Agricultural Robotic Candidate Overview using Co-model Driven Development

Martin Peter Christiansen[(a)], Peter Gorm Larsen[(b)], Rasmus Nyholm Jørgensen[(c)]

[(a),(b),(c)]Department of Engineering, Aarhus University, Finlandsgade 22, 8200 Aarhus, Denmark
[(a)]Agro Intelligence, Agro Food Park 13, Skejby, 8200 Aarhus N

[(a)]mpc@agrointelli.com, [(b)]pgl@end.au.dk, [(b)]rnj@end.au.dk

ABSTRACT
Multi-disciplinary technologies can be used to explore and compare design candidates in order to enhance the time-to-market development for robotic systems. The Crescendo technology lets software designers and engineers collaborate in the development for models containing discrete-event (DE) parts of the controller and continuous-time (CT) parts for the robot-environment interaction. Such models are defined as collaborative (co-models) and their joint execution is called a co-simulation. In this paper, we illustrate the development of a robot mink feeding system using the model-based Crescendo technology. The results of the co-simulations provided an overview of the candidate solutions in the chosen design space entirely in a virtual setting. The candidate overviews provided valuable input for selecting a candidate to develop into an actual robot. The selected candidate solution was subsequently deployed directly on a robot operating system (ROS) based platform and tested on a mink farm.

Keywords: 20-sim, Agriculture, Co-simulation, Crescendo-tool, Robotics

1. INTRODUCRION

Modelling and simulation are gradually being adopted as integral parts of the development process for robotic systems Harris and Conrad (2011); Longo and Muscato (2013). Modelling provides developers with the capability to explore the interaction between hardware and software solutions before developing the actual component. The modelling and simulation approach allows for the automatic evaluation of a much larger potential design space compared to a manual trial and error approach. The alternative approach to robotic systems involves significant time spent on ad-hoc trial-and-error testing to reach a usable system configuration of the physical system. The prime challenge here is that many complementary disciplines are necessary to determine viable solutions i.e. electrical, mechanical, software, embedded systems and signal processing Murata et al. (2000); Pannaga et al. (2013). Each discipline has different cultures, tools and methodologies, which can restrict the development of a cross-disciplinary project. Collaborative simulations (co-simulations) allow developers to examine different aspects of the system to explore design alternatives. Co-simulations are based on models that the developers utilise to describe the different aspects of the robotic system. Co-modelling and co-simulation are performed using the Crescendo technology Fitzgerald et al. (2014)[1]. Design space exploration (DSE) is the analysis of different candidate solutions using co-simulation. The idea behind DSE is to explore the various candidates being considered by the developers to determine a viable candidate solution. The design challenge presented in this paper is based on a robotic feeding system for agricultural farming applications.

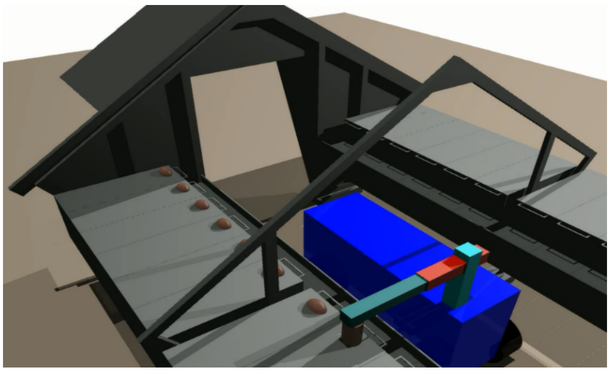

Figure 1: 3D visualisation of final version of co-simulated fodder dispensing robot operating inside mink farm house.

2. Co-model driven development

The co-model was designed for a robot system to dispense mink food along a row of cages at predetermined locations. The robot co-model was evaluated according to the overall system performance demands for the different system configurations.

[1]See www.crescendotool.org.

Proceedings of the Int. Conf. on Integrated Modeling and Analysis in Applied Control and Automation, 2016
ISBN 978-88-97999-82-9; Bruzzone, Dauphin-Tanguy, Junco and Longo Eds.

