Autonomous Systems

Microsoft Autonomous Systems for Manufacturing

Solution Playbook for Building AI to Control and Optimize Manufacturing Lines
June 2019

What is a Solution Playbook?

A solution playbook is a template for designing and training AI based autonomous systems. We call this process machine teaching.

What is Microsoft Autonomous Systems?

Microsoft Autonomous Systems is a toolchain for building, training and deploying sophisticated decision-making AI to control or optimize industrial systems. There are four primary components:

1. **Machine Teaching**: A technique that can human and machine intelligence

2. **Simulation**: A training environment for decision-making AI

3. **AI Engine**: Automated management of neural networks and algorithms

4. **Runtime**: deploy and scale your models in the real world

What kind of manufacturing lines can I control using AI built with Microsoft Autonomous Systems?

Much work has been done to apply Machine Learning (ML) and Artificial Intelligence (AI) to areas of manufacturing like Predictive Maintenance, but very little has been done to help control or optimize the core asset of continuous manufacturing operations: the manufacturing line itself.

This solution playbook provides a template for using Machine Teaching to develop AI systems that can autonomously control and optimize continuous and batch manufacturing lines. Here are a few examples:

- Train an AI to control the input feed policy for a polymer reactor.
- Train an AI to control a continuous evaporation process.
- Train an AI to control a food extrusion process
- Train an AI to control a thermal reactor for silicon manufacturing.
- Train an AI to control a cement manufacturing process.
- Train an AI to control a carbon fiber laying process.

How can AI help me improve my continuous manufacturing line operations?

Optimization goals

Continuous manufacturing lines are typically optimized for the following goals:

- **Throughput**: The primary optimization goal is the amount of manufactured goods that meets quality standards produced per unit time. Here are a few examples:
    - Pounds (lbs) of pet food produced by an extrusion process per hour.
    - Liters (L) of ammonia produced in a chemical reactor per minute.
    - Tons (t) of cement produced per hour by a continuous manufacturing process.
- **Efficiency**: When the product demand is high, it makes sense to maximize throughput. In this case, it might prove effective to run the lines faster even though more product is scrapped because the overall throughput will be higher. However, when the product demand is not sufficiently high it makes more sense to optimize efficiency: what percentage of product meets quality standards without requiring any rework.
- **Energy Consumption**: Energy is often a key input, and the line many need to be tuned for energy efficiency to ensure margins are adequate.
- **Quality**: Quality standards must be met in order to generate revenue from the product of the manufacturing line.

Current Methods

In order to evaluate the potential impact of AI on the continuous control of manufacturing lines, we must first evaluate the current control methods. In most continuous manufacturing scenarios that we've encountered, the equipment is controlled at a high frequency (10hz – 100hz) by low level controllers. The setpoints for those low-level controllers are given by operators or Advanced Process Control (APC) systems:

	Low-Level Equipment Control	Supervisory Control
Operator Control	PID or MPC controllers control the equipment at a high frequency.	Operators provide supervisory controller setpoints in real time during the operation.
Advanced Process Control		MPC or other advanced process control algorithms provide supervisory decisions in real time during the operation.

Limitations of Current Methods

Both operators and APC systems have known limitations that prevent the manufacturing line from achieving maximum throughput and efficiency at quality:

Challenge	Operator Limitations	APC Limitations
Many manufacturing scenarios that each require different control decisions. Here's a few examples: • Raw material permutations. For example, the composition of the corn meal ingredient to pet food varies in fat, protein, carbohydrate and water content when it enters the process. • Line configuration scenarios • Machines wear states • Product specifications.	• Very few operators are experts. • Experts train for years before they can supervise the line from the control room. • Each expert tends to favor an approach or school of thought that works best in specific scenarios. • Line performance varies across experts depending on training and experience.	APC systems can struggle when operating conditions vary widely.
Many parameters to control continuously. The number of potential knobs to adjust is high and it is difficult to know which knobs to turn to control for a specific scenario.	Humans struggle to manage more than 3 or 4 variables simultaneously, yet most manufacturing lines require control of 10 – 100 parameters.	APC systems must be tuned for different scenarios. Typically, some scenarios result in poor performance.
Multiple and changing optimization goals. For example, under high product demand the primary goal is throughput. Under lower product demand, the primary goal is efficiency. When energy is expensive, energy efficiency becomes a primary goal.	Operators must be trained to control the process differently as goals change.	Most APC techniques optimize for one thing at a time, and it is difficult to adjust to a mix of goals.

