Proceedings of IMECE 2007
2007 ASME International Mechanical Engineering Congress and Exposition
November 11-15, 2007, Seattle, Washington, USA

IMECE2007-42890

DRAFT: NANOMETER RANGE CLOSED-LOOP CONTROL OF A STEPPER MICRO-MOTOR FOR DATA STORAGE

Mihai Patrascu*
Sensors and Actuators
IMEC-NL/Holst Centre
High Tech Campus 42
5656AE Eindhoven
The Netherlands
Email: Mihai.Patrascu@imec-nl.nl

Stefano Stramigioli
Meint de Boer
Gijs Krijnen
EWI-EEMCS
Universiteit Twente
P.O. Box 217
7500 AE Enschede
The Netherlands
Email: S.Stramigioli@ieee.org,
{M.J.deBoer, G.J.M.Krijnen}@ewi.utwente.nl

ABSTRACT

We present a nanometer range, closed-loop control study for MEMS stepper actuators. Although generically applicable to other types of stepper motors, the control design presented here was particularly intended for one dimensional shuffle actuators fabricated by surface micromachining technology. The stepper actuator features 50 nm or smaller step sizes. It can deliver forces up to 5 mN (measured) and has a typical range of about 20 µm. The target application is probe storage, where positioning accuracies of about 10 nm are required. The presence of inherent actuator stiction, load disturbances, and other effects make physical modeling and control studies necessary.

Performed experiments include measurements with open- and closed-loop control, where a positioning accuracy in the order of tens of nm or better is obtained from image data of a conventional fire-wire camera at 30 fps.

keywords: closed-loop, MEMS, micro stepper, FPGA, positioning, state diagram.

INTRODUCTION

The advent of probe storage as a potential future application has brought a diversity of research directions, some of them related to actuator design and positioning with extreme accuracy at the nanometer scale. The issue of combining sensors, actuators, and control circuitry is a great challenge in the world of MEMS devices [1]. MEMS positioning sensors typically have a low signal-to-noise ratio (SNR). At the same time actuators are limited in force, velocity, reliability, life time, and more. Designing integrated MEMS sensors and actuators is increasingly complex due to the compromises required at the fabrication stage. Typical control system problems involve the hardware for the control software and coping with the sensor SNR as well as (highly non-linear) actuator behavior. Altogether, closing the loop in MEMS has proven to be a hard task. A few successful examples of closed-loop MEMS-based systems can be found in literature [1,2]. In general however, one or more parts of the total system are not (yet) at the micro scale.

In the μSPAM (Micro Scanning Probe Array Memory) project carried out at the University of Twente, one of the targets is to achieve an accuracy of about $10\,nm$ in positioning. This is necessary for read and write tasks in novel data storage concepts [3–5]. Positioning speed is highly related to the requirements on data rate ($bits/s$) and to the distance between the bits on

*Address all correspondence to this author.

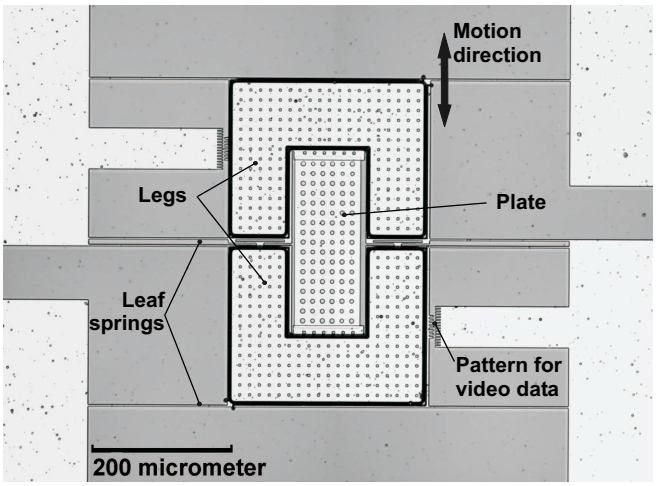

Figure 1. A top view of the micro stepper.

the recording medium. The current specifications in the μSPAM project indicate inter plate distances of several tens of *nm* and data rates of up to 10 $kbits/s$ for each read/write head. Translated to the performance of the micro actuator under attention, this results in actuation rates of a few $kstep/s$.

Initial efforts related to modeling, fabrication, and characterization of shuffle micro motors were reported in [6–8]. Physical modeling with lumped elements and bond graphs resulted in models where the impact of different inputs, dimensions, and material choices can be directly inspected [9–11].

This work is about controlling the micro stepper for read/write commands in one dimension. Actuators for two dimensional motion have been fabricated and tested [12, 13], but the accent here is on closed-loop control of a one dimensional shuffle motor, called the μWalker. This 1D device is the basic stone for the upcoming 2D actuators [14]. The first section of this article shortly introduces the electrostatic micro actuator under attention. Next, the experimental setup and the sub-micron video measurement algorithm are enlightened. Open and closed-loop control is the topic of the following section, after which the main results will be reviewed together with the outlook.

