

## DEVELOPMENT OF FREE-FLOWING PILE PUSHING ALGORITHM FOR AUTONOMOUS MOBILE FEED-PUSHING ROBOTS IN CATTLE FARMS

Rudolfs Rumba, Agris Nikitenko

Riga Technical University, Latvia

rudis.rumba@gmail.com, agris.nikitenko@rtu.lv

**Abstract.** The paper proposes a novel approach of transferring a free-flowing pile of different objects and materials significantly larger than the capabilities of the robot. Whilst trying to automatize feed-pushing in cattle farms with mobile robots, it was found, there may be no algorithm for mobile robots to do the job of this kind. With a robot measuring the impact force of his to the environment, it is possible to approximate how the overall mass pile is transferred as well as the overall change of mass. Change of the mass available to cattle is data that have not been available to farmers. This approach proposes that the feed pusher could give such information as a step closer to fully automatized cattle farms. As for feed pushers of nowadays, they are designed to drive the path hard-coded or even built into the farm, whereas this approach may give additional automation possibilities for the feed pusher, by using power consumption and position as a measurement, the proposed algorithm provides robot capabilities to approximate trajectories for the next passes and thereby be more effective. Approximations can be done by applying divide and conquer strategy to the pile itself, decomposing it in mass points with their relative area of action and the pre-design path it can move about. With this strategy it is proposed that the point is fixed to its path, but the mass is a variable, calculated based on the impact of a robot to one or multiple points of mass. The authors are proposing iterative algorithm to approximate the feed pushing robot trajectory based on sensed force feedback. The change of mass over time can be used to estimate the overall change in the pile's mass – actual consumption, as well as overall transfer over time. It is possible to describe most of the use-cases in the feed-pushing process, but there might be a wide range of other applications that include different free-flowing elements stacked in a pile that require reordering, gathering or transfer.

**Keywords:** mobile robot, feed pushing, automation, cattle farm automation.

### Introduction

To reduce production costs in a dairy farm it is needed to increase the milking yields and lower the labour costs. Although there are many technologies available specifically designed to automate dairy farms, still a lot of labour force is needed to keep the farm running. One of the tasks that must be done is feed-pushing. It is important to have cows eating for as much time as possible, to increase the milking yields (*litres per-cow per-day*). Although the food is brought into farm, the design of the feeding alley is limiting the access to the food. While looking through the pile of food the cow is trying to find the sweetest piece and while doing that with its nose the food gets transferred further away until it is out of reach. This may cause stress and injury and overall lowered milking yields.

Automatic feed-pushers [1-6] are designed to regularly push the otherwise unreachable food back to the cows causing no stress by doing the task slowly and quietly.

Although, the automatic feed-pushers may significantly decrease the labour force on this task, the task must be supervised. To convert the system to fully autonomous and redundant it must implement the technology developed for mobile robots, avoiding hard-coded tasks, decomposing the task of feed-pushing for mobile robots. Charging, path-planning [7; 8; 9], indoor navigation [10; 11] are algorithms that are available in different forms, but the feed-pushing itself as of transfer of free-flowing pile relatively larger than the robot in a desired location is not yet described as an algorithm for mobile robots.

The authors propose this as a novel approach in mobile robotics.

### Feed-pushing

The task is to push unavailable feed back towards the cows to keep the food available. The feed is available, if it is located not further from the feed alley sides than a distance –  $d_{\min}$ .

The limitations of feed-pushing are tied to simple mechanics, where the maximum pushing force –  $F_b$  is directly proportional to the robot's mass, but even with constant mass it may vary over different weather conditions like humidity, temperature, significantly decreasing  $F_b$ . There are possibilities to overcome  $F_b$  limitation by increasing the weight of the robot, but the goal of the algorithm is to be aware that there is such limitation it must adapt to.

The planning of this task must result in transfer of the feed pile in the area between the feed alley wall and  $d_{\min}$  choosing the path, where  $F_b$  is maximised, yet still under physical limitation.

