A Reconfigurable Computing Platform for Plume Tracking with Mobile Sensor Networks

Byung Hwa Kim*[a] , Colin D'Souza*[a], Richard M. Voyles[†b], Joel Hesch[‡b], Stergios I.Roumeliotis[£b]
[a]Dept. of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455
[b]Dept. of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455

ABSTRACT

Much work has been undertaken recently toward the development of low-power, high-performance sensor networks. There are many static remote sensing applications for which this is appropriate. The focus of this development effort is applications that require higher performance computation, but still involve severe constraints on power and other resources. Toward that end, we are developing a reconfigurable computing platform for miniature robotic and human-deployed sensor systems composed of several mobile nodes. The system provides static and dynamic reconfigurability for both software and hardware by the combination of CPU (central processing unit) and FPGA (field-programmable gate array) allowing on-the-fly reprogrammability. Static reconfigurability of the hardware manifests itself in the form of a “morphing bus” architecture that permits the modular connection of various sensors with no bus interface logic. Dynamic hardware reconfigurability provides for the reallocation of hardware resources at run-time as the mobile, resource-constrained nodes encounter unknown environmental conditions that render various sensors ineffective.

This computing platform will be described in the context of work on chemical/biological/radiological plume tracking using a distributed team of mobile sensors. The objective for a dispersed team of ground and/or aerial autonomous vehicles (or hand-carried sensors) is to acquire measurements of the concentration of the chemical agent from optimal locations and estimate its source and spread. This requires appropriate distribution, coordination and communication within the team members across a potentially unknown environment. The key problem is to determine the parameters of the distribution of the harmful agent so as to use these values for determining its source and predicting its spread. The accuracy and convergence rate of this estimation process depend not only on the number and accuracy of the sensor measurements but also on their spatial distribution over time (the sampling strategy). For the safety of a human-deployed distribution of sensors, optimized trajectories to minimize human exposure are also of importance.

The systems described in this paper are currently being developed. Parts of the system are already in existence and some results from these are described.

Keywords: sensor networks, plume tracking, reconfigurable computing

1. INTRODUCTION

In case of natural gas leak or intentional release of harmful chemical contaminant or biological agent in a terrorist act the gas spreads in a plume and its spread is affected by wind and the topology of the landscape. In such a situation, decision-making demands real-time forecasting of the agent's concentration in the atmosphere. The accuracy of these time and spatial predictions depends in turn on source parameters such as the time, location, and total amount of the release. Accurate modeling and estimation of the parameters of the advection-diffusion equation for this airborne agent provide situation awareness for the event and lead to informed decisions for dealing with the emergency, such as, neutralizing the source, predicting its spread, and evacuating affected areas.

The objective for a dispersed team of ground and/or aerial autonomous vehicles (or hand-carried sensors) is to acquire measurements of the concentration of the chemical agent from optimal locations and estimate its source and spread. This requires appropriate distribution, coordination and communication within the team members. The key problem is to determine the parameters of the distribution of the harmful agent so as to use these values for determining

* bhkim@ece.umn.edu, colin.dsouza@ieee.org; phone +1-612-625-8544; fax +1-612-625-0572
† voyles@cs.umn.edu; phone +1-612-624-8306; fax +1-612-625-0572
‡ joel@cs.umn.edu; phone +1-612-625-2217; fax +1-612-625-0572
£ stergios@cs.umn.edu; phone +1-612-626-7507; fax +1-612-625-0572

its source and predicting its spread. The accuracy and convergence rate of this estimation process depend not only on the number and accuracy of the sensor measurements but also on their spatial distribution over time (the sampling strategy). Besides all these computations have to be done in real time to determine the trajectories that the teams have to follow.

A major constraining factor for the real time operation is that multiple tasks such as vision for navigation, PID control for locomotion, wireless communication, raw sensor data processing, etc. have to be done simultaneously with the evaluation of the diffusion parameters and the trajectory planning steps. Vision by itself is computationally expensive and with the added tasks of matrix inversions, divisions and power operations required for the plume detection algorithms, an on board processor is insufficient. This can be augmented by dedicated hardware, however that would tie us into using a fixed algorithm.

To overcome this, the plume detection algorithms will be run on a new reconfigurable computing platform that is being developed for miniature robotic and human-deployed sensor systems composed of several mobile nodes. The system provides static and dynamic reconfigurability for both software and hardware by the combination of CPU (central processing unit) and FPGA (field-programmable gate array) allowing on-the-fly reprogrammability.

Reconfigurable computing lies midway between the software and hardware domains and is becoming more feasible due to the introduction of new devices like modern FPGAs which have a large gate count allowing different tasks to be mapped on a single FPGA. Traditional computations were done either in software or hardware. Software while flexible and easy to develop, suffered from low performance and higher power consumption due to presence of extraneous units. Hardware systems exhibited extremely high performance and efficiency, but required great expertise to design and once fabricated were very difficult if not impossible to modify.

