AALBORG UNIVERSITY

DENMARK

Self-learning Processes in Smart Factories

Deep Reinforcement Learning for Process Control of Robot Brine Injection

Andersen, Rasmus Eckholdt ; Madsen, Steffen; Barlo, Alexander Bendix Krukow; Blegebrønd Johansen, Sebastian; Nør, Morten; Andersen, Rasmus Skovgaard; Bøgh, Simon

[Link to publication from Aalborg University](#)

Self-learning Processes in Smart Factories: Deep Reinforcement Learning for Process Control of Robot Brine Injection

Rasmus E. Andersen[a], Steffen Madsen[a], Alexander B. K. Barlo[a], Sebastian B. Johansen[a], Morten Nør[a], Rasmus S. Andersen[a,b], Simon Bøgh[a,b,*]

[a]*Dept. of Materials and Production, Aalborg University, Fibigerstræde 16, Aalborg Øst, DK-9220, Denmark*
[b]*Robotics & Automation Group, Dept. of Materials and Production, Aalborg University, Fibigerstræde 16, Aalborg Øst, DK-9220, Denmark*

Abstract

The goal of this paper is to investigate the application of adaptive learning algorithms, which enables industrial robots to cope with natural variations exhibited in a brine injection process related to the production of bacon. Due to the variations in bacon meat, the traditional needle-based brine injection process is not capable of injecting the correct amount of brine, leading to either ruined or unflavored bacon. In the presented work a Deep Deterministic Policy Gradient (DDPG) reinforcement learning algorithm is introduced in the injection process to improve process control. To accelerate training of the reinforcement learning algorithm, a simulation environment of the brine absorption is generated based on 64 conducted experiments. The simulation environment estimates the amount of absorbed brine given injection pressure and injection time. Tests are run in the simulation where the starting mass is generated from a normal distribution with mean 80.5g, and a standard deviation of 4.8 g and 20.0 g respectively. With a target of 15 % mass increase, the agent can produce an average mass increase of 14.9 % for the first test and 14.6 % for the second test. This indicates that the model can successfully adapt to a high variety input, thereby showing potential for process control in brine injection, coping with natural variation in meat structure.

Keywords: Self-learning Smart Factories; Deep Reinforcement Learning; Process Control.

1. Introduction

In the meat industry, a brine mixture is added to bacon to mature, tenderize, and enhance the flavour of the meat. Brine is a salt-mix mixture consisting of phosphate, nitrate, and sugar or flavouring. Two methods of curing bacon are 1) immersion curing and 2) needle-based injection. In immersion curing, the bacon is soaked for a specific time period, whereas the needle-based method relies on needles inserted in the bacon, which by pressure forces the brine 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.
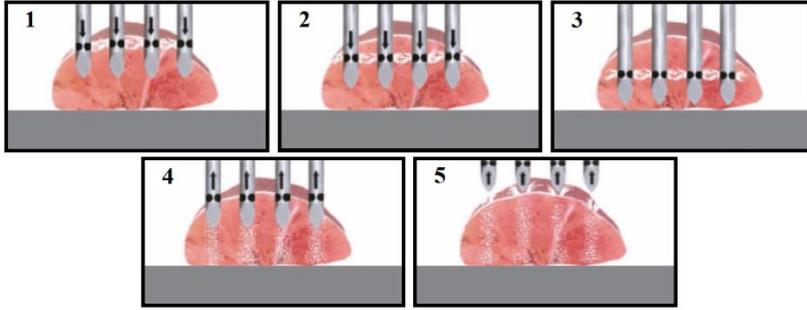
Fig. 1. (1-3) The needles are inserted into the meat while injecting brine (4-5) While the needles are subtracted from the meat, the brine pressure is turned off. After the needles are fully subtracted, the process is finished

The needle-based injection process is difficult to control due to high product variation e.g. amount of fat, meat, and skin. 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 as they are open-loop controlled with manual visual inspection by a worker.

2. Related work

To control needle-based injection an adaptive model is needed due to the variation in meat. One domain of adaptable models is Reinforcement Learning. Reinforcement Learning has previously shown potential to be applicable on a wide range of different control applications. In an industrial context several implementations are known such as robot control [9], robot motion planning [10], scheduling, and predictive maintenance of industrial plants [11]. An application area of great potential is specially in tasks of technical process control [12].

