ARMS project, a supplementary robot, equipped with an eye-in-hand camera, will provide the location of the guide line in space. by following this line the separation of the meat can be achieved. Thus the guideline is the visual primitive that must be taken into account in our environment. In section II-B, it is shown that the surface can be discretized in order to create an intermediate cutting layer. By using the location of these points, the surface can be reconstructed in the control environment using a surface interpolation procedure, the Matlab function **TriScatteredInterp**. Since the interpolated surface is known, the guide line can be reconstructed at any given depth. By extracting the guide line at the maximum height, the equivalent visual primitive is created.

III. CONTROL SCHEME

In this section, the problem of controlling the robotic cell is addressed. Each arm is controlled independently in their respective tasks while the coupling effects are felt through the interaction with the deformable body. A desired value of a parameter is represented as the same variable with the superscript d . The prefix Δ denotes the difference between a desired value and the current value of variable, for example $\Delta\mathbf{x}_i = \mathbf{x}_i^d - \mathbf{x}_i$. A global overview of the control scheme is given in Fig. 4.

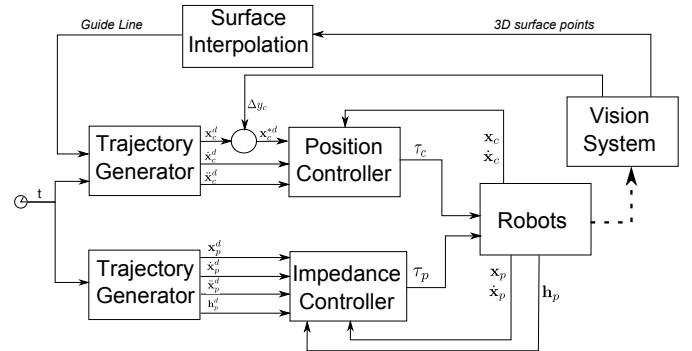


Fig. 4. Global Control Scheme

A. Task Definition

1) *Primary Task*: The primary task is the separation of the two meat muscles. To complete this task, the spring-damper links, representing the aponeurosis, must be removed by the passage of the knife. The cutting tool must only interact with the aponeurosis and avoid cutting into the meat muscles at either side. This necessitates a series of cuts, called passages, at increasing depths along the visible guide line. The pulling robot is responsible for creating an opening so that the knife can pass, unobstructed, along the guide line. After each passage the opening will increase allowing the knife to move deeper into the valley until the two objects have been completely separated.

2) *Secondary Task*: Since the robots have 7-DOF, the null space motion must be controlled. A secondary task \mathbf{Z} , is used to damp any motion in the null space. The secondary task is projected into the primary task using the classical orthogonal projector $\mathbf{P}_i = \mathbf{I} - \mathbf{J}_i^+ \mathbf{J}_i$, where \mathbf{I} is the 7×7 identity matrix and $+$ denotes the pseudoinverse.

B. Cutting Robot

At the beginning of each passage, the visual primitives are used to reconstruct the guide line as outlined in section II-D. A curve is then fitted to the guide line. This curve is represented by a polynomial expression. For a given cutting depth z , the desired trajectory is defined by:

$$y^d = a_2(x^d)^2 + a_1x^d + a_0 \quad (7)$$

The total curvilinear length, D of the polynomial curve is obtained by integrating (8), where a and b are the extremities of the surface.

$$D = \int_b^a \sqrt{1 + \frac{\partial y^d}{\partial x^d}} dx^d \quad (8)$$

A variable $s(t)$ representing the curvilinear distance along the curve is defined using the temporal constraints (9), (10), (11).

$$s(t=0) = 0 \quad s(t=t_{final}) = D \quad (9)$$

$$\dot{s}(t=0) = 0 \quad \dot{s}(t=t_{final}) = 0 \quad (10)$$

$$\ddot{s}(t=0) = 0 \quad \ddot{s}(t=t_{final}) = 0 \quad (11)$$

For every t , $x^d(t)$ can be obtained by substituting the value of s into (8) and solving for the upper limit b . $y^d(t)$ is then calculated from (7). To complete the cutting task definition, the orientation of the knife must be considered. Before each passage, the orientation of the knife is equal to the 3×3 rotation matrix \mathbf{R}_{init} given in (12). The cutting side of the knife must be aligned to the cutting direction, the approach angle is defined by the angle θ , a rotation around the z axis. The desired rotation matrix during the passage, $\mathbf{R}^d(t)$ is calculated from (14).