Static reconfigurability of the hardware manifests itself in the form of the "morphing bus" architecture that permits the modular and interchangeable connection of various sensors with no bus interface logic. It also allows us to tailor the hardware to match the sensors being used. Dynamic hardware reconfigurability provides for the reallocation of hardware resources at run-time as the mobile, resource-constrained nodes encounter unknown environmental conditions that render various sensors ineffective. By changing the hardware on the fly to service different sensors, we are able to support advanced algorithms for the sensors which may not be able to fit on a single FPGA.

Two power PC cores run a RTOS which manages sequencing of reconfiguration, managing communication tasks and data processing that are more efficient if run in software. Thus this system has necessary capabilities for supporting various gas sensors, accelerating the plume detection and tracking algorithms as well as controlling the motion of the robots.

We propose three novel thrusts:

- Optimization of trajectories to consider such factors as minimal exposure
- Porting of algorithms and utilities to a reconfigurable computing platform
- Implementation of the algorithms on real robots

This paper proposes a new system and describes it using plume detection. Section 2 lists other works in both reconfigurable computing for robotics and in plume detection. Section 3 gives an overview of our reconfigurable platform. Section 4 describes the novel “morphing bus” architecture. Section 5 gives a summary of results obtained using prototype system of FPGA. Section 6 and 7 describes plume detection and simulation result. As the sensor information and signal processing algorithms are classified we are unable at present to implement the combined system. However results on the individual parts have been obtained.

2. RELATED WORK

2.1. Reconfigurable computing platform

The benefit of software-based computation over hardware-based computation is the ability to reconfigure on-the-fly. This Run Time Reconfiguration (RTR) of computation is the basis for the flexibility and rapid growth of software-based solutions. But software requires a hardware target on which to run.

FPGA chips can be reconfigured, too, but most only permit Compile Time Reconfiguration (CTR). However, a new breed of FPGA chips, such as XC6000 and Virtex-II Pro, allow Run Time Reconfiguration (RTR) of logic [5]. Furthermore, some of these new FPGA chips, such as the Virtex-II Pro, have embedded microprocessor units (MPU), making it possible to build power-efficient and highly reconfigurable system-on-chip designs. These systems combine reconfigurable software, a power-efficient hardware target on which the software can run, and reconfigurable hardware all on a single chip.

"HW/SW co-design" usually refers to methodologies that permit the hardware and software to be developed at the same time - splitting some functions to be implemented in hardware for additional speed, while others are implemented in software to free up logic resources. This is normally done offline. The combination of HW/SW co-design techniques with online RTR capability at both the hardware and software levels can optimally assign functions between the FPGA and software dynamically [6].

In order to achieve this level of RTR, the system specification must be partitioned into temporal exclusive segments, a process known as temporal partitioning. A challenge for RTR is to find an execution order of a set of sub-tasks that meet system design goals, a process known as context scheduling. Several approaches can be found in the literature describing these problems (e.g. [7]). All these approaches depend on performance and resource requirements of the requisite sub-tasks to make an optimal tradeoff [6].

FPGA-based SoC designs have been widely applied in digital system applications and RTR research has been addressed by many researchers. Elbirt et al. explored FPGA implementation and performance evaluation for the AES algorithm [8]. Weiss et al. analyzed different RTR methods on the XC6000 architecture [5]. Shirazi described a framework and tools for RTR [9]. Noguera and Badia proposed a HW/SW co-design algorithm for dynamic reconfiguration [6]. FPGA power modeling and power-efficient design have also been studied by various researchers [10-14].

2.2. Recent methods for Plume Tracking

Recent work has focused on developing methods for estimating the two most pertinent characteristics of an airborne agent, namely its source and spread. *Chemotaxis* and *Anemotaxis* are two popular methods used in Chemical Plume Tracking (CPT); the former exploits knowledge of the agent's concentration gradient to select a suitable direction for the sensor to travel in, whereas the latter dictates sensor motion based on the mean wind direction. Although well accepted, both these methods suffer from serious drawbacks: if the agent's Reynolds number is high, chemotaxis-based strategies may not provide sufficient information for determining the agent's source, and anemotaxis will not be feasible in areas with little or no ventilation [15, 16]. The problem of predicting the spread of an airborne agent has also been examined by numerous researchers. The prominent approach among these is to compute the parameters of the advection-diffusion equation which governs the spread of the agent [17]. A non-linear least-squares method for estimating these parameters offline is presented in [18], and an exploration of agent spread under continuous release assumption is given in [19, 20]. Several other methods have been proposed, and an overview of these is available in [2].

3. RECONFIGURABLE COMPUTING PLATFORM

At the heart of the system is the Virtex-II Pro chip from Xilinx Inc. These devices consist of up to four 32-bit IBM® PowerPC™ 405 processor hard cores, multi-gigabit transceivers, high-density on-chip memory, and DSP blocks. The low-power, high-performance PowerPC cores are fully embedded within the FPGA fabric, where all processor nodes are controlled by the FPGA routing resources, proving system flexibility in hardware/software partitioning, (e.g. high-speed logic implementation in FPGA fabric and high-flexibility software code in Power PC). This unique architectural capability is ideal for distributed robotics applications, which comprise both speed-intensive raw-data (image) processing and control-intensive system management (communications and control).

