Mixed-Signal Odometry for Mobile Robotics

Michael J. Kuhlman, Tsung-Hsueh Lee, and Pamela A. Abshire

Department of Electrical and Computer Engineering and Institute for Systems Research
University of Maryland, College Park, U.S.A.

ABSTRACT

Miniature robots present a number of challenging problems in controls, as they often exhibit nonlinear dynamics and have strict power and size constraints. These constraints limit the sensing and processing capabilities drastically. Many control techniques require knowledge of the robot's position, so the position must be estimated when it cannot be sensed directly. We report a mixed signal odometry circuit that maps motor commands to estimated and predicted changes in position in Euclidean space (x, y, θ). We compare the mixed-signal implementation with other approaches and find that the mixed-signal implementation offers significant reductions in power consumption at an acceptable loss of precision.

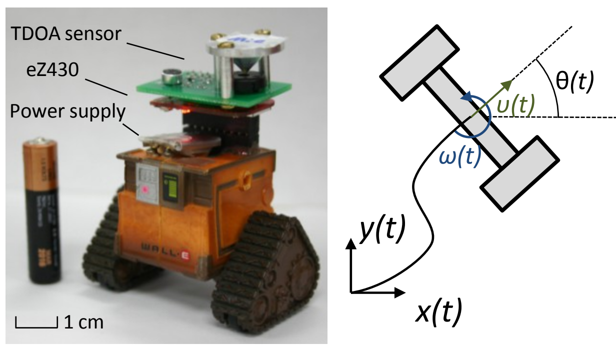

Figure 1. Left: Miniature robot platform. AAA battery shown for reference. Right: mobile robot's state space (x, y, θ) and control space (υ, ω).

1. INTRODUCTION

Advances in sensing, actuation and battery technology have allowed for the development of very small robots (sub-cm^3).[1–3] Autonomous operation generally requires local modeling of system dynamics, since many control strategies require knowledge of the system state and direct realtime sensing of position is not always possible due to size and power constraints. Further, Model Predictive Control techniques such as Randomized Receding Horizon Control require many predictions of the robot's future state ($10^4 - 10^5$ predictions to be done in faster than real time) before executing a given command sequence.[4] On-board computation for small platforms is limited by available micro-controllers, which are relatively large, power-hungry, and slow. To alleviate this challenge we propose a mixed-signal architecture implementing an odometry function that maps motor commands to estimated changes in position using a kinematic model. The architecture is designed to support control of a differential-drive miniature robot such as the one shown in Fig. 1.

1.1 Simulation of System Dynamics

The inputs to the kinematic model are assumed to be the left and right motor commands, u_L and u_R, or the linear and angular control variables, υ and ω. Assuming a kinematic drive-train model, these are linearly related as $\begin{bmatrix} \upsilon \\ \omega \end{bmatrix} = \begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix} \begin{bmatrix} u_L \\ u_R \end{bmatrix}$. The outputs of the kinematic model are the coordinates (x, y, θ) from the nonlinear equations of motion (1).

Further author information: (Send correspondence to P.A.A.)
P.A.A: E-mail: pabshire@umd.edu, Telephone: 1 301 405 6629

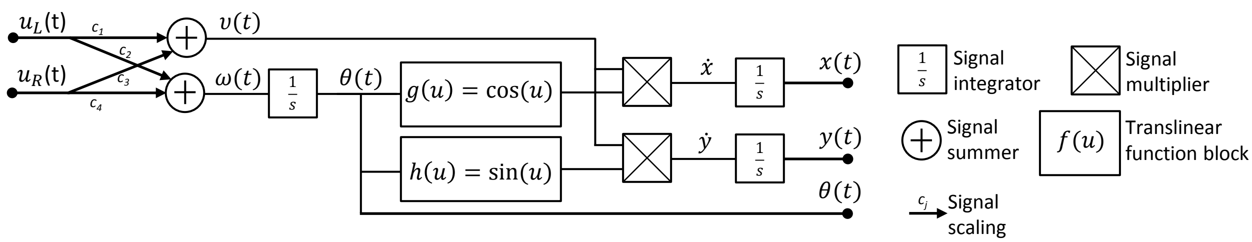

Figure 2. System-level design of analog robot kinematics simulator highlighting signal flow and primary computational blocks.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \upsilon \cos\theta \\ \upsilon \sin\theta \\ \omega \end{bmatrix} \tag{1}$$

Alternatively, closed form solutions of (1) can be implemented directly for piecewise constant (υ, ω) (2):

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{\upsilon_k}{\omega_k}(\sin\theta_{k+1} - \sin\theta_k) + x_k \\ \frac{\upsilon_k}{\omega_k}(\cos\theta_k - \cos\theta_{k+1}) + y_k \\ \omega_k(t_{k+1} - t_k) + \theta_k \end{bmatrix} \tag{2}$$

In Section 2 we describe the system architecture and its main components, and in Section 3 we describe the results of system level simulations. Section 4 summarizes the work.