THE μWALKER

Geometry and dimensions

The micro stepper features bi-directional in-plane motion and was fabricated by surface micromachining. It has three basic parts, namely two U-shaped legs placed symmetrically and the actuator plate that connects them (Fig. 1). At each corner of the legs, long leaf springs prevent the device from continuously sticking on the walking surface. With this construction, stiction is avoided when the device is not actuated. An additional stiction limiting effort represents the presence of bumps texture on the contact region of the legs and plate. This texture limits the contact area between any of the three parts and the surface to only a few contact points instead of a plane, in which case the actuator would fail from stiction after the first contact with the surface underneath [7].

At the same time, the leaf springs act as conductors for the electrostatic legs and plate. When a potential is offered to either of the three basic parts via the corresponding spring(s), the electric field formed between the part and the surface at zero potential attracts it towards the surface. Pull-in instability may occur, which leads to snapping of the part onto the surface. To avoid short-circuiting and reduce wear, a hard, insulating silicon nitride layer has been added above the conducting ground electrode. The three basic parts are mechanically connected but electrically they are isolated using trench isolation technology [8].

In the following, the walking principle is shortly discussed (Fig. 2). When a sufficiently high voltage is applied to one leg, this is attracted onto the surface. The vertically generated contact friction on the leg prevents it from shifting sideways when actuated. The stiction force of the leg depends amongst others on the applied potential, the contact area, the surface roughness, and material hardness.

A potential offered to the plate bends it downwards. This vertical bending of the plate decreases the distance between the two ends causing one step, which is between 0 and about 100 *nm* depending on the leaf springs load condition, the direction, and the voltage actuation. In general, the size of one step is about 50 *nm*. Steps smaller than the standard step can be obtained reproducibly by choosing different actuation schemes than the standard one presented in Fig. 2 [15]. Indeed, smaller steps are necessary for obtaining positioning accuracies in the order of 10 *nm*.

Now, by creating a set of input signals for the legs and the plate like for instance depicted in Fig. 2, a displacement can be obtained in either direction. Repeating this sequence results in displacements larger than one step and up to a range of a few tens to hundreds of μm, depending on specific actuator design and especially on the leaf springs. Although a various number of designs have been successfully fabricated and tested lately [13], only the basic μWalker geometry and dimensions are presented (Fig. 3). Table 1 reveals some relevant dimensions.

Performance

In [11] it has been shown that the useful life time of such a micro positioner may be as large as 4.5 billion steps, a unique performance compared to other stepper motors of similar size. Typical stepping speeds vary from less than 1 step to more than 100.000 steps per second, with typical step sizes varying around 50 *nm*.

To give a quantitative impression of the μWalker reproducibility, repetitive measurements of 20 steps in one direc-

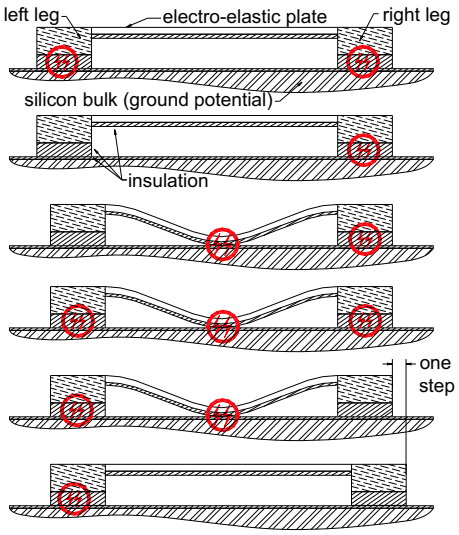

Figure 2. Six changes of the three inputs produce one step.

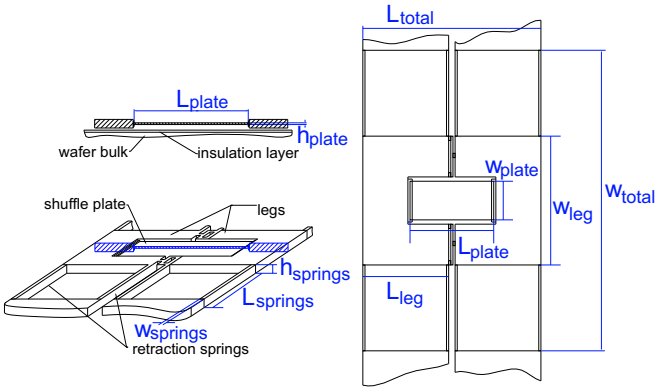

Figure 3. Principal micro-motor shape and dimensions.

tion were conducted (open-loop). The total mean distance was 1.033 μm and the standard deviation amongst the 100 samples was only 6.52 nm [13].

Another important performance factor is the generated force. The highest force exerted on the springs by a typical μWalker is about 2.3 mN. Increased plate and legs surface increases the maximum force up to 4.8 mN [13], which is again highly remarkable in the world of electrostatic micro actuators. None of the above is pure coincidence, as it has been indicated that the μWalker has the highest generated force density among the electrostatic microactuators [7].

Table 1. Dimensions of the basic μWalker type in μm.

