

Contents

1. Collaborative Robotics for Deformable Object Manipulation with Use Cases from Food Processing Industry	1
1.1 Introduction	1
1.2 Manipulation of Deformable Objects	2
1.3 Design of Robotic Meat Cutting Cell	3
1.3.1 Overview	3
1.3.2 Simulation of Robotic Meat Cutting Cell	5
1.3.3 Experimental Validation of robotic meat cutting cell	9
1.4 Conclusions	17
<i>Bibliography</i>	19

Chapter 1

Collaborative Robotics for Deformable Object Manipulation with Use Cases from Food Processing Industry

COPY PASTE, needs to be modified

1.1 Introduction

The food industry in the U.S accounts for an estimated 12% of the total amount workers employed in manufacturing and 14% of the total value of shipments and receipts for services [USC (2017)]. Moreover, according to the United Nations, food production will have to grow by 50% by 2050 in order to cope with the expected population increase [FAO (2017)]. Meanwhile, while much of the packaging has been robotized the bulk of the processing tasks are carried out using manual labor where wages are roughly 80% national average in the U.S [USC (2017)]. Hence, there is an enormous potential for robotization in the food industry, however to do so several challenges must be overcome.

One of the principal differences between food industry and classical manufacturing is that the target objects are typically deformable [Masey *et al.* (2010)], meaning it is necessary to adapt classical methods for both perception [Petit *et al.* (2017)] and manipulation [Navarro-Alarcon *et al.* (2016)]. In this chapter, recent advances in deformable object manipulation for food manufacturing industry are discussed.

For instance, in France meat processing accounts for over 25% of the food industry's total employees and includes over 2,000 companies. The robotization of meat cutting tasks is of increasing importance for several reasons. The unsocial working hours along with the strenuous, uncomfortable working conditions have created a shortage of skilled labor at a time when competition from low cost labor regions, notably from the MERCOSUR countries, is growing. Furthermore the physical tasks involved in the work lead to a high rate of musculoskeletal injuries [?]. The robotization of the meat processing industry has been the focus of several works worldwide. A general overview of the role of robots in the meat processing industry is outlined in [??]. 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 [?]. In ??, a specific robotic meat cutting cell is analyzed from the point of view of the cutting parameters, while

using bones as a positional guide. In Japan, robots have been widely introduced in poultry cutting operations [?]. The previous works deal with highly repeatable scenarios in controlled environments, often aiming to optimize a well known existing process.

1.2 Manipulation of Deformable Objects

Manipulation tasks concerning deformable objects can be divided into two sub-categories. Firstly, where the deformation is seen as a supplementary challenge of the manipulation task and the control is focused on eliminating or mitigating its effects [Long *et al.* (2014a, 2015)]. Secondly, where the deformability is part of the task itself, sometimes referred to as shape control [Das and Sarkar (2011)]. In this case the objective is to control the internal object configuration variables such that they converge to a desired state.

In both cases, approaches vary from model based approaches to sensors based approaches. For model based controllers an precise representation of the system deformation is required. Using this model to predict deformations the robot's controller can be adapted to ensure the minimization of deformations.

The precise control of deformable objects is an important subject, not only in the aforementioned food industry but also in the medical sectors [?]. There exists two classes of approaches to deal with the object's supplementary degree's of freedom model based control and sensor based control. For model based controllers [Long *et al.* (2014a, 2015)] an accurate an exact model of the system deformation is required. Using this model to predict deformations the robot's controller can be adapted to ensure the minimization of deformations. Typically for these applications, the object deforms in a large non-linear fashion in response to external forces. Moreover, the variability of the target object means that manipulation solutions need to use sensor rather than model based control. The accurate control of flexible objects by robots is an interesting subject due to its industrial importance, notably in the medical [1] and food processing [2] sectors. The proposed solutions dealing with object flexibility are into two classes, sensor based [3], [4] and model based [5], [6]. In this work, we focus on model based solutions, where for closed chain cooperative robots research has been typically approached from an object oriented viewpoint. Cooperative robots can be considered as redundantly actuated systems since all the degrees of freedom of each manipulator are motorized. Thus the system can achieve secondary goals, such as load distribution [7], backlash elimination [8] and optimization of independent actuators [9]. In addition to this, the objects internal state can be controlled. Depending on the object type and the grasp structure, the internal variables may represent the objects internal loading [10], shape [11], [12] or vibration [5]. Suppose there exists a number of fixed frames on an object. The object can be classified according to the relative behavior between these frames [Long *et al.* (2015)]. Three possible types of object behavior can be defined:

Rigid: There is no relative motion between the frames, in spite of the object motion or external forces.

Deformable: There is relative motion between frames where the motion of one point can be obtained from the motion of the second point using information about the object's properties.

Articulated: The frames can move freely with respect to each other, where the velocity of a frame has no effect on the velocity of a different frame.

1.3 Design of Robotic Meat Cutting Cell

1.3.1 Overview

In this section, we describe recent work into the separation of beef meat muscles within the framework of the ARMS project¹. The proposed robotic cell consists of a multi-arm system equipped with an array of exteroceptive sensors, notably force and vision sensors. The proposed control approach is first presented in a dynamic simulation environment before validation on an experimental cell.

1.3.1.1 Modeling Separation of Deformable Objects

In the food industry, manipulated objects must be frequently cut, sliced and separated to create different portion sizes, remove waste matter and defects. To do so, the cutting instrument must follow a deformable contour, apply a sufficient force while ensuring that global deformation or *rupture* is avoided. Deformable object cutting presents a number of supplementary challenges with respect to classical contour following tasks. Firstly, in order to cut or sever the object the tool must necessarily pass through the target contour. Secondly, the cutting force opposes the direction of motion, whereas in the majority of contour following tasks, it is orthogonal to the motion. Finally, the target object is deformable and typically heterogeneous, thus both the contour shape and required cutting force is variable throughout the task.