### Divide and conquer approach

Robot's impact on the feed pile is only local and so the authors are suggesting not looking at a pile as one whole object (length of 100 m + ) or connected objects, but rather divide the feed pile in multiple influence areas in the size relative to the robot's dimensions. The area would consist of one dimension – length of the feed alley wall it has the influence on.

With the influence areas set, the pile can be abstractly described, so that every influence area with identifier  $i$  has its mass of the pile in this area –  $m_i$ , concentrated in a single point in a distance from the feed alley wall –  $d$ . Both of which are determined from feedback of the last run.

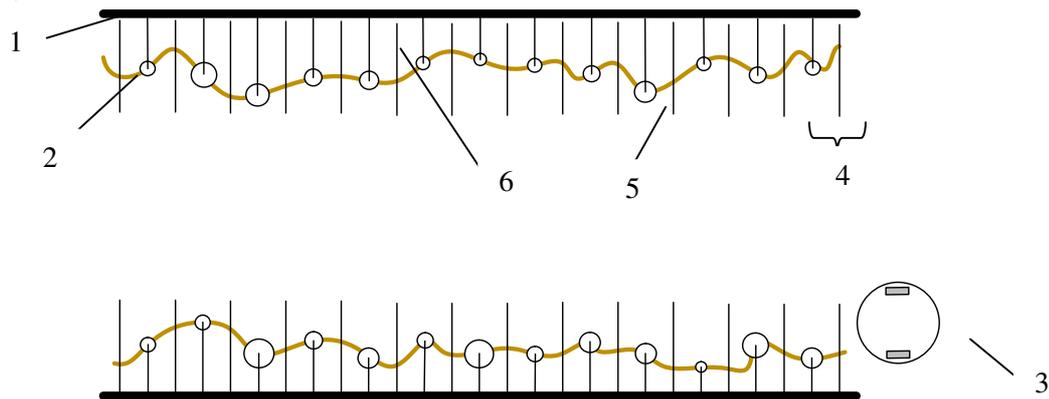


Fig. 1 **Schematic of feed alley and abstractions:** 1 – feed alley wall; 2 – actual position of the feed; 3 – feed-pushing robot; 4 – area of influence; 5 – point of mass; 6 – distance from the wall

Schematically the feed alley is shown in Figure 1; thick black line is the working area, that usually is 5 m in width and 100 + m in length.

This model of influence areas is not suggesting that points of mass are moving and transferred, but rather the pile is transferred and changing over time. Once the feedback is received, the particle mass and positions (distance from the wall) are updated.

### Feedback processing

Algorithm is a planner specifically designed for pushing purposes; the first part consists of feedback processing, but the second part – path planning. To collect the feedback, motor commands are executed one at a time changing state parameters of the robot – position; direction; induced  $F_b$ , which are used to calculate the points of mass. All involved mass points are recalculated, and the results fused in the map for the planner to calculate the next path. Each moment in time can be visualised as shown in Figure 2, where in reality pushing a pile of feed, it is possible to abstract the situation by knowing the exact location of the robot and defining the areas of influence.

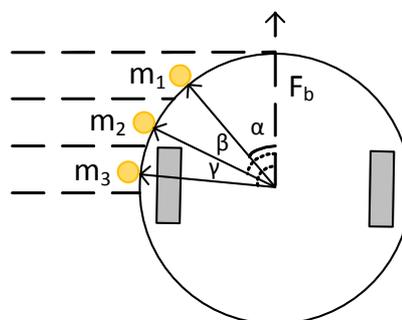


Fig. 2 **Evaluation of mass point [12]:**  $\alpha$ ,  $\beta$ ,  $\gamma$  – effective reaction angle of  $m_1$ ,  $m_2$  and  $m_3$

As the robot is entering the area of influence, a mass point with its mass is reacting on the robot's  $F_b$ . After series of measurements, where the time intervals between the measurements are minimised,

mass for all involved areas can be approximated. Note that the robot shape is circular, and the areas of influence right after  $m_3$  are ignored instead of negative reaction force [13].