2. SYSTEM OVERVIEW

We describe an odometry circuit that solves (1) directly rather than compute (2). This approach requires fewer operations and thus fewer circuit stages to model the nonlinear system dynamics: 4 signal scaling elements, two summing nodes, three integrators, two trigonometric function blocks and two signal multipliers (see Fig. 2). Inputs may be presented to the system as either motor commands $u_L(t)$ and $u_R(t)$ or control variables υ and ω. For a physical interpretation of these control variables, refer to Fig. 1 First, ω is integrated to provide an estimate of the angle θ, then the sine and cosine of the angle are computed. Next the product of $\cos(\theta)$ and the linear velocity υ is integrated to find the position x, and the product of $\sin(\theta)$ and υ is integrated to find the position y.

2.1 Integration of θ and State Calibration

Integration of ω produces an estimate of the angle θ. This is accomplished using differential current-mode signals by sourcing the positive component of the signal onto a node with capacitance C_θ, and sinking the negative component away from the same node. This results in $\Delta I_\omega = I_{\omega_+}(t) - I_{\omega_-}(t)$ and $V_\theta(t) = \frac{1}{C_\theta}\int \Delta I_\omega dt$.

It is also of great importance to calibrate circuit timing, slew rates, and reference angles from robot state trajectories X (in real time) to circuit voltage signals V and differential current signals ΔI (in faster than real time). The linear calibration map can be generated from the properties in (3):

$$Cal : (t_{wor}, X, \dot{X}) \leftrightarrow (t_{cir}, V_X, \dot{V}_X) \tag{3}$$

$$\begin{aligned} t_{wor} &= \tau t_{cir} && \text{time warping} \\ \frac{\frac{dV_\alpha(t_{cir})}{dt_{cir}}}{|SR_{\alpha,max}|} &= \frac{\frac{d\alpha(t_{wor})}{dt_{wor}}}{|\dot{\alpha}_{max}|} && \text{calibration of slew rates to state differentials} \\ \theta(t_{cir}) &= \frac{dV_\theta(t_{cir})}{dt_{cir}} \frac{4\pi}{V_{max} - V_{min}} V_\theta && \text{angle calibration for trigonometric circuits (linear phase)} \end{aligned}$$

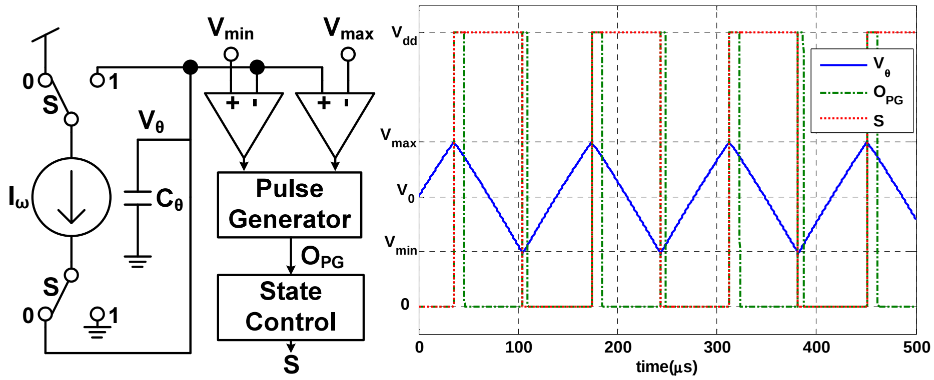

Figure 3. System-level design of integration and state control circuit.

where $\alpha \in \{x, y, \theta\}$, and $X = (x, y, \theta)$. We also assume that $|SR_{\alpha,max}| = \frac{|\Delta I_{\alpha,max}|}{C_\alpha}$ and that (v, ω) are piecewise constant. To calculate the mapping between states X and V_X, the key is to integrate both sides of the slew rate calibration equation in (3) with respect to the *same* time reference frame. Care must be taken to use the other calibration substitutions.

The capacitor size and current range set the time scale. For current mode signals with amplitudes between 1nA and 100nA, the dimensionless time scale constant τ defines how many seconds of world time are simulated in one second of the circuit's operation. In this implementation, the maximum rotation rate was set to $|\Delta I_{\omega,max}| = 100nA \leftrightarrow |\omega_{max}| = 2\pi \frac{\text{rad}}{\text{s}}$.

By looking at the definition of V_θ, it is possible to relate $\theta(t_{wor})$, the estimation of the robot's angle in the physical world, to $V_\theta(t_{cir})$. One can use (3) to establish the linear relationship mapping modeled system states to circuit states (4) (ignoring initial conditions):

$$V_\theta(t_{cir}) = \frac{|\Delta I_{\omega,max}|}{C_\theta |\omega_{max}|} \frac{\theta(t_{world})}{\tau} \tag{4}$$

Notice that (4) is dimensionally self-consistent. Increasing τ reduces C and thus the chip area required for the capacitor. For this design, we assigned $\tau = 10^5$ and designed $C \approx 0.1\text{pF-}1\text{pF}$. Slew rate measurements estimate the node capacitances to be $C_\theta = 2\text{pF}$, $C_x = 0.4\text{pF}$ and $C_y = 0.4\text{pF}$.

