On Structuring Energy-aware Sequence-control Software

Reynaldo Cobos Mendez
University of Twente
Enschede, The Netherlands
r.cobosmendez@utwente.nl

Eddo Hobert
University of Twente
Enschede, The Netherlands
e.c.hobert@alumnus.utwente.nl

Douwe Dresscher
University of Twente
Enschede, The Netherlands
d.dresscher@utwente.nl

Jan Broenink
University of Twente
Enschede, The Netherlands
j.f.broenink@utwente.nl

ABSTRACT

This paper presents a framework to structure sequence-control software that accounts for the communication of energy information by design when interfacing planners with loop controllers. Communicating velocity or force values as setpoints to a loop controller can be characterised as energy that is downstreamed to the rest of the control structure and exchanged with the environment. Awareness of this energy information is useful for addressing dependability aspects in robotics where energy plays a role. This framework comprises metamodels and models for composing and structuring energy-aware sequence-control software that provides information of the energy supplied, for instance, by a trajectory planner. In addition, this paper gives an overview of the computation and communication requirements of this energy information. We present a use case where this structure facilitates using energy as a physical-interaction constraint and dependability metric for robot control.

KEYWORDS

control structure, software architecture, energy awareness, energy computation, planning, robot control, metamodel

ACM Reference Format:
Reynaldo Cobos Mendez, Eddo Hobert, Douwe Dresscher, and Jan Broenink. 2022. On Structuring Energy-aware Sequence-control Software. In *4th International Workshop on Robotics Software Engineering (RoSE'22), May 9, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3526071.3527517

1 INTRODUCTION

The concept of energy awareness in literature on software for robotics is extensively related to energy-efficient computation and battery-life extension. Examples reported in [14, 15, 29] use information of the system's energy consumption as constraint for robot mission's quality of service (QoS), or as metric to enhance battery autonomy. Using energy information is furthermore convenient in control design to shape robot behaviour within a larger stability area [21–23]. Examples reported in [3, 12, 26, 28] use energy information as a measure of robot-control stability, fault tolerance and safety. This is possible because energy – besides being a restrictive resource – influences aspects of control and physical behaviour.

Multiple papers explicitly address the importance of energy in loop control [11, 13, 20], particularly in robot control. However, little attention has been given to the influence of sequence-control actions in the energetic behaviour of robotic systems. A sequence controller communicating velocity or force values as a stream of setpoints to the loop controller can be characterised as an energy supply. For instance, trajectories computed by a planner can represent energy that is downstreamed to the rest of the control structure and exchanged with the environment. From the physics point of view, there is an implicit energetic relation between sequence controllers and the physical interaction of robots.

Awareness of this energy information and its communication in the system is useful for addressing dependability aspects in robotic applications. For instance, this information is used in robot safety, as constraint for safer trajectory tracking [10, 27] and safer human-robot interactions [17]; in robot autonomy, to reduce the energy consumption in robots following a certain trajectory [18, 19]; in control design, as a metric to provide stability guarantees to loop controllers [7]. These are examples of energy awareness focusing on different aspects in robotics that are influenced by energy.

Designing sequence controllers that are aware of the energy they supply requires software structures that integrate constraints of the physics domain which support the communication of energy information. This also calls for extending the concept of energy awareness to include different aspects influenced by energy in robot control. Communicating energy information in the software by design can ease addressing concerns in robotics where energy plays a role.

This paper presents a framework to structure sequence-control software that accounts for the communication of energy information when interfacing planners with loop controllers. We address energy awareness in robot software at an architectural level by providing composition metamodels and models that facilitate structuring sequence controllers with embedded information of the energy they supply.

Structuring the software as proposed here makes the energetic relation between sequence controllers and the physical interaction of robots explicit. This is exemplified with a use case where we implement the energy-aware sequence-control structure to a mobile

RoSE'22, May 9, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9317-1/22/05…$15.00
https://doi.org/10.1145/3526071.3527517

robotic platform to gather energy information that can be used as a dependability metric and physical-interaction constraint.

The remaining part of this paper proceeds as follows: Section 2 concerns the different aspects of energy awareness in robotics. Section 3 addresses the concept of sequence control and its structural considerations. Section 4 is an overview of the requirements to compute and communicate energy information in software. Section 5 presents our framework. Section 6 presents the use case on physical-interaction control of robots. Finally, Section 7 addresses the conclusion of this paper and recommendations.

2 ASPECTS OF ENERGY AWARENESS

Figure 1 depicts different aspects that energy awareness covers in robotics. This is because energy awareness has different connotations and purposes in, for instance, the software and control domains. To give clarity, we have identified these four major aspects that consider different roles that energy can have in robotics, namely:

(1) as a restrictive resource – e.g., for energy efficiency in robotic operations;
(2) as a metric for control stability – e.g., to preserve loop-control stability;
(3) as property to characterise the robot's physical interaction – e.g., to shape compliant robot behaviour;
(4) as criteria for higher-level decision making – e.g., to decide whether to execute certain robot actions.