One method to approach this problem is to utilize an advanced object model whose properties are updated as the cutting progress. This problem has been typically studied with respect to surgical applications [Misra *et al.* (2008)], where the objective is to mimic deformable body behavior, thus the cutting force, if considered, is used as a haptic output rather than robot input. On the other hand, the object model can be used to obtain the deformation in response to the tool interaction. As expected, the accuracy of these models is largely dependent on computation time. The most accurate models are numerical models such as FEM (Finite element methods) [Mendoza and Laugier (2003)] and BEM (Boundary element methods) [Meier *et al.* (2005)]. These models have the added benefit of being able to handle extremely complex geometries relatively easy . However the computation time is a

¹<http://www.agence-nationale-recherche.fr/Project-ANR-10-SEGI-0008>

significant drawback as is difficultly re-meshing to account for cutting. Discrete models for instance mass spring damper systems have a much low complexity and thus have a low computation low. They have been used successfully in the food separation tasks [Nabil *et al.* (2015)], however often these models suffer from a pure physical likeness, leading to tiresome parameter tuning [Delingette *et al.* (1999)]. If, as is often the case, the desired cutting volume is known, the system can be modeled as a set of deformable objects attached together. These methods are known as *regional models* [Delingette *et al.* (1999)] and combine realistic behavior with a real time computation speeds [Vigneron *et al.* (2004)]. Finally a promising mesh-less approaches based on displacing nodal coordinates is proposed in [Jin *et al.* (2014)].

1.3.1.2 Robotic Cutting: Slice & Press

Historically, robots have been successfully applied in tasks that require the separation of rigid objects such as milling or bone cutting. A desired cutting rate or *feed* is defined for a given object based on material properties or empirical evidence. For deformable objects, such an approach is impractical as the properties can vary with different fixations, ambient conditions and in particular with structural changes during cutting. To avoid global deformations and rupture, it is desirable to minimize cutting forces. One method of achieving this is to apply a shearing force.

Intuitively, it is obvious that the cutting force can be reduced when shearing or slicing is applied. This means that instead of a force purely orthogonal to the cutting surface, a force parallel to the surface is simultaneously applied. This is known as the *pressing* and *slicing* approach [Atkins *et al.* (2004)]. There are several ways to explain the reduction in cutting forces, for instance by studying the stress concentration at the contact area [Zhou *et al.* (2006); Reyssat *et al.* (2012)] or by treating the cutting action as a crack propagation problem [Mahvash and Hayward (2001)], or as in the following using an energy balance formulation [Atkins *et al.* (2004)].

As shown in Fig.1.1, the shearing force can be either when *slicing* velocity is added to the tool velocity or alternatively the tool is positioned at an angle to the material [Arnold *et al.* (2009)]. To cut the material the robot controlled knife must along the x -axis of the tool. A movement of distance of Δx_t means the robot must overcome a resistive force, denoted as ${}^t f_x$, meaning the work done by the cutting tool is written as:

$${}^t f_x \Delta x_t = K_c w \Delta x_t \quad (1.1)$$

where K_c is the material's fracture toughness while w is the width of the blade in contact with the material. If a shearing motion is added, the work required to propagate the cut is now a product of the work done in both the *pressing* and *slicing* directions:

$${}^t f_x \Delta x_t + {}^t f_z \Delta z_t = K_c w \Delta x_t \quad (1.2)$$

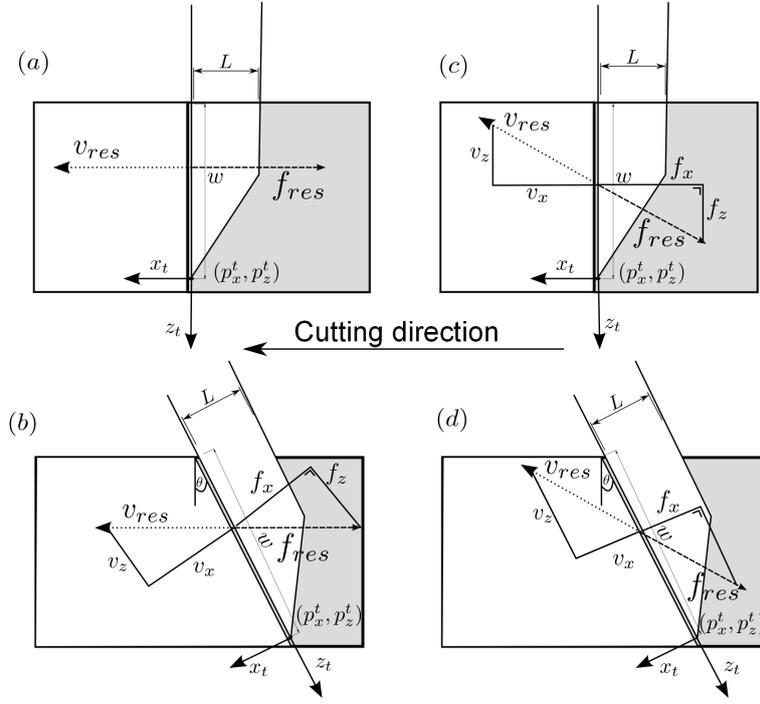


Fig. 1.1 Cutting Cases: (a) Cutting angle zero pure pressing, (b) Cutting angle θ pure pressing, (c) Cutting angle zero, pressing and slicing, (d) Cutting angle θ , pressing and slicing

The resultant tool force and displacement are given respectively as ${}^t f_r = \sqrt{({}^t f_x^2 + {}^t f_z^2)}$ and $\Delta p_t = \sqrt{(\Delta x_t^2 + \Delta z_t^2)}$. Therefore assuming the resultant forces are used purely to cut the material, the energy balance can also be written as:

$${}^t f_r \Delta p_t = K_c w \Delta x_t \quad (1.3)$$

$$\xi = \frac{\Delta z_t}{\Delta x_t} \quad (1.4)$$

By introducing the *slice/push* ratio, given in (1.4), into (1.2) and (1.3), the authors derive the following relation:

$$\frac{{}^t f_r}{K_c w} = \sqrt{\frac{1}{1 + \xi^2}} \quad (1.5)$$

From (1.5), it can be seen that an increase in ξ reduces the resultant forces provided K_c is constant.

