aerospace

MDPI

Article

Self-Organizing Control-Loop Recovery for Predictive Networked Formation Control of Fractionated Spacecraft

Florian Kempf [1,*], Julian Scharnagl [1], Stefan Heil [2] and Klaus Schilling [1]

[1] Zentrum für Telematik e.V., 97074 Wuerzburg, Germany
[2] Department of Computer Science, University of Wuerzburg, 97074 Wuerzburg, Germany
* Correspondence: florian.kempf@telematik-zentrum.de

Abstract: Going beyond the current trend of cooperating multiple small satellites we arrive at fractionated satellite architectures. Here the subsystems of all satellites directly self-organize and cooperate among themselves to achieve a common mission goal. Although this leads to a further increase of the advantages of the initial trend it also introduces new challenges, one of which is how to perform closed-loop control of a satellite over a network of subsystems. We present a two-fold approach to deal with the two main disturbances, data losses in the network and failure of the controller, in a networked predictive formation control scenario. To deal with data loss an event based networked model predictive control approach is extended to enable it to adapt to changing network conditions. The controller failure detection and compensation approach is tailored for a possibly large network of heterogeneous cooperating actuator- and controller nodes. The self-organized control task redistribution uses an auction-based methodology. It scales well with the number of nodes and allows to optimize for continuing good control performance despite the controller switch. The stability and smooth control behavior of our approach during a self-organized controller failure compensation while also being subject to data losses was demonstrated on a hardware testbed using as mission a formation control scenario.

Keywords: networked predictive control; fractionated spacecraft; controller failure recovery; self-organization; formation control; auction based task assignment

Citation: Kempf, F.; Scharnagl, J.; Heil, S.; Schilling, K. Self-Organizing Control-Loop Recovery for Predictive Networked Formation Control of Fractionated Spacecraft. *Aerospace* **2022**, *9*, 529. https://doi.org/10.3390/aerospace9100529

Academic Editor: Shuang Li

Received: 8 August 2022
Accepted: 10 September 2022
Published: 20 September 2022

1. Introduction

For a few years now the traditional space sector has been complimented by the new-space sector, which is known for short development and innovation cycles. With a focus on smaller satellites, Commercial off-the-Shelf (COTS) components and solid standardization like the Cubesat standard, this new space sector is still gaining more and more momentum and has already delivered notable successes beyond the scope of pure university research. In the next five years the main application areas are expected to be communication and Earth observation, with an expected annual growth rate of 26% of the Cubesat market and 19% for the whole small satellite market [1,2]. Contributing to the growth of this sector is the progress in manufacturing processes of miniaturized COTS satellite platform and payload components, which leads among other things to an ever increasing capability of small satellites. This allows these satellites to perform missions previously reserved only to much bigger satellites all at a fraction of the launch and satellite maintenance costs. Miniaturization and the utilization of COTS components, however, do not come for free, as they often lead to a decrease in single-component lifetime and resilience to external and internal disturbances. Examples are the higher susceptibility of micro chips with smaller process size to radiation [3] and the required increased rotation speed of momentum exchange devices and therefore lifetime reduction in comparison to their big satellite counterparts which have a higher moving mass. One way to account for these effects is to utilize the increased processing capabilities of modern COTS components by employing

adaptive or intelligent software that prevents or mitigates failures on the single component level. This can be realized for example by using the single component more efficiently over its lifetime or by detecting failures early on and then compensate for the failure through redundancy. The approach taken in this work can be attributed to the latter class. Another major key driver of the success in the new-space sector, apart from improvements on the component level, is the further extension of the mission scope of smaller satellites by employing one of the core principles of many new-space satellite applications: To utilize multiple smaller satellites in a cooperative fashion to achieve a common mission objective. By using multiple satellites cooperatively the mission key performance indices like ground coverage can be improved significantly while also improving on the typical desired "-ilities" of a space mission: Reliability, flexibility and maintainability.

Going beyond inter-satellite cooperation, projecting this trend into the future leads to a combination of inter- and intra-satellite cooperation. Here, taking further modularization and standardization of satellites subsystems into account, the components of a single- and multiple satellite system are generalized as independent wirelessly connected interacting functional nodes who are jointly working towards a common goal. By classifying the physical satellite platform components as either actuator-, sensor- or computing node according to their main role in the satellite system, the physical border between satellites becomes blurred and a bigger system of cooperatively interacting nodes emerges. Allowing both intra- and inter-satellite cooperation of functional satellite elements deepens the already existing advantages of today's multi-satellite structures while further extending them with an increased extend-ability, re-usability and robustness of the overall system [4–9]. An illustrative example of such a spacecraft equipped with an optical instrument as sensor and a thruster as actuator is sketched in Figure 1 and is used as reference architecture in the rest of the work.

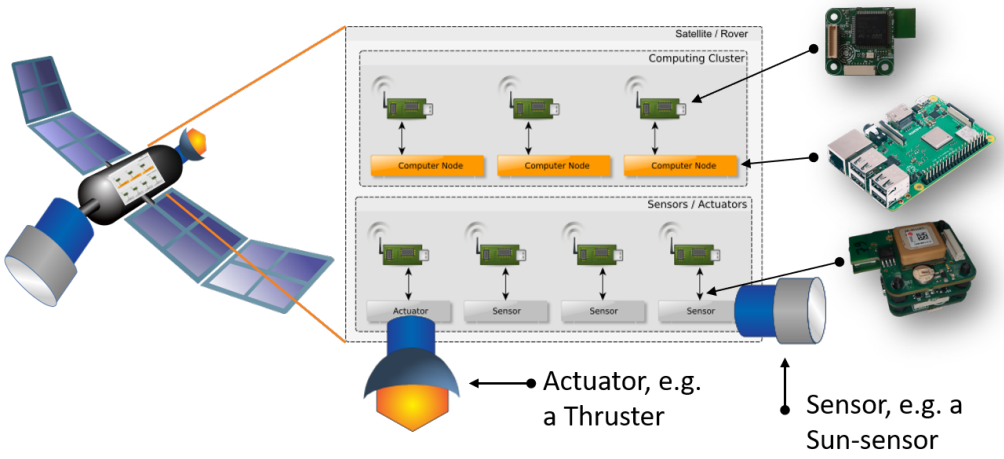


Figure 1. Sketch of a fractionated spacecraft consisting of cooperating, wirelessly connected sensor-, actuator- and computing hardware nodes. Sensor data acquisition is handled by sensor nodes, low-level driving of the actuators is handled by the actuator nodes and high level data processing and control is dealt with by the computing nodes.

Distributed spacecraft systems with the described architecture were pioneered by the F6 program of DARPA [10,11] which was unfortunately cancelled before its in-orbit demonstration in 2013 [12]. Distributed modular spacecraft systems, due to their apparent benefits, have nevertheless seen further continuation in several other projects both with a theoretical [5,13–17] and application focus [18,19]. Notable projects include the iBOSS initiative, that demonstrated the reconfiguration of the building blocks/nodes of a modular satellite with robotic manipulators on ground [20,21], the Innocube satellite mission, which is scheduled to launch in 2023 [22] and is expected to demonstrate the wireless harness of a modular satellite under space conditions [23] and finally the Yete-1 [5,24] and Yete-2

projects [25–27], which developed formation and attitude control for distributed spacecraft and constitute the precursor missions for the Tamariw project, which is scheduled to demonstrate these technologies with an in-orbit assembly of two Cubesats in 2025.

Despite the aforementioned advantages of such distributed spacecraft architectures, there are also several new challenges that these missions face. Especially the coordination and control of such a system of systems over a possibly unreliable and imperfect communication network poses a major task. It is therefore of utmost importance to have suitable control approaches for attitude and formation control, two of the most common control tasks in most missions, that are able to deal with the intra- and inter-satellite node network characteristics. The influencing characteristics of the node network on the control performance are mainly the delay and possible loss of information exchanged in the control loop. These effects can occur in the distributed control-loop on the network connection between sensor-node and controller-node and between controller-node and actuator-node. There already exist several control approaches which take one or more of these delay/loss scenarios into account. Specifically for wireless control networks there are recent developments to employ event-driven control to such systems to compensate for network effects and limit the required communication amount [28–30], some of them are already tailored for formation and attitude control of spacecraft [31–33]. A distributed robust h_∞ control approach where one controller node aboard one modular satellite controls the actuators of two satellites under the presence of intermittent sensor data losses was investigated in [34]. A further increase in control constraint capabilities while allowing information losses on the controller to actuator communication link in addition to sensor data losses was presented for model predictive attitude control in [25]. The development of optimal predictive formation control subject to sensor- and controller-data losses and delays was realized in [27].

All previous control approaches focus on how the controller node can compensate for the network effects such as information loss and delay and how it can optimize the single- or multi satellite performance. What happens however, if the controller node itself fails, either because of a hardware/software fault on the controller node or loss of connectivity of the controller node for a significant amount of time? This constitutes a central point of failure and can lead to partial or complete loss of a spacecraft for the mission objective. Therefore, in order to further increase the resilience and robustness of the fractionated multi-satellite architecture, a new approach to detect and then heal or mitigate the loss is needed. Here the intra- and inter-satellite cooperation aspect could be utilized by offloading the computational effort of the lost controller node unto another node in the system. The re-assignment process of control-authority should have several desired properties:

- It should keep the work load distribution in the intra- and inter- satellite node network optimal while still maintaining the required closed-loop performance of the original control process.
- It needs to keep the computational burden of the distribution low and scale with the number of possible node choices.
- In should allow multiple optimization objectives to take the heterogeneity of the nodes into account

The main contribution of this work is to propose a solution to the controller node failure problem via a distributed self-organized control reassignment process while realizing the aforementioned desired properties.

This is realized in a two step approach:

1. Robust detection of controller loss/failure
2. Intelligent re-assignment of controller responsibilities by performing a distributed multi-objective optimization using a robust control task auctioning procedure

As example control application formation control was chosen, building upon the networked formation control work in [27] combined with refined distribution optimization

objectives from [25]. The proposed solution is not limited to formation control and can be applied to other control objectives and applications like attitude control as well.

The rest of this work is structured as follows: First the employed test hardware, software and network architecture that is representative for a typical fractionated spacecraft system is introduced in Section 2.1. Followed by a description of the relative dynamics of the satellite formation as well as the Model Predictive Control (MPC) formulation. In Section 2.5 we discuss the required adaption rule of the MPC to the changing network characteristic. After the controller formulation we discuss the detection of controller failures in Section 2.6 followed by the description of the self-organizing controller failure recovery in Section 2.7. A first validation of our approach executed in a hardware testsetup is presented for pure network-loss in Section 3.2 and for additional controller failure in Section 3.3. Finally a conclusion and outlook is given in Section 4.

2. Materials and Methods

In this section the individual components of our main approach are described in more detail, both the theory as well as the hardware/software side. Regarding the common notation throughout this work: Vectors are lower-case and bold like $\mathbf{x}$, matrices are upper-case and bold like $\mathbf{A}$ and n-dots represent the n-th time derivation of a variable like $\dot{x}$.

2.1. SW Support Architecture

In order to understand how the individual sensor-, actuator-, and computing hardware nodes visualized in Figure 1 can mutually interact to build a single satellite-border transcending cooperating system, the SW architecture that has been used, albeit not required from theoretical point of view, is briefly introduced.

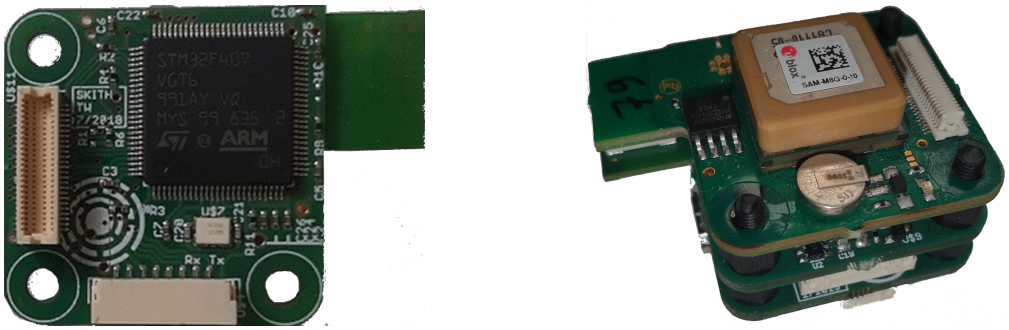
As mentioned previously, the fractionated architecture consists of hardware nodes which are very heterogeneous in nature, both regarding their primary use/objective and their hardware composition. Sensor and actuator nodes for example usually have very limited processing capability due to their primary task of reading and forwarding sensor data or, in case of an actuator, receiving input commands and processing it on the connected actuator hardware. This is in contrast to the computing nodes where sensor-data filtering, control command computation and many more computationally intensive tasks require hardware with strong computing capabilities. In addition, the means by which two nodes are connected in the node network might differ. Nodes might have a wired or wireless connection to the node network. Two nodes might communicate directly or need to rely on intermediary nodes that relay the exchanged data.