The system employs a processor-centric architecture, where one PowerPC core is used to realize vision functions, another one is used for control and communication, and the FPGA fabric is used for custom logic and interfaces. Fig. 1 shows the block diagrams of the FPGA multiprocessor which integrates Pulse Width Modulation (PWM) for the motors, encoder counters, a camera interface, dual-port memory, a PowerPC for vision (with FFT and DCT hardware) and a PowerPC for control and wireless communication.

For Gas Detection the processing could be done either on the PowerPC or in the FPGA, or partly on both. Analog sensors would need Analog-to-Digital converters to interface to the FPGA. However it can be envisioned that very soon FPGAs with analog inputs and inbuilt A/D converters will be available. Then sensors can be directly connected to the system.

A key feature is the communication system. It uses Bluetooth in a multi hop network for transfer of images to an operator. It is also used to share information between robots about gas concentrations for a cooperative multi robotic system. Bluetooth is the preferred means of communication since high data rates are required for the lossless transfer of images back to the operator.

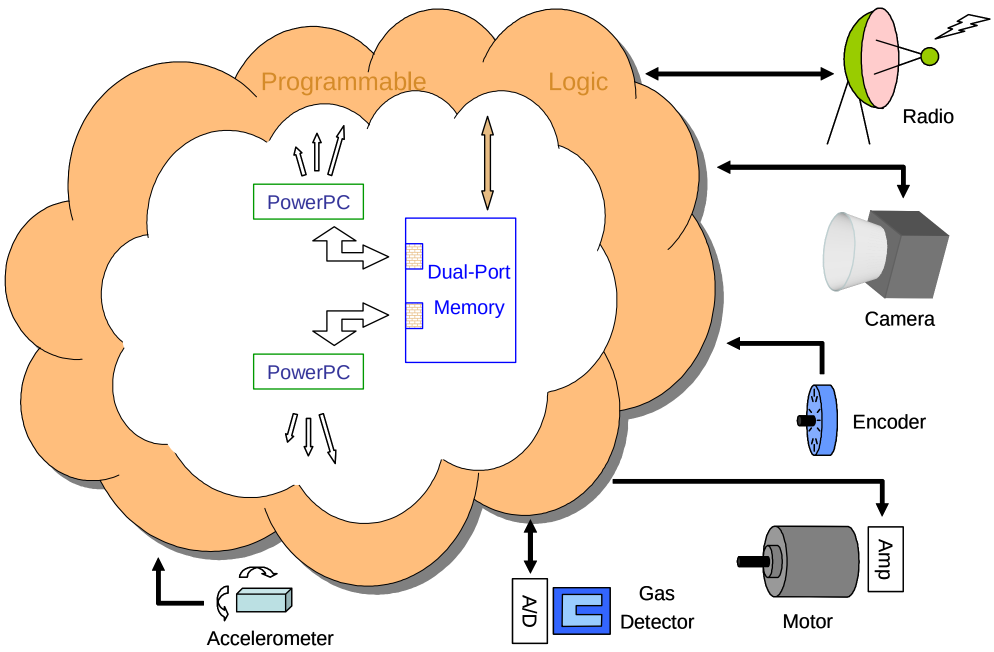


Figure 1: Block diagram of reconfigurable computing platform.

4. MORPHING BUS

4.1 Morphing bus

The morphing bus exploits the static reconfigurability of the FPGA to provide an interface to modular sensors without bus interface logic.

There have been numerous bus protocols such as I2C, USB, serial bus, etc. Some like the USB achieve plug and play capability by storing interface logic on the device. Thus the protocol is able to query the device to gather interface information from it. Also logic is required for bus arbitration in case multiple devices need to be serviced at any given time. This standard bus is described in Fig. 2.

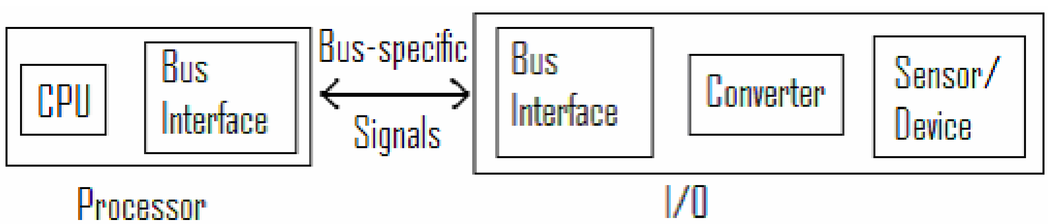


Figure 2: Standard bus (static).

The morphing bus architecture in Fig. 3 gets rid of the need for interface and arbitration logic by providing a dedicated rather than a multiplexed bus for each device with the flexibility to swap the position of each device. The required data handshaking, data translation and signal processing is done on the FPGA.

