Optimization of the Brine Injection Process into Bacon using Supervised Learning and Reinforcement Learning

Alexander B. K. Barlo, Morten Nør*, Rasmus E. Andersen, Sebastian B. Johansen & Steffen Madsen

Department of Materials and Production, Aalborg University
Fibigerstraede 16, DK-9220 Aalborg East, Denmark
Email: mnar12@student.aau.dk,
Web page: http://www.mechman.m-tech.aau.dk/

Abstract

This project is carried out as a part of the MADE Digital research program, focusing on the Danish meat industry. It is motivated by a challenge in the process of injecting brine into bacon. This is done to mature, tenderize, and enhance flavour of the meat. In cooperation with Danish Meat Research Institute (DMRI) and Danish Crown, as the industrial case, the research project seeks to control the process of injecting brine into bacon using adaptive machine learning. The project contains the development of an experimental setup, defining the environment for a Deep Deterministic Policy Gradient Actor-Critic Network (ACN). Using a developed compound of animal-based gelatine and Sodium Polyacrylate, a simulation model of the environment is based on conducted experiments. The model is a four dimensional second order polynomial found by multivariate regression. It consists of two control variables; injection time and injection pressure, and states of mass and volume. The simulation model is used to train the implemented Actor-Critic Agent. It was found that a composition of 14 % 220 Bloom gelatine, 1.2 % Sodium Polyacrylate relative to water, was able to absorb up to 28.9 % water. With a target of 15 % mass increase the composition was a suitable bacon imitator. A vision based numerical integration method was developed, able to calculate volume of a test object with a 7.1 % overestimation. The ACN was capable of adapting a model to a simulated environment with different parameters. It produced a mean of 14.94 % mass increase and standard deviation of 2.59 %. This result was achieved with starting masses drawn from a normal distributions of $\mathcal{N}(80.5\text{ g}, 4.79\text{ g})$.

Keywords: Machine Learning, Reinforcement Learning, Robot Vision, Meat Industry, Brine Injection

1. Introduction

In the meat industry, a brine solution is added to bacon to mature, tenderize, and enhance the flavour of the meat. Brine is a salt-mix solution consisting of phosphate, nitrate, and sugar or flavouring.

Two methods of curing bacon are immersion curing and needle based injection. (i) in immersion curing, the bacon is soaked for a specified time period, and (ii) needle based injection where needles are inserted, and pressurised brine is forced into the bacon [1]. Immersion curing secures the most uniform distribution of the brine. However, this is a more time consuming process. Therefore the needle based injection method is preferred despite the distribution of the brine being less uniform. Traditional needle based injection is displayed on Fig. 1.

The needle based injection process is difficult to control due to high product variation. If too much brine is injected, there is risk of washing out the proteins as a result of a too high salt concentration [2]. To avoid this, less brine is injected as it is not possible to control the distribution of the brine, when using current needle based injection technology.

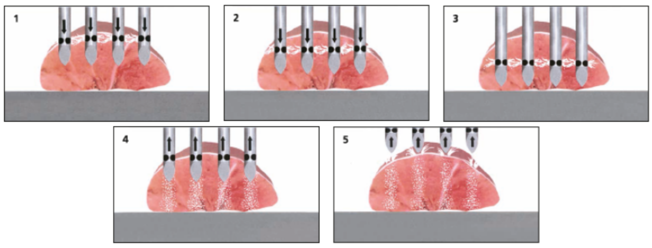

Fig. 1 (1) The needles are forced into the meat, and the brine is injected. (2) As the needles travel down into the meat brine is injected continously. (3) The needles reach the bottom, and reverse the direction of travel. (4) The brine is collected in the holes, as the needles are pulled up. (5) When the needles leave the meat, an amount of brine will escape from the meat.

Due to the variation in meat, an adaptable model is needed. One domain of adaptable models is reinforcement learning, which has shown a great potential within technical control problems [3].

The research carried out in this paper is based on similar research from the Danish Meat Research Institute (DMRI) and the MADE Digital project. This leads to a natural similarity between some of the research objectives, as well as in the design of an experimental setup.

2. Research Objectives

The following research objectives were identified, and will be addressed in the following sections.