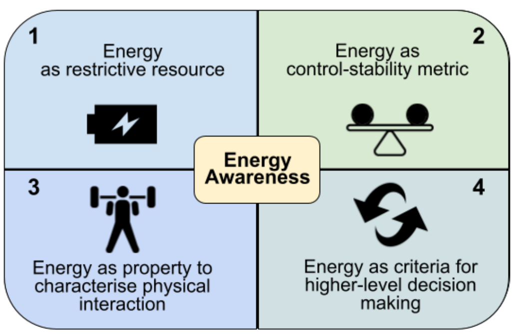

Figure 1: Aspects of energy awareness in robotics

Taking these aspects into account in control design can ease extending the concept of energy awareness in robotics. In this paper, we aim our contribution towards aspects 3 and 4 of Figure 1.

3 STRUCTURAL CONSIDERATIONS

It is convenient to structure the functionality and properties of the control software into layers with different sections of responsibility [1, 2, 8]. We adopt the layered architecture described in [4, 9], of which the basic structure is shown in Figure 2. Every control layer can exist without the layers above it (to the left side of each block in the figure) but requires the layers below it (the right side of each block in the figure) to operate. This paper focuses on the *sequence-control layer* and its interaction with the *loop-control layer*.

The sequence-control layer controls the sequence of robot movements. We consider that the sequence controller contains a planning algorithm – e.g., a trajectory planner as described in [16]. The sequence of robot movements is defined by the stream of motion-setpoints computed by the planner and downstreamed to the loop-control layer. The loop-control layer contains the control-law algorithm that computes steering values and downstreams them to the robot. This layered architecture depicts sequence control as a higher-hierarchy controller, meaning that it has no direct interaction with the physical world. We consider not interfacing sequence controllers directly with the physical world as a good practice in robot-control design.

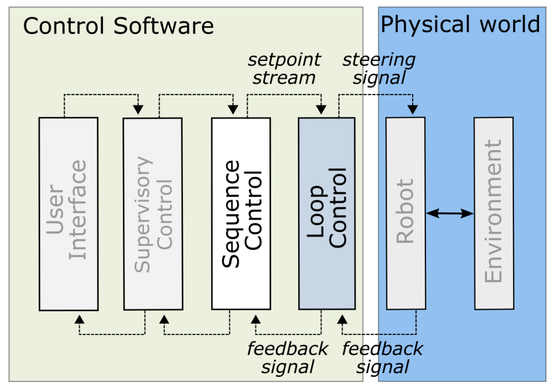

Figure 2: Basic structure of the layered software architecture (inspired by [9])

Designing controllers as composable and reusable *software components* is also a good practice. The RobMoSys[1] project proposes adopting a composition-oriented approach as a strategy to satisfy the demands of the robotics market. This entails constructing systems out of reusable, *off-the-shelf* software components with well-defined properties and interfaces. This approach is compatible with the layered structure in Figure 2.

We use both the layered control architecture and the RobMoSys approach to structure energy-aware sequence-control software. This paper fills in the two bespoke blocks in Figure 2 with a composition of components that compute energy information (i.e., in Joules).

4 ENERGY COMPUTATION AND COMMUNICATION

The correctness of the energy information depends not only on computation but also on the *time* at which the information is produced and communicated. These requirements are imposed by the physics domain and have to be considered when embedding energy information in the control software. Doing this right is important, for instance when the energy information is used for control stability or safety purposes [24]. This section gives a brief description of these physics-domain aspects.

[1]https://robmosys.eu/

In [25], the authors present a physics-conformal method to compute energy information. This is particularly useful when interfacing the control software with the physical world. In summary, we can calculate an energy quanta, ΔH (in Joules), representing the energy exchange between the loop-control layer and the robot in the communication of steering and feedback signals in Figure 2. This information can be computed using (1) for a force-output/position-input computation causality. In (1), F is the force (in Newtons) used as steering signal, x is the *sampled* position (in meters) of the robot used as feedback signal, and k is a positive integer that denotes the kth sample interval with time T (in seconds).

$$\Delta H(kT) := F(kT)\ (x(kT)\ - x((k-1)\,T)) \tag{1}$$

We can extend this method to a velocity-output/momentum-input computation causality using (2), where v is the velocity (in m/s) used as steering signal and p is the sampled momentum of the robot (in kg·m/s) used as feedback signal.

$$\Delta H(kT) := (p(kT)\ - p((k-1)\,T))\ v(kT) \tag{2}$$

Accurate ΔH values require the timely computation of (1) (or (2)), and the timely communication of the corresponding inputs and outputs. *Timeliness* implies that these events must happen at the same time. Therefore, these time constraints are handled by *real time* (RT) systems.

We are interested in the communication of the energy information, as it is binding for composing energy-aware sequence controllers. Previous work defines data-communication services that bring the physics-domain constraints of energy exchange to software models [5]. We refer to these communication services, to (1) and (2), and to the layered software architecture in Section 3 to constrain the communication, computation and composition of energy-aware sequence controllers.

5 FRAMEWORK

This section describes the energy-aware sequence control framework using metamodels and block-diagram models. The metamodels can help giving clear and unambiguous direction on the composition and structure of sequence controllers and their communication. The block-diagram models can help providing a different view of this approach. The blocks depicted in the metamodels and models represent components which functionality and communication is described along this section.