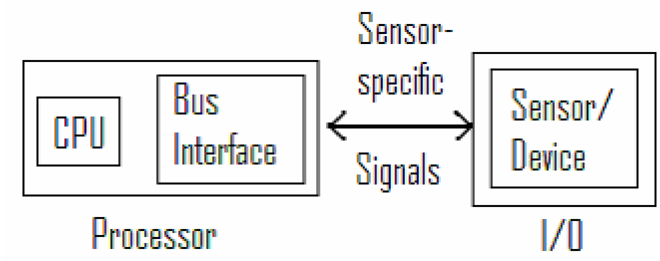


Figure 3: Morphing bus (dynamic).

The bus is made up of circuit boards each of which is dedicated to only one sensor or actuator. This allows the boards to have low complexity and hence small size. Each wedge has electrical connectors at both ends. All the wedges provide the same interface to the preceding and succeeding stages. Thus their position in the bus can be swapped. Each wedge uses as many routes as required to support the logic on that wedge and the remaining are fed to the next connector of the next stage which in turn does the same and so on.

The input lines to a wedge are used as follows: few initial lines are dedicated to power and ground. These are common to all wedges and run through all of them. Starting from the next connection the wedge circuitry uses as many I/O pins as it requires. The remaining lines are shifted to the output connector such that the unused lines are now immediately after the power lines (Fig. 4).

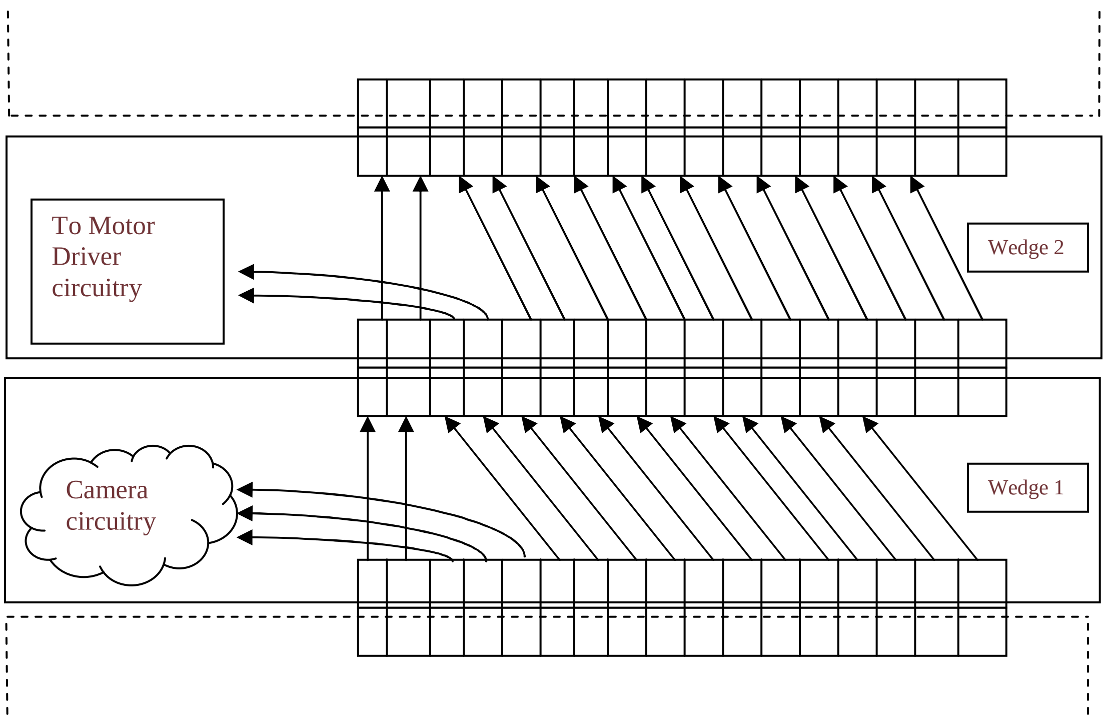
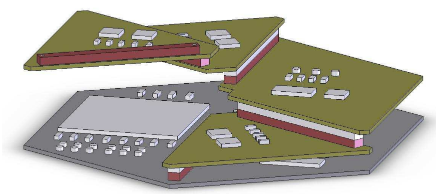


Figure 4: Wedge diagram for morphing bus.

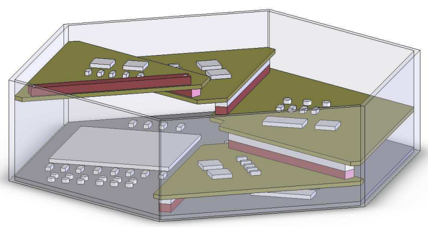
The morphing bus is currently being designed for use in the TerminatorBot [4] and its structure is shown in Fig. 5. Because of the shapes of the wedges, when they are stacked up they take the form of a spiraling staircase. To provide support to this structure mechanical reinforcements are provided. Air is blown from the base upward which follows the path along the spiral cooling the ICs on every wedge. The whole structure is enclosed in wrap to maintain rigidity. If excessive cooling is required the wrap can be made from a conductive material and the various boards can be soldered to the wrap to provide additional conductive cooling.

