Model Based Position–Force–Vision Sensor Fusion for Robot Compliant Motion Control

Ruben Smits, Herman Bruyninckx, Wim Meeussen, Johan Baeten, Peter Slaets, Joris De Schutter
Department of Mechanical Engineering, Katholieke Universiteit Leuven,
Celestijnenlaan 300B, B-3001 Leuven, Belgium.
Email: Ruben.Smits@mech.kuleuven.be

***Abstract*— This paper shows how multi sensor fusion with position, force and vision sensors can help to improve robot control. It shows how Bayesian filtering helps in the fusion and how the extra information from fusing the sensors can be used in different control aspects. The paper gives an overview of increasingly more complex control tasks, whose realization depend on increasingly more "high level" sensor fusion.**

I. INTRODUCTION

This paper gives an overview of the realizations in multi sensor fusion (i.e. position, force, vision) for robot *compliant motion* control in the authors' research of the last decade. Compliant motion involves robot tasks that need *controlled* contact between the robot end effector and the environment. So, the focus of this paper is on *realtime* use of the sensor fusion results within the control loop, and the overview shows how the progress in on-line sensor fusion for position, force and vision has allowed to execute increasingly more complex tasks.

Commercial robot controllers typically use only the *proprioceptive* joint position sensors because they mainly focus on trajectory planning in joint and/or Cartesian space. This works very well if the environment of the robot tasks never changes and motion in contact isn't necessary. But by introducing *exteroceptive sensors* (force and vision) real *interactions* with the environment become possible. Force and vision are two very complementary and reasonably cheap sensors for intelligent robot control. Force sensors give *local* information about the (contact point with the) environment, which is crucial in compliant robot motion. Vision sensors give *global* information about the position of objects in the manipulation area of the robot.

The evolution in robot tasks is to allow more uncertainty about the environment. Therefor better sensing is needed; not just adding more sensors but also improving *sensor fusion* between several sensors. The authors' approach to sensor fusion is the Bayesian framework, basically Kalman filters and particle filters.

Section II explains how multi sensor fusion can be used in the control loop. Section III to V explain which sensor fusion algorithms have been applied to the cases of position–force, position–vision and position–force–vision fusion.

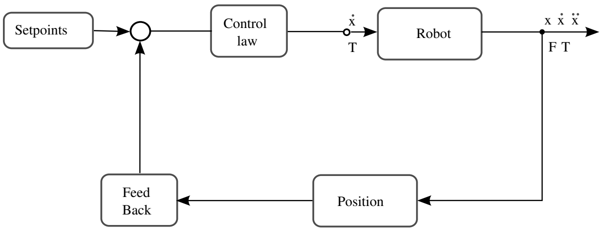

Fig. 1. Robot control with position feedback.

II. ROBOT CONTROL

Fig. 1 shows the basic *feedback* loop used in robot control: one measures some system parameter S of the robot (joint positions x, velocities $\dot{x}$, accelerations $\ddot{x}$, or torques T). A control law then tries to make the measured system parameter S_{meas} move towards a setpoint value S_{des}. Most robotic systems use control laws whose output is a joint velocity or a joint motor current. The joint motions are related to the Cartesian movements required in the task, using the kinematics of the robot and the *Task Frame* specification for robot control [1], [2]. So, this paper works with *velocity resolved* robot control:

$$\dot{x} = K_{FB}(x_{meas} - x_{des}), \tag{1}$$

with $\dot{x}$ the velocity command to the robot, K_{FB} the feedback gain, x_{meas} the measured position, and x_{des} the desired position. Sensor measurements are not only useful for feedback control, but they can also increase the "intelligence" of the robot in several ways. Fig. 2 shows a generic control scheme which integrates all the improvements in intelligence that

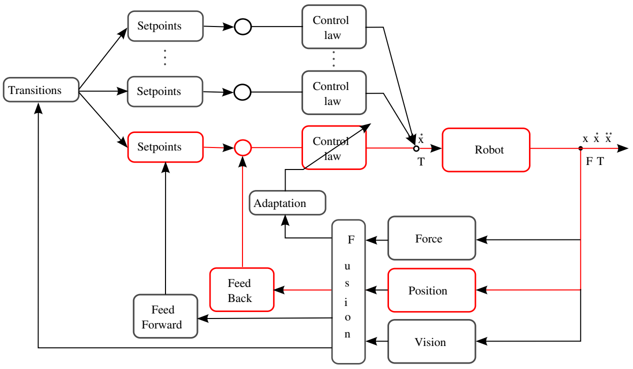

Fig. 2. Multi sensor fusion for robot control.

multi sensor fusion can offer, and which are illustrated in the examples described later in this text.

