LEARNING-BASED CONTROL AND ESTIMATION FOR ATTITUDE REGULATION OF A REUSABLE LAUNCHER FOR LANDING SCENARIO

V. Renganathan[(1)], A. Cervin[(1)], P. Rosa[(2)], A. Marcos[(3)], J. Belhadj[(4)], J. P. Belfo[(2)], A. Rantzer[(1)], D. Navarro-Tapia[(3)], J. Gronqvist[(1)], N. Somma[(2)], A. Botelho[(2)], G. Tofanelli[(2)], M. Casasco[(4)], S. Bennani[(4)]

[(1)]*Department of Automatic Control - LTH, Lund University, Sweden, venkat@control.lth.se*
[(2)]*DEIMOS Engenharia SA, Lisbon, Portugal, paulo.rosa@deimos.com.pt*
[(3)]*Technology for AeroSpace Control (TASC), Martock, UK, andres.marcos@tasc-group.com*
[(4)]*European Space Agency, Noordwijk, Netherlands, joris.belhadj@esa.int*

ABSTRACT

The careful blending of artificial intelligence with the mathematically rigorous control theory comes with numerous promises for space applications and thereby is projected to revolutionise the space industry. In the same spirit, this article comes as a part of the ESA-i4GNC software framework developed towards analysing the usage of artificial intelligence in guidance, navigation and control for futuristic space vehicles. A learning-based control and state estimation for attitude regulation of a reusable launcher for landing scenario is proposed in this paper. Specifically, a learning-based model predictive control is used for regulating the attitude dynamics of the reusable launch vehicle model during its landing scenario and a learning-based Kalman Filter implementation using Gaussian Process for state estimation is proposed in this work. The combination of noise filtering by Gaussian process improving the state estimation and the usage of learning-based model predictive control to estimate the unmodeled dynamics from the input-output data ensures that landing scenario for the reusable launch vehicle is efficiently handled.

1 INTRODUCTION

Artificial Intelligence (AI) has great potential to solve completely many real-world problems and aid in the process of finding good sub-optimal solutions for difficult real-world tasks. Though the usage of AI-based techniques proposed as early as [1] has been setting the trend in many industries, its usage in space applications as mentioned in [2]–[6] is still in its infancy. This is mainly due to the lack of rigorous safety guarantees for many AI-based techniques and combined with the space industry being an expensive arena where failure tolerance is extremely small. However, the current idea of blending AI techniques into the mathematically grounded control theory approaches seems to have opened up a fertile area of application for the AI and thereby mitigating their application hindrance with the inculcation of new safety guarantees from control theory perspectives. Interested readers are referred to the recent works on AI for space research by many worldwide space research organizations like ESA [2], NASA [7], DLR, ISRO, and CNSA. This paper contains two main problem statements addressed in relation to the project concerning the usage of artificial intelligence for guidance, navigation and control for futuristic space vehicles. Specifically, within that project, a two stage to orbit launcher called RETALT1 was defined by the consortium, and a flight dynamic simulator was developed by Deimos Space [8].

1.1 Learning-based Model Predictive Control for Attitude Regulation of a Reusable Launcher

We present the learning-based model predictive control technique as described in [9] for controlling the attitude dynamics of the RETALT vehicle during its landing phase. Specifically, the linearized model at different points along the nominal reference trajectory was utilized as the main nominal plant for the design of control action. We consider a more accurate modelling of the RETALT attitude dynamics parameterized by different time steps with ground truth values of parameters defining the model being estimated. Specifically, a simplified single axis rotation dynamics was assumed be the nominal model. Given the nominal plant, it would not be suitable to device a control strategy for just the nominal dynamics if the plant can possibly have potentially nonlinear unmodelled dynamics. This clearly calls for learning the unmodelled dynamics while aiming to control the RETALT vehicle and such a learning-based regulation is necessary to ensure soft landing despite having bounds on the knowledge of system dynamics. The unmodelled dynamics were learned using an oracle function employing a linear least squares based estimation. This specific learning part involving the estimation of the unmodelled dynamics ensures that both the optimal cost is reduced using the learning-based adaptive model and the constraint satisfaction is guaranteed using the nominal model. That is, both optimised performance objective and the robustness guarantees are obtained simultaneously. This specific control ideology was applied to regulate the attitude error dynamics of the RETALT model developed as a part of the ESA-i4GNC software framework with the simulations resulting in promising results. While the methodology is tailored for learning the unmodelled dynamics using a linear oracle module, it can very well be extended to learn complex nonlinear unmodelled dynamics using a potential neural network based nonlinear oracle module.

1.2 Learning-based Kalman Filter Using Gaussian Process for Attitude Regulation of Reusable Launcher

Most aerospace applications assume that there already exists a high-performing navigation system that provides estimates of all the state variables needed for feedback and learning algorithms. It is interesting to study how navigation (state estimation) can be improved by online learning of unmodelled and possible state-dependent random disturbance processes because the quality of the state estimates directly impact the achievable control performance at the lower loop level. While the Kalman filter and its various extensions that rely on accurate models of the plant is realistic for space vehicles, knowing in advance the type and intensity of disturbances can sometimes be impossible. Hence learning can be crucial for fine-tuning the navigation system during a mission. We propose to apply the Learning-based Kalman filter to the RETALT reusable launch vehicle model, where we estimated two components of the vehicle attitude vector (the yaw or the pitch) and their derivatives (the yaw rate or the pitch rate) based on noisy measurements and knowledge of the control inputs. Specifically, the Gaussian process based approach is advocated to learn the unknown stochastic function which represents the state-dependent disturbance vector. A simulation sequence with a learning-based Kalman filter with the linear correlation model for the RETALT vehicle highlighted a significant improvement in estimator performance.

Following a short summary of notations used in the paper, the rest of the paper is organised as follows: the problem formulation for the attitude regulation of a reusable launcher is established in §2 and the control module using the learning-based MPC is explained in 2.1 and the learning-based state estimation is discussed in 2.2. The proposed approaches are demonstrated in §3 using the RETALT vehicle model developed in the ESA-i4GNC software framework. The paper is finally closed in §4 along with the summary of results and directions for future research.