1.3.2 Simulation of Robotic Meat Cutting Cell

The *kuka lwr* is the robotic platform for both simulation and experimental validation, while MSC Adams is used simulation environment as shown in Fig. 1.2 [Long

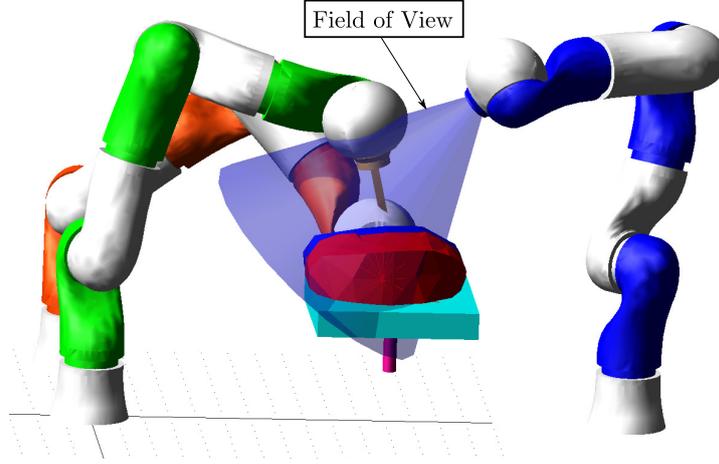


Fig. 1.2 Simulated robotic cell using MSC Adams. The system is composed of three Kuka LWR robots, a cutting robot (green), a pulling robot (orange) and a vision robot (blue).

et al. (2013)]. Using the angle axis representation for orientation, the Cartesian position, kinematic screw and acceleration of the i^{th} robot is given as

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{p}_i \\ \mathbf{u}\psi_i \end{bmatrix}, \quad \mathbf{V}_i = \mathbf{J}_i \dot{\mathbf{q}}_i, \quad \dot{\mathbf{V}}_i = \mathbf{J}_i \ddot{\mathbf{q}}_i + \dot{\mathbf{J}}_i \dot{\mathbf{q}}_i. \quad (1.6)$$

The dynamic model of each robot can be written as:

$$\boldsymbol{\tau}_i = \mathbf{A}_i \ddot{\mathbf{q}}_i + \mathbf{c}_i + \mathbf{J}_i^T \mathbf{h}_i, \quad (1.7)$$

where \mathbf{V}_i is the twist, \mathbf{J}_i is the kinematic Jacobian matrix and \mathbf{q}_i the vector of joint coordinates while $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are the velocities and accelerations respectively. The inertial parameters are taken from the equivalent CAD model. The inertia matrix and the matrix of centrifugal, Coriolis and gravity torques are denoted as \mathbf{A}_i , and \mathbf{c}_i . The Cartesian wrench is denoted as \mathbf{h}_i while $\boldsymbol{\tau}_i$ is the joint torque.

The deformable object is modeled using the *regional models* approach, i.e the cutting is restricted to an a prior defined region. Hence two distinct types of deformable model are generated, a model that represent the meat muscles and a second model that represents the aponeurosis, similar to tendons that act as links between muscles.

The meat muscles are modeled using a finite element model approach. Firstly a visual scan of a generic beef round is obtained after separation and converted into a 3D-geometry. The two muscles are reconstructed and the exact cutting surface is extracted. The muscles are simplified, reducing the computational cost, however the exact cutting surface is used. The models are discretized volumetrically and nodes are generated on the cutting surface of each muscle. For each node on the cutting surface of one muscle, there exists a corresponding attachment point on the other muscle, which is coincident at the beginning of the simulation. These nodes

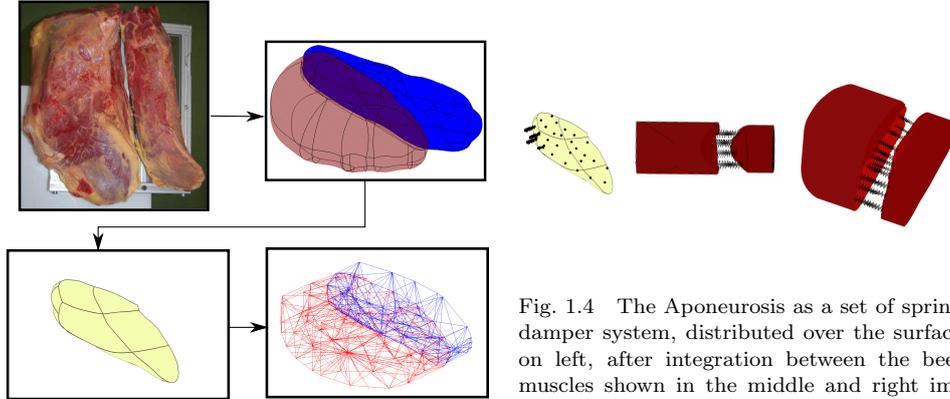


Fig. 1.3 From 3D scan to Finite element mesh, for clarity the different muscles are shown in blue and red.

Fig. 1.4 The Aponeurosis as a set of spring damper system, distributed over the surface on left, after integration between the beef muscles shown in the middle and right images.

can transmit forces from one muscle to the other. A modal analysis is performed for each muscle and the resulting system is imported into the simulation environment. This file contains the object geometry, the orthonormalization of the Craig-Bampton modes and the generalized mass and stiffness for the mode shapes.

The aponeurosis are modeled as the second deformable object located in an intermediate layer in the beef shoulder. The aponeurosis can store elastic energy, then recoil when unloaded, thus it is appropriate to model them as spring damper systems which are then fixed to partner nodes on the cutting surfaces. The simulation commences at the equilibrium condition i.e., when the muscles are perfectly mated and net spring force is zero.