2.2 Circuit Implementation of the Modulus Circuit

State	Desired mapping $\theta' \mapsto \theta$	sine	cosine
S = 0	$\theta = \theta'$	$\sin\theta = \sin\theta'$	$\cos\theta = \cos\theta'$
S = 1	$\theta = 2\pi - \theta'$	$\sin\theta = -\sin\theta'$	$\cos\theta = \cos\theta'$

Table 1. States of operation for $\theta' \mapsto \theta$

We define the trigonometric shaping circuits over two periods in order to avoid problems associated with discontinuities at the boundaries. We defined two states of operation as summarized in Table 1. The integration and state machine circuits are shown in Fig. 3. In state S=0, both switches (controlled by state) flip to the left and ΔI_ω is sourced to capacitor C_θ. Here we assume that the robot can only rotate in one direction. Therefore, ΔI_ω is always greater than or equal to zero and V_θ is monotonically increasing and decreasing in state S=0 and S=1 respectively. In state S=1, both switches flip to the right and ΔI_ω is drained from capacitor C_θ. In this state, a mapping is required to compute the correct θ; this mapping is accomplished by flipping the differential output of the sine shaping circuit (see Table 1). However, the switching of state causes undesired coupling and drift which we will discuss in section 3.2.

The state machine works as follows. The integrated voltage V_θ is compared to two thresholds using comparators. When V_θ crosses a threshold (V_{min} or V_{max}, which correspond to $\pm 2\pi$), the output of the corresponding comparator rises high. This rising signal is fed into a pulse generating circuit, converting the rising signal into a pulse. The rising edge of the pulse toggles the state control circuit and changes the state as shown in the example waveforms of Fig. 3.

2.3 Sine Shaping Circuit

A sine shaping circuit maps a (DC) voltage to its sine, i.e. $V_\theta \mapsto \sin(V_\theta)$. Series expansion using hyperbolic tangents[5] offers the capability to define a valid approximation over an arbitrary range. We use the hyperbolic transfer function (5) of MOSFET differential pairs operating in weak inversion:

$$\Delta I_{out} = I_B \tanh\left[\frac{\kappa}{2U_T}\left(V_\theta - V_{ref}\right)\right] \tag{5}$$

to implement this circuit. Connecting five differential pairs in parallel (see Fig. 4) extends the validity of the sine approximation over $\pm 2\pi$. Self-cascoded current mirrors produce the tail currents in the differential pairs. The circuit design is based on a previously reported implementation[6] with the following modifications: first, differential pair biasing was achieved by a resistor network instead of changing the well potential of the PMOS transistors; second, source degeneration was introduced in the differential pairs in order to extend the linear range. We model source degeneration as a reduction in κ of the source-coupled transistors:

$$\text{SD: } \kappa \mapsto \frac{\kappa^2}{1+\kappa} \tag{6}$$

This approximation is valid in the range of $V_{sg} = 1\text{V} - 2.25\text{V}$, and outside that range κ is attenuated even further.

The sine shaping circuit characteristics are modeled by (7):

$$\Delta I_{out} = \sum_{i=-2}^{i=2} (-1)^i I_B \tanh(\lambda(V_\theta - V_{ref_i})) \approx \sin V_\theta \tag{7}$$

where $\lambda = \frac{SD(\kappa)}{2U_T} = \frac{\kappa^2}{2U_T(\kappa+1)}$ and $V_{ref_i} = \{1, 1.5, 2, 2.5, 3\}$. In Fig. 4 this characteristic is compared with simulation results using a BSIM3.3 model in PSPICE. The large signal model had an accuracy error of 5% and an RMS error of 3%. The sine and cosine fitting error (i.e. the accuracy of the approximation), are around 8% and 5% respectively. These errors can be reduced further by circuit parameter tuning, and improving the circuit model to account for deterministic variations due to Early effect and κ's sensitivity to changes in V_{sg}.

2.4 Cosine shaping circuit

Three design modifications to the sine shaping circuit are needed to create a cosine shaping circuit. The trigonometric identity $-\sin(\theta - \frac{\pi}{2}) = \cos\theta$ suggests using a phase shift in the reference voltages, plus flipping the differential pair current outputs to account for the negative sign. However, this also shifts the operating range of the approximation to $[-2\pi - \frac{\pi}{2}, 2\pi - \frac{\pi}{2}]$. An additional differential pair must be added to ensure that the circuit functions over the interval $[-2\pi, 2\pi]$. Adding this additional differential pair adds an additional current source or sink, shifting the current output when $\sin V_\theta = 0$ to I_B. An additional constant current sink is needed to shift this zero-level current output.