To account for the aforementioned peculiarities of such a multi-spacecraft node network, we rely on the RODOS middleware to provide an interface that abstracts the varying hardware and means of communication of the nodes. RODOS is specifically designed as software support infrastructure for heterogeneous distributed systems and serves a similar purpose as an operating system on PC hardware does. It runs on a multitude of hardware platforms ranging from small microcontrollers to PC grade hardware. RODOS takes care of multi-tasking, inter-process and inter-node communication via the publisher/subscriber principle, has a solid hardware-abstraction-layer and guarantees a robust information exchange between two nodes via flood-routing if they are reachable by any means in the network. Further information on RODOS and its architecture can be found in [5,35] and details on the robust data exchange, hot-plug capability and cooperation advantages of nodes in a spacecraft context are elaborated in [26,36].

2.2. Node Hardware

The sensor and actuator node hardware we use as low-power reference hardware in our work consists of a base board with a common STM32F4™microcontroller and a DWM1000™ultra-wide-band transceiver. The base board is extendable with various daughter-boards by stacking the boards together, thus specializing the node unit for a specific application, e.g., as sensor or actuator node. These so-called SKITH-boards were

developed for research applications in the domain of fractionated and distributed spacecraft and Unmanned Aerial Vehicle (UAV) by the Chair for Aerospace Information Technology of the University of Würzburg. The base board and an already extended unit with daughter boards specialized as GPS sensor node can be seen in Figure 2. Our reference computing nodes are ARM9™-Core based Raspberry PI-3™embedded PC boards equipped with WiFi and a direct connection to a SKITH-board with a DWM1000™ transceiver as relay.



(**a**) (**b**)

Figure 2. (**a**) The sensor/actuator HW base board with a STM32F4 microcontroller and a DWM1000 ultra-wideband transceiver. (**b**) Base board with daughter board extensions to form a GPS sensor node unit.

2.3. Node Network

Wired networking between the nodes is realized via the Controller-Area-Network (CAN) protocol and wireless networking is realized using a self-organizing adaptive Media Access Control (MAC) protocol supporting the dynamic addition and removal of nodes during run time while keeping collisions in the wireless channel to a minimum. The protocol is described in further detail in [37]. Although the ultra-wide band characteristics of the wireless communication hardware and the adaptive MAC protocol should reduce the risk of collisions on the wireless channel in the dynamically changing node network, packet/data loss due to collisions or interference are, apart from transmission delays, still one of the major disturbances in the network. In the case of identical independent distributed random loss events, the network can be modeled by a two-state Markov chain or Bernoulli process with the following state transition probabilities:

$$\begin{aligned} P(\theta(t_k) = 1 | \theta(t_{k-1}) = 1) &= p_l \\ P(\theta(t_k) = 1 | \theta(t_{k-1}) = 0) &= p_l \\ P(\theta(t_k) = 0 | \theta(t_{k-1}) = 1) &= 1 - p_l \\ P(\theta(t_k) = 0 | \theta(t_{k-1}) = 0) &= 1 - p_l \end{aligned} \tag{1}$$

For $\theta(t_k) = 1$ all packets are lost during $t \in [t_k, t_{k+1})$. The probability for a lost packet between two nodes is denoted with p_l and a successful transfer has a probability of $1 - p_l$. The loss probability is assumed to be independent of the current network state and the transition from one network state to another is equal to p_l and $1 - p_l$ respectively. This is visualized in Figure 3.

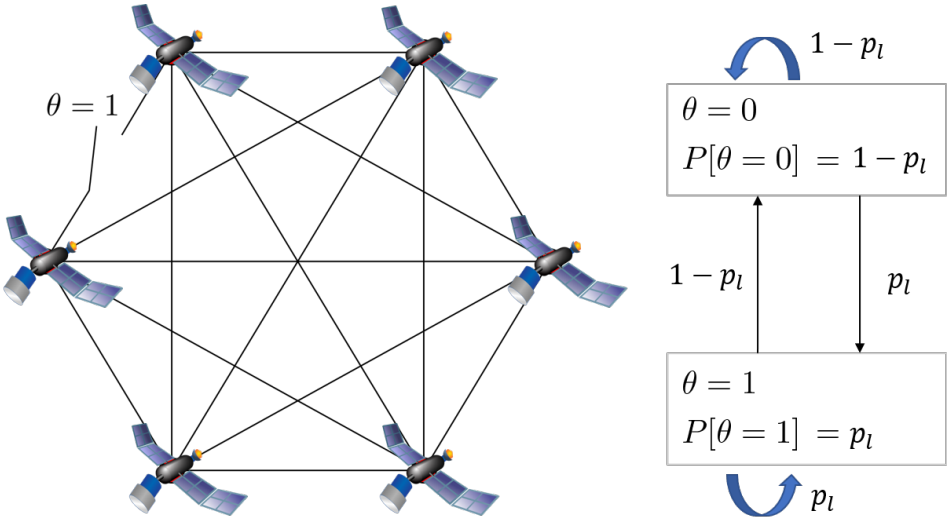


Figure 3. Node network modelled as two-state markov chain with equal loss probability p_l for both states θ.

The limited range of the wireless transceiver of about 100 m is denoted as d_{cmax}. Usually there exists a direct point-to-point link between two nodes in range and we model the distance between nodes as number of hops that a data packet must traverse until it reaches the endpoint as visualized in Figure 4. Each hop introduces processing-delays on the relay nodes and more hops could in general also be attributed to higher physical node distances. Therefore, the hop-distance is later taken into account as one objective that should be minimized in the control task distribution.

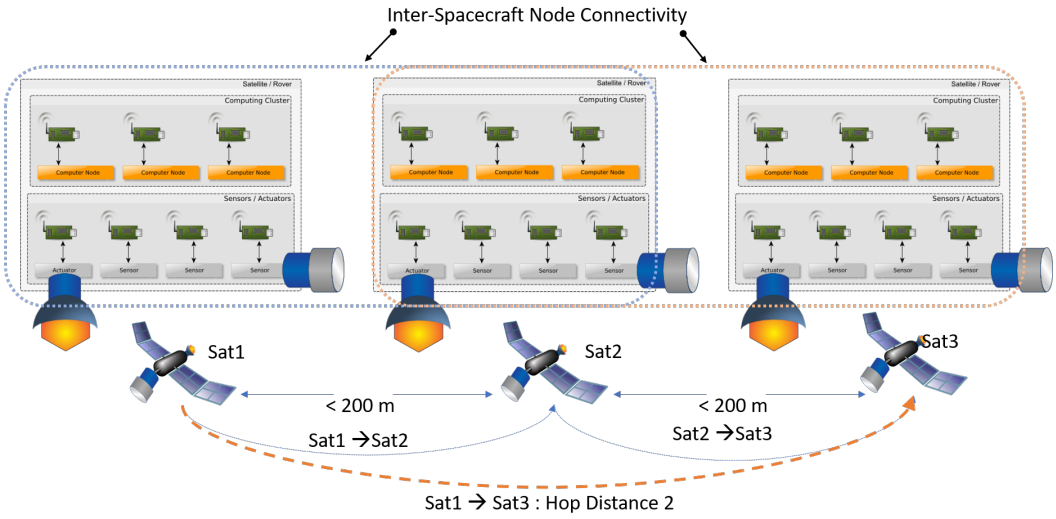


Figure 4. Multi hop data exchange between satellite nodes that are further apart.

Compensation for bounded delays in the network of the closed-loop control system could directly be incorporated in the controller design as demonstrated in [25] for attitude control and [27] for formation control. Strategies for dealing with short term packet loss in the closed-loop system are proposed as well in the aforementioned publications and as we are building upon these designs, bounded delays and short term packet loss are not further considered in the rest of this work.

2.4. Satellite Formation-Dynamic Model

The primary control objective that needs to be maintained in the presence controller loss is formation control of two or more satellites. In the following paragraph the dynamic model that describes the motion of the satellites in the formation is introduced.

The formation description utilized in this work is a so-called leader-follower approach, with one designated leader satellite S_0 also called chief and one or more follower satellites $S_{[1,2,...]}$, also called deputies. The chief satellite is uncontrolled within the formation and can be a real satellite or virtual reference point, whereas the deputy satellites are controlled to achieve a certain relative position and movement with respect to the chief.

The motion of each individual satellite can be described in an Earth-Centered Inertial (ECI) frame by the position vector $\mathbf{r_i}$ from the center of the Earth to the position of the satellite along its orbit and the velocity vector $\mathbf{v_i} = \dot{\mathbf{r}}_\mathbf{i}$. The motion of a single satellite S_i in Earth's gravitational potential with respect to the ECI reference frame can be described as follows [38]:

$$\ddot{\mathbf{r}}_\mathbf{i} = -\nabla U_g + \mathbf{F_{d,i}} + \mathbf{F_{u,i}} \quad \text{with} \quad \nabla U_g = \frac{\mu}{r_i^3}\mathbf{r_i} \tag{2}$$

where U_g is describing the Earth gravitational potential field assuming a perfect homogeneous mass distribution of Earth. $\mathbf{F_{d,i}}$ contains all disturbance forces per unit mass acting on satellite S_i, $\mathbf{F_{u,i}}$ is the controlled thrust force per unit mass and μ is the gravitational constant of the Earth.

The relative movement of the deputy satellites with respect to the chief S_0 can be described in a chief centered coordinate frame, centered at the position of S_0. This so-called Local-Vertical, Local-Horizontal (LVLH) frame is depicted in Figure 5 and its unit vectors are defined as follows: $\hat{X}$ is radially outwards, $\hat{Y}$ is in along-track/satellite velocity direction and $\hat{Z}$ is perpendicular to the orbital plane. With $\boldsymbol{\rho_i} = \mathbf{r_i} - \mathbf{r_0} = [x_i, y_i, z_i]^T$ as the relative position of the i-th deputy satellite and $\mathbf{r_0}$ as the osculating ECI position of the chief.

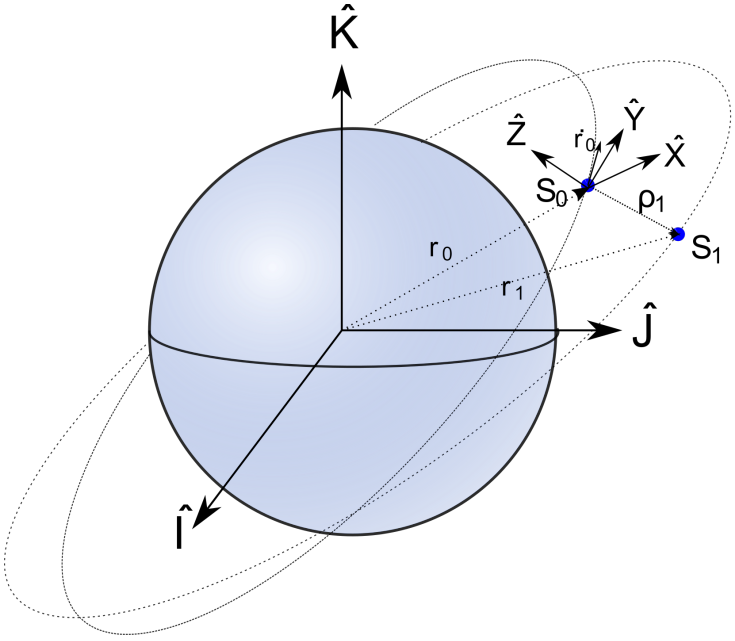


Figure 5. Local-Vertical, Local-Horizontal coordinate frame

The model of the relative dynamics that we use for the i-th satellite in the LVLH frame are the well known Hill–Clohessy–Wiltshire (HCW) Equations [39] given by:

$$\begin{aligned} \ddot{x}_i - 2n\dot{y}_i - 3n^2 x_i &= u_{i,x} \\ \ddot{y}_i + 2n\dot{x}_i &= u_{i,y} \\ \ddot{z}_i + n^2 z_i &= u_{i,z} \end{aligned} \tag{3}$$

with mean motion $n = \sqrt{\mu / r_0^3}$ and $\mathbf{u}_i = [u_{i,x}, u_{i,y}, u_{i,z}]^T$ as control input to the satellite.

