Robotic Meat Cutting

Philip Long, Amine Abou Moughlbay, Wisama Khalil and Philippe Martinet

Institut de Recherche en Communications et Cybernétique de Nantes, UMR CNRS n° 6597
1 rue de la Noë, École Centrale de Nantes, 44321, Nantes, France.
Email:Philip.Long@irccyn.ec-nantes.fr

***Abstract*—In this paper, the modeling, simulation and control of a robotic meat cutting cell is described. A multi-arm system is used in the separation task. A control scheme, using vision and force sensors, is proposed that copes with on-line object deformation. The control scheme and modeling process draws on those currently in use in the field of medical simulation. However the global objectives are fundamentally different. The contributions of this work are twofold. Firstly the modeling of a robotic cell 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.**

I. Introduction

Whether domestically or industrially, in order to cut and separate meat, three distinct tools must be used: a retention system, a trajectory generator, and the cutting system. For humans, these tools are knives, forks and eyes. The fork clamps and holds the object in place. The eyes act as a trajectory generator providing and updating the path for the cutting tool. The knife tracks this path while applying a force in order to separate the object. Likewise a butcher in an industrial environment, carefully separates the beef muscle with a knife while identifying guide features visually. The meat is held in place with a clamping system that either comprises of the butcher's free hand or a specialized jig. In this work, the modeling and control strategy of a robotic meat cutting cell is outlined. A multi-arm system is proposed that functions in a similar way to a human worker, i.e. the first arm follows a cutting trajectory while the second is used to open up the object by pulling and tearing at it. The third and final arm acts as the eyes of the system extracting the location of the guide line.

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 difficult challenges, principally the increasingly difficulty in adequately filling the skilled labor requirements. This is primarily due to the difficult working conditions including both the repetitive, strenuous tasks leading to musculoskeletal illness [2], and the uncomfortable working environment. As a consequence the sector must deal with rising labor costs at a time when the competition from regions of lower labor costs is growing. Therefore it has become imperative to robotize certain tasks.

The ARMS[1] project, *A multi arms Robotic system for Muscle Separation* is an ANR (Agence Nationale de la Recherche) ARPEGE project, that aims to contribute to the robotization of the meat processing industry, in particular to the separation of beef rounds (hindquarters). The goal is to design a robotic cell that can autonomously separate the beef rounds in an industrial environment.

A general overview of the role of robots in the meat processing industry is outlined in [3], [4]. The Danish pig slaughter industry is an example of a successful robotization of a manual process in order to maintain competitiveness in a high cost labor region. Studies have shown that the automation process has improved both hygiene, accuracy and animal welfare in the manufacturing environment [5]. In [6] the design of an intelligent deboning robot is described in order to improve an already semi-robotized process. In [7], a specific robotic meat cutting cell is analyzed from the point of view of the cutting parameters, using bones as a positional guide.

Before the experimental work commences it is important to validate control schemes as much as possible using simulation techniques. The simulation of cutting tasks has been investigated mainly with respect to surgical applications [8]. Generally, a virtual reality model is built to train surgeons. This work has parallels with the meat separation task however the final objectives are fundamentally different. For example 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 [9], linked volumes [10], fracture mechanics [11] and numerical finite or boundary element techniques [12]. 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 [13].

In the meat cutting application, the region of the cut is known. Thus this region can effectively be modeled by a separate process in a manner that allows the interaction of the cutting tool and the meat muscle. On the other hand, the shape of the cutting surface within this region is unknown. Hence the cutting trajectory, which is the input to the cutting robot, must be either measured or estimated during the task. 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].

[1]www.arms.irccyn.ec-nantes.fr

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 third robot is positioned using a visual servoing control law. This robot is responsible for the extraction of the visual primitives.

