2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Verification of Low-Dimensional Neural Network Control

Johan Grönqvist and Anders Rantzer

***Abstract*— We verify safety of a nonlinear continuous-time system controlled by a neural network controller. The system is decomposed into low-dimensional subsystems connected in a feedback loop. Our application is a rocket landing, and open-loop properties of the two-dimensional altitude subsystem are verified using worst-case simulations. Closed-loop safety properties (crash-avoidance) of the full system are obtained from composition of contracts for open-loop safety properties of subsystems in a fashion analogous to the small-gain theorem.**

I. Introduction

Following in the footsteps of impressive applications using neural networks to control dynamical processes, verification of nonlinear controllers has seen progress in recent years [1].

Worst-case simulations [2] can give reachability bounds for general nonlinear systems, and techniques specialized for neural networks have been studied by several groups [3], [4], [5], [6], [7].

One issue with verification of neural network systems is the computational complexity of the method when the numbers of parameters and of nonlinear components in the neural network are large.

We use a computationally simpler method to verify a system with a given nonlinear controller. Our application is a rocket landing, schematically shown in Fig. 1. The system is decomposed into a closed loop shown in Fig. 2 with altitude and attitude subsystems, each controlled by its own neural network controller.

Department of Automatic Control, Lund University. johan.gronqvist@control.lth.se, anders.rantzer@control.lth.se

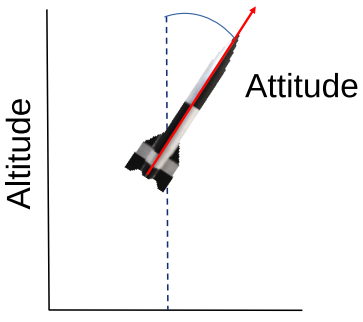

Fig. 1. Schematic view of the neural network controlled rocket landing scenario, indicating altitude and attitude (rotational) dynamics. We verify robust crash-avoidance via decomposition into subsystems.

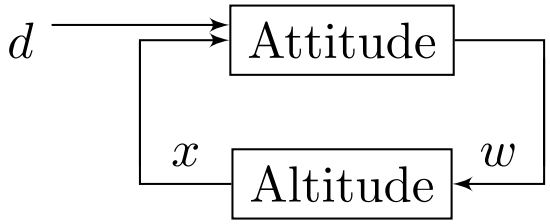

Fig. 2. Closed loop block diagram connecting attitude and altitude dynamics. Model uncertainties and the external disturbance d lead to deviations from nominal dynamics within one subsystem perturbing the other through w and x.

Similar decompositions have been used before, and we mention one study [8] decoupling satellite control into attitude control and momentum control. The study found near optimal performance and good robustness compared to using a single controller for all degrees of freedom.

Our model for the altitude subsystem takes the form of a double integrator with nonlinear terms and disturbances,

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ f(x) \end{bmatrix} + w, \tag{1}$$

where $f(x)$, representing altitude dynamics and neural network controller, contains model uncertainties. The disturbance w represents the effects of non-nominal attitude dynamics.

We obtain reachability bounds on the two-dimensional

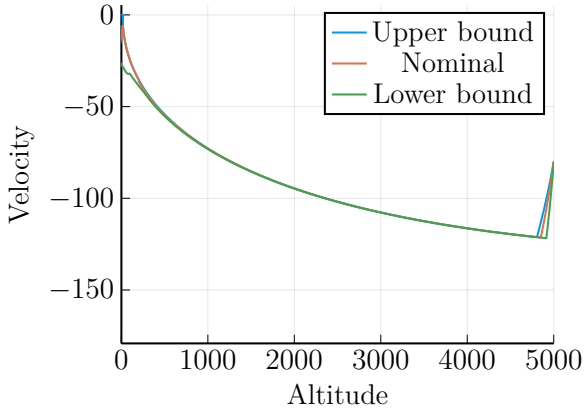

Fig. 3. Three curves are shown, the middle curve showing the nominal trajectory, and the upper and lower curves showing reachability boundaries. Where the three curves are very close, the bounds confine all trajectories to a narrow corridor through state space. (Parameters as in Fig. 6 with $w_2^{\max} = 3.0$.)

