

# Managing Emergent Behavior in Distributed Control Systems<sup>1</sup>

**H. Van Dyke Parunak, Ph.D.**  
Scientific Fellow  
Industrial Technology Institute  
PO Box 1485  
Ann Arbor, MI 48106

**Raymond S. VanderBok**  
Principal Member of Technical Staff  
Industrial Technology Institute  
PO Box 1485  
Ann Arbor, MI 48106

## KEYWORDS

Nonlinear Dynamics, Chaos, Emergent Behavior, Distributed Control

## ABSTRACT

Distributed control architectures are becoming increasingly popular because their modularity makes them easy to install, configure, and modify. These benefits do not come for free. A population of asynchronously executing processes without central top-down control can exhibit unexpected or “emergent” behavior at the system level. To the plant engineer, this behavior may look like noise or error conditions, but it is generated by deterministic interactions among control elements, not random events or unit malfunctions, and it must be managed accordingly. Drawing on experiences in the Auto Body Consortium’s Intelligent Resistance Welding project, we illustrate the potential for this kind of behavior among welding robots in an automotive body shop and in other applications, show how recent research in nonlinear systems theory and agent-based control can be used to detect and manage such interactions, and identify some requirements that these agent techniques place on emerging standards for data and control models.

## POTENTIAL FOR EMERGENT BEHAVIOR

This section defines emergent behavior, identifies its causes, and illustrates it in several industrial contexts.

### *What is Emergent Behavior?*

The proliferation of VLSI<sup>2</sup> computation and inexpensive, robust networking means that control systems are increasingly being designed as distributed systems. The previous paradigm of a central PLC with separate I/O lines for each sensor or actuator is giving way to smart devices networked with one another and with a community of small controllers, each responsible for some portion of the entire system.

In the former paradigm, systems designers defined the desired overall system behavior and enforced the coordination of the various parts in the central controller. The new approach encourages designers to focus on the behaviors of the individual devices and subsystems. This change of focus is beneficial, because it leads to highly modular systems that can be more readily configured and maintained than

---

<sup>1</sup> Some of the work reported in this paper is supported by the Intelligent Resistance Welding project, funded by the NIST MVMT ATP program through the AutoBody Consortium.

<sup>2</sup> Very Large Scale Integration

monolithic systems. At the same time, the distributed approach can produce system behavior that is more complex than the behavior of the individual components. We call such behavior "emergent behavior," because it emerges from the interactions within the overall system, often in ways not intended by the original designers. As engineers gain more experience with distributed systems, the potential for beneficial emergent behavior is expected to form one of the major benefits of this approach [Parunak 1997]. Such behavior can also surface in undesirable ways, and engineers need to give attention to it.

Emergent behavior can take different forms. Sometimes the system will be attracted to a fixed, stable point, which may or may not correspond to the behavior intended by the designer. Sometimes it may oscillate. Sometimes it may enter a formally chaotic regime, which superficially looks like random behavior.<sup>3</sup> This latter case is particularly important. A system that behaves randomly because of noise in the environment can often be corrected by improved quality control, but these methods will not improve formally chaotic behavior that emerges in a distributed system. In this case, the solution lies in structural modification or parameter tuning, not tighter control over varying environmental conditions.

### *Sources of Emergent Behavior*

In a linear system, the whole is equal to the sum of the parts. Various forms of nonlinearity are responsible for emergent behavior. Three of the most common sources of nonlinearity are capacity limits, feedback loops, and temporal delays.<sup>4</sup>

In a manufacturing setting, buffer sizes are a common form of capacity limit, leading to nonlinear couplings among devices. When a workstation attempts to draw parts from an empty buffer, or feed them into a full one, it must stop operation until the occupancy of the buffer moves away from the limit. The tendency of JIT strategies such as Kanban to shrink buffer sizes increases the frequency with which a system is likely to exceed a buffer capacity, and thus behave in a nonlinear fashion.

Feedback loops are ubiquitous in modern control systems. In addition to feedback in control, some manufacturing processes introduce feedback loops in the flow of material through the plant. For example, respot stations reprocess defects in automotive welding lines, and semiconductor lines route a wafer repeatedly through the same set of tools to build up successive layers of circuitry. Other feedback loops are unintentional, such as electromagnetic coupling between different devices, or the impact of the current drawn by one device on the plant's supply voltage and thus the behavior of other devices.