They provide a Linear Time-Invariant (LTI) description of the non-linear relative satellite motion under several assumptions. First we assume a circular orbit of the real or virtual reference satellite S_0 which results in a constant mean motion n and norm of $\mathbf{r_0}$. As second assumption we consider a scenario with close proximity of the satellites so that inter-satellite node communication can be realized. This allows to neglect differential disturbance forces and other higher order un-modelled terms of the relative-distance satellite dynamics in the employed linear HCW formation model. These model simplifications pose no loss of generality of the proposed approach in this work as more complex non-linear relative-dynamic models could be incorporated in the same way by choosing an appropriate solver of the networked Optimal Control Problem (OCP). Indeed the GRAMPC solver utilized in this work is capable of solving non-linear OCP problems as well, which was demonstrated for a networked OCP with a non-linear formation dynamics model in [27]. However increasing model complexity would require significantly more computing power from the node hardware and is not necessary in close proximity scenarios like for example rendezvous-and-docking and joint-earth observation satellite formations. For the sake of keeping common nomenclature the controlled state of the i-th satellite is henceforth referred to as $\mathbf{x}_i = [x_i, y_i, z_i, vx_i, vy_i, vz_i]^T$ with $v\{x, y, z\}_i$ as the x,y,z velocity component of the state. Finally the state space representation of the HCW equations is given by:

$$\begin{aligned} \dot{\mathbf{x}}_i(t) &= \mathbf{A_i}\mathbf{x}_i(t) + \mathbf{B_i}\mathbf{u}(t) \\ \mathbf{A_i} &= \begin{bmatrix} & \mathbf{0}_{3x3} & & & \mathbf{I}_{3x3} & \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix} \\ \mathbf{B_i} &= \begin{bmatrix} \mathbf{0}_{3x3} \\ \mathbf{I}_{3x3} \end{bmatrix} \end{aligned} \tag{4}$$

2.5. Networked Model Predictive Formation Control

The control method that we rely on for controlling a satellite in the formation is a MPC approach. MPC not only allows to tune the controlled system behavior for either optimal fuel usage or reference tracking depending on the application, but also can include complex input- and state constraints in the controller formulation at the same time. MPC has not only seen wide adoption in terrestrial applications like autonomous driving [40–42] but in the spacecraft domain as well [43]. The ever growing processing power of mobile processors and efficient SW implementations also allow the generation of online solutions to the MPC optimal control problem in real-time for many dynamic systems [44]. In the case of our networked closed-loop system we also need to guarantee stability in the event of successive packet loss in the sensor-node to controller-node and/or controller-node to actuator-node communication. For that we build upon a specific MPC formulation from [25] which is an extension of the networked MPC definition from [45]. Here, in order to cope with sensor to controller losses, the internal nominal system model is used to compensate for lost system state updates by predicting the missing states based on the last received measured ones. Losses in the controller to actuator direction are handled by sending the whole computed control trajectory of length N_c of the MPC at each sampling interval to the actuator and storing it there so that in the event of one or multiple actuation

command losses the actuator-node executes the local stored control commands in an open-loop manner. One way to guarantee that the control trajectory $u(t_k : t_{k+N_c})$ of one control horizon of length N_c leads to a stable system behavior in an open-loop condition, is to ensure that for a given Lyapunov function $V(\mathbf{x})$ and a corresponding stable control law $h(\mathbf{x})$ the MPC computed control input $u(t_k : t_{k+N_c})$ leads to an equal or greater decrease of $V(\mathbf{x})$ over the control horizon time as the control input generated by $h(\mathbf{x})$ starting on the same initial conditions would lead to. The general networked MPC continuous time formulation is given below:

$$\begin{aligned}
&\min_{\mathbf{u}} \int_{t_k}^{t_{k+N}} [||\tilde{\mathbf{x}}(\tau)||_{\mathbf{Q}} + ||\mathbf{u}(\tau)||_{\mathbf{R}}] d\tau + F(\tilde{\mathbf{x}}(t_{k+N})) \\
&\text{s.t.} \quad \dot{\tilde{\mathbf{x}}}(t) = \mathbf{f}(\tilde{\mathbf{x}}(t), \mathbf{u}(t)) \\
&\qquad \dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), h(\hat{\mathbf{x}}(t_k + jT_s))) \\
&\qquad\qquad \forall t \in [t_k + jT_s, t_k + (j+1)T_s], \quad j \in [0, 1, ..., N-1] \\
&\qquad \mathbf{u}(t) \in \mathbb{U} \\
&\qquad \tilde{\mathbf{x}}(t) \in \mathbb{X} \\
&\qquad \hat{\mathbf{x}}(t_k) = \tilde{\mathbf{x}}(t_k) \\
&\qquad V(\tilde{\mathbf{x}}(t)) \leq V(\hat{\mathbf{x}}(t)), \quad \forall t \in [t_k, t_k + N_c T_s]
\end{aligned} \tag{5}$$

With $N \geq N_c$ as the prediction horizon length, N_c as the control horizon length, T_s as the control sample time, $\tilde{\mathbf{x}}$ as the propagated state subject to the MPC control inputs, $\hat{\mathbf{x}}$ as the propagated state subject to the Lyapunov stable controller inputs in a sample-and-hold fashion, $\mathbf{f}(\mathbf{x}, \mathbf{u})$ as the system dynamics, $\mathbb{U}$ as the allowed control input set and $\mathbb{X}$ as the admissible state set. The quadratic cost on the state is given by $||\mathbf{x}(t)||_{\mathbf{Q}} = \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)$ with $\mathbf{Q} = \mathbf{Q}^T$ and $\mathbf{Q} > 0$. The quadratic cost on the control input is given by $||\mathbf{u}(t)||_{\mathbf{R}} = \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)$ with $\mathbf{R} = \mathbf{R}^T$ and $\mathbf{R} > 0$. The final state cost penalty $F(\mathbf{x})$ that is often used to guarantee stability of the MPC formulation will be omitted as the Lyapunov constraint imposed on the state trajectory in the MPC formulation already serves as a stability guarantee.

Assuming a Bernoulli characteristic of the network model in Equation (1) the control horizon N_c must satisfy the following criteria, so that within the time interval $N_c T_s$ since the last successfully received packet at least one new control packet is transmitted and received successfully with probability p_s:

$$N_c \geq \frac{\log(1 - p_s)}{\log(p_l)} \tag{6}$$

The control horizon length N_j at time step j is regularly updated during run time so that the controller can react to changing network conditions. This is realized by adapting N_j whenever feedback from the actuator regarding its current state of the last updated control command buffer arrives at the controller node. Especially the number of already consumed/executed commands $N_{con,j}$ from the stored command buffer at the actuator plays an important role. N_j is then smoothly updated via an asymmetric Exponentially-Weighted-Moving-Average (EWMA) filter using the following update rule:

$$N_{j+1} = \begin{cases} \beta_{inc} \left(N_{min} + N_{con,j} + N_{th}\right) + \left(1 - \beta_{inc}\right) N_j & \text{if } N_{con,j} \geq N_{con,j-1} \\ \beta_{dec} \left(N_{min} + N_{con,j} + N_{th}\right) + \left(1 - \beta_{dec}\right) N_j & \text{if } N_{con,j} < N_{con,j-1} \end{cases} \tag{7}$$

By choosing $0 < \beta_{dec} < \beta_{inc} < 1$ a faster adaption of N to increasing data loss on the network than to decreasing data loss events is realized, therefore following a more conservative adaption strategy. The threshold constant $N_{th} \in \mathbb{N}$ is used to account for additional network uncertainty and increases the number of time steps until $N_{con,j}$ reaches the N_{min} command reserve. This command reserve is chosen, so that in case of controller

failure enough command reserve exists to facilitate a redistribution of the control task before the command buffer at the actuator runs out. The first control horizon length N_0 is set to $N_0 = N_{min} + N_{th} + N_{pl,ps}$ with N_{p_l,p_s} calculated according to Equation (6) using the expected network loss probability p_l and the desired probability p_s for at least one successful transfer from the controller before $N_{con,j}$ reaches the N_{min} command reserve. Results of this adaption rule are shown in Section 3.2.

To implement the MPC formulation given in Equation (5) on a digital controller, the problem needs to be reformulated as a discrete time problem with sample time *Ts*. In addition, the system dynamics $\mathbf{f}(\mathbf{x}, \mathbf{u})$ are realized as discrete state space model of the relative satellite motion described in Equation (4) by assuming a sample-and-hold application of the control input $\mathbf{u}_i$ to the system for the duration of the sample time T_s. In addition we assume the prediction horizon length N and control horizon length $N_c = N_j$ for time step j to be equal which does not impact stability as the Lyapunov stability constraint is independent of control and prediction horizon length [45]. This leads to the following discrete time MPC formulation with solution $\mathbf{u}^*_{0:N}$:

$$\begin{aligned} \min_{\mathbf{u}_{0:N} \in \mathbb{U}} & \sum_{k=0}^{N} \tilde{\mathbf{x}}_k^T \mathbf{Q} \tilde{\mathbf{x}}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \\ \text{s.t.} \quad & \tilde{\mathbf{x}}_{k+1} = \mathbf{A}_d\, \tilde{\mathbf{x}}_k + \mathbf{B}_d\, \mathbf{u}_k \\ & \hat{\mathbf{x}}_{k+1} = \mathbf{A}_d\, \hat{\mathbf{x}}_k + \mathbf{B}_d\, \mathbf{h}(\hat{\mathbf{x}}_k) \\ & \tilde{\mathbf{x}}_k \in \mathbb{X} \\ & \hat{\mathbf{x}}_0 = \tilde{\mathbf{x}}_0 \\ & V(\tilde{\mathbf{x}}_k) \leq V(\hat{\mathbf{x}}_k), \quad \forall k \in [0, ..., N] \end{aligned} \tag{8}$$

With $\mathbf{A}_d$ as system matrix and $\mathbf{B}_d$ as control input matrix of the discrete time LTI model of the relative satellite motion with sampling time T_s. The admissible control set $\mathbb{U}$ is defined as $\{\mathbf{u} \in \mathbb{R}^3 \mid ||\mathbf{u}||_2 \leq u_{max}\}$ and describes a ball of radius u_{max} around the origin. Solving the MPC formulation in Equation (8) not only provides a corresponding optimal control input sequence $\mathbf{u}^*_{0:N}$ as input to a specific actuator but also the estimated future state trajectory $\tilde{\mathbf{x}}_{0:N}$ of the associated satellite for the next N time steps. The final missing component for the networked MPC problem described in Equation (8) is a discrete-time Lyapunov stable controller $\mathbf{h}(\mathbf{x}_k)$ and its corresponding Lyapunov function $V(\mathbf{x}_k)$. There are several ways to get such a stabilizing controller given a LTI system. We reformulated the discrete MPC problem as discrete infinite horizon OCP with a constant gain feedback control law $\mathbf{h}(\mathbf{x}_k) = \mathbf{K}\, \mathbf{x}_k$. The gain $\mathbf{K}$ is then derived by solving the resulting Discrete-Time Algebraic Riccati Equation (DARE) and relaxing the control constraint set $\mathbb{U}$. Given a stabilizing control law $\mathbf{h}$ the corresponding Lyapunov function $V(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \mathbf{x}$ can be obtained by solving the Discrete Lyapunov Equation $\mathbf{A}_{cl}^T \mathbf{P} \mathbf{A}_{cl} - \mathbf{P} + \mathbf{W} = 0$ for $\mathbf{P}$ and $\mathbf{W}$ using a DARE again. The closed loop system matrix $\mathbf{A}_{cl}$ is given by $\mathbf{A}_{cl} = \mathbf{A}_d + \mathbf{B}_d\, \mathbf{K}$ and describes a stable closed loop system by design of the constant gain controller $\mathbf{h}$. Therefore both $\mathbf{P}$ and $\mathbf{W}$ are positive definite and $V(\mathbf{x}_k)$ is a valid Lyapunov function for the controller $\mathbf{h}$ [46]. To incorporate the control limits $\mathbf{h}$ is changed to $\mathbf{h}(\mathbf{x}_k) = \alpha \mathbf{K} \mathbf{x}_k$ with $\alpha \in \mathbb{R}^+$. Using α the gain $\mathbf{K}$ is scaled depending on the initial state $\mathbf{x}_0$, so that a decreasing $V(\mathbf{x}_k)$ over time step k ensures that $||\mathbf{h}||_2$ never exceeds u_{max}.

