Augmented Self-Assembly

Solving the Intrinsic System Augmentation problem
for a self-assembly task

Midterm Presentation

Loïc Matthey

Supervisors:

Vijay Kumar
Grégory Mermoud
Alcherio Martinoli

Special contributions:

Spring Berman

Penn
University of Pennsylvania

Master Project - MP12 - 07.07.08

EPFL
École Polytechnique Fédérale de Lausanne

Contents

1. Introduction

Context

- Joint work with the GRASP Lab from University of Pennsylvania (Penn), Prof. Vijay Kumar.
- Combine interests in assembly and multi-robots teams.
- Strong theoretical foundations needed.
- Considered problem:
  - **Stochastic assembly of products**

1. Introduction

Augmented assembly

- Problem of stochastic: poor yield
- Idea: add agents to the initial system to improve performance.

  ▶ **Augmented system.**

- Questions:
  - How to formulate ?
  - How to augment the system ?

2. Goals

- Propose a theoretical framework for the Augmented System problem.
- Validation using a higher-level assembly task (biological scale).
  - Use Webots as physics simulator.
- Develop mathematical models and simulations fitting the tasks.
  - Use Chemical Reactions formalism.
- Optimize the Augmented System using mathematical foundations.

3. Stochastic Assembly

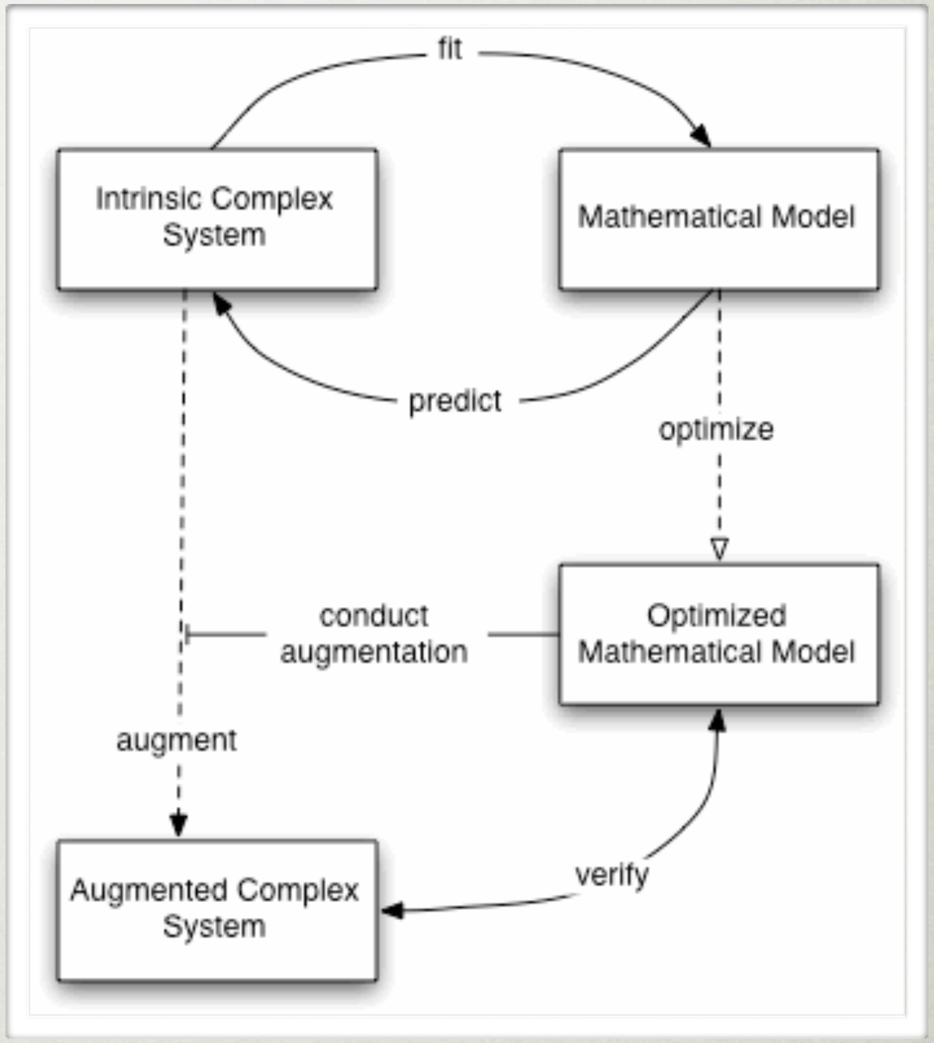
Definition

- Let M_i pieces of different types, assembling with bi-directional connections.
- Let those pieces move and assemble randomly in an arena of size A.
- Let the final assembled products be known as S_j.
- Let a system of reactions R showing the assembling of pieces via their connections. These reactions can contain disassembling reactions too.
- Then this system will create a certain amount $|S_j|$ after a time T_f.

  - Goal: obtain the bigger $|S_j|$ after the smaller T_f.

4. Intrinsic System Augmentation

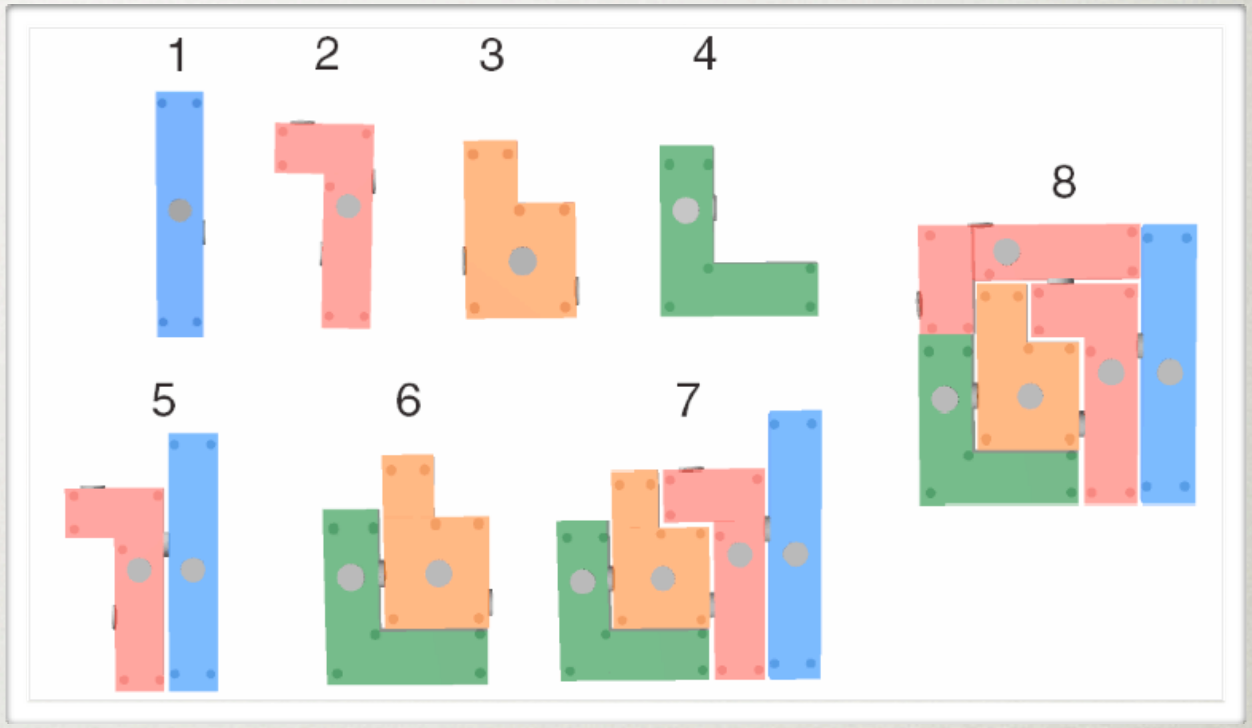
- Define two systems:
  - Intrinsic (initial) system.
  - Augmented system.
- Define a common metric (systems should be measurable).
- Model the intrinsic system behavior.
- Optimize the model of the intrinsic system.
- Link the Augmented system to the optimized model.



5. Puzzle test case

Description

- Simplification of an assembly task.
- Specific pieces to assemble.
- Robot to carry the pieces.
  - still self-assembly? Yes.
- Measure scalability and efficiency of assembly.
  - Stochastic controllers
  - “Designed” controllers



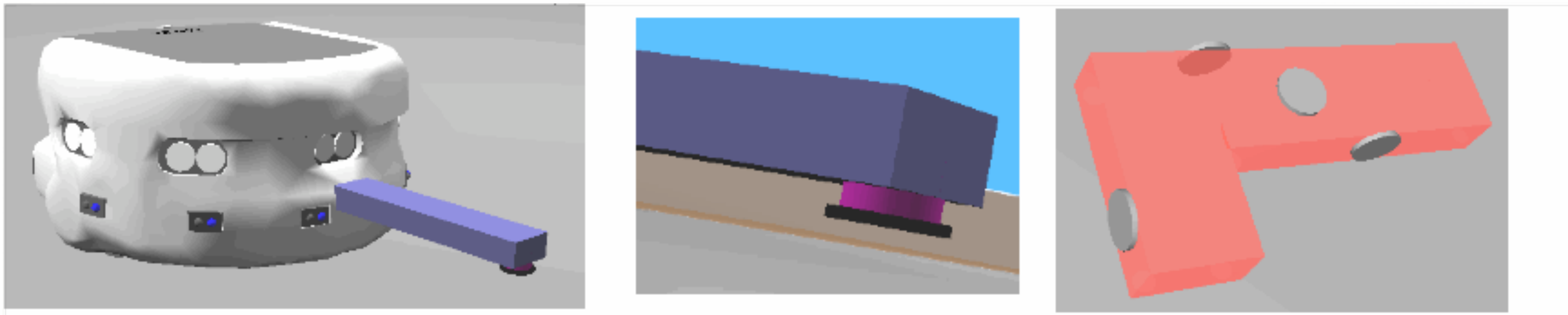
5. Puzzle test case

Relation to Augmented System

- Intrinsic and Augmented system ?
- Stochastic controllers:
  - Like self-assembly of pieces, assuming they move and assemble randomly.
- "Designed" controllers:
  - "Augment" virtually the system by changing the stochastic behavior for something better.
  - This is similar to adding extra robots to conduct this new behavior.