II. SYSTEM MODEL

The simulator [17] is executed in Msc Adams software, a multi-body dynamic simulation software. 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, the pulling robot is at point $[0.0, 0.4, 0.0]$ and the vision robot at point $[1.0, 0.0, 0.0]$. 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. Robotic System

The system is composed of three Kuka LWR robots, a lightweight kinematically redundant robot of 7-DOF [18]. The Modified Denavit-Hartenberg (MDH) notation [19] 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 $i = c, p, v$, (cutting, pulling and vision) the following models are obtained:

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{R}_i & \mathbf{p}_i \\ 0 & 1 \end{bmatrix} \qquad \dot{\mathbf{x}}_i = \mathbf{J}_i\dot{\theta}_i \quad \ddot{\mathbf{x}}_i = \mathbf{J}_i\ddot{\theta}_i + \dot{\mathbf{J}}_i\dot{\theta}_i \tag{1}$$

$$\tau_i = \mathbf{A}_i\ddot{\theta}_i + \mathbf{H}_i + \mathbf{J}_i^T\mathbf{h}_i \tag{2}$$

The Cartesian position, kinematic screw, acceleration are denoted as $\mathbf{x}$, $\dot{\mathbf{x}}$, $\ddot{\mathbf{x}}$. $\mathbf{R}_i$ 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

Joint	α	d	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 represents the beef shoulder that must be separated. 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. Moreover since the aponeurosis having a whitish string-like appearance; they can be distinguished from the meat muscle visually. The meat muscle is precomputed offline and behaves as a linear elastic object. To model the object, the following steps are taken:

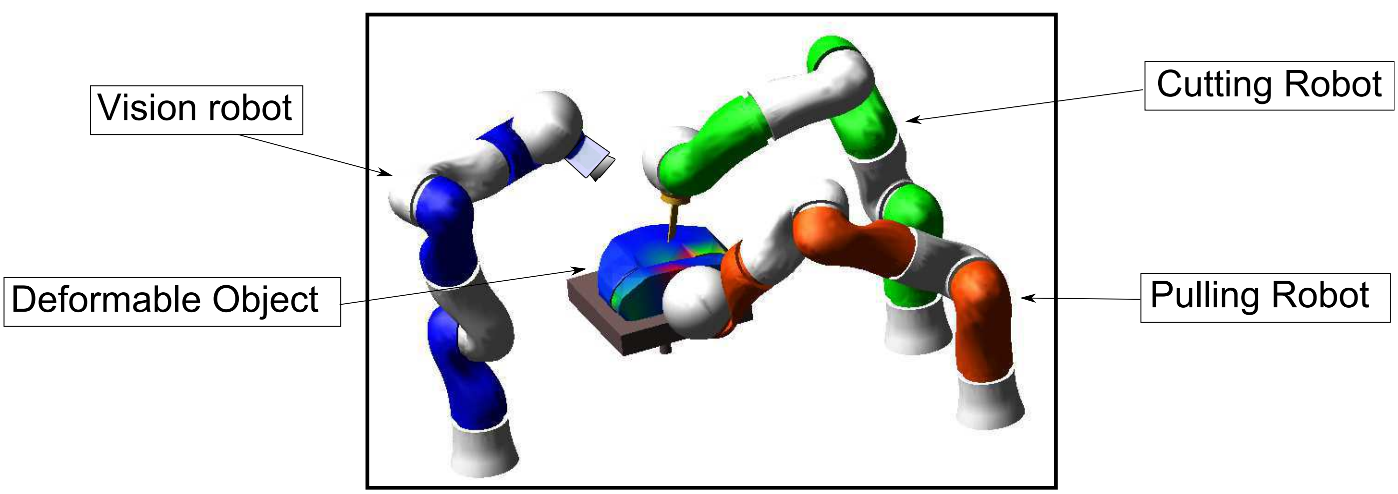

Fig. 1: Global Simulator

1) **Geometry**: A visual scan of a generic beef round is obtained. The cutting surface, is extracted from the geometry, then two objects are modeled using the surface as demonstrated in Fig.2. This means that a simplified model is obtained from the homogeneous beef muscles parts whereas the interface surface is modeled exactly.

2) **Discretization:** The first deformable model, representing the two beef muscles, is 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.
The second deformable model, representing the aforementioned aponeurosis, is constructed using these attachment points. The aponeurosis are modeled as a series of spring-damper systems spread across the cutting the surface terminating at the nodes. When the two objects are perfectly mated and at rest, the spring damper systems are at their equilibrium points.