To generate an online solution to the MPC problem described in Equation (8) we integrated the non-linear gradient-based MPC solver GRAMPC as a task in the RODOS middleware running on the computing nodes. GRAMPC is a free software framework written in the C language capable of solving the MPC problem for parameterized linear or non-linear MPC problems with native support for linear inequality constraints. Due to its self contained nature it needs no external support libraries and has a small code size which makes it very suited for embedded applications. In addition online solution generation allows us to update required MPC parameters like the control horizon length N_c or the Lyapunov constraint during run time. Further advantages like fast speed or even

real-time capability are further described in [44]. The algorithmic side of GRAMPC is well documented in [47,48].

Now all required components necessary to facilitate predictive formation control that compensates for the effect of a distributed closed-loop network are realized. An overview of the wireless closed-loop system and the interaction of the nodes is shown in Figure 6. The MPC task running on the controller node generates a control input trajectory and sends it to the actuator, where a command execution task takes care to realize the control inputs according to their execution time by processing the received buffer one-by-one until new data arrives. In addition the actuator feeds back the current buffer state, e.g., the number of executed commands since the last update to the controller. This information is then used on the controller side to adapt the control horizon to compensate for the current network condition. The control performance and the network effects on the control horizon length over time are further discussed in Section 3.2.

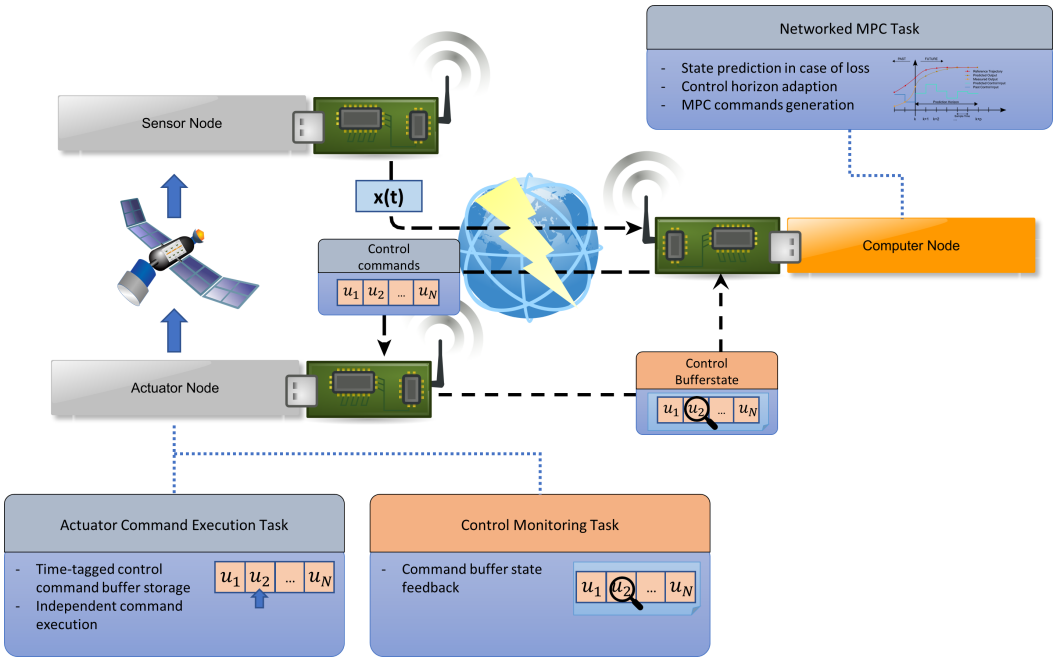


Figure 6. Diagram of the networked closed control loop with sensor-, actuator- and controller node. The necessary tasks running on each node and the corresponding exchanged data packets are also visualized.

The next step now is to compensate not only for intermediary losses on the network, but also to recover from a complete loss of a controller node. This is realized in two steps:

1. Robust detection of controller loss/failure
2. Recovery by intelligent re-assigning a new controller node to perform the original controller's task

2.6. Controller Node Failure-Detection

The first step is the reliable detection if a controller node has failed. This is realized on the actuator side by keeping track of the number of already consumed commands $N_{con,j}$ in the control command buffer at time step j. Under nominal conditions the length N_j of control commands generated at time step j by the controller guarantees that the command reserve at the actuator is always kept around a certain desired level N_{min}. However once $N_j - N_{con,j}$ is less than N_{min} we either have a severe connection problem between controller- and actuator node or the controller node failed completely. In both cases we consider the controller node as faulty and initiate a control node recovery. In case the network is sensitive to bigger single data transfers like the control command list sent by the controller,

regular small size beacons sent from the controller to its corresponding actuator node could also be used to identify controller node failure on the actuator side.

2.7. Controller Node Failure-Recovery

Once a controller node failure is detected by the actuator node a recovery procedure is initiated. The aim of the recovery procedure is to intelligently identify the most suitable candidate among all available inter- and intra-satellite computing nodes to take over the control task of the failed controller node. As mentioned in the introduction the recovery procedure should meet certain design criteria:

DC1: It should scale with the number of nodes in the network
DC2: Candidate selection should put a low processing burden on the computing power limited initiator node
DC3: Candidate selection should not take longer than $N_m in\ T_s$ to avoid an empty command buffer at the actuator node.
DC4: Required communication should be kept minimal
DC5: The possibility to include multiple objectives in the candidate selection

While meeting the primary objectives of the new control assignment:

AO1: High Quality-of-Service (QoS) of the new controller for the control-loop
AO2: Avoidance of unnecessary burden on the network during control execution
AO3: Avoidance of single-point of failures and unbalanced load on the computing nodes in the network

The problem of re-allocation of the control task to a new controller node falls under the domain of task allocation in a multi-agent cooperative setting and is a problem in the field of classical Artificial Intelligence (AI). In general allocating the control task while also taking the m-tasks already running on the n-available computing nodes into account leads to the optimal distribution of (m+1) tasks among the n nodes, which is a combinatorial General-Assignment-Problem (GAP) and is NP hard [49].

2.7.1. Auction Based Task Allocation

To satisfy design criteria DC1-3, e.g., low computational complexity on the allocating node, good scalability with the number of computing nodes and a deterministic allocation time almost independent of the node number, we decided to choose an auction-based sequential single task allocation approach [50–54]. In the auction-based sequential single task allocation the initiator or auctioneer, in our case the actuator node, offers a single task to prospective bidders. It does this by sending a task-auction initiation message containing the task context via broadcast to all reachable bidders, i.e., computing nodes. The bidders then locally calculate their suitability for the task expressed as a single scalar value or vector of scalar values. For the computation, they only use the provided task context and local available information. Subsequently, they send the calculated values back as bid to the auctioneer. The auctioneer then selects the winning bid using a simple selection strategy, e.g., selecting the lowest value and announces the corresponding winner thereby ending the auction process. Design criterion DC4 to have low network load is also satisfied, because in the best case, e.g., no network loss, the bidders communicate only once placing their bids, while the auctioneer communicates twice, once to announce the auction and once to announce the winner. Mitigation procedures for communication losses are discussed later, which require a few more communication events, but the overall burden on the network is still very limited. Finally, design criterion DC5, the possibility for multiple objectives driving the allocation choice is also realized by encoding each objective as appropriately scaled scalar value and fusing all of them together, e.g., with a weighted sum, to a single bid value.

The assignment objectives AO1-3 lead to the two following main selection criteria that we consider important: Proximity to the actuator node and computational load of the controller candidate node.

2.7.2. Proximity Criterion

First, proximity of the controller candidate to the actuator node. The controller to sensor-node proximity is disregarded because we assume that the sensor-node and actuator node are part of the same spacecraft. Proximity is measured in number of network hops which describes how many individual physical links need to be traversed until data sent from one node reaches its destination node. Links directly connected to each other, e.g., via a wireless link, have a network hop distance of one. Measuring node distance in terms of hops is an efficient and convenient way to encode both physical distance and routing effort in one value. The implicit encoding of physical distance emerges because of the limited communication distance of the wireless transceiver hardware and the otherwise fully connected network topology of all nodes in range. That means nodes farther apart than the maximal transceiver distance have to communicate through other nodes acting as relays therefore increasing the hop count of exchanged data by the number of relays taken. This is of course a rough estimation and simplification of physical distance, but serves out purpose well in this case. The hop count value between the auctioneer node $n1$ and bidder node $n2$ in the network at time step k denoted as $\#hop_k(n1, n2)$ can be easily retrieved by the bidder upon receiving the auction initiation message from the network layer of RODOS at the k-th time step. On the other hand, an estimation of the lower-bound of the expected hop count based on the node distance between node $n1$ and node $n2$ at time step k is given by:

$$\#\hat{hop}_{k,min}(n1, n2) = \left\lfloor \frac{||\mathbf{x}_{k,n1} - \mathbf{x}_{k,n2}||_2}{d_{cmax}} \right\rfloor \tag{9}$$

with $||\mathbf{x}_{k,n1} - \mathbf{x}_{k,n2}||_2$ as Euclidean distance between the node $n1$ and node $n2$ position states and d_{cmax} as the maximum transceiver transmission distance. Optimizing for low hop counts reduces closed-loop latency and network loss probability. The former is mainly influenced by data processing on the relay nodes and the latter by the number of physical links, where most data losses happen. This optimization criterion therefore leads to better QoS (AO1) for the closed-loop in terms of delay and loss and less routing overhead and travel distance in the network, therefore keeping the network load low (AO2). To map the hop count $\#hop_k(n1, n2)$ between node $n1$ and node $n2$ to a scalar cost Cd_k we utilize the following cumulative cost function:

$$Cd_k(n1, n2) := \left\lfloor \alpha_d \left(\#hop_k(n1, n2)^2 + \#hop_k(n1, n2)\right) \right\rfloor + \beta_d \tag{10}$$

With the scaling factor $\alpha_d = 0.5$, $Cd_k(n1, n2)$ becomes $Cd_k(n1, n2) = \beta_d + \sum_{i=1}^{\#hop_k} i$, which penalizes hop-counts in a non-linear fashion. The cost function is depicted in Figure 7. We opted for a non-linear penalization of hops, in order to account for the increasing network load induced by communication over a larger number of hops. The value $\beta_d = 2$ is a bias value that weighs the hop-distance criterion with regard to the other criteria used in the fused scalar sum discussed later.

By utilizing the projected future position states $\tilde{\mathbf{x}}_{n1/2,k:k+N}$ of each satellite associated with one of the nodes $n1$, $n2$ computed as part of the last MPC solution and sent int the task context, the instantaneous distance cost $Cd_k(n1, n2)$ can be extended to a distance cost sequence of length m:

$$\begin{aligned} Cd_{k:k+m}(n1, n2) = [&Cd_k(n1, n2), \hat{Cd}_{k+1}(n1, n2), ..., \hat{Cd}_{k+m}(n1, n2)], \\ &\text{with } m \in [0, ..., N] \end{aligned} \tag{11}$$

where $\hat{Cd}_{k+i}$ is the future expected distance cost at time $k + i$ which uses the expected hop count $\#\hat{hop}_{k+i,min}(n1, n2)$ instead of the measured one. If the controller/ bidder node

$n2$ is located on the same satellite as the actuator/ auctioneer node $n1$, Cd_k reduces to $Cd_k(n1, n2) := \beta_d, \forall k$.

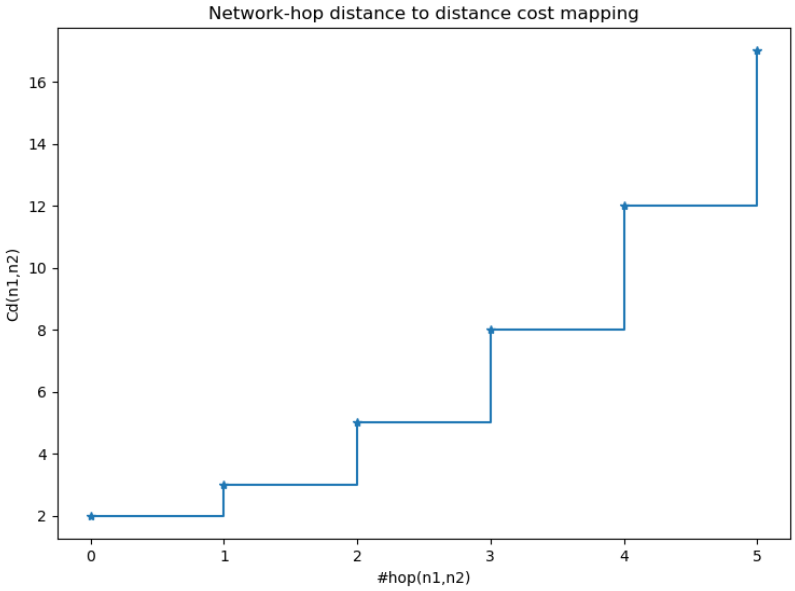


Figure 7. Network hop count between two nodes $n1$, $n2$ to distance cost mapping for the bid with $\beta_d = 2$.