High supervisory control decision frequency.	Some processes simply move too quickly for humans to control well, even at a supervisory level.	Some APC techniques like MPC are compute and time intensive at inference.
Low margin of error between good quality and bad, good throughput and bad.	Experts outperform less experienced operators by significant margins. For example, one client reported that their expert operators produce a batch of product in half the time that it takes less experienced operators to produce the same amount of product at quality.	Often extremely accurate models are required to design and build APC systems.
Many rules and constraints that must be followed.	It is difficult to document, train for and track adherence to the rules and constraints.	APC systems can be costly and time consuming to build and test.

Opportunity for AI to Provide Superior Control

Microsoft Autonomous Systems offers two innovative techniques that can produce AI that outperform existing methods:

- **Machine Teaching**: Method to modularize manufacturing line control into a hierarchy of independent decisions that can be extracted and codified from expert operators.
- **Deep Reinforcement Learning**: Deep Reinforcement Learning provides a method to train AI to make highly non-linear decisions across a variety of scenarios:
  - This method has produced groundbreaking results in game research, in Go, Atari games, and DOTA)
  - Microsoft has shown the method to be effective for industrial systems at Shell, Schneider Electric and others.

Opportunity	Machine Teaching	Deep Reinforcement Learning
Many manufacturing scenarios that each require different control decisions.	Interviews with expert operators and data science analysis can each provide ways to build a hierarchy of simpler control decisions.	Deep Reinforcement Learning (DRL) can develop AI that is very adaptable to diverse scenarios.
Many parameters to control continuously. The number of potential knobs to adjust is high and it is difficult to know which knobs to	Expert operators can often help reduce the difficulty of the learning problem by describing which parameters are relevant in each scenario	DRL can learn to control high-dimensional processes, and the exploration during training can help find creative policies

turn to control for a specific scenario.		that work better than current procedures.
Multiple and changing optimization goals. For example, under high product demand the primary goal is throughput. Under lower product demand, the primary goal is efficiency. When energy is expensive, that energy efficiency becomes a primary goal.	Machine teachers can describe which goals matter when, letting the AI learn how to optimize for them.	The training regime of practicing in simulation across many scenarios and being rewarded based on optimization goal forces AI brains to learn how best to control the line as optimization goal priorities change.
High supervisory control decision frequency.		AI based systems can react very quickly.

What are the prerequisites to training an AI brain to control my manufacturing line using Microsoft Autonomous Systems?

There are three necessary elements:

- Measure the Process
- Simulate the Process
- Find Machine Teachers

Measure the process

Each variable that is used by operators or APC systems to calculate process control decisions, must be measured and stored at the supervisory control frequency in order to train an AI for autonomous control.

Quality measurements, whether taken continuously by an automated system or whether taken in a lab on samples of line output, must be recorded and stored in a system where the data can be programmatically accessed for deployment of the AI system.

Simulate the process

Microsoft autonomous systems train by practicing in a simulation and being rewarded when they achieve optimization goals. Simulation is crucial to the success of your autonomous manufacturing system.

Simulation Strategy	Advantages	Disadvantages
First Order Simulation Model	Accurate	Long development time. Tend to be very slow and compute intensive. Sometimes difficult to model. Licensing concerns, integration challenges
Model from data or proxy model	Fast, easy to connect, develop using process data only	Data science talent required, lower fidelity, sim to real transition

Hybrid – Train on proxy model and First order simulation model	Can drastically speed up training while still providing high-accuracy results	Requires development of both proxy and first-order model. Training setup more complex.
Hybrid – Use multiple simulations to model different parts of the process	Modular, suggests machine teaching approach	May be time consuming and difficult to integrate together.

