University of Zurich[UZH]

# Agent-based Financial Economics
# Lesson 10: Learning

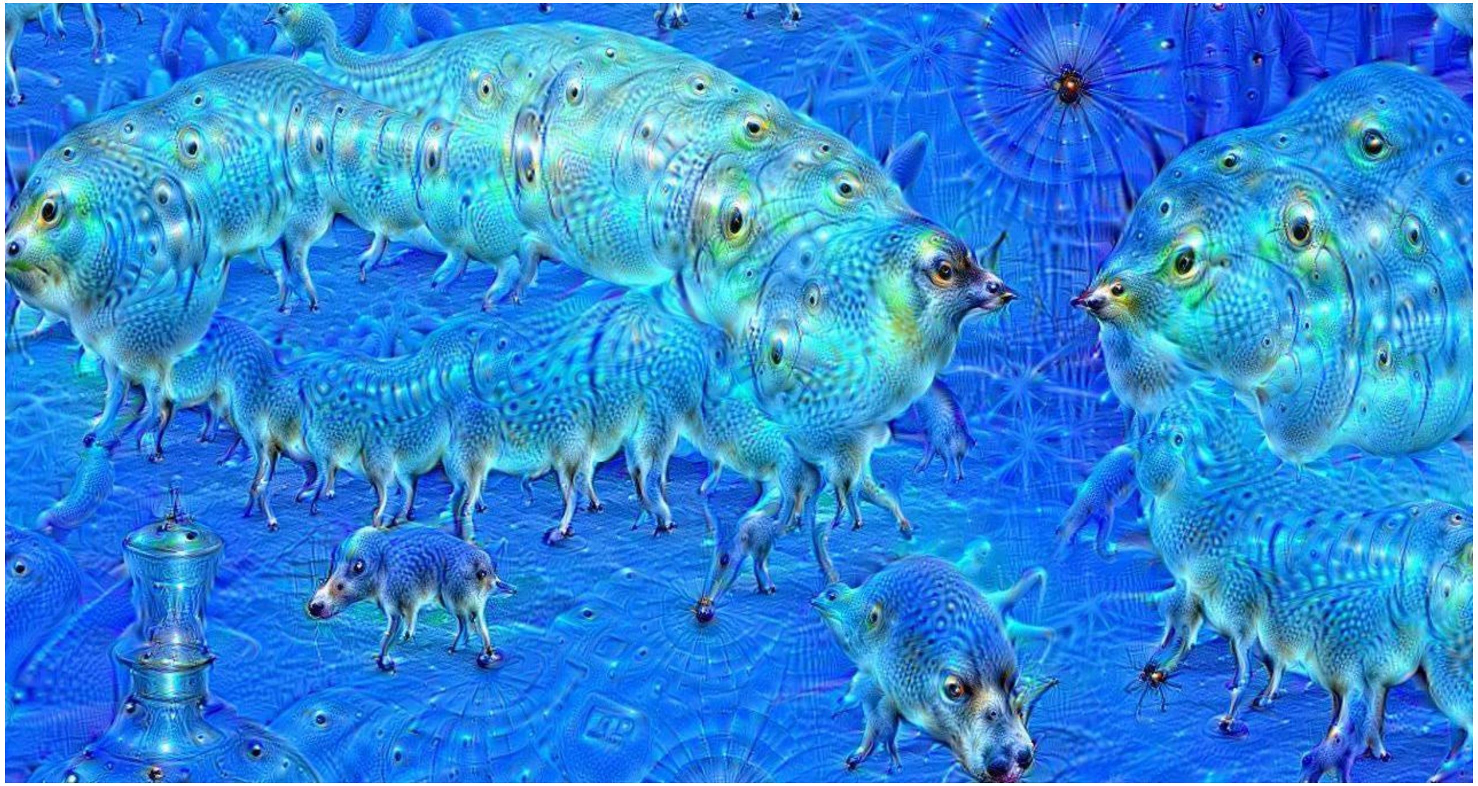
Luzius Meisser, Prof. Thorsten Hens

luzius@meissereconomics.com

“What I cannot create, I do not understand.”

- Richard Feynman

# Today



- Santa Fe Artificial Stock Market
- Learning
- Model predictive control
- Open agent discussion

# Course Outlook

November 30$^{th}$: capital (?)

December 7$^{th}$: team 105, team 102, team 104

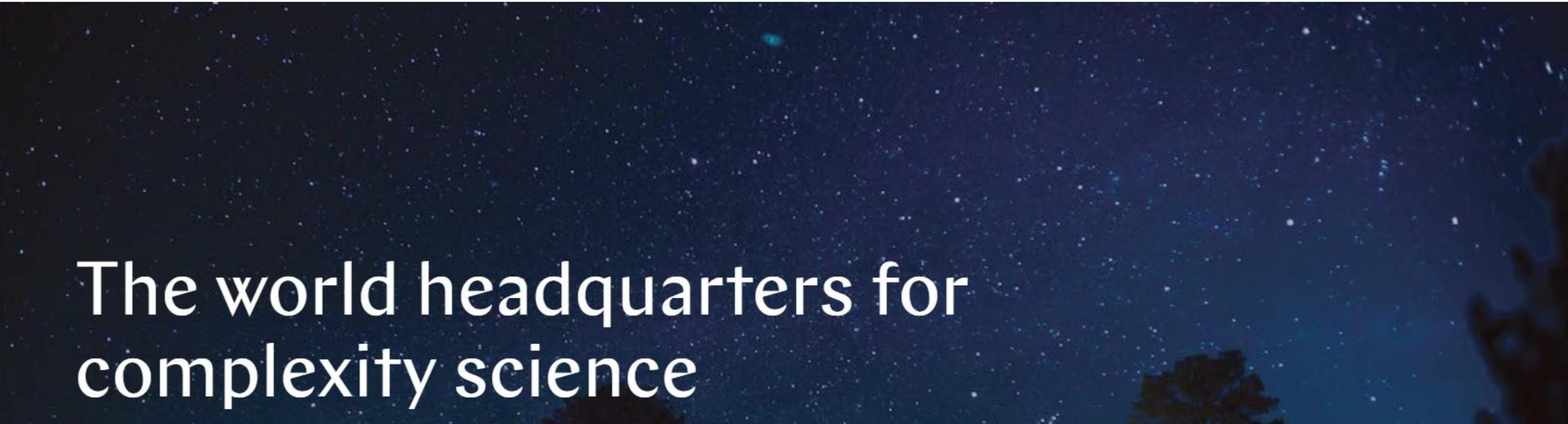
December 14$^{th}$: team 100, team 101, team 103