$$\mathbf{R}_{init} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (12)$$

$$\theta = \frac{\partial y^d}{\partial x^d}(x^d(t)) \quad (13)$$

$$\mathbf{R}^d(t) = \mathbf{R}_{init} \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (14)$$

From the above a desired trajectory is generated in position and orientation, velocity and acceleration, i.e. \mathbf{x}^d , $\dot{\mathbf{x}}^d$ and $\ddot{\mathbf{x}}^d$. To track the desired variables, using Cartesian computed torque [21] the desired Cartesian acceleration, \mathbf{w}_x , is defined as:

$$\mathbf{w}_x = \ddot{\mathbf{x}}^d + \mathbf{K}_d(\Delta \dot{\mathbf{x}}) + \mathbf{K}_p(\Delta \mathbf{x}) - \dot{\mathbf{J}}_i \dot{\mathbf{q}} \quad (15)$$

where \mathbf{K}_d \mathbf{K}_p are positive gains. \mathbf{w}_x is then transformed to the joint space, and a new desired acceleration exploiting the redundancy of the system is defined

$$\mathbf{w} = \mathbf{J}_i^+(\mathbf{w}_x + \mathbf{P}_i \mathbf{Z}) \quad (16)$$

Finally a joint torque realising this acceleration is obtained

$$\tau_i = \mathbf{A}_i \mathbf{w} + \mathbf{H} \quad (17)$$

However the object deforms during the passage of the knife changing the profile of the cutting surface. This is due both to the force applied by the pulling robot and to the effects of cutting the aponeurosis in the intermediate layer. In order to compensate for this motion, the desired position is updated online by using, y_g , the exact position of the guide line extracted from the visual primitive. y^d is updated as:

$$y^{*d}(t) = y^d(t) + \Delta y \quad (18)$$

$$\Delta y = y_g - y_c \quad (19)$$

C. Pulling Robot

The pulling robot works is force controlled. The desired behavior is a gradual opening of the cutting valley as the cutting depth increases. For each passage the number of links between the deformable objects is reduced leading to a smaller retaining force. An impedance controller [22] is applied where $\Delta \mathbf{h} = \mathbf{h}_d - \mathbf{h}$ the difference between the desired and current pulling force, where λ is a gain matrix representing the inverse of the desired inertial behavior. Equation (15) is modified to include these terms:

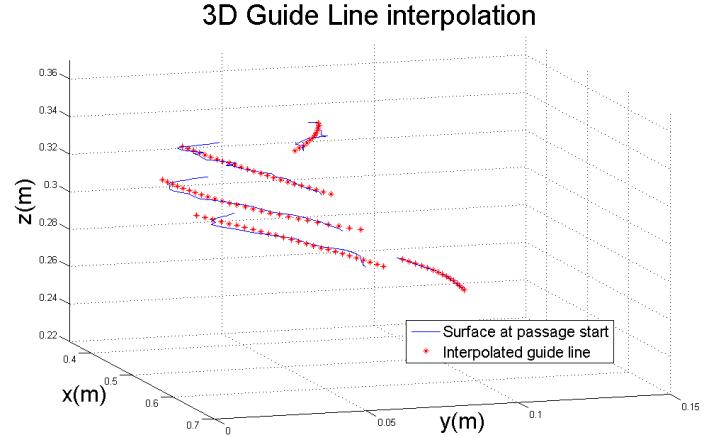


Fig. 5. Guide Line Interpolation

$$\ddot{\mathbf{x}}^d = \ddot{\mathbf{x}} + \lambda (\mathbf{K}_d(\Delta \dot{\mathbf{x}}) + \mathbf{K}_p(\Delta \mathbf{x}) - \mathbf{K}_f(\Delta \mathbf{h})) - \dot{\mathbf{J}}_i \dot{\mathbf{q}} \quad (20)$$

IV. RESULTS

Two different experiments are discussed in this section, differing with respect to the reference trajectory:

- 1) Using the interpolated guide line state at the beginning of each passage
- 2) Locally updating the guide line using predicted errors

In order to fully separate the muscles the knife must cut a distance of 80mm. The meat is separated by a repeatedly cutting along the surface of separation with the knife. The meat is positioned on a table at a height of 270mm. The cutting depth per passage in the world coordinates is shown in Table II and a 3D view of the interpolated cutting trajectories is given in Fig.5. A snapshot of the system after certain cutting passages is given in Fig.6