1) Develop an experimental setup that can measure states and take actions in the brine injection process.
2) Develop a compound that can imitate bacon.
3) Develop a vision system that is able to determine the volume of the test objects.
4) Implement an Actor-Critic Network algorithm and develop a simulation environment.
5) Integrate subsystems in ROS.

This research does not strive towards an optimization of the throughput time.

3. Development of Experimental Setup

In order to train on the reinforcement learning algorithm, a physical experimental setup is needed. The setup presented in the following, is heavily inspired by an experimental setup at the DMRI facilities in Taastrup, Denmark, where similar research is conducted.
The setup consists of a combination of stock components and custom designed components. A 3D CAD rendering of the experimental setup at Aalborg University can be seen on Fig. 2.

The setup is able to control the desired actions along with detecting the relevant states of the test objects. When doing brine injection, two parameters are used to control the process, injection pressure and injection time. These two parameters are combined as a set, and are classified as the action a reinforcement learning (RL) algorithm will take. To control these actions, a fluid system is developed.
The RL algorithm also needs some information about the current state of the test objects, these are defined by a set of volume and mass measurements of the specific test object. In order to obtain the current states of the

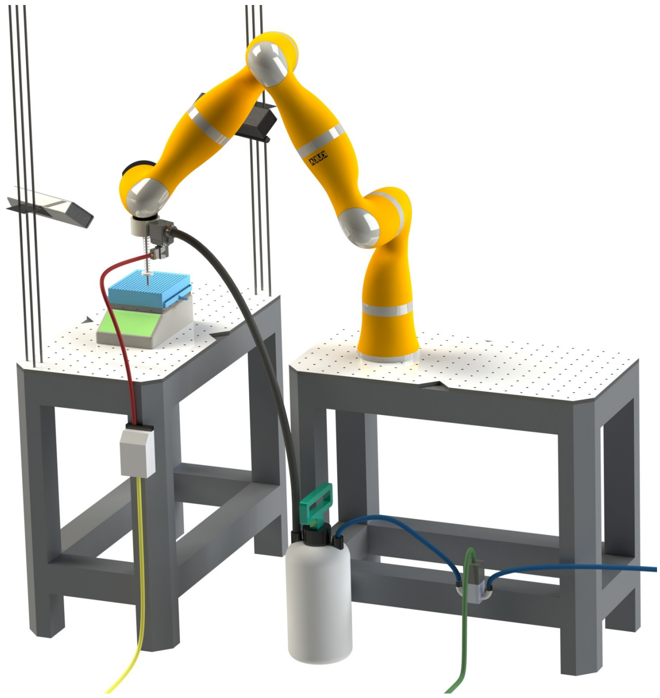

Fig. 2 3D CAD rendering of the experimental setup at Aalborg University

test objects, a high-precision scale and multiple cameras are included for mass and volume detection respectively. Tab. I displays the different components divided into categories.

Category	Component (Manufacturer, Comment)
Fluid System	Injection Needle (Charbroil, marinade needle) Injection Needle Fixture (Custom Design) Pneumatic Valve (MetalWorks "Regtronic") Magnet Valve (RS, servo-assisted) Relay (Danbit) KUKA LWR 4 (KUKA, needle manipulator)
Vision	ASUS Xtion Pro (ASUS) Kinect v2 (Microsoft)
Mass detection	KERN KB 10K0.5N (KERN, 0.05 g precision) Drain Plate (Custom Design)

Tab. I Components used for experimental setup at Aalborg University.

4. Development of Bacon Imitation Compound

Due to the relatively large amount of testing needed to train an algorithm, it has been necessary to find an alternative to the bacon, used at DMRI. For this, inspiration have been taken from forensic ballistic science. In this branch of science, a gelatine solution, mixed from animal-based gelatine, is often used to imitate ballistic impact with humans, which is equivalet to pork and its tissue [4]. The bacon processed in the industrial case is dead meat. It is anticipated that there

is a difference between living and dead tissue, however, it shows that imitating dead tissue is not a well explored field of study. Therefore, gelatine, as used to simulate ballistic impact, will also be applied in this project.
Furthermore, the application of gelatine based test objects, allows for the use of smaller test objects, in this case cubes measuring 50x50x50 mm. Equipment allows for the making of 32 test objects per batch. These test objects will however be single injection objects, and not multi-injection as done in the industrial case. Initial experiments showed, that a compound consisting of only gelatine does not meet the demands of the industrial case, why an additive to increase the water absorption is introduced. This additive is Sodium Polyacrylate, categorised as a Superabsorbant Polymers (SAP).