2.1. Background

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. The backbone of many reinforcement learning algorithms is the Markov decision problem (MDP) consisting of the tuple

- S is a finite set of states.
- A is a finite set of actions.
- $P_a(s, s')$ is the probability of transitioning from state s to s'.
- $R_a(s, s')$ is the reward of transitioning between state s and s'.
- γ is a discount factor to limit importance of states far into the future.

Here an agent acts in an environment transitioning between states and is given a reward based on its performance. The goal of an agent is to maximize the accumulated discounted reward by taking actions. With a finite number of

possible actions, the agent can estimate the accumulated discounted reward using a deep Q-network [7][8]. The agent can learn from new rewards by minimizing the temporal difference as shown in Eq. 1. [5]

$$min: r + \gamma \; argmax_{a'}\big(Q(s', a')\big) - Q(s, a) \tag{1}$$

When undertaking continuous action-spaces, policy-gradient-based actor-critic algorithms such as Deep Deterministic Policy Gradient (DDPG) has proven successful [6]. DDPG is a model-free off-policy deep reinforcement learning algorithm, which applies some techniques introduced with Deep Q-Networks (DQN), such as target networks and experience replay [7][8]. Target networks has proven useful for stabilizing the learning by making parts of Eq. (1) constant and thus the optimization becomes less sensitive to changes. When performing the optimization in Eq. (1), batch sampled uniformly from previous timesteps is used. This allows the agent to remember earlier episodes where it may have performed well. In order to represent the policy function (i.e. the actions to execute) independently of the value function approximating the accumulated future reward, an actor-critic learning algorithm is applied. DDPG uses two neural networks, one for the actor and one for the critic. A general actor-critic setup is displayed in Fig. 2.
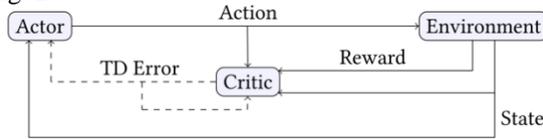
Fig. 2. Actor critic setup.

3. Experimental Setup

In order to train a reinforcement learning algorithm, a physical experimental setup is needed. The setup consists of a combination of off-the-shelf and custom designed components. A 3D CAD rendering of the experimental setup at Aalborg University can be seen on Fig 3.

Fig. 3. Experimental setup. A KUKA lightweight robot equipped with an injection needle above a high precision scale.

When doing brine injection, two parameters are used to control the process; injection pressure and injection time. Together these two parameters make up the action a reinforcement learning algorithm will take. To control these

actions, a fluid system is developed. The reinforcement learning algorithm also needs some information about the current state of the test objects, these are defined by a mass measurement of the specific test object. In order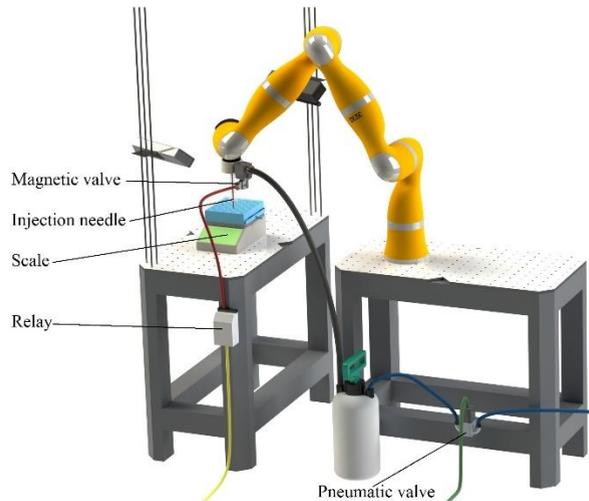 to obtain the current states of the test objects, a high-precision scale is included for mass. Tab. 1 displays the different components divided into categories.

Table 1. Components used for experimental setup.

