2023 European Control Conference (ECC)
June 13-16, 2023. Bucharest, Romania

Improved Positioning Precision using a Multi-rate Multi-sensor in Industrial Motion Control Systems

Chaitanya Jugade, Daniel Hartgers, Sajid Mohamed,
Dip Goswami, Andrew Nelson, Gijs van der Veen, and Kees Goossens

Abstract— **Industrial motion control systems, e.g. pick-and-place tasks in semiconductor manufacturing equipment, require precise positioning for achieving high machine throughput. Linear encoders are the standard industrial sensors used for position feedback due to their relatively low cost, high resolution, and high operating frequency. The challenge is that the linear encoders measure the positions at the points-of-control of the equipment, e.g. motors, and not at the points-of-interest, e.g. pick-and-place positions. The coupling between a point-of-control and the point-of-interest is affected by external disturbances such as mechanical misalignment of the product, friction, and warping of the material, and linear encoders fail to sense these disturbances. Vision-based sensing is a potential alternative to achieve robust sensing and high-precision control. However, vision processing has a long computational delay and affects the machine throughput.**

In this paper, we propose a multi-rate multi-sensor fusion approach to improve the positioning accuracy of industrial motion control systems with different points-of-control and points-of-interest. We present a multi-rate Kalman filter with bias correction to fuse accurate but slow and delayed vision sensor data with fast but less accurate linear encoder data for high-precision position control. We validate the proposed method in an evaluation framework by considering an industrial case study of a semiconductor die-bonding machine. A design-space exploration is done to evaluate the performance of the proposed solution with respect to various relevant design parameters. The effectiveness of the proposed solution depends on the type of disturbances and vision processing delay. For the parameter range under consideration, we achieve a positioning accuracy of $1\mu m$.

I. Introduction

Semiconductor manufacturing machines are common examples where high-precision and high-speed motion systems are extensively used. The positions of the motors of these systems are often sensed by linear encoders [1]. The encoders are precise, but they measure the position at the point-of-control at which the controller operates, e.g. motors, and not at the points-of-interest, i.e., the actual position, e.g. pick-and-place positions of the product, which are different due to external disturbances. The coupling between a point-of-control and the point-of-interest is affected by external disturbances that these motion control systems may experience. An external disturbance occurs in industrial processes due to uncertain operating environments, e.g., ageing deterioration, mechanical vibrations, elastic deformation of the wafer surface, mechanical joint misalignment, and so on [2]. These disturbances cause displacements of the product being handled, lead to positional errors in the process, and influence the control performance, such as positioning accuracy. Linear encoders do not typically detect such disturbances. Vision-based object detection techniques offer a higher sensing accuracy to measure the true position of the product at the point-of-interest [3] and are a means to overcome the above-mentioned sensing limitation. In recent years, many applications have shown the benefits of vision-based approaches in control systems [4]. The major bottleneck in vision-based (motion) control systems is the long computation delay caused by the vision processing, leading to a long sampling period. In recent literature, many approaches, such as parallel and pipeline processing, have been reported to reduce the vision processing time [5], [6], [7]. However, achieving a short enough processing time to meet the high-frequency requirements of a typical high-precision motion control application is still challenging.

To address the aforementioned challenges, the key contributions of this paper are:

1) A *multi-rate multi-sensor fusion* algorithm fusing accurate but slow and delayed vision sensor data and fast but less accurate linear encoder data along with a *bias* correction solution to correct the effect of external disturbances. As a whole, the proposed fusion algorithm improves the position accuracy at a high operating frequency by combining measurements from a vision sensor with an encoder sensor, compared to the accuracy achieved by using only encoder.
2) An evaluation framework to analyze and optimize the performance of multi-rate multi-sensor industrial motion control systems.
3) A design-space exploration to analyze the impact of various relevant design parameters on the performance of the proposed solution.

II. Related Work

Vision-based perception and control is considered a promising technology for industrial applications to achieve robust positioning control. Many advanced industrial applications such as robotics and semiconductor manufacturing consider vision-based control as a key technology. [8] proposes the idea of using vision technology for fault diagnosis of machine tools by detecting the machine surface texture.

This work is supported by the ECSEL Joint Undertaking under grant agreement no. 101007311 (IMOCO4.E).

C. Jugade, D. Hartgers, D. Goswami, A. Nelson, and K. Goossens are with the Department of Electrical Engineering, Eindhoven University of Technology (TU/e) (e-mail: c.jugade@tue.nl, d.hartgers@student.tue.nl, D.Goswami@tue.nl, A.T.Nelson@tue.nl, K.G.W.Goossens@tue.nl).

S. Mohamed and G. van der Veen are with ITEC B.V., Netherlands (e-mail: sajid.mohamed@itecequipment.com, gijs.van.der.veen@itecequipment.com).

The approach is a useful solution for fault diagnosis of machine tools. [9] presents the multi-sensor fusion in advanced driver-assistance systems (ADAS) applications using camera sensors, radar, etc. In such applications, multiple sensors are required to deal with uncertainty in the process. Sensor fusion is used in such applications to produce more reliable information from different sensors. However, [9] does not demonstrate the effectiveness of the sensor fusion approach considering the different operating rates of multiple sensors. [8] presents vision technology for fault diagnosis but do not use the vision-based solution in performance-critical closed-loop systems. [10] summarizes the advanced digital twin approaches for enhancing machine performance considering smart integration of sensors for motion control applications.

In this paper, we present the idea of fusing vision and encoder sensors which operate at different rates to overcome the effect of external disturbance in the process to improve the positioning accuracy of the motion control system. We use an advanced Kalman filter for this purpose. The Kalman filter is a promising technique for handling multi-sensor information. The Kalman filter is used to optimally estimate the state based on measurements (with noise). It offers a recursive solution to the discrete-data linear filtering problem [11]. The filter computes optimal state estimations based on a discrete-time state space model and state measurements in two parts, i.e., prediction and update. The algorithm uses a predefined linear model of the system to predict the state at the next time instance and update for errors in the model using the measurements. The prediction and update are combined using the Kalman gain, which is calculated to minimize the mean square error of the state estimate [12]. The optimal estimation from the two different measurements is then used as input to the control system.