2.5 Multiplier Circuit

The multiplier is implemented using a four quadrant translinear Gilbert cell[7] with differential current-mode signal inputs and outputs. Accuracy is degraded by non-ideal current mirrors, each at a different operating point. These nonlinearities were suppressed by using long transistors to reduce the Early effect and by using self-cascoded current mirrors to increase the output resistance.

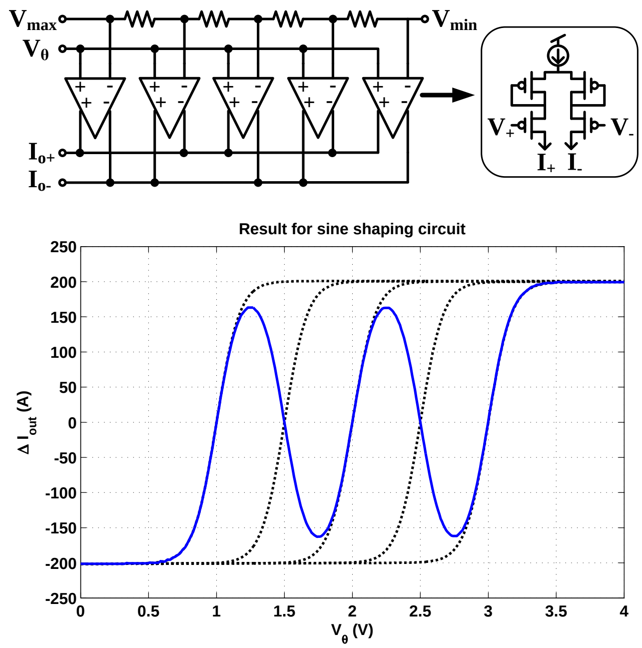

Figure 4. Top, sine shaping circuit comprising five differential pairs. Bottom, comparison of the analytical model with simulation. Blue is the large signal model (MATLAB), green is the PSPICE simulation, and the black dotted lines are the individual differential pair current outputs.

2.6 Digital Application-Specific Integrated Circuit (ASIC)

In order to perform comprehensive comparisons between different types of implementations, we also designed a digital ASIC which computes the closed form odometry solutions in (2). The number of bits for inputs, outputs, and internal nodes were chosen based on the requirements of dynamic range and precision for this application; v, ω, x, y, and θ are 8, 8, 12, 12, and 8 bits respectively. The sine and cosine functions are generated using a lookup table which contains only the sine values from 0 to $\frac{\pi}{2}$ (65 elements in this case). The cosine values and other sine values are calculated by exploiting the symmetry of these functions. Since the requirement for operation time is not strict here, folding technique was applied to reduce area and hopefully the leakage power. As a result, the computations in (2) can be carried out in two clock cycles with only one multiplier. The design was implemented in Verilog HDL. Synthesis was done by Cadence Encounter RTL Compiler using 0.5 μm 2P3M technology. The reported area is 0.70 mm^2 and the highest speed is 61.8 MHz. These values could be improved if the system had access to read-only memory (ROM) to implement the lookup table instead of using combinational logic circuits.

3. SIMULATED SYSTEM PERFORMANCE

3.1 Circuit Power

We can break down the power consumption of the analog circuit for each circuit component. We compare this with the power required to compute (2) using a digital microcontroller and a custom designed digital ASIC. We assume that computations are implemented on a microcontroller comparable to the TI MSP430 with a hardware multiplier. (2) requires 11 multiplications, 1 division, and 12 additions using Taylor series expansion. We assume that each operation takes five clock cycles ignoring memory access costs. The energy required to perform the computation is independent of the clock, but assuming 100μA/MHz, $V_{DD} = 3$V and a 32 MHz clock, the equations of motion can be solved in 0.1μs consuming about 35nJ of power.

3.2 Error metrics

It is of practical interest to compare the circuit performance to the closed form solutions (2). However, commonly used error metrics such as mean squared error are sensitive to small changes in signal frequency or phase, even

Component	N elements	current draw per element [A]
Gilbert multiplier[7]	2	1u, 1.4u
sine shaper[6]	1	1u
cosine shaper[6]	1	1.2u
Integrator and state control	1	$\approx$ 18u
signal Integrator	2	$\approx$ 0.4u
Total		23u
Power at 5V		0.1mW

Table 2. Circuit static current draw for each circuit component.

Implementation	Design Setting	Energy per operation (nJ)
Mixed Signal	$\tau = 10^4$	11.5
	$\tau = 10^5$	1.15
	$\tau = 10^6$	0.12
Digital ASIC	Clk = 1 MHz	5.4
	Clk = 1 MHz	15
μController	Power: 100 $\frac{\mu A}{MHz}$	35 (est.)

Table 3. Power comparison between different implementations and design settings.

if the signals are "qualitatively" similar. We propose a means to account for this by first, estimating relevant signal parameters using a numerical optimization routine. Second, a parameter normalization transform "locks" the expected signal's and the simulated signal's estimated amplitude, phase, and drift. In other words, we wish to develop a parameter-invariant error metric, then account for the parameter errors separately. The resulting error metrics in Table 4 have a physical interpretation and suggest sources of error in the circuit design.