A. *Feedforward*

The first extension to the basic control loop is to add a *feedforward* contribution: the sensor information (with or without fusion) is used to *identify* the parameters in a *model* of the environment, and the next velocity setpoint of the robot is (partly) constructed from this adapted model information:

$$\dot{x} = K_{FB}(x_{meas} - x_{des}) + \mathbf{K}_{FF}\dot{\mathbf{x}}_{FF}, \tag{2}$$

with $\dot{x}_{FF}$ the feedforward velocity and K_{FF} the feedforward gain. If the feedforward is derived from a correct model, the control loop yields smaller feedback errors. Sections III-A–V-B show how force sensing can be used as feedforward signal. Sections V-A–V-C show how a feedforward signal can be generated from vision sensing.

B. *Model Based Feedback*

A second extension to the basic control loop is model based feedback: not only the feedforward but also the feedback component comes from a model. Or rather, the feedback *must* be derived ("observed") from a model, because the controlled system parameter S *cannot* be measured directly, equation (2) changes to

$$\dot{x} = K_{FB}(\mathbf{x}_{model} - x_{des}) + K_{FF}\dot{x}_{FF}, \tag{3}$$

with $x_{model} = f(measurements)$. In this paper, the functions f are Bayesian filters.

C. *Hybrid Event Control*

Every "intelligent" robot control must be able to switch between different control subtasks, on the basis of the *interpretation* of the available sensor information within a model of the ongoing task. In compliant motion, different subtasks are required for each different contact situation, i.e., different velocity and force controlled degrees of freedom, each with appropriate setpoints, feedforward and feedback. The required *hybrid event control* consists of three different parts, each guided by the available sensor information.

1) The subtask to be executed must be *recognised.*
2) The *decision* when to change to a different control task, and with which control parameters, must be made.
3) A smooth *transition* between the current and the new subtasks must be executed.

D. *Adaptation of control parameters*

A fourth extension to the basic control loop is the adaptation of the control parameters. Multi sensor fusion can help to estimate the uncertainties in the control model parameters, and *to adapt the control law* accordingly. So, it's not just the feedforward contribution that is determined by the model, but also the control law:

$$\dot{x} = \mathbf{K}_{FB}(meas)(x_{model} - x_{des}) + \mathbf{K}_{FF}(meas)\dot{x}_{FF}, \tag{4}$$

with K_{FB} and K_{FF} being function of the sensor measurements.

The next sections discuss several examples of how multi sensor fusion has been used to add each of the above-mentioned "intelligence" components to compliant motion robot control.

III. POSITION FORCE FUSION

The development of position/force control was crucial for the execution of robot tasks in contact with the environment. [4] showed how such compliant motion can be specified and controlled using the Task Frame as "model" for the task.

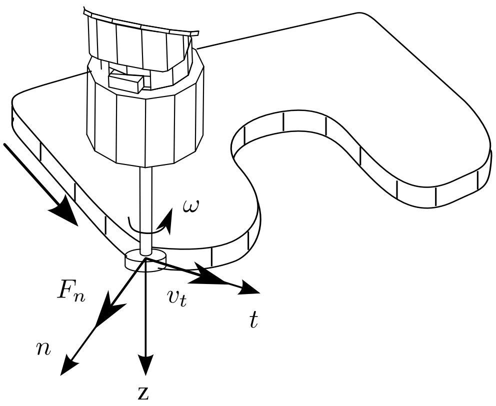

Fig. 3. Task Frame specification of 2D contour following.

A. *2D contour following*

A first common compliant motion task is 2D-contour following of an (unknown, planar) contour. The force sensor information is used to create a feedforward signal to improve the tracking of the contour. Fig. 3 shows how the contour following task is specified in a Task Frame at the tooltip: the tangent velocity v_t is velocity controlled; the force normal to the contour f_n is force controlled. Without a feedforward signal following the contour has to be executed slowly to prevent contact loss or excessive contact forces. If the curvature of the contour is known the rotational velocity ω, around the normal of the plane of the contour can be used as a feedforward signal and the task can be executed faster without losing contact and excessive contact forces. The control law is

