University of Zurich[UZH]

Agent-based Financial Economics
Lesson 9: Learning

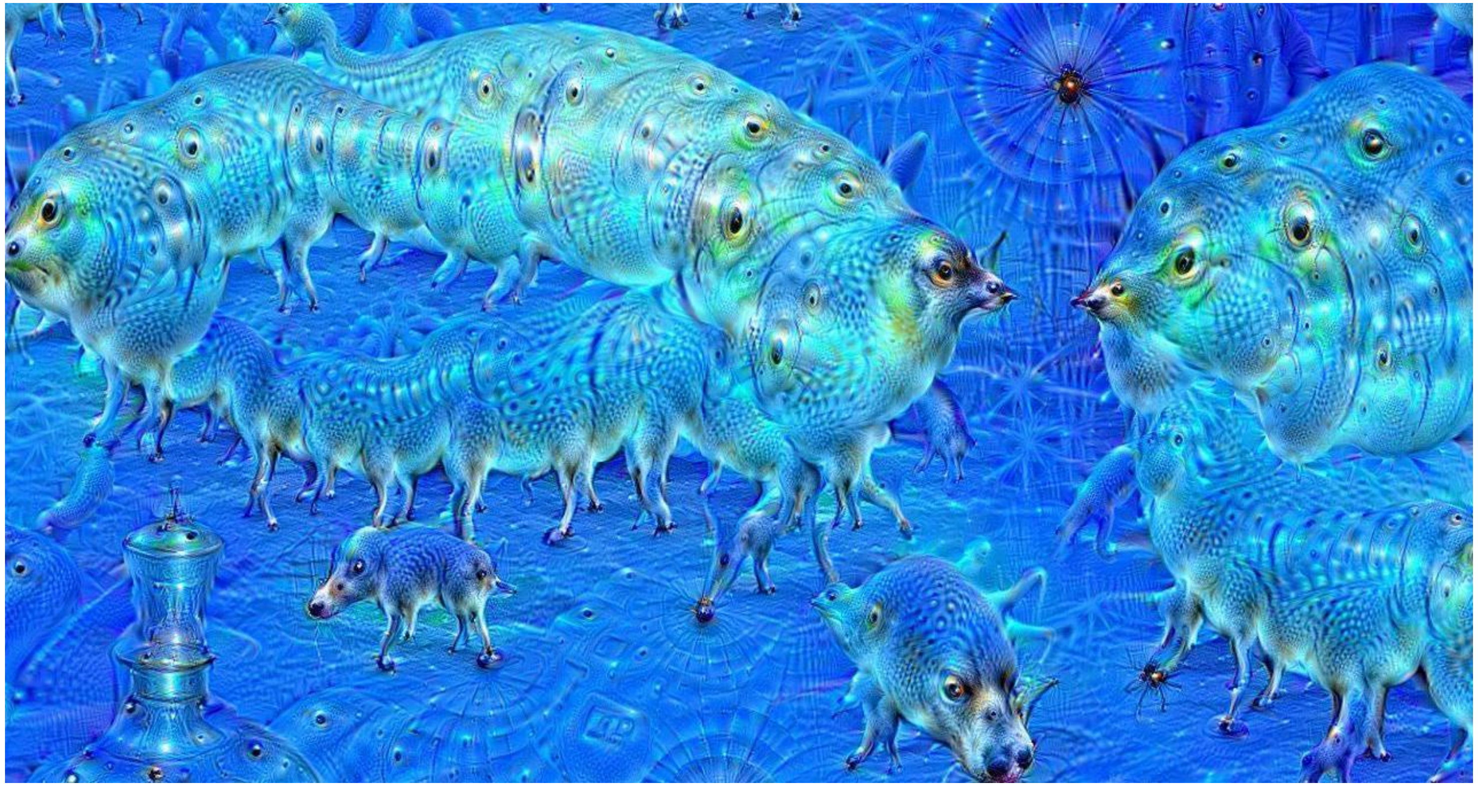
Luzius Meisser, Prof. Thorsten Hens

luzius@meissereconomics.com

"What I cannot create, I do not understand."

- Richard Feynman

Today



- Simulation Update
- MOSES Model
- Learning
- Competition hints

Simulation Update

Made simulation faster. Measures:

- Population cut in half. Only 7 farms (from previously 10).
- Consumers now follow LimitedSampleYieldPickingStrategy(3), i.e. look at three random stocks and pick the one with the highest yield when buying.
- Funds already enter at day 500, simulation shortened to 3000 days.
- No more even distribution of inheritances. First newborn gets everything.
- Eliminated a bug that caused inventory access to be slower than it should be.

→ Now runs below 10 seconds (from previously over 1 minute) on my computer.

Course Outlook

November 30th: competitive simulation camp, presence optional. We will try out different fund strategies together and I can assist with implementing your strategies.

December 6th: team 201, 203, 207

December 13th: teams 202, 205, 208, 210

December 22nd:: open topic

See also https://github.com/meisser/course2019/blob/master/exercises/journal/exercise06-task.md for more information about the presentation.