2.7.3. Load Criterion

The second selection criterion is the prospective computational load of the controller candidate after it has taken over the new control task. Taking the prospective load into account in control assignment optimizes the control task distribution for an even load among all prospective candidates. This should ensure that a node is not overburdened with the new task, therefore satisfying the QoS objective AO1 and at the same time avoiding single-point of failures serving objective AO2.

The prospective load on a node mainly depends on the following factors:

1. The way in which the operating system running on the node schedules the individual tasks
2. The characteristics and run times of other tasks already running on the node
3. The hardware processing capability of the node

The middle-ware/operating system we are using in our experiments is RODOS, which utilizes a round-robin task scheduling approach. Using the measured average runtime share $L_{idle} \in [0, 1]$ of the idle task running on the node for a fixed time ΔTl and the empirical determined estimated runtime $\hat{T}_{ctrl}$ of the new control task we can estimate the expected runtime share of the idle task $L_{idle,+}$ after the acquisition of the new control task:

$$L_{idle,+} = L_{idle} - \hat{T}_{ctrl}\, f_{ctrl} \tag{12}$$

With $f_{ctrl} = \frac{1}{Ts}$ as the task frequency of the control task. The estimated task runtime $\hat{T}_{ctrl}$ is visualized for different MPC control horizon lengths in Figure 8.

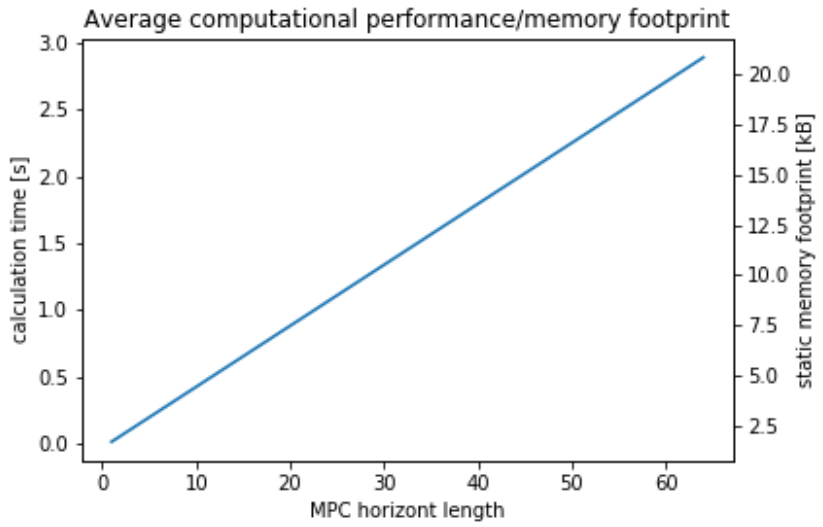


Figure 8. Empirically determined average runtime $\hat{T}_{ctrl}$ of the MPC control task for different control horizon lengths running on the STM32F4 node.

If $L_{idle,+}$ is less than zero the node would be overburdened by the new control task and the bid is sent back marked as infeasible. Otherwise the appropriately scaled cost of computational load Cl is then computed by:

$$Cl = e^{15\,(1-L_{idle,+})-13.5} - e^{-15\,(1-L_{idle,+})} + 1 \tag{13}$$

Equation (13) maps the occupied computational resources after the new task acquisition denoted by $1 - L_{idle,+}$ to the load cost by exponentially penalizing high loads ($L_{idle,+} < 0.2$) and keeping costs on an almost even level for idle loads which have a similar effect on the task performance ($0.2 \geq L_{idle,+} < 0.8$). The effect of this mapping is visualized in Figure 9.

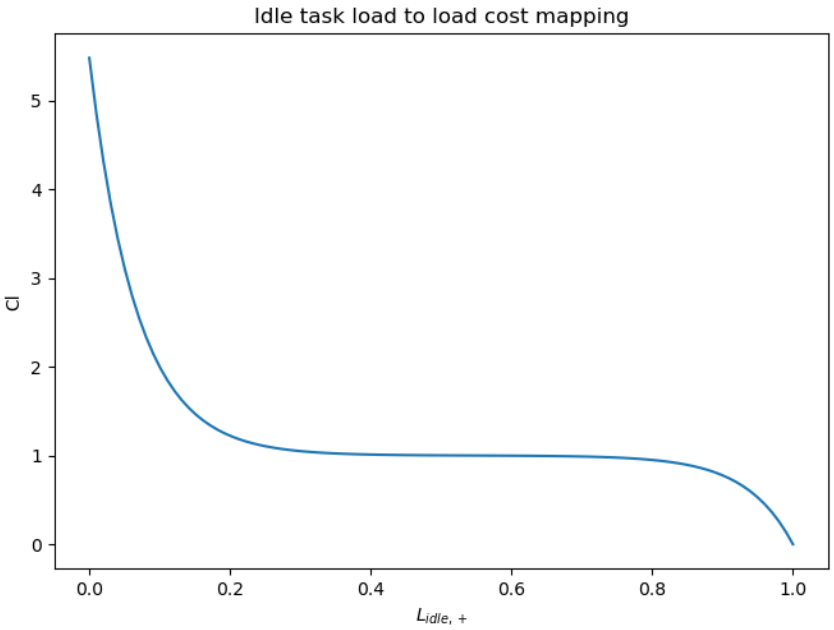


Figure 9. Mapping from free computing load after task acquisition $L_{idle,+}$ to bid load cost Cl.

Finally both the distance cost $Cd_{k:k+m}$ and the load cost Cl are fused to form the bid **Cg** of a candidate node:

$$\mathbf{Cg} = Cl\,\mathbf{Cd} \tag{14}$$

With $\mathbf{Cd} \in \mathbb{R}_{1xm}$ as the distance cost sequence of length m in row vector representation.

2.7.4. Auction Sequence

The whole auction sequence is visualized in Figure 10 and follows the process described in Section 2.7.1. Losses over the network are either compensated by repeated message broadcasts for some time, e.g., the task offer message during the auction announcement period, or are guarded with explicit acknowledgement (ack) messages, which trigger a resend of the original message if no ack message was received from the recipient for some time.

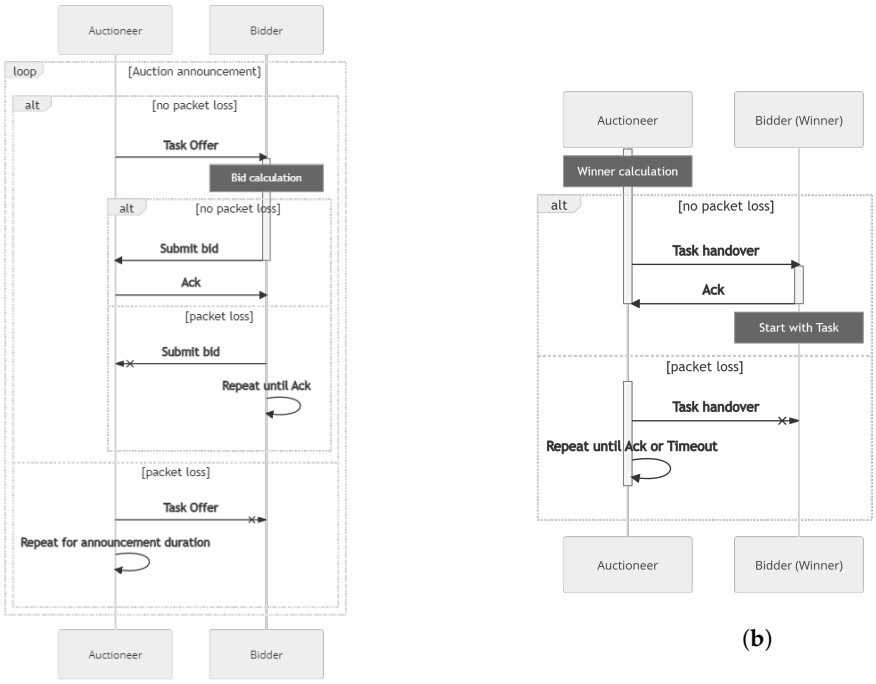


Figure 10. (**a**) The communication sequence between auctioneer and bidders for the auction announcement and bid reception phase. (**b**) The communication sequence between auctioneer and bidders for the winner announcement and task handover phase.

2.7.5. Optimality of Control Task Assignment

Although this kind of sequential single task allocation is optimal with regard to the chosen assignment objectives and a single task at a single time instance it might lead to a suboptimal distribution over time if all already distributed tasks are also taken into account. An example is given in Figure 11, where after the first failure event of the controller on satellite S2, the controller node of satellite S3 wins the bid and takes over the control of the actuator on satellite S2. The next failure event of the controller on satellite S4 then forces the controller of satellite S1 to take over, because it's the only one with free capacity. This leads to a suboptimal distribution in terms of the inter-node closed-loop proximity of all nodes, which would be four hops in total (#hop(S1<->S4) + #hop(S2<->S3) $= 3 + 1$) as compared to two hops in an optimal distribution where satellites are controlled by their next neighbour (#hop(S1<->S2) + #hop(S2<->S3) $= 1 + 1$).

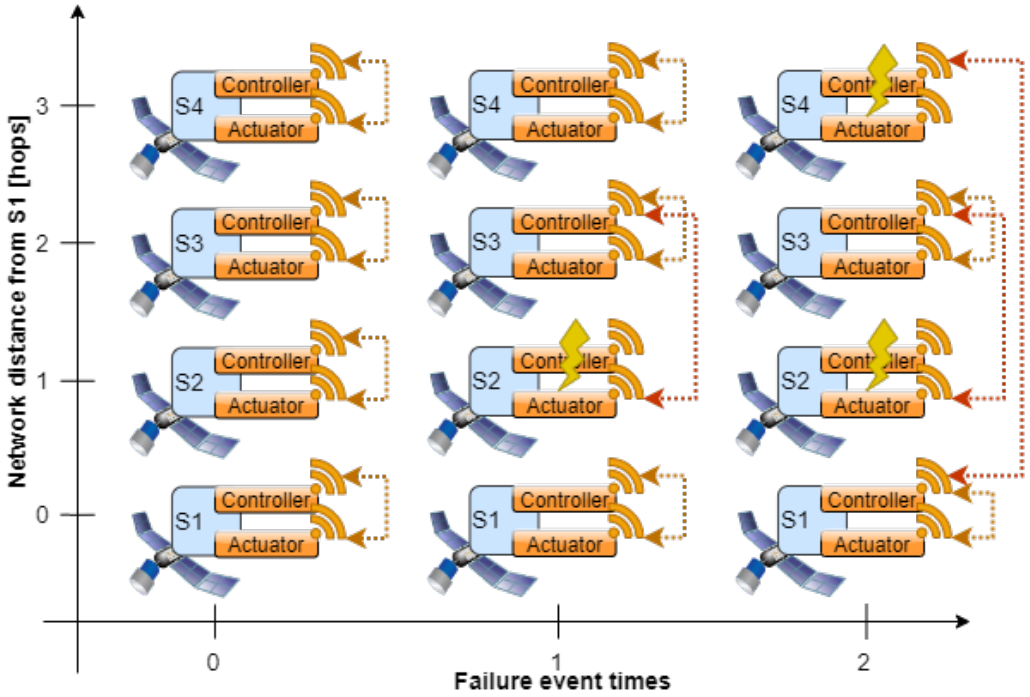


Figure 11. Suboptimal control task distribution with sequential-single-step allocation. The red dotted arrows represent the re-allocated control relationship. The last distribution could be optimized by having each of the satellites take care of the control task of its failed direct neighbour.