Regardless which simulation strategy you choose, your simulation environment must meet the following criterial in order to successfully train an AI brain to control optimize your manufacturing line:

Accurate

The response of the simulator to control actions and environment variables must match reality close enough that a good policy in simulation is a good policy in real life. Because the AI learns over time, a fast and roughly accurate model that simulates control and measurement noise can often work better than very a slow and accurate model.

Note that you do not necessarily need to simulate the underlying complex physical or chemical processes behind some of the complexity. For example, there may be complex physical reasons for temperature increases in a particular machine. If the way to react to that doesn't depend on the details of why it gets hot, it's enough to simulate normal and high temperatures, not the underlying cause.

Fast

Each AI brain training session requires anywhere from 1M to 100M steps of the simulation model. Depending on the complexity of the AI system you are training, this may take a long time. We set a target of 8 hours to complete a brain training experiment. Use the following table to help determine the feasibility of your simulation environment for AI brain training:

Simulation Speed (Time to return one time step at control frequency)	Time to train Brain of average complexity 1 sim, 10M iterations	Parallel Instances Required to Run 10M iterations in 8 hours
1ms	2.7 hours	1
1s	115 days	340
1m	Not feasible	21000
1h	Not feasible	Not feasible (1.2M!)

Configurable

To train a Brain to control the process in many different scenarios, the simulator must be able to simulate those scenarios, and it must be possible to configure it to run a particular scenario on demand.

Machine teachers

Our approach is predicated on the fact that the subject matter experts about your process must be involved in teaching the machine, to help it learn more quickly, and ensure that the solution covers all the nuances of what it takes to succeed in production. This means that people with deep familiarity in the current control methods need to be included in the project from the beginning to the end.

Methods: Machine Teaching for Continuous Manufacturing

The machine teaching approach combines standard engineering techniques such as problem decomposition with AI. This section walks you through defining the problem, identifying subject matter expertise that can help decompose the problem into multiple *concepts* or constrain the solution space, then teaching each concept.

Specify the problem

What are you controlling?

What are the knobs? Usually controlling settings for low-level controllers that are responsible for maintaining set-points, etc.

What are the optimization goals?

Examples include:

- Minimize changeover time: switching the line to a different product
- Maximize throughput
- Maximize efficiency

The optimization goals may change over time — for example, switching from maximizing efficiency in normal operation to maximizing throughput when demand is very high.

What are constraints on system state?

Implicit in the optimization goals above are constraints that must be satisfied while optimizing. Examples:

- Measures of final product quality: density, shape, color, fineness, etc
- Safety constraints: temperatures, pressures,
- Equipment constraints: amounts of material in various parts of the line.

What are the constraints on actions?

This includes global constraints (“the screw temperature setting should always be between 60C and 120C”), as well as situation-specific constraints (“when the pressure is above 150 kPa, the temperature setting should be below 90C”).

What range of scenarios should the solution cover?

Variability in process scenarios is one of the primary reasons for implementing an AI solution. A key first step is to define the target scenarios: what types of situations, parameter ranges, system modes. Examples:

- **Input variability**: The moisture content of the starting pellets can be between 12% and 34%
- **Input cost:** Electricity cost can vary from $0.08 to $0.14 / kWh
- **Equipment variability:** the diameter of the extrusion nozzle varies between 5.2 and 5.4 mm due to wear and manufacturing differences.
- **Sensor variability:** The temperature sensors in the reactor can be off by up to 2C.

What information is available to base automated control decisions on?