2.1. Collaborative Modelling Framework

Crescendo combines discrete event (DE) modelling of a digital controller and continuous time (CT) modelling of the plant/environment for a co-simulation. The Overture tool and the Vienna development method (VDM) formalism Fitzgerald et al. (2005); Larsen et al. (2010) were used to model the DE controller and 20-sim was used for the CT components. 20-sim is a modelling and simulation tool, able to model complex multi-domain dynamic systems, such as combined mechanical, electrical and hydraulic systems Filippini et al. (2007). VDM Real Time (VDM-RT) Verhoef (2006) is the dialect used in Crescendo with the capabilities to describe real-time, asynchronous, object-oriented features. Both VDM and 20-sim are well-established formalisms with stable tool support and a record of industry use.

The Crescendo co-simulation engine coordinates the simulation between 20-sim and the VDM tool through a protocol for time-step synchronisation between the tools. Crescendo binds the domain models together using the Crescendo contract and is responsible for exchanging information between the tools. The contract contains the parameters and variables that CT and DE developers need to be aware of when developing a combined model.

Crescendo contains a functionality enabling the developer to carry out DSE of a co-model Pierce et al. (2012b). DSE can be used to test and evaluate different system configurations like actuator, controller or sensor combinations in the design space that the developers plans to explore.

2.2. System Boundary Definition

The chosen robot is a four-wheeled vehicle with front wheel steering and the rear wheel differential driving Christiansen et al. (2015). The robot receives sensory inputs from a laser-range scanner, radio frequency identification (RFID) tag reader, IMU, and rotary encoders on the back wheel and front wheel kingpins. Actuators control the vehicle steering, driving, and feeding system based on the sensory inputs. A feeder arm system mounted on the robot dispenses the food on the cages at the predetermined locations. RFID tags are placed along the animal cage rows, to provide fixed reference locations. Fused sensory data based on the sensor input are utilised to determine the current location and enable the robot to perform actions in the environment.

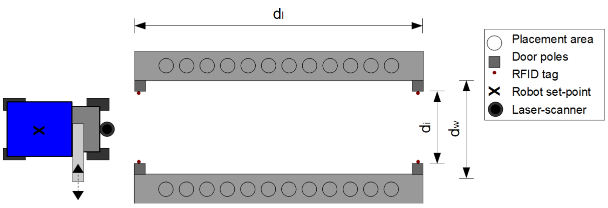

Figure 2: Sketch of example robot vehicle and feeding area inside a mink farm house where the fodder must be placed at specific locations.

System performance demands defines what the robot must achieve to be perceived as a viable solution. The project stakeholder for the system performance were as follows:

- Maximum vehicle speed of $0.25\frac{m}{s}$ (conforming to ISO-10218 2 (2013)).
- Feeding with a precision of $\pm 0.05m$ inside the placement areas.
- No collisions with the surroundings (see Figure 2).

Note that the performance demands are non-domain specific and focus on the overall response of the robot in action. Based on experience the standard length of row of mink farm houses d_l can be from $30m$ to several hundred metres. The widths of the entrance and exit d_i were 1.2-$1.55m$ and are the narrowest parts the robot must pass. The number of mink farm houses can differ from farm to farm, and they tend to be aligned in parallel. A lack of collision means that neither the vehicle nor feeding arm collide with the surroundings.

2.3. Model Development

We looked at different feeding arm solutions using co-modelling and co-simulation to determine a viable solution. We considered solutions with single- and double-sided feeding arm outputs with two either prismatic or revolute joints. The goal of this analysis was to determine the most viable candidate for development into an actual system. Double-sided feeding was considered based on an idea of better utilisation of the feeding robot. Feeding with both sides would double the output placement of fodder at the same vehicle speed. Shifting the arms half a cage length would allow the use of the same pump system for both sides while still allowing individual fodder amounts to be output.

The DSE functionality was used to evaluate each co-simulation robot system based on collision checking and placement of the mink fodder. For collision checking purposes, the robot's 2D bounding box was two rectangles: one for the vehicle body and one for the feeding arm system. If any bounding box comes within the range of the stored obstacles, the robot's pose is invalid. We utilised the method described in Fares and Hamam (2005) to performed the collision checking.

Evaluating the feeding output requires information on the placement of each fodder dispensed in the operational environment. When a co-simulation was run, the dispensed fodder from the robot was logged in a 2D XY grid with 0.01m intervals covering the mink cages. The logged fodder positions were used in the 3D visualisation like that portrayed in Figure 1 and processed afterwards to determine if the placement was a success. The execution was only run for a single house of mink-cages because the task would just be repeated for multiple houses without providing further insight when using co-simulation. We also limited the co-simulation to the first four meters of a mink farm house because we did not expect the remaining part

Proceedings of the Int. Conf. on Integrated Modeling and Analysis in Applied Control and Automation, 2016
ISBN 978-88-97999-82-9; Bruzzone, Dauphin-Tanguy, Junco and Longo Eds.

to provide any new insight. Throughout model development, we ran the same scenarios using DSE on the four different candidate solutions. The DSE results were used to determine when the co-model was working as intended and to allow the project stakeholders to compare the solutions.

