Limiting Static and Dynamic Characteristics of an Induction Motor under Frequency Vector Control

Istvan Vajda[1], Yury N. Dementyev[2], Kirill N. Negodin[2], Nikolay V. Kojain[2], Leonid S. Udut[2], Irina. A. Chesnokova[2]

[1]Óbuda University, Kandó Kálmán Polytechnic, Bécsi út 96/b, 1034 Budapest, Hungary, e-mail vajda@uni-obuda.hu

[2]Tomsk Polytechnic University, Institute of Power Engineering, av. Lenina 30, 634050 Tomsk, Russian Federation (e-mail: dementev@tpu.ru, knn1@tpu.ru, kojain@tpu.ru, udut@tpu.ru, taksimo@tpu.ru)

Abstract: Static and dynamic characteristics of an induction motor (IM) under frequency vector control are reviewed. Limiting static characteristics enabling to determine the limits of an automatic electric drive, as well as regions of short-term and admissible continuous performance of an induction motor under frequency vector control are presented. Recommendations on the choice of maximum phase voltage of an inverter, DC link voltage of a frequency converter and supply voltage of an induction motor as well as possible ways to reach maximum angular velocity of an induction motor under frequency vector control are suggested.

Keywords: induction motor; frequency control; vector control; three-phase inverter; limiting characteristics

1 Introduction

A squirrel cage AC induction motor drive is widely applied in adjustable electric drive systems, which are currently used in industry and operate mainly in continuous static modes with a constant or slowly varying load moment. This electric drive consumes more than half of all power generated [1].

The widespread use of a squirrel cage induction motor in the systems of an adjustable electric drive, which are in high demand in industries, can be attributed to its high reliability due to the absence of a brush-collector unit and permanent magnets, a simple design, a small size and a rotor inertia moment, absence of switching constraints on speed and current, etc. [2-4]. The most common law for developing automatic control systems (ACS) of a frequency-controlled induction

motor drive, which implement the assigned static values, was, at an early stage, a simple proportional law of voltage amplitude control of an induction motor stator in its frequency function. However, in some works [3] it is proved that application of this control law makes it impossible to achieve both acceptable mechanical and energy characteristics of an electric drive under a wide range of rotation changes per minute and load changes due to the influence of active resistance and leakage inductance of the stator of an induction motor.

In that regard, a more promising principle of frequency-vector control of an induction motor drive [3-5] was developed. It enables to consider an induction motor as a two-channel object (an analogue of a separately excited DC motor) oriented along the vector of the rotor flux linkage. A vector-frequency control of a squirrel-cage induction motor allows for providing an independent control of the rotor flux linkage vector and electromagnetic moment. Due to that a two-region rotations per minute can be controlled in the vector control system similar to a dc drive [6].

Currently, of particular interest for the research are limiting static characteristics of an induction motor. Corresponding either to the rated motor voltage or maximum output voltage of an inverter of a frequency converter under the assigned voltage of the supply network for various control systems of a three-phase inverter [7]. Thus, enabling to estimate feasibility of reaching a desired speed depending on the load moment in a vector VFD.

The purpose of the article is to analyse the limiting static design characteristics of the motor $\omega(T_{EM})$ and $\omega(T_{1ph})$ in the "frequency converter - induction motor" system open at *q*-axis coordinate system at the assigned value of the rotor flux linkage and in the closed system of the induction motor drive under frequency-vector control with controlled flux.

2 Vector Method of Frequency Control of an Induction Motor

The vector systems of induction motors frequency control are based on a structural scheme of a two-phase motor in rotating coordinates *d*, *q* [8-12]. In the closed loop vector control system, the voltage component U_{1d} sustains the rotor flux linkage Ψ_{2d}=const constant and the voltage component U_{1q} ensures equality of the motor electromagnetic moment to the static moment on the shaft $T_{EM}=T_{load}+\Delta T_{lmotor}$ in the steady-state operation mode.

The automatic control system of an induction motor drive with a frequency-vector control is made of two independent but related control systems: a maintenance system of the assigned value of the rotor flux linkage with current I_d and a maintenance system of the assigned speed with the motor moment (current I_q).

The control system of the motor flux linkage is auxiliary and ensures the operation of an induction motor drive control system. The speed control system is the main control system of an induction motor drive and ensures compliance of its characteristics with the requirements. Obviously, static modes of an induction motor drive, both in open and closed systems, can be studied only under the following assumption: a flux control system ensures constancy of the assigned value of the rotor flux linkage [13-16].

If to assume that at a constant voltage supply of an induction motor $U_{1ph} = \text{const}$ the control system maintains constancy of the flux linkage $\Psi_{2d}(I_{1d}) = \text{const}$ along the *d*-axis, fulfilment of the condition $T_{EM}(I1q)=T_{EMref}$ will depend not only on the voltage component value $U_{1q} = \sqrt{\left(\sqrt{2} \cdot U_{1ph}\right)^2 - U_{1d}^2}$, but also, primarily, on the angular velocity of the induction motor rotation.

Therefore, if at a constant voltage of an induction motor is to take a flux linkage equal to $\Psi_{2d} = \Psi_{2d\,set}$, then the motor static characteristics $\omega(T_{EM})$ and $\omega(I_{1ph})$ in the "frequency converter-induction motor" system open along the *q*-axis can then be calculated at the assigned value of the rotor flux linkage.

A block diagram of an induction motor in two-phase rotating coordinates $d,\ q$ for the static operation mode of an induction motor drive in "frequency converter-induction motor" system at frequency-vector control is shown in Fig. 1.

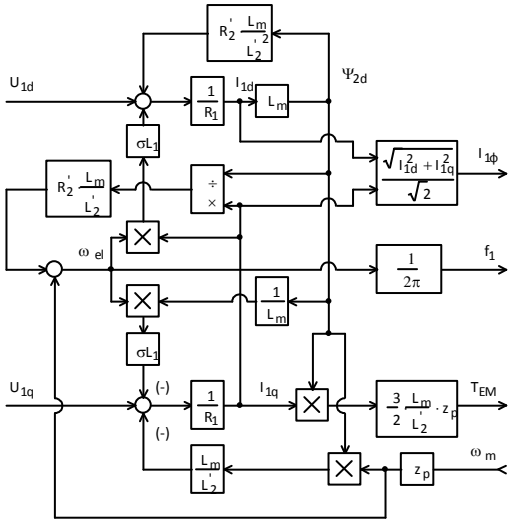


Figure 1

A block diagram of an induction motor in two-phase rotating coordinates *d, q* for the static operation mode under vector control

The system of equations describing the block diagram in fig. 1 can be presented as follows:

$$T_{EM} = \frac{3}{2} \cdot \frac{L_m}{L'_2} \cdot z_p \cdot \Psi_{2d} \cdot I_{1q}; \qquad \Psi_{2d} = I_{1d} \cdot L_m;$$

$$I_{1q} = (U_{1q} - \frac{L_m}{L'_2} \cdot z_p \cdot \Psi_{2d} \cdot \omega_{motor} - \frac{\sigma \cdot L_1}{L_m} \cdot \Psi_{2d} \cdot \omega_{electric1}) \cdot \frac{1}{R_1};$$

$$I_{1d} = (U_{1d} + R'_2 \cdot \frac{L_m}{L'^2_2} \cdot \Psi_{2d} + \sigma \cdot L_1 \cdot I_{1q} \cdot \omega_{electric1}) \cdot \frac{1}{R_1};$$

$$\omega_{electric1} = z_p \cdot \omega_{motor} + R'_2 \cdot \frac{L_m}{L'_2} \cdot \frac{I_{1q}}{\Psi_{2d}}.$$

It is known that under vector control, both a module and a spatial position of the stator current vector change [10-13, 16, 17]. The current vector changes so that the projection of the stator current vector $\vec{I}_1$ of the induction motor on *d*-axis, oriented along the vector of the rotor flux linkage $\vec{\Psi}_2$ remains unchanged and it can be determined for the first control area of the induction motor speed ($f_1 \le f_{1n}$) under the flux linkage $\Psi_{2d} = \Psi_{2n} = \text{const}$ in the following way

$$I_{1d} = \frac{\Psi_{2n}}{L_m} = \text{const}; \tag{1}$$

Component I_{1q} of the stator current vector $\vec{I}_1$, the value of which determines the motor moment, can be calculated, in the steady state mode, with the following equation:

$$I_{1q} = \frac{T_{EM}}{\frac{3}{2} \cdot \frac{L_m}{L'_2} \cdot z_p \cdot \Psi_{2n}}. \tag{2}$$

To meet the conditions (1) and (2) voltage values U_{1d} and U_{1q} must be maintained in accordance with the next equations:

$$\begin{aligned} U_{1d} &= R_1 \cdot I_{1d} - R'_2 \cdot \frac{L_m}{L'^2_2} \cdot \Psi_{2n} - \\ &- \frac{\sigma \cdot L_1 \cdot R'_2 \cdot L_m}{L'_2 \cdot \Psi_{2n}} \cdot I^2_{1q} - \sigma \cdot L_1 \cdot z_p \cdot I_{1q} \cdot \omega_{motor} \end{aligned}; \tag{3}$$

$$U_{1q} = \left(R_1 + \frac{\sigma \cdot L_1}{L'_2} \cdot R'_2 \right) \cdot I_{1q} + \left(\frac{L_m}{L'_2} + \frac{\sigma \cdot L_1}{L_m} \right) \cdot z_p \cdot \Psi_{2n} \cdot \omega_{motor}. \tag{4}$$

The active value of the motor phase voltage U_{1ph} and voltage vector components $\bar{U}_{dq}$ in a two-phase rotating coordinate system *d*, *q* are connected by the following relation:

$$\left(\sqrt{2}\cdot U_{1ph}\right)^2 = U_{1d}^2 + U_{1q}^2 . \quad (5)$$

The given equations (1) - (5) allow calculating static mechanical $\omega_{motor}(T_{EM})$ and electromechanical $\omega_{motor}(I_{1ph})$ characteristics, as well as dependence of an angular velocity on frequency $\omega_{motor}(f_1)$ for the induction motor in the "frequency converter-induction motor" system open along the *q*-axis at a constant voltage supply of the motor $U_{1ph} = \text{const}$.

Besides, equations (1) – (5) enable to determine the required maximum voltage $U_{1ph.n}$, which ensures the motor operation at the assigned values of the maximum speed of the electrical drive and the maximum moment of the static load.

Of practical importance is calculation of limiting characteristics of the motor, corresponding either to the maximum allowed value of the motor voltage $U_{1ph\,n}$ or the maximum output voltage of the converter $U_{i.ph.m}$ at the assigned value of the supply voltage. In the first case, it is assumed that the supply voltage can be selected in accordance with the allowed value $U_{1ph.m}$. In the second case, the supply voltage is assigned and determines the maximum output voltage of the converter $U_{i.ph.m}$.

3 Control Systems of Three-Phase Frequency Converter Inverters

Currently, control systems of three-phase inverters of frequency converters are implemented with a simple sinusoidal PWM, a sine PWM and a third harmonic in control signals and with a vector PWM [7, 11]. Systems with a vector PWM are controlled by sinusoidal signals and have characteristics similar to the sinusoidal PWM with a superposition of a third harmonic [7].

The control system of three-phase inverter with sinusoidal PWM has common for all three phases of inverter reference sawtooth configuration signal with singular amplitude and f_{PWM} frequency. Three sinusoidal control signals with $u_{1m} \le 1$ amplitude are buckled to input PWM block.

$u_{1a} = u_{1m}\cdot\cos(2\pi\cdot f_1\cdot t)$;

$$u_{1b} = u_{1m} \cdot \cos(2\pi \cdot f_1 \cdot t - \frac{2\pi}{3}) ;$$

$$u_{1c} = u_{1m} \cdot \cos(2\pi f_1 \cdot t - \frac{4\pi}{3}) ,$$

Common for all controls third harmonic signal is buckled to control signals in systems with sinusoidal PWM and third harmonic:

$$u_{3f} = \frac{1}{6} \cdot u_{1m} \cdot \cos(2\pi \cdot 3 \cdot f_1 \cdot t) .$$

Realization principle of control system of three-phase inverter with sinusoidal PWM and putting in three-phase system of control influences of third harmonic signal is shown on Fig. 2.

The third harmonic putting in system of control signals of inverter leads to shape change and amplitude of resulting control on input PWM block reducing to $\sqrt{3}/2 = 0.866$ times. It allows to improve amplitude of resulting control influences $u_{1a}^*(t)$, $u_{1b}^*(t)$, $u_{1c}^*(t)$ to $k = 2/\sqrt{3} = 1.1547$ times to amplitude of sawtooth reference voltage and to increase amplitude of the first harmonic of output inverter voltage to the same times.

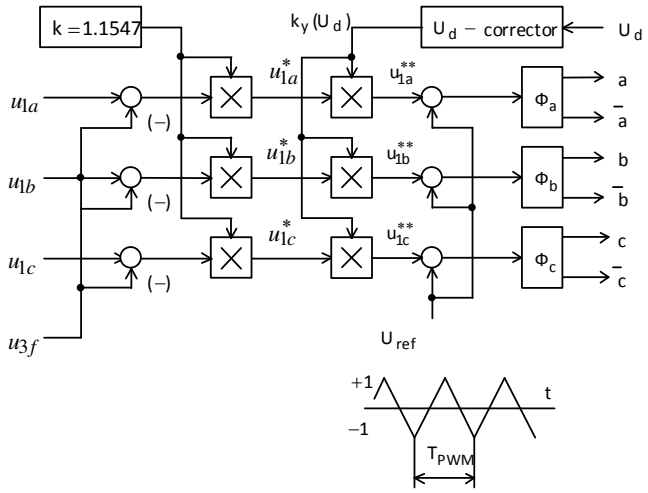


Figure 2

Realization principle of sinusoidal PWM with third harmonic putting and inverter control correction

The comparative evaluation of a simple sinusoidal PWM system and a system with an additional third-harmonic signal and an amplification gain of the modulated signal k=1.1547 are given in [7].

The automatic control system of an induction motor drive with frequency-vector control, primarily, creates and maintains the assigned (nominal, in the first region)

induction motor flux, and then creates the desired moment [1, 3, 5]. When an inverter is in an under-voltage status, the desired flux values of an induction motor and, mainly, the moment values will be achieved by changing rotational emf of an induction motor, i.e. by reducing the angular velocity of an induction motor. It can be achieved in the "frequency converter - induction motor" system by decreasing frequency of the inverter output voltage:

$$f_1 = \frac{\omega_{electric1}}{2 \cdot \pi} = \frac{1}{2 \cdot \pi} \cdot \left(z_p \cdot \omega_{motor} + R'_2 \cdot \frac{L_m}{L'_2} \cdot \frac{I_{1q}}{\Psi_{2d}} \right)$$

Thus, under vector control the inverter output voltage is the main factor when creating flux and the desired moment of an induction motor. Its value can be determined by its regulating characteristic (fig. 3) which is limited at the level $U_{iph} = U_{i.ph.m}$.

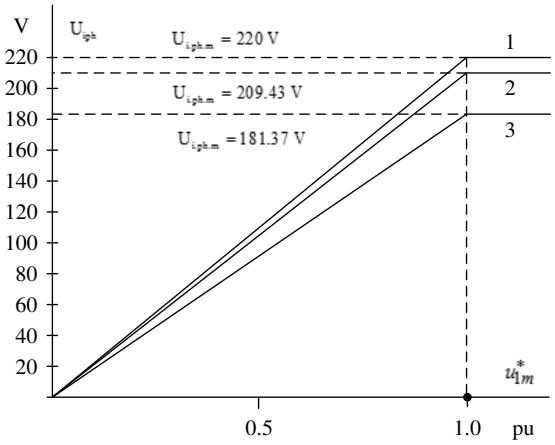


Figure 3

Regulating characteristics of a voltage inverter: 1 and 2 - sinusoidal PWM with a third harmonic at $U_{epn} = 400V$ and $U_{epn} = 380V$; 3 - a simple sinusoidal PWM and $U_{epn} = 380V$

As can be seen from Fig. 2, the inverter voltage limitation affects only in the upper part of the speed control range and almost does not affect the control system operation of the frequency converter in its lower part. Induction motor AB250S6 natural mechanical characteristic 1 ($f_1 = 50$ Hz and $U_1 = U_{1ph.n} = 220$ V) and limiting mechanical characteristics 2, 3 and 4 of the open "frequency converter-induction motor" system are shown in Fig. 3 for the next implementations of a three-phase inverter control system, respectively:

- $U_{epn} = 400$ V, a sinusoidal PWM with a third harmonic superposition and k=1.1547 ($U_{i.ph.m} = 209.43$ V);

- $U_{epn} = 380\,B$, a sinusoidal PWM with a third harmonic superposition and k=1.1547 = ($U_{i.ph.m} = 209.43\,V$);

- $U_{epn} = 380\,B$, a simple sinusoidal PWM ($U_{i.ph.m} = 181.37V$).

Characteristic 5 is a static mechanical characteristic of the motor calculated at $U_{1ph} = 181.37$ V and $f_1 = 50 \cdot \frac{181.37}{220} = 41.22$ Hz. It must be noted that a voltage drop in the inverter circuit is neglected and is taken to be equal to $U_{1ph.m} = U_{i.ph.m}$ in the given calculations of the inverter voltage.

The analysis of characteristics given in Fig. 4 shows that when a frequency converter is powered from the mains with the rated voltage, limitation of the inverter output voltage causes a substantial reduction of the induction motor speed control range in the upper part of the control region at a rated load moment up to the speed value:

$$\omega_{ED.max} < \omega_{motor.n} \cdot \frac{U_{i.ph.m}}{U_{1ph.n}}\ , \text{Rad / s},$$

and reduction of drive overload at high speeds.

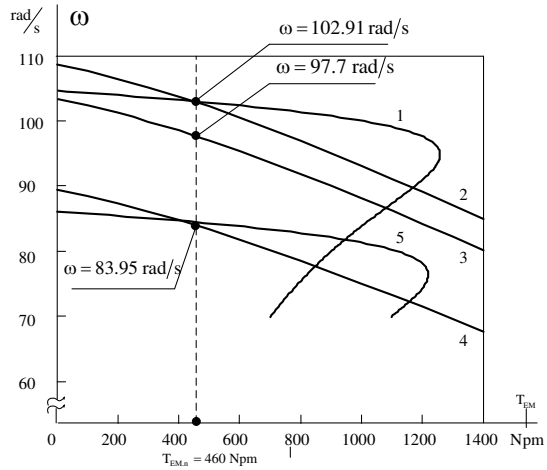


Figure 4
Static mechanical characteristics of induction motor AB250S6

Characteristics explanation of Fig. 4 is follow: 1 – natural characteristic $f_{1n} = 50$ Hz and $U_{1ph.n} = 220$ V; 2, 3 and 4 limiting characteristics of the open-loop "frequency converter-induction motor" system under vector control $U_{1ph.m} = 220$ V, $U_{1ph.max} = 209.43$ V *and* $U_{1ph.max} = 181.37$ V respectively; 5 –

forced characteristic at $U_{1ph} = 181.37$ V and $f_1 = 41.22$ Hz.

Fig. 5 shows static mechanical characteristics of the closed-loop system of an electric motor drive with frequency-vector control of AB250S6 induction motor obtained under the following conditions: $U_{1ph.max} = 209.43$ V current carrying rating $I_{ED.max} = 170$ A corresponding to the maximum electromagnetic moment of the motor $T_{ED.max} = 945$ Npm.

As can be seen from Fig. 5, the peak characteristic 2 of the open-loop system limits the maximum speed of an induction motor drive in the first control region depending on the load moment. For example, the angular velocity at point 2 is limited by the value $\omega_2 = 97.7$ rad/s, and at the maximum motor moment $T_{EM.n} = 460$ Npm it is limited by the speed value $\omega_2 = 89$ rad/s at point 1.

Characteristics of transient processes and a dynamic characteristic of an induction motor drive with closed-loop vector control at a constant nominal rotor flux linkage and an idling torque $T_{EM} = 92$ Npm, when performing the speed assignment, corresponding to fig. 5 $\omega_{set} = 97.7$ rad/s, are shown in Figs. 6, 7, respectively. The transient curves of a frequency-controlled induction motor drive, shown in Figs. 6, 7 were obtained at a constant value of the rotor flux linkage $\Psi_{2dset} = \Psi_{2n}$ and next parameters of the control system of an induction motor drive:

- supply voltage $U_{epn} = 400$V;
- PWM inverter frequency $f_{PWM} = 5$ kHz;
- AD capacity of a current transducer $n_{ADC\,cs} = 10$;
- interval for calculating in the loop current is 0.0002 sec.;
- number of speed sensor pulses (quadrupled) per shaft speed is 4000;
- interval for calculating in the speed loop is 0.002 sec.

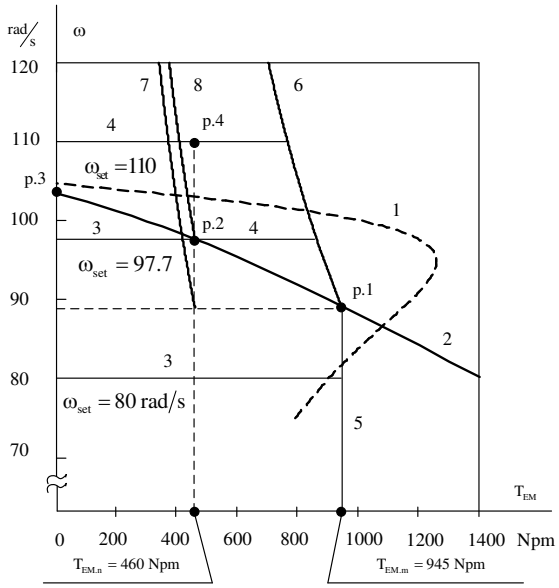


Figure 5

Static mechanical characteristics of an induction motor drive under frequency-vector control

Characteristics explanation of Fig. 5 are as follows: 1 - natural characteristic of AB250S6 induction motor; 2 - limiting characteristic of the open loop system at $U_{1ph.m} = 209.43\,V$; 3, 4 - limiting characteristics of the closed-loop system in the first and second region; 5, 6 - characteristics of admissible short-term duty; 7, 8 - characteristics of admissible long-term duty.

If we know parameters of an induction motor and assign a load moment and an angular motor speed, e.g. $T = T_{EM\,n}$ and $\omega = \omega_{motor\,n}$, the desired voltage values U_{1d} and U_{1q} can be calculated with equations (1) - (5) for an induction motor at a given point. Then, inverter voltage, DC link voltage and mains supply can be calculated with the following equations:

$$U_{i.ph.m} = 1.05 \cdot \frac{\sqrt{U_{1d}^2 + U_{1q}^2}}{\sqrt{2}};\; U_d = \sqrt{3} \cdot \sqrt{2} \cdot U_{i.ph.m};\; U_{epn} = \frac{U_d}{1.35},$$

where, the coefficient 1.05 takes into account a voltage drop in the inverter circuit.

For AB250S6 induction motor the above given load and speed values can be achieved with a sinusoidal PWM inverter, a superposition of a third harmonic and *k*=1.1547 only when the inverter voltage is $U_{i.ph.m} > U_{1ph.n} = 220\,V$. It is possible when a frequency converter is supplied from the mains with $U_{epn} > 420\,V$.

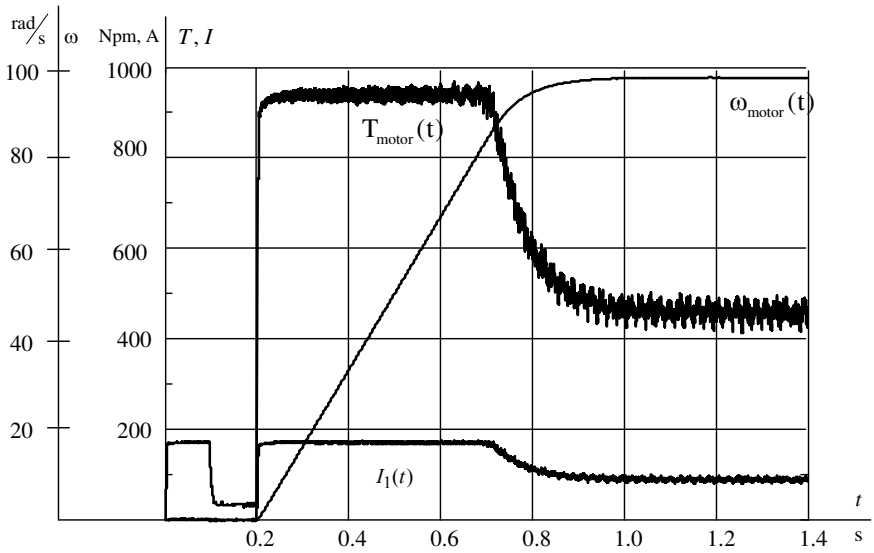


Figure 6

Transient processes $I_{1ph}(t)$, $T_{EM}(t)$ and $\omega_{motor}(t)$ when an induction motor drive performs the assignment $\omega_{set} = 97,7 \text{ rad/s}$

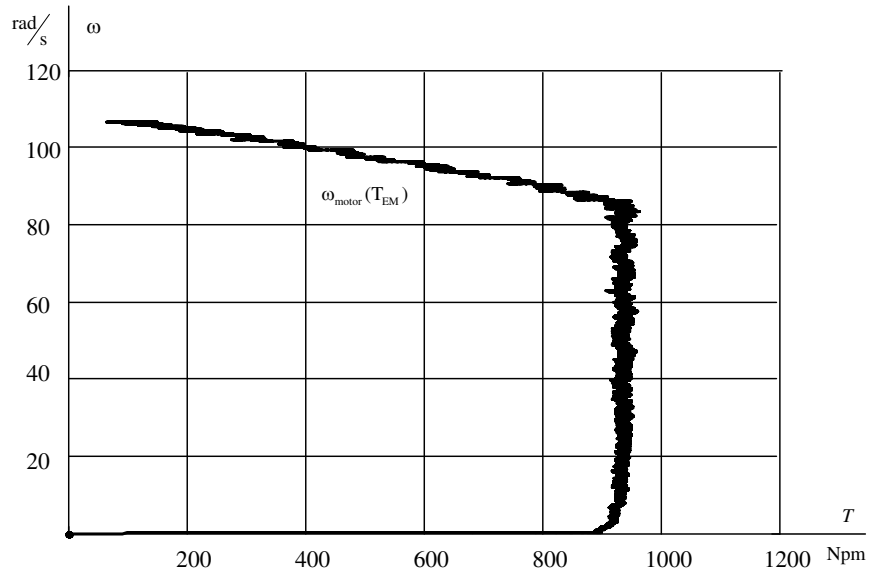


Figure 7

Dynamic characteristic $\omega_{motor}(T_{EM})$ when an electric drive performs $\omega_{set} = 97,7 \text{ rad/s}$

The desired maximum angular velocity of an induction motor under frequency-vector control in an induction motor drive can be achieved if:

- frequency converter is supplied from $U_{epn} \geq 420\,\mathrm{V}$ network, the maximum inverter voltage is limited at $U_{i.ph.m} = 1.05 \cdot U_{1ph.n}$, maintaining the coefficient *k* in accordance with the expression:

$$k = \frac{\sqrt{3} \cdot \sqrt{2} \cdot 1.05 \cdot U_{1ph.n}}{U_d} \cdot 1.1547 \leq 1.1547 \qquad (6)$$

- to increase the amplitude of control signals $u^*_{1m} > 1$ at $U_{epn} < 420\,\mathrm{V}$ network, allowing 10% increase of the maximum value of the inverter output voltage. However, it can cause a substantial increase of higher odd (higher than a third one) harmonics (the 5^{th} – up to 20%, the 7^{th} – up to 14.3%) in the inverter output signal (similar to the inverter with π-switching);

- at $U_{epn} < 420\,\mathrm{V}$ network to select a motor with excess power and run it with a constantly weakened flux, or, more reasonably, to implement a two-region speed control with flux weakening only in the second region, at the speed rate higher than the base speed according to the equation:

$$\Psi_{2d\,set} = \Psi_{2n} \cdot \frac{\omega_{start}}{\omega}, \text{ where } \omega \geq \omega_{start},$$

where ω_{start} is the selected value of the initial speed of field weakening. Here we understand base speed as speed values corresponding to the limiting characteristic of the open-loop system under true values of a load moment.

From the condition of maximum field weakening speed we must start at the initial speed at point 1 $\omega_{start} = \omega_1$ (Fig. 6). However, in this case the induction motor will have a much weaker excitation flux at a steady-state operating mode. For example, when the induction motor operates at point 2 with torque $T_{EM} = 460\,\mathrm{Npm}$ and at angular speed $\omega_2 = 97.7\,\mathrm{rad/s}$ the final value of flux linkage will be equal to

$$\Psi_{2d} = \frac{\omega_1}{\omega_2} \cdot \Psi_{2n} = 0.9 \cdot \Psi_{2n}$$

at the desired value

$$\Psi_{2d} = \frac{\omega_2}{\omega_2} \cdot \Psi_{2n} = \Psi_{2n}.$$

Similarly, when an induction motor operates at point 4 at angular velocity $\omega_4 = 110\,\mathrm{rad/s}$ the final value of flux linkage will be equal to

$$\Psi_{2d} = \frac{\omega_1}{\omega_4} \cdot \Psi_{2n} = 0.73 \cdot \Psi_{2n}$$

at the desired value

$$\Psi_{2d} = \frac{\omega_2}{\omega_4} \cdot \Psi_{2n} = 0.89 \cdot \Psi_{2n}$$

From the foregoing, it follows that the value of the initial field weakening speed must be chosen in accordance with the final value of the motor electromagnetic moment and changed in accordance with the moment changes. Thus, the initial field weakening speed of an induction motor is the function of electromagnetic moment $\omega_{start} = f(T_{EM}, U_{1ph.m})$ and represents a limiting characteristic under the final voltage value $U_{1ph.m}$, e.g., characteristic 2 at $U_{1ph.m} = 220$ V in Fig. 4.