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

state-space of the altitude subsystem using worst-case simulations. Fig. 3 shows an example of good reachability bounds. The reachability bounds allow formulating a contract satisfied by the altitude subsystem in open loop, and we use a formalism [9] for composition of contracts to obtain closed-loop guarantees from open-loop safety properties of subsystems.

The next section describes the background of this work and a conceptual overview of our workflow for verification. We then present our worst-case simulation and composition of contracts, before discussing the details of the model, numerical implementation and results.

II. Background

This work builds on our experiences working within the AI4GNC project [10] funded by the European Space Agency. Within that project, neural network controllers were trained using reinforcement learning to perform safe landing in simulation with nonlinear models. One case modeled only altitude dynamics, while other cases included both altitude and attitude dynamics with a single controller for all states. Within the project, we used verification techniques describing nonlinearities using quadratic forms (similar to [3], [5]). Issues with computational complexity and conservativeness limited our successes. The work presented here is our followup work using computationally cheaper techniques.

We view altitude dynamics of a neural network controlled rocket in upright orientation as our nominal model for altitude dynamics. Deviations from nominal attitude dynamics are seen as disturbance inputs on altitude dynamics, and vice versa.

The altitude model reads (with y denoting altitude)

$$\begin{aligned} m\ddot{y} &= mf(y,\dot{y}) \\ &= T_{\max}u(y,\dot{y}) - mg_0\frac{R^2}{(R+y)^2} + c\dot{y}^2\exp(-y/H), \end{aligned} \tag{2}$$

where the right-hand-side terms denote control, gravity and air resistance[1]. We treat mass as an uncertain parameter, with a nominal mass of 80 tonnes. Our numerical results assume a mass between 40 and 100 tonnes.

The state vector is $x = \begin{bmatrix} y & \dot{y} \end{bmatrix}^\top$, and we add disturbances w to both states, giving us the model of Eq. (5).

The AI4GNC project did not work with a separate model and controller for attitude dynamics only, and we do not analyze attitude dynamics in detail, but only formulate the open-loop contract it is required to satisfy to ensure closed-loop safety.

III. Workflow

This section outlines the conceptual steps required, while details are left for later sections.

1) We decompose the system so that our subsystem validation is limited to models with state-space in $\mathbb{R}^2$. Subsystems are connected using feedback and our closed loop system is shown in Fig. 2.
2) In the two-dimensional state space of a single subsystem, we locally define "worst-case" behaviors of uncertainties and disturbances, and form a "worst-case" ODE whose solutions act as reachability bounds for the subsystem dynamics. A conceptual description and implementation details are given in sections V and VIII, respectively.
3) Reachability bounds are formulated as open-loop contracts on subsystems, capturing the subsystem properties of interest. Specifics for our application are discussed in sections IX and XI.
4) Results on composing contracts [9], as discussed in section VI, give a closed-loop contract from open-loop contracts.
5) Our closed-loop contract implies the system safety (for us, crash avoidance), and our verification is successful.

IV. Notation

We write $x^\top$ for the transpose of x, and $x^\top n$ for the inner product between x and n. A dotted letter, as in $\dot{x}$, is used to denote the time-derivative of a function. The sign function is denoted $\operatorname{sgn}(x)$.

V. Worst-Case Simulation

This section introduces our low-dimensional worst-case simulations. Model and implementation details will be described in later sections.

In $\mathbb{R}^N$ trajectories are 1-dimensional and reachability boundaries are $N-1$-dimensional. In $\mathbb{R}^2$ both trajectories and boundaries are 1-dimensional curves, and reachability boundaries can be obtained from a "worst-case" ODE. If two trajectories of our model cross, it is due to uncertainties or disturbances. At each point in state space, we characterize worst-case values of uncertainties and disturbances, making our safe landing most difficult. This leads us to a "worst-case" ODE whose solutions show "worse" behavior than any trajectory compatible with our altitude dynamics[2].