4.1 Test Object Preparation

A series of experiments showed that the way the objects are mixed and prepared is important in order to obtain test objects with uniform properties.
For the first step of the mixture, water, gelatine powder, a power drill (with attached whisk), and a 2 L Polypropylene (PP) beaker is needed. The water is stirred. and gelatine powder is slowly poured into the water in order to avoid lumps. Afterwards the solution is stirred for approximately 30 s, reaching a homogeneous state. SAP powder is slowly added to the mixture during continuous stirring. This turns the mixture into semi-solid state. Compartments in a mould are filled, while gentle stamping of the contents of the compartments occurs. The stamping process has through trials shown to improve the test objects mechanical properties.

All test objects are stored at 5 °C for 24-48 hours before use. During storage, the mould is tightly wrapped in a plastic bag in order to minimize condensate absorbed by the objects.

4.2 Experimental Procedures and Results

32 different compounds are tested in order to determine a suitable composition for the target of 15 % mass increase post injection.

The object is weighed prior to the injection, using the high precision scale. The needle is then inserted so that the outlet nozzles are approximately in the center of the test object. The injection is carried out with a water pressure of two bars and a duration of approximately one and a half second. The object is once again weighted immediately after injection.

The results of the test indicated, that a composition consisting of 14 % gelatine powder and 1.2 % SAP powder, relative to water, is able to absorb up to a mass increase of 28.9 %. Thus a suitable compound for brine injection experiments has been found, as both under- and over-absorption is possible. An issue regarding increasing batch sizes appeared, resulting in the water absorption of the final objects being reduced by up to 50 %.

5. Development of Vision System

Being able to detect changes in volume is a desirable ability in order to monitor the distribution of the brine in the meat. Change in mass is measured by the high-precision scale, and change in volume is detected by point clouds generated by two 3D cameras.

5.1 Camera Calibration

Two 3D cameras are included in the setup, and these needs to be calibrated in order to obtain useful data. One important aspect when applying adaptive learning to a process, is to obtain good and consistent data. Two types of camera calibrations are performed, (i) intrinsic calibration, and (ii) extrinsic calibration.

The intrinsic calibration corrects the focal length variation and lens imperfections that naturally exists in every camera. Through the intrinsic calibration, parameters such as an accurate focal length and distortion parameters are found [5]. As intrinsic calibration is a standard step in a computer vision, a variety of ROS toolboxes are available. For the purpose of the cameras used in this setup, the package camera_calibration is used for the ASUS Xtion Pro camera [6] and kinect2_calibration for the Microsoft Kinect v2 camera [7].

The extrinsic calibration is done in order to find the transformation from the robot base frame to a 3D camera. It is carried out by applying the calibration driver calib_handeye_kinect2 [8] [9]. The robot base frame and the camera frame perceive the world coordinates differently as displayed on Fig. 3.

In order to obtain usable point clouds, the robot base frame is used as a common reference. Therefore, all points in the point clouds undergoes the transformation displayed in Eq. 1.

$$^{B}P_i = ^{B}_{Cam}T \cdot ^{Cam}P_i \tag{1}$$

Even when having performed the two calibrations, an error between the two point clouds occur. To minimise this, an Iterative Closest Point (ICP) optimisation

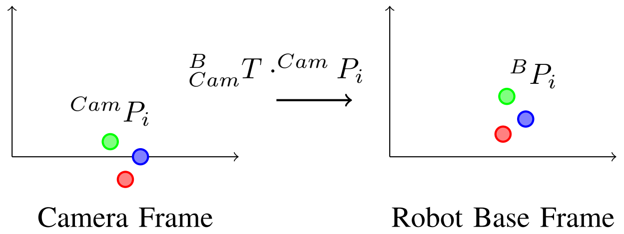

Fig. 3 Point in the camera frame Cam_i^P is transformed to the point in the robot base frame ${}^{B}P_i$ using the wanted transformation ${}^{B}_{Cam}T$. Note that this is a general representation using three points.

algorithm is applied. After performing ICP, there was still an error when capturing point clouds from test objects, likely due to the optical properties of them. As the Kinect v2 showed better results, it was decided to only uses this for the volume calculations.