$$\begin{aligned} v_n &= K_{FB}K_{env}^{-1}(F_{meas} - F_{des}), & (5)\\ \omega_{ff} &= -\kappa v_t, & (6)\\ v_t &= K_{FF}v_{tdes}, & (7)\end{aligned}$$

with κ the curvature of the contour, K_{env} the stiffness of the interaction, and K_{FF} a feedforward gain between 0 and 1. If the contour information is not available, the contour orientation error can be "fused" from the tangent and normal force and velocity components (in case of frictionless contact):

$$\Delta\theta_t = \arctan(-\frac{F_t}{F_n}) = \arctan(\frac{v_n}{v_t}), \tag{8}$$

$$\kappa = d\theta_t/ds, \tag{9}$$

with κ the contour curvature, and $d\theta_t$ the change of the contour orientation along the contour.

The "measurement" (8) has also been used for the on-line identification and matching with a predefined model of the contour using Non-Recursive Deterministic Least Squares, Extended Kalman Filters or a Page–Hinkley test [5]. In this approach we have a model of the contour, but do not know exactly where we are on the contour, but the uncertainty is limited. Using a Non Minimal State Kalman Filter (NMSKF) large uncertainties of the initial position are permitted [6]. Figure 4 shows how the NMSKF estimates the contour position on the contour starting from a uniformly distributed probability along the contour.

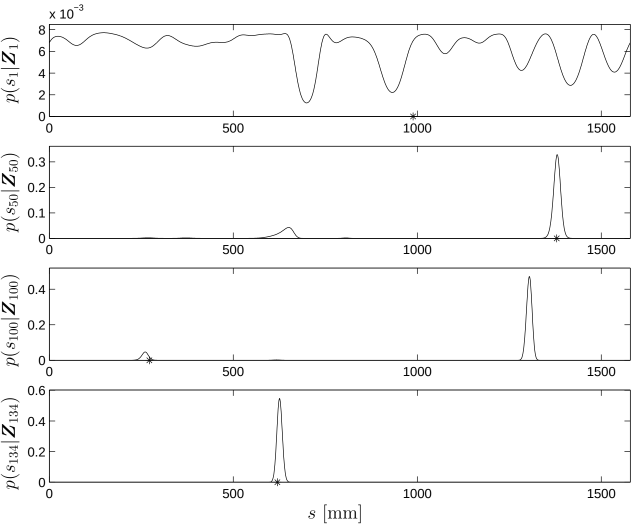

Fig. 4. Probability density function of the contact point (star) along the contour, after 1, 50, 100 and 134 measurements.

This experiment shows how force and position measurements are combined in Bayesian filters to create a *feedforward* signal (II-A). The *adaptation of the control law* (II-D) is used by changing the tangent velocity feedforward gain K_{FF} in Eq. (7). If the contour curvature is not yet known or the uncertainty is still too big, K_{FF} should be small; if the uncertainty decreases, the tangent velocity can be increased until $K_{FF} = 1$. The evolution in the different Bayesian filters allows more and more uncertainty in the model of the contour. Section V-A shows how the model of the contour is replaced by a vision sensor.

B. Cube in Corner experiment

A second application is the Cube–in–Corner experiment (see Fig. 5). In this application position and force information is fused to estimate uncertain geometric parameters of the manipulated object and the environment, and to recognize contact situations and contact state transitions. This has been done with Extended Kalman Filters [7], Iterated Extended Kalman Filters and Non-Minimal State Kalman Filters [8], [6], and particle filters [9], [10], each time with increased robustness against uncertainty in the position and orientation of the cube with respect to the environment. Possible applications are (i) human demonstration, where a task is demonstrated by the human and the robot executes the same task afterwards, or (ii) online estimation, where the geometrical parameters and contact transitions are estimated during the execution of a task and used to adapt the controller parameters.

Fig. 5. Cube in Corner experiment

Fusing position and force sensors even more, the geometric model of the object and the environment can even *be built from scratch* [11]. So, once the 3D contact model is known, the controller can be adapted according to this model and optimize the control. This is an example of II-D. [6], [11] and [10] also show how transitions between contact situations can be detected. This information enables switching between different controllers (II-C). A result of using particle filters for contact formation recognition is shown in Figure 6. Fig. 7 shows a result of the parameter estimation of the environment in a Cube-in-Corner experiment.