More precisely, we assume a dynamical system given as a differential inclusion with trajectories x satisfying $\dot{x}(t) \in F(x(t))$ for a given function $F : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$.

A boundary is a curve $z : [a,b] \longrightarrow \mathbb{R}^2$ with outward direction specified by a set of normal vectors $n : [a,b] \longrightarrow \mathbb{R}^2$ satisfying $|n(t)| > 0$ and $\dot{z}(t)^\top n(t) = 0$. The curve z is a *boundary* to our dynamical system if $x(t) = z(t')$ implies $\dot{x}(t)^\top n(t') < 0$ for any system trajectory x.

The additional property required to ensure that no trajectory crosses z in the outward direction is (for all t)

$$\dot{z}(t)^\top n(t) > \max_{\delta z \in F(z(t))} \delta z^\top n(t) \tag{3}$$

[1]The velocity will always be negative, and air resistance will always be positive.

[2]Our model includes only additive noise. With disturbances on the input to a neural network, a "worst-case" ODE would require reachability bounds on the neural network.

For any trajectory x, boundary curve z and crossing point $x(t) = z(t')$, it follows that

$$\begin{aligned}\dot{x}(t)^\top n(t') &\leq \max_{\delta x \in F(x(t))} \delta x^\top n(t') \\ &= \max_{\delta z \in F(z(t'))} \delta z^\top n(t') \\ &< \dot{z}(t')^\top n(t') = 0\end{aligned} \tag{4}$$

showing that $\dot{x}(t)^\top n(t') < 0$[3]

In words, at any state $z(t)$, the tangent $\dot{z}$ of the boundary curve z points "more" outward than the tangent $\dot{x}$ of the "worst-case" solution x to the differential inclusion. Therefore no trajectory can cross it in the outward direction.

We thus arrive at reachability bounds via a first order ODE in $\mathbb{R}^2$ in which the neural network appears as a deterministic nonlinear function in our model. Our verification scheme scales well with the size of the neural network involved.

As an example of reachability bounds, Fig. 3 shows the nominal trajectory together with two boundary curves. The boundary curves are valid for some specified uncertainty bounds and no trajectory compatible with the uncertainty bounds can cross the upper or lower boundary curves in the outward direction. The boundary curves show that trajectories are restricted to a narrow corridor, providing good reachability bounds for the model and controller used here.

VI. Contracts for Closed Loop Guarantees

In this section, we discuss the assume-guarantee framework [9] used to obtain closed loop guarantees.

The block-diagram in Fig. 2 shows our interconnected system, and will formulate subsystem contracts that hold in open-loop for the two subsystems.

Our constraints will be specified by set-inclusions of the form $d(t) \in D$, $w(t) \in W$ and $x(t) \in X$ $(\forall t)$, for the external input d, and the internal inputs w and x, shown in Fig. 2. The sets W and D will be boxes, and the set $X \subset \mathbb{R}^2$ will contain the points between upper and lower boundary curves in graphs like Figs. 3 and 9.

Our open-loop contracts for the subsystems are

Contract 1 (Altitude Dynamics)
Assume: *Under specified model uncertainties, and with $w(t) \in W$ $(\forall t)$.*
Guarantee: *Until we reach an altitude of 15 meters, we have $x(t) \in X$ $(\forall t)$. We eventually reach 15 meters altitude, and we do so with a safe downward speed.*

Contract 2 (Attitude Dynamics)
Assume: *Under specified model uncertainties, and with $d(t) \in D$ and $x(t) \in X$ $(\forall t)$.*
Guarantee: *We have $w(t) \in W$ $(\forall t)$*

Our desired closed-loop contract is

Contract 3 (Closed Loop)
Assume: *Under specified model uncertainties and with $d(t) \in D$ $(\forall t)$.*
Guarantee: *We will eventually reach 15 meters altitude and we will do do with a safe speed.*

We remark on two properties of the contracts.