When the supply voltage and the induction motor load (i.e. motor supply current) change, the initial field weakening value of the induction motor flux will also depend on the actual voltage on DC link and is determined with the following dependence:

$$\omega_{start} = f(T_{EM}, U_{i.ph.m} = 1.05 \cdot U_{1ph.n}) \cdot \frac{U_d}{\sqrt{3} \cdot \sqrt{2} \cdot 1.05 \cdot U_{1ph.n}}$$

where, $U_d \leq 420$ V.

The block diagram of formation of the assignment at the input control loop of the rotor flux linkage in the frequency-vector control system of the double-region induction motor drive is shown in Fig. 8. The function converter forms limiting characteristic of the open-loop system of an induction motor drive at maximum voltage of the inverter $U_{i.ph.m} = 1.05 \cdot U_{1ph.n}$

$$\omega^*_{start} = f(T_{EM}, U_{i.ph.m} = 1.05 \cdot U_{1ph.n}). \quad (7)$$

To ensure efficiency of the flux linkage control device the following conditions must be met:

- when assigning functional converter characteristics (7) the next condition must be met: $\omega^*_{start} < \omega_{start}$ at a common moment value;

- an inertial filter (F) must be in the flux linkage control channel at the time constant T_F.

The desired nature of transient processes in the second region of speed control can be achieved by selecting the time constant of the filter T_F. Selecting smaller values of the initial field weakening speed ω^*_{start} on the functional converter characteristic in the low moment area we can make an induction motor operate in the low

moment area with weakening of the flux and the lowest current consumption.

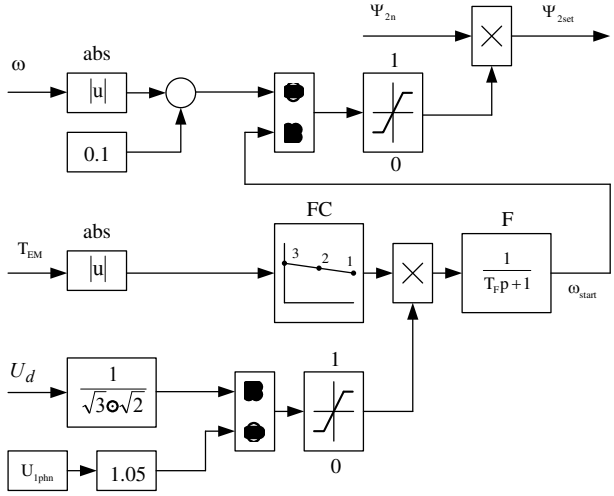


Figure 8

Block diagram of the rotor flux linkage formation at the input control loop

Fig. 9 shows dependence of the current consumed by induction motor AB250S6 when operating at the assigned speed $\omega_{set} = 0.9 \cdot \omega_n$ and the following values of the rotor flux linkage: characteristic 1 - $\Psi_{2set} = \Psi_{2n}$, characteristic 2 - $\Psi_{2set} = 0.8 \cdot \Psi_{2n}$ and characteristic 3 - $\Psi_{2set} = 0.6 \cdot \Psi_{2n}$. To reduce current consumption of an induction motor and exclude its thermal overheating under heavy loads the nominal flux linkage value must be assigned. On the contrary, under light load it is advisable to reduce the flux linkage value.

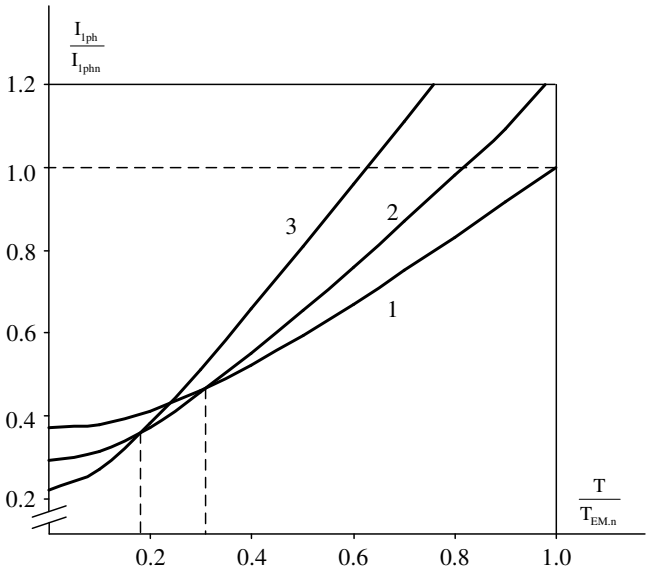


Figure 9

Dependence of current consumed by an induction motor on the load moment at: 1 - $\Psi_2 = \Psi_{2n}$; 2 - $\Psi_2 = 0.8 \cdot \Psi_{2n}$; 3 - $\Psi_2 = 0.6 \cdot \Psi_{2n}$

Characteristics of transient processes of an induction motor drive under two-region speed control when achieving the assigned speed: $\omega_{set} = 97.7$ rad/s and $\omega_{set} = 110$ rad/s at the load moment $T_{EM.n} = 460$ Npm are shown in Figs. 10-13. Characteristics of transient processes shown in Fig. 10 prove greater efficiency of the induction motor drive with controlled flux of an induction motor as compared to the characteristics of the induction motor drive with constant flux of an induction motor (Fig. 6). Characteristics in Fig. 10 and characteristic 2 in Fig. 13 correspond to the adjustment with the constant value $\omega_{start} = \omega_1$, while the diagram in Fig. 12 and characteristic 2 in Fig. 13 correspond to the adjustment with the variable value of the initial speed of weakening of the induction motor flux.

The induction motor drive systems with the constant speed value of field weakening start exhibit a slightly higher speed in the second control region, though the motor flux in the steady-state mode is too weak (characteristic 2 in Fig. 13). The main advantage of the systems with selection of speed of the field weakening start in accordance with the motor electromagnetic moment is to provide optimum value of the flux linkage at a steady-state mode of the induction motor drive (characteristic 1 in Fig. 13). It reduces the motor current consumed from the inverter and allows increasing the moment at the rated motor current in the steady state mode (characteristic 8 in Fig. 5).

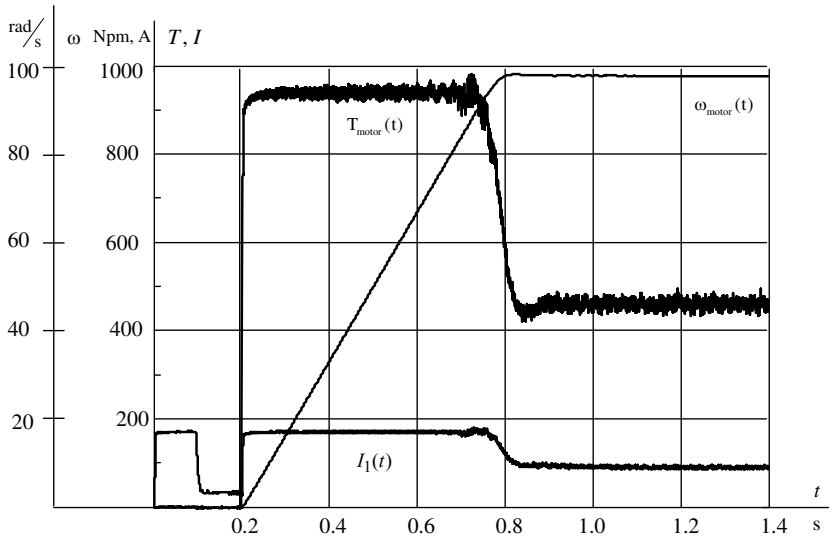


Figure 10

Transient processes $I_{1ph}(t)$, $T_{EM}(t)$ and $\omega_{motor}(t)$ when performing $\omega_{set} = 97.7$ rad / s in a double-region electric drive

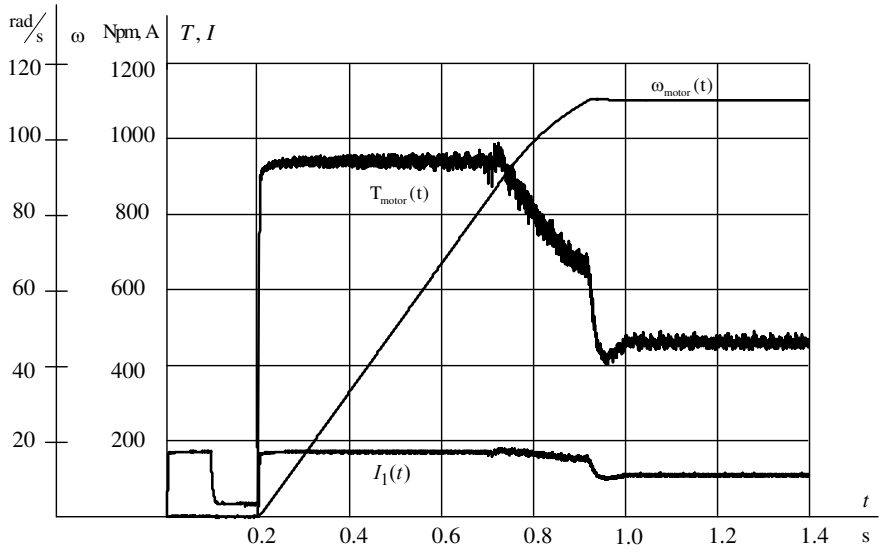


Figure 11

Transient processes $I_{1ph}(t)$, $T_{EM}(t)$ and $\omega_{motor}(t)$ when performing $\omega_{set} = 110$ rad / s and $\omega_{start} = \text{const}$ in a double-region electric drive

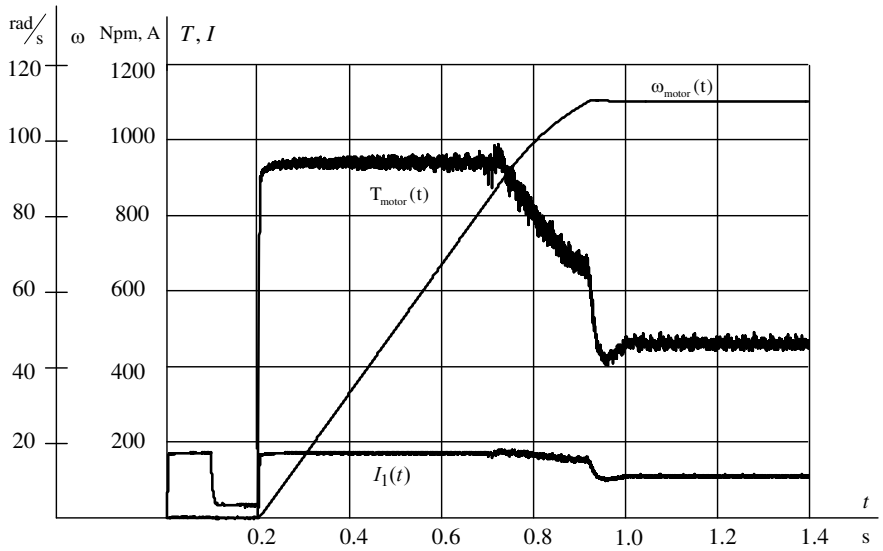


Figure 12

Transient processes $I_{1ph}(t)$, $T_{EM}(t)$ and $\omega_{motor}(t)$ when performing $\omega_{set} = 110\ rad/s$ and $\omega_{start} = f(T_{EM})$ in a double-region electric drive

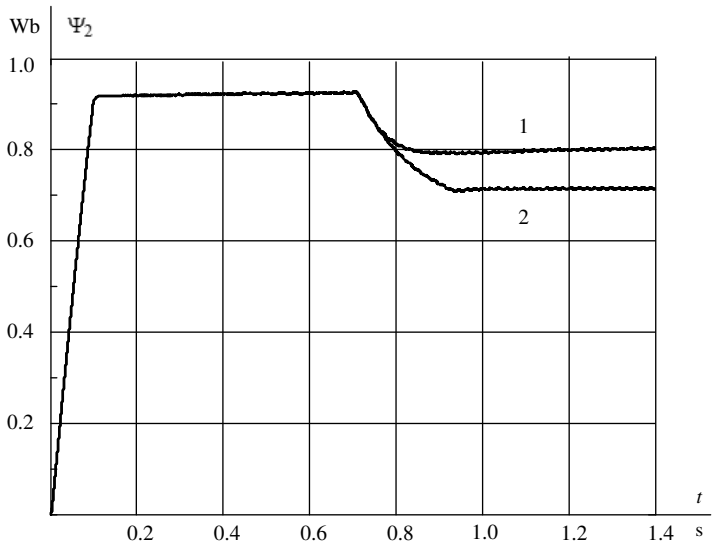


Figure 13

Flux linkage changes $\Psi_2(t)$ when performing $\omega_{set} = 110\ rad/s$ in a double-region electric drive: 1 – $\omega_{start} = f(T_{EM})$; 2 – $\omega_{start} = const$

Dynamic characteristics of an induction motor drive with a frequency-vector control given in Figs. 9-12 prove greater efficiency of an induction motor drive with controlled flux of the induction motor. Moreover, an induction motor drive with the constant speed value of initial weakening of the induction motor flux $\omega_{\text{start}} = \omega_1$ exhibits a slightly higher speed in the second control region.

4 Peculiarities of Frequency Inverter Voltage Selection under Vector Control Considering PWM of the Inverter Output Voltage

If a frequency converter has a sinusoidal PWM of the inverter, with the introduction of a third harmonic and control signals gain $k = 1.1547$, the sequence of choice of frequency inverter voltages under frequency vector control of an induction motor is as follows:

1) If a supply voltage is selected, then, using equations (3) - (7) and at the assigned values of maximum speed of an induction motor drive $\omega_{\text{ED.max}} \le \omega_{\text{motor.n}}$, static load moment M_c and known parameters of an induction motor, the desired value of maximum voltage of the motor $U_{1\text{ph. n.}}$ can be accurately calculated, and, further, if maintenance conditions permit it, the maximum output voltage of an inverter can be chosen equal to:

$$\mathrm{U_{i.ph.m}} \approx 1.05 \cdot \mathrm{U_{1ph.n}} \cdot \frac{\omega_{\text{ED.max}}}{\omega_{\text{motor.n}}}, \text{ V.}$$

$$\mathrm{U_d} \ge \sqrt{3} \cdot \sqrt{2} \cdot \mathrm{U_{i.ph.m}}, \text{ V;}$$

$$\mathrm{U_{epn}} \ge \frac{\mathrm{U_d}}{1.35}, \text{ V,}$$

In this case, the maximum value of an inverter amplification gain is taken equal to

$$\mathrm{k_i} = \mathrm{k_{i.max}} = \sqrt{2} \cdot \mathrm{U_{i.ph.m}}.$$

In a double-region electric motor drive at the assigned value of the maximum angular velocity of an electric motor drive $\omega_{\text{ED.m}} > \omega_{\text{motor.n}}$, the maximum value of phase output voltage of an inverter can be determined as

$$\mathrm{U_{i.ph.m}} = 1.05 \cdot \mathrm{U_{1ph.n}}$$

while DC link voltage of a frequency converter and the supply voltage can be calculated and chosen according to the next formulas and conditions:

$$U_d \geq \sqrt{3} \cdot \sqrt{2} \cdot 1.05 \cdot U_{1ph.n}$$

$$U_{epn} \geq \frac{U_d}{1.35}, \text{ V},$$

Then, the maximum value of an inverter amplification gain must be chosen and taken equal to:

$$k_i = k_{i.max} = 1.05 \cdot \sqrt{2} \cdot U_{1ph.n}.$$

2) If supply voltage U_{epn} is given, then DC link voltage and the maximum value of inverter output voltage can be calculated with the next equations:

$$U_d = 1.35 \cdot U_{epn}, \text{ V};$$

$$U_{i.ph.m} = \frac{U_d}{\sqrt{3} \cdot \sqrt{2}}. \text{ V}.$$

In this case, the inverter amplification gain is equal to

$$k_i = \sqrt{2} \cdot U_{i.ph.m} = \frac{U_d}{\sqrt{3}}$$

and the maximal speed of an electric drive when operating with a nominal magnetic flux will be limited by the value:

$$\omega_{ED.m} \approx \omega_{motor.n} \cdot \frac{1}{1.05} \cdot \frac{U_{i.ph.m}}{U_{1ph.n}}, \text{ Rad / s}.$$

It should be noted that final values of the maximum output voltage of an inverter and the inverter amplification gain essentially depend on the supply voltage and the motor load:

$$U_{i.ph.m.fact} = (1.41 \div 1.35) \cdot \frac{(0.85 \div 1.1) \cdot U_{epn}}{\sqrt{3} \cdot \sqrt{2}}, \text{ V};$$

$$k_{i.fact} = (1.41 \div 1.35) \cdot \frac{(0.85 \div 1.1) \cdot U_{epn}}{\sqrt{3}}.$$

If $U_{i.ph.m} > 1.05 \cdot U_{1ph.n}$, then it must be limited at the level $U_{i.ph.m} = 1.05 \cdot U_{1ph.n}$ while reducing the amplitude of inverter control signals in the function of DC link voltage U_d in accordance with the equation (6).

Therefore, when calculating settings of an electric motor drive control system, the maximum value of an inverter amplification gain must be considered:

$$k_i = k_{i.max} = 1.05 \cdot \sqrt{2} \cdot U_{1ph.n}$$

Conclusions

1) It is found that in an induction motor drive with frequency-vector control in case of output under-voltage of an inverter of the frequency converter the desired values of the induction motor flux and moment are achieved due to decreasing its angular velocity resulting from decreasing output frequency f_1.

2) In case of the mains under-voltage, a two-region control of the induction motor speed is reasonable for the "frequency converter-induction motor" system to meet the condition $\omega_{ED.m} \geq \omega_{motor.n}$. The initial speed of weakening of an induction motor flux is to be selected in accordance with the final value of the induction motor moment, using a limiting static characteristic of the "frequency converter - induction motor" system open at speed.

3) It was proved that the main advantage of control systems which have an option to select the speed of initial weakening of an induction motor flux in accordance with an electromagnetic moment of the motor $\omega_{start} = f(T_{EM})$ is a possibility to maintain the optimal value of a flux linkage in steady state modes of an induction motor drive. It ensures a bigger moment at rated current of an induction motor in steady state mode.

4) To reduce current consumption of an induction motor and exclude its overheating, the rated flux linkage value must be assigned at high load, while at light loads the flux linkage value must be reduced.

Acknowledgments

The research is funded from Tomsk Polytechnic University Competitiveness Enhancement Program grant, Project Number TPU CEP_IPE_97\2017.

References

[1] Udut L. S., Maltseva O. P., Kojain N. V. Design and Study of Automatic Electrical Drives. Part 8. Induction motor drive with frequency control. - Tomsk Polytechnic University. – 2d revised and corrected edition. – Tomsk: TPU Publ., 2014, 648 p.

[2] Sandler A. S., Sarbatov R. S. Automatic Frequency Control of Induction Motors. Moscow, Energy Publ., 1974, 328 p.

[3] Pankratov V. V. Vector Control of Induction Motor Electric Drives. Novosibirsk, NGTU Publ., 1999, 66 p.

[4] H. K. Lam, F. H. F. Leung, P. K. S. Tam Stable and Robust Fuzzy Control for Uncertain Nonlinear Systems, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, Vol. 30, No. 6, pp. 825-840, 2000

[5] Rudakov V. V., Stolyarov I. I., Dartau V. A. Induction Motor Electric Drive with Vector Control. Leningrad. Energoatomizdat Publ., 1987, 136 p.

[6] Blaschke F. Das Prinzip der feldorientierung die Grundlage fur die Transvektor – Regelung von Drehfeldmaschinen.//Siemens Zeitschrift, 1971/Bd.45, – H.10. – S. 757-760

[7] Dementyev Yu. N. , Bragin A. D. , Koyain N. V., Udut L. S. Control System with Sinusoidal PWM Three-Phase Inverter with a Frequency Scalar Control of Induction Motor // 2015 International Siberian Conference on Control and Communications (SIBCON): proceedings, Omsk, May 21-23, 2015, IEEE Russia Siberia Section, 2015, pp. 1-6

[8] Chernyshov A. Yu., Dementyev Yu. N., Chernyshov I. A. Electrical AC Drive. – Tomsk Polytechnic University. – 2d edition. – Tomsk: TPU Publ., 2015, 210 p.

[9] Shrejner R. T. Mathematical Modeling of AC Drives with Solid-State Frequency Converters. Ekaterinburg. URO RAN Publ., 2000, 654 p.

[10] Teryokhin V. B., Dementyev Yu. N. Computer Modelling of AC and DC Drives Systems. – Tomsk Polytechnic University. – Tomsk: TPU Publ., 2015, 307 p.

[11] Dementyev Y. N., Umurzakova A. D. The Engine Mechanical Coordinates Measuring in the Asynchronous Motor // (Article number 01017) // MATEC Web of Conferences, 2014, Vol. 19, pp. 1-5

[12] Odnokopylov I. G., Dementyev Y. N., Usachyov I. V., Lyapunov D. Y., Petrusyov A. S. Load Balancing of Two-Motor Asynchronous Electric Drive // 2015 International Siberian Conference on Control and Communications (SIBCON): proceedings, Omsk, May 21-23, 2015, IEEE Russia Siberia Section, 2015, pp. 1-4

[13] M. Malinowski, M. P. Kazmierkowski, S. Hansen, F. Blaabjerg and G. D. Marques, Virtual–Flux–based Direct Power Control of Three–Phase PWM Rectifiers, IEEE Trans. on Industry Applications, Vol. 37, No. 4, 2001, pp. 1019-1026

[14] Tishihiko Noguchi, Hiroaki Tomiki, Seiji Kondo and Isao Takahashi, "Direct Power Control of PWM Converters without Power−Source Voltage Sensors", IEEE Trans. on Industry Applications, Vol. 34, No. 3, 1998, pp. 473-479

[15] R. E. Precup, S. Preitl PI-Fuzzy Controllers for Integral Plants to Ensure Robust Stability, Information Sciences, Vol. 177, pp. 4410-4429, 2007

[16] A. Gharbi, M. Benrejeb, P, Borne Study of the Stabilization of Uncertain Nonlinear Systems Controlled by State Feedback, Acta Polytechnica Hungarica, Vol. 13, No. 4, pp. 21-38, 2016

[17] Glazachev A. V., Dementyev Y. N., Negodin K. N., Umurzakova A. -. Mathematical Description of an Asynchronous Motor with the Indirect Control of the Output Mechanical Variables // EPJ Web of Conferences, 2016, Vol. 110, Article number 01044, pp. 1-6

The Cheapest Way to Obtain Solution by Graph-Search Algorithms

Benedek Nagy

Eastern Mediterranean University, Faculty of Arts and Sciences,
Department Mathematics,
Famagusta, North Cyprus via Mersin 10, Turkey
E-mail: nbenedek.inf@gmail.com

Abstract: Graph-search algorithms belong to the set of basic problem-solving algorithms in Artificial Intelligence. There are systematic graphs search algorithms and also heuristic ones. Depending on the aim, e.g., to find any solution, all solutions, the best solution, one can choose an appropriate algorithm. The best-first algorithm is apt to find the best solution if a good heuristic is provided. Even, the obtained solution itself is the cheapest one, the way to obtain it may contain several useless branches. In this paper, a modified approach is shown which finds a solution having the minimal number of useless branches (depending also on the used heuristic). For the new algorithm, called minimum total cost search, the concept of the heuristic function is also changed: instead of predicting the cost of the closest goal state a kind of directed heuristic function is used: providing an estimation to the closest goal state from the given state to the given direction.

Keywords: Artificial Intelligence; problem solving; graph-search algorithms; cheapest way to obtain solution; minimum total cost search; best-first search; backtracking; heuristic search

1 Introduction

Some of the basic Artificial Intelligence algorithms are the backtracking and graph-search algorithms [1, 2, 7, 8]. Based on a state space (and the corresponding graph) representations, with their help one can find a solution of the modelled problem. Backtracking algorithm use minimal memory, actually, only the actual path is stored. At operator applications, a node with the newly obtained state is concatenated to the path. However, at backtrack steps the algorithm loses the information about the states (nodes) from that this step is made. Therefore, problems with graphs other than trees need special care. Graph-search algorithms provide other ways to obtain solutions. They store all states (nodes) that are already visited. Usually, they search/try several paths in parallel [5]. They explore

all the states that can be obtained by an operator application from the state stored at the node by expanding that node. In this way, graph-search strategies do not enter to cycles and do not repeat to try paths proved to be dead end. This is one of their main advantages over backtracking.

Optimal search is a graph-search that provides optimal solution, i.e., path connecting the start node (initial state) with a goal state with a minimal weight. In a sense, this algorithm is very similar to the Dijkstra algorithm. Here, we want to recall the main difference between the two types of problems that are addressed by the Dijkstra algorithm and the optimal search. The Dijkstra algorithm is a very efficient shortest path search algorithm for graphs. In the problem, the graph is already given, and the algorithm, using dynamic programming technique, provides shortest path(s) starting from a given node. Opposite to this, in Artificial Intelligence, in the state space, the whole graph is not already given. The algorithm builds and explores those parts of the state space and its graph that are needed to find the optimal solution. The difference is more considerable when the graph cannot be discovered for free, but we need to pay for the discovering, i.e., for building the graph itself. In this paper, we consider the problem in this latter way. Our aim is not to find the cheapest path from the start to a goal state, but to find a solution and spend the least amount of cost for building the necessary part of the graph, i.e., the total cost of the whole construction (i.e. the search) is optimized.

2 Preliminaries

In this section we are describing the basic graph-search algorithms, and especially, the best-first search, but first we recall the concepts of state space and graph representation based on [7, 8]. For all non explained concepts the reader is referred to these standard textbooks on Artificial Intelligence.

2.1 The State Space

In Artificial Intelligence one of the most known methods to model problems is provided by state space which involves deterministic actions and complete information. The state space consists of four parts which is defined by 4-tuple (S,i,O,G). A set of states that is defined by S with a single initial state i, and a set of (target or) goal states G. O is a set of available operators (actions); for every $o \in O$ it is described for each state s whether the operator o is applicable by the function $o(s) = (s',c)$ that identifies the successor state s' of some state $s \notin G$ when action o is taken and also the cost c of applying o for s (where $o \in O$, $s \in S$). The value of $o(s)$ is empty in case o is not applicable on the state s.

There is a one to one correspondence between the definition of state space and its graph representation. The graph is defined by (N,E,v_0,T), where N is the set of vertices (nodes). Each state $s \in S$ is represented by a unique vertex, i.e., there is a bijection from S to N. The initial state i is represented by the start node v_0, where $v_0 \in N$. The set of edges E contains an edge, i.e., $(v,v') \in E$ if and only if the vertices v and v' correspond to states s and s' and we have $o \in O$ such that $o(s) = (s',c)$ and the edge (v,v') have the label (o,c) indicating which operator with which cost is applied. The set of terminal nodes $T \subseteq N$ represents the goal states, i.e., there is a bijection between G and T. Since we have a bijection between S and N we may identify every vertex by a state, from here, we do not make difference between a vertex and the state it represents (if we do not have other ambiguity). Thus, we may simply refer to a vertex v by the state s assigned to it.

The aim is to find the/a path in the graph representation of the state space starting at the start node (representing the initial state) to a terminal node (representing a target state): the sequence $v_0 o_1 v_1 o_2 v_2 \ldots v_{n-1} o_n v_n$ is such path if v_i represents state s_i and $o(s_i) = (s_{i+1},c_i)$ for some cost c_i and $s_n \in T$. The cost of a path is defined by $\sum_{i=1}^{n} c_i$, as usual.

In this paper we will use heuristic search methods, therefore we need the concept of heuristic function. It is usually defined in the following way:

A function which is an estimation of the distance of the state from the closest target state (if any) is called heuristic function, which assigns a nonnegative real value to each state $h : S \to \mathbb{R}^{\geq 0}$. It is assumed that $h(s) = 0$ if and only if $s \in T$.

See [3], a survey on heuristic functions in Artificial Intelligence, for more details.

2.2 Graph-Search Algorithms

In this subsection, we give the pseudo code of a general graph-search algorithm. These algorithms generate the graph representation of the problem, called search graph, dynamically. They use list data structure to represent nodes which includes: the (actual) state, the parent node, the operator was applied for generating the actual state, and the cost from the initial state to the actual state [8]. We note that if we do not have real costs, then the number of steps, i.e., the depth of the node is stored instead.

The data structure for a node is built up from

- a state s,
- pointer pt to the parent node,
- operator o that was applied to obtain s,
- cost c,
- heuristic value $h(s)$.

For non-heuristic search algorithms we do not need the heuristic values, that is we can write 1 (or the smallest positive unit used by the algorithm) for states not in T and 0 for states in T. This value will not modify the work of the algorithm.

The basic steps of graph-search algorithm are expanding the search graph as follows:

Algorithm 1 (Function Expand)

function Expand (v).

1. For each operator o applicable to the state s stored at node v do
   - 1.1. Apply o to s and hence obtain (s',c)
   - 1.2. If there is not vertex v' such that it stores s',
     - Then create vertex v' with data: s',v,o,c_v+c, where c_v is the cost stored at vertex v. Put vertex v' into *OPEN*.
     - Else decide whether or not to change the pointer pt of v' to v and its cost to c_v+c.
       - If v' is found in the list *CLOSED,*
         - Then decide for each of its descendants in G whether or not to rewrite the stored cost, accordingly.
2. End for
3. Return % End Expand

Algorithm 2 (Graph-search)

1. Create a *search graph, G,* containing only the start node, s:
2. Let the list *OPEN* contain only the initial state in the start node, s.
3. Create a list called *CLOSED* that is initially empty.
4. While (*OPEN* is non empty)
   - 4.1. Select the first node in *OPEN,* remove it from *OPEN,* and put it on *CLOSED.* Call this node $v \in N$.
   - 4.2. If $v \in T$, i.e., it is a target node,
     - Then terminate successfully with the solution obtained by tracing a path along the pointers from v to s in G.
   - 4.3. Call the function Expand with parameter v.
   - 4.4. Reorder the list *OPEN,* %either according to some arbitrary scheme or according to heuristic merit%.
5. End while
6. Return failure (there is no solution with this representation).