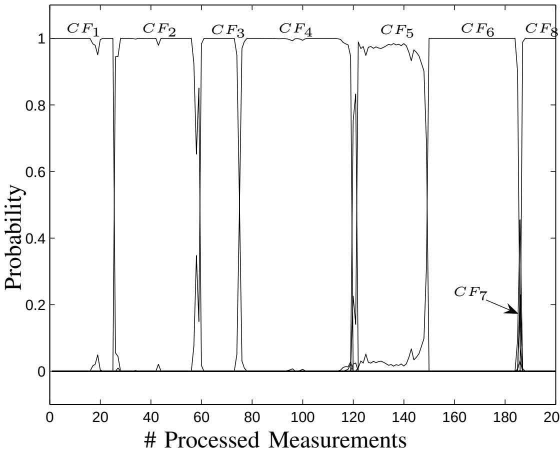

Fig. 6. The probability of eight different contact formations ($CF_1 - CF_8$) in a human demostration of the Cube-in-Corner experiment. Each line represents the probability of each contact formation.

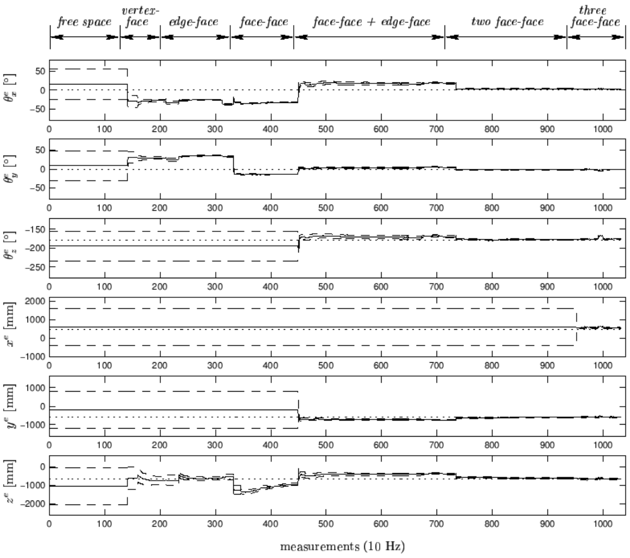

Fig. 7. The estimation of the corner's geometric parameters: Euler angles, θ_x, θ_y, θ_z and 3D position, x, y, z while executing a Cube-in-Corner experiment with succession of different contact formations, from free space to three face-face contacts, with large initial uncertainties imposed on the parameters of the corner. The full line represents the estimation of each parameter, the dashed line represents the covariance of the estimation.

IV. POSITION VISION FUSION

A. *Bayesian Filters for Visual Servoing*

Vision sensors are typically slower than the control loop around the position sensor information. To avoid too large tracking errors in the visual feedback, a Kalman filter can be used to track an object: the Kalman filter enables extrapolating the measured position of the object using the estimated velocity and acceleration. If the velocity or acceleration is needed in the control, a Kalman filter gives much better information (less *phase shift*) than differentiating the position measurements. The system model of the Kalman filter for 1D tracking with estimation of velocity and acceleration looks like:

$$\begin{bmatrix}\tilde{x}_k\\ \tilde{\dot{x}}_k\\ \tilde{\ddot{x}}_k\end{bmatrix} = \begin{bmatrix}1 & T_s & \frac{T_s^2}{2}\\ 0 & 1 & T_s\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\hat{x}_{k-1}\\ \hat{\dot{x}}_{k-1}\\ \hat{\ddot{x}}_{k-1}\end{bmatrix}, \tag{10}$$

with $\tilde{x}_k$ the predicted position at time step k, $\hat{x}_{k-1}$ the estimated position at time step k-1 and T_s the time between k and k-1. Only x is measured. The estimated velocity $\tilde{\dot{x}}$ can be used for feedforward control (II-A). This leads to a better tracking behavior of the controller. The feedback control does not use the *measured* position but the *estimated* position; this is an example of model based feedback control (II-B). Fig. 8 shows the estimation of the position and velocity of a moving object which is measured by a vision sensor. The model used in the Kalman filter includes position, velocity and acceleration of the object.