The Crescendo contract in table 1 defines the parameters and variables to be exchanged during co-simulation. Shared design parameters are defined using the **sdp** keyword, variables operated by the CT side are defined by the **monitored** keyword and variables controlled by the DE side are defined by the **controlled** keyword. Shared design parameters represent values for which the developers want to explore the effect. $x_{init}, y_{init}, \theta_{init}$ define the starting position of the robot in the global coordinate frame. The crescendo DSE functionality was used to start the robot at different initial positions and evaluate if the co-model conforms to the project stakeholder demands. The controlled variables were the input to robot movement and the feeding arm. The robot movement input was transmitted to the drive motor u_o and front wheel steering actuator δ_{f_o}. The feeding arm transmitted the desired arm position y_{arm}, z_{arm} and current feeding output p_o in kilograms to the CT model. Local_Pose is the abstraction of the fused sensor input into a estimated position x_s, y_s, θ_s in the global reference coordinate frame. The abstraction of the fused sensor system is a general strength of modelling and simulation because this component did not need to be develop yet, so we were able to focus development efforts on steering and feeding control. IMU was also only represented by a single rotational variable rotated in the world frame, whereas the actual sensor may contain acceleration and rotation sensors for all three dimensions.

Table 1: Crescendo co-model contract

	Name	Type	Parameter symbol
sdp	Initial_Position	array	$[x_{init}, y_{init}, \theta_{init}]$
controlled	Wheel_Angle	real	δ_{fo}
controlled	Feeder_arm_pos	array	$[y_{arm}, z_{arm}]$
controlled	Feeder_out	real	p_o
controlled	Speed_out	real	u_o
monitored	Local_Pose	array	$[x_s, y_s, \theta_s]$
monitored	IMU	real	r_s
monitored	Encoder_Back	real	V_s
monitored	Encoder_Front	real	δ_{fs}

2.4. Discrete Event Modelling

The robot controller consists of a steering controller that can follow a pre-determined path and a feeding controller system to place fodder at the pre-selected positions. The steering controller steers the robot along the predetermined path, which is defined by a sequence of waypoints. The steering controller utilises the modal mode concept illustrated in Figure 3. The current modal controller mode is dependent on movement inside or outside the feeding area because two different operational strategies are used. The RFID tags at the entry and exit of the mink farm house determine the steering current mode and when the feeding arm should be deployed.

A combination of feedforward and feedback control is used to set the steering angle of the front wheels. The feedforward response is based on the kinematic bicycle model where L is the length of the wheelbase and V_s is the speed measured by the wheel encoders.

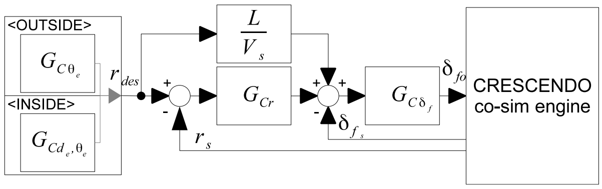

Figure 3: Block diagram structure of modal steering controller.

Inside the mink farm house, the robot needs to move along the cages in straight lines and to ensure that the feeding arms are held straight over the cages. Correct operation is ensured by maintaining a fixed distance and orientation to the sides of the mink cages. The control law employed by Nagasaka et al. (2004); Noguchi et al. (1997), which is given by Equation (1), was chosen for inside operation. The robot rotational angle speed r_des was set to be proportional to the errors in distance d_e and orientation θ_e:

$$r_{des} = \begin{bmatrix} K_{11} & 0 \\ 0 & K_{22} \end{bmatrix} \begin{bmatrix} d_e \\ \theta_e \end{bmatrix} \quad (1)$$

The controller parameter is tuned by the Ziegler-Nichols closed loop method. The parameter K_{22} is determined first and tuned to diminish the angle error θ_e. The procedure is then repeated for the K_{11} parameter for the distance error d_e. When the robot moves outside from the feeding area, the heading error θ_e in relation to the predetermined path of the robot is selected as the steering concept. A classic PD controller is used to steer the robot outside the mink farm houses, based on the method described in Bevly (2009).