---

<sup>3</sup> The popular press uses the term "chaotic" informally to describe emergent behavior in general. We use "chaotic" in its formal sense, to describe one particular form of emergent behavior with specific distinguishing characteristics (e.g., sensitive dependence on initial conditions, a white noise power spectrum, and a trajectory in phase space that does not intersect itself, occupies non-zero but finite volume, and has fractal structure). The distinctive feature of emergent behavior (including fixed points and oscillations as well as formal chaos) is that it can appear unexpectedly from the interactions of explicitly designed components.

<sup>4</sup> The term "nonlinear," like "chaotic," has been broadly and inconsistently applied in the popular press. We use the term "linear" to refer to a linear autoregressive process (equivalently, one whose time series structure is completely captured in the frequency domain by the power spectrum, without the need for phase information). A nonlinear process is any process that is not linear. The resulting phase sensitivities lead to nonadditivity of component behaviors and thus to unexpected emergent behavior. While one can construct artificial examples of feedback loops and temporal delays that are linear in this sense, real-world systems of this sort commonly behave nonlinearly.

Even temporal delays in the movement of information and material between process steps can lead to nonlinearities. Information delays are steadily shrinking through the use of computer technology, but material movement will continue to be delayed by physical factors for the foreseeable future.

### *Examples of Emergent Behavior*

Emergent behavior can manifest itself in many different ways.

Consider, for example, the problem of supply voltage for a collection of high but intermittent loads, such as spot welding guns. If too many guns fire at once, the supply will be unable to provide the current needed across all guns, and some of the resulting welds will be defective. In one corrective strategy, the controller for each gun monitors the line voltage, and only fires when the voltage is above a certain threshold. When this strategy is applied to all the guns in a facility, the entire set of controllers will sense a low voltage condition and delay their firing. Then, when the voltage rises, they will all recognize the availability of power and fire nearly simultaneously, pulling the voltage down and leading to another mass withdrawal. As a result of this feedback among the controllers, the system will tend to oscillate between overload and under-utilization.

Material handling systems can exhibit emergent behavior in the form of "traffic jams," a phenomenon also observed in automobile freeways. In principle, one expects a line of cars on a highway (or automatic guided vehicles in a factory) to maintain even spacing and a constant flow rate. This constancy can be realized with a single centralized transport system, such as a powered conveyor. When vehicles are individually powered and controlled, they may maintain their separation by sensing their relative separation and adjusting speed accordingly. This feedback amplifies inevitable spatial fluctuations to form congested regions, separated by regions of lower than normal vehicle density.

The experience of a major automotive company shows how a bottleneck in part flow can emerge even in a material handling system whose individual components can keep up with line speed [VanderBok & Sauter 1990]. The configuration in question has three components: two non-accumulating conveyors and a gravity chute that feeds one conveyor from the other. The source conveyor is turned off when a part is ready to fall down the chute and a place is clear on the receiving conveyor. Even though each conveyor and the chute can individually keep up with line speed, interaction among these elements leads to a limitation in capacity. The cause of this emergent limitation is the delay introduced by the chute.

Parallel cooling pumps can exhibit an annoying emergent behavior. Inevitably, one pump has slightly less output than does the other. Over time, the buildup of sediment in the weaker pump's inlet grows faster than in the stronger pump, because of reduced flow. Soon the stronger pump is taking most of the load, at which point it shuts down because of overload. Now the weaker pump takes the load, but the load is limited by sediment. The increased flow clears away the sediment to the point where this pump is now carrying the full load, so it overloads and shuts down, too. The tradesman comes in to find two pumps shut down without sediment blockage. The pumps are restarted, only for the cycle to repeat.

The centralized nature of traditional control architectures has restricted their application to operations within a single company or plant, that is, within the domain of a single supervisory computer. Distributed techniques open up the possibility of peer-to-peer coordination among multiple firms, for example, in a supply chain. An emergent phenomenon that has been observed in manually operated supply chains illustrates potential behavior in automated ones as well. It is widely known that the variation in the demand experienced by a low-tier supplier is much wider than that experienced by the OEM, sometimes increasing by as much as two times from one tier to the next. Simulation experiments

show that this variation can be explained as a result of the delayed movement of demand information down the supply chain and of material back up.