Presentation template: http://meissereconomics.com/assets/abfe-template.pptx

Stock-Flow Consistent Modelling

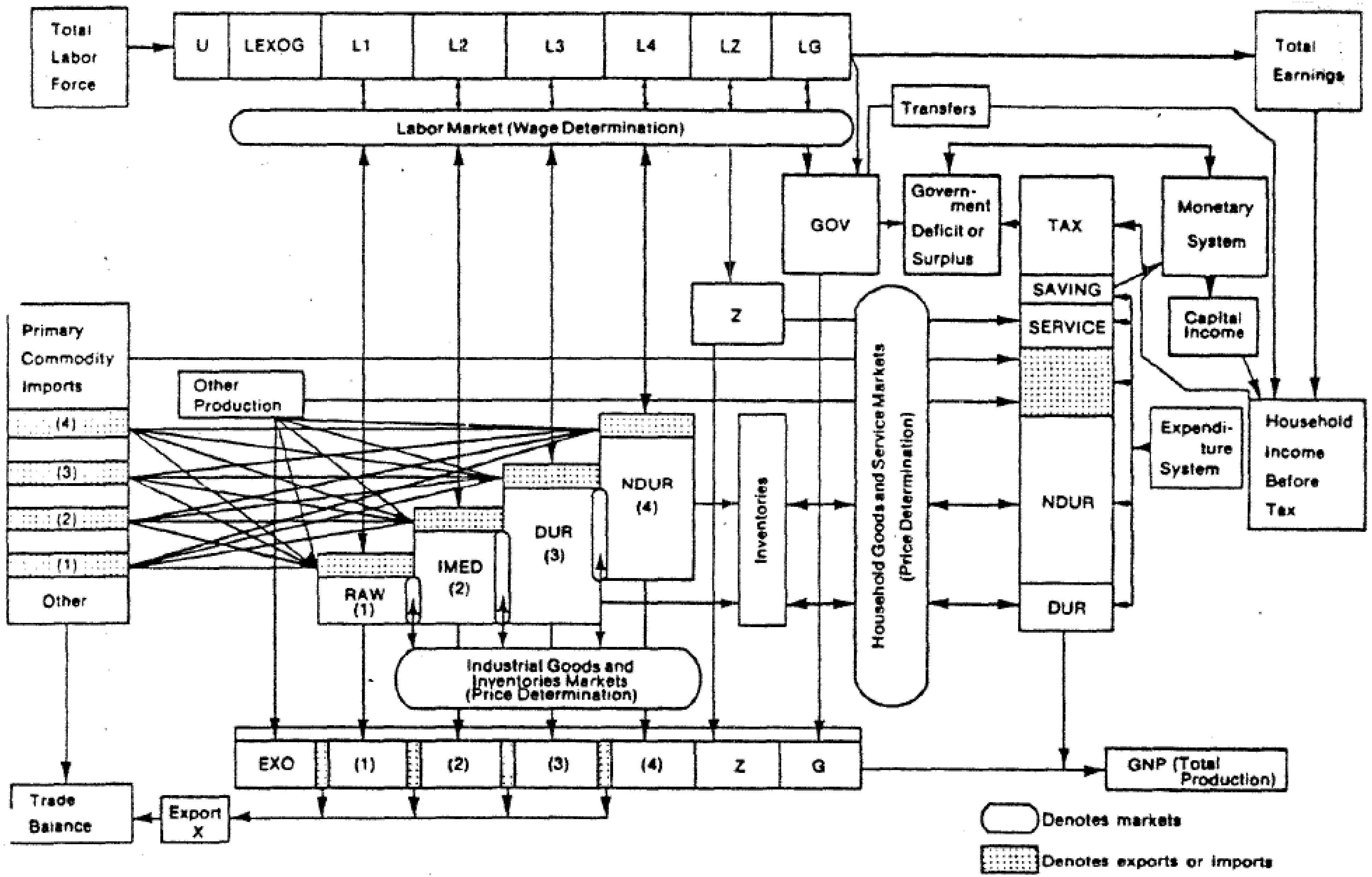
- Stock as in “stockpile” or “inventory”
- Flow: flow of goods and money
- Often, economic models do not keep track of stocks and flows, at least not explicitly
- Stock-flow consistent models explicitly do so
- Our model is stock-flow consistent. Nothing can appear or disappear out of the blue unless we specifically say so.
- Many financial agent-based models are not. For example: the seminal model of Lux and Marchesi from “Scaling and criticality in a stochastic multi-agent model of a financial market” is not, as it contains “noise traders” that never go bankrupt. Same applies to Thurner et al. leverage model from last time.

The MOSES model by Eliasson

- <u>M</u>odel <u>o</u>f <u>S</u>wedish <u>e</u>conomic <u>s</u>tudies
- Built an agent-based model of Sweden in the 80ies
- Consumers modeled in aggregate
- Large firms modeled individually
- Asked managers about how they decide and used these answers to model firm behavior
- Used to advise government on decisions such as “should Sweden build the Gripen?”, “should Sweden bailout its shipping industry?”, etc.

Gunnar Eliasson, author of MOSES model, as described in “THE MOSES MODEL Database and Applications”, 1989

Figure II:1 Macro Delivery and Income Determination Structure of Swedish Model



The MOSES model by Eliasson

We have much better computers now, can model much larger and more complex models with much nicer, automatically generated charts.

Nonetheless, MOSES could produce useful results and provide input to Swedish policy determination.