When the robot moves into the feeding area, it stops to deploy the feeding arm system to the preselected position by updating y_{arm}, z_{arm}. Robot movement cannot continue before the feeding arm system has been completely moved in or out when the robot is entering or exiting a mink farm house. The robot has a feed map in the form of a sequence of amounts and positions of fodder to placed. The feeding arm system starts the feeding process using the output p_o when the next position in the map is reached.

2.5. Continuous Time Modelling

The 20-sim block diagram in Figure 4 represents the steering wheel actuator and mechanical setup to operate the front-wheel orientation δ_f. The input and output signals from the steering wheel are controlled using limiter function blocks to model operational range. The closed loop inner system represents the steering angle rate response to the requested steering angle δ_{f_o}.

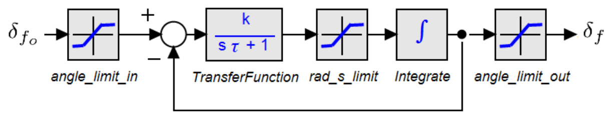

Figure 4: Model of steering wheel system based on method described in Bevly (2009)

The model of the robot vehicle utilises the bicycle approach meaning that lateral forces in the left and right wheels are assumed to be equal and summed together. This assumption holds for typical agricultural vehicle operation velocities (<7.5$\frac{m}{s}$) Karkee and Steward (2010). The bicycle structure, is also known as a half-vehicle (Figure 5). The model allows for yaw and lateral motions with the steering of the front wheel angle δ_f.

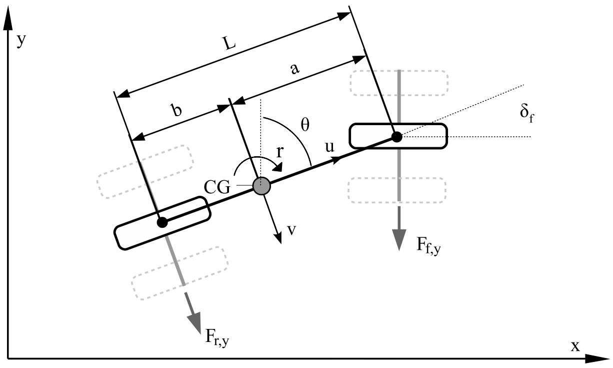

Figure 5: Dynamic bicycle model of the vehicle part of the robot system.

The velocities u, v are at the center of gravity of the vehicle. L is the wheelbase where a is the longitudinal distance to the front wheel and b is the longitudinal distance to the rear wheel. For a constant forward velocity, the vehicle motion is given by

$$m(\dot{v}+ur) = F_{f,y}cos(\delta_f) + F_{r,y} \tag{2}$$

where r is the angular rate about the yaw axis. Similarly, the vehicle yaw motion is expressed by

$$I_{zz}\dot{r} = aF_{f,y}cos(\delta_f) - bF_{r,y} \tag{3}$$

where I_{zz} is the moment of inertia along the yaw axis. We only considered the sideways force of the tire surface interaction because it provides the influence on the vehicle dynamics. The four factors which influence the lateral force are the normal force N, cornering stiffness C_α, rolling angle α and friction factor μ. Formula (4) and (5) are used to calculated the roll angles or the back and front wheel, respectively. The main difference is that (4) also considers the angle of the steering wheel.

$$\alpha_f = \frac{v+ar}{u} - \delta_f \tag{4}$$

$$\alpha_r = \frac{v-br}{u} \tag{5}$$

The sideways force is expressed for the linear and nonlinear cases by a piecewise-defined function.

$$F(\alpha) = \begin{cases} -C_\alpha tan(\alpha), & if\,|C_\alpha \alpha| < \frac{\mu N}{2} \\ -\mu N \frac{\alpha}{|\alpha|}(1-\rho(\alpha)), & if\,|C_\alpha \alpha| \geq \frac{\mu N}{2} \end{cases} \tag{6}$$

The linear case is used when the sideways stress in the contact patch upholds $|C_\alpha \alpha| < \frac{\mu N}{2}$. The function $\rho(\alpha)$ is defined by

$$\rho(\alpha) = \frac{\mu N}{4C_\alpha\,|tan(\alpha)|} \tag{7}$$

The output from the fodder outlet is modelled by standard first order differential describing output flow rate. When the fodder leaves the outlet their movement onto the cages are modelled using mass and earth gravity.