Firstly, we note that the structure of the open-loop contracts match the structure of the block-diagram. Each open-loop contract for a subsystem assumes bounds on its inputs and provides guarantees on its outputs. The assumptions on internal inputs in the subsystem contracts, (w and x in the altitude and attitude contracts, respectively) are guaranteed by the other subsystem contract. Remaining assumptions, on model uncertainties and the external input d, are assumptions in the closed lop contract. The guarantee in the closed loop contract is an immediate consequence of guarantees in the subsystem contracts.

Secondly, the wording of the assumptions and guarantees in the three contracts are chosen to suit our subsystem verification method and our closed-loop safety criterion. The formalism for composition of contracts i agnostic to the formulations of the clauses in the contracts and to the verification method.

Following Theorem 2 of the paper [9] presenting the assume-guarantee framework, the closed-loop contract is valid in closed loop if all subsystems satisfy their open-loop contracts in open loop, and two additional technical assumptions hold. The first requires a one-sided continuity property, stating that if a contract is valid for all times less than t, it is also valid at time t. The second is a property of extending the time interval of validity, and is only required for one block per cycle in the block diagram (we choose our altitude block). If the altitude contract is valid until time t, then the guarantee clause is valid until some (slightly) later time t'.

Our subsystem state $x(t)$ does not jump, and our contracts are formulated using set inclusion with closed sets. A first violation of a contract can therefore only happen at some specific time t, ensuring the first property. The second property is ensured by our worst-case simulation actually being slightly "worse" than "worst-case". This is easy to implement and ensures that we always have a margin. With continuous dynamics (without jumps), if a contract with a margin is valid until a time t, there is a later time t' up until which the guarantee clause of the contract also holds.

VII. Model for Altitude Control

While our method is applicable for more general cases, our verification work is based around a simple model of altitude control.

Our model has a state $x(t) \in \mathbb{R}^2$ and disturbances $w(t) \in \mathbb{R}^2$ and reads

$$\dot{x}(t) \in \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix} x(t) + \begin{bmatrix}0\\ f(x(t))\end{bmatrix} + w(t) = F(x(t)) \tag{5}$$

The function $f(x) = f(x_1, x_2)$ as introduced in Eq. (2) is a set-valued function due to the uncertain mass, and the disturbances $w = \begin{bmatrix}w_1\\ w_2\end{bmatrix}$ are also set-valued with ∞-norm bounds $|w_1(t)| < w_1^{\max}$, $|w_2(t)| < w_2^{\max}$ $(\forall t)$, for fixed

[3] We use strict inequalities to avoid crossings where the tangents of x and z are parallel at the crossing.

values $w_1^{\max}$ and $w_2^{\max}$. In the nominal model, with nominal mass and $w = 0$, $F(x)$ takes a single value and our nominal model is an ODE.

VIII. Numerical Implementation

The model as presented in Eq. (5) is a first order differential inclusion with state in $\mathbb{R}^2$ and uncertainties parameterized by m and w, and therefore matches the discussion in section V. We now describe our numerical implementation of the worst-case simulation.

Our notion of boundary requires knowing the tangent of the boundary. We estimate the local tangent along a curve $z(t)$ using an exponentially decaying average of the values of $\dot{z}(t)$. We introduce an additional state $v(t) \in \mathbb{R}^2$, with the dynamics

$$\dot{v}(t) = -\lambda v(t) + \dot{z}(t) \tag{6}$$

The timescale $1/\lambda$ is chosen to be much shorter than those of our model dynamics, but much longer than the step-sizes used by our ODE solver[4]

Given the estimate v of the local tangent, the local outward normal $n(t)$ is obtained by rotating $v(t)$ by 90 degrees. We use clockwise and counterclockwise rotations for the upper and lower bounds in Fig. 3, respectively.