Notations

The cardinality of the set A is denoted by $|A|$. The set of real numbers, integers and the natural numbers are denoted by $\mathbb{R}, \mathbb{Z}, \mathbb{N}$ respectively and the subset of real numbers greater than a given constant say $a \in \mathbb{R}$ is denoted by $\mathbb{R}_{>a}$. The subset of natural numbers between two constants $a, b \in \mathbb{N}$ with $a < b$ is denoted by $\mathbb{N}_a^b$. The operator $\ominus$ denotes the set difference. For a matrix $A \in \mathbb{R}^{n \times n}$, we denote its transpose and its trace by $A^\top$ and $\mathbf{Tr}(A)$ respectively. We denote by $\mathbb{S}^n$ the set of symmetric matrices in $\mathbb{R}^{n \times n}$. For a symmetric matrix $A \in \mathbb{S}^n$, we denote by $A \succ 0$ to mention that A is positive definite and by $A \succeq 0$ to mention that A is positive semi-definite. An identity matrix of dimension n is denoted by I_n. Given $x \in \mathbb{R}^n, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times n}$, the notations $\|x\|_A^2$ means $x^\top A x$.

2 PROBLEM DESCRIPTION

2.1 Control Using Learning-based Model Predictive Control (LBMPC)

First, we investigate the problem of controlling the attitude dynamics of the RETALT vehicle during its landing phase. Specifically, the linearized model at different points of the nominal reference trajectory is given and the resulting linear system is utilized as the main nominal plant for the design of control action. We describe how to utilize the learning-based model predictive control to regulate the attitude dynamics of the RETALT plant at different reference trajectory points. Such a learning-based regulation is necessary to ensure soft landing despite having bounds on the knowledge of system dynamics. We consider a more accurate modeling of the RETALT attitude dynamics parameterized by different time steps with ground truth values of parameters defining the model being estimated. Assuming them to be the nominal model at the given time step, we present here the attitude dynamics regulation using the LBMPC procedure.

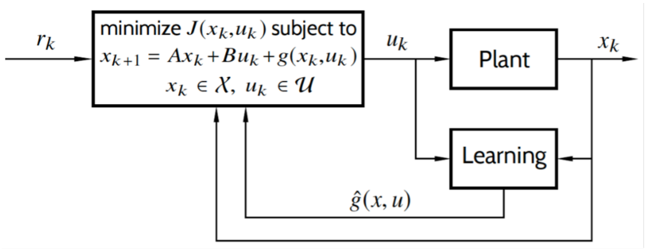
r_k
minimize $J(x_k, u_k)$ subject to
$x_{k+1} = Ax_k + Bu_k + g(x_k, u_k)$
$x_k \in \mathcal{X},\ u_k \in \mathcal{U}$
u_k
Plant
x_k
Learning
$\hat{g}(x, u)$

Figure 1: Learning-based Model Predictive Control (LBMPC) general principle

2.1.1 Nominal System Model

We consider a model of the RETALT vehicle defined using a simplified single axis rotation dynamics. Specifically, we consider both the yaw and pitch dynamics to be identical and decoupled resulting in the total attitude dynamics. Our model consists of two parameters name the aerodynamic instability coefficient a_6 and the control efficiency parameter k_1 and the model is given by

$$G_{LV}(s) := \frac{k_1}{s^2 - a_6}. \tag{1}$$

The knowledge of the true values of both the parameters at different times of landing is assumed to be known in advance and this in turn gave us the required nominal model to start the LBMPC procedure. The system defined by the above dynamics at different time steps is depicted using the sequence of bode plots in Fig. 2.

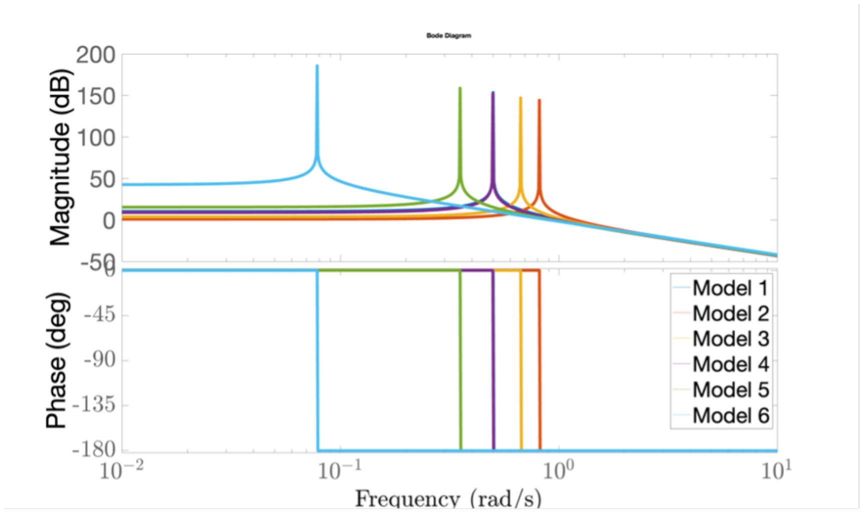

Figure 2: Bode plot containing the a_6, k_1 parameter variation based on linearization at six different time steps is shown here.

2.1.2 LBMPC Theory

The LBMPC theory presented here is adopted from [9] which is basically a modified Robust MPC as in [10] with additional learning component module attached to it. Consider a discrete-time LTI system with modelling error given by

$$x_{t+1} = Ax_t + Bu_t + w_t + g(x_t, u_t), \tag{2}$$

where at time $t \in \mathbb{N}$, $x_t \in \mathbb{R}^n$ denotes the system state, $u_t \in \mathbb{R}^m$ denotes the control inputs, $w_t \in \mathbb{R}^n$ denotes the process disturbance and $g(x_t, u_t) \in \mathbb{R}^n$ denotes the unmodelled nonlinear dynamics and is considered to be a function of both system states and control inputs at time t. The matrices $A \in \mathbb{R}^{n\times n}$, and $B \in \mathbb{R}^{m\times n}$ denoting the system matrix and control input matrix respectively are known in advance. The system is supposed to operate under operational constraints namely the state constraints $\mathcal{X} \subseteq \mathbb{R}^n$, control input constraints $\mathcal{U} \subseteq \mathbb{R}^m$, and terminal state constraints $\Omega \subseteq \mathbb{R}^n$. For simplicity, the constraint sets $\mathcal{X}$,$\mathcal{U}$ and Ω are assumed to be convex polytopes in appropriate dimensions.