Consider signals of the following form:

$$x(s, C) = C_1 \sin(C_2 s + C_3) + C_4 s + C_5 \tag{8}$$

where x is a continuous function, and $s \in \mathbb{R}$. Continuous solutions from (2) given control signals (υ, ω) calibrated into the circuit space can be equated to the form of (8) to calculate C for the ideal analog computer signal output.

We will also consider discrete time sampled signals from the PSPICE simulation to be of the following form:

$$\hat{x}(\hat{t}, K) = K_1 \sin(K_2 \hat{t} + K_3) + K_4 \hat{t} + K_5 + \epsilon(\hat{t}) \tag{9}$$

We will denote sampled signals as (t, x), where $t, x \in \mathbb{R}^N$, and N is the number of sample points. t is not necessarily uniformly sampled as is the case with PSPICE simulations. The ideal sampled signal is (t, x), while $(\hat{t}, \hat{x})$ is the result of the PSPICE simulation.

$(t, \epsilon(t))$ represents the residual from fitting (fitting error). We used the Nelder-Mead simplex (direct search) method to minimize the Euclidean norm of the residual when estimating K for (9). Initial conditions for K were chosen using some heuristics but were easily approximated by observing the signal. Parameters were also prescaled to improve fitting.

Error analysis comprises comparing the expected signal parameters C and the estimated parameters K from the PSPICE simulation results, and calculating the root mean squared residual error. The parameter normalization transform T can be represented by a sequence of 4 linear transforms (on a suitably defined vector space). These transforms are not necessarily commutative:

$$T : (\hat{t}, \hat{x}, x(\cdot, C)) \mapsto (\hat{t}, x^T, \hat{x}^T) \tag{10}$$

$$\begin{aligned}
\hat{x}^T &= \hat{x} - K_4\hat{t} - K_5 && \text{remove drift and bias from } \hat{x} \\
t &= \frac{K_2}{C_2}\hat{t} - \frac{C_3 - K_3}{C_2} && \text{phase shift cancellation} \\
x^T &= x(t, C) - C_4 t - C_5 && \text{evaluate } x \text{ at the phase compensated points} \\
\hat{x}^T &\mapsto \frac{C_1}{K_1}\hat{x}^T && \text{amplitude normalization}
\end{aligned}$$

This transform also preserves the residual such that$(\hat{t}, \epsilon(\hat{t})) = (\hat{t}, x^T - \hat{x}^T)$. Once the signals have been transformed, they can be plotted to show both phase information and the residual to provide insight into the source of the error.

Period offset	$\frac{2\pi}{K_2} - \frac{2\pi}{C_2}$
amplitude scaling	$\frac{K_1}{C_1}$
drift	K_4 (assume $C_4 = 0$)
residual error	$RMSE(t, \epsilon(t))$

Table 4. Error metrics

Period offset was chosen over frequency offset due to the nature of the modulus circuit. Undesired quantities of charge (from leakage or "impulse" quantities from circuit resets) accumulate on the integrating capacitor V_θ and reflect a linear shift in period, but a nonlinear shift in frequency. Observe that since $f = \frac{1}{T}$, for sufficiently small changes in period: $\frac{d}{dT}f = -\frac{1}{f^2}$. Amplitude scaling can result from mismatch in current mirrors, and variations in capacitance on the integrating nodes. Changing the integrator's capacitance changes the slew rate, providing an additional source of amplitude scaling. In addition, drift results from undesired currents being injected onto the integrating capacitor on V_x or V_y.

Finally, using the rectangle rule for numerical integration:

$$RMSE(t, \epsilon(t)) = \sqrt{\frac{1}{t_N - t_1}\sum_{i=1}^{N}(\epsilon(t_i)\Delta t_i)^2} \tag{11}$$

captures any nonlinear circuit module deviations from the ideal transfer functions. Phase offsets (C_3 vs. K_3), and DC offsets (C_5 vs. K_5) were not controlled parameters during testing and have been ignored in the analysis.

3.3 Mixed-Signal Simulation Results

In Fig. 5, we simulate results when $\omega = 0$ and υ is constant to show that x increases linearly with time. In this case, $\Delta I_\upsilon = 0$ nA and $\Delta I_\upsilon \in [2, 4, 6, 8, 10]$ nA.

For the more general cases of input control variables, we performed simulations for the following combinations of υ and ω, where $\Delta I_\omega \in [20, 40, 60, 80, 100]$ nA and $\Delta I_\upsilon \in [2, 4, 6, 8, 10]$ nA. In Fig. 6 and 7, families of graphs of $V_x(t)$ and $V_y(t)$ with constant ω are the same color. These families are also shifted with a DC offset to improve visibility. There is a current mismatch occurring in the circuit which results in an undesired drift of the signals. Note that as predicted by (2), the frequency of the circuit scales linearly with ω and the amplitude of the sinusoid is proportional to $\frac{\upsilon}{\omega}$.