In the worst-case ODE, we compute $n(t)$ at every step in time, and select the values of $F(z(t))$ maximizing the inner product with n, allowing all uncertainties and disturbances to take on worst-case values. Componentwise, the increment $\dot{z}_i(t)$ should be maximal if $n_i(t) > 0$ and minimal otherwise. We additionally add a small value ε to ensure that our boundary curve is "worse" than "worst-case".

Starting from the nominal initial condition x_{init}, we now perform simulations of the nominal model with state x and $w = 0$, as well as two simulations in the worst-case model with extended state $[z, v]$. The two simulations differ in the direction of rotation to obtain n from v.

If our two boundary trajectories reach a state with $z_1 = 15$, we stop our simulations there and connect the endpoints of the upper and lower boundary trajectory by a line at $z_1 = 15$.

Worst-case simulation with the two different normal directions give us bounding trajectories that will not be crossed by any trajectory of our uncertain system. Trajectories are therefore guaranteed to stay within the region until they exit at $z_1 = 15$, between the upper and lower bound[5].

IX. Robustness

In the present work, we discuss two questions

- For specified initial altitude and downward speed, how large disturbances from non-nominal attitude dynamics can we tolerate?
- For specified initial altitude and bounds on disturbances from non-nominal attitude dynamics, how large initial downward speed can we tolerate?

This section will only concern open-loop bounds on altitude dynamics, and contract composition will be used to ensure that the conclusions for the open-loop properties of the altitude subsystem also hold in closed loop.

Our robustness analyses focus on x^{init}, containing initial altitude and velocity, $w_1^{\max}, w_2^{\max}$ bounding our disturbances, and the velocity at 15 meters altitude.

We require reachability boundaries to ensure negative velocity throughout (to ensure that we reach 15 meters eventually), and to ensure that we have a safe velocity at 15 meters altitude (crash-avoidance). Concretely, we require that

- the upper boundary satisfies $z_2(t) < -\varepsilon$ $(\forall t)$ for some $\varepsilon > 0$,
- the lower boundary reaches $z_1(t) = 15$ for some t with a safe final velocity $z_2(t)$.

We assume that a safe velocity is given, such that if the rocket reaches an altitude of 15 meters with a higher velocity[6], the controller that takes over for the last 15 meters is guaranteed to land safely.

The first question posed in this section is answered by asking for the largest $w_1^{\max}$ and $w_2^{\max}$ that still ensure the two requirements on boundary curves for specified x^{init}.

The second question is answered by asking for the most negative $\dot{x}_2^{\text{init}}$ that still ensures the two properties for specified $w_1^{\max}, w_2^{\max}$ and x_1^{init}.

In all cases, the corresponding contract on attitude dynamics, as discussed in section VI must be fulfilled and additional technical assumptions satisfied. Without those additional requirements, we could only discuss open-loop robustness of altitude dynamics. The additional requirements allow us to conclude that the closed-loop contract holds, guaranteeing the same safety properties in closed loop.

X. Numerical results

Our numerical results use $x^{\text{init}} = [5000, -80]^\top$, except for Figs. 8 and 9.

We first set $w_2^{\max} = 0$, and consider how the impact velocity varies with $w_1^{\max}$. Figs. 4 and 5 show the impact of $w_1^{\max}$ on the final velocity. The effect is small for small values of $w_1^{\max}$, and there is a very sharp change to a very large impact around $w_1^{\max} = 0.11$.

In our closed loop, our altitude dynamics is driven by deviation from nominal attitude dynamics affecting the acceleration in altitude dynamics. Acceleration is determined by a disturbance in x_2, and we now set $w_1^{\max} = 0$ and plot final velocity vs. $w_2^{\max}$. Results shown in Figs. 6 and 7 show a gradual increase in the effect of disturbances on acceleration.