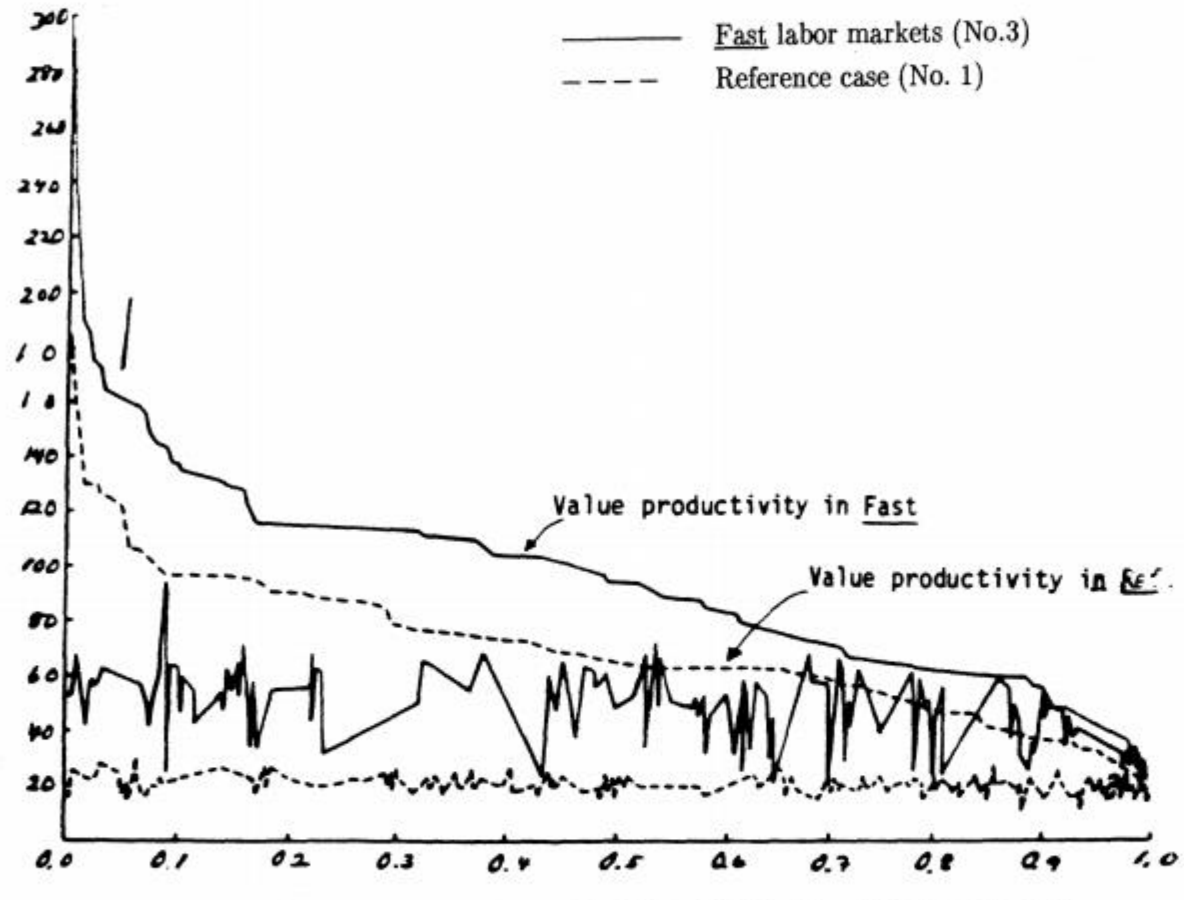


Figure 4 Wage cost and value productivity distributions 1992 in reference case and in fast market experiment

Learning

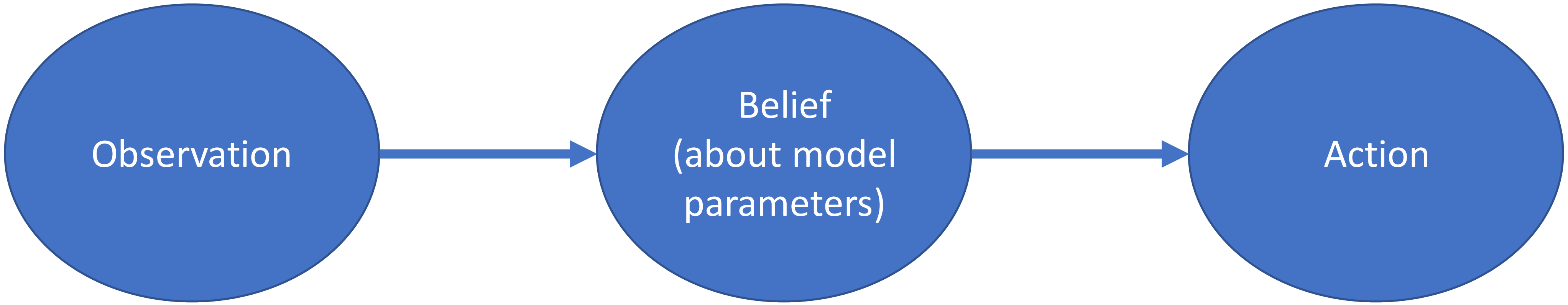
An overview over different models and methods of learning.

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?