The number of devices that can be connected in the morphing bus architecture is limited by the number of available pins routed from the FPGA through the wedges, since each board has a dedicated connection to an FPGA pin. This is determined by size of the connector that can fit on the wedge which in our case is limited by the diameter of the Terminatorbot. Besides a large portion of the wedge is taken up by the feed through routing of unused lines. A second issue is that the boards are not hot swappable i.e. they have to be plugged in and the device has to be configured before the system is turned on.

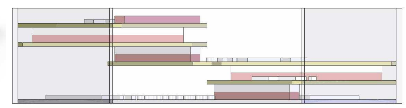
The first issue is acceptable since although it fixes a cap on the number of devices, we have the great advantage of being able to do without interface and arbitration hardware on the devices plugged in. The second is remedied by providing tools that take in the order of the devices and HDL interface descriptions of each and automatically generating a top level file and a corresponding pin configuration file. This can be used in the place and route process.



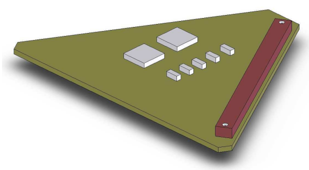
(a) Spiral structure without wrapping around



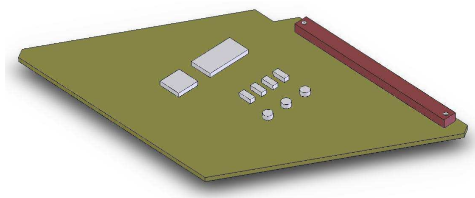
(b) Spiral structure with wrapping around



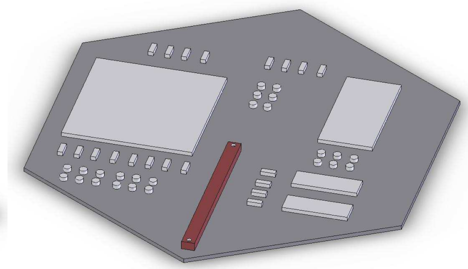
(c) front view



(d) basic cheese wedge



(e) double cheese wedge



(f) FPGA mainboard

Figure 5: Spiraling diagram for TerminatorBot.

5. PROTOTYPE IMPLEMENTATION

5.1. Experimental platform and Motor control interface design

As a prototyping system for the reconfigurable computing platform, we used experimental platform and performed DC motor control [1]. Example components of a complete system for motor control include a trajectory generator, a PID module, a PWM module, an amplifier and motor, a shaft encoder, and an encoder interface. The trajectory generator is implemented in software, the PWM module and encoder interface is implemented on the FPGA, and the amplifier, motor, and shaft encoder are external to the system. The PID module, which is the focus of this prototype system, is implemented both in hardware on the FPGA, and for comparison, in software.

5.2. Function Test

A performance evaluation is meaningful only after the design is verified as functionally correct. Several different PID hardware designs were implemented and used to perform step response control of a motor[1]. Additionally, a software implementation was developed and tested. The same parameters and sampling period were applied to the hardware PID implementations in the FPGA to perform the step response control tests. Experiment results of step response control for all designs are shown in Fig. 6.

To test motor control for each design, the motor was set to an initial position of 1000, then a desired position command of 1200 was issued. From Fig. 6, the horizontal dashed line is the desired position, while the other curves are the real responses sampled from the encoder counter. The results show that all the designs performed correctly and similarly. Response speed is fast, overshoot is small, and static accuracy is high. The average rise time is 30.32ms and the standard deviation of the rise time is 0.7451. The steady state error is 0.

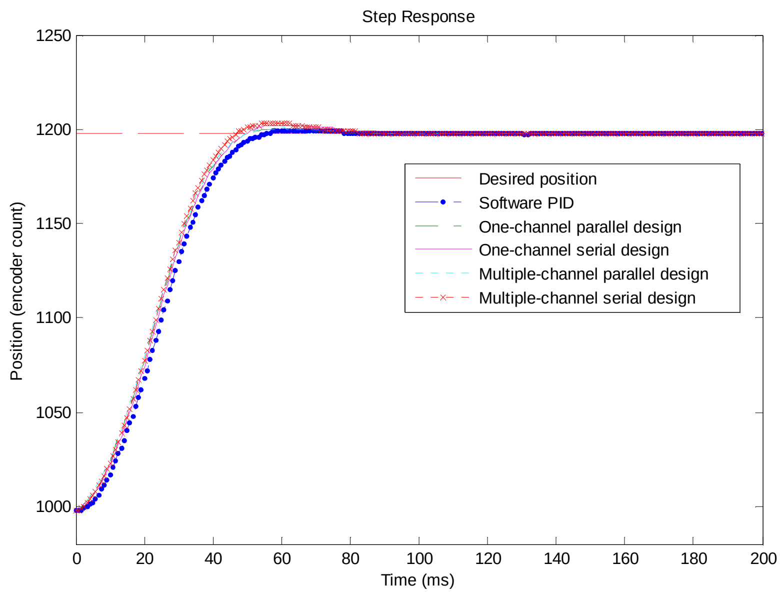


Figure 6: Step response control experiment results. (Reprinted from [1].)