During the simulation experiments if the line segment representing the knife intersects the line segment representing the aponeurosis, the corresponding spring damper system is deactivated. Thus the link is severed between the nodal attachment points.

1.3.2.1 Global controller

A global decentralized control scheme is employed to separate the muscles as shown in Fig. 1.5. The cutting robot follows an off-line generated trajectory which is modified on-line with the output of vision system to compensate for the deformation of the meat muscles. To track the desired variables, a computed torque is used, therefore the desired Cartesian acceleration, \mathbf{w}_c , is defined as:

$$\mathbf{w}_c = \dot{\mathbf{V}}^d + \mathbf{K}_d (\Delta \mathbf{V}) + \mathbf{K}_p (\Delta \mathbf{x}) - \dot{\mathbf{J}}_c \dot{\mathbf{q}} \quad (1.8)$$

where \mathbf{K}_d \mathbf{K}_p are positive gains. \mathbf{w}_c is then transformed to the joint space, and a new desired acceleration is defined:

$$\ddot{\mathbf{q}}^d = \mathbf{J}_c^+ (\mathbf{w}_c + \mathbf{P}_c \mathbf{z}) \quad (1.9)$$

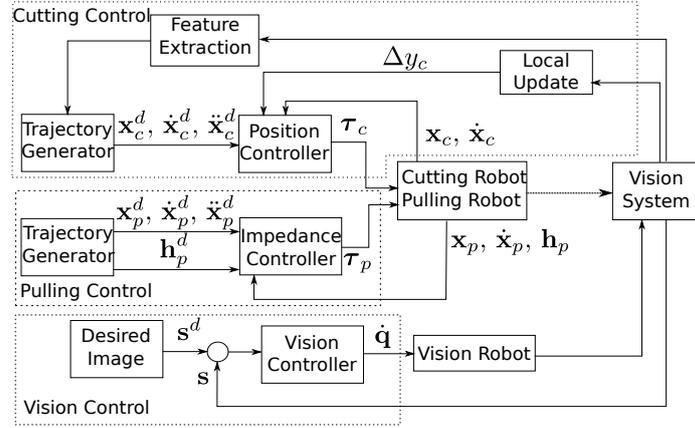


Fig. 1.5 Global Control Scheme

\mathbf{z} is a secondary criterion used to shift the solution away from joint limits, and \mathbf{P} projects \mathbf{z} into the null space of the primary solution. Finally a joint torque realizing this acceleration is obtained as

$$\tau_c = \mathbf{A}_c \ddot{\mathbf{q}}^d + \mathbf{c}_c + \mathbf{J}_i^T \mathbf{h}_c. \quad (1.10)$$

The on-line update is computed by using y_g , the exact position of the guide line extracted from the visual primitive. y^d is updated as:

$$y^{*d}(t) = y^d(t) + \Delta y \quad (1.11)$$

$$\Delta y = y_g - y_c \quad (1.12)$$

Equation (1.8) is modified to create an impedance relationship [Hogan (1985)] between the desired position and empirically learned pulling that is necessary to aid the separation and allow greater access for the vision system.

$$\mathbf{w}_p = \dot{\mathbf{V}} + \lambda (\mathbf{K}_d (\Delta \mathbf{V}) + \mathbf{K}_p (\Delta \mathbf{x}) - \mathbf{K}_f (\Delta \mathbf{h})) - \dot{\mathbf{J}}_p \dot{\mathbf{q}}, \quad (1.13)$$

where λ is the inverse of the desired inertial behavior.

Finally, the global controller guides the vision robot to maintain the cutting surfaces within its field of view. To do so, the robot is controlled in image space, by minimizing the error between a desired image denoted by the feature vector \mathbf{s}_d and the current camera image \mathbf{s}_{im}

$$\dot{\mathbf{q}}_v^d = -\mathbf{K}_p (\mathbf{L}_s \mathbf{J}_v)^+ (\mathbf{s}_d - \mathbf{s}_{im}), \quad (1.14)$$

where \mathbf{L}_s is known as the interaction matrix [Chaumette and Hutchinson (2006)].

1.3.2.2 Simulation 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

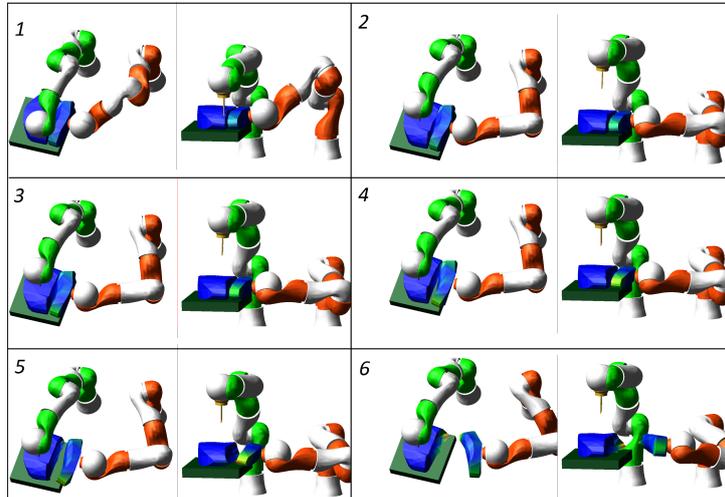


Fig. 1.6 Snapshot of separation process

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 repeatably cutting along the surface of separation with the knife. Aan overview of the resulting behavior can be seen in Fig.1.6. This image is split into six panes. Each pane gives two separate views of the simulator. By examining the image, the evolution of the system can be seen as the cutting progresses.

Fig.1.7 and Fig.1.8 show the results of the cutting task with and without the local vision update. In both cases, large changes in the cutting profile (guide line) are noted as the aponeurosis are severed due to the applied pulling forces. The local vision system compensates for these change by simultaneously applied a corrective translational and angular velocity.