5.2 Volume Calculation Based on Point Cloud Data

The approach selected for volume calculation is a combination of numerical integration and binning. Having divided all the points into different bins, the volume can be calculated using Eq. 2.

$$V = \sum_{i,j=1}^{n} zBinAvg(i,j) \cdot interval_x \cdot interval_y \quad (2)$$

Here, the $zBinAvg$ is the average z-value of the points in each bin, taking the offset from the origin of the coordinate system to the bottom of the object into account, and $interval_i$ is the side lengths of bins.

As binning is a type of discretisation of the object, the method is not completely accurate. One thing to consider is the number/size of the bins the points are divided into. If the bin size is too small, a situation where empty bins are present can occur. The correlation between bin size and the calculated volume can be seen in Fig. 4.

To secure robustness in the method, a constant number of bins chosen to be 60. As can be seen on Fig. 4, this is a compromise between the accuracy of the calculated volume and the number of empty bins. In a finalized solution, the bin size could potentially be set to a fixed distance. Having 60 bins per axis this results in a bin size of 0.45 mm.

Having specified the different values, the average height of the test object above the drain plate can be seen in Fig. 5. This enables the possibility to detect the volume before and after the injection thereby getting the volume increase for the test object.

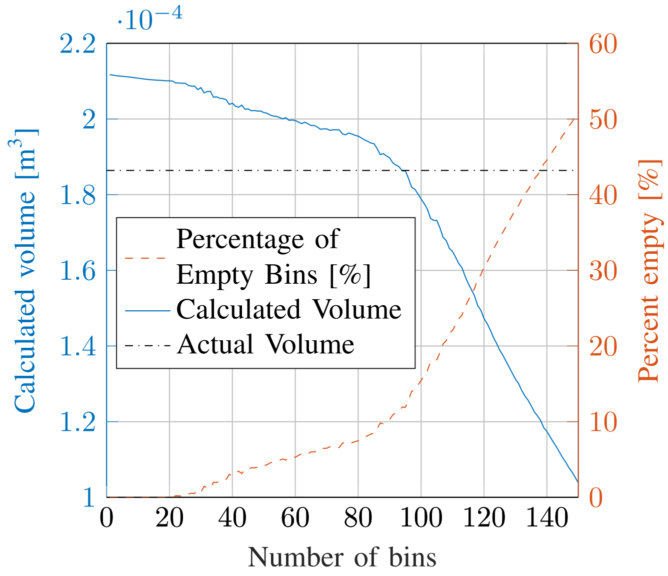

Fig. 4 Volume as a function of bins.

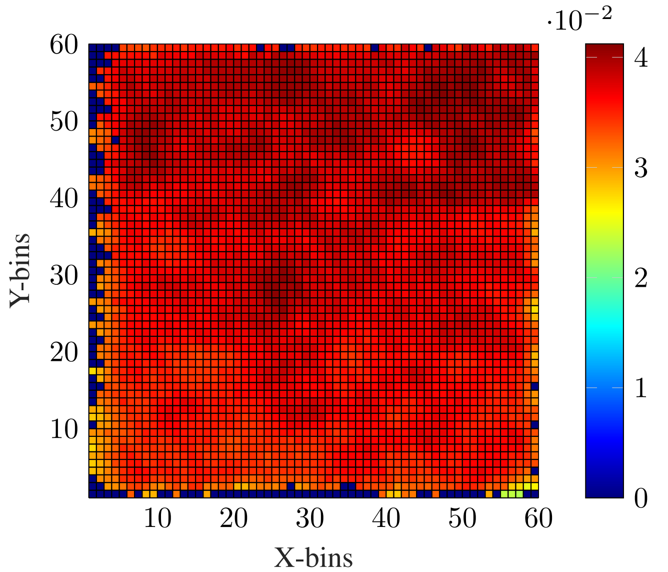

Fig. 5 Average z-height from drain plate in each bin measured in meters.

While the Kinect v2 was able to obtain a good point cloud before injection, it was unable to capture a full point cloud of the test object after injection due to the changes in optical properties caused by the injected water. Therefore, volume increase calculations could not be performed. In the industrial case, where the bacon is travelling along a conveyor belt, it is expected that the use of cameras will slow down the process, why an alternative could be to utilize a laser scanner.