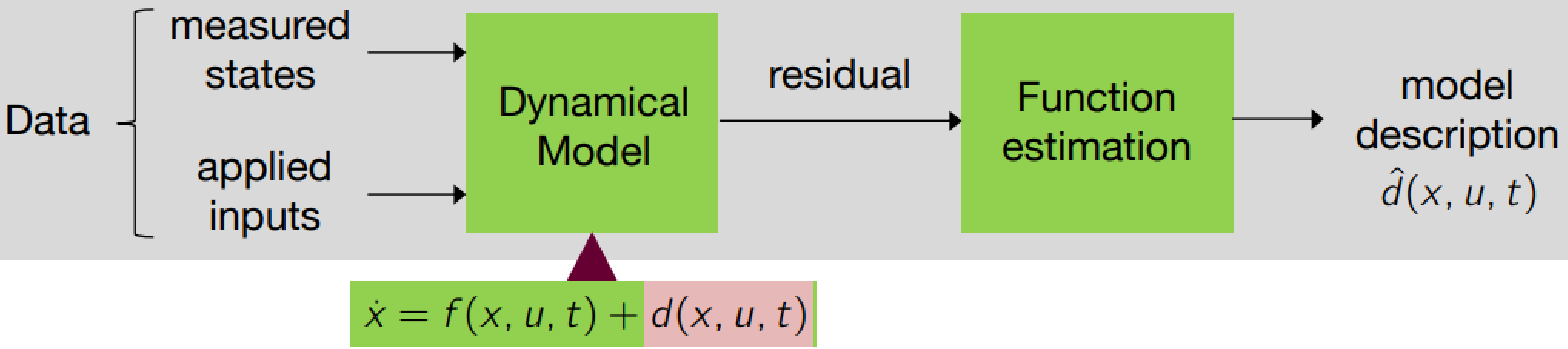
I can charge 7$ per potatoe.

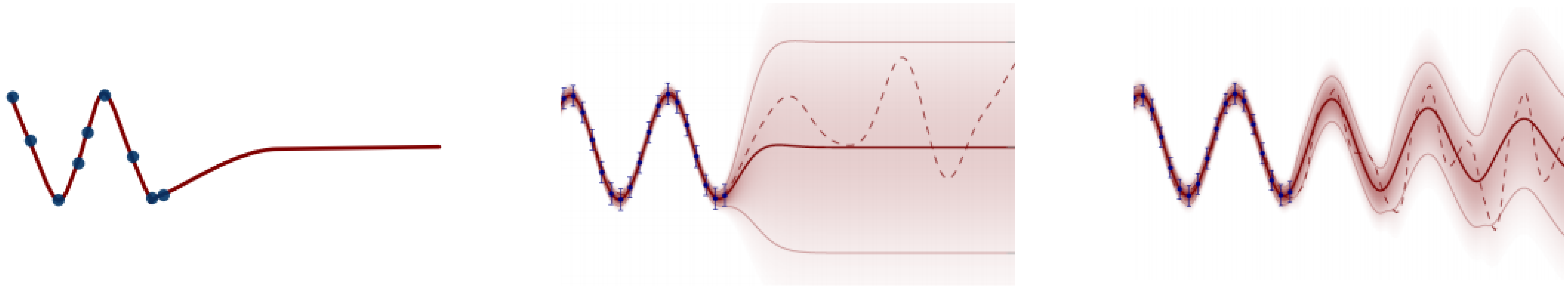
I sell most of my potatoes for 7$. But also a small amount for 7.10$ as a test.

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

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

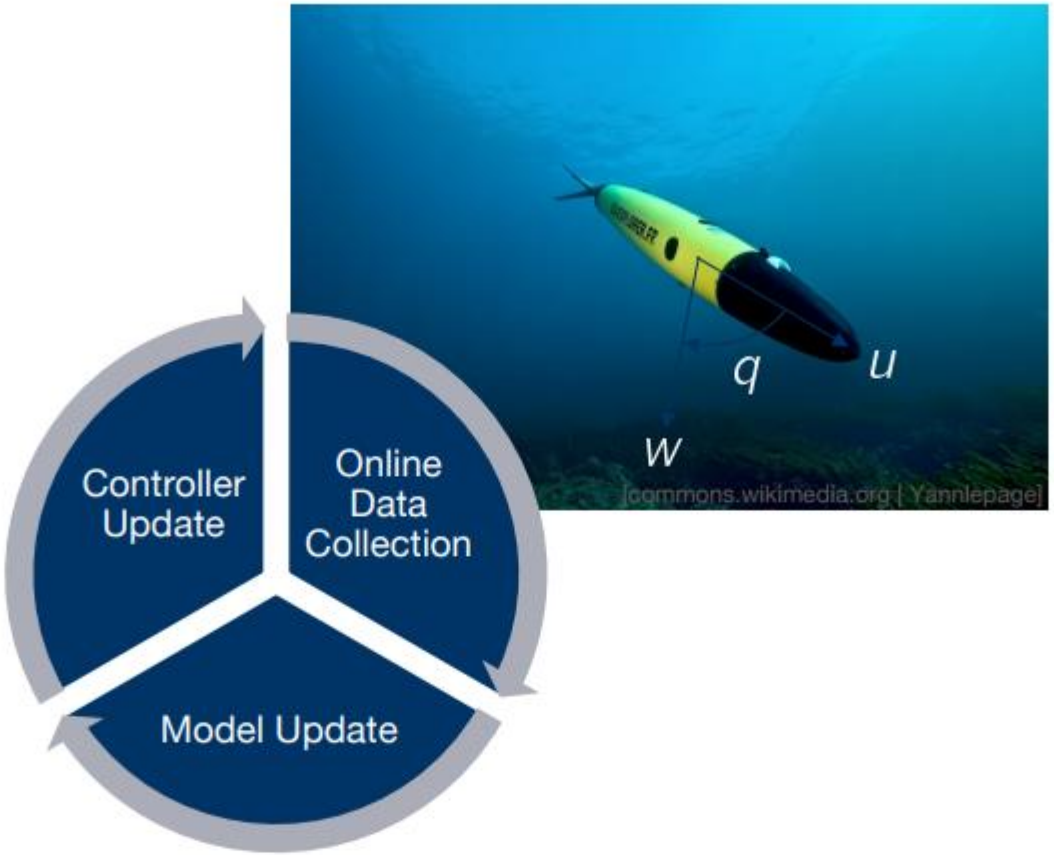
Constraints: min & max pitch velocity / angle
min & max rudder deflection

