Óbuda University

PhD Dissertation

Tools for Efficient Soft Computing Modelling and Feasible Optimal Control of Complex Dynamic Systems,

with Application to Multi-Rotor Unmanned Aerial Vehicle Navigation and Obstacle Avoidance

Nemes Attila

Prof. Dr. Mester Gyula

Doctoral School on Safety and Security Sciences

Budapest, 2017

Examination Committee / Szigorlati Bizottság:

Head of the Examination Committee / Elnök:

Prof. Dr. Rajnai Zoltán, egyetemi tanár, ÓE

Participants / Tagok:

Prof. Dr. Somló János, egyetemi tanár, ÓE

Dr. habil. Farkas Tibor, egyetemi docens, külső NKE (external)

Public Defence Committee Members / Nyilvános védés bizottsága:

Head of the Defence Committee / Elnök:

Prof. Dr. Rajnai Zoltán egyetemi tanár, ÓE

Secretary / Titkár:

Horváthné Dr. Drégelyi-Kiss Ágota egyetemi docens, ÓE

Participants / Tagok:

Dr. habil. Molnár András egyetemi docens, ÓE

Dr. habil. Farkas Tibor egyetemi docens, külső, NKE (external)

Dr. Tóth András adjunktus, külső, NKE (external)

Reviewers / Bírálók:

Prof. Dr. Kóczy T. László egyetemi tanár, külső, SZE (external)

Prof. Dr. Szabolcsi Róbert egyetemi tanár, ÓE

Budapest, 2017.

…………………………..

CONTENTS

INTRODUCTION

Formulation of the Studied Scientific Problem

The focus for this dissertation is on studying mechanical systems of complex dynamics. Unmanned aerial vehicles and robotic manipulators are typical examples of complex dynamics systems. The difficulty in wide spread studding robotic manipulators lies in their relative low availability and high cost. A wide area of robotics research is dedicated to aerial platforms, which have very similar dynamics and are more simple to build and also commercially available in wide ranges. Versatile flying structures and configurations have been developed to allow 3D movements [35], [60]. For example, there are blimps, fixed-wing planes, single rotor helicopters, bird-like prototypes, quadrotors, hexa-rotors, octa-rotors, etc. Each of these has advantages and drawbacks. The vertical take-off and landing (VTOL) requirements exclude some of the aforementioned configurations.

The quadrotor architecture has low dimensions, good manoeuvrability, simple mechanics and good payload capability. The main drawback is the relatively high energy consumption and difficult precision flight control; however, the trade-off results are very positive. This structure can be attractive in several applications, in particular for surveillance, for imaging dangerous environments, and for outdoor navigation and mapping. The study of kinematics and dynamics helps to understand the flight mechanics of the quadrotor and its behaviour [33], [12]. Together with system modelling, the definition of the control algorithm structure is very important. Soft computing methods can be efficiently applied together with, and even instead of conventional controllers [63].

Multi-rotors like quad- and hexa-rotors are popular representatives of VTOL unmanned aerial vehicles (UAVs) as they are relatively simple to build, while being of versatile applicability, also capable of vertical take-off and landing. Also the multi-rotor architecture has simple mechanics, high relative payload capability and good manoeuvrability. The study of multi-rotor kinematics and flight dynamics is based on the physics of aerial platforms - flying bodies, a good description of such can be found in [35]. The kinematics and general force and torque dynamics, flight mechanics of any symmetric multi-rotor (quad-, hexa- or any other number of rotors) is equivalent.

This work presents an efficient toolset improvement proposal for multi-rotor aerial vehicles control system design. Efficient autonomous navigation and obstacle avoidance requires a fast, direct method for calculating time and energy efficient feasible trajectories. Efficient control systems in real-life outdoor environment require robust adaptive system models. Designing robust fuzzy systems require efficient global and precise local optimisation techniques.

The first part of this theses collection proposes improvements of multi-objective stochastic search by a new vector comparison ordered inequality operator and ranking method. The efficiency analysis of the new method is presented on well-studied genetic algorithms of proven convergence capability and carefully designed, mathematically sound, difficult multi-criteria optimisation problems of various Pareto-front form and search space density.

The second part proposes a novel representation of fuzzy-partitions based inference systems for universal function approximations. The proposed fuzzy-partition parameter representation of non-linear parameters of these systems makes it possible to be subjected to efficient unconstrained optimisations by global search algorithms and fine-tuning with gradient descent based methods. Linear parameters of these fuzzy-systems are best calculated based on singular value decomposition to achieve mean square error minimisation. The new multi-objective stochastic search methods introduced in the first part are used to find non-dominated fuzzy system solutions where both the fuzzy structure complexity, the number of membership functions and rules, and the function approximation error, both the absolute maximum error and the mean square sum of the error is first globally minimised then locally fine-tuned by a gradient descent method.

The third part proposes a new method for robust fuzzy-system based modelling of complex dynamic systems as robot manipulators and mobile robots, like free flying multi-rotors. For the six degree of freedom multi-rotors a special extension is proposed for periodic continuous extension of fuzzy systems. This new method follows the grey box identification approach, making use of well-known system properties of robotic manipulators. My proposal results in a system approximation, which has all the benefits of robust fuzzy systems, and also manifests all the analytical properties of dynamic models that are used for analysing system and system control properties. Application of these fuzzy system based grey box models in classical hard computing control techniques is straightforward, as it is possible to explicitly analytically extract all system states and their derivatives.

The fourth part introduces a direct, iteration free single-pass algorithm using simple closed formulas, to design time and energy efficient trajectory parametrisations with pre-defined time derivative constraints. The method enables designing trajectories tuned to system (including control actuator) capabilities and ensuring oscillations free control possibilities. The method is presented for both multi-rotor trajectory designs, where higher derivative smoothness is a must for efficient control, and it is also presented for 3D overhead crane trajectory designs to analyse its oscillations free property.

The fifth part presents a new method for fuzzy-system training data set reduction.

Research Objectives

As concluded and highlighted in the introductory chapter for efficient multi-rotor autonomous navigation and obstacle avoidance improvement it is necessary to master the following design and engineering tools:

1. *Efficient multi-objective search*

My first goal is to define a new operator for comparing two vectors, which can be used as basis for an efficient multi-objective ranking method for Genetic Algorithm (GA), which performs better than the existing Pareto dominance based algorithms.

2. *Efficient genetic fuzzy system universal function approximation*

My second goal is to define a new method for an efficient unconstrained optimisation of Takagi-Sugeno-Kang (TSK) Fuzzy Logic Systems (FLS) subject to both a GA based global search and further ANFIS like gradient descent based local fine tuning of fuzzy partition antecedent Membership Function (MF) parameters. The MF rule base has to remain intact; complete fuzzy partitions have to remain with keeping the pre-defined linguistic variable order. The resulting FLS has to be capable of acting as a universal function approximation.

3. *Efficient robust modelling method for autonomous control of complex systems of nonlinear dynamics*

My third goal is to define a new efficient robust system dynamics modelling method, which results in a system model that can be readily used for efficient autonomous, system state model based control of complex nonlinear dynamics systems such as robot manipulators (RM) and multi-rotor unmanned aerial vehicles (UAV) navigation dynamics.

4. *Efficient trajectory design method for autonomous control of complex nonlinear dynamic systems*

My fourth goal is to define a new method for an efficient real-time direct path parametrisation design algorithm for generating physically feasible, time-and energy optimal, bounded, continuous trajectories that induce no system oscillations. The notion of time and energy optimality is not to be used in some mathematics theory manner but in real life physically feasible engineering manner. Finding optimal trajectories is focused on finding the appropriate parametrisation for the path vector function, given the pre-defined feasibility limits on the displacement time derivatives.

5. *Efficient genetic fuzzy system training data set reduction method*

My fifth goal is to define a new method for an efficient genetic fuzzy system (GFS) training data set reduction, which will significantly reduce the data set size, while maintaining the quality of the identification process, and thus significantly increase the identification process performance, independent of the system to be identified.

Research Hypothesis

Hypothesis I: there exists a vector comparison method which is capable of guiding a multi-objective stochastic search more efficiently than the Pareto dominance relation.

Hypothesis II: there exists a more suitable parametrisation method for antecedent fuzzy partition MF components of a TSK FLS, which is still simple to directly compute and optimise without any restrictions, both by stochastic search and/or with gradient descent methods; and for every case the formed parameters will inherently satisfy all of the required constraints for a stable fuzzy partition antecedent structure, keeping the associated linguistic values.

Hypothesis III.a: the singular value decomposition (SVD) algorithm is efficient enough **to extract each basic component of a dynamic system described by Euler-Lagrange** equation, when there are a sufficient number of good quality training samples available. Further on the nonlinear inertia component functions can be robustly identified with TSK FLSs, while the nonlinear functions describing the centrifugal and Coriolis effects can be exactly derived from the identified TSK FLSs for the inertia components. The nonlinear parameters of TSK FLSs modelling a Robotic Manipulator (RM) dynamic system can be efficiently found by a multi-objective hybrid GA together with gradient descent method fine-tuning, while all the linear parameters of the used TSK FLSs, including constants of the model can be directly calculated with SVD based robust least squares (LS) method. The RM trajectory used for collecting training samples have to be sufficiently exciting to reveal all the characteristics of the RM system equation.

Hypothesis III.b: it is possible to extend the TSK FLSs in a way that they become periodic and of continuous output, even for the 0-2π transitions of attitude Euler angle system inputs. Then for modelling multi-rotor flight dynamics each nonlinear component of a flight dynamics formulated by Euler Lagrange approach can be identified in a similar manner as stated in my previously described hypothesis III.a for RMs.

Hypothesis IV: system trajectories can be designed in harmony with the system dynamics and its actuator characteristics. Such trajectories are energy efficient as no oscillations are induced, and they are feasible, time optimal in terms that no trajectory exists with faster transients, such that the system can precisely track it with lesser energy consumption. These harmonic trajectories are continuous up to the required number of time derivatives, and they can be made bounded in their any number of time derivatives. For a realistic, feasible control input of multi-rotor UAV the designed path has to be such that the sixth time derivative of the body displacement function must be continuous and its fourth time derivative transient has to be feasible for the control actuator. For a realistic, feasible control input of direct brushless DC electric motor (BLDC) actuated systems (RMs, cranes, wheeled vehicles) the designed path has to be such that the fourth time derivative of the planned displacement must be continuous, while the planned body rotation must be such that the feasible body torque transients are proportional to the possible motor torque transients; equivalently the feasible second derivative of the body displacement has to be proportional to motor shaft feasible angular velocity.

Hypothesis V: for dynamic system GFS identifications the necessary training data set of collected samples along real trajectories can be reduced without significant loss in the quality of the identification result, while significantly improving the efficiency of the identification process.

Research Tools and Thesis Validation Methods

From the first introductory chapter it is obvious that my research is multidisciplinary, as such various research and test methods are necessary to test my hypothesis. As my goals are more general than finding a single specific method which is only applicable to UAV design, but are applicable to wide range of multidisciplinary field, I and not using a single UAV example to validate my theses. For each hypothesis I am also using a method well matched to the nature and specifics of the problem, so that my results are appropriately tested and presented in a general way.

1. *Validating Quality of Multi-objective Search and Optimisation*

The proposal is to be validated on well-studied, mathematically sound GA hard multi-objective benchmark problems like:

a) Simple Two Objective Optimisation Problem

b) Deceptive Multi-objective Optimisation Problem

c) Multi-modal Multi-objective Problem

d) Convex and Non-convex Pareto-optimal Fronts

e) Discontinuous Pareto-optimal Front

f) Biased Search Space

g) Generalisation of Two Objectives to Four, Eight and Sixteen Objectives

2. Validating Quality of Genetic Fuzzy System Function Approximation

The proposal is to be validated on well-studied, mathematically sound, versatile, difficult benchmark identification problems of high complexity like:

a) Predicting Future Values of Chaotic Time Series of Mackey and Glass

b) Identification of Gas Furnace Model of Box and Jenkins

c) Identification of Generalised Rastrigin Function

3. Validating Quality of Complex Nonlinear Dynamics System GFS Modelling

The proposal is to be validated on well-studied robot manipulator dynamics modelling and quadrotor flight dynamics modelling simulations.

4. Validating Quality of System Trajectory

The proposal is to be validated on well-studied 3D crane and quadrotor flight trajectory design.

5. Validating Quality of Genetic Fuzzy System Training Data Sets

The proposal is to be validated on well-studied quadrotor flight dynamics modelling simulations.

1 QUADROTOR UNMANNED AERIAL VEHICLES

This chapter describes the system in focus of this research – the multirotor unmanned aerial vehicle and the existing major tools used to achieve system control suitable for its autonomous navigation and obstacle avoidance.

As well summarised in [60] [63] rotary wing aerial vehicles have distinct advantages over conventional fixed wing aircrafts in surveillance and inspection tasks because they can take-off and land in bounded spaces and easily fly above the target. A quadrotor is a four-rotor copter. An example of one is shown in Figure 1 [60].

Multirotor copters are theoretically dynamically stable, but of poor stability augmentation and of limited control authority, thus very difficult to control; suitable advanced robust control methods are needed to make them steadily manoeuvrable. Although a very sensitive dynamics is not desirable from control stability point of view, it is good from the system agility, manoeuvrability point of view. The system complexity comes from wide range of changes in the rotorcraft parameters and from unpredictable real-life environmental disturbances such as a wind gusts or air density variations.

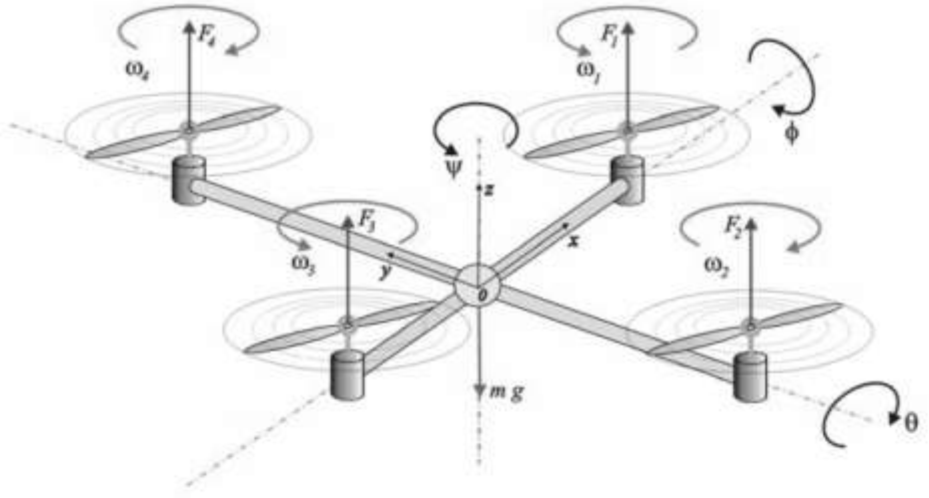


Figure 1.
3 D motion, commonly used model of the quadrotor [60].

1.1 Modelling Multi-rotor Flight Dynamics

A quadrotor is controlled only by varying rotor speeds, thereby changing the lift forces and rotor torques [33], [12]. It is an under-actuated dynamic vehicle with four input forces and six outputs coordinates. One of the advantages of using multi-rotor copters is the increased payload capacity. Quadrotors are highly manoeuvrable, which allows for vertical take-off and landing, as well as flying into hard-to-reach areas. Disadvantages are the increased rotorcraft weight and increased energy consumption due to extra motors. Since the machine is controlled via rotor speed changes, it is more suitable to utilize electric motors. Large helicopter engines, which have a slow response, may not be integrated to a multi-rotor system satisfactory without incorporating a proper gear-box system.

Unlike typical helicopter models and regular helicopters, which have variable pitch angles, a quadrotor has fixed pitch angle rotor blades, and only the speed of individual rotors is controlled in a suitable manner in order to produce the desired lift force, rotation torques and vehicle position displacements.

The quadrotor is satisfactory well modelled with four rotors in a cross configuration as presented in Figure 1. This cross structure is usually quite thin and light, however it has to show robustness by linking mechanically the motors, which are heavier than the cross structure itself. Depending on the mechanical characteristics of the used electromotor each propeller can be connected to the motor through a reduction gear. Each propeller blade axis of rotation is fixed and they are parallel to each other. Furthermore, they have fixed-pitch blades and their airflows point downwards to get an upward lift. These considerations point out that the structure is expected to be quite rigid and the only characteristics that can dynamically vary are propeller blade rotation speeds.

As shown in Figure 1 [60], one pair of opposite propeller blades of a quadrotor rotates clockwise (rotors 2 and 4), whereas the other pair of blades rotates counter clockwise (rotors 1 and 3). This way it is possible to avoid the yaw drift due to reactive torques. This configuration also offers the advantage of enabling lateral displacement motions without changing the pitch of the propeller blades. Having fixed pitch rotors significantly simplifies rotor mechanics and reduces gyroscopic effects. Movement direction control of quadrotors is achieved by commanding different speeds to different propellers, which in turn produce differential aerodynamic forces, torques and moments. For hovering, all four propellers rotate at the same speed. For vertical motion, the speed of all four propellers is increased or decreased by the same amount, simultaneously. In order to pitch and move laterally in a desired direction, speed of propellers 3 and 1 is changed conversely. Similarly, for roll and corresponding lateral motion, speed of propellers 2 and 4 is changed conversely. To produce yaw, the speed of one pair of two oppositely placed propellers is increased while the speed of the other pair is decreased by the same amount. This way, the overall produced thrust is the same, but the differential drag moment creates a yawing motion. Since having only four actuators, the quadrotor is still an under-actuated six degree of freedom (6 DOF) system.

To describe the motion of a 6 DOF rigid body it is usual to define two reference frames: the earth inertia frame (E-frame), and the body-fixed frame (B-frame) – see Figure 2 [63]. Equations of motion are more conveniently formulated in the B-frame because the inertia matrix is time-invariant, advantage of body symmetry can be taken to simplify equations, also measurements taken on-board are easily converted to B-frame and control forces are readily available in the B-frame.

The E-frame (*OXYZ*) is chosen as the right-hand reference inertia system. Axis *Y* points toward the North, *X* points toward the East, *Z* points upwards with respect to the Earth, and *O* is the axis origin. This frame is used to define the linear position (in meters) and the angular position (in radians) of the quadrotor. The B-frame (*oxyz*) is attached to the multirotor body centre of mass. Axis *x* points toward the quadrotor front, *y* points toward the quadrotor left, *z* points upwards and *o* is the axis origin. The origin *o* is chosen to coincide with the centre of mass of the quadrotor cross structure. This reference is right-hand, too.

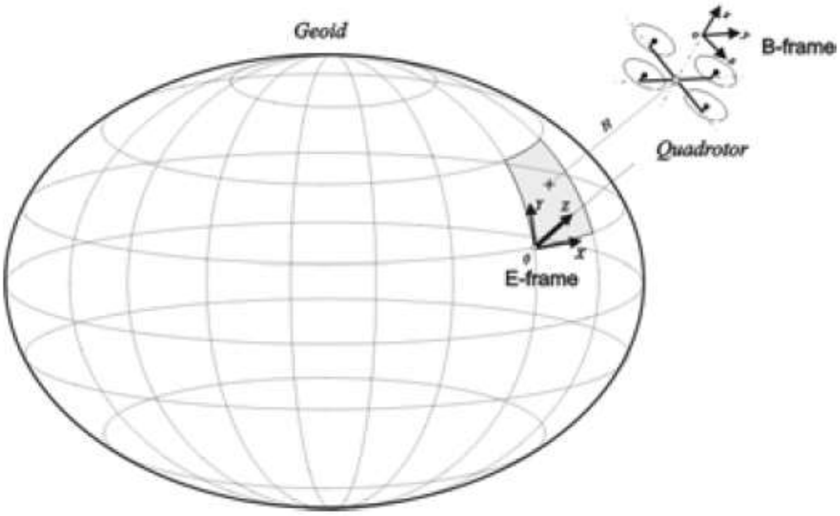


Figure 2.
Earth- and Body-frames used for modelling of the quadrotor system [63].

The body linear velocity v (m/s), the angular velocity of the body Ω (rad/s), the forces F (N) and the torques T (Nm) acting on the body are defined in this frame. The linear position of the copter (X, Y, Z) is determined by coordinates of the vector between the origin of the B-frame and the origin of the E-frame according to the rotation matrix in equation (2). The angular position or attitude of the copter (ϕ, Θ, ψ) is defined by the orientation of the B-frame with respect to the E-frame. This is given by three consecutive rotations about the main axes which take the E-frame into the B-frame. The "roll-pitch-yaw" set of Euler angles can be used. The vector that describes the quadrotor position and orientation with respect to the E-frame can be written in the form:

$$\boldsymbol{q} = [X \quad Y \quad Z \quad \phi \quad \Theta \quad \psi]^T, \tag{1}$$

where: $\boldsymbol{q}$ is the copter system position and orientation vector; X, Y, Z are the E-frame translational coordinates; ϕ, Θ, ψ are the "roll-pitch-yaw" set of Euler angles describing the E-frame orientation.

The rotation matrix between the E- and B-frames we conveniently chose to have the following form [8]:

$$\boldsymbol{R} = \begin{bmatrix} c_\Psi c_\Theta & -s_\Psi c_\Phi + c_\Psi s_\Theta s_\Phi & s_\Psi s_\Phi + c_\Psi s_\Theta c_\Phi \\ s_\Psi c_\Theta & -c_\Psi c_\Phi + s_\Psi s_\Theta s_\Phi & -c_\Psi s_\Phi + s_\Psi s_\Theta c_\Phi \\ -s_\Theta & c_\Theta s_\Phi & c_\Theta c_\Phi \end{bmatrix}, \tag{2}$$

where: $\boldsymbol{R}$ is the rotation matrix between the E- and B-frames; s and c are abbreviations for the sinus and cosine functions of Euler angles as: $s_* = \sin(*)$, $c_* = \cos(*)$.

The corresponding transfer matrix has the form:

$$\boldsymbol{T} = \begin{bmatrix} 1 & s_\Phi t_\Theta & c_\Phi t_\Theta \\ 0 & c_\Phi & -s_\Phi \\ 0 & s_\Phi / c_\Theta & c_\Phi / c_\Theta \end{bmatrix}, \tag{3}$$

where: $\boldsymbol{T}$ is the transfer matrix for Euler angle rates between E-frame and B-frame. As in the previous case a notation has been adopted: s_* =sin(*), c_* =cos(*), t_* =tan(*).

The system Jacobian matrix, taking (2) and (3), can be written in the form:

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{T} \end{bmatrix}, \tag{4}$$

where: $\boldsymbol{0}_{3x3}$ is a 3 by 3 zero-matrix.

The generalized quadrotor velocity in the B-frame has a form of:

$$\boldsymbol{v} = [\dot{x} \quad \dot{y} \quad \dot{z} \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\psi}], \tag{5}$$

where: $\boldsymbol{v}$ is the generalized quadrotor velocity in the B-frame; $\dot{x}$, $\dot{y}$, $\dot{z}$ are the B-frame velocities along the appropriate coordinate axis; $\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$ are the "roll-pitch-yaw" rates of Euler angles in the B-frame.

Finally, as the result of this nomenclature the kinematical model of the quadrotor can be defined in the following simple vector equation form:

$$\dot{\boldsymbol{q}} = \boldsymbol{J} \cdot \boldsymbol{v} \tag{6}$$

The dynamics of a generic 6 DOF rigid-body system takes into account the mass of the body *m* (kg) and its inertia matrix $\boldsymbol{M_B}$ (kgm^2). Two assumptions are commonly made:

• The first assumption states that the origin of the body-fixed frame is coincident with the centre of mass of the body. Otherwise, another point should be taken into account, which could make the body equations considerably more complicated without significantly improving the model accuracy.

• The second common simplification assumption specifies that the axes of the B-frame coincide with the body principal axes of inertia. In this case the inertia matrix $\boldsymbol{M_B}$ is diagonal and, once again, the body equations become simpler.

The dynamic model of a quadrotor can be defined in the following matrix form:

$$\boldsymbol{M_B}\dot{\boldsymbol{v}} + \boldsymbol{C_B}(\boldsymbol{v})\boldsymbol{v} - \boldsymbol{G_B} = \boldsymbol{\lambda}, \tag{7}$$

where: $\boldsymbol{M_B}$ is the system inertia matrix; $\boldsymbol{C_B}$ represents the matrix of Coriolis and centrifugal forces; $\mathbf{G_B}$ is the gravity matrix; $\boldsymbol{\lambda}$ is the generalized force vector including forces and torques along body axes. These matrices have known forms as presented in [8].

A generalized force vector $\boldsymbol{\lambda}$ has the form:

$$\boldsymbol{\lambda} = \boldsymbol{O_B}(\boldsymbol{v})\boldsymbol{\Omega} + \boldsymbol{E_B}\boldsymbol{\Omega}^{\mathbf{2}} \tag{8}$$

where: $\boldsymbol{O_B}$ is the gyroscopic propeller matrix; $\boldsymbol{E_B}$ is the movement aerodynamic matrix; $\boldsymbol{\Omega}$ is the propellers' speed vector as defined below in equation (12).

$\boldsymbol{O_B}$**,** the gyroscopic propeller matrix is:

$$\boldsymbol{O_B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \dot{\theta} & -\dot{\theta} & \dot{\theta} & -\dot{\theta} \\ -\dot{\phi} & \dot{\phi} & -\dot{\phi} & \dot{\phi} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

$\boldsymbol{E_B}$**,** the movement aerodynamic matrix has the form:

$$\boldsymbol{E_B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b & b & b & b \\ 0 & -b \cdot l & 0 & b \cdot l \\ -b \cdot l & 0 & b \cdot l & 0 \\ -d & d & -d & d \end{bmatrix}, \tag{10}$$

where b (Ns^2) and d (Nms^2) are the thrust and the drag factors of one propeller blade; l (m) is the distance between the quadrotor centre of mass and the propeller blade centre of mass.

Equation (11) defines the overall propeller blades rotation speed (rad/s) used in equation (8).

$$\omega = -\omega_1 + \omega_2 - \omega_3 + \omega_4, \tag{11}$$

where: ω is the overall propellers' speed; ω_i is the angular speed of the ith propeller. Positive sign is taken for clockwise rotations and negative sign for counter clockwise rotations.

$\boldsymbol{\Omega}$, the speed vector of propeller blades is defined as:

$$\boldsymbol{\Omega} = [\omega_1 \quad \omega_2 \quad \omega_3 \quad \omega_4]^T \tag{12}$$

Equations (1) to (12) take into account the entire quadrotor nonlinear flight mechanics model including the most influential aerodynamic effects of rotor blade trust and drag.

For high speed airplanes, it is common to introduce a new $\hat{X}$ stability axis, which is aligned into the direction of the oncoming air in steady flight. The stability axis is projected into the plane made by the X and Z body axes when there is sideslip. The used model of quadrotors is not taking into account the sideslip, thus the notion of the stability axis is not used.

High speed aerial platforms in open door environments are highly nonlinear systems also subject to many further nonlinear perturbations like:

i. drag like effects:

   a. blade flapping,

   b. induced drag,

   c. translational drag,

   d. profile drag and

   e. parasitic drag,

ii. ground effect,

iii. in vertical descent further nonlinear effects have to be accounted for as:

   a. vortex ring state,

   b. turbulent wake state,

   c. windmill brake state as described in [68].

None of these nonlinear effects are commonly considered when modelling multirotor flight dynamics. One feasible approach is to account for these effects as disturbances, which have to be compensated by the controller. In my research I am going one step further, by accounting for all unknown disturbances already in the system model in form of using robust fuzzy systems to initially reduce the effects of all input disturbances.

Precision, robustness and adaptability of the applied dynamic model are the starting point to achieve a precise and efficient autonomous control of the system [1]. Fuzzy systems are capable of robust modelling and control of complex systems including helicopters [71].

1.2 Unmanned Aerial Vehicles Path Tracking Control Solutions

The nonlinear, multivariable and coupled system state and response characteristics make the quadrotor difficult to control. In a general approach two loops are used for the quadrotor UAV control system: the outer loop is the position controller and the inner loop is the attitude controller. The controller of the outer loop for position includes information such as instantaneous position and speed. While the inner loop for attitude controller includes the orientation attitude posture information.

The position controller receives as inputs the difference between the desired E-frame position (X,Y,Z) and the actual current position; the outputs are the required B-frame attitude rotation angles (φ,θ,ψ) to move towards the desired position. The required attitude angles are provided for the attitude controller in the inner loop. The attitude controller outputs the desired propeller blade rotation speed value to each of the four motor controllers, which adjusts the rotation speed of the corresponding motor.

By studding the available literature one can observe that motor controllers are generally taken for granted as mere mathematical tools capable of instantaneously generating any desired rotation speed value, their actual physical dynamics is considered neither for multirotor control, nor for multirotor trajectory design. One of my major research targets is to remedy this deficiency.

Control mechanisms as proportional-integral-derivative (PID) controller, back stepping (computed torque) controller and their fuzzy variants have been also well studied and successfully used for decades on many systems including multi-rotors [s1].

1.2.1 PID Controllers with Fuzzy Systems Based Adaptive Gain Parameters

To achieve flight stabilisation and path tracking of quadrotor UAVs, as described in [66] and [34] fuzzy logic systems can be used for adaptive tuning of PID controllers. A PID controller consists of a proportional, an integral and a derivative feedback control action, which represents the current, past and anticipated future errors that cover all the time history of the error signal. The relevant PID gains are named K_P, K_I and K_D. By adjusting these parameters, the desired performance and stability of the system can be achieved. The mathematical representation of PID controller is given as:

$$\boldsymbol{u}(t) = K_P\boldsymbol{e}(t) + K_I\int_0^t \boldsymbol{e}(t)dt + K_D\frac{d}{dt}\boldsymbol{e}(t), \tag{13}$$

where: $\boldsymbol{u}$ is the control signal; t is the time; $\boldsymbol{e}$ is the error signal; K_P, K_I and K_D are the PID gain parameters.

PID controllers with fixed gain parameters provide good performance only for linearized systems, or in a single selected narrow quasilinear operation range for nonlinear systems. To overcome the system nonlinearity control problem fuzzy logic systems can be used for adaptive tuning of PID gain parameters [72], [78]. Fuzzy logic consists of four components: fuzzification, fuzzy rule base, inference engine, and defuzzification:

• Fuzzification refers to the process of transforming crisp input values into grades of membership using linguistic terms of fuzzy sets.

• Fuzzy rule base is the main part of fuzzy logic systems. This component is based on if-then rules. The fuzzy rule base defines how to react to each input combination.

• Inference engine applies the fuzzy rule base to form the output for defuzzification.

• Defuzzification is a method to obtain numerical data from the output of a fuzzy rule base.

For a self-tuning fuzzy PID controller the tracking error can be considered as an input for two controllers: first time for a classical PID control algorithm to minimize the position error, and second time to a fuzzy logic system for adjusting the K_P, K_I, and K_D gain parameters of the PID controller equation (13).

In [66] three fixed position and fixed size triangular membership functions (MFs) were used for fuzzification of the error signal as input and for the gain parameter output. A fixed set of three fuzzy rules were defined. Results of [66] show that the performance of both the classical PID and the fuzzy logic based self-tuning PID control method is acceptable in static cases of no load variation. The self-tuning PID control based on fuzzy logic is able to compensate for variations in payload and to achieve good path tracking. The classical PID control algorithm by itself cannot track the circular path, while the self-tuning PID based on fuzzy logic provides a good performance solution also for this problem [66].

In [34] a wide range of disturbances are modelled. Gravity and buoyancy are combined into a single force. The fluid inertia force related to the acceleration of the rotorcraft motion is represented by additional mass. In the B-frame the direction of the lift force remains constant. While in the inertial system E-frame the driving force of lift is decomposed into three directions along the three coordinate axes. The quadrotor craft is more susceptible to influences of drag, air resistance during flight because of its large surface area. In [34] six fuzzy adaptive controllers are proposed for tuning the gain parameters of PID controllers, having one control output for each state variable of equation (1). For each fuzzy adaptive controller in [34] seven fixed MFs (of linear Z and linear S and of triangular types) are defined for input fuzzification and output defuzzification. The fuzzy rule base is defined constant by expert knowledge. Results in [34] show that for a quadrotor airship model the fuzzy adaptive PID algorithm has a better performance than the classical PID control system.

1.2.2 Lyapunov Stable Back-stepping Control with an Adaptive Fuzzy Model

In [15] it is presented how a robust control method with an adaptive fuzzy model can overcome wind disturbance, which buffet the vehicle with periodic wind vortices. For adaptive-fuzzy altitude control of the roll, pitch, and yaw the nonlinear functions of equation (6) are modelled by fuzzy systems. The notation used is as follows – from equation (8), where the generalized force is represented as a vector of control inputs to each of the quadrotor motors: $\boldsymbol{\lambda} = [u_1, u_2, u_3, u_4]$. From equation (1) a new state vector $\boldsymbol{x}$ is derived as $\boldsymbol{x} = [\boldsymbol{q}, \dot{\boldsymbol{q}}]$ so that the system equation (6) becomes of form $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ as:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) = \begin{pmatrix} x_{10} \\ x_{11} \\ x_{12} \\ x_{10}x_{12}a_1 + x_{10}a_2\omega + b_1u_2 \\ x_5x_{12}a_3 + x_5a_4\omega + b_2u_3 \\ x_{10}x_5a_5 + b_3u_4 \\ x_9 \\ x_7 \\ x_8 \\ -g + c_{x_1}c_{x_3}u_1/m \\ g(x)u_1/m \\ h(y)u_1/m \end{pmatrix}, \tag{14}$$

where *m* is the body mass; system parameters a_i and b_i are defined as:

$$a_1 = \frac{I_Y - I_Z}{I_X}, a_2 = -\frac{J_r}{I_X}, a_3 = \frac{I_Z - I_X}{I_Y}, a_4 = \frac{J_r}{I_Y}, a_5 = \frac{I_X - I_Y}{I_Z}, b_1 = \frac{l}{I_X}, b_2 = \frac{l}{I_Y}, b_3 = \frac{l}{I_Z},$$
$$g(x) = \left(c_{x_4}s_{x_5}c_{x_6} + s_{x_4}s_{x_6}\right),\ h(y) = \left(c_{x_4}s_{x_5}s_{x_6} - s_{x_4}c_{x_6}\right),$$

where $I_{X,Y,Z}$ are body inertia terms; J_r is the rotor blade inertia term; l is the lever of the motor, which is the rotor blade axis distance from the craft centre of mass; and *m* is the mass of the system.

In [15] the fuzzification encoding of input $\boldsymbol{x}$ consists of normalized Gaussian MFs with centres placed on a fixed, evenly spaced lattices of widths σ like $\Gamma_i(\boldsymbol{x}) = \frac{\exp(-(\boldsymbol{C_i}-x)^T(\boldsymbol{C_i}-x)/\sigma\text{^}2)}{\sum_{i=1}^{N}\exp(-(\boldsymbol{C_i}-x)^T(\boldsymbol{C_i}-x)/\sigma\text{^}2)}$, where each $\boldsymbol{c_i}$ is a centre on the lattice and N is the number of MFs. The decoding of control outputs is accomplished with one dimensional Gaussian MFs of width σ, but with centres that can be changed adaptively. The output of the decoding is $\boldsymbol{\Gamma}^T\boldsymbol{c} = [\Gamma_1\Gamma_2 \ldots \Gamma_N] \cdot [c_1c_2 \ldots c_N]^T$ where $\boldsymbol{c} \in \mathbb{R}^N$ is the vector of output MF centres.

According to the standard approximation theory, if the density of the encoding lattice points is high enough then the output of the decoding scheme can uniformly approximate nonlinear functions in a local region as $\dot{x_k} = \Gamma_i^T(\boldsymbol{x})\boldsymbol{c}_i + \epsilon_i(\boldsymbol{x})$ where k = 4, 5, 6 for i = 1, 2, 3. Thus $\epsilon_i(\boldsymbol{x}) \leq \epsilon_{max}$ the approximation error of nonlinear functions is bounded. If the actual centres of output MFs are $\hat{\boldsymbol{c}}$ the error between actual and ideal centres is $\tilde{\boldsymbol{c}} = \boldsymbol{c} - \hat{\boldsymbol{c}}$.

An obvious conclusion is that the more MFs and corresponding rules we have, the more precise our fuzzy system function approximation will be. For real time control applicability on the other hand we need as small amount of computations as possible. This is the inherent conflict of fuzzy control system identification objectives.

My research actively targets finding appropriate Pareto-optimal solutions to these multi-objective optimisation problems.

Further on by taking filtered tracking errors $z_i = Lx_i + x_{(i+3)}$ for i = 1, 2, 3 where L is a positive constant, a Lyapunov control function can be defined as $V(\boldsymbol{z}, \tilde{\boldsymbol{c}}) =$

$\sum_{i=1}^{3}\left(z_i^2 + \tilde{\boldsymbol{c}}_i^T \tilde{\boldsymbol{c}}_i\right)/2$. Assuming there is a bounded external disturbance, then the derivative of V is bounded and Lyapunov stability of the proposed control method is presented in [15]. The resulting control is stable, computationally efficient, and theoretically robust to disturbance. It achieves high performance while eliminating centre drift [15].

1.2.3 Direct PD Like Fuzzy Controllers for Position and Altitude Control

In [56] and [55] a direct fuzzy proportional-derivative (PD) controller approach is used for altitude control and path tracking of a quadrotor UAV. To control each three angular position error a normalized input/output Sugeno type fuzzy engine was used with three MFs on both input and output and a fixed rule base defined by expert knowledge. The fuzzy attitude control design proposed in [56] was verified within simulations by comparison to a back-stepping approach control design. Path tracking efficiencies are very similar for both attitude control systems. In [56] the proposed fuzzy attitude control revealed slightly better performance in case of a rapid trajectory direction changes. The only significant difference was in the first segment of the flight, where the proposed fuzzy solution obtains the desired trajectory much faster than the back-stepping solution [56].

In [50] a PD like fuzzy controller is described for position control to compensate for nonlinear disturbance such as the wind. For the position controller of the flying robot, there are two inputs to the fuzzy logic controller. The first is e the position error, which is the difference between the target position of the robot and its actual position. The second input is $\dot{e}$ the first derivative of the position error with respect to time. For such a setup the fuzzy controller output is the generalized force necessary for moving the body to the target position. For the output of fuzzy controller in [50] a simplified fuzzy inference method is used with fixed MFs and an expert knowledge based pre-defined constant fuzzy rule set. The result in [50] is that the proposed fuzzy controller is more suitable for path tracking in outdoor conditions than a simple PD controller. The proposed fuzzy controller still presented not very good results for the response and steady-state error when a fair wind was blowing, as a fixed MF and constant rule base fuzzy system cannot actively adapt to environmental changes.

In my research I am using mean square error optimal real time adaptable fuzzy rule parameters and also the MF parameters can be fine-tuned along gradient descent in an outer, not real time update cycle.

1.2.4 Fuzzy Control System Based Visual Servoing

Computer vision techniques can provide UAVs with an additional source of information to perform visually guided tasks like tracking and visual servoing, inspection, pursuit and flying in formations. [51] presents a fuzzy servoing strategy using a real time flying object tracking method based on only visual information to generate commands in a dynamic “look and move” control architecture. Considering a flying object moving with an unknown trajectory in the world space and a flying robot with an attached fixed, calibrated, pinhole camera, both having idealized flying dynamics. The control goal is to command the flying robot in order to track the target object by keeping it always in the camera focus with a fixed separation distance.

The target is modelled as an ideal spherical surface, the projection point can be considered as the image projection of the target’s sphere centroid with coordinates in the camera frame. The projected diameter can be used to estimate the distance to the target, because it is inversely proportional to the distance from the camera. In [51] the problem of tracking is approached by exploiting the colour characteristic of the target. A basic colour is defined to the target by assuming a simple coloured mark to it and tracking this mark.

This process is not always perfect, and changes still occur in colour distributions over time. An algorithm that has proven to deal with this issue by dynamically adapting to changes in probability distributions is the Continuously Adaptive Mean Shift [7]. This algorithm is based in the mean shift originally introduced by Fukunaga and Hostetler [25].

In [51] two Mamdani fuzzy controllers were used, which are based on visual information (previously described) to generate yaw and pitch commands for the UAV. All variables of these two controllers are defined using triangular membership functions. The complete controller design is fixed, based on expert knowledge. Both controllers have two inputs and one output. The controller of the yaw or heading of the UAV has for the first input the angle estimation in radians, between the UAV (the centre of the image) and the centre of the object to follow. The second input is the difference between the last angle estimation and the actual angle. This controller sends velocity commands (degrees per seconds) for change of the heading position of the aircraft. The second controller acts on the pitch state of the UAV. It takes the data about the size of the object in pixels, to follow and to estimate the distance from the target. Using for the first input the actual size of the object and for the second input the difference between the last size measure and the actual size is taken. The output of the controller is velocity commands to go ahead, in the case that the object is far away; stay in the same position if the object is near at a predefined safe distance, or go back if it is very close to the UAV.

Real tests on outdoors scenarios demonstrated excellent behaviour of fuzzy controllers, which were generating yaw and pitch commands based on visual information, performing the action of tracking the target object from a safe distance [51].

In [52] the same setup was used as in [51]. A Fuzzy Logic controller based on expert knowledge has been developed to automatize the collision avoidance. This controller acts changing the heading of the aircraft, keeping the obstacle to avoid at the right side (or left) of the image until the object can be overtaken. Excellent results have been obtained in real tests using a commercial quadrotor with a quick response and low error estimation [52].

These visual servoing methods, and common global positioning system (GPS) navigations, graph based map tracking global root selections are proven good strategies to define the next flight coordinates in a point-to-point control strategy. Research also exists on optimally connecting certain pre-defined waypoints with higher ordered polynomials or splines, but there are no solutions for completely autonomous real-time applicable trajectory planning for arbitrarily waypoints, when desired velocity, acceleration and higher order displacement derivatives are prescribed and bounded.

My research also deals with point-to-point trajectory planning that fulfils these complex requirements, along with appreciating system capabilities so that the resulting trajectory induces no system state oscillations and it is time and energy optimal in a feasible manner. Feasible trajectory optimality is such that it does not exist only mathematically on the design board, but it is directly applicable to the system, which is capable of exactly tracking the prescribed trajectory along the complete time signal, and there exist no other feasible trajectory, which could be completed in a shorter time manner at lower energy costs.

1.2.5 Fuzzy Controller Performance

In [63] benchmarking and qualitative evaluation of different autonomous quadrotor flight controllers is presented. Three characteristic representatives of frequently used flight control techniques are considered: PID, back-stepping and fuzzy. Dynamic performances, trajectory tracking precision, energy efficiency and control robustness upon stochastic internal and/or external perturbation was considered.

Two experimental scenarios were considered as characteristic benchmarking procedures: dynamic quadrotor flight in a 3D-loop manoeuvre and a typical cruising flight along the trajectory introduced by setting waypoints with the pre-defined GPS coordinates. In case of fuzzy control six Takagi-Sugeno-Kang fuzzy systems of fixed, constant parameters were used, one FLS for each state variable of equation (1). Each fuzzy system has two inputs: error and error rate; fuzzification with three fixed triangular membership functions; and three singletons were used for output MFs. A fixed rule base is defined based on expert knowledge.

Analysing the simulation results in conclusion that the back-stepping method ensures the best control performances, in the sense of trajectory tracking precision. The other two concurrent algorithms have slightly better characteristics, in the sense of energy efficiency (having lower energy consumption). By increasing of flight speed dynamic effects become influential upon the system performances; the back-stepping method is more sensitive to changing of flight speed than other two controllers, the PID and the fuzzy logic controllers [63].

My research goals are set by realising that fuzzy system robustness is required for disturbance rejection and precise modelling and classical (back-stepping) control along with feasible optimal trajectory planning is required for precision and energy efficiency of navigation.

1.2.6 Adaptive Fuzzy Back-stepping Control

Research [77] presents an adaptive fuzzy control strategy to solve the problem of trajectory tracking for quadrotor unmanned aerial vehicle in the presence of model parameter uncertainties and external disturbances. A fuzzy system is employed to approximate directly a model based control law developed using back-stepping techniques. The adaptive laws for tuning the adjustable parameters of the fuzzy system are derived based on the Lyapunov theorem. The stability analysis of the designed adaptive fuzzy back-stepping controller is shown by the Lyapunov theory. The proposed controller yields asymptotic tracking, robustness in the presence of external disturbances affecting the six degrees of freedom, and parameters uncertainties. It is proved that all signals in the closed-loop system are semi globally uniformly ultimately bounded, and the tracking error converge to a small neighbourhood of the origin. Numerical simulation results are provided to illustrate the good tracking performances of the proposed adaptive control approach.

1.3 Conclusions of Analysing Literature on Soft-computing Autonomous Multirotor Navigation with Obstacle Avoidance

Fuzzy systems are capable of robust modelling and control of complex systems like multi-rotors. Designing and fine tuning of fuzzy systems is a complex challenge.

Gradient descent optimisations are capable of finding only the nearest extremum point, often a sub-optimal solution.

Optimisation methods based on thorough search are excessively computation expensive. Various **stochastic search** optimisation methods based on stochastic gradient descent variations like simulated annealing, taboo search; swarm intelligence methods like ant colonies, bacterial search; and evolutionary algorithms like genetic algorithms have been successfully used for fuzzy system optimisation [32], [16], [2], [9]. **Genetic algorithms** are efficient, well studied simple stochastic optimisation methods of proven convergence, capable of **global multi-objective search** and optimisation of versatile complex systems.

Mathematical model design of complex real systems can readily take the so-called black-box common approach, which uses exclusively numerical system input-output data pairs for constructing the model. Without deeper understanding of the problem, these black box models can easily end up being clumsy and working only in some specific setups, without any guaranties for general precision or robustness.

In contrast to black-box there is white box (also called glass box or clear box) modelling, which uses extensive, state of the art physics and mathematics analysis, presuming to know all necessary information; still just to end up with only simplified models, as real complex nonlinear systems can in the end be only approximated.

Grey-box modelling builds on both input-output data and also on essential expert knowledge; it efficiently incorporates them into the model structure used for system identification. Fuzzy logic system (FLS) modelling can be conducted as black-box modelling where all the system knowledge is mere input-output data, however when **expert knowledge is readily available**, we should take advantage of it – fuzzy grey-box modelling is a rational choice.

Multi-input single-output complete first order Takagi-Sugeno-Kang type FLSs are having large number of interdependent nonlinear parameters, whose number is proportional to the number of antecedent membership functions. The number of FLS linear parameters is even larger, proportional to the product of the number of membership functions over each input. **Singular value decomposition** (SVD) based least squares optimization can determine the optimal value of these linear parameters [s10], [s11].

To overcome the problem complexity of finding good values for the nonlinear parameters of a flexible structured FLS, **global search and optimization methods** as genetic algorithm (GA) can be used [88]. Genetic fuzzy system (GFS) optimisation problems like system identification inherently require **multi-objective approach** as not just the maximum absolute error and the mean square error of the identification has to be simultaneously minimal, but also the system complexity as the number of membership functions and number of rules should also be minimized for computation efficiency [s4]. Based on the standard approximation theory these objectives are clearly competing, thus the required multi-objective optimisation problem is of high complexity.

Angular orientations and induced torques of flying body systems are naturally continuous and periodic. It is our $[0, 2\pi)$ orientation representation that results in a discontinuity at full turn when returning to the origin. A proper dynamic model, be it fuzzy system based or not, must not have a jump in the output when the input continuously changes between any two posture orientation attitude angles. One possible solution is to transform the intuitive 3D Euler angles to quaternions, and perform the entire maths in this transformed space. Quaternion solutions may be called elegant, by whoever likes them, but are surely not simple and intuitive. For a proper intuitive soft computing approach to flying body modelling new efficient tools have to be designed [s13].

From **autonomous multi-rotors** it is expected to **precisely track the desired path** and **to avoid obstacles** [75]. For quadrotor flight efficiency **minimizing the energy consumption** of the system is more beneficial than planning for a minimum time or a minimum distance trajectory; a proposal for designing minimum fuel trajectories is elaborated in [14].

For quad rotors off the shelf, camera based products exist for implementing simple visual point-to-point waypoint tracking, where **the biggest challenge is agility of the quad rotor and precision of path tracking** [29].

Minimum-snap polynomial trajectories are proven very effective as quadrotor trajectories, “since the motor commands and attitude accelerations of the vehicle are proportional to the snap, or forth derivative, of the path” – citation from [37]. The rotor blade velocity is considered as an arbitrary control input. As 7th order minimum-snap polynomial trajectories are discontinuous in displacement crackle, the fifth time derivative of displacement, my claim is that this is still a sub-optimal approach. My research considers the **reality of the rotor blade** velocity not being an arbitrary theoretical control signal, but a real, electro-mechanical physical system, subject to aero dynamical load conditions, and as such having a specific transient behaviour. As I will present the second time derivative of the actuator torque has to be continuous; for multi rotor flight dynamics this is equivalent to having a continuous displacement pop, the sixth time derivative of the body displacement. To achieve feasible energy efficient trajectories, one must take into account both the base system and the control actuator dynamics.

To efficiently generate trajectories for agile quadrotor flight through maps of real-world environments is addressed in [59], where the straight-line route is translated into a smooth dynamically feasible polynomial trajectory and iteratively refined by a time allocation scheme that naturally performs a trade-off to minimize accelerations while attempting to fly at a desired velocity [59].

Based on analysis above the ultimate goal of my research is to achieve improvements of autonomous multi-rotor navigation with obstacle avoidance. In forthcoming chapters the above highlighted essential system design tools are analysed and, possible improvements are suggested for the following subjects:

a.) multi objective search and optimisation

b.) robust function approximation by fuzzy logic systems using a.)

c.) complex dynamic system identification by b.)

d.) optimal feasible trajectory design for c.)

e.) reduction of d.) to training data set for c.)

2 GENETIC ALGORITHMS FOR MULTI-OBJECTIVE SEARCH AND OPTIMISATION

2.1 Literature Synopsis

A genetic algorithm (GA) is constructed on bases of imitating natural biological processes and natural Darwinian evolution [27]. GAs are widely used as a search and optimisation tool [28]. Real-life optimisation problems often have multiple objectives. The comparison of two vectors in this case is not trivial. Often a simple weighted sum is used, but its drawbacks are widely known [13].

The principle of multiple criteria, multi-objective optimisation is different from that of a single objective optimisation. In single objective optimisation, the goal is to find the best design solution, which corresponds to the minimum (or maximum) value of the objective function, while in a multiple criteria optimisation of conflicting objectives there is no single optimal solution. The interaction among different objectives gives rise to a set of compromised solutions, known as Pareto-optimal solutions that comprise the so called Pareto-front region. Without any further considerations and new preferences none of the Pareto-optimal solutions of the Pareto-front can be identified in Pareto sense to be better than the others. The goal in a multi-objective optimisation is to find as many Pareto-optimal solutions as possible. Once such solutions are found, it usually requires a higher level decision making with other considerations to choose one of them for implementation.

There are two general objectives in a multiple criteria optimisation:

i.) find solutions close to the true Pareto-front and

ii.) find solutions that are widely different from each other.

The first task is intended to satisfy all the optimality conditions solution candidates. The second task is designed to ensure that no bias exists towards any particular Pareto non-dominated solution region.

Evolutionary algorithms are particularly suitable to solve multiobjective optimisation problems because they deal simultaneously with a set of possible solutions, the so-called population. This allows finding an entire set of Pareto optimal solutions in a single run of the algorithm, instead of having to perform a series of separate runs as is the case in traditional mathematical programming techniques. Evolutionary algorithms are less susceptible to the shape or continuity of the Pareto front, whereas these two issues are a real concern for other mathematical programming techniques.

Multiobjective optimisation (also called multi-criteria optimisation, multi-performance or vector optimisation) can be defined as the problem of finding a vector of decision variables which satisfies constraints and optimises a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria that are usually in conflict with each other. Hence, the term “optimise” means finding such a solution that would give the values of all the objective functions acceptable to the designer [53].

Formally, we can state it as follows: we want to find the vector $\bar{x}^* = \left[x_1^*, x_2^*, \ ... \ , x_n^*\right]^T$ that will satisfy the m inequality constraints $g_i(\bar{x}) \geq 0 \quad i = 1,2, \ ... \ , m$ and the p equality constraints: $h_j(\bar{x}) = 0 \quad j = 1,2,\ \dots\ , p$, while optimising the vector function: $\bar{f}(\bar{x}) = \left[f_1(\bar{x}), f_2(\bar{x}),\ \dots\ , f_k(\bar{x})\right]^T$, where $\bar{x} = [x_1, x_2,\ \dots\ , x_n]^T$ is the vector of decision variables.

The concept of Pareto optimum was formulated by Vilfredo Pareto in the XIX century [54], and constitutes by itself the origin of research in multiobjective optimisation. We say that a point $\bar{x}^* \in F$ is Pareto optimal if for every $\bar{x} \in F$ we have $\underset{i \in I}{\wedge}\left(f_i(\bar{x}^*) \nprec f_i(\bar{x})\right) \wedge (\exists i \in I)\left(f_i(\bar{x}^*) \succ f_i(\bar{x})\right)$ where $\nprec$ means "is no worse than" and $\succ$ means "is better than" [24].

In other words, this definition says that $\bar{x}^*$ is Pareto optimal if there exists no feasible vector $\bar{x}$ that would improve any criterion without causing a simultaneous worsening in at least one other criterion. The Pareto optimum almost always gives not a single solution, but rather a set of solutions called no inferior or non-dominated solutions. The following definitions ensure clear understanding if a set of solutions belongs to a local or global Pareto-optimal set, similar to the definitions of local and global optimal solutions in single objective optimisation problems:

- *Local Pareto-optimal non-dominated set*: If for every member *x* in a set *P*, there exist no solution *y* satisfying the infinity (or any other) norm condition $\|y - x\|_\infty \leq \varepsilon$, where ε is a small positive number (in principle, *y* is obtained by perturbing *x* in its close neighbourhood), which dominates any member in the set *P*, then the solutions belonging to the set *P* constitute a local Pareto-optimal non-dominated set.

- *Global Pareto-optimal Pareto-front set: If there exists no solution in the search* space which dominates any member in the set *P*, then solutions belonging to the set *P* constitute a global Pareto-optimal set, the true global Pareto-front.

There exists a major difference between a non-dominated set and a Pareto-front set. A non-dominated set is defined in the context of a sample of the search space. In a sample of search points, solutions that are not dominated (according to the above definition of Pareto dominance) by any other solutions in the sample space are non-dominated solutions. A Pareto-optimal, Pareto- front set is the ultimate non-dominated set, when the sample is the entire search space. The optima in the Pareto sense are going to be in the boundary of the design region, or in the locus of tangent points of objective functions. The region of Pareto-optimal points is called the Pareto-front. In general, it is not easy to find an analytical expression of the line or surface that contains these points. The common procedure is to find a large enough number of points in the search space and compute their corresponding fitness values. When we have a sufficient amount of non-dominated solutions, we may proceed to take the final decision [19].

Evolutionary algorithms are also capable of find solutions that are widely different from each other by applying a suitable niching operator. Due to genetic drift - stochastic errors associated with its operators, the genetic algorithm (GA) tends to converge to a single solution when used with a finite population [21], this phenomena is present in biological systems as well. Holland suggested the use of a "crowding" operator to identify situations in which more and more individuals dominate an environmental niche, since in those cases the competition for limited resources increases rapidly, which will result in lower life expectancies and birth rate.

DeJong experimented with such a crowding operator, which was implemented by having a newly formed offspring to replace the existing individual more similar to itself. The similarity between two individuals was measured in the genotype, by counting the number of bits along each chromosome that were equal in the two individuals being compared. DeJong used two parameters in his model: generation gap (G) and crowding factor (CF). The first parameter indicates the percentage of the population that is allowed to reproduce. The second parameter specifies the number of individuals initially selected as candidates to be replaced by a particular offspring [21].

In [Goldberg, 1987] a different approach is elaborated, where the population was divided in different sub-populations according to the similarity of the individuals in two possible solution spaces: the decoded parameter space (phenotype) and the gene space (genotype). They defined a sharing function as $\phi(d_{ij})$ follows: $\phi(d_{ij}) = \begin{cases} 1-\left(\dfrac{d_{ij}}{\sigma_{sh}}\right)^{\alpha} \\ 0,\ \text{else} \end{cases}$, $d_{ij} < \sigma_{sh}$, where normally $\alpha = 1$, and d_{ij} is a metric indicative of the distance between designs *i* and *j*, and σ_{sh} is the sharing parameter which controls the extent of allowed nice sharing. The fitness of a design *i* is then modified as: $f_{s_i} = \dfrac{f_i}{\sum_{j=1}^{M} \phi(d_{ij})}$, where M is the number of designs located in vicinity of the *i*th design. Deb and Goldberg proposed a way of estimating the parameter σ_{sh} in the phenotypic space as: $\sigma_{sh} = \dfrac{r}{\sqrt[p]{q}}$ where $r = \sqrt{\sum_{k=1}^{p} \dfrac{\left(x_{k,\max} - x_{k,\min}\right)^2}{2}}$ is the volume of a *p* dimensional sphere, where *p* refers to the number of variables $x_{k,1}, x_{k,2}, \ldots, x_{p,i}$ encoding in the *i*th individual design of the GA. In phenotypic sharing d_{ij} the distance between two individuals is measured in decoded parameter space, and can be calculated with a simple Euclidean distance in a *p* dimensional space. In genotypic sharing d_{ij} is defined as the Hamming distance between the strings and σ_{sh} is the maximum number of different bits allowed between the strings to form separate niches in the population. The experiments performed in [17] showed sharing as a better way of keeping diversity than crowding, and indicated that phenotypic sharing was better than genotypic sharing.

2.1.1 Sum–dominance Vector Comparison Method

The simplest GA objective vector comparison method is basically the one-norm distance from the origin. One popular way of formulating this approach is the weighted average objective function method, where the sum of objectives is multiplied by individual preference, weight factors and directly used as the scalar merit value for fitness calculation. It has been well studied in many publications; its drawbacks are well known [13].

For the sake of representation uniformity with my thesis I'm giving here one possible alternative formulation to this method: let's define a dominance relation $<_s$(***a*, *b***) (or briefly ***a*** $<_s$ ***b***) between two vectors of *n* elements ***a*** = (a$_i$*)* and ***b*** = *(b$_i$)*, for *i=1..n*, *n* ∈ $\mathbb{N}^+$, where each *i*th element type has a well-defined scalar '<' (less than) strict partial order binary endorelation and also the equivalence relation '=' is defined.

Vector $\boldsymbol{a}$ **sum-dominate**s vector $\boldsymbol{b}$, or briefly: $\boldsymbol{a} <_s \boldsymbol{b}$ if and only if $\sum_i a_i < \sum_i b_i$, for $i=1..n$.

We can define a **measurement value** for $<_s(\boldsymbol{a}, \boldsymbol{b})$ as $d_{<s}(\boldsymbol{a}, \boldsymbol{b}) = \sum_i (b_i - a_i)$.

This method is valid only if all objective functions are in the same range (normalised to the same interval of [0,1] for example). This can be easily achieved as we are investigating a finite number of individual results when determining the fitness of an individual in the population.

2.1.2 Sum–dominance Vector Comparison Method

A general multi-objective optimisation problem consists of a number of scalar minimisation objectives where every scalar objective function $f_i(\boldsymbol{x})$ is to be minimised simultaneously, where $\boldsymbol{x}$ is a p dimensional vector of parameters. As maximisation can be easily transformed to minimisation, the generality of the previous statement stands.

For minimisation a vector $\boldsymbol{x}^{(1)}$ Pareto-dominates $\boldsymbol{x}^{(2)}$, when no scalar component of $\boldsymbol{x}^{(2)}$ is less than the appropriate scalar component of $\boldsymbol{x}^{(1)}$, and at least one component of $\boldsymbol{x}^{(1)}$ is strictly smaller than the appropriate component of $\boldsymbol{x}^{(2)}$.

Up to date Pareto-comparison based methods are proven to be the most efficient multi-objective optimisation methods. Pareto based comparison [54] is the bases of a few popular methods like NSGA (Non-dominated Sorting GA) [18] and MOGA (Multi-Objective GA) [24].

2.1.3 Overview of MOGA – Block Type Non-dominance Ranking

MOGA is the rank based fitness assignment method introduced in [23]. Consider an individual $\boldsymbol{x}_i$ at generation t, which is dominated by $\boldsymbol{p}_i^t$ individuals in the current population. Its current position, individual’s rank can be given by: $rank(\boldsymbol{x}_i; t) = 1 + \boldsymbol{p}_i^t$.

All non-dominated individuals are assigned rank 1. Concerning fitness assignment, one should note that not all ranks would necessarily be represented in the population at a particular generation. The traditional assignment of fitness according to rank is extended as follows:

1. Sort population according to rank.

2. Assign fitness values to individuals by interpolating from the best (rank 1) to the worst (rank n) in the usual way, according to some function, usually linear but not necessarily.

3. Average the fitness values of individuals with the same rank, so that all of them will be sampled at the same rate. Note that this procedure keeps the global population fitness constant while maintaining appropriate selective pressure, as defined by the function used.

The fitness assignment method just described appears as an extension of the standard assignment of fitness according to rank, to which it maps back in the case of a single objective case, or that of non-competing objectives [24].

As the representation of this method for two objective functions forms blocks of individuals that dominate one possible solution, we shall refer to this ranking method as the Block-type ranking. Note that the basis of comparing two vectors in the original MOGA algorithm is the Pareto-comparison.

2.1.4 Overview of NSGA – Slice Type Non-dominance Ranking

NSGA is the following algorithm: Before the selection is performed, the population is ranked on the basis of an individual's Pareto non-domination. The non-dominated individuals present in the population are first identified from the current population. Then, all these individuals are assumed to constitute the first non-dominated front in the population and assigned a large dummy fitness value.

The same fitness value is assigned to give an equal reproductive potential to all these non-dominated individuals. These non-dominated individuals are ignored temporarily to process the rest of population in the same way to identify individuals for the second non-dominated front. These new set of points are then assigned a new dummy fitness value which is kept smaller than the minimum dummy fitness of the previous front.

This process is continued until the entire population is classified into a number of fronts. The population is then reproduced according to the assigned dummy fitness values. Since individuals in the first front have the maximum fitness value, they always get more copies than the rest of population. This method was intended to search for non-dominated regions or Pareto optimal fronts [18]. A slightly improved variant NSGA-II was developed in [20], where both rank (order of dominated precedence of the front to which the solution belongs) and within the same rank the distance from the Pareto front is used for fitness value assignment.

As the representation of this method for two objective functions forms slices of non-dominated individuals, we shall refer to this ranking method as the Slice-type ranking. Note that the basis of comparing two vectors in the original NSGA algorithm is the Pareto-comparison.

2.1.5 Overview of Pareto-dominance

Pareto-dominance is the classic method for comparing two GA individual's objective vectors:

- $\boldsymbol{a} = [a_i]$ Pareto-dominates $\boldsymbol{b} = [b_i]$ ($\boldsymbol{a} <_P \boldsymbol{b}$) if every a_i component of $\boldsymbol{a}$ is no worse than the corresponding b_i and there exists at least one a_j such that it is better than the corresponding b_j.

- No measurement of the amount of Pareto-domination exists.

Pareto dominance is not a complete order binary relation, as there is no clear Pareto dominance results for every $\boldsymbol{a}$ and $\boldsymbol{b}$ vectors. Pareto comparison of vectors of n elements can decide if one vector clearly dominates the other only for limited cases.

Without loss of generality we can assume for minimisation that each vector component is normed in the interval of $a_i, b_i \in [0,1]$. Thus in case for $n=2$ we can assume the average random solution objective value is $\boldsymbol{b} = [0.5, 0.5]$. This means that we can deduce Pareto dominance $\boldsymbol{a} <_P \boldsymbol{b}$ of vector $\boldsymbol{a} = [a_1, a_2]$ only for $a_i \leq 0.5$, which is only ¼ of the complete base vector space, similarly for $a_i \geq 0.5$ we can deduce $\boldsymbol{b} <_P \boldsymbol{a}$, which includes another ¼ of the complete base vector space.

Obviously in the observed general average case for two objectives, based on Pareto dominance we can draw conclusions only for half of the search space; for the other half of the space we cannot conclude any Pareto based relation.

Extending this observation to *n* dimensional vectors, for an average vector the Pareto dominance relation can yield results only for $1/2^{(n-1)}$ part of the search space. We can observe that this “Pareto indecisiveness” grows exponentially with the dimension of the search space.

Notice that in [27] GA efficiency compared to random search it is proved to be of order $O(n^3)$, where *n* stands for the number individuals in the population. Also a rule of thumb instruction is defined that the size of the population is proportional to the complexity of the problem.

Without loss of generality we can assume that the problem complexity for multi-objective optimisations is proportional to the number of objective functions - for a minimalistic approach we assume only linear *O(n)* complexity dependence. In this case of a *n*-objective GA relaying on Pareto dominance in the ranking operator hinders the search efficiency to $n^3/2^{(n-1)}$ as presented in Figure 3.

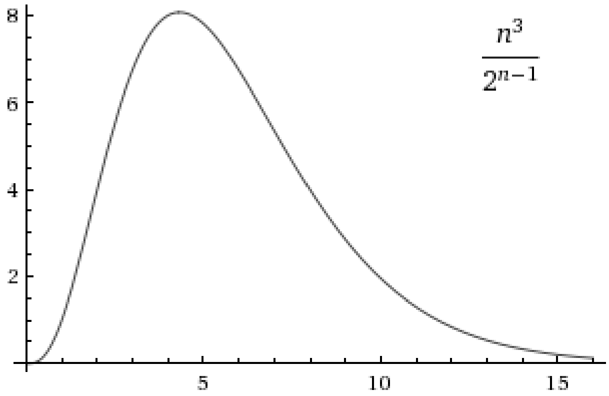


Figure 3.
Efficiency degradation of n-objective GA relaying on Pareto dominance comparison.

By my *Hypothesis I* this paper will propose and present the validity of a new general vector comparison operator, suitable for applications in any scientific field, including stochastic search, including evolutionary algorithms. The new method provides more information when comparing two vectors than the classic Pareto-based comparison, thus the GA convergence is faster, more efficient in its search.

2.1.6 Validating Quality of Multi-objective Search and Optimisation

Various stochastic search and optimisation methods based on stochastic gradient descent variations like simulated annealing, taboo search; swarm intelligence methods like ant colonies, bacterial search; and evolutionary algorithms like genetic algorithms have been successfully used for fuzzy system optimisation [32], [16], [2], [9]. Genetic algorithms are efficient, well studied simple stochastic optimisation methods of proven convergence, capable of global multi-objective search and optimisation of versatile complex systems. Capabilities and performance characteristics of properly formed GAs are defined in [27] so the effect on the GA performance becomes obvious if any supplementary operator – like a new vector comparison or a new niching operator is included into a simple GA.

This paper will prove the validity of a new general vector comparison operator, suitable for applications in any mathematical or engineering field, including stochastic search, and evolutionary algorithms. For a relation to be a proper strict inequality relation it must have the properties of irreflexivity, antisymmetry and transitivity. I shall mathematically prove these traits by analysing basic set cardinality properties of my proposed vector comparison operators. These new vector comparison methods provide more information when comparing two vectors than the classic Pareto-based comparison, thus the GA is faster, more efficient in its search.

Many benchmark problems were designed to test the effectiveness of a multi-objective GA. Of the most interest to us, thus what I will use is the analytically designed GA hard multi-objective function set presented in [79]; I have also used their generalisation to 4, 8 and 16 objectives. To have a clear, familiar simple baseline I have also implemented a single objective GA, which minimises the sum of 42, 84, 168 and 336 parameters – since the 2, 4, 8 and 16 multi-objective problems use this many parameters in total.

2.1.6.1 Simple Two Objective Optimisation Problem

According to [79] a simple two objective optimisation problem can be defined as:

$$\text{Minimize } f_1(\underline{x}) = f_1(x_1, x_2, \ldots, x_m),$$

$$\text{Minimize } f_2(\underline{x}) = g(x_{m+1}, \ldots, x_N) \cdot h(f_1(x_1, \ldots, x_m), g(x_{m+1}, \ldots, x_N)). \quad (15)$$

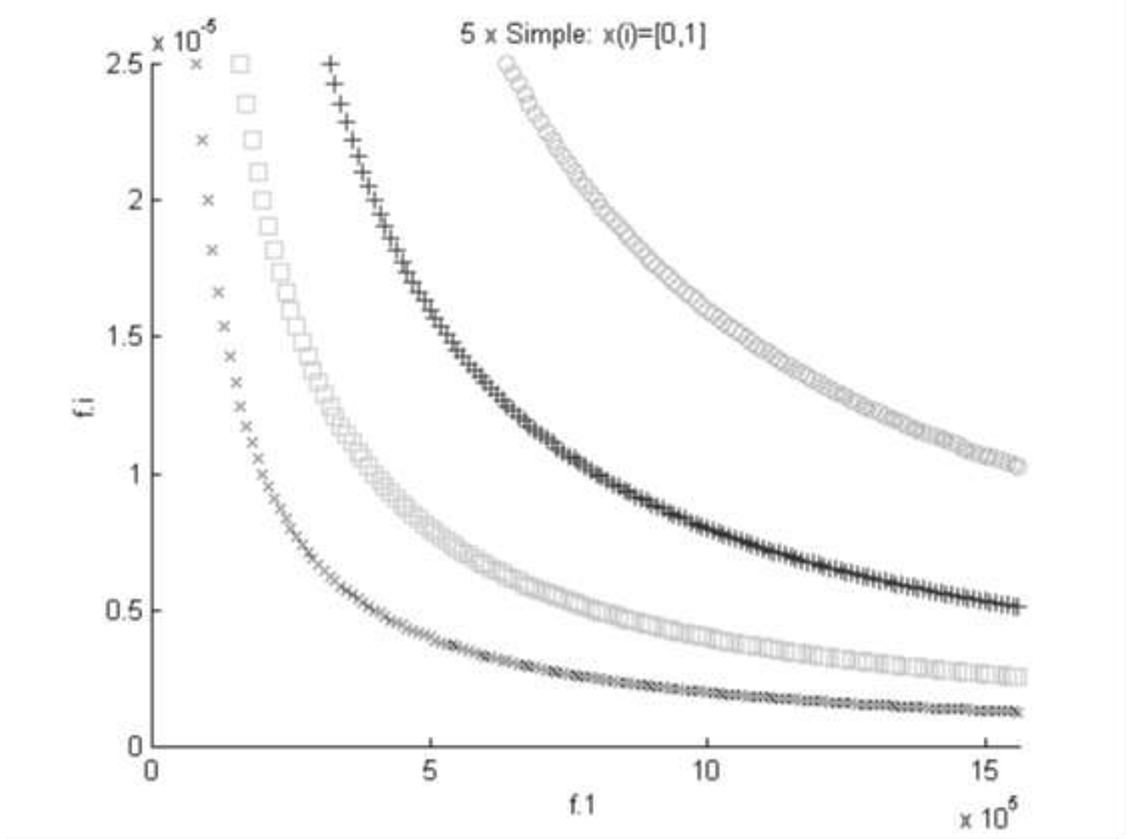


Figure 4.
Four hyperbolic lines (*f1f2 = c*) with $c_1 < c_2 < c_3 < c_4$ are shown.