The actual bound on the final velocity must be set based on a value for the safe impact velocity, and we arbitrarily

[4]The actual numerical curve produced by the ODE solver we use contains unphysical fluctuations on timescales of the solver step size, which is much shorter than the timescales of the physical dynamics. The vector $v(t)$ is a good approximation to the tangent of the physical trajectory, without being sensitive to the unphysical fluctuations in the ODE-solver.

[5]As already remarked, the model and controller were intended to be used for altitudes above 15 meters, and our characterization of crash-avoidance therefore depends on the velocity it has when reaching an altitude of 15 meters.

[6]The velocities are negative and higher velocity corresponds to lower downward speed.

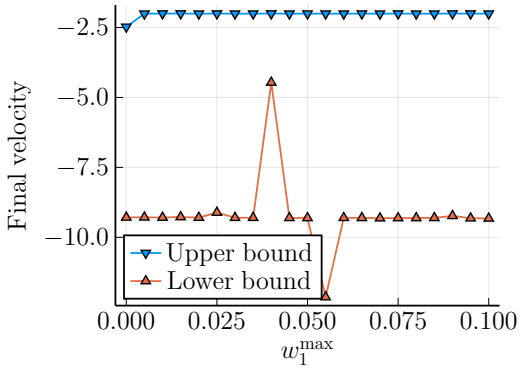

Fig. 4. Upper and lower bounds on final velocity for $w_2^{\max} = 0$ and $w_1^{\max} < 0.1$. The figure shows that the impact of small disturbances on the final velocity is small.

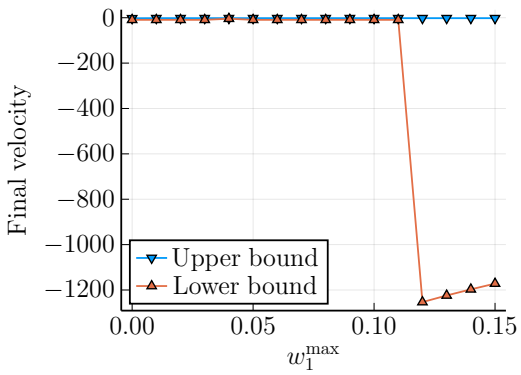

Fig. 5. Upper and lower bounds on final velocity for $w_2^{\max} = 0$ as varying $w_1^{\max}$. At $w_1^{\max} \approx 0.11$, there is a transition to very large impact of the disturbance on the final velocity.

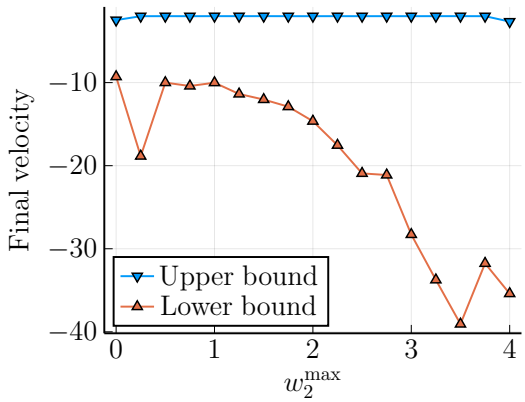

Fig. 6. Upper and lower bounds on final velocity for $w_1^{\max} = 0$ and $w_2^{\max} < 4$. The figure shows that the impact of small disturbances on the final velocity is small.

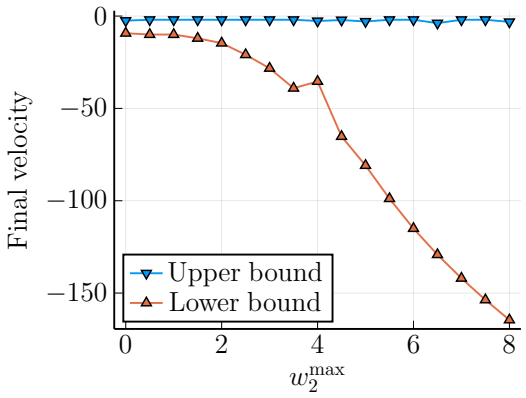

Fig. 7. Upper and lower bounds on final velocity for $w_1^{\max} = 0$ and varying $w_2^{\max}$. For $w_2^{\max} > 4$, we see a gradually growing impact of the disturbance on the final velocity.