There exist several approaches in the literature that cope with the problem of optimality in auctions [50,55]. In a Multi-Bid approach [56] bids for each already distributed task including the new task are collected from all bidders at the auctioneer which optimizes over this bid set, e.g., via Mixed-Integer-Programming. Another common approach is to pool multiple dependent tasks together, both unassigned and already assigned ones and auction subsets of them as one package. Considering that at any time we only need distribute a single new task in a timely manner collecting and optimizing over a possibly growing number of tasks at the actuator node is not efficient. In addition the network load would increase a lot which is not desirable in low-power low-latency space applications. The other option to have the bidders bid on task pools or subsets of tasks ultimately leads to a combinatorial task distribution once optimality should be assured, which scales exponentially with the number of tasks and bidders [51]. Therefore to keep the computational burden at the actuator node low and to profit from the polynomial scaling of sequential single task auctions we propose an extension to the standard single task auction process. In addition to the bid for the current task, a bidder also transmits a second bid under the assumption of one of its previously assigned tasks being released and the cost penalty for releasing its already assigned task. If the difference between the cost including release and the second best bid is larger than the penalty associated with the task release the auctioneer selects the corresponding bidder. The bidder then retires from the execution of its tasks selected for release, thereby triggering their redistribution. To be able to submit the penalty of task release, the information transmitted during task assignment is extended by the cost of the second best bidder. This re-assignment approach solves the suboptimal assignment shown in Figure 11 however edge cases still exist for which the extended auction algorithm produces suboptimal solutions. This is however regarded as acceptable considering the network load and computational benefits of the approach.

The extended closed-loop networked control architecture is summarized in Figure 12 which extends the original networked control architecture from Figure 6 with the auction and failure detection related tasks on actuator and controller side.

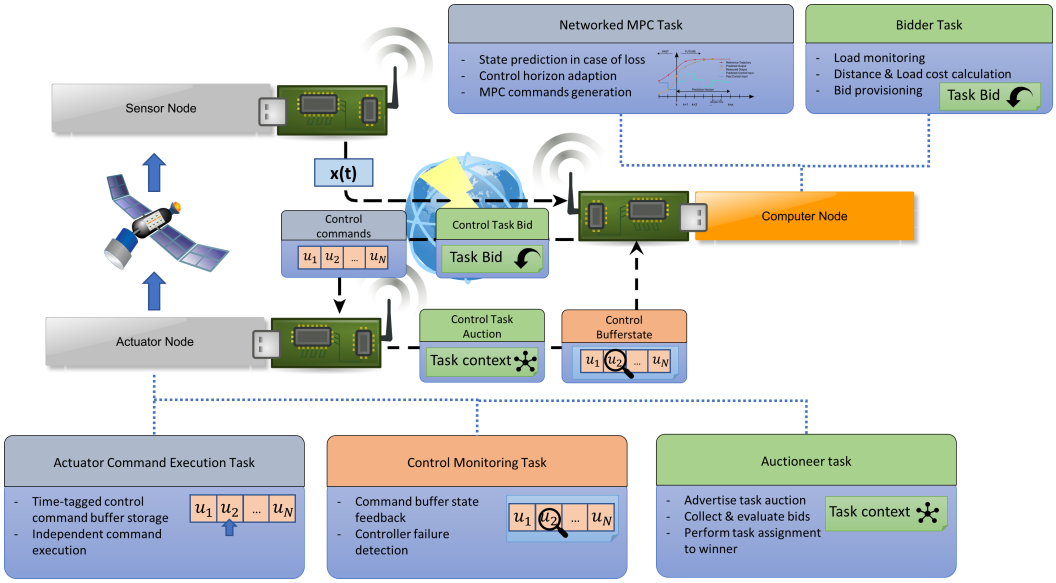


Figure 12. Diagram of the networked control loop architecture extended to deal with controller losses.

3. Results

3.1. Hardware Testsetup

In this section the hardware test setup we utilize for our experiments is introduced. It is shown in Figure 13 and represents a basic two 1-U satellite setup. Both units are joined physically and are sharing the same Electric Power Supply (EPS) unit for simplicity reasons. The upper unit contains three wireless connected STM32F4 SKITH nodes, which are described in Section 2.2. The SKITH nodes can be reconfigured depending on the scenario to run either exclusively actuator-, sensor- or controller SW units, acting in exclusive node mode or to run a multi-purpose mix of SW units if the node is configured as a combined node. If the node is in exclusive mode, the node represents the single type of SW unit it is running, e.g., it represents an actuator node if it runs an actuator SW unit. In combined node configuration the otherwise external physical network used for the data exchange between the SW units of different type running on the node is simulated using the Markovian network model introduced in Section 2.3. Using the publisher/subscriber and Hardware-Abstraction-Layer (HAL) architecture of RODOS allows a transparent transfer of a SW unit from an exclusive to a combined HW node. Combined nodes have the advantage that multiple SW units, e.g., multiple controllers, actuators and sensors could be run on a single HW with more control over the performance influencing parameters like the network between those units. However this advantage comes at the cost of higher load on the HW and influence of network modelling errors. The lower unit contains the EPS used to power all HW nodes and another two configurable nodes that are wirelessly connected with the nodes in the upper unit. The two units are stacked on an electrical interface board which allows monitoring of all HW involved from an external PC.

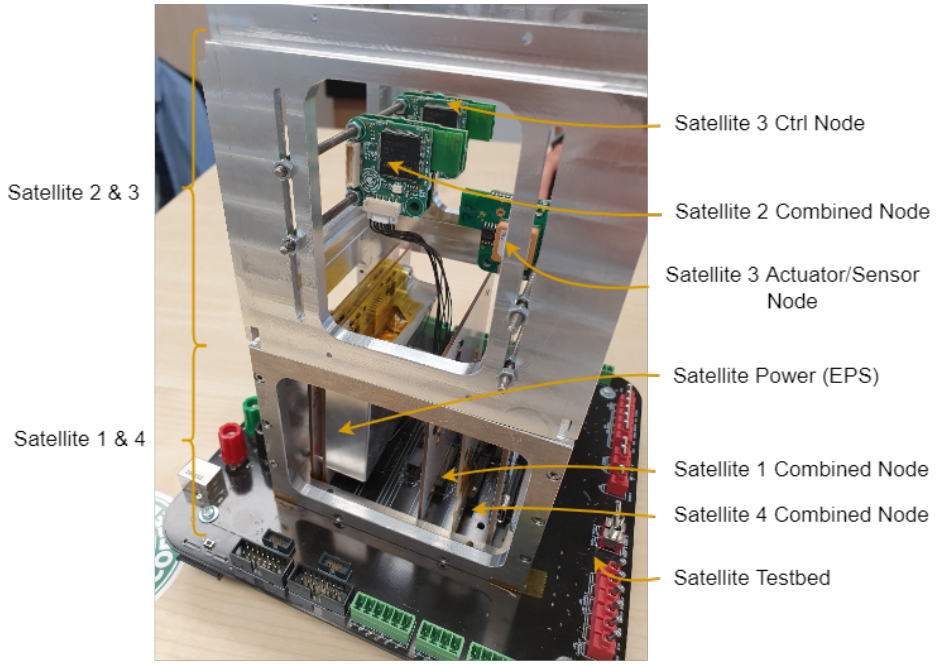


Figure 13. Hardware testbed used for the experiments: Two cube units each containing one or more connected hardware nodes which are associated with one of four satellites. The units are stacked on an electrical test interface board used for external monitoring.

3.2. Control Performance under Nominal Conditions

In the first scenario the control performance subject to network influences is evaluated to establish a baseline for the results once controller failure occur in scenario two. The satellite under observation is satellite 3 represented by two SKITH nodes, an exclusive controller which runs the MPC control task and a combined actuator/sensor node. Initially all satellites are in an along-track formation on the same orbit, except Satellite 3 which has an initial state $\mathbf{x}_{0,S3}$ with a deviation of 10 m in in-plane ($\hat{X}$), along-track ($\hat{Y}$) and out-of-plane ($\hat{Z}$) direction from its desired reference position which is also the origin of the moving Hill coordinate frame. This is visualized in Figure 14. The along-track separation between two adjacent satellites is less than d_{cmax} and bigger than d_{cmax} for all satellites that not direct neighbours along the orbit. This leads to communication hop-counts of one for direct satellite neighbours and bigger than one for non-direct neighbours.

The control goal is therefore to drive satellite 3 to it reference, the moving coordinate system origin $[0, 0, 0]^T$. The networked control loop is run subject to a Markovian network loss, following the network model introduced in Section 2.3 with $p_l = 0.3$. In addition the minimal control horizon length N_{min} is set to 15, the horizon threshold N_{th} is set to 10 at a control cycle duration of $Ts = 6s$. The control input buffer size on the actuator side during the maneuver is shown in Figure 15 where multiple network losses of the control command sequence from the controller to the actuator are visible. Whenever a successful transmission is achieved the control input buffer on the actuator is replenished. In addition an increase of the control input horizon length is visible as a result of the adaption to the network losses, keeping the remaining control input buffer at the actuator well above $26 > N_{min} + N_{th}$ remaining control inputs. The position state evolution of satellite 3 subject to the actuated control inputs is shown in Figure 16. The networked predictive controller drives the system to the desired state $[0, 0, 0]^T$ and exhibits stable performance. During a prolonged open-loop execution with multiple successive packet-losses and while the system is still away from its equilibrium condition a set-point deviation can be seen (t = 80 s until t = 110 s) which is corrected once new command trajectories reach the actuator (t = 110 s until t = 130 s).

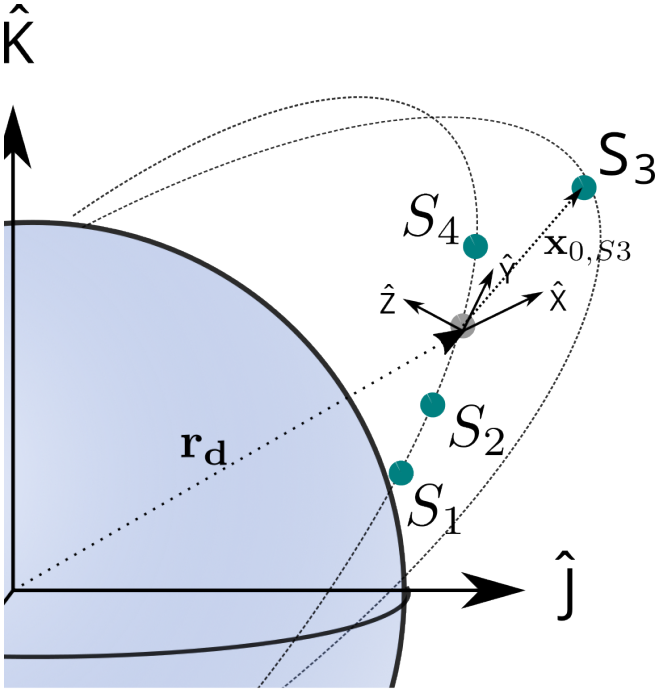


Figure 14. Initial state of all four satellites with the moving Hill coordinate frame centered at the desired position of satellite 3 in the formation.

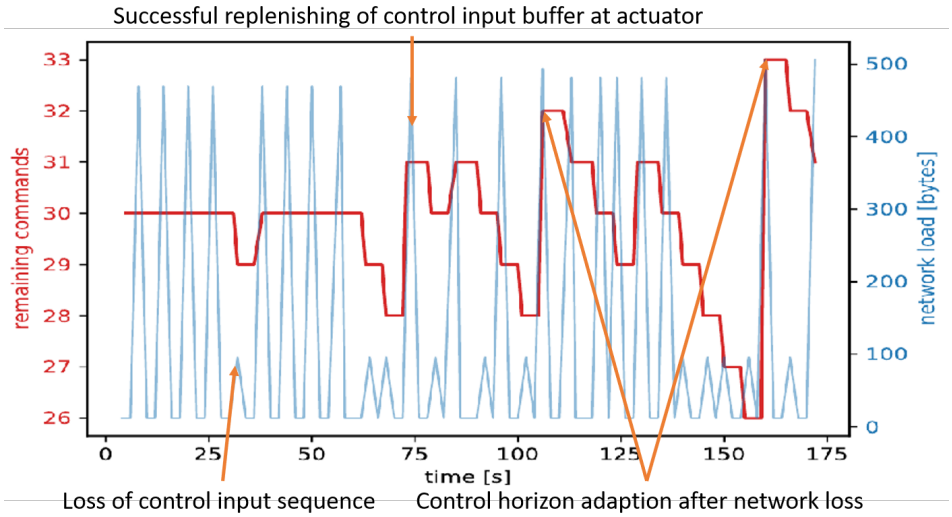


Figure 15. Command buffer state on actuator side under the influence of packet losses. The adaption of the control horizon length as reaction to the measured losses prevents the remaining control input buffer size to fall below $N_{min} + N_{th} = 25$.