The most simple case is where $f_1 = x_1$ and $g(x_2)$ (>0) is a function of x_2 only, and $h = g/f_1$. The first objective function f_1 is a function of x_1 only and the function f_2 is a function of both x_1 and x_2. In the function space (that is, a space with (f_1; f_2) values), the above two functions obey the following relationship: $f_1(x_1; x_2) \cdot f_2(x_1; x_2) = g(x_2)$.

For a fixed value of $g(x_2) = c$, a $f_1 – f_2$ plot becomes a hyperbola ($f_1 f_2 = c$) as shown for a 5 objective case in Figure 4, where 4 objectives $f_{2,3,4,5}$ are competing with the first objective f_1, along a Pareto-front like $f_1 f_i = c_i$, for i=2,3,4,5. Results of the optimisation that correspond to the smallest achievable c are the global, Pareto-optimal solutions.

For the general case of (15) the Global Pareto-optimal solutions are: $0 \leq x_1 \leq 1$ and $x_i = 0$ for $i = 2, 3, \ldots, N$. In [79] the proposal is $N=20$.

2.1.6.2 Deceptive Multi-objective Optimisation Problem

A deceptive g function is defined over binary alphabets, thereby making the search space discontinuous. Let us say that the following multi-objective function is defined over l bits, which is a concatenation of N substrings of variable size l_i such that $sum(l_i) = l$:

$$\text{Minimise } f_1 = 1 + u(l_1),$$

$$\text{Minimise } f_2 = sum(g(u(l_i)) / (1+u(l_1)), \tag{16}$$

where $u(l_1)$ is the unitation of the first substring of length l_1.

The first function f_1 is a simple one-min problem, where the absolute minimum solution is to have all 0s in the first sub-string. A one is added to make all function values strictly positive.

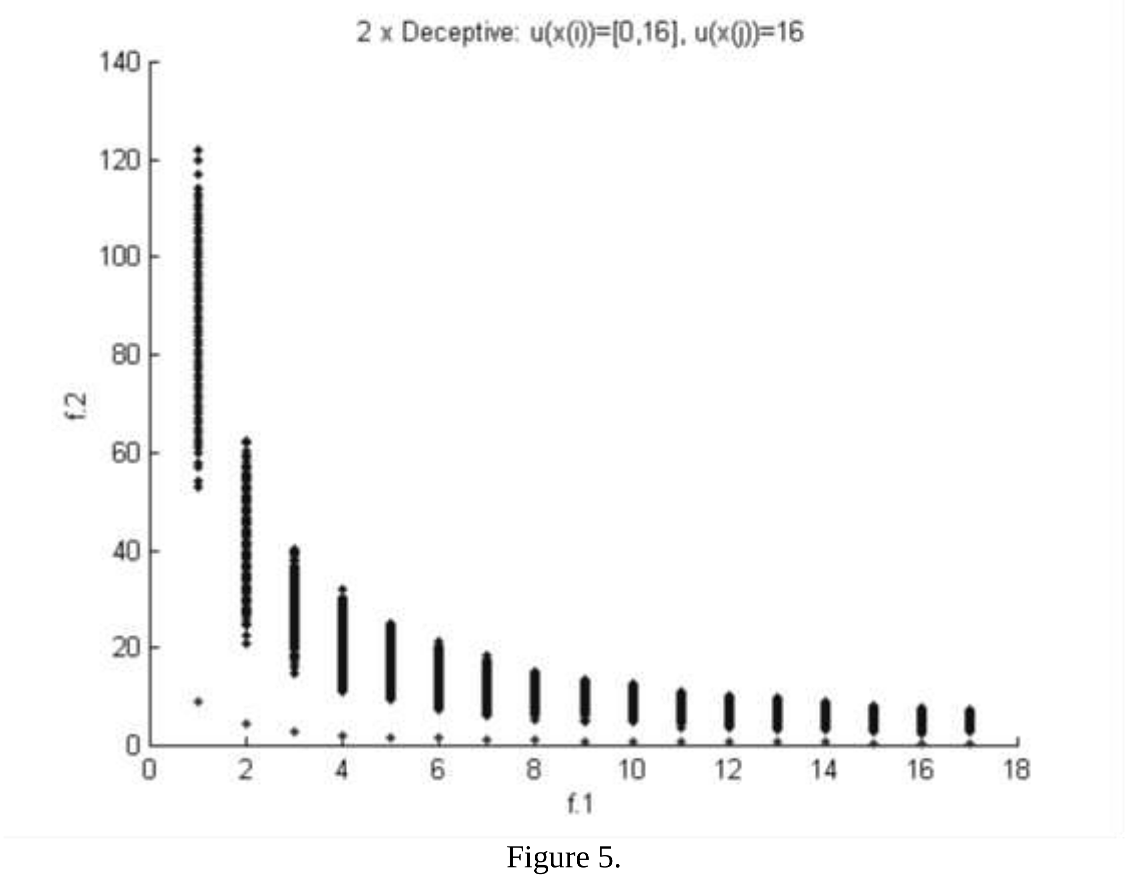


Figure 5.
10 000 randomly generated individuals and the true deceptive two objectives Pareto-front.

The function $g(l_1)$ is defined as $g(u(l_1)) = 2 + u(l_1)$; if $u(l_1) < l_1$, or $g(u(l_1)) = 1$; if $u(l_1) = l_1$.

This makes the true attractor (with all values of one in the substring) to have worst neighbours and with a function value $g(l_1) = 1$ and the deceptive attractor (with all 0s in the substring) to have good neighbours and with a function value $g(0) = 2$. Since, most of the substrings lead towards the deceptive attractor, GAs may find difficulty to converge to the true attractor (all 1s) as in Figure 5. Since each g function has two minima (one true and another deceptive), there are a total of 2^N-1 local minima, of which only one is global. Global Pareto-optimal solutions are: $0 \leq u(x_1) \leq l_i$ and $u(x_i) = l_i$ for $i = 2, 3,\dots,N$. In [79] the proposal is $N=20$.

2.1.6.3 Multi-modal Multi-objective Problem

When the function $g(x_2)$ is multi-modal with local x_2 and global $\underline{x}_2$ minimum solutions, the corresponding two objective problem also has local and global Pareto-optimal solutions corresponding to solutions $(x_1;\ x_2)$ and $(x_1;\ \underline{x}_2)$, respectively. The Pareto-optimal solutions vary in x_1 values.

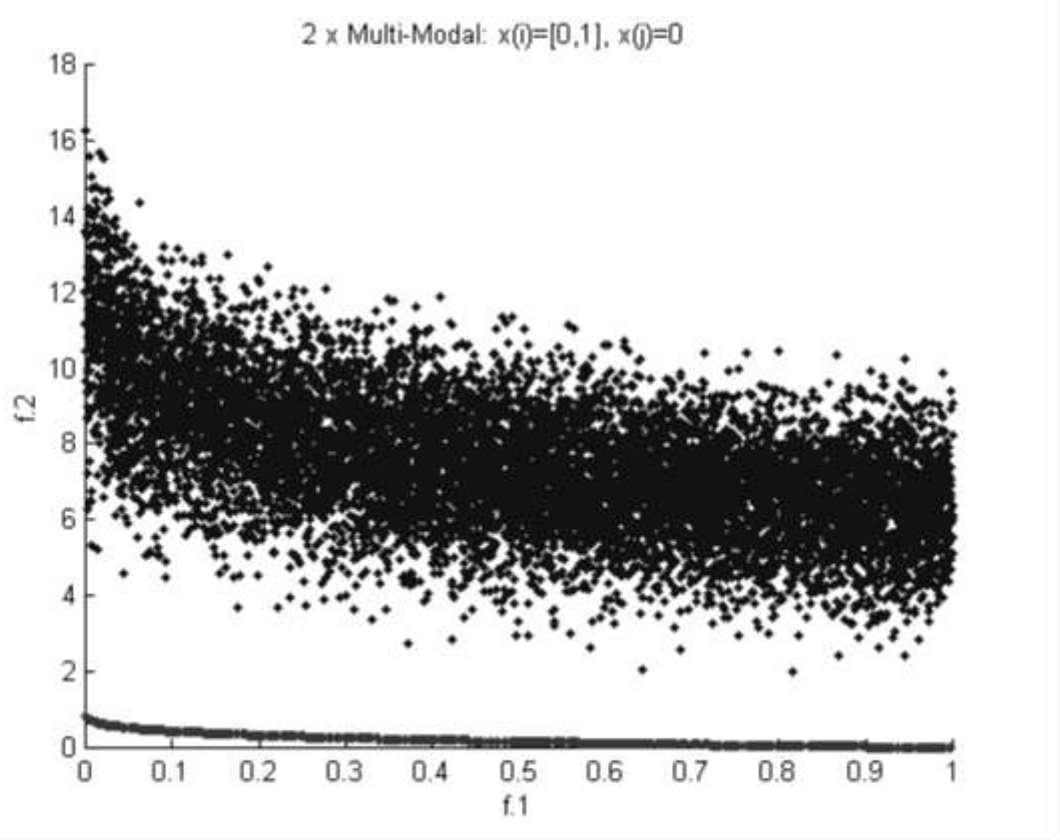


Figure 6.
10 000 randomly generated individuals and the true multi-modal 2 objectives Pareto-front.

We create a multi-modal, two objective optimisation problem by choosing a multi-modal $g(x_2)$ function:

Minimize $f_1(\underline{x}) = f_1(x_1, x_2,\dots,x_m)$,

Minimize $f_2(\underline{x}) = g(x_{m+1},\dots,x_N) \cdot h(f_1(x_1,\dots,x_m),g(x_{m+1},\dots,x_N))$,

$h = g/f_1$, where

$$g(x_2,\dots,x_N) = 1 + 10(N-1) + \mathrm{sum}(x_i^2 - 10\cos(2\pi x_i)). \quad (17)$$

Global Pareto-optimal solutions are: $0 \leq x_1 \leq 1$ and $x_i = 0$ for $i = 2, 3,\dots,N$ as in Figure 6. In [79] the proposal is $N=20$.

To highlight the specifics, where the difficulty lies with optimisations along a multi-modal Pareto front, please observe the magnified part of the search space of the solution distribution in Figure 7.

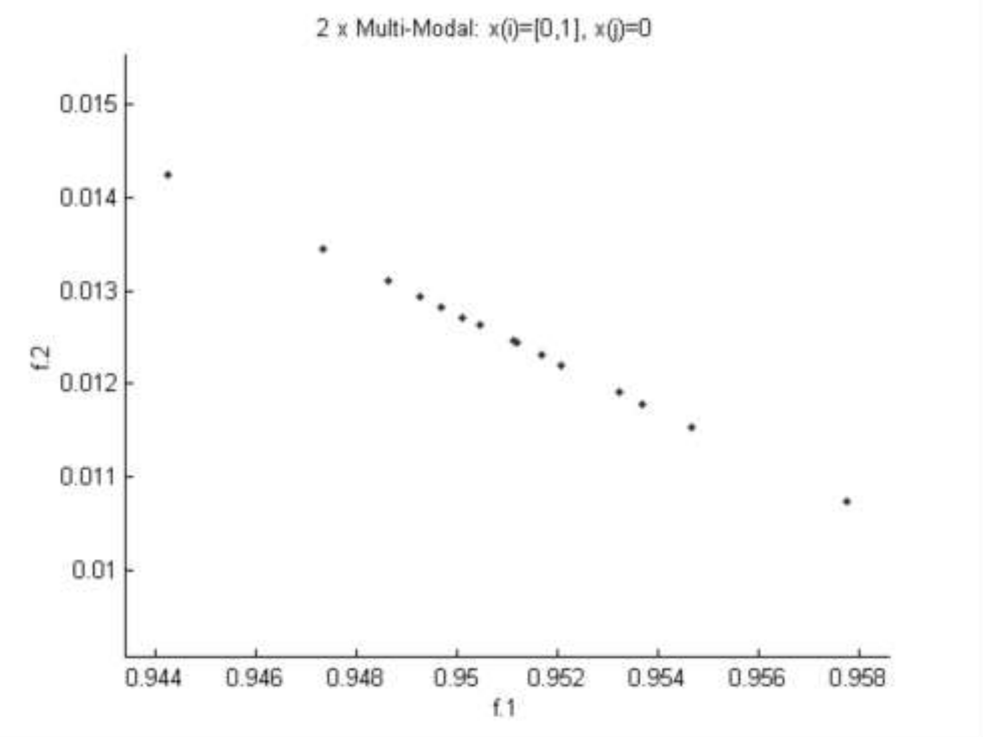


Figure 7.
A magnified portion of the true multi-modal Pareto-front.

2.1.6.4 *Convex and Non-convex Pareto-optimal Fronts*

We choose the following function for *h*:

Minimize $f_1(\underline{x}) = f_1(x_1, x_2,\ldots,x_m)$,

Minimize $f_2(\underline{x}) = g(x_{m+1},\ldots,x_N)^{\cdot}h(f_1(x_1,\ldots,x_m),g(x_{m+1},\ldots,x_N))$, where

$h(f_1, g) = 1 - (f_1/\beta g)^{\alpha}$, if $f_1 \leq \beta g$; and 0 otherwise. (18)

With this function, we may allow $0 \leq f_1$, *g* may be any function $0<g$. The global Pareto-optimal set corresponds to the global minimum of *g* function. The parameter β is a normalisation factor to adjust the range of values of functions f_1 and *g*. To have a significant Pareto-optimal region, β may be chosen as $f_{1,\max}/g_{\min} \leq \beta$, where $f_{1,\max}$ and $g_{\min}$ are the maximum value of the function f_1 and the minimum (or global optimal) value of the function *g*, respectively.

The above function can be used to create multi-objective problems having convex Pareto-optimal set by setting $\alpha \leq 1$.

Global Pareto-optimal solutions are: $0 \leq x_1 \leq 1$ and $x_i = 0$ for $i = 2, 3,\ldots,N$ as in Figure 8. In [79] the proposal is to use *N=20* in equation (18), which results in a suffitienty versatile search space.

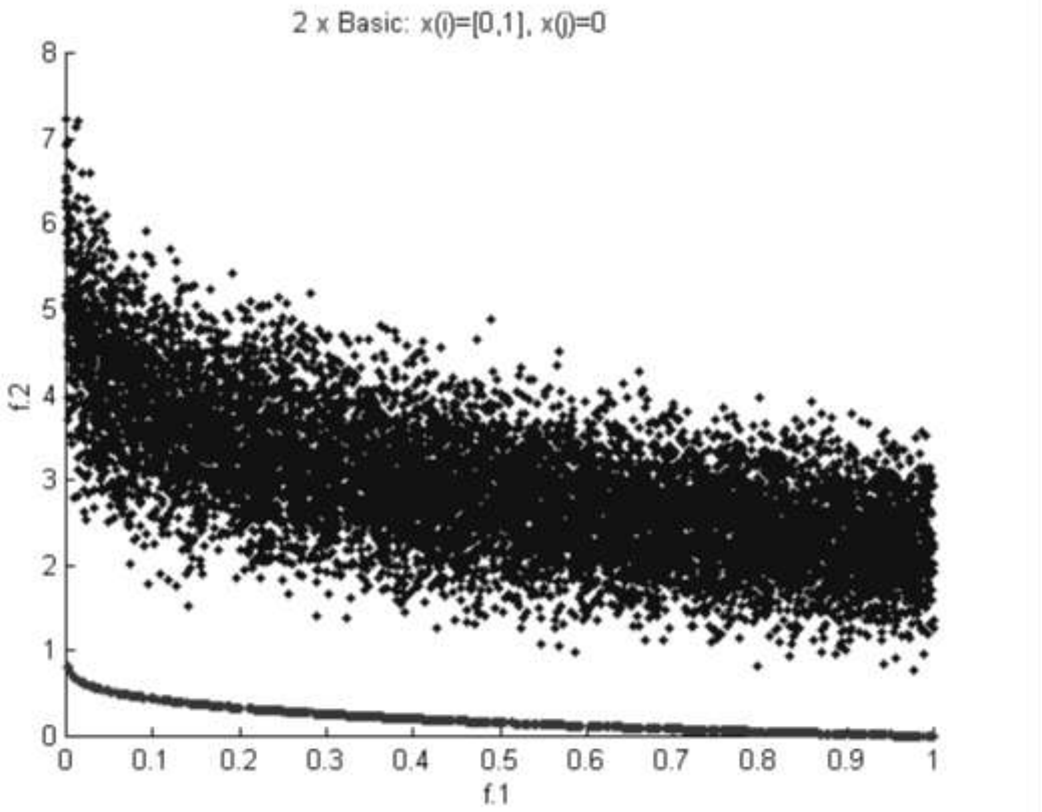


Figure 8.
10 000 randomly generated individuals and the true convex two objectives Pareto-front.

Note that when $\alpha > 1$, the resulting Pareto-optimal front is non-convex. It is important to note that when $\alpha > 1$ is used, the classical weighted sum method cannot find any intermediate Pareto-optimal solution by using any weight vector.

Although there exist other methods (such as ε -perturbation method or goal programming method), they require problem knowledge and, moreover, require multiple application of the single objective optimiser.

Global Pareto-optimal solutions are: $0 \leq x_1 \leq 1$ and $x_i = 0$ for $i = 2, 3, \ldots, N$ as presented in Figure 9. In [79] the proposal is to use $N=20$ in equation (18), which results in a suffitienty versatile search space.

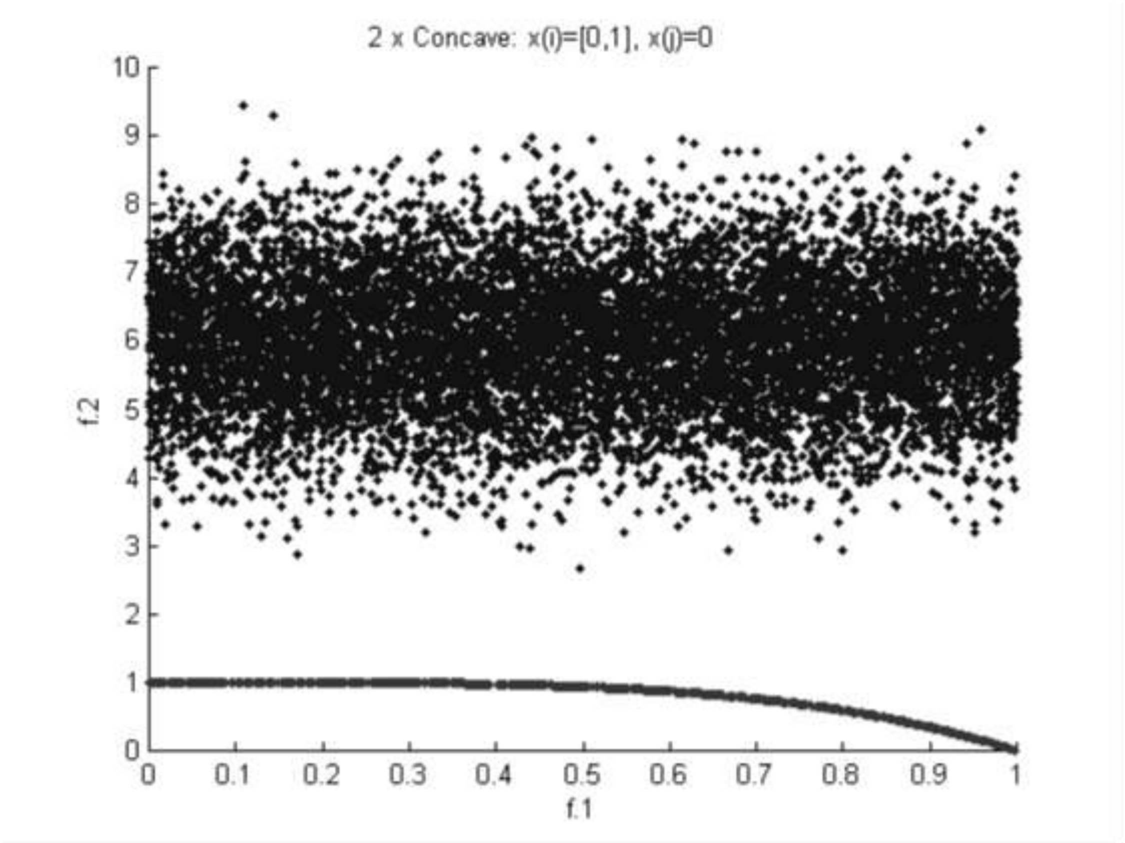


Figure 9.
10 000 randomly generated individuals and the true non-convex two objectives Pareto-front.

2.1.6.5 Discontinuous Pareto-optimal Front

We have to relax the condition for h being a monotonically decreasing function of f_1 to construct multi-objective problems with a discontinuous Pareto-optimal front. In the following, we show one such construction where the function h is a periodic function of f_1:

$$\text{Minimize } f_1(\underline{x}) = f_1(x_1, x_2,\ldots,x_m),$$

$$\text{Minimize } f_2(\underline{x}) = g(x_{m+1},\ldots,x_N)\cdot h(f_1(x_1,\ldots,x_m),g(x_{m+1},\ldots,x_N)),$$

$$h(f_1, g) = 1 - (f_1/\beta g)^{\alpha} - (f_1/\beta g)\sin(2\pi q f_1) \qquad (19)$$

where g may be any function $0<g$. The parameter q is the number of discontinuous regions in an unit interval of f_1. Since the h (and hence f_2) function is periodic to x_1 (and hence to f_1), we generate discontinuous Pareto-optimal regions.

Global Pareto-optimal solutions are: $0 \le x_1 \le 1$ and $x_i = 0$ for $i = 2, 3,\ldots,N$ as in Figure 10. In [79] the proposal is $N=20$.

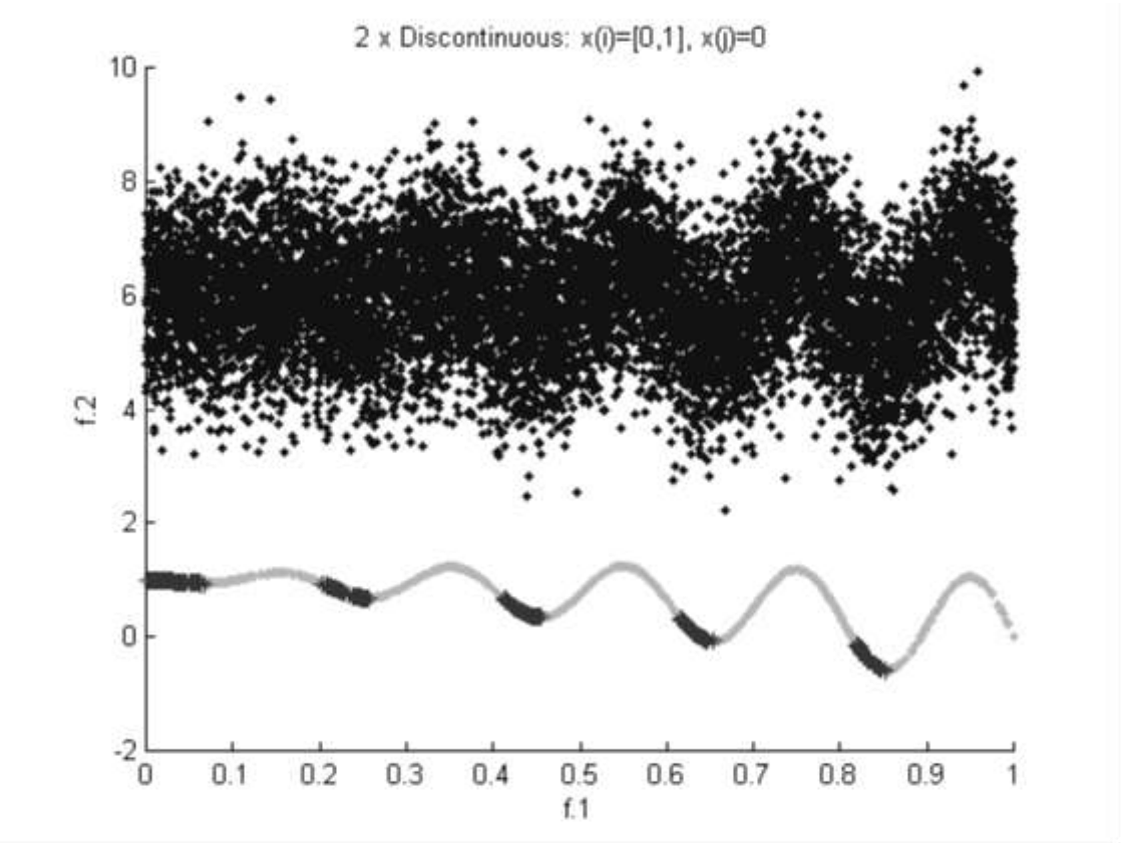


Figure 10.
10 000 randomly generated individuals and the true discontinuous two objectives Pareto-front along the non-dominated region.

2.1.6.6 Biased Search Space

The function *g* makes a major role in introducing difficulty to a multi-objective problem. Even though the function *g* is not chosen to be a multi-modal function nor to be a deceptive function, with a simple monotonic *g* function the search space can have adverse density of solutions towards the Pareto-optimal region. Consider the following function for *g*:

$$\text{Minimize } f_1(\underline{x}) = f_1(x_1, x_2,\ldots,x_m),$$

$$\text{Minimize } f_2(\underline{x}) = g(x_{m+1},\ldots,x_N)\cdot h(f_1(x_1,\ldots,x_m),g(x_{m+1},\ldots,x_N)),$$

$$h = g/f_1, \text{ where} \qquad (20)$$

$$g(x_{m+1},\ldots x_N) = g_{\min} + (g_{\max} - g_{\min}) ((\mathrm{sum}(x_i) - \mathrm{sum}(x_{i,\min})) / (\mathrm{sum}(x_{i,\max}) - \mathrm{sum}(x_{i,\min})))^{\gamma}$$

where $g_{\min}$ and $g_{\max}$ are minimum and maximum function values that the function *g* may take. The values of $x_{i,\max}$ and $x_{i,\min}$ are minimum and maximum values of variable x_i. It is important to note that the Pareto-optimal region occurs when *g* takes the value $g_{\min}$. The parameter γ controls the biasness in the search space. If $\gamma < 1$ then the density of solutions increases while getting further away from the Pareto-optimal front.

Global Pareto-optimal solutions are: $0 \le x_1 \le 1$ and $x_i = 0$ for $i = 2, 3,\ldots,N$ as in Figure 11. In [79] the proposal is *N=20*.

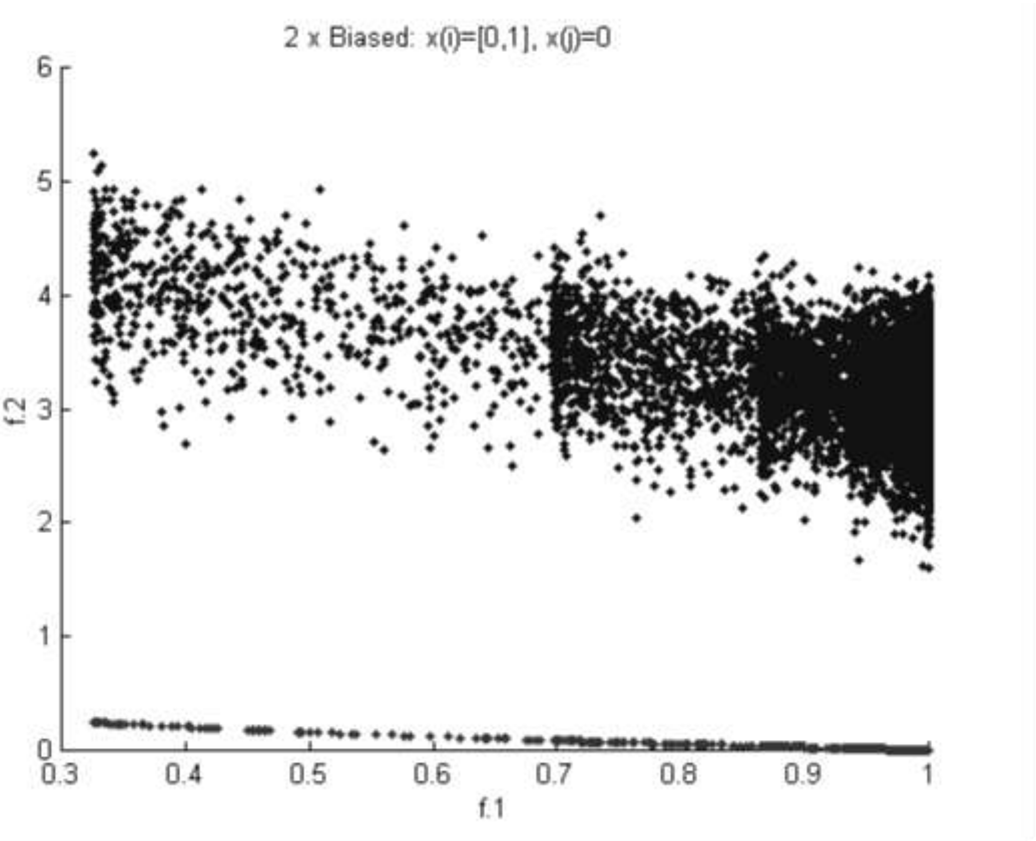


Figure 11.
10 000 randomly generated individuals and the true biased two objectives Pareto-front.

Random search methods are likely to face difficulty in finding the Pareto-optimal front in the case with γ close to zero, mainly due to the low density of solutions towards the Pareto-optimal region.

Although multi-objective GAs, in general, will progress towards the Pareto-optimal front, a different scenario may emerge. Although for values of γ greater than one, the search space is biased towards the Pareto-optimal region, the search in a multi-objective GA with proportionate selection and without mutation or without elitism is likely to slow down near the Pareto-optimal front. In such cases, the multi-objective GAs may prematurely converge to a front near the true Pareto-optimal front. This is because the rate of improvement in g value near the optimum ($x_2 \approx 0$) is small with $\gamma << 1$. Nevertheless, a simple change in the function g with a change in γ suggested above will change the landscape drastically and multi-objective optimisation algorithms may face difficulty in converging to the true Pareto-optimal front.

2.1.6.7 Generalisation of Two Objectives to Four Objectives

The idea is to simply introduce a third function f_3 that is similar to the first one f_1 but having different domains. The fourth function f_4 is chosen to be similar to the second function f_2, but also having different domains. Thus a general 4 objective problem is:

$$\begin{aligned}
&\text{Minimize } f_1(\underline{x}) = f_1(x_1, x_2,\ldots, x_m),\\
&\text{Minimize } f_2(\underline{x}) = g(x_{m+1},\ldots, x_N)\cdot h(f_1(x_1,\ldots, x_m), g(x_{m+1},\ldots, x_N)),\\
&\text{Minimize } f_3(\underline{x}) = f_1(x_{N+1}, x_{N+2},\ldots, x_{N+m}), \qquad (21)\\
&\text{Minimize } f_4(\underline{x}) = g(x_{N+m+1},\ldots, x_{N+N})\cdot h(f_3(x_{N+1},\ldots, x_{N+m}), g(x_{N+m+1},\ldots, x_{N+N})),
\end{aligned}$$

where function f_1, g and h can be chosen identically as in equations (15)-(20). With the same approach we can duplicate the four objective problems to eight competing objectives and further on to 16 competing objectives.

2.2 New Scientific Achievements

2.2.1 New Vector Comparison Operators

New methods for comparing two GA objective vectors are introduced in my publication [s2]. Two proper strict inequality operators are formulated in my first thesis, and also an additional comparison method is proposed.

2.2.1.1 Quantity-dominance Vector Inequality Operator

My idea behind the definition of a new vector comparison algorithm, named quantity-dominance is to extend the Pareto-dominance relation in a way that a domination decision could be also made for vectors, which are not comparable by Pareto dominance, while a human heuristic would name a clear preference. For example in case of maximisation problem solutions (1,1,0,1) and (0,0,1,0), where each scalar defines a normed merit of preference for maximisation the three out of four loci 'perfect' solution is clearly a better choice than a single loci of 'perfect' solution.

To introduce the quantity-dominance definition for a minimisation problem, let's define a dominance relation $<_n(\boldsymbol{a}, \boldsymbol{b})$ (or briefly $\boldsymbol{a} <_n \boldsymbol{b}$) between two vectors of n elements: $\boldsymbol{a} = (a_i)$ and $\boldsymbol{b} = (b_i)$, for $i=1..n$, $n \in \mathbb{N}^+$, where each i^{th} element type has a well-defined scalar '**<**' (less than) strict partial order binary endorelation and also the equivalence relation '=' is defined.

Let's define a helper function $\#_{n<}(\boldsymbol{a}, \boldsymbol{b})$, which for vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ defines two values $(g_a, l_a) = \#_{n<}(\boldsymbol{a}, \boldsymbol{b})$, where $g_a, l_a \in \mathbb{N}_0$ and g_a is equal to the cardinality of set $G_{ab}=\{ a_i \mid b_i < a_i \}$, $i=1..n$; and l_a is equal to the cardinality of set $L_{ab}=\{ a_j \mid a_j < b_j \}$, $j=1..n$.

THESIS I.a - DEFINITION:

For a minimisation problem vector $\boldsymbol{a}$ **quantity-dominate**s vector $\boldsymbol{b}$, or briefly: $\boldsymbol{a} <_n \boldsymbol{b}$ if and only if $\boldsymbol{g_a} < \boldsymbol{l_a}$.

We can define a **measurement value** for $<_n(\boldsymbol{a}, \boldsymbol{b})$ as $\mathbf{d}_{<n}(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{l_a} - \boldsymbol{g_a}$.

The defined dominance measurement is 0 for $\boldsymbol{a} = \boldsymbol{b}$ and for cases when no clear dominance can be determined. The same measurement value can be defined for Pareto-dominance as well. We can observe that quantity-dominance is a natural extension of Pareto-dominance, as the definition of quantity-dominance will degrade to the definition of Pareto-dominance if instead of condition $g_a < l_a$ we take a more strict criterion for dominance like: vector $\boldsymbol{a}$ Pareto-dominates vector $\boldsymbol{b}$, or briefly: $\boldsymbol{a} <_p \boldsymbol{b}$ if and only if $l_a > 1$ *AND* $g_b = 0$.

Notice that $l_a > 1$ *AND* $g_b = 0$ implies $g_a < l_a$, this means that if $\boldsymbol{a}$ Pareto-dominates $\boldsymbol{b}$ then $\boldsymbol{a}$ will also quantity-dominate $\boldsymbol{b}$, so quantity-dominance is an extension of Pareto-dominance.

Let's prove that the quantity-dominance operator $<_n(a, b)$ $(a <_n b)$ is a strict partial order binary endorelation, by proving that it is irreflexive, antisymmetric and transitive.

First of all, let us conclude that without loss of generality we can discard any K number of scalar components of vectors $\boldsymbol{a'}$ and $\boldsymbol{b'}$, such that $a_k = b_k$:

- for $K = n$ we have the trivial case of $\boldsymbol{a'} = \boldsymbol{b'}$; for $K = n\text{-}1$ we have a case of scalar comparison $a < b$ or $b < a$, which is a trivial, well defined problem, as by definition each scalar component has a well-defined scalar '**<**' (less than) strict partial order binary endorelation;

- for $K < n\text{-}1$ we have an new vector comparison problem $\boldsymbol{a} < \boldsymbol{b}$ where $\boldsymbol{a}$ and $\boldsymbol{b}$ are vectors of $n\text{-}k$ scalars, such that for every a_i, b_i scalar component of $\boldsymbol{a'}$, $\boldsymbol{b'}$ it holds that either $a_j < b_j$ or $b_i < a_i$, so the $\boldsymbol{a'}, \boldsymbol{b'}$ dominance question is equivalent to the $\boldsymbol{a}, \boldsymbol{b}$ dominance question for any $\boldsymbol{a'}$ and $\boldsymbol{b'}$ with any number of $a_k = b_k$ components additional to $a_j < b_j$ or $b_i < a_i$.

Reflexivity

To test the reflexivity property we observe $<_n(\boldsymbol{a}, \boldsymbol{a})$, where we have $(g_a, l_a) = \#_{n<}(\boldsymbol{a}, \boldsymbol{a})$. By the definition of $<_n$ we have $g_a = \#\, G_{aa} = \#\{\, a_i \mid a_i < a_i \,\} = 0$ and $l_a = \#\, L_{aa} = \#\{\, a_i \mid a_i < a_i \,\} = n$, for $i=1..n$. Since $g_a = l_a$, so we have $g_a \not< l_a$, which by definition means that vector $\boldsymbol{a}$ does NOT quantity-dominate itself. Thus we can conclude that the operator $<_n(\boldsymbol{a}, \boldsymbol{b})$ is **irreflexive**.

Symmetry

To test the symmetry property of quantity-dominance, without loss of generality we only analyse cases $\boldsymbol{a} \neq \boldsymbol{b}$, we observe $<_n(\boldsymbol{a}, \boldsymbol{b})$ and $<_n(\boldsymbol{b}, \boldsymbol{a})$ where we have $(g_a, l_a) = \#_{n<}(\boldsymbol{a}, \boldsymbol{b})$ and $(g_b, l_b) = \#_{n<}(\boldsymbol{b}, \boldsymbol{a})$.

By the definition of $<_n(\boldsymbol{a}, \boldsymbol{b})$ and $<_n(\boldsymbol{b}, \boldsymbol{a})$ we have:

$$g_a = \#\, G_{ab} = \#\{\, a_i \mid b_i < a_i \,\}, \#\{\, a_j \mid a_j < b_j \,\} = \#\, L_{ab} = l_a$$

$$l_b = \#\, L_{ba} = \#\{\, b_j \mid b_j < a_j \,\}, \#\{\, b_i \mid a_i < b_i \,\} = \#\, G_{ba} = g_b$$

We can observe that $l_b = g_a$, $g_b = l_a$. Let's assume $\boldsymbol{a} <_n \boldsymbol{b}$, which is by definition equivalent to $g_a < l_a$. Then from our observation we have: $l_b < g_b$, which implies $g_b \not< l_b$ what is by definition equivalent to $\boldsymbol{b} \not<_n \boldsymbol{a}$, for cases when $\boldsymbol{a} <_n \boldsymbol{b}$. Thus we can conclude that to the operator $<_n(\boldsymbol{a}, \boldsymbol{b})$ is **antisymmetric**.

Notice that based on the already proven irreflexive property we can even relax the $\boldsymbol{a} \neq \boldsymbol{b}$ condition for the antisymmetric property**.**

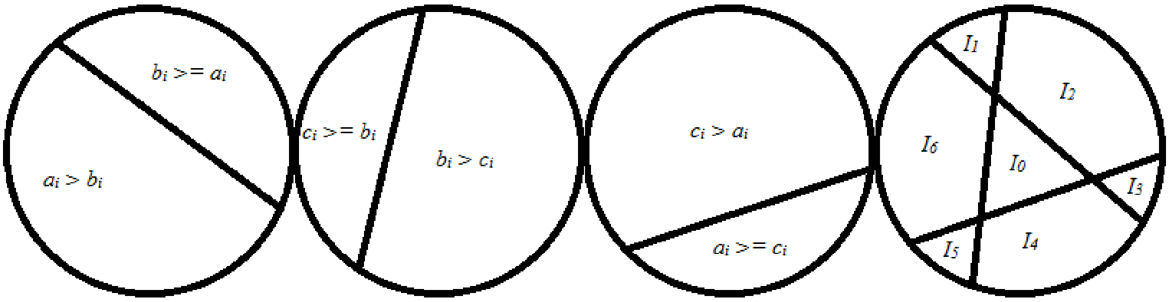
We can also introduce a corresponding operator $>_n(\boldsymbol{b}, \boldsymbol{a})$ for $g_b > l_b$, such that $\boldsymbol{a} >_n \boldsymbol{b} \iff \boldsymbol{b} <_n \boldsymbol{a}$, which has the same general properties as $<_n(\boldsymbol{a}, \boldsymbol{b})$.

Transitivity

For testing the transitivity property we observe $<_n(\boldsymbol{a},\boldsymbol{b})$, $<_n(\boldsymbol{b},\boldsymbol{c})$ and $<_n(\boldsymbol{c},\boldsymbol{a})$, where we have $(g_{ab}, l_{ab}) = \#_{n<}(\boldsymbol{a}, \boldsymbol{b})$, $(g_{bc}, l_{bc}) = \#_{n<}(\boldsymbol{b}, \boldsymbol{c})$ and $(g_{ca}, l_{ca}) = \#_{n<}(\boldsymbol{c},\boldsymbol{a})$. Though this transitivity analysis makes only sense for $\boldsymbol{a} \neq \boldsymbol{b} \neq \boldsymbol{c}$, we must accept to have any number of $a_j = b_j$ and $b_k = c_k$.

Let's introduce a second helper function $\#_{n=}(\boldsymbol{a}, \boldsymbol{b})$ for vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, which defines one value $e_a = \#_{n=}(\boldsymbol{a}, \boldsymbol{b})$, where $e_a \in \mathbb{N}_0$ and e_a is equal to the cardinality of set $E_{ab} = \{\, a_i \mid a_i = b_i \,\}$, $i=1..n$.

Let's perform a grouping of our vector indexes into sets $I_{0,1,2,3,4,5,6}$ in the following manner:



$G_{ab} = \{ a_i \mid b_i < a_i \};$ $G_{bc} = \{ b_i \mid c_i < b_i \};$ $G_{ca} = \{ c_i \mid a_i < c_i \};$

$LE_{ab} = \{ a_i \mid a_i \leq b_i \};$ $LE_{bc} = \{ b_i \mid b_i \leq c_i \};$ $LE_{ca} = \{ c_i \mid c_i \leq a_i \};$

Let's have $<_{\boldsymbol{n}}(\boldsymbol{a}, \boldsymbol{b})$, so that G_{ab} contains all those a_i for which $b_i < a_i$, where $i \in I_{ab} = I_0 \cup I_1 \cup I_2 \cup I_3$ and LE_{ab} contains all those a_i for which $a_i < b_i$ or $a_i = b_i$, where $i \in I_{ba} = I_4 \cup I_5 \cup I_6$.

Notice $\#G_{ab} = \#(I_0 \cup I_1 \cup I_2 \cup I_3)$, $\#LE_{ab} = \#(I_4 \cup I_5 \cup I_6)$, where $\#LE_{ab} = \#L_{ab} + \#E_{ab}$. Since by definition of $<_{\boldsymbol{n}}(\boldsymbol{a}, \boldsymbol{b})$ we have $g_{ab} < l_{ab}$, for any e_{ab} it will also hold that $g_{ab} < l_{ab}+e_{ab}$, *so* $\#G_{ab} < \#LE_{ab}$.

Let's also have $<_{\boldsymbol{n}}(\boldsymbol{b},\boldsymbol{c})$, so that G_{bc} contains all those b_i for which $c_i < b_i$, where $i \in I_{bc} = I_0 \cup I_2 \cup I_3 \cup I_4$ and LE_{bc} contains all those b_i for which $b_i <= c_i$, where $i \in I_{cb} = I_1 \cup I_5 \cup I_6$.

Notice $\#G_{bc} = \#(I_0 \cup I_2 \cup I_3 \cup I_4)$, $\#LE_{bc} = \#(I_1 \cup I_5 \cup I_6)$, and by definition of $>_{\boldsymbol{n}}(\boldsymbol{b}, \boldsymbol{c})$ we have $g_{bc} < l_{bc} + e_{bc}$.

To come to a contradiction let's assume that for $<_{\boldsymbol{n}}(\boldsymbol{a}, \boldsymbol{b})$ and $<_{\boldsymbol{n}}(\boldsymbol{b}, \boldsymbol{c})$ it is possible to have $<_{\boldsymbol{n}}(\boldsymbol{c}, \boldsymbol{a})$. Let's again perform the separation of indexes for G_{ca} to contain all those c_i for which $a_i > c_i$, where $i \in I_{ca} = I_0 \cup I_1 \cup I_2 \cup I_6$ and LE_{ca} to contain all those c_i for which $c_i <= a_i$, where $i \in I_{ac} = I_3 \cup I_4 \cup I_5$.

Notice $\#G_{ca} = \#(I_0 \cup I_1 \cup I_2 \cup I_6)$ and $\#LE_{ca} = \#(I_3 \cup I_4 \cup I_5)$.

Now we observe:

$$\#I_0 = \#\{ i \mid a_i > b_i \ \wedge \ b_i > c_i \ \wedge \ c_i > a_i \} = 0,$$

since by definition the '<' operator used for vector elements is a well-defined scalar '<' (less than) strict partial order binary endorelation, so it is transitive.

$$\#I_1 = \#\{ i \mid b_i \geq a_i \ \wedge c_i \geq b_i \ \wedge c_i > a_i \} = n_1$$

$$\#I_2 = \#\{ i \mid b_i \geq a_i \ \wedge b_i > c_i \ \wedge c_i > a_i \} = n_2$$

$$\#I_3 = \#\{ i \mid b_i \geq a_i \ \wedge b_i > c_i \ \wedge a_i \geq c_i \} = n_3$$

$$\#I_4 = \#\{ i \mid a_i > b_i \ \wedge b_i > c_i \ \wedge a_i \geq c_i \} = n_4$$

$$\#I_5 = \#\{ i \mid a_i > b_i \ \wedge c_i \geq b_i \ \wedge a_i \geq c_i \} = n_5$$

$$\#I_6 = \#\{ i \mid a_i > b_i \ \wedge c_i \geq b_i \ \wedge c_i > a_i \} = n_6$$

So we have $n_4 + n_5 + n_6 = g_{ab} < l_{ab} = n_1 + n_2 + n_3$ and $n_2 + n_3 + n_4 = g_{bc} < l_{bc} = n_1 + n_5 + n_6$. Through summing up these two inequalities we can conclude $n_4 < n_1$.

Also by this index separation we have $g_{ca} = n_1 + n_2 + n_6$; $l_{ca} = n_3 + n_4 + n_5$ and our assumption is that it is possible to have $g_{ca} < l_{ca} \Leftrightarrow n_1 + n_2 + n_6 < n_3 + n_4 + n_5$, but it is not possible! Since by substituting and summing up equations for inequalities $g_{ab} < l_{ab}$ and $g_{ca} < l_{ca}$ it would conclude that $n_6 < n_3$, and doing the same for $g_{bc} < l_{bc}$ with $g_{ca} < l_{ca}$ would conclude that $n_2 < n_5$, but this is not possible! Since then we would have simultaneously $n_4 + n_5 + n_6 = g_{ab} < l_{ab} = n_1 + n_2 + n_3$ and from summing up the inequalities from our conclusions $n_4 + n_6 + n_2 < n_1 + n_3 + n_5$, which is equivalent to have at the same time both $n_5 < n_2$ and $n_2 < n_5$, and that is clearly not possible. So our assumption is false that it is possible to have $<_n(\boldsymbol{c}, \boldsymbol{a})$ for $<_n(\boldsymbol{a}, \boldsymbol{b})$ and $<_n(\boldsymbol{b}, \boldsymbol{c})$, thus we can conclude that $<_n$ is **transitive**.

By this it is proven that the **quantity-dominance operator $>_n(\boldsymbol{a},\boldsymbol{b})$ ($\boldsymbol{a} >_n \boldsymbol{b}$) is a strict partial order binary endorelation**.

2.2.1.2 Quality–dominance Vector Inequality Operator

My idea behind the definition of quality-dominance is to further extend the quantity-dominance relation in a way that a decision could be made also for vectors, which are not comparable by quantity-dominance, but a human heuristic would name a clear preference like for example in case of maximisation problem solutions (0.9,0,0,0.9) and (0,0.1,0.1,0).

Let's define a dominance relation $<_q(\boldsymbol{a}, \boldsymbol{b})$ (or briefly $\boldsymbol{a} <_q \boldsymbol{b}$) between two vectors of n elements $\boldsymbol{a} = (a_i)$ and $\boldsymbol{b} = (b_i)$, for $i=1..n$, $n \in \mathbb{N}^+$, where each i^{th} element type has a well-defined scalar '<' (less than) strict partial order binary endorelation and also the equivalence relation '=' is defined.

Let's define a helper function $\#_{\mathbf{q}<}(\boldsymbol{a}, \boldsymbol{b})$, which for vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ defines two values $(g_a, l_a) = \#_{\mathbf{q}<}(\boldsymbol{a}, \boldsymbol{b})$, where $g_a, l_a \in \mathbb{N}_0$ and g_a is equal to the cardinality of set $G_{ab}=\{ a_i \mid b_i < a_i \}$, $i=1..n$; and l_a is equal to the cardinality of set $L_{ab}=\{ a_j \mid a_j < b_j \}$, $j=1..n$.

THESIS I.b - DEFINITION:

For a minimisation problem vector $\boldsymbol{a}$ **quality-dominates** vector $\boldsymbol{b}$, or briefly: $\boldsymbol{a} <_q \boldsymbol{b}$ if $g_a < l_a$ or in case of $g_a = l_a$ $\boldsymbol{a}$ quality-dominates vector $\boldsymbol{b}$ if $\sum_i(a_i - b_i) < \sum_j(b_j - a_j)$, where i is such that $a_i \in G_{ab}$ and j is such that $a_j \in L_{ab}$.

We can define a **measurement value** for $<_q(\boldsymbol{a}, \boldsymbol{b})$ as
$$d_{<q}(\boldsymbol{a}, \boldsymbol{b}) = \begin{cases} l_a - g_a, & \text{for } g_a < l_a \\ \sum_j \frac{(b_j - a_j)}{l_a} - \sum_i \frac{(a_i - b_i)}{g_a}, & \text{for } g_a = l_a \end{cases}$$
, where i is such that $a_i \in G_{ab}$ and j is such that $a_j \in L_{ab}$.

The defined dominance measurement is 0 for $\boldsymbol{a} = \boldsymbol{b}$ and for cases when no clear dominance can be determined. Notice that for cases of $g_a = l_a$ we have actually a more simple equivalent formula for $d_{<q}(\boldsymbol{a}, \boldsymbol{b}) = \sum_j \frac{(b_j - a_j)}{l_a} - \sum_i \frac{(a_i - b_i)}{g_a} = \sum_k \frac{(b_k - a_k)}{l_a}$, for $k=1..n$.

Reflexivity

To test the reflexivity property we observe $<_q(\boldsymbol{a}, \boldsymbol{a})$, where we have $(g_a, l_a) = \#_{\mathbf{q}<}(\boldsymbol{a}, \boldsymbol{a})$. By the definition of $<_q$ we have $g_a = \# G_{aa} = \#\{ a_i \mid a_i < a_i \} = 0$ and $l_a = \# L_{aa} = \#\{ a_i \mid a_i < a_i \} = 0$, for $i=1..n$. Since $g_a = l_a = 0$, and $\sum_i(a_i - a_i) = 0$ so we have $g_a \nless l_a$ and

$\sum_i(b_i - a_i) \ngtr 0$, which by definition means that vector ***a*** does NOT quality-dominate itself. Thus we can conclude that the operator $<_q$***(a,b)*** is **irreflexive**.

Symmetry

To test the symmetry property we observe $<_q$(***a, b***) and $<_q$(***b, a***) for ***a*** ≠ ***b***, where we have (g_a, l_a) = $\#_{q<}$(***a, b***) and (g_b, l_b) = $\#_{q<}$(***b, a***). We can also define $q_b = \sum_i(b_i - a_i)$, $q_a = \sum_j(b_j - a_j)$, where *i* is such that $a_i \in G_{ab}$ and *j* is such that $a_j \in L_{ab}$.

By the definition of $<_q$***(a,b)*** we observe two general cases:

- for cases of $g_a < l_a$ we have simple quantity-dominance, for which I have already proven the antisymmetric property.
- for cases of $g_a = l_a$ we have $\sum_i(a_i - b_i) < \sum_j(b_j - a_j)$, which is equivalent to $q_a < q_b$, that implies that $q_b \nless q_a$ and $\sum_j(b_j - a_j) \nless \sum_i(a_i - b_i)$, so that $\nless_q$***(a,b)***.

This means that $\boldsymbol{b} \nless_n \boldsymbol{a}$ when $\boldsymbol{a} <_q \boldsymbol{b}$, thus we can conclude that the operator $<_q$***(a,b)*** is **antisymmetric**.

Notice that based on the already proven irreflexive property we can relax the ***a*** ≠ ***b*** condition for the antisymmetric property**.**

We can also introduce operator $>_q$(***b,a***) for $g_b > l_b$, such that $\boldsymbol{a} <_q \boldsymbol{b} \iff \boldsymbol{b} >_q \boldsymbol{a}$, which has the same general properties as $<_q$***(a,b)***.

Transitivity

For testing the transitivity property we observe $<_q$(***a, b***), $<_q$(***b, c***) and $<_q$(***c, a***), where we have (g_{ab}, l_{ab}) = $\#_{q>}$(***a, b***), (g_{bc}, l_{bc}) = $\#_{q>}$(***b, c***) and (g_{ca}, l_{ca}) = $\#_{q>}$(***c, a***), and $q_{ab} = \sum_i(a_i - b_i)$, $q_{ba} = \sum_j(b_j - a_j)$, $q_{ac} = \sum_i(a_i - c_i)$, $q_{ca} = \sum_j(c_j - a_j)$, $q_{cb} = \sum_i(c_i - b_i)$, $q_{bc} = \sum_j(b_j - c_j)$, where $a_i \in G_{ab}$ and $a_j \in L_{ab}$, $c_i \in G_{ac}$ and $c_j \in L_{ac}$, $b_i \in G_{bc}$ and $b_j \in L_{bc}$. This makes only sense for ***a*** ≠ ***b*** ≠ ***c***.

By the definition of $<_q$***(a, b)***, $<_q$***(b, c)*** and $<_q$***(c, a)*** we observe two general cases:

- for cases of $g_{ab} < l_{ab}$ and $g_{bc} < l_{bc}$, we have simple quantity-dominance, for which I have already proven the transitivity property.
- for cases of $g_{ab} = l_{ab}$ or $g_{bc} = l_{bc}$ we need to analyse the less than '<' relation between scalar values of q_{xy}, where $x, y \in \{a, b, c\}$. By definition we have a well-defined scalar '<' (less than) strict partial order binary endorelation, which means that the transitivity property is fulfilled for all q_{xy}.

This means that $<_q$***(a, b)*** is also **transitive**.

By this it is proven that the **quality-dominance operator** $>_q$**(*a*,*b*)** (***a*** $>_q$ ***b*****) is a strict partial order binary endorelation**.

2.2.1.3 Any–dominance Vector Comparison Method

My idea behind the definition of any-dominance for GAs objective vector comparison is to define a comparison method, which uses the most detailed information on two objective vectors – both their quantity-dominance and the sum-dominance results.

Let's define a dominance relation $<_a(\boldsymbol{a}, \boldsymbol{b})$ (or briefly $\boldsymbol{a} <_a \boldsymbol{b}$) between two vectors of n elements $\boldsymbol{a} = (a_i)$ and $\boldsymbol{b} = (b_i)$, for $i=1..n$, $n \in \mathbb{N}^+$, where each ith element type has a well-defined scalar '<' (less than) strict partial order binary endorelation and also the equivalence relation '=' is defined.

THESIS I.c - DEFINITION:

For a minimisation problem vector $\boldsymbol{a}$ **any-dominates** vector $\boldsymbol{b}$, or briefly: $\boldsymbol{a} <_a \boldsymbol{b}$ if and only if ($\boldsymbol{a} <_q \boldsymbol{b}$ OR $\boldsymbol{a} <_s \boldsymbol{b}$).

We can define a **measurement value** for $<_a(\boldsymbol{a}, \boldsymbol{b})$ as
$d_{<a}(\boldsymbol{a}, \boldsymbol{b}) = \frac{l_a - g_a + \sum_i (b_i - a_i)}{2n}$, where $i=1..n$.

This method is valid only if all scalar components a_i and b_i are in the same range (normalised to the closed interval of [0,1] for example). For GAs this normalisation can be simply achieved as we are investigating a finite number of objective vectors when determining the fitness of an individual in the population.

2.2.1.4 Non-dominance Measurement Based Ranking

Measurement based ranking in multi-objective GAs is a new possibility in rank assignment, which is made possible by the definition of measurements in my Thesis I.a, I.b, I.c. In analogy to MOGA – Block Type Non-dominance Ranking introduced in [23] we can calculate with all $\boldsymbol{b}_i^t$ individuals, by which the observed vector is dominated; but instead of the pure number of such vectors, I'm proposing to sum up the measurements of being dominated.

THESIS I.d - DEFINITION:

At generation t the **non-dominance measurement based rank** of the ith individual $\boldsymbol{a}_i^t$ in a GA population, which is dominated by $\boldsymbol{b}_j^t$ individuals in the current population is the ith individual current position; the individual's rank can be defined as:

- $rank_i(\boldsymbol{a}_i^t)$ = sum of the non-dominated comparison measurements for every other $\boldsymbol{b}_j^t$ individual of generation t in correlation to the ith individual.

$rank(\boldsymbol{a}_i^t) = \sum_{j=1}^{n} \boldsymbol{d}_{<*}(\boldsymbol{b}_j^t, \boldsymbol{a}_i^t)$, where '*' can stand for any comparison method: {'s', 'P', 'n', 'q' or 'a'}.

2.2.1.5 Dominance Based Ranking

In analogy to MOGA – Block Type Non-dominance Ranking introduced in [23] we can calculate with all $\boldsymbol{b}_i^t$ individuals, but not those that dominate the observed vector, but with those, which are dominated by the observed vector.

THESIS I.e - DEFINITION:

At generation t the **dominance based rank** of the ith individual $\boldsymbol{a}_i^t$ in a GA population is the count of all $\boldsymbol{b}_j^t$ individuals in the current population, which are dominated by $\boldsymbol{a}_i^t$ is the ith individual current position; the individual's rank can be defined as:

- $rank_i(\boldsymbol{a}_i^t)$ = 1 + sum of the dominated $\boldsymbol{b}_j^t$ individuals of generation t in correlation to the ith individual.

rank($\boldsymbol{a}_i^t$) = $1+\#L_{ab}=1+\#\{\, \boldsymbol{a}_i^t \mid \boldsymbol{a}_i^t <_* \boldsymbol{b}_j^t \,\}$, where ‘*’ can stand for any comparison method: {‘s’, ’P’, ’n’, ’q’ or ’a’}.

2.2.1.6 Dominance Measurement Based Ranking

Similarly to non-dominance measurement based ranking we can sum up the measurements of dominance for each dominated $\boldsymbol{b}_j^t$ individual in the current population *t*.

THESIS I.f - DEFINITION:

At generation *t* the **dominance measurement based rank** of the i^{th} individual $\boldsymbol{a}_i^t$ in a GA population, which dominates all $\boldsymbol{b}_j^t$ individuals in the current population is the i^{th} individual current position, the individual’s rank can be defined as:

- $rank_i(\boldsymbol{a}_i^t)$ = sum of the dominated comparison measurements for every other $\boldsymbol{b}_j^t$ individual of generation *t* in correlation to the i^{th} individual.

rank($\boldsymbol{a}_i^t$) = $\sum_{j=1}^{n} \boldsymbol{d}_{<*}(\boldsymbol{a}_i^t, \boldsymbol{b}_j^t)$, where ‘*’ can stand for any comparison method: {‘s’, ’P’, ’n’, ’q’ or ’a’}.

2.2.2 Implementation of New Multi-objective Genetic Algorithms

The first key question is the coding of the possible solutions to the problem, which will evolve through a number of generations. Fundamental schemata theory suggests that small alphabets are good, because they maximise the number of schema available for genetic processing, so binary coding is implemented [27]. In order to avoid Hamming cliffs Gray code is used. To set equal grounds to all methods the very same initial population of 25 random individuals is used for every GA of the same number of objectives.

Low probability mutation is an important part of every GA, but to exclude its random possible improvement effect, I am not using any mutations for these experiments, where I want to concentrate only on effects of using different objective vector comparisons and ranking methods.

As chromosomes are simply the concatenated bit strings of all the parameters with fixed position for every gene, high probability (0.8) simple two point crossover will ensure low disruptiveness and high rate of inheritance during the reproductive phase. Stochastic universal sampling having minimal spread and zero bias is used for selection with a rather low selection pressure. Continuous exploration of the search space is achieved along with consistent convergence by the combination of genetic operators in this manner.

There are 5 vector comparison methods implemented and tested in this paper:

1. **PGA denotes the classical Pareto-comparison.

2. **NGA denotes the new quantitative comparison.

3. **QGA denotes the new quality comparison.

4. **DGA denotes the classical sum difference comparison.

5. **AGA denotes the result of (Q or D) comparison.

where ** can be substituted by MO, MM, DO, DM and NS according to the used rank assignment cases, as presented in Table I. Summary of examined GA variations.

There are 5 ranking methods tested in this paper:

1. MO*GA denotes the "block-type" multi-objective non-dominance ranking method proposed in [23].

2. NS*GA denotes the "slice-type" multi-objective non-dominance ranking method proposed in [18].

3. MM*GA denotes my proposal of a simple scalar ranking of non-dominance measurement, which is made possible for multi objective vector comparisons by the definition of my Thesis I.d.

4. DO*GA denotes my proposal of "block-type" multi-objective dominance ranking method by the definition of my Thesis I.e.

5. DM*GA denotes my proposal of a simple scalar ranking of dominance measurement, which is made possible for multi objective vector comparisons by the definition of my Thesis I.f.

where * can be substituted by P, N, Q, D and A according to the used comparison method as listed in Table I. Summary of examined GA variations.

Table I below summarizes the nomenclature and types of tested GAs.

GA-type	Rank assignment	Comparison method
MO(P)GA [23]	Block-type non-dominance	Pareto
MONGA	Block-type non-dominance	Quantity
MOQGA	Block-type non-dominance	Quality
MODGA	Block-type non-dominance	Sum Difference
MOAGA	Block-type non-dominance	Quality or Sum Difference
NS(P)GA [18]	Slice-type non-dominance	Pareto
NSNGA	Slice-type non-dominance	Quantity
NSQGA	Slice-type non-dominance	Quality
NSDGA	Slice-type non-dominance	Sum Difference
NSAGA	Slice-type non-dominance	Quality or Sum Difference
MMPGA	Measurement-based non-dominance	Pareto
MMNGA	Measurement-based non-dominance	Quantity
MMQGA	Measurement-based non-dominance	Quality
MMDGA	Measurement-based non-dominance	Sum Difference
MMAGA	Measurement-based non-dominance	Quality or Sum Difference
	Continued on the next page	

GA-type	Rank assignment	Comparison method
DOPGA	Block-type dominance	Pareto
DONGA	Block-type dominance	Quantity
DOQGA	Block-type dominance	Quality
DODGA	Block-type dominance	Sum Difference
DOAGA	Block-type dominance	Quality or Sum Difference
DMPGA	Measurement-based dominance	Pareto
DMNGA	Measurement-based dominance	Quantity
DMQGA	Measurement-based dominance	Quality
DMDGA	Measurement-based dominance	Sum Difference
DMAGA	Measurement-based dominance	Quality or Sum Difference

Table I. Summary of examined GA variations.

There were 6 minimisation problems previously described as equations (15)-(20) for 2 objectives. Also six of each of 4, 8, and 16 objective minimisation problems were constructed as described in (21), every one of these six multi-objective problems corresponds to one specific type of problems described in (34-39). To have a clear, familiar simple baseline I have also included a simple single objective problem, where the goal is to minimise the sum of parameters; I have used 4 such sets with the same number of parameters as the 2, 4, 8, and 16 objectives problems have; so the baseline goal is to minimises the sum of 42, 84, 168 and 336 parameters.

Each GA was tested 25 times on each test function; the total of 17500 GA runs was executed, and the average of these results is used for the result evaluation. The number of generations was limited to the maximum of 50, but a GA run could have stopped earlier, when saturation has occurred. The condition for saturation was a lower than 0.1% relative change of the mean value of all objective values over the entire population.

2.2.3 Results of Multi-objective Genetic Algorithm Evaluations

First of all I would like to point out the well-known disadvantage of the NS ranking method [18] – its computation complexity is $O(N^3)$. The computation complexity of the MO ranking [23], and all my proposed rankings MM, DO and DM is $O(N^2)$. To have a feel of what it means in this particular set of experiments, notice that when an MO, MM, DO or DM evaluation of a population takes 0.5 seconds, then the same evaluation for NS takes 3.5 seconds. For this particular setup of 25 individuals this is not a real problem, but for a population of 1000 individuals, the 7 seconds of $O(N^2)$ genetic operator evaluations turns into half an hour of NS $O(N^3)$ nightmare for each generation.

Results in Tables II-V below are the averages of 17500 experiments performed for all the six types of GA hard functions (34-39) and the simple single objective function. The same round is performed for all 2, 4, 8 and 16 objectives problems.

2.2.3.1 Results of Vector Comparison Method Analysis

Table II below presents the results for each tested vector comparison method – named in the first column: P as proposed in [54], A (Thesis I.c), D the classical average of weighted sum method, N (Thesis I.a), Q (Thesis I.b). The second column is for the number of generations required to find the final result. The third column presents the percentage of non-dominated individuals in the final generation. The fifth column contains the average distance of each free parameter of each individual in the final generation. The fifth column contains average of the non-dominated (minimal) distances of each free parameter of each non-dominated individual in the final generation.

Comparison method	Average of evaluated generations	Average of non-dominated individuals%	Average of Pareto-front mean distance	Average of Pareto-front minimal distance
P	17.3	75.47	0.414	0.0064
A	14.5	85.89	0.393	0.0066
D	14.5	86.02	0.395	0.0068
N	16.8	78.69	0.398	0.0065
Q	15.1	81.39	0.398	0.0065

Table II. Performance of vector comparison methods.

All GAs, no matter which vector comparison method is used, efficiently find the proximity of the true Pareto-front. Precision wise we can conclude that all tested vector comparison methods, the two classical (P and D) and my three new proposals (A, N and Q) are in average equally precise when it comes to locating the true Pareto-front.

By looking at the number of generations required to find the result we can conclude that all my proposed new vector comparison methods (A, N, Q) outperform the Pareto comparison by 20%.

By looking at the percentage of non-dominated individuals in the final generation we can again conclude that all my proposed new vector comparison methods (A, N, Q) outperform the Pareto comparison by 5-15%.

By this analysis I conclude that my Thesis I.a, I.b and I.c are proven valid.

2.2.3.2 Results of Ranking Method Analysis

Table III below presents the results for each tested ranking method named in the first column: DM (Thesis I.f), DO (Thesis I.e), MM (Thesis I.d), MO as proposed in [23], NS as proposed in [18]. The other columns contain the same categories as Table II.

Ranking method	Average of evaluated generations	Average of non-dominated individuals%	Average of Pareto-front mean distance	Average of Pareto-front minimal distance
DM	14.4	86.26	0.400	0.0067
DO	14.3	84.76	0.399	0.0066
MM	17.1	76.66	0.401	0.0068
MO	16.3	80.12	0.398	0.0066
NS	16.1	79.67	0.400	0.0062

Table III. Performance of ranking methods.

All GAs, no matter which ranking method is used, efficiently find the proximity of the true Pareto-front. For all non-dominated individuals having some 0.0065 for the average value of minimal distances from the Pareto front means that in average each of the 150 free parameters (40, 80, 160 and 320 free parameters for 2, 4, 8, and 16 objectives problems) arrives no further than 0.00005 (5e-05) to the theoretical optimum. This is quite a good result for a GA as small as 25 individuals running for 15 generations (which takes in average 8 seconds) – especially if we keep in mind that these optimisation problems are mathematically constructed to be hard to evaluate for GAs.

Looking at the population average distances from the Pareto front we see no significant differences between any of the ranking methods.

Precision wise we can conclude that all tested ranking methods, the two classical (MO and NS) and my three new proposals (MM, DO and DM) are in average equally precise when it comes to locating the true Pareto-front.

The number of generations required to find the result is the measurement of the performance of a GA. Obviously the dominance based ranking methods: DO (Thesis I.e) and DM (Thesis I.f) outperform the other ranking methods by 20%.

The measurement of efficiency of a GA is how many individuals of the final generation are non-dominated. The dominance based ranking methods: DO (Thesis I.e) and DM (Thesis I.f) outperform the other ranking methods by 5-10%.

By this analysis I conclude that my Thesis I.e and I.f are proven valid, while my **Thesis I.d is not sufficiently supported** by these results.

2.2.3.3 Results of Multi-objective Genetic Algorithms Analysis

Table IV below presents the results for each tested multi-objective GA named in the first two columns: first column naming the vector comparison method the second column naming the ranking method. The other columns contain the same categories as Table II and III.

Compar. method	Ranking method	Average of evaluated generations	Average of non-dominated individuals%	Average of Pareto-front mean distance	Average of Pareto-front minimal distance
P	DM	**14.1**	***85.57***	0.415	***0.0065***
P	DO	**14.1**	***85.99***	0.414	***0.0066***
P	MM	19.7	69.75	0.414	***0.0063***
P	MO	19.2	68.15	0.413	***0.0063***
P	NS	19.6	67.90	0.415	***0.0064***
A	DM	***14.5***	***86.59***	***0.392***	0.0070
A	DO	***14.5***	***86.29***	***0.393***	0.0071
A	MM	***14.6***	***84.32***	***0.392***	0.0068
A	***MO***	14.6	**87.82**	***0.393***	***0.0066***
A	*NS*	***14.3***	***84.43***	***0.397***	***0.0057***
D	DM	***14.5***	***87.01***	***0.394***	0.0070
D	DO	***14.4***	***85.53***	***0.394***	***0.0065***
D	***MM***	***14.5***	***85.54***	***0.395***	***0.0065***
D	MO	***14.5***	***85.30***	***0.394***	0.0073
D	***NS***	***14.5***	**86.73**	***0.397***	***0.0066***
N	DM	***14.3***	***84.91***	0.404	***0.0063***
N	**DO**	**14.1**	***82.11***	***0.402***	***0.0060***
N	MM	19.0	75.55	***0.397***	0.0075
N	MO	18.8	75.54	***0.395***	***0.0063***
N	NS	17.8	75.33	***0.395***	***0.0065***
		Continued	on the next	page	

Comparison method	Ranking method	Average of evaluated generations	Average of non-dominated individuals%	Average of Pareto-front mean distance	Average of Pareto-front minimal distance
Q	***DM***	***14.6***	**87.21**	***0.395***	***0.0066***
Q	DO	***14.5***	***83.87***	***0.394***	0.0068
Q	MM	17.9	68.13	0.408	0.0068
Q	***MO***	***14.4***	***83.79***	***0.395***	***0.0066***
Q	NS	***14.2***	***83.98***	***0.396***	**0.0057**
	min	**14.1**	67.90	**0.392**	**0.0057**
	max	19.7	**87.82**	0.415	0.0075
	average	***15.6***	***81.49***	***0.400***	***0.0066***
	stdev	2.0	6.63	0.008	0.0004

Table IV. Performance of multi-objective GAs.

The last four rows of table IV present the minimum, maximum, average values and standard deviation of numerical values in columns 3 to 6. Highlighted with ***bold-italic*** typeset is the average and better than average values, highlighted with **underlined bold** typeset are the (almost) non-dominated values. Plain underlined typeset highlights some combinations I will refer to in my summary below.