1.3.3 *Experimental Validation of robotic meat cutting cell*

The proposed meat cutting cell is transferred to an experimental setup [Long *et al.* (2014b)]. There are several supplementary difficulties for the experimental validation absent in the simulations. Firstly, the vision system can no longer be considered as ideal and in order to get a precise view of the cutting zone it is more expedient to replace the cutting robot by an *eye-in-hand* system that focuses on local deformation. Secondly, the force control must be modified to consider the resistive force during the cut, to do so the force controller is coupled with the vision controller. In the following the proposed force controller is examined in detail.

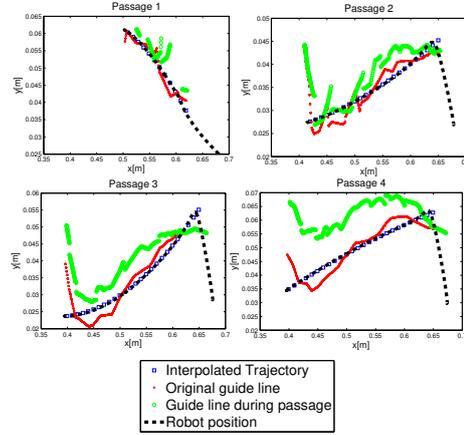


Fig. 1.7 Robot Trajectory for each passage, using off-line trajectory planner based on model data.

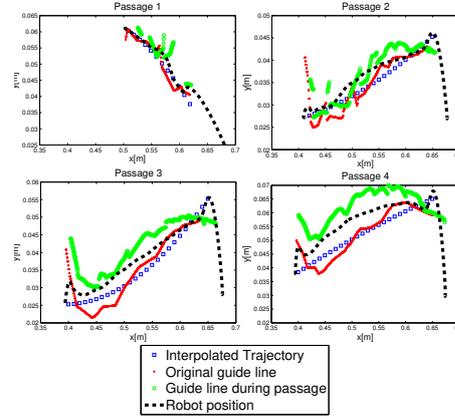


Fig. 1.8 Robot Trajectory for each passage, using off-line model-based trajectory modified by visual feedback.

1.3.3.1 Novel Cutting Force Controller

When cutting deformable objects it is desirable to minimize the required cutting force at the tool frame for two reasons. Firstly if a large cutting force can induce global deformation rather than rupture, leading to the clustering of material around the cutting tool. This global deformation reduces the product quality. Secondly a smaller cutting force reduces the energy input of the system whereas a larger cutting force may be outside the capabilities of the tool. Equation (1.15) is used to describe the cutting process, where W_r is the work done by the cutting tool, defined as the sum of W_c , the energy required to cut the material; W_f , the work done in overcoming the frictional effects on the blade and U , the strain energy due to global deformation of the soft material.

$$W_r = W_c + W_f + U \quad (1.15)$$

During a pure cutting motion, it is assumed that the global deformation caused by the cutting tool is negligible, $U = 0$, therefore $W_r = W_c + W_f$. From, examination of (1.15) we propose to modify the ratio ξ , from (1.5) in response to the presence of resistive forces. ξ can be increased by changing the cutting angle or by generating a velocity parallel to the cutting surface. It is undesirable to increase the cutting angle during the trajectory due to both the practical difficulties and the reduction in material feed. Therefore the *slicing* velocity is linked to the resistive cutting force by an impedance controller.

As shown in Fig. 1.1, the z axis is defined as parallel to the cutting surface. In this section, a series of experiments are described that demonstrate the feasibility of this force controller in a simple cutting scenario. For each experiment the robot followed a straight line cutting trajectory with a constant velocity. This trajectory

is defined by a linear interpolation from point to point. The control law is given as:

$${}^t\mathbf{V}_t = {}^t\mathbf{S}_{ob} \left(\mathbf{k}_p d\mathbf{X} + \mathbf{k}_v \mathbf{V}^d \right) + \begin{bmatrix} 0 \\ 0 \\ k_z^t f_c \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (1.16)$$

$$\dot{\mathbf{q}} = {}^t\mathbf{J}^+ {}^t\mathbf{V}_t \quad (1.17)$$

where $d\mathbf{X}$ is the position error in the object frame and \mathbf{V}^d is the desired cutting velocity. When the knife exits the media, due to the slicing effect of the controller, the robot returns to the initial position to restart the passage. The behavior of the force controller is investigated with respect to changes in the cutting angle, θ as shown in Fig.1.1, and the gain k_z^t .

In total twelve experiments are carried out. The test matrix and the quality of the cut for each test is shown in Table 1.1. The quality of the cut, which depends on the level of global deformation and rupture in the object, was decided by visual inspection. An example of three cases is shown in Fig.1.12. These cases are described as:

- **Good:** No global deformation, an extremely clean cut
- **Medium:** Slight global deformation, in the cutting region
- **Poor:** Large global deformation and permanent damage to surrounding area

	$k_z = 0.0$	$k_z = 0.001$	$k_z = 0.005$	$k_z = 0.01$
$\theta = \frac{\pi}{12}$	Poor*	Medium	Good:	Good
$\theta = \frac{\pi}{6}$	Poor	Medium	Good	Good:
$\theta = \frac{\pi}{4}$	Poor	Good	Good	Good:

The table shows that as expected, the quality of the cut can be increased either by changing the cutting angle or by increasing the force gain. It should be noted that for the experiment $k_z = 0.0$, $\theta = \frac{\pi}{12}$, the knife deformed the object without any cutting. This resulted in a constant increase in force until the experiment was stopped, to prevent damage to the robot and the tool. The increase in force can be seen in Fig.1.9.

