Mixed-Signal Architecture of Randomized Receding Horizon Control for Miniature Robotics

Michael J. Kuhlman, Eduardo Arvelo, Shuoxin Lin, Pamela A. Abshire, and Nuno C. Martins
Department of Electrical and Computer Engineering, University of Maryland, College Park
{ mkuhlman, earvelo, slin07, pabshire, nmartins } @umd.edu

***Abstract*— Control of miniature mobile robots in unconstrained environments is an ongoing challenge. Miniature robots often exhibit nonlinear dynamics and obstacle avoidance introduces sigificant complexity in the control problem. Furthermore, miniature robots have strict power and size constraints, drastically reducing on-board processing power and severely limiting the capability of digital implementations of nonlinear model predictive controllers. To accommodate the demands of this application area, we describe the architecture of a mixed-signal mobile robot control system using randomized receding horizon control. We compare the proposed mixed-signal implementation with purely digital control systems in terms of power requirements and precision and find that the mixed-signal implementation offers significant reductions in power consumption at an acceptable loss of precision.**

I. Introduction

Advances in sensing, actuation and battery technology have allowed for the development of very small robots (sub-cm^3). However, on-board computation for small platforms has been limited by available micro-controllers, which are relatively large, consume significant power, and are too slow for some applications (such as flying micro robots). In this paper we propose a mixed-signal architecture implementing receding horizon control (RHC) strategies on small robotic platforms under significant power and space constraints. The architecture is designed for the control of a differential-drive miniature robot such as the one shown in Fig. 1. RHC controllers are highly adaptable to changing environments and can be used in a wide variety of systems. Our mixed-signal architecture is based on a randomized search of the allowable control inputs (actions) which can potentially be fast, small and low power compared to digital systems.

A. Overview of Receding Horizon Control

Receding horizon control (also known as model predictive control) is a class of control strategies in which the control input (action) at time k is obtained by solving a finite horizon optimization problem that models the system's current state and future behavior of the system. The solution to this problem is a finite sequence of control actions, *but only the first element is applied to the real system*. At the next time step $k+1$, the procedure is repeated (see Fig 2). The repetition of this procedure effectively "closes the loop," providing updated information about the current system state to the controller. When the system state is not immediately available, an estimator that relies on sensor measurements is used.

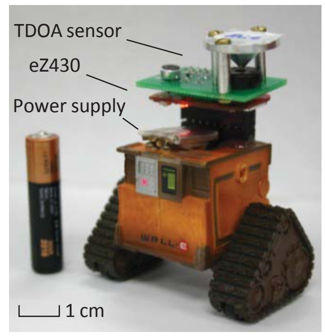

Fig. 1. Miniature robot platform. AAA battery shown for reference.

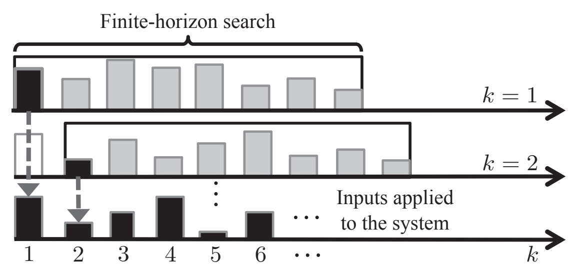

Fig. 2. Overview of receding horizon control. At each time instance a finite solution is found, but only the first element is applied to the sytem.

In order to implement RHC, the future behavior of the robot must be simulated using a model of its dynamics. The approximate behavior of the discretized system is given by:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \tag{1}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. Based on the knowledge (or an estimate) of the current state $\mathbf{x}_k$, the controller needs to find a control sequence $\mathbf{u}_k^N = [\mathbf{u}_{1|k}, \cdots, \mathbf{u}_{N|k}]$ that minimizes a cost given by:

$$J_N(k) = \sum_{j=1}^{N} g(\mathbf{x}_{j|k}, \mathbf{u}_{j|k}) \tag{2}$$

where the stage cost $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is convex and $\mathbf{x}_{j|k}$ satisfies (1) with $\mathbf{x}_{0|k} = \mathbf{x}_k$. The cost is used to set the objective of the robot, and usually penalizes deviation from a desired state and power consumption. Moreover, the states and control actions are constrained in that:

$$\mathbf{x}_{j|k} \in \mathbb{X}_k,\ \mathbf{u}_{j|k} \in \mathbb{U} \tag{3}$$

978-1-4673-2527-1/12/$31.00 ©2012 IEEE

for all time instances $j \in \{1, ...N\}$. For robotics applications (3) represents obstacle constraints and is allowed to change over time and actuation constraints. Moreover, the final state in the horizon has to satisfy a terminal constraint (which guarantees stability):

$$\mathbf{x}_{N|k} \in \mathbb{X}_k^T \tag{4}$$

for some set $\mathbb{X}_k^T$, which can be tuned to achieve a desirable performance. Finally, once a solution is found, only the first element $\mathbf{u}_{1|k}$ is implemented. At the next time step $k+1$, the process is repeated with knowledge of $\mathbf{x}_{k+1}$ (or an estimate $\hat{\mathbf{x}}_k$). This recursive procedure results in a sequence of control actions given by: $[\mathbf{u}_{1|0}, \mathbf{u}_{1|1}, \cdots, \mathbf{u}_{1|k}, \cdots]$. For a comprehensive survey on RHC, see [1].

B. Randomized RHC

Each finite horizon subproblem is usually solved using numerical algorithms, which enables RHC strategies to handle "hard" problems. This, however, poses a challenge: the time alloted for the controller to decide on a control action is limited by the sampling period of the system dynamics. For this reason for many years RHC was used only for problems whose dynamics were slow enough so that the optimization program had time to reach an acceptable solution. Many different ways of coping with this real-time constraint have been suggested. Most methods are beyond the processing capabilities of a state-of-the-art microcontroller [2], [3]. We propose an alternate solution addressing real-time processing constraints using a mixed-signal architecture.

Our architecture performs a randomized search in the space of allowed control inputs. Randomized strategies have been suggested in [4] that use control Lyapunov functions, which may not be available. When using a randomized approach, it is practically impossible to find an optimal solution. Therefore, we need to accept feasible solutions that are not optimal, but still stabilize the system. In [5], it is shown that feasibility is sufficient for stability if we impose an extra constraint on the cost. For this architecture, we consider that the added stabilizing cost constraint is given by:

$$h(J_k) \leq 0 \tag{5}$$

where the function $h : \mathbb{R} \mapsto \mathbb{R}$ can be tuned to achieve a desirable performance.

The sub-optimality relaxation changes each optimization problem to the problem of finding one (any!) feasible solution. Here a feasible solution is a finite sequence $\mathbf{u}_f$ that satisfies (3), (4), (5). The high-level description of the proposed algorithm is as follows:

1) Set $k = 0$
2) Estimate $\mathbf{x}_k$.
3) (Randomly) generate a candidate solution $\mathbf{u}_k^N$.
4) Propagate states N steps using (1).
5) Check constraints (3), (4) and (5). If check fails, return to step 3.
6) Implement $\mathbf{u}_{1|k}$. Set $k = k + 1$. Return to step 2.

In the subsequent sections, we propose a mixed-signal architecture to implement randomized RHC for a differential-drive two-wheeled robot.

II. System Architecture

For the RHC miniature robot controller, we assume that $\mathbf{x}_k = (x_k, y_k, \theta_k)$ and $\mathbf{u}_k = (u_{lk}, u_{rk})$, where $n = 3$ and $m = 2$. $\mathbb{U}$ is the space of all feasible motor controls.