L_{total}	overall device length	440
w_{total}	overall width	686
h_{total}	overall height	5.5
L_{leg}	leg length	216
w_{leg}	leg width	286
L_{plate}	plate length	208
w_{plate}	plate width	94
h_{plate}	plate height	1.2
$h_{springss}$	retraction springs height	5.5
$L_{springs}$	retr. springs length	200
$w_{springs}$	retr. springs width	2
d_{leg}	distance leg surface	0.33
d_{plate}	distance plate surface	1.88

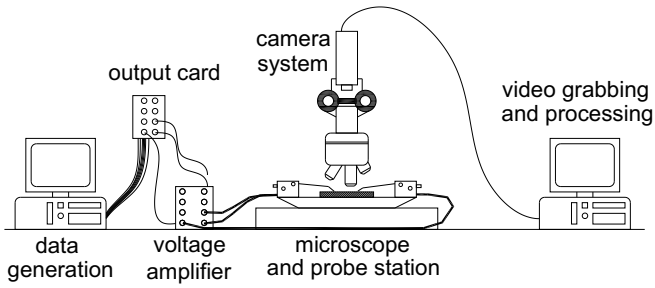

Figure 4. Sketch of the actuation and measurement setup.

MEASUREMENT SYSTEM

Experimental Setup

Once a data file with the desired motion has been generated, it is being translated into signals for the three channels by a D/A data output card or by an FPGA (Fig. 4). The latter has the advantage that state diagrams for hybrid control may be incorporated directly. The channel signals are amplified by a high-end amplifier to potentials around 30-50 V for proper actuation of the micro stepper legs and plate. A probe station electrically connects the amplifier channels to the μWalker under the microscope.

It is useful to enlighten the method used for measuring position with sub-micron accuracy, as this could be applied without much additional effort to various other types of displacement measurements. The measurement part consists of a standard b/w fire-wire camera on top of the microscope, acquiring data at 30 fps. Image data processing is done with a number

of Matlab® scripts. The technique resides on recording a series of regularly patterned light and dark regions and then measuring the phase shift of the patterns of subsequent frames. This shift is directly proportional to the original displacement and can give results as accurate as 10 to 15 *nm*, depending on the number of regular patterns, video sharpness, vibration isolation of the setup, and other aspects. For each frame, the calculated position is saved to a text file which may be used later for further processing.

OPEN AND CLOSED-LOOP CONTROL

Open-Loop Control

Even if physical models and simulations have been used throughout the research, for performance increase an open-loop relying on experimental data was implemented.

The main reason is that at this point we do not seek to study parameter uncertainties in models. Instead, we intend to show how closed-loop systems help coping with problems related to irregular stiction of μWalker components.

A very crucial point is the relation between step size on one hand and position and walking direction on the other hand. When the μWalker is off-center, the walking direction has an influence on step size due to loading of the retraction springs. This is an effect of the positive definite behavior of the retraction springs' potential energy and basically means that the stored potential energy either increases step size – when the energy potential decreases with time, or it decreases the step when going 'up-hill' energetically spoken. The step size as a function of displacement can best be obtained by running the actuator from one end to the other end of the range (Fig. 5) and thereby deducing the step size at each point with respect to the center position.

Based on this information, we parameterized a linear fifth order polynomial by LSQ optimization, i.e. of the form:

$$s(d) = \sum_{n=0}^{5} \alpha_n d^n \qquad (1)$$

where d is the measured off-center displacement and $s(d)$ is the estimated step size. For the basic μWalker, the α vector is:

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{bmatrix} = \begin{bmatrix} -5.1770 \cdot 10^{-2} \\ +4.2190 \cdot 10^{-4} \\ -6.1314 \cdot 10^{-5} \\ -9.5807 \cdot 10^{-5} \\ +1.4910 \cdot 10^{-6} \\ +3.5155 \cdot 10^{-7} \end{bmatrix}.$$

Figure 6 shows both the measurement data points and the obtained approximation. The rotated S-shape of the relation between step size and position is typical for the leaf springs construction.

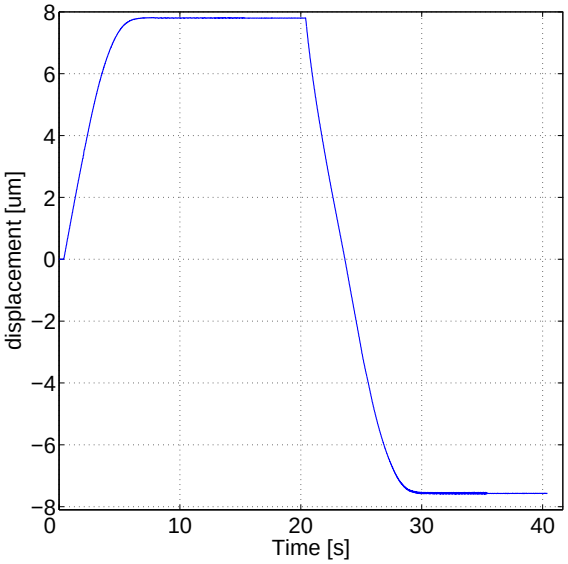

Figure 5. Measurement of the actuator position as function of time.