Figure 3 shows the composition metamodel of the Energy-aware Sequence Controller (EaSC). Figure 4 is an example block-diagram model of the EaSC that conforms to Figure 3. The EaSC contains – among possibly other elements – a *setpoint-computing* component and an *energy estimator*. The setpoint-computing component communicates setpoint values, *sp*, containing information of the trajectory to be followed by the robot. The energy estimator computes and communicates an energy budget, ΔH_{budget}, containing information of the energy required by the robot to follow *sp*.

In return, the EaSC receives feedback information on the state of the robot, Fb_{rbt}, and the state the loop controller, Fb_{LC}. Additionally, the EaSC receives information on the kinetic energy of the robot, E_{kin}, the potential energy in the loop controller, E_{pot}, and an energy-tank level, E_{tank}. We call these *energy-state* signals. Details on the flow of information in the EaSC are given later in this section.

5.1 Setpoint-computing component

The setpoint-computing component contains a trajectory planner – e.g., as described in [16] – which solution is used as setpoint for the loop controller. The setpoint signal (*sp* in Figure 4) contains information of velocities or forces corresponding to a single or multiple time horizons (i.e., one waypoint or a set of waypoints to be tracked by the robot).

We extend the setpoint-computing component with feedback and energy-state signals from the loop controller. This can be used, for instance, to adapt trajectories based on the energetic state of the system. It is up to the control designer using this information which is available by design in the control structure.

5.2 Energy estimator

The energy estimator, as defined in [3], is an element computing an estimate of the energy required by the robot to track the setpoint. We call this energy information *energy budget*, ΔH_{budget} (see Figure 4 and Figure 6), because it can be used as a constraint in the system, as we show later in the use case. Figure 5 depicts the composition metamodel of the Energy Estimator and Figure 6 is an example block-diagram model conforming to this metamodel.

The energy estimator simulates the controlled system dynamics using models of the loop controller, the robot and (if available) the environment. The communication between models is interpreted as exchange of physical *power* (i.e., in Watt), as it contains estimate information of forces and velocities. This requires the estimated steering and feedback signal-pairs, $[\tilde{St}, \tilde{Fb}]$, between model-blocks in Figure 6 to correspond to the same coordinate frame and be communicated at the same time [5]. The {power} constraint in Figure 5 indicates this requirement.

The *energy-sampling* component computes the energy budget using (3), where $\tilde{St}_{\text{LC}}$ and $\tilde{Fb}_{\text{rbt}}$ are the signals communicated between the loop-control and robot models. We use this particular pair of signals because the information exchanged between these two models can be interpreted as the energy that the robot is expected to spend to reach the setpoint.

$$\Delta H_{\text{budget}}(kT) := \tilde{St}_{\text{LC}}(kT)\ (\tilde{Fb}_{\text{rbt}}(kT) - \tilde{Fb}_{\text{rbt}}((k-1)T)) \tag{3}$$

For instance, if *force* is chosen as steering signal, $\tilde{St}_{\text{LC}}$, and *position* as feedback signal, $\tilde{Fb}_{\text{rbt}}$, then (3) conforms to (1). Similarly, if *velocity* is chosen as steering signal, $\tilde{St}_{\text{LC}}$, and *momentum* as feedback signal, $\tilde{Fb}_{\text{rbt}}$, then (3) conforms to (2). Note that the energy-sampling component does not alter the content of $\tilde{St}_{\text{LC}}$ and $\tilde{Fb}_{\text{rbt}}$ in any way.

5.3 Loop controller

The communication between the control law and the robot can also be interpreted as power (described by the {power} constraint in Figure 3). This gives a physical meaning to the interaction between the software and the robot, and allows describing it as an exchange of energy. We extend the loop controller with an *energy observer*

