Webology (ISSN: 1735-188X)
Volume 19, Number 3, 2022

Rocket Engine – A Rocket Engine Propulsion Package In The Tool Command Language (Tcl)

Frank Morlang

German Aerospace Center (DLR)

Abstract

DLR's Space Liner orbiter concept flight simulator bases on a thrustless flight dynamics model for associated descent/approach trajectory analyses. Investigations of the rocket-propelled ascent phase demand the implementation of simulated thrust. For that purpose, a Tcl rocket engine propulsion package has been developed. Its configurable parametrization including the transient regime is described. Package usage test results in a standalone application show a thrust update interval below 4 milliseconds (median). Future options of different human-in-the-loop real-time system of systems integration capabilities are presented and discussed.

Keywords: aerospace, Tcl/Tk, distributed simulation

Nomenclature

A_{ne}	=	nozzle exit cross section area
A_{nt}	=	nozzle throat cross section area
dm/dt	=	mass flow
F	=	thrust
□	=	isentropic exponent
M	=	molecular weight of the exhaust species
p_{at}	=	atmosphere pressure
p_{ch}	=	chamber pressure
p_{ne}	=	nozzle exit pressure
R	=	universal gas constant
T_{ch}	=	chamber temperature
v_{ne}	=	exhaust gas velocity at nozzle exit
WCET	=	worst-case execution time
WCTT	=	worst-case transmission time

1. Introduction

Space Liner is DLR's advanced concept for a suborbital, hypersonic, winged passenger transport [1]. A thrustless flight dynamics model for the commercial flight simulation software "X-Plane" (Figure 1) has been developed [2].

Figure 1. Space Liner during simulated final approach

It is used in integration examinations of space traffic hypersonic gliding descent trajectories. To apply the simulation in future analyses of rocket propelled flight phases, the need for the incorporation of a thrust model arose. For its development, the following requirements were identified:

- Keep it as simple as possible with a minimum of development effort on the one hand.
- Allow a simple and flexible integration in the simulation environment on the other hand.
- Address the capability of its use in a future system of systems context.

Beside Tcl's general-purpose and rapid prototyping strengths, the potential to benefit from an X-Plane interface package [3] drove the decision to realize a thrust model in Tcl.

2. Method

The rocket engine related package input parameters are shown in Figure 2. Their abbreviations p_{ch}, T_{ch}, A_{nt}, A_{ne}, p_{ne} and p_{at} refer to the chamber pressure, chamber temperature, nozzle throat cross section area, nozzle exit cross section area, nozzle exit pressure and atmosphere pressure, respectively.

