Distributed Intelligent Systems – W10
Multi-Level Modeling: Complex Examples and Combination with Machine Learning
Outline

• Multi-level modeling framework (recap)

• Complex examples:
  – Collaborative stick pulling
  – Distributed seed assembly

• Combined modeling and machine-learning methods
  – The example of collaborative stick pulling
  – Homogenous and heterogeneous learning
  – Diversity and specialization
Multi-Level Modeling Framework (recap from W9)
Multi-Level Modeling Methodology

**Macroscopic**: representation of the whole swarm (typically a mathematical model)

**Microscopic**: multi-agent models, only relevant robot features captured, 1 agent = 1 robot

**Submicroscopic**: intra-robot (e.g., S&A, transceiver) and environment (e.g., physics) details reproduced faithfully

**Target system** (physical reality): information on controller, S&A, communication, morphology and environmental features
Model Structure and Metrics

- **Exploit controller blueprint** at submicroscopic/physical level as structure for higher level of abstraction (“behavior = state”); use it for both microscopic and macroscopic levels
- **State granularity arbitrary** but (non spatial) performance metrics must be computable explicitly at all modeling levels

E.g., Performance = f(#robots_{search}, #robots_{grip})
Performance = f(#robots_{search}, #robots_{avoidance})
Model Parameters

- **Number of parameters**: decreasing with increasing abstraction
- **Incremental calibration**: a given level can be calibrated on the underlying one using aggregation techniques (e.g., first moment for a distribution at lower abstraction level)
- **Explicit representation**: any system parameter of interest should be captured explicitly at a given level
- **Multiple calibration methods** for model parameters:
  - Ad hoc experiments (e.g., interaction time, sensor transfer functions)
  - System identification techniques (with constrained parameter fitting)
  - Statistical verification techniques (e.g., trajectory analysis)
- **Submicroscopic models**: large parameter space (e.g., individual sensor and actuator features).
- **Micro- and macroscopic models**, essentially two parameter types:
  - State durations
  - State transition probabilities
Nonlinear Example – Collaborative Stick Pulling
The Stick-Pulling Case Study

Physical Set-Up

- 2-6 robots
- 4 sticks
- 40 cm radius arena

Collaboration via indirect communication

- IR reflective band
- Proximity sensors
- Arm elevation sensor
Systematic Experiments

Real robots

• [Martinoli and Mondada, ISER, 1995]
• [Ijspeert et al., AR, 2001]

Submicroscopic model
Results of Experiments and Submicroscopic Modeling

- Real robots (3 runs) and submicroscopic model (10 runs)
- System bifurcation as a function of \#robots/\#sticks

\[
N_{\text{robots}} > N_{\text{sticks}} \quad \text{for } N_{\text{robots}} > N_{\text{sticks}}
\]

\[
N_{\text{robots}} \leq N_{\text{sticks}}
\]
State Transition Probabilities

\[ A_a = \text{surface of the whole arena} \]

\[ p_s = \frac{A_s}{A_a} \]
\[ p_r = \frac{A_r}{A_a} \]
\[ p_R = p_r(N_0 - 1) \]
\[ p_w = \frac{A_w}{A_a} \]
\[ p_{g1} = p_s \]
\[ p_{g2} = R_g p_s \]

**Note:** slightly different space-to-time conversion as that explained in W9 [Correll and Martinoli, ISER 2004]!
Experimental Validation of Spatiality Assumptions

Symmetry of Stick Distribution

Default

# sticks 12
From Reality to Abstraction

Deterministic robot’s flowchart
Nonspatiality & microscopic characterization

PFSM Probabilistic agent’s flowchart
**Full Macroscopic Model**

For instance, for the average number of robots in searching mode:

\[
N_s(k+1) = N_s(k) - [\Delta g_1(k) + \Delta g_2(k) + p_w + p_R]N_s(k) + \Delta g_1(k-T_{nga})\Gamma(k;T_a)N_s(k-T_{nga}) + \Delta g_2(k-T_{ca})N_s(k-T_{ca}) + p_wN_s(k-T_a) + p_RN_s(k-T_{ia})
\]

with time-varying coefficients (nonlinear coupling):

\[
\Delta g_1(k) = p_{g1}[M_0 - N_g(k) - N_d(k)]
\]

\[
\Delta g_2(k) = p_{g2}N_g(k)
\]

\[
\Gamma(k;T_{SL}) = \prod_{j=k-T_g-T_{SL}}^{k-T_{SL}}[1 - p_{g2}N_s(j)]
\]