Category	Component (Manufacturer, Comment)
Fluid system	Injection Needle Fixture (Custom Design)
	Pneumatic Valve (MetalWorks "Regtronic")
	Magnet Valve (RS, servo-assisted)
	Relay (Danbit)
	KUKA LWR 4 (KUKA, needle manipulator)
Mass detection	KERN KB 10K0.5N (KERN, 0.05 g precision)
	Water drain Plate (Custom Design)

3.1. 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 bacon. For this, inspiration has been taken from forensic ballistic science. In this branch of science, a gelatin solution, mixed from animal-based gelatin, is often used to imitate ballistic impact with humans, which is equivalent to pork and its tissue [4]. The bacon processed in the industrial case is dead tissue. 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, gelatin, as used to simulate ballistic impact, will also be applied in this research. Furthermore, the application of gelatin based test objects, allows for the use of smaller test objects, in this case cubes measuring 50x50x50 mm. Initial experiments show, that a compound consisting of only gelatin does not meet the demands of the industrial case, why an additive to increase the water absorption is introduced in the form of Sodium Polyacrylate, categorized as a Superabsorbant Polymers (SAP).

3.2. 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 14.0 % gelatin powder and 1.2 % SAP powder, relative to water, can 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.

3.3. Implementation of the DDPG Algorithm

Since the DDPG 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. 4, and the parameters used for the NN in the algorithm can be found in Tab. 2.
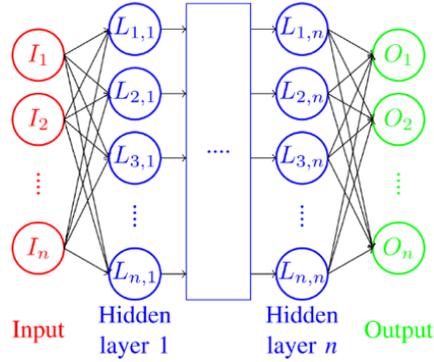
Fig. 4. General setup of a Neural Network, consisting of nodes (blue circles) connected through weights (lines between circles) as well as an input layer with observations of the state and an output layer estimating the value of each possible actions.

Table 2. The parameters used for the DDPG algorithm.

Exploration		Ornstein-Uhlenbeck
Activation Function		ReLU
Actor	Number of layers	3
	Nodes in layer	128
Critic	Number of layers	3
	Nodes in layer	128

The DDPG algorithm is implemented on the industrial case. The reward function is shown in Eq. 2. It has a target mass increase of 15 % as well as two terms penalizing actions the further away from 1.5 bar and 1 s they are, i.e. the middle of the action space. The constants were derived by an iterative process and 0.1 was found to be suitable values that produced good results.

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

For simulation purposes, the environment the agent traverses, has been modelled from experiments with the aforementioned gelatin compound. 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. 3.

$$f(x_1, x_2, x_3) = c_1 x_1^2 + c_2 x_2^2 + c_3 x_3^2 + c_4 x_1 x_2 + c_5 x_1 x_3 + c_6 x_2 x_3 + c_6 x_2 x_3 + c_7 x_1 + c_8 x_2 + c_9 x_3 + c_{10} \quad (3)$$

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

4. Experimental results

Running the agent in the simulation environment, yields the results seen in Fig. 5.
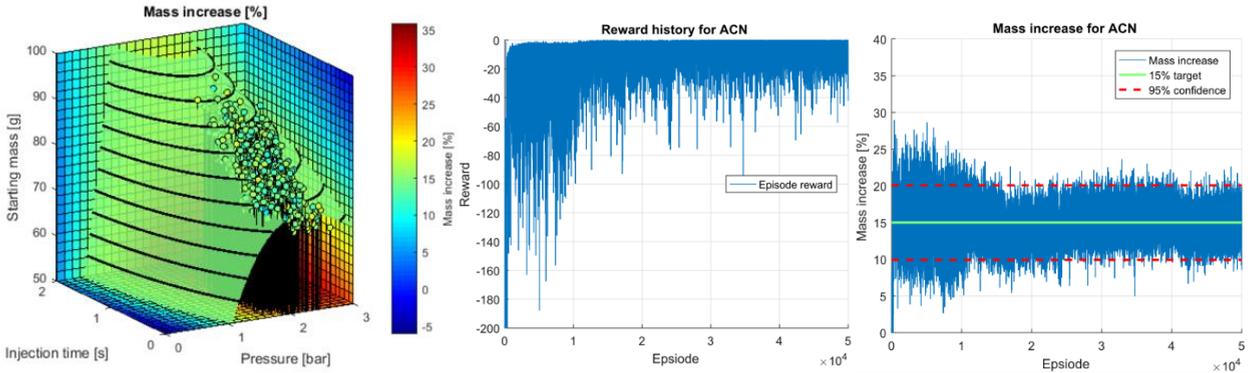

Fig. 5. (left) The surface represents the target mass increase, and the spheres are the achieved mass increases. (center) The reward the agent obtained during learning from Eq. 2. (right) The mass increase for each episode.