3) **Importation:** 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.

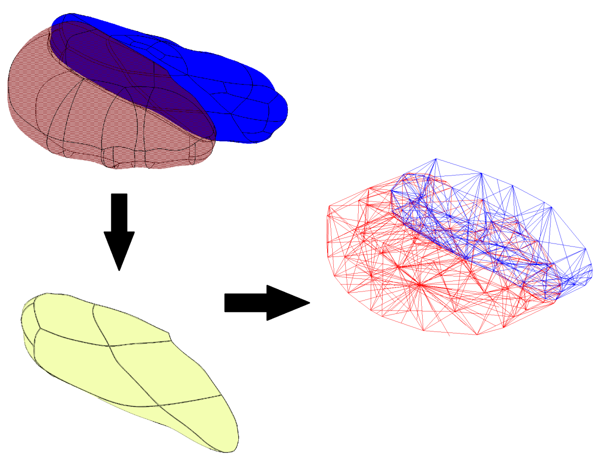

Fig. 2: From 3D scan to deformable body

C. 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. PROPOSED CONTROL SCHEME

A desired value of a parameter is represented with the same variable as the current parameter but with the addition of 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. 3.

A. Task Definition

The primary task is the separation of the two meat muscles. To complete this task the aponeurosis must be removed or severed. 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.

At each passage the guide line location is re-evaluated. This is done by a vision robot that observes the scene. Redundancy management ensures that the vision robot avoids occlusions and optimizes the field of view.

Thus the multi-arm system is coupled indirectly by the deformable motion of the meat muscles.

B. Cutting Robot

At the beginning of each passage, a curve is 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:

$$y^d = a_2(x^d)^2 + a_1 x^d + a_0 \tag{3}$$

The total curvilinear length, D of the polynomial curve is obtained by integrating (4), where a and b are the extremities of the surface.

$$D = \int_b^a \sqrt{1 + \frac{\partial y^d}{\partial x^d}}\, \partial x^d \tag{4}$$

A variable $s(t)$ representing the curvilinear distance along the curve is defined using the temporal constraints (5), (6), (7).

$$s(t=0) = 0 \qquad s(t = t_{final}) = D \tag{5}$$
$$\dot{s}(t=0) = 0 \qquad \dot{s}(t = t_{final}) = 0 \tag{6}$$
$$\ddot{s}(t=0) = 0 \qquad \ddot{s}(t = t_{final}) = 0 \tag{7}$$

For every t, $x^d(t)$ can be obtained by substituting the value of s into (4) and solving for the upper limit b. $y^d(t)$ is then calculated from (3).

