

# Integrated Parallel Printing Systems with Hypermodular Architecture

David Biegelsen\*<sup>a</sup>, Lara Crawford<sup>a</sup>, Minh Do<sup>a</sup>, Dave Duff<sup>a</sup>, Craig Eldershaw<sup>a</sup>, Markus Fromherz<sup>a</sup>,  
Haitham Hindi<sup>a</sup>, Greg Kott<sup>b</sup>, Dan Lerner<sup>a</sup>, Barry Mandel<sup>b</sup>, Steve Moore<sup>b</sup>, Bryan Preas<sup>a</sup>, Wheeler  
Ruml<sup>a</sup>, Greg Schmitz<sup>a</sup>, Lars Swartz<sup>a</sup>, Rong Zhou<sup>a</sup>

<sup>a</sup>Palo Alto Research Center, Inc, 3333 Coyote Hill Rd, Palo Alto, CA USA 94304;

<sup>b</sup>Xerox Corporation, 800 Phillips Road, Webster, NY USA 14580

## ABSTRACT

We describe here a system consisting of multiple, relatively inexpensive marking engines. The marking engines are interconnected using highly reconfigurable paper paths. The paths are composed of hypermodules (bidirectional nip assemblies and sheet director assemblies) each of which has its own computation, sensing, actuation, and communications capabilities. Auto-identification is used to inform a system level controller of the potential paths through the system as well as module capabilities. Motion control of cut sheets, which of necessity reside physically within multiple hypermodules simultaneously, requires a new abstraction, namely a sheet controller which coordinates control of a given sheet as it moves through the system. Software/hardware co-design has provided a system architecture that is scalable without requiring user relearning. Here the capabilities are described of an exemplary system consisting of 160 modular entities and four marking engines. The throughput of the system is very nearly four times that of a single print engine.

**Keywords:** printing, parallel, module, hypermodular, planning, control

## 1. MODULAR PRINTING SYSTEMS

Throughput and uptime are two major concerns in the printing industry. Combining print engines in parallel can achieve a multiplicative effect on throughput and also allow a system to remain in operation, at partial productivity, even if an engine fails. Such a parallel printing system requires a paper path to link the parallel engines and produce automatically collated output. This paper describes a modular parallel printing system connected by a highly modular paper path.

Xerox has a long history of increasingly deep modularization of printing systems. The present work demonstrates an integrated system of parallel marking engines enabled by a paper path glue layer that has a level of modularity near the finest granularity of the design spectrum ('hyper-modularity'). The paper path consists of a small number of module types -- nip modules to provide bidirectional sheet motion and two types of directors for dynamic definition of path topology. Each module is capable of acting, sensing, computing and communicating. Modules, including marking engines, are hot swappable, and the system is capable of auto-configuring. Real-time planning and control software, like the hardware, is designed to be modular, distributed, reconfigurable and scalable. The system can handle exceptions, such as sheet jams, while maintaining (reduced) throughput.

The printing system presented here consists of two towers. Each tower houses two marking engines capable of printing single side images at 55 pages per minute (ppm). Paper path modules are pinned into a frame providing a fixed pitch array. Frames, called highways, hold a 1x7 array of nip modules and directors. Smaller 1x1 frames are stacked or hung from highway frames to provide on and off ramps as well as inversion and sheet purge functionalities at the input and output of each marking engine. All modules, including marking engines are removable from the array by sliding from the frame, perpendicular to the array.

Because the sheet trajectories are bidirectional, baffles between modules are interdigitated. To enable module extraction the interdigitated baffles are retractable. This decoupling of elements of the paper path also allows for novel rotary spindling of jammed sheets and sheet extraction perpendicular to the process direction, as described below.

\* [dave.biegelsen@parc.com](mailto:dave.biegelsen@parc.com); phone 1 650 812 4137

Figure 1 shows the full two-tower fixture. Each marking engine is set to print identifiable patterns as shown on sheets affixed to the covers. Two sheet feeders are on the left side of the array and two output trays are on the right. Frames (1x7 and 1x1) can be seen from the side. Nip modules are inserted into the frames at a fixed pitch of 130 mm, determined by the constraint that the shortest allowable sheet should always be engaged by at least one nip pair. Spaces between nips are populated with directors – modules which can dynamically select possible paths between nips.