**Path planning**

Feed can be pushed at any distance  $d$  from the wall, but the limitation is the robot's  $F_b$ . To maximise the work done,  $d$  has to be minimised. Path planning task is to find the distance  $d_{F_{bmax}}$  in each area of influence from the pushing force feedback generated in the last run. The assumption of the crossing force function must be made for every area of influence – a function that can be determined by feedback variables. A simplified version is proposed in Equation 1 and visualisation of the function in Figure 3a.

$$F_s(d) = \frac{1}{xd}, \tag{1}$$

where  $d$  – distance from wall;  
 $x$  – distribution constant;  
 $F_s$  – force needed to cross an area.

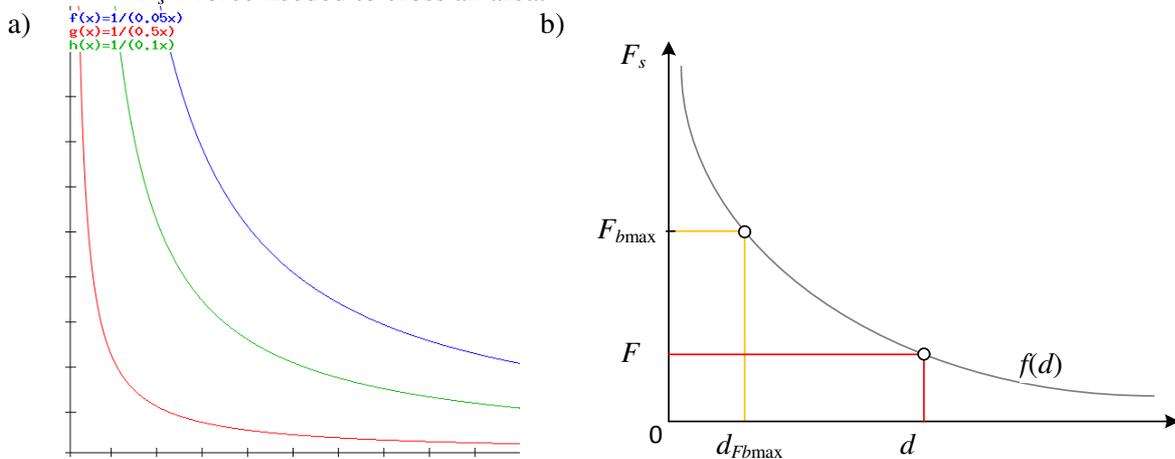


Fig. 3. **Crossing force function:** a – finding function; b – finding  $d_{F_{bmax}}$ , where  $F_{bmax}$  – device specific  $F_b$ ;  $d_{F_{bmax}}$  – approximated distance  $d$  of the limitation  $F_{bmax}$ ;  $F$  – feedback pushing force;  $d$  – feedback distance

Once the crossing force function is approximated, it is possible to evaluate the crossing distance for the next path, where the crossing force  $F_s$  is close to  $F_b$ , as shown in Figure 3b –  $d_{F_{bmax}}$ . Finding one  $d_{F_{bmax}}$  though is giving info only for a single area of influence. Full path planner will use all areas of influence feedback and approximation of  $d_{F_{bmax}}$ .

Further path planning considers the minimal distance the robot should push the feed as shown in Figure 4.

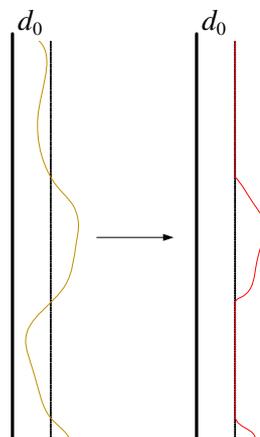


Fig. 4. **Path planner adjustments regarding  $d_0$ :** where  $d_0$  – minimal distance feed should be pushed to

As discussed, there is such distance from the wall that is considered as a maximal, where cattle may reach with no risk of injuries, therefore it may be proposed to push as close as possible. It may be false, since it may affect cattle wellbeing or disrupt the contact of cattle and feed or even physically push the cattle away from the feed. Therefore, it is proposed to have an additional constant distance from the wall, not necessary to cross –  $d_0$ . As additional algorithm goal is to push the feed closer than  $d_{\min}$ , yet still no more than  $d_0$ .