## **TECHNIQUES FOR DETECTING AND MANAGING EMERGENT BEHAVIOR**

Research in nonlinear systems theory and agent-based control offers a variety of techniques for detecting and managing emergent behavior. Agent-based simulation is an important tool in both of these activities.

### ***Detecting Emergent Behavior***

As noted above, emergent behavior can exhibit a wide range of dynamics, including unintended fixed states, oscillation, and formal chaos. Some behaviors are hard to distinguish from noise. Other behaviors happen so quickly or slowly that it is hard for people to see the patterns. Thus the first step in dealing with this kind of behavior is to detect that it exists in a particular system.

Emergent fixed points (such as a bottleneck in a manufacturing line) can be recognized in two ways: they are unexpected, and they often change location when the configuration of the system changes. No one is surprised when a bottleneck appears at a machine whose capacity is lower than the other elements in a system, but when it appears at an element whose individual characteristics seem adequate, emergent behavior becomes a reasonable suspect. Because this behavior is generated by nonlinear interactions with other elements in the system, changes in those elements can lead to a shift in the fixed point. These two clues lead to a suspicion of emergent behavior, but identifying the specific nonlinearity responsible can be difficult. Simulation experiments with a model of the system are a powerful tool to learn how the overall system behavior changes as individual elements are modified.

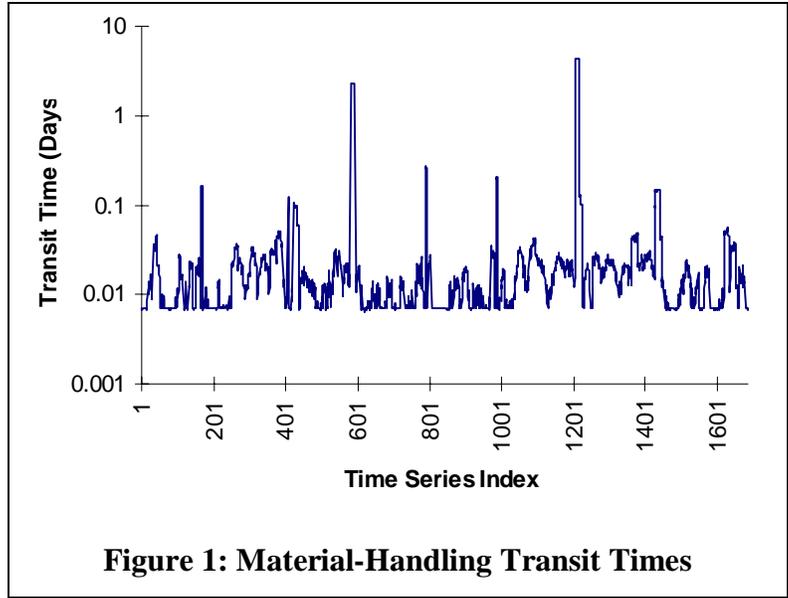
Oscillation is not usually a designed behavior in manufacturing systems, so in most cases the very presence of oscillation is unexpected and attributable to emergent behavior. If a number of different frequencies are present, oscillation may superficially look the same random noise, but can be detected by applying a Fourier transform to time series data. The availability of such transforms even in PC spreadsheets makes such a test easy and inexpensive. White noise (a flat frequency spectrum) suggests either random noise or the presence of emergent chaos, while individual dominant frequencies indicate the presence of oscillations. In this case, the frequencies themselves can be a clue to the lengths of delays involved and therefore a guide to identifying the nonlinearity that is causing the problem. Again, simulation methods are the most promising confirmatory technique, though they are still in their infancy.

Emergent chaos and randomness can give the same superficial appearance of unordered and unpredictable activity, but they have different origins and must be managed differently. True randomness results from unexpected variability in the environment. Emergent behavior results from the nonlinear interaction of completely deterministic processes. We must be able to tell the difference between the two situations if we are to manage them effectively.