It is not possible to name a single, Pareto optimal multi-objective GA. If ones goal is to reach the best solutions in a small population or in cases where the evaluation of an individual is numerically intensive, so the $O(N^3)$ cost of NS ranking is not a big handicap, then one may go with this type of ranking, but my proposal is to use instead of the usual Pareto comparison either my quality based comparison (Thesis I.b) or even better the “any of quality or sum difference” comparison method (Thesis I.c) – these two GAs are highlighted with *underlined italic* typeset. On the other hand for hybrid GAs, where the final fine tuning of individuals is anyhow for a more efficient gradient-descent method my proposal is to use either the DO (Thesis I.e) or DM (Thesis I.f) ranking method, since they are the most efficient ones; my preference is quantity based comparison (N - Thesis I.a) and dominance ranking (DO – Thesis I.e) GA.

There are numerous publications referring to and some fewer even proving or presenting that the simple weighted average based ‘sum of fitness’ evaluation of a true multi-objective problem, especially for the non-convex cases is not to be used as it does not find all the possible solutions [13, 18, 19, 20, 23, 24, 47, 53, 79]. These studies omit to calculate with the effect of fitness sharing (or niching) in GAs. Please take a look at Figures 21-23 and table VI below. These results are for a non-convex (CO) objective function as in equation (18) for $\alpha = 2$, which was presented in Figure 9 of chapter 2.1.6.4.

The ‘D’ - simple “sum of objective value differences” comparison method, equivalent to the weighted average objective value with unit weights, actually outperforms the Pareto comparison in all observed criteria. GAs using the ‘D’ based ranking methods outperform the ‘P’ based GAs in number of performance (number of generations needed for the final result 14.7 vs. 16.5), efficiency (88.44% vs. 75.8% of the final population is non-dominated) and precision (3.6e-3 vs. 3.9e-3 for the minimal distance from the true Pareto optimum) as presented in Table V.

non-convex objective: Comparison method	Average of evaluated generations	Average of non-dominated individuals%	Average of Pareto-front mean distance	Average of Pareto-front minimal distance
P	*16.5*	*75.80*	*0.390*	*0.0039*
A	14.7	90.57	0.359	0.0039
D	***14.7***	***88.44***	***0.360***	***0.0036***
N	15.1	86.45	0.367	0.0036
Q	15.2	85.58	0.362	0.0038

Table V. Performance of vector comparison methods for the non-convex (CO) objective function (18) for $\alpha = 2$.

By looking at Figure 12 we see no issues with the actual distribution of the non-dominated individuals (as single dominated individual is lost in the 'upper left corner') all non-dominated solutions are neatly aligned along the true Pareto-front; both ends of the concave domain are represented.

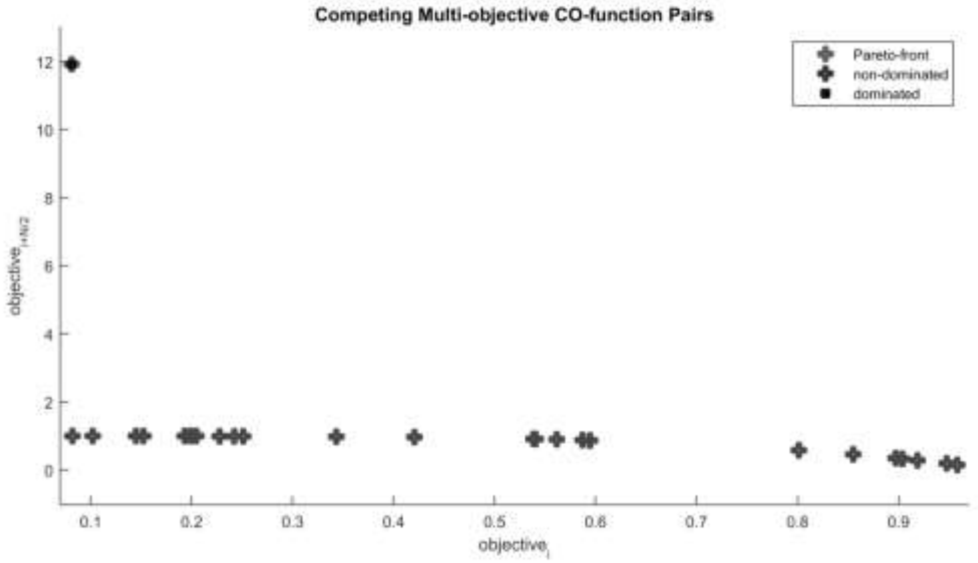


Figure 12.
CO objective values of the final generation D.DO.GA.

By looking at Figure 13 we see a textbook example of a fast, exponential GA convergence.

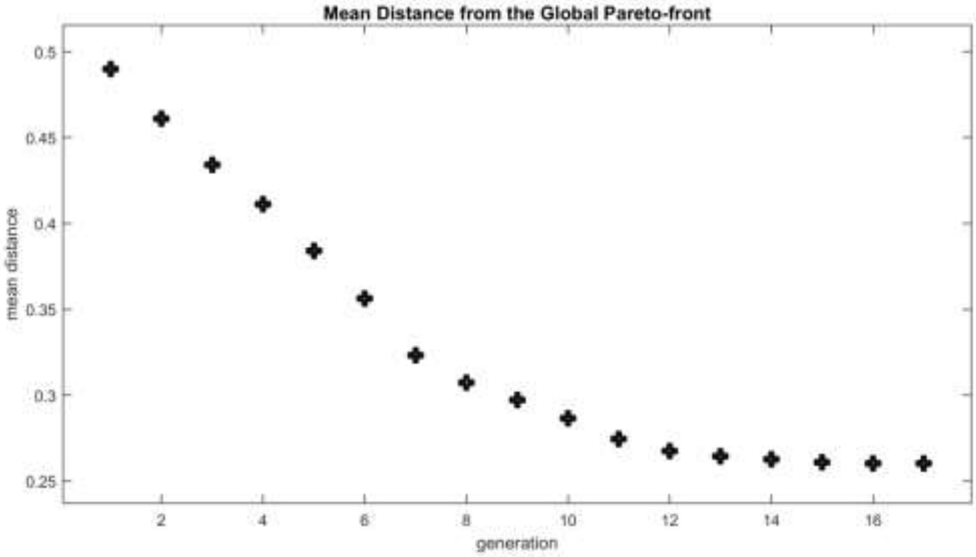


Figure 13.
Evolution, convergence of mean objective values D.DO.GA for CO objectives.

Figure 14 presents a neat dynamics in the evolution of the number of non-dominated individuals expressed as their relative percentage compared to the complete population size.

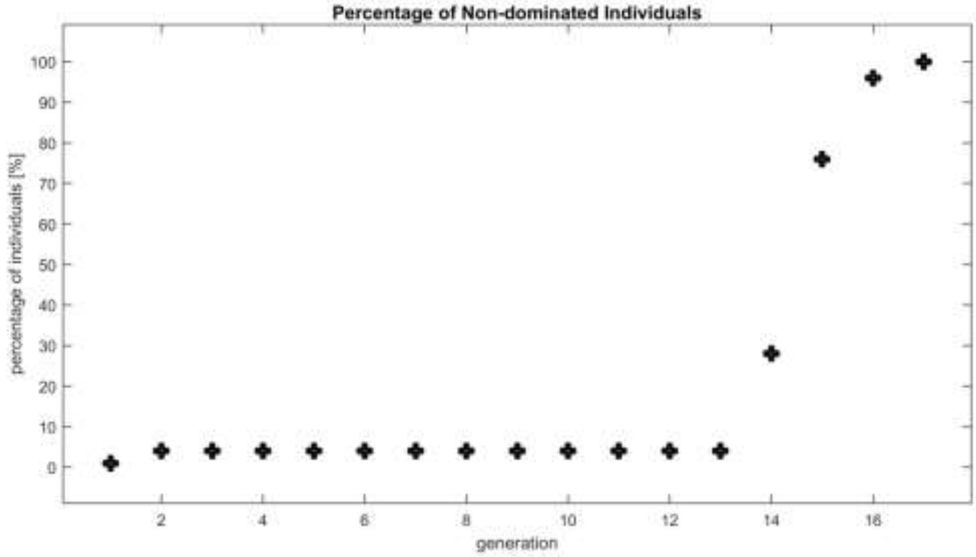


Figure 14.
Evolution, convergence of non-dominated solutions D.DO.GA for CO objectives.

Figure 15 presents an exponential dynamics in the saturation of individuals (in the non-dominated niche – the proximity of the true Pareto-front), expressed as the relative percentage of individuals in the final (non-dominated) niche compared to the complete population size.

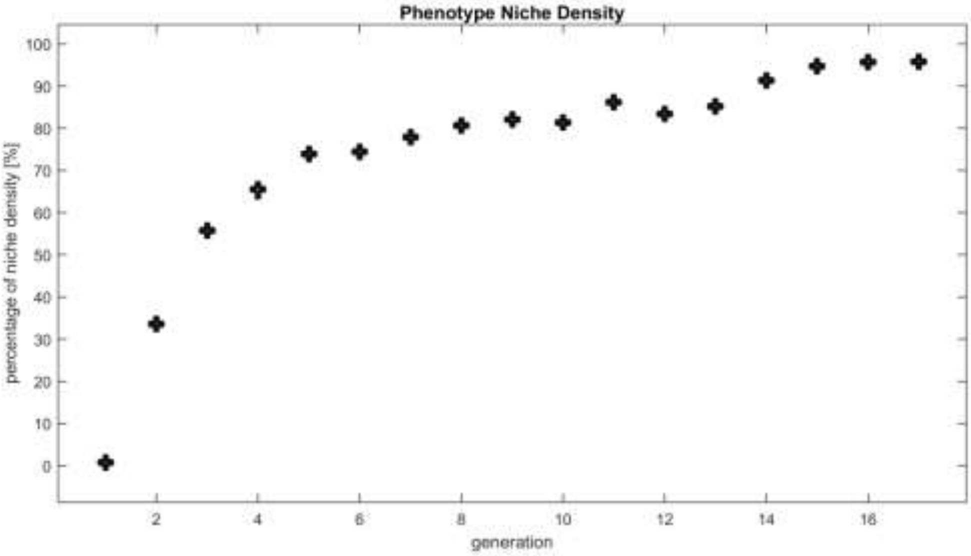


Figure 15.
Evolution, convergence of niche saturation around non-dominated solutions D.DO.GA for CO objectives.

3 UNIVERSAL FUNCTION APPROXIMATION BY FUZZY SYSTEMS

3.1 Literature Synopsis

Mathematical model design of complex real systems is a must for many scientific and engineering tasks. The so-called black-box common approach to modelling uses exclusively numerical system input-output data pairs for the construction of the mathematical model. Grey-box modelling incorporates some expert knowledge into the model structure used for identification of the system. Fuzzy modelling can only be conducted as black-box modelling when all the knowledge of the system is mere input-output data, however when expert knowledge is also available, it should be used – fuzzy grey-box modelling is a preferable choice [70].

Expert knowledge is valuable in the initial stage but system adaptability, automatic fine-tuning of the human provided knowledge in the context of other automatically derived rules is a tough but necessary challenge to cope with a continuously changing system environment. Preserving linguistic value natural meaning and ordering of membership function defining variables imposes numerous hard constraints on the parameters that are used for forming the membership functions associated with a linguistic value. Without constraints the readability and common sense interpretability of the model is lost as linguistic value *low* must precede *medium* which comes before *large* even after automatic fine tuning [76].

There are many applications when the completeness of the model is required; we must ensure that the model output corresponds to the real system output for any real input signal. Forming fuzzy partitions is also important in numerous applications where certain properties of the fuzzy logic system have to be ensured [30]. The necessity of uniformly covering the complete input space and having for every antecedent a rule consequent derived from the provided input-output data establishes another set of constraints on the set of membership function defining parameters. Forming fuzzy-partitions by antecedent membership functions ensures that there cannot be a numerical input within the defined input range that will not result in firing at least one rule consequent of the fuzzy model. Keeping the specific properties of fuzzy-partitions (that will be described in the next chapter) imposes another set of hard constraint on the parameters of membership functions.

This paper will present a novel method for fuzzy identification based on a new simple method for representing all the nonlinear parameters and unconstrained tuning of fuzzy systems that for rule antecedents have Zadeh-type membership functions forming fuzzy partitions. The method is also capable of dynamically defining and changing the fuzzy system complexity by discarding unnecessary membership functions and corresponding rules. The method is validated on well-studied, widely used proven benchmark systems used for measuring the performance of fuzzy modelling.

3.1.1 Fuzzy System Modelling by Zadeh-formed Membership Functions

The identification method to be proposed is applicable to any type of fuzzy logic systems (FLSs) that use Zadeh-formed membership functions (MFs). For example let us consider Takagi-Sugeno-Kang (TSK) type FLSs having *n* inputs and 1 output. These FLSs can be formulated as:

$$f(\boldsymbol{x}) = \sum_{l=1}^{M} \omega_l(\boldsymbol{x}) \cdot y_l(\boldsymbol{x}) \,/\, \sum_{l=1}^{M} \omega_l(\boldsymbol{x}) \tag{22}$$

where: M is the number of Rules, $\boldsymbol{x}$ is the vector of n input variables, y_l is a scalar function of n input variables, and y is most commonly either a constant for Wang type FLSs or a linear function of inputs for the first order TSKs. Thus y_l is defined by one or $(n + 1)$ parameters respectively.

The antecedent, the premise part of a fuzzy rule is:

$$\omega_l(\boldsymbol{x}) = \prod_{i=1}^{n} \mu_{F_l(i)}(x_i) \tag{23}$$

where: $\mu_{Fl(i)}(x_i)$ is the membership function of the i^{th} input variable in the l^{th} rule that defines the linguistic value $F_{l(i)}$.

The linguistic form of the l^{th} rule from the previously described first order TSK FLS is [76]:

$$\text{IF } (x_1 \text{ is } F_{l(1)}) \text{ and } (x_2 \text{ is } F_{l(2)}) \text{ and } (x_\text{n} \text{ is } F_{l(n)}) \text{ THEN } y_l = \sum_{j=1}^{n} c_{l(j)} \cdot x_j + c_{l(0)}, \tag{24}$$

In the focus of this paper are Zadeh-formed MFs. They are the second order polynomial Z-, the S-, and the π-functions (named after their shape) defined respectively as:

$$mfz(x,b_1,b_2) = \begin{cases} 1 & x \le b_1 \\ 1 - 2((x-b_1)/(b_2-b_1))^2 & b_1 < x \le \tfrac{1}{2}(b_2+b_1) \\ 2((b_2-x)/(b_2-b_1))^2 & \tfrac{1}{2}(b_2+b_1) < x \le b_2 \\ 0 & x > b_2 \end{cases}$$

$$mfs(x,b_1,b_2) = 1 - mfz(x,b_1,b_2)$$

$$mf\pi(x,b_1,b_2,b_3,b_4) = \begin{cases} mfs(x,b_1,b_2) & x \le b_2 \\ 1 & b_2 < x \le b_3 \\ mfz(x,b_3,b_4) & x > b_3 \end{cases} \quad , \tag{25}$$

where: $b_1 \le b_2 \le b_3 \le b_4$ are the parameters defining the MFs.

When there is more than one value x such that the degree of membership of x is equal to one, the interval where the $\mu_k(x, b) = 1$ (the interval $[b_2, b_3]$ for $mf\pi$ type μ_k) is called the plateau of the μ_k MF.

When having for example three naturally ordered linguistic values $l \in \{a, b, c\}$ (for example a = '*low*', b = '*medium*', c = '*large*') constraints on b_i parameters to preserve this ordering are:

$$b_{a1} < b_{b1} < b_{c1}$$

$$b_{a2} \le b_{a3} < b_{b2} \le b_{b3} < b_{c2} \le b_{c3} \, . \tag{26}$$

$$b_{a4} < b_{b4} < b_{c4}$$

3.1.2 Fuzzy Partitions

When a linguistic variable can be assigned K different linguistic values, each described by a MF $\mu_k(x, b)$ such that for every input x it holds that $\sum_{k=1}^{K} \mu_k\,(x,\ b) = 1$, the MFs are said to form a fuzzy-partition.

By imposing these restrictions on all linguistic variables of the FLS and additionally assuming that the rule base is complete in the sense that it covers the whole input domain, it immediately follows that the TSK model structure (22) simplifies to [30]:

$$f(\boldsymbol{x}) = \sum_{l=1}^{M} \omega_l(\boldsymbol{x}, \boldsymbol{b}) \cdot y_l(\boldsymbol{x}, \boldsymbol{c}). \tag{27}$$

Automatic fine tuning all the b_i parameters of a TSK FLS that satisfies all of the above constraints is a significant problem. The search space is large and usually deceptive, so global stochastic optimisation processes are needed like GAs. FLSs evolved by GAs are called Genetic Fuzzy Systems (GFSs).

As GAs are not efficient in finding the exact optimum, the application of gradient decent methods is required as post processing the GA results.

By my *Hypothesis II* this paper will propose and present the validity of a novel method that will simplify the b_i parameter optimisation of fuzzy systems like equation (27), while preserving all the required constraints and properties as described above. All linear c_i parameters of equation (24) will be determined by a singular value decomposition (SVD) based robust least squares (LS) method.

3.1.3 Validating Quality of Genetic Fuzzy System Function Approximation

Function identification results on the below listed benchmark systems achieved by the proposed method are compared to the basic nearest neighbourhood clustering method (NNC), the table look-up method (TAB) and the adaptive-network-based fuzzy inference systems method (ANFIS) that are described in [31]. Further on results achieved by various other methods from the literature are also cited for comparison.

3.1.3.1 Chaotic Time Series of Mackey and Glass

The Mackey-Glass chaotic time series is a well-known identification benchmark system. The delay-differential equation used is:

$$\dot{x}(t) = \frac{ax(t-d)}{1+x^c(t-d)} - bx(t) \tag{28}$$

The benchmark set is generated using the parameters a = 0.2, b = 0.1, c = 10, d = 17 and $x(t<0)$ = 0.8. The selection of d = 17 results in a quasi-periodic series with a characteristic period $T_c \approx 50$ [30]. The equation is integrated up to t = 5500, with points from t = 200 to t = 3200 used for training and points from t = 5000 to t = 5500 used for testing as in Figure 16.

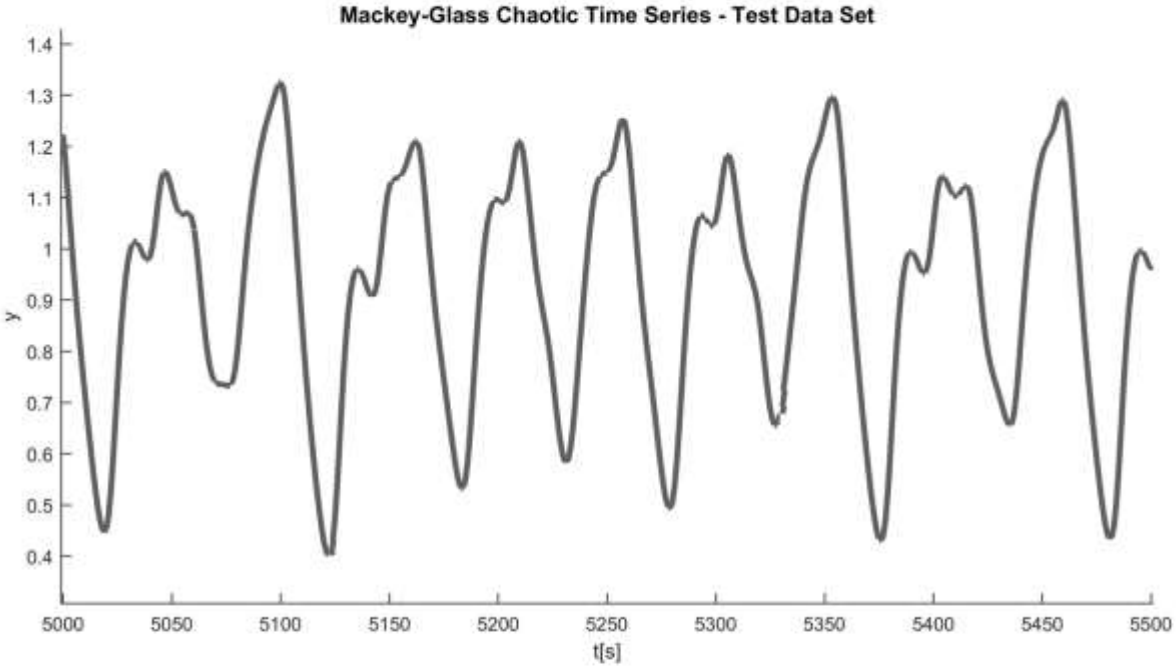


Figure 16.
The Mackey-Glass chaotic time series.

Since the Mackey-Glass time series is chaotic, it is difficult to predict for values of *T* greater than its characteristic period T_c of approximately 50. In the literature a number of different prediction points have been tried: $T \in \{1, 6, 84\}$. For approaches that use as input a time window of past values it is common to use the four delays *t, t-6, t-12* and *t-18* [36].

3.1.3.2 Gas Furnace Model of Box and Jenkins

The Gas Furnace model is the so-called J data series from the well-known Box-Jenkins identification benchmark data set [6]. Using a time-discrete formulation, the dynamics of the system is represented by the relationship:

$$y(t+1) = f(u(t-2), y(t)), \tag{29}$$

where *y(t)* and the corresponding *u(t)* values are provided as 296 input-output data pairs as a benchmark data set [30] as in Figure 17.

The complete, correct Box-Jenkins gas furnace system input-output benchmark data set is listed in Appendix I. Because of the very limited sample data set size the complete set is used for training and testing as well.

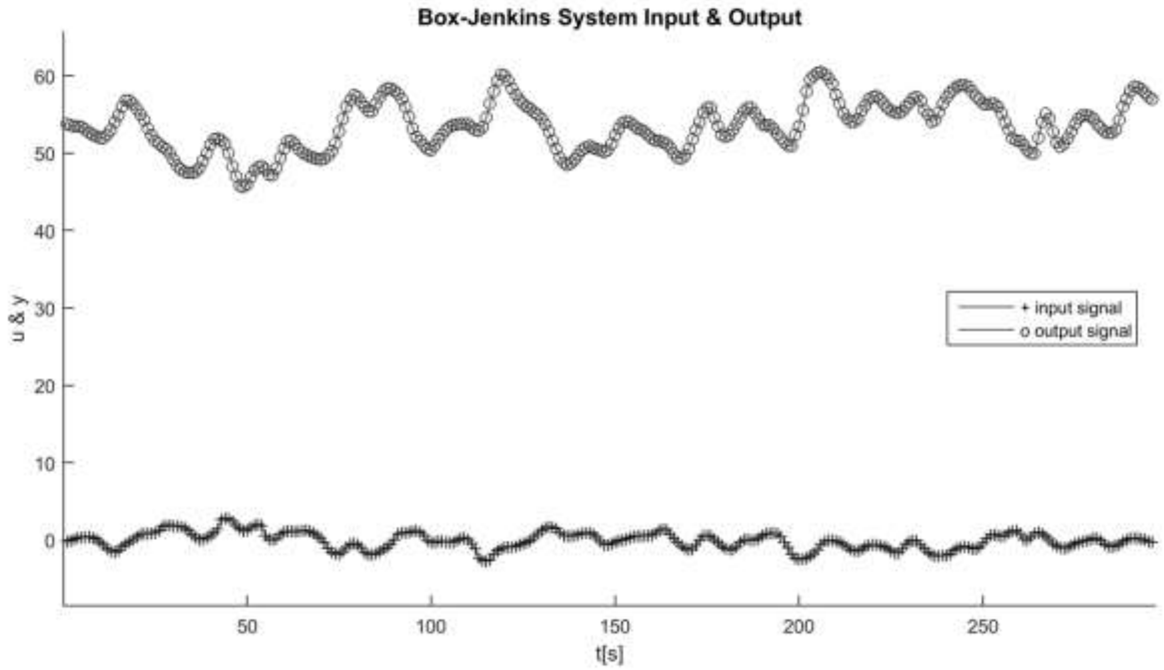


Figure 17.
The furnace model of Box and Jenkins.

3.1.3.3 Generalised Rastrigin Function

The generalised Rastrigin function is a strongly multi-modal function of two inputs:

$$y(x, z) = x^2 + z^2 - \cos(18x) - \cos(18z). \quad (30)$$

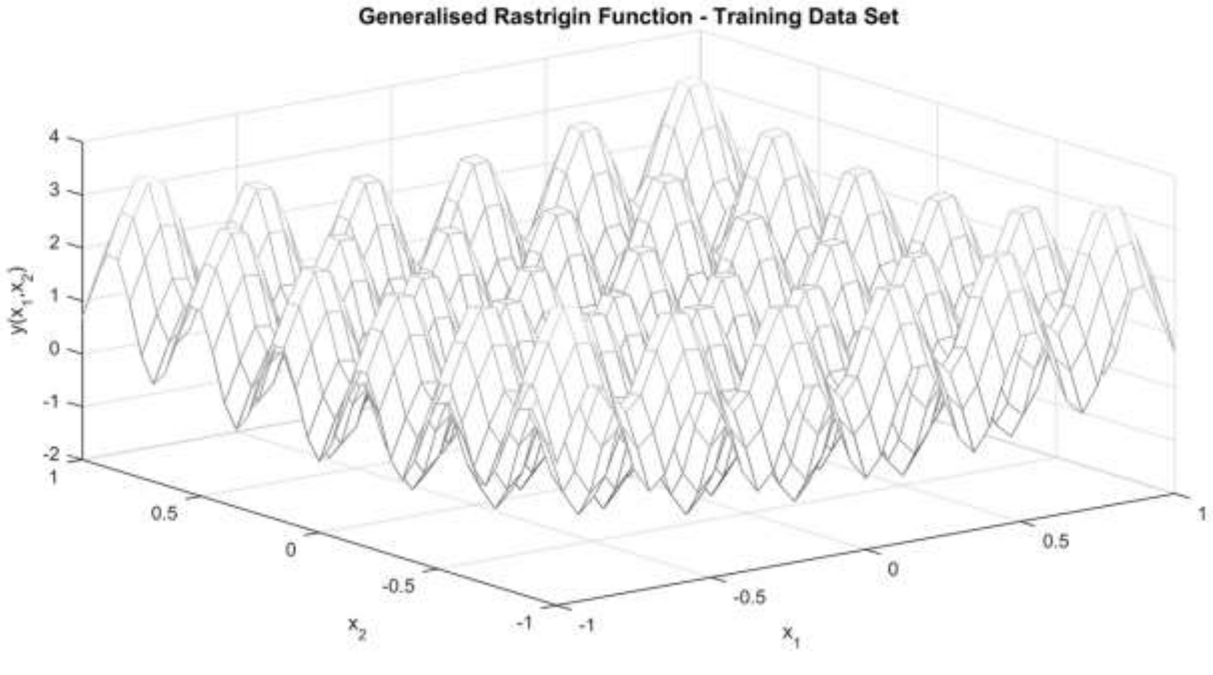


Figure 18.
The generalised Rastrigin function training data set.

A set of 1681 (x, z) input vectors have been generated by taking 41 uniformly distributed values from the closed interval [-1, 1] as in Figure 18. The test series is a randomly formed set of 168 input vectors from the same range [30].

3.2 New Scientific Achievements

A new method for representing Zadeh-type fuzzy partitions for efficient unconstrained function identification is introduced in my publication [s3]. In my second thesis for the nonlinear membership parameters of fuzzy partitions I introduce a simple, minimal number of parameters, such that standard unconstrained tuning is made possible without a loss of linguistic meaning, as in MF ordering.

3.2.1 New Minimalistic Parametrisation of Zadeh-type Fuzzy Partitions for Function Identification by Unconstrained Tuning

As described in [s3] the nature of Zadeh-formed MFs is such that simply making equal the last two parameters of the preceding MF to the first two parameters of the succeeding MF we easily form fuzzy partitions. This way a fuzzy partition of K MFs is defined by *2(K-1)+1*parameters. Let our input space be normalised ($x_{\min}=0$ and $x_{\max}=1$). If we do not want to allow any plateaux, parameter b_2 must be equal to b_3 in (25), thus the number of parameters for a fuzzy partition consisting of K pieces of Zadeh-type MFs is further reduced to the minimum of *(K-1)*.

If we take into consideration all of the constraints (26) we end up with a series of strictly ordered parameters:

$$b_1<b_2<\ldots<b_{K-1}. \tag{31}$$

Let us add two more constraints, which are possible as the input space is normalised:

$$0<b_1 \text{ and } b_{K-1}<1. \tag{32}$$

Let us define the first MF to be:

$$mfz(x,0,b_1), \tag{33}$$

and the K^{th}, the last one, to be:

$$mfs(x,b_{K-1},1). \tag{34}$$

Let us define the general intermediate k^{th} MF to be:

$$mf\pi(x,b_{k-1},b_k,b_k,b_{k+1}) \tag{35}$$

for $k = 2, \ldots,(K-1)$. This way the ordered series of *(K-1)* parameters (31) together with border conditions (32) are the minimal number of parameters to define a fuzzy-partition of Zadeh-formed MFs, which can represent any such partition.

This minimal number of nonlinear parameters is a very important issue for optimisation as over parameterised systems are hard to optimise. The only problem now remains to be that when we are tuning these interdependent b_k nonlinear parameters of a FLS having an n dimensional input space, we must comply with $\sum_{i=1}^{n} K_i$ pieces of hard constraints. Although there are a number of constrained optimisation methods it is obvious that an unconstrained optimisation method would be more efficient. My proposal is to represent the b_k parameters in a different manner.

THESIS II - DEFINITION:

For **a minimal independent parametrisation of Zadeh-type MF based fuzzy partitions, that can be optimised without any constraints**, let us consider *K* pieces of rational, positive or zero parameters as:

$$a_\kappa \in R_0^+, \kappa = 1, ..., K \ . \tag{36}$$

Let us form the b_k nonlinear parameters of Zadeh-type MFs forming fuzzy-partitions for a FLS as:

$$b_k = \sum_{j=1}^{k} a_j \Big/ \sum_{\kappa=1}^{K} a_\kappa \ , \tag{37}$$

then for every $k = 1, \ldots, K$ all the constraints (31) and (32) are automatically fulfilled for every b_k from (37) without any further restrictions on a_κ.

Notice that for cases when an $a_\kappa = 0$ we obtain $b_k = 0$, thus the fuzzy partition is reduced by one MF. In cases when there is only a single $a_\kappa > 0$ the fuzzy system degrades to a simple linear equation. And finally in the case where all $a_\kappa = 0$ the fuzzy partition degrades to a constant, and the fuzzy system becomes indifferent to that input channel; it becomes independent of the corresponding input variable.

These a_κ parameters can be fine-tuned with any gradient descent based method; also the Jacobian of the FLS can be calculated with regards to a_κ for advanced nonlinear least squares data fitting methods.

Further, an ANFIS like optimisation of all FLS parameters (linear least square (LS) for consequent and gradient based optimisation for antecedent parameters) can be directly applied to tune all the a_κ parameters of the FLS. Only a little bit of extra preliminary symbolic derivation is required – we must in addition to the usual derivatives evaluate $\partial b_k / \partial a_\kappa$ and chain it to $\partial e / \partial b_k$ in the well-known Δb_k equation (38) to acquire the amount of modification of the a_k parameter from the identification error *e* [31].

$$\Delta b_k = -\partial e / \partial b_k \, / \sqrt{\sum_k (\partial e / \partial b_k)^2} \tag{38}$$

3.2.2 Implementation of the New Genetic Fuzzy System Parametrisation

The proposed fuzzy-partition representation has been incorporated into a multi-objective genetic algorithm. One chromosome consists of the number of MFs for every input and the corresponding parameters as in equation (36) for every possible MF partition of each input. The population size is then times the number of parameters, for parameter coding I am using binary Grey-coded chromosomes. Chromosomes are subjects to a DONGA (N.DO) multi-objective genetic algorithm from Table I (Table IV), described in chapter 2.2.2.

The objective functions I have used are:

- *maxE*: the maximum absolute error of the identification,
- *MSE*: the mean squared error of the identification,
- *RuleN*: the number of used fuzzy rules for the identification divided by the maximum possible number of rules for the selected design

- *RankW*: that is calculated from the matrix rank of $\boldsymbol{W}(\boldsymbol{x},\boldsymbol{b})$**,** the FLS identification linear equation of consequents $\boldsymbol{c}$, based on equation (27) as: $f(\boldsymbol{x}) = \sum_{k=1}^{M} \omega_k(\boldsymbol{x},\boldsymbol{b_k}) \cdot y_k(\boldsymbol{x},\boldsymbol{c_k}) = \sum_{k=1}^{M} \omega_k(\boldsymbol{x},\boldsymbol{b_k}) \cdot \left(\sum_{j=1}^{n} c_{k,j} \cdot x_j + c_{k,0}\right) = \boldsymbol{W}(\boldsymbol{x},\boldsymbol{b}) \cdot \boldsymbol{c}$, which is normed to the theoretically maximal possible rank of the same FLS structure.

Based on the transformation of equation (27) to the $f(x) = W(x,b) \cdot c$ format, we split the identification into 2 problems. One nonlinear problem of finding the optimal $W(x,b)$, which is defined as a function of system input x and the nonlinear MF b_k parameters of equation (25) – my Thesis II is the proposed simple solution to this problem by transforming a_κ to b_k, after which any nonlinear unconstrained optimisation can be applied to a_κ parameters. The second part of the problem is a simple linear equation issue, which is best solved by the SVD decomposition method as: $c = VS^{-1}U^T \cdot W(x,a_\kappa)$ for the SVD decomposition of $W(x,a_\kappa) = USV^T$.

For the applied GA chromosomes are evaluated through the following eight steps:

1. K_i the number of MFs for each of the *n* inputs is decoded from the chromosome.

2. a_κ parameters are decoded from the chromosome.

3. All required parameters of Zadeh-formed MFs of equation (25) that form fuzzy partitions are calculated as proposed in (37).

4. All possible antecedents are formed from the MFs and their numerical values are evaluated as in equation (23).

5. Corresponding fuzzy rule consequent parts $y_l(x)$ for all numerical inputs are evaluated by a singular value decomposition (SVD) based LS method from equation (27).

6. Parameters from Step 2. are further optimised by a simple gradient decent based method for no more than $\sum_{i=1}^{n} K_i$ steps, where *n* is the dimension of the input space and K_i is the number of used MFs. For every iteration Step 4. is repeated.

7. The resultant fuzzy system is evaluated.

8. The *maxE, MSE, RuleN/maxRuleN, RankW/maxRankW* of the identification is calculated, where *maxRuleN* = $\prod_{i=1}^{n} K_i$, and *maxRankW = maxRuleN·(n+1).*

To increase the efficiency of the GA, the chromosome defining a fuzzy system is updated with its optimised MF parameter values after each evaluation.

3.2.3 Results of the New Function Identification with the New Genetic Fuzzy System Parametrisation

Results of benchmark system identifications are presented in Tables VI-X. In the first column of every table the identification method is named. The second column points out the complexity of the identified model – the number of rules for FLSs. Identification mean square error (MSE) values of test samples are in the third column. The fourth column presents the processing effort, the time duration of completing a process (for a current mid-level PC one tick corresponds to one second).

My proposed identification based on Thesis II can be used in multiple schemas. It can be used for a fast and efficient direct fuzzy identification, where uniform fuzzy partitions ($a_\kappa = 1, \forall\kappa$) are used, and only the linear parameters $\boldsymbol{c_k}$ of the FLS equation (27) are calculated by the SVD method – such results are presented in rows named "LinLSzFLS". These uniform fuzzy partitions can be further subject to gradient descent based nonlinear optimisation, where the uniform a_κ parameters are locally optimised – such results are presented in rows named "NonlinLSzFLS". Further on the non-linear a_κ parameters of fuzzy partitions can be pre-optimised by GA – such results are presented in rows named "GAzFLS"; results of GA optimisation are the average values of 10 runs.

Each method can be used for different number of MFs – the number of MFs is marked in brackets after the name of the identification method. Notice that not all possible MF combinations have to be used for fuzzy rules of the system – there can be incomplete systems, where certain rules are omitted, thus the number of rules can less than the product of MFs. Notice that such incomplete rule bases can be acceptable for simple data fitting, but in case of control engineering this means incompletes of the training data, thus there can be no uniform precision guaranty for the identified system model – as the model response is not defined for untrained input space areas. The nearest neighbourhood clustering, the table lookup and the ANFIS method I took from the standard Matlab2015 Fuzzy toolbox implementation; the other reference numbers are from literature.

Method	**Number of rules**	**MSE of test data**	**Effort [tics]**
LinLSzFLS (3 MFs)	81	8.6503e-8	4.5569e-1
NonlinLSzFLS (3 MFs)	81	5.4159e-8	2.8114e+1
GAzFLS (3 MFs) Average of 10 runs	76	2.8242e-8	8.8100e+2
GAzFLS (4 MFs) Best of 10 runs	152	2.6197e-7	3.758e+3
NNC	67	1.3464e-3	8.1477
TAB	301	9.6185e-5	6.2069
ANFIS	199	2.2074e-8	5.508e+3
Linear Predictor [26]	na	3.27e-2	-
MLP [26]	25	1.02e-2	-

Table VI. Mackey-Glass chaotic time series, equation (28) – predicting y(t+1)

Method	**Number of rules**	**MSE of test data**	**Effort [tics]**
LinLSzFLS (3 MFs)	81	1.4304e-6	5.1010e-1
NonlinLSzFLS (3 MFs)	81	1.1090e-6	2.8381e+1
GAzFLS (3 MFs) Average of 10 runs	74	5.1280e-7	1.2020e+3
GAzFLS (4 MFs) Best of 10 runs	174	5.3702e-7	3.822e+3
NNC	67	2.3841e-3	8.1092
TAB	301	3.155.e-4	6.3051
ANFIS	214	2.1694e-7	6.207e+3
Linear Predictor [26]	-	7.17e-1	-
MLP [26]	25	5.11e-2	-
DCS-LLM [26]	200x200	5.50e-3	-
BGALR [26]	150	2.37e-1	-
SOM [26]	35x35	4.80e-3	-

Table VII. Mackey-Glass chaotic time series, equation (28) – predicting y(t+6).

Method	Number of rules	MSE of test data	Effort [tics]
LinLSzFLS (3 MFs)	81	1.8246e-4	4.5911e-1
NonlinLSzFLS (3 MFs)	81	1.3383e-4	5.6149e+1
GAzFLS (3 MFs) Average of 10 runs	73	7.9297e-5	1.2130e+3
GAzFLS (4 MFs) Best of 10 runs	149	1.1172e-4	2.767e+3
NNC	67	1.1107e-2	8.4509
TAB	301	3.289e-3	6.3102
ANFIS	241	1.3625e-6	7.745e+3
Linear Predictor [26]	-	1.5035	-
MLP [26]	25	4.60e-1	-
DCS-LLM [26]	200x200	3.00e-2	-
BGALR [26]	150	2.64e-1	-
SOM [26]	35x35	2.20e-2	-

Table VIII. Mackey-Glass chaotic time series, equation (28) – predicting y(t+84).

Method	Number of rules	MSE of test data	Effort [tics]
LinLSzFLS (3 MFs)	9	1.3212e-1	1.1693e-1
NonlinLSzFLS (3 MFs)	9	1.2818e-1	1.0157
GAzFLS (7 MFs) Average of 10 runs	37	8.5986e-2	1.0e+2
NNC	36	7.6222e-1	6.0726e-1
TAB	52	2.5163e-1	1.5032e-1
ANFIS	38	4.5166e-1	6.0334
Box&Jenkins [30]	na	7.10e-1	-
Yoshinari, Pedrycz&Hirota [30]	na	2.99e-1	-
Lee, Hwang&Shih [30]	na	2.11e-1	-
Wang&Langri [30]	na	6.6e-2	-
Nakoula [30]	na	1.75e-1	-

Table IX. The furnace model of Box and Jenkins, equation (29) – system modelling.

Method	Number of rules	MSE of test data	Effort [tics]
LinLSzFLS (9 MFs)	81	5.964e-4	2.9100e-1
NonlinLSzFLS (9 MFs)	80	7.469e-4	2.0366e+1
GAzFLS (9 MFs) Average of 10 runs	81	8.5168e-4	1.006e+2
NNC	99	4.7453e-1	3.8081
TAB	100	5.7492e-1	9.2975e-1
ANFIS	464	8.6129e-6	6.592e+3
GFIM [30]	253	2.102e-1	-
GFIM [30]	73	1.1183	-

Table X. The generalised Rastrigin function, equation (30) – function approximation.

The ANFIS method produces exceptionally precise approximations with large number of rules, and requires a significant computation effort (for a current mid-level i5 CPU based PC 7200 ticks corresponds to 2 hours of computation). The drawback of this method is that each consecutive fine tuning also requires similar significant computational effort, thus real-time adaptability of ANFIS based systems can not directly be performed.

The newly proposed 'LinLSzFLS' method provides exceptionally fast results, while the identification precision is already very good, second only to ANFIS and my other two proposed fuzzy identification methods, the 'NonlinLSzFLS' and the 'GAzFLS', albeit all these more precise methods are significantly more time consuming, up to 10 000 times slower.

The performance of getting good results within half a second on a current mid-level PC the newly proposed 'LinLSzFLS' method provides a feasible update strategy even for real time applications.

The newly proposed 'NonlinLSzFLS' method is a gradient descent extension of the 'LinLSzFLS', as such it is some 100 times slower, while it results in less than 50% improvements.

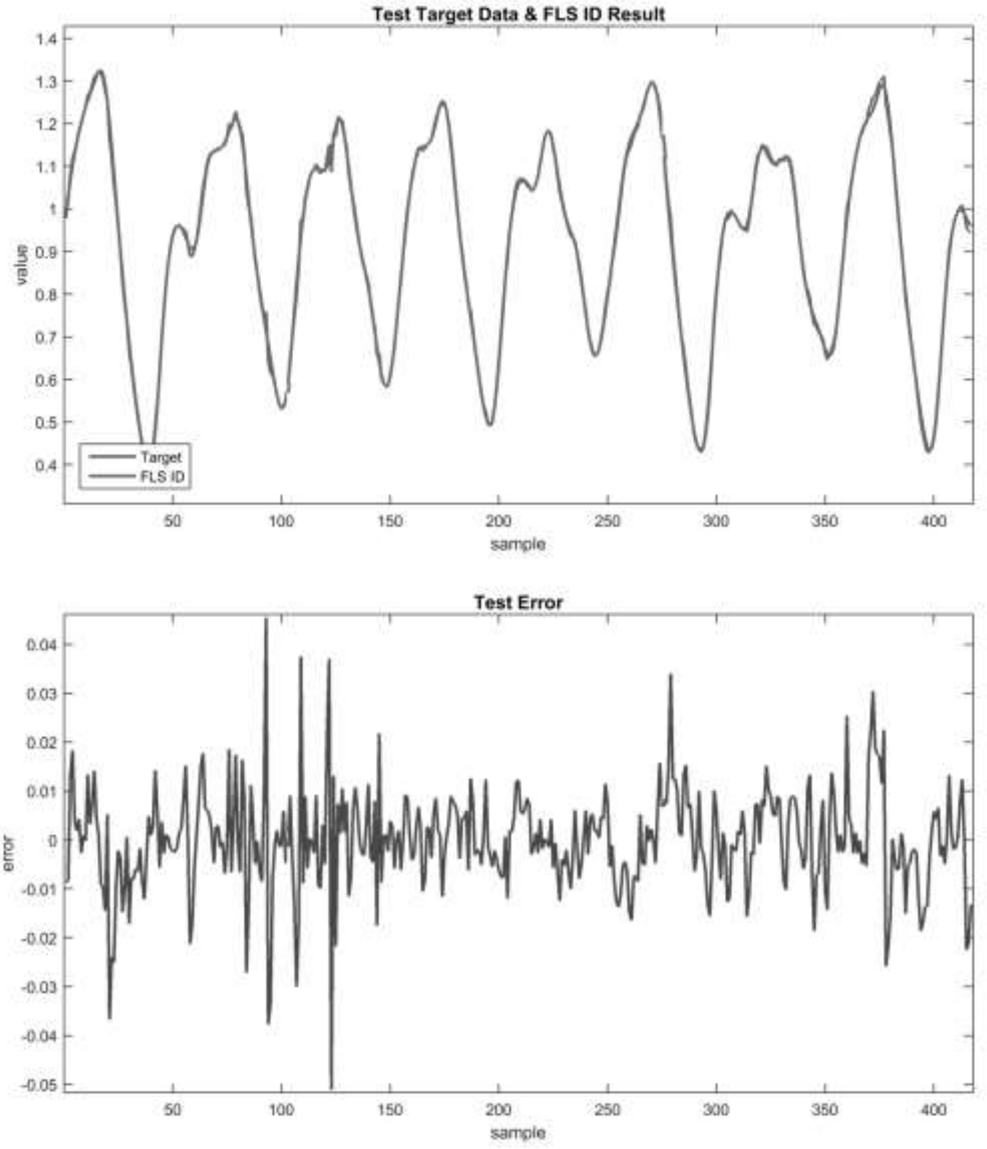


Figure 19.
Test error with test target data and FLS identification result of my proposed method for Mackey-Glass chaotic time series – predicting y(t+84).

Results with the proposed ‘GAzFLS’ method are in average achieved within 15 GA generations. Rules with a pure zero consequent part were discarded.

The ANFIS and the ’GAzFLS’ are in by far the most time consuming methods, suitable for offline design methods only, but in return they can find an order of magnitude more precise results.

The precision of my proposed fuzzy identification method is superb, even the *y(t+84)* prediction of the Mackey-Glass chaotic time series as in Figure 19 is extraordinary, while the number of rules is small.

By these results my proposal for a successful fuzzy identification strategy is to take the ‘GAzFLS’ method as an off-line preliminary identification method, apply the results while keeping a continuous real-time ‘LinLSzFLS’ update mechanism in place for continuous fine tuning with fresh measurements, thus ensuring adaptability of the system.

By this analysis I conclude that my Thesis II is proven valid.

4 GENETIC FUZZY MODELLING OF COMPLEX SYSTEM DYNAMICS

There are two popular, well studied complex nonlinear systems for which we will examine the applicable soft computing system identification, complex dynamics modelling tools – a general robot manipulator dynamics model and its special case the multi-rotor flight dynamics model.

4.1 Literature Synopsis

4.1.1 Modelling Robot Manipulator Dynamics

In my research the modelling of robot manipulators (RMs) dynamics, mapping the position, velocity and acceleration of joints to forces and torques exerted to the structure is based on the Lagrange formulation, which ensures the appropriate structure of the dynamic model that is commonly used in control algorithms.

RMs are known to be highly nonlinear multi-input multi-output systems. To preserve the known structure of the Lagrange formulation, common to all system equations of RMs and other dynamic systems such as navigation dynamics of missiles and aeroplanes, I have chosen the grey-box modelling approach.

Forces exerted to joints of the RM are the sum of four components modelling consequently the torque resulting from the inertia (*H*), the Coriolis effects and centrifugal forces (*C*), the gravity forces (*g*) and the viscose friction (*f*). Individual knowledge of all these components is important for precise, model based robot control algorithms, yet it is impossible to directly, explicitly measure these components in any general real life system. When designing a grey-box model advantage can be taken of other commonly known facts of robotics like H and g are nonlinear functions of joint positions and the driving torque τ is linear in the joint accelerations. The centrifugal and Coriolis effects are quadratic in the joint velocities and nonlinear in the joint positions and *f* is linear in joint velocity [80].

The dynamic model identification method uses the measured resultant torque and joint variables along suitably chosen paths for every joint. The application of Lagrange dynamic equations for a robot manipulator in the joint space formulates the resultant torque τ_i acting on the *i*th joint for all the *p* joints of the RM as a function of following vectors:

$$\sum_{j=1}^{p}\left(\boldsymbol{D}_{ij}(\boldsymbol{q})\cdot\ddot{\boldsymbol{q}}_j\right)+\sum_{j=1}^{p}\sum_{k=1}^{p}\left(\dot{\boldsymbol{q}}_{\boldsymbol{j}}\cdot\boldsymbol{D}_{ijk}(\boldsymbol{q})\cdot\dot{\boldsymbol{q}}_k\right)+\boldsymbol{D}_i(\boldsymbol{q})+f_i=\tau_i \tag{39}$$

where: $\boldsymbol{q}$ is the vector of joint positions; $\dot{\boldsymbol{q}}$ are joint velocities; $\ddot{\boldsymbol{q}}$ are joint accelerations; $\sum_{j=1}^{p}\left(\boldsymbol{D}_{ij}(\boldsymbol{q})\cdot\ddot{\boldsymbol{q}}_j\right)$ is commonly referred to as $\boldsymbol{H}\cdot\ddot{\boldsymbol{q}}$ or $\mathbb{J}\cdot\ddot{\boldsymbol{q}}$ the inertia matrix component of the torque; $\sum_{j=1}^{p}\sum_{k=1}^{p}\left(\dot{\boldsymbol{q}}_{\boldsymbol{j}}\cdot\boldsymbol{D}_{ijk}(\boldsymbol{q})\cdot\dot{\boldsymbol{q}}_k\right)$ is commonly referred to as $\boldsymbol{C}\cdot\dot{\boldsymbol{q}}$ or $\mathbb{C}\cdot\dot{\boldsymbol{q}}$, describing the centrifugal forces and the Coriolis effects; $\boldsymbol{D}_i(\boldsymbol{q})$ is commonly referred to as $\boldsymbol{g}$ the gravitational force component; f_i stands for the viscous friction; and τ_i is the resultant torque acting on the ith joint – identities in the common robotics notation are:

$$H_{ij}=D_{ij}(\boldsymbol{q}),C_{ik}=\sum_{j=1}^{p}\dot{q}_j\cdot D_{ijk}(\boldsymbol{q})\,,g_i=D_i(\boldsymbol{q}),f_i=const_i, \tag{40}$$

where: D_{ij}, D_{ijk}, D_i are in general, highly nonlinear scalar functions of $\boldsymbol{q}$, the joint position vector. They may contain sin($*$) and cos($*$) functions of joint positions and/or of their products and sums defined by the geometry of the RM.

There are well known general relations that can be used to reduce the number of unknown elements, like D_{ijk} are the Christoffel symbols of D_{ij} [80]:

$$D_{ijk} = \frac{1}{2}\left(\frac{\partial D_{ij}}{\partial q_k} + \frac{\partial D_{ik}}{\partial q_j} - \frac{\partial D_{jk}}{\partial q_i}\right), D_{ijk} = D_{ikj}, D_{kij} = -D_{jik}, D_{kjk} = 0, \forall i, k \geq j, \quad (41)$$

It should be well noted that direct measurement of any single sub-component from equation (40) is not possible. The only information on the output of the system joint is the resultant torque (39). The identification of all nonlinear functions under these terms is a considerable problem.

By my *Hypothesis III.a* this paper will propose and present the validity of a new method that will identify the RM dynamics through finding the D_{ij} nonlinear functions of equation (39) as TSK FLSs, while calculating D_{ijk} nonlinear functions as in equation (41). All linear parameters of the system will be determined by SVD based robust LS method. Nonlinear parameters will be evolved by multi-objective GA and fine-tuned by a gradient descent method.

4.1.2 Modelling Multi-rotor Flight Dynamics

The complete dynamics of an aircraft, taking into account aero-elastic effects, flexibility of wings, internal dynamics of engines, and the whole set of changing environmental variables is quite complex and somewhat unmanageable for the purpose of autonomous control engineering [10]. This paper deals with the flight dynamics of multi-rotors.

Multi-rotor UAV manoeuvres are controlled by varying angular speeds of its propellers. Each rotor blade produces a thrust and a torque, whose combination generates the main trust, the yaw torque, the pitch torque, and the roll torque acting on the multi-rotor. Motors produce a force proportional to the square of the angular speed and the angular acceleration of the rotor; the acceleration term is commonly neglected as the speed transients are short thus exerting no significant effects. Motors of a multi-rotor can only turn in a fixed direction, so the produced force can be always presumed positive.

Motors are set up so that opposites form pairs rotating in the same direction (clockwise/counter-clockwise), while their neighbouring motors are rotating in the opposite direction (counter-clockwise/clockwise). This arrangement is chosen so that gyroscopic effects and aerodynamic torques are cancelled in trimmed flight. The main trust is the sum of individual trusts of each motor. The pitch torque is a function of difference in forces produced on one pair of motors, while the roll torque is a function of difference in forces produced on other pair of motors. The yaw torque is sum off all motor reaction torques due to shaft acceleration and blades drag. The motor torque is opposed by a general aerodynamic drag.

For a full navigation dynamic model of a multi-rotor system both (i) the centre of mass position vector of $\boldsymbol{\xi} = (x, y, z)$ in fixed frame coordinates and (ii) the orientation Euler angles: roll, pitch, yaw angles *(ϕ,θ,ψ)* around body axes *X*, *Y*, *Z* are considered. Using the Euler-Lagrange approach it can be shown how the translational forces $\boldsymbol{F}_{\xi}$, applied to the rotorcraft due to main trust, can be fully decoupled from the yaw, pitch and roll torques $\boldsymbol{\tau}$ as defined by equations (24-25).

$$m \cdot (\ddot{\boldsymbol{\xi}} + g \cdot [0\ 0\ 1]^T) = \boldsymbol{F}_\xi \tag{42}$$

where: $\ddot{\boldsymbol{\xi}}$ is the second time derivative (acceleration) of the E-frame coordinates of the rotorcraft centre of mass $\boldsymbol{\xi} = (x, y, z)$, m is the multi-rotor mass; g is the gravitational constant, which is acting only along the third axis $\vec{z}$; $\boldsymbol{F}_\xi$ is the vector of translational forces.

$$\mathbb{J}(\boldsymbol{q}) \cdot \ddot{\boldsymbol{q}} + \mathbb{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \cdot \dot{\boldsymbol{q}} = \boldsymbol{\tau} \tag{43}$$

where: $\mathbb{J}$ is a 3x3 matrix, called the inertia matrix, $\mathbb{C}$ is also a 3x3 matrix that refers to Coriolis, gyroscopic and centrifugal terms, $\boldsymbol{q} = [\phi, \theta, \psi]$ is the state vector of Euler angles, its time derivatives are $d\boldsymbol{q}/dt = \dot{\boldsymbol{q}} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]$ and $d\dot{\boldsymbol{q}}/dt = \ddot{\boldsymbol{q}} = [\ddot{\phi}, \ddot{\theta}, \ddot{\psi}]$.

For the scope of this research we shall address only equation (43) as that is the complex nonlinear challenging part of the multi-rotor flight dynamics model to be identified.

Equation (43) can be analysed as three resultant torques τ_i acting along the $[\phi, \theta, \psi]$ axes for $i,j,k \in (\phi, \theta, \psi)$ as:

$$\sum_j (D_{ij}\,(\boldsymbol{q}) \cdot \ddot{q}_j) + \sum_j \sum_k \big(\dot{q}_j \cdot D_{ijk}(\boldsymbol{q}) \cdot \dot{q}_k\big) = \tau_i \tag{44}$$

The similarity with RM dynamics equation (39) is obvious, the first component of equation (44) is the inertia matrix part expansion, the second is the Coriolis matrix term expansion, whose components are highly nonlinear functions containing $sin(\boldsymbol{q})$ and $cos(\boldsymbol{q})$ components, and also their products and sums defined by the rigid body system geometry as described in [67].

There are general relations that can be used for reducing the number of unknown inertia and Coriolis components: $\mathbb{J}$ is symmetric and $\mathbb{C}$ is defined by Christoffel symbols of $\mathbb{J}$:

$$D_{ijk} = (\partial D_{ij}/\partial q_k + \partial D_{ik}/\partial q_j - \partial D_{jk}/\partial q_i)/2 \tag{45}$$

These properties result in further inherent relations as:

$$D_{ij} = D_{ji}, D_{ijk} = D_{ikj}, D_{kij} = -D_{jik}, D_{kjk} = 0,\ \forall i, k \geq j \tag{46}$$

It should be noted that direct measurement of any single D_{ijk} or D_{ij} component of equation (44) is not possible. Measurable data vector pairs are $(\ddot{\boldsymbol{q}}, \boldsymbol{\tau})$ angular accelerations as system input and resultant torques proportional to rotation speed of motors as system output.

Determining all D_{ijk} and D_{ij} nonlinear functions is a considerable grey-box identification problem, but when achieved the model is usable in efficient robust and precise model based control implementations, as this model preserves all $\ddot{\boldsymbol{q}}, \dot{\boldsymbol{q}}$ state variables in an explicit form.

By my *Hypothesis III.b* this paper will propose and present the validity of a new method that will identify the multi-rotor flight dynamics equation (44) D_{ij} components by specially constructed continuous and periodic TSK FLSs, while calculating D_{ijk} nonlinear functions as in equation (45). For modelling multi-rotor flight dynamics I propose to extend the definition of TSK FLSs in a way that they become periodic and of continuous output, even for the attitude Euler angle system inputs 0-2π transitions.

4.1.3 Validating Quality of Complex Nonlinear Dynamic System Genetic Fuzzy System Modelling

4.1.3.1 SCARA Robot Manipulator as Modelling Test System

The proposed fuzzy system identification method is tested for a SCARA type RM described in chapter 4.1.1. The training data set is reduced to 179 points [s12] as proposed in my Thesis V. The input space is normalised to the unit hyper-cube. The fuzzy-partition representation has been incorporated into a multi-objective hybrid genetic algorithm [s4] as proposed im my Thesis I.

One chromosome consists of only four a_k integer Zadeh-type MF parameters as proposed im my Thesis II. One parameter defines a complete fuzzy partition of three MFs. There are only two D_{ij}s (D_{11} and D_{12}) that are nonlinear functions of only two inputs q_2 and q_4 [s11]. These four a_k parameters are all that is required to model the nonlinearity of a SCARA RM. The remaining twelve linear parameters of the RM and the two times twenty seven linear parameters of the two TSK FLSs having two inputs, nine rules each is determined by the LS method.

4.1.3.2 Quadrotor Unmanned Aerial Vehicles as Modelling Test System

The proposed continuous periodic fuzzy system identification method is tested for a multi-rotor system simulation from [35] as described in chapter 1.1 and 4.1.2 with parameter values listed in Table XI. The identification method optimizes b_k parameters of the continuous periodic FLS (cpFLS) based multi-rotor model in a manner that the torque output of the fuzzy model best fits the multi-rotor system simulation torque output along a reduced data set trajectory. The genetic fuzzy system identification method is based on my Thesis I and II.

The training data set is collected from a simulation along a trajectory with defined sinusoid pop for (x,y,z) and ψ defined so that position changes simultaneously along a main diagonal of a cube, while performing a full circle rotation in yaw motion. The used trajectory generation method is based on my Thesis VI. Roll ϕ and pitch θ is calculated by equation (54). For the cpFLSs input variables ϕ and θ are converted to $[0, 2\pi)$ by eliminating unnecessary $2k\pi$ extensions, then transformed by the “seesaw” function of equation (47) and finally normalized to the $[0, 1]$ closed interval for cpFLS inputs. The training data set is reduced by my Thesis V.

parameter	**value**	**unit**
gravity constant, g	9.81	m/s2
mass, m	6	kg
torque lever, l	0.3	m
trust factor, k	121.5e-6	
drag factor, b	2.7e-6	
body inertia along axes X, I_{XX}	0.6	kgm2
body inertia along axes Y, I_{YY}	0.6	kgm2
body inertia along axes Z, I_{ZZ}	1.2	kgm2
simulation time, T	55	s

Table XI. Parameters of quad-rotor system dynamics.

4.2 New Scientific Achievements

4.2.1 New Genetic Fuzzy System Grey-box Modelling of Complex Dynamics Systems

As described in [s4], [s5], [s10], [s11] my proposed identification method for a general Robot Manipulator (RM) Dynamics equation (39) identification is to use Zadeh-formed membership functions (MFs) as in equation (25) for antecedents as in equation (23) in a Takagi-Sugeno-Kang (TSK) type FLS having n inputs and 1 output as defined in (27). MF nonlinear parameters are represented as in equation (37). Centrifugal and Coriolis components are calculated from the Inertia component as in equation (41).

This chapter relays heavily on many complex equations described in chapters 3 and 4.1 – I have repeated here the bare equations to support a brief background overview for my Thesis III.a:

$$\sum_{j=1}^{p}\left(\boldsymbol{D}_{ij}(\boldsymbol{q})\cdot\ddot{\boldsymbol{q}}_j\right)+\sum_{j=1}^{p}\sum_{k=1}^{p}\left(\dot{\boldsymbol{q}}_{\boldsymbol{j}}\cdot\boldsymbol{D}_{ijk}(\boldsymbol{q})\cdot\dot{\boldsymbol{q}}_k\right)+\boldsymbol{D}_i(\boldsymbol{q})+f_i=\tau_i \qquad (39)$$

$$f(\boldsymbol{x})=\sum_{l=1}^{M}\omega_l(\boldsymbol{x})\cdot y_l(\boldsymbol{x}). \qquad (27)$$

$$y_l=\sum_{j=1}^{n}c_{l(j)}\cdot x_j+c_{l(0)} \qquad (24)$$

$$\omega_l(\boldsymbol{x})=\prod_{i=1}^{n}\mu_{F_l(i)}(x_i) \qquad (23)$$

where each MF $\mu_k(x,\boldsymbol{b})$ is chosen from equations (25) in such a manner that for every input *x* it holds that $\sum_{k=1}^{K}\mu_k\,(x,\ b)=1$, the MFs are said to form a fuzzy-partition. A common partition scheme is to start with *mfz*, and finish with *mfs*, while having arbitrary number of $mf\pi$s in-between.

$$mfz(x,b_1,b_2)=\begin{cases}1 & x\le b_1\\ 1-2((x-b_1)/(b_2-b_1))^2 & b_1<x\le \tfrac{1}{2}(b_2+b_1)\\ 2((b_2-x)/(b_2-b_1))^2 & \tfrac{1}{2}(b_2+b_1)<x\le b_2\\ 0 & x>b_2\end{cases}$$

$$mfs(x,b_1,b_2)=1-mfz(x,b_1,b_2)$$

$$mf\pi(x,b_1,b_2,b_3,b_4)=\begin{cases}mfs(x,b_1,b_2) & x\le b_2\\ 1 & b_2<x\le b_3\\ mfz(x,b_3,b_4) & x>b_3\end{cases}, \qquad (25)$$

$$b_k=\sum_{j=1}^{k}a_j\Big/\sum_{\kappa=1}^{K}a_\kappa\,, \qquad (37)$$

$$D_{ijk}=\frac{1}{2}\left(\frac{\partial D_{ij}}{\partial q_k}+\frac{\partial D_{ik}}{\partial q_j}-\frac{\partial D_{jk}}{\partial q_i}\right),D_{ijk}=D_{ikj},D_{kij}=-D_{jik},D_{kjk}=0,\forall i,k\ge j, \quad (41)$$

THESIS III.a - DEFINITION:

(Robot manipulator) **Complex system dynamics** equation (39) **can be precisely identified by approximating its $\boldsymbol{D}_{ij}$ inertia, *D*$_i$ gravity and *f*$_i$ friction components with FLSs** as:

$$\boldsymbol{D}_i(\boldsymbol{q}) \text{ and } \boldsymbol{D}_{ij}(\boldsymbol{q}) = \ \textstyle\sum_{l=1}^{M}\big((\prod_{i=1}^{n}\mu_{F_l(i)}(q_i,\boldsymbol{b_l}))\cdot(\sum_{j=1}^{n}c_{l(j)}\cdot q_j + c_{l(0)})\big), \text{ and}$$

$$f_i(q_i) = \ \textstyle\sum_{f=1}^{F}\Big(\mu_{F_f(i)}(q_i,\boldsymbol{b}_f)\cdot\big(c_{f(i)}\cdot q_i + c_{f(0)}\big)\Big)$$

where *n* is the number of position state variables (number of RM joints); *M* and *F* is the designed number of FLS rules; *q*$_i$ is the i$^{\text{th}}$ position state variable (RM joint position); $c_{l(j)}$ and $c_{f(i)}$ are the linear parameters to be identified; $\boldsymbol{b_l}$ and $\boldsymbol{b_f}$ are vectors of nonlinear parameters of Zadeh MF formed fuzzy partitions as in equation (25). Components of $\boldsymbol{b_l}, \boldsymbol{b_f}$ are formed as equation (36), (37) in my Thesis II:

$$b_{*,k} = \sum\nolimits_{j=1}^{k} a_{*,j} \Big/ \sum\nolimits_{\kappa=1}^{K} a_{*,\kappa}\ ,\ a_{l,\kappa} \in R_0^+, \kappa = 1,\ldots,K_* \qquad (37)$$

where *K*$_*$ corresponds to *K*$_l$ and *K*$_f$, the designed number of input membership functions of fuzzy partition for ***D***$_{ij}$, ***D***$_i$ and *f*$_i$ FLS antecedents.

Dynamics equation (39) $\boldsymbol{D}_{ijk}$ components are to be expressed from the FLS form of $\boldsymbol{D}_{ij}$ inertia components by applying Christoffel symbols as:

$$D_{ijk} = \tfrac{1}{2}\left(\frac{\partial D_{ij}}{\partial q_k} + \frac{\partial D_{ik}}{\partial q_j} - \frac{\partial D_{jk}}{\partial q_i}\right), D_{ijk} = D_{ikj}, D_{kij} = -D_{jik}, D_{kjk} = 0, \forall i,k \geq j, \quad (41)$$

$$D_{ij} = D_{ji}, D_{ijk} = D_{ikj}, D_{kij} = -D_{jik}, D_{kjk} = 0,\ \forall i,k \geq j \qquad (46)$$

For $a,b,c\ \in \{i,j,k\}$ we have $\frac{\partial D_{ab}(\boldsymbol{q})}{\partial q_c} = \sum_{l=1}^{M}(\frac{\partial(\prod_{i=1}^{n}\mu_{F_l(i)}(q_i,\boldsymbol{b_l})}{\partial q_c}\cdot(\sum_{j=1}^{n}c_{l(j)}\cdot q_j + c_{l(0)}) + (\prod_{i=1}^{n}\mu_{F_l(i)}(q_i,\boldsymbol{b_l})\frac{\partial(\sum_{j=1}^{n}c_{l(j)}\cdot q_j + c_{l(0)})}{\partial q_c})$.

The (RM) system dynamics equation (39) can now be stated as: $\tau_i = \sum_{j=1}^{N}A_{ij}(\ddot{\boldsymbol{q}},\dot{\boldsymbol{q}},\boldsymbol{q},a_{ijl\kappa})\cdot c_{il(j)}$, and the complete body dynamics is now of the form: $\boldsymbol{\tau} = \boldsymbol{A}(\ddot{\boldsymbol{q}},\dot{\boldsymbol{q}},\boldsymbol{q},\boldsymbol{a_\kappa})\cdot\boldsymbol{c}$, where τ_i is the i$^{\text{th}}$ body torque; $\boldsymbol{q}$ is the body position system state, $\dot{\boldsymbol{q}}$ is its first time derivative and $\ddot{\boldsymbol{q}}$ is its second time derivative; A_{ij} is a very complex nonlinear equation to write down, while relatively simply expressed by the stated FLS identification procedure.

Linear system parameters c_* – components of vector $\boldsymbol{c}$ are to be calculated by SVD decomposition based LS optimal method as: $\boldsymbol{c} = \boldsymbol{VS^{-1}U^T}\cdot\boldsymbol{\tau}$ for SVD decomposition of $\boldsymbol{A}(\ddot{\boldsymbol{q}},\dot{\boldsymbol{q}},\boldsymbol{q},\boldsymbol{a_\kappa}) = \boldsymbol{USV^T}$.

Nonlinear system parameters $a_{*,\kappa}$ – components of vector $\boldsymbol{a_\kappa}$ are to be identified with a global stochastic search method, like GAs of my Thesis I, and fine-tuned by a gradient descent method.

4.2.2 Implementation of the New Genetic Fuzzy System Grey-box Modelling of Robot Manipulator Dynamics

My inertia base function choice to form H- the inertia matrix components is multi-input single-output complete first order Takagi-Sugeno-Kang (TSK) type FLSs, having membership functions (MFs) in form of Zadeh type fuzzy partitions [s3]. Components of C - the Coriolis effects and centrifugal forces I evaluate as the Christoffel symbols of FLSs forming H- the inertia matrix; for simplicity I take the friction to be a mere constant. The nonlinear parameters of Zadeh type fuzzy partitions I identify by a multi-objective hybrid evolutionary optimisation method [s4]. The number of nonlinear parameters is reduced to its minimum, the number of fuzzy partitions - FLS inputs is defined by the geometry of the RM, thus cannot be reduced without significant loss in modelling precision.

A gradient-descent method is used for fine tuning all the nonlinear parameter of the dynamic model. The SVD based LS optimal calculation method is used for finding the linear parameters of every fuzzy system (40), also to their required Christoffel symbols (41) and the remaining linear parameters of the RM equation (39) [s5].

4.2.3 Results of the New Genetic Fuzzy System Grey-box Robot Manipulator Modelling

The result of a quick, small evolutionary search is a chromosome of (37522, 32020, 65333, 53411) for the four nonlinear a_K parameters. Linear parameters of the dynamic model are listed in Table XII below. The first column names the parameter, the second contains its exact value, results of geometrical an mechanical analysis of joints and its configuration. The third column contains an earlier result [s10], where the D_{ijk} components were identified as simple FLSs, independent of D_{ij}. Result of the proposed method is listed in the fourth column of Table XII. Table XIII Contains the parameters of MFs forming Zadeh-type fuzzy partitions. Table XIV lists all the linear parameters that form consequent parts of all nine rules of both FLSs for D_{11} and D_{12}.

The identification error of torque acting on joints 1-4 is presented in Figure 20.

	Exact	LSQ [s11]	SVD [s5]
D_{11}	Func_{11}	FLS_{11}	FLS_{11}
D_{12}	Func_{12}	FLS_{12}	FLS_{12}
D_{14}	0.004	1.1315	0.0098776
D_{22}	1.1454	0.19586	1.1388
D_{24}	0.004	-0.25933	0.005855
D_{33}	130.2521	130.2521	130.2521
D_{44}	0.409	-0.45519	0.40839
D_{112}	$\mathrm{f}(\mathrm{Func}_{11})$	FLS_{112}	$\mathrm{f}(\mathrm{FLS}_{11})$
D_{114}	0	0.0022199	0
D_{214}	0	0.00024719	0.023923
D_3	67.1985	67.1985	67.1985
f_1	14.5031	14.5031	14.5031
f_2	13.8	13.7869	13.5866
f_3	3948.9	3948.9	3948.9
f_4	13.4	13.4001	13.4008

Table XII. Linear parameters of the dynamic model.