### Visualisation of the feed

As an auxiliary feature, visualisation is a way to validate the algorithm's feedback, generated feed pile model, determining precision of approximations.

As feedback is measured as a force, the more mass may be found in the area, the higher the reaction force, relative to the distance from the wall.

By having different mixtures, the density of the feed may vary and hence the volume of the feed, so as the profile of each is of influence, as Figure 5 shows. As the feed may be distributed as just a stacked pile, Figure 5a; a pile with reduced mass, where the cattle can access the feed, Figure 5b.

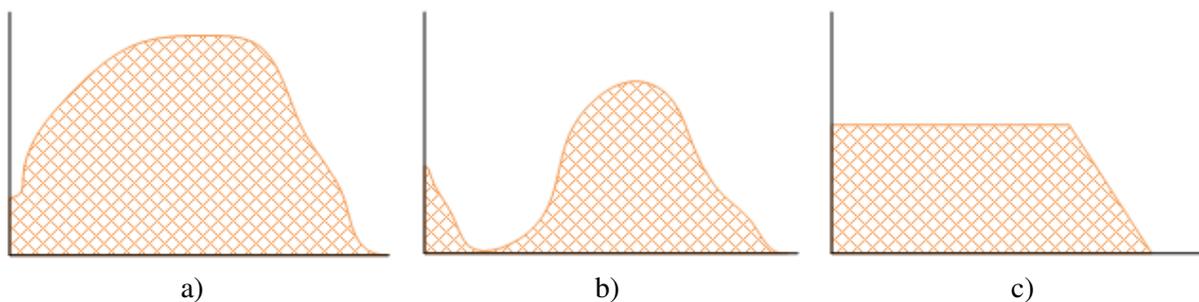


Fig. 5. Different possible profiles of feed in single area of influence

Without visual sensors, the actual distribution of feed is indeterminable. Therefore, the authors propose simplified visualisation as in Figure 5c, based on distinct variables of feed and feedback as shown in Figure 6 and Equation 2.

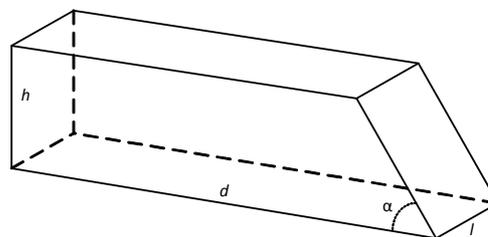


Fig. 6. Simplified component for visualisation

$$h = \operatorname{tg}(\alpha) \left( d - \sqrt{d^2 - \frac{2m}{\operatorname{tg}(\alpha)\rho l}} \right), \quad (2)$$

- where  $h$  – height of the component;  
 $\alpha$  – angle at which the feed holds stable onto itself;  
 $d$  – feedback distance from the wall;  
 $m$  – approximated mass;  
 $\rho$  – density of the feed;  
 $l$  – width of a single area of influence.

Although visualisation is not crucial to this algorithm, at best of our knowledge it has not been implemented in any system in cattle farms, generating a new kind of information, enabling such metrics for the farmer as overall mass on the feeding alley, so as for each area of influence –  $m_{\text{tot}}$ ,  $m_i$ ; change of the mass both overall and in a single area of influence –  $\Delta m_{\text{tot}}$ ,  $\Delta m_i$ ; hence, knowing the time interval between feedback, the activity of cattle can be measured.

Overall visual material could be generated as shown in Figure 7.