- 6 states: 5 DE + 1 cons. EQ
- \(T_i, T_a, T_d, T_c \neq 0; T_{xyz} = T_x + T_y + T_z\)
- \(T_{SL}\) = Shift Left duration
- [Martinoli et al., *IJRR*, 2004]
Swarm Performance Metric

Collaboration rate: # of sticks per time unit

\[ C(k) = p_{g2} N_s(k-Tca) N_g(k-Tca) \]

: mean # of collaborations at iteration k

\[ \sum_{k=0}^{T_e} C(k) \]

\[ C_t(k) = \frac{\sum_{k=0}^{T_e} C(k)}{T_e} \]

: mean collaboration rate over \( T_e \)
Results (Standard Arena)

Discrepancies because of ODE approximation (nonlinearities + discrete exact vs. average quantities)
Results: 4 x #Sticks, #Robots and Arena Area

![Graph showing collaboration rate vs gripping time parameter for different numbers of robots and different arena sizes.]

- **Submicro (10 runs)**
- **Micro (100 runs)**
- **Macro (1 run)**
Reducing the Macroscopic Model

Goal: reach mathematical tractability

\[ T_i, T_a, T_d, T_c \ll T_g \rightarrow T_i = T_a = T_d = T_c = 0 \]
Reduced Macroscopic Model

Nonlinear coupling!

\[ N_s(k+1) = N_s(k) - p_{g1}[M_0 - N_g(k)]N_s(k) + p_{g2}N_g(k)N_s(k) \]

\[ + p_{g1}[M_0 - N_g(k-T_g)]\Gamma(k;0)N_s(k-T_g) \]

\[ N_g(k+1) = N_0 - N_s(k+1) \]

\[ \Gamma(k;0) = \prod_{j=k-T_g}^{k}[1 - p_{g2}N_s(j)] \]

Initial conditions and causality

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N_s(0) = N_0, N_g(0) = 0)</td>
<td>(N_s = \text{average # robots in searching mode})</td>
</tr>
<tr>
<td>(N_s(k) = N_g(k) = 0)</td>
<td>(N_g = \text{average # robots in gripping mode})</td>
</tr>
<tr>
<td>(N_s(k) = N_g(k) = 0)</td>
<td>(N_0 = # \text{robots used in the experiment})</td>
</tr>
<tr>
<td>(M_0 = # \text{sticks used in the experiment})</td>
<td></td>
</tr>
<tr>
<td>(\Gamma = \text{fraction of robots that abandon pulling})</td>
<td></td>
</tr>
<tr>
<td>(T_e = \text{maximal number of iterations})</td>
<td></td>
</tr>
<tr>
<td>(k = 0, 1, \ldots T_e)</td>
<td>(iteration index)</td>
</tr>
</tbody>
</table>

\(\text{successful}\) \\
\(\text{unsuccessful}\)
Results Reduced Micro/Macroscopic Models

- Reduced microscopic (100 runs) and macroscopic models overlapped
- Only qualitative agreement with submicroscopic/real robots results

- 4 robots, 4 sticks, $R_a = 40$ cm

- 16 robots, 16 sticks, $R_a = 80$ cm
Steady State Analysis (Reduced Macro Model)

- Steady-state analysis \([N_n(k+1) = N_n(k)]\) → It can be demonstrated that:

\[
\exists T_g^{opt} \text{ for } \frac{N_0}{M_0} \leq \frac{2}{1 + R_g}
\]

with \(N_0 =\) number of robots and \(M_0 =\) number of sticks, \(R_g \propto\) approaching angle for collaboration