TABLE II. CUTTING DEPTHS PER PASSAGE

Passage number	1	2	3	4	5
Cutting Depth (m)	0.35	0.33	0.31	0.29	0.27

In Fig.7 and Fig.8, the motion in the $x - y$ plane for all passages of the cutting tool is shown. The graphs show: the initial guide before cutting has commenced, the interpolated trajectory for this line, the position of the guide line during the cutting trajectory and the position of the robot cutting tool. A large difference between the initial guide line before cutting has begun and the position of the guide line during the trajectory is shown. This is due to the deformation of the object when links are severed. In Fig.8, we can see that the local visual update compensates for these changes. This compensation changes the robot motion the cutting tool is closer to the current guide line position.

Fig.9 shows the sensed forces at pulling frame with the location of the pulling frame. The Pulling forces remain constant at 100N. It can be seen that during the trajectory the position of the pulling frame changes in the Y-position, corresponding to the cutting of the aponeurosis. As the links

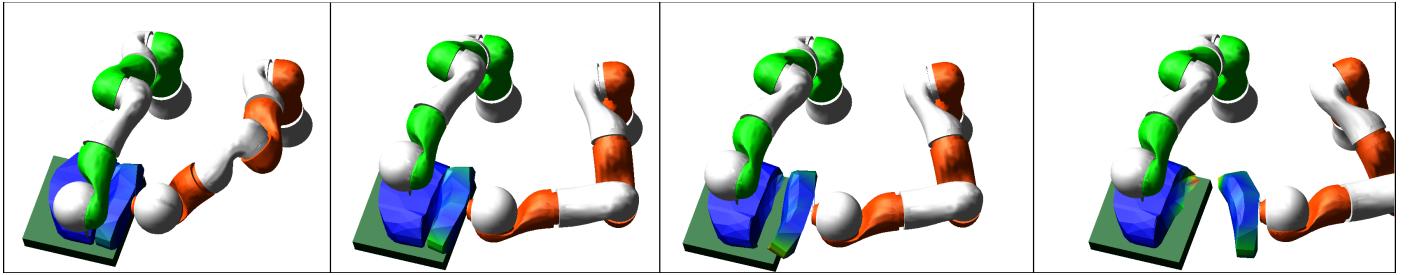


Fig. 6. Snapshot of separation process

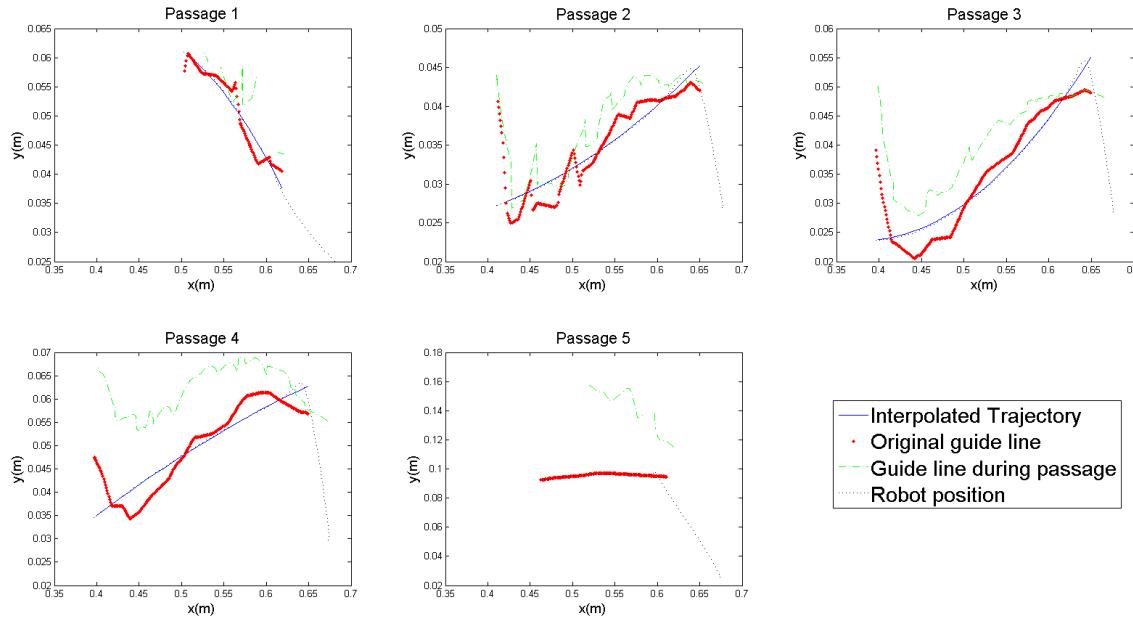


Fig. 7. Robot Trajectory for each passage

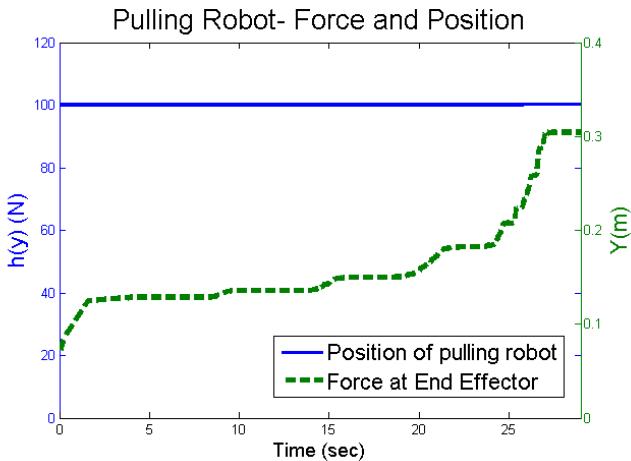


Fig. 9. Force and Position in Y-direction

are cut the retaining elastic forces are unable to withstand the pulling forces leading to a gradual opening of the deformable object.

V. CONCLUSIONS

This paper has described the modeling and control of a robotic meat cutting cell. The modeling process for the robot, the meat, visual primitives and the interaction between modules are described. A control scheme is proposed using visual and force data to cope with uncertainties about the object behavior. The results show how, both local and global visual primitives, can be used to compensate for object deformations.

Future work will concentrate on the experimental validation of the proposed control schemes in an industrial environment. Furthermore studies will be carried out to integrate a third eye-in-hand robot into the simulator, to optimize redundant behavior.

ACKNOWLEDGMENT

This paper describes work carried out in the framework of the ARMS project. ARMS : A multi arms robotic system for muscle separation, is partly funded by the ANR (Agence Nationale de la Recherche), reference ANR-10-SEGI-000.