A mass increase with a mean of 14.9 % and standard deviation of 2.6 % was obtained, using starting masses drawn from a normal distribution of N(80.5 g, 4.8 g). The plot of the mass increase in Fig. 5 shows that even though the agent can produce a mean mass increase close to the target 15 %, the agent is not able to minimize the standard. The same behavior is seen in the reward plot with a noisy reward even after convergence. Fig. 5 (left) shows that even though the results are noisy, the agent is able to follow the curvature of the target mass increase, indicating that some learning has occurred. Tuning the hyper parameters of the DDPG may increase the convergence, however, this was not tested. The agent was run a second time using starting masses drawn from a normal distribution of N(80.5 g, 20.0g), showing the same standard deviation and a mean of 14.6 g, indicating that the variance in starting mass does not have an impact on the agents ability to adapt a model to the environment.

5. Conclusion

One of the challenges in brine injection is large variety, thus requiring an adaptive model to control. An experimental setup was developed using hardware measuring states defined as mass. Injection pressure and injection time was possible to control by an adaptable DDPG reinforcement learning model. Since it was not realistic to use bacon, a compound of SAP, gelatin and water was used. The compound was able to absorb water equal to a 28.9 % mass increase. The compound was then further used to create an approximation model decrease training time.

The DDPG model could adapt a model to a simulated environment with different parameters. With a target mass increase of 15 % was it capable of producing a mean of 14.9 % mass increase with a standard deviation of 2.6 %. This result was achieved with starting masses drawn from a normal distributions of N(80.5 g, 4.8 g). In combination, this means that an DDPG model was applied to the process control of brine injection in the simulation environment, and a mass increase was achieved. However, the distribution of the injected brine was not measured.

6. Future work

The presented solution is only carried out with small test objects. This was enough 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. For future work, it should be investigated how a potential replacement for the KUKA LWR 4 robot could be designed.

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 DDPG is able to adapt a model on physical test objects as well.

Acknowledgements

5. References

[1] C. Vestergaard, Salt Diffusion and Salt Distribution in Meat Curing (2004).

[2] M. Philipsen, R. Andersen, O. Madsen, T. Moeslund, Adaptive and Self-Learning Slaughter Robots, 6th Aalborg U Robotics Workshop, (2017) 16.

[3] R. Hafner, M. Riedmiller, Reinforcement Learning in Feedback Control, 84 (2011) 137-169.

[4] J. Jussila, Preparing ballistic gelatine - review and proposal for a standard method, Forensic Science International, 1 (2005) 91-98.

[5] R. Sutton, A. Barto, Introduction to Reinforcement Learning (2018).

[6] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, CoRR, abs/1509.02971 (2015).

[7] V. Mnih, K Kavukcuoglu, D. Silver, A. Rusu, A. Andrei, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, K. Andreas, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature, 518 (2015) 529-533.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller Playing Atari with Deep Reinforcement Learning, CoRR, abs/1312.5602 (2013).

[9] J. Peters, S. Schall, Policy Gradient Methods for Robotics, IEEE, (2006) 2219-2225.

[10] R. Meyes, H. Tercan, S. Roggendorf, T. Thiele, C. Buscher, M. Obdenbusch, C. Brecher, S. Jeschke, T. Meisen, Motion Planning for Industrial Robots using Reinforcement Learning, Procedia CIRP, 63 (2017) 107-112.

[11] P. Koprinkova-Hristova, Reinforcement Learning for Predictive Maintenance of Industrial Plants, Information Technologies and Control , 11 (2013) 21-28.

[12] R. Hafner, M. Riedmiller, Reinforcement learning in feedback control, Springer US, 84 (2011) 137-169.