Assumption 1: For simplicity, the system's unmodeled dynamics is assumed to be linear [1] in system state and control inputs. That is, the system's unmodeled dynamics is governed by *unknown* matrices $H \in \mathbb{R}^{n\times n}, G \in \mathbb{R}^{m\times n}$ and is given by

$$g(x_t, u_t) = Hx_t + Gu_t, \quad \forall t \in \mathbb{N}, \tag{3}$$

Assumption 2: There exists an Oracle module (linear Oracle [2]) which gives the estimate of the system's unmodeled dynamics when provided with the history of system's input-output data. That is,

$$\hat{g}_t(x_t, u_t) = \mathcal{O}_t(x_t, u_t) = \hat{H}_t x_t + \hat{G}_t u_t. \tag{4}$$

The ideology behind the LBMPC technique as shown in Fig. 1 is to

1. use a constant LTI model (nominal model) to predict the system's response and guarantee robust constraint satisfaction and,

[1]Future work will seek to have nonlinear unmodelled dynamics.

[2]Future work will seek to have nonlinear Oracle based on neural networks to capture nonlinear unmodelled dynamics.

2. use a separate model to adaptively approximate the true (possibly nonlinear) system dynamics as new data becomes available, which yields more accurate state and input predictions to be used in the objective function.

Let us denote the nominal state of the system that evolves without any disturbance or unmodelled dynamics at time t by $\bar{x}_t$. Similarly, let us denote the adaptive state of the system that evolves with the effect of disturbance and the estimated unmodelled dynamics at time t by $\tilde{x}_t$. The main essence of the LBMPC algorithm is to repeatedly solve a constrained optimization problem in a rolling horizon fashion with prediction horizon length given by $N \in \mathbb{N}$, starting both the nominal and adaptive system states from a given state x_t and applying the first control input $\check{u}_k$ from the resulting control input sequence $\mathbf{u}_{k,N} := \left\{\check{u}_t \mid t \in \mathbb{N}_k^{k+N-1}\right\}$. That is, starting from a state $\bar{x}_t = x_t$ and $\tilde{x}_t = x_t$, solve the following constrained optimization problem in a rolling horizon fashion:

$$\begin{aligned}
\underset{\mathbf{u}_{k,N}}{\text{minimize}} \quad & \|\tilde{x}_{t+N}\|_P^2 + \sum_{k=t}^{t+N-1} \left(\|\tilde{x}_k\|_Q^2 + \|\check{u}_k\|_R^2\right) && \text{(5a)}\\
\text{subject to} \quad & \tilde{x}_{t+k+1} = A\tilde{x}_{t+k} + B\check{u}_{t+k} + \mathcal{O}_t(\tilde{x}_{t+k}, \check{u}_{t+k}), && \text{(5b)}\\
& \check{u}_{t+k} = K\bar{x}_{t+k} + c_{t+k} \in \mathcal{U} \ominus (K\mathcal{R}_k), \forall k \in \mathbb{N}_0^{N-1}, && \text{(5c)}\\
& \bar{x}_{t+k+1} = A\bar{x}_{t+k} + B\check{u}_{t+k} \in \mathcal{X} \ominus \mathcal{R}_{k+1}, \forall k \in \mathbb{N}_0^{N-1}, && \text{(5d)}\\
& \bar{x}_{t+N} \in \Omega \ominus \mathcal{R}_N. && \text{(5e)}
\end{aligned}$$

2.2 State Estimation Through Learning-based Kalman Filter (LBKF) Using Gaussian Process

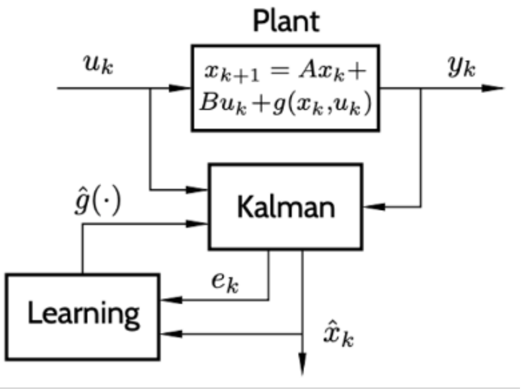

Figure 3: Learning-based Kalman filtering (LBKF) general principle

It is a common practice to assume that there already exists a high-performing navigation system for the relaunch vehicle that provides estimates of all the state variables needed for feedback and learning algorithms. It is nevertheless interesting to study how navigation (state estimation) can be improved by online learning of unmodeled and possible state-dependent random disturbance processes. Navigation is an essential part in a GNC system, since the quality of the state estimates directly impact the achievable control performance at the lower loop level. State estimation algorithms—such as the Kalman filter and its various extensions— rely on accurate models of the plant, which are used to predict the future state of the vehicle based on the current estimated state and the current control input (if any). The predicted state is combined with filtering of measurements (partial observations of the plant state) to yield a new state estimate. Besides a linear or nonlinear dynamics model of the plant, it is also necessary to know the statistical properties of the various random disturbances acting on the system. While the former is realistic for space vehicles, knowing in advance the type and intensity of disturbances can sometimes be impossible. Hence, learning can be crucial for fine-tuning the navigation system during a mission. There exist many variants of LBKF. The approach applied here represents a slightly simplified version of the Gaussian-process-based approach proposed in [L2020]. The algorithm assumes an augmented linear state-space system

$$x_{k+1} = Ax_k + Bu_k + Gg(x_k, u_k) + w_k, \tag{6}$$

$$y_k = Cx_k + v_k, \tag{7}$$

where the (a-priori) unknown stochastic function $g(x_k)$ represents a state-dependent disturbance vector of size no greater than the measured output vector y_k. Disregarding the nonlinear function $g(\cdot)$, standard Kalman filter assumptions apply: The disturbances processes w_k and v_k are assumed independent, white, and normally distributed with zero mean and known covariances R_1 and R_2, and the initial state is Gaussian with mean $\hat{x}_0$ and covariance P_0. A block diagram of a Learning-based Kalman filtering application is shown in Fig. 3. The Kalman filter uses knowledge of the plant model (the matrices A, B, G, and C) together with the online collected control inputs u_k and measured plant outputs y_k to create the current state estimate $\hat{x}_k$. Based on the estimates and the prediction errors

$$e_k = \hat{x}_k - Ax_{k-1} - Bu_{k-1} \tag{8}$$

The goal of the learning part if to establish a statistical relationship between the current estimate $\hat{x}_k$ and the output of the unknown function $g(x_k, u_k)$. This knowledge can then be used to produce a state- and/or input-dependent disturbance estimate $\hat{g}(\hat{x}_k, \hat{u}_k)$, thus improving the performance.