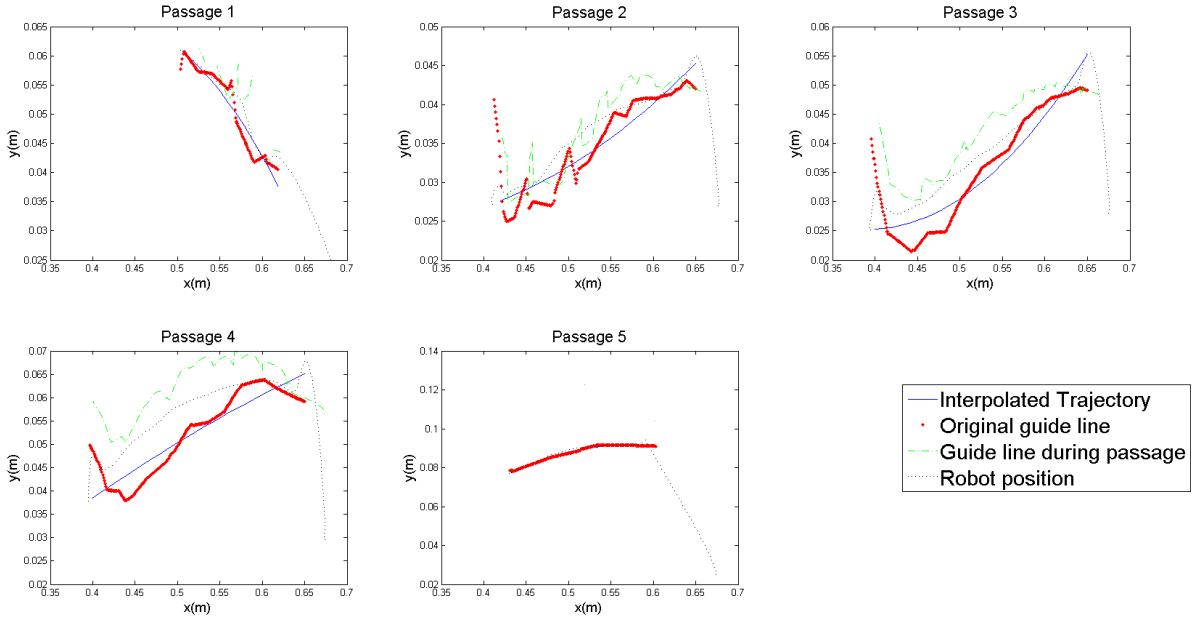


Fig. 8. Robot Trajectory modified by local updates

REFERENCES

- [1] Institut national de la statistique et des études économiques (INSEE), “Industrie agroalimentaire,” http://www.insee.fr/fr/themes/document.asp?ref_id=T10F181, 2009, accessed: 2013-04-22.
- [2] G. Purnell, “Robotic equipment in the meat industry,” *Meat science*, vol. 49, pp. S297–S307, 1998.
- [3] J. Billingsley, A. Visala, and M. Dunn, “Robotics in agriculture and forestry,” in *Springer handbook of robotics*, B. Siciliano and O. Khatib, Eds. Springer-Verlag, Berlin, Germany, 2008, pp. 1065–1077.
- [4] L. Hinrichsen, “Manufacturing technology in the danish pig slaughter industry,” *Meat science*, vol. 84, no. 2, pp. 271–275, 2010.
- [5] G. Guire, L. Sabourin, G. Gogu, and E. Lemoine, “Robotic cell with redundant architecture and force control: application to cutting and boning,” in *2010 IEEE 19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD)*. IEEE, 2010, pp. 99–104.
- [6] S. Misra, K. Ramesh, and A. Okamura, “Modeling of tool-tissue interactions for computer-based surgical simulation: a literature review,” *Presence: Teleoperators and Virtual Environments*, vol. 17, no. 5, pp. 1–41, 2008.
- [7] U. Kühnapfel, H. Cakmak, and H. Maaß, “Endoscopic surgery training using virtual reality and deformable tissue simulation,” *Computers & Graphics*, vol. 24, no. 5, pp. 671–682, 2000.
- [8] S. Friskin-Gibson, “Using linked volumes to model object collisions, deformation, cutting, carving, and joining,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 333–348, 1999.
- [9] M. Mahvash and V. Hayward, “Haptic rendering of cutting: A fracture mechanics approach,” *Haptics-e*, vol. 2, no. 3, pp. 1–12, 2001.
- [10] U. Meier, O. Lopez, C. Monserrat, M. Juan, and M. Alcaniz, “Real-time deformable models for surgery simulation: a survey,” *Computer methods and programs in biomedicine*, vol. 77, no. 3, pp. 183–197, 2005.