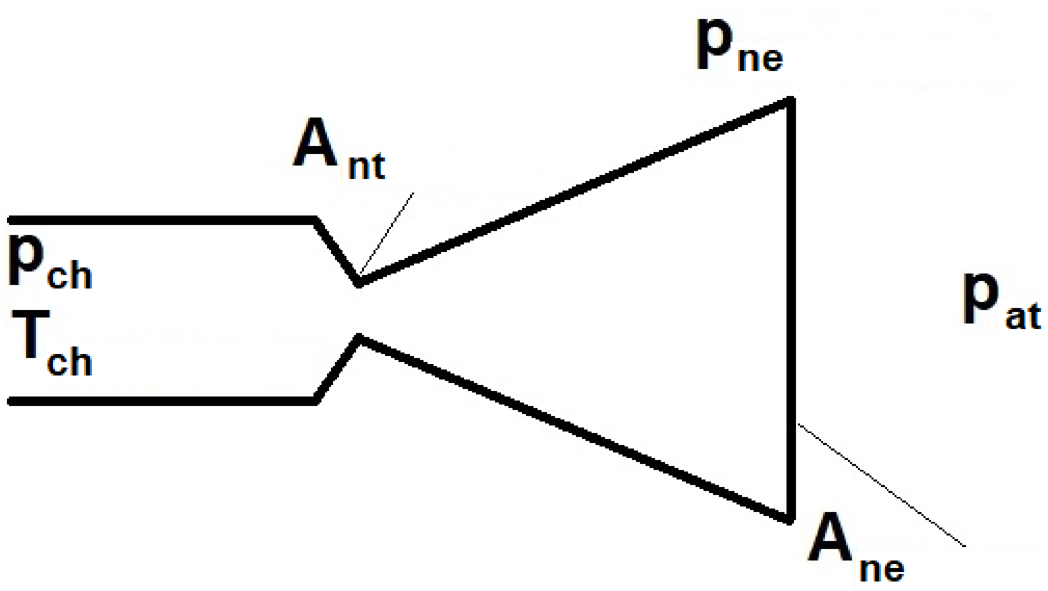

Figure 2. rocket Engine input parameters

Fuel specific characteristics are addressed by the additional input parameters R, M and (universal gas constant, molecular weight of the exhaust species and isentropic exponent). The thrust is given by:

$$F = \frac{dm}{dt} v_{ne} + A_{ne} (p_{ne} - p_{at}) \qquad (1)$$

Here, dm/dt and v_{ne} refer to the mass flow through the nozzle and the exhaust gas velocity at nozzle exit, which are defined as follows:

$$\frac{dm}{dt} = \frac{A_{nt} p_{ch} \gamma}{\sqrt{\frac{\gamma R T_{ch}}{M}}} \sqrt{\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{\gamma-1}}} \qquad (2)$$

$$v_{ne} = \sqrt{\frac{RT_{ch}}{M} \frac{2\gamma}{\gamma-1} \left[1 - \left(\frac{p_{ne}}{p_{ch}}\right)^{\frac{\gamma-1}{\gamma}}\right]} \qquad (3)$$

The package consists of three internal and two external procedures:

- ::rocket Engine::Calculate Gas Exit Velocity
- ::rocket Engine::Calculate Mass Flow

- ::rocket Engine::Calculate Max Thrust
- ::rocket Engine::in I Defines
- ::rocket Engine::calculate Thrust

Calculation of exhaust gas velocity at nozzle exit, mass flow and maximum thrust is done by the internal procedures according to (3), (2) and (1). When the package is used, the input parameters firstly have to be initialized with the external procedure ::rocket Engine::ini Defines. The actual thrust can then be acquired via the external procedure ::rocket Engine::calculate Thrust, taking the elapsed time since engine start in milliseconds and the atmospheric pressure in Pascal as input parameters. The transient regime to the establishment of the maximum thrust phase is approximated with a two-phase approach according to Figure 3. Its parametrization is done in ::rocket Engine::ini Defines by the parameters time1, time max thrust and fraction, where fraction refers to thrust1/thrust max (Figure 3). Random thrust fluctuations during the three phases (1st phase, 2nd phase and max. thrust phase) can be defined with three spread parameters.