3 SIMULATION RESULTS

In this section, we demonstrate our proposed approach of learning-based control and learning-based state estimation applied for relaunch vehicle model which is integrated into the ESAi4GNC framework developed. We start by discussing the simulation results using the LBMPC followed by simulation results of LBKF.

3.1 LBMPC Problem Setup

We consider a linear perturbation to our considered nominal model (A, B). That is, $g(x_k, u_k) = Hx + Gu$, where $H = 0.001A, G = 0.001B$. The considered continuous system was then discretized using a sampling time of 0.1 seconds. Since both pitch and yaw dynamics are considered to be decoupled, we assumed a non-zero initial condition of $\begin{bmatrix} \frac{\pi}{8} & 0 - \frac{\pi}{8} & 0 \end{bmatrix}^\top$ for the angular positions and zero angular velocities. We used a maximum time-step of 100 and considered an MPC horizon length of N = 10 time steps. The state and input penalty matrices were chosen to be $Q = I_n, R = I_m$. The states were given a lower bound of $-\pi$ and an upper bound of π. The noise was assumed to be zero for simplicity. The first and third state in the modeling error was assumed to be in $\begin{bmatrix} -0.0001(x_0(1))^2 & 0.0001(x_0(1))^2 \end{bmatrix}^\top$ and $\begin{bmatrix} -0.0001(x_0(3))^2 & 0.0001(x_0(3))^2 \end{bmatrix}^\top$ respectively and these bounds were used to compute the modeling error polytope for performing robust constraint tightening. For ease of exposition, we had used a quadratic nonlinearlty. Since, the thrust was assumed to be constant, we essentially had to use only the other two control inputs in our linear model and the bounds was chosen from the RETALT parameters definition excel file and they were $[-0.1745328, 0.1745328]$ both the inputs respectively. The oracle module of the LBMPC performed a simple linear regression from the input-output data to estimate the unmodelled dynamics (in our case, the F & G perturbation matrices). Finally, an infinite horizon LQR gain was used to construct the

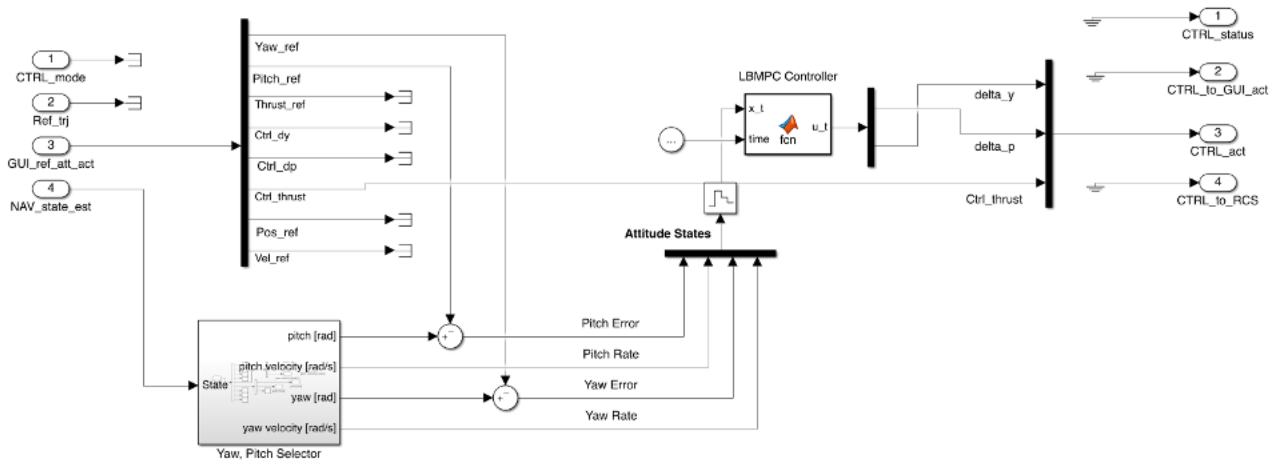

Figure 4: Simulink model of LBMPC within the ESA-i4GNC framework is shown here.

robust disturbance invariance set which was then used for robust constraint satisfaction by the state and control trajectories of the LBMPC algorithm. The LBMPC optimization problem was solved using CVX and the robust constraint tightening was performed using the MPT Toolbox.

3.2 LBMPC Implementation

Since the LBMPC module is integrated within the ESA-i4GNC software framework, the implementation details using the Simulink model is shown in Fig. 4. Initially, the system dynamics at different time steps corresponding to different a_6, k_1 values are prepared and with each system dynamics, the corresponding control invariant set and the other error polytopes are generated using the MPT toolbox. Once the polytopes are generated, the LBMPC simulation can be started. When the LBMPC simulation ends, the stored values of system states, attitude errors and other interesting variables of interest are plotted by the ESA-i4GNC tool. The detailed steps are as follows:

1. Retrieve the system matrices according to the time of operation
2. Build the observation matrix $Y = X(t+1) - AX(t) - BU(t)$ using all the data from time zero
3. Call the oracle to output the estimate of the perturbation matrices $(\hat{H}, \hat{G})$ based on the observation matrix using the linear regression
4. Using the newly obtained estimates of the perturbation matrices, perform the LBMPC optimization where the cost function uses the learned model while constraint satisfaction (tightening) is taken care by the nominal dynamics to return the control input $\breve{u}(t)$.
5. Apply the first input to propagate one step forward to $t = t + 1$ and repeat the process again.