- **Counterintuitive conclusion:** an optimal \(T_g\) can exist also in scenarios with more robots than sticks if the collaboration is very difficult (i.e. \(R_g\) very small)!
Analysis Verification
(Micro and Macro Full Model)

Example: \( \tilde{R}_g = \frac{1}{10} R_g \) (collaboration very difficult)

20 robots and 16 sticks (optimal \( T_g \))
Optimal Gripping Time

• Steady-state analysis → $T_{g}^{opt}$ can be computed analytically in the simplified model (numerically approximated value):

\[
T_{g}^{opt} = \frac{1}{\ln(1 - p_{g} R_{g} \frac{N_0}{2})} \ln \left( \frac{1 - \frac{\beta}{2}(1 + R_{g})}{1 - \frac{\beta}{2}} \right) \quad \text{for} \quad \beta \leq \beta_{c} = \frac{2}{1 + R_{g}}
\]

with $\beta = N_0/M_0$ = ratio robots-to-sticks

• $T_{g}^{opt}$ can be computed numerically by integrating the full model ODEs or solving the full model steady-state equations

An Example with Nonlinearities and Time-Varying Parameters: Distributed Seed Assembly
Robot Behavior

- Reactive, non-communicating, non-adaptive behavior
- 1 robot state: loaded, free
- **Qualitative stigmergy significant:**
  - 2 rules in interaction with cluster
  - Avoid if interaction with the cluster body
  - Manipulate if interaction with cluster tips
- **Quantitative stigmergy minimal:**
  - the bigger, the more stable the cluster
  - big cluster (> 2) = number of manipulation sites as cluster of 2 seeds
  - almost no difference between cluster incrementing and decrementing probabilities
Metrics

- Stigmergy means leaving signs in the environment
- All metrics are related to the evolution of the seed assemblies (clusters) on ground (e.g., number of clusters, size of the biggest cluster, etc. see also Week 6).
- System states must also be capturing this evolution in order to be able to formulate a metric leveraging them.
Robot Controller – Intuitive

Start → Search

Object?

Seed?

Seed picking-up

Seed dropping

Loaded?

Interference

Obstacle Avoidance

Robot?
State Granularity Choices

• Idea: split the search state in search (loaded) and search (free) and keep dedicated avoidance/interferences states for each of the split states

• Motivation:
  • Given the chosen metrics, seeds need to be tracked also when they are in the robot grippers (conservation law of seeds)
  • Status loaded (carrying a seed) or free (not carrying a seed) can only change in a deterministic fashion (e.g., cannot be changed by an avoidance operation) and only by going to one of the seed dropping or picking states → can be explicitly represented with a larger state space and deterministic transitions
  • Such enlarged state space facilitates the writing of the ODEs
Robot Controller – Restructured

Start

Search (free)

Object?

Y

Seed?

N

Seed picking-up

Search (loaded)

Object?

Y

Seed?

N

Seed dropping

Interference

Y

N

Obstacle Avoidance

Y

N

Robot?
Robot Controller – Restructured

1. Start

2. Search (free)
   - Object?
     - Seed?
       - Seed picking-up
         - Search (loaded)
           - Object?
             - Seed?
               - Seed dropping
               - Interference
                 - Obstacle Avoidance
   - Interference
     - Obstacle Avoidance
   - Robot?

3. Obstacle Avoidance
   - Interference
     - Robot?

4. Interference
   - Robot?

5. Obstacle Avoidance
   - Interference
     - Robot?
Robot Controller

From Agassounon et al, 2004 (restructured representation):
Micro and Macroscopic Models

From Agassounon et al, 2004 (restructured representation):

Robots always active
(no worker allocation)
Parameter Calibration

Geometric Estimations

• **Incrementing** probabilities

![Incrementing probabilities diagram]

• **Decrementing** probabilities

![Decrementing probabilities diagram]

Perimeters are relevant for computing the cluster modifying probabilities: robot turns on the spot for object distinction before approaching the cluster!
Models: Explanations and Predictions

Single cluster? All models predicted yes and in roughly how much time!