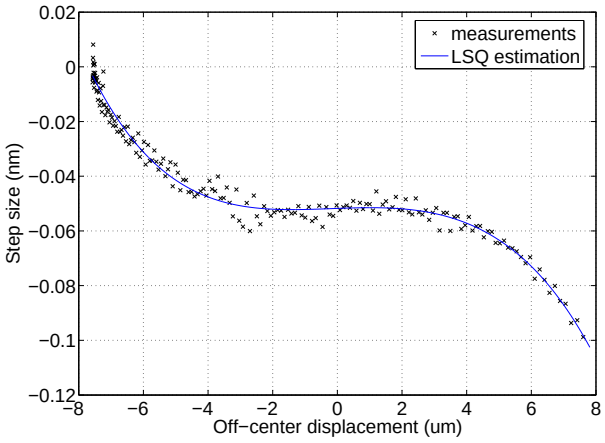

Figure 6. Measured and estimated step size as a function of position and direction.

To investigate the validity of the proposed open-loop model, a test sequence was composed (Fig. 7). The intended motion is composed of scanning six different positions, followed by a scan covering the whole range. The measurement time was chosen to be about 17 *s* in order to have enough video frames during each part of the motion. For the same reason, the number of steps per

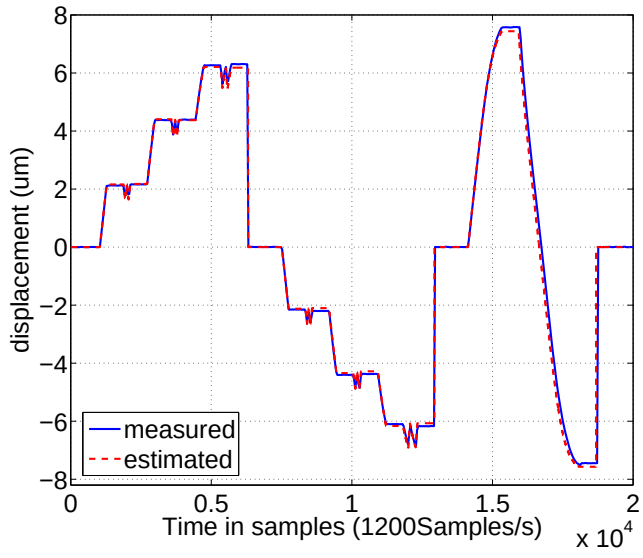

Figure 7. Test sequence: open-loop estimation and measurement.

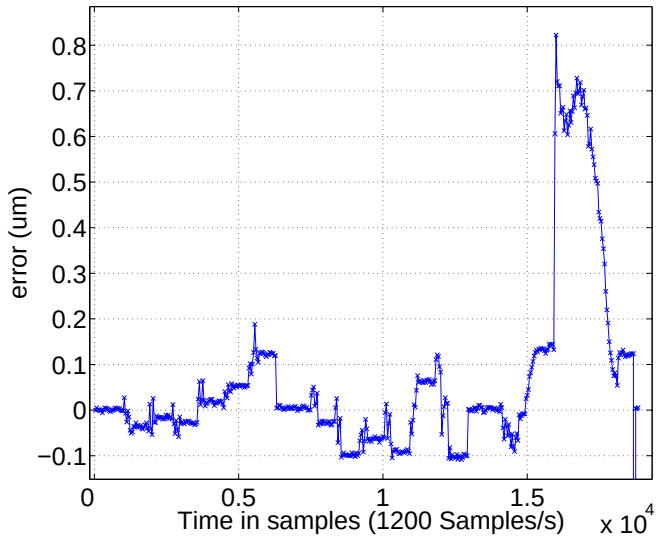

Figure 8. Discrepancies between measured and estimated motion.

second was limited to 200 (1200 $Samples/s$ and 6 $Samples/step$ required).

Figure 8 shows the error between the expected position according to the open-loop model and the measured position. Understanding the data in this plot requires some explanation. First of all, the sampling frequency of the measurements equals the video system frame rate (30 $Samples/s$), whereas the model output is being estimated with each sample, in this case at 1200 $Samples/s$. Second, random frame drops induce discrepancies between the real position and the logged position. A relatively small delay introduces error peaks up to a few hundred of *nm*, see for instance Fig. 8, right from $1.6 \cdot 10^4$ *Samples*.

Assuming that the sampling frequency and delays are not being considered, then the actual error is at most 140 *nm* and normally smaller than 100 *nm*. Related to the total range of 15.38 μm, the maximum error is only 0.91 %.

In practice, we sometimes encountered problems related to irregular legs stiction onto the surface, which prevent the feed-forward model from being sufficiently accurate and consequently rose the need for feedback control.

Closed-Loop Control

A higher performance can be achieved when implementing closed-loop control, based on the inverse feed-forward model and on video data from the fire-wire camera. Basically, in this way the positioning accuracy can be reduced down to the limits of the position detection algorithm, about 10 *nm* in our case.

One of the particularities related to the closed-loop control of the μWalker MEMS actuator is the presence of different bandwidths in one control loop. This property seems to be rather specific for closed-loop positioning of MEMS stepper motors and other mechanically actuated systems with very small dimensions and high non-linearities like for instance the presence of pull-in. Partly, this property is present due to the high eigen-frequencies of the plant and the relatively small step sizes, which require a large rate of steps per unit time in order to meet the required performance. On the other hand, the sensing hardware – be it a camera, a MEMS capacitive sensor or something else – is limited to far below the required actuation rate, so that no 'ordinary' feedback with only one bandwidth in the loop for both actuation and sensing can be used for reasonable positioning performance.