3.3 Discussion of Results

The results show that a near-perfect attitude control for the RETALT vehicle is achieved by the LBMPC algorithm for all the linear models obtained from the reference trajectory information. It is clear from Fig. 5 that LBMPC was able to stabilize and regulate the system dynamics with different a_6, k_1 parameter values defining the attitude system dynamics at different time steps along the reference trajectory of the RETALT vehicle. It was observed that the system dynamics at later time steps during the landing phase near the touch down was the harder one to regulate as it took by far

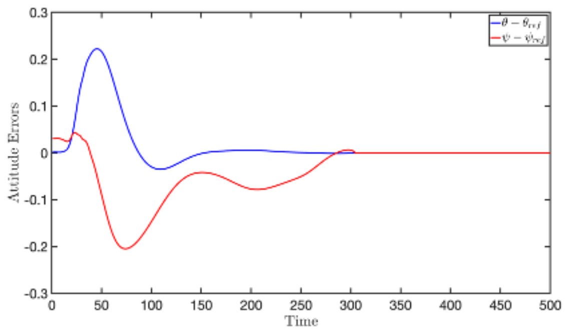

Figure 5: Attitude reference tracking is shown here.

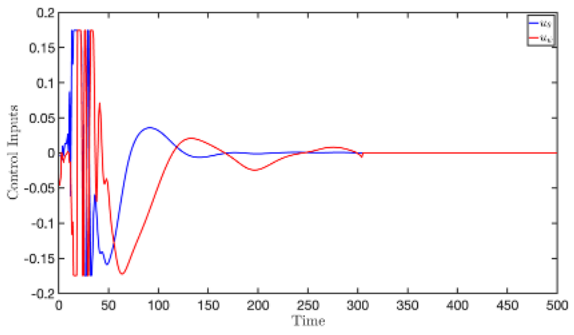

Figure 6: LBMPC control signals are plotted here.

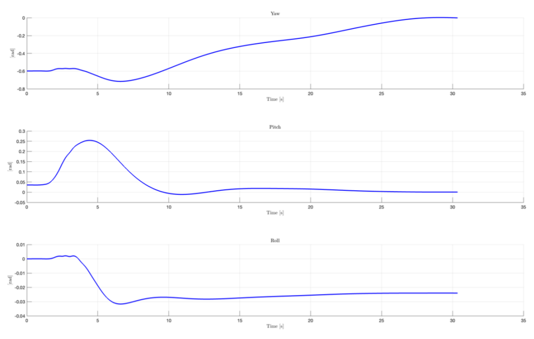

Figure 7: Regulation of angular states is shown here.

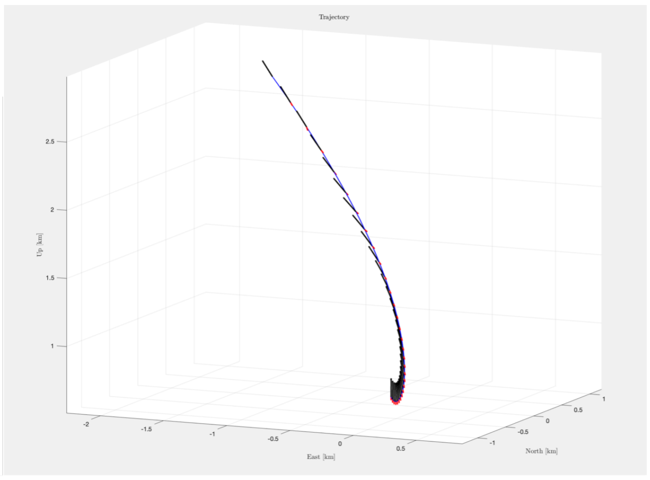

Figure 8: RETALT 3D state trajectory is shown here.

the most number of time steps to achieve near perfect regulation. In the next simulation, we changed the problem formulation from attitude regulation to attitude tracking, where the LBMPC controller was implemented to successfully track the attitude values from the reference trajectory. The control thrust reference from the reference trajectory was used without any modification. The error signal formed using the reference attitude and the actual attitude were fed into the LBMPC controller which calculated the necessary control signal using the procedure described above. We see similar results in the attitude reference tracking as shown in Fig. 7 and Fig. 6. To qualitatively study how the learning enabled the working of LBMPC, we plotted the learning error $H - \hat{H}$ and $G - \hat{G}$ and found that they had reduced over time as shown in Fig. 9. Further, we observed that the dynamics switching had happened before the linear regression had effectively converged to a greater accuracy and it was found to be the effect of starting the linear regression process with both estimates $\hat{H}$ and $\hat{G}$ being initialized to zero. With intelligent starting guesses, this could be rectified. This also shows that switching times should happen ideally when the estimates from one model have converged to their true values sufficiently closer, so that the true perturbation is learned to a greater extent by the LBMPC procedure. This clearly shows that an even better nonlinear oracle such as the neural network can also be employed to estimate the perturbation quantities which however is out of the scope at this point in time.

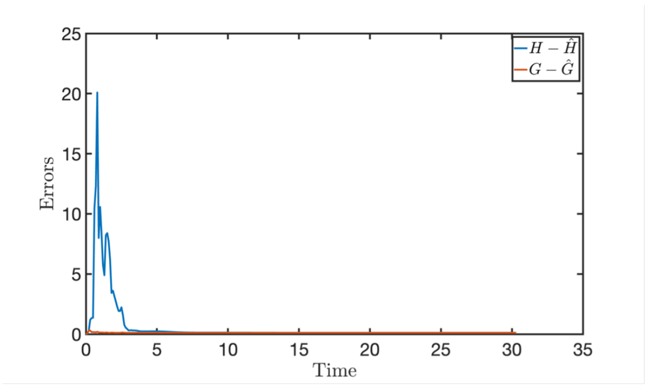

Figure 9: Estimation errors by the LBMPC while learning the unmodelled dynamics is plotted here.