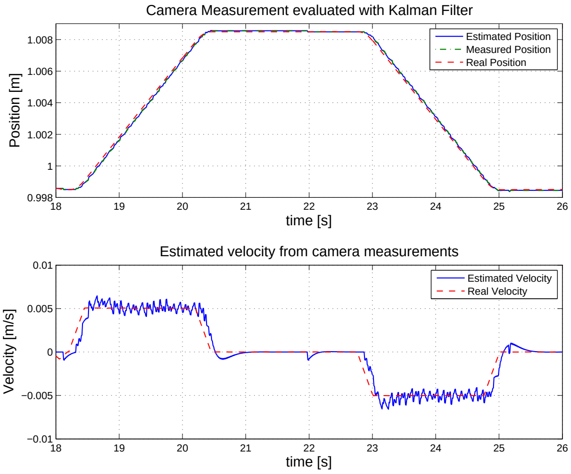

Fig. 8. 1D estimation of position and velocity of camera measurement using a Kalman Filter. In the upper image the full, dotted and dashed line represent the estimated position of the object, the measured position from the camera and the real position. In the lower image the full and dashed line represent the estimated velocity of the object and the real velocity.

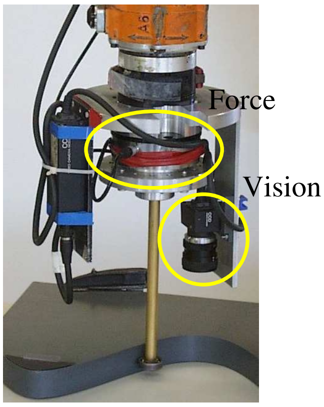

Fig. 9. 2D Contour following with force and vision sensors

V. POSITION FORCE VISION FUSION

A. *2D Contour Following*

Our first successful combination of force and vision for 2D contour following used a vision sensor to get the contour information [12] without having a predefined model. Fig. 9 shows the experimental setup with the camera mounted on the robot end effector. Fig. 10 shows how the vision information is used to calculate the curvature κ using a least squares solution. κ can then be used in Eq. (6) to calculate the feedforward signal. Fig. 11 shows the result of using a feed forward signal for the contact force II-A. Model based feedback control II-B is used to cross sharp corners. Even with feedforward the tangent velocity has to be lower in these corners. The vision sensor can detect these corners and the tangent velocity is decreased by lowering K_{ff} in equation (7).

B. *Shared Control*

Another example of multi sensor fusion is Shared Control. Here a robot manipulator and a human are controlling the

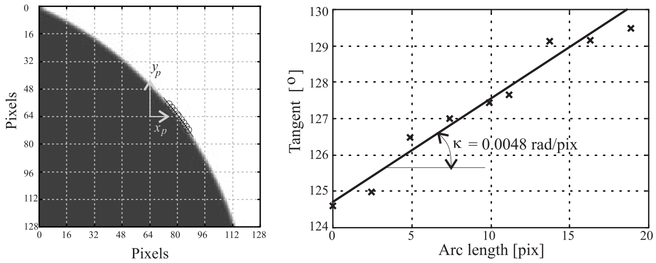

Fig. 10. Tangent to contour in nine points, every point is represented by a circle (**left**); corresponding least squares solution for the curvature computation(**right**).

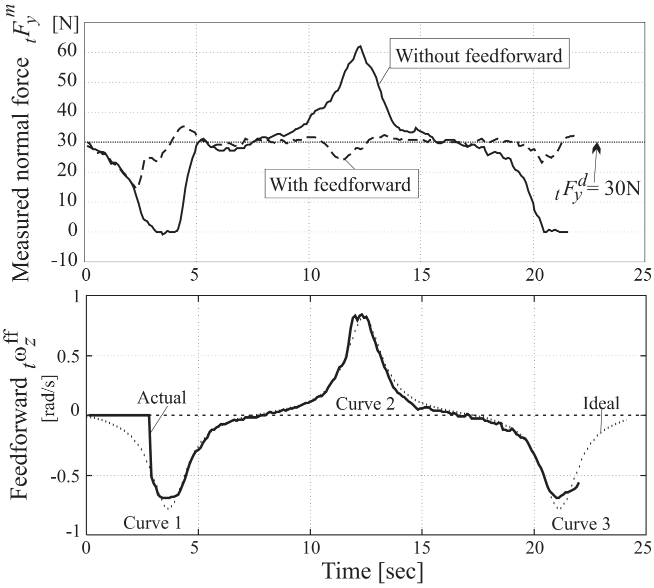

Fig. 11. Measured normal contact forces with and without the use of vision based feedforward in the contour following experiment. The contour consists of three successive corners curve 1 to curve 3 with the same radius and an alternating sign. In the upper image the full line represents the measured normal force without feedforward control. The dashed line represents the measured normal force with feedforward control. The desired normal force is $30N$. In the lower image the full line represents the calculated feedforward velocity ω_{ff} along the contour. The dotted line represents the ideal feedforward velocity.

motion of the same object. The robot manipulator helps the human to move the object, Fig. 12. The easiest way to do this is to use a force sensor to measure the human-robot interaction forces and use force feedback control. To get a smooth motion along a trajectory requires the use of a Kalman filter to estimate the velocity with which the human wants to move the robot. The state of the Kalman filter consists of the position and velocity of the desired motion. The measurements are the force input from the human and the robot position. This way the desired velocity of the object by the human is estimated and used as feedforward control.