Figure 1. Two-tower hypermodular printing system

## 2. HYPERMODULES

Figure 2 shows a nip module. The left side is the back side which is inserted first into the frame. The long shafts have spring loaded pins which allow the module to be inserted into holes laser cut into the back and front faces of the frames. A stepper motor drives a shaft holding two drive rollers in opposition to two sprung idler rolls. Upper and lower retractable nip baffle pairs (here sprung into their retracted position) are interdigitated within a pair to facilitate bidirectional sheet guidance without stubbing. The baffle retraction pins are used by the director (described below) to pivot the baffles into interdigitated alignment with the mating baffles of the director path guides. Optical edge sensors are mounted in the windows in the baffles to provide information about sheet leading and trailing edge timing as well as skew. The printed circuit board at the rear end carries a digital signal processor (DSP) which oversees the motor controller, sensors, local computation and communication. The gold pins at the rear of the PCB are spring-loaded connectors which enable hot swappable connections for power and communication. A final function, integrated onto all modules, is the auto-ID optical transceiver. Each nip module has two nearest neighbors with transceivers. DSPs on each module have a stored code which points to a data base enumerating the capabilities of the module. Simple nearest neighbor communications allow the module DSPs to learn their neighbors and transmit the information over the bus. A

central processor can then deduce all possible sheet paths and module capabilities (including marking engine characteristics, health state, etc.)

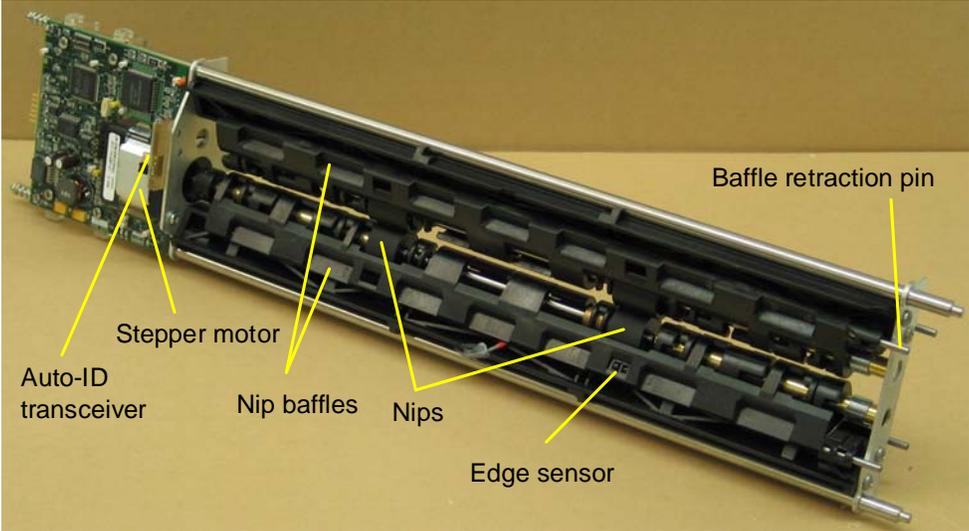


Figure 2. Nip module

Figure 3 shows a ‘3-way’ director. The left side corresponds to the front side of the module. The aluminum plate at the bottom is part of the drawer mechanism for sliding the module out of the frame. As can be seen in the inset the inner walls of the paper path are formed by three flippers sharing a common axis. Idle rollers help guide sheets through the 55 mm radius turns. The flippers are driven by the solenoids mounted on the end plate and driven by the PCB hidden below the bottom director baffle. The flippers take ~10 ms to change state in either direction. Cut sheets incident to any of the three ports can be guided to either of the two other ports, thus allowing six possible trajectories through the module.