The following results are for V_x, but are similar to those for V_y (V_y's drift is notably different). Fig. 8 demonstrate how the parameter invariant transform (10) matches ideal waveforms to simulation results. The residual $(t, \epsilon(t))$ is also apparent.

Figures 9 and 10 highlight the relationships between choice of $(\Delta I_\upsilon, \Delta I_\omega)$ to the error metrics defined in 4.

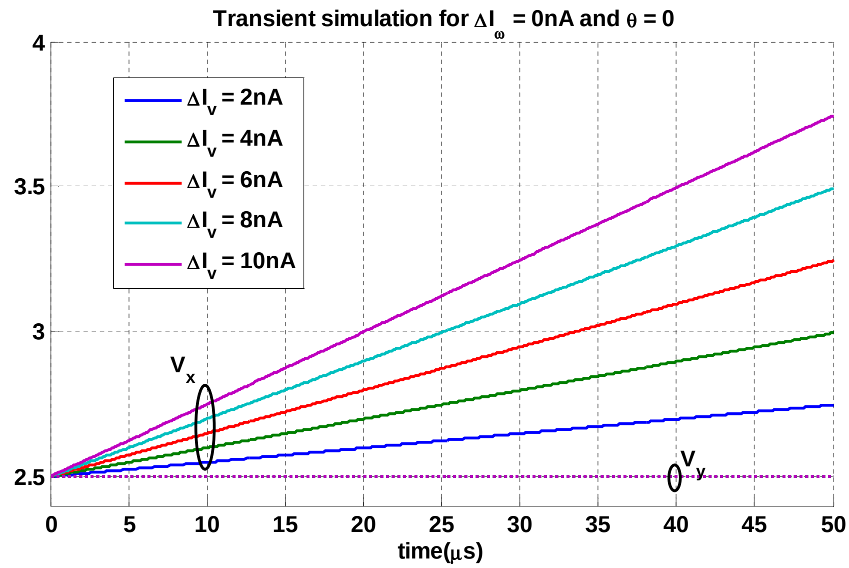

Figure 5. Simulation results with $\omega = 0$. Sweeping values for υ, the relationship is linear

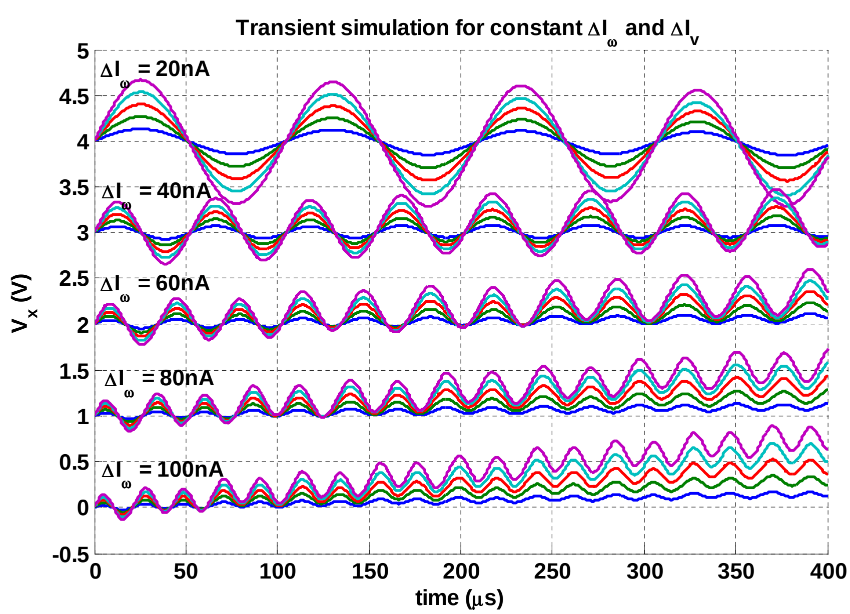

Figure 6. Simulation results for V_x with (υ, ω) constant.