The "smoking gun" that distinguishes emergent chaos from true randomness is the presence of an underlying determinism or order. Scientists have developed a wide variety of statistical tools to search for such order. This paper illustrates two of these techniques that researchers from the Industrial Technology Institute and the University of Michigan have applied to shop-floor time series of material handling systems. A convenient tutorial introduction to a number of other techniques is available in the form of a PC-based program [Sprott & Rowlands 1996].

The data in both of these examples come from an American factory that produces several different models of specialty vehicles, and from a portion of that factory that is similar in its processes and structures to an automotive body and paint shop. Parts are mounted on carriers, which in turn travel on a power-and-free conveyor system. Sensors placed at critical points in the process record the time and identity of passing carriers.

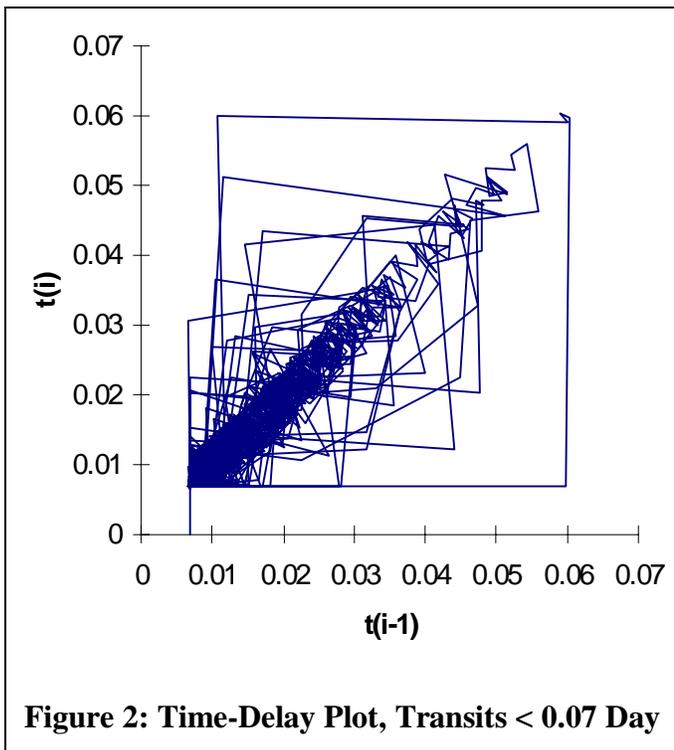
The first technique, the time-delay plot, can be applied to the sequence of transit times of carriers past two successive sensors along the material handling system. Figure 1 shows a plot of over 1600 successive transit times between a pair of successive readers over a two-week period. The vertical axis is logarithmic, showing that the delays range over three orders of magnitude. Transit times greater than 0.1 day recur fairly regularly, but shorter delays appear to be random.



**Figure 1: Material-Handling Transit Times**

Figure 1 shows a plot of over 1600 successive transit times between a pair of successive readers over a two-week period. The vertical axis is logarithmic, showing that the delays range over three orders of magnitude. Transit times greater than 0.1 day recur fairly regularly, but shorter delays appear to be random.

We denote the first transit time in the series by  $t(1)$ , the second by  $t(2)$ , and so forth. To generate a time-delay plot, we plot a point for every two successive transit times, with an  $x$  coordinate given by the first of the two ( $t(i-1)$ ) and the  $y$  coordinate given by the second ( $t(i)$ ). Even after eliminating the longer transits, the resulting picture (Figure 2) is far from random. A more detailed analysis [Parunak 1995] shows that the square structures results from line stoppages (a problem that can be corrected by better machine maintenance), while the diagonal band reflects delays due to traffic jams (an emergent behavior, which requires more careful system tuning).



**Figure 2: Time-Delay Plot, Transits < 0.07 Day**

while the diagonal band reflects delays due to traffic jams (an emergent behavior, which requires more careful system tuning).

Another powerful tool is the  $S$  statistic [Savit & Green 1991], which identifies the presence of deterministic dependencies among successive elements in a time series. Using this and other techniques, [Manuca, Parunak, & Savit 1997] identify stable and unstable loading policies in the vehicle plant described above, and recommend an operating policy to maintain the plant at a high level of throughput.