An extra sensor in this application is the vision sensor. The camera is used to align the manipulated object with the part of the environment that is on the opposite side from the user. In this control state we have the desired position at one side of the object from the vision sensor, the desired position at the human side, and the velocity of the object coming from the human through the force sensor. This way the control of the object is over-constrained. Weighting between the different constraints leads to a desired spring behavior between the desired camera position and the desired human position.

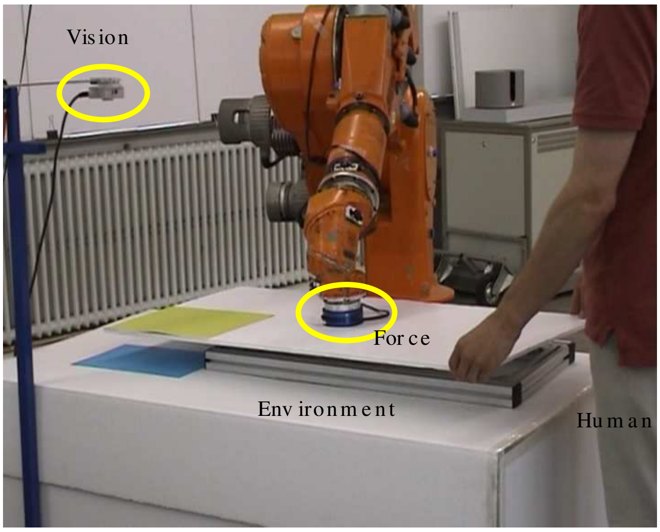

Fig. 12. Shared control with force and vision sensors

This task is too complicated to be specified using the Task Frame, so a more flexible specification framework is used, [3]. Using this framework, the degrees of freedom can be controlled in different frames on different objects. This approach is more appropriate for multi sensor fusion, where each sensor typically processes its measurements in its own frame. The framework can also deal with overconstrained motion which is definitely the case in this experiment.

C. Milling on a moving workpiece

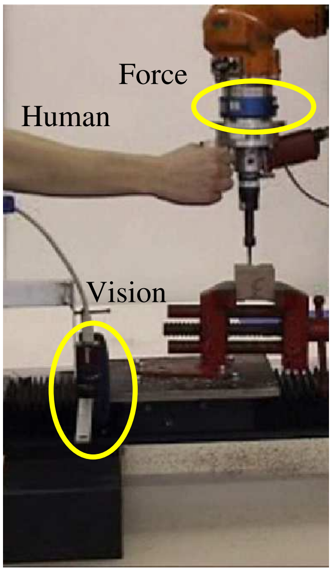

Fig. 13. Milling on a moving workpiece

Another application using this new specification framework using different sensors, is the milling on a moving workpiece. Figure 13 shows the experimental setup. The workpiece can be moved by a XY-table to simulate the disturbing movement of the workpiece. The planar movement is measured with a camera using 2D template matching and the Kalman filter of IV-A to estimate its position, x_{camera}, velocity, $\dot{x}_{camera}$, and acceleration. The velocity is used as a feedforward signal in the position control loop. The XY-movement of the mill is controlled by a feedback on both the movement of the workpiece and the force input of the human. Sensor fusion has to be done to apply both sensor measurements in the same controlled degree of freedom. Equations (12)–(15) show how the sensor information is fused. In equation (11) the desired velocity $\dot{x}_{force}$ is calculated using force feedback with F_{des}, the desired force equal to zero, K_{env} the stiffness of the interaction between the robot and the human and K_{FB} a feedback gain. A new desired position of the mill from the force input x_{force} is calculated using equation (12), with T_s the timestep between the previous position at timestep k and the new position at timestep k-1. Equations (13) and (14) represent the sensor fusion with x_{des} the new desired position of the mill and $\dot{x}_{ff}$ the feedforward velocity, both composed from the camera and force measurements. In equation (15) the output velocity $\dot{x}_{robot}$ is calculated using model based feedback II-B and feedforward II-A for the fused vision and force sensor information.