The nodes in *OPEN* are the tip nodes of the graph-search, and the nodes on *CLOSED* are the non-tip nodes. In Breadth-first and Depth-first search algorithms there is no real cost function, unit cost, e.g., the depth is used. In the Breadth-first search algorithm with a queuing function the newly generated states put at the end of the queue, after all the previously generated states. However, the Depth-first search algorithm puts the newly generated states at the front of the queue (more like a pushdown stack architecture).

The algorithm ends successfully whenever the selected node for expansion is a target node. The desirable path $v_0o_1v_1o_2v_2\ldots v_{n-1}o_nv_n$ can be obtained by tracing the pointers from v_n to v_0 in the reverse order (if the/a solution exists). Whenever the set of terminal nodes are empty, i.e., $T = \emptyset$, or no terminal node can be obtained from the start node using the given operators, the algorithm terminates reporting unsuccessful search (failure).

Since this is a graph-search algorithm some of the states obtained by Expand may already be in *OPEN* or *CLOSED*, i.e., they have already been generated. Recognizing and deciding what to do if some of the newly generated data are already generated before, has some computational cost. In the simplest cases, e.g., when the cost of the obtained solution is not of high importance, there is no extra process for the states that are already in the database. For Breadth-first, Depth-first, and, as we will see, for best-first search algorithms in the else branch of step 1.2 of the function Expand can be deleted (no change will happen if a state is already stored in some nodes of the database, i.e., in the already obtained search graph).

In this paper, we would like to obtain a solution as fast and in the cheapest way as possible. Therefore, two specific graph-search algorithms, namely, the optimal search and the best-first search are the most important for us. We briefly recall them in the next two subsections.

We note that relations of graph-search and parallel algorithms are investigated in [1, 5]. However, in this paper, we mainly deal with the traditional, sequential approach.

2.3 The Optimal Search

A systematic search algorithm that provides the optimal (minimal cost) solution (in case a solution exists) is the optimal search. Since the aim is to find the best solution (minimal cost path from the start node to a terminal node) in the Expand function if a cheaper path is found to a node v' than the actually stored path (through the links to the parent nodes), then the parent node of v' and the cost of the path to node v' are updated. However, it is not possible to find a cheaper path to node that is already in the list *CLOSED*. In this search algorithm, the list *OPEN* is sorted in a non decreasing way by the stored cost of the path to the node.

At the next algorithm, we do not want to get the optimal solution (for sure), but we want to obtain a solution as fast as possible. This can be done by the use of a heuristic function.

2.4 The Best-First Search

One of the search algorithms that searches in a graph by selecting and expanding the most promising node based on the special rule is the Best-First search. This algorithm selects the promising node by using a heuristic function, $h(s)$, which takes a state (stored in a node v) as argument and returns a non-negative real number. At this algorithm, the nodes in the list *OPEN* are ordered (may be based on the variety of heuristics ideas) so that the "best" one will be the first in the list, and thus, it can be easily selected. The heuristic function estimates the cost of a path from node v to a target node. We then expand the node to generate its successors and quit if one of them is a target. Otherwise, among all the other nodes (including the newly generated ones), again the most promising node (i.e., the node containing the state with the smallest heuristic value among the nodes stored in the list *OPEN*) is chosen and the process continues. By using a heuristic function, the search can be informed about the direction to a target and predict how close the end of a path is to a target [6, 7]. It is known that Best-First search can provide a solution in the fastest way if the heuristic function is informative enough.

The function Expand for best-first search algorithm is specified from Algorithm 1, as we have already indicated, by simply deleting the Else branch of step 1.2. In Algorithm 2 the reordering of the list *OPEN* (step 4.4) goes by non-decreasing values by the heuristic values of the states stored.

We note that in [4] search algorithm based on the best-first is presented using the memory in an efficient way.

3 The New Approach

The new approach is somewhat similar to the Best-First search algorithm, it is a kind of modification of the usual graph-search algorithms reducing the cost of the applied operators. The heuristic function is also redesigned for this purpose.

3.1 Appropriate Type of Heuristic Functions

The heuristic function, to obtain our aim, has two arguments, not only a state, but also a chosen direction to go forward from that state, i.e., an operator. By the help

of this new feature, we can reduce the cost of the applied operations. In this new algorithm, the data structure that is stored contains vertices with the following data structure:

- a state *s*,
- pointer *pt* to the parent node,
- operator *o* that was applied to obtain *s,*
- cost *c,*
- an applicable operator *o*' (that is chosen to apply to state *s*)
- heuristic value *h*(*s*,*o*').

The heuristic function is applied to each element of the *OPEN* list. In this algorithm, we have again the two lists. At a node the state, pointer, operator and cost are the same as before. The new part is a chosen operator and the new type of heuristic function. Now, we are ready to present the new algorithm with the appropriately modified Expand function.

3.2 The Minimum Total Cost Search

To reduce the cost (of the applied operations) in the Expand function, we should not use all the applicable operations for a given state at a time, we continue the search only in a chosen direction. In this way, the algorithm has a similarity to the backtracking search. The list *OPEN* is assumed to be sorted by the stored heuristic value *h*(*s'*,*o*) in non-decreasing order.

Algorithm 3. (Function Best-Continue)

function Best-Continue(*v*)

1. Apply *o*' stored in *v* to *s* stored in *v* and hence obtain (*s'*,*c*).
2. If there is no such node in *OPEN* and *CLOSE* that stores the state *s'*
   Then
   2.1. If *s'* is a goal state,
   Then terminate with the solution reaching *s'* from *s*
   (and its predecessors).
   2.2. For each operator *o* applicable to the state *s'* do
   Create a node *v'* to store state *s'*, *pt* to node *v*, *o'*, *c*, then the applicable operator *o* and the heuristic value *h*(*s'*,*o*).
   Insert *v'* into the list *OPEN* in non-decreasing way by *h*(*s'*,*o*).
   2.3. End for
3. Return % End Best-Continue

Algorithm 4. (Minimum Total Cost Search, MTCS)

1. Create a *search graph, G,* containing only the initial state with its applicable operators:
2. For each operator o applicable to the initial/start state s do
   2.1. Let the list *OPEN* contain the node: *s. NUL, NUL,* 0*, o,* $h(s,o)$. Insert it into *OPEN* by the values $h(s,o)$ in non-decreasing way.
3. End for
4. Create a list *CLOSED* that is initially empty.
5. While (*OPEN* is non empty)
   5.1. Select the first node of *OPEN,* remove it from *OPEN,* and put it into *CLOSED.* Call this node $v \in N$.
   5.2. Call the function Best-Continue with parameter v.
6. End while
7. Return failure (there is no solution with this representation).

The algorithm is complete and correct: For finite search graphs, if a solution exists, it will find a solution; and it terminates with failure if there is no solution. However, for problems represented by infinite graphs, it may not produce solution if, e.g., the heuristic function directs the search to an infinite branch without solution. This is actually, the same phenomenon for other heuristic search algorithms, including heuristic backtracking and best-first. (To overcome this issue one may use the optimal search, or its mix with the best-first, i.e., the A- or A*-algorithms.) However, if the heuristic is not completely misleading in an infinite search space, the new algorithm can be applied. Moreover, due to the new type of heuristic function, the total cost of the explored search tree is minimized (according to the quality of the heuristic function).

The new approach is between the graph-search and backtracking, by breaking the expand function into parts applying operators one-by-one, similarly as they are used in backtracking algorithms. On the other side, since we have no backtrack operation, we keep all the already explored parts of the search graph, the new method also inherits the advantages of graph-search algorithms (e.g., loops and alternative paths to the same dead end do not cause such problems that could occur at backtracking).

We note that by efficient practical design we do not really need to store the repeated information (i.e., s, pt, o and c) at nodes storing the same state. By using linked (multi)lists we can store this part of the information separately and this part can be linked by a pointer to the real node that contains additionally o' and $h(s,o')$.

3 Example

In this section, an example is shown. We also compare the work of the heuristic backtracking, the optimal search, the best-first and our new MTCS algorithm. Let the graph representation of our problem be given as it is shown in Figure 1. The cost assigned to the operator application is written at the left side of the edges by blue color. The heuristic values (indicated by red) are given in Table 1.

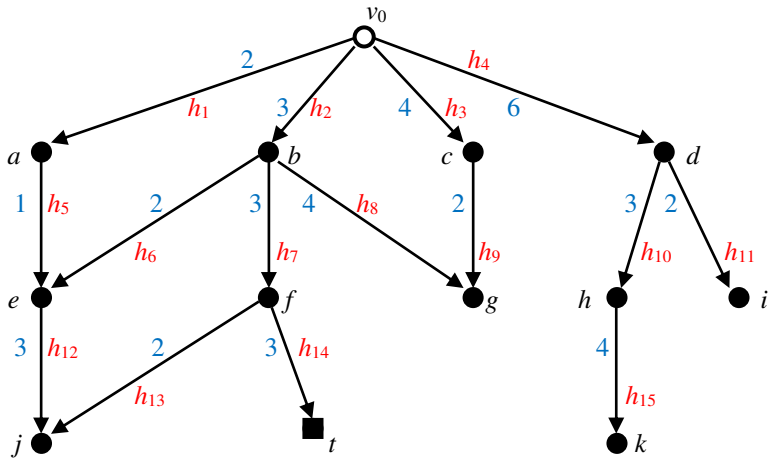


Figure 1

The graph of the state space representation of the example

In the example, we use two heuristic functions, the perfect heuristic and a non-perfect (another) heuristic (which is more realistic), as specified in Table 1.

Table 1

Heuristic values used in the example

h_i	i =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
perfect		99	9	99	99	99	99	6	99	99	99	99	99	99	3	99
another		5	10	11	14	4	5	6	4	7	6	8	3	4	3	4

$h(v)$	v:	v_0	a	b	c	d	e	f	g	h	I	j	t	k
perfect		9	99	6	99	99	99	3	99	99	99	99	0	99
another		10	4	5	7	7	3	3.5	99	4	99	99	0	99

The upper part of the table shows the new type of heuristic functions that are apt to use for backtracking and the MTCS algorithm. The lower part of the table

shows traditional heuristic functions that can be used for the best-first algorithm. For the "another" heuristic the average new-type heuristic values of the out-edges are used at the corresponding nodes.

Let us see, how the solution is obtained by various algorithms. Let us start with the optimal search. Obviously, it starts by expanding the node v_0. By applying all the 4 operators it obtains 4 new nodes (*a*,*b*,*c*,*d*) and this costs 2+3+4+6=15. The nodes in *OPEN* are ordered by their cost value, and thus, node *a* is expanded in the next step with cost 1: node *e* is obtained. In the next 2 steps nodes *b* and *e* are expanded, by costs 9 and 3, respectively; adding the nodes *f*, *g* and *j* to the explored search graph. Then, node *c* is expanded having a better path to *g*, and adding 2 to the total cost (of the graph we have paid so far). Now the nodes *j*, *f*, *g* and *d* are expanded: this gives 0+5+0+5 cost and adding nodes *t*, *h* and *i* to the explored part of the graph. Node *i* has the smallest cost among the nodes in *OPEN*, therefore it is expanded (it is moved to *CLOSED* without other changes). Now, *t* is among the nodes with the smallest cost in *OPEN,* and it can be tested that it contains a goal state. The solution is found: v_0 to *b*, then to *f* and from *f* to *t*. Even the cost of the solution is 9, the cost of the explored part of the search graph is 40.

Now let us see how the heuristic search algorithms works with perfect heuristic. The best first must start by expanding the start node v_0 and, hence, obtaining nodes *a*, *b*, *c*, and *d*, it costs 15. Then, the smallest heuristic value is 6 and it is at node *b*. By expanding *b* the nodes *e*, *f* and *g* are obtained with additional cost 9. Now, the smallest heuristic value is 3 (at *f*) among the nodes in *OPEN*. Consequently, expanding *f*, the nodes *j* and *t* are explored (cost 5), and the node *t* has 0 heuristic value. The search is finished, the solution is found, and the total cost of the search was 29.

The backtracking using the perfect heuristic given in the first row of the table, will go from v_0 to *b* and, then to *f*, and it finds the terminal node *t* in the next step. The total cost of the search is 9, it is exactly the same as the cost of the solution (it was obtained without backtrack steps).

Algorithm MTCS works in a similar manner. In the beginning, it has basically 4 edges in the open list indicated by the values h_1, h_2, h_3 and h_4. It choses the edge v_0 to *b* as h_2 is the smallest among the values in the list *OPEN*. Using Best-continue, the edges indicated by h_6, h_7 and h_8 are found and stored in *OPEN*. Then, the edge with h_7 is chosen for Best-continue. In the next round, by the edge with h_{14} the solution is found. The total cost of the MTCS search was 9, same as the total cost of the backtracking (with perfect heuristic values).

Let us see the performance of the heuristic search algorithms with a non-perfect heuristic function. One can easily check that, in our example, the best-first search finds the solution by using the heuristic values from the last row of Table 1, with an even larger total cost than previously: it is 33. It is usual that without a really

good heuristic function the performance of the heuristic search algorithms are worst than with them. (Actually, the quality of the heuristic function is essential.)

Let us see, how the backtracking works: It will go from v_0 to a, and then to e, and to j (with total cost 6, so far). In the next step, it realizes the dead end, and backtracks to v_0. Then, it goes to b and then to g. The total cost of used operators is 6+7=13, so far. However, it realizes the dead end, and backtracks to b. Then it continues to e, again, moreover to j (the total cost is 13+5=18, so far). Again, realizing the dead end backtracks and backtracks to b. Now, it goes to f, and, finally, from f it finds the terminal node t. The total cost of the search is 24, it is much larger than the cost of the solution, even the backtrack steps had no direct extra costs.

Finally, we show the performance of MTCS with the non-perfect heuristic values specified in Table 1. Again, the list *OPEN* contains the edges indicated by the values h_1, h_2, h_3 and h_4. However, in this case, h_1 looks the best choice (and it costs 2). Then instead of that edge the edge with h_5 will be in the list *OPEN*. In the next round, it seems the best choice and the edge indicated by h_{12} will replace it in *OPEN*. Then edge with h_{12} is chosen and moved to *CLOSED*. The total cost of the search is 6, so far. In the next round the edge h_2 is chosen for Best-continue. Consequently, *OPEN* will include the edges with h_6, h_7 and h_8. Then the edges with h_8 and h_6 are chosen, and both become *CLOSED*. The total cost is 6+9=15 up to this point. Then the choice of the edge with h_7 to Best-continue gives two new *OPEN* edges: the ones with h_{13} and h_{14}. The latter one is a better choice and the algorithm terminates with the solution. The total cost to find it was 21, less than the costs of the other algorithms. Of course, without perfect heuristic values we usually need to pay some extra costs to explore some parts that are not directly needed for the solution, however, by our algorithm this extra cost is minimized.

Backtracking algorithms keep only a path in their memory, and thus, with a heuristic backtracking algorithm with a good heuristic function the solution may be obtained without any backtrack steps (see, the solution with perfect heuristic). In this way, the total cost to find the solution is exactly the same as the cost of the solution itself. However, if the heuristic function is not good enough, the total cost to obtain the solution increases, and disadvantages of the backtracking algorithm may occur, e.g., by applying (and paying again for) the same operator at the same state reached in a newer path, as we have seen at the case of "another" heuristic.

Even best-first could be believed as a method exploring the minimal part of the search graph to obtain a solution, we have shown that our new algorithm can work with even less cost. The best-first pays extra fees when at expand, it is applying operator(s) at a node not only in the ideal direction.

Conclusions

There are some cases in real life when we do not have a chance to freely see what and how will happen if we do something (modelled by applying an operator). In these cases, to try and analyze how a given step may help to find a solution (to reach a goal state), one may need to pay the cost of this step (operator). Consequently, the problem to find the cheapest (optimal) solution shifts to the problem to obtain a solution in the cheapest way, i.e., exploring a minimum-cost part of the search tree that is needed for the solution.

In this paper, we have modified the well-known search algorithms and we have obtained a general heuristic algorithm to have a search algorithm with minimum total cost. The algorithm is related to the best-first graph-search algorithm and also to heuristic backtracking algorithms uniting their advantages for the considered types of problems.

References

[1] Kenneth A. Berman, Jerome L. Paul: Algorithms: Sequential, Parallel, and Distributed. Thomson/Course Technology, 2005

[2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Introduction to Algorithms. MIT Press and McGraw-Hill, 1990 (3rd edition, 2009)

[3] Fred Glover, Harvey J. Greenberg: New Approaches for Heuristic Search: A Bilateral Linkage with Artificial Intelligence, European Journal of Operational Research 39/2 (1989) 119-130

[4] Richard E. Korf: Linear-Space Best-First Search, Artificial Intelligence 62/1 (1993) 41-78

[5] Benedek Nagy: On the Notion of Parallelism in Artificial and Computational Intelligence, Proceedings of the 7th International Symposium of Hungarian Researchers on Computational Intelligence, Budapest, Hungary (2006) pp. 533-541

[6] Ira Pohl: Heuristic Search Viewed as Path Finding in a Graph, Artificial Intelligence 1/3-4 (1970) 193-204

[7] Elaine Rich, Kevin Knight, Sivashankar B Nair: Artificial Intelligence. Tata McGraw-Hill, 3rd edition, 2008 (Elaine Rich, Kevin Knight: Artificial Intelligence, 1st edition, 1983)

[8] Stuart Russell, Peter Norvig: Artificial Intelligence – A Modern Approach. Prentice Hall, 1995 (3rd edition, 2009)

External Rapid Prototyping Validation System for the Automotive Development Cycle

Ioan Gheorghe Dubar[1], Razvan Bogdan[2], Mircea Popa[2]

[1]FEV ECE Automotive, Neuenhofstraße 181, 52079 Aachen, Germany, dubar_i@fev.com

[2]Politehnica University of Timisoara, Faculty of Automation and Computers, No. 2, blv. Vasile Pârvan, RO-300223 Timişoara, România, razvan.bogdan@upt.ro, mircea.popa@upt.ro

Abstract: The most pressing requirement currently faced by the automotive industry is the speed at which a certain product can be offered to the market. The present paper discusses the applicability of the Rapid Prototyping method in the automotive industry, illustrated as a development approach of a potential solution to the early evaluation phase of a system. The offered method advances an implementation of a pilot approach in the context of the final product. The functioning of the prototype is measured in different test scenarios, the results being increasingly encouraging towards the industrial adoption of this technique from a technical as well as an economic point of view.

Keywords: Automotive; Engine Control Unit; External Rapid Prototyping; Performance

1 Introduction

The automotive industry is among the leading divisions of the currently-emerging economy. Different corporations have been created with the aim of meeting the complex requirements of a large spectrum of customers. A plethora of technologies, open problems as well as products are living proof that this field is one of the most dynamic industrial domains. Different customers from all the continents offer large amounts of money so that competitive products can be launched on the market. The key ingredient of such an industry has become the speed at which a corporation offers a certain product to the market. Different techniques, processes and methodologies are being researched and tested so that the life cycle of a product is shortened as much as possible.

In order to face such a situation, in which automotive control systems are more and more complex and the economic factor is an impactful constraint in developing a final product, the great urge of testing and validating new concepts

during the development process has led towards new prototypes [1]. These are executable models of a system that accurately reflect a chosen subset of its properties. In other words, one of the solutions to such a problem is to create a system that allows to develop and test new implemented requirements and concepts that are aimed at being included in the final automotive product. To obtain these prototypes, one of the methods that could be applied at the software and hardware level is the Rapid Prototyping method. This paper presents such a solution that could be successfully applied in the automotive industry.

The Rapid Prototyping method can be applied in two different approaches, the External Rapid Prototyping (eRPT) and the Internal Rapid Prototyping (iRPT). The External Rapid Prototyping does not affect the processing capabilities of the Electronic Control Unit (ECU). All the processing resources that must be available for eRPT implementation are ensured by an external Rapid Prototyping Unit (RPU). There is also a second version of Rapid Prototyping, which is an alternative to External Rapid Prototyping and is called Internal Rapid Prototyping (iRPT) [2]. These two methods of Rapid Prototyping have basically the same principles, however they differ at the implementation level. The SDA-iRPT (System Design Automation) uses free resources of the running ECU and can be implemented only if the ECU has enough unused memory space for the model code, the model RAM and the model calibration data in the ECU memory. Furthermore, enough free processor runtime is needed to run the bypass-model.

To have a better overview of the system, it should be mentioned that the experimental setup of our approach is a serial production ECU that can be found on Ford C-MAX cars. This ECU is equipped with a MPC561 microcontroller, provided by Freescale. This microcontroller provides only 32-kbytes static RAM and 40 Mhz processing frequency. Due to these limited resources, the chosen method for applying Rapid Prototyping is the External Rapid Prototyping (eRPT) method. In order to be able to apply such a method, external data processing power will be applied and the validation will be performed by means of a new injectors control algorithm. The external processing unit and the hardware interface that makes possible the communication with the ECU are provided by dSPACE as well as the entire tool chain, from control tools for experiments to specific eRPT libraries for Matlab Simulink for the experimental models.

The content of the paper is structured in four main sections as follows: the first reviews previous cases where Rapid Prototyping has been successfully applied; the second describes the proposed solution aimed at system overview, hardware and software implementation and the adaptation of the model for External Rapid Prototyping; the third section discusses the results achieved, whereas the last section formulates conclusions and future research directions.

2 Related Work

Rapid prototyping methods are present in many branches of the industry, especially in those where the development phases are expensive and time consuming. According to the already published state-of-the-art scientific literature, this method has not been previously applied in the area of Electronic Control Units. Therefore, this section of the paper will have two distinct directions: it will present those automotive areas where the applicability of this technique has been researched and different results are available, and then other industrial domains which have successfully benefitted from this method will be presented.

Rapid prototyping has been successfully applied in different fields of the automotive industry. In [2], two approaches are presented. First, a methodology and integrated tool set for rapid prototyping of communication systems is discussed which uses programmable boards from high-level requirements. The prototype for the system is obtained via synthesis. Secondly, a Virtual Component CO-Design (VCC) environment is used to model the distributed system being composed of Electronic Control Units (ECUs), different functions and communication protocols. Even if this solution supports two methodologies, namely virtual rapid prototyping of communication protocols and virtual and physical prototyping of applications, only preliminary results have been obtained so far. In [3] the case of applying the rapid prototyping approach to design and test active vibration control systems is presented. This solution uses the dSPACEMicroAutoBox rapid prototyping systems, with the support of tools such as MATLAB/Simulink/Stateflow. By using such a method, it has been successfully proven that important reductions in noise and vibrations can be achieved. In [4] a case study of rapid prototyping from the automotive industry is presented, namely the identification of usability problems at the beginning of the software life cycle. This paper aims at offering some results on the effectiveness and performance of prototypes in usability scenarios. In [5] a Real-Time model that substitutes a real DC motor is presented for designing a cost effective Power-Hardware-In-the-Loop. The rapid prototyping method is used in order to simulate behavior characteristic of the real DC motor.

In additon to the automotive industry, rapid prototyping has been used in the case of robot manipulators as well [6]. Force sensors are also used in robotic systems, providing critical information about the robot manipulators. The main downsides of force sensors are excess sizes, cost and fragility. In order to improve these aspects, 3-D printing is used for obtaining a quick and inexpensive development process. The big advantage of rapid prototyping, combined with 3-D printing, is that a final product is developed and built faster, it is easy to customize it and can be shared with other developers in the field in an open source approach. Other application domains of rapid prototyping method are wireless sensor networks,

internet of things, FPGAs [7-16], as well as medicine [17, 18], robotic systems [19] and power electronics [20].

3 Proposed Solution

The solution advanced by this paper is based on the implementation of the External Rapid Prototyping method in order to address the above mentioned shortcomings within the automotive industry. An automotive hardware/software system which allows the practical implementation of the External Rapid Prototyping has been implemented and the performance of the obtained system has been measured, as well as the gain in terms of V-cycle development process. The motivation for applying this method is to obtain an improved performance of control algorithms for injectors, through, which is made the fuel injection in a thermal motor. Moreover, an important aspect is the fact that the system is based on a serial production Engine Control Unit (ECU SID 803), found on mass production cars. The following steps are proposed to be followed in order to obtain the solution:

a. Hardware implementation of the system
b. Mathematical model Matlab\Simulink for Rapid Prototyping
c. Software implementation
d. Performance analysis of the entire system.

The concept of External Rapid Prototyping is quite new in the automotive field and the word “external”, placed before Rapid Prototyping, means that the processing data capabilities for automated code generation from an improved model of the target system and the execution of this code are ensured by external hardware. In this way, the impact on the computing resources of the engine control unit is smaller.

The following subsections will present an overview of the system and all the stages of the implementation, hardware, software and model adaptation.

3.1 System Overview

A system overview from the hardware viewpoint of the proposed solution is presented in Figure 1.

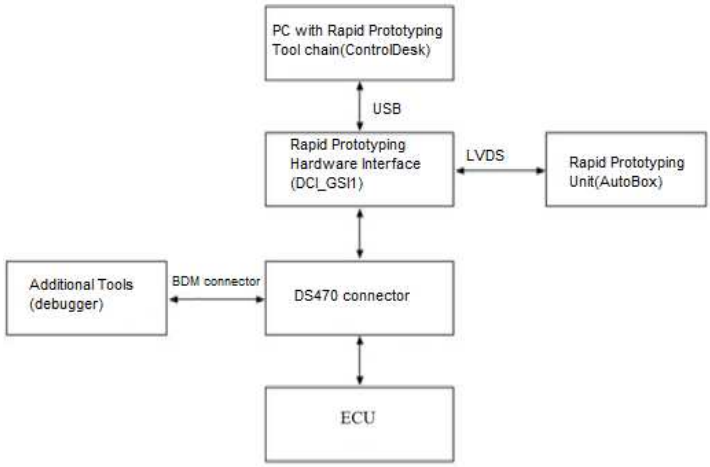


Figure 1
Hardware overview

The main device of our solution is the ECU (Electronic Control Unit) of the car. The ECU SID 803 is used, that is found on the Ford C-Max (Engine: Duratorque 2.0L, DW10B, Power: 136HP, Injection: Common Rail Piezo, Transmission: Manual 6 gears) and it is equipped with the MPC561 micro-controller.

The DCI_GSI_CON1 (DS470) is a NEXUS/READI connector adapter for the MPC561 microcontroller which makes possible the communication between the ECU and the Rapid Prototyping Unit. So as not to alter the functionality of the whole circuit and in order to have both devices functional, i.e. the debugger (Trace32) and the DCI_GSI1 (Generic Serial Interface), a hardware wiring of the mapping was made directly between the corresponding debugger adaptor pins and the DCI_GSI_CON1 pins. In this way, the native debug interface remains functional.

Many microcontrollers used in the automotive ECUs today provide internal units, for example overlay RAM for supporting ECU calibration, measurement or debugging. These internal units can be accessed by a chip interface such as a debug interface or any other serial interface which allows memory read/write access by an appropriate protocol (for example, NBD/AUD or Nexus). The DCI-GSI1 uses this interface to access the internal units to perform ECU calibration, measurement and bypassing. The DCI_GSI1 uses the NEXUS/READI interface to facilitate the communication between ECU and AutoBox.

The Rapid Prototyping Unit for our solution is the Autobox. This hardware is provided by dSPACE and has several features that make it suitable for eRPT. Among them one can mention AutoBoot options for making stand-alone operations, existence of several hardware interfaces for connection and communication with a host PC or supply voltage from a car battery. Usage of

Autobox is well suited to experiments performed directly on real-time systems inside a car. It is equipped with a rugged case, inside of which one can mount up to six boards, processing data collected from the system, or even interfacing with other equipment boards.

By means of a host PC/Laptop we can control the whole development process, for example collecting useful data from the ECU or making the desired adaptation on the ECU code. For this, the PC must be equipped with a certain tool chain that consists of:

a. SDA-Matlab/Simulink with RPT configuration –SDA (System Design Automation) is a Matlab/Simulink base tool;

b. ControlDesk - a software tool for controlling, monitoring and automation experiments using dSPACE hardware;

c. INCA - a tool that allows, based on a .a2l file, the monitoring of all variables contained in the code flashed on the ECU.

On the other hand, from an architectural viewpoint, the system is structured as presented in Figure 2. This illustrates the dependencies between different levels of the whole system.

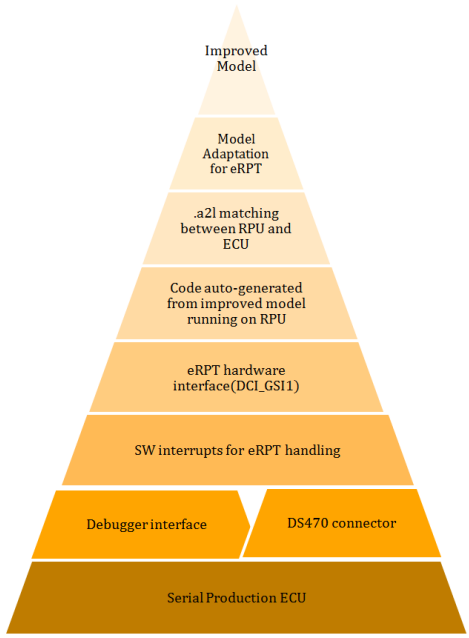


Figure 2
System architecture

At the bottom of the pyramid is placed the ECU and after that the DS470 connection tower and debugger connector. These two devices are placed on the same level because through them the ECU communicates with the superior layers. Through the debugger interface, the software suitable for Rapid Prototyping can be flashed on the ECU. By having this layer available, through the Rapid Prototyping hardware interface, the communication with the RPU (Rapid Prototyping Unit) is ensured. On this unit, the code is generated from a prototype model, which can be improved for a certain functionality.

The last three levels of the pyramid show the link between the improved model and the RPU, through a model adaptation in the input signals part of the model, where all the input and output signals of the model are present and a couple of .a2l files that make the address matching of the variables from ECU and RPU.

In terms of system implementation, the activities could be grouped into three major parts: hardware, software and prototype model adaptation implementation.

3.2 Hardware Implementation

In order to have the hardware implementation, it is necessary to adapt a serial engine control unit, having the role to communicate with the external rapid prototyping unit, AutoBox, through a hardware interface, DCI_GSI1 (Figure 3). These are both delivered by dSPACE. This is possible through a third-party connector tower, DS470, for the MPC561 microcontroller of the engine control unit.

