Abstract—Computer vision is often used in robotics where image-based feature detection is an important part. The obtained features can be exploited, e.g. for path planning, process monitoring or feedback control. In this paper the focus is on vision-based vibration damping of robots with flexible links. The measured values for control are obtained by extracting image features. The required image processing framerate depends on the link dynamics. Image processing in general is a computationally expensive task since the complexity for pixel operations is of order $O(n^2)$. Efficient algorithms for online feature tracking have to be used. In an experiment, image processing is performed on a low cost computer and results regarding the computational time are presented. The feature detection performance is validated by results of the vision-based vibration damping control.

I. INTRODUCTION

In modern robotics applications reduction of cycle times is a critical aspect. Lightweight robots are ideal for fast operations due to their lower link inertia compared to typical industrial robots. Also power consumption is reduced. Nevertheless, the mechanical structure of lightweight robots leads to an inherent low link stiffness which causes undesirable vibrations. However, in contact with the environment increased compliance introduced by link flexibility might be even required, especially when robots are interacting with humans.

In order to damp the resulting link oscillations, additional sensors are required since the robotic system is underactuated. The elasticity of the links represent the unactuated degrees of freedom, see also [1].

Usually strain gauges, accelerometers or optical sensors are used for vibration control, see also [2]. In [3] strain gauges are used for curvature feedback control. Since the mounting of strain gauges is quite complex and error-prone, accelerometers are often used instead, because they are easier to apply. The acceleration measurements can be directly used for feedback control or for state estimation of flexible link robots. An example of vibration damping with accelerometers can be found in [4].

The tasks performed by robots are often monitored by an external camera system. This can be used e.g. for safety in the robot environment, process monitoring for fault detection or supervision of manipulating tasks. For guidance of the tool center point (TCP), a camera can be mounted on the robot. In this case the relative pose between the TCP and a target object can be estimated, e.g. for grasping. Since cameras are widely spread in robotic applications they can be used for vibration damping while no additional sensors are required.

Features in the camera image are used for detection of the link vibrations. The approach for vibration damping is to extend PD control of the motor angles by PD control of the feature positions transformed into the joint space using a linearization of the image Jacobian in an operating point. The dynamics of the flexible links are modeled by concentrated elasticity in the joints (lumped element).

Different methods can be used for feature detection and tracking. The image processing rate is critical for successful vision-based vibration damping, since at least the first link eigenfrequency has to be detected. However, many algorithms have high computational costs. Therefore, in this paper some feature detection methods have been tested on a low cost computer. The first approach was the markerless estimation of the optical flow, which has already been used successfully for vibration damping of a flexible link robot in [5]. Since this approach did not work with the given setup, markers are used for feature tracking. Markers are either detected by blobs or their contours. Circular shaped markers are projected as ellipses in the image plane. With detected blobs the marker centroid is calculated. By contour detection an ellipse has to be approximated in a further processing step.

The performance of the implemented feature tracking methods is compared. Also, the quality of a feature tracking method has to be validated in combination with the vibration damping control.

II. MODELING AND CONTROL

In this section a control law for a flexible link robot using image features is presented. Also, the equations of motion used for simulation and control design are shown.

A. Dynamic Modeling

Assuming a three degrees of freedom (3-DOF) flexible link robot, the link vibrations can be modeled using a concentrated joint elasticity. This simplifying approach results in a dynamic model sufficient for the purpose of vibration damping. The equations of motion

$$M_M \ddot{q}_M + \tau_f (\dot{q}_M) + \tau_A = \tau_M$$  
$$M_A (\dot{q}_A) + g_A (q_A, \dot{q}_A) = \tau_A$$  
$$K(q_M - q_A) = \tau_A$$

are partitioned into the dynamics of the motor angles $q_M \in \mathbb{R}^3$ and the virtual link angles $q_A \in \mathbb{R}^3$. The motor dynamics (1) and the link dynamics (2) are coupled via (3). The