Objective: track set point point for pitch angle

Model:

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

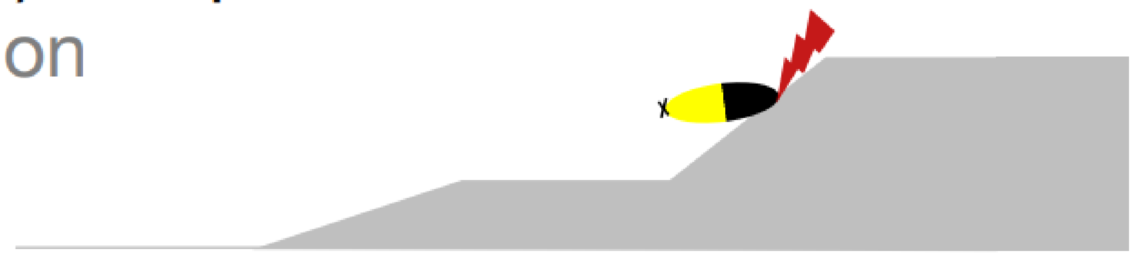
nonlinear resistance unknown

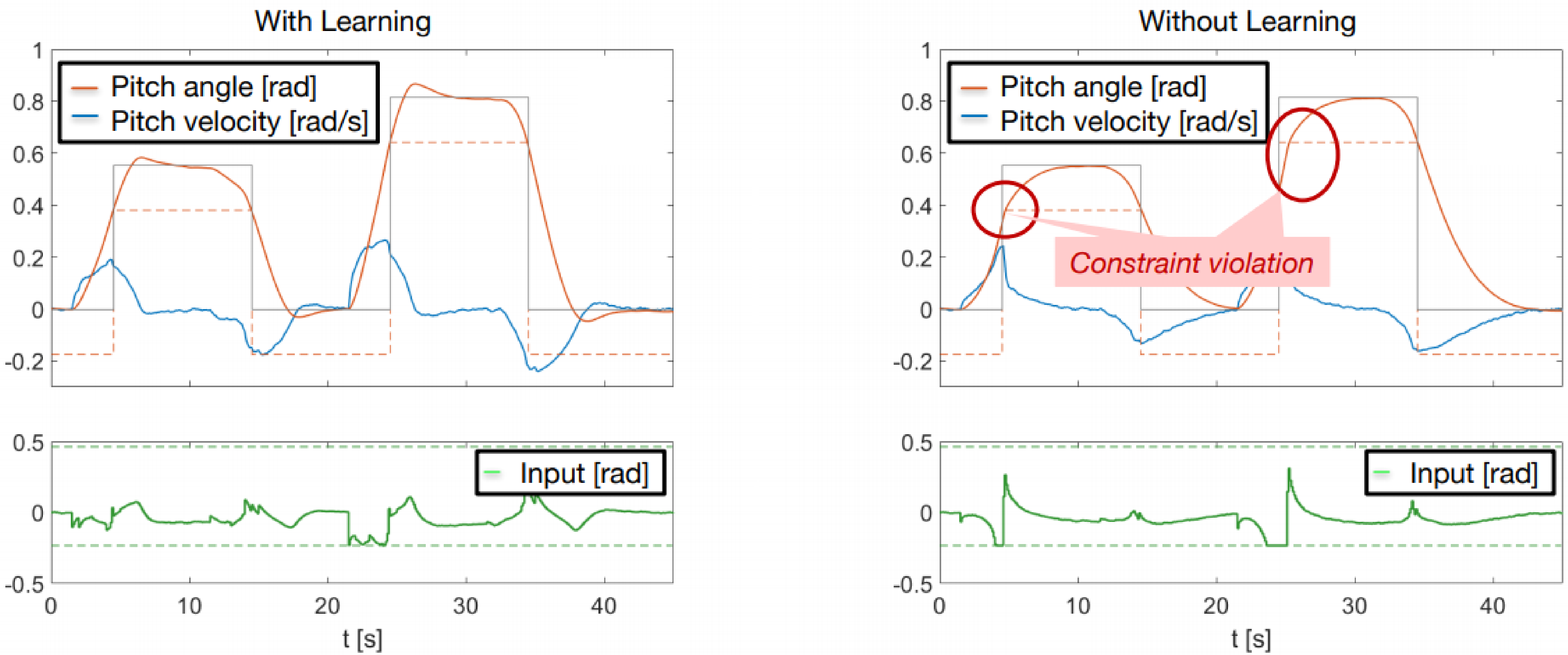


Slide copied from Melanie Zeilinger:
https://www.ethz.ch/content/dam/ethz/special-interest/dual/riskcenter-dam/Dialogue%20Events/Melanie%20Zeilinger.pdf