6. Webots Implementation

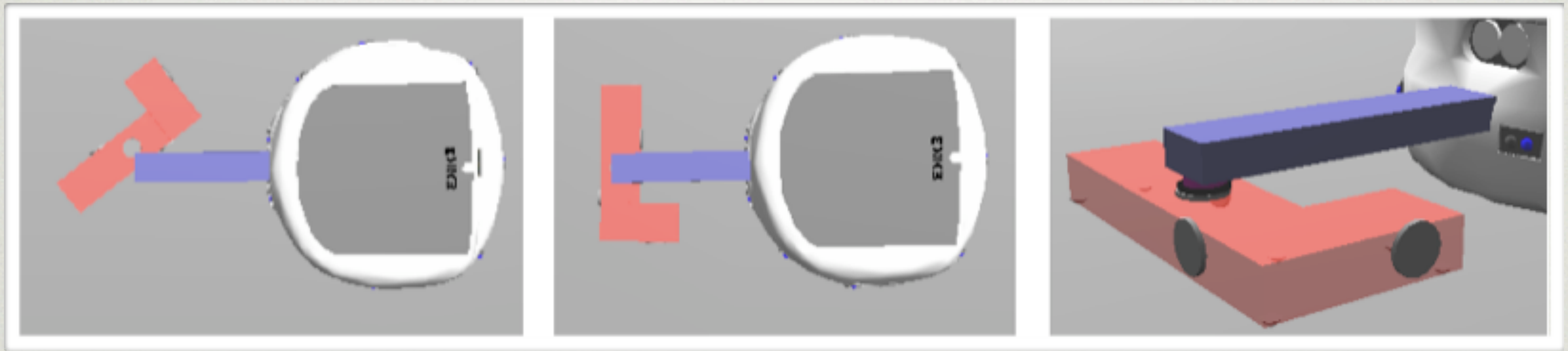
Basic components


(a) (b) (c)

- Khepera III modified.
  - Protruding arm to carry the pieces.
- Arm with rotating connector.
- Pieces with several connectors.
  - Top connector for the robots.
  - Side connectors to assemble to other pieces.

6. Webots Implementation

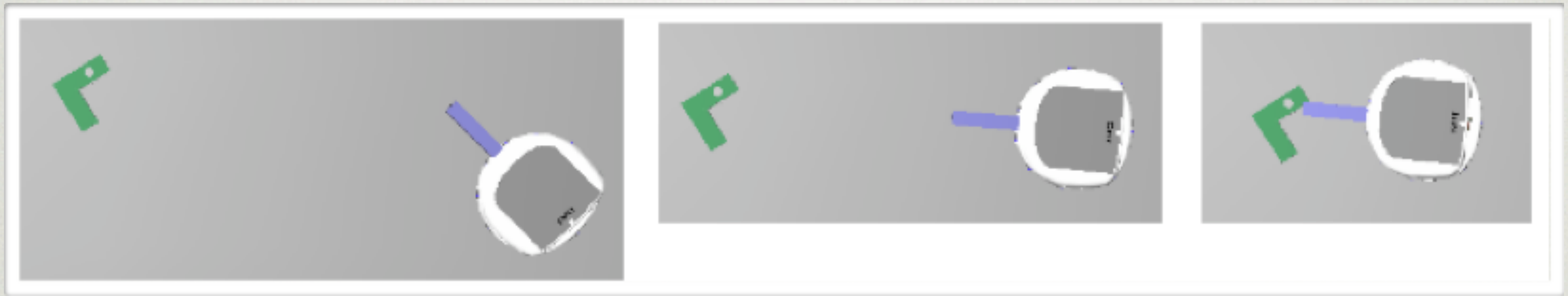
Capabilities



- Rotating connector:
  - Allows full access to every side connectors
- Pieces and robots communicate with emitter/ receiver.
  - Offers relative positioning for alignment

6. Webots Implementation

Capabilities



- Rotating connector:
  - Allows full access to every side connectors
- Pieces and robots communicate with emitter/ receiver.
  - Offers relative positioning for alignment

6. Webots Implementation

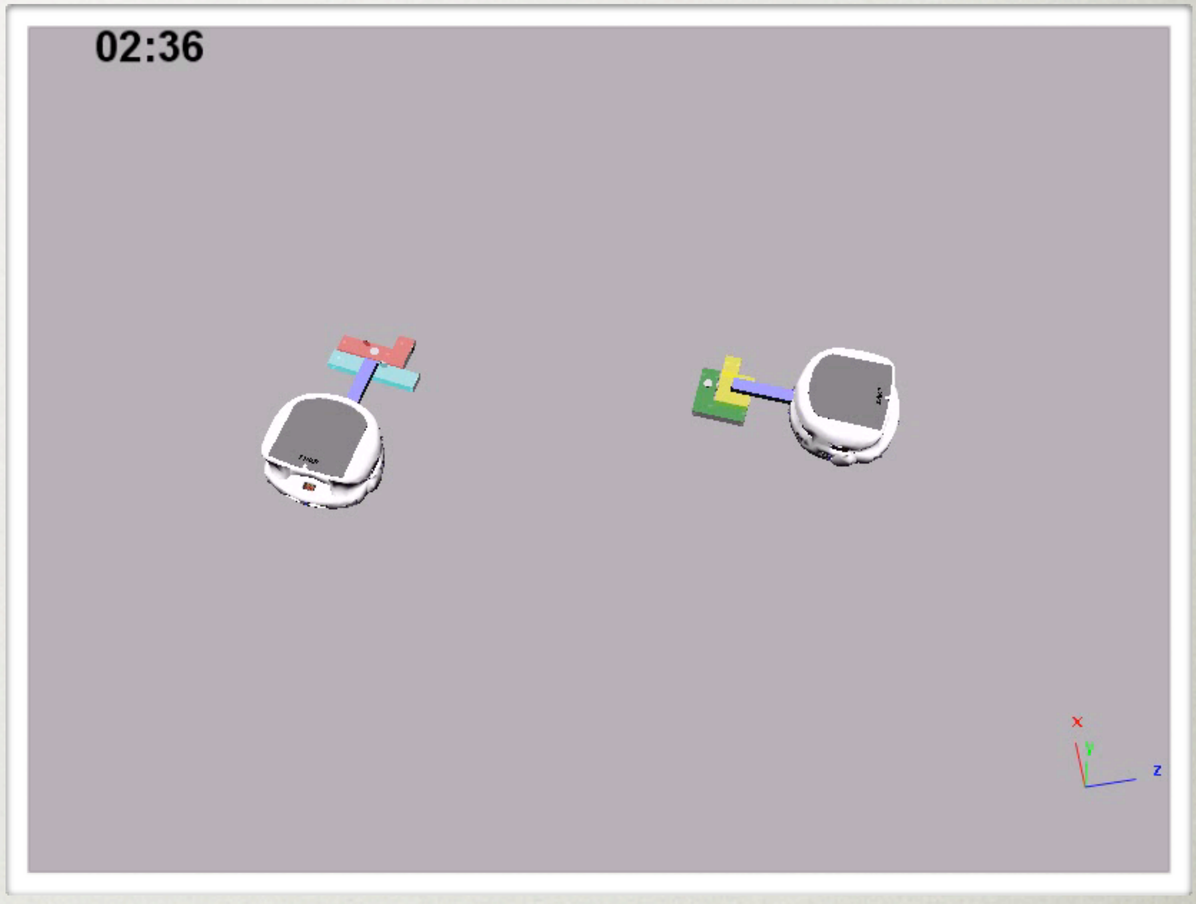
Systematic experiments

- Webots World Generator written in Python, to generate specific number of robots and pieces.
- Random initial positioning for every robot and piece in the arena.
- Supervisor handling the experiments and writing to files.

6. Webots Implementation

Systematic experiments

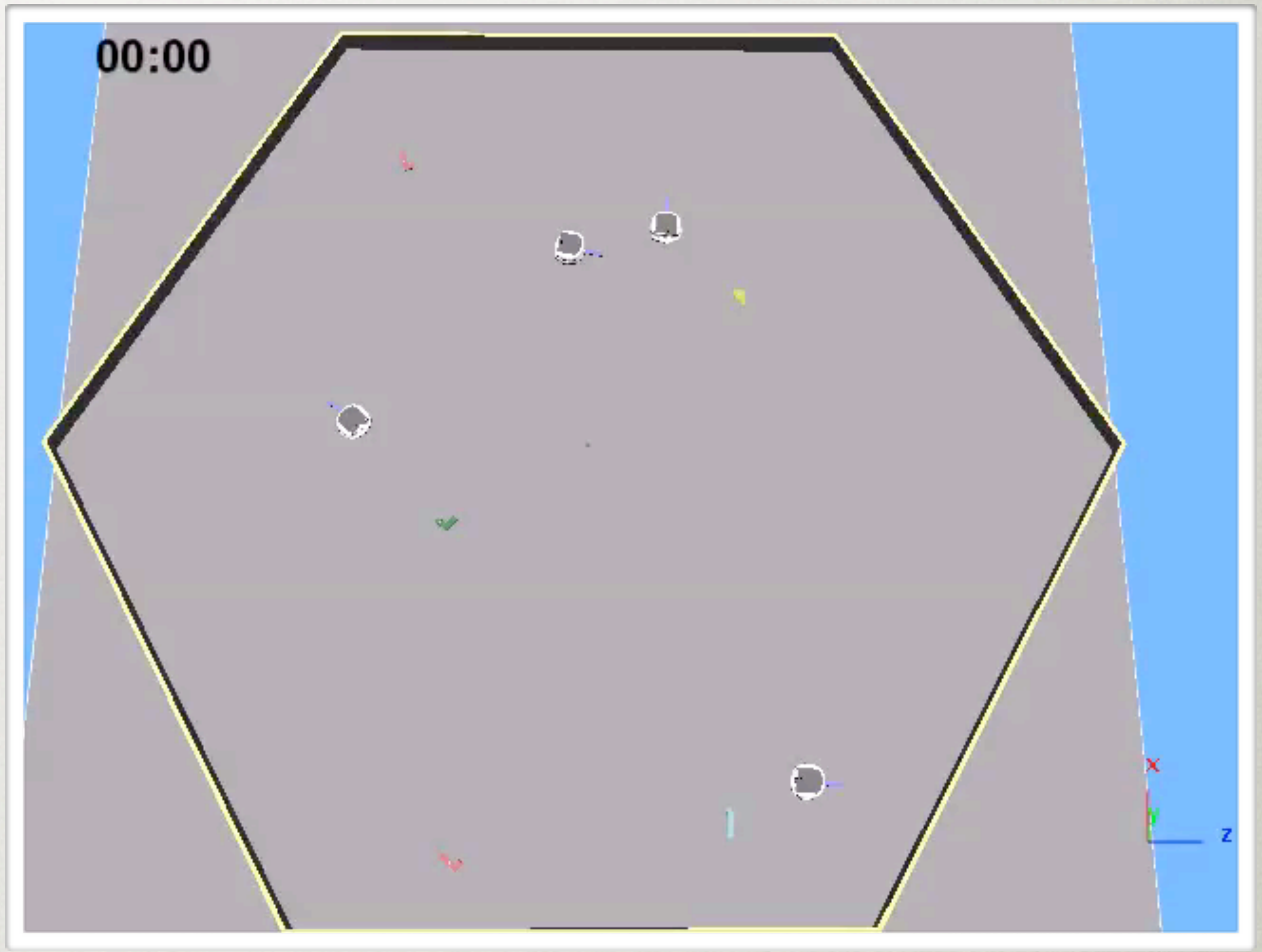
- Webots World Generator written in Python, to generate specific number of robots and pieces.
- Random initial positioning for every robot and piece in the arena.
- Supervisor handling the experiments and writing to files.