1Institute of Robotics, Johannes Kepler University Linz, Altenbergerstraße 69, 4040 Linz, Austria, www.robotik.jku.at, {florian.pucher, hubert.gattringer, a.mueller}@jku.at
inertia matrices are $M_M$ and $M_A(q_A)$ respectively. Motor friction is denoted by $\tau_f(q_M)$ and the generalized motor torques are $\tau_M$. The joint torques $\tau_A$ are resulting from the virtual spring stiffness matrix $K$. The centrifugal and Coriolis terms, as well as link damping and gravity are combined in $g_A(q_A, q_A)$.

B. Camera Model

For vision-based vibration damping a camera model, as shown in Fig. 1, is required. The perspective projection of a point $P$ with $C_P = (x_{CP}, y_{CP}, z_{CP})$ onto the image plane with distance $f$ along the optical axis $C_z$ from the camera center $C$ is

$$
\begin{pmatrix}
  u \\
  v
\end{pmatrix} = \frac{1}{z_{CP}} \begin{pmatrix}
  f_u x_{CP} \\
  f_v y_{CP}
\end{pmatrix} + \begin{pmatrix}
  u_0 \\
  v_0
\end{pmatrix}.
$$

(4)

The projected point is denoted by $p$ with image coordinates $p = (u, v)$. The focal lengths $f_u, f_v$ and the camera center $(u_0, v_0)$ are the intrinsic camera parameters. The position vectors of $C$ and $P$ from the inertial point $I$ are $r_C$ and $r_P$, respectively.

C. Vision-Based Vibration Damping

The vibration damping control law for flexible link robots using a camera was presented in [6]. Image feature points are transformed into the joint space and state feedback is applied. For that, a camera is mounted at the TCP (eye-in-hand). Differentiating (4) w.r.t. time leads to

$$
\dot{r}_p = J_p \dot{q}_A,
$$

(5)

with $J_p = \begin{pmatrix} J_{p,v} & J_{p,\omega} \end{pmatrix}$

$$
J_{p,v} = \begin{bmatrix}
  -\frac{f_u}{z_{CP}} & 0 & \frac{v_0}{z_{CP}} \\
  0 & -\frac{f_v}{z_{CP}} & \frac{u_0}{z_{CP}} \\
\end{bmatrix},
$$

(6)

$$
J_{p,\omega} = \begin{bmatrix}
  \frac{u v_c}{f_c} - \frac{f_u}{f_c} v_c & \frac{u v_c}{f_c} - \frac{f_u}{f_c} v_c \\
  \frac{f_u}{f_c} v_c & \frac{u v_c}{f_c} - \frac{f_u}{f_c} v_c \\
\end{bmatrix}.
$$

(7)

Therein the image Jacobian is $J_p \in \mathbb{R}^{2 \times 6}$, the camera velocities are $\dot{z}_C$, and abbreviations $u_c = u - u_0$ and $v_c = v - v_0$ are used. The image Jacobian $J_p = J_p(u, v, z_{CP})$ depends on the unknown distance $z_{CP}$ of the feature point. Possible solutions for this problem are addressed in [7]. In this paper $J_p(u, v, z_{CP}) \approx \tilde{J}_p = J_p(u, v, \tilde{z}_{CP})$ where $\tilde{z}_{CP}$ is an approximation for $z_{CP}$.

The camera velocities can be expressed as $\dot{z}_C = J_C(q_A) \dot{q}_A$ with the geometric Jacobian $J_C(q_A) \in \mathbb{R}^{6 \times 3}$ regarding the angular velocities of the links $\dot{q}_A$. The unknown arm angles are replaced by the desired values $J_C(q_A) \approx J_C(q_{A,d})$. The image velocities $\dot{r}_p$ are given with

$$
\dot{r}_p = J_p J_C(q_{A,d}).
$$

(8)