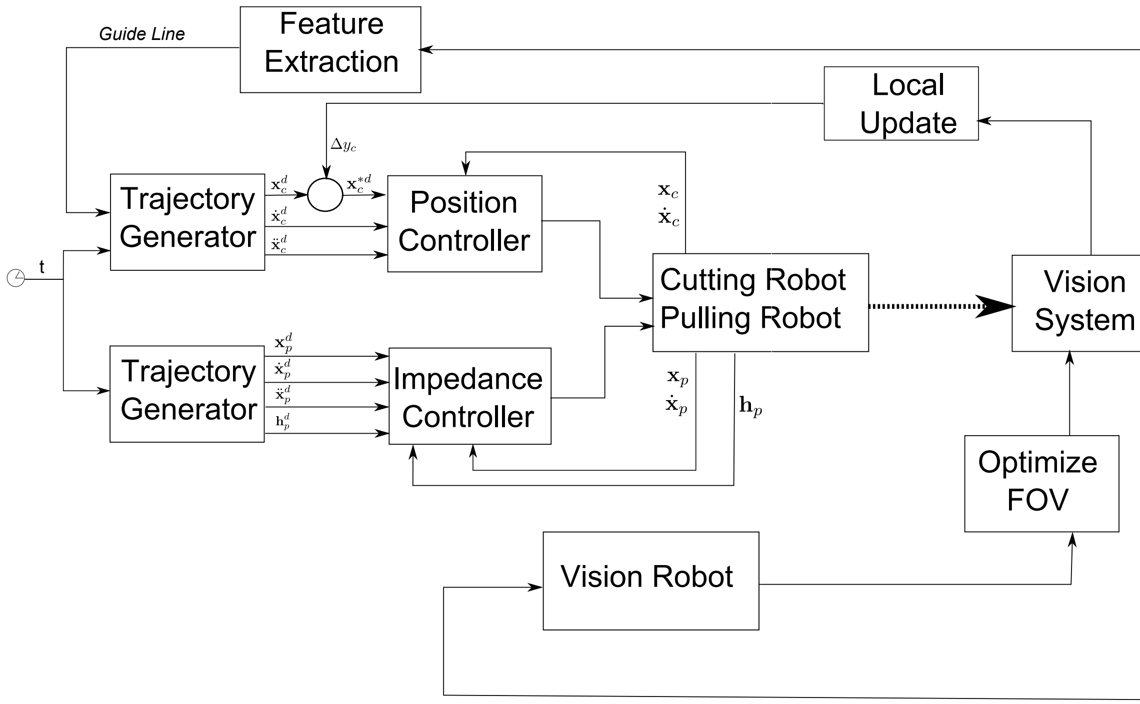

Fig. 3: Global Control Scheme

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 (8). 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 (10).

$$\mathbf{R}_{init} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{8}$$

$$\theta = \frac{\partial y^d}{\partial x^d}(x^d(t)) \tag{9}$$

$$\mathbf{R}^d(t) = \mathbf{R}_{init} \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

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 $\ddot{\mathbf{x}}^\mathbf{d}$. To track the desired variables, using Cartesian computed torque[20] the desired Cartesian acceleration,$\mathbf{w}_x$, is defined as:

$$\mathbf{w}_x = \ddot{\mathbf{x}}^\mathbf{d} + \mathbf{K_d}\left(\mathbf{\Delta}\dot{\mathbf{x}}\right) + \mathbf{K_p}\left(\mathbf{\Delta x}\right) - \dot{\mathbf{J}}_\mathbf{i}\dot{\mathbf{q}} \tag{11}$$

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

$$\mathbf{w} = \mathbf{J}_i^+\left(\mathbf{w}_x + \mathbf{P}_i\mathbf{Z}\right) \tag{12}$$

Finally a joint torque realizing this acceleration is obtained

$$\tau_i = \mathbf{A}_i\mathbf{w} + \mathbf{H} \tag{13}$$

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:

$$y^{*d}(t) = y^d(t) + \Delta y \tag{14}$$

$$\Delta y = y_g - y_c \tag{15}$$

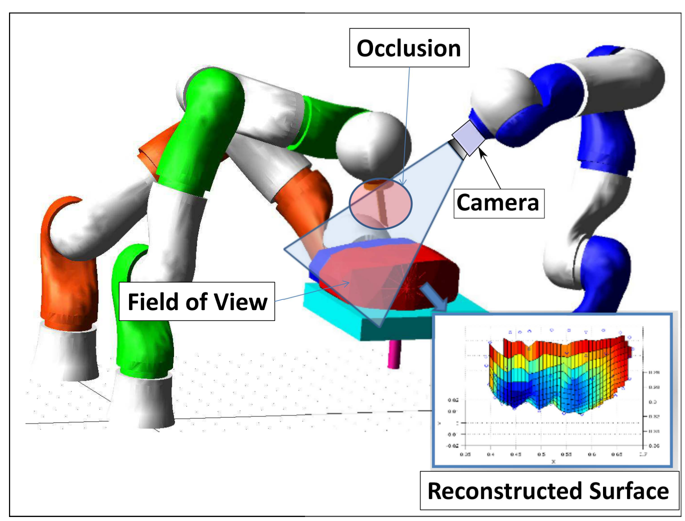

Fig. 4: Vision System

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 [21] is applied where $\Delta \mathbf{h} = \mathbf{h}_d - \mathbf{h}$ is the difference between the desired and current pulling force, where λ is a gain matrix representing the inverse of the desired inertial behavior. Equation (11) is modified to include these terms:

$$\ddot{\mathbf{x}}^{\mathbf{d}} = \ddot{\mathbf{x}} + \lambda \left(\mathbf{K_d} \left(\mathbf{\Delta \dot{x}} \right) + \mathbf{K_p} \left(\mathbf{\Delta x} \right) - \mathbf{K_f} (\mathbf{\Delta h}) \right) - \dot{\mathbf{J}}_{\mathbf{i}} \dot{\mathbf{q}} \tag{16}$$

D. Vision Robot

The visual robot is controlled in image space. The guide line is extracted by the camera and parameterized into a feature variable $\mathbf{s}_{im}$. A desired image, which is defined to optimize the field of view is defined as $\mathbf{s}_d$. The desired velocity is generated using the interaction matrix [22]:

$$\dot{\mathbf{x}}^d = -\lambda \mathbf{L}_s^+ \left(\mathbf{s}_d - \mathbf{s}_{im} \right) \tag{17}$$

Fig. 4 shows the system from the point of view of the vision robot. A field of view is supposed as a cone with its origin at the origin of the camera. The task of the robot is to keep the cutting surface within the field of view at all times. An occlusion occurs when a foreign object prevents the vision robot from obtaining the surface parameters. The position of the node points are then exported to the simulator environment where they are used to reconstruct a surface and thus generate a trajectory.

IV. DISCUSSION

A snapshot of the system after cutting passages is given in Fig. 5. The figure shows the gradual successful separation of the meat muscles, in particular the state of the system is