- What sensors are available?
    - This includes traditional sensors that measure pressures, and temperatures, as well as more complex sensors like cameras and audio sensors.
- Sanity check: How is process controlled today? Is enough sensor data available? For example, if operators look at the color of extruded material or listen to the noise a machine makes to make decisions, there need to be sensors that capture the relevant info.

Codify Subject Matter Expertise on how to control the process

In addition to specifying the problem correct, subject matter experts can help teach the AI by guiding how it learns. This section describes several ways to guide.

What skills and strategies do operators use?

Skills and strategies are ways to control the process that help get good results but are hard to describe precisely and thus hard to program. For example, when manufacturing polymers, one strategy is to get the melting point right, then start to control for the desired density.

What rules are used to control the process?

Rules are policies that precisely describe what the controller should do in certain situations. For example: whenever the pressure goes above 175 kPa, open the relief valve.

What aspects of the problem can be controlled separately?

Just as in traditional engineering, tackling parts of the problem one at a time can make it easier for the AI to learn. Subject matter experts can define which parts are independent enough that it's possible to control them separately. For example:

- For cement manufacturing, the kiln and the mill can be controlled separately.
- The temperature of the extrusion nozzle and the extrusion screw speed can be controlled separately when producing certain products, but are intertwined and need to be controlled jointly to meet quality criteria for other products.

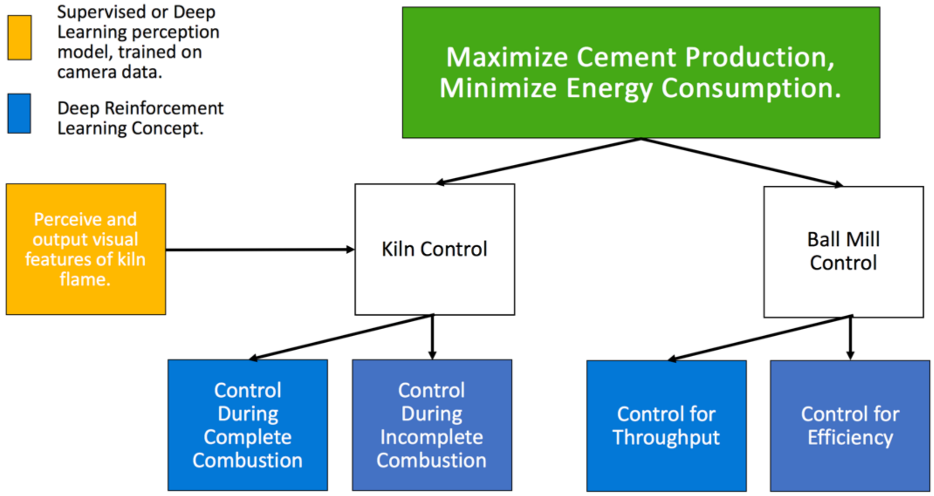
In some cases, can control systems separately given some higher level interaction. For example, throughput between sequential machines needs to be matched up, but they can be independent given a target throughput setting.

In addition to expert input and first principles, data-science-based analysis on historical logs can help assess independence among control areas. An additional heuristic is that if you can simulate part of the process separately, it can most likely be controlled separately.

Define a concept plan

Based on your earlier analysis of which parts of the process can be controlled separately, and where learning is necessary, define a concept plan. A concept plan defines how state information about the process gets turned into control decisions. Before we dive into the details, here is an example to convey the idea:

Concept plan for cement production



Types of concepts

There are three main types of concepts.

1. First are concepts that turn raw sensor inputs into actionable information, such as determining whether a kiln in cement manufacturing is running with complete or incomplete combustion based on images of the flame.
2. Second are programmed concepts that implement rules and procedures. These can be as simple switching a subsystem on or off based on a threshold, or as complex as an existing advanced controller, e.g. using model-predictive control.
3. Finally, there are learned concepts, which use machine learning to determine how to solve their piece of the overall problem.

Concept network structure

Concept plans can include parallel, hierarchical, and sequential components.