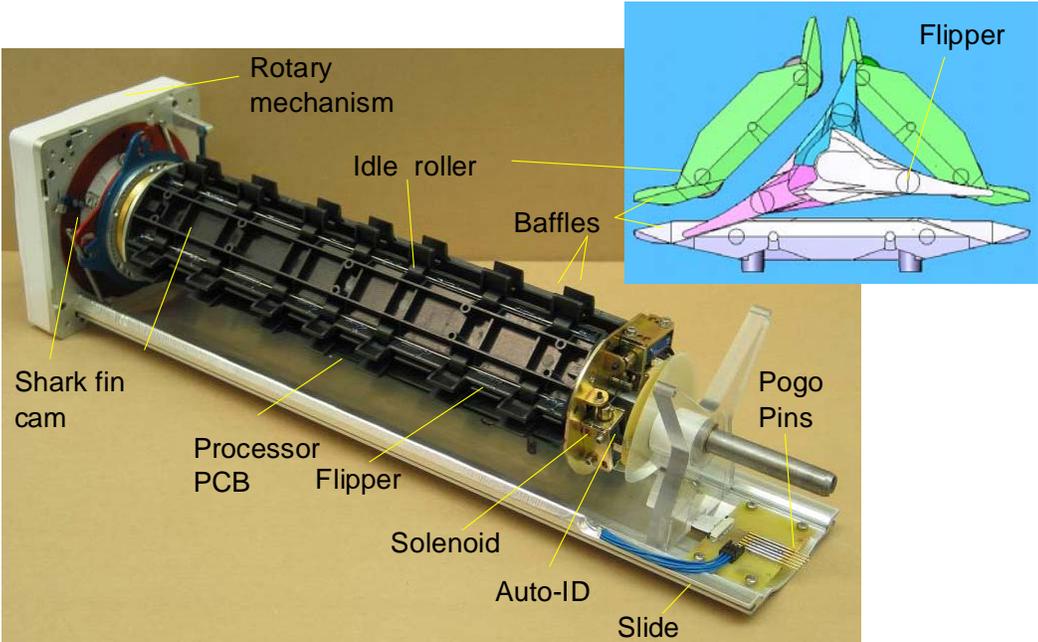


Figure 3 Three way director module



Figure 4 Rotary jam clearance

As will become evident from the discussion below, for architectures with redundant parallel sheet transport paths it becomes very difficult to access sheets within the system for jam clearance. For this reason we have developed directors which allow sheet extraction in a direction perpendicular to the sheet path. During the initiation of rotary motion the shark-fin-like structures (blue and red in Figure 3) allow the retraction pins on the nip baffles to spring open. Then the pin engagement rides up and over the pins to allow full rotation of the director core. At the end of rotary travel the unit unlatches from the frame, is depowered, and can be pulled out with any jammed sheet roughly centered on the director now spindled and removable through the opening in the frame. Figure 4 shows a director after spindling a jammed sheet and being partially extracted from the array. Neighboring nip motors assist the sheet as it is pulled into the director.

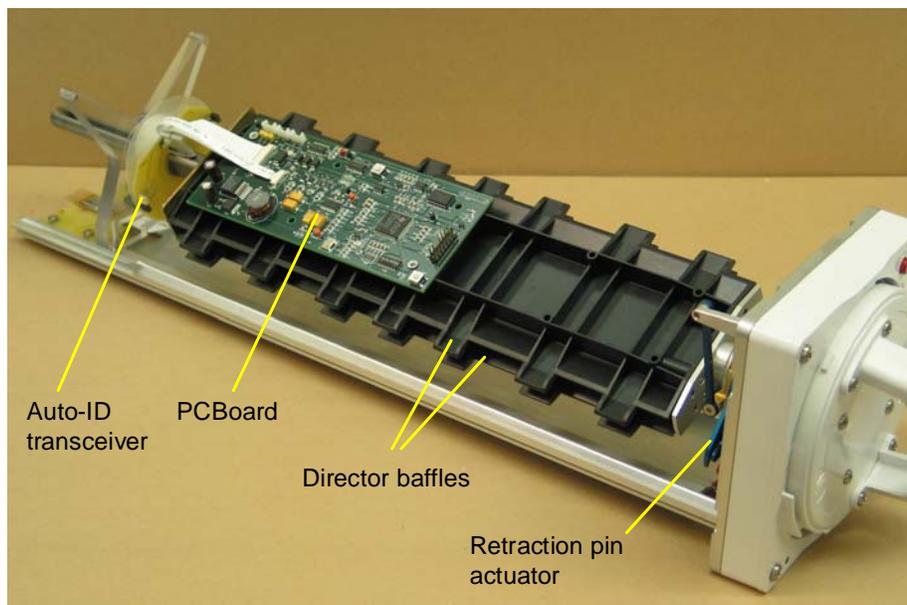


Figure 5 Straight-through director module

Figure 5 shows a straight-through director module (in an oppositely oriented view). Many of the parts are reused, such as the top director baffle. The handle at the right has been rotated part way through the  $\sim 360^\circ$  of its full rotation so that the PC board can be observed. An auto-ID transceiver can be seen again. Only when the director is in its proper angular position will the system accept the module as functional.