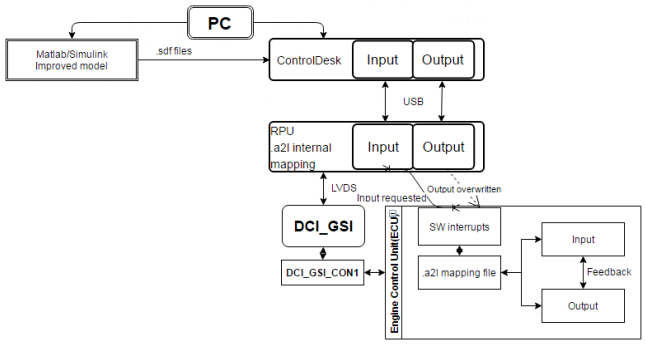


Figure 3
Data flow: from the Matlab/Simulink improved model, up to physical output

Nevertheless, an important problem remains. By using simultaneously the debugger hardware interface and DCI_GSI1, in order to flash code on the ECU, serious damage might be caused to the devices. This problem is present due to the fact that the DS470 connector has also dedicated pins for the debug interface of

the microcontroller and they are included in the auxiliary NEXUS port for communication with DCI_GSI1.

To solve this issue, a mapping of the associated pins of BDM debugger connector signals, debugging pins of the microcontroller and DS470 pins is required. The required mapping between the three hardware components is presented in Table 1.

Table 1
Required mapping between hardware components

BDM Connector Pin	BDM Signal	DS470 Pin	DS470 Signal
8	DSDI	15	MDI0
4	DSCK	17	MCKI
10	DSDO	19	MDO0
1	VFLSO	25	MSEO0
9	VREF	8	VREF

After a hard wired connection of the mapping described in the table above, both communication of the microcontroller with the DCI_GSI1 and the debug interface, were functional. A physical limitation is that it is not recommended to use both debug interface and DCI_GSI1 interface simultaneously because electrical problems could appear.

3.3 Software Implementation

In order to make the system compatible for external Rapid Prototyping, a simple software re-flash is not enough. Therefore, an interrupt mechanism which activates the External Rapid Prototyping unit, AutoBox, during each 10 ms recurrence, is triggered by adding the code presented in Figure 5. This must be done because on the AutoBox the code generated from the improved model will run and at each 10 ms new values of the targeted variables are delivered. These values are different from the ones delivered by the software flashed on the engine control unit. The interrupt mechanism is presented in Figure 4.

Also, at this level, after compiling the whole serial production project with the modifications mentioned above, an .a2l file is generated. This file is important because it also contains the memory mapping of all variables present in the software.

```
void interrupt_mechanism(void)
{
        #ifdef RPT_FUP_10MS  /* If the module has a RPT macros defined */
        C_RPT_SND_DATA(RPT_FUP_10MS
                /* Step 1. Load calculated valid signals from the Rapid Prototyping Unit,
                   from previous recurrence of the current module */
        #endif

        /* Step 2. Calculate signal fup_fil through the module logic */

        DPSWX(ip_fup_mes, fup_fil)
                /* Interpolate on the map axis/dependency with the fup_fil value from the previous
                recurrence */

        fup_fil = IP_1DW(ip_fup_mes)
                /* Define a map variable as a calibration vector or matrix */
                /* The new calculated value is obtained through an interpolation in the ip_fup_mes map*/

        #ifdef RPT_FUP_10MS /* If the module has a RPT macros defined */
        C_RPT_RCV_DATA (RPT_FUP_10MS)
                /* Step 3. The new calculated signal is received by the Rapid Prototyping Unit; at this level,
                the new value will be included in the calculation loop */
        #endif
}
```

Figure 4
The interrupt mechanism

3.4 Prototype Model Adaptation for External Rapid Prototyping

In order to make a better overview of the serial production software that is flashed on the engine control unit, this is divided into smaller modules. Each software module is developed based on a model developed in a plug-in tool, based on Matlab/Simulink. Each module of the software can be either manually coded or auto generated and introduced in the whole project.

A specific module that contains a control algorithm of the injection has been chosen and the according model adapted for External Rapid Prototyping. For this, in the model, there is a special section, Stimuli, where this adaptation is made. It contains dedicated blocks for mapping the memory addresses of the targeted variables between the engine control unit and the Rapid Prototyping unit.

Read blocks are dedicated blocks for model communication with DCI_GSI1. Those can be found in the Simulink library, at section "dSPACERTIBypassBlockset", named "RTIBYPASS_READ_BLx" (Figure 5). In the model created for the application, several blocks of this type were introduced because each block can handle a maximum of ten variables.

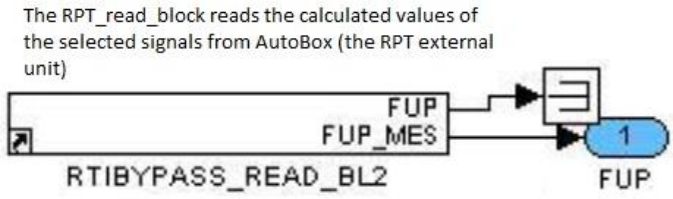


Figure 5
Read blocks for targeted variables

This is made through a couple of .a2l files. The first one is the .a2l file generated after building the project, as it was mentioned in the previous section. The second one is an .a2l file, used internally by the Rapid Prototyping Unit, which contains the addresses of the targeted variables from its own environment. By its own environment, the memory mapping of the microcontroller from the development boards can be understood, on which, runs the code generated from the improved model.

At run-time it is ensured that selected variables from the prototype model can be read, and even more, the new calculated values overwritten in real-time on the engine control unit, at the corresponding addresses, without being necessary to make a re-flash of the whole software. In order for these blocks to have the desired functionality, some specific settings must be made. The first thing to set is the interface used for bypass, in our case GSI. A second .a2l file is used which enables communication between ECU and Autobox through DCI_GSI1. This file basically realizes the communication port mapping, length of the messages exchanged between the equipment and other communication settings.

On the other hand, there are also the RTIBYPASS WRITE BLx blocks. These blocks write variables back to the ECU that are output variables of the calculated bypassed functions on the prototyping system. The variables to be written to the ECU should be selected, and specify the bypass interface and the service instance which should be used for writing the variables. The setup block must be placed within the SDA-eRPT RPT WRITE subsystem. These blocks offer the possibility to select the service instance of the ECU where the block has to write the variables. The names of the service instances are unique, and each service instance corresponds to one service ID. The Service Instance drop-down list contains the available service instances, determined by the selected imported database files (bypass .a2l file) for the selected bypass interface.

4 Results

The results of the proposed solution can be quantified in two directions. The main line of the study has been to understand how the new system affects the load on the microprocessor of the engine control unit. The second line was the economic one, more precisely which steps of the V cycle development process are eliminated by implementing the proposed method. For the first direction, five test cases were designed. For test case 0 an arbitrary number of variables to be read were chosen from the prototype system and decrease them sequentially up to test case 4. Figure 6 illustrates the results, more precisely, how the CPU load varies depending on the number of variables that are read and the engine speed. A maximum number of 76 signals were monitored. This is the exact number of input and output signals of the module that was chosen to be improved.

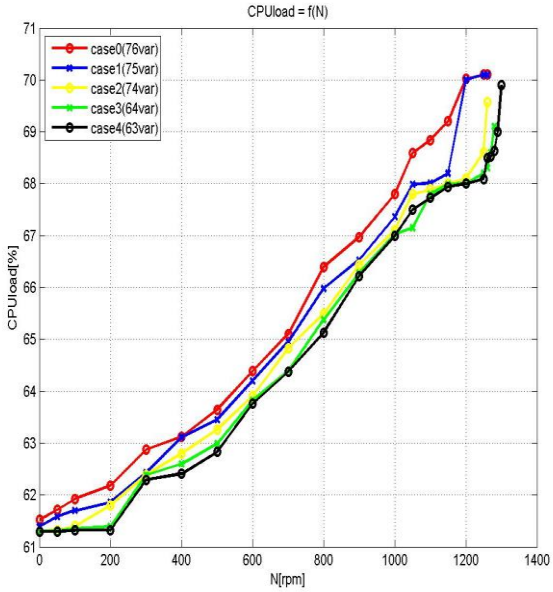


Figure 6
CPU load as a function of engine speed

One can notice the decrease of *CPUload* from one case to another, with the biggest load when the system reads 76 variables, and less charging in the final case. Aditionally, it can be remarked that the variation is about 0.4% *CPUload* with a change in the speed N of 100 units, during the same measurement, and 0.15-0.2% in a measurement to another for the same values of N. Another criterion was to measure if the time segment, which has a correlation with the engine speed, will be affected by the CPU load increase. This is important because

the relation between the engine speed and the segment time is an inversely proportional one, meaning a higher engine speed, a smaller time segment. All the logic contained by the functions called at segment time must be executed, so a decrease of segment time means an increase of CPU load. Adding the CPU load increase due to Rapid Prototyping system, the segment time could be affected.

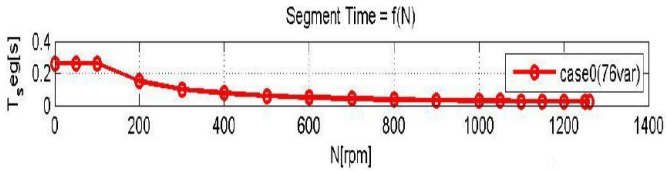


Figure 7

Segment time as a function of engine speed (case 1)

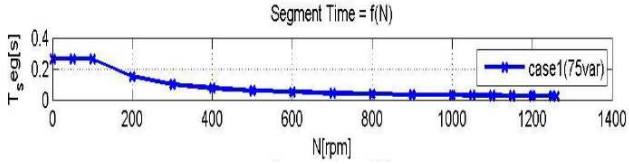


Figure 8

Segment time as a function of engine speed (case 2)

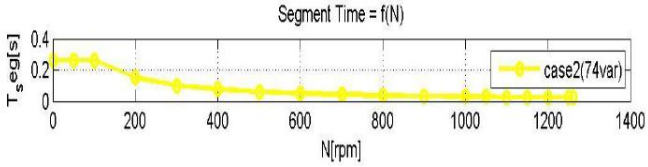


Figure 9

Segment time as a function of engine speed (case 3)

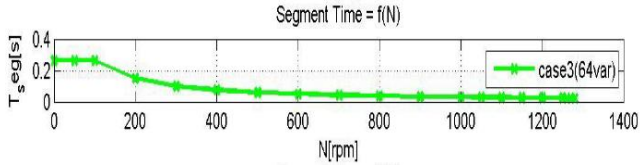


Figure 10

Segment time as a function of engine speed (case 4)

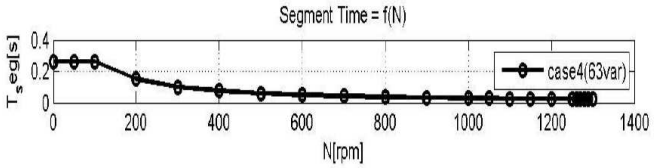


Figure 11
Segment time as a function of engine speed (case 5)

It can be noticed (Figure 7 – Figure 11) that the segment time is the same during all five measurements, given the fact that the Rapid Prototyping is negligible, the overall performance of the system is not affected.

On the other hand, the main output from the improved prototyped model is the fuel pressure in the common rail of the thermal motor, PFU. The pressure is obtained based on the value from the previous recurrence and is made at segment recurrence, which may last under 1 ms, at high engine speeds. This implies a signal acquisition at a very high rate. Figure 12 shows that the engine speed gradient is the same for all measurements, but still the amount of acquired data affects the acquisition.

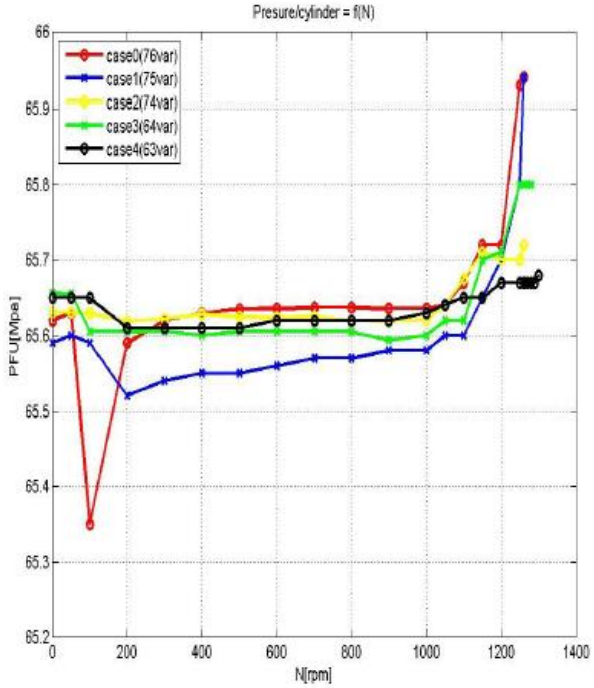


Figure 12
Fuel pressure as a function of engine speed

The first measurement in Figure 12 is quite unstable with high variations of PFU. In the next measurements, a decrease of the variables that are read through Rapid Prototyping system is operated, the logic of the model being maintained from a measurement to another. The smaller the number of signals acquired from the system, the more accurate the read of the PFU is made. To sum up, it is important to know what meaningful data must be read through the Rapid Prototyping system in order to have accurate data available.

The second direction of the results achieved is aimed at the economic improvement.

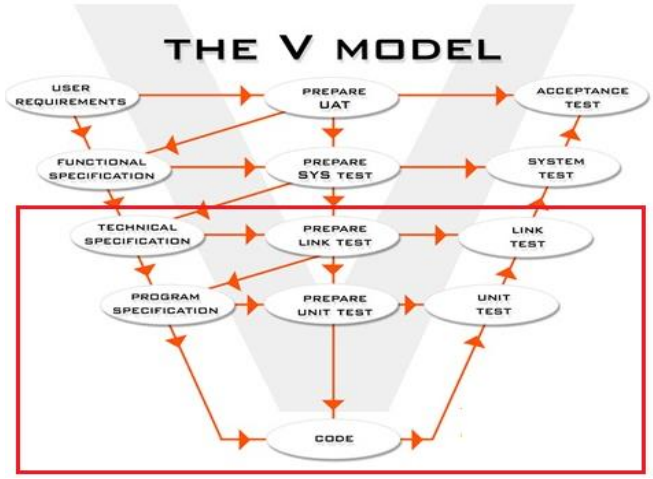


Figure 13
Eliminated steps from the developing process

Figure 13 describes a classic V cycle development, which is very common in the automotive field. Through the system that was developed, all the steps that are marked in the red section are mainly removed from the whole process. By this, there is a huge gain in terms of time and costs because the system allows a true validation [21] of the new concepts implemented in the prototype model because of the serial produced ECU, which is the main core piece of the system.

Conclusions

Considering that a classic development cycle on different functionalities from the engine control unit lasts between two and three years, by using the method proposed by this paper, this amount of time can be reduced significantly to only several months.

The novelty of the system produced for this paper is that it is the first time when the eRPT method has been applied on a serial production ECU, not on development or prototype ECUs.

According to the available state-of-the-art scientific literature, different automotive and industrial areas have harnessed this technique, yet it has not been previously applied on serial productions ECUs. The system reproduces 100% the final environment in which the new functionality will run, while the obtained results illustrate how the key parameters of the engine control unit will be affected by the Rapid Prototyping system. By this approach, the hardware validation is removed from the V-cycle, resulting in cost reduction from a time and money perspective. The proposed experiments show how each targeted key parameter have acceptable, even negligible deviations. In this manner a highly accurate validation of the developed prototype models is ensured. Furthermore, the prototypes can be developed in an evolutionary approach and the code corresponding to the final prototype model can be introduced directly in the final software product.

Acknowledgement

The implementation of the system underlying this paper has been facilitated by Continental Automotive Romania, and all the information presented in the paper has the approval of the company to be made public.

References

[1] F. Kordon, "An Introduction to Rapid System Prototyping", IEEE Transactions on Software Engineering, Vol. 28, Issue 9, 2002, pp. 817-821, DOI: 10.1109/TSE.2002.1033222

[2] B. O'Rourke, P. Giusto, T. Demmeler, S. Wisniewski, "Rapid Prototyping of Automotive Communication Protocols", in Proc. 12th International Workshop on Rapid System Prototyping, Monterey, USA, 2001, pp. 64-69, DOI: 10.1109/IWRSP.2001.933840

[3] K. Kowalczyk, H. J. Karkosch, P. M. Marienfeld, and F. Svaricek, "Rapid Control Prototyping of Active Vibration Control Systems in Automotive Applications", in Proc. IEEE Conference on Computer-Aided Control Systems Design, Munich, Germany, 2006, pp. 2677-2682, DOI: 10.1109/CACSD-CCA-ISIC.2006.4777062

[4] A. Holzinger, O. Waclik, F. Kappe, S. Lenhart, G. Orasche, B. Peischl, "Rapid Prototyping on the Example of Software Development in Automotive Industry: The Importance of their Provision for Software Projects at the Correct Time", Proc. International Conference on e-Business (ICE-B), Seville, Spain, 2011, pp. 1-5

[5] J. Chalupa, R. Grepl, V. Sova, "Design of Configurable DC Motor Power-Hardware-in-the-Loop Emulator for Electronic-Control-Unit Testing", Proc. 21st International Conference on Automation and Computing (ICAC), Glasgow, United Kingdom, 2015, pp. 1-6, DOI: 10.1109/IConAC.2015.7313987

[6] Samuel B. Kesner, Robert D. Howe, "Design Principles for Rapid Prototyping Forces Sensors Using 3-D Printing", IEEE/ASME Transactions on Mechatronics, Vol. 16, Issue 5, 2011, pp. 866-870, DOI: 10.1109/TMECH.2011.2160353

[7] P. L. Evans, A. Castellazzi, C. M. Johnson, "Design Tools for Rapid Multidomain Virtual Prototyping of Power Electronic Systems", IEEE Transactions On Power Electronics, Vol. 31, No. 3, 2016, pp. 2443-2455, DOI: 10.1109/TPEL.2015.2437793

[8] C. P. Kruger, A. M. Abu-Mahfouz, G. P. Hancke, "Rapid Prototyping of a Wireless Sensor Network Gateway for the Internet of Things Using Off-the-Shelf Components", Proc. IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, 2015, pp. 1926-1931, DOI: 10.1109/ICIT.2015.7125378

[9] S. Buso, T. Caldognetto, "Rapid Prototyping of Digital Controllers for Microgrid Inverters", IEEE Journal of Emerging and Selected Topics in Power Electronics, Vol. 3, Issue 2, 2015, pp. 440-450, DOI: 10.1109/JESTPE.2014.2327064

[10] K. Buchenrieder, "Rapid Prototyping of Embedded Hardware/Software Systems", Design Automation for Embedded Systems, Vol. 5, 2000, pp. 215-221

[11] C. Bieser, K. D. Mueller-Glaser, "Rapid Prototyping Design Acceleration Using a Novel Merging Methodology for Partial Configuration Streams of Xilinx Virtex-II FPGAs", Proc. 17th IEEE International Workshop on Rapid System Prototyping (RSP), Chania, Crete, 2006, pp. 193-199, DOI: 10.1109/RSP.2006.32

[12] R. Selvamuthukumaran, R. Gupta, "Rapid Prototyping of Power Electronics Converters for Photovoltaic System Application Using Xilinx System Generator", IET Power Electronics, Vol. 7, Issue 9, 2014, pp. 2269-2278, DOI: 10.1049/iet-pel.2013.0736

[13] B. D. Rouhani, E. M. Songhori, A. Mirhoseini, F. Koushanfar, "SSketch: An Automated Framework for Streaming Sketch-based Analysis of Big Data on FPGA", Proc. IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM) 2015, pp. 187-194, DOI: 10.1109/FCCM.2015.56

[14] S. Gadelovits, M. Sitbon, A. Kuperman, "Rapid Prototyping of a Low-Cost Solar Array Simulator Using an Off-the-Shelf DC Power Supply", IEEE Transactions on Power Electronics, Vol. 29, Issue 10, 2014, pp. 5278-5284, DOI: 10.1109/TPEL.2013.2291837

[15] A. D. Lantada, V. P. Klaus Plewa, N. Barie, M. Guttmann, M. Wissmann, "Toward Mass Production of Micro-Texturedmicro-Devices: Linking Rapid Prototyping with Microinjection Molding", The International Journal

of Advanced Manufacturing Technology, Vol. 76, Issue 5-8, 2015, pp. 1011-1020, http://dx.doi.org/10.1007/s00170-014-6333-2

[16] A.Tisan, J. Chin,"An End-User Platform for FPGA-based Design and Rapid Prototyping of Feedforward Artificial Neural Networks with On-Chip Backpropagation Learning", IEEE Transactions on Industrial Informatics, Vol. 12, Issue 3, 2016, pp. 1124-1133, DOI: 10.1109/TII.2016.2555936

[17] L. de Melo, M. T. da Silva, J. Martins, D. Newman, "A Microcontroller Platform for the Rapid Prototyping of Functional Electrical Stimulation-based Gait Neuroprostheses", Artificial Organs, Vol. 39, Issue 5, 2015, pp. 56-66, DOI: 10.1111/aor.12400

[18] A. Taddese, M. Beccani, E. Susilo, P. Volgyesi, A. Ledeczi, P. Valdastri, "Toward Rapid Prototyping of Miniature Capsule Robots", Proc. IEEE International Conference on Robotics and Automation (ICRA) 2015, pp. 4704 – 4709, DOI: 10.1109/ICRA.2015.7139852

[19] L. Simoni, M. Beschi, D. Colombo, A. Visioli, R. Adamini,"A Hardware-In-the-Loop Setup for Rapid Control Prototyping of Mechatronic Systems", in Proc. IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA) Luxembourg, 2015, pp. 1-4, DOI: 10.1109/ETFA.2015.7301628

[20] H. Vardhan, B. Akin, H. Jin,"A Low-Cost, High-Fidelity Processor-in-the Loop Platform: For Rapid Prototyping of Power Electronics Circuits and Motor Drives", IEEE Power Electronics Magazine, Vol. 3, Issue: 2, 2016, pp. 18-28, DOI: 10.1109/MPEL.2016.2550239

[21] S. Folea, R. De Keyser, I. R. Birs, C. I. Muresan, C. Ionescu, "Discrete-Time Implementation and Experimental Validation of a Fractional Order PD Controller for Vibration Suppression in Airplane Wings", Acta Polytehnica Hungarica, Vol. 14, Issue: 1, 2017, pp. 191-206, DOI: 10.12700/APH.14.1.2017.1.13

The Fitting Disc Method, a New Robust Algorithm of the Point Cloud Processing

Gábor Nagy[1], Tamás Jancsó[1], Chongchen Chen[2]

[1]Óbuda University, Alba Regia Technical Faculty, Institute of Geoinformatics
[2]Fuzhou University, Spatial Information Research Center
E-mail: nagy.gabor@amk.uni-obuda.hu

Abstract: This article presents a new robust LiDAR processing method. This method fits a regression plane to a point cloud in any horizontal position by fitting a disc (with R radius) on it, which contains a specified portion (q) of points under the disc plane in all three sectors of the disc. This method can be used to create digital elevation models even without any filtering process. This article also describes an analysis, which compares the results of the fitting disc method using different parameters in processing of digital elevation models.

Keywords: LiDAR; Point Cloud; Digital Elevation Model

1 Introduction

Laser scanning is a very efficient and developing technology of three-dimensional spatial data capturing. The processing of captured point clouds is a key element of these surveys. The point cloud is a typical case of big data: it contains a lot of information, and we have to assort the essential elements for our aim.

One of the most important results of the processing of airborne laser scanning (LiDAR) data is the Digital Elevation Model (DEM) of the surveyed region. The ground surface must be determined from the point cloud where the points generated by natural or artificial objects should not influence the result. There is a need for a new method that does not interfere with all points situated over the ground surface.

Usually this is achieved by methods that filter the point cloud before the surface is generated or filter the created surface. This article describes a new method (called fitting disc method) that can examine the elevation in any vertical position without filtering the point cloud. This method can be used to create digital elevation models or to recognize the terrain objects.

This article also contains an examination of the suggested method. The elevations derived from a LiDAR point cloud by the suggested method are compared to the result of surveying.

2 Solutions of the Creating Elevation Models from Point Clouds

Most of the developed methods solve this problem by using the major part of the points located over the ground surface, but the points located under the ground surface are generated only as measurement noise. The searching of the ground surface is a process of searching the lowest coherent surface.

2.1 Current Processing Methods

One group of the processing methods tries to filter the points out over the terrain surface, and uses the filtered point cloud to create a digital elevation model. The slope-based filter [14] excludes the points that result into a bigger slope than a specified angle as the maximum slope of the surface. The filtered point cloud does not contain any point pairs, where the slope between the points is more than the maximal slope. This filtering method removes the points that were created on the vegetation or buildings, if the slope resulted by these points was greater than the maximal slope. An advanced method (suggested by [12]) uses variable slope limit adapted to the terrain.

Another LiDAR data processing method filters the created surface to eliminate the effect of the non-ground points. For example, the morphological filter [3] or the methods suggested by [9].

The morphological filters are built from erosion and dilation operations, where the lower (erosion) or higher (dilation) elevation of the neighboring area (window) of the surface is assigned to an element of the elevation model. The size of this neighboring area is defined by a window. The combination of the erosion and dilation operation is called Dual Rank Filter.

The principle of the widely used Progressive Morphological Filter [17] is that the output surface of the morphological filtering is used for filtering the point cloud in the next step. The points near the surface are chosen in this filtering step. This filtered point cloud will be used in another step to create a new surface, with a simple morphological filter. These steps can be repeated several times in order to achieve the final result. The size of the window will be smaller and smaller in each step.

[2] suggests a fuzzy-based planar segmentation method for processing LiDAR data. [18] describes a method that uses cloth simulation (well-known tool in

computer graphics) for processing the ground surface: Put a cloth sheet over the upside-down point cloud, and the ground surface will be the surface of the sheet driven by the gravity and the collisions by the point of the cloud.

[13] compares different filtering methods. [10] presents details of several methods. [6] studies the possibility of the GPU-based acceleration of the LiDAR point cloud filtering methods.

2.2 Selecting the Lowest Point

The principle of our suggested method is that most of the points of a LiDAR point cloud are located higher or close to the ground surface. Or the assumption is that the theoretical surface representing the ground surface is located in such a position where only a small portion of the points remains under it. This robust method can determine the ground surface without filtering.

First, the lowest point of the selected section (for example, where the vertical distance from a position is lower than a radius) of the point cloud can be the central point of this part. The point, which is at a higher position than a certain proportion of points (for example 1 or 10 percent), can be used instead of the lowest point, which means the wrong points may be filtered this way.

If this operation is done in different positions, such as at the nodes of a DEM, the digital elevation model can be created. The elevation will be the lowest elevation of the chosen area that has an unfavorable effect on a sloping terrain.

2.3 Plane (Disc) Fitting

The disadvantage of the last method (selecting the lowest point) presented above is that it derives from the terrain with a horizontal plane, and it is not the true level. Therefore, the resulted elevation does not characterize the center of the area but, rather, the elevation of the border of the area (if the small portion of the points are under the horizontal plane). To overcome this, an oblique plane can be used instead of a horizontal plane.

The horizontal plane can be defined by unique parameter (the elevation of the plane), and this one parameter can be determined by one condition: defined portion of the points will be under the plane. The oblique plane needs three conditions because it has three parameters (for example, the elevation in a position and the slope in x and y direction). The area can be divided into three parts, and by applying three conditions to these three parts we can define a portion of points under the oblique plane.

The circle-shaped area (the disc) can be divided with three half-lines started from the center and, as a result, three sectors are created (Figure 1). The three parameters of the fitted plane may be the elevations of the centers of the sectors,

and these points are called control points (other parameters of the plane can be calculated from this data-set).

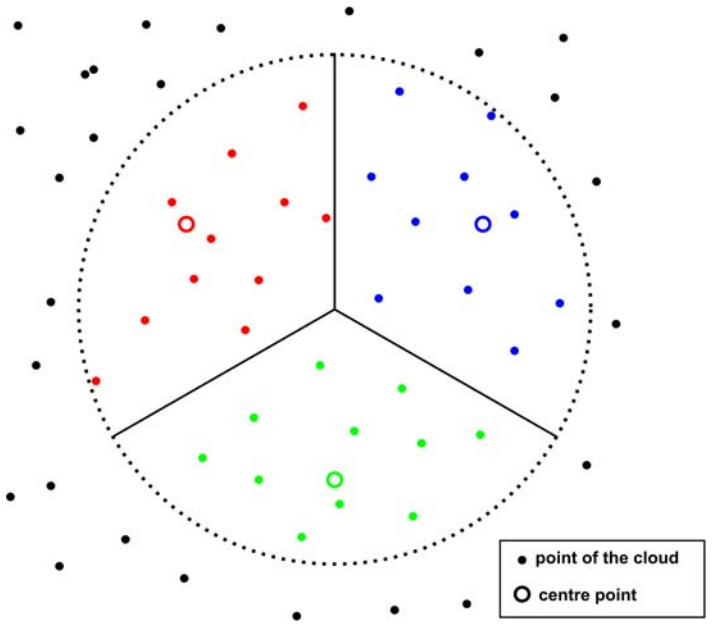


Figure 1

The points collected from the analyzed position (the area of the disc) are divided into three sectors. These points belonging to different sectors are marked by different colors. The center of each sector is represented by a small circle.

In the initial step of the plane fitting, the elevations of the sector centers will be determined as a portion (denoted by q, that means percent) of the points of this sector laying under this elevation. The q means quantile. Furthermore, the process is circulating around the sectors and it modifies the control point of the sector, according to the q portion of the points of the sector laying under the plane defined by this modified control point and the control points of the other two sectors. This step is repeated until all three sectors have q portion of the points under the plane without changing the elevation of the control points.

This method determines not only the elevation in a position (at the center of the disc), but also the slopes of the terrain are determined by the plane of the fitting disc. The result of the operation may be the elevation of the analyzed horizontal position and the slopes in x and y directions.

The principle of the method is demonstrated in the Figure 2, as a two-dimensional example.