III. Motivation and Problem Statement

We consider the wafer stage of a die-bonding machine as the case study [2]. Fig. 1 (a) shows the simplified schematic of the wafer stage of a semiconductor die-bonding machine. The linear encoder measures the position of the motor attached to the wafer table. The camera is mounted on top, focusing on the wafer containing an array of dies placed horizontally and vertically. In this work, we focus only on the horizontal motion along the x-axis. Each die has a dimension of $200\mu m \times 200\mu m$, and they are placed $20\mu m$ from each other in an ideal scenario without disturbances. Therefore, along the x-axis, the distance between the center of two consecutive dies is $220\mu m$, and a die has to move (from left to right) $220\mu m$ to reach the target position. Fig. 1 (b) shows the top view of the die on the wafer table with the current die position and the target position where the die needs to be positioned. Here, the point-of-control is the position of the wafer table, which can be moved along the x-axis, and the point-of-interest is the true position of the die in the wafer, which is the same as the wafer table position without external disturbances.

In the presence of external disturbances, the die position may change and differ from the linear encoder measurement. The idea is to use a camera focusing on detecting the true position of the die and using that as feedback. Generally, the linear encoder is light in terms of computation and runs at a higher operating frequency, while vision-based sensing operates at a slower rate due to a higher computation time. The main question is how to fuse both sensors' data to improve the closed-loop positioning accuracy of the die bonding machine.

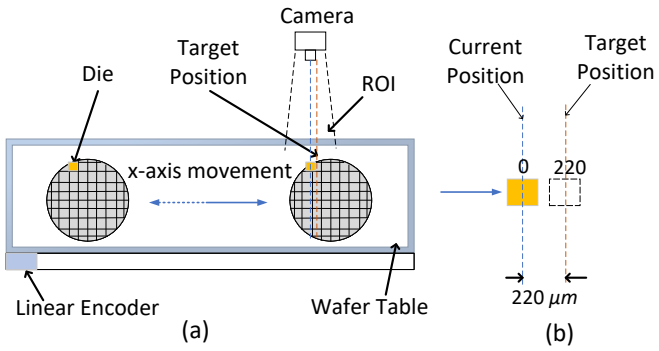

Fig. 1. (a) Schematic of a die bonding platform with camera and linear encoder. (b) Current position without external disturbances and target position.

A. System Model

The dynamic behaviour of a die-bonding machine can be closely modelled by a second-order mass-damper system, where the movement across one axis is represented by a mass moving on a surface. The equation of motion for the mass is given as follows,

$$f = m.a, \tag{1}$$

$$-c_v\dot{x} + f_{ext} = m\ddot{x}, \tag{2}$$

$$m\ddot{x} = -c_v\dot{x} + f_{ext}. \tag{3}$$

m is the mass, f is the total force, a is the acceleration, c_v is the viscous damping constant, f_{ext} is the external force applied by the motor, and x is the position of the die. The continuous state-space model is obtained from the dynamics in eq (3) considering two states - position x_1 and velocity x_2. u is the input to the system which is external force f_{ext}. z is the output measurement i.e., position. The state space model of the system is given by:

$$\dot{x} = Ax + Bu, \tag{4}$$

$$z = Cx, \tag{5}$$

where A, B, and C are given by,

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -c_v/m \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 1/m \end{bmatrix}, \; C = \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T, \tag{6}$$

Given that the encoder samples at a higher rate, we consider the encoder sampling period $h_{encoder}$ as the base sampling period. The corresponding discrete-time state-space model with sampling period $h_{encoder}$ is as follows:

$$F = e^{Ah_{encoder}}, \; G = \int_0^{h_{encoder}} e^{At} dt B, \tag{7}$$

where F and G are the discrete-time state space matrices. We further consider process and measurement noise w_k and v_k respectively as follows:

$$x_{k+1} = F_k x_k + G_k u_k + w_k, \tag{8}$$

$$z_k = C_k x_k + v_k. \tag{9}$$

where $x_k = \begin{bmatrix} x_{1,k} & x_{2,k} \end{bmatrix}^T$, $w_k \sim N(0,Q)$, $v_k \sim N(0,R_e)$ for encoder, and $v_k \sim N(0,R_v)$ for vision. Q is the process noise covariance matrix, R_e is the encoder measurement noise covariance and R_v is the vision measurement noise covariance. F_k, G_k are the discrete-time state space matrices at the k^{th} sampling instance. C_k is the output matrix. Two noise sources are independent i.e., $\mathbb{E}[w_i, v_i^T] = 0$. z_k is the (true) position of the die under consideration at the k^{th} sampling instance. Further, we denote the encoder and the vision measurement at the k^{th} sampling instance by $z_{e,k}$ and $z_{v,k}$, respectively. The control problem is to design a u_k that brings z_k to r_k (reference) at k^{th} sampling instance.

B. Effect of External Disturbances

Fig. 2 (a) shows the first three dies in an ideal scenario without any external disturbances. In ideal scenario, dies are located equidistant from each other with a distance of $20\mu m$. The region of interest (RoI) of the camera focuses on the first die. The center of the first die is located at position $z_0 = 0\mu m$ without disturbances. The control problem is to center the first die to the target position $220\mu m$ at the end of the first iteration. The center of the second die is initially located at the left of the first die at $-220\mu m$, and the third die is positioned at $-440\mu m$. This way, there are several other dies placed along the x-axis.

Fig. 2 (b) shows an example of a scenario when the semiconductor die positions are disturbed due to external disturbances. When external disturbance exists, the dies are not equidistant from each other. In this case, the first semiconductor die (along with all subsequent dies) on the wafer stage is moved to the right by $50\mu m$ and $z_0 = 50\mu m$. This position inaccuracy of $50\mu m$ can not be measured by the linear encoder since it only senses the position of the wafer table, i.e., $z_{e,0} = 0\mu m$. However, the vision sensor is able to measure the true position of the die position using the captured images and vision processing algorithm, i.e., $z_{v,0} = 50\mu m$. Vision measurements are only available at a lower rate. The idea is to fuse both vision and encoder data to obtain an accurate and faster estimate $\hat{z}_k$ of the true die position to improve the closed-loop control performance. The reference position r_k should be adjusted accordingly to the estimated position $\hat{z}_k$.