### 3. ARCHITECTURE AND RELIABILITY

Many ways exist for configuring multiple marking engines to work together. We have chosen a tower arrangement as the highest level of modularity. Each tower can host stacked print engines. (We have used commercially available printers which have been stripped of extraneous features such as duplex return paths and paper trays; the housing sizes, kept for expediency, limited us to using two engines per tower.) It is desirable in a highly parallel printing system to be able to transport a sheet from any engine output to any other input. For example, if one marking engine is a color marking engine and another is monochrome and a given sheet calls for one side to have color and the other to be monochrome, then it can be optimal within a job to transport a sheet in a retro direction from one marking engine to another. To enable optimal system throughput the paper paths must therefore have sufficient speed and parallelism to enable the system throughput to be limited only by the productivity of the marking engines printing at maximum capacity in parallel.

One architecture which satisfies these goals (and corresponds to the photo in Figure 1) is shown in Figure 5. Two 2-marking engine towers have been abutted. The paper path modular frames (shown as dashed lines) form a fixed pitch array populated alternately by nips and directors. Solid lines between nips indicate possible paper paths. The three horizontal paths in the center (highways) enable bidirectional redundant paths for connecting any outputs and inputs. Sheet inversion (and purging) can occur at any of the path terminations. The degree of inter-highway connectivity can clearly be chosen at will. The choice is related to system throughput (including the event of jam clearance), desired redundancy, reliability of parts, relative speeds of marking engines and path nips, etc. As a system scales to more towers, either more highways can be inserted or higher speed path elements can be used to keep pace with the higher marking engine capacity.

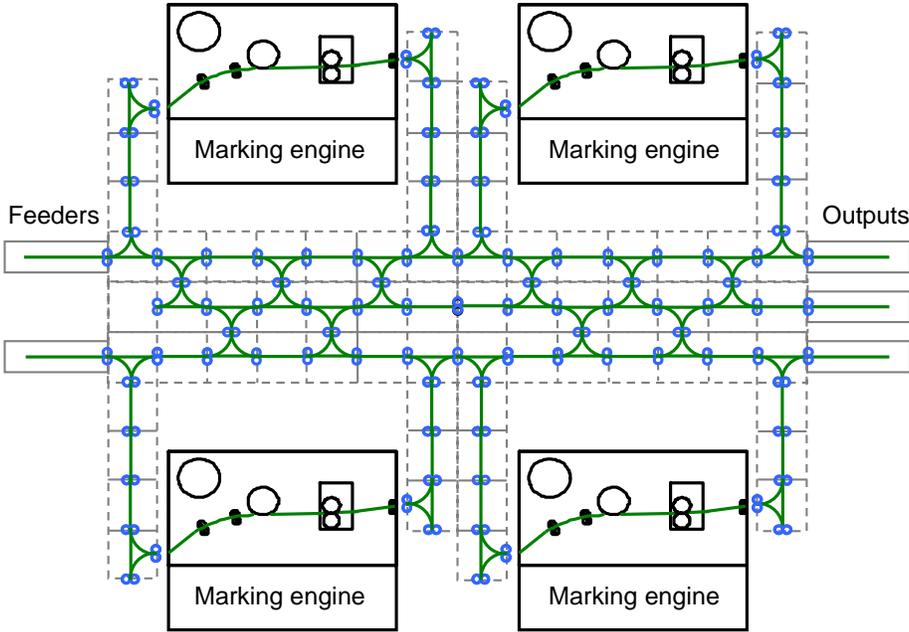


Figure 5 Two-tower, 3-highway architecture

All nips must be not only bidirectional but also capable of driving at a range of speeds. For example, the vertical on- and off-ramps must buffer sheet speeds between fast highway speeds and the relatively low speed of the marking engines. Similarly, obtaining maximum throughput requires sheet speed changes to enable trajectory crossing and timing. Because multiple nips in contact with a given sheet need to run at the same speeds, tight control between motor controllers needs to be supported (see discussion below).