The graphical results for $\theta = \frac{\pi}{12}$, $\theta = \frac{\pi}{6}$ and $\theta = \frac{\pi}{4}$ are shown in Fig.1.9, Fig.1.10 and Fig.1.11, respectively. Each figure consists of six sub-figures arranged in two rows and three columns. The top row shows the cutting forces as the cutting distance is increased. The bottom row shows the corresponding cutting depth as

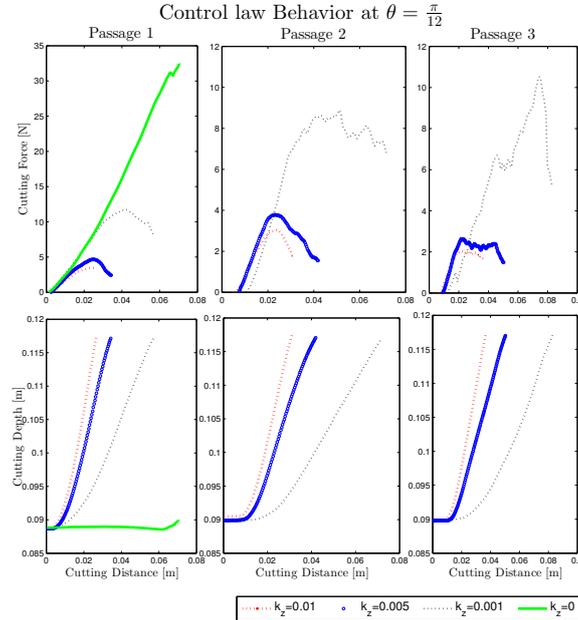


Fig. 1.9 Cutting Force versus displacement with $\theta = \frac{\pi}{12}$. The cutting forces versus cutting distance are shown on the top row. The cutting depth versus cutting distance are shown on the bottom row. Each column represents a passage.

the cutting distance is increased. Each of the three columns shows the results of a particular passages. Although in the case of $k_z = 0.0$, the robot completes is only one passage since there is no slicing action.

For each cutting angle, it can be seen that by increasing the value of k_z , the resulting resistive force is reduced. Furthermore for each value of $k_z > 0$, the results show a decrease in the cutting forces as the controller begins the slicing phase. This generates a *n*-shaped for the force response and thus shows the effectiveness of the proposed controller. In contrast, the position controller $k_z = 0.0$, results not only in a poor quality, as shown in Table 1.1, but also high forces on the cutting tool reaching up to 32 Newtons in Fig.1.9.

However, a drawback of increasing the force gain is the reduction in cutting distance. For example Fig.1.11, the control law with $k_z = 0.001$ has cut a distance of over 200mm at the end of the third passage whereas $k_z = 0.005$ has cut less than half this distance.

By increasing the cutting angle, the force on the blade is decreased for all tests. This is expected since the cutting angle also increases the *slice/press* ratio. For this set of experiments, the cutting depth was constant, however in practice by increasing the cutting angle, the possible cutting depth and therefore cutting feed is reduced.

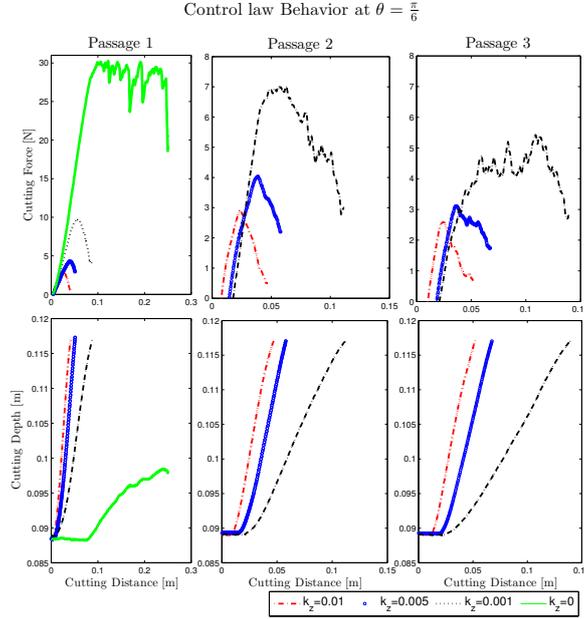


Fig. 1.10 The cutting forces versus cutting distance are shown on the top row. The cutting depth versus cutting distance are shown on the bottom row. Each column represents a passage.

1.3.3.2 Experimental Validation of Force/Vision Controller

The proposed force controller is coupled with a visual feedback. The cutting robot follows a polynomial curve, \mathcal{C} defined in the object frame \mathcal{R}_{ob} . At any instant the knife's desired location, defined by the 4×4 homogenous transformation matrix in the object frame is given as ${}^{ob}\mathbf{T}_t^d = {}^{ob}\mathbf{T}_c {}^c\mathbf{T}_t(\theta)$, where ${}^c\mathbf{T}_t(\theta)$ is used to make the trajectory consistent with the cutting angle θ . The position of ${}^{ob}\mathbf{T}_c$, denoted ${}^{ob}\mathbf{p}_c$, is defined as:

$$\mathbf{p} = [p_x^d \ p_y^d \ p_z^d]^T \quad (1.18)$$

The orientation of ${}^{ob}\mathbf{T}_c$, denoted ${}^{ob}\mathbf{R}_c$, is defined as:

$${}^{ob}\mathbf{R}_c = [\mathbf{t} \ \mathbf{n} \ \mathbf{a}]$$

$$\mathbf{t} = \left[\frac{1}{\sqrt{\left(1 + \frac{\partial y^2}{\partial x^2}\right)}}, \frac{\frac{\partial y}{\partial x}}{\sqrt{\left(1 + \frac{\partial y^2}{\partial x^2}\right)}}, 0 \right]^T \quad (1.19)$$

$$\mathbf{n} = \left[\frac{-\frac{\partial y}{\partial x}}{\sqrt{\left(1 + \frac{\partial y^2}{\partial x^2}\right)}}, \frac{1}{\sqrt{\left(1 + \frac{\partial y^2}{\partial x^2}\right)}}, 0 \right]^T \quad (1.20)$$