- Parallel means multiple concepts each controlling a certain part of the process.
- Hierarchical means a high level concept that selects which sub-concept to apply at any given time. This can be a pre-programmed decision ("if pressure > X, use the high-pressure controller, otherwise the low-level one"), or can be a more complex learned concept that learns when each skill is appropriate.
- Sequential components are a simple case of hierarchical, where the selection policy is to use sub-concepts in a certain order.

	Examples
Parallel	Cement manufacturing: One concept controls kiln, a parallel concept controls mill.
Hierarchical	Cement manufacturing: The kiln control policy is very different under complete and incomplete combustion. A selector based on current combustion type chooses the appropriate sub-concept.

Sequential	Polymer manufacturing: first run a concept that gets melting point of polymer to be right, then run a concept that gets the density to the target value.

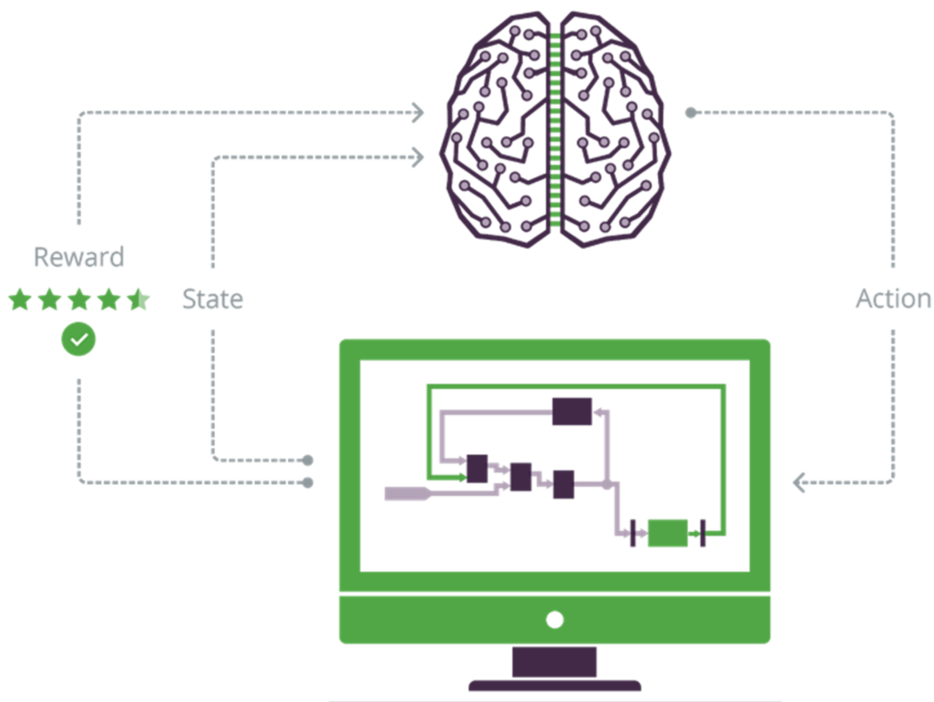
How to program each concept

- For rules, program the rules directly.
- For incorporating existing controllers or perception models, plug them in.
- For concepts where the correct behavior isn't clear, use learning.

Teaching learned concepts

Reinforcement learning

Our platform for machine teaching is built on Deep Reinforcement Learning under the hood. Reinforcement learning (RL), illustrated below, is a machine learning technique for controlling or optimizing a system or process. Deep in Deep RL refers to using deep neural networks to capture the problem complexity. In RL, an agent takes actions in an environment, getting feedback in the form of reward. The agent explores different action strategies or policies and learns to maximize cumulative reward.



Controller inputs

Controller inputs are also called "state" in RL. The system will learn what output (also called action) is best in any state. To make this work, the state needs to include all relevant information needed to make the decision. This typically involves current values of process variables, and often some history of the process variables as well. History is especially useful when the system must adjust to unknown variability in the underlying process – e.g. machine wear, sensor noise, etc. The history lets the Brain behave differently in response to different system dynamics.