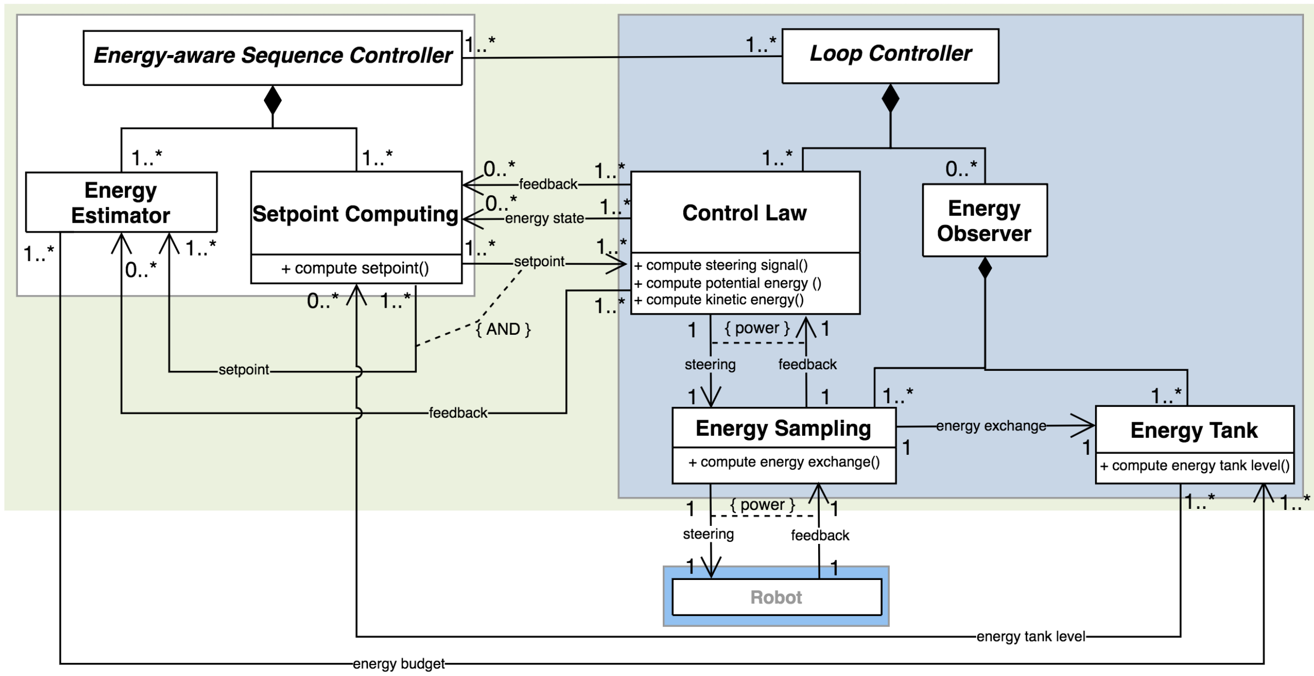

Figure 3: Energy-aware Sequence-control and loop-control composition metamodels

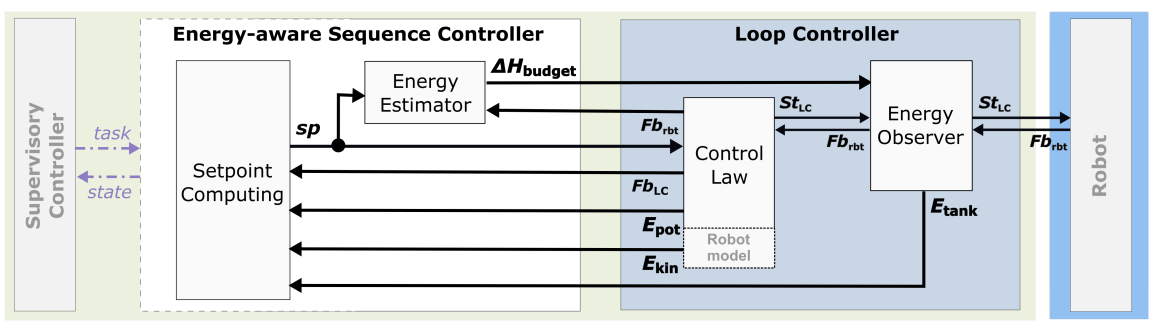

Figure 4: Block-diagram model of the basic composition and communication of the Energy-aware Sequence Controller. This model conforms to Figure 3.

component (see Figure 3 and Figure 4) to account for this energy information.

The energy observer contains an energy-sampling component and an *energy tank*. The energy-sampling component computes the exchange of energy between the control law and the robot, ΔH_{ex}, using (4). Note that (4) should conform to either (1) or (2). The energy tank is a *memory* element that keeps track of the energy *information* that enters and leaves the system at all time in the communication of steering and feedback signals. This information is stored in the *energy-tank level*, E_{tank}, and is calculated using (5).

$$\Delta H_{\text{ex}}(kT) := St_{\text{LC}}(kT)\,(Fb_{\text{rbt}}(kT) - Fb_{\text{rbt}}((k-1)T)) \qquad (4)$$

$$E_{\text{tank}}(kT) = E_{\text{tank}}((k-1)T) + \Delta H_{\text{budget}}(kT) - \Delta H_{\text{ex}}(kT) \qquad (5)$$

Comparing the energy budget with the actual energy exchanged between the software and the robot can be used, for instance, as a metric to determine whether the robot is spending more energy than needed/desired. If $E_{\text{tank}} < 0$, it shows that the robot is spending more energy than expected due to unforeseen conditions in the environment or in the robot itself. This can be useful for shaping robot behaviour (e.g., by adapting trajectories), or for characterising the physical interaction of the robot (e.g., by detecting changes in the environment).

Knowing the energetic state of the system is useful for getting insight on the physical interaction of the robot with its environment. This requires communicating physical information on the potential and kinetic energy of the system to the sequence controller. The loop controller can share information on its potential energy, E_{pot}

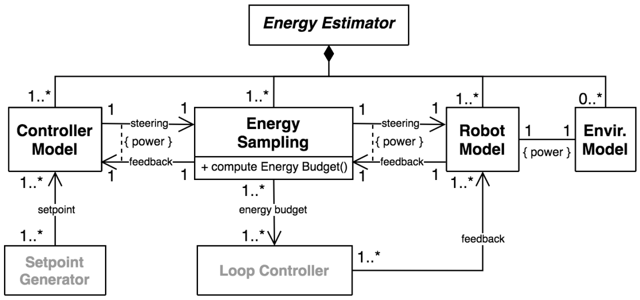

Figure 5: Composition metamodel of the Energy Estimator (inspired by [3])

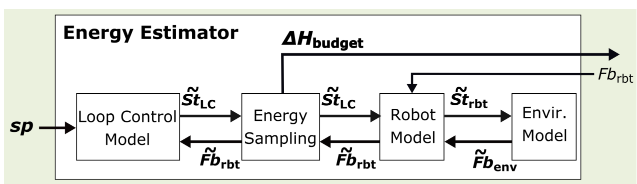

Figure 6: Block diagram of the basic composition of the Energy Estimator. This model conforms to Figure 5.