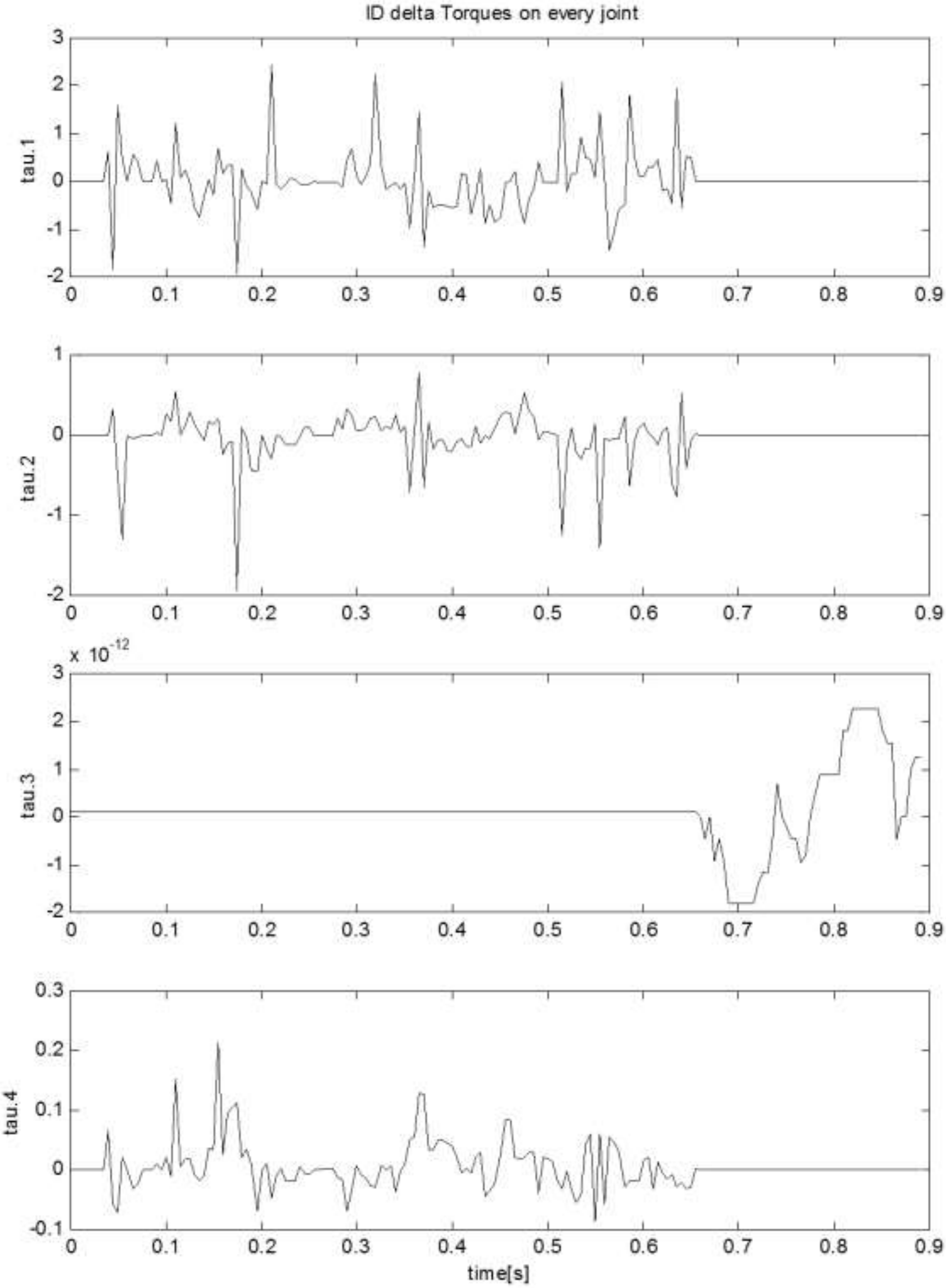


Figure 20.
SVD based SCARA RM torque identification error in [Nm] for joints 1,2,3 and 4.

	b_1	b_2	b_3
${}_{11}zMF_{11}(q_2)$	0	0.5725	-
${}_{11}\pi MF_{12}(q_2)$	0	0.5725	1
${}_{11}sMF_{13}(q_2)$	0.5725	1	-
${}_{11}zMF_{21}(q_4)$	0	0.48862	-
${}_{11}\pi MF_{22}(q_4)$	0	0.48862	1
${}_{11}sMF_{23}(q_4)$	0.48862	1	-
${}_{12}zMF_{11}(q_2)$	0	0.99693	-
${}_{12}\pi MF_{12}(q_2)$	0	0.99693	1
${}_{12}sMF_{13}(q_2)$	0.99693	1	-
${}_{12}zMF_{21}(q_4)$	0	0.81503	-
${}_{12}\pi MF_{22}(q_4)$	0	0.81503	1
${}_{12}sMF_{23}(q_4)$	0.81503	1	-

Table XIII. Nonlinear parameters ${}_{11}\mathrm{xMF}_{\mathrm{xx}}$ for FLS_{11} and ${}_{12}\mathrm{xMF}_{\mathrm{xx}}$ for FLS_{12}.

	c_0	c_1	c_2
${}_{11}y_{11}(q_2, q_4)$	1.5836	-0.31855	0.0059997
${}_{11}y_{12}(q_2, q_4)$	-0.072244	0.014675	-0.033884
${}_{11}y_{13}(q_2, q_4)$	1.5836	-0.31855	0.0059997
${}_{11}y_{21}(q_2, q_4)$	0.43328	-0.25865	0.10659
${}_{11}y_{22}(q_2, q_4)$	0.95086	-0.42537	-0.80804
${}_{11}y_{23}(q_2, q_4)$	0.43328	-0.25865	0.10659
${}_{11}y_{31}(q_2, q_4)$	1.5836	-0.31855	0.0059997
${}_{11}y_{32}(q_2, q_4)$	-0.072244	0.014675	-0.033884
${}_{11}y_{33}(q_2, q_4)$	1.5836	-0.31855	0.0059997
${}_{12}y_{11}(q_2, q_4)$	0.3856	0.15897	-0.0065901
${}_{12}y_{12}(q_2, q_4)$	0.098905	-0.43382	0.022196
${}_{12}y_{13}(q_2, q_4)$	0.3856	0.15897	-0.0065901
${}_{12}y_{21}(q_2, q_4)$	-0.79427	-0.34648	0.0096337
${}_{12}y_{22}(q_2, q_4)$	0.12912	0.77157	-0.053574
${}_{12}y_{23}(q_2, q_4)$	-0.79427	-0.34648	0.0096337
${}_{12}y_{31}(q_2, q_4)$	0.3856	0.15897	-0.0065901
${}_{12}y_{32}(q_2, q_4)$	0.098905	-0.43382	0.022196
${}_{12}y_{33}(q_2, q_4)$	0.3856	0.15897	-0.0065901

Table XIV. Linear parameters of rule consequent ${}_{11}\mathrm{y}_{\mathrm{xx}}$ for FLS_{11} and ${}_{12}\mathrm{y}_{\mathrm{xx}}$ for FLS_{12}.

The mean square error is *0.05*, the maximal absolute error is *2.44* Nm and there are no more than nine such points where the error is greater than three times the standard deviation of the error. The relative value of the maximal error is *2.83%*, *2.23%*, *0%*, *0.66%* for RM joints 1, 2, 3 and 4 respectively. These results are of more than satisfactory precision for an advanced computed torque (aka. back tracking) model based control algorithm implementation.

By this analysis I conclude that my Thesis III.a is proven valid.

4.2.4 New Continuous Periodic Fuzzy Logic Systems

As described in [s6], [s13], [s14] my proposal is to transform the general FLS equation (27) to form a continuous periodic FLS (cpFLS). Such cpFLSs are ready to be used for modelling systems which are inherently continuous and periodic, for example the orientation angle input based torque function of a multi-rotor dynamics in equation (43).

For physical systems in the Euclidian space orientation angles are naturally defined on the [0, 2π) interval. Any angular value α below *0* or above *2π* is equivalent to a value *β=α±2kπ*, where k is such an ordinary number that *β∈ [0,2π)*. For orientation angles selection of the origin is arbitrary and transition between two orientation angles is continuous and smooth, as in having a continuous first derivative. As orientation angle of 2π is equivalent to angle 0 the transition from 2π-ε to 0+ε also has to be continuous.

For FLSs defined by equation (27) we can make the input space continuous and periodic over the *[0, 2π)* interval by applying a simple piecewise linear “seesaw” function transformation as in Figure 21, whose output is in [-1,1] as defined by:

$$\hat{q} = \begin{cases} 2 * (\pi - q)/\pi, & 3 * \pi/2 < q > \pi/2 \\ 2 * (q - \pi)/\pi, & q > 3 * \pi/2 \\ 2 * (q)/\pi, & else \end{cases} \tag{47}$$

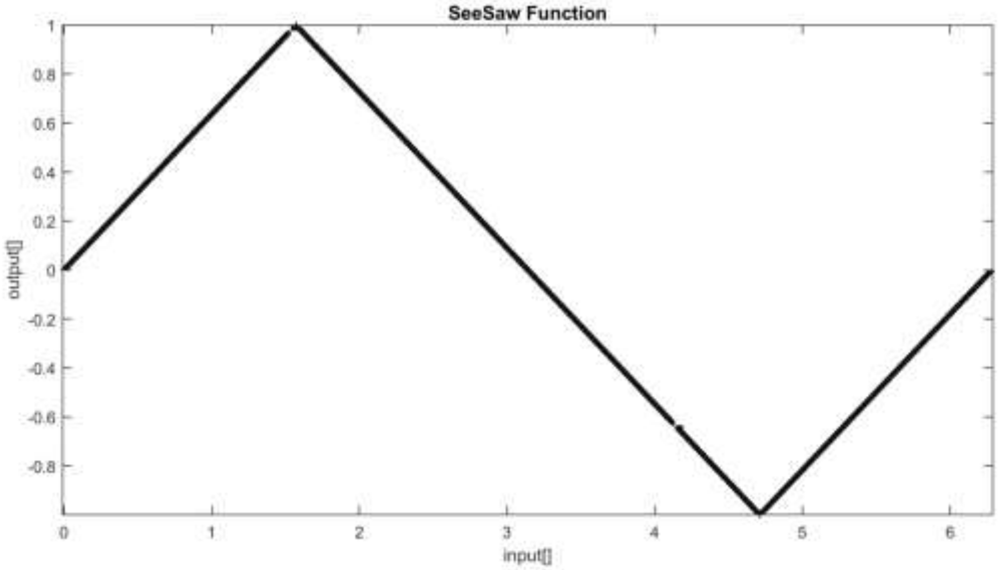


Figure 21.
The piecewise linear “seesaw” function.

With transformation (47) of the cpFLS input space we make sure that there is no discontinuity between angular cpFLS inputs of any two values, we simply force critical *2π-ε* input values to become equal to *0+ε* for all $\varepsilon < \pi/2$. This step is needed to ensure the output space $y_l(\boldsymbol{q})$ consequence part can become continuous over the ***q*** *∈[0, 2π)* input space even for full circle rotations.

We also have to make the antecedent fuzzy partition “circular” by combining the first μ_z and the last μ_s MF of the partition as defined in equation (25) into a single virtual μ_π MF to be substituted into equation (23), so that fuzzy rules applied to the first z-MF equally apply to the last s-MF. We achieve this by making all the linear parameters of the last rule for each fuzzy partition in equation (27) equivalent to the first rule of the same partition as $c_{jK_i} = c_{jK_1}$, where *n* is the number of cpFLS inputs, and each input is covered by a fuzzy partition of K_i MFs for *i=1..n*.

By this procedure we have ensured to have a continuous periodic fuzzy system (cpFLS) such that for $\forall \boldsymbol{q} \in \mathbb{R}^n, \forall k \in \mathbb{Z}$ and any arbitrary small $\boldsymbol{\varepsilon}$ there is a similarly small $\mu(\boldsymbol{\varepsilon})$ for which we have:

$$cpFLS(\boldsymbol{q} \pm 2k\pi) = \; cpFLS(\boldsymbol{q}),\; cpFLS(\boldsymbol{q} \pm \boldsymbol{\varepsilon}) = cpFLS(\boldsymbol{q}) \pm \mu(\boldsymbol{\varepsilon}), \qquad (48)$$

Table XV presents the linear parameter triplets for all 25 fuzzy rule consequents $\boldsymbol{c_l} = (c_{0l} c_{1l} c_{2l})$, of equation (27) for a TSK FLS interpretation of a cpFLS with *n=2* inputs, where each antecedent is a Zadeh formed fuzzy partition of 5 MFs.

	μ_{z11}	$\mu_{\pi12}$	$\mu_{\pi13}$	$\mu_{\pi14}$	μ_{s15}
μ_{z21}	$\boldsymbol{c_1}$	$\boldsymbol{c_2}$	$\boldsymbol{c_3}$	$\boldsymbol{c_4}$	$\boldsymbol{c_1}$
$\mu_{\pi22}$	$\boldsymbol{c_6}$	$\boldsymbol{c_7}$	$\boldsymbol{c_8}$	$\boldsymbol{c_9}$	$\boldsymbol{c_6}$
$\mu_{\pi23}$	$\boldsymbol{c_{11}}$	$\boldsymbol{c_{12}}$	$\boldsymbol{c_{13}}$	$\boldsymbol{c_{14}}$	$\boldsymbol{c_{11}}$
$\mu_{\pi24}$	$\boldsymbol{c_{16}}$	$\boldsymbol{c_{17}}$	$\boldsymbol{c_{18}}$	$\boldsymbol{c_{19}}$	$\boldsymbol{c_{16}}$
μ_{s25}	$\boldsymbol{c_1}$	$\boldsymbol{c_2}$	$\boldsymbol{c_3}$	$\boldsymbol{c_4}$	$\boldsymbol{c_1}$

Table XV. TSK FLS-like interpretation of cpFLS linear parameter triplets.

In the compact cpFLS interpretation form of the same fuzzy system as in Table XV, we have to consider only linear parameter triplets for 16 unique fuzzy rule consequents as presented in Table XVI, where $\mu_{\pi ji}$ for *i>0* are equivalent to Table XV MFs of the same index, and $\mu_{\pi j0}(q) = \mu_{zj1}(q) + \mu_{sj5}(q)$ for *j=1,2*.

Triplets $\boldsymbol{c_l}$ in Table XVI are equivalent to triplets of matching index from Table XV.

	$\mu_{\pi10}$	$\mu_{\pi12}$	$\mu_{\pi13}$	$\mu_{\pi14}$
$\mu_{\pi20}$	$\boldsymbol{c_1}$	$\boldsymbol{c_2}$	$\boldsymbol{c_3}$	$\boldsymbol{c_4}$
$\mu_{\pi22}$	$\boldsymbol{c_6}$	$\boldsymbol{c_7}$	$\boldsymbol{c_8}$	$\boldsymbol{c_9}$
$\mu_{\pi23}$	$\boldsymbol{c_{11}}$	$\boldsymbol{c_{12}}$	$\boldsymbol{c_{13}}$	$\boldsymbol{c_{14}}$
$\mu_{\pi24}$	$\boldsymbol{c_{16}}$	$\boldsymbol{c_{17}}$	$\boldsymbol{c_{18}}$	$\boldsymbol{c_{19}}$

Table XVI. The compact cpFLS interpretation.

THESIS III.b - DEFINITION:

All FLSs that for antecedent use Zadeh-type MF based fuzzy partitions, whose last (*smf*) and first MF (*zmf*) can form a single continuous MF (*πmf*), and the consequent part of rules is a constant (like in Mamdani FLS) or a continuous function of the input signal (like a TSK FLS), can be made **continuous and periodic fuzzy logic system (cpFLS)** as in equation (48) by:

1. applying equation (47) as a preliminary transformation to the input signal

2. making all parameters c_{jK_i} (like $c_{5,10,15,20,25}$ and $c_{21,22,23,24,25}$ in Table XV) of fuzzy rule consequents whose premise includes the *smf (*like $\mu_{s1,5}$ and μ_{s25} in Table XV*)* identical to c_{jK_1} parameters (like $c_{1,6,11,16,1=21}$ and $c_{1,2,3,4,1=5}$ in Table XV) of rule consequents for the matching *zmf* of the same input.

4.2.5 New Genetic Fuzzy System Grey-box Modelling of Multi-rotor Flight Dynamics

As described in [s6], [s13], [s14] my proposal is to use continuous periodic FLS (cpFLS) for modelling systems which are inherently continuous and periodic, for example the orientation angle input based torque function of a multi-rotor dynamics in equation (43).

THESIS III.c - DEFINITION:

(Multi-rotor) **flight dynamics can be precisely identified by continuous and periodic fuzzy logic systems** by taking system components for cpFLSs (my Thesis III.b) as in equations (39), (46), (50), (51), (52) for $\boldsymbol{q}$ = *(ϕ, θ, ψ),* by applying the identification method as equation (53), which is detailed in my Thesis III.a as:

$$\sum_{j=1}^{p}\left(\boldsymbol{D}_{ij}(\boldsymbol{q})\cdot\ddot{\boldsymbol{q}}_j\right)+\sum_{j=1}^{p}\sum_{k=1}^{p}\left(\dot{\boldsymbol{q}}_{\boldsymbol{j}}\cdot\boldsymbol{D}_{ijk}(\boldsymbol{q})\cdot\dot{\boldsymbol{q}}_k\right)+\boldsymbol{D}_i(\boldsymbol{q})+f_i=\tau_i, \tag{39}$$

$$D_{13}(\theta)=f_1(\theta), D_{22}(\phi)=f_2(\phi), D_{23}(\phi,\theta)=f_3(\phi,\theta), D_{33}(\phi,\theta)=f_4(\phi,\theta), \tag{50}$$

$$D_{11}=I_{xx}, D_{12}=0, D_{21}=D_{12}, D_{31}=D_{13}, D_{32}=D_{23}, \tag{51}$$

$$\begin{gathered}D_{122}=-\frac{1}{2}\frac{\delta D_{22}}{\delta\phi}, D_{123}=\frac{1}{2}\left(\frac{\delta D_{13}}{\delta\theta}-\frac{\delta D_{23}}{\delta\phi}\right), D_{322}=\frac{\delta D_{23}}{\delta\theta},\\ D_{133}=-\frac{1}{2}\frac{\delta D_{33}}{\delta\phi}, D_{223}=\ -\frac{1}{2}\frac{\delta D_{33}}{\delta\theta}, D_{312}=\frac{1}{2}\left(\frac{\delta D_{23}}{\delta\phi}+\frac{\delta D_{13}}{\delta\theta}\right)\end{gathered} \tag{52}$$

$$D_{ij}=D_{ji}, D_{ijk}=D_{ikj}, D_{kij}=-D_{jik}, D_{kjk}=0,\ \forall i,k\geq j, \tag{46}$$

$$\left(\mathbb{J}^*(\boldsymbol{q},\boldsymbol{a_\kappa})\cdot\ddot{\boldsymbol{q}}+\mathbb{C}^*(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{a}_\kappa)\cdot\dot{\boldsymbol{q}}\right)\cdot\boldsymbol{c}=\boldsymbol{A}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}},\boldsymbol{a}_\kappa)\cdot\boldsymbol{c}=\boldsymbol{\tau}, \tag{53}$$

For such a flight dynamics system model the minimal number of $\boldsymbol{a_\kappa}$ nonlinear parameters is 24 and the number of $\boldsymbol{c}$ linear parameters is 113 to achieve a good quality model, when the f_i friction components are neglected; $\boldsymbol{D_i}$ gravity components for a free flying object are non-existent.

In analogy to my Thesis III.a, linear flight dynamics system parameters c_* – components of vector $\boldsymbol{c}$ are to be calculated by SVD decomposition based LS optimal method as: $\boldsymbol{c}=\boldsymbol{VS^{-1}U^T}\cdot\boldsymbol{\tau}$ for SVD decomposition of $\boldsymbol{A}(\ddot{\boldsymbol{q}},\dot{\boldsymbol{q}},\boldsymbol{q},\boldsymbol{a_\kappa})=\boldsymbol{USV^T}$.

Nonlinear system parameters $a_{*,\kappa}$ – components of vector $\boldsymbol{a_\kappa}$ are to be identified with a global stochastic search method, like GAs of my Thesis I, and fine-tuned by a gradient descent method.

4.2.6 Implementation of New Genetic Continuous Periodic Fuzzy System Grey-box Modelling

As proposed in [s5] we identify D_{ij} components of the dynamic model in equation (44) as FLSs defined by equations (24) to (27), where the FLS general input variable $\boldsymbol{q}$ will be substituted for appropriate state variables of *(ϕ, θ, ψ).*

Instead of simple TSK FLSs we will use cpFLSs. Where the D_{ij} inertia matrix components are modelled by cpFLSs, forming the D_{ijk} components as Christoffel symbols is to be expressed by partial derivatives of equation (27) like:

$$\partial f(\boldsymbol{q})/\partial q_i=\sum_{i=1}^{M}\left(\partial\omega_l(\boldsymbol{q})/\partial q_i\cdot y_l(\boldsymbol{q})+\omega_l(\boldsymbol{q})\cdot\partial y_l(\boldsymbol{q})/\partial q_i\right), \tag{49}$$

The unknown four inertia matrix components of torques defined in equation (43), which have to be identified for a 3DOF rigid body rotational motion model, are:

$$\boldsymbol{D}_{13}(\theta) = f_1(\theta), \boldsymbol{D}_{22}(\phi) = f_2(\phi), \boldsymbol{D}_{23}(\phi,\theta) = f_3(\phi,\theta), \boldsymbol{D}_{33}(\phi,\theta) = f_4(\phi,\theta), \tag{50}$$

Based on the multi-rotor system structure and inertia matrix symmetry the remaining inertia components are known to be:

$$D_{11} = I_{xx}, D_{12} = 0, \boldsymbol{D}_{21} = \boldsymbol{D}_{12}, \boldsymbol{D}_{31} = \boldsymbol{D}_{13}, \boldsymbol{D}_{32} = \boldsymbol{D}_{23}, \tag{51}$$

where I_{xx} is the constant multi-rotor body inertia around the *x* axis.

Based on equation (46) the following Coriolis term matrix $\boldsymbol{D}_{ijk}$ components can be calculated by equations (49):

$$\begin{aligned} &\boldsymbol{D}_{122} = -\frac{1}{2}\frac{\delta \boldsymbol{D}_{22}}{\delta\phi}, \boldsymbol{D}_{123} = \frac{1}{2}\left(\frac{\delta \boldsymbol{D}_{13}}{\delta\theta} - \frac{\delta \boldsymbol{D}_{23}}{\delta\phi}\right), \boldsymbol{D}_{322} = \frac{\delta \boldsymbol{D}_{23}}{\delta\theta} \\ &\boldsymbol{D}_{133} = -\frac{1}{2}\frac{\delta \boldsymbol{D}_{33}}{\delta\phi}, \boldsymbol{D}_{223} = -\frac{1}{2}\frac{\delta \boldsymbol{D}_{33}}{\delta\theta}, \boldsymbol{D}_{312} = \frac{1}{2}\left(\frac{\delta \boldsymbol{D}_{23}}{\delta\phi} + \frac{\delta \boldsymbol{D}_{13}}{\delta\theta}\right) \end{aligned} \tag{52}$$

The remaining $\boldsymbol{D}_{ijk}$ components are trivial identities as defined in equation (46). This way we can model the complete multi-rotor rotation dynamics as defined in equation (43) by only 1 linear constant and 4 cpFLSs, where 2 cpFLSs are functions of a single input, and 2 are functions of 2 inputs.

We have for these fuzzy systems 6 Zadeh type fuzzy partitions. Each partition consists of 1 μ_z-, 1 μ_s-, and 3 μ_π-type MFs as presented by equations (25) and (37), such a fuzzy partition is defined by 3 nonlinear a_i parameters; 6 partitions totalling in 24 nonlinear $\boldsymbol{a}_\kappa$ parameters.

These 4 cpFLSs consist of 2 times 4 rules for single input functions an 2 times 16 rules y_l as defined in equation (27) for two input functions. Each rule consequent y_l is defined by 2 (single input case) or 3 (two inputs case) c_{il} linear parameters, these 4 cpFLSs total in 112 linear parameters. The grand total for our model is 24 nonlinear $\boldsymbol{a}_\kappa$ and 113 linear **c** parameters.

Linear parameters are best directly evaluated by a singular value decomposition (SVD) based least squares (LS) fitting method. We first substitute equations (16, 18, 20, 48) to (51, 52, 53) and all to (43) expressed as (53), then we express all the 113 linear $\boldsymbol{c}$ parameters as:

$$(\mathbb{J}^*(\boldsymbol{q}, \boldsymbol{a}_\kappa) \cdot \ddot{\boldsymbol{q}} + \mathbb{C}^*(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{a}_\kappa) \cdot \dot{\boldsymbol{q}}) \cdot \boldsymbol{c} = \boldsymbol{A}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{a}_\kappa) \cdot \boldsymbol{c} = \boldsymbol{\tau}, \tag{53}$$

For SVD decomposition of $\boldsymbol{A}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{a}_\kappa) = U \cdot S \cdot V^T$ we obtain $\boldsymbol{c} = V \cdot S^{-1} \cdot U^T \cdot \boldsymbol{\tau}$.

The used training data is from calculated roll and pitch motions based on my Thesis IV, as presented in Figure 22. The simulated resultant torque full training data set is presented in Figure 23; its size is reduced for calculations based on my Thesis V.

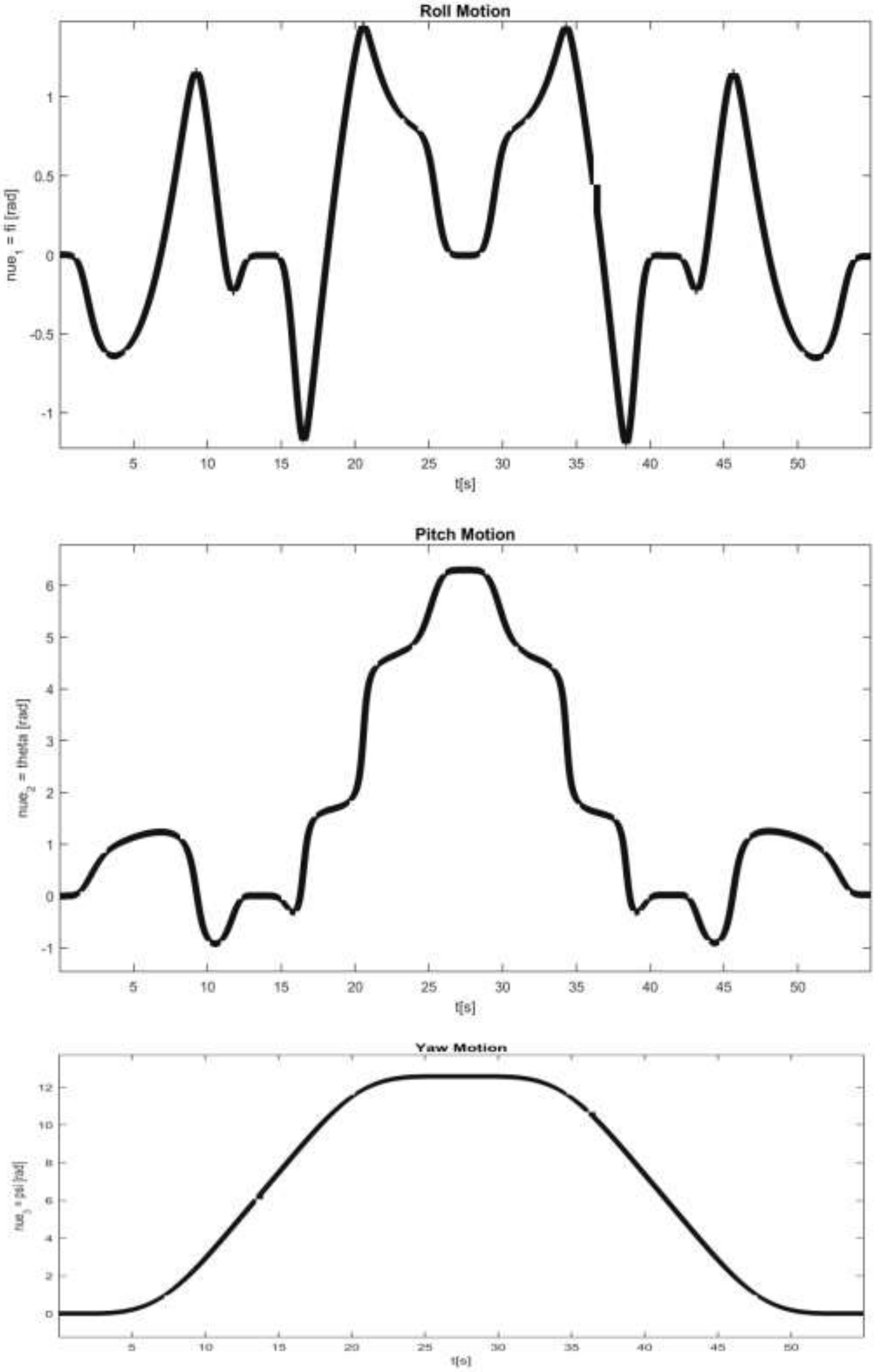


Figure 22.
Smooth roll, pitch and yaw motions of the multirotor.

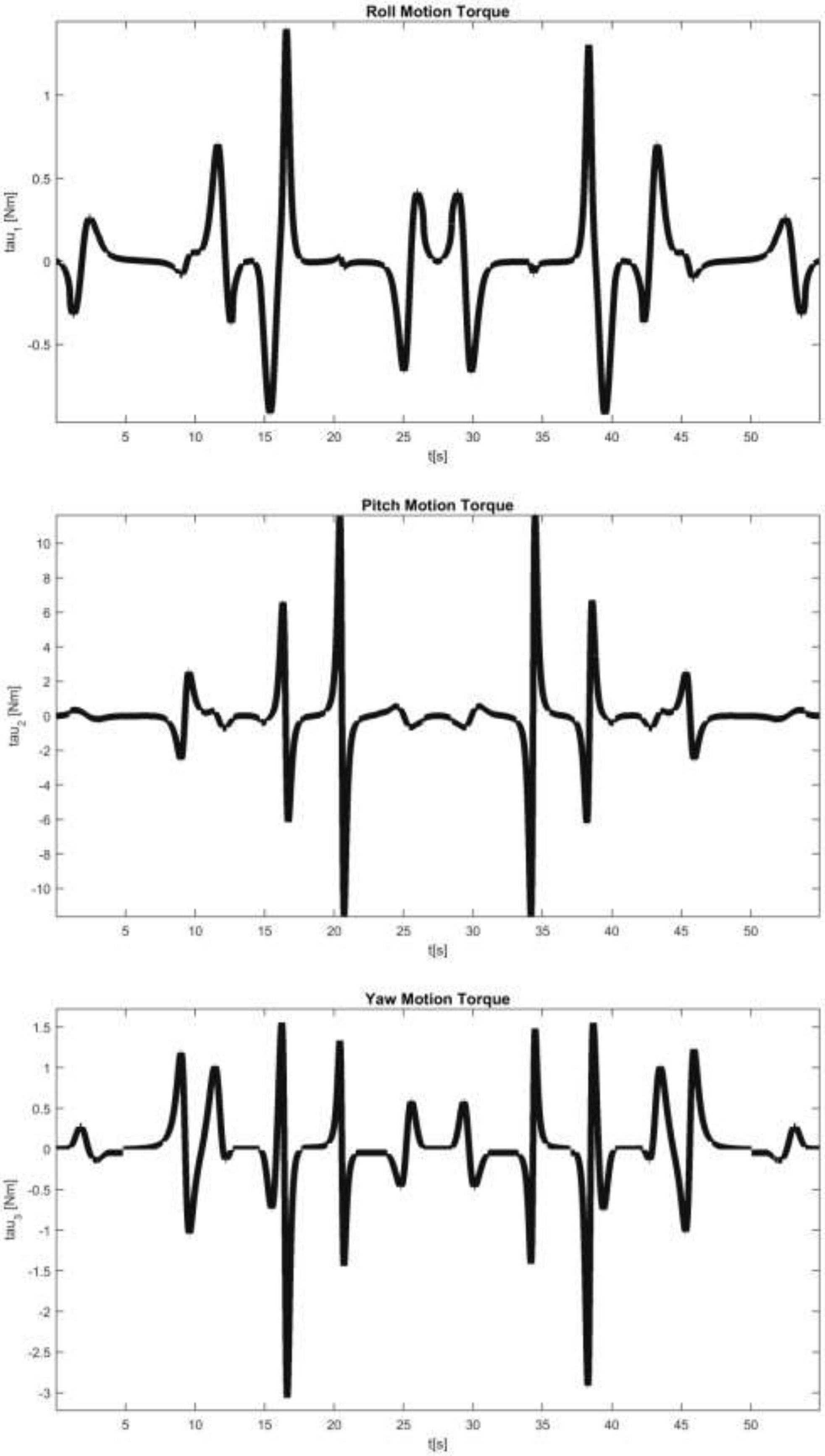


Figure 23.
Smooth resultant torques – full training set before reduction.

The list of nonlinear cpFLS parameters consists of six times four integer parameters for defining six fuzzy partitions having five MFs each, where each partition consists of three classical π-type MFs and for the cpFLS setup one virtual π-type MF composed by one Z-type MF at the beginning of the input interval and one S-type MF in the end of the input interval as in equation (25). These six fuzzy partitions serve as antecedents for the four fuzzy systems like in equation (27) and (48), used for identifying D_{ij}, $ij=(13, 22, 23, 33)$ as defined in equations (44)-(46) and (49)-(52).

The unknown linear parameter D_{11} of the multi-rotor model as in equation (53), together with 112 linear parameters of the four TSK FLSs (2 FLSs with 5 MFs on one input, each rule with 2 *c* parameters, plus 2 FLSs with 5 MFs on both of the 2 inputs, each rule with 3 *c* parameters) of equations (43) are first substituted to (27) and (49), then based on (45), (46) and (44) to (50), (51) and (52), that they can be determined by the SVD-based LS method from equation (43) expressed as (53).

Concluded from equation (25) and (37) six fuzzy partitions (antecedent part of 2 FLSs with 1 input, plus 2 FLSs with 2 inputs are covered by 6 independent fuzzy partitions) are represented by a vector of six times four parameters, which are optimized by a multi-objective hybrid genetic algorithm as detailed in [s9]. Chromosomes are evaluated and subjected to a local gradient based search. Chromosome values are updated with the result of fine-tuning after each evaluation, so the GA does not waste time on local optimization; only global search capabilities of the GA are utilized.

The GA is set to work on a population of 200 chromosomes, divided into 5 subpopulations, with migration rate 0.2 taking place after each 5 completed generations. Chromosomes are comprised of 24 Gray-coded integers, each consisting of 16 bits. The initial population is set up in a completely random manner. Crossover rate, generation gap and insertion rate is set to 0.8, selection pressure is 1.5. In each generation 4% of individuals are subject to mutation, when 1% of the binary genotype is mutated.

Matrix of the linear equation $\mathbb{Q}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ from equation (53) is pre-processed, as FLSs like equation (27) and their partial derivatives like equation (49) are substituted as defined by equations (50)-(52). Unknown linear parameters are D_{11} and the 112 $\boldsymbol{c}$ parameters of fuzzy rule consequents.

Evaluation of each individual is conducted as follows:

(a) Convert coded a_i values from the chromosome to b_k by equation (37).

(b) Evaluate all MFs, and antecedents which will comprise six fuzzy partitions from each of six b_k quadruplets by equations (23) and (25). Also evaluate antecedent derivatives of equation (49).

(c) Calculate the matrix coefficients of linear equations $\mathbb{Q}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ by values of triggered MFs and their partial derivatives.

(d) Linear components $[D_{11}, \boldsymbol{c}]$ of equations (27), (43) and (49) are calculated by SVD decomposition as in (53).

(e) Fine-tune a_i parameters, for example by the Matlab “lsqnonlin” function, while re-calculating steps (a)-(d) for each a_i tuning iteration.

(g) Re-insert optimized a_i parameters into the evaluated chromosome.

For the multi-objective rank assignment described in [s2], the objective vector is created from:

(i) the mean square of the identified torque error,

(ii) the maximum absolute torque identification error and

(iii) the condition number of the matrix of the linear equation.

Stochastic universal sampling is used for selecting the next generation without explicit elitism. To speed up the GA processing, a database of evaluated chromosomes and their objective vectors is created, so only unique new individuals are evaluated in each generation.

4.2.7 Results of New Genetic Continuous Periodic Fuzzy System Grey-box Modelling of Multi-rotor Flight Dynamics

Results of the identification process - averages of 10 simulations for each training data set size are presented in Table XVII. For identification quality comparison to the new cpFLS model [s6], results of a simple FLS model, described in [s9] are also presented.

Results of a selected non dominated model are followed by the average values and the variance of 10 runs of each method and parametrization, as defined in the first 4 rows.

FLS type	**FLS[s9]**	**cpFLS**	**cpFLS**	**cpFLS**	**cpFLS**	**cpFLS**
Training points	5487	5487	2743	1170	685	685
Population #	500	500	500	500	500	75
Generation #	200	200	200	200	200	25
MSE selected	0,0007	0,0008	0,0009	0,0009	0,0008	0,0008
maxE selected	0,1531	0,1771	0,1786	0,1648	0,1952	0,2080
Cond selected	2707,1	5506,4	3399,4	4145,9	4262,2	3032,0
MSE mean	0,0008	0,0009	0,0010	0,0010	0,0012	0,0011
MSE variance	0,0001	0,0001	0,0001	0,0002	0,0003	0,0002
maxE mean	0,1845	0,1977	0,2912	0,2329	0,2968	0,2408
maxE variance	0,0324	0,0330	0,2568	0,0540	0,1020	0,0382
Cond mean	3815,5	7342,8	5651,6	6953,7	5368,1	5202,9
Cond variance	1701,2	8493,5	5259,3	5274,6	2434,5	2039,3

Table XVII. Maximum error, mean square error and condition number results.

Convergence of applied multi objective GAs of population size (Population#) 200, (or 75) is achieved in <200, (or <25) generation evaluations (Generation#), when the typical mean square error (MSE) is <1e-3, the typical maximum torque error (maxE) is <0.2 Nm, which means that the typical relative error is <10%.

For fuzzy partitions defined by non-dominated chromosomes the typical condition number (Cond) of $\boldsymbol{A}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{a_\kappa})$ the matrix of linear equations is ~5000.

There is neither significant quality, nor identification performance degradation when using the new cpFLS compared to the simple FLS method described in [s9]. While the most significant benefit of smooth output transitions for (2kπ-ε) – (0+ε) input space changes of continuous periodic FLSs cannot be achieved by a simple TSK FLS.

In average there is no significant difference in the identification quality for any of the reduced or the full data set. The only real, significant difference is the time required for evaluation, which is proportional to the increased speed of singular value decompositions of different sized samples. The 685 point sample set reduced to 1/8th of the full set is significantly faster evaluated than any other larger set.

Further on the proposed GA setup method is robust enough to compensate for a significant reduction of the GA size parameters as well. The identification result is still very good when reducing the population size to 75 and generations evaluated to 25, which result in a (500*200)/ (75*25)≈5000% fold GA execution time gain.

Numerical values representing a_i of equation (37) for the selected typical non-dominated chromosome are:

[3157 31387 59087 34526 61856 23999 31983 5100 21985 53525 13592 15164 41416 52669 17091 8246 27195 36846 42384 27934 32215 55957 57320 24610], which defines fuzzy partition MF parameters by equation (37) as:

$\boldsymbol{b_i}$ $i=1,2,3$ for cpFLS modelling D_{13}: [0.024633, 0.26954, 0.7306].

$\boldsymbol{b_i}$ $i=1,2,3$ for cpFLS modelling D_{22}: [0.50315, 0.69836, 0.95851].

$\boldsymbol{b_i}$ $i=1,2,3$ for D_{23}: [0.21085, 0.72421, 0.85457; 0.34681, 0.78784, 0.93095].

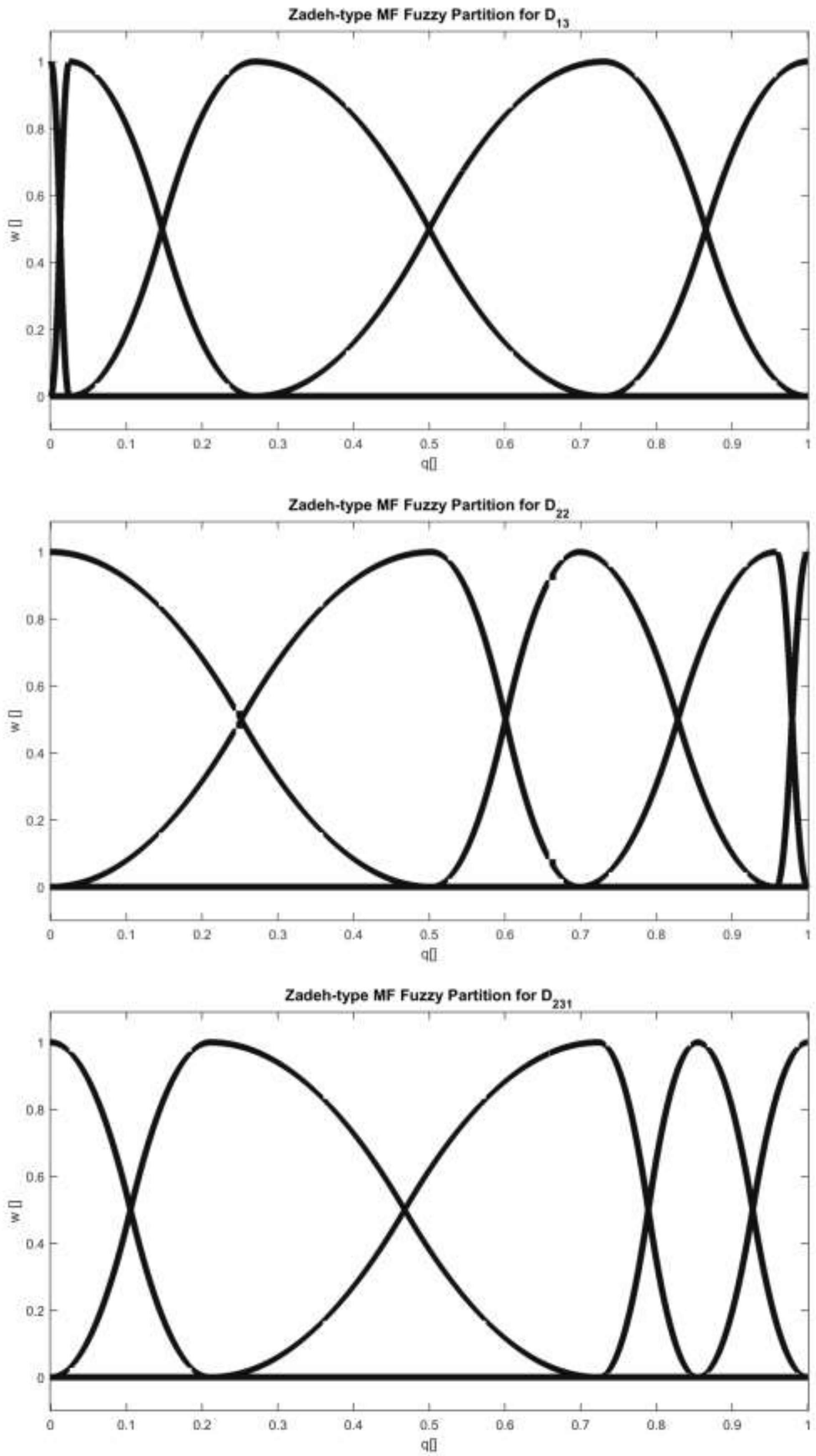
$\boldsymbol{b_i}$ $i=1,2,3$ for D_{33}: [0.20241, 0.47664, 0.79209; 0.18939, 0.51835, 0.85532].

The graphical representation of a fuzzy partition antecedents defined by this chromosome is shown in Figure 24a and Figure 24b.

The torque identification error along the complete training data set for a cpFLS based quad-rotor model defined by the selected non-dominated chromosome is presented in Figure 25.

The cpFLS identified torque is continuous as there are no discontinuities in the absolute error, and the identification error is minor, having MSE <1e-3.

By this analysis I conclude that my Thesis III.b and III.c are proven valid.



Zadeh-type MF Fuzzy Partition for D_{13}

Zadeh-type MF Fuzzy Partition for D_{22}

Zadeh-type MF Fuzzy Partition for D_{231}

Figure 24a.
Antecedent fuzzy partition for D_{12}, D_{22}.

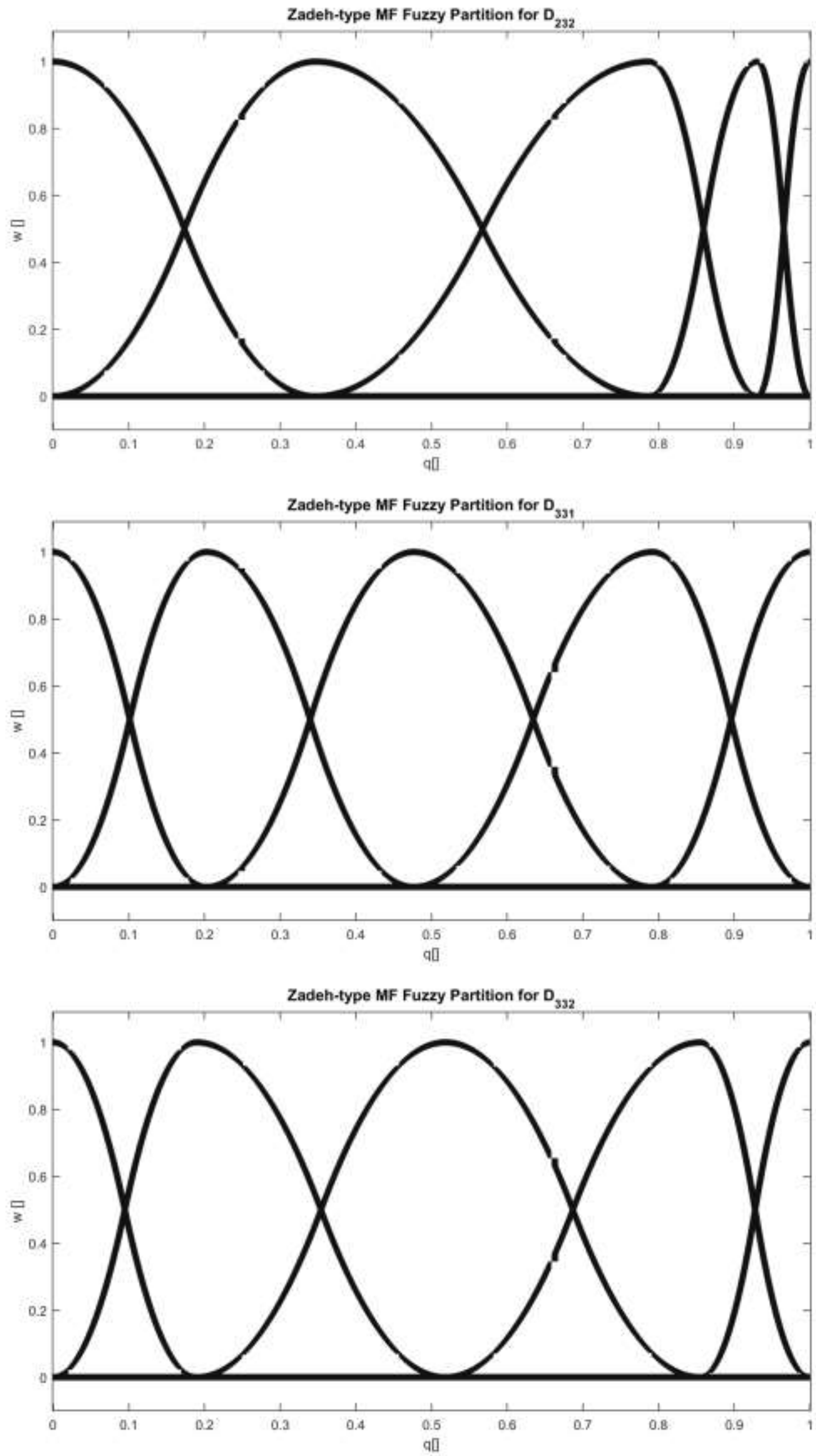


Figure 24b.
Antecedent fuzzy partition for D_{23}, D_{33}.

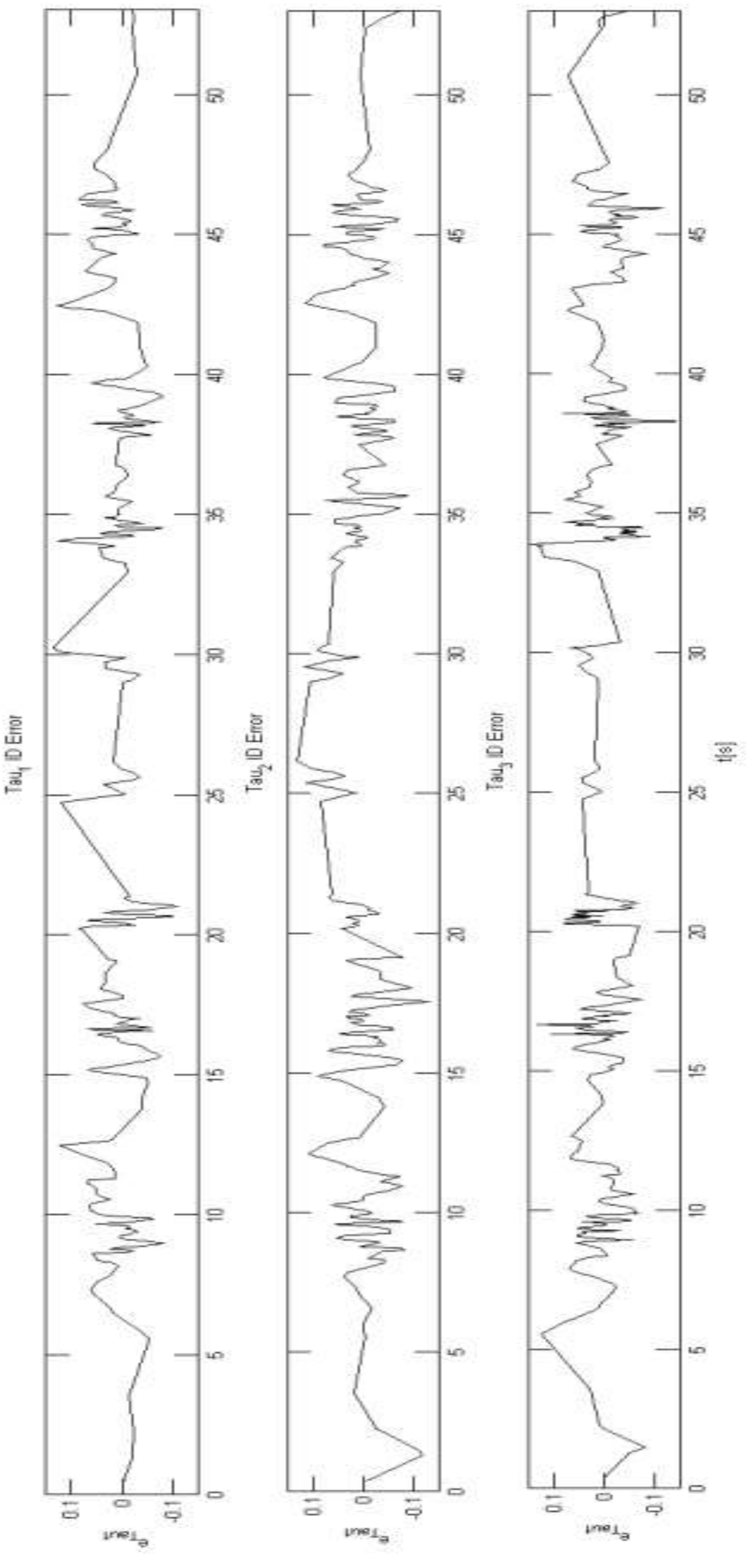


Figure 25.
Absolute identification error for each roll, pitch yaw motion torque in [Nm].

5 OPTIMAL SYSTEM TRAJECTORY DESIGN

5.1 Literature Synopsis

Optimal trajectory design is still a challenging task, though it is usually considered well formulised since Pontryagin's work in the early '60s [58]. For all differentially flat systems state variables $\boldsymbol{q}$ as part of the complete trajectory ($\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}},$) are directly influenced by the desired path of the system [4].

5.1.1 Basic Approaches to Optimal Trajectories

The basic trajectory design problem to solve is: moving from point A to point B. In general, there are set boundaries on the trajectory in terms of allowed geometrical regions or obstacles, velocity, acceleration or even jerk (third time derivative of displacement) limits are defined.

Usually time optimality and/or energy optimality requirements are associated with a planned movement along a chosen path. The pure theoretical approach starts with finding possible 3D geometric path curves from point A to point B. The exact time optimal solution is then selecting the shortest geometric displacement curve s and traveling along this path with v_{max} the maximum allowed velocity for the shortest time of $t_f = s/v_{max}$. See Figure 26, where for a very modest v_{max} = *0.08m/s* an up to 8'g' *(8* times *9.81m/s*^{2}*)* acceleration is required, not to mention the jerk of ~10^5m/s^3…

We must never overlook that the kinematic results have to be applied to actually moving not a virtual point, but a body of mass m, while we have real actuator outputs as the means of control instead of arbitrary mathematical signals.

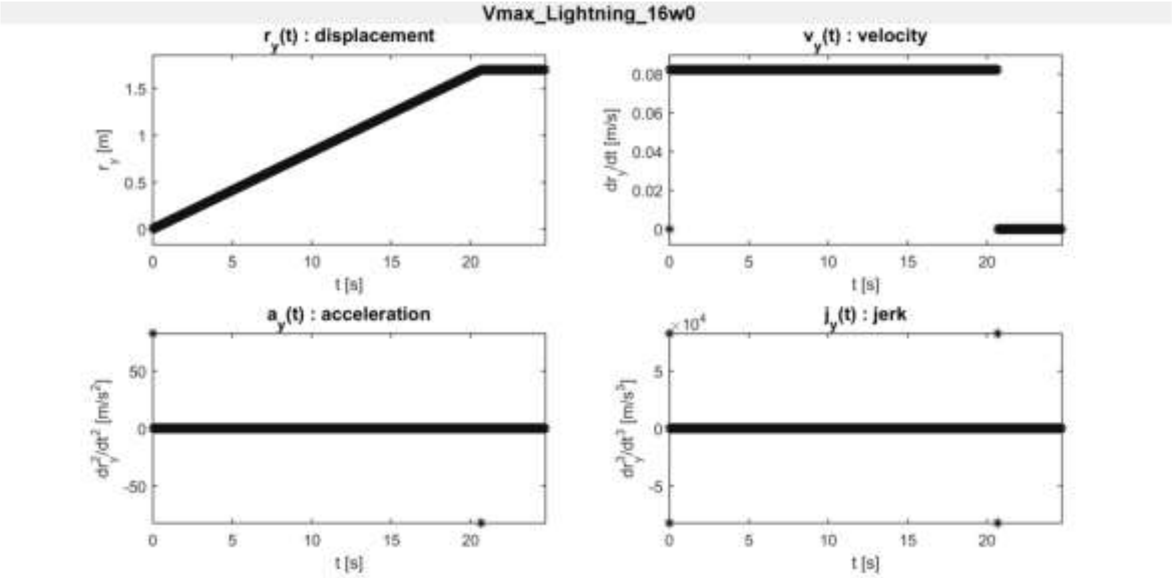


Figure 26.
Instantaneous v_{max} trajectory components.

For moving a body of mass m, such a time optimal trajectory, where an instant jump-start is demanded with v_{max} maximum speed from point A, and then it is planned to have an abrupt stop to zero velocity at time t_f in point B is, is just physically not feasible; obviously, there must be an acceleration and a deceleration period – we cannot physically generate mathematical Dirac force impulses having extremely precise integral value. This realization was adopted in „bang-bang" trajectory plans – see Figure

27, where the first part of the trajectory was planned for a_{max} constant maximal acceleration until reaching v_{max}, then traveling with v_{max} for the appropriate distance, and finally decelerating with a constant maximal deceleration $-a_{max}$ to reach the target. Notice the jerk Dirac impulse of *10m/s*^{3} despite a very modest speed and barely existing acceleration levels.

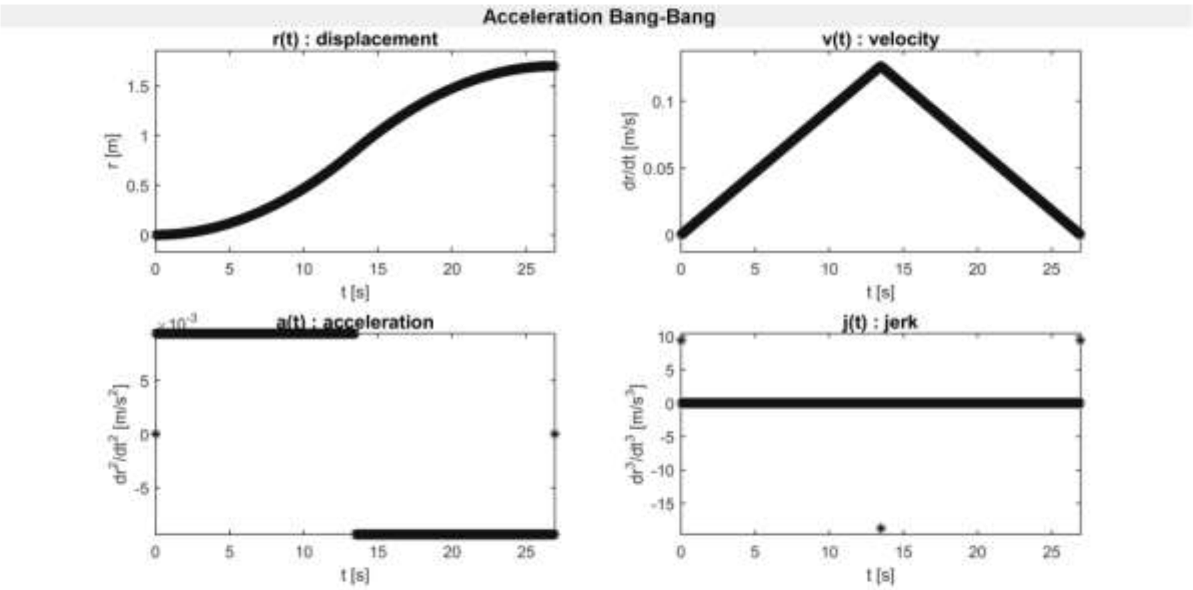


Figure 27.
Acceleration bang-bang trajectory components.

These “bang-bang” trajectories were dubbed „time optimal” trajectories, though obviously for $a_{max} = v_{max}/(t_f/2)$ we get $\sqrt{s/a_{max}} > s/v_{max}$, so actually they are no longer ‘theoretically optimal’ in the pure mathematical sense.

My comment after analysing Figure 27 is: notice the Dirac impulse of *10m/s*^{3} in the jerk component at t_0, t_f and an even larger at time instance $t_f/2$. These „bang-bang” trajectories imply that we plan for using discontinuous force actions, as we have initially $F(t_0) = 0$, then in the next infinitesimally close time $F(t_0 + \varepsilon) = F_{max} = m \cdot a_{max}$; also when reaching the target this trajectory design plans for an instant drop from $F(t_f - \varepsilon) = -F_{max}$ to $F(t_f) = 0$. Even a more significant discontinuity is planned for $F(t_f/2 - \varepsilon) = F_{max}$ and then $F(t_f/2 + \varepsilon) = -F_{max}$, which is also not feasible in real life dynamics of non-rigid bodies, even if undergraduate studies comfortably operate with similar kinematic models. So I claim that these, and many other ‘optimal’ trajectories – as I will show – are not ‘feasibly optimal’ in the engineering sense, as no actuator coupled system can precisely track them.

Please, take a step back, and analyse what are control engineers actually trained for: when control engineering faces these ‘optimal’ trajectories – or any other not feasible trajectory, the best what can be done is to apply fast and strong enough control loops so that the controlled system transient period is acceptably small, while the overshoot and the settling time also remains “controlled” – as obviously, these trajectories induce vibrations, significant system state oscillations. When analysing such control signals well noticeable are the immense energy spikes used for achieving, or at least trying to achieve these fast, discontinuous transients, compensating for overshoots and the resulting vibrations, oscillations after the rise time – be it mechanical or electrical in nature. These effects are unwanted – they increase wear and reduce the life span of physical systems, in applications like crane transport the oscillations are just uncomfortable and reduce the task duration, in certain applications like remote brain surgery end effector oscillations are absolutely not acceptable, and in extreme cases of high speed vehicles they are source of catastrophic accidents.

A lot of research effort has been put into studying and dampening, controlling vibrations. Vibrations are highly undesirable for any precise path tracking. Be it cranes or robotic manipulators (RMs) – the lowest vibration levels were identified to correspond to the magnitude of the second time derivative of the induced torque, while an appropriately constructed trajectory can decrease oscillations and energy consumption by a factor of 10, – as presented in [73].

Soon after introducing optimal trajectory planning, it had been realised that also the control effort can and should be minimised. First minimum acceleration trajectories had been devised aiming for a minimal force action, later special cost functions had been adopted to consider the direct control energy effort. In [58] Pontryagin defined the mathematical theory of optimal processes. To minimize the total used energy E_{t_f} of moving mass m from A to B, a cost function like $E_{t_f} = \int_0^{t_f} \boldsymbol{P_{abs}}(t)dt$ is devised by cumulating the absolute value of the instantaneous applied power $\boldsymbol{P_{abs}}(t) = |\boldsymbol{F}(t)\boldsymbol{v}(t)| + |\boldsymbol{\tau}(t)\boldsymbol{\omega}(t)|$ through which the product of the absolute acceleration and velocity function profile of the shortest geometric path is minimized. Since $\boldsymbol{P_{abs}}(t) = \sum_i m_i \cdot |\boldsymbol{a_i}(t)| \cdot \boldsymbol{v_i}(t)$ this minimization process is like looking for the minimal acceleration trajectory, which results in a polynomial trajectory of order 3, with a discontinuity in acceleration for $t = 0$ and $t = t_f$.

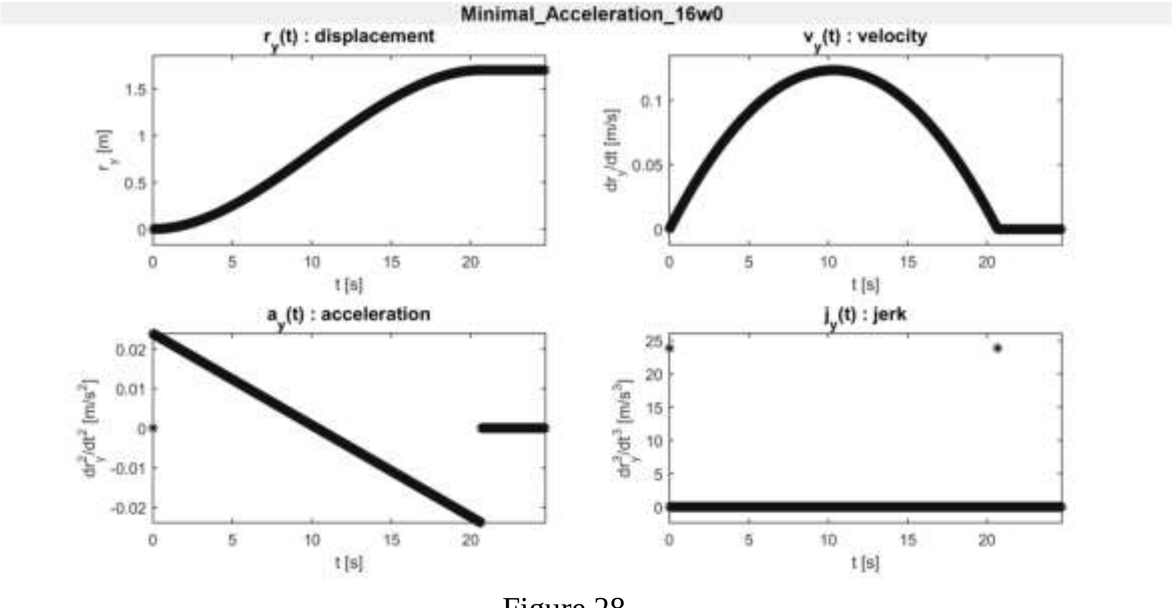


Figure 28.
Minimal acceleration trajectory components.

A minimal acceleration trajectory is presented in Figure 28 - notice the considerable jerk Dirac impulse of *24m/s*^{3} despite a very modest speed and acceleration levels.

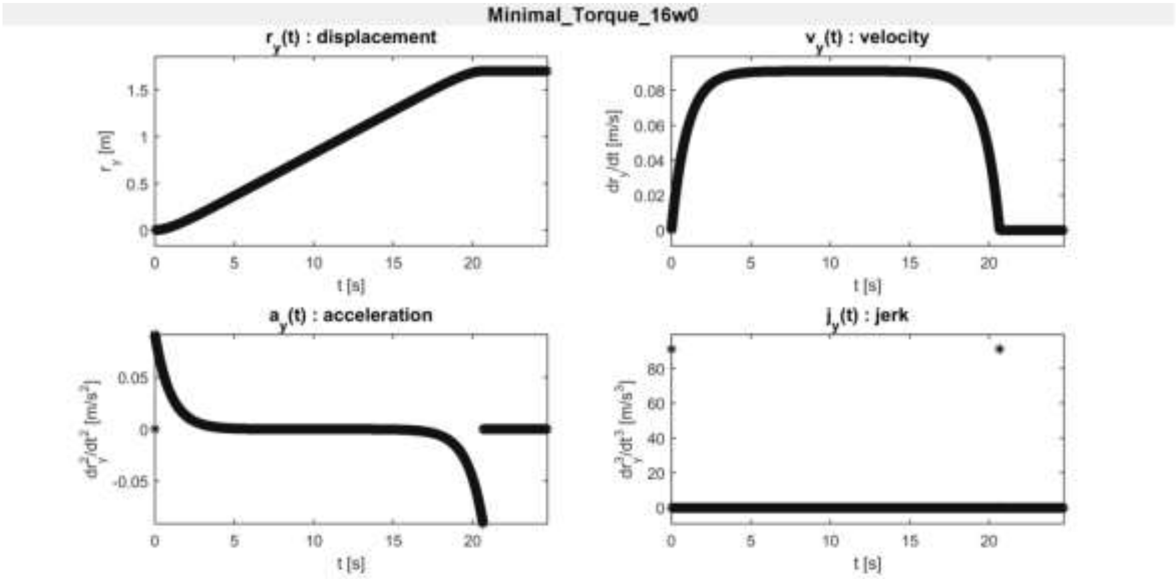


Figure 29.
Minimal torque trajectory components.

A minimal torque trajectory is presented in Figure 29 - notice the high jerk Dirac impulse of *90m/s*^{3} despite a very modest speed and acceleration levels.

Despite these advanced optimal trajectories system vibrations remained to be a substantial issue and targeted by myriads of research effort. Research had pointed out that vibrations are in correlation with the jerk. Experiments on train passenger comfort have proven that jerk levels higher than *3m/s*^{3} are already unacceptable [82].

The first time derivative of acceleration, which is the third time derivative of the displacement is called jerk $\boldsymbol{j} = d^3\boldsymbol{s}/dt^3 = \dddot{\boldsymbol{s}}(t)$. Polynomial trajectories of order 5 can be designed as minimum jerk trajectories – as in Figure 30. The construction of minimal jerk trajectories, like for the previously mentioned minimum acceleration and minimum energy trajectories, is by minimization of a cost function; in case of minimum jerk, it is: $\boldsymbol{C}(s) = \frac{1}{2}\int_0^{t_f}\dddot{\boldsymbol{s}}(t)^2dt$.

Calculus of variation or Hamiltonian with Lagrange functions is one of the common tools to solve this mathematical problem, where a perturbation function $\boldsymbol{\delta}(t)$ is added with a constant multiplier α in the form of $\boldsymbol{s}(t) + \alpha\cdot\boldsymbol{\delta}(t)$, such that for boundary conditions the perturbation and its derivatives are 0 like $\boldsymbol{\delta}(x) = \dot{\boldsymbol{\delta}}(x) = \ddot{\boldsymbol{\delta}}(x) = \dddot{\boldsymbol{\delta}}(x) = 0$ for $x = 0$ and $x = t_f$. Based on calculus of variations, instead of minimizing the function $\boldsymbol{C}(\boldsymbol{s}(t) + \alpha\cdot\boldsymbol{\delta}(t)) = \frac{1}{2}\int_0^{t_f}(\dddot{\boldsymbol{s}}(t) + \alpha\cdot\dddot{\boldsymbol{\delta}}(t))^2dt$, the zero point of its partial derivative for $\alpha = 0$ is calculated like $\partial\boldsymbol{C}(s + \alpha\delta)/\partial\alpha|_{\alpha=0} = \int_0^{t_f}\left(\dddot{\boldsymbol{s}}(t) + \alpha\cdot\dddot{\boldsymbol{\delta}}(t)\right)\cdot\dddot{\boldsymbol{\delta}}(t)dt\Big|_{\alpha=0} = \int_0^{t_f}\dddot{\boldsymbol{s}}(t)\cdot\dddot{\boldsymbol{\delta}}(t)dt$.

Using boundary conditions the result is $-\int_0^{t_f}\boldsymbol{s}^{(\boldsymbol{6})}(t)\cdot\boldsymbol{\delta}(t)dt = 0$, which per the calculus of variations is equivalent to the requirement of having the sixth derivative of the displacement equal to zero: $\boldsymbol{s}^{(\boldsymbol{6})}(t) = 0$. Knowing all the boundary conditions of the trajectory $\boldsymbol{s}(x) = s_0, \dot{\boldsymbol{s}}(x) = v_0, \ddot{\boldsymbol{s}}(x) = a_0$ at $x = 0$, and similarly for $x = t_f$; the a, b, c, d, e parameters can be calculated as a polynomial trajectory $\boldsymbol{s}(t) = a + bt +$

$ct^2 + dt^3 + et^4 + ft^5,\ \dot{\boldsymbol{s}}(t) = b + 2ct + 3dt^2 + 4et^3 + 5ft^4,\ \ddot{\boldsymbol{s}}(t) = 2c + 6dt + 12et^2 + 20ft^3$. A minimal jerk trajectory is presented in Figure 30.