C. Multi-rate Multi-Sensor Fusion Problem

We consider two sensors – a linear encoder that operates at a sampling period of $h_{encoder}$ and a vision sensor that operates at a sampling period h_{vision}. The two sensors operate at different rates with $h_{vision} \gg h_{encoder}$. Fig. 3 shows the relative timing of multi-rate signals in the closed-loop die bonder system. The relation between vision period and the encoder period is captured by h_v as follows:

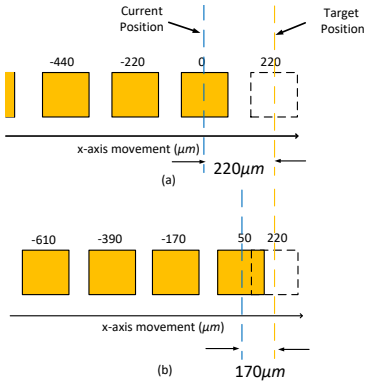

Fig. 2. (a) Ideal die positions without external disturbance i.e., $z_{e,k} = 0\mu m$ and $z_{v,k} = 0\mu m$ (b) Position of the semiconductor die with external disturbance of $50\mu m$ i.e., $z_{e,k} = 0\mu m$ and $z_{v,k} = 50\mu m$.

$$h_v = \left\lceil \frac{h_{vision}}{h_{encoder}} \right\rceil. \tag{10}$$

i.e., the vision period is h_v times longer than the encoder period. The vision measurement starts with capturing the image at instances $k = s, s+h_v \ldots$. The captured images are processed to obtain position information $z_{v,k}$ using a classical object localizer [13]. The worst-case processing time of the object localizer is given by τ_{vision}. Further, the vision measurement $z_{v,k}$ available at the instances $k = s+\tau_v, s+\tau_v+h_v, \ldots$, where

$$\tau_v = \left\lceil \frac{\tau_{vision}}{h_{encoder}} \right\rceil. \tag{11}$$

That is, the vision measurement at k uses the image captured at τ_v sample time ago, and we denote such vision measurement as $z_{v,k-\tau_v}$. The problem is to fuse the above two sensor data and obtain an optimal estimation $\hat{z}_k$ such that it converges to the actual position z_k. That is,

$$\hat{z}_k = f(z_{e,k}, z_{v,k}, \tau_v, h_{encoder}), \tag{12}$$

$$s.t \ (z_k - \hat{z}_k) \to 0. \tag{13}$$

IV. Proposed Multi-rate Multi-sensor Algorithm

A. Proposed sensor fusion algorithm

Fig. 3 shows the relative timing diagram of the proposed sensor fusion algorithm based on the encoder and vision measurements running at different rates. The base sampling period is $h_{encoder}$ which is used in the control algorithm. Therefore, we need an estimate $\hat{z}_k$ at every $h_{encoder}$. While encoder measurement $z_{e,k}$ is available in each sample, the vision measurement $z_{v,k}$ is available only once in every h_v samples, where h_v is defined as shown in Eq. (10). Moreover, the position information from an image captured at k is only available at $(k+\tau_v)$ due to processing delay, where τ_v is determined by Eq. (11). Based on the availability of the sensor measurements, the algorithm (or samples) runs in three modes as explained below.

Mode 1 $(\mathbf{s}+\mathbf{i}\mathbf{h_v}+\boldsymbol{\tau_v} < \mathbf{k} \leq \mathbf{s}+(\mathbf{1}+\mathbf{i})\mathbf{h_v})$: In this mode, only the encoder measurements $z_{e,k}$ are available. This mode runs in the samples $s+ih_v+\tau_v < k \leq s+(1+i)h_v$ as shown in

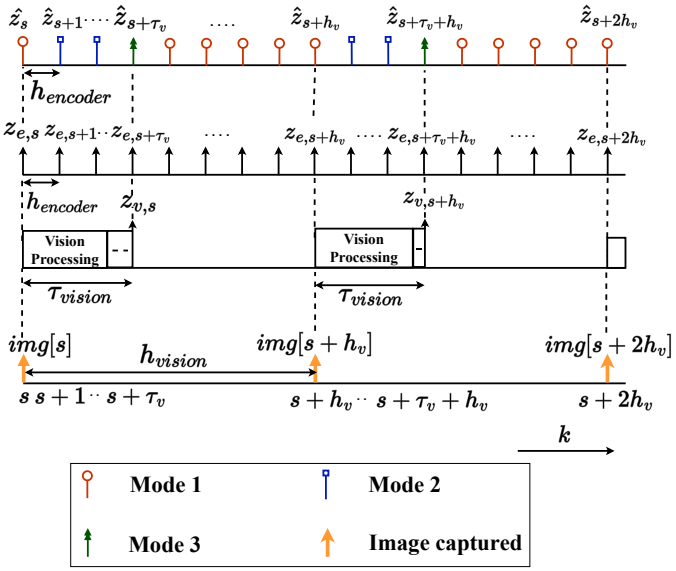

Fig. 3. Timing diagram of multi-rate multi-sensor fusion algorithm. τ_{vision} is the worst-case execution-time of the vision processing, img[s] is the captured image, $z_{v,s}$ is the vision measurement, $z_{e,s}$ is the encoder measurement, and $\hat{z}_s$ is the multi-rate multi-sensor Kalman estimation output at time instance s.

Fig. 3. Algorithm 1 shows the computation steps performed in Mode 1. The algorithm starts in Mode 1 when the first image frame is captured at $k=0$. At $k=0$, Steps 1-3 initialize P_0, R_e, R_v, Q, and $\hat{x}_k(+)$ matrices where P is the estimation covariance, R_e and R_v are the measurement noise covariance matrices for the linear encoder and vision sensor respectively. Q is the process covariance matrix, and $\hat{x}_k(+)$ is the Kalman state estimate. We consider the following initialization:

$$P_0(+) = \begin{bmatrix} var(x_{1,0}) & 0 \\ 0 & var(x_{2,0}) \end{bmatrix}, \quad (14)$$

$$R_e = \sigma^2_{encoder}, \quad (15)$$

$$R_v = \sigma^2_{vision}, \quad (16)$$

$$Q = \sigma_v^2 \cdot G_0 G_0^T, \quad (17)$$

$$\hat{x}_0(+) = 0. \quad (18)$$

We consider the intrinsic noise level in the encoder as $\sigma_{encoder} = 5E-9$m, which is highly precise. However, as already explained, the encoder cannot measure the external disturbances, the presence of which further increases the noise level depending on their nature. The noise level in the vision sensor $\sigma_{vision} = 3.5E-6$m given that we consider $3.5\mu m$ accuracy per pixel of the image. The values of $\sigma_{encoder}$ and σ_{vision} are obtained from the specifications of the industrial die-bonder machine. Q is the process noise covariance matrix.