3.4 LBKF Problem Setup

Applying the LBKF to the RETALT reusable launch vehicle model, we focused on the problem of estimating two components of the vehicle attitude vector (the yaw or the pitch) and their derivatives (the yaw rate or the pitch rate) based on noisy measurements and knowledge of the control inputs. There are several possible absolute attitude sensors for space vehicles, including various optical sensors and magnetometers, as well as relative movement sensor such as gyroscopes. For the purpose of illustrating the application of LBKF, we use an abstract measurement model, where some absolute angle sensor provides a noisy reading of the yaw and pitch and regular intervals. The measurement noise is modeled as Gaussian white noise. We can also model that the angle sensor becomes unavailable during some time interval. The LBKF Simulink implementation is shown in Fig. 10.

3.4.1 Dynamics Models

To provide experimental data for testing the LBKF during a landing sequence, we considered a scenario where the simulated RETALT vehicle should perform a vertical decent. Initial errors in the yaw or pitch are regulated to zero by simple, independent PID controllers acting on the thrust vector controls (TVCs). A separate, independent PID controller regulates the thrust of the vehicle according

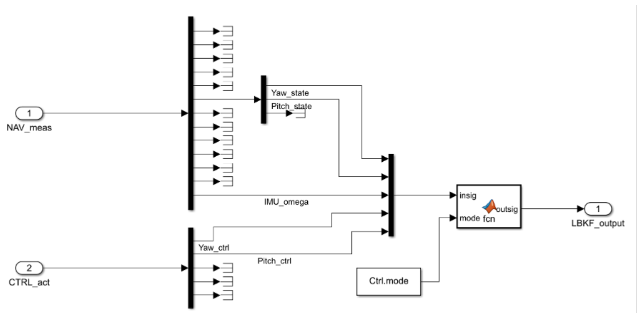

Figure 10: Simulink diagram of the LBKF implementation is shown here.

to a pre-specified decent trajectory. For the purpose of state estimation, we model each angle dynamics (yaw, pitch) independently as shown in Fig. 11. In the simulation model, all the control loops interact through the nonlinear rigid body dynamics. For small angles, however, the interaction can be neglected in the filter designs. A double integrator model for each attitude axis was identified from

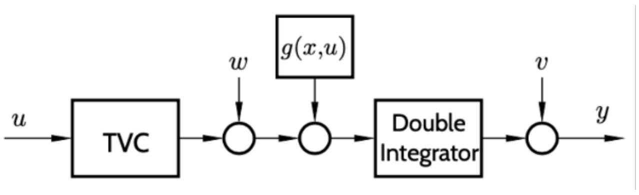

Figure 11: Simplified process dynamics model for single-axis attitude estimation

the RETALT simulation model at the middle of the nominal descent trajectory as used as the basis of the Kalman filter. In the simulation model, the measurement noise v is explicitly added (and its variance hence being known exactly), while the process noise w represents unmodeled disturbances and unknown initial state deviations. In the Kalman filter, the assumed variance of the process noise was tuned to give reasonably fast convergence in nominal conditions. The control signal u was assumed to be known by the state estimator, but the TVC dynamics were assumed unknown. The simulation model includes some low-pass filter dynamics as well as a saturation as well as a rate limitation (both symmetrical) for the actuator. The PID controllers were (de)tuned so that the saturation and rate limitations would be active during parts of the landing sequence. The output of the Kalman filter was not used for feedback control, as this would complicate the interpretation of the results. Rather, the various filters were running in open loop based on the simulated measurement and control signals, and their outputs were stored and analysed offline.

3.4.2 Input Data

Some typical test inputs (both measured outputs and known control inputs) generated by the simulation model are displayed in Fig. 12 and Fig. 13. Here an example of yaw angle control is shown, but the pitch angle dynamics work very similarly (except for a different in sign for the control signal applied). Measurement samples are taken every 0.2 seconds, and the simulation length is 30 seconds (just before the landing of the space craft). For the learning techniques, we consistently used the data from the first 7 seconds (i.e., only 35 samples) for training and then the remaining 23 seconds for performance evaluation. It is seen that the regulation performance is quite poor and that the actuator limitations are significant—again, this is to provide sufficiently exciting input data to the learning algorithm and to amplify the performance difference between the linear and learning (nonlinear) estimation techniques within a very short sequence.

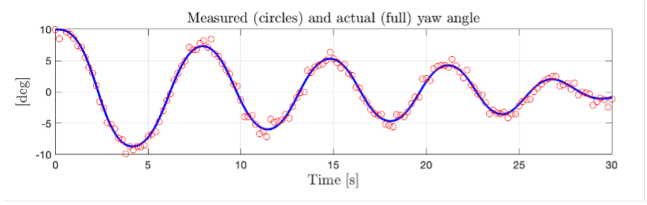

Figure 12: Measured output corresponding to input sequences for the filters, generated by the RETALT model.

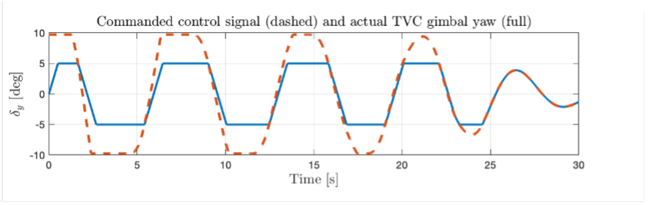

Figure 13: Typical input sequences for the filters, generated by the RETALT simulation model.

3.4.3 Filter Designs

For the filter design, we considered several different estimators of increasing complexity:

1. A *standard Kalman filter*, tuned according to the nominal conditions around the process linearization point. The assumed process noise parameter R_1 was tuned to give a reasonable compromise between convergence and measurement noise rejection.

2. A *Kalman filter with learned noise intensity*, where the training data is used to estimate the process noise intensity R_1. The variance estimation will interpret the unmodeled nonlinear dynamics as extra noise, which will in turn make the Kalman filter rely more on the measurements and less on the linear dynamics model.

3. A *Kalman filter with learned linear model* between the states/inputs and the residual process noise. The predicted extra process disturbance is the next sample is by given by $\hat{g}(\hat{x}_k, u_k) = A\hat{x}_k + Bu_k$, where A and B are given by linear regression on the training data.