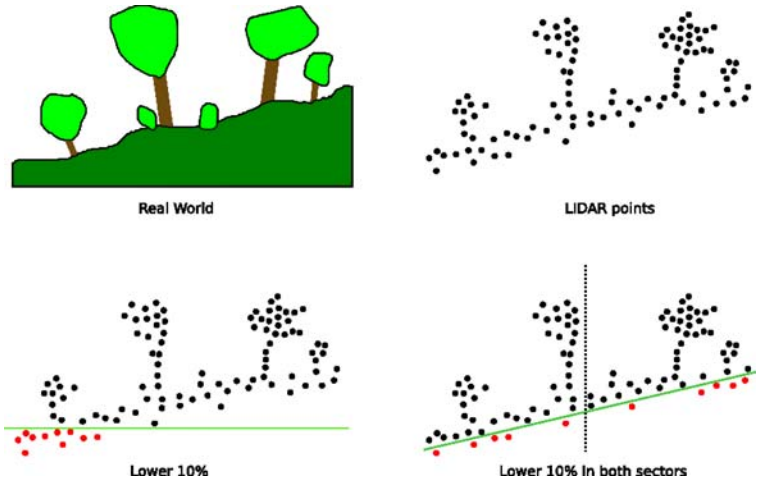


Figure 2

The principle of the method in two dimensions. The upper left picture symbolizes the real world, and the right upper picture is the LiDAR point cloud made from this terrain. In the left lower picture, a horizontal plane (here it is a line in 2D) is located over 10 percent of the points. The right lower picture shows the fitting disc method in two dimensions. The oblique plane (line) is located over 10 percent of the points in both sectors divided by a dashed line. The majority of the points are forming the trees and bushes, but the line fits to the ground surface.

2.4 Limitations

This method is based on the assumption that, as a good approximation, the terrain is flat (oblique) and close to the analyzed position. This is a better model than a terrain with a horizontal plane, but it is not too good if the terrain surface has significant curvature in the analyzed area. If the size of the area is small (the radius of the disc is small), the plane approximation will be good, but the points will form a sparse point cloud.

3 Implementation of the Proposed Method

The presented algorithm is implemented as a Python 3 application. One of the functions of the fitdisc module can read point cloud data from a simple text file to a point cloud object, and another function can determine the fitting disc to the point cloud at a specified horizontal position.

Inside the fitting disc function, the program uses a local coordinate system. The points are transformed because the origin of this local system is the analyzed point; the coordinates are scaled after the translation because the radius of the area

(the radius of the disc) in the local system equals 1. The equation of the transformation, where the uppercase letters (*X*, *Y*) are the original, while the lowercase letters (*x*, *y*) are the local coordinates:

$$\begin{aligned} x &= \frac{X - X_0}{R} \\ y &= \frac{Y - Y_0}{R} \\ z &= Z \end{aligned} \tag{1}$$

The X_0 and Y_0 are the coordinates of the analyzed horizontal position, and the R is the radius of the disc. Before applying this transformation, the program selects the points where the elevation is not changed. The program uses the original elevations for the plane fitting. Therefore, the elevation, as a result, is also calculated in the original elevation coordinate system.

The plane of the fitting disc is defined in several ways. One possibility is an equation in a local system:

$$z = z_0 + ax + by \tag{2}$$

The z_0 is the elevation of the plane in the origin (the center of the disc) of the local coordinate system, which matches the analyzed position; a and b are the slopes of the plane in the local system. The original slopes can be calculated, if these numbers are divided by R. If z_0, a and b are known, the elevation of the plane can be calculated in an arbitrary x, y position; for example, in the horizontal positions of the points of the point cloud, these elevations may be compared to the elevation of the point to decide whether the point is under or over the plane.

Another way to define the plane is when there are three points (that do not lie on a common line) of the plane, which are defined. These points may be the centers of the sectors. The horizontal position of these points is constant, so the plane can be defined by the elevations of these points, which are denoted by z_{w0}, z_{w1} and z_{w2}. These three values are more advantageous in the plane-fitting method because one of these values influences more the plane position in its own sector than in the other two sectors. These values are denoted z_{wi}, where i is the number of the sector, an integer value between zero and two.

The parameters of the (2) can be calculated from the elevations of the center points:

$$\begin{aligned} z_0 &= \frac{z_{w0} + z_{w1} + z_{w2}}{3} \\ a &= \frac{\sqrt{3}(z_{w2} - z_{w0})}{2} \\ b &= \frac{\sqrt{3}(z_{w1} - z_{w0})}{2} \end{aligned} \tag{3}$$

The reverse calculation is simpler, because only (2) is used in the centers of the areas:

$$z_{w0}=z_0-\frac{\sqrt{3}a}{3}-\frac{b}{3}$$
$$z_{w1}=z_0+\frac{2b}{3} \tag{4}$$
$$z_{w2}=z_0-\frac{\sqrt{3}a}{3}+\frac{b}{3}$$

In the initial step, the program determines the z_{w0}, z_{w1} and parameters. These values will be the elevations that are situated over the q portion of the points of the point cloud in the sectors around the center points.

In the next step, the z_{wi} value (i is initially zero, and in the further steps it will be *(i+1) mod 3*) must be modified, so that q portion of the points will be under the plane determined by z_{w0}, z_{w1} and z_{w2} in the sector number i. The $z_{w[i\pm1 \bmod 3]}$ values are not modified, and the points of sectors number *(i±1) mod 3* are not analyzed in this step. The new value of z_{wi} will be an integer multiplied with t value (the default value in the program is *0.01*). This step is repeated until $z_{\mathrm{wi}}$is not changed in three consecutive steps. The fitting disc is calculated, finally from the z_{w0}, z_{w1} and z_{w2} parameters. The z_0, a and b values can be calculated by (3). The z_0 is the elevation in the analyzed position, and slope can be calculated from a and b by the $\frac{a}{R}$ and $\frac{b}{R}$ formulas.

The program separates three (not two) categories when it analyzes the situation related to the plane. If a point is near the plane (closer than *1.6t*), this point will be ranked to the nearby category.

The t is the resolution of the elevation, the default $t=0.01$ means the centimeter resolution. (the z_{wi} values are integer multiple of t.)

The nearby category is necessary to avoid the infinite loop. The multiplication by *1.6* is needed for the points from the side of the disc, because the elevation of the disc's plane is changing about *1.6* times in this place rather than in the center of the sector. Other points will be ranked, obviously, to the under and to the over categories. The program counts the number of the points in under (n_{-1}), nearby (n_0) and over (n_{+1}) categories. The condition defined by a q value is true if the portion of the under category is less or equal than q and the portion of under and nearby points is greater or equal than q:

$$\frac{n_{-1}}{n_{-1}+n_0+n_{+1}}\leqslant q\leqslant\frac{n_{-1}+n_0}{n_{-1}+n_0+n_{+1}}=1-\frac{n_{+1}}{n_{-1}+n_0+n_{+1}} \tag{5}$$

If this condition is not true, the z_{wi} value must be increased (when portion of the under category is greater than q), or decreased (when portion of the under and nearby category is less than q). The initial value of the modification is t, and it will be doubled until the category is nearby or opposite to the initials. In the latter case,

the nearby position will be searched by the bisection method between the last two values.

The program can calculate the elevations in each node of a grid, and the created digital elevation model can be written to ArcInfo ASCII GRID format. The slopes may be calculated and exported similarly.

4 Examination of the Parameters

The fitting disc method needs two parameters. One of these is the radius of the disc (R). The other parameter is the q, the portion of the points that are under or nearby the plane in every sector.

Increasing the value of R the number of points is growing. More points provide better fitting and allow to use less q values, but the smaller details of the terrain cannot be evaluated, because the obtained value characterizes a greater area. The R value may be determined dynamically as a smaller radius around the analyzed position when every sector has at least N points from the processed point cloud.

Different q values make different surfaces. If the q is greater, the elevation is higher because more points must be located under the surface. The distances between the surfaces generated by different q values may be a feature of the point cloud. If the points of the cloud are diffused (for example in a wooden area), this value will be greater, and in other areas (fields or artificial surfaces) may be less.

5 Possible Applications of the Proposed Method

The presented method is used primarily to create digital elevation models from LiDAR data. The advantage of this method is that it can eliminate the effect of the points created around various natural or artificial objects with suitable parameters, if enough points were created in the ground surface.

The differences between the results are calculated with different parameters relating to the distribution of the points of the cloud near the analyzed position, and these values may be used similarly to multispectral data to interpret the land cover.

5.1 Creating DEM

The presented method can calculate an elevation from the point cloud in any horizontal position. These positions may be the points of a grid, and a digital elevation model can be created. Raster files can be created from other values of

this method, such as slope and aspect data. Figure 3 shows some DEM generated by the described algorithm from LiDAR data with different *R* and *q* parameters.

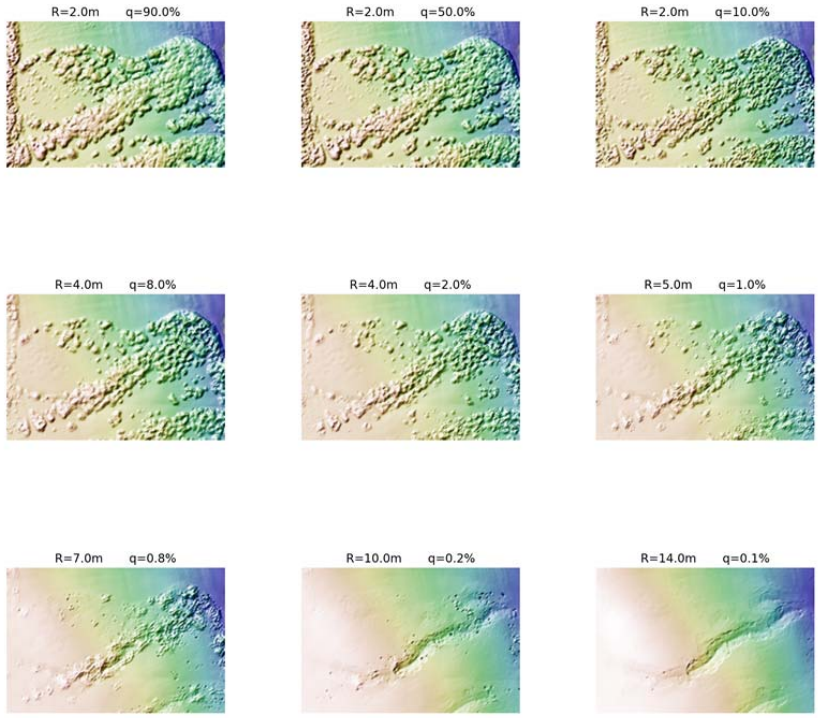


Figure 3.
Different elevation models created by different *q* parameters. The *R=10m*. The impact of the trees and bushes is less with lower *q* value.

TIN (Triangulates Irregular Network) models may be created by a method similar to method described in [5]. Another TIN generation method was described in [1].

5.2 Examination of the Ground Surface in Woodland Area

[11] and [8] analyzed the problem of the LiDAR data processing in woodland areas. [5] called this function as Virtual DeForestration (VDF). [16] described an additional method for processing LiDAR data in a woodland area.

The method presented in this article is suitable for this task with appropriate parameters: sufficiently low *q* and enough large *R* to the sectors has enough points (the points count is proportional to R^2). Of course, this method is not universal. If the vegetation is too dense, and points were not created in the ground surface, the

method cannot determine the elevation of the terrain (more correctly, a wrong elevation is calculated).

The presented method may be used for separation of the woodland areas. The different surfaces created by different q values may be useful for this work.

5.3 Recognizing Artificial Objects

In the urban areas, many points are created on the roofs of the buildings. The roof usually contains continuous and plane parts. The roofs are similar to the terrain surface, as points under these surfaces are not created because the laser beam cannot pass through them.

The presented method may be useful to recognize and evaluate roofs because it can provide a spatial position of a fitting disc, not only an elevation. The closer points may be grouped to a roof element, if the fitting discs match together. Another useful feature of this method is that this fitting disc process can be calculated in any vertical position. The edges of the roof elements may be interpreted smoother, if these advantages will be utilized.

The road surfaces can be recognized by a fitting disc that has a bit lesser R than the half of the width of the road. [7] and [15] demonstrated some their algorithm for road surface processing from LiDAR data.

6 Study of the Results

One LiDAR dataset was processed by different software applications to study the results. We made a surveying in the area of the LiDAR survey.

6.1 The LiDAR Survey

The LiDAR survey was made in May 2008, in the area of Iszkaszentgyörgy, close to Székesfehérvár, by the TELECOPTER Company. This survey project resulted in a true orthophoto from this area, which is used for samples for some figures of this article. The area covers six by four kilometers in E18.2280-18.3085 and N47.2130-47.2495. The flight altitude was 1400 meters (4593 ft).

The presented method, and the other processing software, use the last echo points. The maker guaranteed 0.15 meter as the maximum height error.

6.2 Surveying

One part of the LiDAR survey was surveyed by geodetic technology. This is about 25 hectares around the N47.2370 E18.2560 point. We surveyed 195 points in all types of vegetation (field, bushes and woods).

We used Leica TC 407 total station and Sokkia Stratus GNSS station in the survey. All of the points were surveyed by the total station, the GNSS observations were used to measure the positions of the station. We carried out the survey in August 2015.

The result of the surveying will be the reference in later analyzes. The processed LiDAR data will be compared to this surveying.

6.3 LiDAR Data Processing by Other Methods

The last echo LiDAR data was processed by GRASS [4]. As a first step, the point cloud was filtered by the v.outlier command. In the next step, we used the v.lidar.edgedetection command to search edges in the point cloud and, after that, we applied the v.lidar.growing and v.lidar.correction commands. In the final step, a surface was created by the v.surf.bspline command. We created another surface by the v.surf.bspline command from the original, unfiltered last echo point cloud.

The LiDAR data had been processed by the TopoSys software previously. This data was provided by the maker of the LiDAR survey. One of this data sets is the Digital Terrain Model (DTM) file, which was calculated from the last echo points. This file contains empty values where the ground surface was not determined reliably. In the FDTM (Filled DTM) file, these holes are filled in starting from the neighboring known values.

6.4 The Result of the Studies

The elevations were calculated in the horizontal positions of the surveyed points by bilinear interpolation from the data of the analyzed elevation models (denoted H_{DEM}), and these values were compared to the elevation of the surveying (denoted H_{GEOD}). We calculate the mean, the median and the

deviation of the difference (H_{DEM}-H_{GEOD}), and we also calculated the mean of the absolute value of the differences ($\frac{\sum|H_{\mathrm{DEM}}-H_{\mathrm{GEOD}}|}{n}$), and square mean of the differences ($\sqrt{\frac{\sum(H_{\mathrm{DEM}}-H_{\mathrm{GEOD}})^2}{n}}$).

A part of points (37 points) are in the holes of the TopoSys DTM data, the analysis can be done at the other 158 points. In the other models, the values can be calculated from all of the 195 surveyed points.

In the analysis of the fitting disc method, we calculated the elevations in the horizontal positions of the surveyed points (this method can calculate an elevation in any horizontal position), and these elevations are compared to the surveyed elevations. The analysis was done with more *R* and *q* parameter combinations.

In the higher q values, the mean of the differences is positive, and in the lower *q* values it changes into negative value. This ensures the principle of the method: when only few points must be under the plane, the plane is located lower. The trend of the medians of the differences is the same as it is with the mean values.

The standard deviations of the differences are lower with lower *q* parameters. The minimal value depends on the *R* parameter.

The trends of the mean of the absolute values of the differences and the square mean of the differences are similar to the standard deviations of the differences.

After trying different parameter combinations, the best value of the mean of the absolute values of the differences is 0.174 where *R=3.67* and *q=0.015*. It is better than the surface created by GRASS (0.904), and it is only less unfavorable than the FDTM surface of the TopoSys software (0.166) but the fitting disc method did not use filtering. The values of TopoSys FDTM and the fitting disc method are near to the error of the LiDAR survey (0.15 meter).

The differences are depending on the vegetation of the neighboring area with analyzed points. This is expressed by a value, which is calculated by the difference of elevations with *q=0.95* and *q=0.053* where *R=4m*, and this value may be called as the thickness of the point cloud.

The values of the Table 1. were calculated separately from the points where the thickness of the point cloud is thinner than 40 centimeters. The part thinner than 40 cm has 98 points, and the part thicker than 40 cm has 97 points. All of the 37 points, which are in the holes of the DTM data are located in the thicker area.

Table 1

The comparison of the elevation from fitting disc method with different R and q values and other software. The values were calculated in different areas depending on the thickness of the point cloud.

thickness	*method*	*GRASS*		*TopoSys*		*Fitting disc method*				
		original	*filtered*	*DTM*	*FDTM*	*R=2m*	*R=4m*		*R=5.19m*	*R=10.37*
						q=0.09	*q=0.016*		*q=0.014*	*q=0.0107*
	mean	0.058	0.060	0.042	0.055	0.005	−0.006	−0.036	−0.054	−0.117
	median	0.062	0.061	0.048	0.048	0.002	−0.007	−0.037	−0.047	−0.094
< 40 cm	**standard deviation**	0.079	0.070	0.158	0.070	0.055	0.056	0.061	0.080	0.145
	mean of abs. diff.	0.080	0.076	0.085	0.072	0.044	0.045	0.055	0.071	0.131
	sq.mean of diff	0.097	0.092	0.162	0.162	0.055	0.056	0.070	0.097	0.186
	mean	1.467	0.683	0.064	0.091	0.471	0.281	−0.007	−0.039	−0.117
	median	0.858	0.371	0.090	0.094	0.071	0.054	−0.003	−0.012	−0.062
> 40 cm	**standard deviation**	1.710	1.086	0.170	0.199	1.154	0.966	0.422	0.229	0.253
	mean of abs. diff.	1.507	0.718	0.126	0.138	0.517	0.366	0.164	0.116	0.151
	sq.mean of diff	2.247	1.278	0.181	0.218	1.240	1.002	0.420	0.231	0.277

The fitting disc method can provide better or, in some cases, only a bit worse result than other methods; it depends the values of the parameters (q and R). The method is especially good in the mean values.

The presented method can provide elevation data from the LiDAR point clouds of similar quality to the other analyzed methods with ideal parameters. The key issue is the correct adjustment of the R and q parameters, which are dependent on the characteristic of the terrain surface. The dense vegetation needs low q value and the low q value needs big R value for enough points, but the big R value is very good for minor details of the ground surface.

Conclusions and Future Work

The fitting disc method is an effective and robust solution for processing LiDAR point clouds. An elevation model can be made by this algorithm in any optional resolution because the fitting disc method can calculate an elevation in any horizontal position (not only the nodes of a raster grid). The method can work even without any filtering process because some wrong points do not influence the result.

The method can be adapted to the characteristic of the terrain by varying the parameters. The choice of the optimal parameters needs additional research. The choice is not a constant pair of R and q values; rather, it will be a method that can determine the parameters from the point cloud data.

The differences of the results, which were calculated from different parameters, may provide useful information of the surveyed ground surface (for example, the thickness, in the analysis, in the Table 1). These values can help to search for the optimal parameters to create a reliable elevation model.

The practical application needs more research for the parameters of the method, and combine the method with filter algorithms. The study of the method in other areas and other measurements (different instruments, flight altitude, etc.) is also required.

The principle of the fitting disc method may be used with polynomial surfaces. The area must be divided into many parts (similar to the sectors in the fitting disc method) as many parameters determine the surface. The parameters of the polynomial surface can be calculated from the elevations of the center points of the parts. The elevations of the center points may be modified in sequence to q portion of the points of the part lying under the surface, until this, criterion does not need modification in all of the parts. The fitting disc method is a special case of this general principle.

The fitting disc method may be regarded as a two-dimensional interpolation method, where the interpolated value is the elevation. The principle is applicable in higher dimension spaces as well. The two-dimensional space must be divided

into three sectors, the *N*-dimensional space must be divided into *N+1* parts because a linear function has *N+1* parameters in an *N*-dimensional space.

The test programs were made in Python 3 language because the ideas invented during the research were implemented easier this way. The final algorithms may be implemented in a low-level programming language (for example C or C++), and this program would be more effective. One important advantage of this method is an excellent parallel calculation because many fitting disc processes can be run at the same time.

Acknowledgement

In the initial state, the research was supported by the IGIT—Integrated geo-spatial information technology—and its application to resource and environmental management towards GEOSS covered by the FP7-PEOPLE-2009-IRSES, PIRSES-GA-2009-247608 project. The later research, and the making of this article, was supported by the project number TÁMOP-4.2.2.B-15/1/KONV-2015-0010, titled "Tudományos képzés műhelyeinek fejlesztése az Alba Regia Műszaki Karon" (in English: Developing workshops of the scientific education in the Alba Regia Technical Faculty).

References

[1] Axelsson P. (2000) DEM Generation from Laser Scanner Data Using Adaptive TIN Models. In International Archives of Photogrammetry and Remote Sensing, pp. 111-118, International Society for Photogrammetry & Remote Sensing

[2] Biosca J., Lerma J. (2008) Unsupervised Robust Planar Segmentation of Terrestrial Laser Scanner Point Clouds Based on Fuzzy Clustering Methods. In ISPRS Journal of Photogrammetry and Remote Sensing, 63:pp. 84-98, Elsevier

[3] Eckstein W. and Muenkelt O. (1995) Extracting Objects from Digital Terrain Models. In SPIE's 1995 International Symposium on Optical Science, Engineering, and Instrumentation, pp. 43-51, International Society for Optics and Photonics

[4] GRASS-Wiki. (2015) https://grasswiki.osgeo.org/wiki/LIDAR

[5] Haugerud R. A. and Harding D. J. (2001) Some Algorithms for Virtual Deforestation (vdf) of Lidar Topographic Survey Data. International archives of photogrammetry remote sensing and spatial information sciences, 34(3/W4):211-218

[6] Hu X., Li X., Zhang Y. (2013) Fast Filtering of LiDAR Point Cloud in Urban Areas Based on Scan Line Segmentation and GPU Acceleration. IEEE Geoscience and Remote Sensing Letters, 2:308-312

[7] Jaakkola A., Hyyppä J., Hyyppä H., Kukko A. (2008) Retrieval Algorithms for Road Surface Modelling Using Laser-based Mobile Mapping. Sensors, 9:5238-5249

[8] Kraus K. and Pfeifer N. (1998) Determination of Terrain Models in Wooded Areas with Airborne Laser Scanner Data. ISPRS Journal of Photogrammetry and Remote Sensing, 53(4):193-203

[9] Lohmann P., Koch A., and Schaeffer M. (2000) Approaches to the Filtering of Laser Scanner Data. International Archives of Photogrammetry and Remote Sensing, 33(B3/1; PART 3):540-547

[10] Meng X., Currit N., and Zhao K. (2010) Ground Filtering Algorithms for Airborne Lidar Data: A Review of Critical Issues. Remote Sensing, 2(3):833-860

[11] Pfeifer N., Reiter T., Briese C., and Rieger W. (1999) Interpolation of High Quality Ground Models from Laser Scanner Data in Forested Areas. International Archives of Photogrammetry and Remote Sensing, 32(3/W14):31-36

[12] Sithole G. (2001) Filtering of Laser Altimetry Data Using a Slope Adaptive Filter. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 34(3/W4):203-210

[13] Sithole G. and Vosselman G. (2004) Experimental Comparison of Filter Algorithms for Bare-Earth Extraction from Airborne Laser Scanning Point Clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 59(1-2):85-101

[14] Vosselman G. (2000) Slope-based Filtering of Laser Altimetry Data. International Archives of Photogrammetry and Remote Sensing, 33(B3/2; PART 3):935-942

[15] Wu B., Yu B., Huang C., Wu Q., Wu J. (2016) Automated Extraction of Ground Surface Along Urban Roads from Mobile Laser Scanning Point Clouds. Remote Sensing Letters, 7:170-179

[16] Wu Q., Lane C., Liu H. (2014) An Effective Method for Detecting Potential Woodland Vernal Pools Using High-Resolution LiDAR data and aerial imagery. Remote Sensing, 11:444-467

[17] Zhang K., Chen S.C., Whitman D., Shyu M. L., Yan J., and Zhang C. (2003) A Progressive Morphological Filter for Removing Nonground Measurements from Airborne Lidar Data. Geoscience and Remote Sensing, IEEE Transactions on, 41(4):872-882

[18] Zhang W., Qi J., Wan P., Wang H., Xie D., Wang X., Yan G. (2016) An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. Remote Sensing, 8:501

Extending JSON-LD Framing Capabilities

Kosa Nenadić[1], Milan Gavrić[2], Imre Lendak[2]

[1]Schneider Electric DMS NS LLC Novi Sad, Narodnog Fronta 25 a,b,c,d, 21000 Novi Sad, Serbia; kosa.nenadic@schneider-electric-dms.com

[2]Faculty of Technical Sciences, University of Novi Sad, Trg D. Obradovića 6, 21000 Novi Sad, Serbia; gavricm@uns.ac.rs, lendak@uns.ac.rs

Abstract: Today, with the increasing popularity of JSON-LD on the Web, there is a need for transformation and extraction of such structured data. In this paper, the authors propose extensions of the JSON-LD Framing specification which are able to create a tree layout based on recursive application of prioritized inverse relationships defined in a frame. The extensions include recursive application of reverse framing, a new @priority keyword which prioritizes reverse properties, a new embedding rule defined with the @first keyword, and the new @reverseRoots keyword used for filtering the result hierarchies of full-length. The proposed Extended Framing Algorithm, together with an extended frame, can be applied on arbitrary JSON-LD input files regardless of the length of its reverse hierarchy chains present in the frame. The proposed solution was tested on JSON-LD documents containing the ENTSO-E CIM Profiles. The two test scenarios were selected because of their complexity and size, each of them containing the ENTSO-E CIM Profiles expressed in CIM RDF Schema and OWL 2 Schema, respectively.

Keywords: Common Information Model; ENTSO-E; Framing; JSON-LD; RDF; Semantic Web

1 Introduction

The Semantic Web represents the Web of Linked Data. With the growth of the Semantic Web, the World Wide Web Consortium (W3C) promoted common data formats and exchange protocols, including the Resource Description Framework (RDF) family of specifications based on the RDF data model. JavaScript Object Notation (JSON) is considered the de-facto standard for data exchange over the Internet, mainly due to its simplicity for developers and its consumption in mobile and web applications [1]. Although JSON syntax is simple and clear there is no associated semantics. In contrast, JSON-LD (i.e. a JSON based serialization for Linked Data) adds meaning to JSON documents. A JSON-LD document is an instance of an RDF data model. The data model of a JSON-LD document represents a labeled, directed graph. A single directed graph can be serialized in

multiple ways and expressing the same information [2]. In practice, many complex models defined as graphs need to be presented in a hierarchical way to allow convenient access to particular data [3]. A hierarchy is commonly defined with a tree data model. For example, various ontology editors such as OntoStudio, TopBraid Composer, Protégé, Web Protégé, ontology browsers such as VectorBase, and ontology libraries such as OntoLink use indented tree visualization to present hierarchical structures associated with ontology entities [4], [5]. In paper [6] various ontology visualization methods were analyzed, including graph and tree structures, and a new approach for an ontology visualization and evaluation based on descriptive vectors is presented. As a JSON native data model is a tree [7], there is a need to transform a JSON-LD graph into a JSON tree that is capable of modeling cyclic structure. For that purpose, a mapping mechanism is defined in the form of JSON-LD Framing [8] – a draft specification published by the Linking Data in JSON Community Group with the aim to query and create specific tree layouts of JSON-LD documents. In this specification, a frame is applied on an input graph defined in a JSON-LD document using the JSON-LD Framing Algorithm, shaping the input document into a tree layout result that satisfies conditions specified in the frame. The JSON-LD Framing Algorithm complements the JSON-LD 1.0 Processing Algorithms and API [9], which defines a set of algorithms (namely, expansion, compaction, flattening and RDF serialization/deserialization) for programmatic transformations of JSON-LD documents.

There is often a need to express data relationships defined in a graph in a reverse direction. For example, a common practice when expressing a parent-child relationship between two entities in a class hierarchy definition is that the relationship is declared from a child to a parent. The aim of the @reverse keyword in JSON-LD is to express an inverse relationship using a reverse property [10].

Currently, the JSON-LD Framing specification supports a basic reverse framing, meaning that each inverse relationship has to be explicitly declared in a frame and its sub-frames in order to be present in the result. In this paper, the authors propose the introduction of three new framing keywords, namely @priority, @first and @reverseRoots, and present the usage of the following extensions of the JSON-LD Framing Algorithm:

- recursive application of reverse framing,
- prioritized inverse relationships using the @priority keyword in a reverse property,
- node object embedding using the @first keyword as a value of the object embed flag (i.e. @embed),
- node object filtering based on the value of the @reverseRoots keyword.

The ultimate goal of the proposed extensions is to define simple frames which result in desired tree layouts.

The paper is organized into sections. Section 2 presents related work, while Section 3 defines the problem. The proposed extensions are described in Section 4 together with the pseudo-code of the algorithm. Section 5 contains the description of the testing methodology and the overview of two test scenarios. The analysis of the input and output data, the results of performance testing and the discussion are presented in Section 6. Additionally, Section 6 demonstrates the application of resulting framed outputs in a custom developed JSON tree viewer for Web-based content visualization.

2 Related Works

The recent advances in the business intelligence applications field and knowledge representation is paired with Semantic Web technology improvements. For instance, numerous new versions of W3C Web Recommendations for RDF were created or updated in the last three years [11]. In the same direction were efforts to convert data from a JSON format to a semantically-enriched, RDF format containing Linked Data URIs [12]. Similarly, a framework for integration of various governmental services was developed using semantically enhanced service descriptions [13]. Nearly at the same time, the W3C created JSON-LD [10] as a lightweight syntax to serialize Linked Data in JSON. JSON-LD gained significant popularity and worldwide adoption when the major search engines (Google, Microsoft, Yahoo! and Yandex) created schema.org – a structured data markup schema – with the aim to enable search engines to understand the information embedded in web pages in order to serve richer search results [14], [15]. In reference [1] JSON-LD is recognized as a format that increases the utility and applications of the Smart City's data.