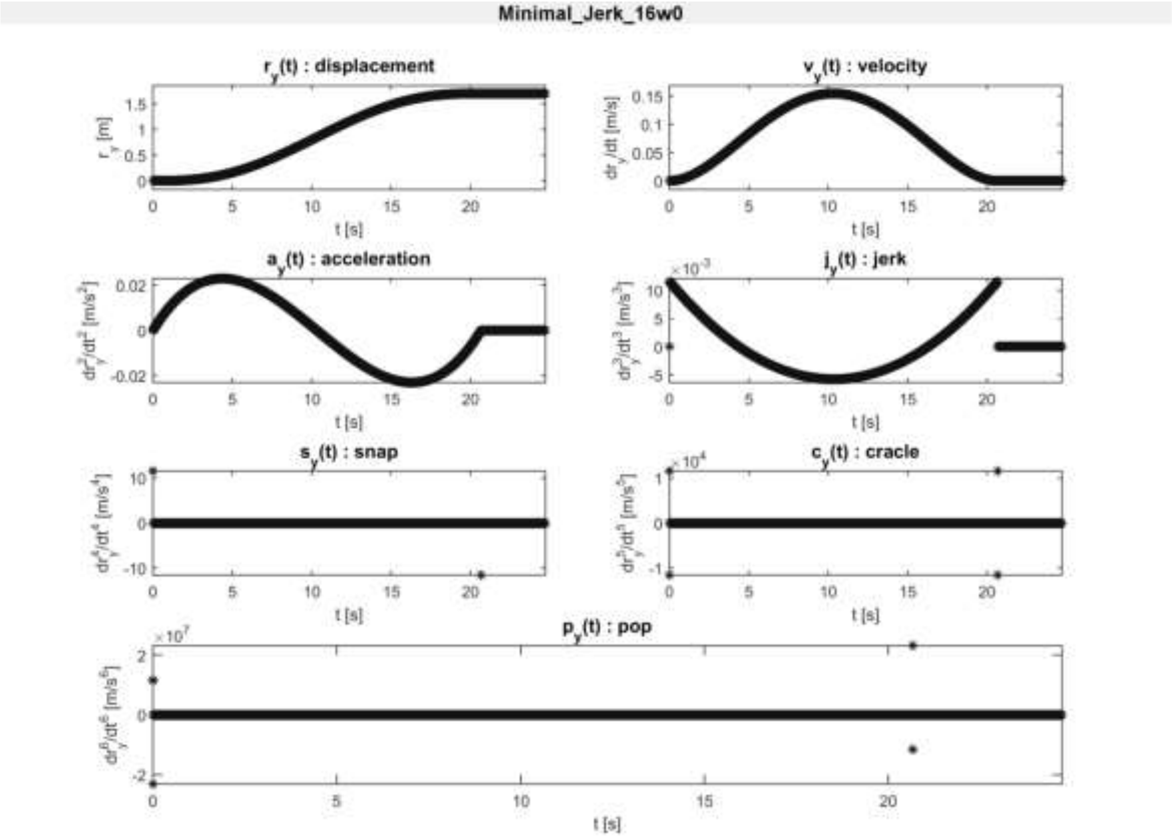


Figure 30.
Minimal jerk trajectory components.

For such trajectories jerk $\boldsymbol{j}(t_0) = \dddot{\boldsymbol{s}}(t = 0) = 6d$, obviously starts with an instantaneous jump from 0 to $6d$, also at the final moment $t = t_f$ there is also a discontinuous a jump from a non-zero to 0 value of the jerk. As we have previously discussed: this induces oscillations. A sudden jerk induces vibrations; in case of vehicles like elevators, high speed trains, roller coasters the ride is very uncomfortable at those points.

5.1.2 Multi-Rotor Unmanned Aerial Vehicle Flight Trajectory

Multi-rotors like quad- and hexa-rotors are popular representatives of UAVs as they are relatively simple to build and easy to control, while being of versatile applicability, capable of vertical take-off and landing. Also the multi-rotor architecture has simple mechanics, high relative payload capability and good manoeuvrability. The study of multi-rotor kinematics and dynamics is based on the physics of aerial platforms [68].

The kinematics with the general force and torque dynamics of any symmetric multi-rotor (quad- or hexa- rotors) is the equivalent 6 DOF dynamic system of mass *m* moved against the gravity acceleration *g*. Generalised translational forces: $m(\ddot{\boldsymbol{\xi}} + g[0\ 0\ 1]^T) = \boldsymbol{F}_\xi$; and the generalised body torques are: $\boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{\tau_B}$, where in analogy with robotic manipulators: $\boldsymbol{J}\ddot{\boldsymbol{q}}$ is the inertia matrix; $\boldsymbol{C}\dot{\boldsymbol{q}}$ is the Coriolis term;and the state vector $\boldsymbol{q}$ is composed of the Euler angles for roll, pitch and yaw $\boldsymbol{q} = [\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi}]$.

The roll and pitch of a multi-rotor UAV can be calculated from the path curve vector function as *($\boldsymbol{x}(t)$,$\boldsymbol{y}(t)$,$\boldsymbol{z}(t)$)* and the required yaw motion *$\psi(t)$* as presented in [35] like:

$$\boldsymbol{\phi} = asin\left(\frac{\dot{x}sin\boldsymbol{\psi}-\dot{y}cos\boldsymbol{\psi}}{\ddot{\boldsymbol{x}}^2+\ddot{\boldsymbol{y}}^2+(\ddot{\boldsymbol{z}}+g)^2}\right), \boldsymbol{\theta} = atan\left(\frac{\dot{x}cos\boldsymbol{\psi}-\dot{y}sin\boldsymbol{\psi}}{(\ddot{\boldsymbol{z}}+g)}\right) \tag{54}$$

Equation (54) highlights that compared to RMs, 3D cranes and any directly actuated system, multi-rotor UAVs introduce yet another layer of complexity: their torque dynamics is similar to RMs, while they are propelled by a lift force of rotating blades fixed to the body $\vec{z}$ axes. Paths $[\boldsymbol{\xi}(t), \boldsymbol{\mu}(t)] = \left[\left(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{z}(t)\right), \boldsymbol{\psi}(t)\right]$ defined along earth coordinate $\vec{x}, \vec{y}, \vec{z}$ axes extended with the desired yaw rotation angle $\psi(t)$ translate to body rotation coordinates $\boldsymbol{\mu}(t) = \left(\boldsymbol{\varphi}(t), \boldsymbol{\theta}(t), \boldsymbol{\psi}(t)\right)$ as defined by equation (54) where $\boldsymbol{\varphi}(t) = \boldsymbol{\varphi}\left(\ddot{\boldsymbol{x}}(t), \ddot{\boldsymbol{y}}(t), \ddot{\boldsymbol{z}}(t)\right)$, and $\boldsymbol{\theta}(t) = \boldsymbol{\theta}\left(\ddot{\boldsymbol{x}}(t), \ddot{\boldsymbol{y}}(t), \ddot{\boldsymbol{z}}(t)\right)$, which means $\boldsymbol{\mu}(t) = \boldsymbol{\mu}\left(\ddot{\boldsymbol{\xi}}(t)\right)$.

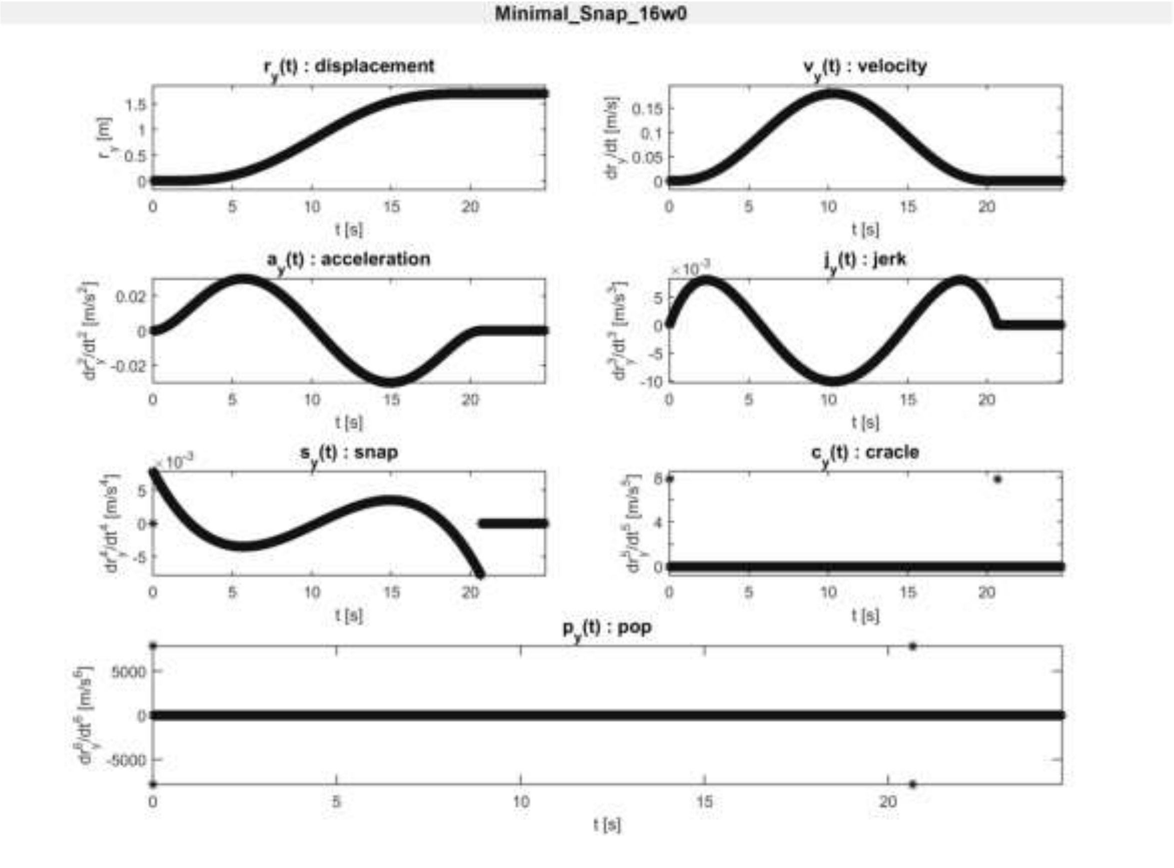


Figure 31.
Minimal snap trajectory components.

Thus, from the torque equation we can conclude $\boldsymbol{\tau_B}(t) = \boldsymbol{\tau_B}\left(\boldsymbol{\xi}^{(4)}\right)$, where $\boldsymbol{\xi}^{(4)} = d^4\boldsymbol{f}(\boldsymbol{t})/dt^4$ is the fourth time derivative of the displacement curve vector function ***f(t)***. This means that we have the body torque being a function of the displacement snap, the fourth time derivative of the displacement. This is the point where [37] draws the conclusion to use minimum snap trajectories. Minimum-snap polynomial (***x,y,z***) trajectories are proven good for quadrotors, “since the motor commands and attitude accelerations of the vehicle are proportional to the snap, the fourth time derivative, of the path” [37]. A minimal snap trajectory is presented in Figure 31; for the sake of some further analysis, please notice the snap discontinuity at t_0 and t_{final} time instances.

Notice that the control signal (commonly the torque output of an electric motor) is not a virtual mathematical quantity to be optimised without constraints. The actuators are real physical systems of studied characteristics, dynamical properties and physical limitations.

Researches like [81] and [48] are pointing out that applying input shaping instead of direct step change in the control signal, for example in BLDC rotor speed control, results in both the unwanted oscillations reduction and the energy consumption reduction; also the responsiveness of the system can be increased - when there are no current spikes, much less energy used [74].

There are multiple approaches for multi-rotor UAV trajectory planning, starting from simple path plotting up to complete trajectory generation: [5], [49], [65].

[51] describes the possibility of defining the major path milestones by visual fuzzy servoing, also any map based three search algorithms can be applied to define the next major target point during a flight mission.

To facilitate both time and energy efficiency of flight the major path milestones are best connected with continuous curvature functions ***f(s)*** [22].

5.1.3 Electric Motors as Actuators

For BLDC actuators it is common to merely associate the torque with the electric current in the actuator $\boldsymbol{i_e}(t) \approx const_2 \cdot \boldsymbol{\tau_{Bi}}(t)$. Whereas the complete electrical equation of an electric motor is:

$$\boldsymbol{v_e}(t) = L_e \frac{\boldsymbol{di_e}}{\boldsymbol{dt}}(t) + R_e \boldsymbol{i_e}(t) + K_b \boldsymbol{\omega}(t) \tag{55}$$

where $\boldsymbol{v_e}(t)$ is the armature voltage; $\boldsymbol{i_e}(t)$ is the armature electric current, $\boldsymbol{\omega}(t)$ is the rotation velocity of the rotor; L_e, R_e are the electrical inductance and resistance of the armature; K_b is the back EMF parameter.

The even more determinative characteristic of an electro motor actuator is defined by the torque equation as:

$$K_\tau \boldsymbol{i_e}(t) - \boldsymbol{\tau_L}(t) = J_M \frac{\boldsymbol{d\omega}}{\boldsymbol{dt}}(t) + \gamma_M \boldsymbol{\omega}(t), \tag{56}$$

where $\boldsymbol{\tau_L}$ is the load torque; K_τ is the torque constant; J_m is the motor inertia; and γ_M is the rotor friction coefficient.

The motor load torque for a rotor blade propeller application is:

$$\boldsymbol{\tau_L}(t) = J_R \frac{\boldsymbol{d\omega}}{\boldsymbol{dt}}(t) + K_d \boldsymbol{\omega}^2(t). \tag{57}$$

where K_d is the rotor drag parameter and J_R is the rotor blade inertia. Further on from equations (55)-(57) we can conclude:

$$\frac{\boldsymbol{di_e}}{\boldsymbol{dt}}(t) = \frac{1}{K_\tau}\left((J_M + J_R)\frac{\boldsymbol{d^2\omega}}{\boldsymbol{dt^2}}(t) + (\gamma_M + 2K_d \boldsymbol{\omega})\frac{\boldsymbol{d\omega}}{\boldsymbol{dt}}(t)\right), \tag{58}$$

For electro-motor torque actuated mechanical systems like common cranes, RMs or multi-rotor UAVs cost functions like $E_{t_f} = \int_0^{t_f} \boldsymbol{\tau_{abs}}(t)dt$ are also used to minimize the total torque, the used actuator electric energy; as in a stationary state of the actuators it is common to use a simplified motor torque model, where the applied current is linearly proportional to the resulting torque $\boldsymbol{i_e}(t) \cong const \cdot \boldsymbol{\tau}(t)$ [3].

5.1.4 Often Neglected System and Actuator Characteristics, Effects of Infeasible Trajectories

Anti-swing control of cranes is still a popular research subject, since their construction simplicity, thus availability along wide industrial practical applicability. Very advanced control mechanisms, including input shaping have been studied; the primary goal is to move the pendulum like system efficiently, keeping the pendulum system oscillations at minimum, so that the payload mass swing – trajectory tracking error is at minimum [11].

The dynamic model of an RM or even that of a 3D crane [11] has a quite complex torque equation like $\boldsymbol{\tau}(\mathrm{t}) = \boldsymbol{H}(\boldsymbol{q}) \cdot \ddot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T \cdot \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \cdot \dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q})$, with highly nonlinear $\boldsymbol{H}(\boldsymbol{q}), \boldsymbol{C}(\boldsymbol{q}), \boldsymbol{G}(\boldsymbol{q})$ functions, where joint variables $\boldsymbol{q} = \boldsymbol{q}(s)$ themselves are not trivially deducted through the systems construction geometry along the end effector path $\boldsymbol{s}$, so solutions to energy optimal trajectories come only either in flavours of drastic reductions to linear approximations of the RM dynamics, or in flavours of numerical iterative methods. In the end linear approximation results for RMs are in general sub-optimal, while numerical iterative solutions are far from real time usability.

Even for the simplest direct electro-motor actuator considering the armature electric current $\boldsymbol{i_e}(t) \cong const_1 \cdot \boldsymbol{\tau}(t) = const_2 \cdot \boldsymbol{a}(t)$ model, the discontinuity in jerk $\boldsymbol{j}(t) = \dot{\boldsymbol{a}}(t) = d\boldsymbol{a}(t)/dt$ means a discontinuity in $\dot{\boldsymbol{i_e}}(t) = d\boldsymbol{i_e}(t)/dt$ and by equation (55) that would translates to armature voltage $\boldsymbol{v_e}(t)$ discontinuity, which is not possible in real life.

Approaching from a different direction we can conclude the same: if we accept the paradigm of not planning for bang-bang trajectories because of discontinuity of the third derivative of the linear displacement ($\dot{\boldsymbol{a}}(t)$), we have to accept the same for the rotation displacement ($\dot{\boldsymbol{\omega}}(t)$) as well. By equation (58) we conclude that armature current change $\dot{\boldsymbol{i_e}}(t)$ is continuous and then by (55) $\boldsymbol{v_e}(t)$ is also continuous.

Vibration levels in RMs have been identified as induced even by discontinuities in the second derivative of the torque. Now this must not come as a surprise, being aware of these described voltage and current discontinuity circumstance and equation (58), we can conclude that $\ddot{\boldsymbol{\tau}}(t) \cong const_1 \cdot \ddot{\boldsymbol{i}}_{\boldsymbol{e}}(t) \cong const_2 \cdot d^2(\boldsymbol{\omega}(t)^2)/dt = const_2 \cdot (2\dot{\boldsymbol{\omega}}^2 + 2\boldsymbol{\omega}\ddot{\boldsymbol{\omega}})$. We have the angular position of the actuator rotor, which is a real physical body with mass that cannot be accelerated in a discontinuous manner.

The applied mechanical force or torque $\boldsymbol{M_B}(t) \approx m * \ddot{\boldsymbol{\mu}}(t)^2$ excreted onto the body is proportional to the second time derivative of the linear position or rotation angle $\ddot{\boldsymbol{\mu}}(t)$ of the body. As the body is driven by a rotor blade, $\boldsymbol{\omega}(t)$ is proportional to $\ddot{\boldsymbol{\mu}}$, the body angular acceleration. We must be open to conclude that in reality no discontinuities can physically occur, not even in the fourth time derivatives of a displacement neither for the controlled system, nor for the control actuator – see equations (55), (58).

My major point is that for limiting the actuator torque vibrations, it must not be considered only as a mathematical problem of limiting $|\dddot{\boldsymbol{q}}|$, the absolute value of the jerk. It is time to realize that in trajectory planning not the extent of discontinuity in a physical quantity is the problem, but the existence of such discontinuity itself is unacceptable or at least a sub-optimal approach.

I would like to highlight that in [37] the rotor blade velocity is considered as an arbitrary control input. As 7th order minimum-snap polynomial trajectories are discontinuous in displacement crackle, the fifth time derivative of displacement, my claim is that this is still a sub-optimal approach. The rotor blade velocity is not an arbitrary theoretical control signal, but a real, electro-mechanical physical system, subject to aero dynamical load conditions.

We can directly conclude that what can be realised for RMs, the same applies to UAVs: the second time derivative of the actuator torque has to be continuous. From equations (43) and (54) this requirement is equivalent to having a continuous displacement pop, $p(t) = \boldsymbol{\xi}(t)^{(6)}$ the sixth time derivative of the $\left[\left(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{z}(t)\right)\right]$ displacement. This realisation of mine is the baseline for my fourth hypothesis.

By my *Hypothesis VI* this paper presents a novel direct path construction real-time algorithm for generating physically feasible, time-and energy optimal, bounded, continuous trajectories that can reach any target displacement with a known minimal error. These trajectories can be designed to arbitrary smoothness – depending on system requirements; they are to be designed smooth up to the 5th time derivative of displacement for multi-rotor UAV trajectories. The term (n times) smooth is used as in being equivalent to having continuous (nth) time derivative.

5.1.5 Validating Quality of System Trajectory

The system path can be defined as a parametrised vector function ***f(s)***, where the nature of parametrisation defines the trajectory characteristics as $\boldsymbol{q} = \boldsymbol{f}(s), \dot{\boldsymbol{q}} = \boldsymbol{f}'(s)\dot{s}, \ddot{\boldsymbol{q}} = \boldsymbol{f}''(s)\dot{s}^2 + \boldsymbol{f}'(s)\ddot{s}$. For these systems, the primary trajectory design challenge is to find such efficient parametrisations that the required system constraints are fulfilled.

The defined boundary conditions of the trajectory, the limits of $\boldsymbol{q}$ displacement time derivatives have to be satisfied. The corresponding limits on maximum values for time derivatives of the displacement parametrisation $\boldsymbol{s}$ have to be obeyed. Continuity and smoothness of every trajectory component $\boldsymbol{s}$ has to be ensured up to the predetermined order (for $\frac{d^5s}{dt^5}$; which is 5 in case of multi-rotors). The term (n times) smooth is used as in being equivalent to having continuous (nth) time derivative.

5.1.5.1 Multi-rotor Test Flight Trajectory

To evaluate the effect of a trajectory to the multi-rotor flight dynamics we will observe plots of the roll and pitch angle, the resulting body torques, the required rotor angular velocities and accelerations. Effects of discontinuities in different time derivatives will be presented. A comparison will be presented for the proposed new harmonic trajectory and the classical trajectories.

The term (n times) smooth is used as in being equivalent to having continuous (*nth*) time derivative. For any signal n times smoothness can be validated by excluding the occurrence of discontinuities in the *(n+1)*st time derivative or even more obviously by excluding occurrence of impulses in the *(n+2)*nd or higher time derivatives.

5.1.5.2 3D Overhead Crane Test Trajectory

3D cranes are the most obvious tools to present systems state oscillations. The most simple and obvious way to present trajectory induced system oscillations is to analyse an up-scaled crane system – with a longer pendulum length, so that all oscillation sings are notably present as either the payload trajectory tracking error or in the state space rate of change; for a crane it is the second time derivative of payload position that will magnify the presence of trajectory induced oscillations in a feed forward control system. The used benchmark 3D overhead crane is described in [11].

Anti-swing control of cranes is still a popular research subject, since their construction simplicity, thus availability along wide industrial practical applicability. Very advanced control mechanisms, including input shaping have been studied; the primary goal is to move the pendulum like system efficiently, keeping the pendulum system oscillations at minimum, so that the payload mass swing – trajectory tracking error is at minimum [11].

Crane system state oscillation effects will be presented for the proposed new harmonic trajectory and the classical trajectories. Trajectory feasibility and oscillatory properties of a crane system are most obviously presented by inspecting the actual system state variables and their planned trajectory – for an optimal feasible trajectory the actual system variable changes will match the planned state trajectory, obviously without any unwanted, unplanned oscillations.

5.2 New Scientific Achievements

5.2.1 New Feasible Optimal Harmonic Trajectories of Bounded, Smooth Time Derivatives

In [37] the rotor blade velocity is considered as an arbitrary control input. As 7th order minimum-snap polynomial trajectories are discontinuous in displacement crackle, fifth derivative of displacement, my claim in [s14], [s15] is that this is still a sub-optimal approach; again: the rotor blade velocity is not an arbitrary theoretical control signal, but a real, electro-mechanical physical system, subject to aero dynamical load conditions.

The goal of this paper is to present a new method for flexible and efficient real-time direct path parametrisation, which is capable of generating physically feasible, time-and energy optimal, bounded, continuous trajectories with minimal induced oscillations; a method even usable for autonomous navigation. The notion of time and energy optimality is not used in mathematics theory manner, but in real life, physically feasible engineering manner [s14], [s15].

The process of finding optimal trajectories is in this paper focused on finding the appropriate parametrisation for the path vector function ***f(t)***, given the pre-defined feasibility limits on the displacement time derivatives, in conjunction with the effects of the path curvature.

The defined boundary conditions of the trajectory have to be satisfied. The defined limits on maximum values for arbitrary time derivatives of the displacement have to be obeyed.

Continuity and smoothness of every trajectory component has to be ensured up to the predetermined order: six times smooth in case of multi-rotors, four times smooth in case of cranes and RMs.

As described in [s14] and [s15], to have realistic, feasible torques along a trajectory, which are efficiently controllable without chattering, we need smooth torque changes. For indirect rotor-blade propulsion systems (ships, multi-rotors) we have the propulsion motor force or torque $\boldsymbol{M_M}(t) \approx const \cdot \boldsymbol{\omega}(t)^2$ proportional to the square of the rotor angular velocity. The applied mechanical force or torque $\boldsymbol{M_B}(t) \approx m * \ddot{\boldsymbol{\mu}}(t)^2$ excreted onto the body is proportional with the second derivative of the linear position or rotation angle $\ddot{\boldsymbol{\mu}}(t)$ of the body. As the body is driven by a rotor blade, $\boldsymbol{\omega}(t)$ is proportional to $\ddot{\boldsymbol{\mu}}$, the body angular acceleration.

In reality no discontinuities can physically occur, not even in third time derivatives of a displacement neither for the controlled system, nor for the control actuator.

For a realistic, feasible control input of direct BLDC actuated systems (RMs, cranes, wheeled vehicles) the designed path has to be such that the planned snap ($\xi^{(4)}$) must be continuous and proportional to the third derivative of the motor shaft rotational displacement . Ultimately for a feasible trajectory for the body rotation we must obey that the feasible body torque transients are proportional to the possible motor torque transients; equivalently the feasible second derivative of the body displacement $\xi^{(2)}(t)$ has to be proportional to motor shaft possible $\omega(t)$. On top of the allowed trajectory transient behaviour there are requirements on its smoothness as well. To be able to optimally control an electric motor with either $\boldsymbol{v_e}(t)$ or $\frac{\boldsymbol{di_e}}{\boldsymbol{dt}}(t)$, the $\frac{\boldsymbol{d^2\omega}}{\boldsymbol{dt^2}}(t)$ signal has to be continuous; equivalently $\xi^{(4)}$(t), snap, the fourth time derivative of body displacement has to be continuous.

After algebraic manipulations of equations (55)-(58) by Laplace transformations we can conclude:

$$I_e = \frac{V_e - K_b W}{R + Ls} \tag{59}$$

by notation $(J_M + J_R) = J, K_T = K_b = K, B = L_e J, C = R_e J + L_e \gamma_M, D = L_e K_d$ we get:

$$-Bs^2 W - sW(C + DW) = KV_e - W(K^2 + R_e \gamma_M + R_e K_d W), \tag{60}$$

where the right hand side represents the stationary mode of the electric motor, and the left hand side represents the dynamic transient mode. For cases of voltage control we can use directly the solution of equation (60), while for current control of the electric motor, the solution has to be substituted to equation (59). For the stationary case we can directly calculate the required $\boldsymbol{v_e}$ for an arbitrary stationary ω_{stat} by making the transient left hand side equal to zero. The solution of the left hand side, keeping the right hand side zero, defines the transient mode characteristic of the rotation angular velocity $\boldsymbol{\omega_t}(t)$ as:

$$\boldsymbol{\omega_t}(t) = -\frac{C}{D} - (\frac{AB}{D}) tanh(\frac{A}{2}(T_h - t)), \tag{61}$$

where $A = \sqrt{(\frac{C^2}{B^2} + \frac{2D}{B}E)}$, T_h; and E are constants calculated for $\boldsymbol{\omega_t}(0) = \omega_0$, $\dot{\boldsymbol{\omega}}_{\boldsymbol{t}}(0) = \omega_{d0}$ boundary conditions. For $\dot{\boldsymbol{\omega}}_{\boldsymbol{t}}(0) = \omega_{d0} = 0$ equation (61) can be presented in the form of:

$$\boldsymbol{\omega_t}(t) = \frac{\omega_{stat}}{2}\left(1 + tanh\left(\frac{\pi}{P}\left(t - \frac{P}{4}\right)\right)\right), \tag{62}$$

where ω_{stat} stands for the targeted stationary rotation speed; P is a system specific parameter describing the transient characteristic - the settling time of the transient; for the given boundary condition $E = E(\omega_0)$ we have:

$$P = \frac{2\pi}{A} = 2\pi\sqrt{\left(\frac{C^2}{B^2} + \frac{BD}{B^2}E\right)^{-1}} = \frac{2\pi L_e(J_M+J_R)}{\sqrt{(R_e(J_M+J_R)+L_e\gamma_M)^2+EL_e(J_M+J_R)L_eK_d}} \tag{63}$$

Dependency of multirotor torque and rotor blade angular velocity on the continuity of the pop function can be also demonstrated by simply calculating and plotting these system values for an artificially created step function-like trajectory pop [s14] – it is well notable that any discontinuity in the trajectory pop will result in a discontinuity in the time derivative of the required rotor angular velocity, which we have already concluded to be a physical not feasible requirement. The most important system variable time signals of such infeasible discontinuous trajectories are presented in Appendix II.

THESIS IV.a - DEFINITION:

For a **realistic, feasible general system trajectory** one must design realistic, feasible control system inputs. For direct BLDC actuated systems (RMs, cranes, wheeled vehicles) the designed path has to be such that the planned body displacement fourth time derivative, the snap ($\xi^{(4)}$) must be continuous and proportional to the third derivative of the motor shaft rotational displacement as in equation (62).

We must notice that multi-rotor UAVs introduce yet another layer of complexity: their torque dynamics is similar to RMs, while they are propelled by a lift force of rotating blades fixed to the body $\vec{z}$ axes. Paths $[\boldsymbol{\xi}(t), \boldsymbol{\mu}(t)] = \left[\left(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{z}(t)\right), \boldsymbol{\psi}(t)\right]$ defined along earth coordinate $\vec{x}, \vec{y}, \vec{z}$ axes extended with the desired yaw rotation angle $\psi(t)$ translate to body rotation coordinates $\boldsymbol{\mu}(t) = \left(\boldsymbol{\varphi}(t), \boldsymbol{\theta}(t), \boldsymbol{\psi}(t)\right)$ as defined by equation (54) where $\boldsymbol{\varphi}(t) = \boldsymbol{\varphi}\left(\ddot{\boldsymbol{x}}(t), \ddot{\boldsymbol{y}}(t), \ddot{\boldsymbol{z}}(t)\right)$, and $\boldsymbol{\theta}(t) = \boldsymbol{\theta}\left(\ddot{\boldsymbol{x}}(t), \ddot{\boldsymbol{y}}(t), \ddot{\boldsymbol{z}}(t)\right)$, which means $\boldsymbol{\mu}(t) = \boldsymbol{\mu}\left(\ddot{\boldsymbol{\xi}}(t)\right)$. Thus, from the torque equation $\boldsymbol{\tau}_{\mathbf{B}}(\mathrm{t}) = \mathbf{H}(\boldsymbol{\mu}) \cdot \ddot{\boldsymbol{\mu}} + \dot{\boldsymbol{\mu}}^{\mathbf{T}} \cdot \mathbf{C}(\boldsymbol{\mu}) \cdot \dot{\boldsymbol{\mu}} = \boldsymbol{\tau}_{\mathbf{B}}\left(\boldsymbol{\xi}^{(\mathbf{4})}\right)$ we can conclude $\boldsymbol{\tau}_{\boldsymbol{B}}(t) = \boldsymbol{\tau}_{\boldsymbol{B}}\left(\boldsymbol{\xi}^{(\mathbf{4})}\right)$, where $\boldsymbol{\xi}^{(\mathbf{4})} = d^4\boldsymbol{f}(\boldsymbol{t})/dt^4$ is the fourth time derivative of the displacement curve vector function ***f(t)***.

This means that we have the body torque being a function of the displacement snap, the fourth time derivative of the displacement. This is the point where [37] draws the conclusion to use minimum snap trajectories.

But this is not the complete picture! For multi rotors the control signal is the angular velocity of the rotor blade, which is not an arbitrary 'just a mathematical' function; it is a real physical system!

5.2.2 New Feasible Optimal Harmonic Multi-rotor Flight Trajectories

For a realistic, feasible control input of multi-rotor UAVs, we must not only consider equation (54), but also (55) and (58), so the designed UAV path has to be such that the displacement pop ($\boldsymbol{\xi}^{(6)}$) must be continuous and the body snap transient has to be feasible by a BLDC: $\boldsymbol{\xi}(t)^{(4)} \sim \boldsymbol{\omega}(t)$.

As described in [s6], [s7], [s13], [s14] to have realistic, feasible torques along a trajectory, which are efficiently controllable without chattering, we need smooth torque changes. The term (n times) smooth is used as in being equivalent to having continuous (nth) time derivative.

THESIS IV.b - DEFINITION:

For a **realistic, feasible multi-rotor trajectory** one must design realistic, feasible control system inputs, such that the planned body displacement sixth time derivative, the pop ($\xi^{(6)}$) must be continuous and proportional to the third derivative of the motor shaft rotational displacement as in equation (62).

For any trajectory we have a target position and generally a limit to the maximum velocity $\xi^{(1)} < \mathrm{V}_{\max}$ for safety. Usually there is also a limit $\left|\xi^{(2)}\right| < \mathrm{A}_{\max}$ to the acceleration and deceleration, too – either for power source capacity, constructional integrity or passenger safety reasons. The jerk is to be bounded$\left|\xi^{(3)}\right| < \mathrm{J}_{\max}$ as it has already been concluded by many researchers either to reduce structural vibrations or just for passenger comfort.

The important message of the proposal of this paper is that we also must not overlook a simple physical constraint of the actuator motor either: the rotation speed of the rotor is bounded – this limit in case of a multi-rotor vehicle is equivalent to limiting the trajectory displacement crackle $\left|\ \xi^{(5)}\right| < \mathrm{C}_{\max}$.

As already highlighted all currently named "optimal" trajectory planning methods, be it minimum time, acceleration bang-bang, minimum energy or fuel, minimum jerk, minimum torque or minimum acceleration trajectories, they all suffer from the same physical infeasibility issue: existence of supposed discontinuity in the second or third derivative of the displacement, which translates to a discontinuity target in the actuator rotor position time derivatives of second or third order. In case of a multi-rotor vehicle not even a minimum snap trajectory qualifies as physically feasible sound trajectory – in terms of targeting continuous actuator actions up to the third time derivative of the actuator rotor displacement.

It is easy to overlook these issues as we have the power of the feedback controller at our disposal. A well-tuned strong, fast feedback loop efficiently copes with system identification deficiencies and unmodelled, random perturbations, so we readily use them - and by the way we mask our physically infeasible trajectories. This is so natural an approach that the most basic tool for measuring the performance of a controller is the step response function. The goal is usually to have the fastest response with just a small overshoot, which induces only limited system vibrations lasting for a planned settling time. But we have to pay the cost of extra energy used for control – a luxury that we cannot allow for battery powered flying bodies.

The important message of the proposal of this paper based on [s6] is that we must not overlook the physical capabilities, constraint of neither the system nor the actuator itself. For multi-rotors their body torques and matching rotation speed of rotors and their transient behaviour is limited – these constraints are proportional to properties of the trajectory displacement snap $\boldsymbol{s}(t) = \boldsymbol{\xi}^{(4)}(t)$. The snap is required to be 2 times smooth, equivalent to pop $\boldsymbol{p}(t) = \boldsymbol{\xi}^{(6)}(t)$ being continuous. Also by equation (61) the transient behaviour of rotor $\boldsymbol{\omega_t}(t)$ rotation speed has to be proportional to $1 + tanh\left(\frac{\pi}{P}\left(t - \frac{P}{4}\right)\right)$.

The proposal of this paper is to use for the snap transient a base function in the form of:

$$\boldsymbol{c_t}(t) = \ \boldsymbol{\xi}_{\boldsymbol{t}}^{(5)}(t) = G \cdot \left(1 - cos\left(\frac{2\pi}{P}t\right)\right), \tag{64}$$

where P is the design parameter responsible for the trajectory duration and also the energy efficiency with oscillation avoidance, its value has to be equal or an integer multiple of the settling time of the $\boldsymbol{\omega_t}(t)$ actuator system - in case of BLDC see equation (11); and G is the design parameter by which we freely control the required displacement length for traveling any distance. By this we obtain the pop base continuous transient function as:

$$\boldsymbol{p_t}(t) = \boldsymbol{\xi_t}^{(6)}(t) = G \cdot \frac{2\pi}{P} \sin\left(\frac{2\pi}{P} t\right). \tag{65}$$

For any trajectory we have a target position and generally a limit to the maximum velocity $\xi^{(1)} < V_{max}$ for safety. Usually there is also a limit $\left|\xi^{(2)}\right| < A_{max}$ to the acceleration and deceleration, too – either for power source capacity, constructional integrity or passenger well-being reasons. For advanced projects also the jerk is to be limited$\left|\xi^{(3)}\right| < J_{max}$ as it has already been concluded by many researchers either to reduce structural vibrations or just for passenger comfort.

THESIS IV.c - DEFINITION:

A **realistic, feasible multi-rotor trajectory parametrisation** of continuous body displacement sixth time derivative pop ($\xi^{(6)}$), such that the snap ($\xi^{(4)}$) is proportional to the motor shaft rotational velocity as in equation (62) can be designed by selecting:

$$\boldsymbol{p_t}(t) = \boldsymbol{\xi_t}^{(6)}(t) = G \cdot \frac{2\pi}{P} \sin\left(\frac{2\pi}{P} t\right). \tag{65}$$

$$\boldsymbol{c_t}(t) = \boldsymbol{\xi_t}^{(5)}(t) = G \cdot \left(1 - cos\left(\frac{2\pi}{P} t\right)\right), \tag{64}$$

where P is either measured, as in equation (62), or calculated based on equation (63)

$$\boldsymbol{\omega_t}(t) = \frac{\omega_{stat}}{2}\left(1 + tanh\left(\frac{\pi}{P}\left(t - \frac{P}{4}\right)\right)\right), \tag{62}$$

$$P = \frac{2\pi}{A} = 2\pi\sqrt{\left(\frac{C^2}{B^2} + \frac{BD}{B^2}E\right)^{-1}} = \frac{2\pi L_e(J_M+J_R)}{\sqrt{(R_e(J_M+J_R)+L_e\gamma_M)^2+EL_e(J_M+J_R)L_eK_d}} \tag{63}$$

The proposal of this paper is to use for multi-rotors a parameterised single sinus wave $\boldsymbol{p}(t) = G\frac{2\pi}{P}\sin\left(\frac{2\pi}{P}t\right)$ as the base function for the displacement pop to reach the desired smooth crackle as $\boldsymbol{c}(t) = \int \boldsymbol{p}(t)dt = G\left(1 - cos\left(\frac{2\pi}{P}t\right)\right)$, which is of transient characteristics physically feasible to match by a BLDC motor. P is the period of $\boldsymbol{p}(t)$ and by this it must match the dynamics of the actuated system. G can be an arbitrary positive real value, which controls the amplitude of the pop base function and thus trajectory displacement length. The integral of a full period $\boldsymbol{c}(t)$ for *t=1..P* is to be used for the ascending part of the jerk function $\boldsymbol{j}^+(t) = \int \boldsymbol{c}(t)dt$, for simplicity we take *0* for the integral constant value. For $\boldsymbol{j}^-(t)$ descending part of jerk the integral of –**c**(t) is taken. In case that the acceleration $\boldsymbol{a(t)} = \int\left(\boldsymbol{j}^+(t) + \boldsymbol{j}^-(t+P)\right)dt$ does not reach the desired level, a constant $\boldsymbol{j_{max}}$ interval is to be inserted between $\boldsymbol{j}^+$ and $\boldsymbol{j}^-$ intervals. The velocity is planned in an analogous manner, by integrating the rising acceleration and the falling deceleration interval, with optional inclusion of a constant acceleration interval to reach the desired maximum velocity, all this without overshooting the reached acceleration limit. By keeping the velocity constant in the middle of the trajectory we ensure feasible time optimally reaching the desired displacement without exceeding the speed limit.

5.2.3 Implementation of the New Feasible Optimal Harmonic Multi-rotor Trajectories of Bounded, Smooth Time Derivatives

A basic smooth trajectory parametrization curve can be defined with a realistic sampling time *dt = 0.001*[s]. A feasible trajectory with dynamic boundary velues can be chosen to have *maximum_snap = 1*[m/s^4], *maximum_jerk = 1*[m/s^3], *maximum_acceleration = 2*[m/s^2], *maximum_velocity = 8*[m/s], *displacement = 64*[m], for displacement *duration = 16*[sec].

This base trajectory parametrisation can be projected to the training path of connecting back and forth the centre and the right most and left most opposite corners along the main diagonal ***(x,y,z)*** *= (0,0,0) -> (64,64,64), (64,64,64) -> (0,0,0), (0,0,0) -> (-64,-64,-64), (-64,-64,-64) -> (0,0,0)* of a 128m cube, while for each segment of flight performing simultaneous a full yaw rotations in each direction ***ψ*** *= (0->2π), (2π->0)*.This test trajectory setup will result in a training data set worth of 68 seconds of flight time.

The algorithm to generate feasible optimal harmonic trajectory of bounded, smooth time derivatives is:

0. Account for all defined limitations in snap, jerk, acceleration, velocity, displacement, duration, for each calculate the boundary consequence on each higher derivative:

   a. *limit_snap* directly limits:

      limit_P=limit_snap/G;

   b. *limit_jerk limits:*

      limit_P=sqrt(limit_jerk/G) and

      *limit_snap=G*limit_P*;

   c. *limit_acceleration limits:*

      *limit_P=nthrooth(limit_acceleration/(2*G),3)* and

      *limit_snap=G*limit_P* and

      limit_jerk=G(limit_P)*^{2}

   d. *limit_velocity limits:*

      *limit_P=nthrooth(limit_velocity/(8*G),4)* and

      *limit_snap=G*limit_P* and

      limit_jerk=G(limit_P)*^{2} and

      *limit_acceleration=G*2*(limit_P)*^{3}

   e. *limit_displacement limits:*

      *limit_P=nthrooth(limit_velocity/(64*G),5)* and limit_snap=G*limit_P and

limit_jerk=G*(limit_P)^2 and

limit_acceleration=G*2*(limit_P)^3 and

limit_ velocity =G*8*(limit_P)^4

f. *limit_duration limits:*

limit_P=limit_duration/16

1. select the minimal calculated limit of all above calculated values for each derivative: *limit_P, limit_snap, limit_jerk, limit_acceleration, limit_velocity,* values calculated in step 0.

2. calculate base feasible values:

   base_P = min([limit_P]) from step 1.

   base_cracle = G*2

   base_snap = base_P

   base_jerk = base_snap*base_P

   base_acceleration = base_jerk*2*base_P

   base_velocity = base_acceleration*4*base_P

   base_displacement = base_velocity*8*base_P

   base_duration = 16*base_P

3. select the smallest admissible *allowed_P*, *allowed_snap, allowed_jerk, allowed_acceleration*, *allowed_velocity* from limit and base values calculated in steps 1 and 2.

4. calculate final trajectory parameters

   a. P = allowed_P

   b. max_cracle = G*2

      l_cracle = P/2

   c. max_snap = max_cracle * l_cracle

      l_snap = 2 * l_cracle

   d. increment_jerk = max(0, allowed_jerk/max_snap – l_snap)

      max_jerk=mas_snap * (l_snap + increment_jerk)

      l_jerk=2 * l_snap + increment_jerk

   e. increment_acceleration=max(0, allowed_acceleration/max_jerk – l_jerk)

      max_acceleration = max_jerk * (l_jerk + increment_acceleration)

      l_acceleration = 2 * l_jerk + increment_acceleration

f. *increment_velocity = max(0, allowed_velocity/max_acceleration – l_acceleration)*

   *max_velocity = max_acceleration * (l_acceleration + increment_velocity)*

   *l_velocity = 2 * l_acceleration + increment_velocity*

g. *increment_displacement = max(0, allolwed_displacement/max_velocity – l_velocity)*

   *max_displacement = max_velocity * (l_velocity + increment_displacement)*

   *l_displacement = 2 * l_velocity + increment_displacement*

h. *increment_duration = max(0, target_duration – l_displacement)*

*max_duration = 16 * P + 8 * increment_jerk + 4 * increment_acceleration + 2 * increment_velocity + increment_displacement + increment_duration*

Notice that for numerical robustness in discrete calculations one must take care of each variable *l_#* and *increment_#*, and also values *#_P*, as they have to be integer multiples of the sampling time, otherwise the required exact balance between positive and negative areas of acceleration and higher derivatives cannot be ensured. The discrete algorithm thus cannot generate trajectories to arbitrary displacement with absolute precision. The generated displacement final position error is proportional to *numerical_endposition_error = 16 * P - integer(16 * P / sampling_time) * sampling_time*.

Such a general basic 16 intervals, smooth sinusoid pop function trajectory setup *with P=1*, and all its corresponding displacement derivatives are presented in Figure 32. The basic smooth trajectory parametrization curve used is with *P = 1*, with sampling time *dt = 0.01[s]* sampling time, which results in a time optimal trajectory with dynamic boundaries of maximum of sinus *pop* = 2π *[m/s^6]* wave of 1[s] period, maximum of *crackle = 2[m/s^5], maximum_snap = 1[m/s^4], maximum_jerk = 1[m/s^3], maximum_acceleration = 2[m/s^2], maximum_velocity = 8[m/s], displacement = 64[m]*, within *duration = 16[sec]*. The integral of absolute jerk is 8[m/s^2], what is proportional to the expended energy (as mass and desired displacement we consider to be constant).

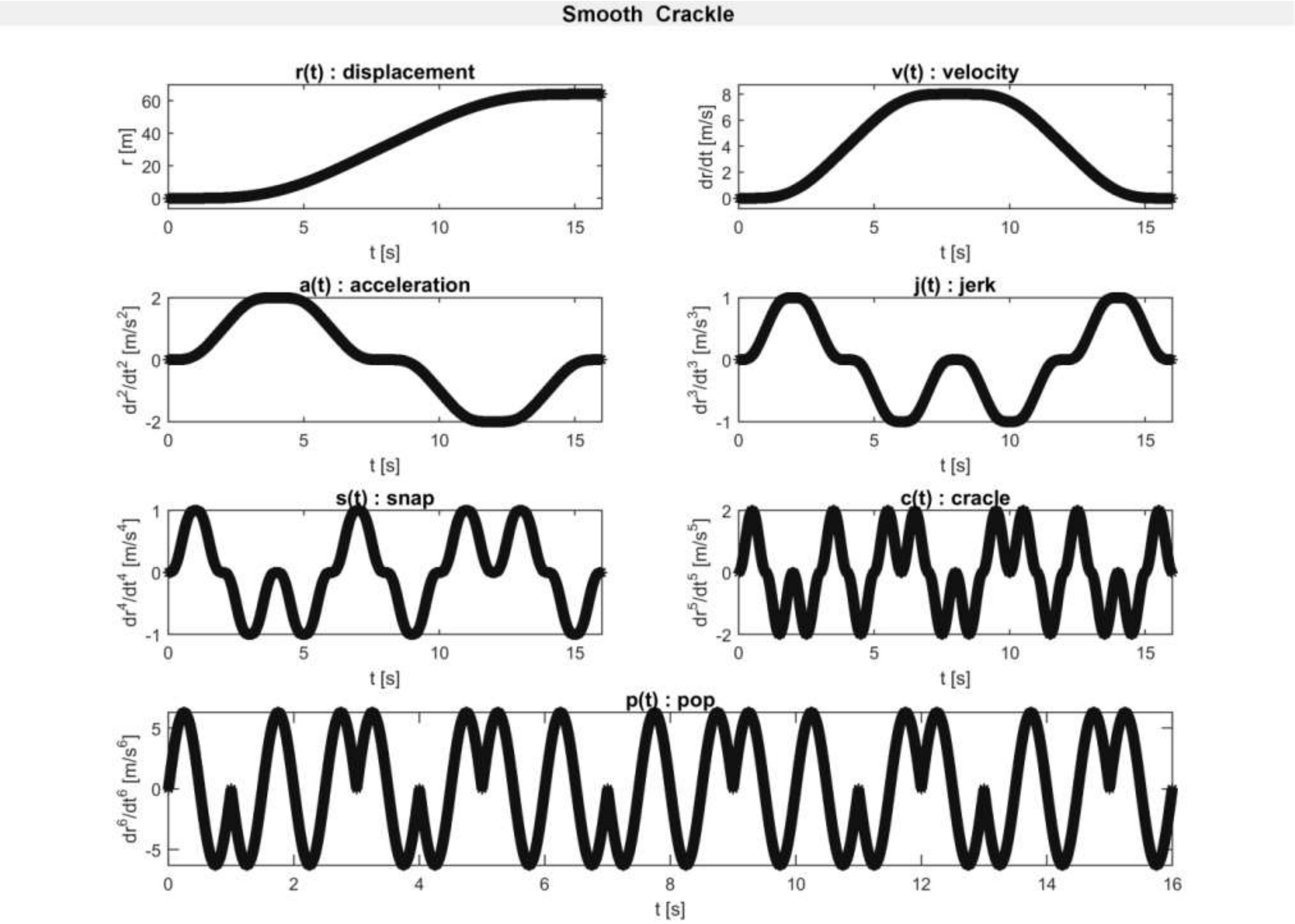


Figure 32.
Harmonic ***(x,y,z)*** displacements with yaw Euler angle orientation and their time derivatives.

5.2.4 Results of the New Feasible Optimal Harmonic Multi-rotor Flight Trajectories of Bounded, Smooth Time Derivatives

To fully understand and appreciate the quality and value of my feasible trajectory proposal, please study first Appendix II where the effects of discontinuities in jerk, snap and pop on the flight dynamics of a multi-rotor are presented. As the trajectory is fully symmetric, for the clarity and manageable size of figures I am here presenting only the first segment of the displacement: ***(x,y,z)*** *= (0,0,0) -> (64,64,64).*

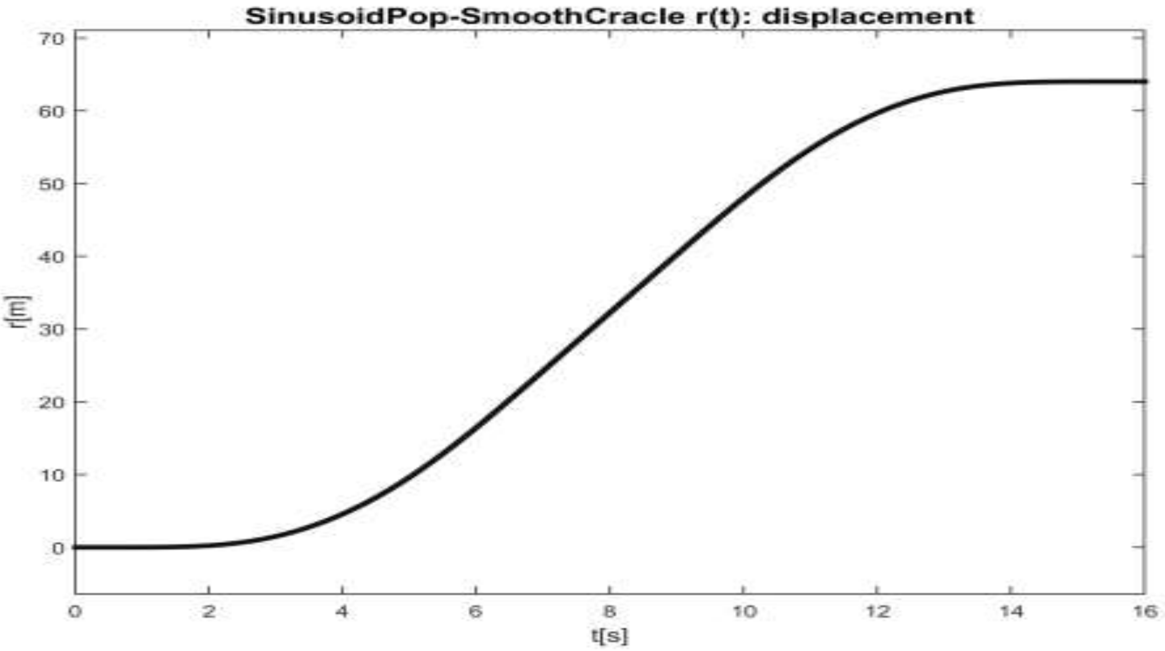


Figure 33.
Harmonic *(x,y,z)* displacements and ψ yaw Euler angle orientation scheme.

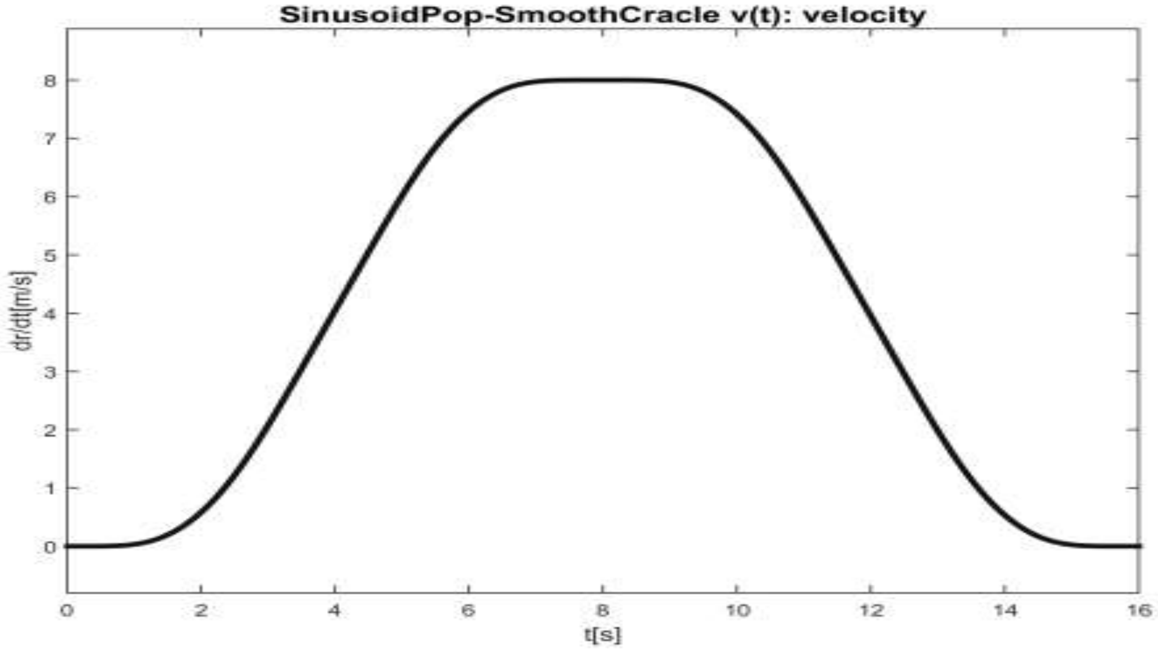


Figure 34.
Harmonic *(x,y,z,ψ)* velocity scheme.

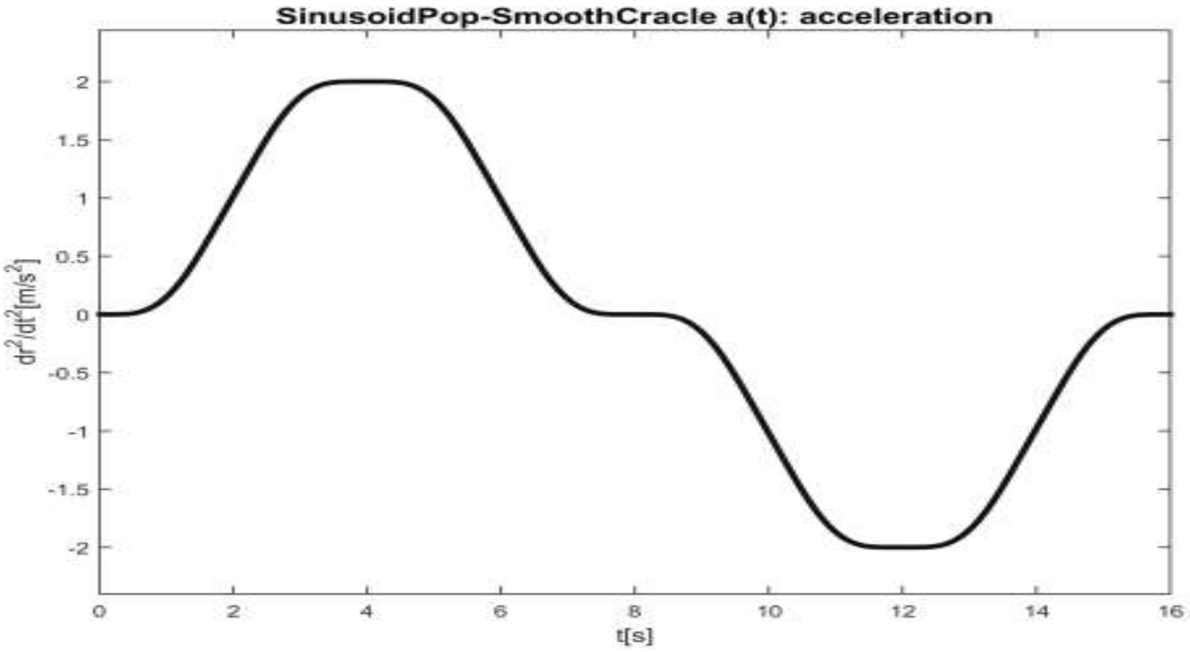


Figure 35.
Harmonic *(x,y,z,ψ)* acceleration scheme.

The above presented displacement, velocity and acceleration profiles, common to harmonic *x,y,z* and *ψ* changes – at the first glance – look quite simple; their nature will be obvious only after inspecting their further time derivatives.

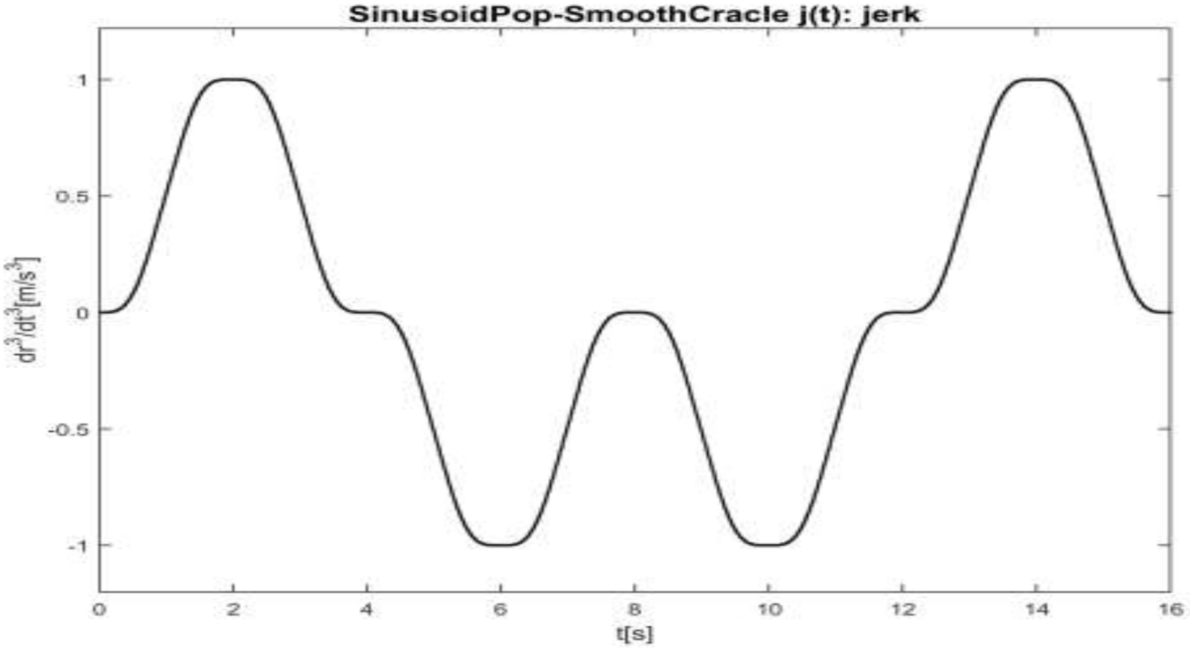


Figure 36.
Harmonic *(x,y,z,ψ)* jerk scheme.

The rhythm typical to my harmonic feasibility trajectories is starting to reveal – each following time derivative will present a double frequency 'mirrored' image of the previous time derivative.

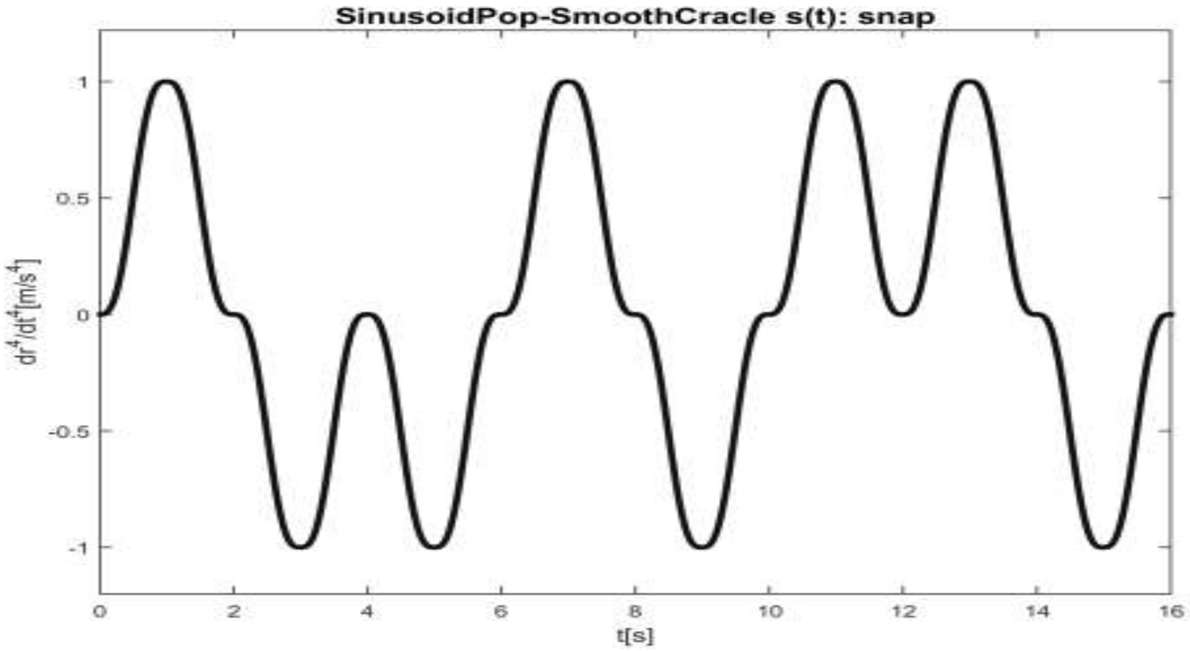


Figure 37.
Harmonic *(x,y,z,ψ)* snap scheme.

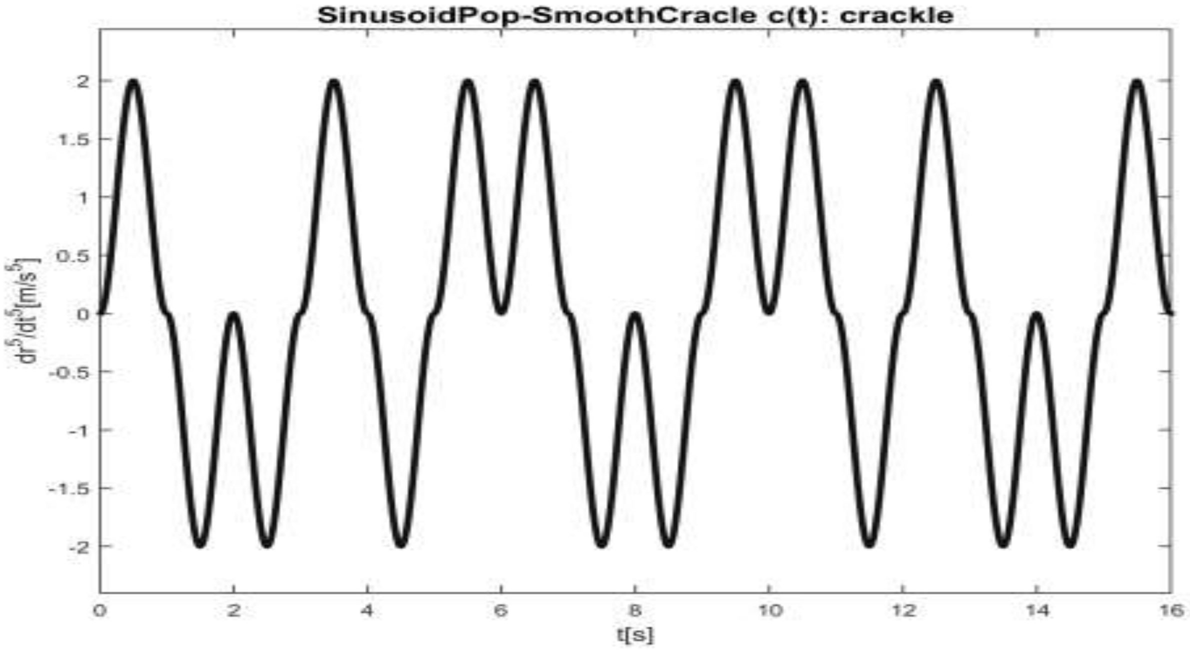


Figure 38.
Harmonic *(x,y,z,ψ)* cracle scheme.

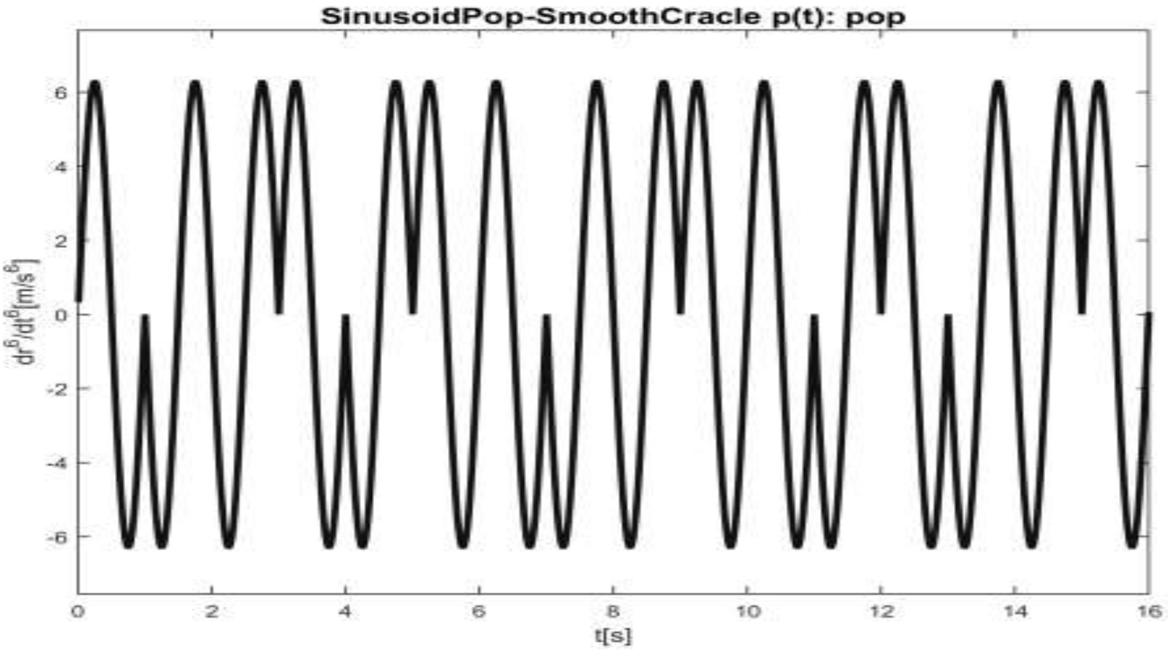


Figure 39.
Harmonic *(x,y,z,ψ)* pop scheme.

Finally the pop function, which a series of sinusoid base functions. Notice that by symmetrically inserting pauses between these sinusoid base functions, we can arbitrary prolong the selected time derivative maximum value, without increasing any values of higher derivatives.

For example introducing a gap of 2 seconds at time instance *t=8*, we change our velocity profile in a manner that the maximum velocity is held for these 2 seconds, thus the displacement is significantly increased, without changing the maximum velocity, or acceleration.

Similarly by introducing a 1 second delay at time instanced *t=4* and *t=12* we change our velocity profile in a manner that the maximum acceleration and deceleration is held for this 1 second, thus the maximum velocity and displacement is significantly increased, without changing the maximum acceleration, or jerk.

Analogously by introducing a 1 second delay at time instanced *t=2*, *t=6* and *t=10, t=14* we change our acceleration profile in a manner that the maximum and minimum jerk is held for this 1 second, thus the maximum acceleration, velocity and displacement is significantly increased, without changing the maximum jerk, or snap. These higher intensity trajectories are presented in Appendix II.

Notice that the smoothness and general scheme of any derivative is unaffected by these pauses in the pop function.

The following figures present the smooth roll and pitch orientation changes, matching torques, rortor angular displacements, and their derivatives, corresponding to the designed ***(x,y,z,ψ)*** profiles.

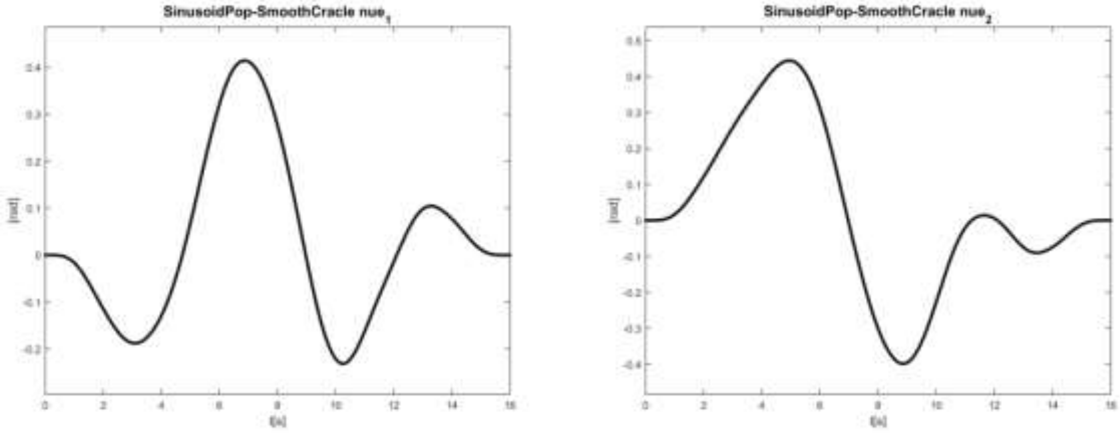


Figure 40.
Resulting smooth roll, pitch Euler angle orientation.

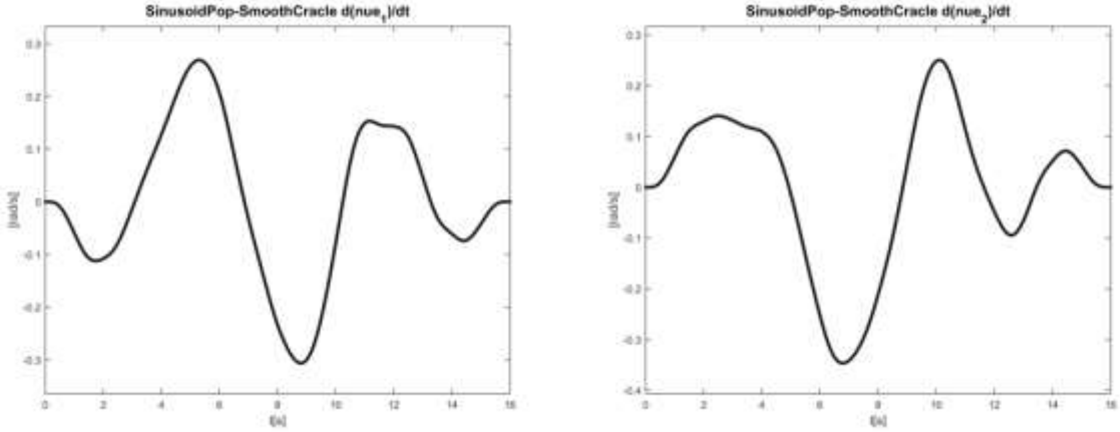


Figure 41.
Resulting smooth roll, pitch orientations first time derivative.