Techniques such as these can identify potential cases of emergent behavior in manufacturing systems. Agent-based simulation is an important tool to confirm these hypotheses. This approach to system modeling represents each entity in the manufacturing system as a software agent, a software object (in the sense of "object-oriented

programming") with its own thread of control and its own initiative [Drogoul 1993]. The Swarm system developed at the Santa Fe Institute [Langton et al. 1997] is a sophisticated and widely available platform for such work. Engineers can incorporate in such a model the forms of deterministic interaction they suspect may be responsible for erratic plant behavior, thus verifying the diagnosis provided by mathematical analysis.

### ***Managing Emergent Behavior***

When emergent behavior is responsible for undesirable plant behavior, several strategies can control it.

The most direct approach is to eliminate the nonlinearity in the system that generates this behavior. Usually, this approach is not feasible. Buffer limitations and non-instantaneous movement of material are unavoidable facts of life in manufacturing, imposed by fundamental laws of physics, and removing feedback loops of information and material would reduce the plant's susceptibility to deterministic emergent behavior only by surrendering it to environmental noise.

One classical approach is to lock all the details of the plant's operation under a centralized control algorithm. This approach may be feasible for a facility that serves a stable market, in which the expense of developing large custom software systems can be amortized over a long period. For most companies in today's economy, the software engineering benefits of decentralized systems are overwhelming, and these economic considerations block this approach to control almost as inevitably as physical considerations do the previous one.

Another classical approach is to damp out any variation in the system by limiting performance well below the optimal level. For example, decreasing the line rate and thus the density of automatic guided vehicles can reduce the incidence of traffic jams in a material handling system. While this approach greatly simplifies the operation of the system, it seriously compromises its efficiency, and is not competitive in modern markets.

A fourth broad category of management techniques takes advantage of the distributed computational control elements that lead to the problem in the first place, and draws on techniques of agent-based coordination. These micro-controllers, acting as agents, can monitor their environment and take local action to damp undesirable forms of emergent behavior. Consider several examples.

The naïve strategy employed by weld controllers that wait for the line voltage to rise above a certain threshold before firing is an example of an agent-based technique, but as noted above, this approach leads to oscillation. A slight modification avoids this problem. When a weld controller wishes to fire, it examines the line voltage, and if it is above a threshold, it fires. If the voltage is too low, it generates a random number and goes to sleep for that period of time, then tries again. The random numbers break the symmetry of behavior among the different controllers so that they do not synchronize. In theory, under this strategy some controllers may not be able to complete their welds in the time window available. Similar arguments were used at one time to argue against Ethernet as a factory automation backbone. As in that case, with appropriate load levels and system tuning this algorithm can give satisfactory results in practice.

More complicated schemes of inter-agent coordination include bidding protocols, in which each controller bids for access to a scarce resource (such as the power mains). In the welding example, the mains can support a certain number of concurrent welds. Each controller is assigned a budget of funny money periodically, and bids for one of the available slots. The amount it bids is calculated based on how wide a window the controller has in which to complete the weld, and perhaps on the structural

importance of the weld in the product. Slots are awarded to the highest bidders. As one controller's budget is depleted, it must wait its turn, so that other controllers that were unable to get a slot earlier have a chance to get one later. This kind of synthetic economy has been used to manage the climate control system in an industrial research facility, and outperforms a conventional system that had been hand-tuned over a two-year period by professional facilities engineers [Clearwater et al. 1996].

As these examples make clear, agent-based control schemes require careful tuning to insure that they alleviate the undesirable emergent behavior and do not cause other problems of their own. Agent-based simulation is again an indispensable tool for tuning and testing these algorithms before they are applied to actual facilities.

## **REQUIREMENTS FOR DATA AND CONTROL MODELS**

Emergent behavior is an inescapable consequence of modern control architectures. Vendors of control technology and system designers can take a number of steps to make such behavior easier to detect, evaluate, and modify. Initially, these capabilities will reduce the potential damage of unexpected system behavior. Ultimately, they will lead to techniques for taking advantage of this behavior to make systems more agile and robust.

Emergent behavior cannot be predicted analytically, but must be detected in an operating system. Thus control elements must incorporate hooks for monitoring system performance in real time. For example, the controller for a material transport system can record transit times, buffer occupancies, and blockages, indexed by the time and location of each measurement.