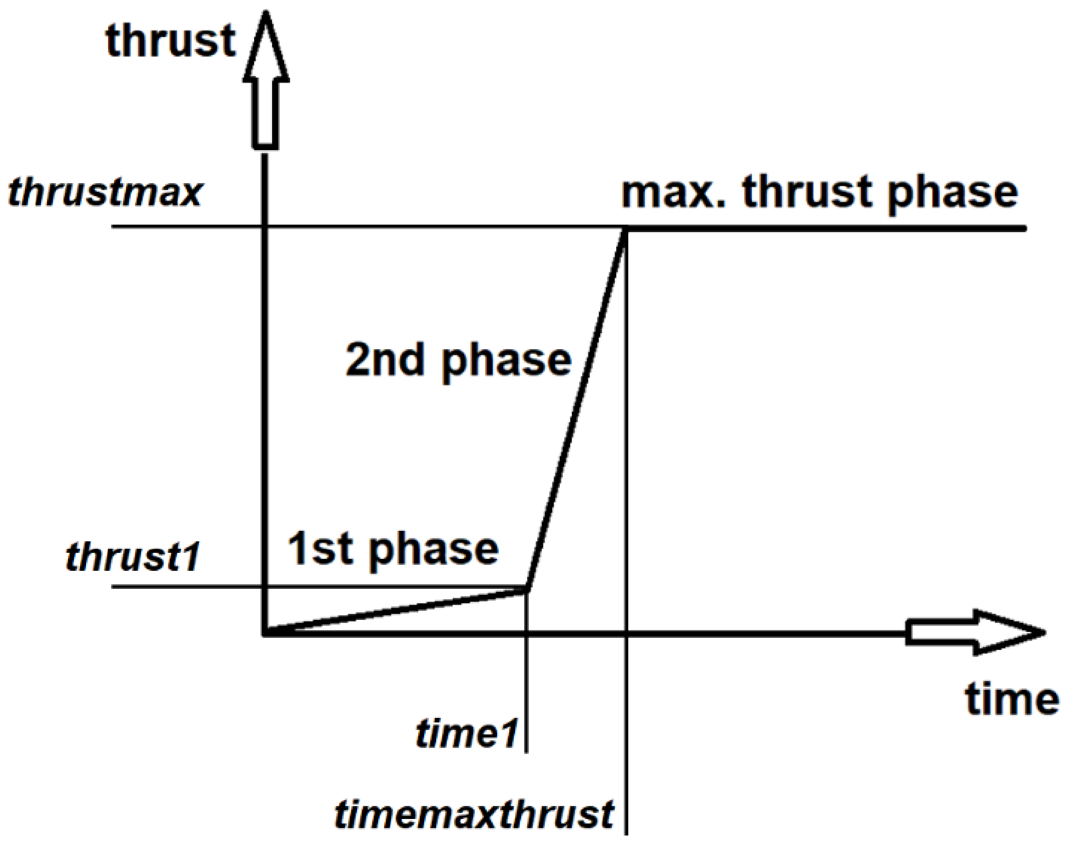

Figure 3. Maximum thrust establishment approximation

3. Results

A local standalone package usage test with the following parameters was made:

• A_{nt}:	0.042 m^2
• A_{ne}:	0.900 m^2
• p_{ch}:	9720000 Pa
• p_{ne}:	43008 Pa
• p_{at}:	101325 Pa
• T_{ch}:	3006 K
• M:	22 kg/kmol
• □:	1.22
• time1:	1 s
• time max thrust:	3 s
• fraction:	10 %
• spread1:	15 %
• spread2:	10 %
• spread3:	2 %

The configuration of hardware, operation system and software are shown in Table 1.

Computer	Lenovo MIIX 320-10ICR 4000 MB memory Intel® Atom(TM) x5-Z8350 CPU @ 1.44GHz Windows 10 Home 64bit (1909)
Tcl/Tk version	Tcl/Tk 8.6.10 (64 bit)

Table 1. Test configuration

The thrust update interval fluctuation of a 10 seconds run lies in a corridor between 1 ms and 15 ms with a median of 3.3 ms (Figure 4 and Figure 5).