December 22$^{nd:}$: open topic

# Santa Fe Artificial Stock Market

SANTA FE INSTITUTE

- Small, alternative research institute in New Mexiko
- Known for complexity science and social simulations
- Created the "Santa Fe Artificial Stock Market" in the 90ies, one of the earliest attempts to construct a financial market model with heterogeneously learning traders.



# Santa Fe Artificial Stock Market

Resources:

- Tesfation: www2.econ.iastate.edu/tesfatsi/SFISTOCKDetailed.LT.htm
- LeBaron: Building the Santa Fe Artificial Stock Market, 2002
- Brian Arthur: Complexity Economics: A Different Framework for Economic Thought, 2013

Blake LeBaron

Brian Arthur

Leigh Tesfatsion

# Santa Fe ASM: Assets

Two assets:

- Risk-free asset, paying $r_f = 0.10$
- Risky stock paying a stochastic dividend: $d_t = \bar{d} + \rho(d_{t-1} - \bar{d}) + \mu_t$

with $\bar{d} = 10$, and $\rho = 0.95$, and $\mu_t \sim N(0, \sigma_\mu^2)$

→ Dividend mean-reverts to 10.

# Santa Fe ASM: Market Clearing

announced by a market maker to all the traders. Agents found a matching rule for the current market conditions, and came to the market with an order to buy or sell 1 share of stock. Most of the time this market was out of equilibrium with either more buyers or sellers. The smaller of these two sets would get satisfied while the other would get rationed. For example if there were 100 buyers $(B_t)$ and 50 sellers $(S_t)$, the sellers would all be able to sell 1 share, and the buyers would get only 1/2 share. The price would then be adjusted to reflect either the excess demand or supply.

$$\boxed{p_{t+1} = p_t + \lambda(B_t - S_t)} \tag{1}$$

I have commented previously on some of the problems with this pricing mechanism[4], but it is important to repeat them here. The first problem is that the results were very sensitive to $\lambda$. Setting $\lambda$ too low generated long persistent trends during which the market stayed in excess demand or supply. Setting $\lambda$ too high yielded markets in which the price overreacted, and the market thrashed back and forth from excess supply to excess demand.[5] The second problem

Price function in the Santa Fe Artificial Stock Market.
Source: “Building the Santa Fe Artificial Stock Market”, 2002, Blake LeBaron

# Demand and Supply

- Demand for stock depends on price prediction
- Each agent makes individual price prediction using prediction rules
- Each agents has 100 prediction rules, but only uses the recently best one.
- Rule-set is evolved through genetic learning
  - Mutation: randomly change rules sometimes a little
  - Selection: delete the worst rules every now and then
  - Reproduction: recombine some of the best rules to replace the deleted ones
- Rules are linear: $\hat{E}_t^i(p_{t+1} + d_{t+1}) = a_j(p_t + d_t) + b_j$
- But only applied if they match current market conditions, see next slide.

# Rule matching

$$(\#, 0, 1, \#; a_j, b_j, \sigma_j^2).$$

Agent always applies the best active rule.
(Unclear from the paper how inactive rules are ranked…)

| Rule | Matches |
|---|---|
| $(1, \#, 1, \#, 0)$ | $(1, 0, 1, 0, 0)$ |
| | $(1, 1, 1, 0, 0)$ |
| | $(1, 0, 1, 1, 0)$ |
| | $(1, 1, 1, 1, 0)$ |