(see Figure 4), which represents the energy stored in the control law. The kinetic energy, E_{kin} (see Figure 4), represents the energy stored in the moving mass of the robot and can be estimated using a model of the robot.

6 USE CASE

An energy-aware sequence controller requires mechanisms to shape the robot's behaviour based on the energetic information of the system. The functionality of such mechanisms depends on the application and cannot be generalised in this framework. Nonetheless, this section presents an example that associates the energy information of the robot with the aspects of energy awareness described earlier.

We use a simulation of a mobile robotic platform (e.g., a floor-cleaning robot) to showcase how the energy information embedded in the composition of the EaSC can be used for:

- detecting faults associated with the energetic interaction between the robot and the environment;
- monitoring the robot's energy expenditure;
- making higher-level decisions based on the system's energetic state.

Figure 7 shows the system of interest. The robotic platform is represented as a 1-DOF body with 1 kg mass and a viscous friction of 1 Ns/m in the direction of movement. The structure of the control software conforms to Figure 3.

The control law in the loop controller is a PID controller steering the robotic cart with a force signal and measuring its position as feedback signal. The sequence controller contains a trajectory planner outputting position setpoints. The goal of the planner is to drive the mobile platform a distance of 1 meter in 1 second in a straight-line trajectory.

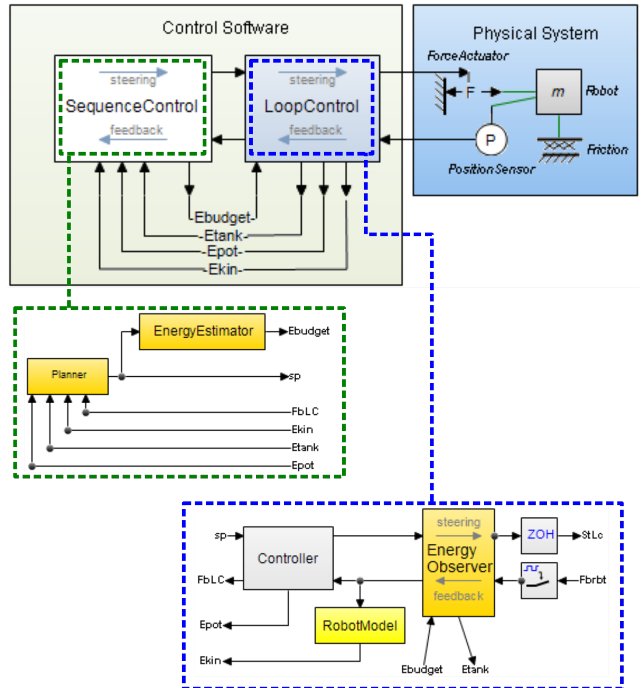

Figure 7: Control structure for the 1-DOF use case. The physical system consists of a mobile robotic platform with viscous friction.

Figure 8 depicts the time series of the position, energy budget and energy-tank level of the system under the ideal situation where the parameters of the models in the energy estimator match the parameters of the physical system. The numeric value of the energy-tank level in Figure 8 is mostly positive and close to zero during the time window in which the energy budget is communicated to the loop controller – i.e., $\Delta H_{\text{budget}} = \Delta H_{\text{ex}}$ in (5). This means the robot does not spend more energy than anticipated by the sequence controller. This can also be seen in the estimated position of the model matching the position of the physical system. We use these results as ground truth for this use case.

We study the behaviour of the system when the mobile robot presents a higher viscous friction than expected (3 Ns/m instead of 1 Ns/m) – e.g., due to changes in the environment or malfunction of the robot. We present two scenarios: 1) higher viscous friction without adapting the trajectory, and 2) higher viscous friction with energy-aware adaptation of the trajectory. The loop controller communicates feedback and energy-state signals to the EaSC in both scenarios but only in the second scenario this information is used to adapt robot behaviour.

6.1 Higher friction without adapting the trajectory

Figure 9 depicts the behaviour of the system when the conditions of the physical system deviate from what is modelled in the energy estimator, and no action is taken to adapt the trajectory. In consequence, the energetic behaviour of the robot deviates from

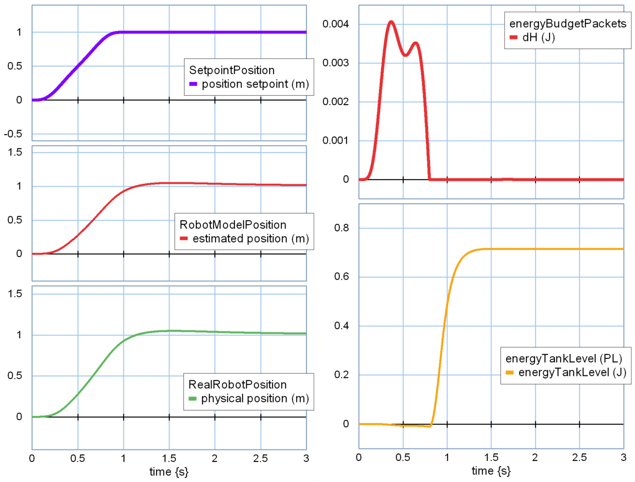

Figure 8: Ideal situation where the parameters of the models match the parameters of the physical system. At the left side, the time series of the position values of the setpoint, the robot model (inside the energy estimator), and the real robot. At the right side, the time series of the energy budget and the energy-tank level.