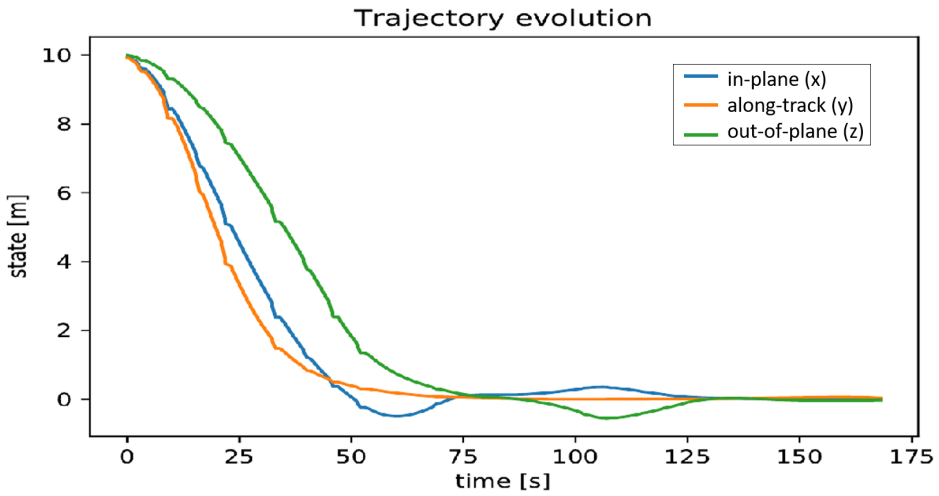


Figure 16. Control performance of the networked formation control of satellite 3 over a network with Markovian network characteristics without controller failures.

3.3. Controller Failure with Reassignment

Now a similar scenario to the previous one is considered. Satellite 3 starts now at position $[20, -10, 15]\ m$ and should be driven again to $[0, 0, 0]\ m$. During the control process the controller node belonging to satellite 3 fails at $t = 35$ s leading to a continuing depletion of the control input buffer at the actuator of satellite 3 which is shown in Figure 17. Once the remaining stored commands fall below $N_{min} = 25$, the controller failure is detected by the actuator and a control task re-assignment is initiated by the actuator node via the described auction procedure at $t = 50$ s.

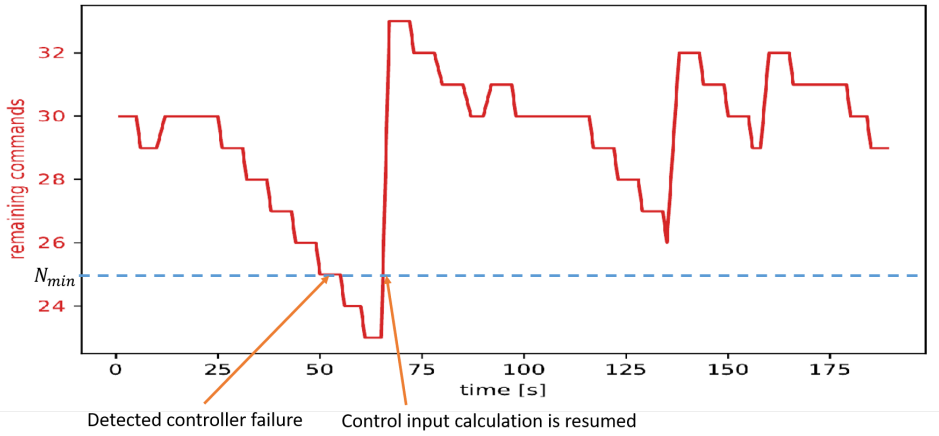


Figure 17. Command buffer state on actuator side under the influence of packet losses. In addition at t = 35 s the currently assigned controller node fails and a control task reassignment is initiated once the remaining control input buffer size to falls below N_{min}. After the task is reassigned the control input buffer at the actuator is filled again.

For the bid calculation the upper control horizon length is set to 40, resulting in a maximum MPC task execution time of $\hat{T}_{ctrl} = 2.5$ s and together with the desired control frequency $f_{ctrl} = \frac{1}{Ts} = \frac{1}{6\,s}$ leads to a maximum estimated task load of $L_{ctrl} = 42\%$. In addition inter-satellite proximity expressed in hops is not expected to be changing during the maneuver , therefore the distance cost $Cd_{0:m}$ is assumed to be constant and is treated as a scalar value Cd of dimension one. The scaling factors for the distance cost Cd are $\alpha_d = 0.5$ and $\beta_d = 2$. This results in the submission of task bids of combined node 2 of satellite 2 and combined node 4 of satellite 4. The combined node of satellite 1 did not submit a bid,

because its expected load after the task acquisition exceeds its capacity. The individual costs and bids are summarized in Table 1.

Table 1. Individual cost and fused bid values for all candidate nodes of the satellites.

Node of Sat	Free Load	Hop Distance to S3	Load Cost Cl	Distance Cost Cd	Bid
1	25%	2	5.4	5	No Bid
2	50%	1	2.35	3	7
4	45%	1	3.86	3	11

After multiple auction re-announcements the actuator node of satellite 3 closes the auction at $t = 60$ s and selects the node with the smallest bid, the combined node of satellite 2, to take over the control task. After one loss event during the winner announcement the assignment is re-transmitted by the actuator and is acknowledged by the winner node. At time $t = 61$ s the control input sequence calculation is resumed and the control input buffer at the actuator is filled again. For the remaining time of the scenario the remaining control input buffer length is mainly influenced by the data loss in the network. The overall control performance is shown in Figure 18. Like before the controller manages to drive the system to its set-point and exhibits stable closed-loop behavior even under the effect of the failure of the initial controller and a necessary control task reassignment. This is smooth control task handover is realized because the remaining control input buffer of length N_{min} successfully bridges the time until the control task is assigned to a new controller.

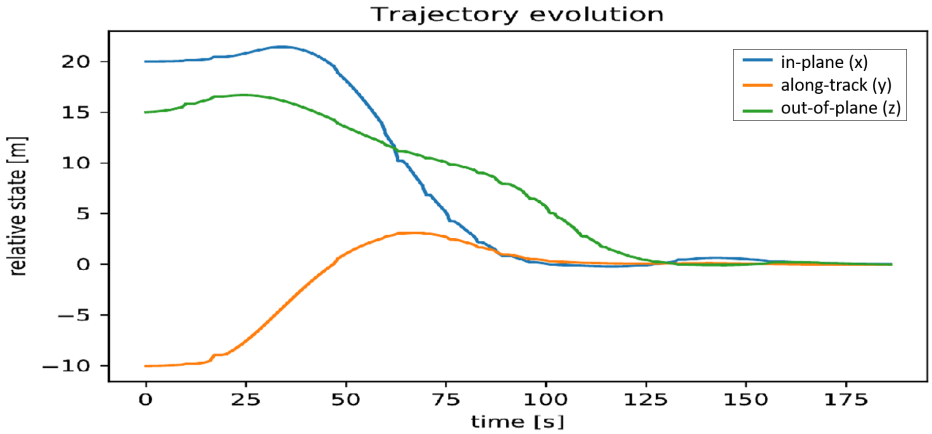


Figure 18. Control performance of the networked formation control of satellite 3 over a network with Markovian network characteristics with failure of the controller node of satellite 3 at time t = 35 s. The remaining control inputs of length N_{min} stored at the actuator successfully bridge the time until the control task is assigned to a new controller.

4. Discussion

In this work we have presented a two-fold approach to deal with data losses and controller failures in a networked predictive formation control scenario. First a state-of-the-art event based networked model predictive control approach was extended to enable it to adapt to changing network conditions by varying the control horizon length. The theoretical foundation of this adaption based on a Markovian network model was elaborated and a reference implementation further tested on a hardware testbed. Both the HW and SW of the testbed were introduced in the process. The second contribution is a controller failure detection and compensation approach tailored for a possibly large network of heterogeneous cooperating actuator- and controller nodes. The actuator both detects the failure and redistributes the control responsibility to a new controller. Using an auction based methodology for the redistribution it scales well with the number of nodes and allows to optimize for continuing good control performance despite the controller switch.

The stability and smooth control behavior during the control handover to another node were demonstrated on the testbed with a controller failure during satellite reconfiguration scenario. Open points for further research are the inclusion of more elaborate models of the network into the control horizon length calculation. This could be realized by having the estimated future network state as part of the MPC state space which some previous scientific contributions in the domain of networked MPC already proposed. Another point for improvement would be to test our approach with more complex scenarios, e.g., a multi-satellite reconfiguration, as well as with non-linear systems with higher dynamic requirements like attitude control. Finally on the control-task redistribution side additional objectives like link-quality between satellites in addition to hop-distance could be used and also inter-task constraints or positive/negative task synergies could be taken into account.

Author Contributions: Conceptualization, F.K. and J.S.; Formal analysis, F.K. and S.H.; Funding acquisition, K.S.; Methodology, F.K., J.S. and S.H.; Project administration, F.K. and J.S.; Software, F.K. and S.H.; Supervision, K.S.; Validation, F.K.; Visualization, F.K. and S.H.; Writing—original draft, F.K. and J.S. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by the German Aerospace Center (DLR) within the "YETE-II" project under the Grant Agreement No. 50RA1733. and by the European Space Agency (ESA) by the Networking/Partnering Initiative (NPI) under Contract No. 4000113715/15/NL/MH/ats.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the writing of the manuscript, and in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
CAN	Controller-Area-Network
COTS	Commercial off-the-Shelf
DARE	Discrete-Time Algebraic Riccati Equation
ECI	Earth-Centered Inertial
EPS	Electric Power Supply
EWMA	Exponentially-Weighted-Moving-Average
GAP	General-Assignment-Problem
HAL	Hardware-Abstraction-Layer
HCW	Hill–Clohessy–Wiltshire
LTI	Linear Time-Invariant
LVLH	Local-Vertical, Local-Horizontal
MAC	Media Access Control
MPC	Model Predictive Control
OCP	Optimal Control Problem
QoS	Quality-of-Service
UAV	Unmanned Aerial Vehicle

References

1. Cappelletti, C.; Battistini, S.; Malphrus, B.K. *Cubesat Handbook from Mission Design to Operations*; Academic Press: London, UK, 2021. Available online: https://www.sciencedirect.com/science/book/9780128178843 (accessed on 1 July 2022).
2. Villela, T.; Costa, C.A.; Brandão, A.M.; Bueno, F.T.; Leonardi, R. Towards the Thousandth CubeSat: A Statistical Overview. *Int. J. Aerosp. Eng.* **2019**, *2019*, 1–13. Available online: https://www.hindawi.com/journals/ijae/2019/5063145/ (accessed on 1 June 2022). [CrossRef]
3. Marinella, M.J. Radiation Effects in Advanced and Emerging Nonvolatile Memories. *IEEE Trans. Nucl. Sci.* **2021**, *68*, 546–572. [CrossRef]
4. Brown, O.; Eremenko, P. *Fractionated Space Architectures: A Vision for Responsive Space*; Technical Report; Defense Advanced Research Projects Agency: Arlington, VA, USA, 2006.
5. Mikschl, T.; Montenegro, S.; Hilgarth, A.; Kempf, F.; Schilling, K.; Tzschichholz, T. Resource Sharing, Communication and Control for Fractionated Spacecraft (YETE). In Proceedings of the 10th Symposium on Small Satellites for Earth Observation, Berlin, Germany, 20–24 April 2015.
6. Guo, J.; Maessen, D.C.; Gill, E.K.A. Fractionated Spacecraft: The New Sprout in Distributed Space Systems. In Proceedings of the 60th International Astronautical Congress: IAC 2009, Daejeon, Korea, 12–16 October 2009.