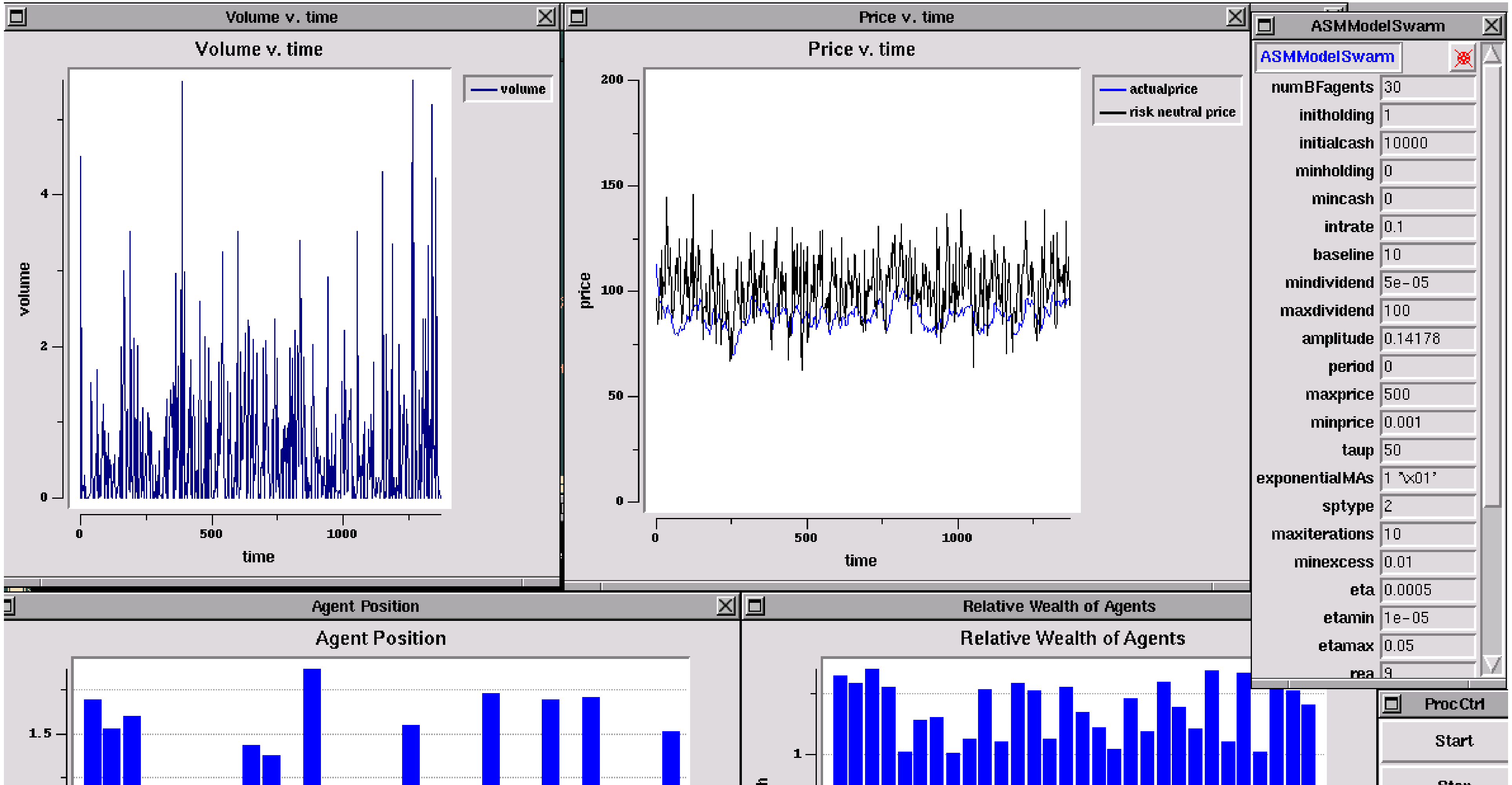
Table 1: Matching Examples

| Bit | Condition |
|---|---|
| 1 | Price*interest/dividend > 1/4 |
| 2 | Price*interest/dividend > 1/2 |
| 3 | Price*interest/dividend > 3/4 |
| 4 | Price*interest/dividend > 7/8 |
| 5 | Price*interest/dividend > 1 |
| 6 | Price*interest/dividend > 9/8 |
| 7 | Price > 5-period MA |
| 8 | Price > 10-period MA |
| 9 | Price > 100-period MA |
| 10 | Price > 500-period MA |
| 11 | On: 1 |
| 12 | Off: 0 |

Table 2: Condition Bits

# Santa Fe Artificial Stock Market

Tried to run it myself, but without success… here's a screenshot from the Internet:

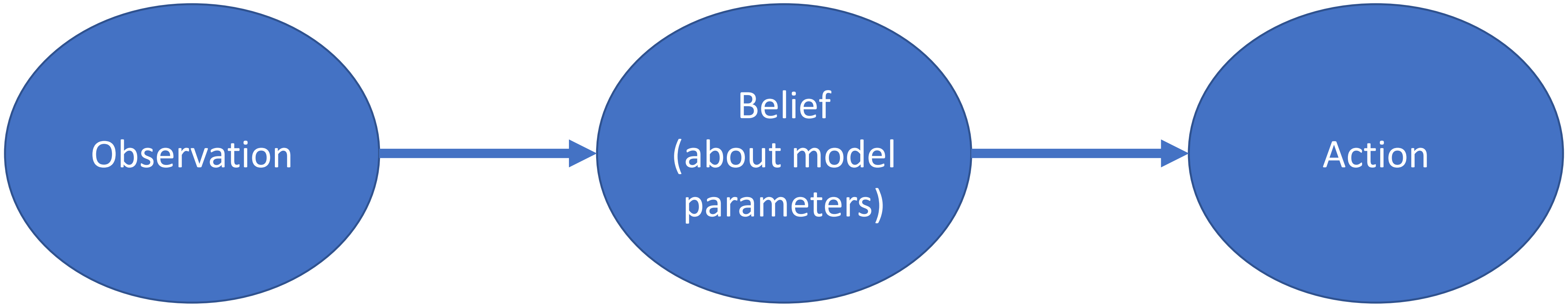


# Learning Method Classification

- Psychology: conscious vs unconscious learning.
  - Conscious learning example: reading a book to learn something
  - Unconscious learning example: learning to ride a bicycle (you can’t do so by reading a book)
- Computer science: online vs offline
  - Online: “learn as you go”, continuously update beliefs as new data comes in
  - Offline: all data from beginning is necessary to update beliefs
- Endogenous vs exogenous learning
  - Endogenous: agents learn within a simulation run, learning is part of the model
  - Exogenous: agents learn between simulation runs, “reincarnating agents”

Many learning methods can be applied in multiple ways.

# Exploration




E.g. was my offer filled?