4. A *Kalman filter with learned Gaussian process model* capturing the (potentially nonlinear) correlation relationship between the states/inputs and the residual process noise. The predicted extra process disturbance is the next sample is by given by the Gaussian process output $\hat{g}(\hat{x}_k, u_k)$ and includes a state-dependent estimate of its uncertainty. This approach is explained in detail in [11].

A preliminary investigation revealed that the control signal u_k contained the bulk of the prediction power. This is not surprising, since the unmodeled actuator dynamics and saturation constitute the major nonlinearities for the given scenario. Including the process state $\hat{x}_k$ in the learned relationship only contributed with more randomness, and in the end, worse estimator performance. Therefore, we let both the linear regression model and the Gaussian process model capture a scalar-to-scalar relationship between u and $\hat{g}$. Training of the linear and Gaussian models was performed using the Matlab commands `cov` and `fitgrp` respectively. A typical result is shown in Fig. 14. It is seen that both models produce very similar results, and that a linear model is probably sufficient for the given unmodeled dynamics. The data points at the control input saturation limits (± 0.17) could possibly be exploited for better predictions if combined with other input data, but it would require a multidimensional model (with some other input variables than $\hat{x}_k$).

3.5 Discussion of Results

The four different algorithm variants (standard Kalman filter, Kalman filter with adapted process variance, learning-based Kalman filter with linear correlation model, and learning-based Kalman filter with Gaussian process model) were tested on a RETALT vehicle landing sequence, where the

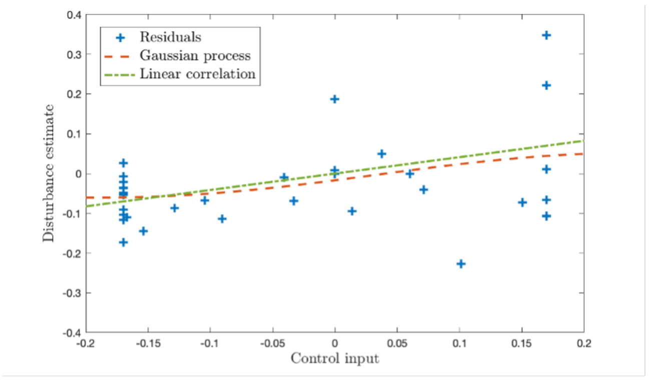

Figure 14: Learned relations between Kalman filter residuals & unmodeled additive disturbances are shown here.

Filter Type	Angle Mean Square Error	Angle Rate Mean Square Error
Standard KF	1.09×10^{-2}	4.18×10^{-2}
KF with Adapted Variance R_1	1.03×10^{-2}	4.02×10^{-2}
LBKF with Linear Model	6.92×10^{-3}	1.77×10^{-2}
LBKF with Gaussian Process	7.68×10^{-3}	2.31×10^{-2}

Table 1: Estimation performance of different Kalman filters are shown here.

yaw angle was regulated from a starting position of 10 degrees during the descent. Noisy angle measurements and the control input from the yaw attitude regulator were fed into the algorithm, which operated in open loop. The first 7 seconds were used to collect data, then possible learning/adaptation took place, and the last 23 seconds of the trajectory were used for evaluating the mean-square error of the yaw angle and angular rate estimates. A simulation sequence with the standard Kalman filter is shown in Fig. 15. The angle estimate looks fairly accurate, while the angular rate estimate has a visible systematic error due to the model error in the actuation model. Using the first 7 seconds to estimate the real process noise variance R_1 yielded an improvement that was not visible by the naked eye, so that graph was not included here. A simulation sequence with a learning-based Kalman filter with the linear correlation model is shown in Fig. 16, and there the improvement in estimator performance is significant. After the learning phase, the angular rate is more accurately tracked. Switching to the more sophisticated Gaussian process model did not yield any further improvements, as seen in Fig. 17. This was not expected, given the similarity of the two models for this use case. To numerically evaluate the performance of the algorithms, 10 different noise sequences were generated and

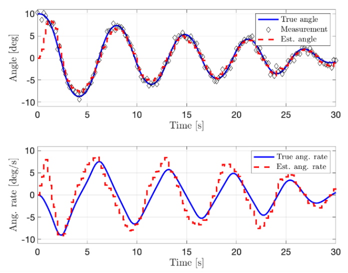

Figure 15: Yaw angle and rate estimation results using standard Kalman filter is shown here.

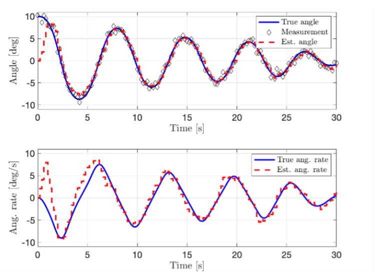

Figure 16: Yaw angle and rate estimation results using LBKF with linear regression model are plotted here.

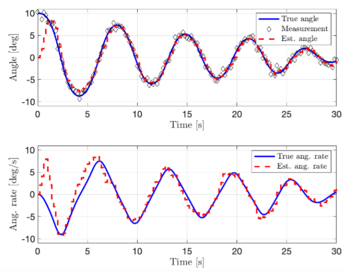

Figure 17: Yaw angle and rate estimation results using LBKF with Gaussian process model are shown here.

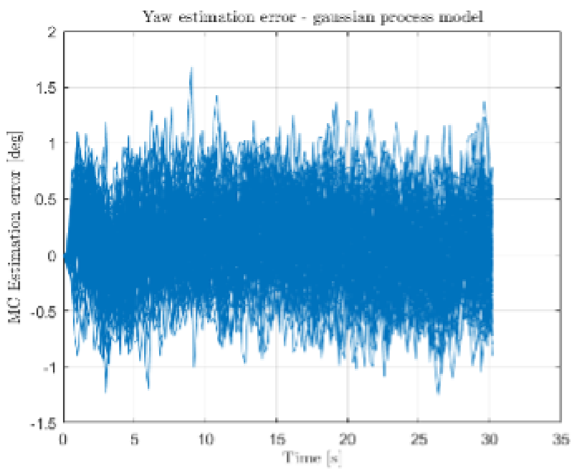

Figure 18: Estimation error from the Monte-Carlo trials are shown here.