However, for control the inverse kinematics is of interest. Since $r_p \in \mathbb{R}^2$ and $\dot{q}_{A,d} \in \mathbb{R}^3$ more than one feature point is needed. For explicit calculation of the camera velocities at least three image points are required. With

$$
\begin{pmatrix}
  \dot{r}_{p1} \\
  \dot{r}_{p2} \\
  \dot{r}_{p3}
\end{pmatrix} = \begin{pmatrix}
  J_{p1} \\
  J_{p2} \\
  J_{p3}
\end{pmatrix} \dot{q}_C
$$

(9)

an inverse $J_p^{-1}$ can be computed. Approximation leads to $J_p^{-1} \dot{J}_F \approx \dot{J}_F = \begin{pmatrix} \dot{J}_{p1} & \dot{J}_{p2} & \dot{J}_{p3} \end{pmatrix}$. Using a linearization of the forward kinematics at $q_{A,d}$, i.e.

$$
\Delta r_F \approx \dot{J}_F J_{C,d} \Delta \dot{q}_A
$$

(10)

with $\Delta q_i = q_{i,d} - q_i$, $i \in \{M, A\}$ and $\Delta r_F = r_{F,d} - r_F$ the control law for vibration damping is

$$
\tau_M = K_{PM} \Delta \dot{q}_M + K_{DM} \Delta \dot{q}_M + K_{PA} J_{C,d} J_p^{-1} \Delta r_F + K_{DA} J_{C,d} J_F^{-1} \Delta \dot{r}_F.
$$

(11)

The Moore-Penrose pseudoinverse is denoted by $(\bullet)^+$. The first row in (11) is a typical PD control of motor angles used for control of robots with rigid links. The second row is used for damping of the link oscillations. Feature positions and/or velocities of feature points are required for (11).

The following sections provide a short overview of considered feature tracking methods for vibration damping. The goal is to find efficient algorithms, since the image processing rate is critical for success. Detection of the first link eigenfrequency is mandatory for this task. The methods for feature tracking are divided into markerless and marker-based techniques.

III. MARKERLESS FEATURE TRACKING

Typical features in an image are edges, corners and blobs. Without additional information about the features or the camera scene, corners are best suited for tracking. Blob detection can be especially useful if markers of known shape, size or color are used. The detection of edges can be used for finding contours of objects. Blob and edge detection are subjects of section IV concerning marker-based feature tracking methods.
A. Corner Detection

Corners can be found with the Shi-Tomasi corner detector or the Harris corner detector [8]. In [9] an improvement of the selection criteria for corners compared to the Harris corner detector is presented. For this reason both algorithms require approximately the same amount of computational time.

B. Optical Flow

The optical flow is a vector field describing the relative displacement of pixels between two consecutive frames of a video. The calculated pixel velocities can be used in (11). The differential methods for estimation of the optical flow are based on the assumption that the illumination \( I(u, v, t) \) between two subsequent frames is constant. The equation

\[
\frac{dI}{dt} = \frac{\partial I}{\partial u} \dot{u} + \frac{\partial I}{\partial v} \dot{v} + \frac{\partial I}{\partial t} = 0
\]  

(12)

is the basis for calculation. The optical flow can be computed using, e.g. the Horn-Schunk method [10] or the Lucas-Kanade method [11].

Dense algorithms compute the optical flow for each pixel, whereas the sparse techniques rely on features. Only sparse algorithms, as the pyramidal implementation of the Lucas-Kanade method [12], are considered here.

Since image corners can vanish over time, in each image a new set of corner features is detected and tracked in the consecutive image. This means the method using optical flow only supplies feature velocities but no feature positions. The vibration damping is achieved solely by feedback of image velocities, i.e. by setting \( K_{PA} = 0 \) in (11).

IV. MARKER-BASED FEATURE TRACKING

The use of objects (markers) of known size, shape and color can greatly reduce the processing time of feature tracking. Since the main goal is the verification of the control law (11) for vibration damping, the image environment is constructed to have only few textures. This makes it easier to detect the markers and reduces computational effort. In this paper three black circular markers on a light gray background are used. Due to projection into the image plane elliptic markers have to be assumed. These markers can be detected by either the blob regions or the boundaries of the regions, i.e. the contours.