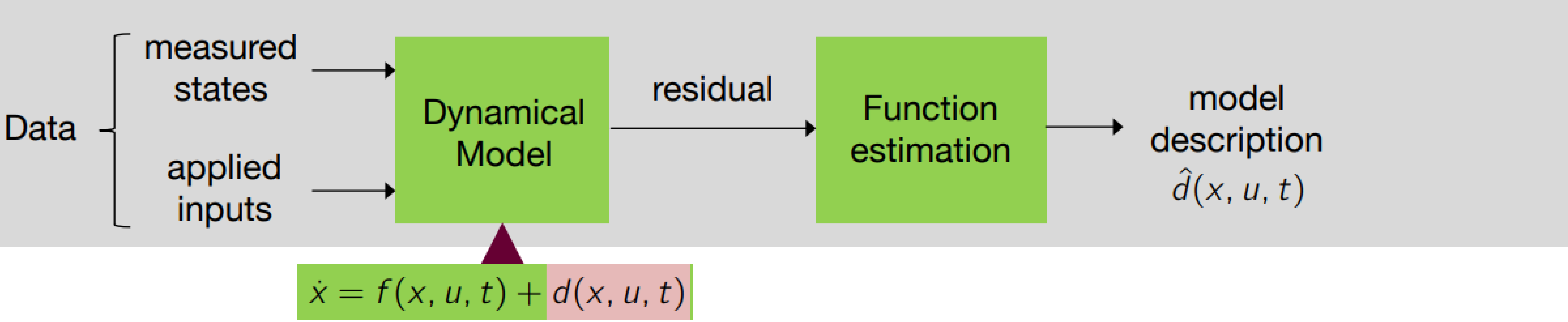
I can charge 7$ per potatoe.

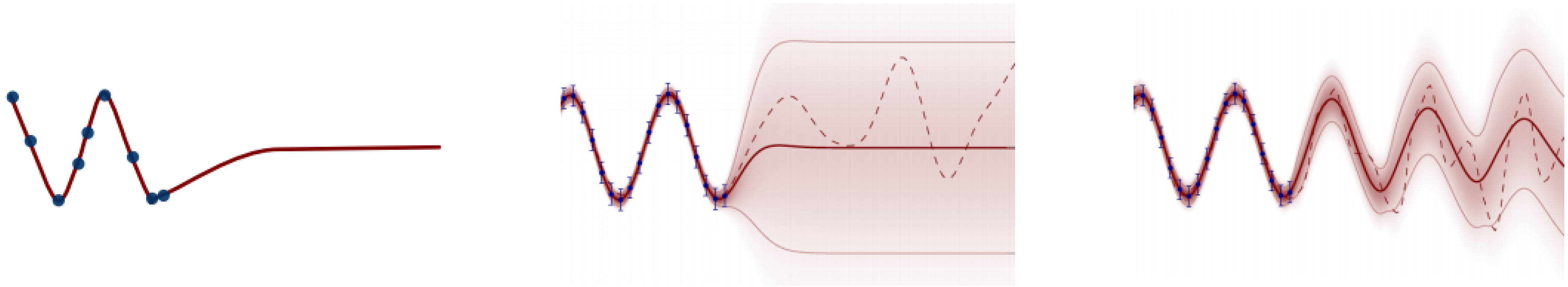
I sell most of my potatoes for 7$.
But also a small amount for 7.10$
as a test.

Actions influence what you can observe, leading a trade-off between expoitation and exploration.

# Automatic Model Improvement from Data

## Online Learning



Slide copied from Melanie Zeilinger:
https://www.ethz.ch/content/dam/ethz/special-interest/dual/riskcenter-dam/Dialogue%20Events/Melanie%20Zeilinger.pdf

# Example

## Autonomous Underwater Vehicle (AUV) – Depth Control

**States**
$$\begin{cases} w\text{, heave velocity [m/s]} \\ q\text{, pitch velocity [rad/s]} \\ \Theta\text{, pitch angle [rad]} \end{cases}$$

**Input:** rudder deflection

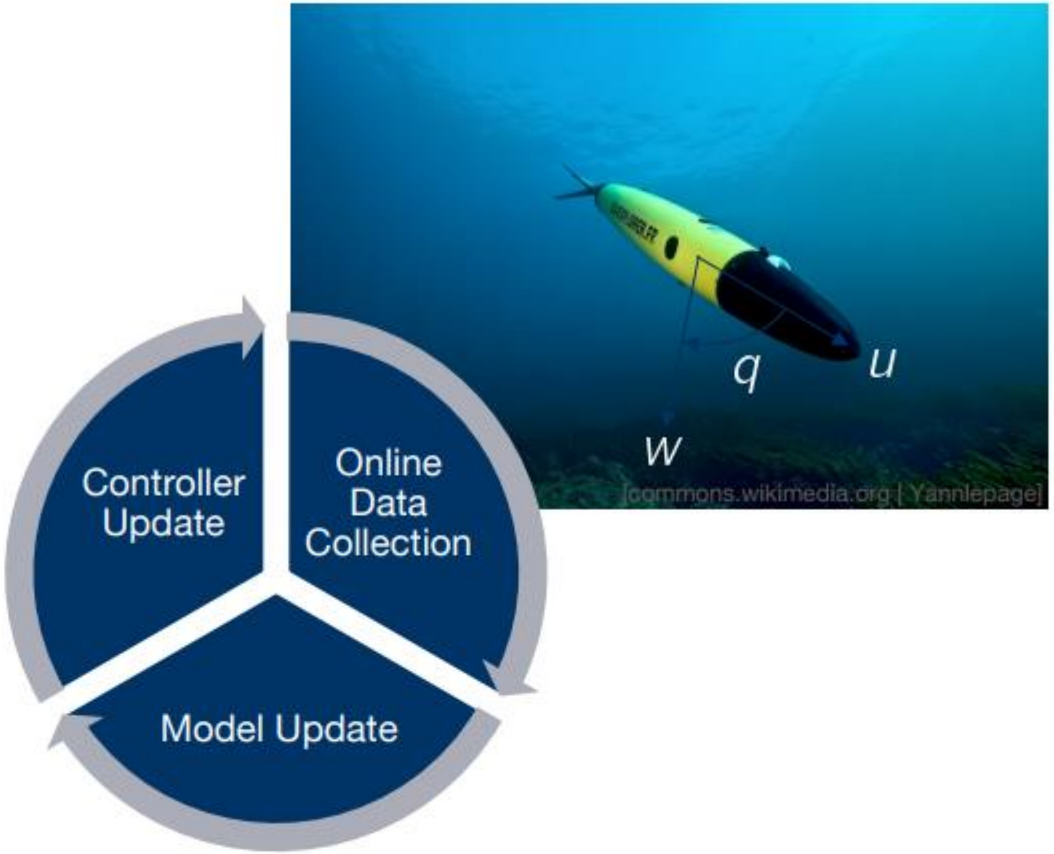
**Constraints:** min & max pitch velocity / angle
min & max rudder deflection