6. Implementation of Actor-Critic Network Algorithm

One of the issues with traditional supervised machine learning is, that once training data is collected, the model will no longer adapt to new inputs. This means, that if the state space drifts over time, the training data can become invalid.

One way to get around this issue, is to apply reinforcement learning (RL). Here an agent acts in an environment and is given a reward based on its performance, as displayed on Fig. 6.

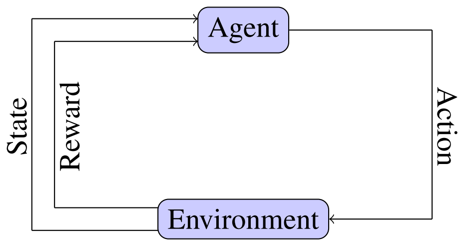

Fig. 6 Basic principle of reinforcement learning, where an agent acts in an environment and receives rewards based on its performance.

6.1 Inverted Pendulum Problem

Three different methods have been investigated; (i) Q-Learning method, (ii) Deep Q Network (DQN) method, and (iii) Deep Deterministic Policy Gradient (DDPG) Actor-Critic Network (ACN) with a Deep Deterministic Policy Gradient (DDPG).

The state and action spaces are both continuous, where the state space consists of mass and volume of the object, and the action space consists of the injection pressure and -time. From the state and action space, the Q-Learning method is disregarded, since this method is known to have poor performance, when the numbers of actions or states increases, based on the principle of 'curse of dimensionality'.

To determine which method of the two remaining (DQN and ACN) have the best performance, these are tested applying the inverted pendulum control problem. For this problem, the pendulum environment have the state space consisting of the normalized x- and y-coordinates of the pendulum (coordinates defined as $[\cos(\phi), \sin(\phi)]$) as well as the angular velocity of the pendulum ($\dot{\phi}$). The action space is defined by a torque applied at the pivot point of the pendulum in the range of $[-2, 2]$ Nm. The reward used as feedback to the algorithm is defined in Eq. 3 [10].

$$r = -(\phi^2 + 0.1\,\dot{\phi}^2) \tag{3}$$

The results of the two simulations can be found on

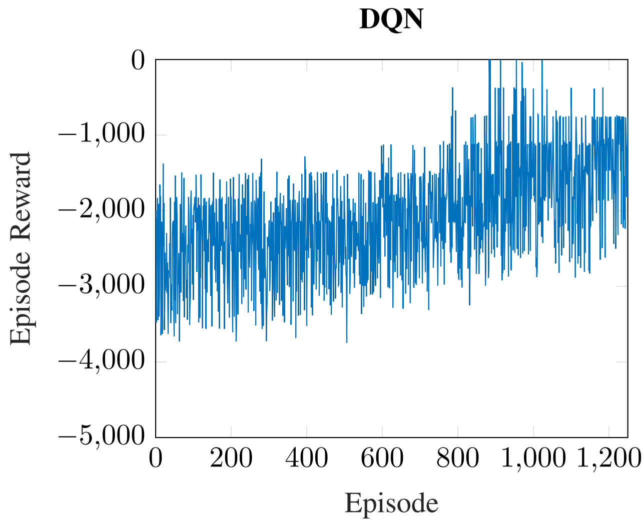

Fig. 7 Result of the DQN method applied to the inverted pendulum method.

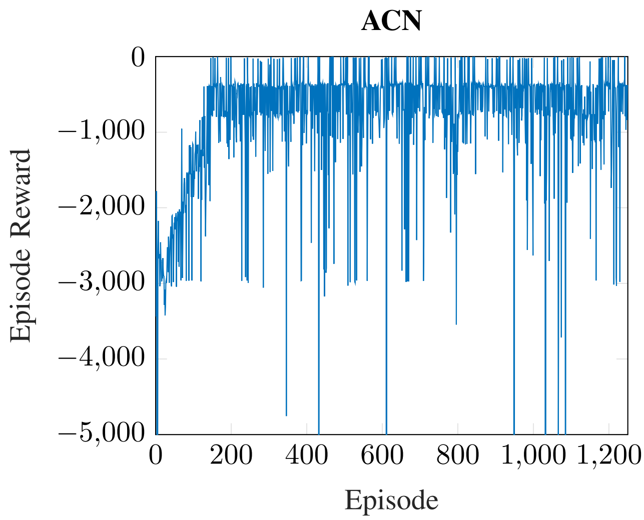

Fig. 8 Result of the ACN method applied to the inverted pendulum method.