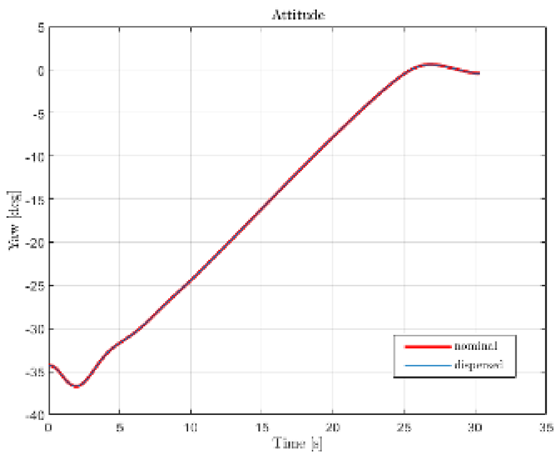

Figure 19: The nominal & dispersed yaw values from Monte-carlo trials are compared here.

tested on each variant, the averaged results over the ten runs being reported in Table 1. The standard Kalman filter and the filter with the adapted process variance performed very similarly, with only a few percent performance improvement for the latter variant. A larger improvement was obtained for both of the learning-based Kalman filters, with the linear model being the overall best one. The failure of the Gaussian process model to improve over the linear correlation model is probably due to overfitting. For the considered problem setup, a linear model was sufficient to capture the major model error in the control signal path. At last, to validate the proposed approach, Monte-Carlo trials were conducted and satisfactory results were obtained as shown in Fig. 18 and Fig. 19.

4 CONCLUSION & FUTURE OUTLOOK

The LBMPC algorithm as implemented is quite general and can handle various types of model errors. The simplest option is to use a linear oracle, and this was deemed sufficient for the attitude control benchmark. Further, LBMPC was shown to result in attitude regulation for RETALT vehicle model at different points along its reference trajectory making it a suitable candidate for implementation. There are several directions for further improving LBMPC implementation in the future such as using better constraint tightening as described in [12]. For instance, the coupling between the attitude and position dynamics were neglected during this design. As a result, exact position tracking was not achieved despite achieving a near perfect attitude error regulation. This could be the possible discussion for the future along with the usage of interpolated a_6, k_1 values between time steps for the online implementation of LBMPC with accurate nominal dynamics.

LBKF is a quite general technique for improving the performance of a state estimator in the presence of unmodeled (possibly nonlinear) dynamics. The method relies on identifying a a relationship between the filter residual process (which under a perfect model should be white) and some state or input signals in the system. A general challenge is to have enough training data to distinguish real trends from noise. This can be especially difficult for relationships involving non-measurable state variables, since these first must be filtered/smoothed. As with any adaptive approach, a blind application of the technique is not likely not achieve good results due to the overfitting to noisy data. We noted that a simple linear regression model yielded similar (even slightly better) results than a Gaussian process model. Using more than one input variable (beside the control signal) degraded the performance and was discarded early. In our application, it was obvious that the control signal path was the main source of model errors, and such domain knowledge is typically needed to achieve meaningful results. For asymmetrical and in general more nonlinear model errors, it could be worth pursuing a Gaussian process. To tune the Gaussian process model parameters, especially the kernel width, the general shape of the nonlinearity should ideally be known by the designer. To the best of our knowledge, there does not exist any closed-loop performance guarantees for learning-based Kalman filters. This is a major disadvantage of the approach. With noisy and insufficient amounts of training data and lack of validation the performance could easily degrade. In our example, training the Gaussian process blindly using all the default parameters in Matlab gave very poor estimation performance (not shown here), which in turn may would actually have destabilized the spacecraft if the filter were used in closed loop.

5 ACKNOWLEDGMENTS

The results presented in this paper have been achieved under funding by the ESA TDE programme with contract ESA contract No. 4000134108/21/NL/CRS. The view expressed in this paper can in no way be taken to reflect the official opinion of the European Space Agency. The authors from Lund university are members of the ELLIIT Strategic Research Area and partially funded by the European Research Council (ERC) under grant agreement No 834142 (ScalableControl).

REFERENCES

[1] S. Assilian, "Artificial intelligence in control of real dynamic systems.," Ph.D. dissertation, Queen Mary University of London, 1974.

[2] D. Girimonte and D. Izzo, "Artificial intelligence for space applications," *Intelligent Computing Everywhere*, pp. 235–253, 2007.

[3] S. Chien and R. Morris, "Space applications of artificial intelligence," *AI Magazine*, vol. 35, no. 4, pp. 3–6, 2014.

[4] G. Furano, G. Meoni, A. Dunne, *et al.*, "Towards the use of artificial intelligence on the edge in space systems: Challenges and opportunities," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 12, pp. 44–56, 2020.

[5] J.-G. Meß, F. Dannemann, and F. Greif, "Techniques of artificial intelligence for space applications - a survey," in *European workshop on on-board data processing*, European Space Agency, 2019.

[6] A. Russo and G. Lax, "Using artificial intelligence for space challenges: A survey," *Applied Sciences*, vol. 12, no. 10, p. 5106, 2022.

[7] E. Baroth, W. Powers, J. Fox, *et al.*, "IVHM (integrated vehicle health management) techniques for future space vehicles," in *37th Joint Propulsion Conference and Exhibit*, 2001, p. 3523.

[8] G. D. Zaiacomo, G. Medici, A. Princi, *et al.*, "Retalt: Development of key flight dynamics and gnc technologies for reusable launchers," *Proceedings of the International Astronautical Congress (IAC)*, September, 2022.

[9] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.

[10] D. Q. Mayne, M. M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[11] T. Lee, “Adaptive learning Kalman filter with Gaussian process,” in *2020 American Control Conference (ACC)*, 2020, pp. 4442–4447.

[12] A. Parsi, P. Anagnostaras, A. Iannelli, and R. S. Smith, “Computationally efficient robust MPC using optimized constraint tightening,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*, IEEE, 2022, pp. 1770–1775.