Number of clusters (inter-distance between seeds < 1seed) monotonically decreases if:

- Probability to create a NEW cluster of 1 seed in the middle of the arena is equal to zero
- No hard partitioning of the arena (robot homogeneously mix clusters)
- Cluster are not broken in two parts by removing one seed in the middle
Long Seed-Assembling Experiments

**Submicroscopic Model**  
(Webots)

- 10% white noise on all sensor and actuators
- Perfectly homogeneous team
- Kinematic mode

**Real robots**  
(Khepera I)

- Electrical floor: continuous power supply in any position and orientation
- Heterogeneities among teammates and components
- Inaccuracies in acting and sensing
- Dynamics (e.g., friction) plays a role
Results – Until a Single Cluster

- 3 robots
- real robots (5 runs), submicroscopic (10 runs), microscopic model (100 runs)
- [Martinoli, Ijspeert, Mondada, 1999]

- Mean size of clusters
- Size of the biggest cluster
- Number of clusters
Examples of Assembled Structures

Noise in S&A and poor navigation capabilities do not allow for precise, controllable structure building.

Submicroscopic

Real robots
Macroscopic Model: Distributed Building Dynamics

- \( d_i(k) = \text{decr}_i \cdot \text{prob}_i \cdot p_{\text{find}}_i(k) \)
- \( c_i(k) = \text{incr}_i \cdot \text{prob}_i \cdot p_{\text{find}}_i(k) \)
- \( p_{\text{find}}_i(k) = \text{finding probability of all the cluster of size } i \)
- If \( n \) = number of seeds -> macroscopic model of environment with \( n \) nonlinearly coupled ODE (\( n \) for each possible cluster size) + robot states

Some Results from Agassounon et al., 2004 (1, 5, 10 robots always active)

**Metric:** average cluster size (20 seeds)

- **1 and 5 robots**
  - Saturation phase: all seeds in a single cluster or in the robots’ grippers

- **10 robots**
Micro and Macroscopic Models

Robots can go resting (worker allocation)

Note: see also Week 6
Some Results from Agassounon et al., 2004 (10 robots with activity regulation)

20 seeds, threshold for abandoning the arena = 25 min, 10 robots

No more saturation: growing phase beyond 10-seeds single cluster

Average cluster size

Number of active robots

Note: see also Week 6
Journal Publications using the Same Modeling Framework

**Stick Pulling**
- [Lerman, Galstyan, Martinoli, Ijspeert, *Artificial Life*, 2001]

**Object Aggregation**

**Robot Aggregation and Swarming** – explore no arena bounds
- [Winfield, Liu, Nembrini, Martinoli, *Swarm Intelligence J.*, 2008]

**Coverage** – use spatial models
Combined Modeling and Machine-Learning Methods
Rationale for Combined Methods (1)

- Any level of modeling (submicro, micro, or macro) allow us to consider certain parameters and leave others; models, as expression of reality abstraction, can be considered as more or less coarse “filters” of the reality

- Combined modeling/machine-learning techniques can be used at any of the abstraction levels; machine-learning techniques will explore the design parameters explicitly represented at a given level of abstraction

- Depending on the features of the hyperspace to be searched (size, continuity, noise, etc.), appropriate machine-learning techniques should be used (e.g., hill-climbing vs. population-based, online vs. offline)

- One particular optimization problem is system identification: the performance to optimize is the matching with the reality (or with a lower abstraction level). See model calibration in [Correll & Martinoli, DARS 2006].
Rationale for Combined Methods (2)

- **Macroscopic + ML?** Most of the time not needed since very fast + continuous; homogeneous systems mainly; standard numerical optimization techniques/systematic search can be used.

- **Microscopic + ML** (see this lecture’s examples); for instance, diversity and specialization can be studied.

- **Submicroscopic + ML** (see Week 11 and 12 examples using PSO); for instance low-level design parameters can be learned.