3. RESULTS

3.1. Design Space Exploration

Each solution was modelled to conform to the given system performance requirements and evaluated based on the DSE co-simulation response. Running DSE co-simulation scenarios throughout the development of the robot allows us to determine that the key obstacle is the mode change between outside and inside. The safe start/stop procedure

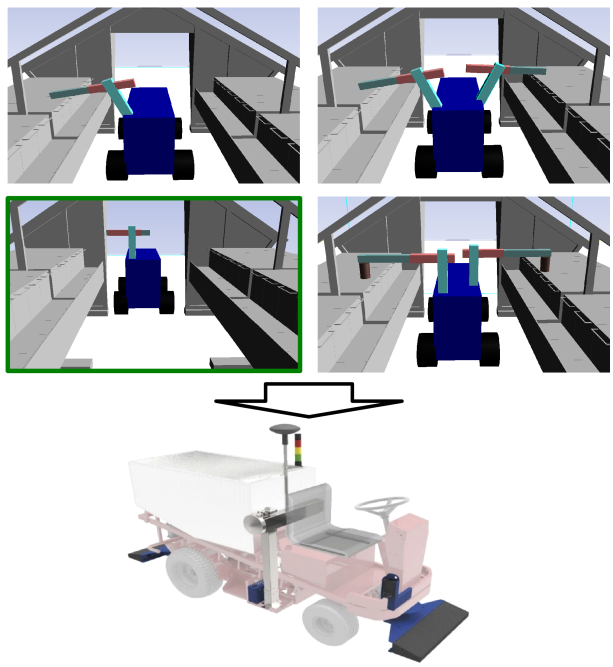

Figure 6: Feeding arm candidate solutions that was experimented with to determine a viable candidate solution.

when the robot needs to retract and deploy the feeding arm is the major factor that influences successful feeding. For the double sided candidate solutions, the number of start/stop procedures was doubled because the arms needed to be operated individually owing to the placement difference. Placing the fodder at the correct position mainly depends on accurate position information and can be achieved by increasing the number of RFID tags used

Proceedings of the Int. Conf. on Integrated Modeling and Analysis in Applied Control and Automation, 2016
ISBN 978-88-97999-82-9; Bruzzone, Dauphin-Tanguy, Junco and Longo Eds.

at reference positions along the cages. In the final version, all four candidate solutions were able to fulfil the stakeholder demands. The four candidate solutions for the four different arm and feeding systems are illustrated in Figure 6.

The single sided solution with prismatic joints was selected for feed-arm operation. Because feeding is generally performed at the same height for each row of cages, a prismatic joint solution was deemed to have better functionality and move with ease between positions. The prismatic solution was also deemed simpler to operate manually should this be needed. The single-side solution was chosen over a double sided solution because the stakeholders deemed it to be a better first prototype design for the actual robot. The double sided solution can easily be used to upgrade the same prototype platform later because no major software upgrades would be needed.

3.2. Experimental Results

To test the candidate solution, it was deployed into a vehicle solution as illustrated in Figure 7. The DE model was rewritten as a solution in the robot operating system (ROS) ecosystem Quigley et al. (2009). The ROS distribution Hydro Medusa was used in combination with ROS components from the Frobomind platform Jensen et al. (2014). The robot solution was implemented on a Norcar Minkomatic 660 DLA mink-feeding vehicle normally used for human operated feeding. The current version of

Figure 7: Robotic mink feeding system based on Norcar Minkomatic vehicle.

the robot system is intended as an add-on function to a standard vehicle platform. The solution also allows the operator to manually control the vehicle if necessary.

To evaluate the envisioned system, testing was performed at an outdoor mink farm in Denmark. Movement outside the mink farm houses is based on localisation using a reference map. The outside map provided in Figure 8 was created using OpenSlam's Gmapping ROS node. One part for each entry to the mink farm houses. Localisation with the created map was performed using an acml ROS node. Both gmapping and amcl are part of the ROS navigation stack. The path the robot must follow to move between houses was pre-planned based on the houses the robot was planned to cover in the current run.

Indoor operation was performed as defined in 2.4. Input to the steering controller was based on laser-range scanner measurements from a SICK TiM551. The laser-range scanner measurement was processed using RANSAC Fischler and Bolles (1981) to determine the robot's relative position to side of cages in terms of angle and distance. The robot's relative position was compared against the chosen reference for the robot to stay at.