In this respect, the closed-loop control scheme could be extended in a straightforward way to steppers featuring any type of position sensors with limited data output. Especially in the micro and nano world, the discrepancy between the sensing and actuation bandwidth is striking. The controller implementation shown here becomes a necessary (but not sufficient) condition for obtaining optimal performance under such circumstances.

In the following, we present a more detailed description of the controller implementation, especially of the state machine. Figure 9 illustrates the blocks of the closed-loop system. The reference signal is the desired μWalker position trajectory in time, an arbitrary scanning sequence like encountered in read/write systems. A handshake protocol between software (20Sim®) and hardware (FPGA) ensures that the error signal is only considered when the micro motor is in a hold state and position data from the camera is available. The error signal, namely the difference between the reference and measured position, is being converted

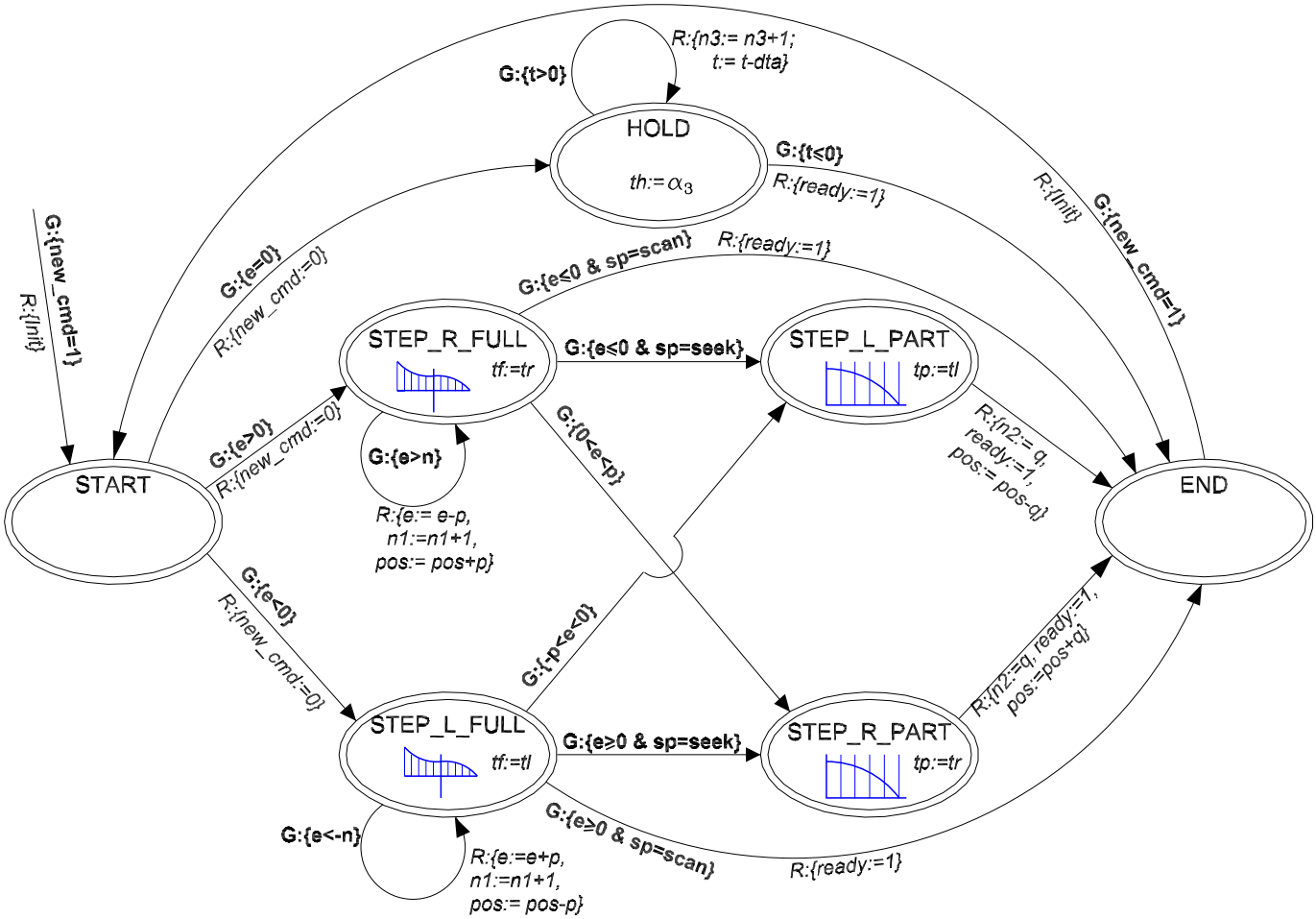

Figure 10. State diagram for closed-loop control, the main automaton.

to a command including the number of steps to be taken and the direction. The use of a state diagram to derive this command has been considered. According to Fig. 10, seven modes were defined for the automaton:

START: start mode,
STEP_L_FULL: full step(s) to the left,
STEP_R_FULL: full step(s) to the right,
STEP_L_PART: partial step to the left,
STEP_R_PART: partial step to the right,
HOLD: hold position,
END: end the motion sequence.