The overall system architecture implementing RHC on a miniature differential drive robot is in Fig. 3. Primary sensor input comes from a distance-only sensor using Time Difference of Arrival [6]. An observer (Extended Kalman Filter, EKF) maps these observations into changes in the system state space (x, y, θ). The random trajectory generator uses multiple random number generators (RNG) and feeds them into an analog shift register, effectively creating a piecewise constant control signal with fixed time period T between steps but random step values. Such control signals are easy to replicate with the motor controller using pulse width modulation (PWM). The dynamics simulator enables feedforward control and testing of candidate control signals by modeling the equations of motion of the vehicle. There are obstacle avoidance constraints using the IR sensors and stability or performance constraints using information from the dynamics simulator. The constraint and cost checker determines how feasible the candidate control signal is. Once a feasible control has been found, the first element of the control sequence will be sent to the motor controller for execution.

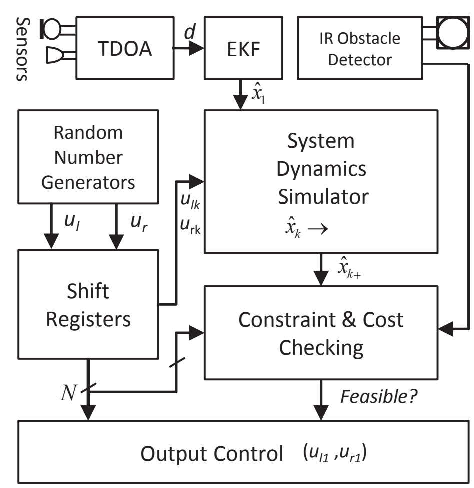

Fig. 3. System-level design of a mobile robot with RHC

A. Random Number Generator & Shift Register

To sample the control space, we randomly generate $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, so that u is parameterized by a collection of random variables X_i. In order to produce random variables that are approximately normal in distribution, the system will use several compact, Bernoulli true random number generators (RNG) based on amplified thermal noise [7]. These RNG's are combined using a digital to analog converter (DAC) to produce an analog random variable, which has a distribution similar to the Gaussian random variable. To have N steps in the piecewise constant signal, 2 RNGs will be fed into two parallel to serial analog shift registers each of length N (one for each motor controller).

B. Simulation of System Dynamics

In order to implement RHC, one must model system state dynamics. We have developed a mixed-signal kinematic model that maps motor commands to estimated and predicted changes in position in Euclidean Space (SE(2)). Specifically, given piece-wise continuous functions of time, we would like to solve the nonlinear differential equations of motion (6). Closed form solutions of (x, y, θ) exist for piecewise constant (υ, ω) which can be implemented directly in a digital system (7):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \upsilon \cos\theta \\ \upsilon \sin\theta \\ \omega \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{\upsilon_k}{\omega_k}(\sin\theta_{k+1} - \sin\theta_k) + x_k \\ \frac{\upsilon_k}{\omega_k}(\cos\theta_k - \cos\theta_{k+1}) + y_k \\ \omega_k(t_{k+1} - t_k) + \theta_k \end{bmatrix} \quad (7)$$

In an analog system, it is preferable to directly solve (6) given the fewer operations required to model the nonlinear system dynamics. At the computational level, this will require 4 signal scaling elements, two summing nodes, three integrators, two trigonometric function blocks and two signal multipliers (see Fig. 4). Translinear circuits offer significant computational capability and will be used to approximate trig functions [8]. It is important to note that since we wish to solve the equations of motion in faster than real time, the circuits representation of time will be much faster (by several orders of magnitude) than the real world clock. The required precision of these components will be discussed later.

C. Constraint Checker and Cost Tracker

The two primary pathways for constraints checking are for obstacle avoidance and stability/performance. Information from the dynamics simulator plus the control signal can be used to assess the convergence of the robot to the desired state. First, controller should progressively shrink the region of acceptable terminal conditions over time, using (8):

$$\|[x_k\, y_k]'\| < \beta \|[x_0\, y_0]'\| \quad (8)$$

$0 < \beta < 1$ in (8) remains fixed, scaling the size of the ball geometrically over time, and $\|\cdot\|$ is a (convex) norm. $L1$ norms are preferable for circuit implementation because they are simpler and operate well over a wide dynamic range. Absolute value circuits for use in the $L1$ norm do not require current scaling unlike squaring circuits for the $L2$ norm used in quadratic costs. In order to guarantee stability, one must also bound the cost $J_N(k) < c$, where c is a tuning parameters that may change for each iteration of the RHC strategy

In addition, IR proximity sensors positioned around the robot detect any obstacles blocking the robot's trajectory. The IR sensors feed directly into the constraint checker, effectively "short-circuiting" the controller to run in a fail-safe mode upon imminent collision, increasing the flexibility of RHC strategies for mobile robot control.