Step 4 computes the state estimation $\hat{x}_k$ based on the discrete-time model. Step 5 computes the uncertainty of the prediction P_k using the recursive equation and considering the process covariance matrix Q. Step 6 computes the Kalman gain K_k using P_k and considering the encoder measurement noise R_e. Step 10 computes the Kalman state estimate $\hat{x}_k(+)$ based on the encoder measurement $z_{e,k}$ and $\hat{x}_k$ obtained in Step 4. Step 11 computes the update in prediction uncertainly using the Kalman gain K_k. Step 12 computes the estimated output $\hat{z}_k$. Additionally, in Steps 7-9 at the $k = s+ih_v$, we initialize the correction matrix $M_k = M_0 = I$ which serves as a correction term on the Kalman gain when the additional information is received from the second sensor, i.e. vision sensor. M_k is updated in other modes to update the necessary correction before the use of vision measurement. M_k is used in Mode 3 to determine the optimal Kalman gain K_k^*.

Algorithm 1 : Mode 1 ($\mathbf{s+ih_v+\tau_v < k \leq s+(1+i)h_v}$)

1: **if** $k=0$ **then**
2: Initialize: $P_k(+)$, R_e, R_v, Q, $\hat{x}_k(+)$
3: **end if**
4: $\hat{x}_{k+1} = F_k\hat{x}_k(+) + G_k u_k$
5: $P_{k+1} = F_k P_k(+) F_k^T + Q$
6: $K_k = P_k C_k^T [C_k P_k C_k^T + R_e]^{-1}$
7: **if** $k = s+ih_v$ **then**
8: $M_k = I$
9: **end if**
10: $\hat{x}_k(+) = \hat{x}_k + K_k[z_{e,k} - C_k\hat{x}_k]$
11: $P_k(+) = [I - K_k C_k]P_k$
12: $\hat{z}_k = C_k\hat{x}_k(+)$

Mode 2 ($\mathbf{s+ih_v < k < s+ih_v+\tau_v}$) : Mode 2 continues after Mode 1. In this mode, the encoder measurements $z_{e,k}$ are available while the vision processing is ongoing using the images captured at $k = s,\ s+h_v,\ \cdots$. Therefore, vision measurements are not available in this mode. The corresponding samples are $s+ih_v < k < s+ih_v+\tau_v$ as shown in Fig. 3. Algorithm 2 shows the computation steps in Mode 2. All computation steps are the same as Algorithm 1 except for Step 4. In Step 4, we update the correction matrix M_k, which is intended to be used in Mode 3. This way, the latest value of the correction matrix is used in Mode 3 when the vision measurement is available.

Algorithm 2 : Mode 2 ($\mathbf{s+ih_v < k < s+ih_v+\tau_v}$)

1: $\hat{x}_{k+1} = F_k\hat{x}_k(+) + G_k u_k$
2: $P_{k+1} = F_k P_k(+) F_k^T + Q$
3: $K_k = P_k C_k^T [C_k P_k C_k^T + R_e]^{-1}$
4: $M_k = (I - K_k C_k) F_k M_{k-1}$
5: $\hat{x}_k(+) = \hat{x}_k + K_k[z_{e,k} - C_k\hat{x}_k]$
6: $P_k(+) = [I - K_k C_k]P_k$
7: $\hat{z}_k = C_k\hat{x}_k(+)$

Mode 3 ($\mathbf{k = s+ih_v+\tau_v}$) : In this mode, at $k = s+ih_v+\tau_v$, both the encoder and vision measurements are available. Algorithm 3 shows the computation steps performed in Mode 3. Step 5 calculates the new optimal Kalman gain K_k^* based on the latest correction matrix i.e., M_k, the old covariance matrix $P_{k-\tau_v}$ calculated at $k-\tau_v$, and the measurement noise covariance matrix in vision sensor i.e., R_v. Step 7 calculates the state estimates $\hat{x}_e$ based on only the encoder measurement $z_{e,k}$. Step 8 computes the extrapolated measurement z_k^* based

on the vision measurement $z_{v,k-\tau_v}$, i.e., position information from the image captured at $(k-\tau_v)$. Step 9 computes the optimal estimate $\hat{x}_k(+)$ using z_k^*, K_k^* and $\hat{x}_e$. Steps 10 and 11 calculate the update on P_k and the position estimate $\hat{z}_k$. Based on the difference between the two sensor measurements, the bias correction is performed in Step 6 which is explained in the following.

Algorithm 3 : Mode 3 ($\mathbf{k = s + ih_v + \tau_v}$)

1: $\hat{x}_{k+1} = F_k\hat{x}_k(+) + G_k u_k$
2: $P_{k+1} = F_k P_k(+) F_k^T + Q$
3: $K_k = P_k C_k^T [C_k P_k C_k^T + R_e]^{-1}$
4: $M_k = (I - K_k C_k) F_k M_{k-1}$
5: $K_k^* = M_k P_{k-\tau_v} C_k^T [C_k^T P_{k-\tau_v} C_k^T + R_v]^{-1}$
6: $z_{e,k} = bias\_correction(z_{e,k-\tau_v}, z_{v,k-\tau_v}, \varepsilon)$
7: $\hat{x}_e = \hat{x}_k + K_k(z_{e,k} - C_k\hat{x}_k)$
8: $z_k^* = z_{v,k-\tau_v} - C_k\hat{x}_{k-\tau_v} + C_k\hat{x}_k$
9: $\hat{x}_k(+) = \hat{x_e} + K_k^*[z_k^* - C_k\hat{x}_e]$
10: $P_k(+) = P_k - K_k^* C_k P_{k-\tau_v} M_k^T$
11: $\hat{z}_k = C_k\hat{x}_k(+)$

B. Bias correction

The bias correction is performed in Mode 3 in order to compensate for the error in encoder measurement due to disturbances. We consider a threshold ε based on the noise-induced variance in the encoder and vision-based measurements. We compare the encoder and vision measurements at $(k-\tau_v)$, i.e., $z_{e,k-\tau_v}$ and $z_{v,k-\tau_v}$. If their difference is more than ε, we correct the encoder measurement $z_{e,k}$ by the vision measurement $z_{v,k-\tau_v}$ and the change in position observed in the encoder measurements. The corrected encoder measurement is considered in Mode 3 (i.e., in Step 6 of Algorithm 3).