5.3. Performance test

PID hardware design was implemented and used to perform step response control of a DC motor. Additionally, a software implementation was developed and tested. Once the correctness of the designs was verified, performance was analyzed. Xilinx provides a variety of performance analyses, including resource utilization, speed, and power consumption, based on simulations of the hardware design, as reported in [1]. Performance was based on these reports.

Resource utilization for each design was measured and analyzed. In one-channel and multi-channel serial designs, all arithmetic operations share one multiplier and one adder, while in parallel designs there are 3 multipliers and 4 adders. Because of this, serial designs have an obvious space advantage. However, some of this space savings is used up with additional control logic.

Area and speed are conversely related. The advantages in area requirements shown for serial design are countered by their disadvantage in speed. While the datapath in the serial design is shorter, thus the delay is shorter, more clock cycles are required. As expected, execution times for serial design were longer.

Power consumption is dependent upon the sampling and control clock frequency. Thus, to compare power performance, we both generated motor commands and ran the PID module at various frequencies. The test data obtained in the step response experiments were used as input to the hardware simulation of each PID design. In multiple-channel designs, for the same sampling frequency of 0.12MHz and control clock frequency, power dissipation increases linearly as the number of channels increases. It was expected that for the same sampling frequency and execution time, the multiple-channel parallel based design would consume less power, because the clock frequency of the parallel based design is lower. For one channel, the parallel based design does consume less power, however, for a large number of channels, the parallel-based design consumes *more* power than the serial-based. The result also shows that for the same sampling frequency, the channel-level parallel design with a serial PID unit consumes the least power, but the area requirements of the channel-level parallel design rapidly exceed the capacity of the FPGA as the number of channels increases.

5.4. Discussion on performance analysis

In this experiment of prototype system, preliminary work was conducted to explore control system design for a resource-constrained robot based on an FPGA technique. Parallel and serial architectures of the PID control algorithm were designed and implemented for one-channel and multiple-channel architectures.

Performance tests show that for a small number of channels, channel-level parallel design with serial PID has the smallest area and consumes the least power. For more channels, the channel-level serial design with serial PID has the smallest area requirements and the channel-level parallel design with serial PID still consumes the least power, but the area requirements of the channel-level parallel design with serial PID increases very quickly.

In order to adapt different environment, robot doesn't need to choose one design to achieve one performance goal, for example, maximum power. On the contrary, robot could change designs or reconfigure the FPGA fabrics to adapt different situation considering all the trade-offs such as circuit area, execution speed, and power consumption, and robot could find optimal configuration for various condition.

6. PLUME TRACKING

There is always a possibility of an accidental gas leak or intentional release of harmful chemical/biological/radiological contaminant in a terrorist act. In such situations the gas spreads in a plume and its spread is affected by wind and the topology of the landscape. The ability of the rescue team to minimize casualties depends on real-time forecasting of the agent's concentration and movement in the atmosphere. The accuracy of these time and spatial predictions depends in turn on source parameters such as the time, location, and total amount of the release. Accurate modeling and estimation of the parameters of the advection-diffusion equation for this airborne agent provide situational awareness for the event and lead to informed decisions for dealing with the emergency, such as, neutralizing the source, predicting its spread, and evacuating affected areas.

The objective for a dispersed team of ground and/or aerial autonomous vehicles (or hand-carried sensors) is to acquire measurements of the concentration of the hazardous agent from optimal locations and estimate its source and spread. This requires appropriate distribution, coordination and communication within the team members. The key problem is to determine the parameters of the distribution of the harmful agent so as to use these values for determining its source and predicting its spread. The accuracy and convergence rate of this estimation process depend not only on the number and accuracy of the sensor measurements but also on their spatial distribution over time (the sampling strategy). Furthermore, the use of human-carried or robot-carried sensors adds additional optimization constraints that, for example, minimize exposure to the contaminant or minimize power consumed by the trajectory. Finally, it is critical that these computations be completed in real time or near real time to determine the trajectories that the teams have to follow.

All these requirements make the reconfigurable platform described earlier ideal for implementing the plume detection algorithms and their interface to sensors. Different sensors are required to detect the radiation and dissimilar types of gases. These can easily be accommodated with minimum effort using the morphing bus which does not require any interface on the sensor side. As most of the sensors deal with concentrations in the parts per billion range, extensive signal processing is required. This can be supported either in software or on the FPGA hardware.