The apparent complexity of this system forces one to question the relative reliability and cost of the hypermodular architecture in comparison with architectures using much coarser-grained modularity, but still with the same number of marking engines. Studies were carried out using as many part-for-part equivalents as possible. The bottom line was essentially that for the same degree of functionality the cost and reliability of the systems were approximately the same. This is due to two factors. First is that the dominant source of unreliability is the marking engine; and for the paper path the costs and unreliabilities are dominated by the actuators (motors and flippers). Because systems with the same functional capabilities have nearly the same numbers of actuators, the costs and reliabilities are quite similar. In comparison with a more custom-built monolithic architecture it was concluded that cost and reliability scaling for equivalent performance strongly favored the modular, multi-marking engine approach.

#### **4. SHEET CONTROL, JOB AND PATH PLANNING, AND EXCEPTION HANDLING**

The software controlling such a complex, parallel machine must be particularly flexible. It must be able to support the hardware reconfigurability, both off- and on-line (while the machine is running), and it must be able to scale to different machine sizes. It must also support the parallel operation of multiple print engines and multiple paper paths handling multiple jobs and optimize the sheet flow through the system to achieve the highest possible productivity. Finally, the system software must be able to handle exception events such as faults or sheet jams in a manner that allows unaffected portions of the system to keep running, taking advantage of the redundancy and flexibility of the hardware.

The control architecture that we chose [1-6] leverages the modularity, redundancy, and flexibility of the physical system design to achieve these goals. The architecture is shown in Figure 6. Each module of the physical system, nip (n), three-way director (t), straight-through director (s), printer, or feeder, has its own DSP (TI F2811) and its own module controller. This local module controller is responsible for driving actuators and monitoring sensors as well as handling communications. Module communication occurs over a CAN bus (Control Area Network - a fault-tolerant serial communications protocol).

Each sheet in the system (shown in red) is associated with a coordination entity called a sheet controller, which mediates communication among the local controllers, monitors the sheet's progress, and interfaces between the local controllers and the higher level of the control hierarchy. A new sheet controller is spawned for each sheet by a sheet controller factory, which in our system resides in a central PC. At the top of the control hierarchy is the planner/scheduler, which is responsible for generating routes and timing for all sheets through the machine.

Giving each module its own controller supports all the benefits of hardware modularity, including scalability and reconfigurability. The module controllers do not need to be aware of the overall structure of the system, and no centralized entity must keep track of details that it otherwise does not need; this is the principle of encapsulation, keeping knowledge where it is needed (much like object oriented programming). With local knowledge, the modules have a significant degree of autonomy, and can be delegated tasks by higher control levels. The modules then need simply to be able to accept delegated commands and then report back to the higher levels if there is a problem (escalation of error conditions). Complexity in system scaling is thus reduced. The nip module controllers use a proximal time-optimal servo (PTOS) control scheme, which has demonstrated good tracking performance in this system.

Nip spacings and module dimensions are defined by the shortest sheet that can be handled. Thus, all sheets are generally contacted by multiple nips simultaneously. The modules acting on a given sheet form a dynamic control group that changes as the sheet moves through the system. The sheet controller fulfills the coordination role for the control group. The sheet controller incorporates into the control group module controllers which are about to receive the sheet so that control is tightly integrated by the time the sheet arrives. The sheet controller stores the itinerary from the planner/scheduler and interprets and disseminates it to the control group. Sensory information is shared among the control group in a synchronized manner via the sheet controller. Finally, should the system not be able to meet the planner/scheduler's itinerary for the sheet, the sheet controller reports the situation to the planner/scheduler for exception handling.