Algorithm 4 : $z_{e,k} = bias\_correction(z_{e,k-\tau_v}, z_{v,k-\tau_v}, \varepsilon)$

1: Input : $z_{e,k-\tau_v}, z_{v,k-\tau_v}, \varepsilon = \frac{\sigma_{encoder}+\sigma_{vision}}{2}$
2: Output : $z_{e,k}$
3: **if** $|z_{e,k-\tau_v} - z_{v,k-\tau_v}| > \varepsilon$ **then**
4: $\quad b_k = z_{e,k-\tau_v} - z_{v,k-\tau_v}$
5: **else**
6: $\quad b_k = 0$
7: **end if**
8: $z_{e,k} = z_{e,k} - b_k$

C. Multi-rate multi-sensor estimation

From an estimation viewpoint, the state estimate $\hat{x}_k$ is computed using only the encoder measurement $z_{e,k}$ in Mode 1 and 2 (see Step 10 in Algorithm 1 and Step 5 in Algorithm 2). However, the state estimation is performed using both encoder measurement $z_{e,k}$ and delayed vision measurement $z_{v,k-\tau_v}$ in Mode 3. In Mode 3, the state estimation is performed in two stages. In the first stage, the estimation $\hat{x}_e$ is obtained using only encoder measurement (see Step 7 of Algorithm 3). In the second stage, the vision-based state estimation is performed using $\hat{x}_e$ as an input (see Step 9 of Algorithm 3). Simplifying Steps 7-9 in Algorithm 3, we obtain the following:

$$\hat{x}_k(+) = \hat{x}_k + (K_k - K_k^* C_k K_k)(z_{e,k} - C_k\hat{x}_k) + K_k^*(z_{v,k-\tau_v} - C_k\hat{x}_{k-\tau_v}). \quad (19)$$

In essence, in Mode 3 samples, the encoder-based estimate is corrected by the vision-based estimates. Mode 1 and 2 have only the encoder-based estimates.

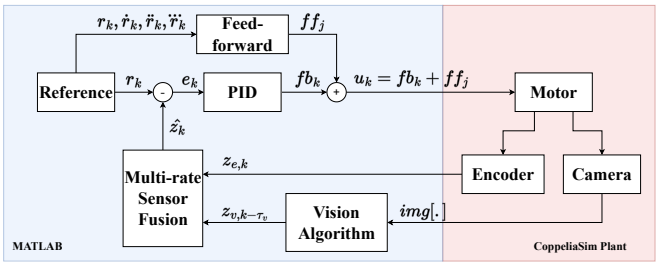

Fig. 4. Control system block diagram with Matlab and CoppeliaSim components.

V. Feedback and feed-forward controllers

The overall control system structure is illustrated in Fig. 4. We use a proportional, integral, and derivative (PID) as the feedback controller and an Iterative Learning controller (ILC) as the feedforward controller in the closed-loop control system. The final control action is the combination of feedback control (fb_k) and feed-forward control (ff_j) i.e., $u_k = fb_k + ff_j$.

A. PID Controller

The discrete-time PID controller equation is

$$fb_k = k_p e_k + k_i \sum_{k=1}^{N} e_k h_{encoder} + k_d \left(\frac{e_k - e_{k-1}}{h_{encoder}} \right), \quad (20)$$

where fb_k is the feedback control action, k_p, k_i, and k_d are the proportional, integral, and derivative controller gains respectively. e_k is the error signal,

$$e_k = \hat{z}_k - r_k, \quad (21)$$

where r_k is the reference at the k^{th} sampling instance.

B. Feed-forward ILC

ILC method utilizes specific aspects of repetitive control tasks to increase the control performance [14]. It exploits the reproducible part of the tracking error (over multiple identical actions) to design a compensation signal. Designing the ILC for the considered case study uses the basis function ILC as explained in [15], which is more suitable for repeating varying trajectories. Consider the following equation of the feedforward signal ff_j,

$$ff_j = k_j \dddot{r}_j + k_a \ddot{r}_j + k_v \dot{r}_j + k_C \cdot sign(\dot{r}_j) + k_s r_j + k_{so}, \quad (22)$$

$$ff_j = \psi_j \theta_j, \quad (23)$$

where,

$$\psi_j = [\dddot{r}_j \quad \ddot{r}_j \quad \dot{r}_j \quad sign(\dot{r}_j) \quad r_j \quad 1], \tag{24}$$

$$\theta_j = [k_j \quad k_a \quad k_v \quad k_C \quad k_s \quad k_{so}]^T, \tag{25}$$

and k_j is jerk, k_a is inertia, k_v is viscous friction, k_C is Coulomb friction, k_s is stiffness and k_{so} is spring offset coefficients to update the feedforward signal [2], where j indicates j^{th} iteration for the feedforward controller. We have assumed k_C as zero for the considered case study. Updates of the feedforward parameters are given by,

$$\theta_{j+1} = Q_{filter}\theta_j + L_{filter}e_j, \tag{26}$$

$$Q_{filter} = (\psi_j^T J^T W_e J \psi_j + W_{\Delta\theta})^{-1}(\psi_j^T J^T W_e J \psi_j + W_{\Delta\theta}), \tag{27}$$

$$L_{filter} = (\psi_j^T J^T W_e J \psi_j + W_{\Delta\theta})^{-1}\psi_j^T J^T W_e. \tag{28}$$

The feedforward addition for the j^{th} iteration for each N discrete sample is determined by the ψ_j vector containing the reference signals and θ_j transposed vector containing the feedforward parameters. The feedforward parameters in θ_j are updated after every iteration, based on the previous parameter settings, the error signal e_j for the j^{th} iteration, filters Q_{filter}, and L_{filter}. These filters are a product of the third-order reference signals, a Toeplitz matrix J containing the impulse response of the process sensitivity, and weight matrices that determine the learning rate of the controller. J is an $N \times N$ lifted representation [14], where N is the number of samples in each ILC iteration. $W_e \in R^{N\times N}$ is a diagonal matrix to impose a weight on the error signal and $W_{\Delta\theta} \in R^{N\times N}$ is a diagonal matrix for weighting the rate of change of the feedforward parameters. More information on the implementation of the basis function ILC can be found in the paper [2].