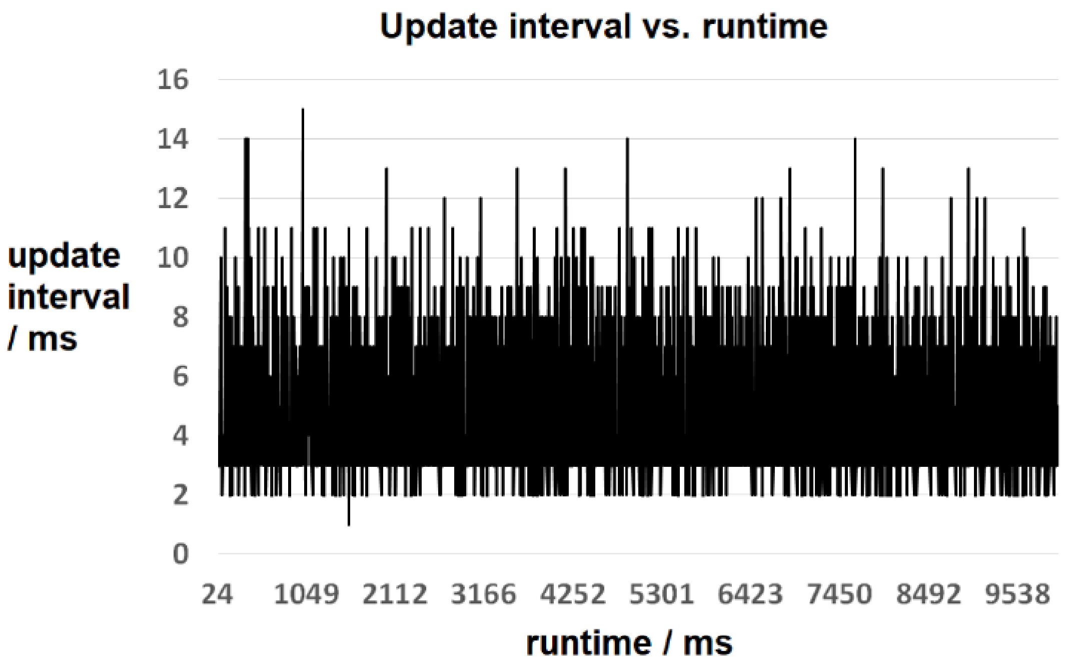

Figure 4. Thrust update interval fluctuation

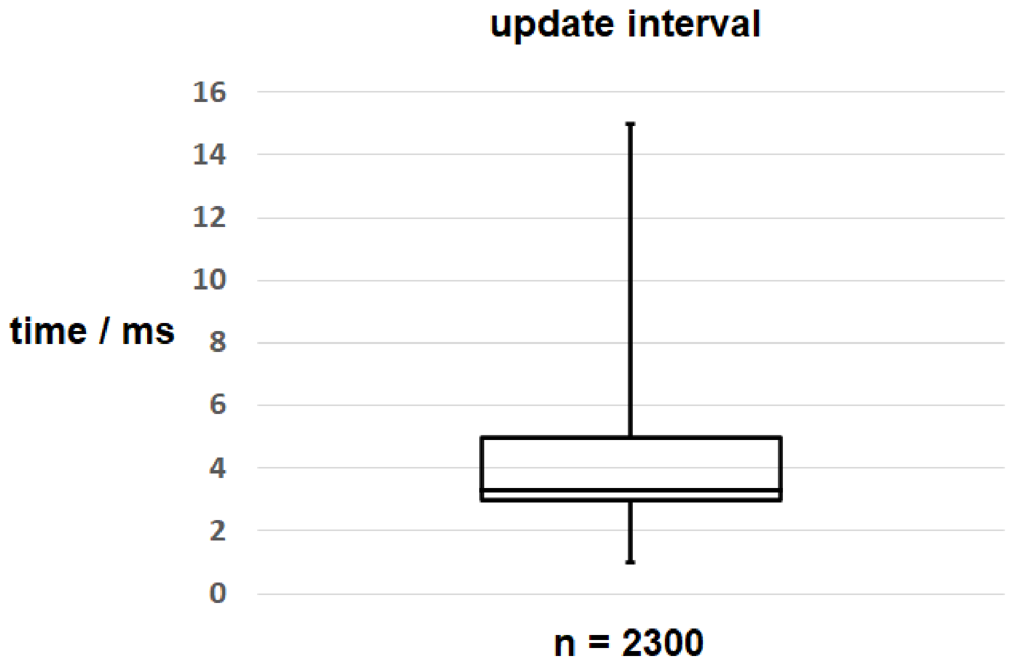

Figure 5. Thrust update interval distribution

The thrust evolution is shown in Figure 6 with a thrust value User Datagram Protocol (UDP) send-receive delay ranging between 1.6 ms and 9.1 ms and a median of 4.1 ms (Figure 7).