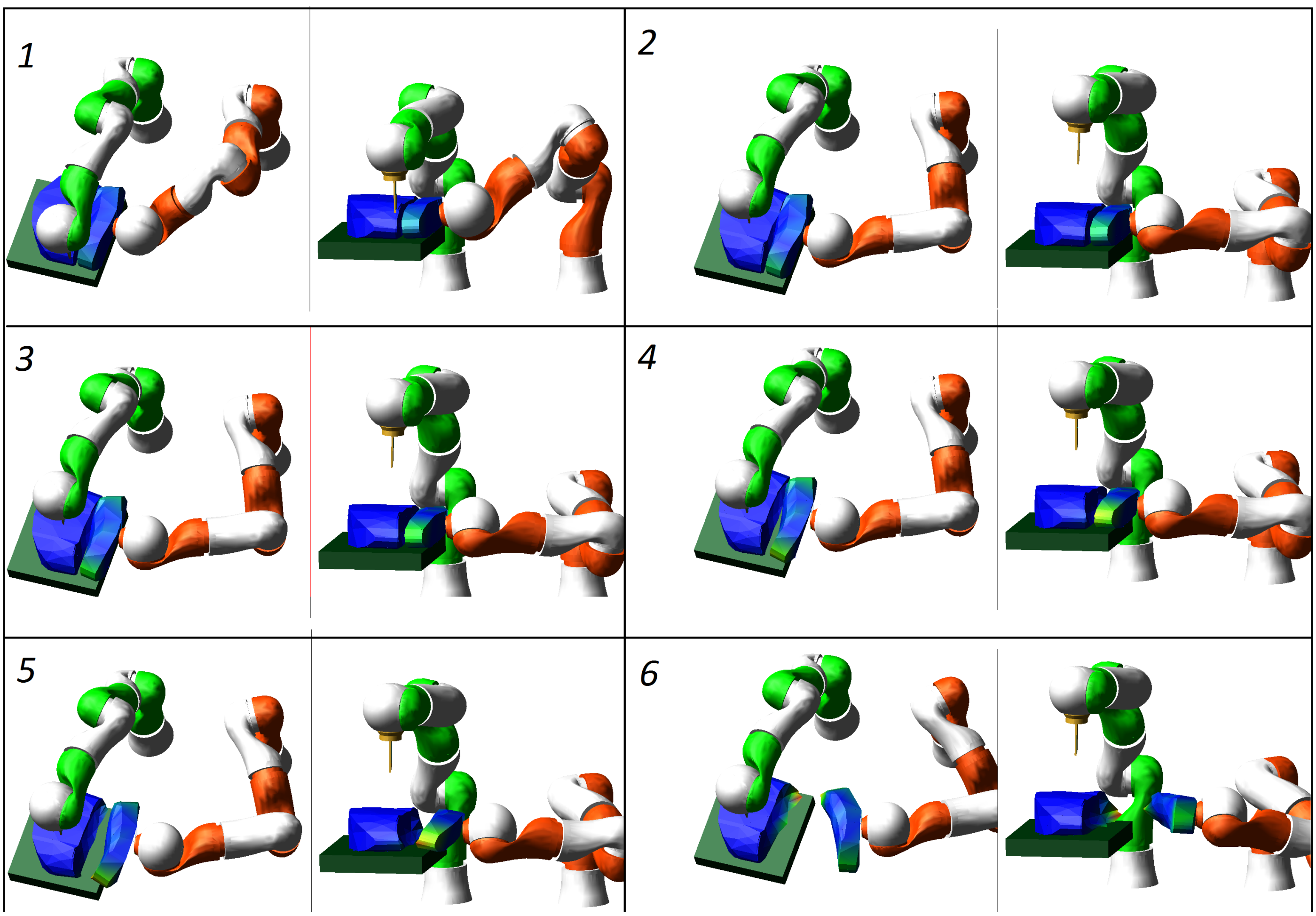

Fig. 5: Snapshot of separation process

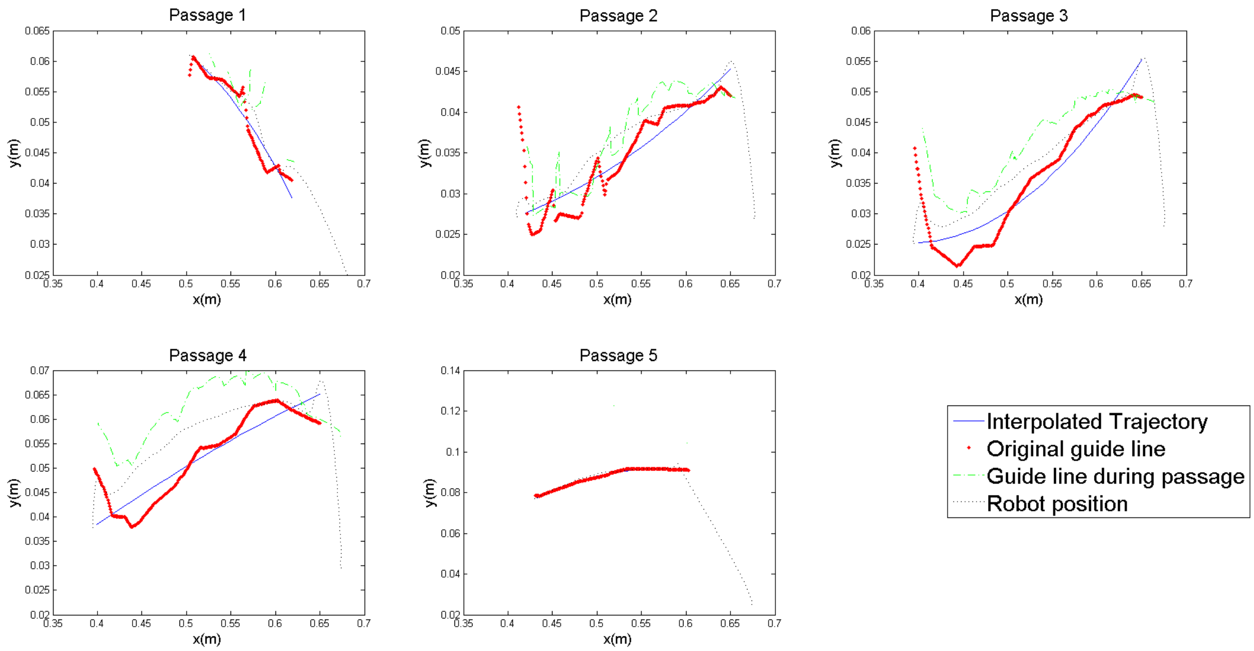

Fig. 6: Robot Trajectory modified by local updates

shown at the end of a cutting passage. It can be seen that after each cutting passage the meat muscles are pulled further apart. This is due to the impedance based force control scheme for the pulling robot. A desired force is given to the pulling robot, by maintaining this force the muscles are gently separated.

It should be noted that after each passage the state of the surface as changed dramatically, therefore the system must be updated by the visual primitive. In fact, even during the passage the cutting trajectory must be locally updated to ensure that the knife does not interact with the beef muscle. Fig.6 shows the resulting trajectory when the local update is taken into account for each passage. 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 interpolated 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. 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.

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. Working with deformable objects is a both challenging and increasingly important topic. There are two principle approaches to dealing with this problem. Firstly advanced modeling techniques that can predict the evolution of the state of the object with respect to external wrenches. Secondly increasing sensory capabilities, in this case vision, to adapt to changes in the object state. Due to the complex interaction between the tool and the object and the inherent variability of the meat muscles the latter is a more realistic choice.

Future work will concentrate on the experimental validation of the proposed control schemes in an industrial environment. An efficient redundancy management scheme should be created that shares the non-essential DOF throughout the system to optimize secondary criteria.

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.

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] P. G. Dempsey and R. W. McGorry, "Investigation of a pork shoulder deboning operation," *Journal of Occupational and Environmental Hygiene*, vol. 1, no. 3, pp. 167–172, 2004.

[3] G. Purnell, "Robotic equipment in the meat industry," *Meat science*, vol. 49, pp. S297–S307, 1998.