- **Target system + ML = adaptation with HW in the loop** (on-board or off-board).
In-Line Adaptive Learning
In-Line Adaptive Learning (Li, Martinoli, Abu-Mostafa, 2001)

- **GTP**: Gripping Time Parameter
- **Δd**: learning step
- **d**: direction
- Underlying low-pass filter for measuring the performance
Algorithm Parameters

<table>
<thead>
<tr>
<th>Algorithmic parameters:</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_m$</td>
<td>2400</td>
<td>averaging period for reinforcement signal (sec)</td>
</tr>
<tr>
<td>$E$</td>
<td>1.9</td>
<td>GTP offset enlarge factor</td>
</tr>
<tr>
<td>$F$</td>
<td>0.3</td>
<td>GTP factor enlarge ratio</td>
</tr>
<tr>
<td>$U$</td>
<td>2</td>
<td>GTP offset shrink divider</td>
</tr>
<tr>
<td>$V$</td>
<td>0.5</td>
<td>GTP factor shrink ratio</td>
</tr>
</tbody>
</table>

From Li et al., *Adaptive Behavior*, 2004
In-Line Adaptive Learning

**Differences with gradient descent methods:**
- Fixed rules for calculating step increase/decrease → limited descent speed → no gradient computation → more conservative but more stable
- Randomness for getting out from local minima (no momentum)
- Underlying low-pass filter is part of the algorithm

**Differences with Reinforcement Learning:**
- No learning history considered (only previous step)

**Differences with basic In-Line Learning:**
- Step adaptive → faster and more stability at convergence
Co-Learning in a Collaborative Framework
Sample Results – Homogeneous Learning

Short averaging window
(filter cut-off $f_{\text{high}}$)

Long averaging window
(filter cut-off $f_{\text{low}}$)

-- Systematic (mean only)
--- Learned (mean + std dev)

Note: 1 parameter for the whole group!
Heterogeneous Learning

Key question: does team diversity enhance performance? I.e., can individual members become specialized?

Performance ratio between 2 caste and homogeneous system (submicro/micro models, systematic search)

4 robots, one per color, micro + learning
Heterogeneous vs. Homogeneous Learning

Performance ratio between heterogeneous (full and 2-castes) and homogeneous groups AFTER learning

Notes:
- large $T_m$ (long averaging window)
- only private strategies
- global = group
  local = individual

[Li et al., Adaptive Behavior, 2004]
Measuring Diversity and Specialization
Diversity Metrics
(Balch 1998)

Entropy-based diversity measure introduced in AB-04 could be used for analyzing threshold distributions

Simple entropy: \[ H(\mathcal{R}) = -\sum_{i=1}^{m} p_i \log p_i. \]
Social entropy: \[ D(\mathcal{R}) = \int_{0}^{\infty} H(\mathcal{R}, h) \, dh. \]

\( p_i \) = portion of the agents in cluster \( i \); \( m \) cluster in total; \( h \) = taxonomic level parameter

**Input:** a swarm system \( \mathcal{R} = \{r_1, r_2, \ldots, r_n\} \) of size \( n \); a difference measure \( d \).
For different level \( h \), the \( C_u \) clustering algorithm does:

1. Initialize \( n \) clusters with cluster \( c_i = \{r_i\} \);
2. For each \( c_i \): for each \( r_j \): if \( d(r_j, r_k) \leq h \) for all \( r_k \) in \( c_i \), add \( r_j \) to cluster \( c_i \);
3. Discard redundant clusters;
4. Calculate \( p_i \) and the entropy \( H(\mathcal{R}, h) \). Note that when \( r_j \) belongs to \( s \) clusters including \( c_i \), its contribution to \( p_i \) is \( 1/sn \).