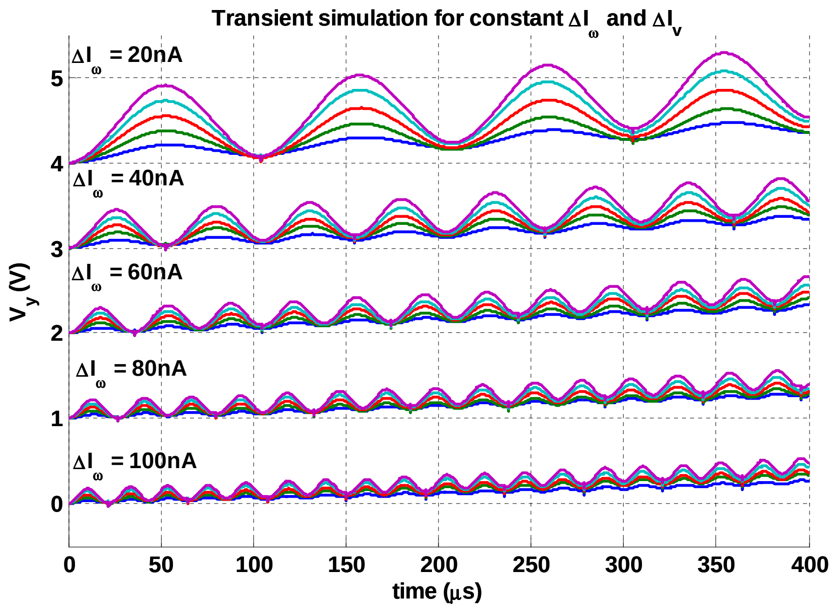

Figure 7. Simulation results for V_y with (υ, ω) constant.

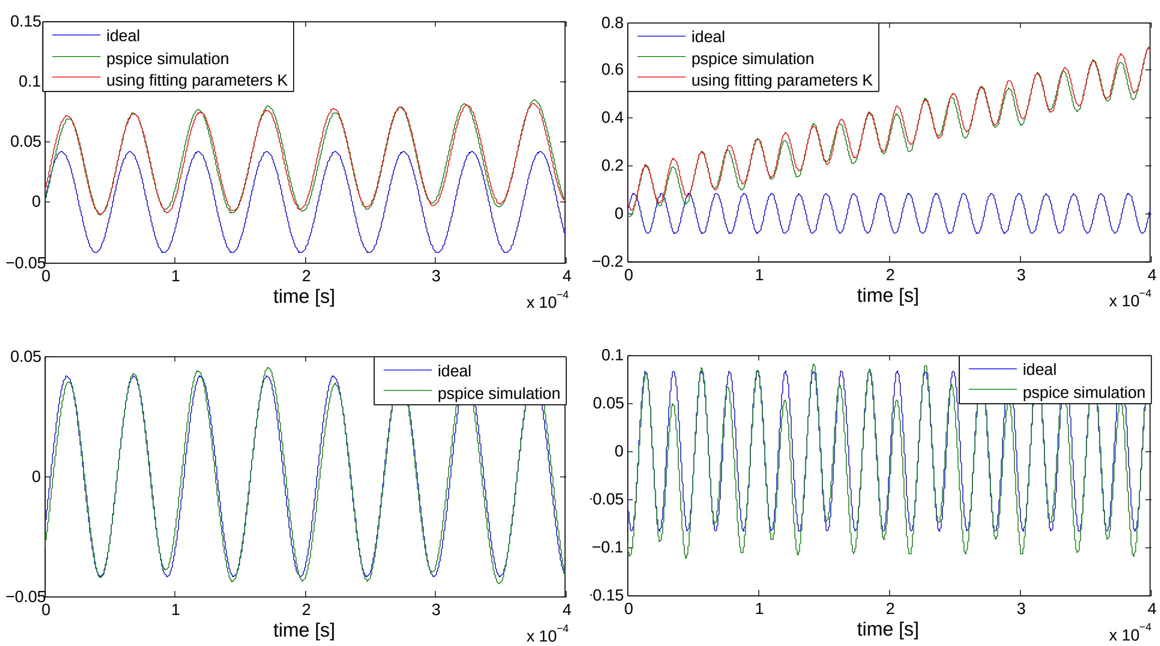

Figure 8. Showing two examples comparing the expected circuit response from the PSPICE simulation, along with the fit from (9) using estimated K parameters (top). The bottom shows both signal phase and the residual. Left:$(\Delta I_\upsilon, \Delta I_\omega) = (2, 40)$ nA. For this simulation run, $RMSE$ = 1.5e-6 Right: $(\Delta I_\upsilon, \Delta I_\omega) = (10, 100)$ nA. For this simulation run, $RMSE$ = 7.2e-6