A. Region of Interest

A method for vastly reducing the computational effort is the use of small image areas, the regions of interest (ROI), where the image processing is performed. The size of the ROI is chosen by using the knowledge of the marker size in the image and the expected displacement of the marker. The ROI are centered around the feature position of the preceding image.

B. Blob Detection

A basic and fast method for marker detection is the generation of a binary image using thresholding. This separates markers from the background. The detected blobs by using thresholding can be either used directly for estimation of marker properties or further processed, e.g. by extraction of the contour.

For conversion of a gray scale image with pixel intensity \( I(u, v) \) into a binary image with \( I_b(u, v) \) a decision based on a threshold value \( I_{Th} \) is used. If the gray level is greater than the threshold, the resulting pixel is white. If not, it is a black pixel, i.e.

\[
I_b(u, v) = \begin{cases} 
1 & \text{if } I(u, v) > I_{Th} \\
0 & \text{if } I(u, v) \leq I_{Th} 
\end{cases}
\]  

(13)

For varying illumination across different image regions an adaptive threshold can be used. Constant threshold is more efficient here, because for each ROI a different value can be used. In Fig. 2 on the left hand side a gray scale image is shown and on the right hand side is the resulting binary image for a constant threshold value.

C. Contour Detection

Detection of contours can be done by finding the borders of blobs or by edge detection.

1) Border Following: In a binary image there are regions of black pixels adjacent to white pixels. The contours are the connected components found by checking the pixel neighbourhood. A border following algorithm is presented in [13]. In Fig. 3 the found contours are shown for the full image only for demonstration purposes. For efficient calculation the regions of interest are used.

2) Edge Detection and Linking: Edges are typically detected in a gray scale image using the Canny edge detector, see [14]. The edges, however, are not connected in general. If edges are used, they need to be connected to obtain the full contours of an object.
D. Marker Position Estimation Methods

Having the blobs or contours detected, the next processing step is the extraction of the marker positions. The blob centroid is an appropriate candidate for the marker position. In case of detected ellipse contours, the ellipse parameters have to be estimated. Possible methods are based on least-squares techniques or Hough transform. The center of the ellipse is the wanted marker position.

1) Statistical Moments: Shape information of detected blobs can be extracted by the use of statistical moments. Here a binary image with pixels \( I_b(u, v) \in \{0, 1\} \) is assumed although the concept of statistical moments is more general and can be also used for gray scale images. The statistical moment of order \( p + q \) is defined as

\[
m_{pq} = \sum_{u, v \in \mathcal{I}} u^p v^q I_b(u, v)
\]

within a region \( \mathcal{I} \). The location of a marker is required for vibration damping. This can be e.g. the centroid

\[
(\pi, \nu) = \frac{1}{m_{00}} \begin{pmatrix} m_{10} \\ m_{01} \end{pmatrix}
\]

of the marker. Thresholding and calculation of the moments can be efficiently done within only one loop over the pixels. The decision if a blob is the wanted marker, can be based on the area

\[
m_{00} = \sum_{u, v \in \mathcal{I}} u v I_b(u, v)
\]

of the blob and the previous marker location.

2) Ellipse Approximation Using Least-Squares: The equation for a general ellipse in image coordinates \((u, v)\) is

\[
a_{uu} u^2 + a_{uv} u v + a_{vv} v^2 + a_u u + a_v v + a_0 = 0
\]

with the parameters \( a^T = (a_{uu} \ a_{uv} \ a_{vv} \ a_u \ a_v \ a_0) \).

An algorithm for least-squares fitting is presented in [15]. The method is based on eigenvector calculation. The center of the ellipse is the feature position used for vibration damping.