Return \( \int_{0}^{\infty} H(\mathcal{R}, h) \, dh \) as the hierarchic social entropy.
Example – Simple Entropy

- \( R = \{r_1, r_2, r_3\} \)
- \( n = 3 \) (three swarm points)
- bi-dimensional space
- define a distance: Euclidian distance
- \( h = \) taxonomic level parameter
- \( m = \) number of clusters

\[
H(R) = -\sum_{i=1}^{3} p_i \log p_i = H(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}) =
\]
\[
-\frac{1}{3} \log \frac{1}{3} = 0.477
\]

\[
H(R) = -\sum_{i=1}^{2} p_i \log p_i = H(\frac{1}{3}, \frac{2}{3}) =
\]
\[
-\frac{1}{3} \log \frac{1}{3} - \frac{2}{3} \log \frac{2}{3} = 0.159 + 0.117 = 0.276
\]
Example – Simple Entropy

\[ H(R) = - \sum_{i=1}^{2} p_i \log p_i = H\left(\frac{1}{3} + \frac{1}{3}, \frac{1}{3} + \frac{1}{3}\right) = H\left(\frac{1}{2}, \frac{1}{2}\right) \]

\[ H(R) = - \sum_{i=1}^{1} p_i \log p_i = H\left(\frac{3}{3}\right) = - \log 1 = 0 \]

\[ -\frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \log \frac{1}{2} = 0.301 \]

Check \[ \sum_{i=1}^{m} p_i = 1 \] with overlapping clusters!
Example – Social Entropy

\[ D(R) = \int_0^\infty H(R, h) dh = 3 \times H\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) \]
\[+ 1 \times H\left(\frac{1}{3}, \frac{2}{3}\right) \]
\[+ 2 \times H\left(\frac{1}{2}, \frac{1}{2}\right) + 0 = 2.309 \]

Note: In contrast to simple entropy \( \geq 1 \)

Contrast with \( R = \{r1, r2, r3\} \) and \( r_1 = r_2 = r_3 \) (homogeneous swarm),
for any \( h \geq 0 \) → single cluster → \( D(R) = 0! \)
Differences with Plain Euclidian Diversity Measure

Underlying distance measure in the solution space might be the same (e.g. Euclidian distance)

Social entropy is looking for possible clustering of the vectors (looking for possible castes) while Euclidian diversity is just looking how spread out/diverse in general are the vectors

\[
d(a, b) = \sqrt{\sum_i (a_i - b_i)^2}
\]

\[
D_{eu} = \frac{1}{N(N-1)} \sum_a \left[ \sum_{b \neq a} d(a, b) \right]
\]

Components in all dimensions

All points from any other point
Specialization Metric

Specialization metric introduced in AB-04:

\[ S = \text{corrcoef}(D; R) \times D. \]

S = specialization; D = diversity (e.g., social entropy); R = swarm performance

Notes

• Idea: “weighting diversity with performance”
• This is useful when the number of tasks to be solved is not well-defined or it is difficult to assess the task granularity a priori. In such cases the mapping between task granularity and caste granularity might not trivial (see the limited performance of a caste-based solution in the stick-pulling experiment)
• Could be used for analyzing specialization arising from a variable-threshold division of labor algorithm (see lecture Week 6)
Sample Results with Canonical Sticks

- 2 serial grips needed to get the sticks out
- 4 sticks, 2-6 robots, 80 cm arena

**Relative Performance**
- Specialists more important for small teams
- Local \( p \) > global \( p \)
- Enforced caste: pay the price for odd team sizes

**Diversity**
- Measured using social entropy
- Flat curves, difficult to tell whether diversity brings performance

**Specialization**
- Specialization higher with global when needed, drop more quickly when not needed
- Enforcing caste: “low-pass filter” effect
Conclusion
Take Home Messages

- The multi-level modeling methodology is a framework that has been successfully used in multiple case studies.
- An additional case study has illustrated how to capture time-varying parameters depending on the environmental modifications introduced by the robots and how to choose an appropriate state granularity for computing the targeted metrics.
- If carefully designed, models allow also for system optimization and closing the loop between analysis and synthesis.
- Different modeling levels can be combined with machine-learning for design and optimization purposes.
- Microscopic models allows for efficiently studying questions concerned with diversity and specialization.
- Specialization is the part of diversity that improves performance.
- The diversity and specialization level of a heterogeneous swarm can be quantitatively measured.
Papers