- [11] B. Balaguer and S. Carpin, “Combining imitation and reinforcement learning to fold deformable planar objects,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1405–1412.
- [12] H. Delingette, S. Cotin, and N. Ayache, “A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation,” in *Computer Animation, 1999. Proceedings*, 1999, pp. 70–81.
- [13] L. Vigneron, J. Verly, and S. Warfield, “On extended finite element method (xfem) for modelling of organ deformations associated with surgical cuts,” *Medical Simulation*, pp. 134–143, 2004.
- [14] B. Espiau, F. Chaumette, and P. Rives, “A new approach to visual servoing in robotics,” *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 3, pp. 313 –326, jun 1992.
- [15] F. Chaumette, “Visual servo control. I. Basic approaches,” *Robotics & Automation, no. December*, pp. 82–90, 2006.
- [16] N. Andreff, T. Dallej, P. Martinet *et al.*, “Image-based visual servoing of gough-stewart parallel manipulators using legs observation,” in *8th International IFAC Symposium on Robot Control (SYROCO 2006)*, 2006.
- [17] K. Hosoda, K. Igarashi, and M. Asada, “Adaptive hybrid control for visual and force servoing in an unknown environment,” *Robotics & Automation Magazine, IEEE*, vol. 5, no. 4, pp. 39–43, 1998.
- [18] V. Lippiello, B. Siciliano, and L. Villani, “Robot interaction control using force and vision,” in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 1470–1475.
- [19] Y. Mezouar, M. Prats, and P. Martinet, “External hybrid vision/force control,” in *Intl. Conference on Advanced Robotics (ICAR07)*, 2007.
- [20] W. Khalil and J. Kleinfinger, “A new geometric notation for open and closed-loop robots,” in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 1986, pp. 1174–1179.
- [21] O. Khatib, “A unified approach for motion and force control of robot manipulators: The operational space formulation,” *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.
- [22] N. Hogan, “Impedance control: An approach to manipulation: Part II applications,” *Journal of dynamic systems, measurement, and control*, vol. 107, no. 2, p. 17, 1985.