VI. Performance Evaluation Framework

A. Plant/System model

We model the plant dynamics in the physics simulator CoppeliaSim [16], [17]. In CoppeliaSim, we model the system described in Section III. We model the x-axis movement of the wafer table. The mass that models the wafer table moves across a surface by applying a force as a control action u_k. The CoppeliaSim is interfaced with MATLAB using the MATLAB remote API.

B. Sensors and Vision Processing

We explain the sensors used in the closed-loop framework and the vision processing algorithm.

Linear Encoder: The linear encoder is used to measure the position of the wafer table or the motor responsible for moving the table.

Vision Sensor: Fig. 1 (a) shows the general schematic of the semiconductor die bonder platform with the camera mounted on top for measuring the true position of the die. Fig. 1 (b) shows the image captured by the camera. The image processing algorithm uses the captured image to extract the position information $z_{v,k}$. The vision sensor precision is 3.5 μm per pixel.

Vision Processing: We use Hough transform [13] to obtain the location of the object/die from the image. The algorithm extracts the line segments and returns the x-coordinate of the center point of a product within the RoI. The ratio between the center point and full image width determines the absolute position of the center point to the axis in the simulation setup. Fig. 5 shows the processing steps. The input RGB image is converted to a grayscale image. Subsequently, we apply the Canny edge detection algorithm to detect the edges on the grayscale image. Once we get the image from edge detection, we pass it to the Hough transform algorithm which extracts the line segments on the image, which is used to compute the x-coordinate of the center point of a die.

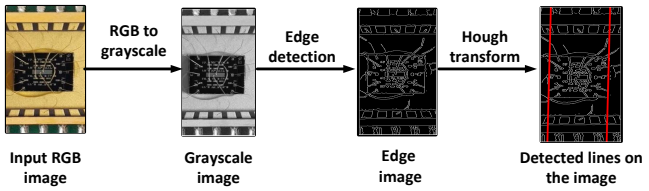

Fig. 5. Vision processing steps

C. Reference Generator

As shown in Fig. 4, the controller requires r_k and its derivatives in every sampling instance. We use the Advanced setpoint generation toolbox [18] to obtain the trajectory for a point-to-point movement within the given bounds. The first reference component contains the position signal that serves as input to the PID controller, and the other reference components are used by the ILC for updating the feedforward parameters.

VII. Results

To evaluate the performance of the closed-loop control system with the proposed multi-rate multi-sensor fusion algorithm, we perform design-space exploration over various relevant parameters and scenarios. The construction of the error scenarios for the design-space exploration is explained. We model two commonly occurring scenarios which are explained in the following.

Fig. 6 (a) shows the scenario without external disturbances. For illustration, we consider 5 dies of dimension $200\mu m \times 200\mu m$ placed with $20\mu m$ gap with each other. The center of any two dies is $220\mu m$ away from each other. We refer to the center of the die as the position of the die. Along the x-axis, the first die is positioned at 0, the second die is positioned at $-220\mu m$, the third die is positioned at $-440\mu m$, and so on. We denote the deviation of these (ideal) positions by Δ_i where $i = \{1,2,3,4,5\}$ for the 5 dies under consideration. Without disturbances, $\Delta_i = 0$ as shown in Fig. 6 (a). The controller is supposed to bring the first die to the target position at $220\mu m$ by moving the wafer table from left to right. Next, the second die moves another $220\mu m$, and so on. Every iteration is of length $50ms$. That is, a die is supposed to reach the target position at the latest by

50ms. Therefore, the 5 dies should reach the target position one by one in 5 iterations to be completed by 250ms. In view of this setup, we consider 2 scenarios under disturbances as described in the following. In all these scenarios, we consider two sensor settings: (i) $h_{encoder} = 0.125ms$, $h_{vision} = 1ms$ (ii) $h_{encoder} = 0.125ms$, $h_{vision} = 8ms$.

- Scenario 1: $\Delta_1 = 20\mu m$, $\Delta_2 = 15\mu m$, $\Delta_3 = 10\mu m$, $\Delta_4 = 5\mu m$, and $\Delta_5 = 0\mu m$ as shown in Fig. 6 (b)
- Scenario 2: $\Delta_1 = 0\mu m$, $\Delta_2 = -5\mu m$, $\Delta_3 = -10\mu m$, $\Delta_4 = -15\mu m$, and $\Delta_5 = -20\mu m$

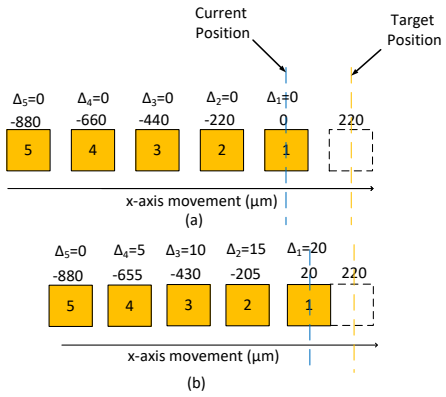

Fig. 6. (a) Scenario without external disturbances (b) Scenario 1: $\Delta_1 = 20\mu m$, $\Delta_2 = 15\mu m$, $\Delta_3 = 10\mu m$, $\Delta_4 = 5\mu m$, and $\Delta_5 = 0\mu m$.

Performance Metrics: We evaluate the closed-loop performance of the control system considering the aforementioned scenarios by (i) Steady-state error (SSE) with e_k at the end of each iteration, (ii) Mean absolute error (MAE) (μm) i.e., $MAE = \frac{1}{N}\sum_{k=1}^{N-1} |(z_k - \hat{z}_k)|$, (iii) Settling time (ST) is the time to reach within 2% of the target position. In this case, e_k should be in the range of $\pm 1\mu m$.