6. Webots Implementation

Behavior

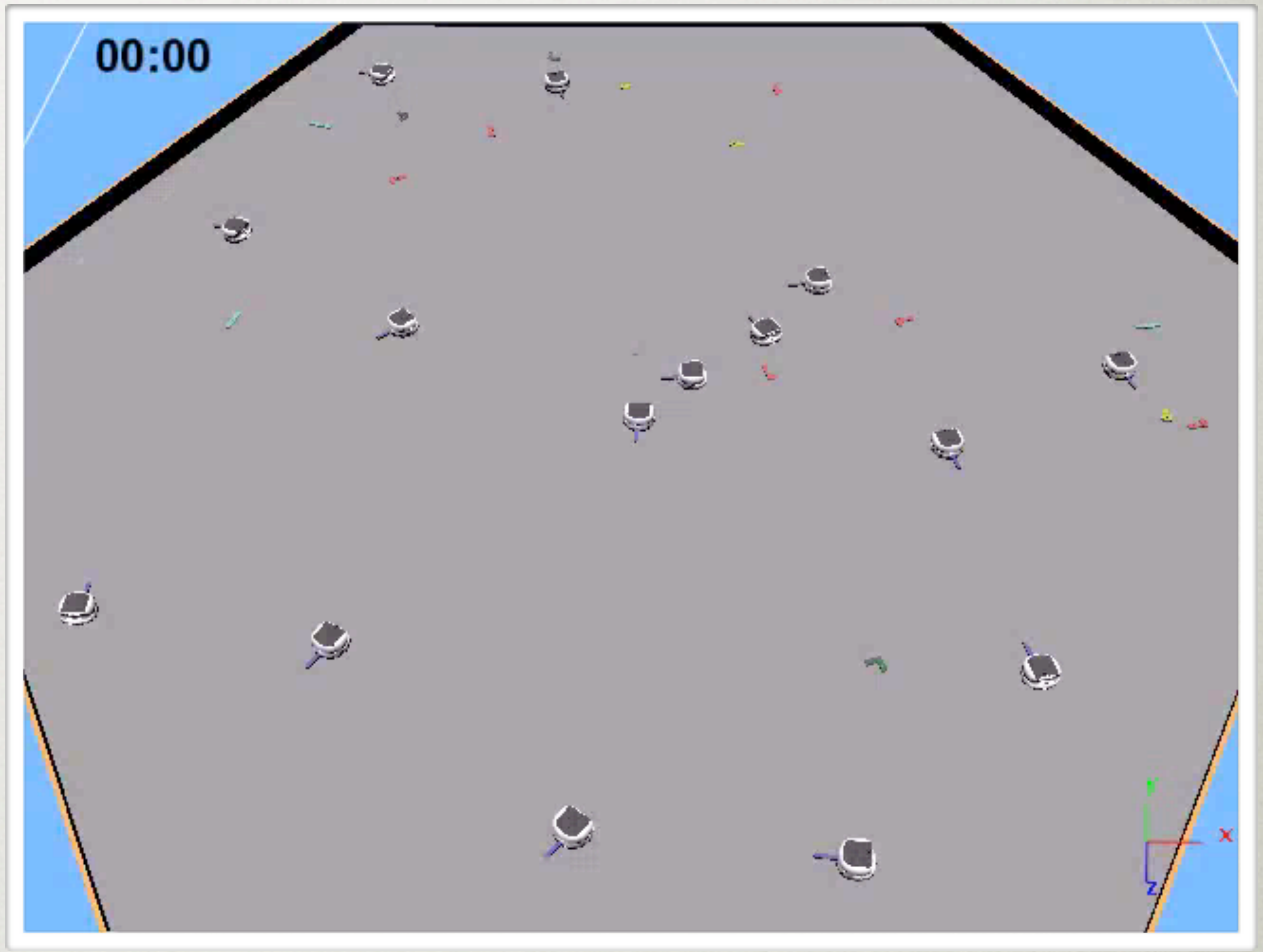
- **Spring's work, adapted.**
- All local radio communications (0.4m radius).
- Robot search for lying pieces and carry them.
- When encounter a robot, check if the assembly is possible.
- Alignment and assembling of pieces.
- Answer to a given plan, written in forms of reactions.



6. Webots Implementation

Behavior

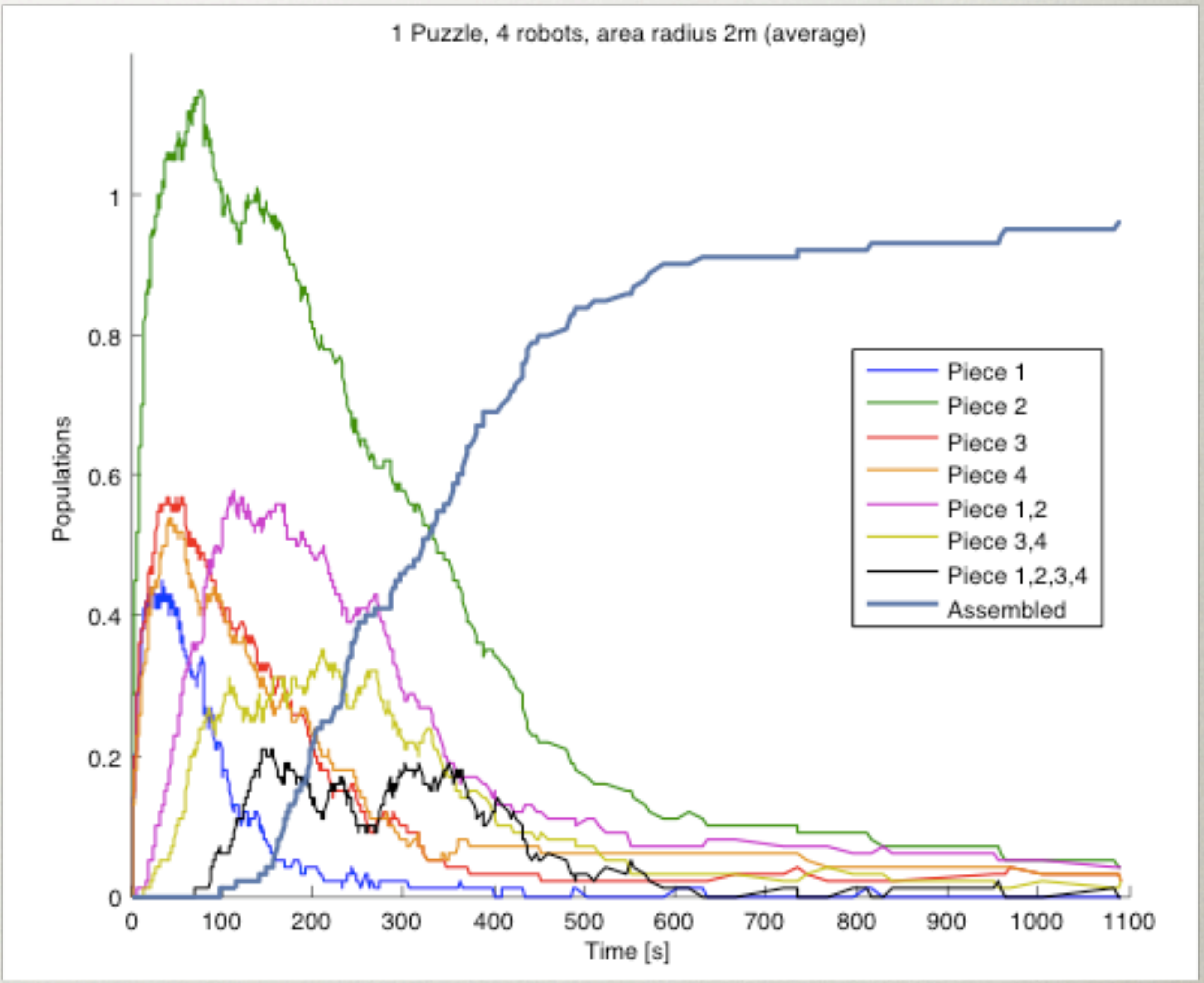
- **Spring's work, adapted.**
- All local radio communications (0.4m radius).
- Robot search for lying pieces and carry them.
- When encounter a robot, check if the assembly is possible.
- Alignment and assembling of pieces.
- Answer to a given plan, written in forms of reactions.