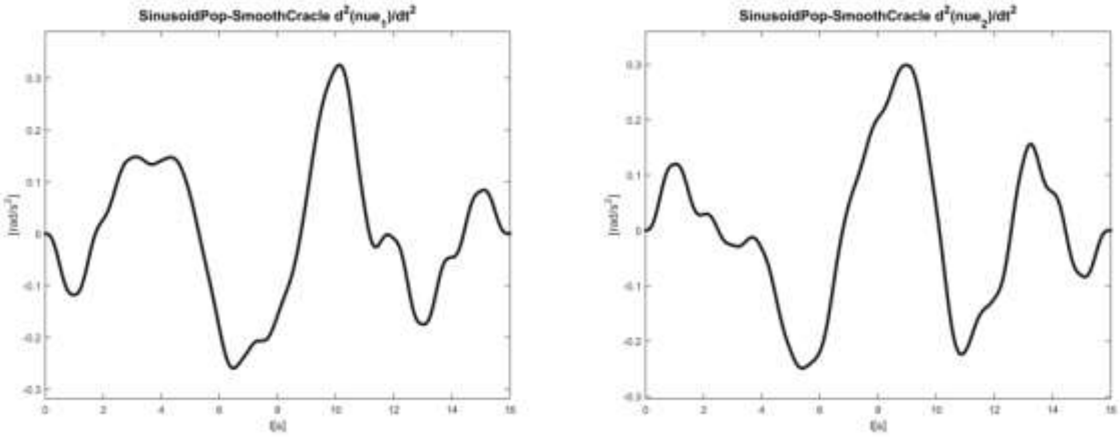


Figure 42.
Resulting smooth roll, pitch orientations second time derivative.

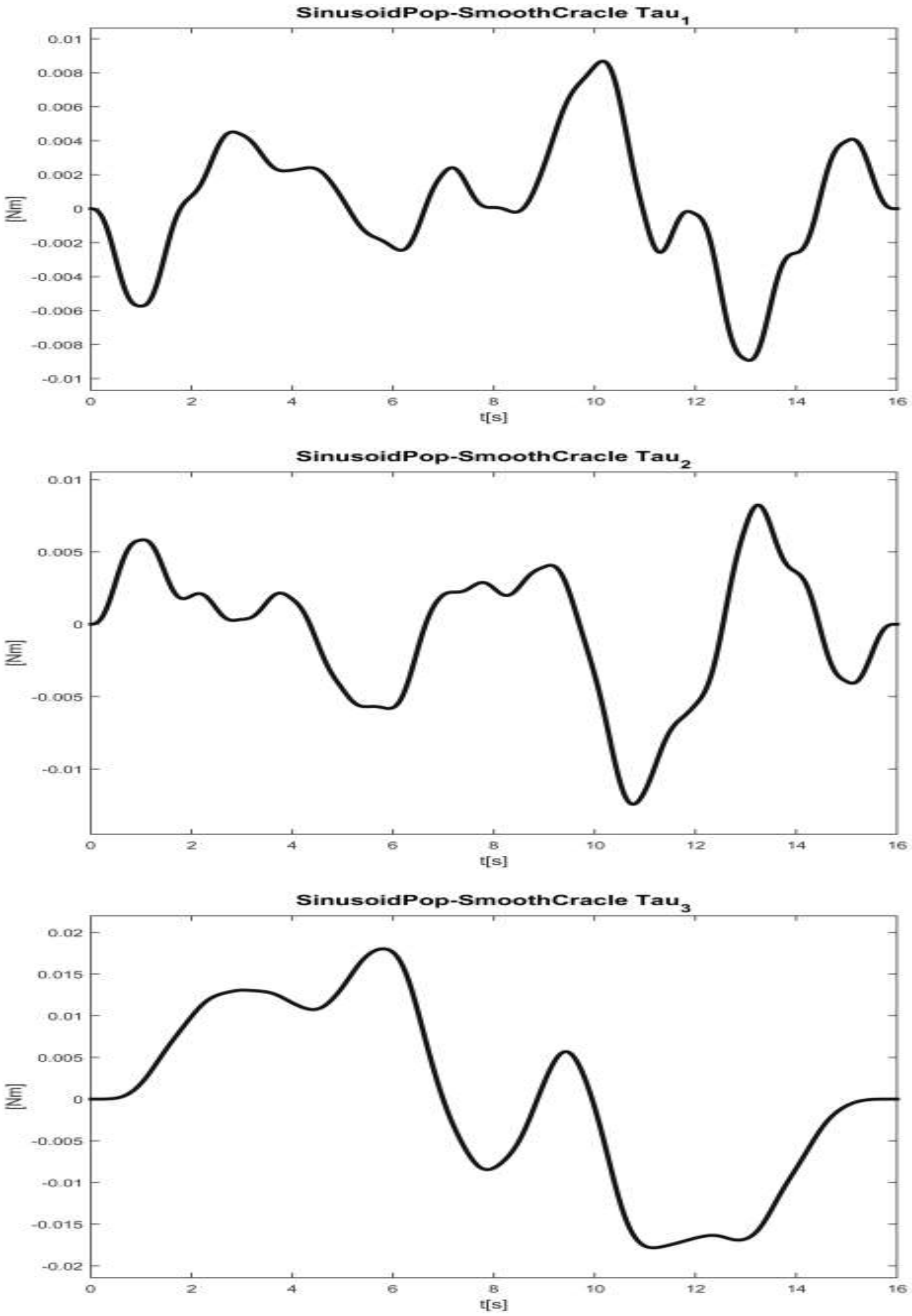
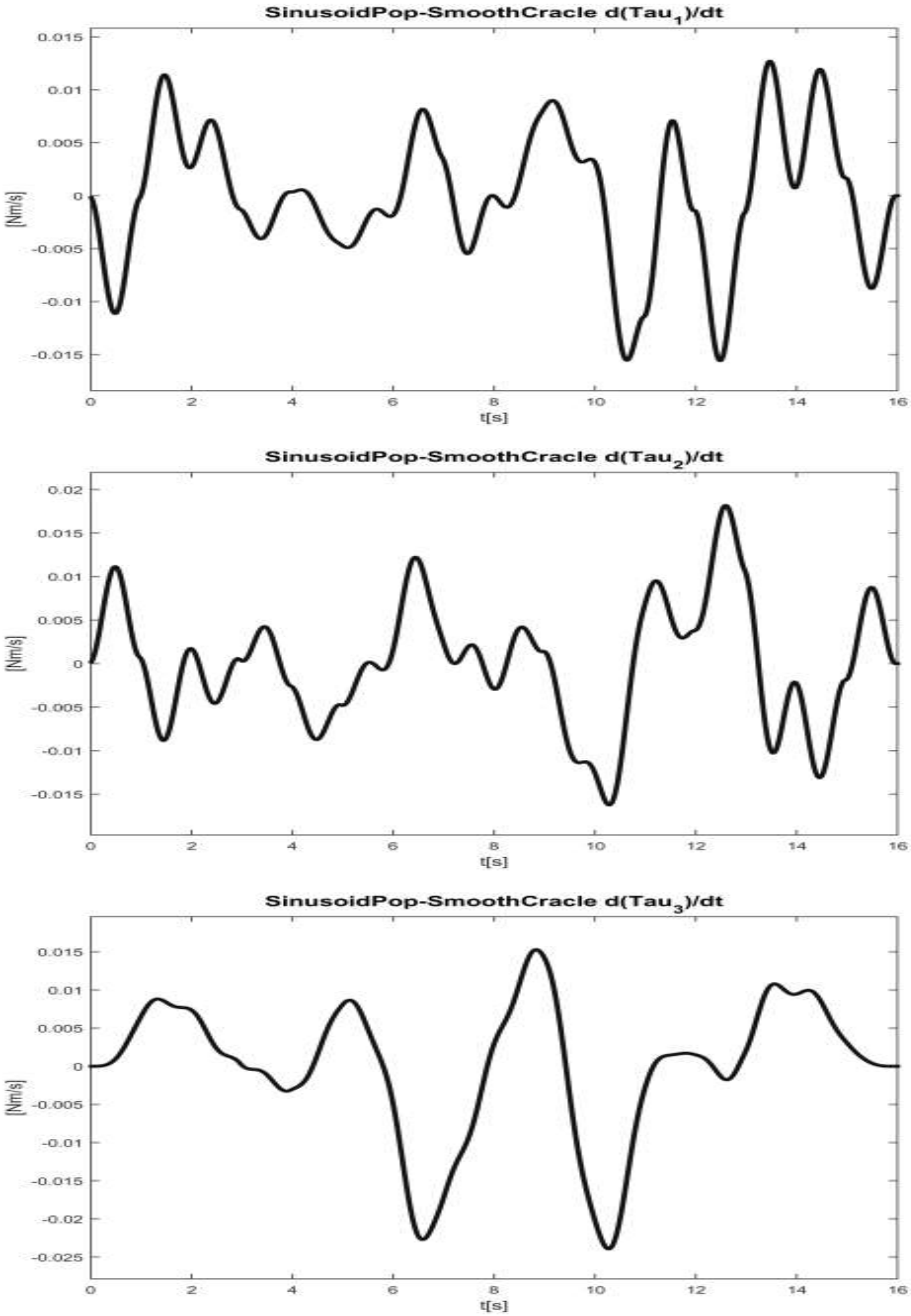


Figure 43.
Resulting smooth roll, pitch, yaw torques.

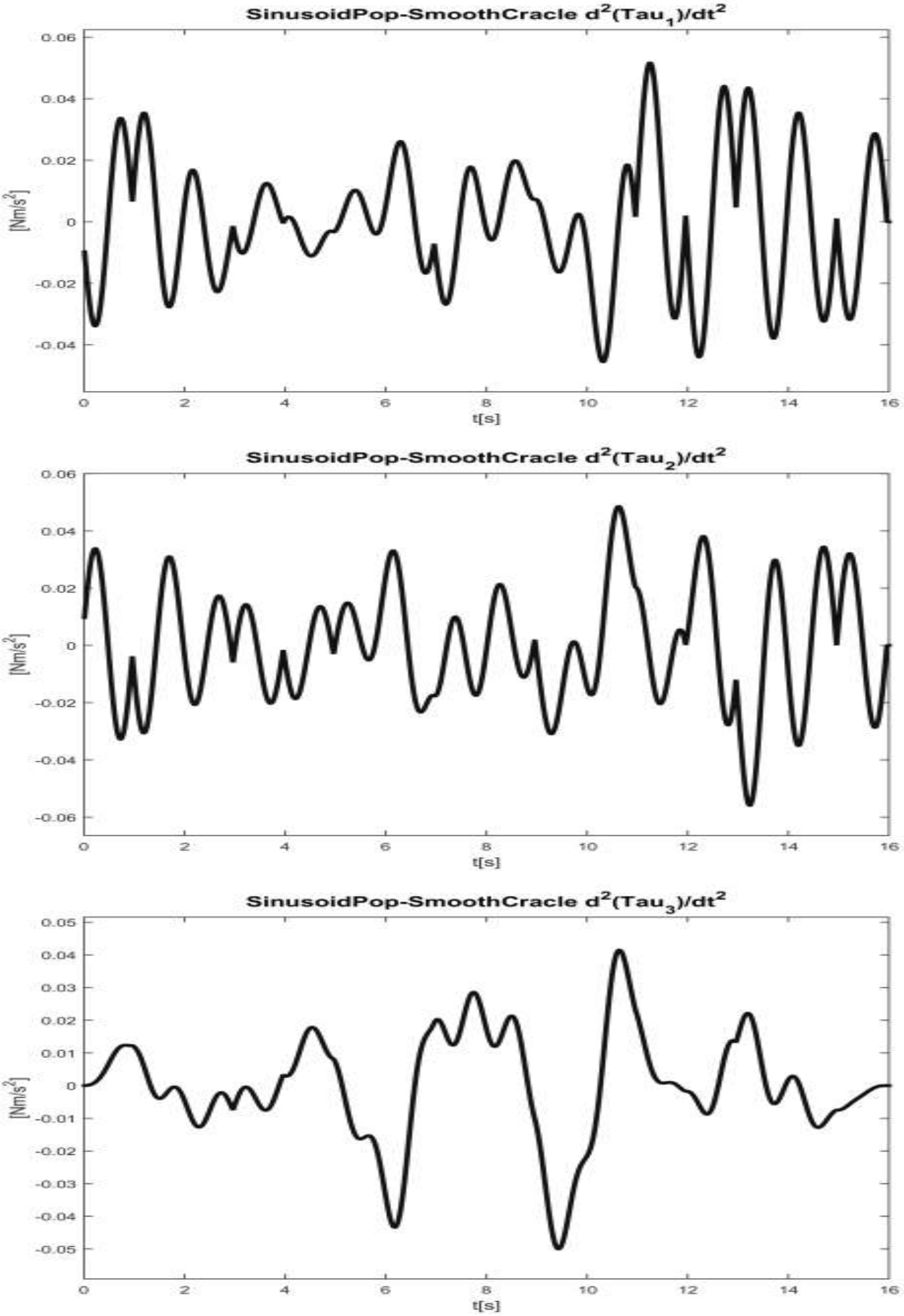


SinusoidPop-SmoothCracle d(Tau$_1$)/dt

SinusoidPop-SmoothCracle d(Tau$_2$)/dt

SinusoidPop-SmoothCracle d(Tau$_3$)/dt

Figure 44.
Resulting smooth roll, pitch, yaw torque first time derivatives.



SinusoidPop-SmoothCracle $d^2(Tau_1)/dt^2$

SinusoidPop-SmoothCracle $d^2(Tau_2)/dt^2$

SinusoidPop-SmoothCracle $d^2(Tau_3)/dt^2$

Figure 45.
Resulting smooth roll, pitch, yaw torque second time derivatives.

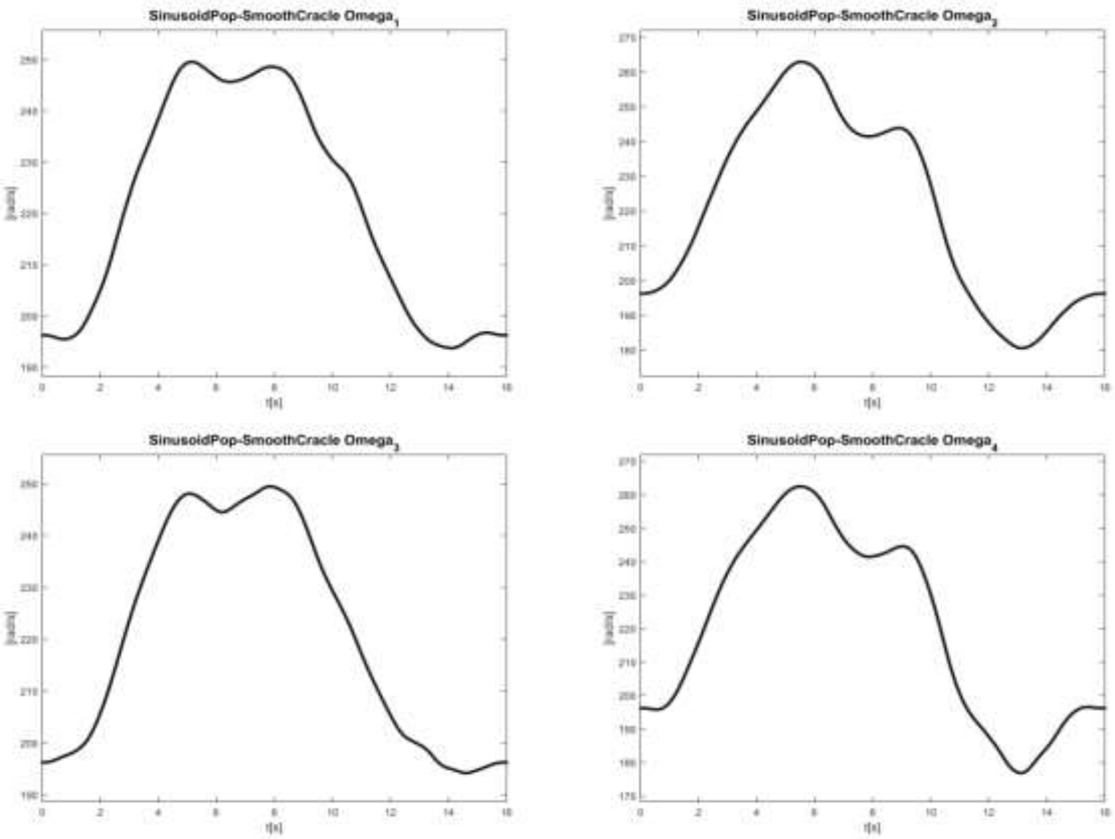


Figure 46.
Resulting smooth rotor blade angular velocities.

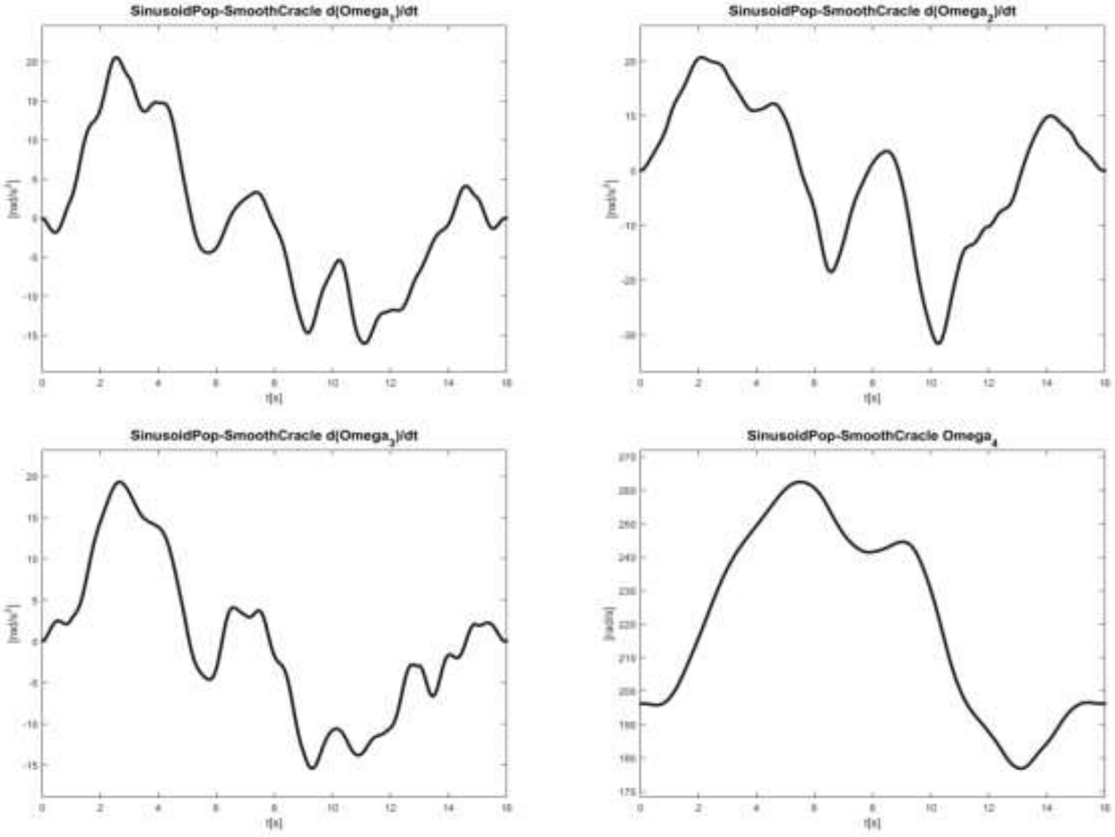


Figure 47.
Resulting smooth rotor blade angular accelerations.

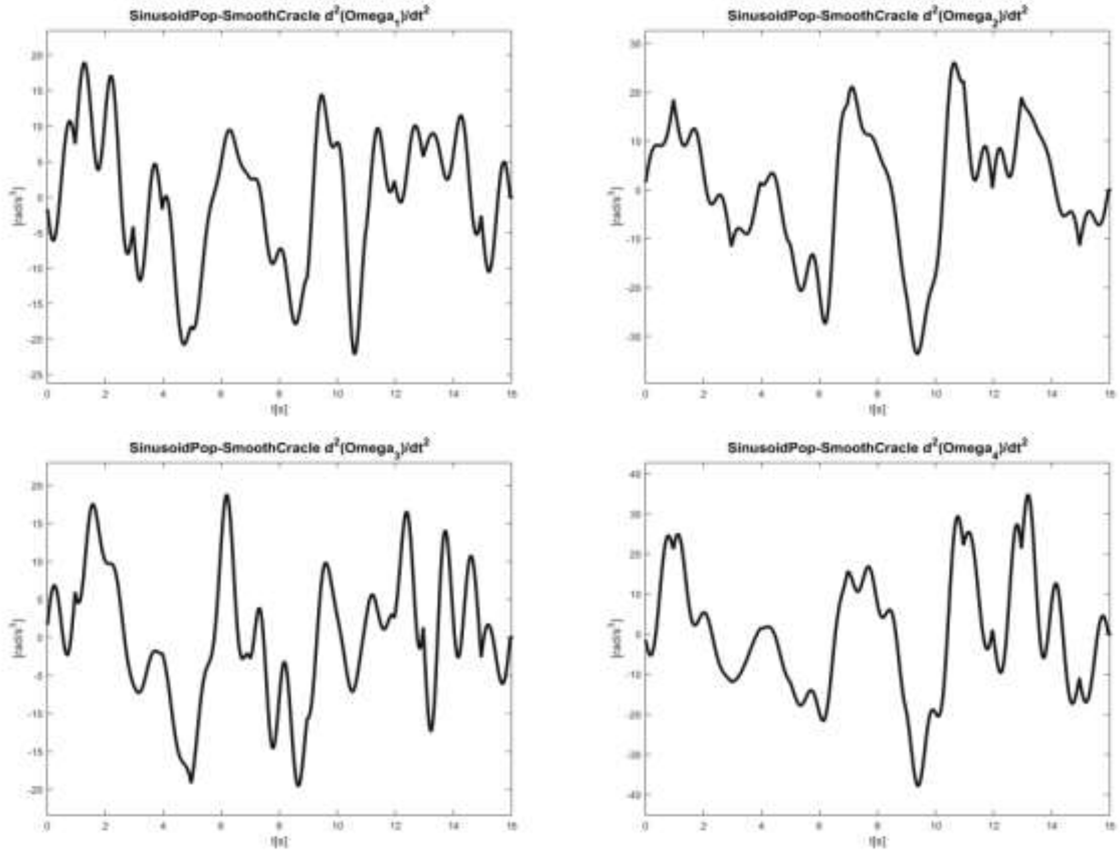


Figure 48.
Resulting smooth rotor blade angular acceleration time derivative.

By examination of above Figures 33-48 we can conclude that all the relevant system variable transients are continuous and smooth. Since this trajectory is planned so that the highest harmonics in changes of angular velocities correspond to the transient frequency of the BLDC actuators, the trajectory is feasible. Since the BLDC actuator is not capable of producing faster changes, there exists no faster trajectory for the planned displacement, so this trajectory is feasible time optimal.

By this analysis I conclude that my Thesis IV.b and IV.c are proven valid.

Energy efficiency and oscillations properties of my feasible optimal harmonic trajectories are examined in the fore coming chapters on a system, which manifests and magnifies these traits to a highly noticeable, visible level.

5.2.5 New Feasible Optimal Harmonic 3D Overhead Crane Trajectories

Compared to the analysed multi-rotor dynamics, a 3D overhead crane model and a RM dynamics model are of the same basic format as equation (39), these systems are more simple as the position of the payload or end effector is directly linked to the position of the actuator rotor shaft – there is no intermediate transfer function like equation (54) for a multi-rotor. This fact predicts that cranes and RMs are not sensitive to discontinuities in the trajectory pop or crackle, only the snap has to be continuous.

Notice that for 3D overhead cranes and a RMs it is expected to be enough to have $\boldsymbol{\xi}_{\boldsymbol{t}}^{(4)}(t) = G \cdot \frac{2\pi}{P} \sin\left(\frac{2\pi}{P} t\right)$, since we have a direct link between the actuator motor shaft position and the system variables $\boldsymbol{q}(t) = \boldsymbol{\xi_t}(t)$ – as opposed to multi-rotors, where equation (54) has to be first applied. When the trajectory is planned with parameter P matching the system transient behaviour, the trajectory is harmonic.

THESIS IV.d - DEFINITION:

When the trajectory is planned as in my Thesis IV.c with parameter P matching the system transient behaviour, the trajectory is harmonic.

For a **harmonic, realistic, feasible multi-rotor trajectories induce no system oscillations.**

5.2.6 Implementation of the New Feasible Optimal Harmonic 3D Overhead Crane Trajectories of Bounded, Smooth Time Derivatives

The generated trajectory can be applied as parametrisation to any vector function defined path ***f****(t)=(****f****$_x$(****r****(t)),****f****$_y$(****r****(t)),****f****$_z$(****r****(t)))*. When determining the constraints on trajectory derivatives, one has to take into consideration both the system limits and curvature properties of the desired path ***f****(t)* and its derivatives. ***f****(t)* has to be smooth at least up to the required smoothness of the trajectory.

Reduction of the method is strait forward to systems with simpler trajectory constraints, like RMs or wheeled vehicles, where it is enough to have smooth trajectories up to the 3rd time derivative of displacement. A 3D crane, as described in [11] is a simple system, which can in a feed forward control setup very notably present system oscillations induced by the prescribed trajectory. The dimensionless cost function $\boldsymbol{C_{el}}(s) = \frac{1}{2}\int_0^{t_f}(\ddot{\boldsymbol{r}}(t)^2 + \dot{\boldsymbol{r}}(t)^2 + \boldsymbol{r}(t)^2)dt$ was minimised by calculus of variation, where *r(t)* is the displacement.

5.2.7 Results of the New Feasible Optimal Harmonic 3D Overhead Crane Trajectories of Bounded, Smooth Time Derivatives

Figure 49 and 50 presents the crane system feed forward output error (payload delta position and pitch angle) for bang-bang acceleration and for a minimal torque (minimal electric energy trajectory). System oscillations are obvious even by observing the payload position only.

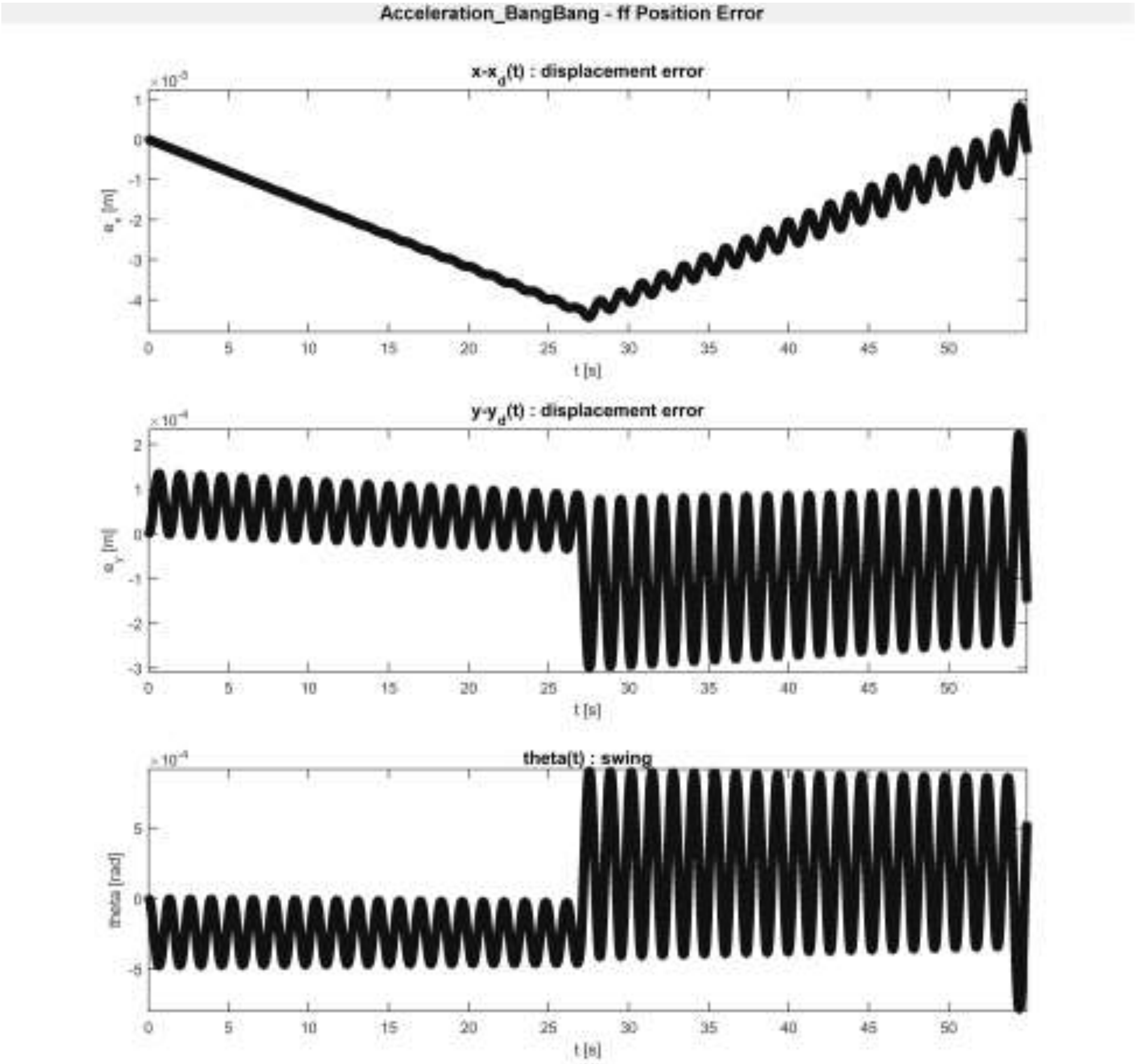


Figure 49.
Crane feed forward response error: classical bang-bang acceleration trajectory.

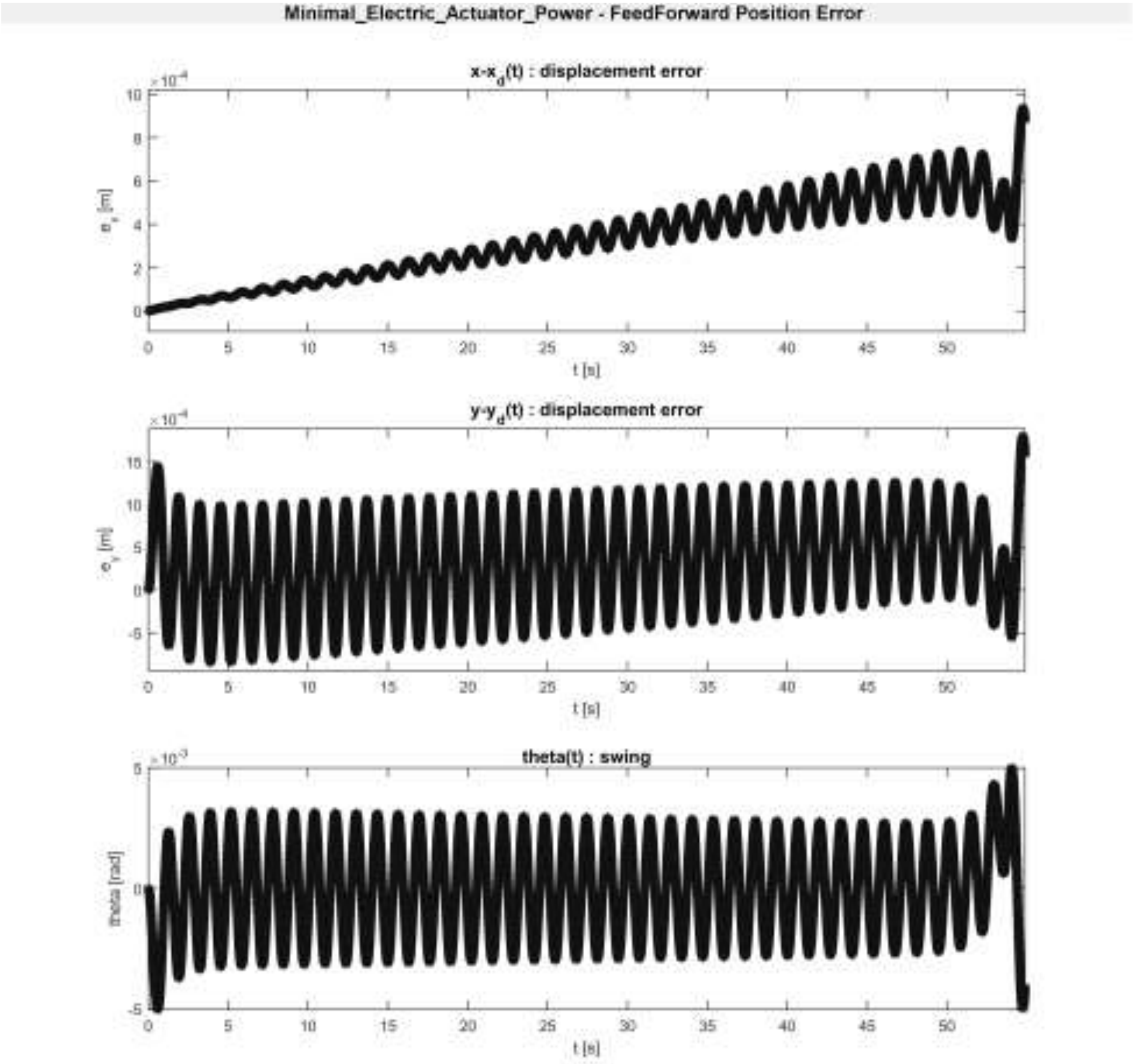


Figure 50.
Crane feed forward response error: classical minimal torque trajectory.

Table XVIII presents the numerical results for maximum payload pitch (Theta), maximum payload tracking error along x and y, and the torque cost (electric energy cost function) for classical 'optimal' trajectories. Table XIX. presents the numerical results for my proposed feasible optimal harmonic smoot trajectories, measured by the same objective function.

One can conclude by looking at the numerical results of 'hastier trajectories' in Table XIX that the more timid, slower changing the trajectory is, the better the performance is along all four objectives. The "w$_0$" reference in the table stands for the used ideal pendulum angular frequency $\omega_0 = \sqrt{g/L}$, where g is the gravity acceleration (9.81m/s^2) and L is the pendulum length; the divisor "w$_0$/n" by the angular frequency in the trajectory name represents the trajectory length multiplier compared to a smooth trajectory defined by a pop base function of period ω_0 (a trajectory of name ending with "w0/2k" takes 2 times longer to complete than that of "w0/k").

TRAJECTORY TYPE / PERFORMANCE	maxTheta	maxErrorX	maxErrorY	maxTorqueCost
Minimal_Jerk w0:	3.72E-04	3.73E-05	1.21E-04	1.25E-01
Acceler_BangBang w0:	9.04E-04	4.44E-05	3.00E-04	1.34E-01
Minimal_Snap w0:	4.76E-04	4.35E-05	1.53E-04	1.48E-01
Minimal_Crackle w0:	6.01E-04	4.90E-05	1.89E-04	1.67E-01
Minimal_Torque w0:	5.02E-03	9.35E-04	1.80E-03	7.24E-02
Minimal_Acceler w0:	7.07E-04	8.97E-05	2.29E-04	9.96E-02
Vmax_Lightning w0:	2.50E+00	1.58E+01	1.58E+01	1.94E+03
HASTIER				
Minimal_Torque w0*2:	1.05E-02	1.88E-03	3.01E-03	1.57E-01
Minimal_Torque w0*4:	2.28E-02	2.14E-03	6.55E-03	3.63E-01
Minimal_Torque w0*8:	6.56E-02	5.77E-03	1.59E-02	9.09E-01
Minimal_Torque w0*16:	2.88E-01	2.35E-02	6.40E-02	2.73E+00
Minimal_Torque w0*32:	7.97E-01	7.84E-02	1.30E-01	6.43E+00

Table XVIII. Numerical results for the crane feed forward control setup – classical trajectories.

All trajectories planned for faster completion than $t_T = 2 * \pi/\omega_0$ end up with oscillations. The proposed smooth trajectories are always inducing less oscillation than the classical "minimum" counterparts. The minimum crackle and the proposed smooth crackle trajectories are the only two trajectory types that starting from $t_T = 2 * \pi/\omega_0$ long trajectory motions, which result in no significant crane pendulum second state derivative oscillations. For longer durations other variants of the proposed smooth trajectories are totally vibration free. One benefit of the proposed smooth trajectory is in the designed, arbitrary bounded derivative maximum values – one can set any velocity, acceleration, jerk, even snap limits. The other benefit is that by increasing the level of smoothness one can ensure absolute oscillation free behaviour and reduce the position error, even to reduce the required energy – of course all this is at the cost of longer trajectory durations.

SMOOTH TRAJECTORIES:				
SmoothCrackle_w0/32:	1.93E-06	1.51E-06	1.70E-06	8.18E-03
SmoothSnap_w0/16:	7.74E-06	3.01E-06	4.30E-06	1.64E-02
SmoothCrackle_w0/16:	7.74E-06	3.01E-06	4.31E-06	1.64E-02
SmoothSnap_w0/8:	3.11E-05	6.03E-06	1.28E-05	3.31E-02
SmoothCrackle_w0/8:	3.11E-05	6.03E-06	1.28E-05	3.31E-02
SmoothSnap_w0/4:	1.26E-04	1.21E-05	4.33E-05	6.73E-02
SmoothCrackle_w0/4:	1.26E-04	1.21E-05	4.34E-05	6.73E-02
SmoothSnap_w0/2:	5.33E-04	2.42E-05	1.65E-04	1.39E-01
SmoothCrackle_w0/2:	5.33E-04	2.43E-05	1.65E-04	1.39E-01
SmoothSnap_w0:	2.51E-03	1.04E-04	7.04E-04	2.95E-01
SmoothCrackle_w0:	2.53E-03	1.05E-04	7.08E-04	2.95E-01

Table XIX. Numerical results for the crane feed forward control setup – my harmonic, smooth trajectories.

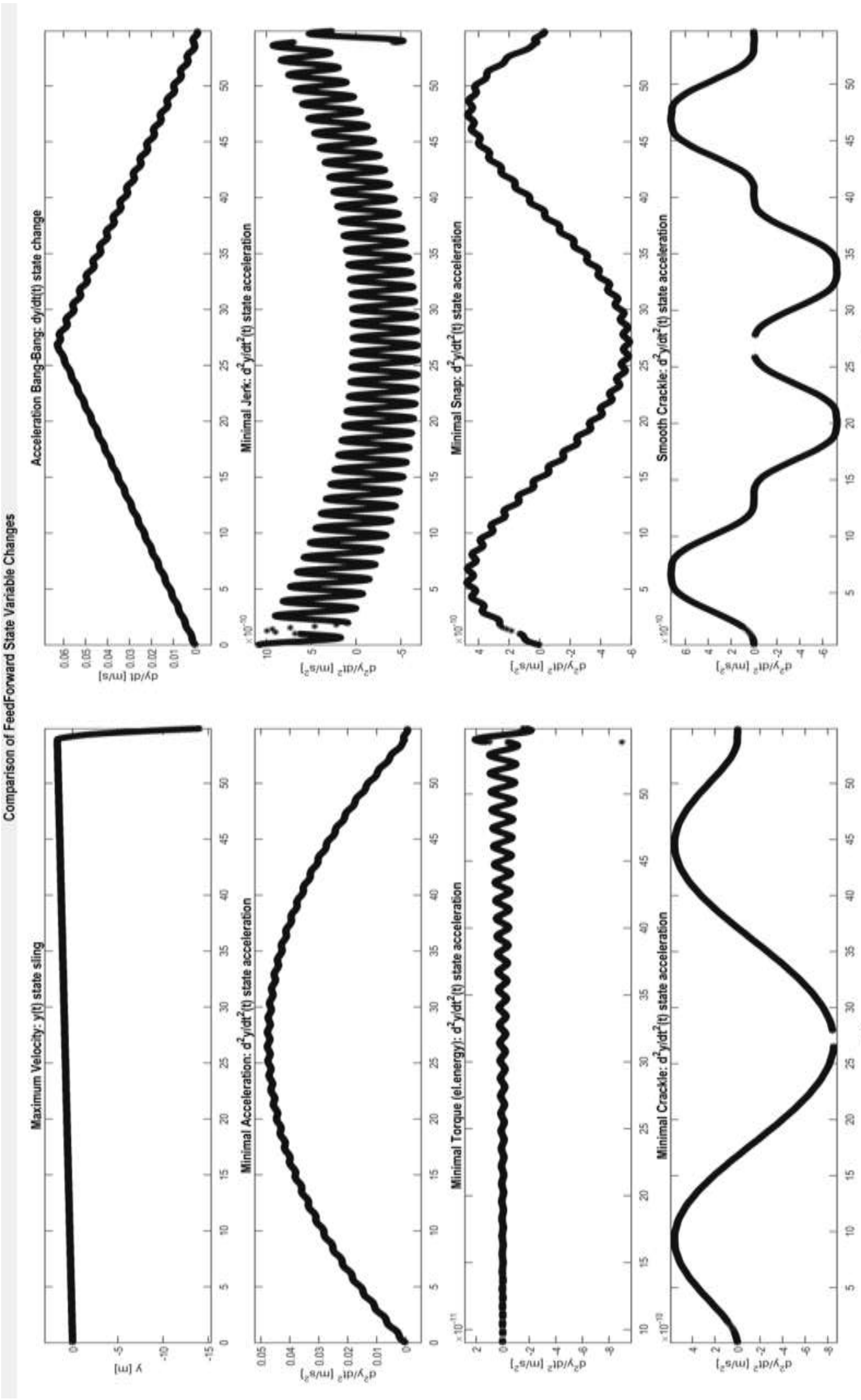


Figure 51.
Crane feed forward response error: various trajectories – observe the oscillations.

Notice that for this crane system example an incomplete mathematical model is used as in [11] – it does not include the actuator dynamics – it counts with the control signal being of arbitrary precision simple mathematical function. For this research paper I have not compensated this model deficiency, so that my results can be directly compared to [11].

Figure 51 presents crane system second time derivative state oscillations induced by different trajectories; for the maximum velocity and the acceleration bang-bang there is just no point in looking at the second time derivative of the y position, only the position and its first derivative is presented, respectively.

Let us examine closer my proposal for a harmonic feasible optimal trajectory of smooth crackle and its effect on system state variables:

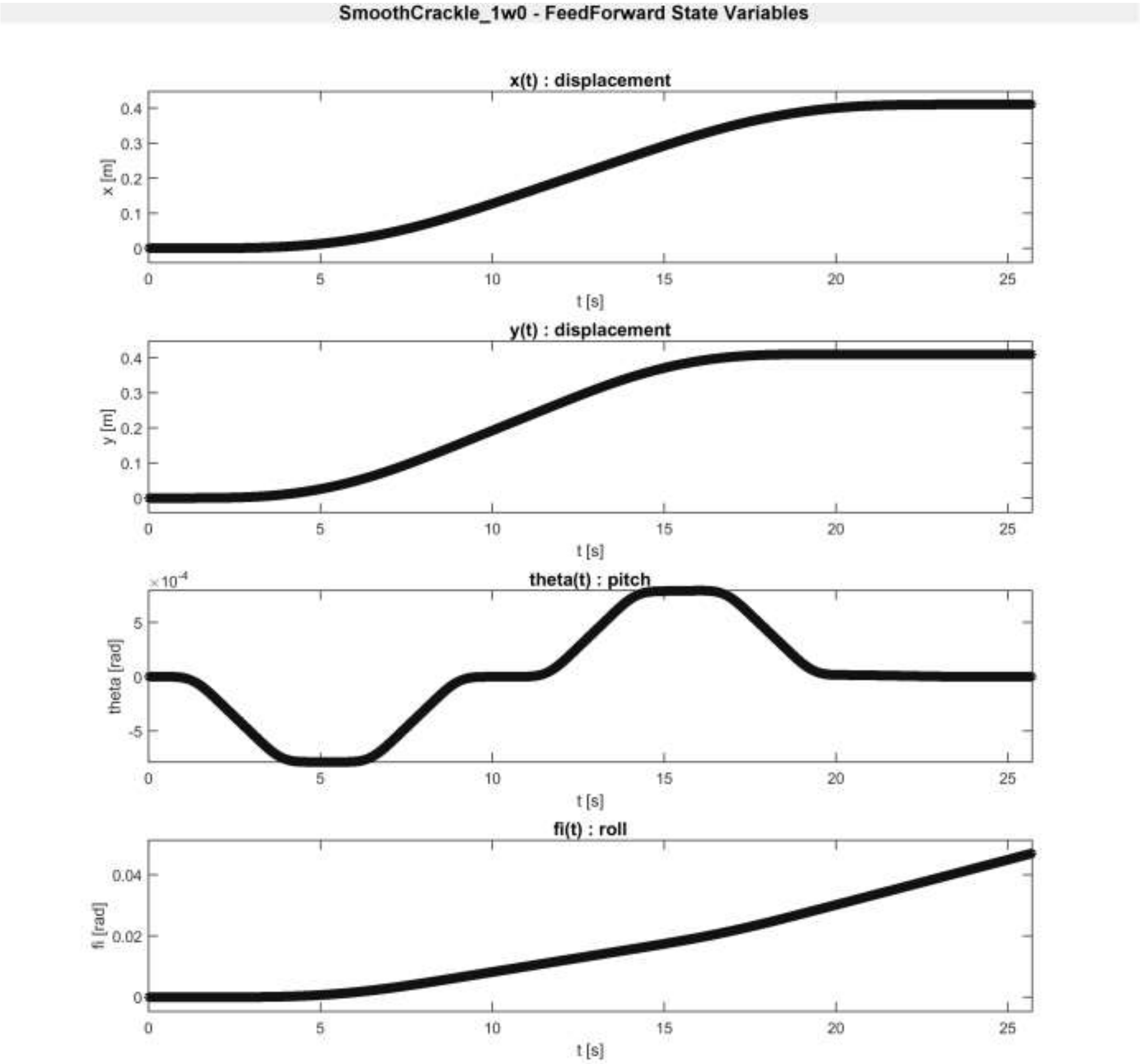


Figure 52.
Crane feed forward state variable: my harmonic optimal feasible smooth crackle trajectory.

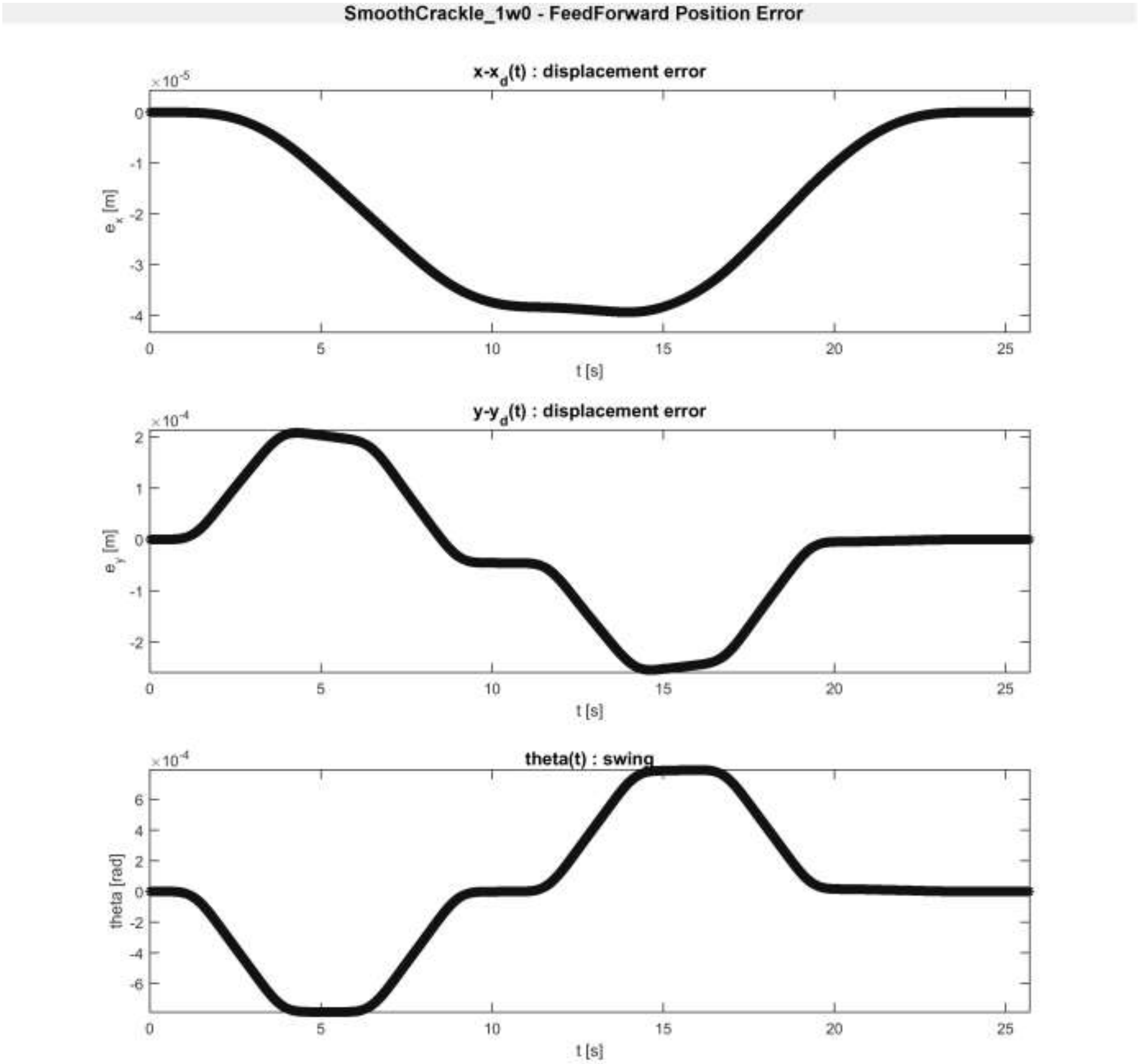


Figure 53.
Crane feed forward position error: my harmonic optimal feasible smooth crackle trajectory.

The trajectory is planned for a realistic laboratory setup comparable to [3]. The planned displacement is of 41 cm. The displacement takes 25 seconds, while the maximal displacement error along all 3 dimensions is extremely small <0.1mm (radians).

One benefit of the proposed smooth trajectory is in the designed, arbitrary bounded derivative maximum values –any velocity, acceleration, jerk, even snap limits. Other benefit is that by increasing the level of smoothness one can ensure absolute oscillation free behaviour and reduce the position error, even to reduce the required energy – of course all this is at the cost of longer trajectory durations.

The next figure presents rates of displacement (velocity) and their change (accelerations) – these signals clearly presents state variable oscillations, if any exists that is.

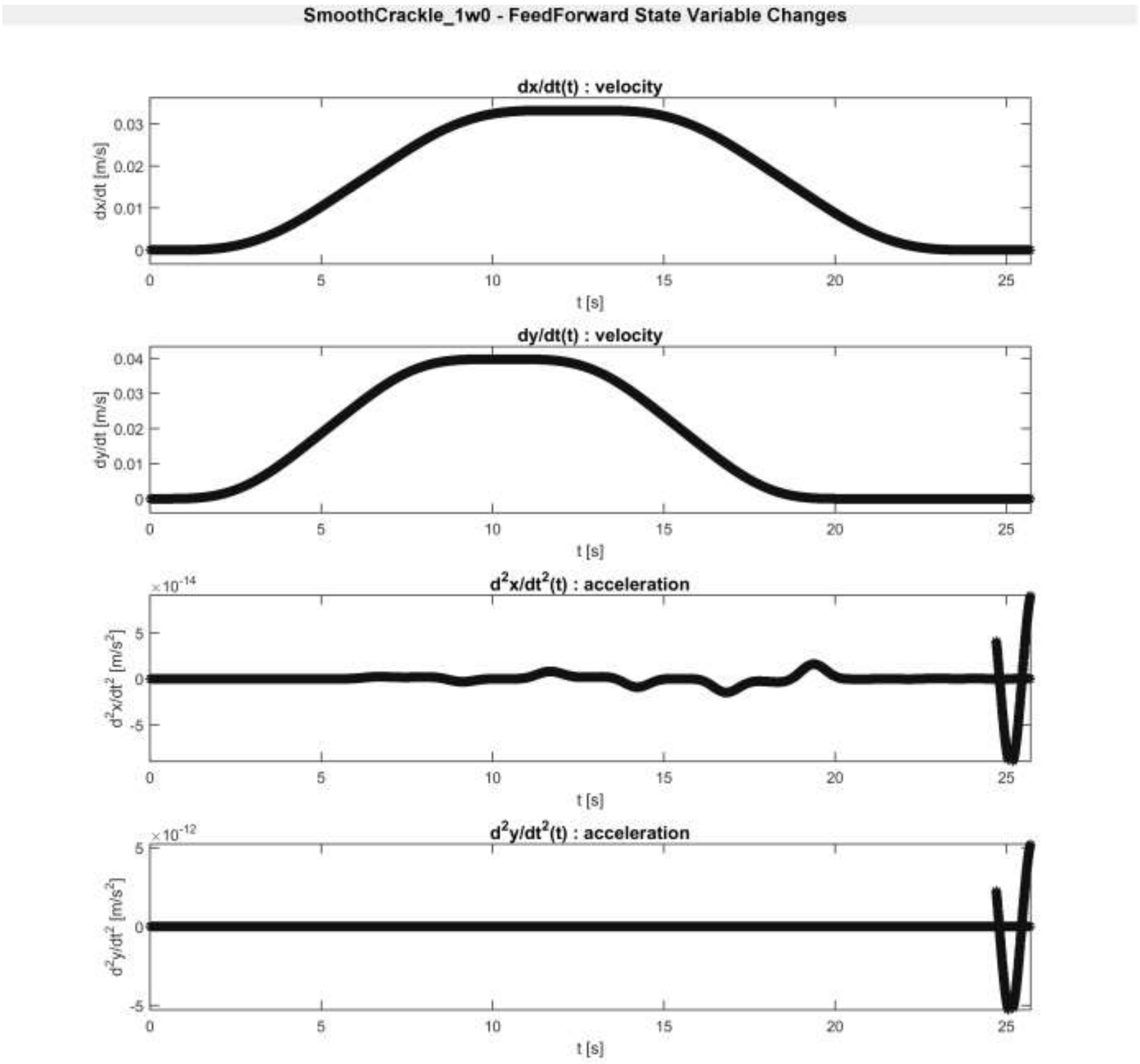


Figure 54.
Crane feed forward state variable time derivatives: my harmonic optimal feasible smooth crackle trajectory – observe no oscillations.

We can observe unquestionably smooth velocity changes along the complete trajectory, up until the stopping point at *t=24* seconds. After the 24th second an extremely small wave appears of a magnitude ~1e-14m and ~1e-12m, which I cannot contribute to anything else but numerical error of the Matlab Simulink environment.

All trajectories planned for faster completion than $t_T = 2 * \pi / \omega_0$ end up with oscillations. The proposed smooth trajectories are always inducing less than the classical “minimum” counterparts; practically no oscillations are induced by harmonic trajectories. The minimum crackle and the proposed smooth crackle trajectories are the only two trajectory types that starting from $t_T = 2 * \pi / \omega_0$ long trajectory motions, which result in no significant crane pendulum second state derivative oscillations. Lower frequency trajectory $t_T = 32 * 2 * \pi / \omega_0$ no oscillations at all

By this analysis I conclude that my Thesis IV.a and IV.d are proven valid.

I am here addressing a criticism that my harmonic feasible optimal trajectories are “too slow – as it takes 24 seconds to travel 41cm with a 1m crane”.

My response to this is: if one requires no oscillations and minimal energy usage, then this is what it takes for the analysed physical system – if one requires faster motions, either a well-designed feedback loop with a more capable actuator is to be used, or one has to accept system oscillations. By the way, would anybody consider traveling in 12 minutes some 41 000 kilometres (all around the Earth’s equator) to be too slow? – because this is what is harmonically feasible with the same crane system setup, when no physical system boundaries in terms of cran size are considered:

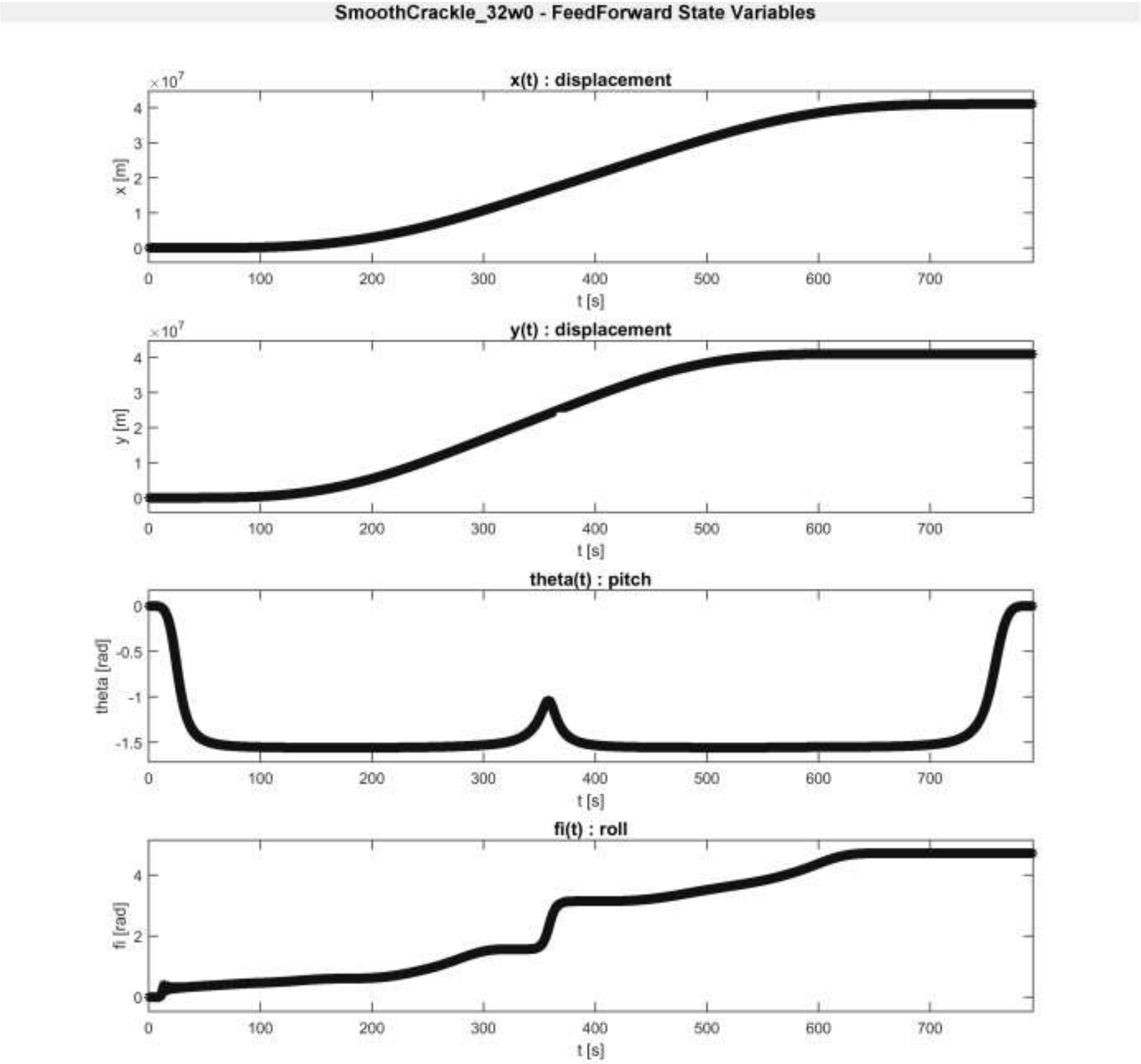


Figure 55.
Crane feed forward state variables: exaggerated smooth crackle trajectory “around the equator in 12 minutes”.

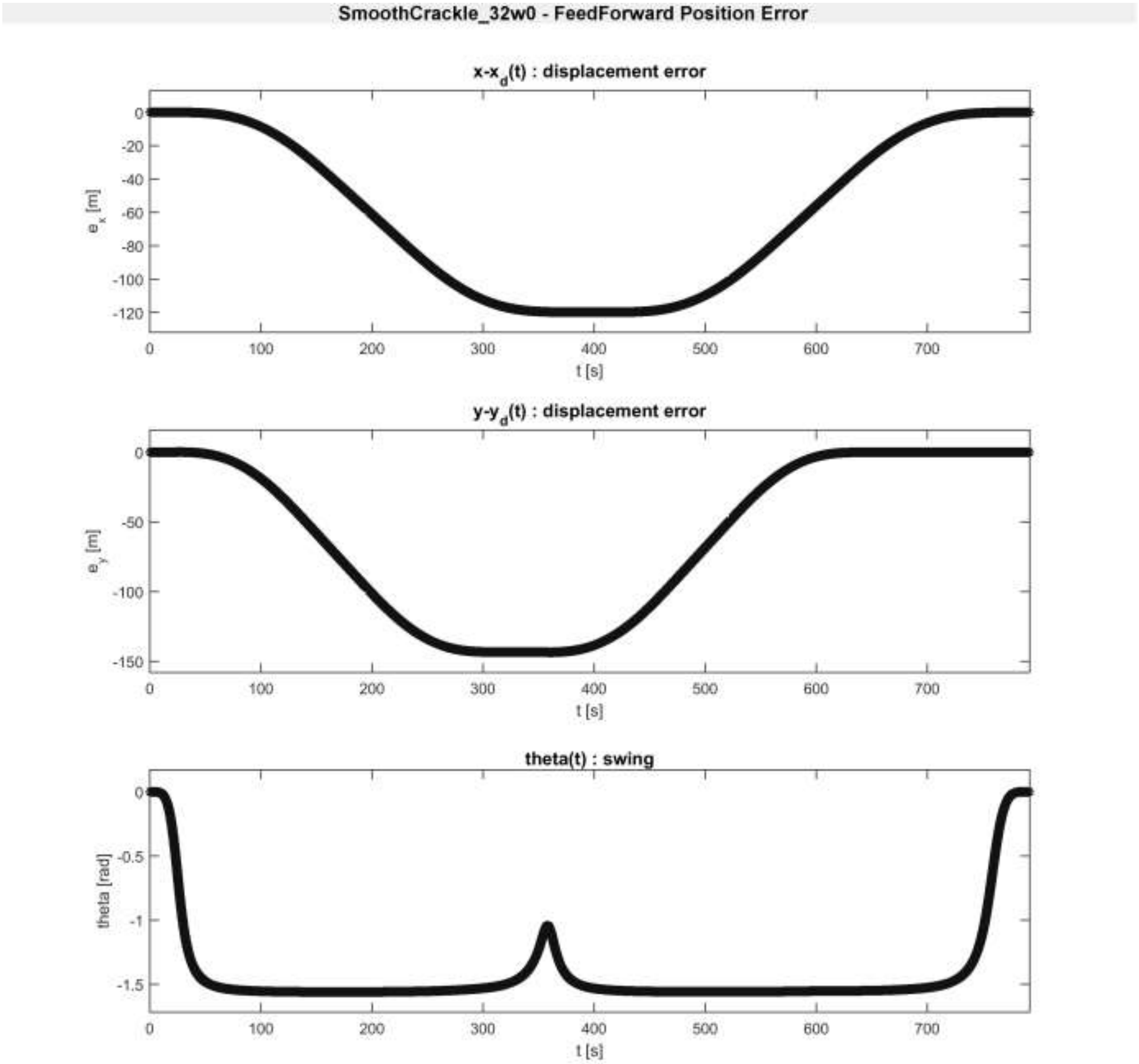


Figure 56.
Crane feed forward position error: exaggerated smooth crackle trajectory “around the equator in 12 minutes”.

The displacement error corresponds to the distance travelled at maximum speed (10e+4 m/s) for the sampling time duration (1ms). Notice that the unrealistic maximum velocity is 100km/sec, which is reached by an even more unrealistic acceleration of 500m/s^2, while the maximal jerk is uniformly <5m/s^3, which is a very comfortable value.

Indeed this “around the equator in 12 minutes” is a Sci-Fi crane setup, its sole purpose is to demonstrate that it is not the harmonic feasible optimal trajectory design method responsible for generating any “slow” trajectories, but it is the nature of physical properties of the system that in reality actually limit our harmonic feasible trajectory “speed”.

Of course one can always relax the harmonic feasibility aspect of a trajectory and then relay on a powerful robust stable feedback loop design and excessive energy consumption to drive a system faster than what it is capable of – and then only the induced oscillations have to be properly dampened. I question that it is possible to create an oscillations free trajectory that takes in overall less time than it takes to track a harmonic feasible trajectory; and I am positive that any other trajectory would require much more energy to be spent.

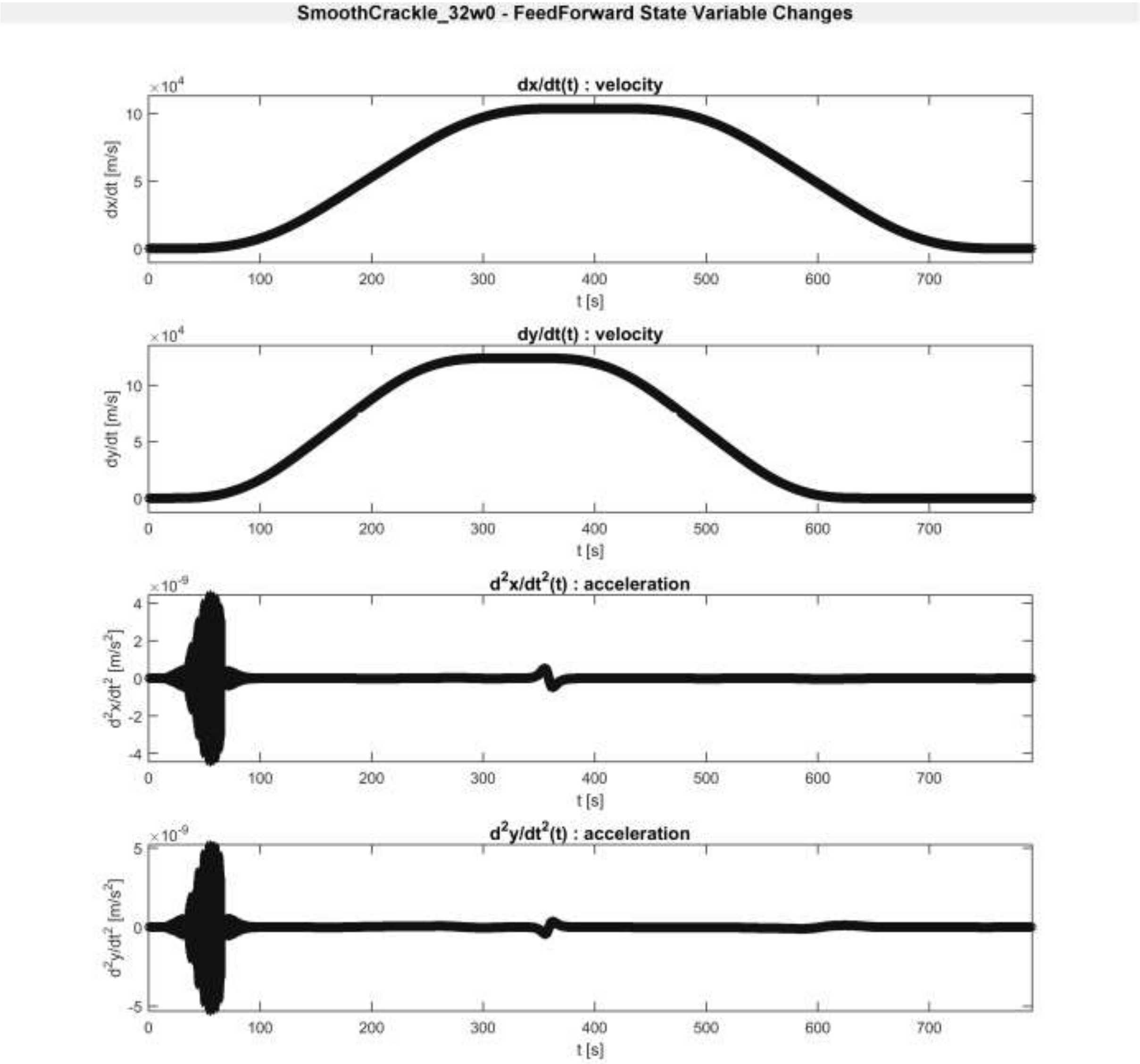


Figure 57.
Crane feed forward state variable time derivatives: exaggerated smooth crackle trajectory “around the equator in 12 minutes” – observe no oscillations.

We can observe smooth velocity changes along most of the trajectory, except for the first 80 seconds, when an extremely small oscillation appears of a magnitude ~5e-9m and ~4e-9m. Since the displacement magnitude is ~ 4e+7 this 4e-9 oscillation is of 1e-16 relative magnitude which I cannot contribute to anything else but numerical error of the Matlab Simulink environment.

6 GENETIC FUZZY SYSTEM TRAINING DATA SET REDUCTION

A prerequisite for system identification is a set of measurements of the system to be modelled while being driven along a pre-defined trajectory. As this training path must be sufficiently exiting so that all system characteristics can be observed, it is natural that we have to operate with extremely large training data sets. Performing Singular Value Decomposition (SVD) of large matrices is extremely time consuming. It is always a challenge to find sufficiently exciting, while being not oversized training data sets.

6.1 Literature Synopsis

Training data sets consist of recorded input-output pairs of the function/system to be identified. A fuzzy system cannot properly respond for totally unknown input space regions, thus for training fuzzy systems a large amount of historical data is collected, as it is important ha the training data covers all possible aspects of the system. The training data collection has to ensure that the complete input space is sampled at least to a certain extent. These aspirations will usually lead to an abandon data set with possibly many unneeded repetitions. Calculations, numerical transformations of huge data sets are always a time consuming task. Neither real time applicability, nor evolutionary algorithms prefer time inefficient long calculation cycles. For real time systems we must conclude our calculations in-between the timeframe of the sampling time. For evolutionary search we perform couple of hundreds evaluations for each generation – if one evaluation takes too long, the complete design calculation easily turns out to be a multi-week server task, which is not a friendly environment for a research study.

In case of modelling complex systems filtering out unnecessary samples, while still leaving all the necessary data for good quality identification is not a trivial strait forward process. In case of every identification problem, specific approaches are used when deemed necessary.

By my *Hypothesis V* this paper presents a novel method that will reduce the necessary training data set size for fuzzy identification of complex dynamic systems. The method is based on finding the minimal subset of the training data, which most efficiently minimises the corresponding condition number of the linear system subject to SVD decomposition when identifying the optimal linear parameters of the system.

6.1.1 Validating Quality of Genetic Fuzzy System Training Data Sets

The nonlinear parameters always have to form complete fuzzy partitions, thus complete uniform input space coverage, what is to be ensured by the proposed new parameter representation method of Thesis II. The quality of a TSK linear parameter set calculated by an SVD based LS optimization method greatly depends on the training data set quality.

By definition the condition number of a parameter data set, which defines a linear system of equations is called the condition number of the equation; it is the ratio of the largest and the smallest singular value. The higher the condition number, the more uncertain the solution is; the more sensitive the solution is to small disturbances of system parameters. The natural goal for a good quality, robust linear system solution is to have as small a condition number as possible.