Fig. 7 and 8. From the results, it can be seen that the ACN method outperforms the DQN method. Based on the presented results, the ACN method is chosen.

6.2 Implementation the ACN Algorithm

Since the ACN method is based on the concept of Neural Networks (NN), the number of hidden layers and activation function is of great importance, these are known as hyperparameters. A basic principle of a NN can be found on Fig. 9, and the parameters used for the NN in the algorithm can be found in Tab. II.

The ACN algorithm will be implemented on the industrial case using the same structure as used for the inverted pendulum problem, but with different

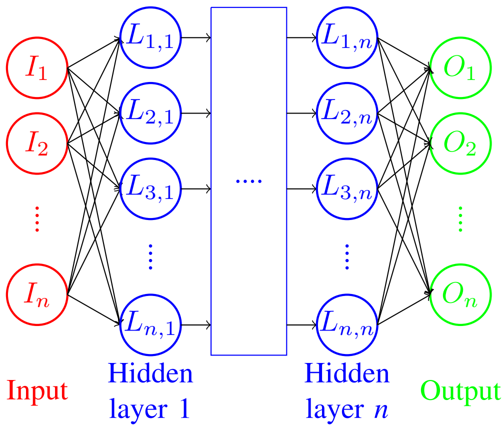

Fig. 9 General setup of a Neural Network, consisting of nodes (blue circles) connected through weights (lines between circles) as well as an input layer with observations of the state and an output layer estimating the value of each possible actions.

Exploration		Ornstein-Uhlenbeck
Activation Function		ReLU
Actor	Number of layers	3
	Nodes in layer	128
	Activation function	max(0,x)
Critic	Number of layers	3
	Nodes in layer	256
	Activation function	max(0,x)

Tab. II Parameters applied to the NN for the actor and the critic.

definitions of the state and action space, and reward function. The reward function is shown in Eq. 4. It has a target mass increase of 15 % as well as two terms penalising actions the further away from 1.5 bar and 1 s they are.

$$r = -\left((p-15)^2 + 0.1\left(a_1 - \frac{3}{2}\right)^2 + 0.1(a_2-1)^2\right) \tag{4}$$

The implemented Actor-Critic Model can be seen in Algorithm 1. For simulation purposes, the environment the agent traverses, has been modelled from experiments. The environment is a four dimensional second order polynomial function that returns the mass increase as a function of injection pressure (x_1), injection time (x_2), and starting mass (x_3). The environment is obtained through application of multivariate polynomial regression and can be found in Eq. 5.

$$\begin{aligned} f(x_1, x_2, x_3) = & c_1x_1^2 + c_2x_2^2 + c_3x_3^2 + c_4x_1x_2 + \\ & c_5x_1x_3 + c_6x_2x_3 + c_7x1 + c_8x_2 \\ & + c_9x_3 + c_{10} \end{aligned} \tag{5}$$

Algorithm 1: DDPG for Brine Injection [11].

Initialize critic $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ
Initialize target critic Q' and target actor μ' with weights $\theta^{Q'} \leftarrow \theta^Q$ and $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for $\underline{episode = 1, M}$ **do**
  Get action $a_e = \mu(s|\theta^Q) + \mathcal{N}$
  Execute action a_e in environment, observe r_e and s_{e+1}
  Store transition (s_e, a_e, r_e, s_{e+1}) in R
  Sample N random transitions, (s_i, a_i, r_i, s_{i+1}), from R
  Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
  Update the critic by minimizing the loss:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - Q(s_i, a_i|\theta^Q)\right)^2$$

  Update the actor policy using the sampled policy gradient:

$$\nabla J \approx \frac{1}{N}\sum_{i=1}^{N} \nabla Q(s_i, \mu(s_i)|\theta^Q)\nabla\mu(s_i|\theta^\mu)$$

  Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\,\theta^{Q'}$$
$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\,\theta^{\mu'}$$

end

Since some solutions exist where a mass increase is non-zero when the injection time is 0 s, the polynomial only partly represent the actual environment.