The JSON-LD Implementation Report [16] gives an overview of JSON-LD implementation statuses in various programming languages. Most of the available libraries that implement JSON-LD, and the JSON-LD Processing Algorithms and API, also support framing in its current state. As the JSON-LD Framing specification is a work in progress, version 1.0 lacks support for several features such as reverse framing, named graphs, re-embedding of the same data in the result [17], etc. Work was recently reactivated on the JSON-LD Framing specification 1.1 [8], focusing on changes regarding the node object embed options, support for the basic reverse framing, and framing of named graphs. The JSON-LD Framing 1.0 considers embedding of node objects as enabled (i.e. *true*) or disabled (i.e. *false*), but the latest version 1.1 also adds the following object embed flags @always, @never, @last and @link as the alternatives.

JSON-LD Framing is applied in specifications on the W3C recommendation track, such as Web Payments HTTP Messages 1.0 (uses JSON-LD Frame and JSON Schema for conformance check of JSON-LD objects of web payment messages

[18]) and Web Annotation Vocabulary (defines JSON-LD frames applicable to the graph of information that generate JSON output conforming to the serialization recommended by the Web Annotation Data Model [19]). In reference [20], a high performance cache for materialized graph views over large RDF graphs is created using JSON-LD Framing for denormalization of the materialized views into a tree data structure that is further indexed into a high performance tree indexing system. A similar approach is taken in [21] where JSON-LD Framing is used to represent RDF graphs of bibliographic data in JSON suitable for indexing with Elasticsearch.

In reference [22] the authors recognized the need for recursive prioritized inverse relationships in frames and developed an early implementation on top of node object filtering, as a basis of a web component which processes and displays machine readable CIM Profiles in a more human friendly form. This paper extends those results and addresses recursive application of prioritized reverse framing paired with node object filtering and introduces additional framing keywords to achieve more flexible results.

3 Problem Definition

In this section, we identify some deficiencies of the existing reverse framing solution, namely:

1. The inability to define a simple frame used for reverse framing.
2. The lack of multiple-relationship based reverse framing in a simple frame.
3. It is not possible to embed node objects on their first occurrence.
4. The lack of advanced filtering used for reverse framing.

3.1 Problem 1: Reverse Framing Complexity

Due to the fact that JSON-LD serializes directed graphs, each property (i.e. directed edge) points from one node to another node or value. Sometimes, it is necessary to express such a relationship in reverse direction. For instance, considering a *subClassOf* relationship, pointing from a subclass to a superclass, explicit definition of a property that designates the opposite direction – *superClassOf* relationship is typically omitted. Such a relationship can be expressed in a JSON-LD graph when the *@reverse subClassOf* property is defined. In this way, a reverse hierarchy is created.

When a long reverse hierarchy chain is needed, creating a desired framed output requires a deeply nested, complex frame, as the JSON-LD Framing specification currently supports only basic reverse framing. This means that a main frame and each embedded frame should define their own @reverse keyword section which may be inconvenient in situations when a creator of a frame is not aware of hierarchy depth in a JSON-LD input. A typical example represents the inheritance hierarchy (i.e. *rdfs:subClassOf*) in any RDF Schema (RDFS) vocabulary or Web Ontology Language (OWL) ontology.

An example of a JSON-LD frame that produces a tree hierarchy based on class inheritance and groups all related class properties based on a property domain is shown in Listing 1. It can be noticed that a subframe assigned to the *children* property must be repeated as many times as needed, depending on a JSON-LD document to be framed.

```
{
    "@context": {  ...
        "children": { "@reverse": "rdfs:subClassOf", "@container": "@set" },
        "properties": { "@reverse": "rdfs:domain", "@container": "@set" }
    },
    "@type": "rdfs:Class",
    "children": {
        "@type": "rdfs:Class",
        "properties": {
            "@type": "rdf:Property"
        },
        "children": {
            "@type": "rdfs:Class",
            "properties": {
                "@type": "rdf:Property"
            },
            "children": { ... }
        }
    }
}
```

Listing 1

Example of JSON-LD Frame Excerpt with Reverse Properties

There is certainly an option to create a custom suited reverse frame programmatically based on a document to be framed and the relationships of interest, but then each document requires its own frame and reverse framing becomes a two-step process. In the first step, a custom program creates a frame based on an input file and inverse relationships of interest, while in the second the frame is applied on the input file to produce a framed output.

3.2 Problem 2: Reverse Framing with Multiple Relationships

Frames with reverse properties become quite complex when additional properties are included or when there are multiple nested reverse properties of the same type creating corresponding hierarchies. For example, a frame with multiple nested reverse properties is shown in Listing 2. Assuming that family members work in the same company this frame can be used to create two hierarchies (company employee and parent-child hierarchies) starting from a person.

```
{
    "@context": { ...
        "employees": { "@reverse": "ex:employeeOf" },
        "children": { "@reverse": "ex:childOf" }
    },
    "@type": "Person",
    "employees": {
        "@type": "Person",
        "employees": {
            "@type": "Person",
            "employees": { ... },
            "children": { ... }
        },
        "children": {
            "@type": "Person",
            "employees": { ... },
            "children": { ... }
        }
    },
    "children": {
        "@type": "Person",
        "employees": { ... },
        "children": { ... },
    }
}
```

Listing 2

Example of JSON-LD Frame Excerpt with Multiple Nested Reverse Properties

Currently, there is no straightforward and standardized way for creating single-relationship or multiple-relationship based reverse tree hierarchies (thereafter referred to as tree hierarchies) of arbitrary depth starting from a flattened JSON-LD document and simple frame.

The order in which each inverse relationship is applied is important since JSON-LD framing uses the depth-first search algorithm when traversing related nodes to produce a framed output. By default, reverse properties are applied in order determined by the Expansion Algorithm, as an expanded frame is one of the inputs of the Framing Algorithm. Depending on how reverse properties are given in the original frame, whether using plain reverse properties or reverse properties with expanded term definitions or their combination, their order may vary in an expanded frame. The order in which inverse relationships are applied should be unambiguously determined and easily understood regardless of the way reverse properties are declared.

3.3 Problem 3: Embed on First Occurrence

Node objects are embedded on their last occurrence (i.e. the embedding rule ”@embed”: ”@last”) unless explicitly defined otherwise in a frame. Consequently, the result of reverse framing may not contain an explicitly expressed chain of full-length (i.e. longest hierarchy) if some element in the reverse chain appears later in the result. In order to overcome this deficiency, the embedding rule of @always could be applied, but then each referenced node and its subtree would be repeated in the result.

3.4 Problem 4: Advanced Filtering

The Framing Algorithm supports filtering based on strict-typing and duck-typing [17], [23]. The first type of filtering uses values of @type for matching. In its absence, filtering of nodes is based on matching included properties. Reverse framing is performed in conjunction with filtering, meaning that the framing process results in an array of node objects that satisfy filtering conditions and may also be roots of a reverse tree hierarchy if a node is related to some other nodes with reverse properties given in a frame. Some of these node objects may already be subtree roots in other reverse tree hierarchies, so there is a need to filter them out. In this way, a resulting framed output would contain only reverse tree hierarchies of full-length.

4 Solution

In order to address the identified problems, the authors propose extensions to the JSON-LD frame definition and the JSON-LD Framing Algorithm.

4.1 The Extended Frame

4.1.1 Recursive Application of Reverse Properties

By this proposal each reverse property is defined in the top-level frame of a JSON-LD frame. Each reverse property is applied recursively, meaning that they are all implicitly passed to any subframe (i.e. using the similar approach applied for object embed flags, the explicit inclusion flag and the require all flag). At the same time, a reverse property can be overridden in a subframe and as such passed to its subframes. In this way, deeply nested reverse properties are avoided in a frame.

4.1.2 Definition of Prioritized Reverse Properties

Problem 2 described in the previous section, regarding the order in which inverse relationships are applied, can be overcome if the order was explicitly declared. For this reason, a frame definition of a reverse property is extended with a new @priority framing keyword (e.g. *children* in Listing 3). The priority of a reverse property is defined with a number value of the @priority keyword (a lower value a higher priority). If the priority is not defined for a reverse property, a default priority is determined by the Expansion Algorithm. A priority does not determine the order in which relationships appear in a resulting tree layout because the layout is compacted using a context included in a frame.

4.1.3 Definition of a New Embedding Rule - First

The authors consider the embedding of node objects on their first occurrence in a JSON-LD graph as equally important to other alternatives. When used with recursive reverse framing it allows for the creation of tree hierarchies of full-length, while leaving node references to already traversed nodes. Therefore, a new value of the @embed framing keyword is proposed, defined with the @first object embed flag. Listing 3 illustrates how the new flag is used in a frame to globally define that node objects are embedded on their first occurrence.

4.1.4 Definition of Hierarchy Roots

In order to keep only the tree hierarchies of full-length in resulting framed outputs the authors introduce the @reverseRoots framing keyword (Listing 3). This keyword acts as a flag. The value of the @reverseRoots keyword is boolean. Setting its value to *true* enables filtering. If @reverseRoots is not specified, its value defaults to *false*.

```
{
    "@context" : {
        "ex" : "http://example.com/",
        "employees" : {"@reverse" : "ex:employeeOf"},
        "children" : {"@reverse" : "ex:childOf"}
    },
    "@type": "ex:Person",
    "@embed": "@first",
    "@reverseRoots": true,
    "employees": {"@priority" : 1},
    "children": {"@priority" : 2}
}
```

Listing 3
Example of Extended JSON-LD Frame

4.2 The Extended Framing Algorithm

The Extended Framing Algorithm (EFA) represents an extension of the Framing Algorithm [8] which supports the proposed, extended frame definition. In addition to the existing framing capabilities, it creates a tree hierarchy on each filtered node based on multiple prioritized reverse properties provided in an input frame.

The very process of creating prioritized reverse tree hierarchies in a JSON-LD tree layout can be split into two portions. The first portion of the EFA (Listing 4) accepts an expanded JSON-LD input file (i.e. *graph*) and expanded frame (i.e. *frame*) together with the global framing options. Essentially, this portion of the overall algorithm initializes the parameters used in the second, recursive portion. It includes:

- Initialization of the current state.
- Flattening of the input graph.
- Identification of relationships to be inverted from the input frame.
- Identification of graph nodes related with the identified relationships.

- Identification of all hierarchy roots and non-blank roots of each identified relationship based on the related nodes, and initialization of the current state.

Identification of non-blank roots is important as it is expected by this implementation that all blank (i.e. anonymous) nodes are embedded inside non-blank (i.e. named) nodes in a created hierarchy. For this reason, based on the identified hierarchy roots, if a root is a blank node, then its hierarchy is searched downstream for the nearest appearance of non-blank descendants which become new hierarchy roots.

```
function Frame(graph, frame, options)
        state       = CreateState(options)
        fGraph      = Flatten(graph)
        revRels     = GetReverseRelationships(frame)
        forEach node in fGraph do
                forEach revRel in revRels do
                        if node has revRel then
                                add node.id into revRel.domain
                                add node.revRel.value into revRel.range
        forEach revRel in revRels do
                forEach id in revRel.range do
                        if not id exists in revRel.domain then
                                add id into revRel.roots
        forEach revRel in revRels do
                forEach id in revRel.roots do
                        if IsBlank(id) then
                                descs = FindNearestNBDescs(id, revRel)
                                add descs into revRel.nonBlankRoots
                        else
                                add id into revRel.nonBlankRoots
        state.revRels       = revRels
        state.subjects      = fGraph.nodes
    return RecurFrame(state, IDs(fGraph.nodes), frame, false, undefined)
end function
```

Listing 4

Extended Frame Algorithm – First Portion

The second, recursive portion of the EFA (pseudo-code in Listing 5) includes:

- Flag initialization using the current frame and state – flags ensure that a property (namely *embed, explicit, requireAll, reverse* and *reverseRoots*) is passed from the current frame to a subframe if the subframe does not override it;
- Filtering of subjects that satisfy the current frame and flags (i.e. matches);
- Prioritization of matches using the identified non-blank roots – meaning that non-blank root matches have precedence over the rest of the matches that are sorted ascending by their ids;
- Each match is processed in the following way:
    - A match is skipped if it is a top-level node that is already traversed in another reverse hierarchy and only the hierarchies of full-length are of interest;
    - Depending on the current *embed* value and state, the way in which the match is referenced in the output is determined or the framing process is continued;

- Based on the content of the current frame, inverse relationships are identified, ordered by their priorities and used to build a tree hierarchy with the match as its root. For each relationship, the match's related nodes are prioritized and traversed recursively with the appropriate subframe, taking care that the related node is skipped if it is already traversed and hierarchies of full-length are of interest. The match and related node are marked as traversed when they are recursively processed;
- The match's own properties are processed;
- Default properties, defined in the current frame, are processed;
- The output is set as a value of the current *parent's property*;
- If the recursive framing of a top-level node is completed and hierarchies of full-length are of interest, then all traversed nodes are globally stored to be checked when a new top-level node is processed.

```
function RecurFrame(state, subjects, frame, parent, property)
        flags      = GetFlags(frame, state)
        matches    = FilterSubjects(state, subjects, frame, flags)
        matches    = PrioritizeNonBlankRoots(matches, state, frame)

        forEach match in matches do
                if property == undefined and flags.reverseRoots and
                match in state.traversedAll then
                        continue

                output = create(match)
                if ProcessEmbedValues(flags.embed, state, output) then
                        continue

                revRels = GetReverseRelationships(frame)
                revRels = OrderByPriority(revRels)

                forEach revRel in revRels do
                        rs = GetRelated(id, revRel)
                        rs = PrioritizeNonBlankRoots(rs, state, frame)
                        implicitFrame      = CreateImplicitFrame(flags)
                        subframe   = GetSubframe(frame, revRel)
                        subframe   = MergeFrames(implicitFrame, subframe)

                        forEach r in rs do
                                if subframe.reverseRoots and
                                r in state.traversed then
                                        continue

                                RecurFrame(state, r, subframe, output.reverse, revRel)

                                if not id in state.traversed then
                                        add id into state.traversed
                                if not r in state.traversed then
                                        add r into state.traversed
                ProcessOwnProperties(match, flags, frame, output)
                ProcessDefaultProperties(frame, output)
                AddFrameOutput(parent, property, output)

                if property == undefined and flags.reverseRoots then
                        add state.traversed into state.traversedAll
end function
```

Listing 5

Recursive Portion of the Extended Frame Algorithm

5 Testing Methodology

Initial testing was conducted against the set of created new reverse API tests included in the JSON-LD Test Suite provided with the implementation of the Extended Framing Algorithm [24]. These tests basically validate a framed output against the expected output for a given input and frame.

For the detailed testing, the authors searched for convenient data sources that are sufficiently large and complex to evaluate the proposed extensions providing at the same time verifiable results. The CIM Profiles which are part of the CGMES defined by the European Network Transmission System Operators for Electricity [14] (ENTSO-E) were chosen as the testing data source. These profiles were used for the 5th interoperability tests conducted by the European Transmission System Operators (TSO) in 2014.

In order to clarify the connections between input and output data, the following terms are defined:

- CIM Profile – a subset of CIM classes, properties and associations including CIM extensions. It may be defined using the CIM RDF Schema [25].
- CIM RDF Schema – an IEC standard, which relies on the subset of RDF classes and properties and set of CIM RDF Schema extensions [25].
- RDF/XML – an XML syntax for RDF graphs.
- CIMXML model exchange format – an IEC standard, defines a CIM Profile serialization using the RDF/XML [26].

CIMXMLs of the ENTSO-E CIM Profiles were used as a starting data source in two test scenarios. In the first test scenario, the profiles were transformed into JSON-LD syntax and used as a testing input. As a CIM Profile does not contain blank nodes related with RDFS properties, it was decided to conduct additional testing using the representation of CIM Profiles in a more expressive OWL 2 (the latest version of OWL). For this reason, the profiles were mapped into the OWL 2 representation in RDF/XML syntax, transformed into JSON-LD syntax afterwards, and as such used as a testing input in the second test scenario. In both test scenarios, the same input frame is applied to create a CIM Profile tree hierarchy.

5.1 The RDFS Test Scenario

In this scenario, the CIMXML files containing the RDFS representation of CIM Profiles were used as a starting data source. Those files were converted into JSON-LD syntax since both RDF/XML and JSON-LD are capable to serialize an RDF graph. The translation was done using the RDF Translator [27]. The frame

shown in Listing 6 was used together with a translated CIM Profile as an input to the Extended Framing Algorithm.

5.2 The OWL 2 Test Scenario

Based on the authors' previous experiences (an analysis of CIM Profile conversion into OWL was presented in reference [28]), a custom converter was implemented in order to transform CIMXML of CIM Profiles into the OWL 2 format. The conversion was accomplished in the following steps:

- RDFS class and property constructs were transformed into corresponding OWL 2 class and property constructs (i.e. *rdfs:Class* into *owl:Class*; *rdfs:Property* into *owl:DatatypeProperty* or *owl:ObjectProperty* depending on a relation designated by *rdfs:Property*).
- The RDFS extensions (i.e. constructs that share *cims* namespace) were transformed into corresponding OWL 2 constructs where possible. *cims:multiplicity* was replaced with OWL object and data property restrictions, *cims:inverseRoleName* was mapped to *owl:inverseOf*, and *cims:dataType* was replaced with *rdfs:range* of an *owl:DatatypeProperty*.
- The rest of the RDFS extensions (namely, *cims:AssociationUsed, cims:stereotype, cims:isFixed, cims:ClassCategory* and *cims:belongsToCategory*) were preserved as meta data of defined classes and properties.
- Classes that model primitive datatypes, such as String, Date, Integer, etc., were skipped and corresponding data types from XML Schema Definition (XSD) namespace were used instead.

In addition to the subset of RDF properties applied in CIM RDFS, the authors used *rdfs:isDefinedBy* property to designate that each defined *owl:Class*, *owl:DatatypeProperty* and *owl:ObjectProperty* is defined by the created *owl:Ontology*. In this way, one more hierarchical level was created in the CIM profile ontology compared to the corresponding profile RDF Schema.

The created CIM Profiles in OWL2 form were validated in Protégé ontology editor (Figure 1). JSON-LD serialization of a CIM Profile is used as an input in the Extended Framing Algorithm together with the frame shown in Listing 6 (see 5.3).

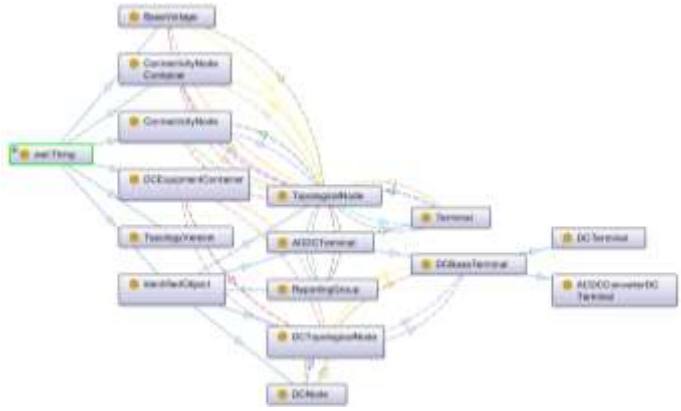


Figure 1
OntoGraph Vizualization of Topology Profile Ontology in Protégé

5.3 Test Frame

The input frame (Listing 6) shapes the initially provided JSON-LD document into hierarchy trees starting from an ontology or class, groups related classes and properties, embeds subclasses based on their inheritance relationship, groups all properties that belong to a class. It ensures that a hierarchy tree is not a subtree of another tree that only explicitly declared properties are included in the output and that node objects are embedded when they are first encountered. The same resulting framed output can be achieved by creating a simpler frame in each test scenario. For instance, in the RDFS test scenario OWL constructs and inverse *rdfs:isDefinedBy* property can be avoided in the frame. However, the authors wanted to keep the same input frame not affecting the framing process. At the same time, the results of such framing served as a confirmation of properly implemented profile conversion.

```
{
    "@context": {
        // owl, rdf, rdfs, xsd, cim, entsoe, tp, ...
        "children": { "@reverse": "rdfs:subClassOf", "@container": "@set"},
        "properties": { "@reverse": "rdfs:domain", "@container": "@set"},
        "defines": { "@reverse": "rdfs:isDefinedBy", "@container": "@set"}
    },
    "@type": ["owl:Ontology", "rdfs:Class", "owl:Class"],
    "@embed": "@first",
    "@reverseRoots": true,
    "@explicit": true,
    "defines": {
        "@priority": 1,
        "@type": ["owl:Class", "owl:DatatypeProperty", "owl:ObjectProperty"]},
    "children": {
        "@priority": 2,
        "@type": ["rdfs:Class", "owl:Class"]},
    "properties": {
        "@priority": 3,
        "@type": ["rdf:Property", "owl:ObjectProperty", "owl:DatatypeProperty"]}
}
```

Listing 6
Frame for CIM Profiles

6 Results and Discussion

The performance testing of the Extended Framing Algorithm implementation based on forked version of jsonld.js, available at [24], was conducted on a computer with an Intel Core i5-4300M/2.60 GHz CPU with 16 GB of RAM and 500 GB HDD running Microsoft Windows 8.1 Enterprise (64-bit) with Node v6.8.1. The testing was done in two scenarios, where 1000 iterations of the framing were executed for each file, and an average time was calculated.

Tables 1-2 present model metrics of the input and output data, and average framing times in RDFS and OWL 2 test scenarios, respectively.

Table 1
Data ModelMetrics per ENTSO-E CIM Profile document in RDFS

Profiles	Input				Output				
	#Classes	#Properties	#Triples	Size [bytes]	#Triples	Size [bytes]	#Hierarchy trees	Longest hierarchy [length]	Average framing time [ms]
GeographicalLocation	10	23	240	25332	58	5015	6	2	6.981
TopologyBoundary	10	28	284	32533	67	5680	7	2	8.075
Topology	18	31	355	36790	88	7085	8	3	10.182
DiagramLayout	21	46	585	58819	118	8789	14	3	15.769
EquipmentBoundary	25	39	567	59392	116	9517	10	5	15.583
StateVariablesProfile	33	63	770	74973	166	11775	24	3	21.362
SteadyStateHypothesis	75	84	1232	125892	292	24728	24	7	35.453
EquipmentProfileCore	177	412	4483	444497	1107	85889	69	7	171.210
EquipmentProfileCore-ShortCircuit	183	399	4417	448550	1093	86127	69	7	154.307
EquipmentProfileCore-Operation	222	417	4799	480497	1207	90106	69	7	187.640
EquipmentProfileCore-ShortCircuitOperation	226	624	6451	633842	1629	126640	69	7	281.026
Dynamics	252	2802	21655	2057966	6067	440244	39	7	1233.987

In order to have better understanding of results, input and output files of both test scenarios are compared and analyzed. The OWL 2 representation of profiles has a slightly smaller number of defined classes due to usage of XSD primitive data types when compared with RDFS representation, while the number of properties is the same. A conversion of CIM Profile representation in RDFS into OWL 2 had a significant impact on number of triples in OWL 2 profiles as some RDFS extensions are represented with several triples in OWL 2 (e.g. *cims:multiplicity* into OWL 2 restrictions). The number of triples in OWL 2 representation is increased for ~48.9% on average in comparison with the RDFS representation, while the size of the input file is increased for ~24.9% on average.

Table 2
Data Model Metrics per ENTSO-E CIM Profile document in OWL 2

Profiles	Input				Output				
	#Classes	#Properties	#Triples	Size [bytes]	#Triples	Size [bytes]	#Hierarchy trees	Longest hierarchy [length]	Average framing time [ms]
GeographicalLocation	7	23	334	30019	59	5635	1	3	8.854
TopologyBoundary	7	28	404	36236	69	6389	1	3	10.512
Topology	16	31	500	43085	93	8100	1	4	13.612
DiagramLayout	16	46	867	72747	123	9987	1	4	22.829
EquipmentBoundary	20	39	821	69265	119	10738	1	6	22.477
StateVariablesProfile	29	63	1158	95947	183	13744	1	4	32.877
SteadyStateHypothesis	70	84	1795	155311	307	28066	1	8	54.195
EquipmentProfileCore	169	412	6830	574211	1161	98731	1	8	311.856
EquipmentProfileCore-ShortCircuit	175	399	6745	584335	1143	99826	1	8	273.016
EquipmentProfileCore-Operation	214	417	7235	617918	1261	104570	1	8	343.537
EquipmentProfileCore-ShortCircuitOperation	218	624	9882	854579	1683	148120	1	8	522.892
Dynamics	247	2802	36035	2833795	6097	509551	1	8	2441.795

The number of output triples in OWL 2 representation is increased for ~4.2% on average when compared with its counterpart in RDFS representation, while the size of an output file is increased for ~14.6% on average. This is the consequence of the introduced *rdfs:isDefinedBy* property and conversion of *rdfs:Property* into *owl:DatatypeProperty* and *owl:ObjectProperty*. The number of hierarchy trees in each OWL 2 output is one, as there is a single ontology root object that defines all class and property node objects in contrast to the corresponding RDFS output in which each hierarchy tree has a class as a root object. This is illustrated in Figure 2 in which the JSON Tree Viewer web component displays the frames of the Topology Profile in RDFS and OWL 2 respectively. At the same time, this is a good example how JSON-LD Framing is used to shape input data to ease further JSON to DOM rendering. The length of the longest hierarchy tree in a framed OWL 2 profile was one level longer than the length of the corresponding RDFS counterpart.

As for the average framing time (later referred to as framing time), Figure 3 illustrates how it relates to other measured values. Figure 3 (a) shows the linear dependence of framing time (presented in a logarithmic scale) with respect to the number of ontology, class and property triples in both test scenarios. Framing time was slightly longer in OWL 2 case. The reason for this was the greater number of other types of triples in input files which is confirmed with results shown in Figure 3 (b). Also, Figure 3 (b) shows that the framing time advances with a linear dependence of the number of input triples in both test scenarios. Similarly, Figure 3 (c) shows the linear dependence of framing time with respect to input file size in

both test scenarios. As for the number of output triples in both scenarios, shown in Figure 3 (d), it is nearly the same since RDFS output contains some triples that were related to defined primitive data types which were removed from OWL 2 input, while in the OWL 2 input a *rdfs:isDefinedBy* relationship was introduced and preserved in the output triples (Figure 2). It should be noted that there is a linear dependence between data shown in Figure 3 (a) and Figure 3 (d) as the output contains ontology, class, property triples together with reverse property triples. The linear dependence was present between output file sizes and framing time in both test scenarios as well Figure 3 (e), which is the consequence of the size of input files, Figure 3 (c).

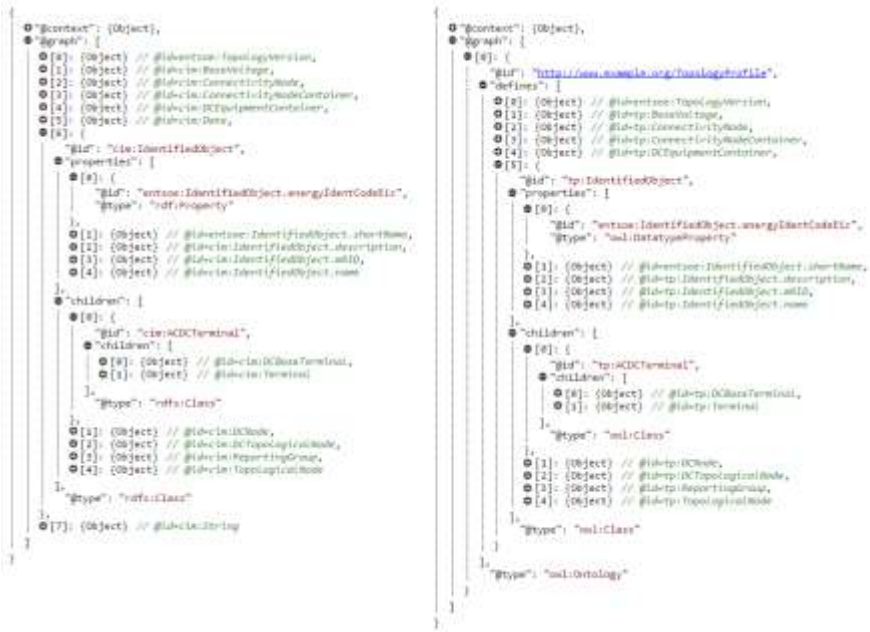


Figure 2
JSON Tree View of TopologyProfile in RDFS (left) and OWL 2 (right)

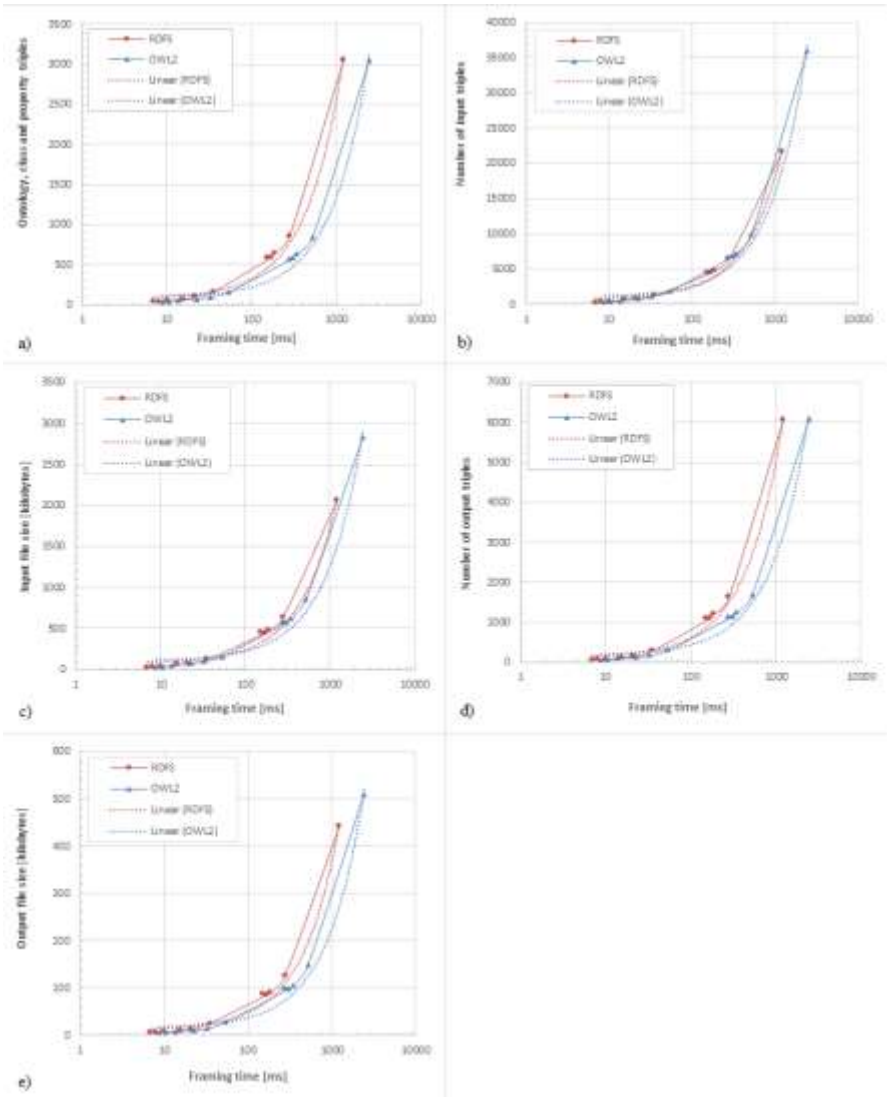


Figure 3
(a) Number of Input Ontology, Class and Property Triples vs. Framing Time,
(b) Number of Input Triples vs. Framing Time, (c) Input File Size vs. Framing Time,
(d) Number of Output Triples vs. Framing Time, (e) Output File Size vs. Framing Time

Conclusion

This paper extends the existing JSON-LD Framing specification with recursive prioritized reverse framing. Defined extensions allow definition of a frame which can be applied on an arbitrary number of input files regardless of the length of a reverse hierarchy chain of the given inverse relationship from the frame. Otherwise, a custom suited frame must be created for each input file. It also

allows combination of multiple inverse relationships in reverse tree hierarchies based on defined priorities, which the authors find suitable for grouping of related nodes using different inverse relationships. The set of existing embedding rules is extended with a new rule which enables embedding of node objects on their first occurrence. This rule, when combined with recursive reverse framing, enables creation of reverse tree hierarchies of full-length. Additionally, one of the proposed extensions enables filtering of such reverse tree hierarchies.