The full-step mode derives the number of steps and the direction of motion from the error and the desired position.

In addition to a sequence of full steps, a partial step may be required in order to achieve the desired position with higher accuracy. The method is presented in [15], where it has been shown that the step size can be decreased in a deterministic way from maximum (50*nm*) down to about 10*nm* by changing the actuation sequence.

Feed-forward data about step dependency on position (Fig. 6) and information of the sub-steps have been quantized in a number of points. Each of these points corresponds to a specific table in the FPGA, which represents the three-channel signals necessary for the required motion.

Having a closer look at Fig. 10, G:{...} represent the guards which are present at the transition between two modes. Generally, the guards test the value of the position error e or of some other state (see Eq. 2) and are a condition for jumping between one mode and another. R:{...} are resets which change elements in the state vector during a mode transition.

The initial state vector of this automaton is formed by a num-

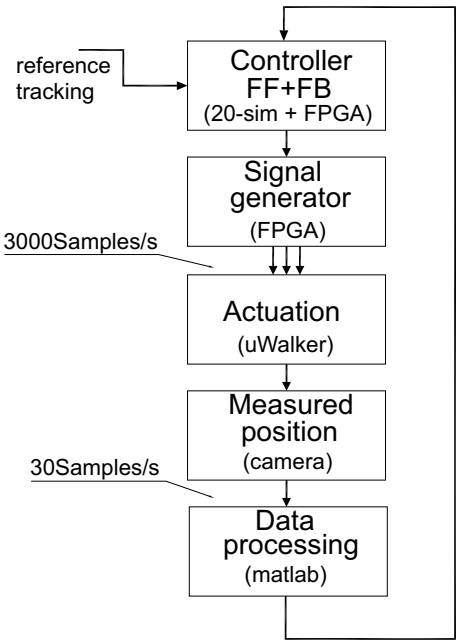

Figure 9. Schematics of the proposed control loop.

ber of states, which are described next:

$$\begin{bmatrix} t_f \\ t_p \\ t_h \\ dir \\ sp \\ pos \\ t \\ e \\ dta \\ ready \\ n_1 \\ n_2 \\ n_3 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \\ \alpha_9 \\ \alpha_{10} \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ meaning } \begin{bmatrix} \text{pointer table full steps} \\ \text{pointer table partial steps} \\ \text{pointer table hold} \\ \text{direction of movement} \\ \text{speed} \\ \text{position} \\ \text{motion sub-part duration} \\ \text{position error signal} \\ \text{sample duration} \\ \text{inform higher automaton} \\ \text{nr. of full steps} \\ \text{type of partial step} \\ \text{nr. of hold steps} \end{bmatrix} \quad (2)$$

Variables t_f and t_p of the state vector have three possible values, namely $\alpha_1, \alpha_2 \in \{0, t_r, t_l\}$. The three possibilities mean: no step action, do step(s) to the right and do step(s) to the left, respectively.

State t_h is a boolean for the HOLD mode ($\alpha_3 \in \{0,1\}$). The state variables *dir* and *sp* represent the motion direction and magnitude of the velocity. They can be derived by performing a *sign* test on the desired velocity function ($\alpha_4 \in \{-1,0,1\}$) and by a readout of the magnitude ($\alpha_5 \in \{hold, seek, scan\}$). *pos* is the momentary estimated position; initially, $\alpha_6 := \mathring{x}(t_{start})$, where t_{start} is the first data point of the sub-sequence under consideration. As the number of steps is calculated, *pos* decreases below

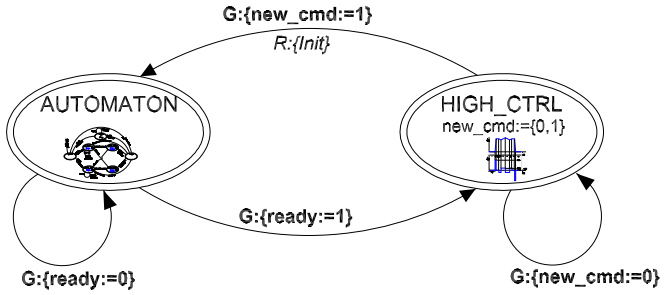

Figure 11. Higher order automaton for generating motion patterns.

one step size. Next, by implementing a trigger on the same velocity plot, we can discriminate between different sequences and define a time t for each sequence ($\alpha_7 \in \mathbf{R}^+$).