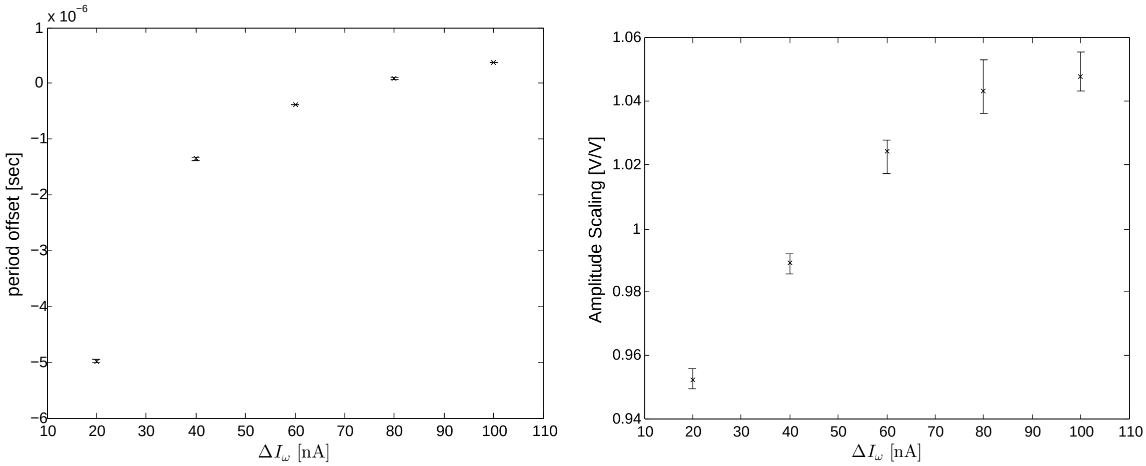

Figure 9. Simulation results for both period offset [left] and amplitude scaling [right] for a given $\Delta I_\omega \in [20, 40, 60, 80, 100]$ nA. Results were averaged over ΔI_v, suggesting that these error metrics are insensitive to changes in ΔI_v . Error bars signify the minimum and maximum observed values for a given ΔI_ω.

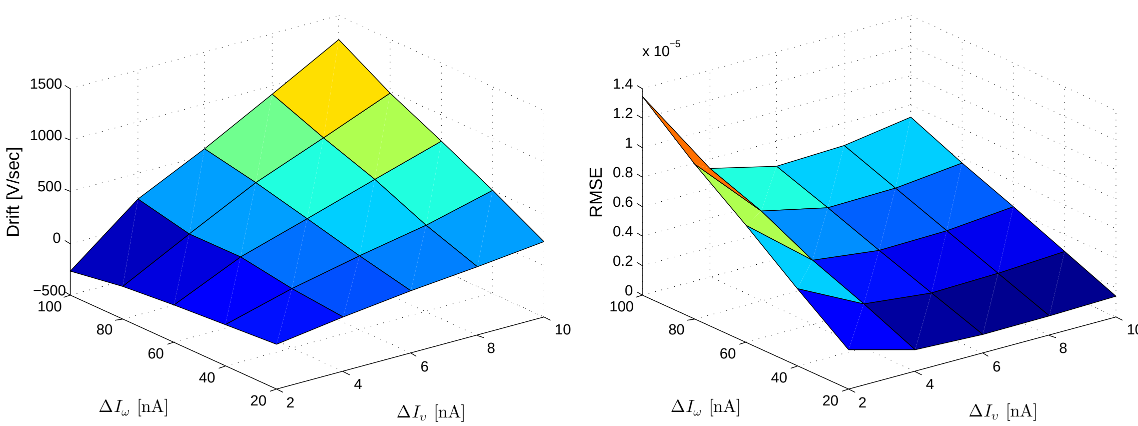

Figure 10. Left: Drift error K_5 as a function of $(\Delta I_v, \Delta I_\omega)$. Right: $RMSE(\hat{t}, \epsilon(\hat{t}))$ error, as a function of $(\Delta I_v, \Delta I_\omega)$.

4. CONCLUSION

The mixed-signal odometry circuit satisfies the strict design requirements of a miniature mobile robot. Detailed analysis of the dynamics simulator suggests that an analog or mixed-signal implementation can dramatically reduce power consumption at an acceptable loss in precision.

Acknowledgment

The authors would like to thank Dr. Timothy K. Horiuchi and the rest of the ANTBOT team. This material is based upon work supported by the National Science Foundation under Award Nos. 0647321, 0755224, and 0931878, and the ONR AppEl Center at UMD.

REFERENCES

[1] Caprari, G. and Siegwart, R., "Mobile Micro-Robots Ready to Use: Alice," in [*2005 IEEE/RSJ International Conference on Intelligent Robots and Systems,*], 3295–3300, IEEE (2005).

[2] Sineriz, G., Kuhlman, M., and Abshire, P., "High Resolution Distance Sensing for Mini-Robots using Time Difference of Arrival," in [*IEEE International Symposium on Circuits and Systems*], (May 2012).

[3] Navarro-Serment, L., Grabowski, R., Paredis, C., and Khosla, P., "Millibots," *Robotics & Automation Magazine, IEEE* **9**(4), 31–40 (2002).

[4] Kuhlman, M., Arvelo, E., Lin, S., Abshire, P., and Martins, N., "Mixed-Signal Architecture of Randomized Receding Horizon Control for Miniature Robotics," in [*2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*], 570–573 (Aug. 2012).

[5] Gilbert, B., "Circuits for the Precise Synthesis of the Sine Function," *Electronics Letters* **13**(17), 506–508 (1977).

[6] Fried, R. and Enz, C., "MOST Implementation of Gilbert sin(x) Shaper," *Electronics Letters* **32**(22), 2073–2074 (1996).

[7] Diotalevi, F. and Valle, M., "An Analog CMOS Four Quadrant Current-Mode Multiplier for Low Power Artificial Neural Networks Implementation," in [*15th European Conference on Circuit Theory and Design, ECCTD*], **1** (2001).