The system allows you to change the algorithm on the fly both in hardware and software. Thus when the robot is far from the gas source, concentration is low and sensor data is small, it can spend more resources on locomotion thus moving toward the source quickly. This helps maximize battery usage in case of robots and in case of hand held sensors reduces exposure to rescue workers. Alternatively, if a particular location in a very low concentration area is deemed to be valuable to the estimation, additional signal processing resources (software and hardware) can be dedicated to the sensing task while the sensor sits still.

Thus we can consider the platform as part of a heterogeneous sensor network that can support different gas and navigational sensors along with a high speed communication link. This makes it well suited for a computationally intensive and time critical task such as plume tracking, while retaining the flexibility to respond to contaminants that are unknown beforehand.

Initially, we consider only a single sensor and ignore the effects of landscape topology[2]. Constant wind is considered in the models. We then extended our investigation to include multiple sensors[3]. In either case, dealing with complex topology, such as buildings and trees, is not practical in real-time with this this approach to estimation of adjection-diffusion parameters.

In [2] we have presented a method for estimating the parameters of the advection-diffusion equation. The algorithm is a generic one that assumes a sensor reading is given to it and is independent of the type of sensor or the composition of the plume. The focus of this work was to decide the best motion strategy (i.e., the optimal locations to measure the agent's density) for a group of mobile sensors so as to maximize the information gained from the sensors. The problem is posed as a non-convex optimization minimization which seeks to minimize the trace of the covariance matrix corresponding to the uncertainty in the estimates of the advection-diffusion equation parameters. Due to the computationally intensive aspects of the non-convex optimization strategy, we proposed two solutions to the problem. The first is the Locally Optimal Trajectory (LOT) optimization which considers only the next position for each sensor when performing the optimization. The second formulation termed Globally Optimal Trajectory (GOT) optimization seeks to minimize the trace of the covariance using the entire trajectories (i.e., all the positions) of all sensors as decision variables.

The trajectories generated by LOT approached that of GOT in 90% of the cases. However for it to be successful a minimum of 5 sensor readings are required. With additional readings accuracy improves. Hence the robot has to move quickly to enable the sensor to take readings at different locations. The fast computing platform accelerates the computation thus allowing the robots to move toward the contamination source rapidly

The performance of a multi robotic system would be far superior to that of a single robot as now we have many robots taking gas concentration readings over a wider area. The high speed bluetooth communication system on the reconfigurable platform turns the sensing robots into a wireless sensor network that can exchange information to better plan their trajectories. This was demonstrated in [3] where it was shown that communication between the robots causes them to behave differently from what they would if they acted independently. The trajectories of the five mobile robots are presented in Fig. 7 given wind velocity u = 0.5 m/sec as demonstrated in that paper.

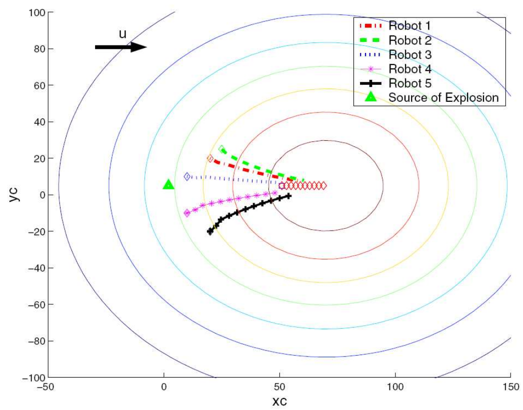


Figure 7: Sample trajectories of the Multi-Robot System for given wind velocity (u = 0.5 m/sec). Reprinted from [3].

7. SUMMARY AND FUTURE WORK

This paper described a new reconfigurable computing platform with the novel “morphing bus” architecture. An application of plume detection was described and the advantages of our system for such tasks were explained.

The platform consists of a Virtex-II pro FPGA which has two embedded PowerPC cores. The external components are interfaced to the FPGA via the morphing bus. This bus consists of multiple circuit boards each dedicated to only one sensor or actuator. Each board uses as many inputs as required and passes on the rest to the next board. Since each connector has same length and all boards use the first few connections the location of the boards can be swapped without having to redesign them. Tools which take the order in which the devices are connected and output top level configuration information will be developed.

Preliminary testing of the system was done on PID control architectures, which experimented with serial and parallel structures. The system was also evaluated in light of plume detection. The reconfigurable hardware gives us the ability to accelerate the involved calculations as well as accommodate new types of sensors.

The plume detection algorithms have been proved via simulation. The trajectories generated approach the global optimum in 90% of the cases, with a minimum of 5 readings. Accuracy increases with the number of readings.

The system is under development, and we will be working on implementing the plume detection algorithms on hardware. Simultaneously tools to design a system using the morphing bus are being developed.

8. ACKNOWLEDGEMENTS

This work was sponsored in part by the NSF Safety, Security and Rescue Research Center and by the National Science Foundation (ITR-0324864, MRI-0420836). The authors wish to thank Alliant Techsystems for discussions and sensors.