7. Kwon, D.; Cheplak, M. Applications of Fractionated Spacecraft Architectures. In Proceedings of the AIAA SPACE 2011 Conference & Exposition, Long Beach, CA, USA, 27–29 September 2011; p. 7131.
8. Mathieu, C.; Weigel, A. Assessing the Flexibility Provided by Fractionated Spacecraft. In Proceedings of the Space 2005, AIAA; Long Beach, CA, USA, 30 August–1 September 2005; p. 6700.
9. Mathieu, C.; Weigel, A. Assessing the Fractionated Spacecraft Concept. In Proceedings of the Space 2006, AIAA, San Jose, CA, USA, 19–21 September 2006; p. 7212.
10. Brown, O.; Eremenko, P.; Bille, M. Fractionated Space Architectures: Tracing the Path to Reality. In Proceedings of the AIAA/USU Conference on Small Satellites, Logan, UT, USA, 10–13 August 2009.
11. Brown, O.; Eremenko, P.; Collopy, P. Value-Centric Design Methodologies for Fractionated Spacecraft: Progress Summary from Phase I of the DARPA System F6 Program. In Proceedings of the AIAA Space 2009 Conference & Exposition, Pasadena, CA, USA, 14–17 September 2009; p. 6540.
12. DARPA Cancels Formation-Flying Satellite Demo. Available online: https://spacenews.com/35375darpa-cancels-formation-flying-satellite-demo/ (accessed on 1 June 2022).
13. Di, F.; Li, A.; Guo, Y.; Wang, C.; Wang, L. Attitude Tracking Control for Fractionated Spacecraft with Actuator Failures under Adaptive Event-Triggered Strategy. In *Advances in Space Research*; Elsevier: Amsterdam, The Netherlands, 2022.
14. Xu, M.; Liang, Y.; Tan, T.; Wei, L. Cluster Flight Control for Fractionated Spacecraft on an Elliptic Orbit. *Celest. Mech. Dyn. Astron.* **2016**, *125*, 383–412. [CrossRef]
15. Chu, J.; Guo, J.; Gill, E. Decentralized Autonomous Planning of Cluster Reconfiguration for Fractionated Spacecraft. *Acta Astronaut.* **2016**, *123*, 397–408. [CrossRef]
16. Wan, S.H.; Song, J.L.; Chen, J.; Hu, M. Hybrid Approach to Optimize the Cluster Flying Orbit for Fractionated Spacecraft Based on PSO-SQP Algorithm. In *Applied Mechanics and Materials*; Trans Tech Publications Ltd.: Wollerau, Switzerland, 2013; Volume 341, pp. 1144–1149.
17. Mosleh, M.; Dalili, K.; Heydari, B. Optimal Modularity for Fractionated Spacecraft: The Case of System F6. *Procedia Comput. Sci.* **2014**, *28*, 164–170. [CrossRef]
18. Li, X.; Yao, Y.; Yang, B.; Wang, L. Guidance Strategy Design for Space Debris Removal Using Fractionated Spacecraft. In Proceedings of the 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC), Nanjing, China, 12–14 August 2016; pp. 264–270.
19. Alandihallaj, M.A.; Emami, M.R. Multiple-Payload Fractionated Spacecraft for Earth Observation. *Acta Astronaut.* **2022**, *191*, 451–471.
20. Schervan, T.A.; Kortmann, M.; Schroder, K.; Kreisel, J. iBOSS Modular Plug & Play-Standardized Building Block Solutions for Future Space Systems Enhancing Capabilities and Flexibility, Design, Architecture and Operations. In Proceedings of the 68th International Astronautical Congress (IAC), Adelaide, Australia, 25–29 September 2017.
21. Kortman, M.; Ruhl, S.; Weise, J.; Kreisel, J.; Schervan, T.; Schmidt, H.; Dafnis, A. Building Block Based iBoss Approach: Fully Modular Systems with Standard Interface to Enhance Future Satellites. In Proceedings of the 66th International Astronautical Congress (Jerusalem), Jerusalem, Israel, 12–16 October 2015; pp. 1–11.
22. InnoCube-Chair of Computer Science VIII-Aerospace Information Technology. 2022. Available online: https://www.informatik.uni-wuerzburg.de/en/aerospaceinfo/wissenschaft-forschung/innocube/ (accessed on 15 July 2022).
23. Grzesik, B.; Baumann, T.; Walter, T.; Flederer, F.; Sittner, F.; Dilger, E.; Gläsner, S.; Kirchler, J.L.; Tedsen, M.; Montenegro, S. InnoCube—A Wireless Satellite Platform to Demonstrate Innovative Technologies. *Aerospace* **2021**, *8*, 127. [CrossRef]
24. Mikschl, T.; Hilgarth, A.; Kempf, F.; Kheirkah, A.; Tzschichholz, T.; Montenegro, S.; Schilling, K. YETE: Distributed, Networked Embedded Control Approaches for Efficient, Reliable Mobile Systems. *DASIA 2014-Data Syst. Aerosp.* **2014**, *725*, 20.
25. Kempf, F.; Santa Cruz, U.; Scharnagl, J.; Schilling, K. Networked and Distributed Cooperative Attitude Control of Fractionated Small Satellites. In Proceedings of the 69th International Astronautical Congress, Bremen, Germany, 1–5 October 2018.
26. Kempf, F.; Hilgarth, A.; Kheirkhah, A.; Mikschl, T.; Tzschichholz, T.; Montenegro, S.; Schilling, K. Reliable Networked Distributed On-Board Data Handling Using a Modular Approach with Heterogeneous Components, In Proceedings of the 4S Symposium, Majorca, Spain, 26–30 May 2014.
27. Catanoso, D.; Kempf, F.; Schilling, K.; D'Amico, S. Networked Model Predictive Control for Satellite Formation Flying. In Proceedings of the 10th International Workshop of Satellites Constellations and Formation, Glasgow, UK, 16–19 July 2019; p. 16.
28. Heemels, W.; Johansson, K.; Tabuada, P. An Introduction to Event-Triggered and Self-Triggered Control. In Proceedings of the 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), Maui, HI, USA, 10–13 December 2012; pp. 3270–3285. [CrossRef]
29. Peng, C.; Song, Y.; Peng Xie, X.; Zhao, M.; Fei, M.R. Event-Triggered Output Tracking Control for Wireless Networked Control Systems with Communication Delays and Data Dropouts. *IET Control. Theory Appl.* **2016**, *10*, 2195–2203. [CrossRef]
30. Zhang, L.; Swain, A.; Zhang, D.; Wen, S. A Discrete Event-Triggered Scheme for Networked Control Systems. In Proceedings of the 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE), Kyoto, Japan, 20–23 June 2021; pp. 1–6.
31. Wang, C.; Guo, L.; Wen, C.; Hu, Q.; Qiao, J. Event-Triggered Adaptive Attitude Tracking Control for Spacecraft With Unknown Actuator Faults. *IEEE Trans. Ind. Electron.* **2020**, *67*, 2241–2250. [CrossRef]
32. Wang, C.; Li, Y.; Hu, Q.; Huang, J. Event-Triggered Adaptive Control for Attitude Tracking of Spacecraft. *Chin. J. Aeronaut.* **2019**, *32*, 454–462. [CrossRef]
33. Wu, B.; Shen, Q.; Cao, X. Event-Triggered Attitude Control of Spacecraft. *Adv. Space Res.* **2018**, *61*, 927–934. [CrossRef]

34. Kheirkhah, A.; Kempf, F.; Tzschichholz, T.; Schilling, K. Robust Distributed Control for a Mechanical-Electrical Demonstrator Considering Communication Constraints. *IFAC-PapersOnLine* **2015**, *48*, 246–251. [CrossRef]
35. Montenegro, S.; Dannemann, F. RODOS-real Time Kernel Design for Dependability. *DASIA 2009-DAta Syst. Aerosp.* **2009**, *669*, 66.
36. Walter, T.; Hilgarth, A.; Mikschl, T.; Montenegro, S. VIDANA: A Fault Tolerant Approach for a Distributed Data Management System in Nano-Satellites. In Proceedings of the 10th Symposium on Small Satellites for Earth Observation, Berlin, Germany, 6–10 May 2019.
37. Mikschl, T.; Rauscher, R.; Montenegro, S.; Schilling, K.; Kempf, F.; Tzschichholz, T. *Collision Free Protocol for Ultrawideband Links in Distributed Satellite Avionics*; University of Würzburg: Würzburg, Germany, 2016.
38. Fortescue, P.; Swinerd, G.; Stark, J. *Spacecraft Systems Engineering*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
39. Clohessy, R.S.; Wiltshire. The Clohessy-Wiltshire Equations of Relative Motion. *J. Aerosp. Sci.* **1960**, *27*, 653–658. [CrossRef]
40. Hrovat, D.; Di Cairano, S.; Tseng, H.; Kolmanovsky, I. The Development of Model Predictive Control in Automotive Industry: A Survey. In Proceedings of the 2012 IEEE International Conference on Control Applications, Dubrovnik, Croatia, 3–5 October 2012; pp. 295–302. [CrossRef]
41. Qian, X.; Navarro, I.; de La Fortelle, A.; Moutarde, F. Motion Planning for Urban Autonomous Driving Using Bézier Curves and MPC. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 826–833. [CrossRef]
42. Yu, S.; Hirche, M.; Huang, Y.; Chen, H.; Allgöwer, F. Model Predictive Control for Autonomous Ground Vehicles: A Review. *Auton. Intell. Syst.* **2021**, *1*, 4. Available online: https://link.springer.com/10.1007/s43684-021-00005-z (accessed on 15 June 2022). [CrossRef]
43. Valmorbida, A. Development and testing of model predictive control strategies for spacecraft formation flying. Ph.D. Thesis, Universita degli Studi di Padova, Padova, Italy, 2014. Available online: https://www.research.unipd.it/handle/11577/3423708?1/valmorbida_andrea_tesi.pdf (accessed on 8 June 2022).
44. Englert, T. A Software Framework for Embedded Nonlinear Model Predictive Control Using a Gradient-Based Augmented Lagrangian Approach (GRAMPC). *Optim. Eng.* **2019**, *20*, 769–809. [CrossRef]
45. Christofides, P.D.; Liu, J.; Muñoz de la Peña, D. Networked and Distributed Predictive Control. In *Advances in Industrial Control*; Springer: London, UK, 2011. Available online: http://link.springer.com/10.1007/978-0-85729-582-8 (accessed on 10 June 2022).
46. Kalman, R.; Bertram, J. Control System Analysis and Design via the Second Method of Lyapunov: (I) Continuous-Time Systems (II) Discrete Time Systems. *IRE Trans. Autom. Control* **1959**, *4*, 112–112. [CrossRef]
47. Burk, D.; Volz, A.; Graichen, K. Towards a Modular Framework for Distributed Model Predictive Control of Nonlinear Neighbor-Affine Systems. In Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France, 11–13 December 2019; pp. 5279–5284. Available online: https://ieeexplore.ieee.org/document/9029800/ (accessed on 1 May 2022).
48. Kapernick, B.; Graichen, K. The Gradient Based Nonlinear Model Predictive Control Software GRAMPC. In Proceedings of the 2014 European Control Conference, ECC 2014, Strasbourg, France, 24–27 June 2014. [CrossRef]
49. Sahni, S.; Gonzalez, T. P-Complete Approximation Problems. *J. ACM* **1976**, *23*, 555–565. [CrossRef]
50. Bertsekas, D.P. The Auction Algorithm: A Distributed Relaxation Method for the Assignment Problem. *Ann. Oper. Res.* **1988**, *14*, 105–123. Available online: http://link.springer.com/10.1007/BF02186476 (accessed on 15 July 2022). [CrossRef]
51. Koenig, S.; Tovey, C.; Lagoudakis, M.; Markakis, V.; Kempe, D.; Keskinocak, P.; Kleywegt, A.; Meyerson, A.; Jain, S. The Power of Sequential Single-Item Auctions for Agent Coordination. In Proceedings of the AAAI, Boston, MA, USA, 16–17 July 2006; Volume 2006, pp. 1625–1629.
52. Lagoudakis, M.; Berhault, M.; Koenig, S.; Keskinocak, P.; Kleywegt, A. Simple Auctions with Performance Guarantees for Multi-Robot Task Allocation. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Sendai, Japan, 28 September–2 October 2004; Volume 1, pp. 698–705. Available online: http://ieeexplore.ieee.org/document/1389434/ (accessed on 10 June 2022).
53. Nanjanath, M.; Gini, M. Dynamic Task Allocation for Robots via Auctions. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 2781–2786. [CrossRef]
54. Schneider, E.; Sklar, E.I.; Parsons, S.; Özgelen, A. Auction-Based Task Allocation for Multi-Robot Teams in Dynamic Environments. In Proceedings of the Conference Towards Autonomous Robotic Systems, Liverpool, UK, 8–10 September 2015; pp. 246–257.
55. Koenig, S.; Tovey, C.A.; Zheng, X.; Sungur, I. Sequential Bundle-Bid Single-Sale Auction Algorithms for Decentralized Control. In Proceedings of the IJCAI, Hyderabad, India, 6–12 January 2007; pp. 1359–1365.
56. Gerkey, B.P.; Matarić, M.J. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *Int. J. Robot. Res.* **2004**, *23*, 939–954. [CrossRef]