**Objective:** track set point point for pitch angle

**Model:**

$$x_{k+1} = Ax_k + Bu_k + B_d d(w_k, q_k)$$

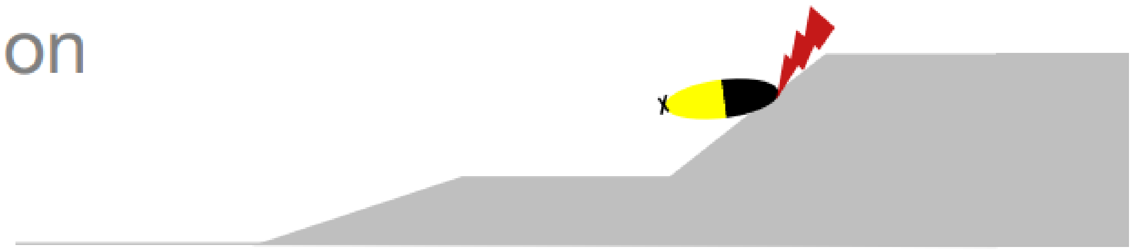
nonlinear resistance unknown

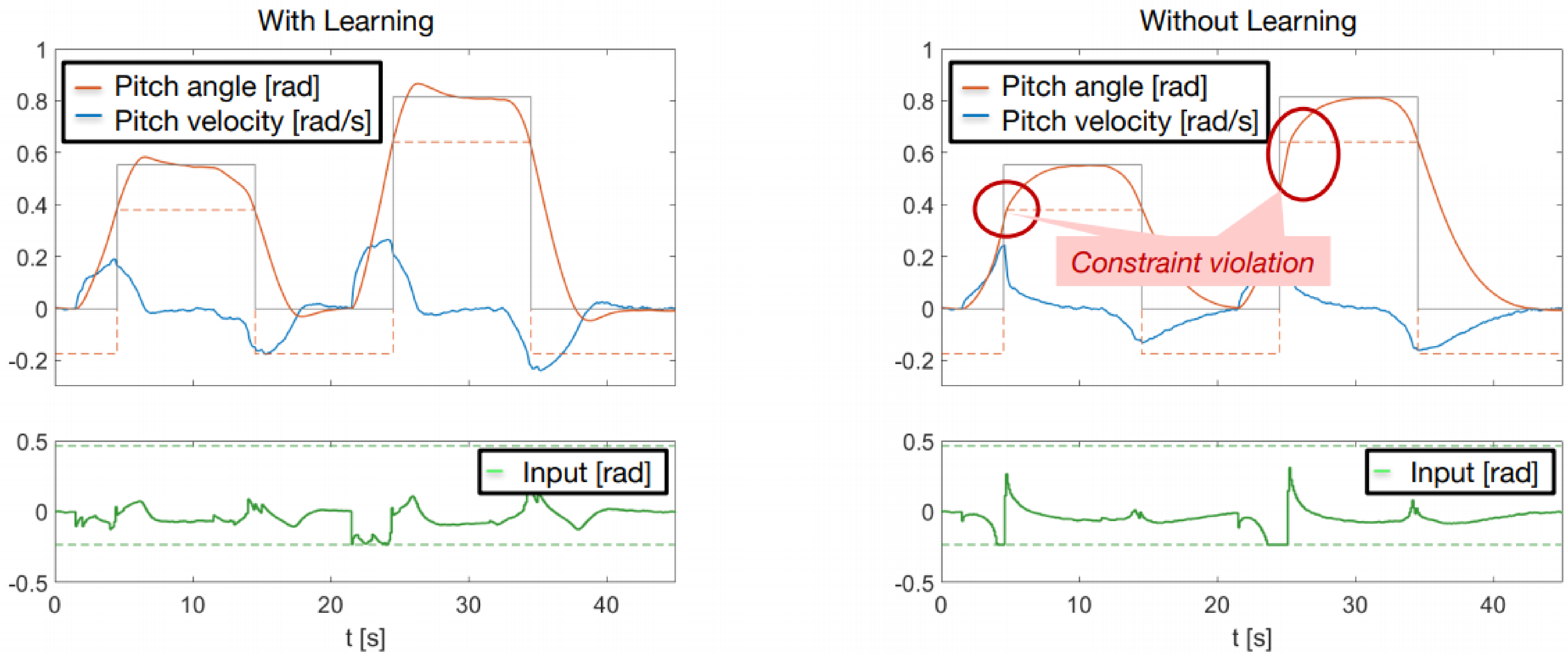



Slide copied from Melanie Zeilinger:
https://www.ethz.ch/content/dam/ethz/special-interest/dual/riskcenter-dam/Dialogue%20Events/Melanie%20Zeilinger.pdf

# Autonomous Underwater Vehicle (AUV) – Depth Control

## No Learning Causes Constraint Violation






Slide copied from Melanie Zeilinger:
https://www.ethz.ch/content/dam/ethz/special-interest/dual/riskcenter-dam/Dialogue%20Events/Melanie%20Zeilinger.pdf
Course: http://www.idsc.ethz.ch/education/lectures/model-predictive-control.html