The planner/scheduler also must support on-line reconfigurability and flexibility in the system. Its goal is to maximize the productivity of the entire, parallel machine. The planner/scheduler accepts multiple job requests and creates near-optimal plans to launch and guide sheets from feeders to output trays via multiple marking engines as needed. It can optimize performance using different criteria such as best system throughput, best component utilization and wear, or longest time before service. The planner/scheduler uses a model-based, on-line, temporal planning approach. It maintains a model of the machine that is separate from the reasoning algorithms, and can thus support different configurations. In addition, if the machine changes on-line, due to a sheet jam, for example, the planner can update its model based on feedback from the rest of the control system and can continue to reason about the best way to maximize productivity. The planning algorithm takes the system model, its internal model of the current system state (including sheets in flight), and the job requests as input and uses a heuristic-driven regression (backward) search through the system state space to find an optimal itinerary for each sheet. Planning efficiency is enhanced through reuse of heuristic computations between sheets and the use of a temporal constraint network to manage the constraints between resources in the system. In addition, the planner/scheduler contains a time-bounded search algorithm, so that a greedy plan can be found in a very short time and then improved upon. If an exception such as a jam occurs, the planner replans on-line for sheets already in flight in order to route around the jam, purge out-of-order sheets, and recreate the jammed pages.

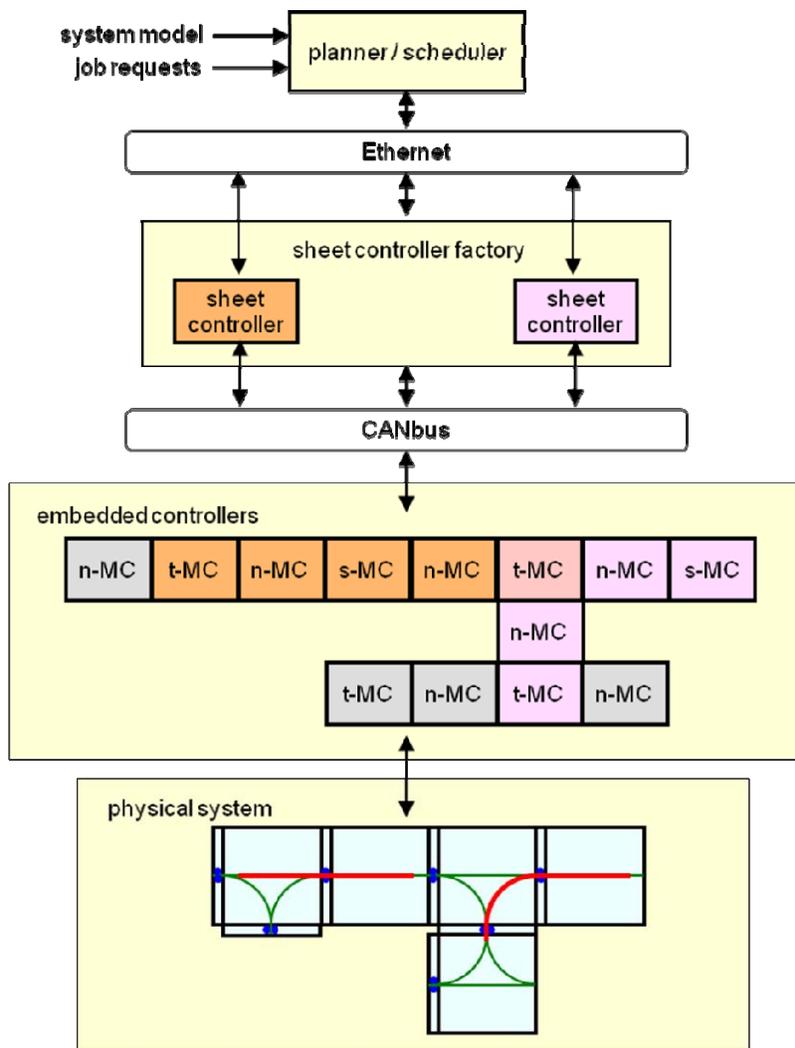


Figure 6. Hypermodular printer control architecture

For printers that can handle multiple jobs at the same time, optimal sheet-level planning by itself may not be adequate to achieve optimal productivity, since improper mixing of jobs can lead to productivity losses or cost increases that can be otherwise avoided. For example, printing two color jobs on a machine with one monochrome and one color engine can only keep half of the print engines busy. On the other hand, while color engines can print monochrome pages, their cost of printing is usually much higher than the black-and-white only engines. To overcome these problems, we developed an intelligent jobs manager that aims at producing the right mix of jobs to match the speed of different print engines.