REFERENCES

1. W. Zhao, B.H. Kim, A.C. Larson and R.M. Voyles, “FPGA Implementation of Closed-Loop Control System for Small-Scale Robot”, in *Proceedings of the 2005 International Conference on Advanced Robotics*, July 2005, pp.70-77.
2. V. Christopoulos and S.I. Roumeliotis, “Adaptive Sensing for Instantaneous Gas Release Parameter Estimation”, in *Proc. 2005 International Conference on Robotics and Automation*, Barcelona, Spain, April 18-22, 2005, pp. 4461-4467.
3. V. Christopoulos and S.I. Roumeliotis, “Multi Robot Trajectory Generation for Single Source Explosion Parameter Estimation”, in *Proc. 2005 International Conference on Robotics and Automation*, Barcelona, Spain, April 18-22, 2005, pp. 2814-2820.
4. Richard M. Voyles, “TerminatorBot: A Robot with Dual-Use Arms for Manipulation and Locomotion”, in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 2000, pp.61-66.
5. Karlheinz Weiss, Ronny Kistner, Arno Kunzmann and Wofgang Rosenstiel, “Analysis of the XC6000 Architecture for Embedded System Design”, in *Proceedings of IEEE Symposium on FPGAs for Custom Computing Machines*, April 15-17, 1998, pp.245-252.
6. Juanjo Noguera and Rosa M. Badia, “HW/SW Codesign Techniques for Dynamically Reconfigurable Architectures”, in *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 10, No.4, August 2002, pp. 399-415.
7. R. Maestre, F. J. Kurdahi, M. Fernandez, R. Hermida, “A Framework for Scheduling and Context Allocation in Reconfigurable Computing”, in *Proc. of the International Symposium on System Synthesis*, 1999, pp. 134-140.
8. Adam J. Elbirt, W. Yip, B. Chetwynd, and C. Paar, “An FPGA-Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists”, in *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 9, No.4, August 2001. pp. 545-557.
9. N. Shirazi, W. Luk and P.Y.K. Cheung, “Framework and tools for run-time reconfigurable designs”, in *IEE Proceedings of Computers and Digital Techniques*, Vol. 147, No. 3, May 2000, pp.147-151.
10. Seonil Choi, Ronald Scrofano, Viktor K. Prasanna, and Ju-Wook Jang, “Energy-Efficient Signal Processing Using FPGAs”, in *Proceedings of ACM/SIGDA 11th ACM International Symposium on FPGA*, Monterey CA, Feb. 23-25, 2003, pp.225-233.
11. Li Shang and Niraj K. Jha, “High-level Power Modeling of CPLDs and FPGAs”, in *Proceedings of 2001 IEEE International Conference on Computer Design*, Sep. 23-26, 2001, pp.46-51.
12. Subodh Gupta and Farid N. Najm, “Power Modeling for High-Level Power Estimation”, in *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 8, No.1, Feb. 2000, pp. 18-29.
13. Li Shang and Niraj K. Jha, “Hardware-Software Co-Synthesis of Low Power Real-Time Distributed Embedded Systems with Dynamically Reconfigurable FPGAs”, in *Proceedings of the 15th IEEE International Conference on VLSI Design (VLSID'02)*, 2002
14. Andres Garcia, Wayne Burleson, and Jean Luc Danger, “Low Power Digital Design in FPGAs: A Study of Pipeline Architectures implemented in a FPGA using a Low Supply Voltage to Reduce Power Consumption”, in *Proceedings of 2000 IEEE International Symposium on Circuits and Systems*, Geneva, vol.5, May 28-31, 2000, pp.561-564.
15. D. Zarzhitsky, D. F. Spears, D. R. Thayer, and W. M. Spears, “Agent- Based Chemical Plume Tracing using Fluid Dynamics,” in *Proceedings of Formal Approaches to Agent-Based Systems (FAABS III)*, 2004.
16. D. Zarzhitsky, D. F. Spears, W. M. Spears, and D. R. Thayer, “A Fluid Dynamics Approach to Multi-Robot Chemical Plume Tracing,” in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, New York, NY, July 2004.
17. H. Versteeg and W. Malalasekera, *Introduction to computational fluid dynamics. The finite volume method*. Longman Scientific and Technical, 1995.
18. R. M. P. Kathirgamanathan, R. McKibbin, “Source term estimation of pollution from an instantaneous point source,” *Research Letters in the Information and Mathematical Sciences*, vol. 3, no. 1, April 2002, pp. 59–67.

19. ——, “Sourse release-rate estimation of atmospheric pollution from non-steady point source - part [1]: Source at a known location,” *Research Letters in the Information and Mathematical Sciences*, vol. 5, June 2003, pp. 71–84.
20. ——, “Source release-rate estimation of atmospheric pollution from a non-steady point source - part [2]: Source at an unknown location,” *Research Letters in the Information and Mathematical Sciences*, vol. 5, June 2003, pp. 85–118.