6. Webots Implementation

Results

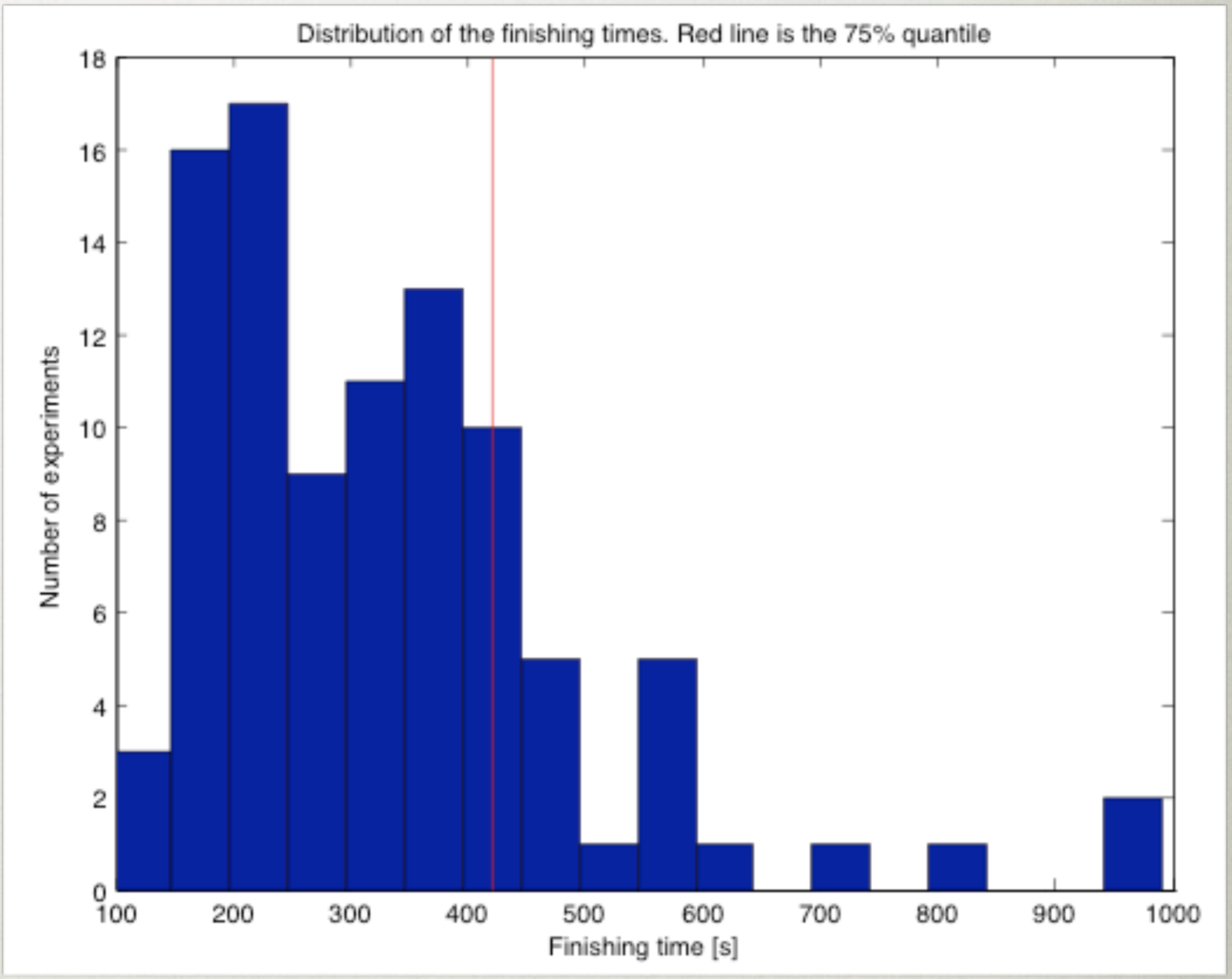
- **1 Puzzle (5 pieces), 4 robots**
  - 100 experiments, 20 min maximum.
  - Initial positions and rotations between each of them.
  - 96% assembly success in 20 min.
  - 75% of all experiments assembled after 6 min.



6. Webots Implementation

Results

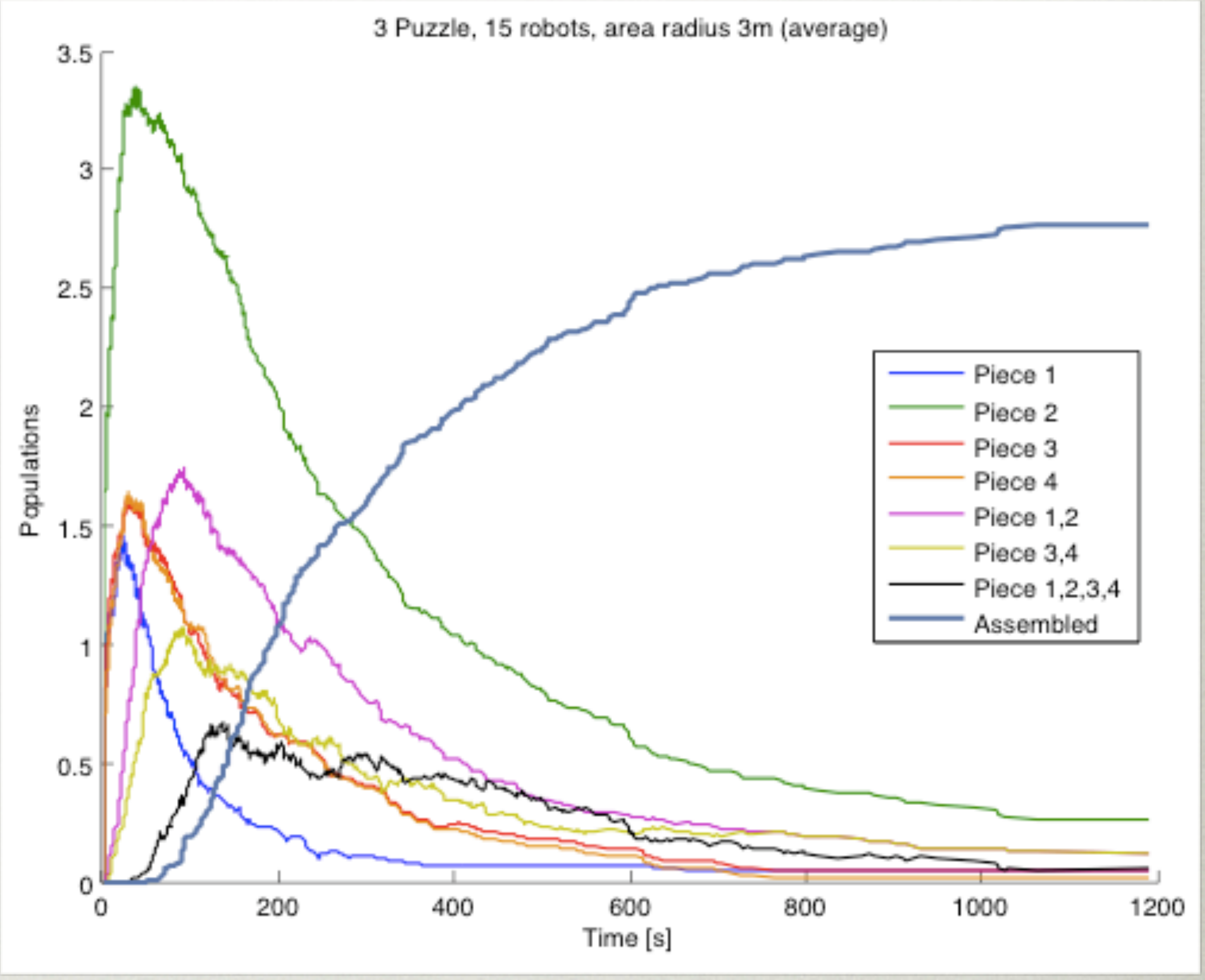
- **1 Puzzle (5 pieces), 4 robots**
  - 100 experiments, 20 min maximum.
  - Initial positions and rotations between each of them.
  - 96% assembly success in 20 min.
  - 75% of all experiments assembled after 6 min.



6. Webots Implementation

Results

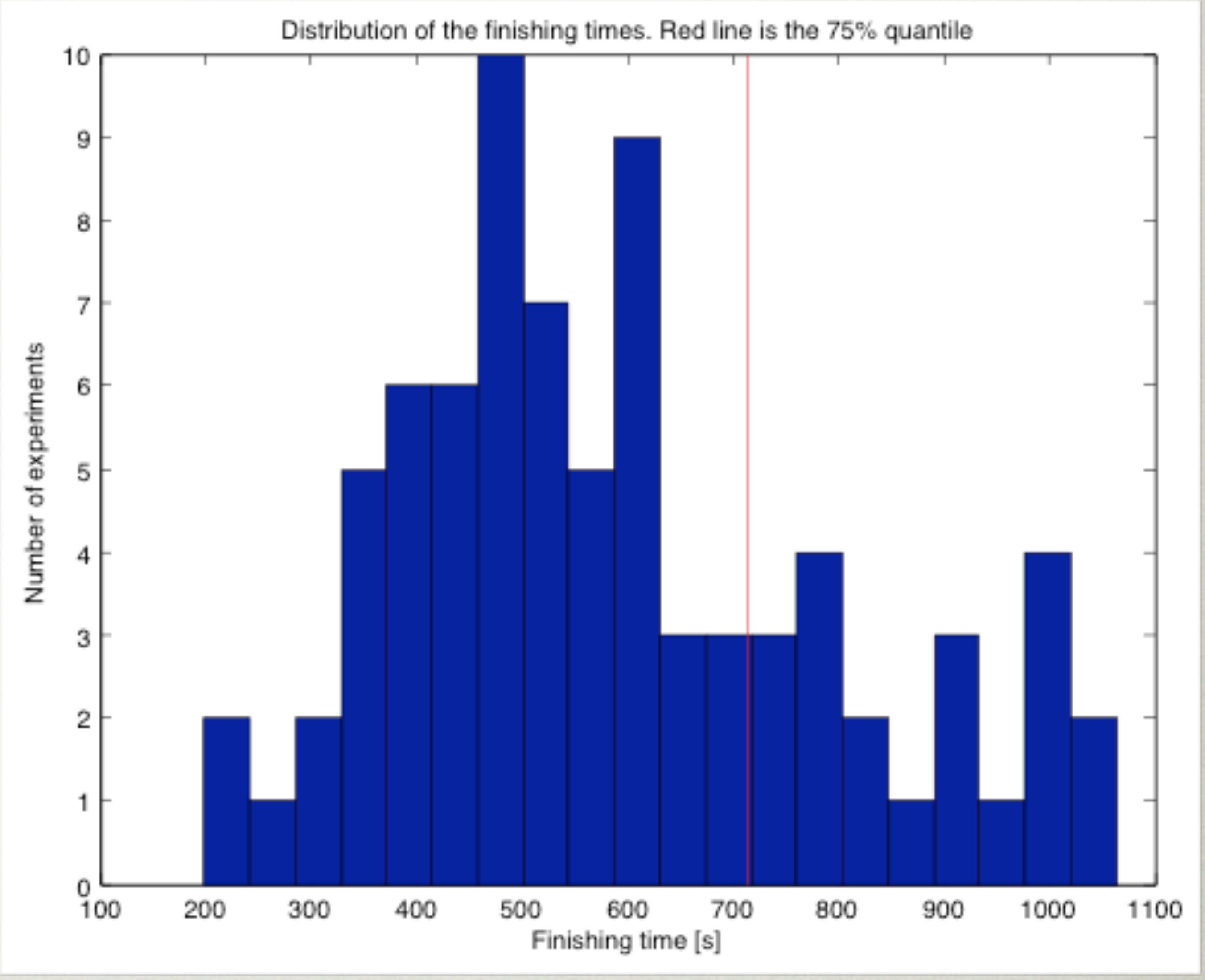
- **3 Puzzle (15 pieces), 15 robots**
    - 100 experiments, 20 min maximum.
    - Initial positions and rotations between each of them.
    - 80% assembly success.
    - 75% of all assemblies done after 11 min.