what was estimated due to the unaccounted increase of the viscous friction. The numeric value of the energy-tank level is mostly negative during the time window in which the energy budget is communicated to the loop controller – i.e., $\Delta H_{\text{budget}} < \Delta H_{\text{ex}}$ in (5). This indicates that the increase in the viscous friction causes the robot to spend more energy than what was estimated by the EaSC. This is also shown in the position of the physical system deviating from the position estimated by the models.

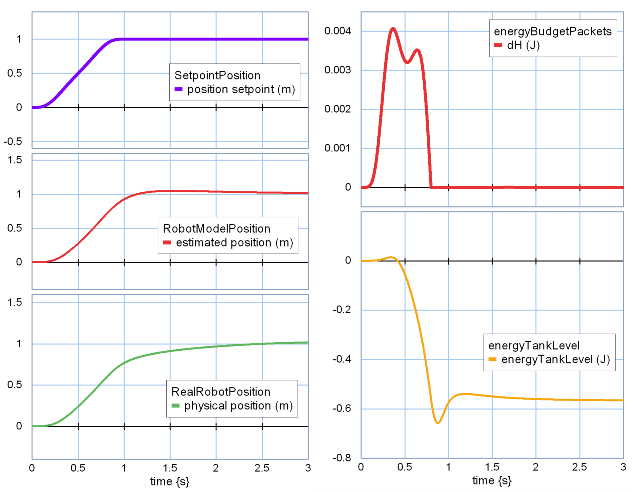

Figure 9: Situation with higher viscous friction without adapting the trajectory. At the left side, the time series of the position values of the setpoint, the robot model (inside the energy estimator), and the real robot. At the right side, the time series of the energy budget and the energy-tank level.

6.2 Higher friction with energy-aware trajectory adaptation

Figure 10 depicts the behaviour of the system when the conditions of the physical system deviate from what was modelled in the energy estimator. However, this time the EaSC takes action so that $E_{\text{tank}} \geq 0$. The EaSC uses information on the state of the energy tank and the velocity of the robot to make an intelligent, energy-aware adaptation of the setpoint trajectory. As a consequence, the planner drives the mobile robot slightly slower, as shown by the time series of the position in Figure 10 (compared to Figure 8 and Figure 9). This results in the numeric value of the energy-tank level being positive during the time window in which the energy budget is communicated to the loop controller.

Adapting the setpoints also results in the robot spending less energy to reach the goal despite the unaccounted increase of viscous friction (compared to Figure 9). This is expected because slower movements require less energy. Please note that the adaptation of setpoint values is rather crude and results in some undesired switching effects in the energy budget and energy-tank level in Figure 10.

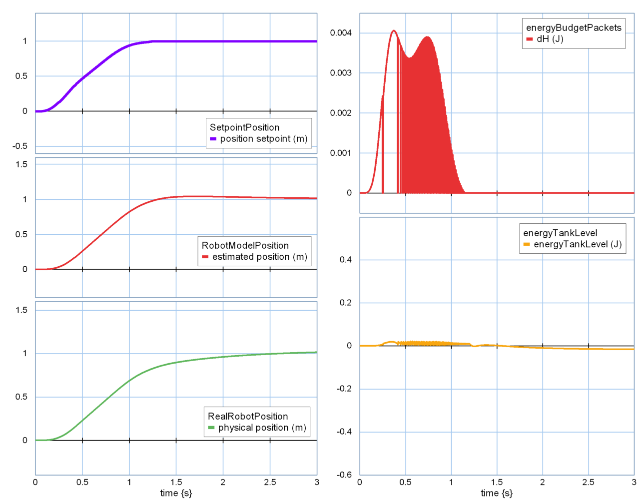

Figure 10: Situation with higher viscous friction but sequence-control actions are adapted so that $E_{\text{tank}} \geq 0$. At the left side, the time series of the position values of the setpoint, the robot model (inside the energy estimator), and the real robot. At the right side, the time series of the energy budget and the energy-tank level.

6.3 Discussion

The state of the energy tank in (5) can be useful to detect anomalies in the physical system. This use case gives an example showing the impact of, for instance, a change in the friction on the the energetic behaviour of the robot. The energy-tank level can also be an indicator of the robot's energy consumption and be used as a trigger for higher-level controllers to take actions to preserve battery life. For instance, the positive increase in the energy-tank level in Figure 8 represents energy being transferred from the environment to the controller. This energy (about 0.7 J) can be used to recharge the mobile robot's battery through regenerative breaking.