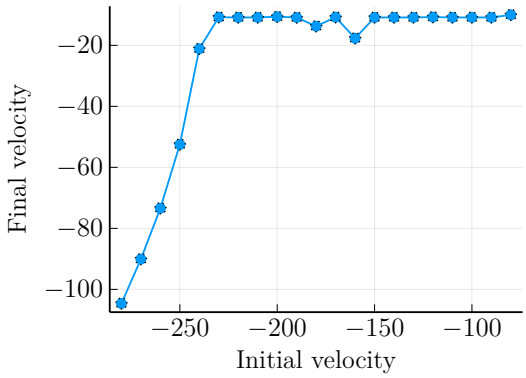

Fig. 8. With $w_1^{\max} = 0$ and $w_2^{\max} = 1$, we plot initial velocity vs. final velocity. We are within our safety limit for initial velocities down to -230.

choose -20m/s as our largest safe velocity at an altitude of 15 meters, for our numerical examples.

For our open-loop altitude subsystem, we numerically find

$$(w_1(t) = 0 \text{ and } |w_2(t)| < 2.3) \implies \text{safe landing.} \quad (7)$$

Turning to the second question posed in section IX, we keep the initial altitude at 5000, set disturbance bounds $w_1^{\max} = 0$ and $w_2^{\max} = 1$, and find the largest initial downward speed we may still consider safe.

Fig. 8 shows initial velocity vs. final velocity, and we see that the initial velocity can reach -230m/s without much impact on the final velocity, but that the impact starts to become large for larger deviations.

Allowing initial velocities in the interval $[-230, -80]$, we perform worst-case simulation with uncertain initial velocity, in addition to the uncertainties discussed above. Upper and lower boundaries are shown in Fig. 9 together with the nominal trajectory. The upper boundary and nominal trajectory are close, as the nominal initial velocity is the upper bound on the allowed interval. The large range of allowed initial velocities gives a large region between the upper and lower boundaries, and this region will form the assumptions in the contract attitude dynamics is required

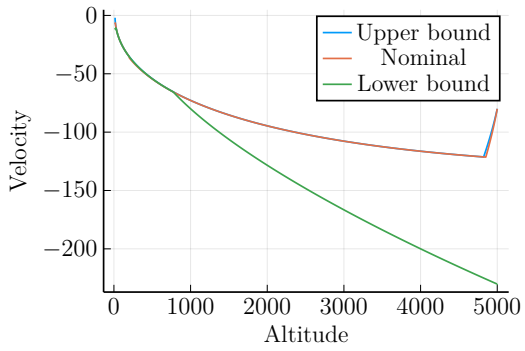

Fig. 9. Upper and lower boundary curves for the case in Fig. 8 with uncertain initial velocity in the range $[-230, -80]$. The upper boundary and the nominal trajectory nearly coincide, while the lower boundary deviates strongly from the nominal curve due to the uncertain initial velocity.

to satisfy. While we still have a safe final velocity, we see that the requirements on the attitude controller will be much stricter than in previous cases, due to the large deviations from nominal altitude dynamics allowed. This shows the trade-off between the strictnesses of the contracts required for the subsystems of altitude and attitude dynamics. This trade-off is again analogous to the structure of the small-gain theorem.

XI. Requirements on Attitude Dynamics

In previous sections, We analyzed the altitude dynamics appearing as the lower part of the block diagram Fig. 2. We obtained bounds on the deviation $x - x^{\text{nominal}}$ from bounds on w. The next step is to combine this with bounds on the upper block in the block-diagram, our model for nominal attitude dynamics. The model would describe nominal attitude dynamics with uncertainties for mass, and would be driven by altitude dynamics via $x - x^{\text{nominal}}$ and by wind disturbances via d.