Running the agent in the simulation environment, yields the results seen in Fig. 10 with reward history on Fig. 11 and mass increases on Fig. 12. A mass increase with a mean of 14.94 % and standard deviation of 2.59 % was obtained, using starting masses drawn from a normal distribution of $\mathcal{N}(80.5 \text{ g}, 4.79 \text{ g})$. The agent was run a second time using starting masses drawn from a normal distribution of $\mathcal{N}(80.5 \text{ g}, 20 \text{ g})$, showing the same standard deviation and a mean of 14.64 g, indicating that the variance in starting mass does not have an impact on the agents ability to adapt a model to the environment.

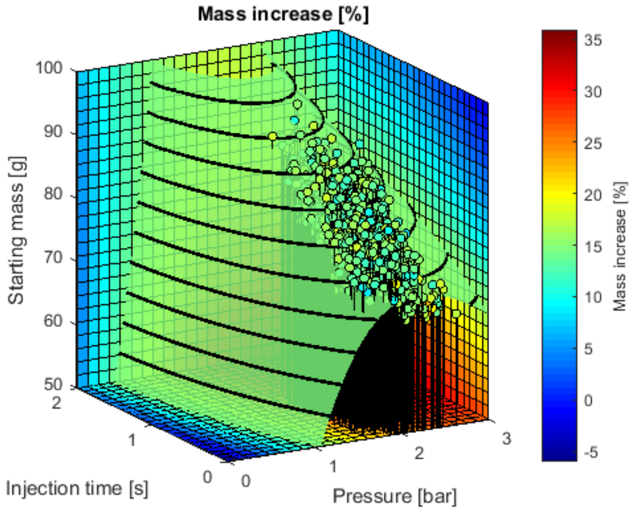

Fig. 10 Simulation results for brine injection using the last 1000 episodes.

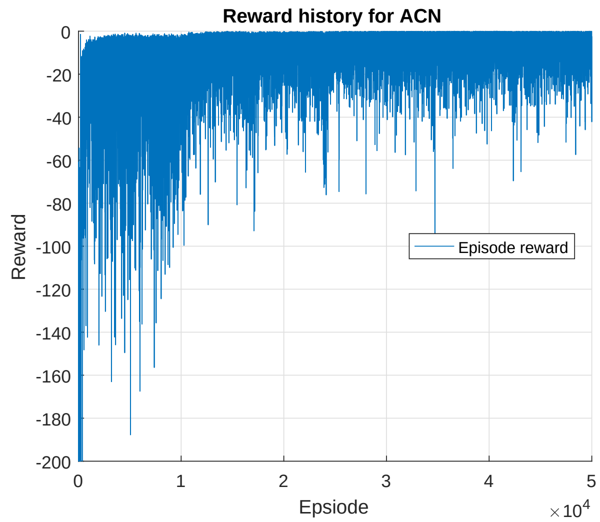

Fig. 11 The reward for each episode.

7. Subsystem Integration in ROS

To allow all of the different subsystems in this project to work together on a shared platform, the open source framework called Robot Operating System (ROS), is utilized. Through a number of ROS topics/services, developed drivers communicate indirectly through an interface handling all reinforcement learning, called *Environment and Reinforcement Learning Interface*. Tab. III displays the different topics and services.

8. Conclusion

One of the challenges in brine injection is large variety, thus requiring an adaptive model to control. A Deep Q-Network and an Actor-Critic Network were both tested on the inverted pendulum environment, where the Actor-Critic Network was found superior. The Actor-Critic Network was capable of adapting a model to an environment having continuous state and action space.

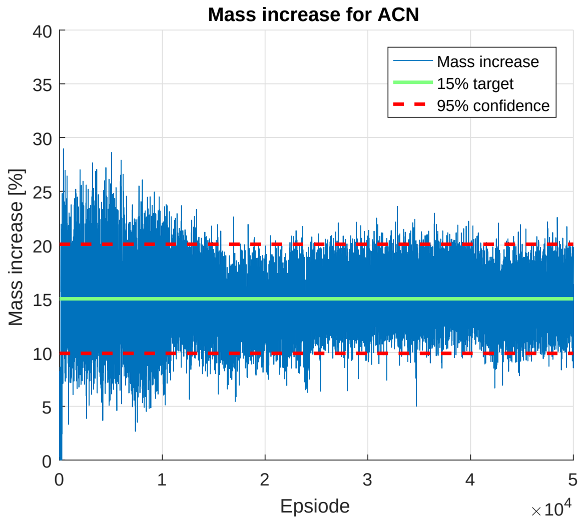

Fig. 12 Mass increase for each episode with a 95 % confidence interval.