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

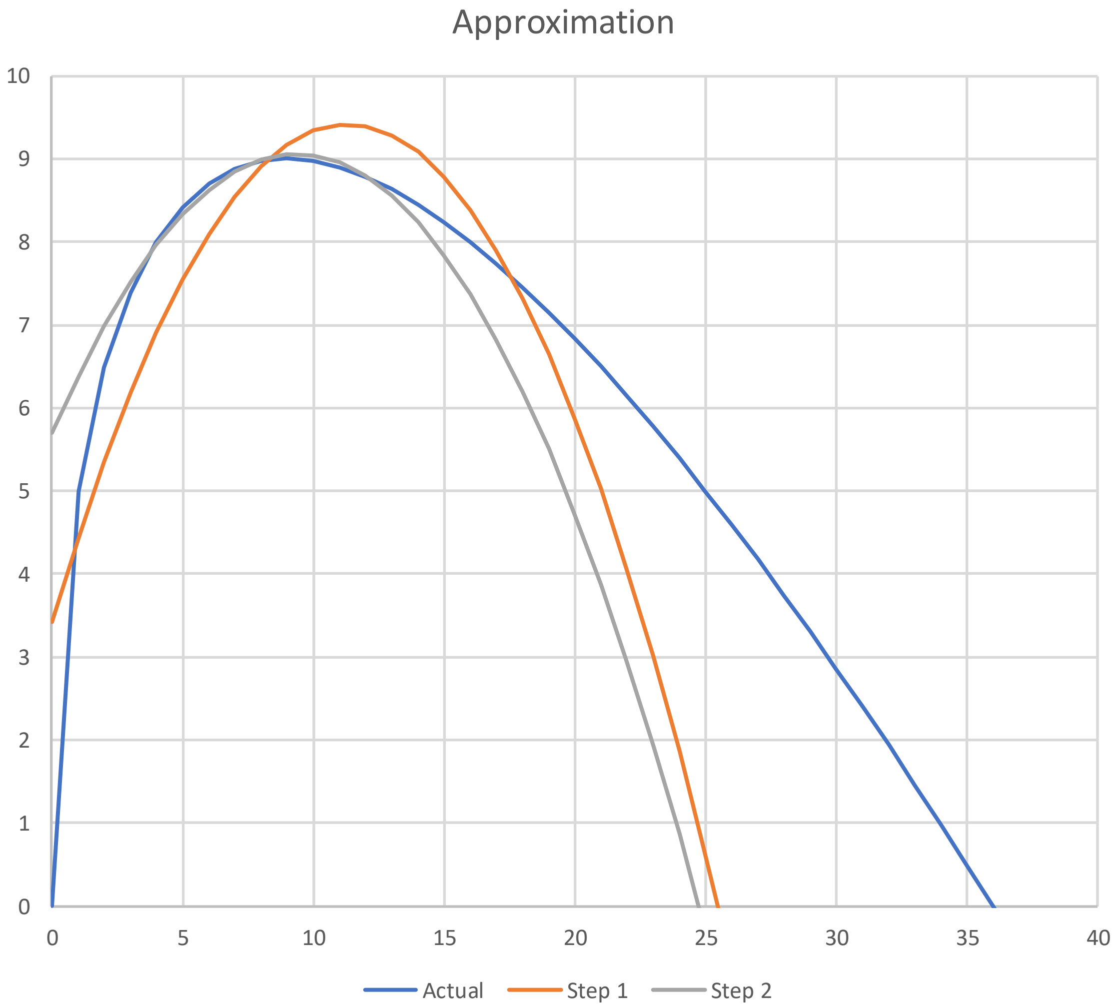
Online regression example

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: $$E[x] \approx m_{1,t+1} = 0.9m_{1,t} + 0.1x_t$$
- Same for second moment: $$E[x^2] \approx m_{2,t+1} = 0.9m_{2,t} + 0.1x_t^2$$
- 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:

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

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 λ 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 θ 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 methods 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.

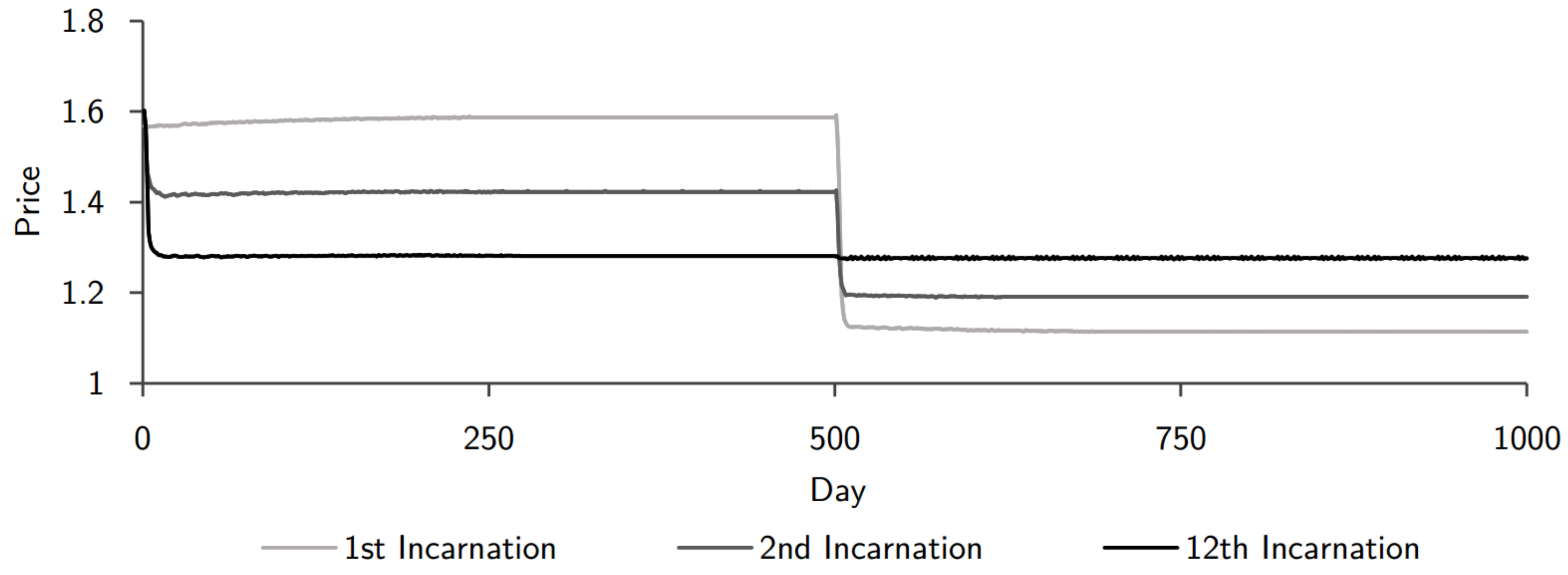
- Every rational expectations equilibrium is a self-confirming equilibrium (Cho and Sargent 2008)
- Self-confirming equilibria are rational expectations equilibria “when applied to competitive or infinitesimal agents” (Cho and Sargent 2008)
- If a RE equilibrium exists and the SC equilibrium is unique, 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%)