The proposed Extended JSON-LD Framing Algorithm is designed and implemented, and results of its application on a set of complex RDFS vocabularies and OWL 2 ontologies, using a single frame, are analyzed showing overall linear dependence of the number of input triples with respect to framing time when multiple inverse relationships are defined in a frame. The framing applied in the test scenarios shows how an arbitrary ontology can be transformed into a tree and used as input for other processes as well as how framing can be used in the validation of properly implemented RDFS into OWL 2 conversions on the examples of ENTSO-E CIM Profiles.

The future work is intended towards research of more advanced property value filtering (e.g. a property value equal to, less than, greater than some value) that could be used in conjunction with recursive prioritized reverse framing.

Acknowledgment

We would like to thank to our colleagues from Schneider Electric DMS NS LLC Novi Sad, Serbia for their support.

References

[1] U. Aguilera, O. Peña, O. Belmonte, and D. López-de-Ipiña, "Citizen-centric data services for smarter cities," *Futur. Gener. Comput. Syst.*, pp. 1-14, 2016.

[2] M. Lanthaler and C. Gütl, "Model Your Application Domain, Not Your JSON Structures," in *Proceedings of the 4th International Workshop on RESTful Design WSREST 2013 at the 22nd International World Wide Web Conference WWW2013*, 2013, pp. 1415-1420.

[3] M. Lanzenberger, J. Sampson, and M. Rester, "Ontology Visualization: Tools and Techniques for Visual Representation of Semi-Structured Meta-Data," *J. Univers. Comput. Sci.*, Vol. 16, No. 7, pp. 1036-1054, 2010.

[4] B. Fu, N. F. Noy, and M.-A. Storey, "Indented Tree or Graph? A Usability Study of Ontology Visualization Techniques in the Context of Class Mapping Evaluation," in *Proceedings of the 12th International Semantic Web Conference - Part I*, 2013, pp. 117-134.

[5] E. S. Alatrish, "Comparison of Ontology Editors," *e-RAF J. Comput.*, Vol. 4, pp. 23–38, 2012.

[6] K. Machová, J. Vrana, M. Mach, and P. Sinčák, "Ontology evaluation based on the visualization methods, context and summaries," *Acta Polytech. Hungarica*, Vol. 13, No. 4, pp. 53-76, 2016.

[7] M. Lanthaler and C. Gütl, "On using JSON-LD to create evolvable RESTful services," in *Third International Workshop on RESTful Design*, 2012, No. April, pp. 25-32.

[8] M. Sporny, G. Kellogg, D. Longley, and M. Lanthaler, "JSON-LD Framing 1.1 An Application Programming Interface for the JSON-LD Syntax," *Draft Community Group Report 04 October 2016*, 2016. [Online]. Available: http://json-ld.org/spec/latest/json-ld-framing/. [Accessed: 06-Oct-2016].

[9] D. Longley, G. Kellogg, M. Lanthaler, and M. Sporny, "JSON-LD 1.0 Processing Algorithms and API, W3C Recommendation 16 January 2014," *W3C*, 2014. [Online]. Available: https://www.w3.org/TR/json-ld-api/. [Accessed: 24-Feb-2016].

[10] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, and N. Lindström, "JSON-LD 1.0, A JSON-based Serialization for Linked Data, W3C Recommendation 16 January 2014," *W3C*, 2014. [Online]. Available: https://www.w3.org/TR/json-ld/. [Accessed: 26-Feb-2016].

[11] W3C, "RDF Current Status," 2016. [Online]. Available: https://www.w3.org/standards/techs/rdf#w3c_all. [Accessed: 24-Nov-2016].

[12] J. Weaver and P. Tarjan, "Facebook Linked Data via the Graph API," *Semant. Web*, Vol. 4, No. 3, pp. 245-250, 2013.

[13] K. Furdík, M. Tomášek, and J. Hreňo, "A WSMO-based framework enabling semantic interoperability in e-government solutions," *Acta Polytechnica Hungarica*, Vol. 8, No. 2, pp. 61-79, 2011.

[14] Schema.org, "About Schema.org," *Schema.org*, 2014. [Online]. Available: https://schema.org/docs/faq.html. [Accessed: 13-May-2016].

[15] P. Mika, "On Schema.org and Why It Matters for the Web," *IEEE Internet Comput.*, vol. 19, no. 4, pp. 52–55, Jul. 2015.

[16] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, and N. Lindström, "JSON-LD Implementation Report," *json-ld.org*, 2015. [Online]. Available: http://json-ld.org/test-suite/reports/.

[17] M. Sporny, G. Kellogg, D. Longley, and M. Lanthaler, "JSON-LD Framing 1.0, An Application Programming Interface for the JSON-LD Syntax," *W3C Community Group Draft Report*, 2012. [Online]. Available: http://json-ld.org/spec/ED/json-ld-framing/20120830/. [Accessed: 03-Mar-2016].

[18] M. Sporny and D. Longley, "Web Payments HTTP Messages 1.0 W3C

First Public Working Draft 15 September 2016," *W3C*, 2016. [Online]. Available: https://www.w3.org/TR/webpayments-http-messages/. [Accessed: 30-Dec-2016].

[19] R. Sanderson, P. Ciccarese, and B. Young, "Web Annotation Vocabulary W3C Candidate Recommendation 22 November 2016," 2016. [Online]. Available: https://www.w3.org/TR/annotation-vocab/. [Accessed: 27-Nov-2016].

[20] T. Kim, S. Campinas, R. Delbru, H. Jung, and S.-P. Choi, "High Performance Indexing of Materialized Graph Views," in *Proceedings of the 12th International Conference on Business Innovation and Technology Management*, 2013, pp. 1-7.

[21] T. Johnson, "Indexing Linked Bibliographic Data with JSON-LD, BibJSON and Elasticsearch," *Code4Lib J.*, No. 19, pp. 1-11, 2013.

[22] K. Nenadić, M. Letić, M. Gavrić, and I. Lendak, "Rendering of JSON-LD CIM Profile Using Web Components," in *Proceedings of the 14th International Symposium on Intelligent Systems and Informatics (SISY)*, 2016.

[23] G. Kellogg, "More specific frame matching #110," *GitHub*, 2012. [Online]. Available: https://github.com/json-ld/json-ld.org/issues/110. [Accessed: 01-Jun-2016].

[24] K. Nenadić, "Fork of jsonld.js with Recursive Prioritized Embedding Using @reverse," 2016. [Online]. Available: https://github.com/knenadic/jsonld.js.

[25] IEC, "Energy management system application program interface (EMS-API) - Part 501: Common Information Model Resource Description Framework (CIM RDF) Schema. IEC 61970-501." 2006.

[26] IEC, "Energy management system application program interface (EMS-API) - Part 552: CIMXML Model exchange format. IEC 61970-552." 2013.

[27] A. Stolz, B. Rodriguez-Castro, and M. Hepp, "RDF Translator: A RESTful Multi-Format Data Converter for the Semantic Web." 2013.

[28] K. Nenadić and M. Gavrić, "Enhancing CIM with Linked Data Capability," in *Proceedings of the 24th Telecommunications Forum (Telfor)*, 2016.

New Adaptation of Actuator Disc Method for Aircraft Propeller CFD Analyses

György Bicsák and Árpád Veress

Budapest University of Technology and Economics, Department of Aeronautics, Naval Architecture and Railway Vehicles
Műegyetem rkp. 3, H-1111 Budapest, Hungary
e-mail: gybicsak@vrht.bme.hu, averess@vrht.bme.hu

Abstract: The present paper is dedicated to introducing an accurate, generally applicable and minimum requirement demanding quasi steady-state CFD simulation method for investigating the effect of an aircraft propeller within the framework of the ESPOSA project. The simplest solution has been looked for in low Mach number flow regime, thus instead of direct discretization or using source terms, the Actuator Disk Method (ADM) has been applied with two different boundary condition settings: applying induced velocities or total pressure. Formerly, the Rotating Domain Model (RDM) has been validated, thus its output can be used as the reference solution. The results of the three models have been investigated both qualitatively and quantitatively. The most problematic part was the engine nacelle and propeller interaction, which has a strong influence on propeller efficiency. The investigation has shown that the ADM with total pressure boundary condition settings can provide acceptably close results to the reference RDM: within 5% amongst the investigated parameters.

Keywords: Actuator Disk Model; Schmitz method; Rotating Domain Model; Reynolds-Averaged Navier-Stokes Simulation; propeller-body interaction

1 Introduction

One of the most challenging problems nowadays is satisfying the continuing demands of increasing air traffic and introducing breakthrough innovations, leading technologies and green, sustainable solutions. These goals require fast and accurate solutions, for which purpose several outstanding R&D (Research and Development) projects are in progress [1], [2], [3], [4] and [5]. The design and optimization [6] of air-flows and structures in relation to jet engines, turboprops, helicopter rotors [7] and other turbo-machinery related configurations is a complex and cost demanding process. Especially for having accurate performances [8] and aerodynamic parameters, while expecting the application of proper material properties to maintain structural integrity [9]. The design and analyses in a spatially distributed manner are becoming more highlighted today

due to its effectivity [10], not only transportation such as aviation, ships and marine propulsion [11], [12] but also in the other segments of the industry. Novel design, sizing and calculation technologies are becoming available in both light and very light jet aircraft according to new trends [13].

In this paper, an accurate and fast CFD (Computational Fluid Dynamics) application is introduced within the framework of the ESPOSA (Efficient Systems and Propulsion for Small Aircraft) project [14]. The project itself is based on cost-efficient solutions, which enable and ensure access to development methods for smaller companies to design their own small aircraft type, while following the most up-to-date safety laws, regulations and recent international project goals, such as, the Clean Sky Project, and still reducing the overall cost demands. The BME (Budapest University of Technology and Economics) Department of Aeronautics, Naval Architecture and Railway Vehicles participates in the ESPOSA project and has the task of improving design specifications of the engine intake channel and nacelle of a newly developed tractor turboprop aircraft with numerical methods. The induced velocity distribution of the propeller has been determined by Schmitz's method and has been used in the flow modelling software as boundary conditions.

Beside the developments and/or applications of different CFD methods in the wide range of engineering coverage - as it is observable today in the state of the publications at different length scales: [4], [7], [15] - these approaches are highly capable also of investigating the airflow around rotor blades – propeller aerodynamics [16], or even the load distribution on the rotors [17]. The goal of this investigation is to establish a RANS (Reynolds Averaged Navier-Stokes Simulations) based (due to its reduced computational demand compared to Large Eddy Simulations) cost effective ADM method, which is capable of providing general, fast and accurate results in the pre- and serial-development phases of a new product by replacing several experimental results. The actuator disk method originates from the pioneering work of Rankine [18] and Froude [19], which still constitutes, in conjunction with the blade element theory, the most used analysis and design tool for aircraft propellers and wind turbines. A landmark review on this matter is presented by Glauert [20]. Wu [21] gives the exact and implicit solution of the flow through a generalized actuator disk. Conway [22] introduced explicitly Wu's solution through a semi-analytical procedure, which was later extended by Bontempo and Manna [23] [24] to ducted rotors with or without an axisymmetric hub of general shape. In order to reduce the computational cost related to the simulation of a geometrically resolved rotor, an actuator disk is often introduced in CFD codes.

The actuator disc method is already a well-determined concept, several papers and articles have been published to present different applications. One of the most significant fields is the modelling of airflow adjacent to wind turbine propellers, as represented by the following papers: an improved ADM was developed by Costa Gomes, Palma and Silva Lopes [25]; the ADM was applied by porous cells for wake modelling purposes in the publication of Gravdahl, Crasto, Castellani and Piccioni [26].

Regarding the aeronautical applications, Yu, Samant and Rubbert [27] developed an Euler solver for predicting flow field for propfan configuration using the actuator disc method. Although the theory behind the specification of the downstream disc surface boundary conditions is based on the total pressure, total temperature and flow angles, the method provided is rather code-dependent; it is difficult to generalize due to the extrapolation of the velocity from the computational domain. In other counterparts of the aerospace applications it is still a particularly important field of investigation, especially for helicopter, hover or tilt-rotor applications, as the following relevant examples show: LeChuiton [28], Visingardi, Khier and Decours [29], Conlisk [30], Farrokhfal and Pishevar [31], or Wald [29]. Lenfers [32] performed research in the design process on turboprop engine rotor, and Jeromin, Bentamy and Schaffarczyk [17] completed a rotor analysis in a wind tunnel.

As Coton, Marshall, Galbraith and Green have discussed the interaction of the rotor and an object nearby has been a subject of discussion for almost 30 years [33], of which most studies focus on the solution. Indeed, within the frame of the present paper, the novel actuator disk model has been developed with such boundary conditions that make it possible to investigate the effect of a nearby object on the propeller efficiency in general. The need for such a solution is required, as the actuator disk model generates a uniform induced velocity at the rotor disk [22], which is a constant boundary condition regardless of the environmental objects and distributions. Because of this, it is also worth mentioning that there is a different way in which the source terms are applied to the actuator disc surface imparting impulse and energy to the fluid, as Le Chuiton published it [28]. However, the generalization and the control of this approach with respect to the blockage in the flow field, for example, is not always possible and rather complicated comparing that with the presently applied procedure.

2 Theoretical Fundamentals

The governing equations in the present CFD simulations are the Navier-Stokes equations (together with mass and energy conservation laws) due to the Knudsen number being below 0.01 within the framework of the ANSYS CFX. The considered system of the nonlinear partial differential equation in their conservative form is valid: for both laminar and turbulent flows, for compressible one component ideal gas in a steady state inertial system, for a homogeneous isotropic material, in which the frictional processes are taken into account together with no potential field, such as gravity or magnetic forces, for example except for the rotating domain, where the inertial forces and their effects are considered and with the assumption of no sources or sinks are available. In order to avoid the uncertainty coming from the turbulent fluctuation while preserving the description of its effect in momentum transfer and from an energetics point of view, the corresponding parameters are averaged by Reynolds. The governing equations are the mass, momentum and total energy equations, in order [35]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{U}) = 0 \tag{1}$$

$$\frac{\partial (\rho \boldsymbol{U})}{\partial t} + \nabla \cdot (\rho \boldsymbol{U} \otimes \boldsymbol{U}) + \nabla p - \nabla \cdot \tau = \boldsymbol{S_M} \tag{2}$$

$$\frac{\partial (\rho h_{tot})}{\partial t} - \frac{\partial p}{\partial t} + \nabla \cdot (\rho \boldsymbol{U} h_{tot}) - \nabla \cdot (\lambda \nabla T) - \nabla \cdot (\boldsymbol{U} \cdot \tau) = \boldsymbol{U} \cdot \boldsymbol{S_M} + \boldsymbol{S_E} \tag{3}$$

where ρ stands for density, $\boldsymbol{U}$ is the velocity vector, p marks the pressure, τ is the stress tensor, h_{tot} is the total enthalpy, λ represents the thermal conductivity finally $\boldsymbol{S_M}$ and $\boldsymbol{S_E}$ are source terms. Following the Boussinesq approximation for modelling the components of Reynolds stress tensor, SST (Shear Stress Transport) turbulence model has been used in the all investigated cases. It combines the advantages of the *k-ω* and *k-ε* models. Blending functions control the usage of a *k-ω* formulation in the inner parts of the boundary layer makes the model directly usable all the way down to the wall through the viscous sub-layer. The SST formulation switches to the *k-ε* behaviour by the blending function in the free-stream and thereby avoids the common *k-ω* problem that the model is too sensitive to the free-stream value of the turbulence variables (in particular *ω*). The further distinction of the SST turbulence model is the modified turbulence eddy-viscosity function. The purpose is to improve the accuracy of prediction of flows with strong adverse pressure gradients and pressure-induced boundary layer separation. The modification accounts for the transport of the turbulent shear stress, which is based on Bradshaw's assumption that the principal shear stress is proportional to the turbulent kinetic energy [34].

Rotating Domain Model

One of the applied models is the RDM. The base of this approach is that the propeller blades with the hub and the adjacent fluid domain rotate with the angular velocity of the turboprop. This model is basically considered as the most accurate one, but to simulate the wall boundary effect correctly, a fine mesh is necessary, which significantly increases the computational time and performance demand.

Because the fluid also rotates together with the rotating blades with a constant angular velocity, additional components are required in the governing equations. The centrifugal force and Coriolis force's effects are included in the momentum equation with the following sources [35]:

$$S_{M,rot} = S_{Cor} + S_{cfg} = -2\rho\Omega \times U - \rho\Omega \times (\Omega \times r) \tag{4}$$

In the energy equation, the advection of the enthalpy has to be replaced by the advection of rothalpy (I) in the following form [35]:

$$I = h + \frac{1}{2}U^2 - \frac{1}{2}\Omega^2 r^2 \tag{5}$$

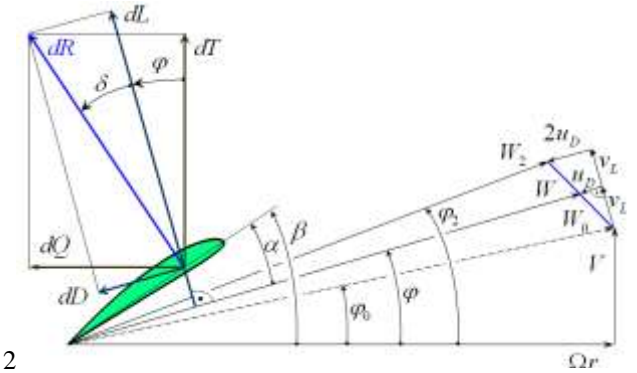
Schmitz Method for the Actuator Disk Model

The output of the present calculation is the required power of the propeller and the distribution of the induced velocities in the radius. The condition of the calculation is that the given power be equal to the calculated power at the belonging blade setting. The results of the present propeller analyses are used for the boundary conditions of the CFD analysis in the ADM models.

The swirling effect of the propeller is simulated by adopting the actuator disc model based on Schmitz method. This model deals with the aerodynamic forces created by the airflow around the propeller blade elements. The calculation uses the combined Blade Element and Momentum Theory [30]. The mathematical model has the following input data:

- air density and temperature (depends on the flight altitude, based on International Standard Atmosphere data; $\rho = 0.9048\ kg/m^3$, $T = 268.4\ K$);
- free stream (flight) velocity of the aircraft (112 m/s);
- diameter and RPM (Rotation Per Minute) of the propeller (2.08 m and 1950 RPM);
- blade number (4 pcs.), blade profile, chord, and the relative blade thickness at 75 % of the blade length;
- c_L^α and c_D^α (angle of attack dependent lift and drag coefficients) curves of the profile NACA 4412.

Although the density here is stated as a constant value, this assumption is valid only to determine the induced velocity components. During the CFD simulation cases the density can be changed. The aerodynamic forces, velocity triangles and notations are illustrated in Figure 1. The induced velocity in the "lift" direction is marked as v_L, while the induced velocity in the "drag" direction is marked as u_D.



2

Figure 1

Velocity and force components at a blade element [36] *//note: modified figure from cited source*

The relative velocity of the profile (W) is expressed by the relative base velocity (W_0) constructed by peripheral (Ωr), flight speed (V) and induced velocity u_D [36]:

$$W = W_0 \cos(\varphi - \varphi_0) - u_D \tag{6}$$

Based on Figure 1 notations, the elementary drag force (dD) and lift force (dL) can be determined by momentum theory from which the first and second connection equations, between the blade-element and momentum theory, can be expressed by the two sides of the right equalities [36]:

$$dD = d\dot{m}(2u_D) = \rho(2\pi r dr)W \sin\varphi\,(2u_D) = Bc_D\frac{\rho}{2}W^2 c\,dr \tag{7}$$

$$dL = d\dot{m}(2u_L) = \rho(2\pi r dr)W \sin\varphi\,(2v_L) = Bc_L\frac{\rho}{2}W^2 c\,dr \tag{8}$$

In equation (7) and (8) c is the chord distribution of the rotor, and actually is the function of the distance from the rotation axis (c=$c(r)$), B is the number of blades, dr is the elementary radius, r is the radius, c_L and c_D are the lift and drag coefficients respectively. Based on the connection equation (7) and velocity triangle in Figure 1, the induced velocity components in the direction of drag and lift forces are the followings [36]:

$$u_D = \frac{B\,c}{8\pi r}\frac{c_D}{\sin\varphi}W \text{ and } v_L = W_0 sin(\varphi - \varphi_0), \tag{9}$$

By combining the first equation in (9) and equation (6), the relative base velocity is next [36]:

$$W_0 = \frac{W}{\cos(\varphi - \varphi_0)}\frac{\frac{8\pi r}{Bc}\sin\varphi + c_D}{\frac{8\pi r}{Bc}\sin\varphi} \tag{10}$$

By substituting equation (10) to equation (8) (with replacing v_L by (9)) one gets:

$$2\pi r \sin\varphi\left[2\frac{W}{\cos(\varphi - \varphi_0)}\frac{\frac{8\pi r}{Bc}\sin\varphi + c_D}{\frac{8\pi r}{Bc}\sin\varphi}\sin(\varphi - \varphi_0)\right] = \frac{W}{2}Bcc_L \tag{11}$$

After simplifying equation (11) it can be written:

$$cc_L - \left[\frac{8\pi r}{B}\sin\varphi + cc_D\right]\tan(\varphi - \varphi_0) = c_L - \left[\frac{4}{\sigma}\sin\varphi + c_D\right]\tan(\varphi - \varphi_0) = 0, \tag{12}$$

which is the base equation of the calculation (propeller solidity: $\sigma = Bc/(2\pi r)$). If the solutions (φ, c_L, c_D) were substituted in (12) the expression becomes 0. But if different values are used than the solutions, there would be a non-zero residuum $\Re$ as it is found below:

$$c_L - \left[\frac{4}{\sigma}\sin\varphi + c_D\right]\tan(\varphi - \varphi_0) = \Re \tag{13}$$

(13) is a non-linear equation for φ, c_L, c_D, which are dependent variables of each other. Thus, the numerical calculation can be done by using Newton iteration to find the value φ (and then c_L and c_D can be calculated by $c_L - \varphi$ and $c_D - \varphi$ plots) [36]:

$$\varphi_{NEW} = \varphi_{OLD} - \frac{\mathcal{R}}{\frac{\partial\Re}{\partial\varphi}} \tag{14}$$

where the derivative of the residuum $\Re$ is [36]:

$$\frac{\partial \Re}{\partial \varphi} = \frac{\partial c_L}{\partial \varphi} - \left[\frac{4}{\sigma}\cos\varphi + \frac{\partial c_D}{\partial \varphi}\right]\tan(\varphi - \varphi_0) - \left[\frac{4}{\sigma}\sin\varphi + c_D\right][1 + \tan(\varphi - \varphi_0)^2] \quad (15)$$

In this work, the tip-loss correction is neglected. Although at the blade tip the induced velocities are 0 in reality, the partner institute hasn't applied it in the project [37], so in this way, the results were more comparable in the final report. Of course, one of the improvement possibilities is to incorporate the blade tip loss. The induced velocities in axial and tangential direction can be calculated if v_L and u_D are determined [36]:

$$v = v_L \cos\varphi - u_D \sin\varphi \quad \text{and} \quad u = v_L \sin\varphi + u_D \cos\varphi \quad (16)$$

In order to use the induced velocities as actual boundary conditions, they have to be multiplied by two and the flight speed is added to the axial induced speed. Furthermore, the static temperature and incoming flow direction are specified also.

This work is dedicated to highlighting the differences between two boundary conditions, and compares the results to the base rotating domain provided data. The computed v and u are used in ADMv1 model, while ADMv2 requires a slight additional calculation: the density is supposed to be constant within an infinitesimal volume of the fluid that moves with the flow velocity, so the flow is supposed to be incompressible (M_{local}=0.34-0.37). Hence, the absolute total pressure distribution along the radius is calculated from the induced speeds and flight velocity (V) with the following equation:

$$p_{total} = p + \frac{\rho}{2} \cdot [(2v + V)^2 + (2u)^2] \quad (17)$$

The static temperature, pressure and density at flight altitude are determined by ISA (International Standard Atmosphere) [37]. As later described (see the description of rotor), the ADMv2 applies total pressure inlet values with constraint airflow direction, instead of defining the velocity vector in the rotor sweeping surface.

3 Introduction of the goals and Analyzed Aircraft Configuration

The goal of the ESPOSA project is to provide designing methods for small aircraft manufacturers by implementing new approaches or coupling already well-established methods. Since this work has been performed during the project, no experimental data were available for the behaviour of the configuration, but wind-tunnel tests have provided data for the propeller itself. Since the rotating domain had been validated earlier as a method that can provide results within 3%, this approach has been used later as a baseline model. The actuator disc method provided induced velocities were applied as boundary conditions (ADMv1) and

the relative total pressure has been set on the actuator disc surface (ADMv2). The point is to introduce that ADMv1 cannot handle the existence of adjacent objects: setting velocities results in an axisymmetric boundary condition, although the presence of wing, nacelle and engine intake influence the velocities, torque induced by the propeller.

The investigated aircraft is a high-wing, twin engine, turboprop, multi-purpose aircraft for transportation of passengers and/or cargo to be designed and produced by a partner company [40]. The nose landing gear of the aircraft is retractable; tail unit is T-shaped. Nine passengers can be transported in the unpressurized cabin with two crew members. Type AVIA tractor propellers are installed on the engines. The propellers have four blades, made from aluminium alloy and can be hydraulically actuated with single acting regulation of constant rotational speed, feathering and reversing with possibility of rotating speed and phase synchronization. The propeller blades are set to low pitch by pressurized engine oil, to the high-pitch by a spring and counter-balances on the blades. The propeller is equipped with an electrical de-icing system [41].

Geometry

The 3-dimensional CAD (Computer Aided Design) model of the analysed configuration has been provided by a partner institute [37]. In order to reduce the computation power requirements only one symmetric part of the aircraft was investigated without the fuselage. Downstream of the rotor, at the lower section of the nacelle, is the air duct inlet which is equipped with a particle and ice separator device. The separated particles are drawn through the sidewall duct outlets overboard. Upstream of the gas turbine section there is a static inlet guide vane air intake device, and downstream of the gas turbine the exhaust pipe, where the hot gas leaves the engine. At the bottom of the nacelle a cross-flow, oil-to-air type heat exchanger cools down the engine oil, which uses ambient ram airflow.

Even with this reduced geometry, the complexity of the CAD model was still too high, so some additional parts have been neglected or simplified. The rotors for the RDM and ADM have been handled in two different ways (see Figure 2):

- for the RDM all four rotor blades have been kept and a cylinder was applied around them,
- for the ADM v1 the same cylinder has been built up, but inside the cylinder everything (blades, nose cone) has been removed,
- for ADM v2 the former model has been modified: the plane surfaces of the cylinder have been divided into 10 annuli, where the boundary conditions are set.

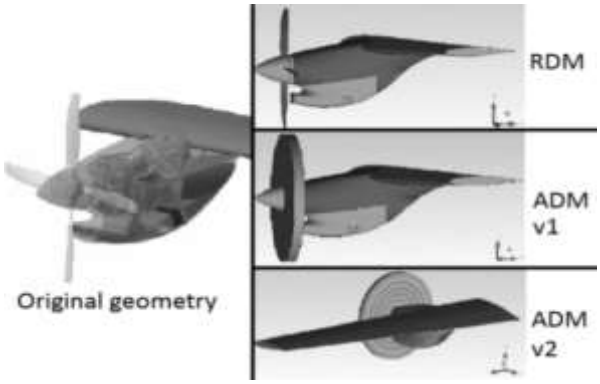


Figure 2

The original geometry [40] and simplified models for the three models *//note: original geometry is adapted from the cited source*

In order to minimise the time demand and complexity of the discretization process, the mesh has been built up using tetrahedron elements. Three-dimensional volume meshes with boundary layer subdivisions have been generated for the all domains of all versions to be investigated. Regarding the flow parameters with high gradients, local mesh refinement was applied and inflation layers provided the necessarily low $y+$ values, to be below 300 depends on the Reynolds number [35]. The global number of elements for RDM was 19,065,034, for ADM v1 and ADM v2 13,428,609.