A very well-conditioned linear system of equations has a condition number of three orders of magnitude less than the reciprocal of the numerical precision of our calculations. In case of double precision floating point digital computer calculous we can rely on a numerical precision of at least ~10^{-16}, which means that any condition number <10^{10} will already result in a sufficient precision of up to six decimal places.

6.2 New Scientific Achievements

6.2.1 New Singular Value Decomposition Based Genetic Fuzzy System Training Data Set Reduction

As described in [s6] when identifying a system, we have to design a sufficiently exciting trajectory, which will properly expose all singular values of the (linear) system. For a stable equation solution for linear parameters it is needed to have all singular values higher than one.

For solving a linear system of equations it is recommended to use an SVD-based decomposition method before calculating the inverse matrix as for equation (53), but calculating SVD decomposition for large matrices is very processor and memory demanding task, which increases exponentially with the data set size.

Data samples collected along sufficiently exciting trajectories tend to be oversized, thus redundant. In [s12] and [s16] it is shown for a robotic manipulator dynamic model identification, that by using only a reduced number of training data points the same quality of system identification can be reached as with the full set, given that the reduced set is representative enough of the full set, which is equivalent to having a similarly low condition number.

My idea to reduce the fuzzy system training data set comes from the applied linear parameter calculation method of Thesis groups II and III – they rely on SVD based linear equation calculation. To efficiently calculate *n* linear parameters one only needs *n* linearly independent equations of *n* variables – whose parameters are represented by an *n*x*n* matrix. This means that if we could obtain the ideal set of training points we would need not more points than what is minimally required to construct an *n*x*n* matrix of linearly independent rows for our system representation. This is my attempt to select only those input data, which are necessary to construct a good quality system parameters.

To successfully train a FLS of a general form as $f(\boldsymbol{x}) = \sum_{l=1}^{M} \omega_l(\boldsymbol{x},\boldsymbol{b}) \cdot y_l(\boldsymbol{x},\boldsymbol{c})$, which is equivalent to the transformed form of $f(\boldsymbol{x}) = \boldsymbol{A}(\boldsymbol{x},\boldsymbol{b}) \cdot \boldsymbol{c}$ the traning data has to be such that it is necessary fo the matrix $\boldsymbol{A}$ to be of full rank, while the condition number of matrix $\boldsymbol{A}$ is limited by the desired numerical precision.

The FLS training in this case is finding the proper $\boldsymbol{b}$ vector of the nonlinear MF parameters, and finding the $\boldsymbol{c}$ vector of linear consequence parameters. For optimising $\boldsymbol{b}$ one can us a method as described in my thesis II. For LS optimal $\boldsymbol{c}$ vector one can use the SVD transformation property as $\boldsymbol{c} = \boldsymbol{V}\boldsymbol{S}^{-1}\boldsymbol{U}^{\boldsymbol{T}} \cdot \boldsymbol{F}_{full}(\boldsymbol{x})$ for the SVD decomposition of $\boldsymbol{A}_{full} = [\boldsymbol{A}(\boldsymbol{x}_i,\boldsymbol{b})] = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\boldsymbol{T}}$, where $\boldsymbol{F}_{full}(\boldsymbol{x}) = [f(\boldsymbol{x}_i)]$ is the vector of the traning data results $f(\boldsymbol{x}_i)$**,** for the input series data $\boldsymbol{x}_i$, *i=1,..,N;* for *N* being the number of traing data inputs.

The proposal of this paper is to apply a selection algorithm to $[\boldsymbol{x}_i]$ and thus to the $\boldsymbol{F}_{full}(\boldsymbol{x})$ training data set, such that we can determine an arbitrary quality / size balanced training data set $[\boldsymbol{x_j}]$ and thus $\boldsymbol{F}_{red} = [f(\boldsymbol{x_j})]$ for FLS based dynamic model identifications.

THESIS V - DEFINITION:

Without compromising the identification quality it is possible to reduce an oversized training data set *F(**x**)* in a manner that we extract only samples $\boldsymbol{x_j}$ such that the selected input-output training data pairs $\boldsymbol{F}_{red} = [f(\boldsymbol{x_j})]$ maximise the condition number decrease of the $\boldsymbol{A}_{red} = [\boldsymbol{A}(\boldsymbol{x_j}, \boldsymbol{b})]$ of the FLS antecedent matrix.

In this setup we can define an arbitrary condition number limit, which when reached means that no further training samples have to be considered for the defined calculation precision. Also an explicit limit can be set to the number of training data samples (driven by the computation complexity of $\boldsymbol{A}_{red} = \boldsymbol{USV^T}$ decomposition), while we can guaranty that only the most relevant samples are included into the training set – those samples that contribute the most to the calculation precision of $\boldsymbol{c} = [c_{ij}] = \boldsymbol{VS^{-1}U^T} \cdot F_{red}(\boldsymbol{x})$**.**

6.2.2 Implementation of New Singular Value Decomposition Based Genetic Fuzzy System Training Data Set Reduction

The algorithm to select the required minimal training data set – an example application for RM dynamics or multi rotor flight dynamics identification problems is:

0. set the reduced data set$\boldsymbol{A}_{red}$ for our system example $\boldsymbol{A}_{red} \equiv \mathbb{Q}_{red}$ to an empty set and start from the full training data set $\boldsymbol{A}_{full} \equiv \mathbb{Q}_{\boldsymbol{full}}$ and perform the following preparations:

   a. evaluate FLS antecedents by equation (24), (25) and (37) using an uniform, equidistant fuzzy partition defined with a_i=*i/K* for equation (37);

   b. prepare evaluation of linear c_{ij} parameters, by substituting all nonlinear $\mathbb{Q}_{full}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ components of the function/system model to be identified – in our case equation (53);

   c. (optional step) perform the SVD decomposition of $\mathbb{Q}_{full}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \boldsymbol{U} \cdot \boldsymbol{S}_{full} \cdot \boldsymbol{V}^T$ and calculate the reference condition number of the full set as: $cond(\mathbb{Q}_{full}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})) = max(diag(\boldsymbol{S}_{full})/min(diag(\boldsymbol{S}_{full})),$

1. select the j^{th} trajectory point input data $\mathbb{Q}_j(\boldsymbol{q_j}, \dot{\boldsymbol{q}}_j, \ddot{\boldsymbol{q}}_j)$ from the full training set $\mathbb{Q}_{full}$, which the most reduces the condition number of the FLS antecedent, being $\max\left(diag(\boldsymbol{S}_{red\cup j})\right) / \min\left(diag(\boldsymbol{S}_{red\cup j})\right)$ value of the reduced $[\mathbb{Q}_{red}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \cup \mathbb{Q}_j(\boldsymbol{q_j}, \dot{\boldsymbol{q}}_j, \ddot{\boldsymbol{q}}_j)]$ matrix;

2. remove the selected j^{th} trajectory point from the full training set $\mathbb{Q}_{full}$ and add it to the reduced training set $\mathbb{Q}_{red}$;

3. repeat steps 1 and 2, while the condition number of the reduced training set $\mathbb{Q}_{red}$ is above the target value, and there remains any selectable points in the full training set $\mathbb{Q}_{full}$ or the targeted maximum size of reduced training set $\mathbb{Q}_{red}$ is not reached.

Notice that the target condition number of the reduced training set $\mathbb{Q}_{red}$ cannot be set to lower than the reference condition number of the full training set $\mathbb{Q}_{full}$ data set (which cn be determined by step c). The target training data set size cannot be set to lower than the number of c_{ij} linear parameters of the system, as $\mathbb{Q}_{red}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ must not be rank deficient.

6.2.3 Results of New Singular Value Decomposition Based Genetic Fuzzy System Training Data Set Reduction

For uniformly distributed MFs in the fuzzy partition of antecedents the $cond(\mathbb{Q}_{full}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}))$ condition number (cond) of the full training set $\mathbb{Q}_{full}$ and $cond(\mathbb{Q}_{red}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}))$ for the reduced training set $\mathbb{Q}_{red}$ is in Table XX; the condition number change for the first 1170 points out of the total set of 5487 points is presented in Figure 58.

	Full set	**reduced to**	**reduced to**	**reduced to**
training points % reduction to	5487 100%	2743 50%	1170 25%	685 12.5%
cond % increase by	5625 0%	5680 +0.98%	5977 +6.25%	6893 +22.54%

Table XX. Condition number change and size of the reduced trading data set

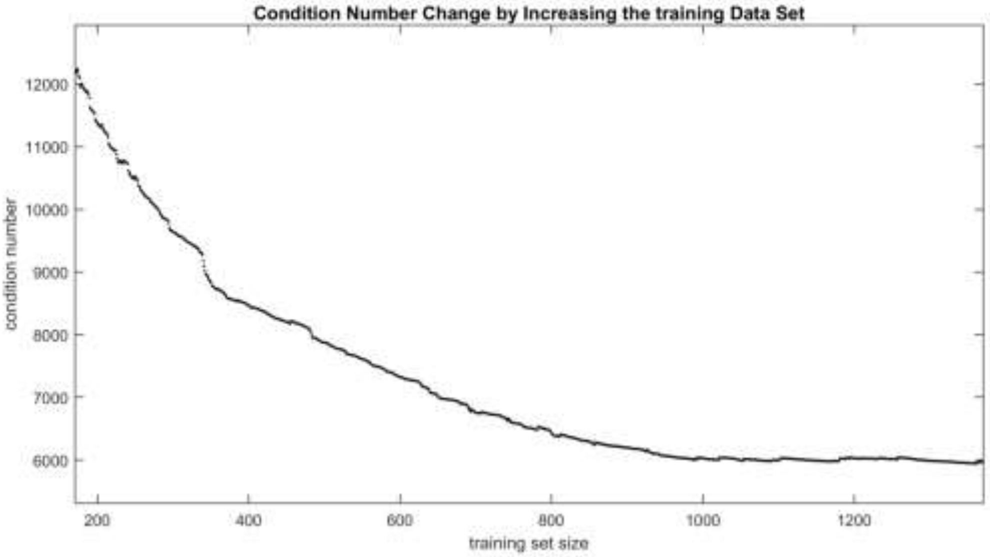


Figure 58.
Condition number changes for the set reduction up to 1200 points

We can observe that my proposed FLS training data set reduction method successfully reduces the multi-rotor flight dynamics data set from 5500 points to below 300 points, while the condition number of the resulting linear system identification problem does not significantly increase. The condition number of 1e+4 corresponds to a 1e-12 numerical precision (assuming 1e-16 numerical precision for computer based calculations) in the calculated linear parameters, which is more than enough for a good quality engineering application. In case we still want to double our precision, we need to double the size of the training data set; notice that it makes no sense to increase the data set above 1000 points, as the identified linear system precision will practically not increase any longer.

By this analysis I conclude that my Thesis V is proven valid.

SUMMARY CONCLUSIONS

New Scientific Achievements

1. *New Vector Comparison Operators*

This paper presents a new vector comparison relation operator, and its extensions that can be used for creating a measurement based new multi-objective ranking operator, which can be the bases for an efficient new multi-objective GA. Also a measurement function is defined for Pareto-dominance. A general measurement based ranking method is proposed. Also a modification of fitness sharing is presented. Numerous multi-objective GA types are evaluated for their performance on GA hard functions.

Each tested GA, no matter which ranking method is used, efficiently finds the close proximity of the true Pareto-front. The proposed new dominance based ranking methods DO and DM both outperform all other tested ranking methods by 20% when it comes to the number of generation evaluations required for convergence, and they also outperform the others by 5-10% when it comes to the number of non-dominated individuals found in the final generation.

Each tested GA, no matter which vector comparison method is used, efficiently finds the proximity of the true Pareto-front. The new vector comparison methods (A, N, Q) outperform the Pareto comparison by 5-15% when it comes to the number of generation evaluations required for convergence, and they also outperform the others by 5-15% when it comes to the number of non-dominated individuals found in the final generation.

2. *New Minimalistic Parametrisation of Zadeh-type Fuzzy Partitions for Function Identification by Unconstrained Tuning*

This paper presents a novel method that simplifies the $\boldsymbol{b_i}$ non-linear parameter optimisation of TSK FLSs based on fuzzy partitions for antecedent MFs like equation (27) that is suitable for unconstrained stochastic and gradient descent based non-linear optimisation, while preserving all the required constraints and properties. All linear parameters of equation (24) are determined by SVD based robust LS method.

The proposed identification method is capable of highly efficient off-line precise identification, and also real-time adaptive fine tuning of fuzzy systems for function approximation or system identification purposes. Furthermore, the proposed minimalistic parameterisation of Zadeh-formed MFs makes it possible to use unconstrained optimisation methods while the initial ordering of MFs and the fuzzy-partitioning properties are preserved.

The presented simple uniform partition based fuzzy precedent definition with SVD-based linear antecedent calculation is a very fast, good enough uniform function approximation technique. The application of my proposed precedent parameter representation enables the application of any numerically efficient unconstrained tuning of the fuzzy system. Applying a gradient-descent like method further improves the identification quality; at a cost of some extra computation effort (usually 15 iterations are satisfactory). Applying an initial efficient GA search for the global optima neighbourhood of the precedent parameters, combined with gradient-based fine toning and SVD-based antecedent parameter calculations result in extremely precise function identifications; at a cost of further extra computation effort (usually <15 generations are needed for a population proportional to the complexity of the problem, proportional to the dimension of the search space and the number of objectives).

This very efficient and minimalistic parametrisation of uniform function approximation fuzzy systems is the starting point of building complex, robust fuzzy system models, which can cope with real life data uncertainties such as the unpredictable aerial environment of an UAV.

3. *New Genetic Fuzzy System Grey-box Modelling of Complex Dynamics Systems*

This paper presents a new method that identifies the RM dynamics through finding the $\boldsymbol{D_{ij}}$ nonlinear functions of equation (39) as TSK FLSs, while calculating $\boldsymbol{D_{ijk}}$ nonlinear functions as in equation (41). All linear parameters of the system are determined by SVD based robust LS method. Nonlinear parameters are evolved by multi-objective GA and fine-tuned by gradient descent method.

This paper presents a new method that identifies the multi-rotor flight dynamics equation (44) $\boldsymbol{D_{ij}}$ components by specially constructed continuous and periodic TSK FLSs, while calculating the $\boldsymbol{D_{ijk}}$ nonlinear functions as in equation (45). All linear parameters of the system are determined by SVD based robust LS method. Nonlinear parameters are evolved by multi-objective GA and fine-tuned by gradient descent method.

The proposed identification method is capable of forming and fine-tuning a soft computing, fuzzy system based dynamic model for a robot manipulator. The number of nonlinear parameters can be kept to minimal and optimised by evolutionary and gradient based methods, too. The value of the linear parameters can be determined by a least squares method. After an initial evaluation the complete identification method is capable of running on-line with a control algorithm if we use an on-line iterative least squares method for the linear parameters [57], while from the background a hybrid evolutionary and gradient based method periodically updates the nonlinear parameters.

The relative value of the maximal error is well within the tolerance level of a model based control algorithms [80]. Parameters identified by this method can be considered as real physical values, in contrast to previous results where some negative numbers appeared for inertia terms.

The proposed identification method is capable of forming and fine-tuning a soft computing, fuzzy system based dynamic model for quadrotors. The quality of identification with the relative torque error being uniformly <10% is suitable for application in model based control algorithms; the torque error is presented in Figure 5. Such good quality UAV flight dynamics models are the prerequisites for quality model based flight control systems.

4. *New Feasible Optimal Harmonic Trajectories of Bounded, Smooth Time Derivatives*

This paper presents a novel harmonic path construction real-time direct algorithm for generating physically feasible, time-and energy optimal, bounded, continuous trajectories that can reach any target displacement with a known minimal error. These trajectories can be designed to arbitrary smoothness – depending on system requirements; they are to be designed smooth up to the 5th time derivative of displacement for multi-rotor UAV trajectories. The term (n times) smooth is used as in being equivalent to having continuous (n^{th}) time derivative. The requirement for feasible trajectories of having minimum 5 times smooth displacement functions in case of UAV is proven. Effects of trajectory discontinuities on system state oscillations are studied in details. It is proven that the proposed harmonic trajectories of appropriate smoothness (defined by the system and control actuator dynamics) do not generate system state oscillations.

The proposed trajectory design method is capable of forming bounded, smooth, energy efficient and time optimal trajectories with a single pass algorithm using closed formulas. The design method is defined and validated on an example for a multi-rotor UAV path planning, where a single parameter controls the trajectory dynamics, as presented in Figure 32.

Dynamic transient properties and energy efficiency of the trajectory can be tuned with a single parameter, but the feasibility of torque transients must not be dismissed along this optimization. The resulting trajectory is always the time optimal solution, which complies with all defined limits.

5. *New Singular Value Decomposition Based Genetic Fuzzy System Training Data Set Reduction*

This paper presents a novel method that reduces the necessary training data set size for fuzzy identification of complex dynamic systems. The method is based on finding the minimal subset of the training data, which most efficiently minimises the corresponding condition number of the linear system subject to SVD decomposition when identifying the optimal linear parameters of the system.

The proposed GFS training data set reduction method, while maintaining the quality of the identification process, is capable of significantly reducing the number of necessary training data points, and thus significantly increases the identification process performance. The method is defined and validated on quadrotor dynamic model identification with GFS, where less than 20% of data points give more than 80% of contribution to the system condition number. A typical rate of condition number change for the most significant 25% of data points is presented in Figure 58.

The training data set reduction to 1/20th of the full set significantly increases the identification process speed, while the proposed reduction method ensures that the identification result quality does not deteriorate to below a pre-defined minimum precision level.

This method implicitly provides information on the quality of the training data set. The condition number is of acceptable magnitude only for full rank matrices. Rank deficiency in case of the proposed fuzzy identification methods means that there is no sufficient data to meaningfully define consequent values for every rule; thus when using such model for control purposes we cannot achieve uniform stability – a model built on a rank deficient fuzzy system is not stable for the complete operational space, even if the antecedent fuzzy partition uniformly cover the complete input space.

Application Possibilities of Results

The main goal of this work is to create new and improve existing tools, by which the complete autonomy with obstacle avoidance of UAV navigation can be enhanced.

My first thesis group gives a set of new tools for evolving, searching near-optimal parameters of complex systems such as fuzzy models of system dynamics as in navigation dynamics of UAV.

My second thesis group gives a new tool for minimalistic representation of fuzzy model parameters and their unconstrained tuning for precise function approximation in modelling complex system dynamics as in navigation dynamics of UAV.

My third thesis group gives a new tool for efficient complete fuzzy modelling of continuous and periodic complex nonlinear dynamics systems as in navigation dynamics of UAV.

My fourth thesis group gives a new tool for feasible optimal trajectory design, which can real-time generate trajectory parametrisations while obeying all the kinematic constraints, such as parametrisation for any geometric UAV path with velocity and acceleration constraints.

My fifth thesis group gives a new tool for efficient genetic fuzzy modelling by reducing the training data set to minimum, while guaranteeing the prescribed quality of the solution.

Along my results to improve the autonomy of UAV navigation I propose to:

- start from a flying UAV, keep the high level strategic way-point selection algorithm

- for planning the exact feasible optimal trajectory between two way-points use my method described in thesis 4

- at the initial stage use the existing control mechanism to track such feasible optimal trajectories and collect measurement signals datasets consisting of at least 3D position, 3D orientation data paired to exact (4 in case of quadrotor) motor rotation velocities for each time sample; if sensors can provide, further data can be collected such as position and orientation velocities and accelerations, motor currents or voltages

- for minimising the training data set size use my method described in thesis 5

- using my methods described in thesis 1 and 2 design a fuzzy system structure as described in thesis 3 to precisely model the UAV flight dynamics along the reduced trading data set

- replace the UAV control system to a back stepping (computed torque) controller, which uses a fuzzy reference model obtained in the previous step

A further improvement possibility exist in adapting the computed torque control algorithm in a manner that it does not apply a simple PID action to the decoupled double integrators, but instead actually calculates a feasible optimal harmonic micro-trajectory which, when super-positioned to the original trajectory, compensates for the occurred trajectory error. This way high speed smooth obstacle avoidance can be achieved, be it a full evasive manoeuvre or just a velocity modification.

As all the tools I have developed are general, they have much broader application possibilities.

1. *Multi-objective Genetic Algorithms with Quality-dominance and Measurement-based Ranking*

The proposed vector comparison operators are strict partial order binary endo-relations, being irreflexive, antisymmetric and transitive – thus they are uniformly usable in any mathematical or engineering process where a decision is to be made based on multipole criteria. The proposed metrics, including those for Pareto and weighted sum operators can be the bases for any ranking process, not just stochastic search, evolutionary algorithms and genetic algorithms. These proposed methods are computational efficient and provide detailed information on the quality, the nature and extent of difference between vectors of the same kind.

These new ranking and vector comparison methods can be freely use in any mathematical, engineering, economics or any other field, when objects of multiple properties are to be objectively compered or ranked. They are very much needed when the task is to optimise very complex, highly nonlinear system as fuzzy UAV flight dynamics models.

Based on the presented analysis my conclusion is that if one does not want to mess with vector comparisons, then the simple weighted sum of objectives will still do the trick; the only recommendation I give for this simple approach is to use the dominance approach to ranking (DO or DM) – measure by how much an individual is better than the others (and not by how much it is worse than the others), as this is a more efficient approach – observe the yellow highlighted D.DO.GA of Table IV.

Finally to offer an alternative to all those that still insist on using the classical Pareto vector comparison for multi-objective GAs: please observe the orange marked P.DM and P.DO GAs in Table IV to conclude that it is still more efficient to base the rank of an individual on the number of how many individuals it dominates (dominance based ranking) instead of looking for how many individuals do not dominate it (non-dominance based ranking).

2. *Free Parametrisation Method for Unconstrained Tuning of Zadeh-type Fuzzy Partitions*

My proposal for a successful fuzzy identification strategy is to take the ‘GAzFLS’ method as an off-line preliminary identification method, apply the results while keeping a continuous real-time ‘LinLSzFLS’ update mechanism in place for continuous fine tuning with fresh measurements, thus ensuring adaptability of the system.

The proposed fuzzy partition representation method performs exceptionally well when it comes to precision and reduced complexity (simplicity) of the solution format – low number of MFs and fuzzy rules. The global search of nonlinear parameters can be performed in a satisfactory fast manner by a well-constructed GA; gradient descent methods can simply and efficiently fine-tuned the system. With all linear parameters being LS optimal, the final quality of the identification is very good. Since the FLS complexity is reduced (low number of MFs and fuzzy rules) the model evaluation is fast. After the necessary offline pre-processing to calculate the system ***a*** then ***b*** and ***c*** parameters, for each online input data we need to perform only 5*(number of system inputs) multiplications and maximum 7*(number of system inputs), in average 4*(number of system inputs) additions; this can all be performed online in real time on practically any simple processor.

The number of multiplications and additions comes from these considerations:

- one input triggers maximum 2 MFs in a fuzzy partition;

- 2 MFs of equation (25) take maximum 3 additions (subtractions) and 2 multiplications (1 multiplication and 1 division); notice that the MF denominator does not take a subtraction for every online input calculation, it is constant, thus it can be pre-calculated; and actually the average number of additions is only 2;
- for each input it takes 2 multiplications for each antecedent of equation (22);
- for each input it takes 1 multiplication and 1 addition for each fuzzy consequence evaluation (24);

It is interesting to note that the proposed antecedent structure has all the positive properties of a second order B-spline antecedent. It has the same minimal number of parameters. Its derivative is continuous up to the second order. The evaluation of the complete rule base for every input can be omitted the same way as for B-splines. Furthermore, the proposed formulation has other benefits over B-splines: no iterative evaluation of the MF values is required. The MF parameters can be directly tuned by gradient based methods. No constraints have to be taken into consideration for its parameters. The consistency of the linguistic values remains intact throughout any fine-tuning so incorporated human knowledge can be fine-tuned without loss of meaning. Instead of nonlinear LS the proposed formulation and unconstrained gradient based method can be used for tuning the consequent parameters of a Mamdani type FLS, too.

3. *Grey-box Genetic Fuzzy System Modelling and Control of Complex Dynamic Systems*

The proposed general identification method is capable of forming and fine-tuning a soft computing, fuzzy system based dynamic models for any mechanical system that can be described by Euler-Lagrange equations, including but not limited to robotic manipulators, cranes, even for flight dynamics. The structure of the model is such that we can guarantee continuity of the model output and also periodicity if required. The model continuity and its bounded nature can be used for mathematically proving stability of control systems using the model. Periodicity of the model is needed to ensure the proper natural behaviour after full circle turns in flight dynamics or with robotic arms where a joint (typically a wrist) can rotate in one direction more than 360 degrees.

This method can be freely used for large, complex systems of many heavily coupled variables, as the number of nonlinear parameters is kept to minimum, and influenced only by the inertia matrix size; the complexity of the Coriolis and the centrifugal components do not introduce any new variables, as they are fully defined by and calculated from inertia components.

The structure of the model is such that after an initial evolutionary nonlinear parameter optimisation, it is possible to continuously, real-time fine tune the linear parameters of the model thus resulting in an adaptive model.

4. *Feasible Optimal Harmonic Trajectories of Bounded, Smooth Time Derivatives*

The proposed trajectory design method results in real-life feasible smooth, bounded torque transients, which is energy efficient for control signal design; while providing a flexible interface to arbitrary velocity, acceleration, jerk and snap limit enforcement. For such cases the trajectory is designed in a way to hold the maximal velocity, acceleration, jerk and snap, so that the desired lesser derivative maximum values are reached without any increase in higher derivatives. Figure 59 presents such trajectory, where the maximum snap is held for 1 second (yellow vertical lines), maximum jerk for 2 seconds (orange double vertical line), maximum acceleration for 4 seconds (pink rectangles) and the maximum velocity for 8 seconds (hollow red rectangles).

Reduction of this method is strait forward for more simple systems where it is enough to have smooth trajectories up to the 3rd time derivative of displacement. The notion of time and energy optimality is not used in some mathematics theory manner but in real life physically feasible engineering manner.

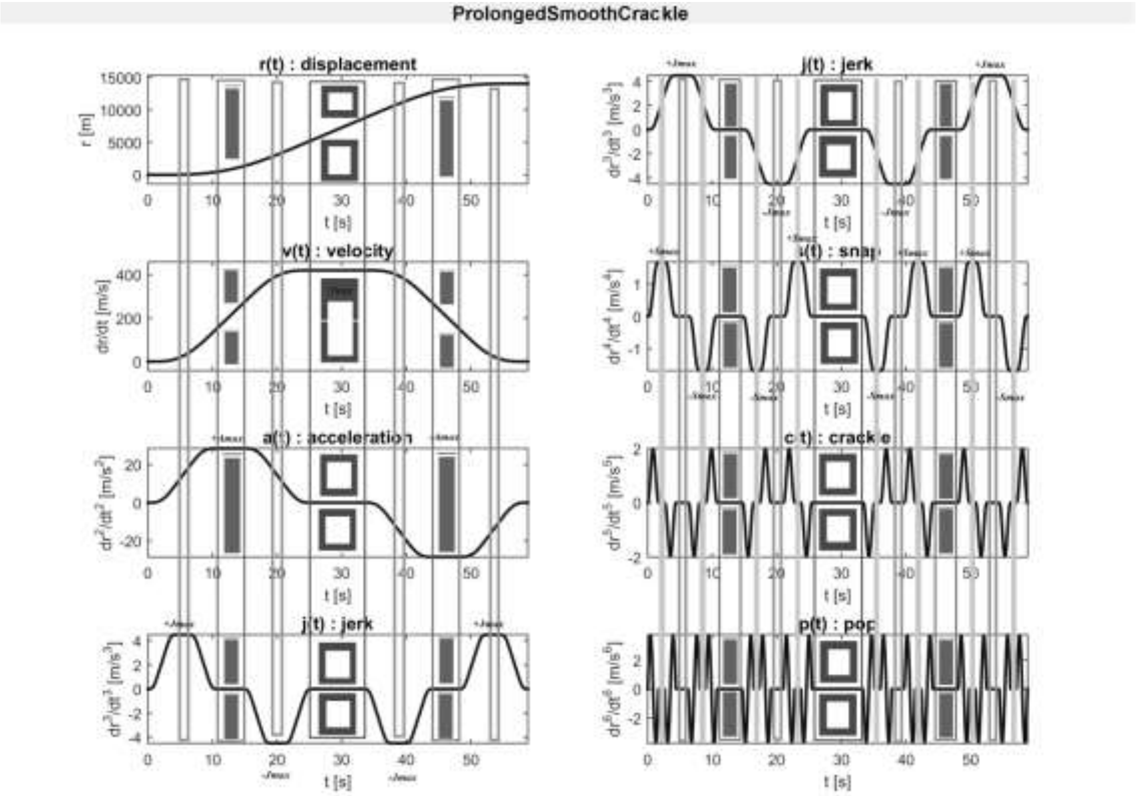


Figure 59.
Design of increased maximal trajectory values

The same basic principle of accounting for system oscillations and the actuator dynamics when planning for system trajectories can be also applied to a crane model and any other than electro motor actuated system, by replacing equations (55) to appropriate ones, and then evaluating their transient behaviour. When the actuator dynamics and its relation to the system trajectory is known, one can use the algorithm and the method described in this paper to design trajectories of required transient dynamics and smoothness by replacing equation (62) to the appropriate one [s6].

These real time generated trajectories can be applied as parametrisation to any vector function defined path *f(s)=(x(s),y(s),z(s)),* which is chosen to accomplish the desired RM or crane task; also for complex UAV missions of arbitrary designed paths, even those including sudden unplanned changes as in obstacle avoidance. When determining the constraints on trajectory derivatives, one has to take into consideration both the system limits (39) or (43) and curvature properties of *f(s)*. The capability of real time trajectory generation with arbitrary constraints on displacement derivatives makes this trajectory design method especially suited for tactical re-planning of flight trajectories.

This paper presents results with feed-forward control scheme, but the trajectory design is applicable to any scheme including feedback setups. Naturally the poles of the system will change in a feedback loop, this will result in changed system dynamics – the period of a critical aperiodic transient will be different. This transient period can either be calculated as in equation (63) or simply measured as in equation (62) and then used as the parameter P in equations (64) and (65).

5. Singular Value Decomposition Based Genetic Fuzzy System Training Data Set Reduction

The proposed training data set reduction method is a general procedure usable for all identification procedures where a significant part of the approximation is linear – as in linear parameter approximation in TSK fuzzy system consequent part. For complex system identification problems we must be sure that the training data set is sufficiently exciting – that it reveals all the typical modes of the system. Datasets of sufficiently exciting system trajectories tend to be oversized. Oversized data is expensive to evaluate, especially in iterative and evolutionary search methods.

The need to reduce the training dataset is real, but for many applications the quality of the result is of outmost importance as in model based control of UAV flight dynamics we must not allow for any unknown, uncontrolled states. Using this proposed training data reduction method we can ensure that we have full control over the quality and also the size of the training data, thus full control off the uniform quality of our UAV flight dynamics model and implicitly of the stability of the model based flight control.

REFERENCES

[1] C. An, C. Atkeson, J. Hollerbach, “Model-Based Control of a Robot Manipulator,” The MIT Press, Cambridge, 1988.

[2] K Balázs, L T Kóczy, “Constructing Dense Fuzzy Systems by Adaptive Scheduling of Optimization Algorithms,” Proc. IFSA 2013 World Congress, Edmonton, June 24-28, 2013, pp. 280-285.

[3] J.T. Betts, “A survey of Numerical Methods for Trajectory Optimization,” Mathematics and Engineering Analysis, Boeing Information and Support Services, Washington, 1998.

[4] J.E. Bobrow, S. Dubowsky, J.S. Gibson, “Time-optimal control of robotic manipulators along specified paths,” Proc Int. J. Robotic Research, Vol.4, Issue 3, pp. 3-17, 1985.

[5] Y. Bouktri, M. Haddad, T. Chettibi, “Trajectory planning for quadrotor helicopter,” Proc. 16th Mediterranean Conference on Control and Automation, Ajaccio, France, 2008.

[6] G.E. Box, G.M. Jenkins, “Time series analysis, forecasting and control,” Holden Day, 1970.

[7] G. R. Bradski, “Computer vision face tracking for use in a perceptual user interface,” Intel Technology Journal, no. Q2, 1998.

[8] T. Bresciani, “Modelling, Identification and Control of a Quadrotor Helicopter,” Department of Automatic Control, Lund University, October 2008.

[9] A. Buruzs, M. Hatwágner, C. Pozna, L. T. Kóczy, “Advanced Learning of Fuzzy Cognitive Maps of Waste Management by Bacterial Algorithm,” Proc. IFSA 2013 World Congress, Edmonton, June 24-28, 2013, pp. 890-895.

[10] G. Carrillo, D. López, R. Lozano, C. Pégard “Modeling the Multi-rotor Mini-Rotorcraft,” Quad Rotorcraft Control, Springer-Verlag, London, 2013.

[11] D. Chwa, “Nonlinear Tracking Control of 3-D Overhead Cranes Against the Initial Swing Angle and the Variation of Payload Weight,” IEEE Transactions on Control Systems Technology, 2009, Vol.17, Issue 4, pp. 876-883.

[12] J. Coelho, R. Neto, C. Lebres, V. Santos, “Application of Fractional Algorithms in Control of a Quad Rotor Flight,” Proc. 2nd Conference on Nonlinear Scienceand Complexity, Porto, Portugal, July 28-31, 2008, pp. 1-12.

[13] C. A. Coello Coello, “A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques,” Knowledge and Information Systems, Springer-Verlag, Singapore, August 1999, Vol.1, Issue 3, pp 269–308.

[14] I. Cowling, “Towards Autonomy of a Quadrotor UAV,” Ph.D. dissertation, Cranfield Univ., 2008.

[15] C. Coza, C.J.B. Macnab, “A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization,” Fuzzy Information Processing Society, IEEE NAFIPS 2006. pp. 454-458.

[16] Z. Dányádi, P. Földesi, L.T. Kóczy, "Solution of a Fuzzy Resource Allocation Problem by Various Evolutionary Approaches," Proc. IFSA 2013 World Congress, Edmonton, June 24-28, 2013, pp. 807-812.

[17] K. Deb, D. E. Goldberg, "An investigation of niche and species formation in genetic function optimisation," Proc. Third International Conference on Genetic Algorithms, 1989, pp. 42-50.

[18] N. Srinivas, K. Deb, "Multiobjective Optimisation Using Nondominated Sorting in Genetic Algorithms," Proc. Evolutionary Computation, 1994, Vol.2, Issue 3, pp. 221-248.

[19] K. Deb, "Evolutionary Algorithms for Multi-Criterion Optimisation in Engineering Design," Proc. Evolutionary Algorithms in Engineering and Computer Science (EUROGEN'99), John Wiley and Sons, London, 1999, pp. 135-161.

[20] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II," IEEE Transactions on Evolutionary Computation, 2002, Vol.6, Issue 2, pp. 182-197.

[21] A. K. DeJong, "An Analysis of the Behaviour of a Class of Genetic Adaptive Systems," Ph.D. Thesis, Univ. of Michigan, 1975.

[22] T. Fraichard A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," IEEE Trans. on Robotics and Automation, 2004, Vol.20, Issue 6, pp. 1025–1035.

[23] C. M. Fonseca, P. J. Fleming, "Multiobjective Optimisation and Multiple Constraint Handling with Evolutionary Algorithms I: A Unified Formulation," Technical Report 564, Univ. of Sheffield, Sheffield, UK, 1995.

[24] C. M. M. da Fonseca, "Multiobjective genetic algorithms with application to control engineering problems," Ph.D. Thesis, Dep. of Automatic Control and System Engineering, Univ. of Sheffield, 1995.

[25] K. Fukunaga, L. Hostetler: "The estimation of the gradient of a density function, with applications in pattern recognition," Information Theory, IEEE Trans.,1975, Vol.21, Issue 1, pp. 32 – 40.

[26] F. Gers, D. Eck, J. Schmidhuber, "Applying LSTM to Time Series Predictable Through Time-Window Approaches," Neural Nets WIRN Vietri-01, Perspectives in Neural Computing, Section 6, 2002, pp, 193-200.

[27] D.E Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," Addison Wesley Publishing Company, 1989.

[28] D. E. Goldberg, et all. "Genetic Algorithms: A Bibliography," IlliGAL Report No.97002, University of Illinois at Urbana-Champaign, 1997.

[29] J. C. Goldin, "Perching Using a Quadrotor with Onboard Sensing," M.S. thesis, Utah State Univ., 2011.

[30] H. Hellendoorn, D. Driankov (Eds.), "Fuzzy Model Identification, Selected approaches", Springer-Verlag Berlin Heidelberg, 1997.

[31] J.R. Jang, C. Sun, E. Mizutani, "Neuro-Fuzzy and Soft Computing, A Computational Approach to learning and Machine Intelligence," Prentice-Hall, 1997.

[32] W. B. Langdon, “Genetic Programming and Data Structures: Genetic Programming + Data Structures = Automatic Programming! Genetic Programming, Kluwer, Boston, 1998.

[33] C. Lebres, V. Santos, N. M. Fonseca Ferreira, J. A. Tenreiro Machado, “Application of Fractional Controllers for Quad Rotor,” Nonlinear Science and Complexity, Part 6, Springer, 2011, pp. 303-309.

[34] Y. Li, G. Wang, “Quadrotor Airship Modeling and Simulation,” Proc. Sensors & Transducers, Vol.157, Issue 10, October 2013, pp. 306-316.

[35] R. Lozano, “Unmanned Aerial Vehicles”, ISTE Ltd, London, 2010.

[36] M. Mackey, L. Glass, “Oscillation and chaos in a physiological control system,” Science 197,287, 1977.

[37] D. Mellinger, V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” IEEE Int. Conf. on Robotics and Automation (ICRA2011), 2011, pp. 2520 –2525.

[38] G. Mester, “Distance Learning in Robotics,” Proc. The Third International Conference on Informatics, Educational Technology and New Media in Education, Sombor, Serbia and Montenegro, 2006, pp. 239-245.

[39] G. Mester, “Improving the Mobile Robot Control in Unknown Environments,” Proc. Conf. YUINFO’ 2007, Kopaonik, Serbia, 2007, pp. 1-5.

[40] G. Mester, A. Rodic, “Modeling and Navigation of an Autonomous Quad-Rotor Helicopter,” E-society Journal: Research and Applications, July 2012, Vol.3, No.1, pp. 45-53.

[41] G. Mester, A. Rodic, “Navigation of an Autonomous Outdoor Quadrotor Helicopter,” Proc. 2nd Int. Conf. on Internet Society Technologie and Management ICIST, Kopaonik, Serbia, 2012, pp. 259-262.

[42] G. Mester, A. Rodic, J. Stepanic, “Nonlinear Control of Aerial Robotics,” invited talk, Workshop Modern Approach to Product Development and Business Improvement, Balatonfüred, Hungary, 16-19th May 2013.

[43] G. Mester, A. Rodic, “Négyrotoros robothelikopter modellje és írányítása,” A Magyar Tudomány Napja a Délvidéken 2012, Vajdasági Magyar Tudományos Társaság, szerkesztő: Szalma József, Újvidék, Szerbia, 2013, pp. 469-476.

[44] G. Mester, A. Rodic, “Simulation of Quad-rotor Flight Dynamics for the Analysis of Control, Spatial Navigation and Obstacle Avoidance,” Proc. 3rd International Workshop on Advanced Computational Intelligence and Intelligent Informatics (IWACIII 2013), Shanghai, China, October 2013, pp. 1-4.

[45] G. Mester, “Backstepping Control for Hexa-Rotor Microcopter,” Acta Technica Corviniensis – Bulletin of Engineering, Tome VIII, Fascicule 3, July – September 2015, pp. 121-125.

[46] G. Mester, “Modeling of Autonomous Hexa-Rotor Microcopter,” Proc. 3rd International Conference and Workshop Mechatronics in Practice and Education, MechEdu 2015, Subotica, Serbia, May 14-16, 2015, pp. 88-91.

[47] A. K. Mishra, Y. Mohapatra, A. K. Mishra, “Multi-Objective Genetic Algorithm: A Comprhensive Survey,” Int Journal of Emerging Technology and Advanced Engineering, 2013, Vol.3, Issue 2, pp. 81-92.

[48] M.Murugan, R.Jeyabharath and P.Veena, “Stability Analysis of BLDC Motor Drive based on Input Shaping,” Int. Journal of Engineering and Technology, 2013, Vol.5, Issue 2, pp. 1169-1177.

[49] A. Naghash, M. Naghshineh, A. Honari, “Minimum Time Trajectory Optimisation for Flying a Quadrotor in an 8-shaped Path,” Int. Micro Air Vehicle Conf. and Flight Competition (IMAV2013), Toulouse, 2013.

[50] S. Noda, H. Masuta, H. Lim, “The Fuzzy Position Control for the Four Rotor Flying Robot,” IEEE SCIS-ISIS 2012, pp. 1238 -1243.

[51] M. A. Olivares-Mendez, I. F. Mondragón, P. Campoy, L. Mejías, C. Martinezm, “Aerial Object Following Using Visual Fuzzy Servoing,” Proc. 1st Workshop on Research, Development and Education on Unmanned Aerial Systems (REDUAS 2011), Centro Avanzado de Tecnologías Aeroespaciales (CATEC), Seville, Spain, 2011, pp. 61-70.

[52] M. A. Olivares-Mendez, L. Mejias, P. Campoy, I. Mellado-Bataller: “Quadcopter See and Avoid Using a Fuzzy Controller,” Proc. 10th International FLINS Conf. on Uncertainty Modeling in Knowledge Engineering and Decision Making (FLINS 2012),World Scientific, Istanbul, Turkey, 2012.

[53] A. Osyczka, “Multicriteria optimisation for engineering design,” John S. Gero, edt. Design Optimisation, Academic Press, 1985, pp 193-227.

[54] V. Pareto. “Cours D'Economie Politique”, Vol.I-II, F. Rouge, Lausanne, 1896.

[55] I. Petrusev, A. Rakic, “Simple Fuzzy Solution for Quadrotor Attitude Control,” Proc. 12th Symposium on Neural Network Applications in Engineering (NEUREL), Belgrade, 2014.

[56] I. Petruševski, A. Rakić, “Simple Fuzzy Solution for Qadrotor Attitude Control,” IEEE NEUREL Symp. Trans., 2014.

[57] W. H. Press, et.all, “Numerical Recipes in C, The Art of Scientific Computing,” Cambridge University Press, Cambridge, 1990.

[58] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, E. F. Mishchenko, “The mathematical theory of optimal processes,” Interscience Publishers, New York, 1962.

[59] C. Richter, A. Bry, N. Roy, “Polynomial Trajectory Planning for Quadrotor Flight,” Proc. Int. Symp. Robotics Research (ISRR), Robotics Research, 2013, pp. 649-666.

[60] A. Rodić, G. Mester, “Modeling and Simulation of Quad-Rotor Dynamics and Spatial Navigation,” Proc. 9th IEEE Int. Symp. on Intelligent Systems and Informatics SISY 2011, Subotica, Serbia, 8–10 September, 2011, pp 23-28.

[61] A. Rodić, G. Mester, “Remotely Controlled Ground-Aerial Robot-Sensor Network for 3D Environmental Surveillance and Monitoring,” invited talk, TAMOP 422 Workshop, Szeged, Hungary, 2011.

[62] A. Rodić, G. Mester, "The Modeling and Simulation of an Autonomous Quadrotor Microcopter in a Virtual Outdoor Scenario," Acta Polytechnica Hungarica, 2011, Vol.8, No.4, pp. 107-122.

[63] A. Rodić, G. Mester, I. Stojković, "Qualitative Evaluation of Flight Controller Performances for Autonomous Quadrotors," in E. Pap (Ed.): Intelligent Systems: Models and Applications, Springer-Verlag Berlin Heidelberg, 2013, Vol.3, pp. 115–134.

[64] A. Rodić, G. Mester, "Control of a Quadrotor Flight," Proc. ICIST Conf., Kopaonik, Serbia, 2013, pp. 61-66.

[65] A. Sanchez, V. Parra-Vega, O. Garcia, F. Ruiz-Sanchez, L. E. Ramos-Velasco, "Time-Parametrization Control of Quadrotors with a Robust Quaternion-based Sliding Mode Controller for Aggressive Maneuvering," Proc. European Control Conf. (ECC), Zürich, 2013.

[66] T. Sangyam, P. Laohapiengsak, W. Chongcharoen, I. Nilkhamhang, "Path Tracking of UAV Using Self-Tuning PID Controller Based on Fuzzy Logic," SICE Annual Conf. 2010, pp. 1265-1269.

[67] M. Spong, M. Vidyasagar, "Robot Dynamics and Control," John Wiley and Sons, Inc. 1989.

[68] R. Stengel, "Flight Dynamics," Princeton University Press, Cloth, 2004.

[69] J. Stepanic, G. Mester, J. Kasac, "Synthetic Inertial Navigation Systems: Case Study of Determining Direction," Proc. 57th ETRAN Conf., Zlatibor, Serbia, June 3-6, 2013.

[70] M. Sugeno, "Fuzzy Identification of Systems and its Applications to Modeling and Control," IEEE Trans. on Systems, Man, and Cybernetics SMC-15(1), 1985, pp. 116-132.

[71] M. Sugeno at al. "Helicopter flight control based on fuzzy logic," Proc. 1st Int. Fuzzy Engineering Symposium, Yokohama, 1991, pp. 1120-1121.

[72] H. Takagi, M. Sugeno, "Fuzzy Identification of Systems amd its Application to Modelling and Control," IEEE Trans. on Systems, Man and Cybernetics, SMC-15(1), 1985. pp. 116-132

[73] N. Varminska, D. Chablat, "Optimal Motion of Flexible Objects with Oscillations Elimination at the Final Point," Eucomes, Springer, 2016.

[74] G. Venu, Dr. S. T. Kalyani, "A New Topology for Speed control of Sensor less BLDC Motor with Reduced Commutator Switches and Improved Input Power Factor," Int. Journal Of Engineering And Computer Science, Vol.3, Issue 11, 2014, pp. 9243-9247.

[75] G. E. Veselov, A. A. Sklyarov, S A. Sklyarov, "Synergetic approach to quadrotor helicopter control with attractor-repeller strategy of nondeterministic obstacles avoidance," Proc. 6th Int. Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), St. Petersburg, 2014.

[76] L. Wang, "Adaptive Fuzzy Systems and Control, Design and Stability Analysis," PTR Prentice Hall, 1994.

[77] F. Yacef1, O. Bouhali, M. Hamerlain, “Adaptive Fuzzy Backstepping Control for Trajectory Tracking of Unmanned Aerial Quadrotor,” Int. Conf. on Unmanned Aircraft Systems (ICUAS), Orlando, 2014.

[78] Z. Y. Zhao, M. Tomizuka, S. Isaka. “Fuzzy gain scheduling of PID controllers,” IEEE Trans. on Systems, Man and Cybernetics, Vol.23, No.5, 1993, pp. 1392 -1398.

[79] E. Zitzler, K. Deb, L. Thiele, “Comparison of Multiobjective Evolutionary Algorithms on Test Functions of Different Difficulty,” Genetic and Evolutionary Computation Conf. (GECCO-99): Bird-of-a-feather Workshop on Multi-criterion Optimization Using Evolutionary Methods, 1999.

[80] The ZODIAC, “Theory of Robot Control,” Springer-Verlag London Ltd., 1996.

[81] R. Zwahlen, T. Chang, “Feedforward Speed Control of Brushless DC Motors with Input Shaping,” The 33rd Annual Conf. IEEE Industrial Electronics Society (IECON), Taipei, Taiwan, Nov. 5-8, 2007.

[82] J.P. Powell, R. Palacín, “Passenger Stability Within Moving Railway Vehicles: Limits on Maximum Longitudinal Acceleration,” Urban Rail Transit, Springer Nature, Berlin Heidelberg, Vol.1, Issue 2, 2015, pp.95-103

[83] G. Mester, S. Pletl, G. Pajor and I. Rudas, “Adaptive Control of Robot Manipulators with Fuzzy Supervisor Using Genetic Algorithms”, Proc. Int. Conf. on Recent Advances in Mechatronics (ICRAM’95), Bogazici University Bebek, Istanbul, Turkey, 1995, Vol. 2, pp. 661–666

[84] G. Mester, “Neuro-Fuzzy-Genetic Trajectory Tracking Control of Flexible Joint Robots”, Proc. I ECPD Int. Conf. on Advanced Robotics and Intelligent Automation, 1995, Athens, Greece, pp. 93-98

[85] G. Mester, S. Pletl, G. Pajor, D. Basic, “Adaptive Control of Rigid-Link Flexible-Joint Robots”, Proc. 3rd Int. Workshop of Advanced Motion Control, Berkeley, USA, 1994, pp. 593-602

[86] G. Mester, S. Pletl, G. Pajor, Z. Jeges, “Flexible Planetary Gear Drives in Robotics”, Proc. Int. Conf. on Industrial Electronics, Control, Instrumentation and Automation - Robotics, CIM and Automation, Emerging Technologies (IEEE IECON '92) , 1992, Vol. 2, San Diego, USA, pp. 646-649

Publications in Support of Thesis

[s1] A. Nemes, “Synopsis of Soft Computing Techniques Used in Quadrotor UAV Modelling and Control,” Interdisciplinary Description of Complex Systems, 2015, Vol.13, No.1, pp. 15-25.

[s2] A. Nemes, “New Genetic Algorithms for Multi-objective Optimisation,” Proc. 1st Int. Symp. of Hungarian Researchers on Computational Intelligence (HUCI 2000), Budapest, 2000.

[s3] A. Nemes, “Function Identification by Unconstrained Tuning of Zadeh-type Fuzzy Partitions,” Proc. 2nd Int. Symp. of Hungarian Researchers on Computational Intelligence (HUCI 2001), Budapest, 2001.

[s4] A. Nemes, “System Identification Based on Multi-Objective Optimisation and Unconstrained Tuning of Zadeh-type Fuzzy Partitions,” Proc. 2nd IEEE Int. Symp. Intelligent Systems and Informatics (SISY 2003), Subotica, Serbia, 2003.

[s5] A. Nemes, “Dynamic Modelling of Robot Manipulators by Zadeh-type Fuzzy Partitions,” Proc. 4th Int. Symp. Hungarian Researchers on Computational Intelligence (HUCI 2003), Budapest, 2003.

[s6] A. Nemes, “Continuous Periodic Fuzzy Logic Systems and Smooth Trajectory Planning for Multi-Rotor Dynamic Modeling,” Acta Polytechncia Hungarica, 2016, Vol.13, No.6, pp. 215-234.

[s7] A. Nemes, “Fuzzy-Genetic Control of Quadrotors Unmanned Aerial Vehicles,” Interdisciplinary Description of Complex Systems, 2016, Vol.14, No.2, pp. 223-235.

[s8] A. Nemes, “Genetic Algorithm-Based Adaptive Fuzzy Logic Systems for Dynamic Modeling of Quadrotors,” Proc. 3rd Int. Conf. MechEdu, 2015, 14-15 May, Subotica, Serbia, pp. 96-103.

[s9] A. Nemes, “Genetic Fuzzy Identification Method for Quadrotor UAVs,” Annals of Faculty of Hunedoara – International Journal of Engineering, 2015, Tome XIII, Fascicule 3, pp. 257-264.

[s10] A. Nemes, B. Lantos, “Genetic Algorithms-Based Fuzzy Logic Systems for Dynamic Modeling of Robots,” Periodica Polytechnica Electrical Engineering, 1999, Vol.43, No.3, pp. 177-187.

[s11] A. Nemes, B. Lantos, “Optimization of Fuzzy Logic Systems for Gray-Box Dynamic Modeling of Robot Manipulators by Genetic Algorithms,” Proc. IEEE Int. Conf. on Intelligent Engineering Systems INES'99, 1999. pp. 353-358.

[s12] A. Nemes, B. Lantos, “Training Data Reduction for Optimisation of Fuzzy Logic Systems for Dynamic Modelling of Robot Manipulators by Genetic Algorithms,” Proc. IEEE Instrumentation and Measurement Technology Conf. (IMTC 2001), 2001, Vol.3, pp. 1418-1423.

[s13] A. Nemes, G. Mester, “Unconstrained Evolutionary and Gradient Descent-Based Tuning of Fuzzy partitions for UAV Dynamic Modeling,” FME Trans., 2017, Vol.45, No.1, pp.1-8

[s14] A. Nemes, G. Mester, “Energy Efficient Feasible Autonomous Multi-Rotor Unmanned Aerial Vehicles Trajectories,” Proc. 4th Int. Scientific Conf. on Advances in Mechanical Engineering (ISCAME), 2016, Vol.1, pp. 369-377.

[s15] A. Nemes, “Feasible Time and Energy Optimal, Minimum Oscillations Trajectory Design,” Proc. XXXII Kandó Conf. (KSC), 2016

[s16] A. Nemes, “Robotkarok leegyszerűsitett szürkedoboz identifikációja fuzzy függvényekkel,” Proc. MicorCAD 2001, Miskolc, Hungary, 2001.

Further Publications

[87] G. Mester, A. Nemes, S. Pletl, T. Mester, “Soft Computimg metoda za optimalno modeliranje buke,” Proc. YOUINFO’99, Kopaonik, Yugoslavia, 1999.

[88] G. Mester, S. Pletl, A. Nemes, T. Mester, “Structure Optimization of Fuzzy Control Systems by Multi-Population Genetic Algorithm,” Proc. 6th European Congress on Intelligent Techniques and Soft Computing, EUFIT’98, Verlag Mainz, Aachen, Germany, 7-10. September 1998, Vol.1, pp. 450–456.

[89] G. Mester, A. Nemes, S. Pletl, J. Varga, “Optimization of Electric Motors by Multi-Population Genetic Algorithm,” Proc. SYMOPIS, Zbornik radova Jugoslovenskog simpozijuma o operacionim istrazivanjima, Herceg-Novi, Yugoslavia, 1998, pp. 8-9.

[90] G. Mester, A. Nemes, P. Odri, S. Pletl, “Design of the Optimal Fuzzy Modeling of Noise Pollution Using Multi-population Genetic Algorithm,” Proc. SCPE’97, Nis, Yugoslavia, 1997.

ABBREVIATIONS

3D – three dimensional in Euclidean space

VTOL – vertical take-off and landing

UAV – unmanned aerial vehicle

GA – genetic algorithm

TSK – Takagi-Sugeno-Kang

FLS – fuzzy logic system

MF – membership function

RM – robotic manipulator

GFS – genetic fuzzy system

SVD – singular value decomposition

LS – least squares

BLDC – brushless DC electric motor

DOF – degree of freedom

PID – proportional-integral-derivative

PD – proportional-derivative

GPS – global positioning system

MOGA – multi-objective GA

NSGA – non-dominated sorting GA

TAB – table look-up method

NNC – nearest neighbourhood clustering

ANFIS – adaptive neural fuzzy inference system

MSE – mean square error

PC – personal computer

zFLS – Zadeh-type fuzzy partition MF based FLS

cpFLS – continuous periodic FLS

**PGA – the classical Pareto-comparison based GAs

**NGA – the new quantitative comparison based GAs

**QGA –s the new quality comparison based GAs

**DGA – the classical sum difference comparison based GAs

**AGA – the result of (Q or D) comparison based GAs

MO*GA – the “block-type” multi-objective non-dominance ranking method based GAs

NS*GA – the “slice-type” multi-objective non-dominance ranking method based GAs

MM*GA – scalar ranking of non-dominance measurement based GAs

DO*GA – “block-type” multi-objective dominance ranking method based GAs

DM*GA – scalar ranking of dominance measurement based GAs

Sin – sine function

Cos – cosine function

Tan – tangent function

Asin – inverse function of sin

Atan – inverse function of tan

LIST OF TABLES

LIST OF FIGURES

APPENDIX I

Box-Jenkins Gas Furnace Benchmark Data

t[s]	input	output		t[s]	input	output		t[s]	input	output
1	-0.109	53.8		100	-0.314	50.4		199	-2.378	52.4
2	0	53.6		101	-0.288	51		200	-2.499	53.5
3	0.178	53.5		102	-0.153	51.8		201	-2.473	55.6
4	0.339	53.5		103	-0.109	52.4		202	-2.33	58
5	0.373	53.4		104	-0.187	53		203	-2.053	59.5
6	0.441	53.1		105	-0.255	53.4		204	-1.739	60
7	0.461	52.7		106	-0.229	53.6		205	-1.261	60.4
8	0.348	52.4		107	-0.007	53.7		206	-0.569	60.5
9	0.127	52.2		108	0.254	53.8		207	-0.137	60.2
10	-0.18	52		109	0.33	53.8		208	-0.024	59.7
11	-0.588	52		110	0.102	53.8		209	-0.05	59
12	-1.055	52.4		111	-0.423	53.3		210	-0.135	57.6
13	-1.421	53		112	-1.139	53		211	-0.276	56.4
14	-1.52	54		113	-2.275	52.9		212	-0.534	55.2
15	-1.302	54.9		114	-2.594	53.4		213	-0.871	54.5
16	-0.814	56		115	-2.716	54.6		214	-1.243	54.1
17	-0.475	56.8		116	-2.51	56.4		215	-1.439	54.1
18	-0.193	56.8		117	-1.79	58		216	-1.422	54.4
19	0.088	56.4		118	-1.346	59.4		217	-1.175	55.5
20	0.435	55.7		119	-1.081	60.2		218	-0.813	56.2
21	0.771	55		120	-0.91	60		219	-0.634	57
22	0.866	54.3		121	-0.876	59.4		220	-0.582	57.3
23	0.875	53.2		122	-0.885	58.4		221	-0.625	57.4
24	0.891	52.3		123	-0.8	57.6		222	-0.713	57
25	0.987	51.6		124	-0.544	56.9		223	-0.848	56.4
26	1.263	51.2		125	-0.416	56.4		224	-1.039	55.9
27	1.775	50.8		126	-0.271	56		225	-1.346	55.5
28	1.976	50.5		127	0	55.7		226	-1.628	55.3
29	1.934	50		128	0.403	55.3		227	-1.619	55.2
30	1.866	49.2		129	0.841	55		228	-1.149	55.4
31	1.832	48.4		130	1.285	54.4		229	-0.488	56
32	1.767	47.9		131	1.607	53.7		230	-0.16	56.5
33	1.608	47.6		132	1.746	52.8		231	-0.007	57.1

t[s]	input	output		t[s]	input	output		t[s]	input	output
34	1.265	47.5		133	1.683	51.6		232	-0.092	57.3
35	0.79	47.5		134	1.485	50.6		233	-0.62	56.8
36	0.36	47.6		135	0.993	49.4		234	-1.086	55.6
37	0.115	48.1		136	0.648	48.8		235	-1.525	55
38	0.088	49		137	0.577	48.5		236	-1.858	54.1
39	0.331	50		138	0.577	48.7		237	-2.029	54.3
40	0.645	51.1		139	0.632	49.2		238	-2.024	55.3
41	0.96	51.8		140	0.747	49.8		239	-1.961	56.4
42	1.409	51.9		141	0.9	50.4		240	-1.952	57.2
43	2.67	51.7		142	0.993	50.7		241	-1.794	57.8
44	2.834	51.2		143	0.968	50.9		242	-1.302	58.3
45	2.812	50		144	0.79	50.7		243	-1.03	58.6
46	2.483	48.3		145	0.399	50.5		244	-0.918	58.8
47	1.929	47		146	-0.161	50.4		245	-0.798	58.8
48	1.485	45.8		147	-0.553	50.2		246	-0.867	58.6
49	1.214	45.6		148	-0.603	50.4		247	-1.047	58
50	1.239	46		149	-0.424	51.2		248	-1.123	57.4
51	1.608	46.9		150	-0.194	52.3		249	-0.876	57
52	1.905	47.8		151	-0.049	53.2		250	-0.395	56.4
53	2.023	48.2		152	0.06	53.9		251	0.185	56.3
54	1.815	48.3		153	0.161	54.1		252	0.662	56.4
55	0.535	47.9		154	0.301	54		253	0.709	56.4
56	0.122	47.2		155	0.517	53.6		254	0.605	56
57	0.009	47.2		156	0.566	53.2		255	0.501	55.2
58	0.164	48.1		157	0.56	53		256	0.603	54
59	0.671	49.4		158	0.573	52.8		257	0.943	53
60	1.019	50.6		159	0.592	52.3		258	1.223	52
61	1.146	51.5		160	0.671	51.9		259	1.249	51.6
62	1.155	51.6		161	0.933	51.6		260	0.824	51.6
63	1.112	51.2		162	1.337	51.6		261	0.102	51.1
64	1.121	50.5		163	1.46	51.4		262	0.025	50.4
65	1.223	50.1		164	1.353	51.2		263	0.382	50
66	1.257	49.8		165	0.772	50.7		264	0.922	50
67	1.157	49.6		166	0.218	50		265	1.032	52
68	0.913	49.4		167	-0.237	49.4		266	0.866	54
69	0.62	49.3		168	-0.714	49.3		267	0.527	55.1

t[s]	input	output		t[s]	input	output		t[s]	input	output
70	0.255	49.2		169	-1.099	49.7		268	0.093	54.5
71	-0.28	49.3		170	-1.269	50.6		269	-0.458	52.8
72	-1.08	49.7		171	-1.175	51.8		270	-0.748	51.4
73	-1.551	50.3		172	-0.676	53		271	-0.947	50.8
74	-1.799	51.3		173	0.033	54		272	-1.029	51.2
75	-1.825	52.8		174	0.556	55.3		273	-0.928	52
76	-1.456	54.4		175	0.643	55.9		274	-0.645	52.8
77	-0.944	56		176	0.484	55.9		275	-0.424	53.8
78	-0.57	56.9		177	0.109	54.6		276	-0.276	54.5
79	-0.431	57.5		178	-0.31	53.5		277	-0.158	54.9
80	-0.577	57.3		179	-0.697	52.4		278	-0.033	54.9
81	-0.96	56.6		180	-1.047	52.1		279	0.102	54.8
82	-1.616	56		181	-1.218	52.3		280	0.251	54.4
83	-1.875	55.4		182	-1.183	53		281	0.28	53.7
84	-1.891	55.4		183	-0.873	53.8		282	0	53.3
85	-1.746	56.4		184	-0.336	54.6		283	-0.493	52.8
86	-1.474	57.2		185	0.063	55.4		284	-0.759	52.6
87	-1.201	58		186	0.084	55.9		285	-0.824	52.6
88	-0.927	58.4		187	0	55.9		286	-0.74	53
89	-0.524	58.4		188	0.001	55.2		287	-0.528	54.3
90	0.04	58.1		189	0.209	54.4		288	-0.204	56
91	0.788	57.7		190	0.556	53.7		289	0.034	57
92	0.943	57		191	0.782	53.6		290	0.204	58
93	0.93	56		192	0.858	53.6		291	0.253	58.6
94	1.006	54.7		193	0.918	53.2		292	0.195	58.5
95	1.137	53.2		194	0.862	52.5		293	0.131	58.3
96	1.198	52.1		195	0.416	52		294	0.017	57.8
97	1.054	51.6		196	-0.336	51.4		295	-0.182	57.3
98	0.595	51		197	-0.959	51		296	-0.262	57
99	-0.08	50.5		198	-1.813	50.9				

Table XX. Adapted Box-Jenkins benchmark system input-output training data pairs.

APPENDIX II

Effects of Displacement Jerk, Snap and Pop Function Discontinuities on Multi-rotor Flight Dynamics

Discontinuity in d^3r/dt^3 – jerk

Discontinuity in d^3r/dt^3 – jerk, the third time derivative of the displacement function occurs with trajectories that have non-smooth profile of the acceleration function. Common examples are: maximum velocity trajectory, bang-bang acceleration trajectory, or minimum acceleration and minimum energy trajectory, that are designed as described by Pontryagin in [58] – see chapter 5.1 Figure 30. Even minimum jerk trajectories of the same kind will result in discontinuities in jerk for the first and the last instance of time – see Figure 60:

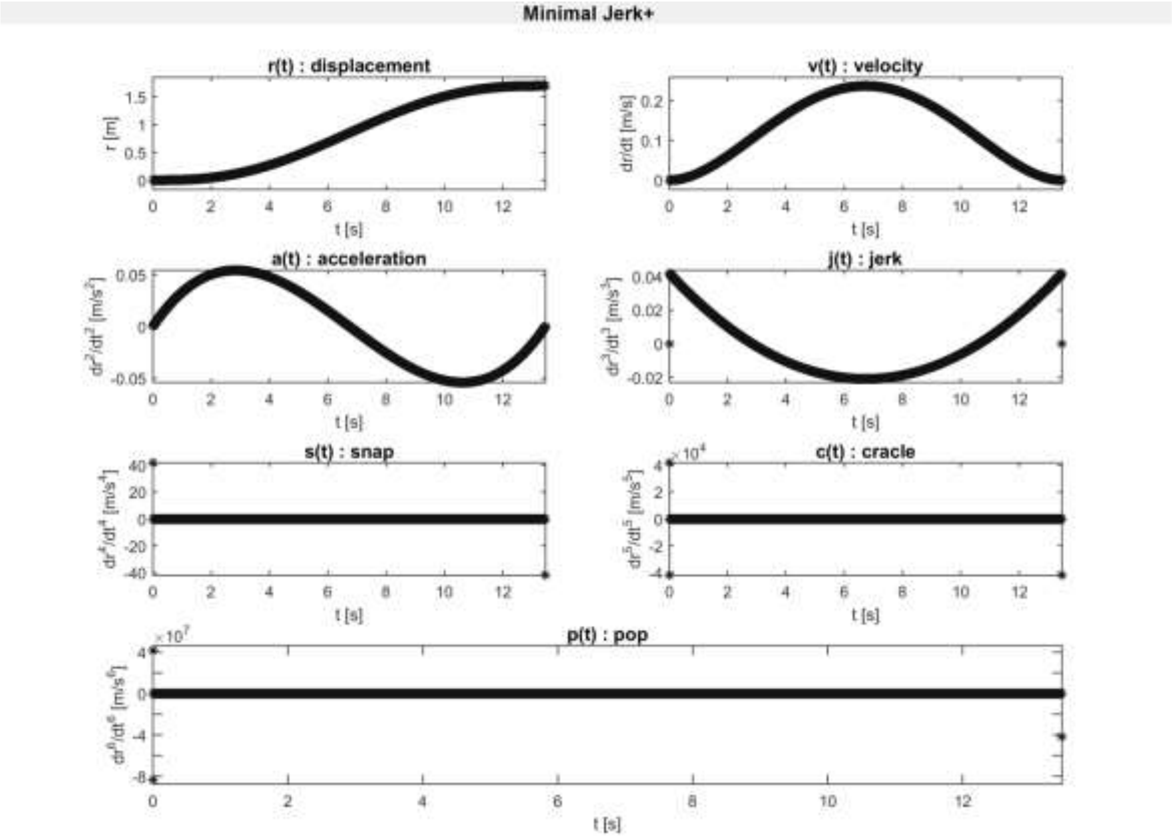


Figure 60.
Minimal jerk trajectory time derivatives – discontinuity in jerk.

The simplest way to visualise the effects of jerk discontinuity on the multi-rotor dynamics, is to take a simple step function for jerk – Figure 61 presents all the important derivatives of such a displacement.

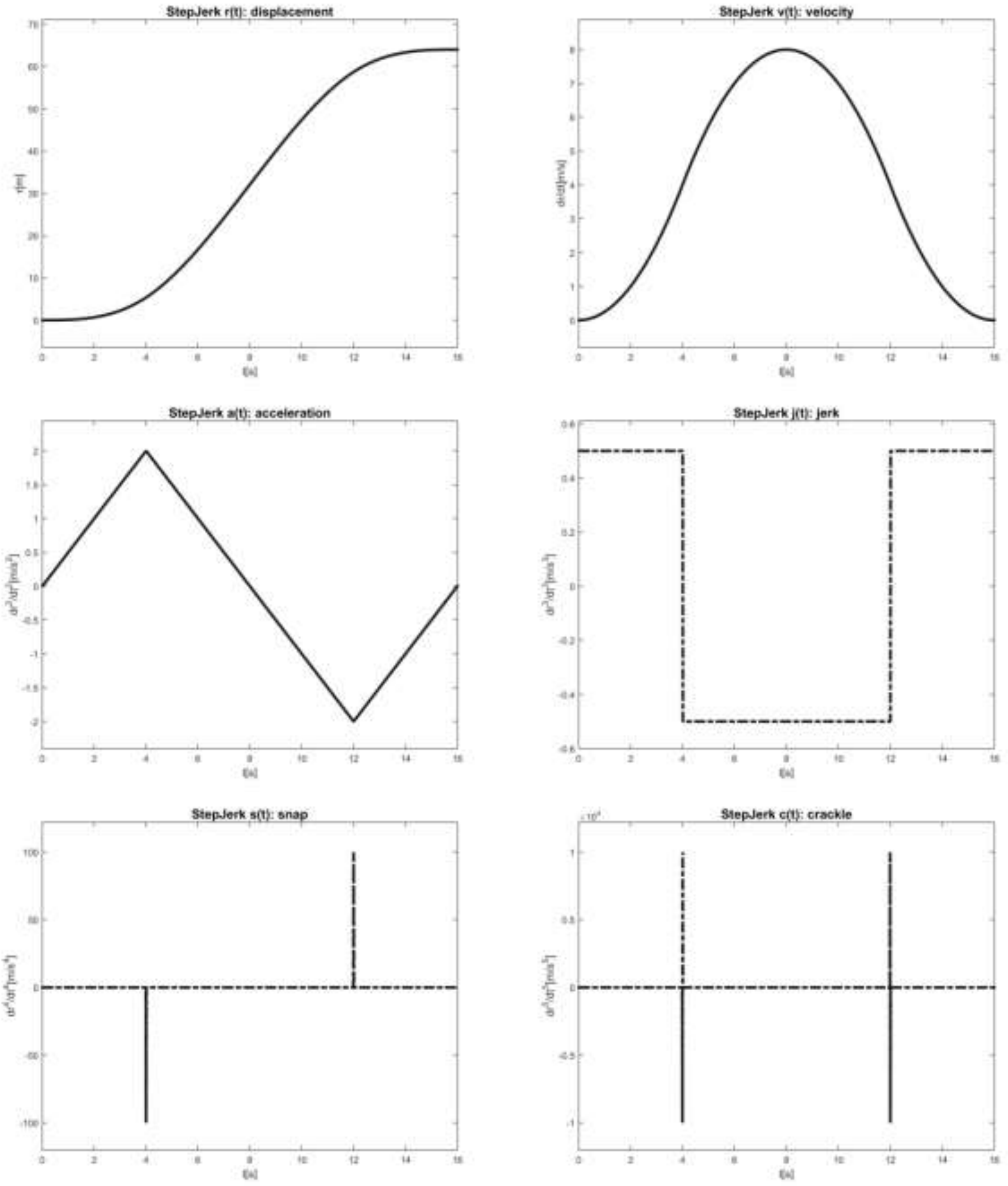


Figure 61.
Step jerk trajectory time derivatives – discontinuity in jerk.