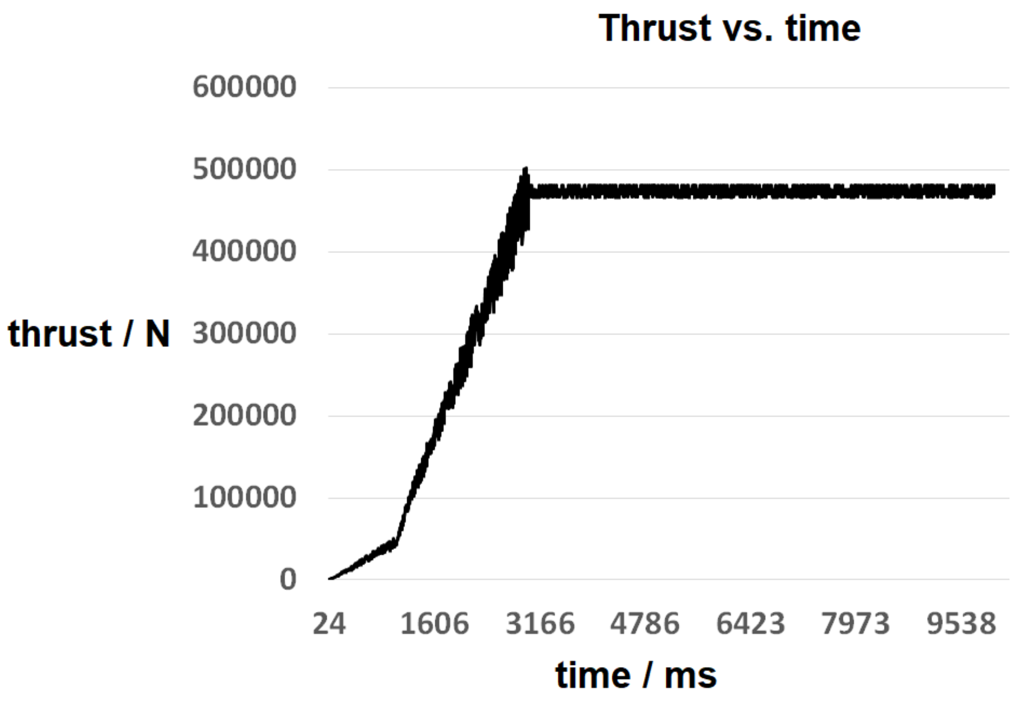

Figure 6. Thrust evolution

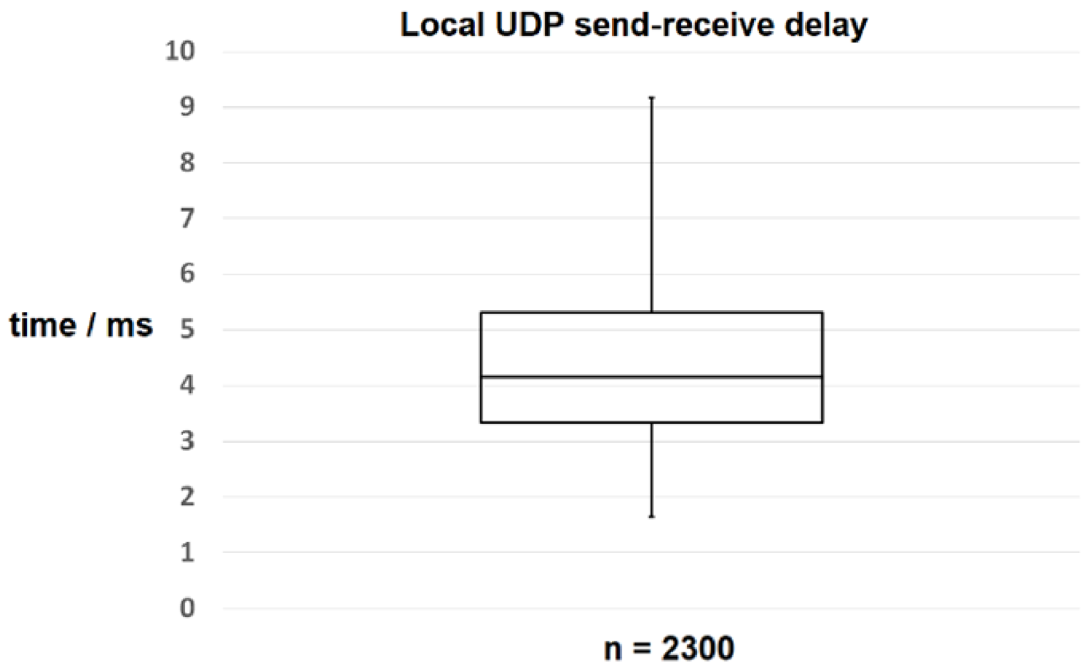

Figure 7. Local UDP send-receive delay distribution

4. Discussion

Taking the maximum thrust update interval of 15 ms as worst-case execution time (WCET) and the maximum UDP send-receive delay of 9.1 ms as worst-case transmission time (WCTT) reveals:

$$\text{WCET} + \text{WCTT} = 24.1 \text{ ms} \qquad (4)$$

This does not meet a 50 Hz hard real time request but gets its relativization by the test set-up on a cheap, low performance computer. The distributions of the thrust update interval and UDP send-receive delay with the associated medians of 3.3 ms and 4.1 ms show a soft real time performance suitable for pilot in the loop simulations at 50 Hz [5].

5. Conclusion

A Tcl rocket engine propulsion package has been developed, taking typical rocket engine design parameters as input values. It allows simple and flexible integration in the simulation environment with a soft real time performance of 50 Hz. Its usage in a future deployment of a rocket engine thrust federate in a distributed spacecraft simulation (Figure 8) is planned.

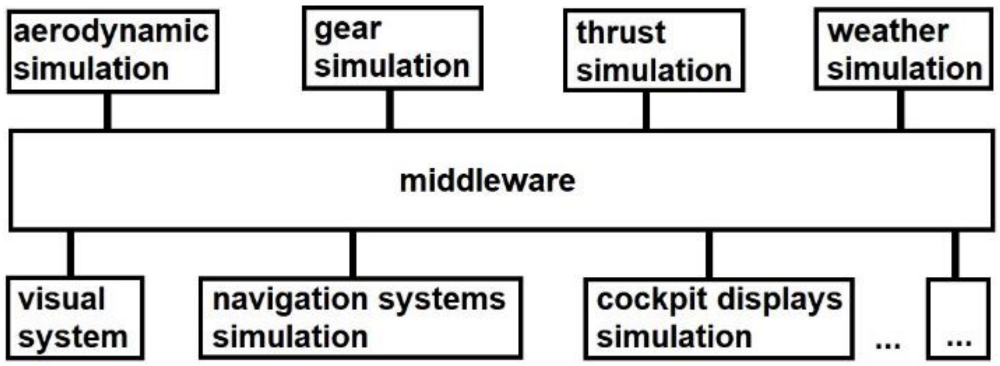

Figure 8. Distributed simulation architecture

Here, further aspects to be analyzed will refer to differences between loose and tight coupling and their influence on the overall simulation ensemble performance. More evolved versions of the package will benefit from alternative calculation procedures in embedded C code for the CriTcl C Runtime in Tcl [6].

6. References

[1] M. Sippel, Editor. System Design of the SpaceLiner Project and Its Latest Technical Progress. Proceedings of the 20th Annual Meeting of the China Association for Science and Technology, **(2018)**; Hangzhou, China.

[2] F. Morlang, Editor. Supersonic and hypersonic flight dynamics realization for the Space Liner real-time Human-in-the-Loop Space Flight Simulator. Proceedings of the 5th International Conference on Recent Trends in Computer Science and Electronics, **(2020)**; Honolulu, USA.

[3] F. Morlang, Editor. Quick and dirty I/O playing with X-Plane without "dirty" - the Tcl way. Proceedings of the 26th Annual Tcl/Tk Conference, **(2019)**; Houston, USA.

[4] European Aviation Safety Agency, Editor, Certification Specifications for Aeroplane Flight Simulation Training Devices, Annex to ED Decision 2012/010/, **(2012)**

[5] G. Dussart, V. Portapas, A. Pontillo and M. Lone, in Flight Physics - Models, Techniques and Technologies, Edited K. Volkov, In Tech, London **(2018)**, pp.49-72.

[6] A. Kupries, Editor. C Runtime In Tcl. Proceedings of the 23rd Annual Tcl/Tk Conference, **(2016)**; Houston, USA.

Author

Frank Morlang

1999 Diploma in Engineering in Materials Science of Technical University of Darmstadt

1999-2001 Project engineer at Aerodata company Braunschweig
2001 - today Researcher at Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Center), DLR, Braunschweig