$$\dot{x}_{force} = K_{FB} K_{env}^{-1} (F_{des} - F_{meas}), \tag{11}$$

$$x_{force}(k) = x_{force}(k-1) + \dot{x}_{force} T_s, \tag{12}$$

$$x_{des} = x_{camera} + x_{force}, \tag{13}$$

$$\dot{x}_{ff} = \dot{x}_{camera} + \dot{x}_{force}, \tag{14}$$

$$\dot{x}_{robot} = K_{FB}(x_{meas} - x_{des}) + \dot{x}_{ff}. \tag{15}$$

VI. CONCLUSIONS AND FUTURE WORK

This paper has given an overview of increasingly complex *compliant motion* robot tasks, where sensor fusion between position, force and vision allows to increase the on-line uncertainty in the relative position and motion between the robot and objects in its environment. The Bayesian approach (with particle filters, and various Kalman filter variants) is the computational framework for the sensor fusion. This sensor fusion is integrated in a flexible task modelling and programming framework, using multiple "feature frames" on the robot, the sensors, and the environment objects.

One of the main objectives in our future research is to develop software support to specify multi sensor robot tasks using our new specification and sensor fusion framework. This software support will enable the use of our framework for multi sensor robot task in industry and other research groups. The handling of the different sensor information will be offered through libraries. End user support as well as development support will be available through easy-to-use user interfaces.

REFERENCES

[1] H. Bruyninckx and J. De Schutter, "Specification of force-controlled actions in the "Task Frame Formalism": A survey," *IEEE Trans. Rob. Automation*, vol. 12, no. 5, pp. 581–589, 1996.

[2] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-11, no. 6, pp. 418–432, 1981.

[3] J. De Schutter, J. Rutgeerts, E. Aertbelien, F. De Groote, T. De Laet, T. Lefebvre, W. Verdonck, and H. Bruyninckx, "Unified constraint-based task specification for complex sensor-based robot systems," in *Int. Conf. Robotics and Automation*, Barcelona, Spain, 2005, pp. 3618–3623.

[4] J. De Schutter and H. Van Brussel, "Compliant Motion I, II," *Int. J. Robotics Research*, vol. 7, no. 4, pp. 3–33, Aug 1988.

[5] S. Demey, H. Bruyninckx, and J. De Schutter, "Model-based planar contour following in the presence of pose and model errors," *Int. J. Robotics Research*, vol. 16, no. 6, pp. 840–858, 1997.

[6] T. Lefebvre, H. Bruyninckx, and J. De Schutter, *Nonlinear Kalman Filtering for Force-Controlled Robot Tasks*, ser. Springer Tracts in Advanced Robotics. Springer Verlag, 2005.

[7] J. De Schutter, D. Torfs, S. Dutré, and H. Bruyninckx, "Invariant hybrid force/position control of a velocity controlled robot with compliant end effector using modal decoupling," *Int. J. Robotics Research*, vol. 16, no. 3, pp. 340–356, 1997.

[8] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Exact nonlinear Bayesian parameter estimation for autonomous compliant motion," *Advanced Robotics*, vol. 18, no. 8, pp. 787–800, 2004.

[9] K. Gadeyne, T. Lefebvre, and H. Bruyninckx, "Bayesian hybrid model-state estimation applied to simultaneous contact formation recognition and geometrical parameter estimation," *Int. J. Robotics Research*, vol. 24, no. 8, pp. 615–630, 2005.

[10] W. Meeussen, J. Rutgeerts, K. Gadeyne, H. Bruyninckx, and J. De Schutter, "Contact state segmentation using particle filters for programming by human demonstration in compliant motion tasks," K.U.Leuven, Leuven, Belgium, Tech. Rep., 2006.

[11] P. Slaets, J. Rutgeerts, K. Gadeyne, T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Construction of a Geometric 3-D Model from Sensor Measurements Collected during Compliant Motion," in *Int. Symp. on Experimental Robotics*, Singapore, Australia, 2004, in press.

[12] J. Baeten, H. Bruyninckx, and J. De Schutter, "Integrated vision/force robotics servoing in the task frame formalism," *Int. J. Robotics Research*, vol. 22, no. 10, pp. 941–954, 2003.