A step jerk function results in a triangular acceleration profile (quite often used in literature) as it defines a seemingly nice continuous velocity profile. Notice that I have set up this trajectory to be comparable to my feasible optimal harmonic trajectory described in chapter 5.2 and presented in Figures 32-39. Sampling time remains *dt = 0.01[s]*, also *maximum_acceleration = 2[m/s^2]*, *maximum_velocity = 8[m/s]*, *displacement = 64[m]*, within *duration = 16[sec]*. What is different because of the step jerk profile is that *maximum_jerk = 0.5[m/s^3]*(instead of 1), *maximum_snap = 100[m/s^4]* (instead of 1) and also maximum *crackle* increases to 10 000 *[m/s^5]* from the previous value of 2*[m/s^5]*! The most important change is that the snap and crackle functions become Dirac-delta impulses.

The discontinuity in jerk results in a noticeable discontinuity in the first derivative of the roll (nue_1) and pitch (nue_2) Euler angles, and a significant spike in their second time derivative – see Figure 62.

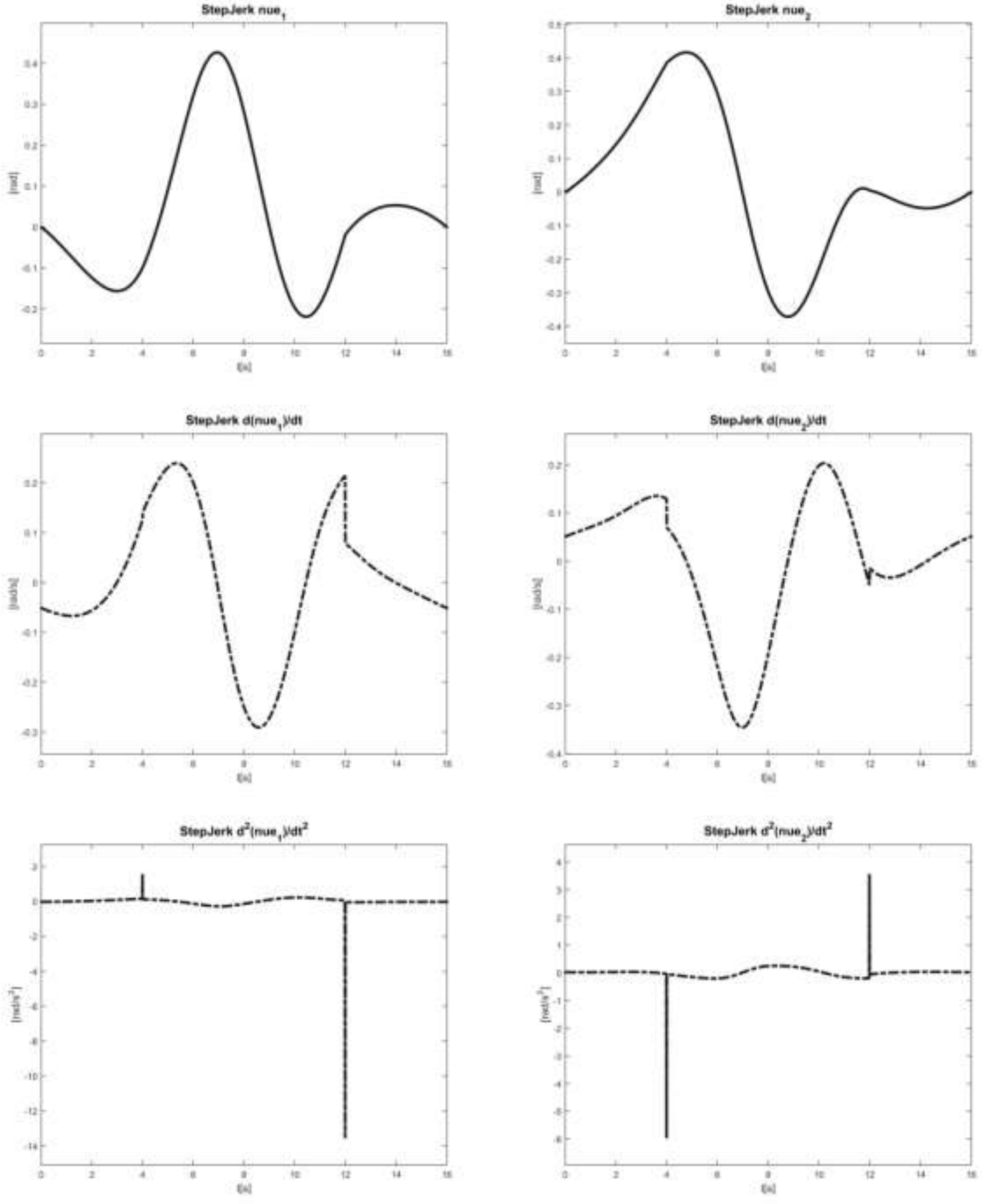


Figure 62.
Step jerk trajectory – system Euler angles and their time derivatives.

Now let’s take a look at the required torques to track such a trajectory; it is full of large spikes – Figure 63.

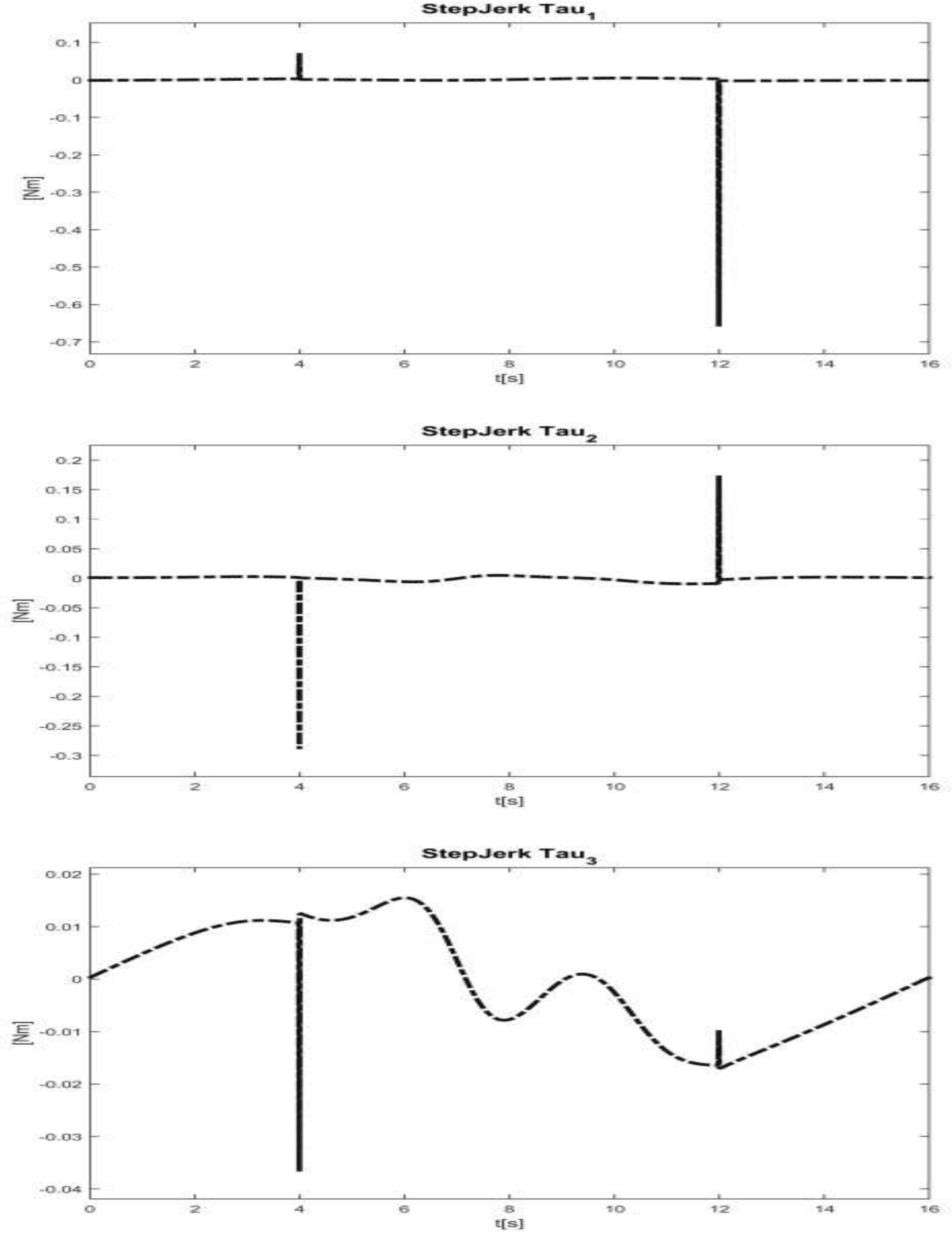


Figure 63.
Step jerk trajectory – system torques.

The corresponding electro motor shaft rotation velocity is presented in Figure 64. This shaft displacement is simply not feasible – we cannot generate instantaneous rotation velocity changes in a real physical system as electric motors.

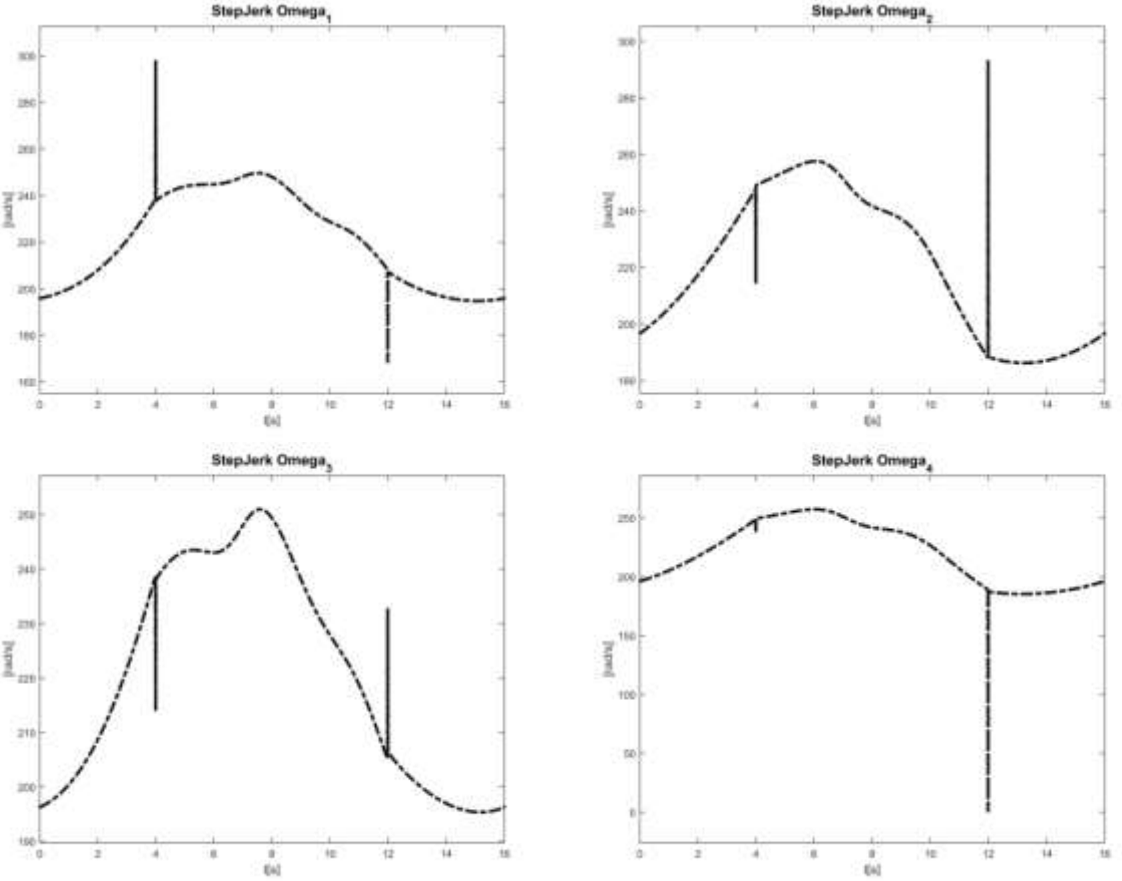


Figure 64.
Step jerk trajectory – required motor shaft rotational velocity.

Discontinuity in d^4r/dt^4 – snap

Discontinuity in d^4r/dt^4 – snap, the fourth derivative of displacement occurs even for minimum snap trajectories as described by Mellinger in [37]. Observe the first and the last instance of time in Figure 65.

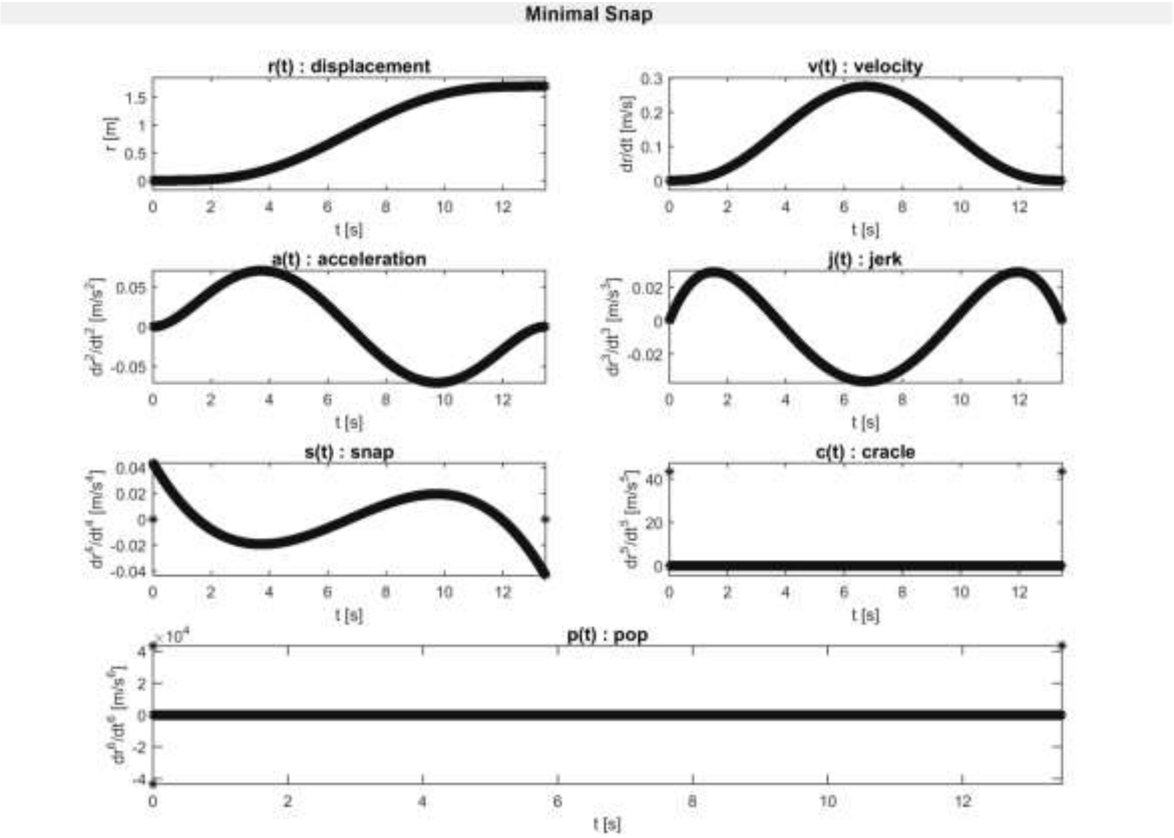


Figure 65.
Minimal snap trajectory time derivatives – discontinuity in snap.

The simplest way to visualise the effects of snap discontinuity on the multi-rotor dynamics, is to take a simple step function for snap – Figure 66 presents all the important derivatives of such a displacement.

A step snap function results in a triangular jerk profile. Notice that I have set up this trajectory to be comparable to my feasible optimal harmonic trajectory described in chapter 5.2 and presented in Figures 32-39. Sampling time remains *dt = 0.01[s]*, also *maximum_jerk = 1[m/s^3]*, *maximum_acceleration = 2[m/s^2]*, *maximum_velocity = 8[m/s]*, *displacement = 64[m]*, within *duration = 16[sec]*. What is different because of the step snap profile is that *maximum_snap = 0.5[m/s^4]* (instead of 1) and also maximum *crackle* increases to 100 *[m/s^5]* from the previous value of 2*[m/s^5]*! The most important change is that the crackle function becomes a combination of Dirac-delta impulses – see Figure 66.

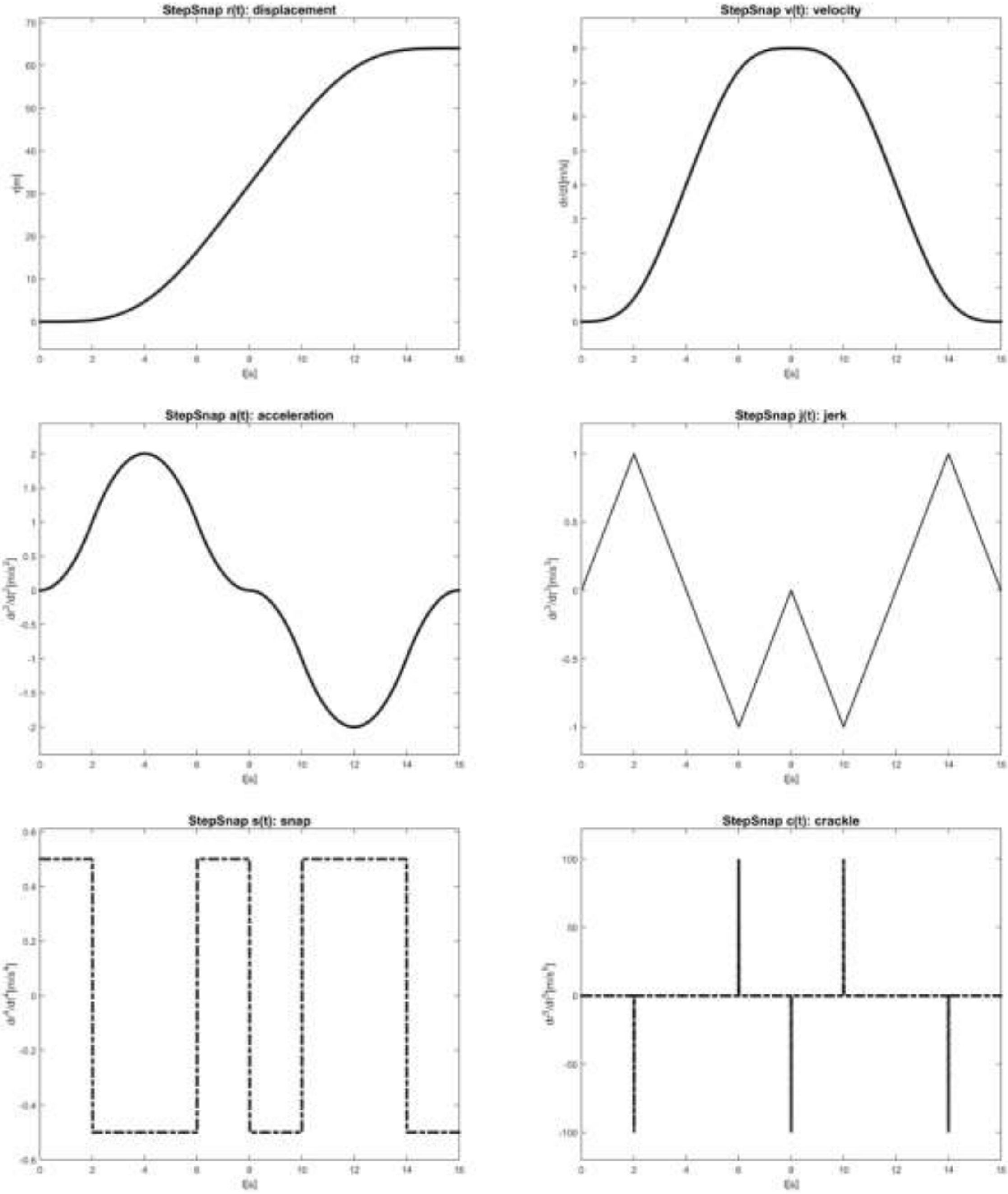


Figure 66.
Step snap trajectory time derivatives – discontinuity in snap.

The discontinuity in snap results in discontinuity in the second derivative of the roll (nue$_1$) and pitch (nue$_2$) Euler angles – see Figure 67.

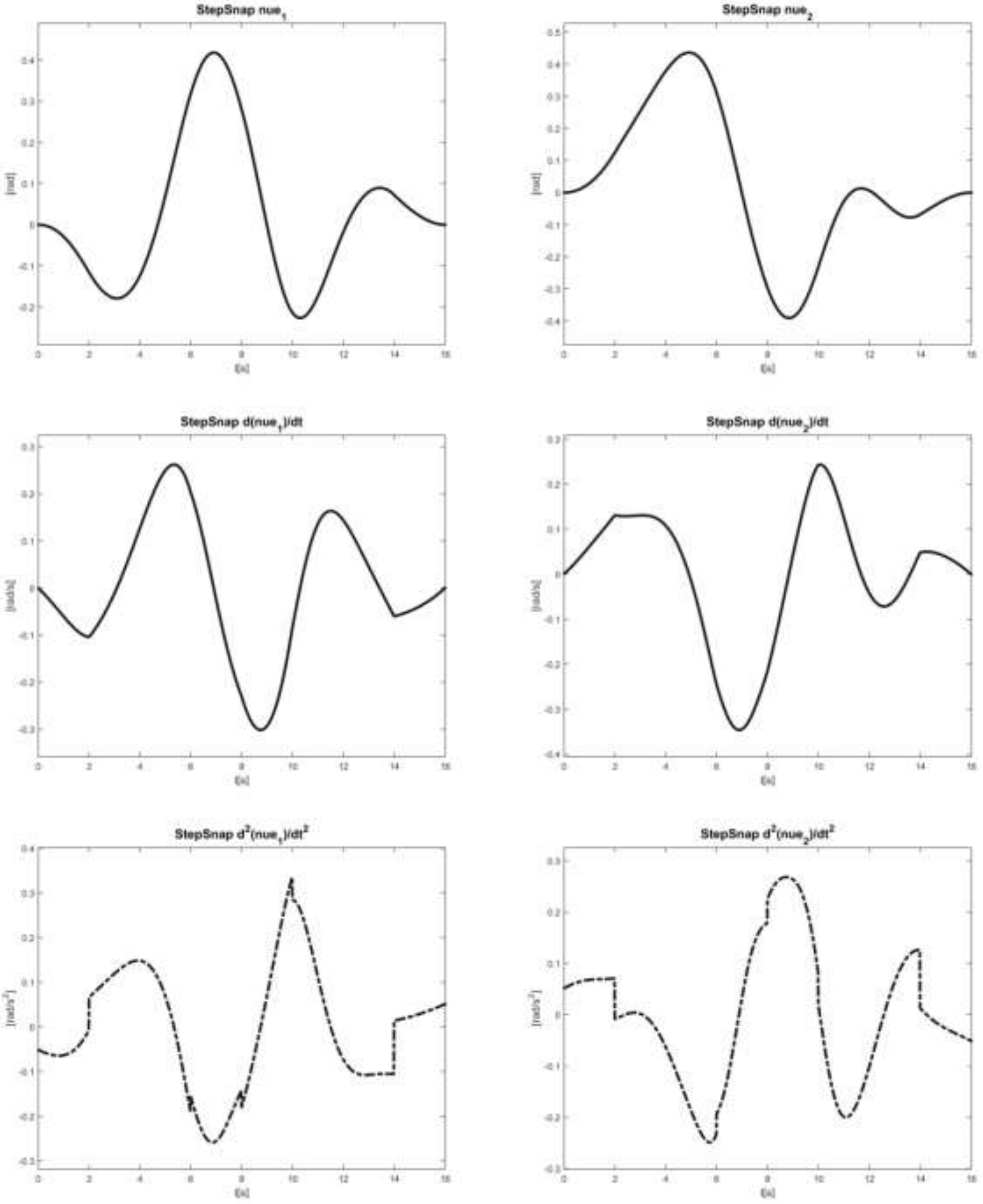


Figure 67.
Step snap trajectory – system Euler angles and their time derivatives.

Now let’s take a look at the required torques to track such a trajectory, it is discontinuous – Figure 68, and first time derivative of the torque is full of spikes – Figure 69.

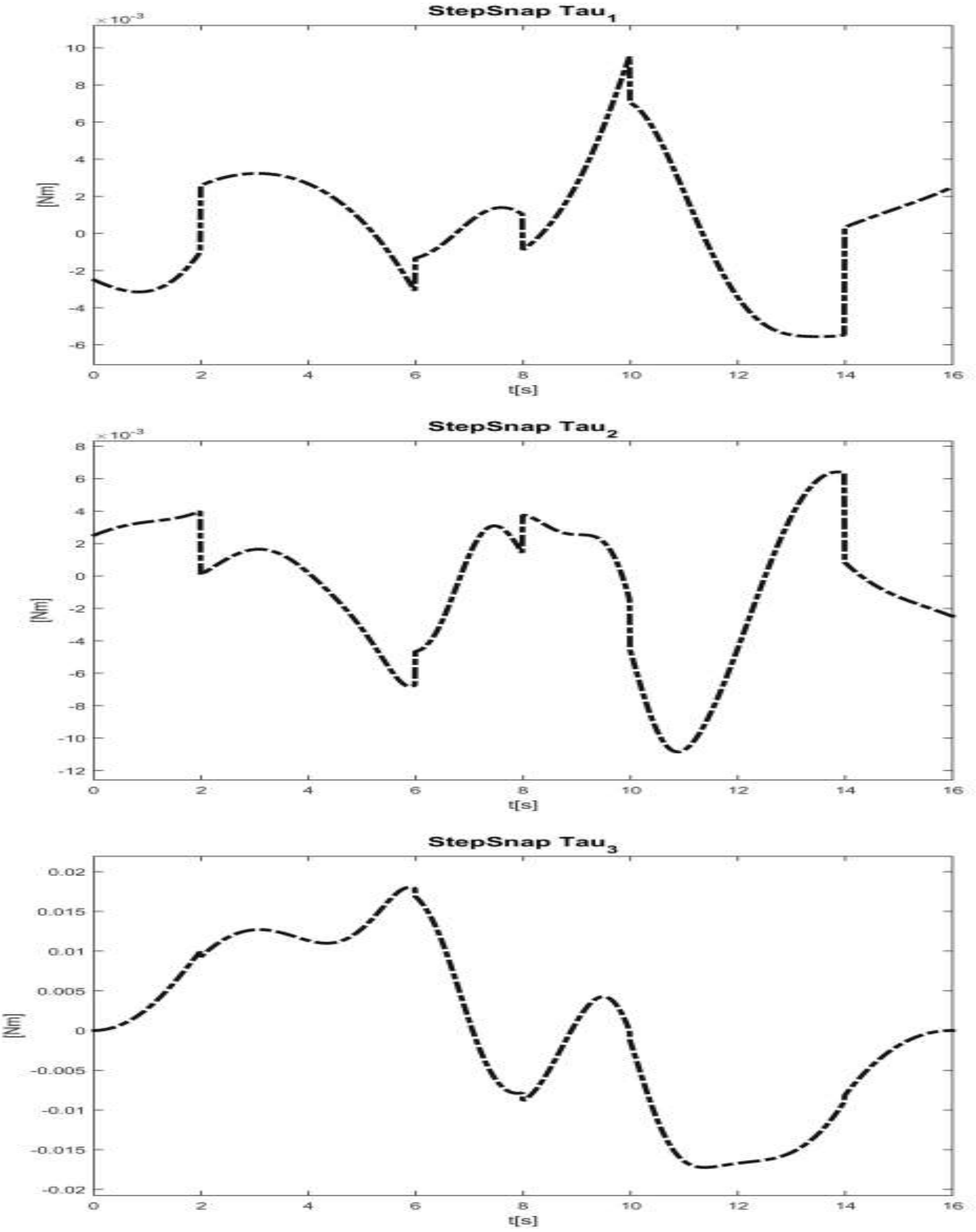


Figure 68.
Step snap trajectory – system torques.

The first time derivative of required system torques is full of discontinuities, spikes, Dirac impulses – Figure 69.

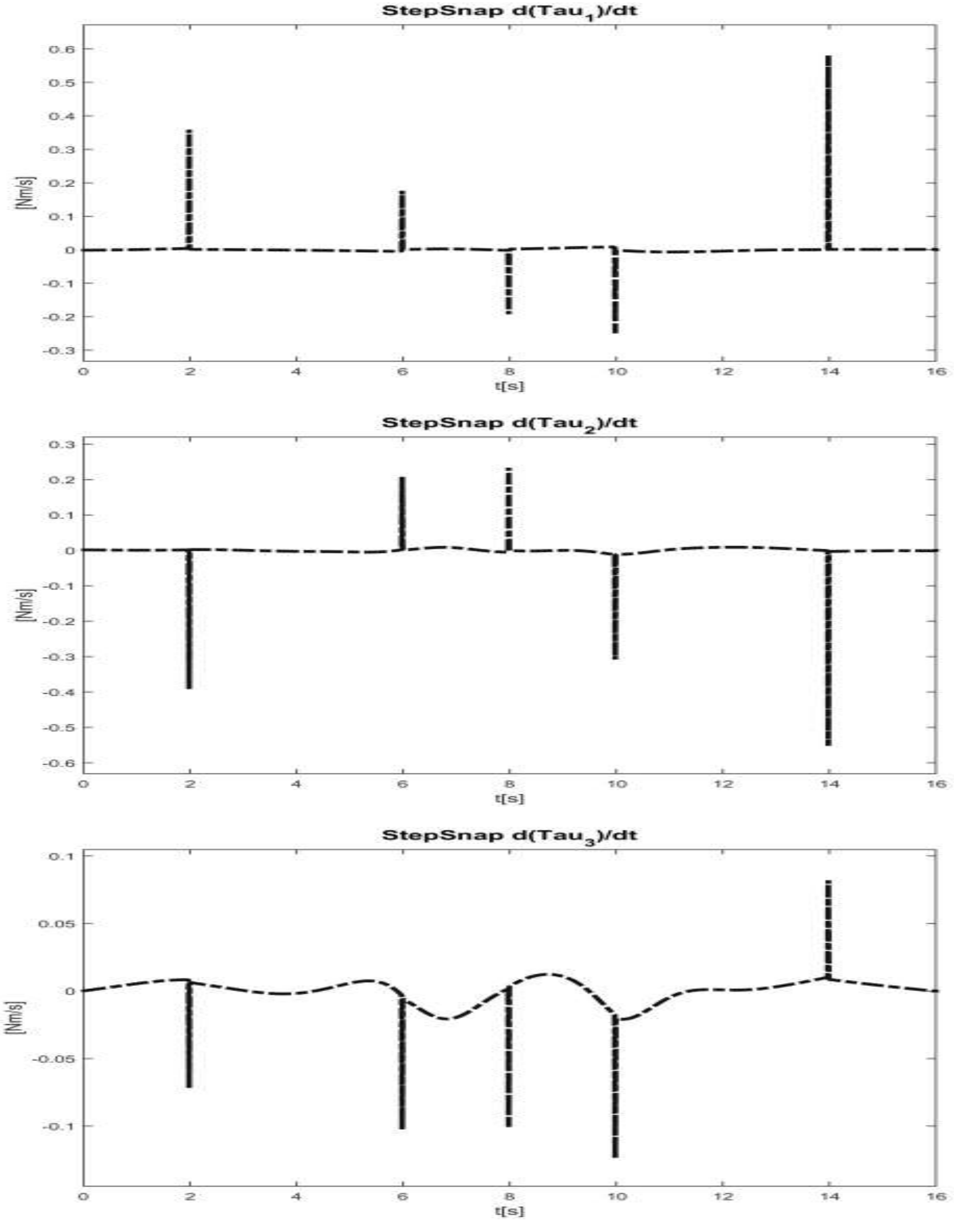


Figure 69.
Step snap trajectory – system torque first time derivatives.

The corresponding electro motor shaft rotation velocity is presented in Figure 70. Even these very small discontinuous jumps in the rotation velocity are simply not feasible – we cannot generate instantaneous rotation velocity changes in a real physical system. The problem is more obvious when we take a look at the rotational acceleration – Figure 71, especially in the light of equation (56).

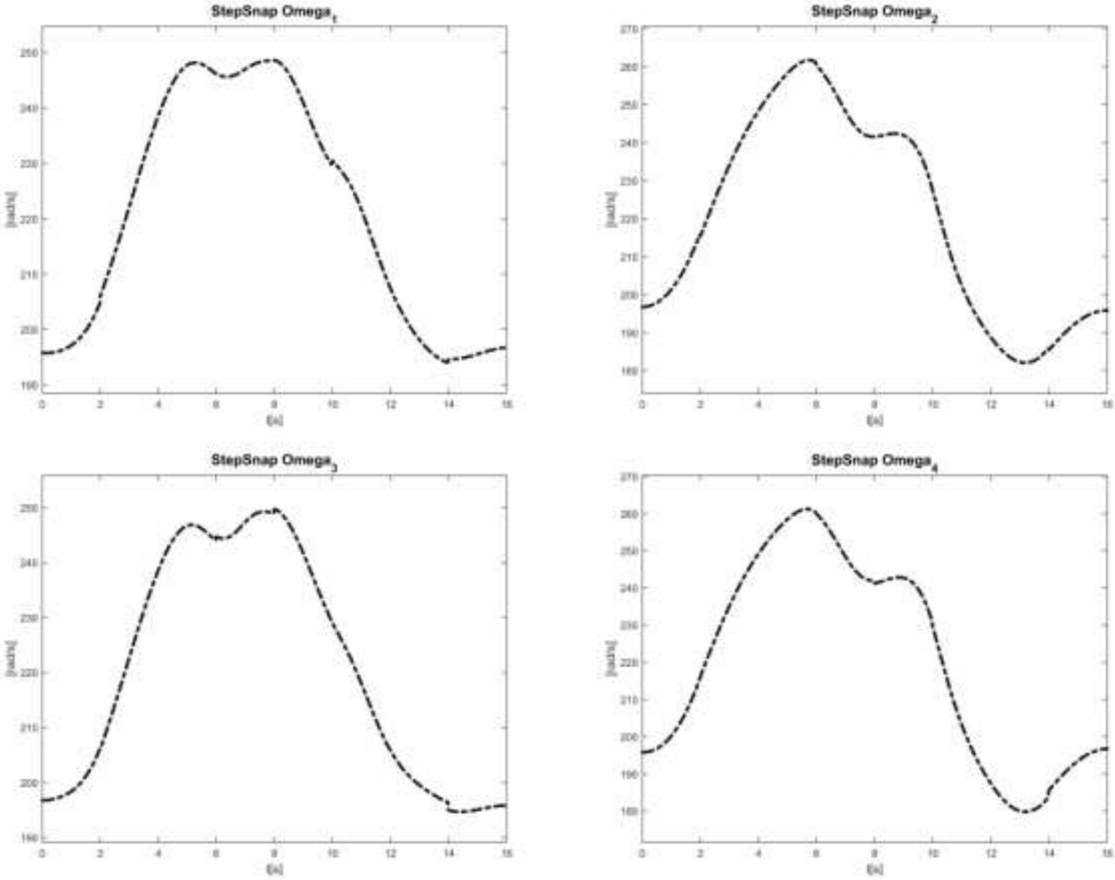


Figure 70.
Step snap trajectory – required motor shaft rotational velocity.

The corresponding electro motor shaft rotation accelerations are presented in Figure 71.

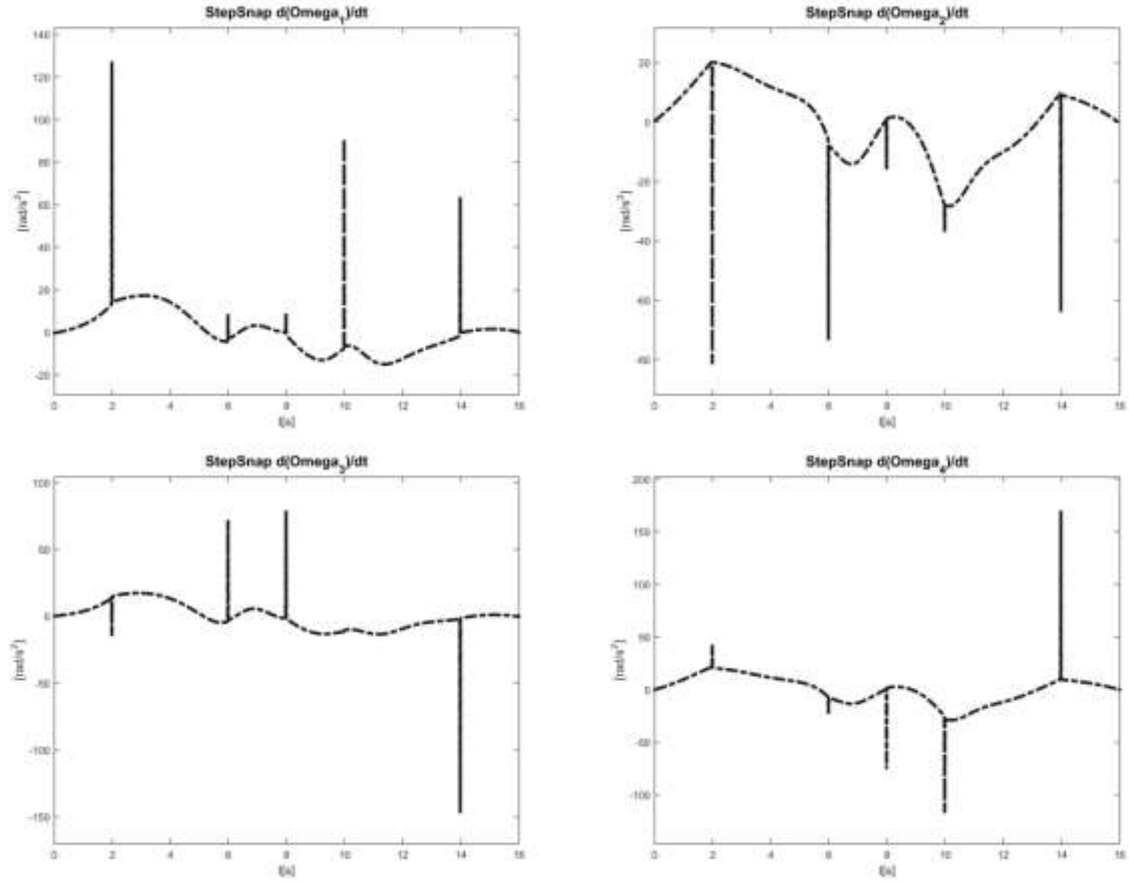


Figure 71.
Step snap trajectory – required motor shaft rotational accelerations.

These spikes, instant jumps back and forth in the rotor shaft acceleration of a real physical system are simply not feasible – we cannot generate instantaneous rotation acceleration changes in a real physical electric motor systems; since by equation (56) $K_\tau \boldsymbol{i_e}(t) - \boldsymbol{\tau_L}(t) = J_M \frac{\boldsymbol{d\omega}}{\boldsymbol{dt}}(t) + \gamma_M \boldsymbol{\omega}(t)$ discontinuity in $\frac{\boldsymbol{d\omega}}{\boldsymbol{dt}}$ would mean discontinuity in at least one of electric motor torque, rotation velocity and/or electrical current, which are obviously real physical properties of the system, and those just cannot be "instantaneously teleported" from one value to an arbitrary other value (like $\pm$100 rad/s^2 changes within 10 milliseconds).

Discontinuity in d^6r/dt^6 – pop

Discontinuity in d^6r/dt^6 – pop occurs with trajectories that have non-smooth crackle; a step function displacement pop will result in a triangular crackle function Figure 72. Notice that I have set up this trajectory to be comparable to my feasible optimal harmonic trajectory described in chapter 5.2 and presented in Figures 32-39. Sampling time remains *dt = 0.01[s]*, also *maximum_snap = 1[m/s⁴]*, *maximum_jerk = 1[m/s³]*, *maximum_acceleration = 2[m/s²]*, *maximum_velocity = 8[m/s]*, *displacement = 64[m]*, within *duration = 16[sec]*. Even the maximum *crackle* remains 2*[m/s⁵]* so all trajectory components are equally bounded as my feasible optimal harmonic trajectory described in chapter 5.2.

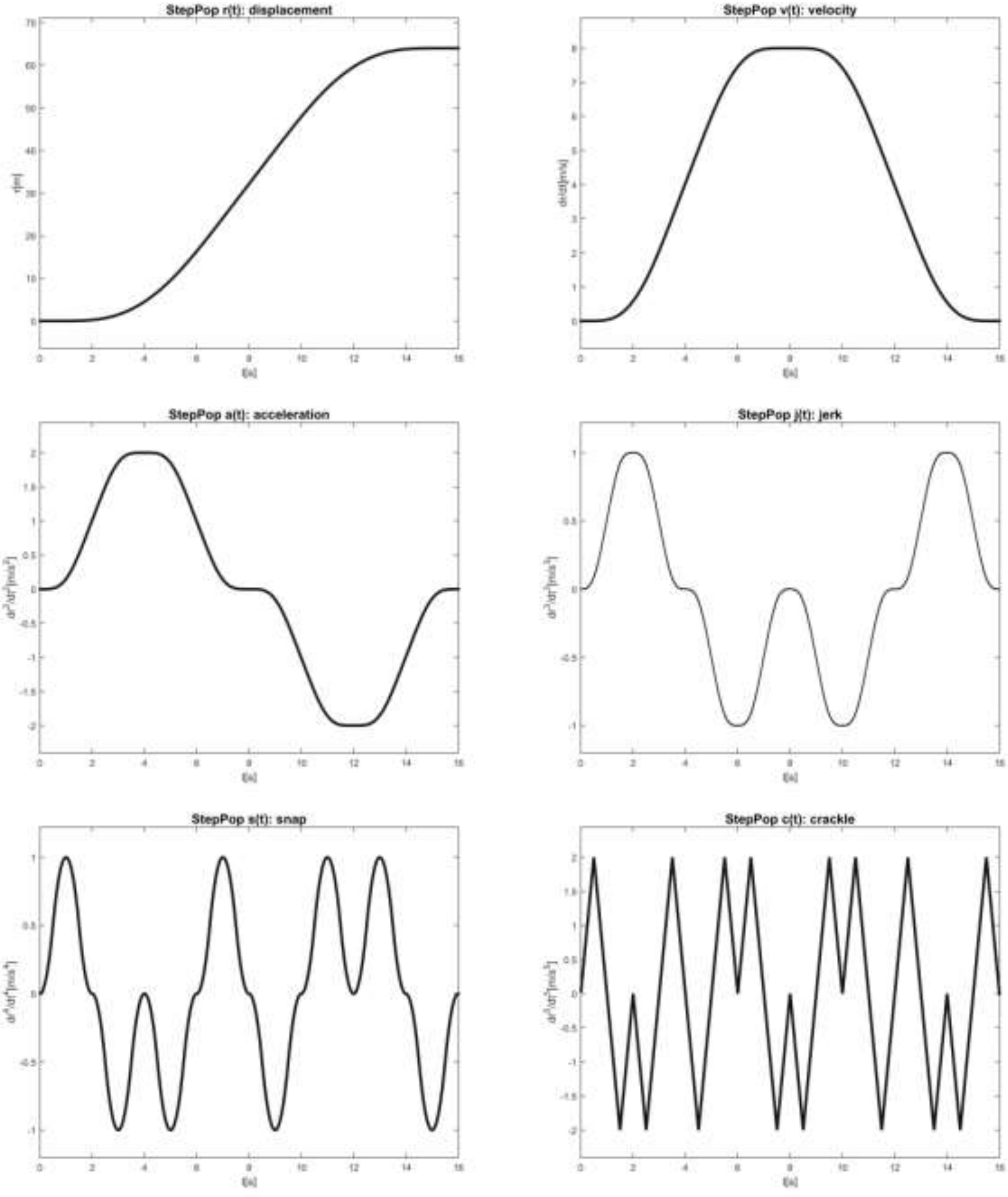


Figure 72.
Step pop trajectory – a trajectory with discontinuity in the displacement pop.

The discontinuity in pop results in no obvious issue with the required roll (nue$_1$) and pitch (nue$_2$) Euler angles or their first and second time derivatives, they are all smooth – see Figure 73.

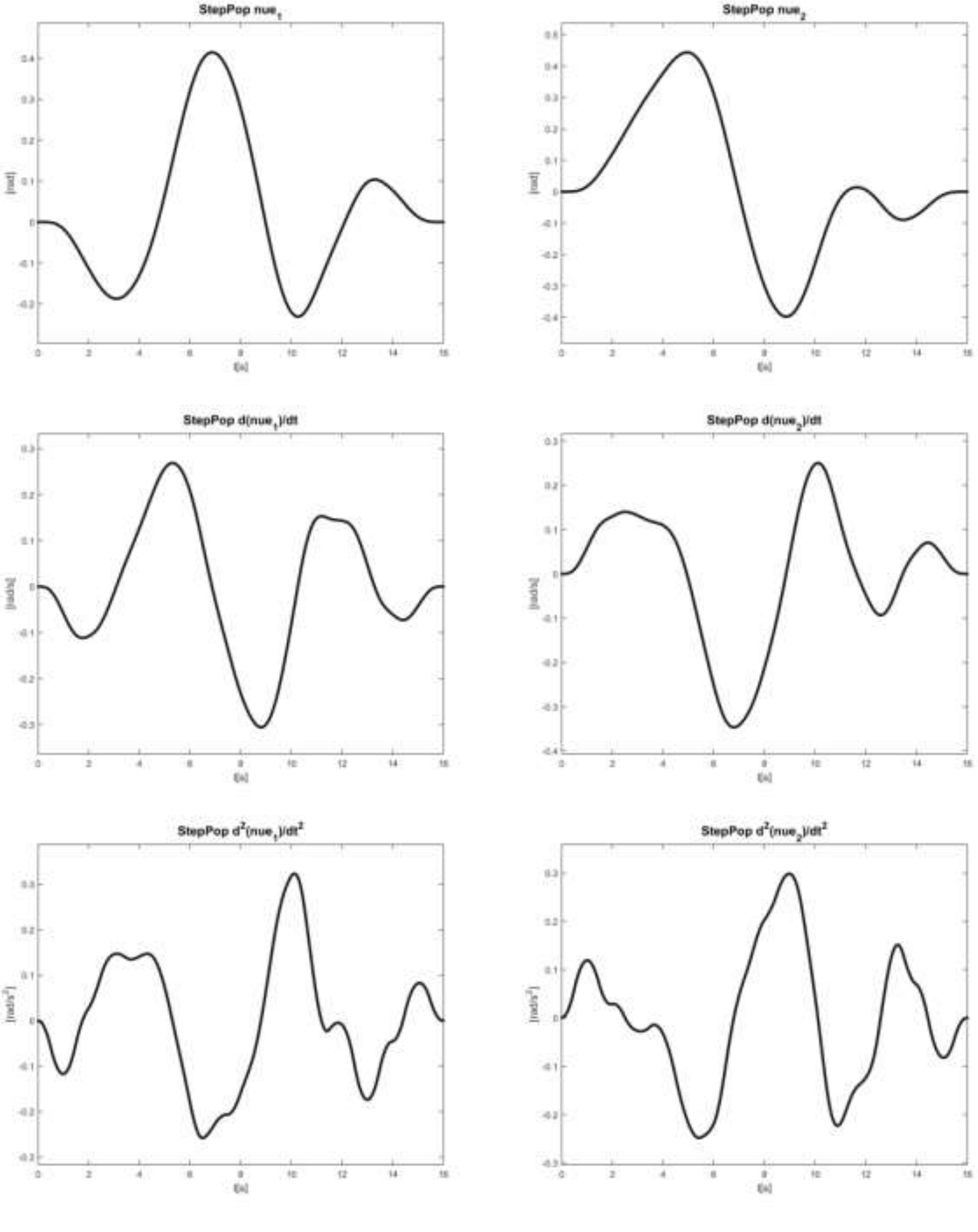
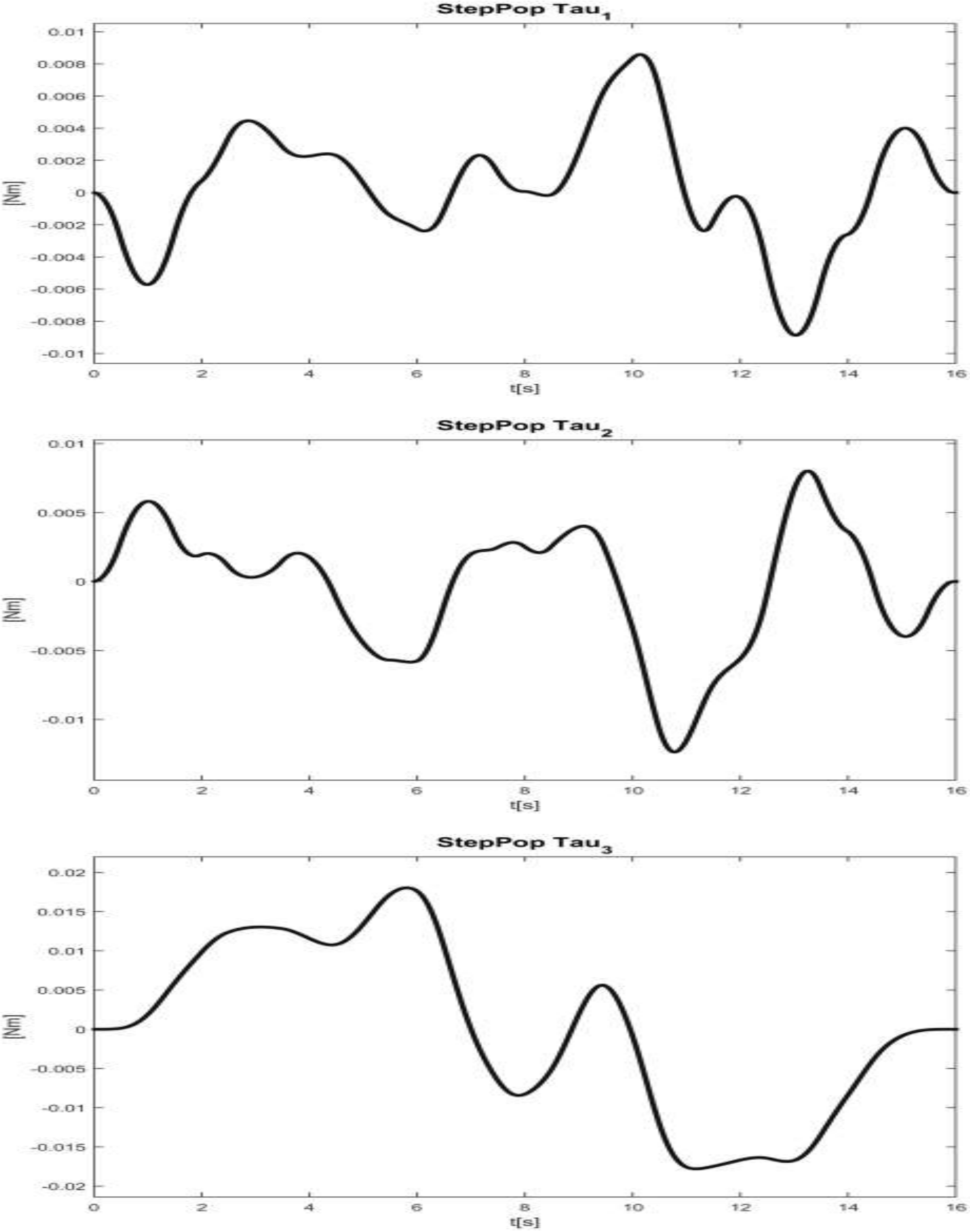


Figure 73.
Step pop trajectory – system Euler angles and their time derivatives.

The discontinuity in pop results in no obvious issue with the required system torques, they also are all smooth – see Figure 74.


StepPop Tau$_1$

StepPop Tau$_2$

StepPop Tau$_3$

Figure 74.
Step pop trajectory – system torques.

The discontinuity in pop results in no obvious issue with the first time derivative of required system torques either, they are all smooth – see Figure 75.

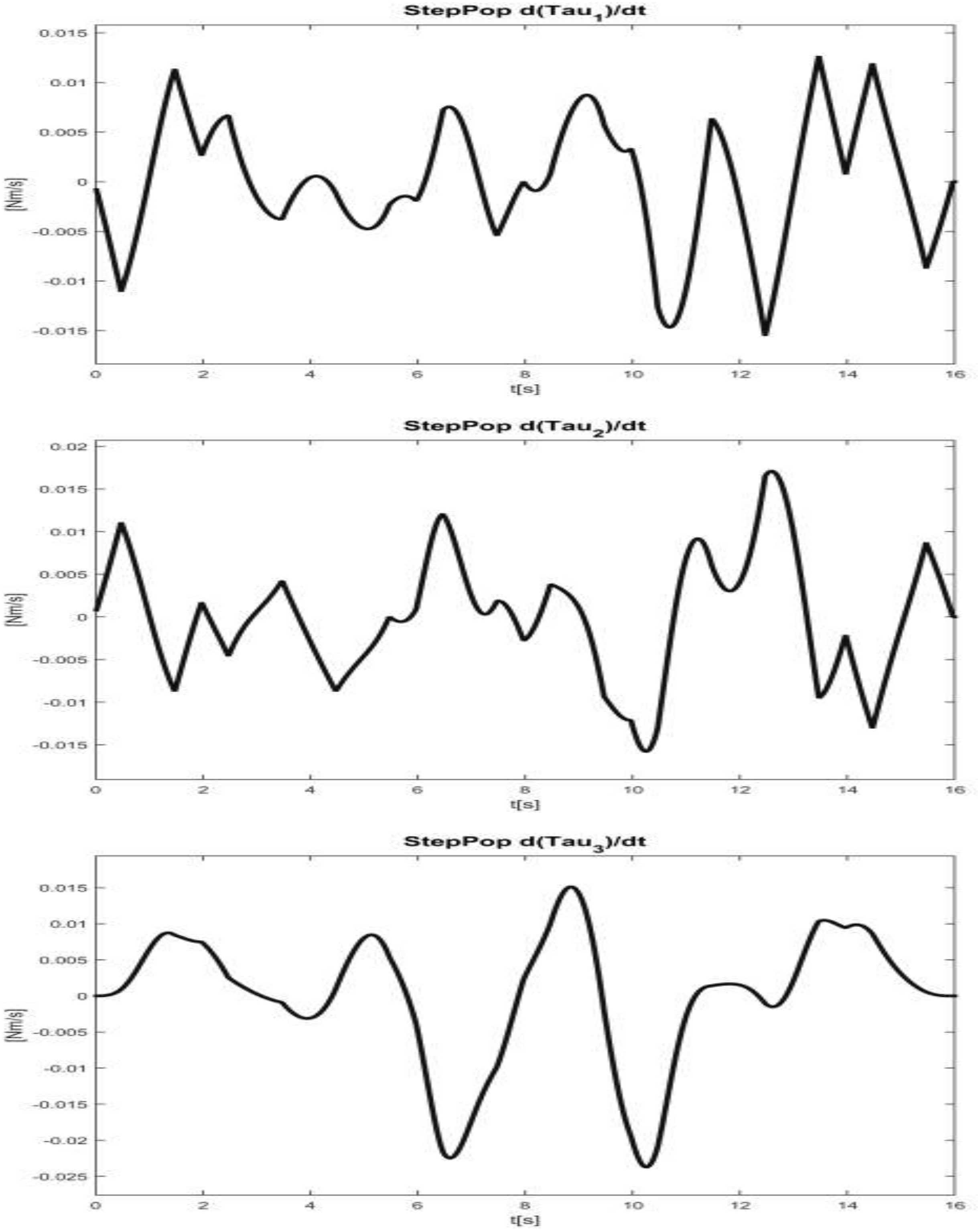


Figure 75.
Step pop trajectory – first time derivative of system torques.

Now observe how the discontinuity in pop results in a discontinuity in the second time derivative of required system torques – see Figure 76.

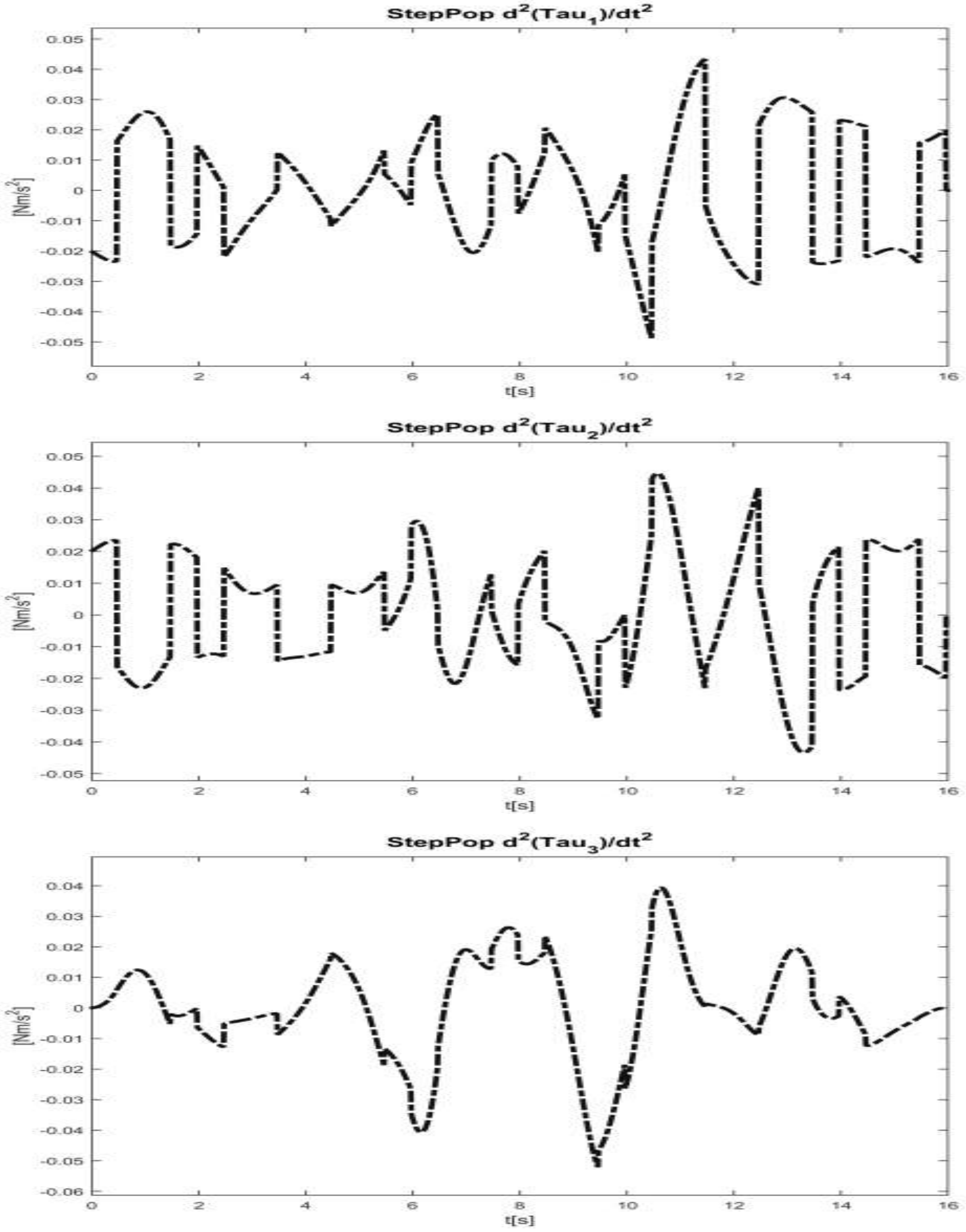


Figure 76.
Step pop trajectory – second time derivative of system torques.

Notice how a discontinuity in the second time derivative of required system torques will induce problems with the required motor shaft rotational profile.

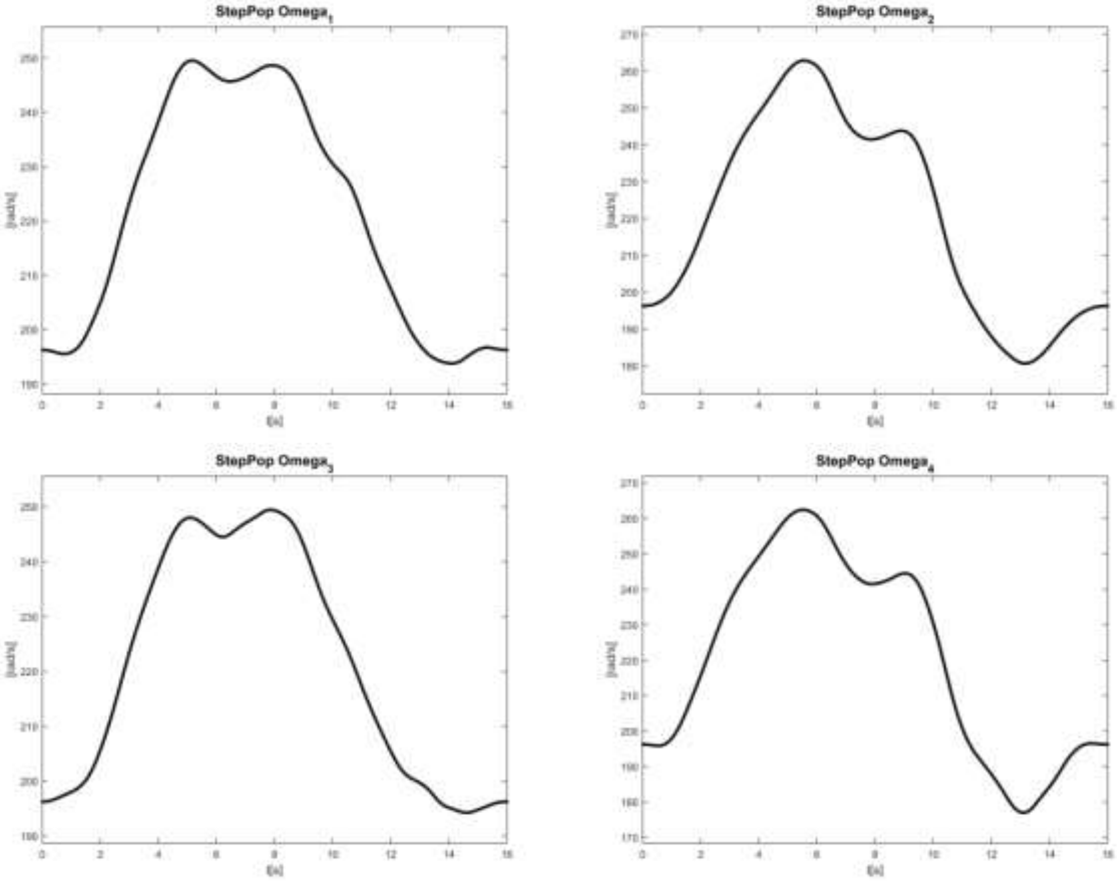


Figure 77.
Step pop trajectory – required motor shaft rotational velocity, $\boldsymbol{\omega}(t)$.

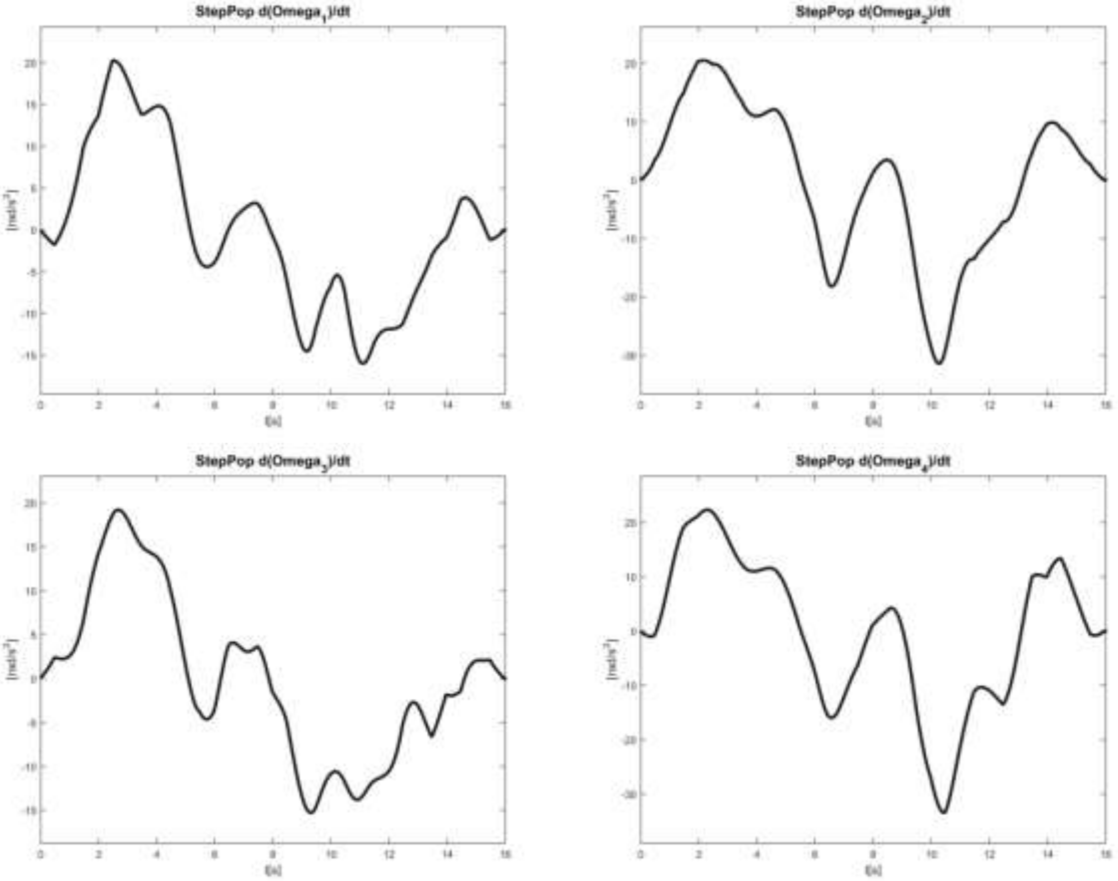


Figure 78.

Step pop trajectory – required motor shaft rotational acceleration, $\frac{d\boldsymbol{\omega}}{dt}(t)$.

Now let's observe how the discontinuity in pop results in a discontinuity in the second time derivative of required motor shaft rotational velocity – see Figure 79. This is understandable if we notice that based on equation (57) the rotational torque of the multi-rotor is mostly influenced by the square of the rotational velocity, thus discontinuity in the second derivative of the multi-rotor body torque (Figure 76) will result also in a discontinuity in the second derivative of the required rotational velocity the rotor shaft, given that all other components are continuous.

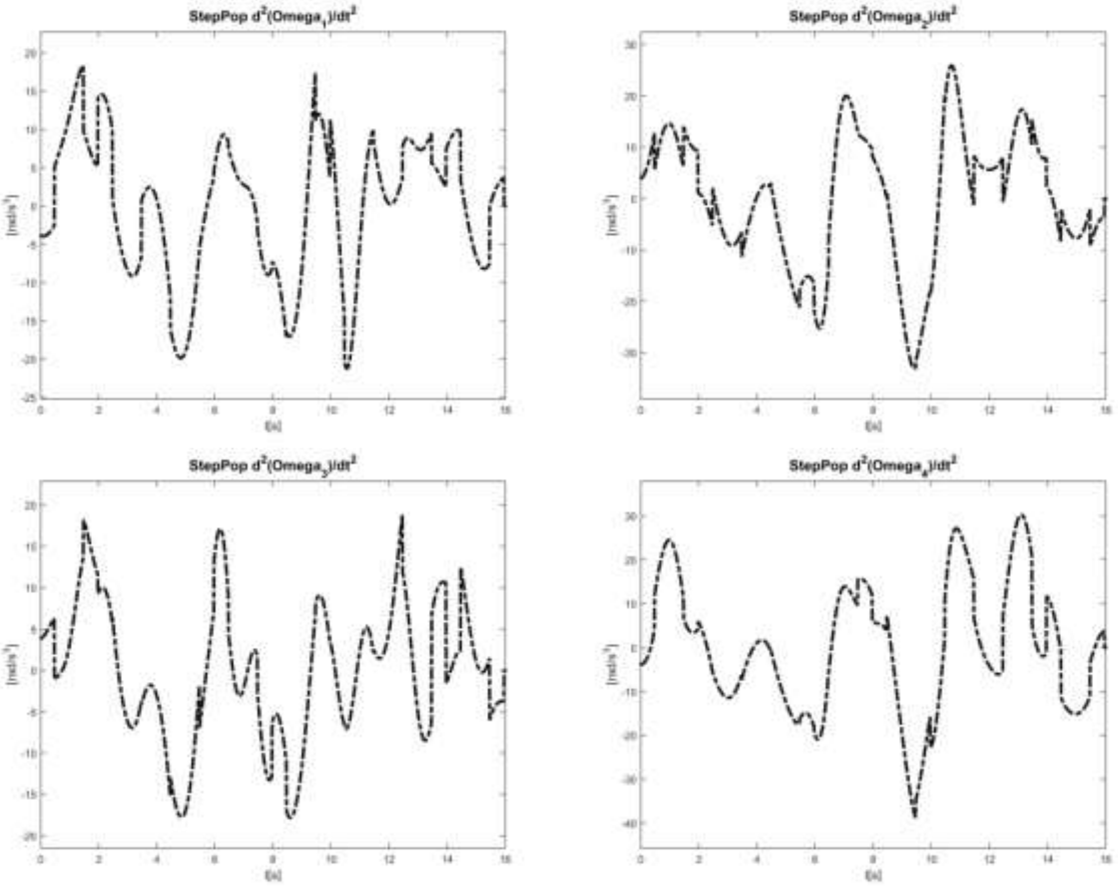


Figure 79.
Step pop trajectory – required motor shaft rotational jerk, $\boldsymbol{\frac{d^2\omega}{dt^2}}(t)$.

Observe the electric motor equation (58) and then (55) described in chapter 5.1.3, formulas repeated here for a reminder:

$$\boldsymbol{\frac{di_e}{dt}}(t) = \frac{1}{K_\tau}\Big((J_M + J_R)\boldsymbol{\frac{d^2\omega}{dt^2}}(t) + (\gamma_M + 2K_d\boldsymbol{\omega})\boldsymbol{\frac{d\omega}{dt}}(t)\Big), \tag{58}$$

$$\boldsymbol{v_e(t) = L_e \frac{di_e}{dt}(t) + R_e i_e(t) + K_b \omega(t),} \tag{55}$$

It is obvious that discontinuity in $\boldsymbol{\frac{d^2\omega}{dt^2}}(t)$ would mean discontinuity in $\boldsymbol{\frac{di_e}{dt}}(t)$ and then in $\boldsymbol{v_e(t)}$ or in $\boldsymbol{i_e(t)}$, which is realistically not possible for real physical systems! We see that even if it seems really farfetched to discuss discontinuities in the sixth time derivative of a multi-rotor displacement or the second time derivative of a motor rotation profile, it is important if we would like to deal with real, physically feasible trajectories of electro motor actuated multi-rotors. Implementation and properties, optimality of feasible trajectories are discussed in chapter 5.2.