Besides monitoring the system's energy to detect changes in the properties of the physical system, the energy information can be useful for addressing other concerns in robotics such as safety, autonomy and system performance. For instance, the energy-aware sequence controller:

- provides energy budgets that can be useful as metric for preserving loop-control stability (as described in [3, 6]) when interfacing with trajectory planners;
- provides information of the energetic behaviour of the robot useful to describe its physical interaction with the environment – e.g., contact with an object;
- provides information on the energy exchange of the system ahead of time to help deciding whether or not to execute certain sequences – e.g., to maintain safety metrics or to preserve battery life.

Nonetheless, the accuracy of the energy information – and therefore its usefulness – does not solely depend on the composition of the control software, but also on its computation and communication concerns. It is important to compose the energy estimator with accurate models of the loop controller, robot and environment such that deviation between the energy budget and the actual energy spent by the robot is minimised. Moreover, the sampling of energy (i.e., computing ΔH_{budget} and ΔH_{ex}) and its communication has to be carried out using the right equations and strict timing constraints, so that it conforms to the physics domain as described in Section 4.

7 CONCLUSIONS

This paper presents a framework for composing energy-aware sequence controllers. We provide metamodels and example models as guidelines for embedding sequence controllers with energy information that can be used to address aspects in the robotics domain where energy plays a role. This extends the concept of energy awareness to higher-control levels, providing them with information on the physical interaction of the robot with its environment.

This approach not only allows sequence controllers to know the energy required to execute the trajectories computed by planners, but also facilitates assessing the behaviour of the system based on its energy information. This allows controllers to make decisions based on the physical interaction of the system. The use case presented here shows this property by using energy information to detect unexpected situations in the controlled system or the environment, and adapt the sequence of robot movements accordingly.

Energy awareness as a system property requires handling the energy information according to physical laws. Doing this accurately is important, as it has an impact on multiple concerns in robotics, including safety. Therefore, following the guidelines described in this paper facilitates the development of control software that conforms to physical laws.

Energy-aware sequence control is the next step towards energy-aware control-software architectures. Next work is on enabling energy awareness in the supervisory-control layer, and designing methods to communicate energy information across the entire system. In addition to energy-aware composition, computation and communication, further work will be done on energy-aware configuration and coordination of robot-control software.

ACKNOWLEDGMENTS

This research received partial funding from the RobMoSys project (EU project No. 732410) under the subproject EGCS https://robmosys.eu/egcs/

REFERENCES

[1] Stuart Bennett. 1988. *Real-Time Computer Control.* Prentice Hall.
[2] William Bolton. 2015. Chapter 13 - Control Systems. In *Instrumentation and Control Systems (Second Edition)* (second edition ed.). Newnes, 281–302. https://doi.org/10.1016/B978-0-08-100613-9.00013-4
[3] Y. Brodskiy. 2014. *Robust Autonomy for Interactive Robots.* Ph. D. Dissertation. University of Twente, Netherlands. https://doi.org/10.3990/1.9789036536202
[4] Johannes F. Broenink, Yunyun Ni, and M. A. Groothuis. 2010. On Model-Driven Design of Robot Software Using Co-Simulation. In *Proceedings of SIMPAR 2010 Workshops International Conference on Simulation, Modeling, and Programming for Autonomous Robots.* TU Darmstadt, 659–668.
[5] Reynaldo Cobos Mendez, Douwe Dresscher, and Jan Broenink. 2021. Power and Energy Communication Services for Control-software Models. In *2021 IEEE/ACM 3rd International Workshop on Robotics Software Engineering (RoSE).* 55–62. https://doi.org/10.1109/RoSE52553.2021.00016
[6] M.C.J. Franken. 2011. *Control of Haptic Interaction : An Energy-Based Approach.* Ph. D. Dissertation. University of Twente, Netherlands. https://doi.org/10.3990/1.9789036531894
[7] Michel Franken, Stefano Stramigioli, Sarthak Misra, Cristian Secchi, and Alessandro Macchelli. 2011. Bilateral Telemanipulation With Time Delays: A Two-Layer Approach Combining Passivity and Transparency. *IEEE Transactions on Robotics* 27, 4 (Aug. 2011), 741–756. https://doi.org/10.1109/TRO.2011.2142430
[8] Gerhard Greeff and Ghoshal Ranjan. 2004. 3 - System Hierarchies and Components. In *Practical E-Manufacturing and Supply Chain Management.* Newnes, Oxford, 26–65. https://doi.org/10.1016/B978-075066272-7/50006-3
[9] Marcel A. Groothuis, Raymond Frijns, Jeroen Voeten, and Johannes F. Broenink. 2009. Concurrent Design of Embedded Control Software. In *Proceedings of the 3rd International Workshop on Multi-Paradigm Modeling (MPM2009) (Electronic Communications of the EASST, Vol. 21).* European Association for the Study of Science and Technology, Netherlands. https://doi.org/10.14279/tuj.eceasst.21.284
[10] Sami Haddadin, Simon Haddadin, Augusto Khoury, Tim Rokahr, Sven Parusel, Rainer Burgkart, Antonio Bicchi, and Alin Albu-Schäffer. 2012. A Truly Safely Moving Robot Has to Know What Injury It May Cause. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems.* 5406–5413. https://doi.org/10.1109/IROS.2012.6386163
[11] B. Hannaford. 1989. A Design Framework for Teleoperators with Kinesthetic Feedback. *IEEE Transactions on Robotics and Automation* 5, 4 (Aug. 1989), 426–434. https://doi.org/10.1109/70.88057
[12] Blake Hannaford and Jee-Hwan Ryu. 2002. Time-Domain Passivity Control of Haptic Interfaces. *IEEE Transactions on Robotics and Automation* 18, 1 (Feb. 2002), 1–10. https://doi.org/10.1109/70.988969
[13] D. Hill and P. Moylan. 1976. The Stability of Nonlinear Dissipative Systems. *IEEE Trans. Automat. Control* 21, 5 (Oct. 1976), 708–711. https://doi.org/10.1109/TAC.1976.1101352
[14] D. Ho, K. Ben Chehida, B. Miramond, and M. Auguin. 2019. QoS and Energy-Aware Run-Time Adaptation for Mobile Robotic Missions: A Learning Approach. In *2019 Third IEEE International Conference on Robotic Computing (IRC).* 212–219. https://doi.org/10.1109/IRC.2019.00039
[15] M. F. Jaramillo-Morales, S. Dogru, L. Marques, and J. B. Gomez-Mendoza. 2019. Predictive Power Estimation for a Differential Drive Mobile Robot Based on Motor and Robot Dynamic Models. In *2019 Third IEEE International Conference on Robotic Computing (IRC).* 301–307. https://doi.org/10.1109/IRC.2019.00056
[16] Steven M. LaValle. 2006. *Planning Algorithms.* Cambridge University Press.
[17] Anis Meguenani, Vincent Padois, Jimmy Da Silva, Antoine Hoarau, and Philippe Bidaud. 2017. Energy Based Control for Safe Human-Robot Physical Interaction. In *2016 International Symposium on Experimental Robotics (Springer Proceedings in Advanced Robotics)*, Dana Kulić, Yoshihiko Nakamura, Oussama Khatib, and Gentiane Venture (Eds.). Springer International Publishing, Cham, 809–818. https://doi.org/10.1007/978-3-319-50115-4_70
[18] Yongguo Mei, Yung-Hsiang Lu, Y.C. Hu, and C.S.G. Lee. 2004. Energy-Efficient Motion Planning for Mobile Robots. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, Vol. 5. 4344–4349 Vol.5. https://doi.org/10.1109/ROBOT.2004.1302401
[19] Peter Ondrúška, Corina Gurău, Letizia Marchegiani, Chi Hay Tong, and Ingmar Posner. 2015. Scheduled Perception for Energy-Efficient Path Following. In *2015*