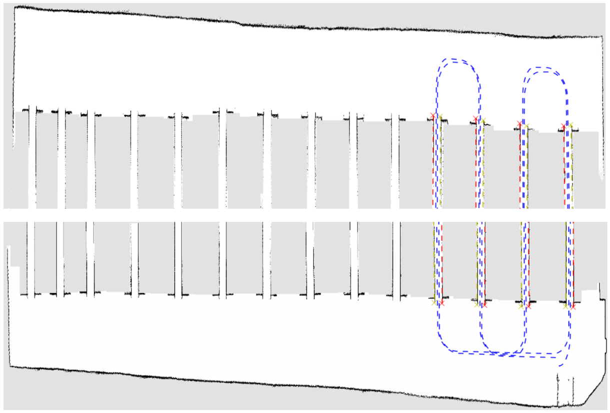

Figure 8: Local map of mink farm used in localisation and navigation outside. A combined version is shown in this picture, for both entrances to the mink farm houses. The blue line represents an example path that the robot can take between the mink farm houses. The farm is surrounded by a fence as indicated by the black border. The area the robot has lacks information on are marked in grey.

4. DISCUSSION

There is still more work to be done on the feeding robot system before a first version is completed. This paper illustrates how co-modelling and co-simulation can be used in robotic development. DSE was used to provide an overview of the candidate system configurations of mink arm feeding systems. Factors such as material, development, implementation and maintenance cost can influence the selection of a candidate configuration. Developers can use the DSE results to select a configuration to develop into an actual robot vehicle solution. DSE will not guarantee optimal solutions, but is a tool assisting with trade-off analysis with multiple candidate solutions and allows the project stakeholders to get an overview. The overview of the multiple feeding arm system solutions illustrates the DSE functionalities in Crescendo and shows that the method can support the selection of viable candidate solutions. The design overview approach would be an asset in the development of other robotic system where new aspects needs to be explored.

A similar simulation analysis could also have been developed directly in ROS, but the solution would lack the multidisciplinary tool based collaboration and development. Based on the results, we deemed that the co-modelling concepts can be used as a supplementary feature in robotics development tools like ROS and Microsoft

Developer Robotic Studio. Other tool combinations can be used for co-model driven development of agricultural and field robotic systems. The development tools should depend on the developer team's preferences and problems faced by the robot must.

5. RELATED WORK

In industrial robotic designs for indoor applications, research has indicated that a model-driven design approach incorporating co-simulation improves the cross-disciplinary design dialogue Broenink and Ni (2012); Broenink et al. (2010). Co-simulation has been used in the development of robot manipulators, where the tools Matlab Simulink and ADAMS were used to model the controller and robot body/environment, respectively Brezina et al. (2011); TIAN and SUN (2006). The ADAMS/Simulink combination has also been used to design a tomato harvesting robot manipulator Jun et al. (2012).

The Crescendo technology has been used to select viable candidate sensor positions on an R2-G2P line-following robot with a fixed controller setup Pierce et al. (2012a). The robot's performance was evaluated against a predefined set of marked curve segments to determine the most viable candidate solution. An adaptive controller solution was designed for an agricultural vehicle with a commercial GNSS based auto-steering solution. The solution was focused on finding the maximum safe speed for different load distributions on a tractor; these results were used to design the controller solution Christiansen et al. (2013).

Feeding robots for animal husbandry have previously been developed and documented in the literature. In Tan et al. (2007), a static feeding system was utilised in combination with an RFID reader to dispense food to cows with an attached RFID tag. The company Lely has developed a commercial mobile cow feeding robot which combines of ground metal wires for line following and ultrasound sensor for in-row movement. To date, the Lely Vector feed robot has not been documented in the academic literature, but it is a well-known commercial product.[2] In Jørgensen et al. (2007), outdoor piglet feeding was realised using a mobile feeding platform. The pig-feeding robot was utilised to influence the behavioural pattern and manure output of the piglets by daily changes of the feeding position in the field.

Feeding minks is a high precision task compared to cows and pigs because the normal feeding area has been empirically determined to be 0.2-0.35m for each cage. Each mink cage must be dosed with a predetermined amount of fodder placed on top. Our design approach looks at the navigational system, feeding controller and ground vehicle solution when making design choices based on the co-modelling. Our co-simulation based development approach to robot design is intended to determine a viable candidate solution for deployment in an actual platform.

[2]See www.lely.com for the Lely Vector.