The closed-loop part comes in the model with variable e which is initialized as α_8 and represents the position error. This is the difference between the reference tracking position $\hat{x}(t)$ and the measured position $\mathring{x}(t)$:

$$\alpha_8 = \hat{x}(t) - \mathring{x}(t), \text{ with} \quad (3)$$

$$\mathring{x}(t) = \begin{cases} \tilde{x}(t) & : \text{ last sequence was seek} \\ \hat{x}(t-1) & : \text{ otherwise} \end{cases} \quad (4)$$

State variable *dta* stands for the time required to run one table:

$$\alpha_9 = \frac{\text{samples per table}}{\text{output samples per second FPGA}}.$$

During experiments, $\alpha_9 \approx \frac{12}{1200} = 0.01s$ such that 200 steps per second are being generated. The reason for choosing only 200 steps steps per second, or about 10 $\mu m/s$, has to do with the sampled video data. Much larger velocities could be interpreted erroneously by the position detection algorithm.

The *ready* signal informs a higher order automaton (Fig. 11) that the main automaton is ready for the next part of the signal ($\alpha_{10} \in \{0,1\}$). Finally, the n_1, n_2, and n_3 define the number of full steps, the specific table to be used for the partial step and the number of hold sequences.

The higher order automaton in Fig. 11 is used to restart the main state machine with the right initial vector depending on the input (reference tracking). As soon as the *new_cmd* is tested high by the main automaton (AUTOMATON), the states are being reset and a new part of the tracking signal is transformed into table rows for the FPGA output. The commands are thus stored in a buffer of the FPGA, where they wait for execution. The *ready* signal informs the HIGH_CTRL machine that the main automaton, AUTOMATON, is ready for the next part of the motion. A new state vector is being generated and the main automaton is restarted.

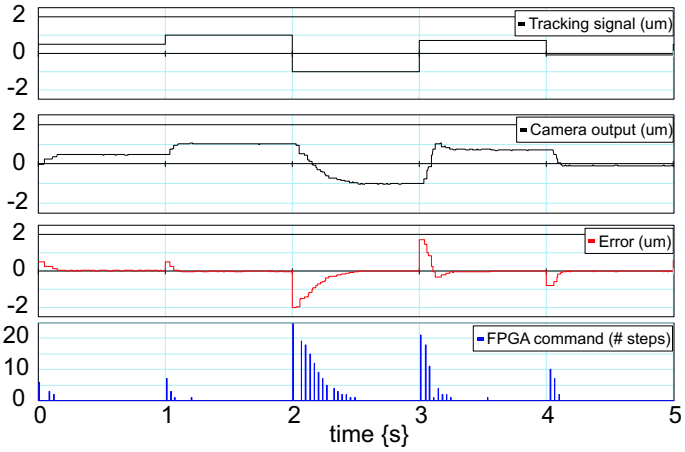

Figure 12. From top to bottom: tracking signal, measured position, error and FPGA output.

Once the type and number of steps have been derived, an output table is generated with all the signals necessary for the requested motion and in chronological order. Subsequently, this data is transformed into output signals on the three channels by the signal generator. The signals pass via a D/A converter (one per channel) and an amplifier, before reaching the μWalker on the wafer.

Figure 12 shows the first results from this closed-loop implementation. The FPGA output rate was set at 3.0 $kSamples/s$ during measurements. The top signal in the figure is the desired tracking signal, generated a-priori and composed of quantized step functions. The measured position from the camera is the second plot. The positioning error is shown in the third plot. The error between the desired and measured position indeed decreases with time and finally disappears. A reduced amount of overshoot is present three seconds after starting the measurement, which could be related to stiction effects. The presence of a non-zero error triggers the state machine to generate correction signals every 33 *ms* in the form of a number of steps into one direction. The lowest plot shows the FPGA output, namely the number of steps expected to reduce the error to zero.

The presence of stiction and other uncertainties at device level explains the fact that instead of one or two corrections, more FPGA commands are necessary to reduce the error. The main physical reasons for the stiction are low humidity, worn-out μWalker legs and extended actuation cycles that lead to surface insulator charging and thus stiction. Especially the start-up period of each change in position seems to be impeded by stiction during the measurements shown.

Note that the positioning accuracy is not being influenced by the camera frame rate. The closed-loop settling time does depend linearly on the frame rate, so higher frame rates would yield smaller settling times. The number of steps with each corrective action (Fig. 12, lowest plot) is not expected to change beyond statistical variation, unless stiction phenomena at the contact surfaces show time dependencies at about the same scale as the frame update rate.

Clearly, it has been shown that in the presence of large uncertainties, the closed-loop effectively reduces the positioning error beyond what would be achievable with open-loop control.

CONCLUSIONS AND OUTLOOK

After a concise introduction to an electrostatic micro actuator for data storage, open and closed-loop control have been demonstrated. A combination of a fire-wire camera and a dedicated position detection algorithm was used to acquire the actuator position with 10 *nm* accuracy each 33 *ms*. For high speed positioning, open-loop control may be the best option, provided a valid model describing the step size and reduced model uncertainties. To diminish the impact of unintended stiction and other effects which cannot be modeled in a feed-forward way, closed-loop control has shown to successfully reduce the error between the desired and the real position. At the heart of the control loop lies a state machine that serves both as feedback and feed-forward discrete controller. Using closed-loop control, the positioning accuracy and settling time are mainly limited by uncertainties, the video grabbing hardware, and the positioning algorithm. The state machine closed-loop control presented in this paper may be used for other systems of arbitrary size, where the sensing bandwidth is (much) lower than the actuation bandwidth.