$$\mathbf{a} = [0, 0, -1]^T, \quad (1.21)$$

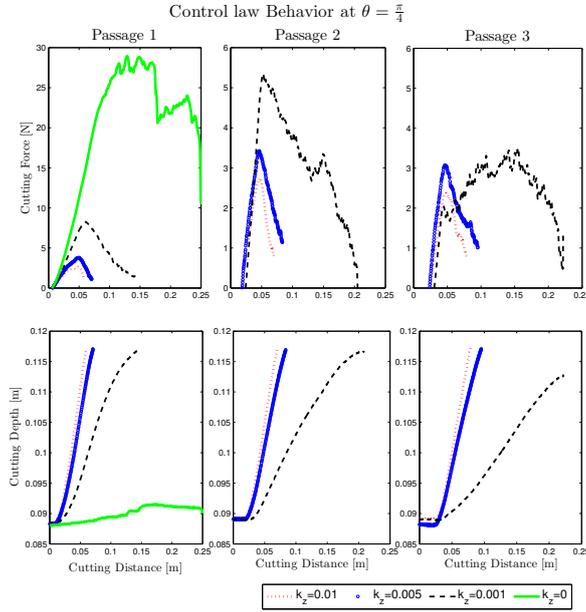


Fig. 1.11 The cutting forces versus cutting distance are shown on the top row. The cutting depth versus cutting distance are shown on the bottom row. Each column represents a passage.

where \mathbf{t} is the desired cutting direction, which is tangential to \mathcal{C} . \mathbf{a} is the axis normal to the object's surface while \mathbf{n} is the remaining orthogonal axis of the frame. $\frac{\partial y}{\partial x}$ is the value of $\frac{\partial y}{\partial x}$ evaluated at p_x^d .

The vision controller updates the ${}^{ob}\mathbf{T}_t^d$ in response to on-line deformations by creating a deviation, denoted as ${}^{ob}d\mathbf{X}_t^v$. The vision system extracts the image coordinates of (u_i, v_i) , (u_j, v_j) and (u_k, v_k) , a series of points ahead of the image projection of tool point. The normalized position of a point i is reconstructed using the intrinsic camera parameters, \mathbf{C} , which relate the image coordinates to the coordinates in the perspective plane:

$$\begin{bmatrix} p_{xi}^v \\ p_{zi}^v \\ p_{yi}^v \\ p_{zi}^v \\ 1 \end{bmatrix} = \mathbf{C} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \quad (1.22)$$

The depth of a point, p_{zi}^v , is estimated using the material height and the tool position, the depth estimation allows the reconstruction of the 3D position of the point. Since the camera gives a local view of the trajectory, the curvature within this window is quite small and can be approximated by a straight line. By fitting this line to the Cartesian position of points i , j and k the vectors \mathbf{t} and then \mathbf{n} are obtained.

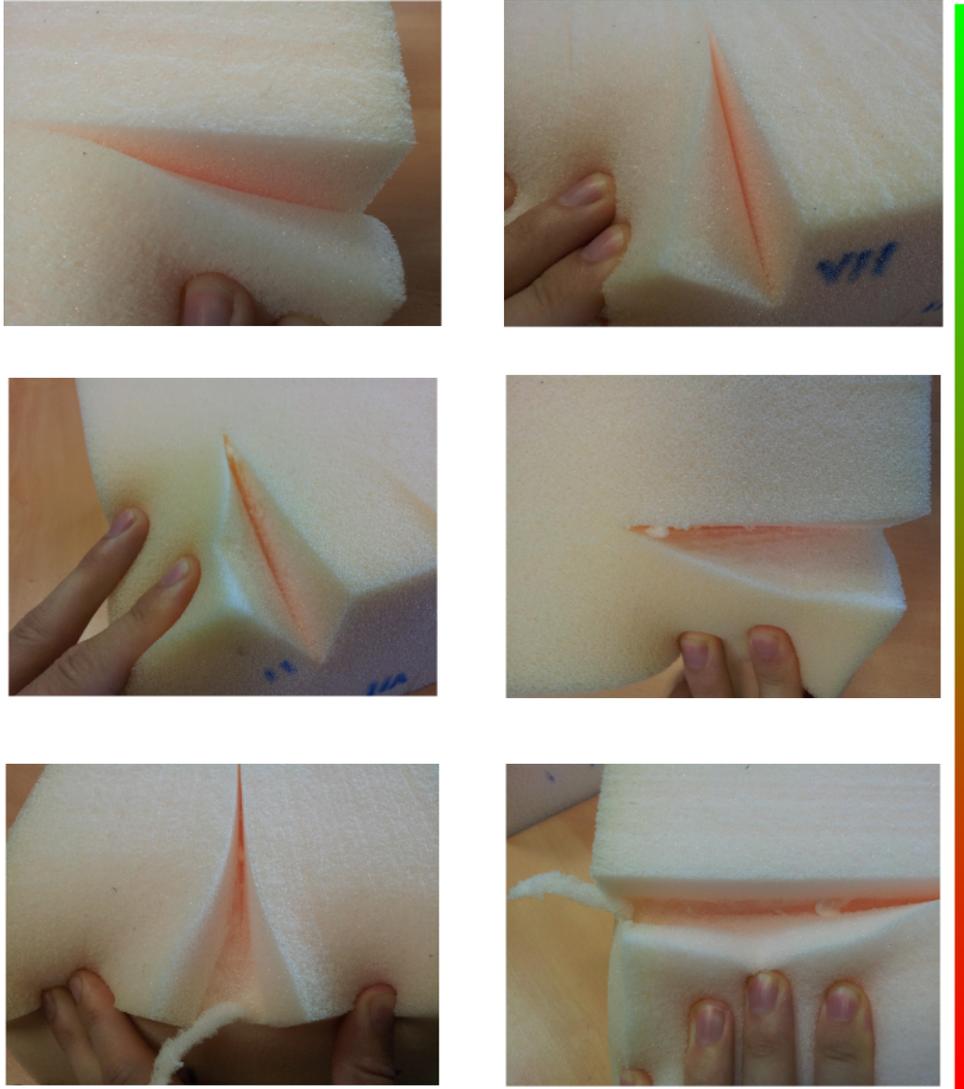


Fig. 1.12 Comparison of Cut Quality for the proposed force controller, (Top) A good quality cut with no global deformations, where the force gain $k_z = 0.01$ and the cutting angle $\theta = \frac{\pi}{4}$, (Middle) A medium quality with some small deformations where the force gain $k_z = 0.001$ and the cutting angle $\theta = \frac{\pi}{12}$, (Bottom) Poor quality with large global deformations where the force gain $k_z = 0.0$ and the cutting angle $\theta = \frac{\pi}{4}$.