Driver	ROS Topic/Service
Modbus Driver	*/modbus_client/write_coils* *"/modbus_client/read_coils"*
Scale Driver	*/serial/read_weight*
Pressure Regulator Driver	*/pressure/measured*
Volume Calculation Driver	*/volume/get_bacon_cloud*

Tab. III Key ROS topics/services for different drivers.

An experimental setup was developed using hardware measuring states defined as mass and volume. Injection pressure and injection time was furthermore possible to control. Since it was not realistic to use bacon, a compound of SAP, gelatine and water was used. The compound was able to absorb water equal to a 28.9 % mass increase. The developed vision system was capable of measuring initial volume of the test objects. However, it was not possible to test the test objects for uniform mass increase, as the vision system had issues adequately detecting the test objects after injection due to the resulting change in optical properties.

The Actor-Critic Model was capable of adapting a model to a simulated environment with different parameters. It produced a mean of 14.94 % mass increase and standard deviation of 2.59 %. This result was achieved with starting masses drawn from a normal distributions of $\mathcal{N}(80.5\text{ g}, 4.79\text{ g})$.

In combination, this means that an Actor-Critic Network was applied to the process control of brine injection in the simulation environment, and a mass increase was achieved. However, the distribution of the mass increase was not measured.

9. Future Work

The first research object concerned the investigation of whether a bacon imitation could be found. The presented solution is only carried out with small test objects. This was sufficient for the purpose of this project, however, objects of the bacon imitation should be developed to fit the size and shape of the actual bacon one to one.

The second research objective concerned the experimental setup. For future work, it should be investigated how a potential replacement for the KUKA LWR 4 robot could be designed. The application of a highly sophisticated robot with 7 degrees of freedom as a simple needle manipulator does not seem like a feasible solution.

The third research objective concerned the development of a vision system for volume determination. The first improvement point is to find an ideal positioning and angle, in order to obtain point cloud data without missing sections. The performance of the volume calculation needs to be verified on test objects both before and after injection, and compared to the data from the scale.

The fourth research objective concerned the implementation of an Actor-Critic Network. The simulation is based on an environment created through experimental work. Improving the model for this environment, especially in the low injection time area, will help increase the accuracy of the model. A way to implement this improvement of the environment would be to include more experimental data in the model. This could also potentially allow the usage of a polynomial of higher order.
One point where the model for the environment suffered, was due to high variance in the initial mass of the test objects. This calls for a tuning of the parameters, where e.g. other explorations methods could be tested.

Finally, the subsystem integration should be tested in the real experimental setup in order to verify that the ACN is able to adapt a model on physical test objects as well.

Acknowledgement
The authors of this work gratefully acknowledge Grundfos for sponsoring the 6th MechMan symposium. Additionally we would like to thank Mark Philip Phillipsen for guidance and knowledge sharing.

References

[1] C. Vestergaard, Salt Diffusion and Salt Distribution in Meat Curing. PhD thesis, 2004.
[2] M. Philipsen, R. Andersen, O. Madsen, and T. Moeslund, Adaptive and self-learning slaughter robots, p. 16. 11 2017.
[3] R. Hafner and M. Riedmiller, “Reinforcement learning in feedback control,” vol. 84, pp. 137–169, 2011.
[4] J. Jussila, “Wound ballistic simulation: Assessment of the legitimacy of law enforcement firearms ammunition by means of wound ballistic simulation,” Forensic Science International, vol. 1, no. 1, 2005.
[5] P. Corke, Robotics, Vision and Control - Fundamental Algorithms in MATLABÂ®, vol. 73 of Springer Tracts in Advanced Robotics. Springer, 2011.
[6] J. Bowman and P. Mihelich, “camera_calibration,” 2018.
[7] T. Wiedemeyer, “IAI Kinect2.” https://github.com/code-iai/iai_kinect2, 2014 – 2015. Accessed June 12, 2015.
[8] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” Proc. SPIE, vol. 1611, pp. 1611–1611–21, 1992.
[9] R. Skovgaard, “calib_handye_kinect2,” 2018.
[10] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “Openai gym,” 2016.
[11] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” CoRR, vol. abs/1509.02971, 2015.