IEEE International Conference on Robotics and Automation (ICRA). 4799–4806. https://doi.org/10.1109/ICRA.2015.7139866

[20] R. Ortega, I. Mareels, A. J. van der Schaft, and B. Maschke. 2000. Energy Shaping Revisited. In *Proceedings of the 2000. IEEE International Conference on Control Applications. Conference Proceedings (Cat. No.00CH37162)*. 121–126. https://doi.org/10.1109/CCA.2000.897410

[21] R. Ortega, A. J. Van Der Schaft, I. Mareels, and B. Maschke. 2001. Putting Energy Back in Control. *IEEE Control Systems Magazine* 21, 2 (April 2001), 18–33. https://doi.org/10.1109/37.915398

[22] R. Ortega and M. W. Spong. 1988. Adaptive Motion Control of Rigid Robots: A Tutorial. In *Proceedings of the 27th IEEE Conference on Decision and Control*. 1575–1584 vol.2. https://doi.org/10.1109/CDC.1988.194594

[23] J.-E. Slotine. 1988. Putting Physics in Control-the Example of Robotics. *IEEE Control Systems Magazine* 8, 6 (Dec. 1988), 12–18. https://doi.org/10.1109/37.9164

[24] Stefano Stramigioli. 2015. Energy-Aware Robotics. In *Mathematical Control Theory I*. Springer, Cham, 37–50. https://doi.org/10.1007/978-3-319-20988-3_3

[25] Stefano Stramigioli, Cristian Secchi, Arjan van der Schaft, and Cesare Fantuzzi. 2005. Sampled Data Systems Passivity and Discrete Port-Hamiltonian Systems. *IEEE transactions on robotics and automation* 21, 4 (2005), 574–587. https://doi.org/10.1109/TRO.2004.842330

[26] T.S. Tadele. 2014. *Human-Friendly Robotic Manipulators: Safety and Performance Issues in Controller Design*. Ph. D. Dissertation. University of Twente. https://doi.org/10.3990/1.9789036537841

[27] T. S. Tadele, T. J. A. de Vries, and S. Stramigioli. 2014. Combining Energy and Power Based Safety Metrics in Controller Design for Domestic Robots. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 1209–1214. https://doi.org/10.1109/ICRA.2014.6907007

[28] Arjan van der Schaft. 2017. *L2-Gain and Passivity Techniques in Nonlinear Control* (third ed.). Springer International Publishing.

[29] G. Zamanakos, A. Seewald, H. S. Midtiby, and U. P. Schultz. 2020. Energy-Aware Design of Vision-Based Autonomous Tracking and Landing of a UAV. In *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*. 294–297. https://doi.org/10.1109/IRC.2020.00054