6. Webots Implementation

Results

- **3 Puzzle (15 pieces), 15 robots**
  - 100 experiments, 20 min maximum.
  - Initial positions and rotations between each of them.
  - 80% assembly success.
  - 75% of all assemblies done after 11 min.



7. Models

Description

- Chemical Reactions Networks.
- Used for chemical and biological processes.
- Well adapted because of flexibility and versatility.

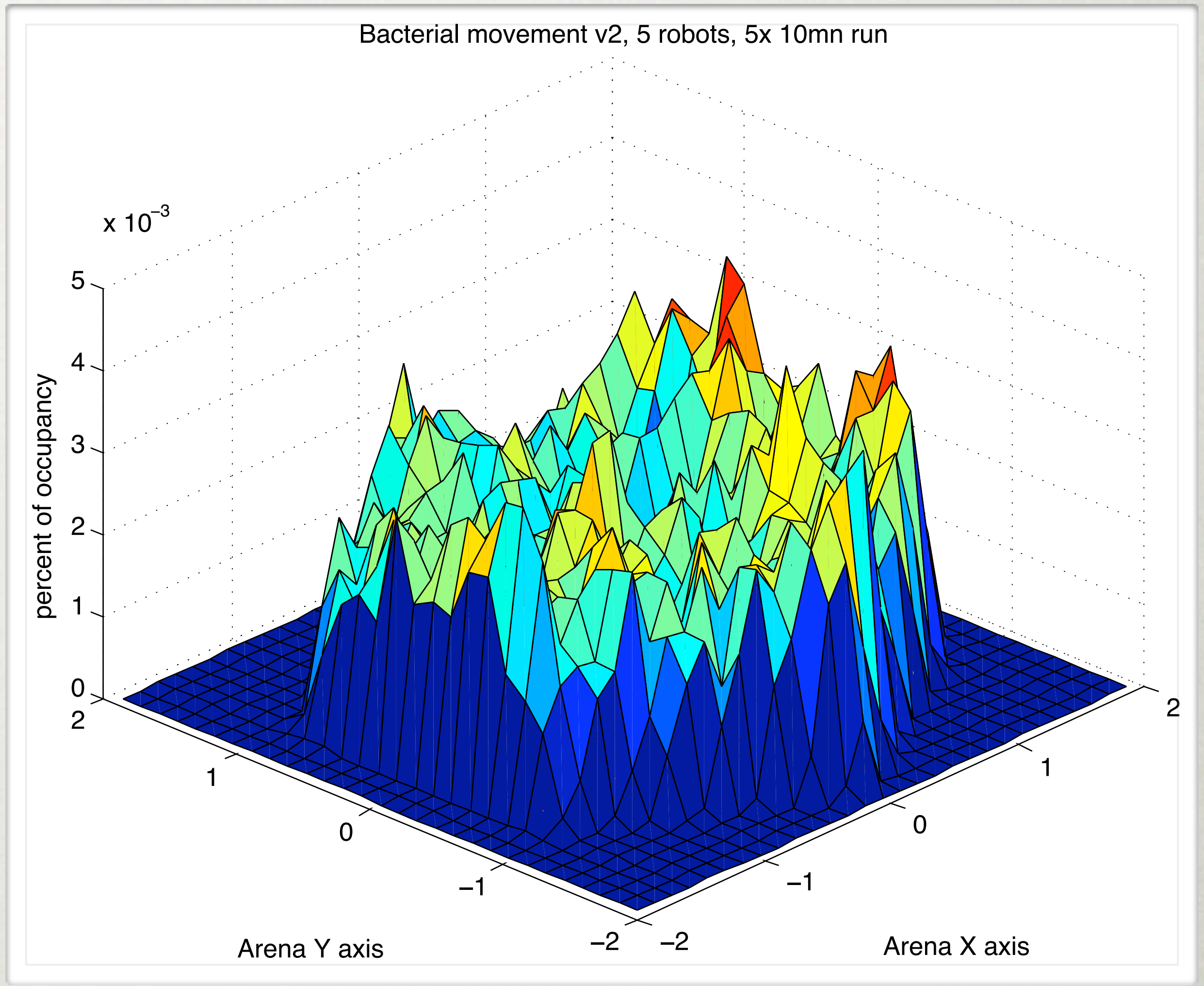
$$
\begin{aligned}
R + P_i^f &\xrightarrow{k_1} P_i^c \\
P_1^c + P_2^c &\xrightarrow{k_2} P_5^c \\
P_3^c + P_4^c &\underset{k_3'}{\overset{k_3}{\rightleftharpoons}} P_6^c
\end{aligned}
\quad \sim \quad
\begin{aligned}
\dot{R} &= -k_1 R P_i^f \\
\dot{P_i^f} &= -k_1 R P_i^f \\
\dot{P}_1^c &= k_1 R P_1^f - k_2 P_1^c P_2^c \\
\dot{P}_2^c &= k_1 R P_2^f - k_2 P_1^c P_2^c \\
\dot{P}_5^c &= k_2 P_1^c P_2^c \\
\dot{P}_3^c &= k_1 R P_3^f - k_3 P_3^c P_4^c + k_3' P_6^c \\
\dot{P}_4^c &= k_1 R P_4^f - k_3 P_3^c P_4^c + k_3' P_6^c \\
\dot{P}_6^c &= k_3 P_3^c P_4^c - k_3' P_6^c
\end{aligned}
$$

7. Models
Simulations

- Lots of literature available.
- Exact simulation: Stochastic Simulation Algorithm (SSA) by Gillespie, 1976.
- Approximate solutions: ODE Models or Hybrid Models (Stochastic + Deterministic).
- Simulation tools chosen:
  - StochKit for SSA. L. Petzold, UCSB.
  - Matlab for ODE Models.

7. Models
Parameters estimation

- Hypothesis:
  - System should be well-mixed.
- Enforced by chemotaxis-like movement of robots.
- We can make non-spatiality assumption then.

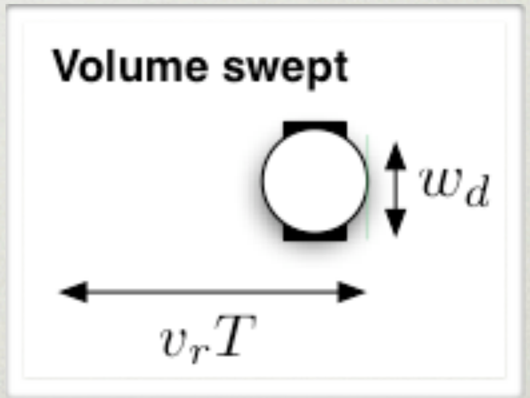


7. Models

Parameters estimation

- Reaction rates depends on encountering probabilities.
  - Measure them in Webots
  - A-priori guess using theoretical informations
- Chose to use the geometric probabilities, like N. Correll did.
  - Actually is the exact application of a chemical simulation formula to large-scale robots.

$$p_e \sim \frac{1}{A_{total}} v_r T w_d$$

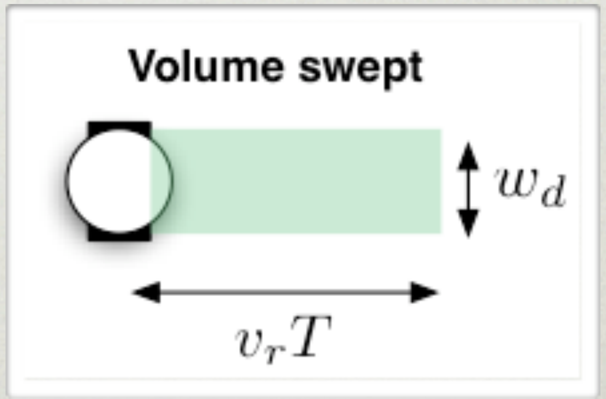


7. Models

Parameters estimation

- Reaction rates depends on encountering probabilities.
  - Measure them in Webots
  - A-priori guess using theoretical informations
- Chose to use the geometric probabilities, like N. Correll did.
  - Actually is the exact application of a chemical simulation formula to large-scale robots.