6. CONCLUSIONS

Developing a robotic system to conform to the overall system requirements is essential. In this article, we described the concept of co-modelling and co-simulation as a robotic design approach. We showed how co-simulation using DSE can provide an overview of cross-disciplinary design candidates in robotic development. The model of the feeding robot combines modelling in VDM and 20-sim into a complete co-model to allow developers to utilise tools specific to their discipline. We believe that co-modelling and co-simulation combined with DSE can be utilised as an early stage development approach to analyse and compare design candidates from different domains.

A new European research project called INTO-CPS[3] will work on further improving the Crescendo technology. Efforts will include covering requirements for heterogeneous models and realisations of both controllers as well as physical components.

ACKNOWLEDGMENT

The work presented here is partially supported by the INTO-CPS project funded by the European Commission's Horizon 2020 programme under grant agreement number 664047. Thanks also go to Conpleks Innovation for their collaboration on the mink feeding case study and to Minkpapir A/S for providing the prototype platform. We also acknowledge Nick Battle for giving invaluable feedback on drafts of this paper.

References

ISO/TC 184/SC 2. Robots for industrial environments - safety requirements - part 1: Robot. Technical report, The International Organization for Standardization and the International Electrotechnical Commission, 2013.

David M. Bevly. *GNSS for Vehicle Control*. Artech House, 1 edition, 2009. ISBN 978-1596933019.

T. Brezina, Z. Hadas, and J. Vetiska. Using of Co-simulation ADAMS-SIMULINK for development of mechatronic systems. *14th International Conference Mechatronika*, pages 59–64, June 2011. doi: 10.1109/MECHATRON.2011.5961080.

J. F. Broenink and Y. Ni. Model-Driven Robot-Software Design using Integrated Models and Co-Simulation. In J. McAllister and S. Bhattacharyya, editors, *Proceedings of SAMOS XII*, pages 339 – 344, jul 2012.

Jan F. Broenink, Yunyun Ni, and Marcel A. Groothuis. On model-driven design of robot software using co-simulation. In E. Menegatti, editor, *Proceedings of SIMPAR 2010 Workshops International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pages 659–668, Darmstadt, Germany, November 2010. TU Darmstadt.

[3]This is an acronym for "Integrated Tool Chain for Model-based Design of Cyber-Physical Systems", see http://into-cps.au.dk/.

Martin Peter Christiansen, Morten Larsen, and Rasmus Nyholm Jørgensen. Collaborative Model Based Development of Adaptive Controller Settings for a Load-carrying Vehicle with Changing Loads. In Dionysis D. Bochtis and Claus Aage Grøn Sørensen, editors, *CIOSTA XXXV Conference*, July 2013.

Martin Peter Christiansen, Peter Gorm Larsen, and Rasmus Nyholm Jørgensen. Robotic Design Choice Overview using Co-simulation and Design Space Exploration. *Robotics*, pages 398–421, October 2015.

Charbel Fares and Yskandar Hamam. Collision detection for rigid bodies: A state of the art review. *GraphiCon 2005*, 2005.

Germán Filippini, Norberto Nigro, and Sergio Junco. Vehicle dynamics simulation using bond graphs. In *Conférence IMAACA*. Citeseer, 2007.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

John Fitzgerald, Peter Gorm Larsen, Paul Mukherjee, Nico Plat, and Marcel Verhoef. *Validated Designs for Object–oriented Systems*. Springer, New York, 2005. ISBN 1-85233-881-4. doi: 10.1007/b138800. URL http://overturetool.org/publications/books/vdoos/.

John Fitzgerald, Peter Gorm Larsen, and Marcel Verhoef, editors. *Collaborative Design for Embedded Systems – Co-modelling and Co-simulation*. Springer, 2014. ISBN 978-3-642-54117-9. doi: 10.1007/978-3-642-54118-6. URL http://link.springer.com/book/10.1007/978-3-642-54118-6.

Adam Harris and James M. Conrad. Survey of popular robotics simulators, frameworks, and toolkits. In *2011 Proceedings of IEEE Southeastcon*, pages 243–249. IEEE, March 2011. ISBN 978-1-61284-739-9. doi: 10.1109/SECON.2011.5752942.

Kjeld Jensen, Morten Larsen, Søren H. Nielsen, Leon B. Larsen, Kent S. Olsen, and Rasmus N. Jørgensen. Towards an open software platform for field robots in precision agriculture. *Robotics*, 3(2):207–234, 2014. ISSN 2218-6581. doi: 10.3390/robotics3020207. URL http://www.mdpi.com/2218-6581/3/2/207.