Control outputs

Controller outputs are also called "action" in RL. Actions can be binary (e.g. "on/off"), discrete (e.g. "put the system into one of several modes"), or continuous ("set the target temperature for the reactor to 152 C").

Optimization objectives and constraints

Your objectives and constraints will determine the rewards you provide to the Brain, and when to terminate a training episode and have the Brain try again. Reward functions should guide the Brain toward the desired behavior, providing more reward as process objectives are improved. When the Brain breaks a constraint during training, terminate the episode and provide a reward penalty to help the Brain learn. Reward and terminal condition definition can get complex, but decomposition into sub-concepts can help alleviate this by dividing the problem.

Configurations and curriculum

As described previously, reinforcement learning is particularly applicable when the control system must be robust to variability in hidden process variables. To ensure that the Brain learns to be robust, it must be exposed to the full range of variability during training. The simulator configuration allows the machine teacher to define what configuration ranges to train on.

The simplest approach is to simply specify the full range of variability required from the beginning. If the system can't learn that right away, curriculum learning can help – here, the teacher specifies a restricted set of typically simpler configurations, and expands the range of variability as the system masters them.

Training in the cloud

Reinforcement learning works by exploring the state and action space, and seeing the effects of different actions in different states. This usually requires simulating hundreds of thousands or millions of actions. To get results quickly, you will need to run many simulators in parallel. Azure provides the needed infrastructure, and our tools make it easy.

Deploying a trained Brain

Integration types

Once trained, AI systems can be deployed to serve as a decision support tool for operator, to directly control the equipment, and in various hybrid configurations. Here are some advantages and disadvantages of each method:

Method	Description	Advantages	Disadvantages
Decision Support	Deploy an AI system in the operator control room or control station to provide recommendations	Human in the loop enables rapid deployment and time to value while keeping existing operators and procedures in place	Optimal recommendations may not be consistently implemented

Direct Control	Deploy and AI system to directly control the manufacturing line	Consistent implementation of recommendations even at high decision frequencies	Longer time to value due to integration with production systems
Hybrid – Sequential	Deploy an AI system to directly control the manufacturing line only after verifying the AI with a human in the loop for a period	Provides the peace of mind and rapid time to value of decision support with the consistent high frequency implementation of direct control	Duplicated work and additional cost of creating a decision support UI and an integration with production systems
Hybrid – Safety Override	Deploy an AI system to directly control the manufacturing line with a manual override option that gives humans system control	Consistent high frequency implementation of direct control with added peace of mind	Additional work and cost to create manual override system
Hybrid – Conditional	Deploy an AI system for direct control under certain conditions and decision support or manual operator control under other conditions	Get the benefits of AI-based control before all possible operating modes have been trained and validated	Requires operators to always be on standby. Lack of practice in routine situations can lead to operator skill decline

Deployment platform

There trained AI can be deployed in the cloud, on an edge device connected to your production line, or embedded in the equipment itself. Brains deployed in the cloud are the simplest to maintain and upgrade, but rely on network connectivity. This makes cloud-based deployment most appropriate for POCs for decision support or low-frequency control with active human supervision. For production deployment, edge or embedded is usually more appropriate. We support Azure IoT to make deployment and updates seamless, and easy to integrate with existing systems.

Deployment best practices

The first secret to successful deployment of Brains trained in simulation is to validate your simulations across the target range of scenarios.

Whenever possible, deploy early candidate Brains as decision support, to uncover missed elements in the simulation, the machine teaching approach, or the technical integration with the sensors and actuators.

Safety: Plan for failure and ensure there is safety code to protect people and equipment if sensors or machines malfunction.

Next steps

If you are interested in using machine teaching to optimize your continuous manufacturing process, you can learn more at https://www.microsoft.com/en-us/ai/autonomous-systems and you can contact Dave Cahill (dave.cahill@microsoft.com), Principal Program Manager.