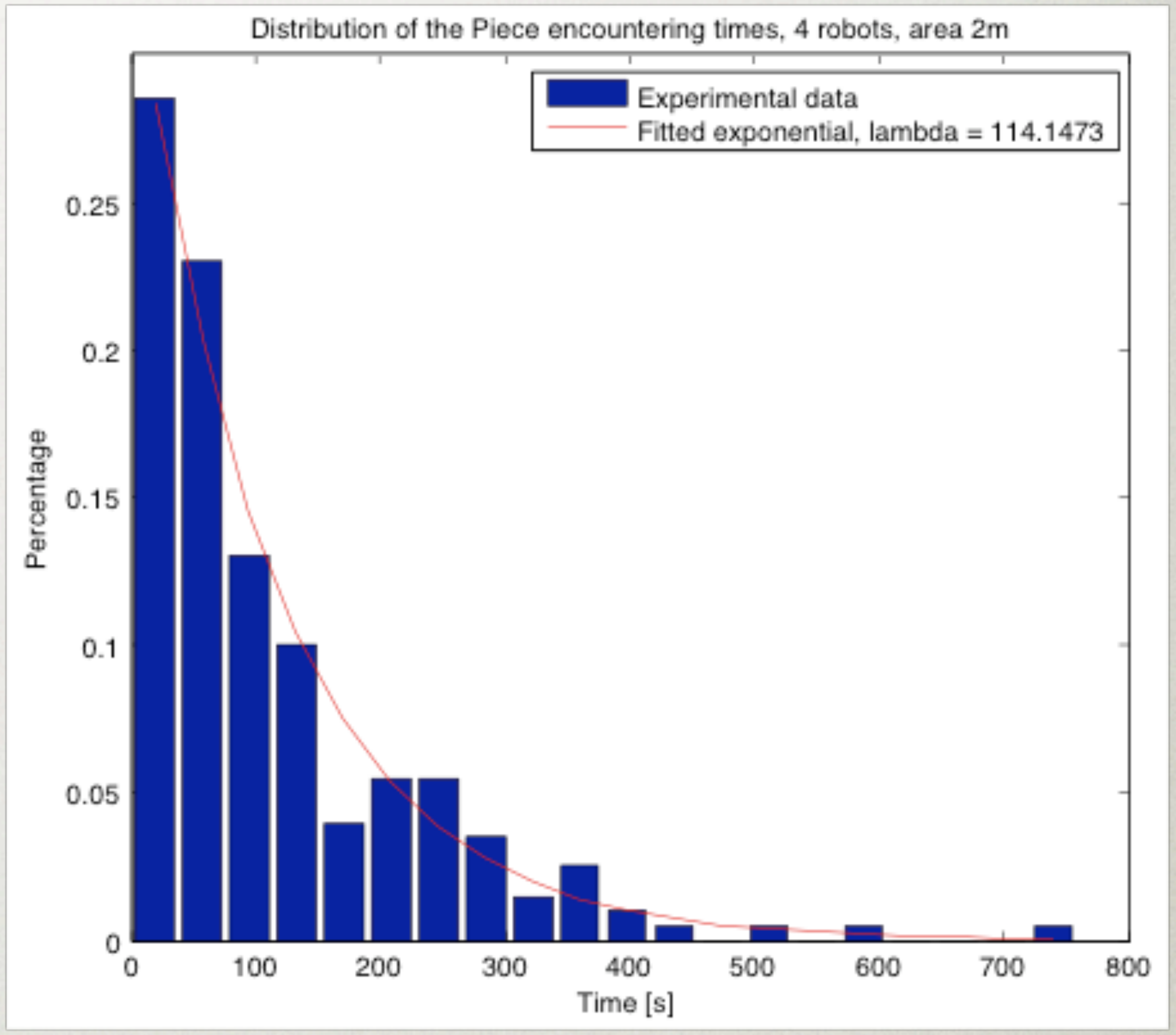
$$p_e \sim \frac{1}{A_{total}} v_r T w_d$$



7. Models

Parameters estimation

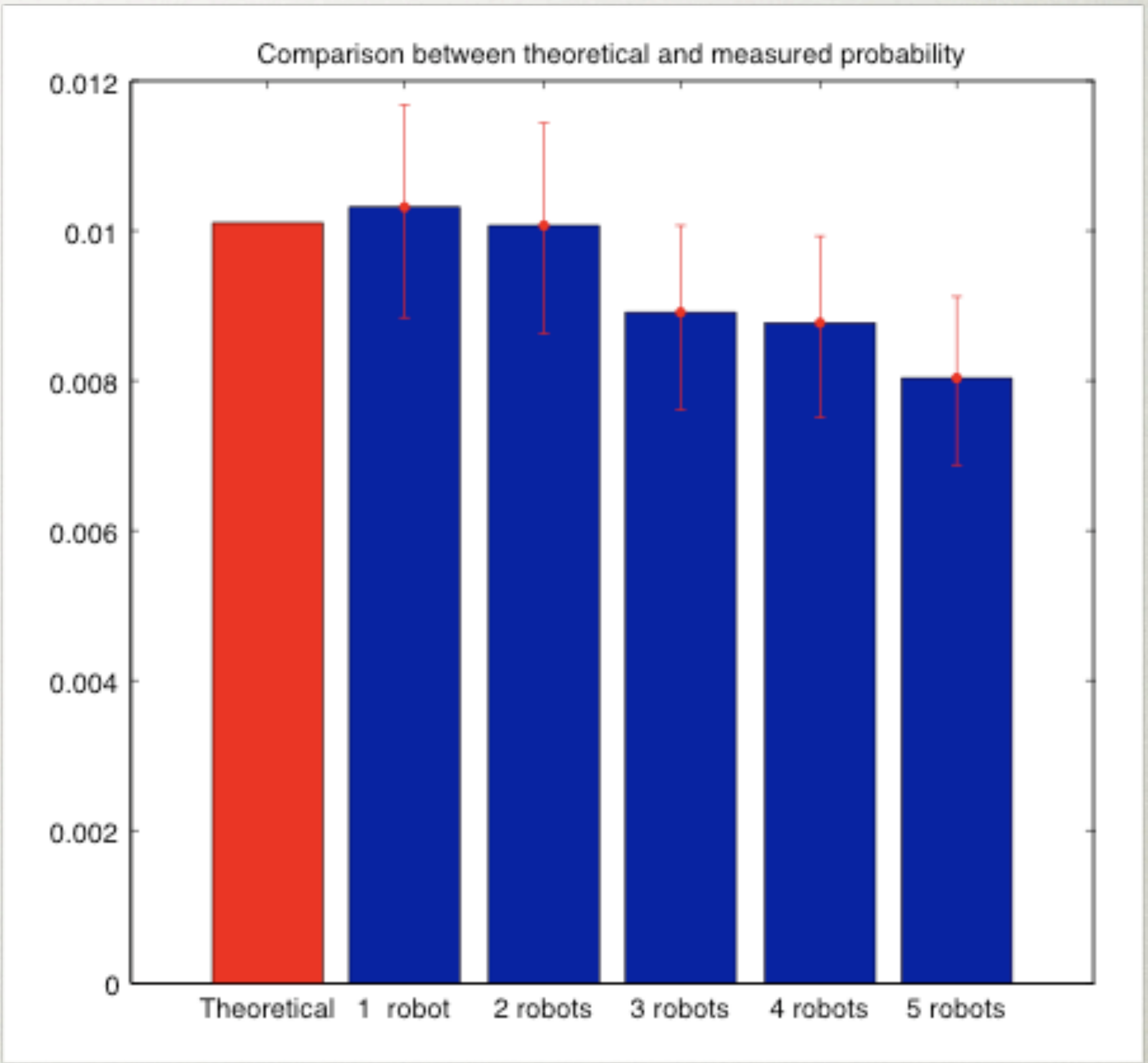
- Rates verifications
- Webots experiments
  - Sample the times to event.
  - 100 experiments.
  - Fit an exponential distribution in Matlab.
- Verify effect of adding "dummy" robots.



7. Models

Parameters estimation

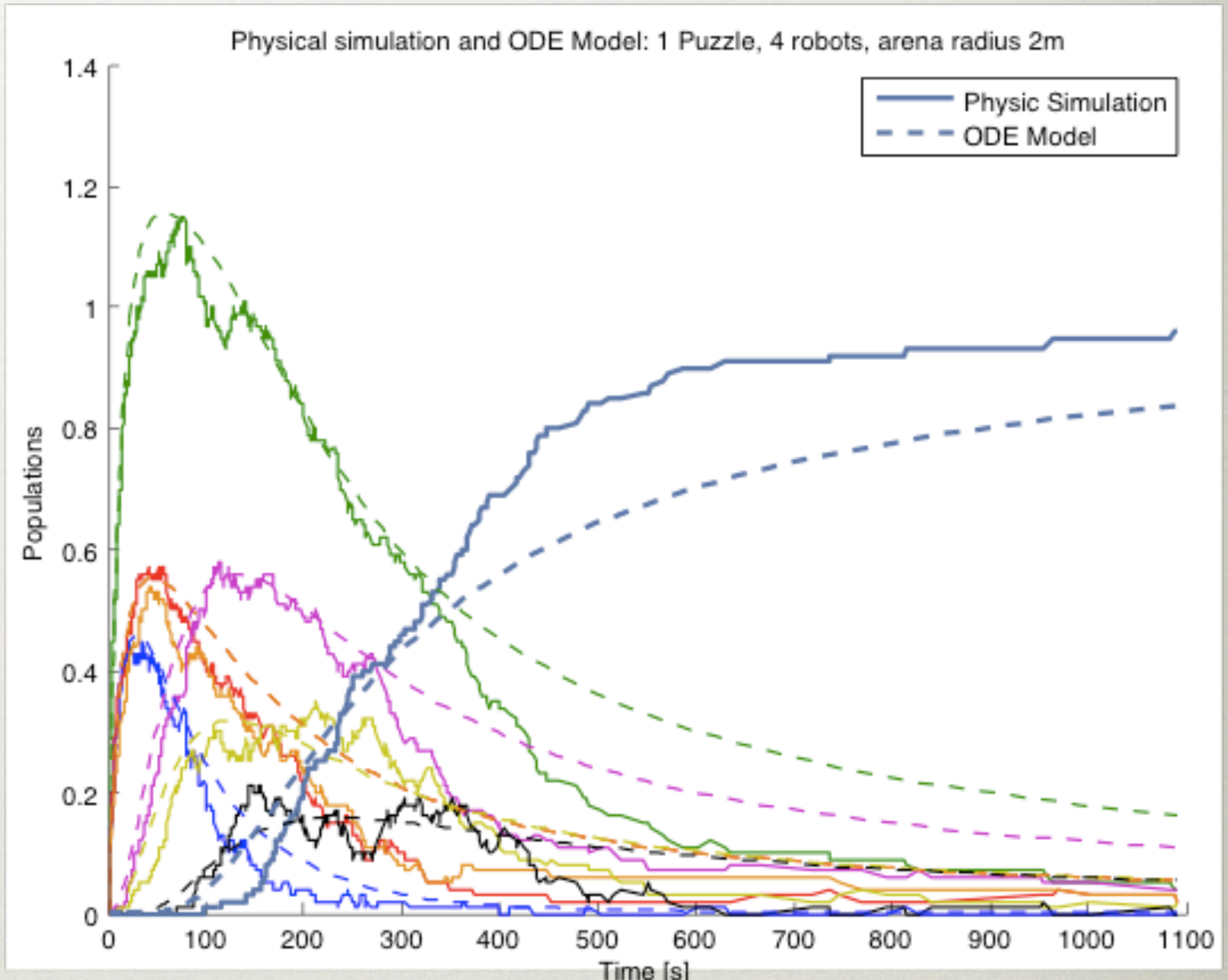
- Rates verifications
- Webots experiments
  - Sample the times to event.
  - 100 experiments.
  - Fit an exponential distribution in Matlab.
- Verify effect of adding "dummy" robots.