From the contract forms discussed in section VI and our numerical simulation for altitude dynamics, we find that the requirement on attitude dynamics.[7]

Contract 4 (Attitude Dynamics)
Assume: $d(t) \in D$, $40000 < m(t) < 100000$ *and* $|x - x^{\text{nominal}}| < 15$
Guarantee: $w_1 = 0$ *and* $|w_2| < 2.3$

We do not describe details of the attitude dynamics here, but it is similar to an inverted pendulum controlled to behave like a damped oscillator, driven by disturbances $x - x^{\text{nominal}}$ and d.

XII. Discussion

The verification presented here decomposes the full system into subsystems, and uses worst-case simulation for subsystems with two-dimensional state-space. Composition of contracts [9] is used to obtain guarantees in closed loop.

[7]The value 15 is the maximal distance between a point on the lower bound curve and its nearest point on the nominal trajectory.

The essential feature enabling our analysis is the dimensionality of the subsystems, and we impose few other requirements on their structure. In particular, our method scales well with the size and complexity of the nonlinearity, represented in our application by a neural network.

The result is enabled by two observations.

- Boundaries and trajectories have the same geometric structure, as curves in $\mathbb{R}^2$, boundaries can be obtained from numerically solving an ODE.
- Reachability bounds on subsystems can be combined using assume-guarantee contracts in the spirit of the small-gain theorem.

XIII. Acknowledgments

This work builds on work performed in collaboration with Deimos and others as part of the AI4GNC project[10]. The model and the RL-trained neural network controller were produced by Deimos as part of their AI4GNC work. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The authors are members of the ELLIIT Strategic Research Area at Lund University.

References

[1] M. Everett, "Neural network verification in control," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 6326–6340.
[2] L. El Ghaoui and G. Calafiore, "Worst-case simulation of uncertain systems," in *Robustness in identification and control*, A. Garulli and A. Tesi, Eds. London: Springer London, 1999, pp. 134–146.
[3] H. Yin, P. Seiler, and M. Arcak, "Stability analysis using quadratic constraints for systems with neural network controllers," *IEEE Transactions on Automatic Control*, vol. 67, no. 4, pp. 1980–1987, 2021.
[4] T. Chen, J. B. Lasserre, V. Magron, and E. a. Pauwels, "Semialgebraic optimization for lipschitz constants of relu networks," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. L. in, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 19 189–19 200.
[5] M. Everett, G. Habibi, and J. P. How, "Efficient reachability analysis of closed-loop systems with neural network controllers," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4384–4390.
[6] H. Hu, M. Fazlyab, M. Morari, and G. J. Pappas, "Reach-sdp: Reachability analysis of closed-loop systems with neural network controllers via semidefinite programming," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 5929–5934.
[7] A. Sadeghzadeh and P.-L. Garoche, "Reachability analysis of linear parameter-varying systems with neural network controllers," in *2022 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2022, pp. 1372–1377.
[8] M. Elgersma, G. Stein, M. Jackson, and J. Yeichner, "Robust controllers for space station momentum management," *IEEE Control Systems Magazine*, vol. 12, no. 5, pp. 14–22, 1992.
[9] A. Saoud, A. Girard, and L. Fribourg, "Assume-guarantee contracts for continuous-time systems." *Automatica*, vol. 134, 2021.
[10] P. Rosa, J. P. Belfo, N. Somma, A. Botelho, G. Tofanelli, J. I. Bravo, R. Hinz, J. M. Lemos, F. Parente, B. Costa, J. Igreja, A. Marcos, D. Navarro-Tapia, M. Martinez, A. Rantzer, A. Cervin, V. Renganathan, J. Gronqvist, J. Belhadj, M. Casasco, and S. Bennani, "Artificial intelligence techniques for gnc design, implementation, and verification (ai4gnc)," *ESA GNC conference*, 2023.