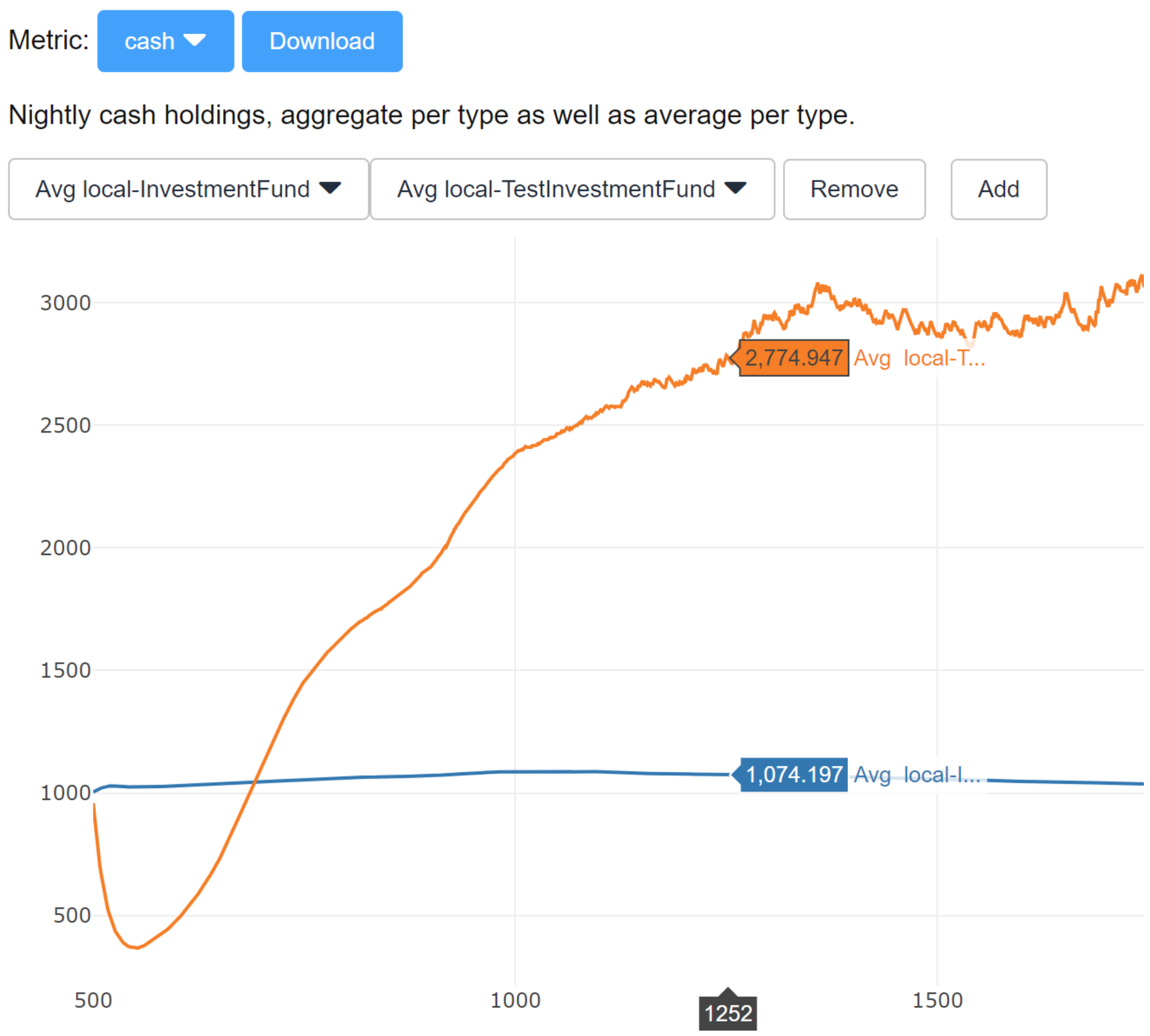


Break

Fund Hints 1

First statistic to consider are the cash holdings. The funds start with 1000.

This is relevant for buying and the dividend rule.