7. Models

Results - 1 puzzle

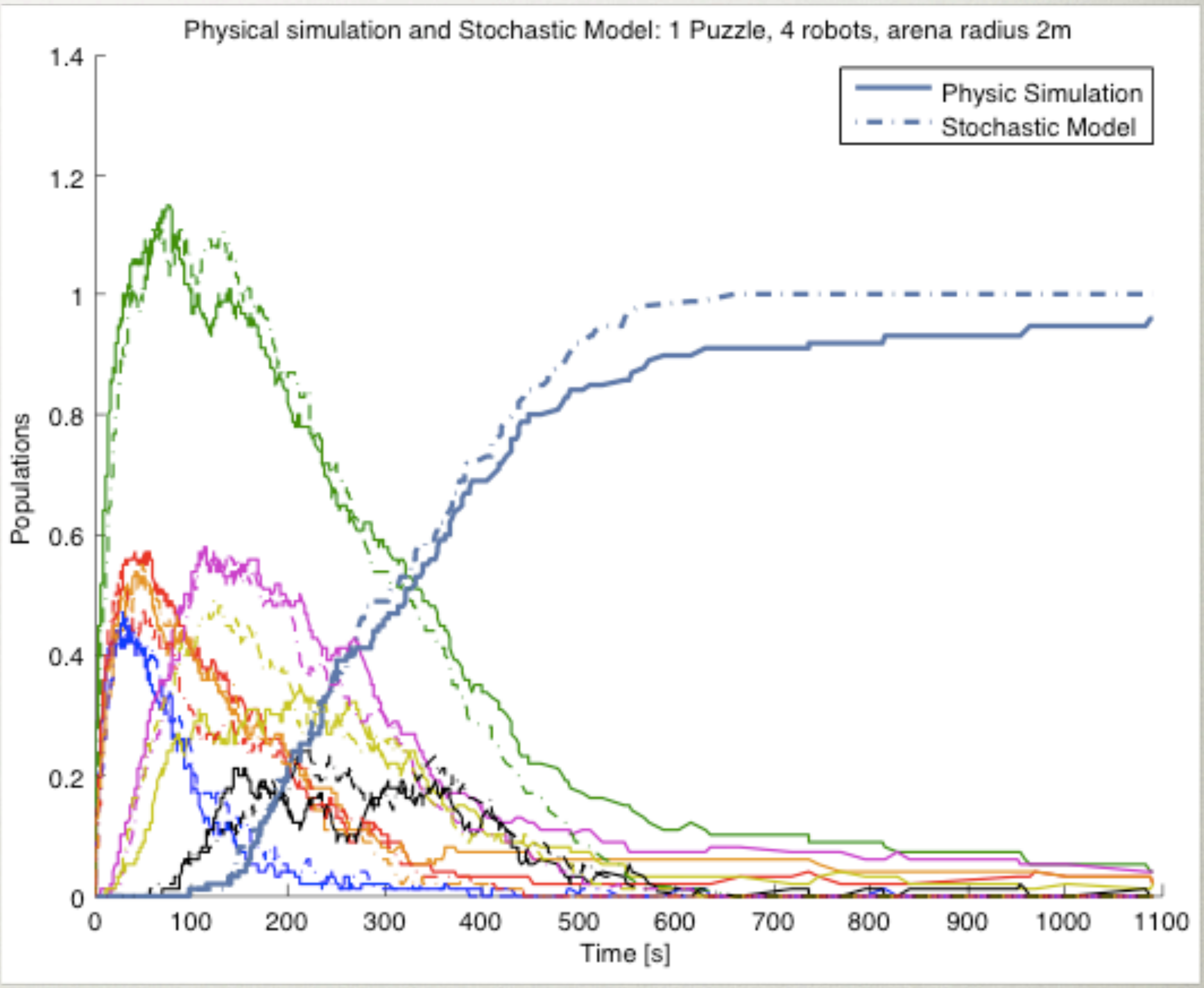
- **1 Puzzle** (5 pieces), 4 robots
- Same rates for both models, from previous theoretical.
- ODE is too low, due to low numbers effects.
- Stochastic is good. Too high because no failures.



7. Models

Results - 1 puzzle

- **1 Puzzle** (5 pieces), 4 robots
- Same rates for both models, from previous theoretical.
- ODE is too low, due to low numbers effects.
- Stochastic is good. Too high because no failures.



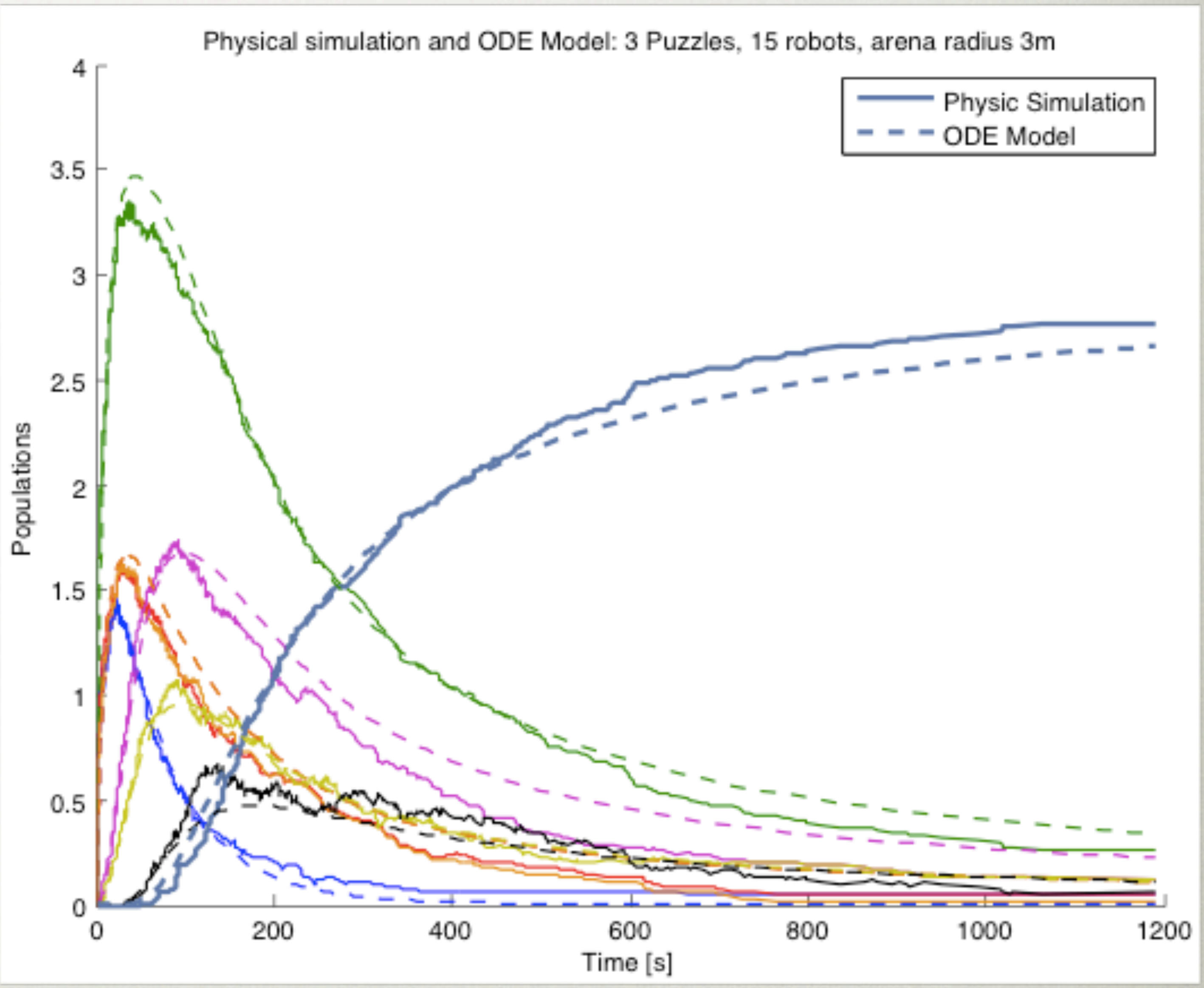
7. Models

Results - 3 puzzles

- **3 Puzzle** (15 pieces), 15 robots
- ODE better but still low. Stochastic still better.

  - **Accurate models.**

- Even with other robots and the possible well-mixed problem.