We designed a so-called “minimax” job-mixing strategy in which the job with the highest *color-to-monochrome ratio* (defined as the number of color pages divided by the number of black-and-white pages) is paired with another job with the lowest color-to-monochrome ratio for simultaneous printing. This increases the likelihood that both the color and the monochrome engines are fully utilized at any given moment. Our experiment showed that the minimax strategy can increase productivity by 25% on average, compared to a random strategy for mixing jobs.

Once the job mix is decided, there remains the question of how fast each job should be printed relative to each other, since many color jobs have black-and-white pages too. We call this the “throttle control” problem, which is formulated as a linear program (LP) with an objective to maximize the number of pages printed per minute (ppm), subject to the constraints that (a) the number of color pages printed per minute is no greater than the combined ppm of the color engines and (b) the number of monochrome pages per minute is no greater than the combined ppm of the black-and-white engines. The decision variables being optimized correspond to the rates at which these simultaneous jobs are printed. For color jobs with relatively even distributions of monochrome pages, the solution to the LP gives high quality throttle control that results in near-optimal machine throughput.

## 5. RESULTS AND CONCLUSIONS

Using nip modules providing 1.25 m/s sheet speeds with sufficient torque at speed for smooth control and four marking engines with process speeds of 0.26 m/s (55 impressions/minute) system throughput was demonstrated at 210 impressions/minute, almost 4x the single processor throughput. Real time replanning and rerouting were also successfully demonstrated as well as rotary jam clearance with computer guidance and nip assist.

As a demonstration of on-line replanning and rerouting, a module in the fixture could be failed after a sheet had already launched, and the sheet would reroute around the failure. Additionally, if a sheet were to jam, the planner would reroute out-of-order sheets to the purge tray and recreate the jammed sheet and all subsequent sheets. Alternatively, the out-of-order sheets could be routed around inside the system until the jammed sheets were reproduced.

A systems analysis indicated that cost and reliability of single hypermodular systems should be closely comparable with custom systems when equivalent functionalities are enabled. However, if the high level of functionality enabled by the described hypermodular system is not required, then a coarser-grained transport system can be designed that achieves a possibly lower cost and/or higher reliability. Hypermodular designs, however, should support a wide variety of configurations (a platform of customizable products) and should allow broad system improvements as individual elements are improved.

The present work has demonstrated a parallel printing system which is deeply modular and reconfigurable in both the physical and software realms. There are clearly many trade-offs involved in the design of modular parallel printing systems and hypermodular approaches in particular. Aspects such as degree and ease of reconfigurability, parts count, scalability, etc. likely lead to different optimizations. However, it has been shown that building systems using fixed pitch placement of paper path elements, using marking engines with inputs and outputs commensurate with the same pitch, and using a hierarchy of modularity (e.g. towers, highways, a few types of hypermodules and a reduced set of standardized parts) can lead to the goals of scalability, easy reconfigurability, path redundancy, etc.

## REFERENCES

- [1] Fromherz, M. P. J., Crawford, L. S., and Hindi, H. A., “Coordinated Control for Highly Reconfigurable Systems,” Proceedings of Hybrid Systems: Computation and Control (HSCC, Zurich, Switzerland, 2005).
- [2] Hindi, H., and Crawford, L. S., “Control in Printing Systems: Modular Reconfigurable Media Paths,” Proceedings of the American Control Conference (2009).
- [3] Crawford, L. S., Hindi, H. A., Zhou, R., and Larner, D., “Synchronized Control in a Large-Scale Networked Distributed Printing System,” Proceedings of the IEEE International Conference on Robotics and Automation (May 2009) pp. 158—163.
- [4] Hindi, H. A., Crawford, L. S., Zhou, R., and Eldershaw, C., “Efficient Waypoint Tracking Hybrid Controllers for Double Integrators using Classical Time Optimal Control” Proceedings of the 47th IEEE Conference on Decision and Control (December 2008).
- [5] Ruml, W., Do, M. B., and Fromherz, M., “On-line Planning and Scheduling for High-speed Manufacturing,” Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS, 2005).
- [6] Do, M. B., Ruml, W., and Zhou, R., “On-line Planning and Scheduling: An Application to Controlling Modular Printers,” Proceedings of the 23<sup>rd</sup> AAAI Conference on Artificial Intelligence, (NECTAR Track, 2008).