[4] 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.

[5] L. Hinrichsen, "Manufacturing technology in the danish pig slaughter industry," *Meat science*, vol. 84, no. 2, pp. 271–275, 2010.

[6] Y. Sugiyama, A. Ito, and S. Manmoto, "Intelligent cutter for pork deboning robot (automatic processing of complete pre-deboning process of pork arm)," *Nihon Kikai Gakkai Nenji Taikai Koen Ronbunshu*, vol. 5, pp. 27–28, 2005.

[7] G. Guire, L. Sabourin, G. Gogu, and E. Lemoine, "Robotic cell with redundant architecture and force control: application to cutting and boning," in *Robotics in Alpe-Adria-Danube Region (RAAD), 2010 IEEE 19th International Workshop on*. IEEE, 2010, pp. 99–104.

[8] 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.

[9] 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.

[10] S. Frisken-Gibson, "Using linked volumes to model object collisions, deformation, cutting, carving, and joining," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 5, no. 4, pp. 333–348, 1999.

[11] M. Mahvash and V. Hayward, "Haptic rendering of cutting: A fracture mechanics approach," *Haptics-e*, vol. 2, no. 3, pp. 1–12, 2001.

[12] 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.

[13] 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.

[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 and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine,*, vol. 13, no. 4, 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] P. Long, W. Khalil, and P. Martinet, "Modeling and control of a meat-cutting robotic cell," in *2013 15th International Conference on Advanced Robotics (ICAR)*. IEEE, 2011, pp. 61–66.

[18] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald *et al.*, "The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*. VDE, 2010, pp. 1–8.

[19] W. Khalil and J. Kleinfinger, "A new geometric notation for open and closed-loop robots," in *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol. 3. IEEE, 1986, pp. 1174–1179.

[20] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 1, pp. 43–53, 1987.

[21] N. Hogan, "Impedance control: An approach to manipulation: Part illapplications," *Journal of dynamic systems, measurement, and control*, vol. 107, no. 2, p. 17, 1985.

[22] Y. Mezouar, M. Prats, and P. Martinet, "External hybrid vision/force control," in *Intl. Conference on Advanced Robotics (ICAR07)*, 2007.