Rasmus Nyholm Jørgensen, Claus G. Sørensen, Helle Frank Jensen, Bent Hindrup Andersen, Erik Fløjgaard Kristensen, Kjeld Jensen, Jørgen Maagaard, and Alastair Persson. FeederAnt2 - An autonomous mobile unit feeding outdoor pigs. In *ASABE Annual International Meeting*, 0300. ASABE, 2007.

Wang Jun, Zhou Zhou, and Du Xiaodong. Design and co-simulation for tomato harvesting robots. In *Control Conference (CCC), 2012 31st Chinese*, pages 5105–5108. IEEE, 2012.

Manoj Karkee and Brian L. Steward. Study of the open and closed loop characteristics of a tractor and a single axle towed implement system. *Journal of Terramechanics*, 47(6):379–393, December 2010. ISSN 00224898. doi: 10.1016/j.jterra.2010.05.005.

Peter Gorm Larsen, Nick Battle, Miguel Ferreira, John Fitzgerald, Kenneth Lausdahl, and Marcel Verhoef. The Overture Initiative – Integrating Tools for VDM. *SIGSOFT Softw. Eng. Notes*, 35(1):1–6, January 2010. ISSN 0163-5948. doi: 10.1145/1668862.1668864. URL http://doi.acm.org/10.1145/1668862.1668864.

Domenico Longo and Giovanni Muscato. Design and Simulation of Two Robotic Systems for Automatic Artichoke Harvesting. *Robotics*, 2(4):217–230, December 2013. ISSN 2218-6581. doi: 10.3390/robotics2040217. URL http://www.mdpi.com/2218-6581/2/4/217/.

S. Murata, E. Yoshida, K. Tomita, H. Kurokawa, a. Kamimura, and S. Kokaji. Hardware design of modular robotic system. *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, 3:2210–2217, 2000. doi: 10.1109/IROS.2000.895297. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=895297.

Yoshisada Nagasaka, Naonobu Umeda, Yutaka Kanetai, Ken Taniwaki, and Yasuhiro Sasaki. Autonomous guidance for rice transplanting using global positioning and gyroscopes. *Computers and Electronics in Agriculture*, 43:223–234, 2004. ISSN 01681699. doi: 10.1016/j.compag.2004.01.005.

N. Noguchi, K. Ishii, and H. Terao. Development of an Agricultural Mobile Robot using a Geomagnetic Direction Sensor and Image Sensors. *Journal of Agricultural Engineering Research*, 67:1–15, 1997. ISSN 00218634. doi: 10.1006/jaer.1996.0138.

N. Pannaga, N. Ganesh, and R. Gupta. Mechatronics – an introduction to mechatronics. *International Journal of Engineering*, 2(8):128–134, 2013.

K. G. Pierce, C. J. Gamble, Y. Ni, and J. F. Broenink. Collaborative Modelling and Co-Simulation with DESTECS: A Pilot Study. In *3rd IEEE track on Collaborative Modelling and Simulation, in WETICE 2012*. IEEE-CS, June 2012a.

Ken Pierce, John Fitzgerald, Carl Gamble, Yunyun Ni, and Jan F. Broenink. Collaborative Modelling and

Simulation — Guidelines for Engineering Using the DESTECS Tools and Methods. Technical report, The DESTECS Project (INFSO-ICT-248134), September 2012b.

Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, 3, page 5, 2009.

CL Tan, Za Kan, MJ Zeng, and Jing-bin LI. Rfid technology used in cow-feeding robots. *Journal of Agricultural Mechanization Research*, 2:169–171, 2007.

Feng-lei TIAN and Rong-lei SUN. The Flexible Robot Arm Simulation System Based on ADAMS and Simulink. *Machinery & Electronics*, 10:020, 2006.

Marcel Verhoef. On the Use of VDM++ for Specifying Real-Time Systems. In J. S. Fitzgerald, P. G. Larsen, and N. Plat, editors, *Towards Next Generation Tools for VDM: Contributions to the First International Overture Workshop, Newcastle, July 2005*, pages 26–43, School of Computing Science, Newcastle University, Technical Report CS-TR-969, June 2006.

Proceedings of the Int. Conf. on Integrated Modeling and Analysis in Applied Control and Automation, 2016
ISBN 978-88-97999-82-9; Bruzzone, Dauphin-Tanguy, Junco and Longo Eds.