Future work will concentrate on further investigating the multiple correction actions required by closed-loop control. From an analysis point of view, the closed-loop stability of switched systems like the μWalker is not trivial but needs to be investigated.

In a more general context, MEMS sensors are being sought that will replace the camera setup for reaching the micro-scale sizes required of the total system. For the final target of positioning in probe storage, positioning and closed-loop control of two dimensional structures will be considered [13, 14].

ACKNOWLEDGMENT

The authors thank Edin Sarajlic for the initial design of the μWalker and Srinivas Vanapalli for the software implementation of the sub-pixel position detection algorithm. Marten Lootsma and Marcel Schwirtz are acknowledged for finalizing the closed-loop software and hardware implementation. Our thank also goes to IMEC-NL for enabling the writing of this publication. The μSPAM project is funded by STW (TES 5178).

REFERENCES

[1] Borovic, B., Liu, A. Q., Popa, D., Cai, H., and Lewis, F. L., 2005. "Open-loop versus closed-loop control of MEMS devices: choices and issues". *Journal of Micromechanics and Microengineering*, **15**, pp. 1917–1924.

[2] Pantazi, A., Sebastian, A., Pozidis, H., and Eleftheriou, E., 2005. "Two-sensor-based H-inf control for nanopositioning in probe storage". *Proceedings of the 44th IEEE Conference on Decision and Control*, December, pp. 1174–1179.

[3] Carley, L. R., Bain, J. A., Fedder, G. K., Greve, D. W., Guillou, D. F., Abelmann, L., and Min, S., 2000. "Single-chip computers with microelectromechanical systems-based magnetic memory (invited)". *Journal of Applied Physics*, **87**(9), 05, pp. 6680–6685.

[4] Bolks, M., Hanssen, F., Abelmann, L., Havinga, P., Hartel, P., Jansen, P., Lodder, C., and Smit, G., 2001. "Micro Scanning Probe Array Memory (uSPAM)". *Proceedings of the Second Progress Workshop, Veldhoven, The Netherlands*, pp. 17–26. ISBN: 90-73461-26-X.

[5] Vettiger, P., Cross, G., Despont, M., Drechsler, U., Dürig, U., Gotsmann, B., Häberle, W., and e.a., M. A. L., 2002. "The "Millipede" - nanotechnology entering data storage". *IEEE Trans. Nanotechnol.*, **1**, pp. 39–55.

[6] Tas, N., Wissink, J., Sander, L., Lammerink, T., and Elwenspoek, M., 1998. "Modeling, design and testing of the electrostatic shuffle motor". *Sensors and Actuators A*, **70**, pp. 171–178.

[7] Sarajlic, E., Berenschot, E., Tas, N., Fujita, H., Krijnen, G., and Elwenspoek, M., 2005. "High performance bidirectional electrostatic inchworm motor fabricated by trench isolation technology". *TRANSDUCERS 05 - Int. Conf. on Solid State Sensors and Actuators - Seoul, Korea*.

[8] Sarajlic, E., Berenschot, E., Krijnen, G.J.M., Elwenspoek, M., 2003. "Versatile trench isolation technology for the fabrication of microactuators". *Microelectronic Engineering*, **67-68**, *pp. 430-437*

[9] Patrascu, and Stramigioli, S., 2006. "Modeling and simulating the stick-slip motion of the μWalker, a MEMS based device for μSPAM". *Microsystem Technologies*, **161**, pp. 1432–1858.

[10] Ligterink, N. E., Patrascu, M., Breedveld, P. C., and Stramigioli, S., 2005. "An energy-based electroelastic beam model for MEMS applications". *Sensors and Actuators A: Physical*, **121**(2), pp. 500–507.

[11] Patrascu, M., and Stramigioli, S., 2006. "Characterization of stiction effects of an electrostatic micro positioner for probe storage". *Journal of Physics: Conference Series, Institute of Physics (IOP)*, **34**, pp. 818–823. online: http://www.iop.org/EJ/abstract/1742-6596/34/1/135.

[12] Sarajlic, E., Berenschot, E., Fujita, H., Krijnen, G., and Elwenspoek, M., 2005. "Bidirectional electrostatic linear shuffle motor with two degrees of freedom". *MEMS 2005 - 18th IEEE International Conference on Micro Electro Mechanical Systems, Miami*, January, pp. 391–394.

[13] Patrascu, M., 2006. "Characterization, modeling and control of the μWalker – a micro actuator for data storage". PhD thesis, University of Twente, September. ISBN:90-365-2398-2.

[14] Patrascu, M., Stramigioli, S, De Boer, M., Krijnen, G.J.K., 2006. "Electrostatic micro-actuators for probe storage". *Proceedings of PPTC, Hilvarenbeek*, October, pp. 115-116. Sarajlic, E., Berenschot, E., Krijnen, G.J.M., Elwenspoek, M., 2003. "Versatile trench isolation technology for the fabrication of microactuators".

[15] Patrascu, M., and Stramigioli, S., 2005. "Stick-slip actuation of electrostatic stepper micropositioners for data storage- the uWalker". *ICMENS*. See also URL `http://csdl2.computer.org`.