The gray scale image with the detected ellipse contours in the regions of interest is shown in Fig. 4. The centers of the ellipses are also drawn in the figure. The least-squares fitting can be also successful if some parts of the contour are missing.

3) Ellipse Extraction Using Hough Transform: Using the Hough transform geometric objects like lines, circles or ellipses can be found in a contour image. Based on the equation of the corresponding geometric object the parameter space is quantized. For each set of parameters an accumulator is increased if the equation is fulfilled for a pixel. The computational effort increases with a high dimensional parameter space, since the method is like a brute-force algorithm. An ellipse has a five dimensional parameter space, therefore the hough transform is quite computationally expensive.

E. Marker Tracking

If more than one ellipse is detected within a region of interest, the one with the smallest euclidian distance from the previous ellipse is chosen. In this case the feature tracking does not require any additional image processing operations.

V. EXPERIMENTAL SETUP AND RESULTS

For the experiment a Raspberry Camera is mounted at the TCP of the 3-DOF flexible link robot ELLA (Elastic Laboratory Robot), developed at the Institute of Robotics at the Johannes Kepler University Linz. The experimental setup is schematically shown in Fig. 5. Three black circular markers are placed in front of the robot with a distance of ca. 0.06 meters from the camera. The first link eigenfrequency lies within the range of 4 to 5 Hertz. The maximum frame rate of the camera is 90 frames per second. Image processing is performed on a Raspberry Pi 2 for gray scale images with a resolution of \(640 \times 480\) pixels. The size of the ROI is \(120 \times 100\) pixels for each of the three regions.

Fig. 4. Gray scale image including ROI and detected ellipses

Fig. 5. Elastic robot with camera-setup
A. Feature Tracking Performance

With the Shi-Tomasi corner detector and the pyramidal implementation of the Lucas-Kanade feature tracker an image processing rate of 25 frames per second is achieved by using a full image. This method was tested in an environment with more image textures than in the marker-based ones. Compared to the first link eigenfrequency the computational time is too high for vibration damping with the given experimental setup.

Hough transform was only tested for circles, which have a three dimensional parameter space, and was omitted immediately, since the performance was insufficient and the computational effort for ellipses is higher than for circles.

With the method using (15) for blob centroid calculation in a thresholded image, 90 frames per second are obtained. The marker detection algorithm with the steps
1) Threshold
2) Contour Detection by Border Following
3) Least-Squares Ellipse Fitting
also reaches an image processing performance of 90 frames per second. The last described algorithm was used in the validation experiment.

B. Experimental Results

In Fig. 6 and Fig. 7 the vision-based vibration damping is compared to the undamped case, i.e. pure PD control of the motor angles. The excitation signal is a step disturbance of 1 Nm in the motor torques. The resulting link vibrations are successfully damped. In this experiment no feature velocities were used in the control law (11), i.e. $K_{DA} = 0$.

In the horizontal oscillations, i.e. Fig. 6 (top) and Fig. 7 (top), it is obvious that vibrations with higher frequencies are difficult to damp using vision-based control.

VI. CONCLUSIONS

In vision-based control feature tracking is a challenging task. Especially for robots with flexible links, where the objective is vibration damping, fast motion detection is required. Therefore, in this paper different feature tracking methods were analyzed regarding computational efficiency. Marker-based techniques have the advantage of greatly reducing the size of processing data with few image processing steps due to a-priori knowledge of the objects. With two methods the maximum possible image processing rate for the given camera setup was achieved. The efficiency of the feature tracking was validated by a vibration damping experiment.

In the future investigations on calculating the optical flow within ROI will be done. Also a comparison between marker position extraction from contours and blobs regarding robustness is of interest. Furthermore, the use of edges for contour detection has to be implemented and compared to border following.

ACKNOWLEDGMENT

This work has been supported by the Austrian COMET-K2 program of the Linz Center of Mechatronics (LCM), and was funded by the Austrian federal government and the federal state of Upper Austria.

REFERENCES