The mathematical tools used to detect emergent behavior generally look for patterns in time series. The longer the time series, the easier it is to detect patterns and the more confidence one can place in those patterns. Thus these performance-monitoring tools need to be automated, and readily integrated into a communications backbone that can route successive readings to a central station for analysis and archiving.

When a pattern in time series data suggests an emergent behavior, the existence of the behavior is verified through simulation experiments with a model of the system. Controls vendors can greatly facilitate such experimentation by making executable device models a standard component of device documentation. Ideally, these models should be available to systems integrators as an electronic catalog to facilitate the construction of system models during design. The same models can then be used to manage emergent behavior during operation of the completed system.

Performance monitoring hooks, data communications mechanisms, and executable device models all support the detection and verification of emergent behavior. Efficient modification of that behavior in a running system requires richer functionality at the level of individual devices and controllers. For example, in addressing the problem of weld power management described above, most sequencers available today are not flexible enough to impose a random delay on firing. In addition, devices should be dynamically reprogrammable, so that when adjusting parameters in a simulation has solved a problem, the new parameters can be downloaded easily to the real devices in the factory.

These various device-level requirements will be greatly leveraged by providing an integrated development and deployment architecture and workbench. This facility should include data archiving and retrieval, a suite of mathematical tools for nonlinear systems analysis, and the basic simulation engine for executing integrated systems models.

## CONCLUSION

The move toward assembling manufacturing systems from distributed asynchronous control components inevitably makes them susceptible to emergent behavior. Knowledge of the individual devices that make up the system is not sufficient to predict when such behavior will occur or what it will look like, and it may be considerably more complex than any of the components that are involved in producing it.

Emergent behavior may be either desirable or undesirable. As our understanding matures, it is likely that future system engineers will be able to take advantage of emergent behavior to develop systems that are more robust and agile than current designs. Our present state of knowledge permits engineers to detect the existence of such behavior in the first place so that they can assess whether or not it is harmful and take corrective action.

The most important tools for dealing with emergent behavior are nonlinear mathematical analysis of time series generated by a system, and simulation experiments with a model of the system. Engineers should add these tools to their problem-solving repertoire so that these new issues can be identified and addressed. In addition, the next generation of control technology should support these techniques by providing for easy monitoring of operational data and straightforward construction and manipulation of simulation models.

## REFERENCES

- [Clearwater et al. 1996] S.H.Clearwater, R.Costanza, M.Dixon, and B.Schroeder, "Saving Energy Using Market-Based Control." S.H.Clearwater, ed., *Market-Based Control: A Paradigm for Distributed Resource Allocation*. Singapore: World Scientific, 253-273.
- [Drogoul 1993] A.Drogoul, *De la Simulation Multi-Agents à la Résolution Collective de Problèmes*. Ph.D. Thesis, University of Paris IV.
- [Langton et al. 1997] C.Langton, R.Burkhart, and G.Ropella, "The Swarm Simulation System." <http://www.santafe.edu/projects/swarm/>
- [Manuca, Parunak, & Savit 1997] R.Manuca, H.V.D.Parunak, and R.Savit, "Getting the Big Picture: A Nonlinear Data-Analytic Approach to Factory Monitoring and Control." Submitted to *Journal of Manufacturing Systems*.
- [Parunak 1995] H.V.D.Parunak, "The Heartbeat of the Factory: Understanding the Dynamics of Agile Manufacturing Enterprises." ITI Technical Memorandum. Available at <http://www.iti.org/~van/heartbt.ps>
- [Parunak 1997] H.V.D.Parunak, "Go to the Ant: Engineering Principles from Natural Agent Systems." Forthcoming in *Annals of Operations Research*. Available at <http://www.iti.org/~van/gotoant.ps>
- [Savit & Green 1991] R.Savit and M.Green, "Time series and dependent variables." *Physica D* 50, 95-116.
- [Sprott & Rowlands 1996] J.C.Sprott and G.Rowlands, *Chaos Data Analyzer: The Professional Version*. Software and manual available from Physics Academic Software, 1-800-955-8275.
- [VanderBok & Sauter 1990] R.VanderBok and J.A.Sauter, "A Case Study: Integrated Methods and Tools for Optimization." *Autofact '90 Conference Proceedings*, November 12-15, Detroit, Michigan, Society of Manufacturing Engineers, MS90-742.