In order to generate an error vector, the curve \mathcal{C} is evaluated at p_{xi} allowing a desired matrix ${}^{ob}\mathbf{T}_i^d$ to be obtained. This in turn is used to calculate the vision generated deviation which acts in one translational direction and three rotational

directions:

$$\Delta p_{yi} = p_{yi}^d - p_{yi}^v \quad (1.23)$$

$$\Delta^{ob} \mathbf{R}_i = {}^{ob} \mathbf{R}_i^d ({}^{ob} \mathbf{R}_i^v)^T \quad (1.24)$$

*Collaborative Robotics for Deformable Object Manipulation with Use Cases from Food Processing Industry*17

1.4 Conclusions

Bibliography

- (2017). The 2016 annual survey of manufactures, annual survey of manufactures: General statistics: Statistics for industry groups and industries: 2016 and 2015, Tech. rep.
- (2017). Faostat statistics database. Tech. rep.
- Arnold, G., Leiteritz, L., Zahn, S., and Rohm, H. (2009). Ultrasonic cutting of cheese: Composition affects cutting work reduction and energy demand, *International Dairy Journal* **19**, 5, pp. 314–320.
- Atkins, A., Xu, X., and Jeronimidis, G. (2004). Cutting, by pressing and slicing, of thin floppy slices of materials illustrated by experiments on cheddar cheese and salami, *Journal of Materials Science* **39**, 8, pp. 2761–2766.
- Chaumette, F. and Hutchinson, S. (2006). Visual servo control. i. basic approaches, *IEEE Robotics & Automation Magazine* **13**, 4, pp. 82–90.
- Das, J. and Sarkar, N. (2011). Autonomous shape control of a deformable object by multiple manipulators, *Journal of Intelligent & Robotic Systems* **62**, 1, pp. 3–27.
- Delingette, H., Cotin, S., and Ayache, N. (1999). A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation, in *Computer Animation, 1999. Proceedings (IEEE)*, pp. 70–81.
- Hogan, N. (1985). Impedance control: An approach to manipulation: Part ii implementation, *Journal of dynamic systems, measurement, and control* **107**, 1, pp. 8–16.
- Jin, X., Joldes, G. R., Miller, K., Yang, K. H., and Wittek, A. (2014). Meshless algorithm for soft tissue cutting in surgical simulation, *Computer methods in biomechanics and biomedical engineering* **17**, 7, pp. 800–811.
- Long, P., Khalil, W., and Martinet, P. (2013). Modeling & control of a meat-cutting robotic cell, in *Advanced Robotics (ICAR), 2013 16th International Conference on (IEEE)*, pp. 1–6.
- Long, P., Khalil, W., and Martinet, P. (2014a). Dynamic modeling of parallel robots with flexible platforms, *Mechanism and Machine Theory* **81**, pp. 21–35.
- Long, P., Khalil, W., and Martinet, P. (2014b). Force/vision control for robotic cutting of soft materials, in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014) (IEEE)*, pp. 4716–4721.
- Long, P., Khalil, W., and Martinet, P. (2015). Dynamic modeling of cooperative robots holding flexible objects, in *Proceedings of the 2015 International Conference on Advanced Robotics (ICAR) (IEEE)*, pp. 182–187.
- Mahvash, M. and Hayward, V. (2001). Haptic rendering of cutting: A fracture mechanics approach, *Haptics-e* **2**, 3, pp. 1–12.
- Masey, R., Gray, J., Dodd, T., and Caldwell, D. (2010). Guidelines for the design of low-cost robots for the food industry, *Industrial Robot: An International Journal* **37**, 6,

pp. 509–517.

- Meier, U., López, O., Monserrat, C., Juan, M. C., and Alcaniz, M. (2005). Real-time deformable models for surgery simulation: a survey, *Computer methods and programs in biomedicine* **77**, 3, pp. 183–197.
- Mendoza, C. and Laugier, C. (2003). Simulating soft tissue cutting using finite element models, in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, Vol. 1 (IEEE), pp. 1109–1114.
- Misra, S., Ramesh, K., and Okamura, A. M. (2008). Modeling of tool-tissue interactions for computer-based surgical simulation: A literature review, *Presence: Teleoperators and Virtual Environments* **17**, 5, pp. 463–491.
- Nabil, E., Belhassen-Chedli, B., and Grigore, G. (2015). Soft material modeling for robotic task formulation and control in the muscle separation process, *Robotics and Computer-Integrated Manufacturing* **32**, pp. 37–53.
- Navarro-Alarcon, D., Yip, H. M., Wang, Z., Liu, Y.-H., Zhong, F., Zhang, T., and Li, P. (2016). Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model, *IEEE Transactions on Robotics* **32**, 2, pp. 429–441.
- Petit, A., Lippiello, V., Fontanelli, G. A., and Siciliano, B. (2017). Tracking elastic deformable objects with an rgb-d sensor for a pizza chef robot, *Robotics and Autonomous Systems* **88**, pp. 187–201.
- Reyssat, E., Tallinen, T., Le Merrer, M., and Mahadevan, L. (2012). Slicing softly with shear, *Physical review letters* **109**, 24, p. 244301.
- Vigneron, L. M., Verly, J. G., and Warfield, S. K. (2004). On extended finite element method (xfem) for modelling of organ deformations associated with surgical cuts, in *Medical Simulation* (Springer), pp. 134–143.
- Zhou, D., Claffee, M., Lee, K., and McMurray, G. (2006). Cutting, 'by pressing and slicing', applied to the robotic cut of bio-materials. ii. force during slicing and pressing cuts, in *Proceedings of 2006 Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE), pp. 2256–2261.