III. Anticipated Performance: Analog vs. Digital

A. Circuit Power

The power consumption requirements will differ for the analog and digital implementations of the simulator. From the system diagram, we can approximate the power consumption of the analog circuit by estimating the number of bias currents required. We assume the use of Gilbert cells for signal multiplication, an operational amplifier with a capacitor for integration, and the equivalent of five differential pairs in circuits used to approximate trigonometric functions [8]. This requires a total of 14 bias currents. Assuming I_{bias} = 100nA, V_{DD} = 5V, and 10μs to 1ms of circuit operation time, the analog circuit consumes 70pJ to 7nJ,

For the digital system, we assume that the closed form solutions to the equations of motion (6) are implemented on a microcontroller comparable to the TI MSP430 with a hardware multiplier. (7) 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. In addition, we also ignore the limiting case of $\omega_k \to 0$, which would further increase the complexity of the digital implementation. Dramatically greater computational costs would occur if time-stepping approximations were used to solve the equations of motion. The energy required to perform the computation is independent of the clock, but assuming 100μA/MHz, V_{DD} = 3V and a 32 MHz clock, the equations of motion can be solved in 0.1μs consuming about 35nJ of power.

B. Circuit Precision

For the analog circuit, we assume that each of the components have a linear distortion error, i.e. $\sigma < 1\%$ [9]. Assuming that the errors are independent, and compound at each stage as a summation of Gaussian random variables, the maximum depth of the circuit is 5 stages so the circuit error would be bounded by a 2.2% error where $\sigma_{system} = \sqrt{\sum_{i=1}^{N} \sigma_i^2}$. This would result in a lower bounded signal to noise ratio (SNR) of 45/1, where $SNR = \frac{1}{\sigma}$. To achieve an SNR of 100/1, the RMS error of all stages would need to be $< 0.4\%$.

The SNR of the digital circuit is determined by the number M of bits used in the circuit. 8 bit calculations result in a maximum theoretical SNR of 256/1, where $SNR = 2^M$. This SNR can be scaled exponentially by a linear increase in the number of bits and power consumed [10] but subsequent operations would not necessarily scale linearly.

C. Summary

The previous analysis suggests that the analog implementations can model the system dynamics with significantly less

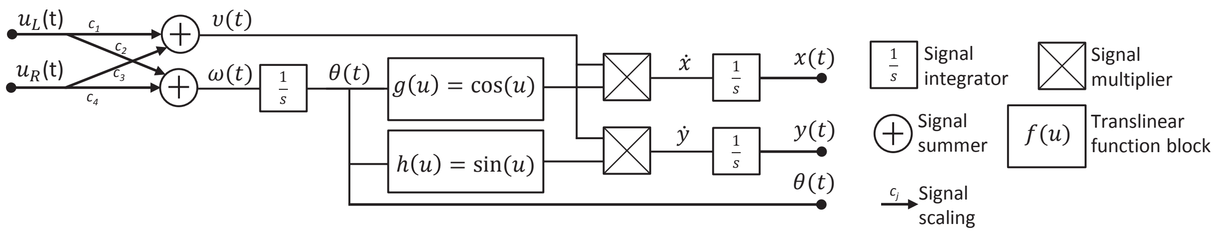

Fig. 4. System-level design of analog robot kinematics simulator highlighting signal flow and primary computational blocks.