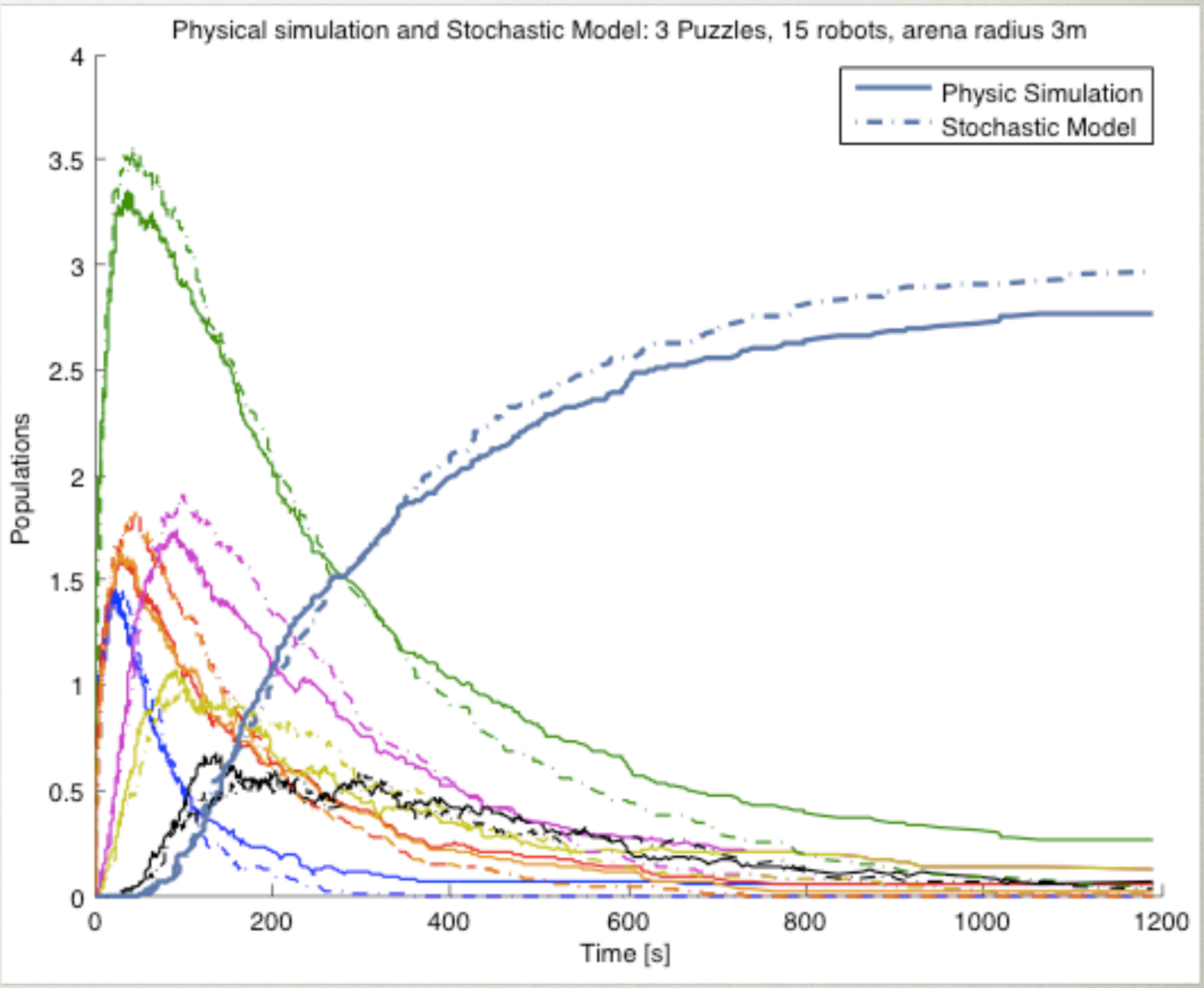
7. Models

Results - 3 puzzles

- **3 Puzzle** (15 pieces), 15 robots
- ODE better but still low. Stochastic still better.

  ▶ **Accurate models.**

- Even with other robots and the possible well-mixed problem.



8. Conclusion

- Created a Webots platform to simulate realistic assemblies.
  - Further tests possible, with more interesting plans (multiple targets).
- Chemical Reactions Networks seems like a good model for this kind of problem.
- Simulations shows that Stochastic is indeed better at representing small populations dynamics.

9. Further work

- Optimize the system !
  - Represent it as a Markov Chain.
  - This is a Multi-affine system that we have to optimize.
- Map the optimized system onto new robot behaviors or added agents.
  - “Augmentation” step.
- Assess the generalizability of this methodology and the possibles applications to other fields.

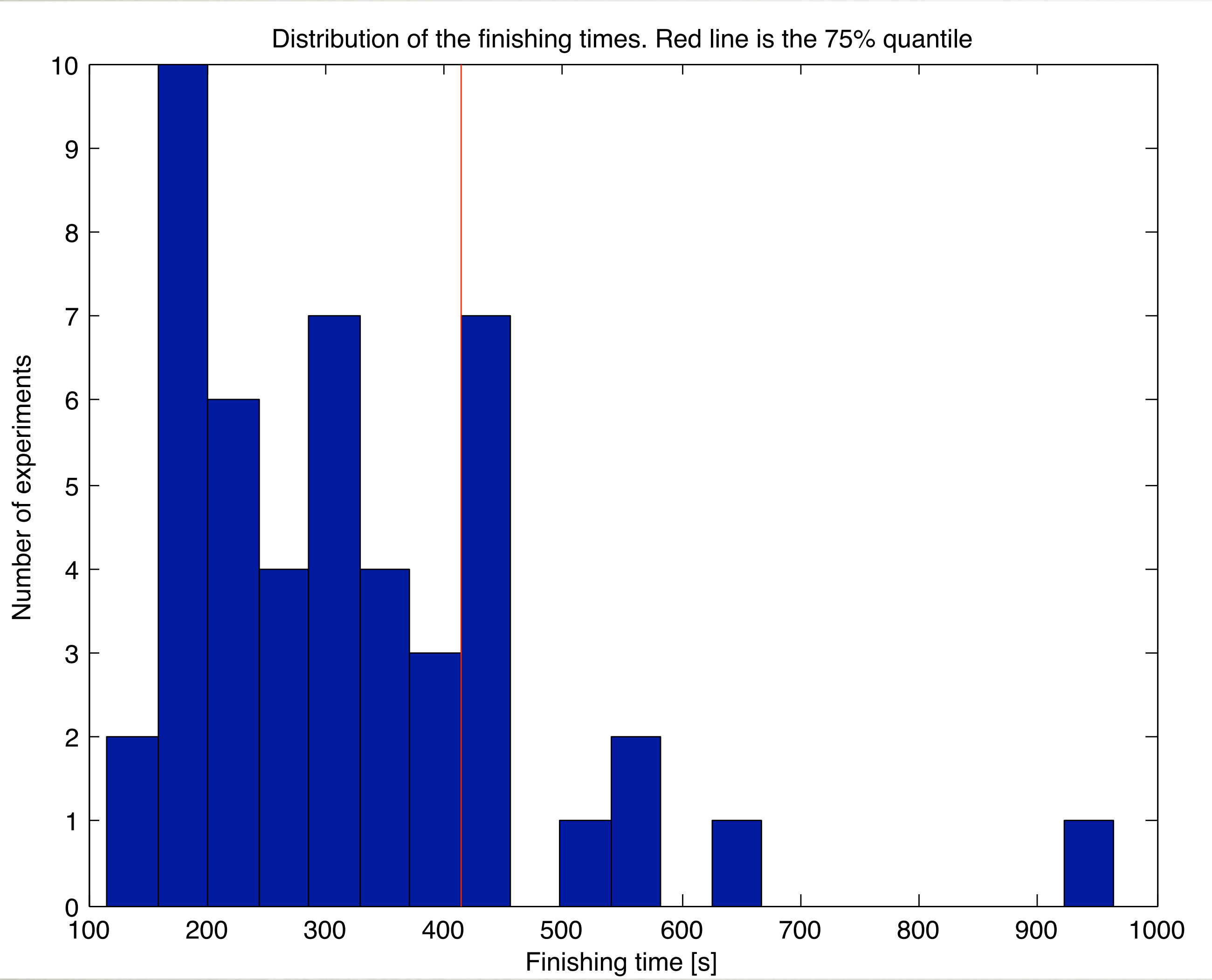
Thank you

Any Questions ?

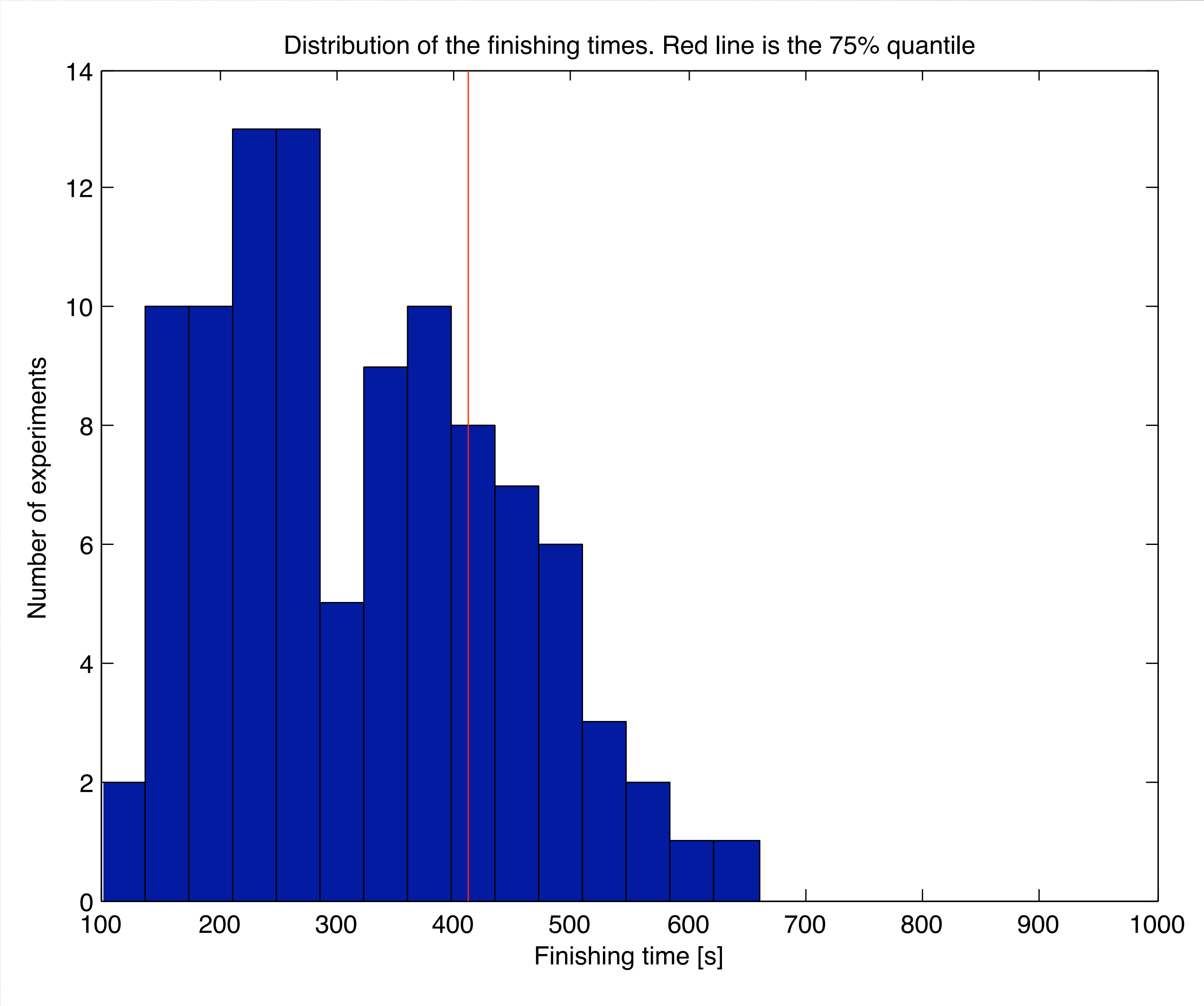
7. Models

Stochastic Model

- Finishing times



Webots



Stochastic model

Optimization

- Chemical Reaction Networks are linear in $\{1, x_1, ..., x_n, x_1x_2, ..., x_nx_m\}$:

$$\dot{x} = E \cdot A \cdot y$$

With x vector of x_i, E is the effector matrix, A the tunable parameters and y a vector of $\{x_i, x_ix_j\}$.

  - Multi-affine system.

- We need to tune A, so that we converge to a desired equilibrium.

- Problem: non-linear.
  - Ideas: calculate several A for small regions (linearization assumption).