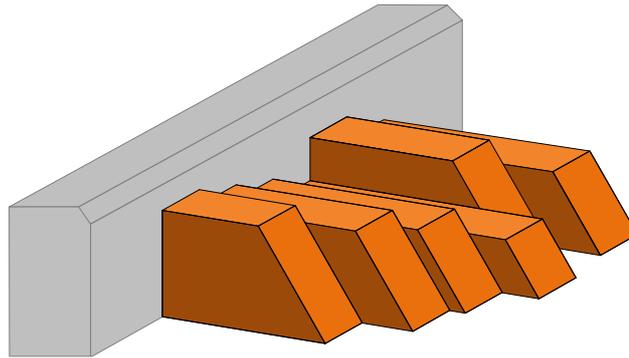


Fig. 7. **3D visualisation of feed pile:** where orange objects are the area of influence with its approximated feed volume component; and the grey object is the feeding alley wall, behind which the cattle are located

**Simulation and results**

A simulation was made to test the assumptions and validate feasibility of the algorithm. Simulation visual representation is shown in Figure 8.

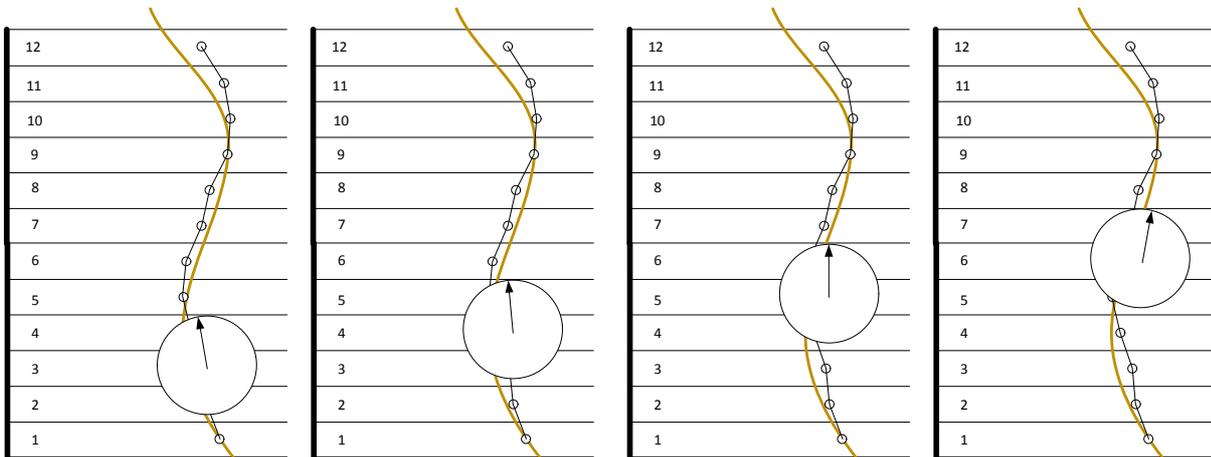


Fig. 8. **Simulation as iterative algorithm**

As a result of simulation, a feedback after execution of the previous path shown in Figure 9a (black line with dots) is monitored and the next pass' path shown in Figure 9b (black line with dots) is calculated.

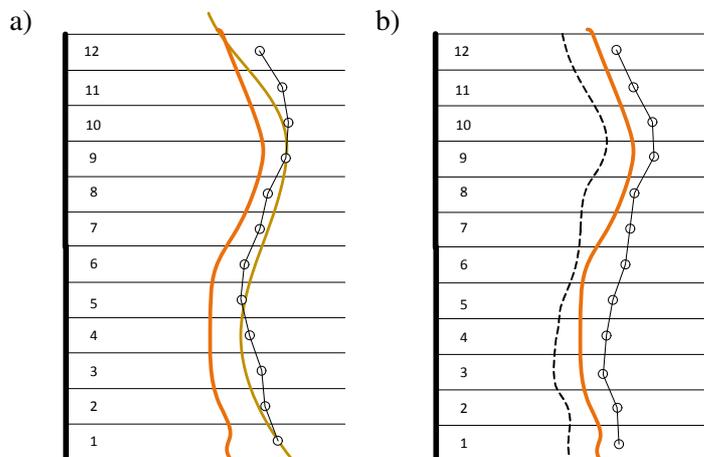


Fig. 9. **Result of simulation:** where yellow line – feed location before the pass; orange line – feed location after the pass; dashed line – potential feed location after the next pass