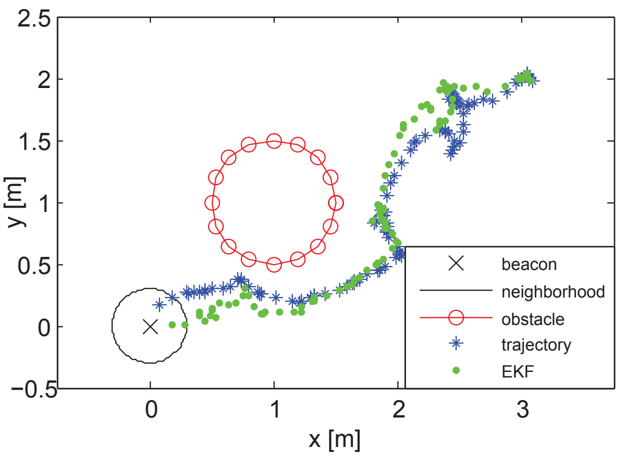

Fig. 5. Simulated example of the proposed architecture.

power usage. However, with digital systems, greater precision can be achieved efficiently by increasing the bit length of the signal. Therefore, analog systems are well suited for fast, low power, but less precise computations, which is applicable to problems in robotics.

IV. Simulation

We developed a system-level simulation of the proposed architecture in Fig 3, including obstacles. The robot must track the position of a single beacon and move to its location avoiding obstacles. Measurement uncertainty, model parameters and system noise were extracted from sensor and motor calibration experiments on the platform shown in Fig. 1 and were incorporated into the simulation. Fig. 5 shows an example trajectory generated by the controller.

The RHC algorithm is sensitive to the distribution of the control variables that are randomly selected. Picking trajectories in either the motor speed space (u_l, u_r) or the robot body speed space (v, ω) will also bias the simulation. For example, a small imbalance in selecting motor speeds for a robot with a small wheelbase can cause the robot to spin quickly if care is not taken.

V. Conclusion

The mixed-signal randomized RHC controller 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. Reductions in power consumption using a mixed-signal implementation compared to a digital implementation will decrease from 35nJ to 70pJ-7nJ per odometry computation. The change in SNR of 255/1 for 8 bit computations to the specified SNR of 100/1 for mixed signal circuits is an acceptable loss in precision.

Acknowledgment

The authors thank 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] D. Mayne, J. Rawlings, and C. Rao, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[2] A. Alessio and A. Bemporad, "A Survey on Explicit Model Predictive Control," in *Nonlinear model predictive control*, L. Magni, D. Raimondo, and F. Allgöwer, Eds. Springer Berlin / Heidelberg, 2009, pp. 345–369.

[3] M. Zeilinger, C. Jones, and M. Morari, "Real-Time Suboptimal Model Predictive Control Using a Combination of Explicit MPC and Online Optimization," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1524–1534, 2011.

[4] H. Tanner and J. Piovesan, "Randomized Receding Horizon Navigation," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2640–2644, 2010.

[5] P. Scokaert, D. Mayne, and J. Rawlings, "Suboptimal model predictive control (feasibility implies stability)," *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 648–654, 1999.

[6] G. Sineriz, M. Kuhlman, and P. Abshire, "High resolution distance sensing for mini-robots using time difference of arrival," in *IEEE International Symposium on Circuits and Systems*, May 2012.

[7] P. Xu, Y. Wong, T. Horiuchi, and P. Abshire, "Compact floating-gate true random number generator," *Electronics Letters*, vol. 42, no. 23, pp. 1346–1347, 2006.

[8] R. Fried and C. Enz, "MOST implementation of Gilbert sin(x) shaper," *Electronics Letters*, vol. 32, no. 22, pp. 2073–2074, 1996.

[9] B. Gilbert, "A precise four-quadrant multiplier with subnanosecond response," *IEEE Journal of Solid-State Circuits*, vol. 3, no. 4, pp. 365–373, 1968.

[10] B. Hosticka, "Performance comparison of analog and digital circuits," *Proceedings of the IEEE*, vol. 73, no. 1, pp. 25–29, 1985.