Fund Hints 1

First statistic to consider are the cash holdings. The funds start with 1000.
This is relevant for buying and the dividend rule.

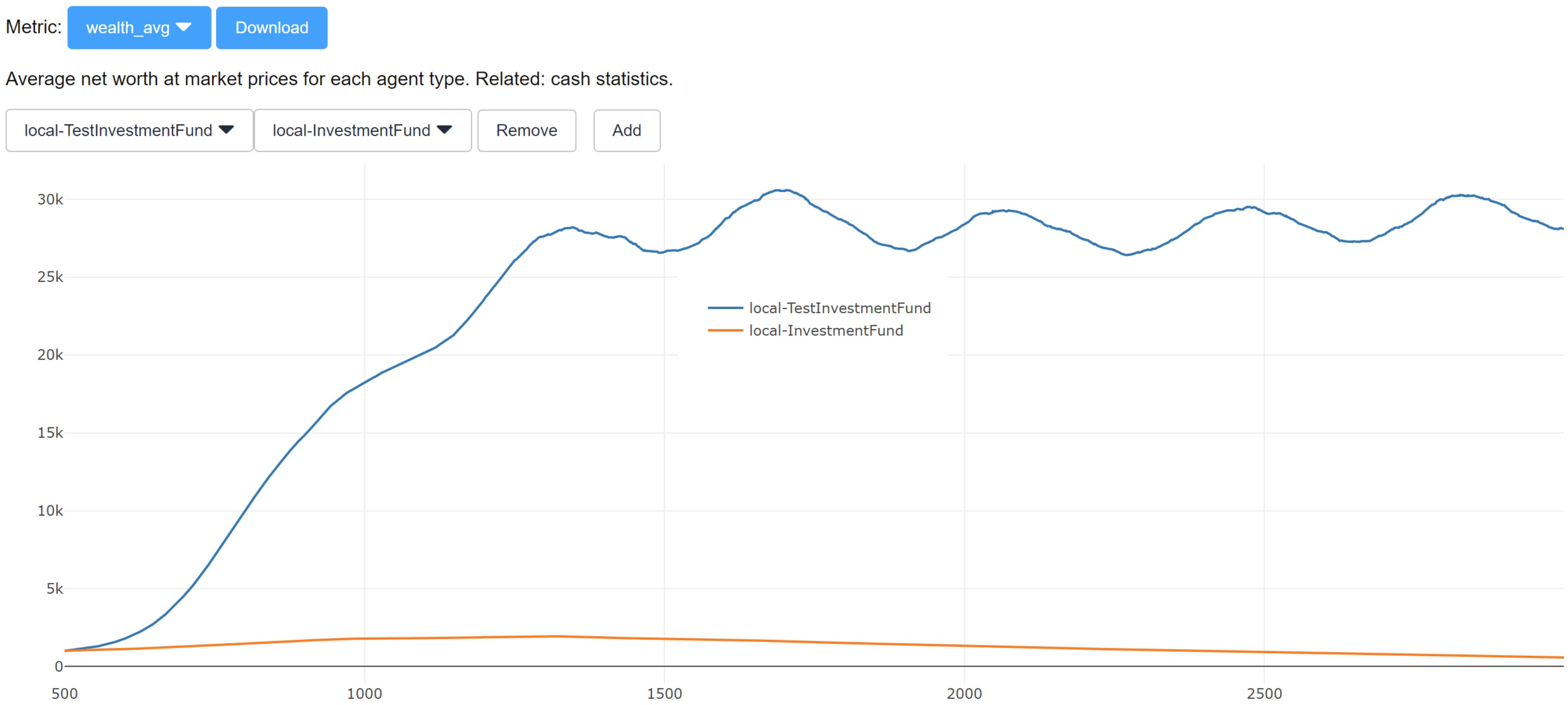
```
public TestInvestmentFund(IAgentIdGenerator world, Endowment end) {
    super(world, end);
    this.reserve = 100;
    this.portfolio = new TradingPortfolio(getMoney(), false);
}

public void managePortfolio(IStockMarket dsm) {
    IStock money = getMoney().hide(reserve);
    portfolio.invest(new HighestYieldPickingStrategy(), dsm, this, money.getAmount() * 0.05);
    portfolio.sell(dsm, this, 0.005);
}

protected double calculateDividends(int day) {
    double cash = getMoney().getAmount();
    if (cash < 1000) {
        return 0.0;
    } else if (cash < 5000) {
        return (cash - 1000) * 0.02;
    } else {
        return 80 + (cash - 5000) * 0.1;
    }
}
```

Fund Hints 2

Make sure the reserve is not too high, so the fund can actually buy things and get rich.
You can verify the value of the funds holding in the wealth statistic.



Fund Hints 3

However, you only can score well if you pay out a dividend, and for paying out a dividend, you must sell some of your holdings sometimes to have the necessary cash.

```
public TestInvestmentFund(IAgentIdGenerator world, Endowment end) {
    super(world, end);
    this.reserve = 100;
    this.portfolio = new TradingPortfolio(getMoney(), false);
}

public void managePortfolio(IStockMarket dsm) {
    IStock money = getMoney().hide(reserve);
    portfolio.invest(new HighestYieldPickingStrategy(), dsm, this, money.getAmount() * 0.05);
    portfolio.sell(dsm, this, 0.005);
}
```

The basic configuration just sells 0.5% of everything every day.
Of course, you can do better. ☺

Additional Fund Hints

Live demo and code discussion.