Performance evaluation: Fig. 7 presents the position of the first die in iteration 1 in Scenario 1 with $h_{encoder} = 0.125ms$ and $h_{vision} = 1ms$. Here, $h_v = 8$ and $\tau_v = 7$ with the worst-case execution time of the vision processing $\tau_{vision} = 0.875ms$. The disturbance $\Delta_1 = 20\mu m$. As shown in Fig. 7 (a), the encoder doesn't measure the disturbance and measures the position as 0 at the start, i.e., $z_{e,0} = 0$. The vision measurement starts with image capture at $k = 0, 8, 16, \cdots$. The vision measurements are available at $k = 7, 15, 23, \cdots$ with a delay of τ_v. The first vision measurement $z_{v,0}$ is available at $k = 7$ or at 0.875ms. We obtain $z_{v,0} = 19.1\mu m$ which is close to the true position of $20\mu m$. The output of the proposed estimator $\hat{z}_k$ should go to the true position with time as what can be noticed from Fig. 7 (b). At the end of the iteration at 50ms, the encoder measurement $z_{e,k}$ still measures an error of around $20\mu m$ (i.e., $z_{e,k} = 200\mu m$) while the true position z_k converges to the target position of $220\mu m$. As shown in Fig. 7 (a), the estimation $\hat{z}_k$ experiences significant drift between 2 to 4ms since the bias correction is triggered. These results demonstrate that the proposed fusion algorithm improves the positioning estimates by integrating the vision and encoder measurements as opposed to the case where only encoder is used for position estimation.

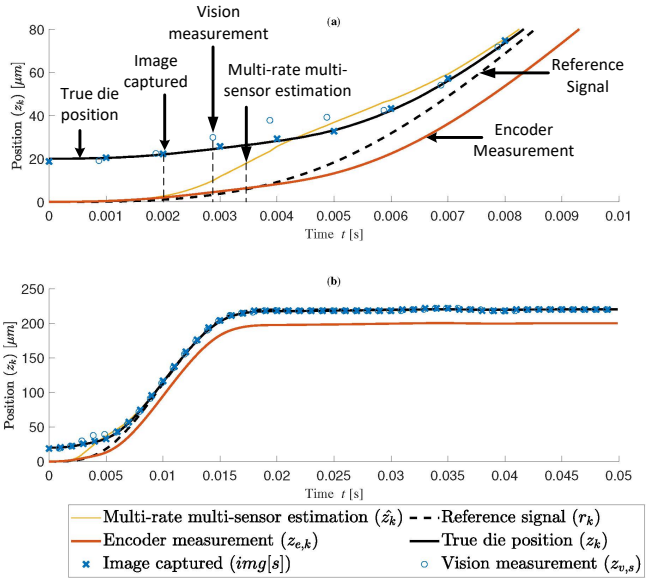

Fig. 7. Scenario 1 with $h_{vision} = 1ms$, and $h_{encoder} = 0.125ms$: position of the first die (a) zoomed in to 0-8ms (b) iteration 1: 0-50ms.

Fig. 8 shows the die position of the 5 dies under consideration. Clearly, for all the dies, the estimated position $\hat{z}_k$ is converging to the true die position with the iteration time of 50ms. This validates the effectiveness of the proposed method. The encoder does not measure the disturbances and hence, it shows a steady-state error at the end of each iteration. Fig. 9 illustrates the true position (closely measured by the vision measurement) along the x-axis after each iteration. Dies are numbered 1,2,3,4, and 5 for illustration. In Fig. 9, in the first iteration, the first die moves $(220 - 20) = 200\mu m$ by the x-axis motion of the wafer table. At the end of the first iteration (at 50ms), the second die also moved $200\mu m$ to the right and positioned at $(-205 + 200) = -5\mu m$. The same holds for the subsequent dies, i.e., the third die at $(-430 + 200) = -230\mu m$, the fourth die at $(-655 + 200) = -455\mu m$ and so on. In the second iteration, the camera focuses on the position of the second die which starts at $-5\mu m$ and reaches the target position $220\mu m$ at the end of the second iteration at 100ms. So, it requires the wafer table to move $225\mu m$ to the right. Therefore, at the beginning of the third iteration, the third die is at $(-230 + 225) = -5\mu m$, the fourth die is at $(-455 + 225) = -230\mu m$ and so on. This motion continues for all the subsequent dies. The true die positions (z_k) and encoder measurements ($z_{e,k}$) for each die are marked in Fig. 8 to correlate the die position movement explained in Fig. 9.

Table I shows the performance in all two scenarios. The SSE values are smaller than $1\mu m$ in almost all cases, although they would further vary based on the nature of the disturbances. In all the cases, the closed-loop system settles with 50ms iteration time. It is notable that the performance in terms of SSE, MAE and ST degrades when $h_{vision} = 8ms$ compared to the case with $h_{vision} = 1ms$. Therefore, it is desirable that the vision delay is reduced and the h_{vision} gets shorter.

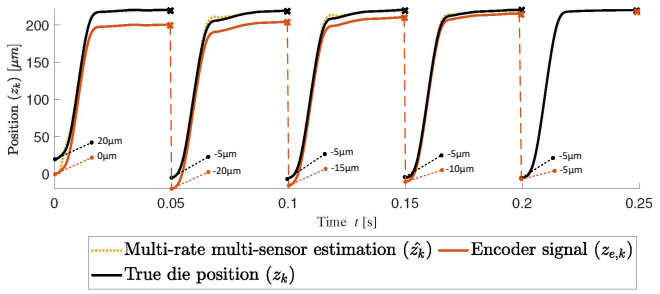

Fig. 8. Scenario 1 with $h_{vision} = 1ms$, and $h_{encoder} = 0.125ms$: position of the five dies, iteration 1: 0-50ms, iteration 2: 50-100ms, iteration 3: 100-150ms, iteration 4: 150-200ms, and iteration 5: 200-250ms.