## Conclusions

This approach of feed pushing was made to fill the gap in mobile robotics and combine the task of feed-pushing with mobile robotics. The algorithm has not yet been implemented in any robotic system, yet it seems to have great potential in this field. Feed-pushing is just a small task in a long chain of tasks, yet another new application of the mobile robotics system that would benefit from the indoor navigation system and evolve feed-pushers.

The method itself is quite complex comparing to PID controlled feed pushers, but this would bring new metrics for the farmer and smart environment dairy farms are heading towards it.

## Acknowledgement

This research has been made as a part of electronic and optical manufacturing competence center of Latvia project “Multi-robot systems for applications in agriculture” which is a joint project of the Riga Technical University and SIA “Robotic Solutions”.

## References

- [1] LELY JUNO automatic feed pusher [online][1.05.2018]. Available at: <https://www.ley.com/solutions/feeding/juno/>
- [2] JOZ MOOV automatic feed pusher [online][1.05.2018]. Available at: <https://www.joz.nl/us/moov>
- [3] WASSERBAUER BUTLER GOLD automatic feed pusher [online][1.05.2018]. Available at: <http://www.wasserbauer.at/en/self-driving-slide-feed/butler-gold.html>
- [4] GEA FRONE automatic feed pusher [online][1.05.2018]. Available at: <https://www.gea.com/en/products/frone-feed-pusher.jsp>
- [5] ROVIBEC RANGER automatic feed pusher [online][1.05.2018]. Available at: <https://rovibecagrisolutions.com/en/produit/roviranger-feed-pusher>
- [6] VALMETAL PRO-FEED automatic feed pusher [online][1.05.2018]. Available at: <https://rovibecagrisolutions.com/en/produit/roviranger-feed-pusher>
- [7] Latombe J.C., “Robot motion planning”, New York, Springer 1991. 586 p.
- [8] Nikitenko A., Liekna A., Andersone I., Ekmanis M., Urtans E. Mobile Robot Path Planning for Indoor Use. No: 13th International Scientific Conference “Engineering for Rural Development”: Proceedings, Latvija, Jelgava, 29.-30. may, 2014. Jelgava: 2014, pp. 366-372.
- [9] Goyal J. K., Nagla K. S., “A new approach of path planning for mobile robots,” in Proceedings of the 3rd International Conference on Advances in Computing, Communications and Informatics (ICACCI '14), IEEE, New Delhi, India, September 2014, pp. 863-867.
- [10] Nikitenko A., Liekna A., Ekmanis M., Kulikovskis G., Andersone I. Single Robot Localisation Approach for Indoor Robotic Systems through Integration of Odometry and Artificial Landmarks. Applied Computer Systems. Vol.14, 2013, pp. 50-58. ISSN 22558683.
- [11] Nikitenko A., Grundspenkis J., Liekna A., Ekmanis M., Kulikovskis G., Andersone I. Multi-Robot System for Vacuum Cleaning Domain. No: Advances in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection: Proceedings of the 12th International Conference (PAAMS 2014), Spain, Salamanca, 4.-6. june, 2014. Heidelberg: Springer, 2014, 363.-366.lpp. ISBN 978-331907550-1
- [12] Weber J., Kaufmann C., Hache C., Schmidt M. Solving the box-pushing problem using a spherical robot, 2015, pp. 7-11. 10.1109/REM.2015.7380366.
- [13] De Berg M., Gerrits D. H. P. Computing push plans for disk-shaped robots. International Journal of Computational Geometry & Applications. 23, 2013, 10.1142/S0218195913500027.