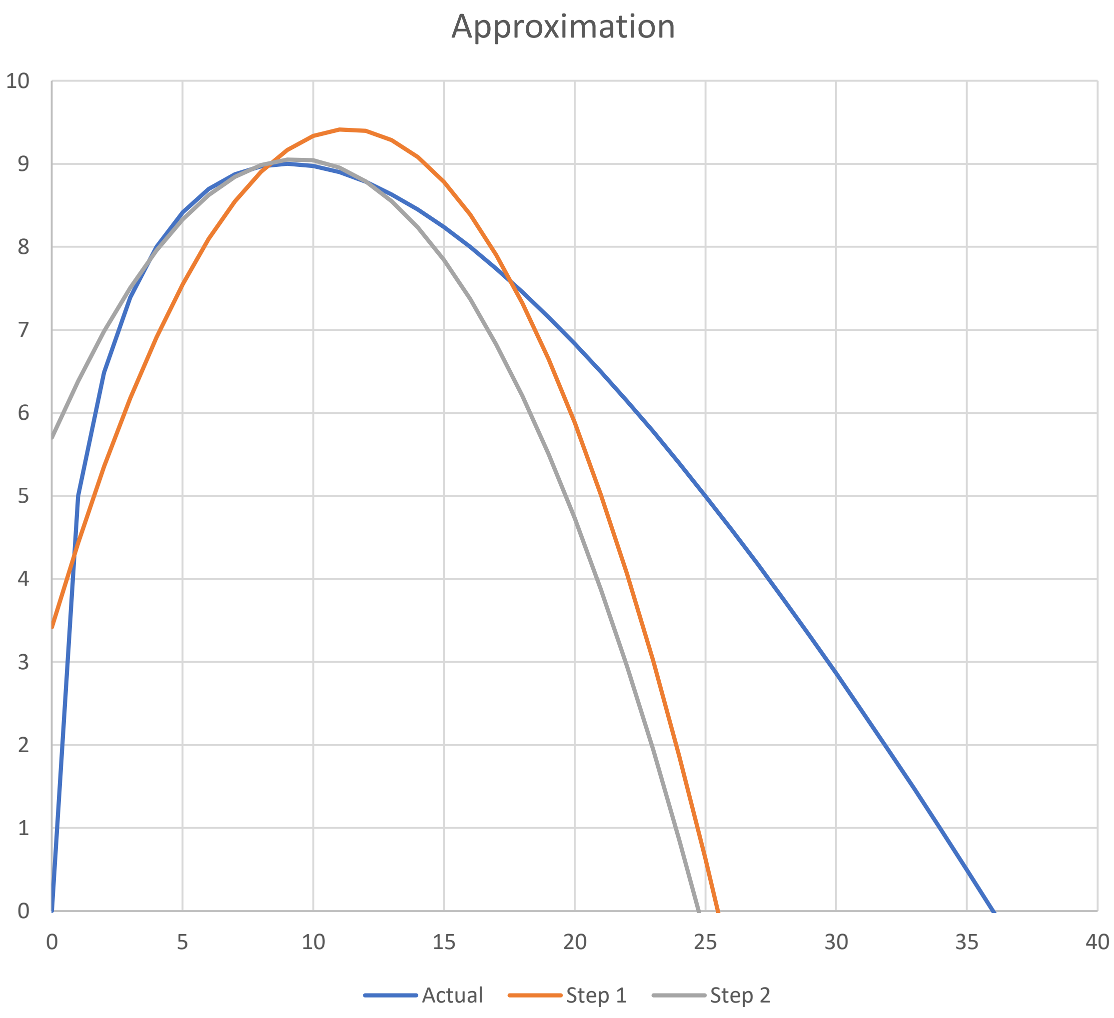
# Online regression example

Firm assumes profits are a 2nd-degree polynomial function of spendings x:

$$\pi = a_0 + a_1 x + a_2 x^2$$

1. Observe spendings and profits
2. Update beliefs a0, a1, a2
3. Action: spending optimally according to beliefs

Implemented in classes QuadraticMaximizer and RecursiveLeastSquares.



# Recursive least squares: intuition

- Exponantially moving average: $$E[x] \approx m_{1,t+1} = 0.9 m_{1,t} + 0.1 x_t$$
- Same for second moment: $$E[x^2] \approx m_{2,t+1} = 0.9 m_{2,t} + 0.1 x_t^2$$
- Combine to get an exponentially moved estimator for the variance: $$Var(X) = E[x^2] - E[X]^2 \approx m_2 - m_1^2$$
- We can do the same trick to get the covariance of two variables:

$$E[xy] \approx m_{xy,t+1} = 0.9 m_{xy,t} + 0.1 x_t y_t$$
$$Cov(X,Y) \; = \; E[xy] - E[X]E[Y] \; \approx \; m_{xy} - m_x m_y$$

Finally: the result of a regression is nothing but a combination of variances and covariances. If we have them, we have an “exponentially moving regression” or “recursive least squares with forgetting”.

(Disclaimer: this slide is just to provide a good intuition. Some these estimators are biased.)

# Recursive least squares

### 4.1 Recursive Least Square Estimation with Forgetting

If the values of the parameters of a system change abruptly, periodic resetting of the estimation scheme can potentially capture the new values of the parameters. However if the parameters vary continuously but slowly a different heuristic but effective approach is popular. That is the concept of forgetting in which older data is gradually discarded in favor of more recent information. In least square method, forgetting can be viewed as giving less weight to older data and more weight to recent data. The "loss-function" is then defined as follows:

$$V(\hat{\theta}, k) = \frac{1}{2} \sum_{i=1}^{k} \lambda^{k-i} \left( y(i) - \phi^T(i)\hat{\theta}(k) \right)^2 \tag{9}$$

where $\lambda$ is called the forgetting factor and $0 < \lambda \leqslant 1$. It operates as a weight which diminishes for the more remote data. The scheme is known as least-square with exponential forgetting and $\theta$ can be calculated recursively using the same update equation (6) but with $L(k)$ and $P(k)$ derived as follows:

$$L(k) = P(k-1)\phi(k) \left( \lambda + \phi^T(k) P(k-1) \phi(k) \right)^{-1} \tag{10}$$

and

$$P(k) = \left( I - L(k)\phi^T(k) \right) P(k-1) \frac{1}{\lambda}. \tag{11}$$

For your reference:

The RecursiveLeastSquares class is based on this document:

pdfs.semanticscholar.org/80eb/236ec16f66e4ce167b2bb0c9804385b03c7f.pdf

Related: Kalmann filters, see

www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/
for an excellent explanation

<table>
<tr><th></th><th>non-conscious learning</th><th>routine-based learning</th><th>belief learning</th></tr>
<tr><td>psychology-based models</td><td>Bush-Mosteller model, parameterised learning automaton</td><td>satisficing, melioration, imitation, Roth-Erev model VID model</td><td>stochastic belief learning, rule learning</td></tr>
<tr><td>rationality based models</td><td></td><td></td><td>Bayesian learning, least-square learning</td></tr>
<tr><td>adaptive models</td><td></td><td>learning direction theory</td><td></td></tr>
<tr><td rowspan="2">belief learning models</td><td></td><td></td><td>fictitious play</td></tr>
<tr><td colspan="3">EWA model</td></tr>
<tr><td>models from AI and biology</td><td></td><td>evolutionary algorithms, Replicator dynamics, selection-mutation equation</td><td>genetic programming, classifier systems neural networks</td></tr>
</table>

Table 1: Classification according to the source of the learning models and according to the classification developed below.

Classification of learning methoda by Thomas Brenner.

http://web.uvic.ca/~mingkang/econ353/project/Brenner.pdf

# Learning: Endogenous vs exogenous

- Endogenous: agents learn within a simulation run
  - Most complex: neural networks “brain” for every agent
    (E.g. Isabelle Salle in “Modeling expectations in agent-based models — An application to central bank's communication and monetary policy”)

- Exogenous: agents learn between simulation runs
  Simulation runs multiple times, and “Reincarnating agents”

# Reincarnating Agents

- Repeat the whole simulation many times
- Allow agents to remember their observations from previous simulation runs

→Self-confirming equilibrium (hopefully)

In stark contrast to most existing agent-based literature, which focuses on learning *within* a single simulation run.
In equation-based models, moments of aggregate variables are used instead of local observations.

# Reincarnating Agents

Self-confirming equilibria provide a well-defined benchmark and are related to rational expectations equilibria.

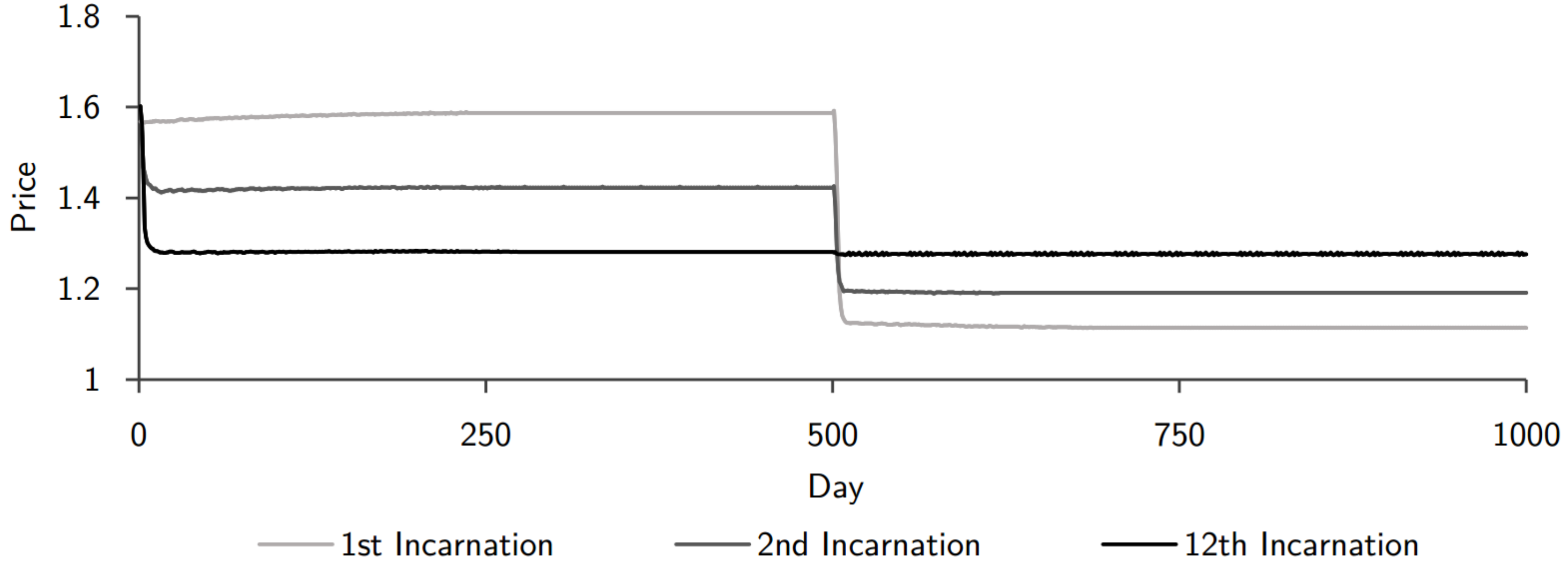
- Every rational expectations equilibrium is a self-confirming equilibrium (Cho and Sargent 2008)
- Self-confirming equilibria are rational expectations equilibria “when applied to competitive or infinitesimal agents” (Cho and Sargent 2008)
- If an RE equilibrium exists and the SC equilibrium is unique, they must be identical. Guess: this is the case in my simulations so far.