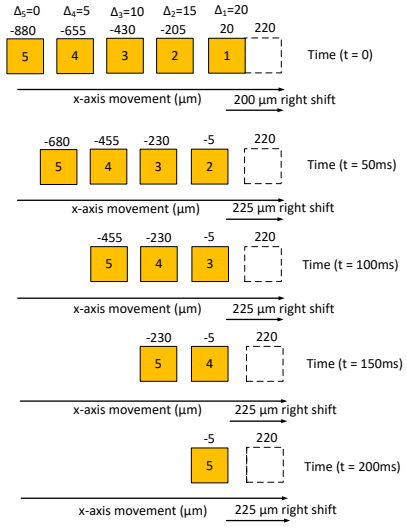

Fig. 9. Scenario 1 with external disturbances as the die position measured by vision sensor.

VIII. Conclusions

We presented the multi-rate multi-sensor fusion algorithm to improve the positioning accuracy of the industrial motion control systems. We fused the accurate, delayed and slow vision measurements with fast and noisy encoder measurements to achieve high-accuracy and high-speed position estimates. Further, we proposed bias correction, ensuring error correction due to the difference in measurement between the two sensors. We designed the performance evaluation framework for the entire closed-loop control system to evaluate the applicability of the proposed solution for the different scenarios by performing design-space exploration. More extensive design-space exploration with real implementation on hardware would be an interesting follow-up direction.

TABLE I

Closed-loop performance in different scenarios.

$h_{encoder}$ (ms)	h_{vision} (ms)		Die position 1	2	3	4	5
Scenario 1: $\Delta_1 = 20$, $\Delta_2 = 10$, $\Delta_3 = 15$, $\Delta_4 = 5$, and $\Delta_5 = 0$.							
0.125	1	SSE (μm)	0.177	0.300	0.296	0.419	0.062
		MAE (μm)	0.691	2.439	2.063	2.050	1.595
		ST (ms)	17.75	29.63	32.13	27.38	30.25
0.125	8	SSE (μm)	0.296	1.249	0.653	0.177	0.892
		MAE (μm)	1.639	2.971	2.171	2.154	1.839
		ST (ms)	30.63	45.38	35.13	28.13	35.50
Scenario 2: $\Delta_1 = 0$, $\Delta_2 = -5$, $\Delta_3 = -10$, $\Delta_4 = -15$, and $\Delta_5 = -20$.							
$h_{encoder}$ (ms)	h_{vision} (ms)		Die position 1	2	3	4	5
0.125	1	SSE (μm)	0.300	0.300	0.419	0.539	0.777
		MAE (μm)	0.035	0.975	2.087	3.206	4.458
		ST (ms)	17.63	17.75	42.25	41.63	42.255
0.125	8	SSE (μm)	0.300	0.777	0.539	0.181	1.016
		MAE (μm)	0.035	2.393	4.455	6.434	7.816
		ST (ms)	17.63	36.25	39.13	45.13	47.38

* All the deviations (Δ_i), where $i = \{1,2,3,4,5\}$ for 5 dies are in μm.

References

[1] J. Yu, X. Zheng, and J. Liu, "Stacked convolutional sparse denoising auto-encoder for identification of defect patterns in semiconductor wafer map," *Computers in Industry*, vol. 109, pp. 121–133, 2019.

[2] G. van der Veen, J. Stokkermans, N. Mooren, and T. Oomen, "How learning control supports industry 4.0 in semiconductor manufacturing," in *ASPE Spring Topical Meeting on Design and Control of Precision Mechatronic Systems*, 2020, pp. 1–5.

[3] S. Ariche, Z. Boulghasoul, A. Haijoub, A. Tajer, H. Griguer, and A. El Ouardi, "Object detection and distance estimation via lidar and camera fusion for autonomous driving," in *International Conference on Electrical Systems & Automation*. Springer, 2022, pp. 43–54.

[4] D. Lee and Y. Park, "Vision-based remote control system by motion detection and open finger counting," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 2308–2313, 2009.

[5] S. Mohamed, D. Goswami, S. De, and T. Basten, "Optimising multiprocessor image-based control through pipelining and parallelism," *IEEE Access*, vol. 9, pp. 112 332–112 358, 2021.

[6] S. Mohamed, "Multiprocessor image-based control: Model-driven optimisation," *Eindhoven University of Technology*, 2022.

[7] S. Mohamed, D. Goswami, V. Nathan, R. Rajappa, and T. Basten, "A scenario-and platform-aware design flow for image-based control systems," *Microprocessors and Microsystems*, vol. 75, p. 103037, 2020.

[8] Y. Liu, L. Guo, H. Gao, Z. You, Y. Ye, and B. Zhang, "Machine vision based condition monitoring and fault diagnosis of machine tools using information from machined surface texture: A review," *Mechanical Systems and Signal Processing*, vol. 164, p. 108068, 2022.

[9] Z. Zhong, Z. Hu, S. Guo, X. Zhang, Z. Zhong, and B. Ray, "Detecting multi-sensor fusion errors in advanced driver-assistance systems," in *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2022, pp. 493–505.

[10] M. Čech, A.-J. Beltman, and K. Ozols, "Digital twins and ai in smart motion control applications," in *IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022.

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[12] J. Gao and C. J. Harris, "Some remarks on kalman filters for the multisensor fusion," *Information Fusion*.

[13] Y. Liu and S. Zhou, "Detecting point pattern of multiple line segments using hough transformation," *IEEE Transactions on Semiconductor Manufacturing*, vol. 28, no. 1, pp. 13–24, 2014.

[14] D.A. Bristow and M. Tharayil, and A.G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[15] Wijdeven, van de, J.J.M. and O.H. Bosgra, "Using basis functions in iterative learning control : analysis and design theory," *International Journal of Control*, vol. 83, no. 4, pp. 661–675, 2010.

[16] A. C. Robotics, "Robot simulator CoppeliaSim: create, compose, simulate, any robot-coppelia robotics," 2020.

[17] C. Jugade, D. Hartgers, P. D. Anh, S. Mohamed, M. Haghi, D. Goswami, A. Nelson, G. van der Veen, and K. Goossens, "An evaluation framework for vision-in-the-loop motion control systems," in *ETFA*, 2022.

[18] P. Lambrechts, "Advanced Setpoints for Motion Systems," 2022, Accessed: 16-02-2022. [Online]. Available: https://nl.mathworks.com/matlabcentral/fileexchange/16352-advanced-setpoints-for-motion-systems