# Reincarnating Agents

Consumption smoothing example #Smoothing:

- From day 500 on, consumers suddenly like pizza twice as much
- Consumers remember average consumption before and after shock
- Consumers put enough pizza aside to double consumption after shock, assuming daily purchases stay the same across simulation run (which they do only in equilibrium)

→ Rational expectations equilibrium is approached (error: 0.26%)



# Break

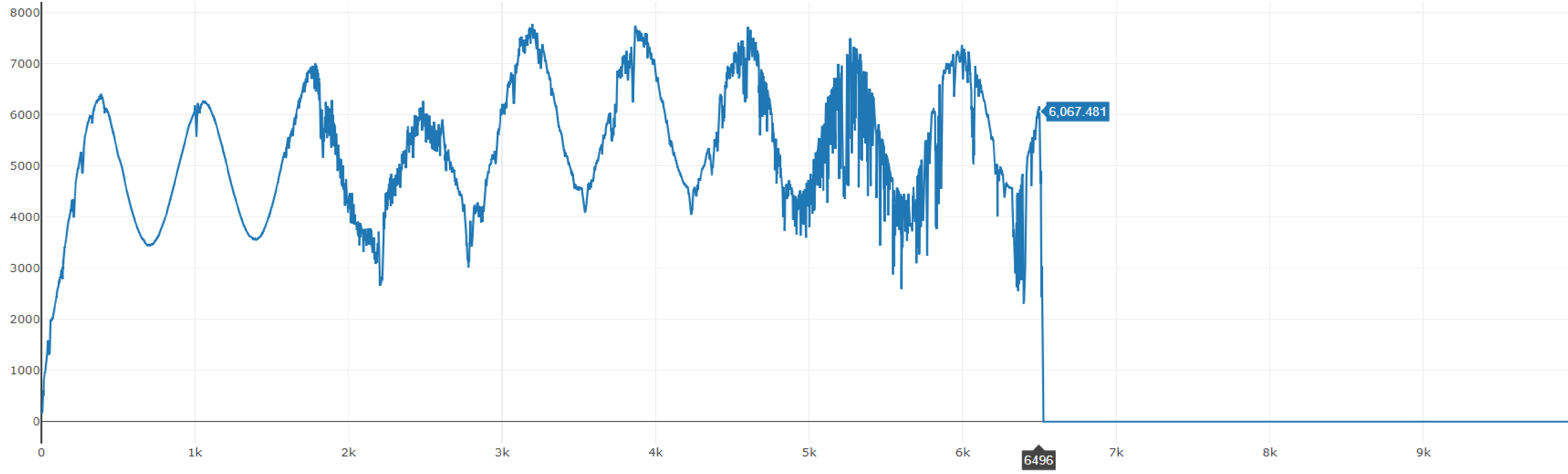
# State of our simulation

Leverage created various unintended consequences…
Here: production breaks down.

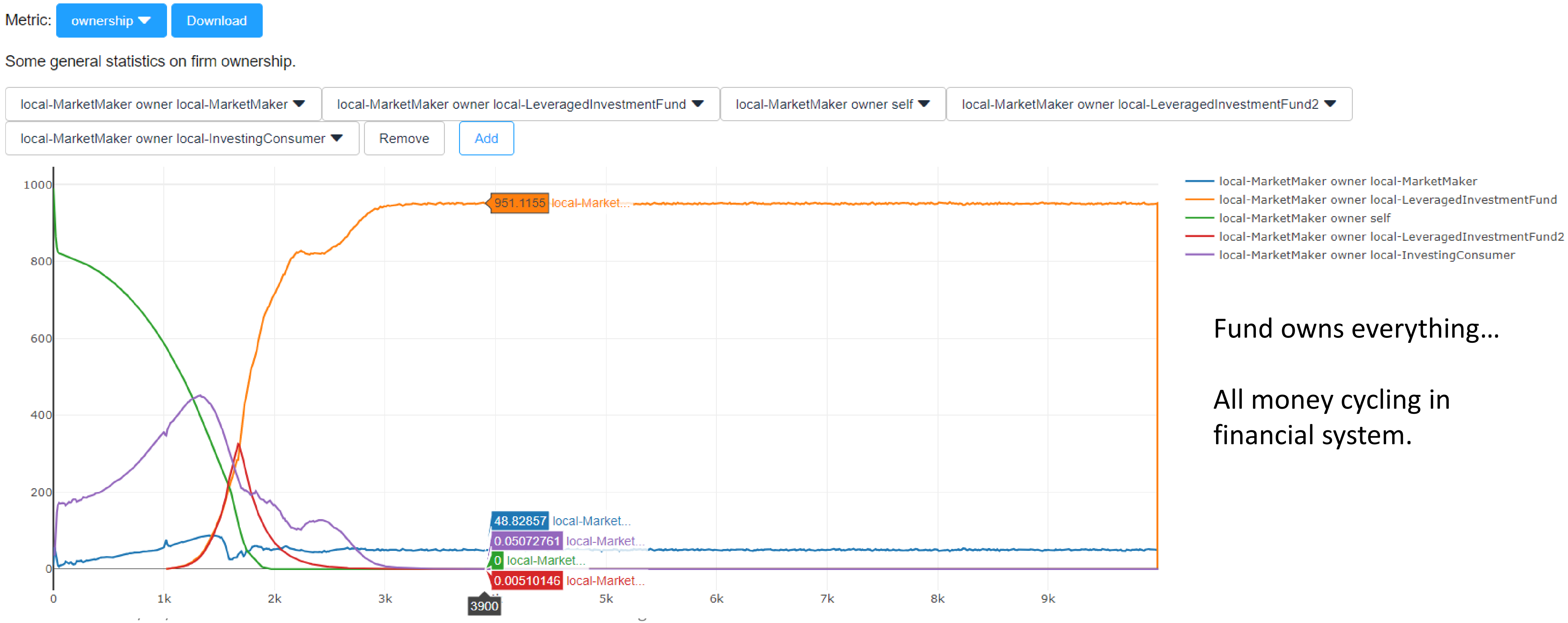
Metric: production Download

Overall production statistics.

Optimal total potatoe output given inputs Add



# State of our simulation




Fund owns everything...

All money cycling in financial system.