Material Properties, Physical Settings and Boundary Conditions

The solution of the non-linear partial differential equations in a discretised form requires additional definitions. The material of the flow domains was air as an ideal gas ($\gamma = 1.4$ and $R = 287.058\ J/(kg\ K)$. The reference pressure of the domains has been calculated by the cruise altitude of the aircraft (3048 m) and ISA (International Standard Atmosphere) and set to 69682 Pa [14]. The effect of gravity is negligible in this case, but the viscous work term must be considered within the total energy model during modelling the heat transfer. Since the scaling varies in wide range Shear Stress Transport (SST) turbulence model is recommended for turbulence closure and 5% inlet turbulent intensity has been applied in each case [35].

Main flow

The main airflow contains the bounding volume of the flow field, the shape of the wing, engine nacelle, nosecone, oil cooler, its intake and outlet and the cylindrical surfaces, which includes the rotor and the engine exhaust surface represents the hot exhaust gas inlet into the main airflow. The main flow boundary condition settings are illustrated in Figure 3. 112 m/s cruise speed of the aircraft with 268.4 K static temperature has been set as an inlet boundary condition. 0 Pa relative pressure has been imposed on the outlet surface. Symmetry boundary condition

has been applied to the inner boundary surface, which makes possible to simplify the model: this boundary condition essentially acts as a frictionless wall. On the top, side face and bottom of the flow field "opening" boundary condition has been set with entrainment mass and momentum option and 0 Pa opening pressure [29]. This approach allows undisturbed inflow and outflow depending on the total pressure of the fluid in the computational domain and at the boundary. The engine exhaust surface has been represented to simulate the hot exhaust gas inlet into the main airflow. The mass flow of this inlet has been set equal to the engine intake outlet air and the fuel mass flow rate: 1.9759 kg/s and the static temperature of the inflowing fluid has been set to 853.3 K, based on the manufacturer's data [40].

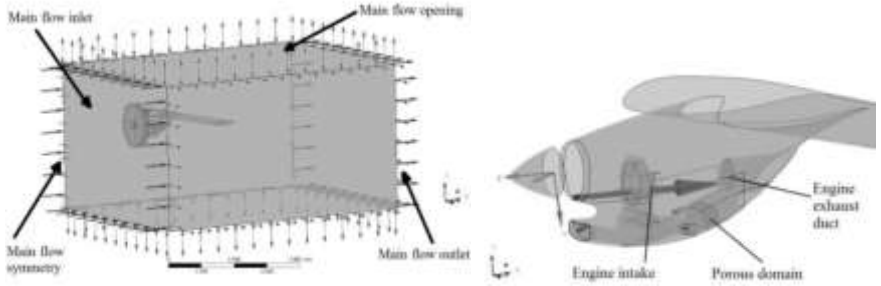


Figure 3

Boundary conditions on the main airflow domain

Heat Exchanger and Air Intake Duct

A local pressure drop is caused by the blockage of the heat exchanger in the main airflow; thus, a local low-pressure zone is formed downstream of the oil cooler heat exchanger for which purpose porous domain was used with the same volume as the heat exchanger. The pressure drop curve in the function of the normal volume flow rate has been provided by the manufacturer and has been used to determine the input parameters for the porous domain, as resistance and loss coefficient and volume porosity [40]. Indeed, the necessary calculations have been made analogically and iteratively. The scope of this paper is to introduce the settings of the rotor boundary conditions and compare the results. Thus, the calculation method is not detailed.

The role of the air duct is to collect air and forward it to the engine compressor unit with the lowest possible pressure drop, flow uniformity; and if icy conditions occur, or it is raining, separate the ice (or simply dust) particles and water droplets from the airflow to protect the compressor stage and the further part of the engine. During the investigation process of ESPOSA project, the air duct optimisation was also the scope. Hence, a separate fluid domain has been created only for the air duct in order to determine the necessary parameters – boundary conditions for the individual simulations. A general fluid to fluid interference connection has been used between the air duct separator part and main airflow. At the outlet section of the air duct (inlet of the compressor), the air demands of the engine has been imposed as a constant mass flow rate with 1.9398 kg/s.

Rotor

1. The configuration of the RDM is the most simple: the same fluid model has to be applied, as in the case of the main flow or for the air duct domain, but a coordinate system is in the rotational axis of the rotor, around which it constantly rotates with 1950 1/min. The rotation caused rothalpy-increment generates the thrust and torque by achieving pressure rise. Since the similar model has already been validated, it is considered to be the baseline model.

2. In the ADMv1 version, the induced velocities are calculated in the function of the distance from the rotational axis based on the propeller characteristics given in Figure 4. The continuous and dotted lines represent the axial and tangential induced velocities from the axis of rotation moving outwards along the blades. Through the front surface of the cylinder the airflow can leave the domain and across the rear surface the fluid enters. The same amount of mass flow rate is defined at the upstream surface of the disc and at the exit of the propeller plane for mass conservation, which is the same at ADM2 version too. In the inlet surface the velocity vectors and static temperature (268.4 K) boundary conditions represent the effect of the propeller. The cylindrical surface close to the tip section of the imagined blades has been set as a free slip wall, assuming that there is no airflow across this boundary.

3. In the ADMv2 version, total pressure values were set on the inlet boundary surface in the function of the radius, constructed from the static far field pressure and dynamic pressure, which is calculated by the ambient density and local velocity vector determined at ADMv1 model. In essence, if there was an object downstream of the actuator disc, the RANS solver would create the corresponding flow field, compatible with those conditions. In our particular case to reduce computational demands, the plane surfaces have been divided into 10 annular sections, and the average total pressure has been calculated for each annulus with Simpson's method. The relative total pressure values in each section are represented by the column chart in Figure 4.

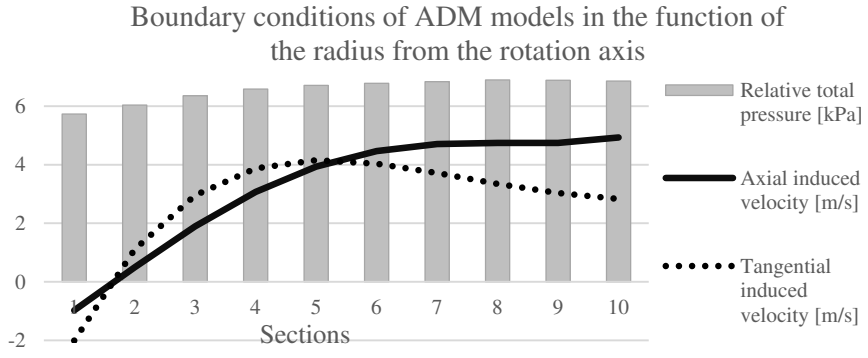


Figure 4

Components of the boundary conditions of ADM models in the function of the applied sections moving from the rotational axis outwards

Solver Properties

The model has been treated as a quasi-steady-state simulation since the interest is the flow parameters and the comparison between the investigated versions during cruise phase, when the flow conditions are assumed to be constant over the time. High-resolution advection scheme has been used, just like the turbulence numeric option. The target iteration number was 1000 for the first approach, but the actual in the RDM simulation was not enough for the residuum to converge. Finally, a four times higher iteration number was needed. The ADM models' imbalances reached $2*10^{-6}$ values sooner, thus the simulation cases can be stated to be converged. The auto timescale has been applied, and the residuum target for RMS (Root Mean Square) values was 10^{-7}.

The goal of the present research was to provide a method, which is achievable with reduced computation capacities, and not necessarily require HPC (High-Performance Computing) solutions to handle the problem. Thus, the three simulations have been executed on the same computer, which is equipped with Intel® Core™ i7-3770 CPU and 8 GB RAM.

Results and Evaluation

The RDM model finally required four times more iteration number to reach the convergence criteria – in comparison with the versions of ADM – and this process took $1.3074*10^{6}$ CPU wall clock seconds. The ADMv1 has reached the criteria after approx. 100 iteration steps, but in order to eliminate any perturbation in the numeric solver algorithm, the computation duration has been expanded untill 600 iteration steps, which required $1.2786*10^{5}$ CPU seconds. In the case of ADMv2 the proceeding was the same, converged early, but the running has been extended to 600 iteration steps and lasted for $6.6713*10^{4}$ CPU seconds.

It is clear that the ADM approaches needs less than 10% CPU time of the RDM. This is caused by the significantly higher mesh number, but also by the longer time demand of the different residuals to drop below the convergence limit since the interface between the rotating and stationary domains causes additional disturbances in the algebraic equations. In number, it was proven that the ADMv1 has converged in 9.78% of the RDM time demand, and ADMv2 has completed in 5.1% of the RDM CPU wall clock time, so if the accuracy of the results is acceptable, the time-factor would be a serious pro for the ADM methods.

One key indicator of the results' quality and plausibility is the $y+$ number. In terms of the recent requirements, for a simulation with the given length scales and Reynolds number, the $y+$ value is maximised in 300. In the simulations, the maximum $y+$ number was 179.3 on the nacelle assembly. Meanwhile, in the case of RDM on the rotor blades, the highest $y+$ value was 107.4. Certainly, mesh sensitivity analysis was completed earlier, but in the framework of this paper, only the already checked mesh has been used. To reach the convergence criteria was no problem for the ADM simulations, generally, within 100 iteration steps the residuals dropped below 10^{-6}, and the imbalances below 1%. The RDM run on the other hand required 3500 iteration steps to achieve the same convergence.

Generally, the airflow pattern corresponds to the expectations. Upstream of the nose cone, where the airflow is decelerated, there is an increased static pressure zone. Downstream of the rotor blade sweep surface the total pressure is increased and kinetic energy has been added to the fluid by the propeller, as it is illustrated in Figure 5 bottom row. It is also obvious that there is a higher-pressure zone upstream of the air duct highlighted by the absolute static pressure distribution (middle row).

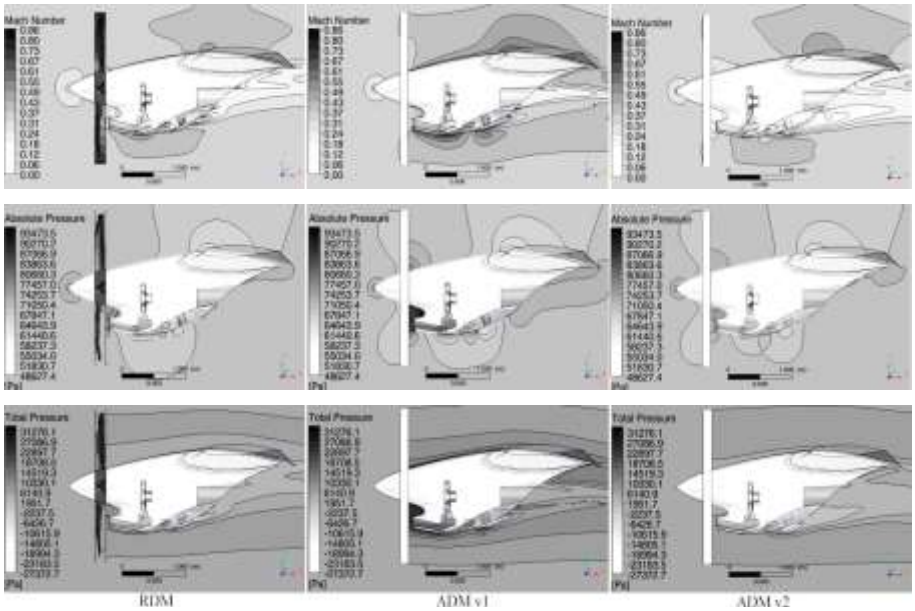


Figure 5

Representation of different parameters illustrated on the vertical mid-section plane of the engine nacelle

The compressor has a constant mass flow rate, thus the unnecessary air quantity escapes, and passes along the nacelle's side, and this blockage effect, together with the engine nacelle, also influences the velocity field downstream of the propeller blades. Basically, this phenomenon represents the problem of the ADMv1. In reality because of the discussed effect, the incoming velocities are less and deviate in particular zones due to the proximity of engine nacelle and the propellers' efficiency slightly drops. In the boundary conditions of the ADMv1 model, the velocity vectors were defined according to the calculated data and are introduced in Figure 4. These velocities are constraints, so the velocity distribution will not change downstream of the propeller. The simulation can reach this condition by increasing the local absolute total pressure at the boundary, which has produced also higher static pressure values. This phenomenon produces 10-15% higher pressure values. It cannot be observed in the other two cases. It can be also concluded that Mach number, relative static and relative total pressure

distribution pattern show similarities in the case of RDM and ADMv2 model according to Figure 5.

The different results are investigated in the function of dimensionless distance from the rotational axis. Six lines have been created downstream of the rotor, see the right upper corner of Figure 6. Both the Mach number distribution and total pressure distributions are averaged over the lines 1, 2, 3 and 4, 5, 6, which mark the significant differences caused by the different boundary condition setting between ADMv1 and ADMv2. Closer to the rotational axis, the interaction between the fluid flow and engine nacelle strongly influences the flow pattern.

In this particular area, the nacelle decreases the efficiency of the rotors, according to that the induced velocities decrease. This effect has been correctly handled by the RDM and ADMv2, but ADMv1 treats the velocities as constraints, hence the Mach number distribution (Figure 6 left upper corner) shows a significant overestimation closer to the propeller blades. The higher induced velocities come with higher local pressure values, thus the total pressure distribution has also similar pattern and confirms the better assumption of ADMv2 boundary conditions.

The total pressure distribution through the propeller/actuator disk also confirms the accuracy of ADMv2: while the average relative error between RDM and ADMv1 is 23.4%, ADMv2 reduced this error to only 3.9%, see Figure 7.

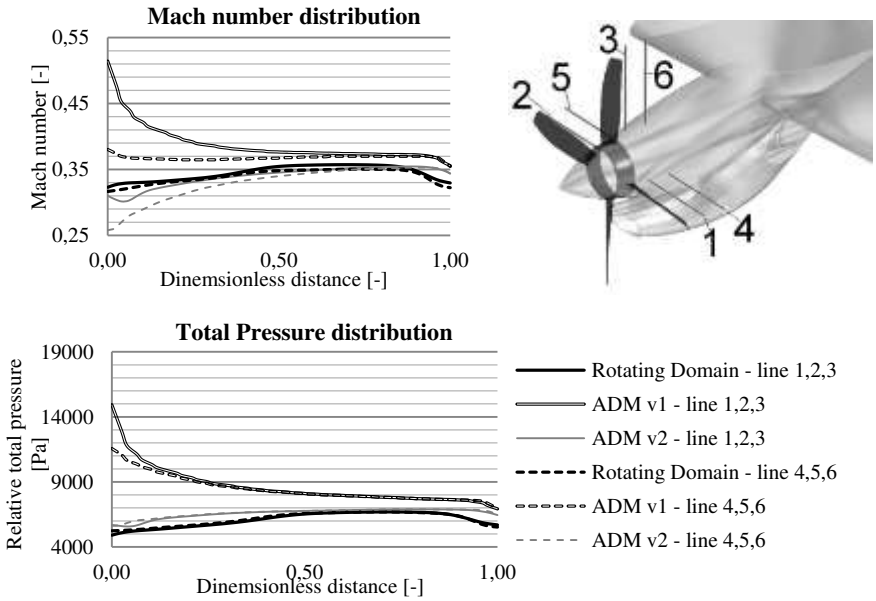


Figure 6

Lines from 1-6 downstream of the propeller sweep surface for parameter investigation (top right) and exported averaged parameters along the specified lines (top left and bottom)

The overestimated total pressure, caused by the used boundary condition treatment of the software, can be observed even better by considering the pressure distribution on the engine nacelle, see Figure 8. This comparison clearly shows that ADMv1 has computed too high-pressure values, in fact, based on Figure 5, it is higher than it is actually possible a maximum of 32% .The maximum total pressure can be calculated in the function of maximum induced and flight speeds at the downstream of the propeller plan is the following: $p_{total} = p + \frac{\rho}{2} \cdot [(2v + V)^2 + (2u)^2] = 69682\ Pa + \frac{0.9048 \frac{kg}{m^3}}{2} \cdot \left[\left(2 \cdot 4.7 \frac{m}{s} + 112 \frac{m}{s}\right)^2 + \left(2 \cdot 3.71 \frac{m}{s}\right)^2\right] =$ 76374.4 Pa. At the same time, neither of the distributions are symmetrical, the tangential components of the induced velocities create an asymmetric pattern. The engine nacelle downstream of the rotor also influences the parameters along the rotor's sweep surface by decreasing the induced velocities. This phenomenon can be observed in the case of RDM and ADMv2 cases, which is a remarkable achievement since the boundary conditions of the Actuator Disk Models are defined axis-symmetrically (considering both the velocity and pressure components), but while ADMv1 keeps the velocity distribution unchanged. ADMv2 is more flexible and can modify the velocity vector distribution, while maintaining the total pressure distribution on the demanded level.

The engine intake duct ensures a constant air-consumption, thus propeller-delivered airflow goes through the intake duct inlet. This process decreases the local absolute pressure in that particular zone. This pressure distribution, can be useful in the sizing process of the engine nacelle, to calculate the surface loads, like Renaud, O'Brien, Smith and Potsdam did it [41].

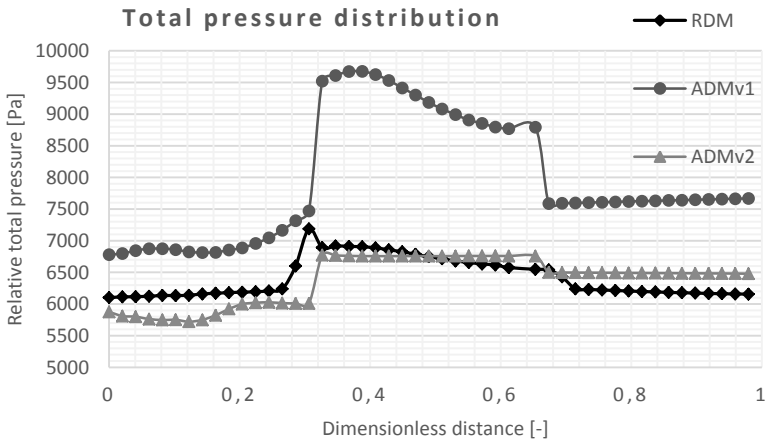


Figure 7

Effect of the propeller on total pressure distribution

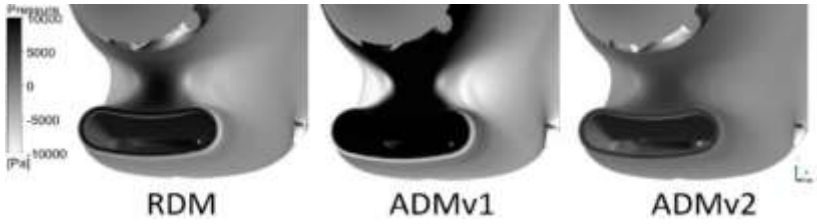


Figure 8

Relative total pressure distributions on engine nacelle downstream of the propeller blades

In order to compare the results of the different models from the propulsion viewpoint, the thrust and torque parameters of the propellers have been investigated. Each aircraft engine generates 364 kW shaft power in cruise phase, from which the generated torque is 1782.54 Nm. The thrust and torque coefficient can be calculated by the following expressions [41]:

$$c_T = \frac{T}{n^2 D^4 \rho} \text{ and } c_Q = \frac{Q}{n^2 D^5 \rho} \tag{18}$$

In equation (18) the density (ρ) was area averaged downstream of the propeller, in the same circular section. From these coefficients, the propeller efficiency in the function of J, advanced ratio, can be determined according to (19).

$$\eta = \frac{J}{2\pi}\frac{c_T}{c_q} \text{ where } J = \frac{V}{nD} \tag{19}$$

As Table 1 represents, the generated thrust forces are really close to each other in the case of the RDM and ADMv2 model, the relative difference is 1.37% between the parameters. The overestimated downstream velocity distribution by ADMv1 is resulted in 21.77% difference. Computing the area averaged density and velocity downstream of the propeller the torque, assuming that RDM is the most accurate – reference model, the thrust – torque coefficients, advanced ratios and propeller efficiencies have been calculated. The relative difference between ADMv1 and RDM were significantly high from the viewpoint of every investigated parameter, while ADMv2 provided results close to RDM, with the maximum value for the relative error of 3.92%. The propeller thrust of the ADMv1 model was not only higher but resulted in unreal propeller efficiency: 108.56%, which is not possible.

Table 1

Simulated thrust and torque parameters of the propeller

	RDM	ADMv1	ADMv2	ADMv1 rel. error [%]	ADMv2 rel. error [%]
T [N]	2968.8	3528.2	2857.6	21.77	1.37
ρ [kg/m^3]	0.8951	0.9407	0.9073	5.09	1.37
$\boldsymbol{c_T}$ [-]	0.1637	0.1897	0.1593	15.87	2.7
$\boldsymbol{c_Q}$ [-]	0.0484	0.0461	0.0478	10.58	3.92
J [-]	1.6568	1.6568	1.6568	-	-
$\boldsymbol{\eta}$ [-]	0.9135	1.0856	0.8793	21.77	1.37

Although for the whole geometry no experimental data is available yet, by observing the results, it can be concluded that ADMv2 model is capable of providing results close to the RDM model; so accurate results with accepted accuracy in a much shorter time can be achieved.

Conclusions

In the framework of the ESPOSA project, an aerodynamic analysis of a turboprop engine, its propeller and internal flow channels has been completed. The goal of the present work is to compare the results of different simulation approaches and develop an accurate, fast and general method for considering the effect of the propeller. Three numerical analyses have been performed and the results of the three different methods have been compared with each other are the next: Rotating Domain Model and Actuator Disk Model with two different boundary condition settings.

The computational time demand for the imbalances to reach 1% and residuum to converge untill $2*10^{-4}$ in the case of the RDM was nearly 15 days, for ADMv1 approx. 1.5 days and for ADMv2 is 0.75 day on the same computer with Intel® Core™ i7-3770 CPU and 8 GB RAM. Also, the RDM required almost 4000 iteration steps, while for ADM cases 200 iterations were sufficient. Consequently, the ADM models have significantly shorter computational time demand. According to the investigated parameters for determining the accuracy of the different approaches, the ADMv2 provides the closest results to the RDM method, which is considered to be the most accurate one due to the detailed physical and numerical representations and based on the previous validation. If the boundary conditions for the ADM (ADMv2) are the total pressure with determined flow direction and static temperature, the solver is capable of taking the effect of geometrical object downstream of the rotor blades into consideration. Hence, ADMv2 is an effective approximation and so the replacements of the Rotating Domain Model, both in terms of results and in computational time demand, are reasonable. The results could be improved by applying a finer mesh or using functional representation instead of the 10 annuli.

The investigation also supports other applications, so the developed method is intended to apply in " Small aircraft hybrid propulsion system development" supported by Hungarian national EFOP-3.6.1-16-2016-00014 project titled " Investigation and development of the disruptive technologies for e-mobility and their integration into the engineering education".

Acknowledgements

The research has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. ACP1-GA-2011-284859-ESPOSA.

References

[1] Rohács, D. and Rohács, J., "Magnetic Levitation Assisted Aircraft Take-Off and Landing (feasibility study – GABRIEL concept)" *Progress in Aerospace Sciences*, 85: pp. 33-50, 2016

[2] Bera, J., and Pokorádi, L.: Monte-Carlo Simulation of Helicopter Noise, Acta Polytechnica Hungarica, 12:(2) pp. 21-32, 2015

[3] Bréda, R., Lazar, T., Andoga, R. and Madarász L.: Robust Controller in the Structure of Lateral Control of Maneuvering Aircraft, Acta Polytechnica Hungarica, pp. 101-124, Vol. 10, No. 5, 2013

[4] Beneda, K.: Numerical Simulation of MEMS-based Blade Load Distribution Control in Centrifugal Compressor Surge Suppression, ICNPAA 2012 Congress. American Institute of Physics, Conference Proceedings 1493, 116 (2012); http://doi.org/10.1063/1.4765479, pp. 116-123, 2012

[5] Andoga, R., Főző, L., Madarász, L. and Karol, T.: A Digital Diagnostic System for a Small Turbojet Engine, Acta Polytechnica Hungarica, pp. 45-58, Vol. 10, No. 4, 2013

[6] Chattopadhyay, A., and Narayan, J. R., "Optimization Procedure for Design of High-Speed Prop-Rotors" *ASCE Journal of Aerospace Engineering* Volume 7, Issue 2 (April 1994)

[7] Cao, Y., and Yu, Z., "Numerical Simulation of Turbulent Flow around Helicopter Ducted Tail Rotor", *Aerospace Science and Technology* 9 (2005) 300-306

[8] Fujii, K., "Progress and Future Prospects of CFD in Aerospace – Wind Tunnel and Beyond"; *Progress in Aerospace Sciences* 41 (2005) 455-470

[9] Edwards, K. L., and Davenport C., "Materials for Rotationally Dynamic Components: Rationale for Higher Performance Rotor-Blade Design", *Materials and Design* 27 (2006) 31-35

[10] Fu, S., and Wang, L., "RANS Modeling of High-Speed Aerodynamic Flow Transition with Consideration of Stability Theory", *Progress in Aerospace Science* 58 (2013) 36-59

[11] Muscari R., and Mascio A. Di, "Simulation of the Viscous Flow around a Propeller using a Dynamic Overlapping Grid Approach", *First International Symposium on Marine Propulsors*, smp'09, Trondheim, Norway, June 2009

[12] Schweighofer, J., van der Meij, K., Gronarz, A., Hargitai, Cs. L., Simongáti, Gy., Demonstration by Simulation: The Four Simulator Demonstrators of the FP7 EU Project MoVeIT!, Proceedings of Congrès SHF: Hydrodynamics and simulation applied to inland waterway and port approaches, Paris, France: Société Générale Hydrotechnique, PIANC, 2015, pp. 1-11, ISBN 979-10-93567-08-2

[13] Huda, Z., Edi, P., Almajid, A. A., and Al-Garni, A. Z. "New Trends in Designing Markets, Configurations, and Materials for Very Light Jet Aircrafts" *ASCE Journal of Aerospace Engineering* Volume 25, Issue 3 (July 2012)

[14] Seventh Framework Programme, Theme [AAT.2011.4.4-4.], Project ESPOSA: Annex I – "Description of Work", Grant agreement no: 284859, Version date: 2011-09-20

[15] Rácz, N., Kristóf, G., Weidinger, T.: Evaluation and Validation of a CFD Solver Adapted to Atmospheric Flows: Simulation of Topography-induced Waves, "Időjárás" / Quarterly Journal of the Hungarian Meteorological Service, 117:(3) pp. 239-275, 2013

[16] Wald, Q. R., "The Aerodynamics of Propellers"; *Progress in Aerospace Sciences* 42, pp. 85-128, 2006

[17] Jeromin, A., Bentamy, A., and Schaffarczyk, A. P., "Actuator Disk Modeling of the Mexico Rotor with OpenFOAM"; *ITM Web of Conferences 2*. 06001, DOI: 10.1051/itmconf/20140206001, 2014

[18] Rankine, W. J. M., "On the Mechanical Principles of the Action of Propellers". *Transactions Institute of Naval Architects 6*, p. 13, 1865

[19] Froude, R. E., "On the Part Played in Propulsion by Differences of Fluid Pressure". *Transactions of the Institute of Naval Architects 30,* pp. 390-405, 1889

[20] Glauert H., "Airplane Propellers". *Aerodynamic theory. Ed. by W. F. Durand. Vol. IV*, Division L. Springer, 1935, pp. 169-360

[21] Wu, T. Y., "Flow through a Heavily Loaded Actuator Disc". *Schiffstechnik 9* (1962) pp. 134-138

[22] Conway J. T., "Exact Actuator Disk Solutions for Non-Uniform Heavy Loading and Slipstream Contraction". *J. Fluid Mech.* 365 (1998) pp. 235-267

[23] Bontempo, R., Manna, M., "Solution of the Flow over a Non-Uniform Heavily Loaded Ducted Actuator Disk". *Journal of Fluid Mechanics* 728 (2013) pp. 163-195. ISSN: 1469-7645

[24] Bontempo, R., Manna, M., "A Nonlinear and Semi-Analytical Actuator Disk Method Accounting for General Hub Shapes: Part I - Open Rotor". *Journal of Fluid Mechanics* (2016) DOI: 10.1017/jfm.2016.98

[25] Costa Gomes, V. M. M. G., Palma, J. M. L. M., and Silva Lopes, A., "Improving Actuator Disk Wake Model", *Journal of Physics: Conference Series* 524 (2014) 012170, TORQUE 2014

[26] Gravdahl, A. R., Crasto, G., Castellani, F., and Piccioni E., "Wake Modeling with the Actuator Disc concept", *Energy Procedia 24* (2012) 385-392

[27] Yu, N. J., Samant, S. S., Rubbert, P. E., "Flow Prediction for Propfan Configurations using Euler Equations", *AIAA 84-1645, 17th Fluid Dynamics, Plasma Dynamics and Lasers Conference*, June 1984, Snowmass, Colorado

[28] Le Chuiton, F., "Actuator Disc Modeling for Helicopter Rotors", *Aerospace Science and Technology* Volume 8, Issue 4 2004, pp. 285-297

[29] Visingardi, A., Khier, W., and Decours, J., "The Blind-Test Activity of Tiltaero Project for the Numerical Aerodynamic Investigation of a Tilt Rotor", *ECOMAS 2004* Jyväskylä, 24-28 July 2004

[30] Conlisk, A. T., "Modern Helicopter Rotor Aerodynamics"; *Progress in Aerospace Sciences*, 37 (2001) 419-476

[31] Farrokhfal, H., and Pishevar, A. R., "Aerodynamic Shape Optimization of Hovering Rotor Blades using a Coupled Free Wake-CFD and Adjoint Method", *Aerospace Science and Technology* 28 (2013) 21-30

[32] Lenfers, C., "Propeller Design for future QESTOL Aircraft in the BNF Project", *30th AIAA Applied Aerodynamics Conference* 25-28 June 2012, New Orleans, Louisiana; AIAA 2012-3334

[33] Coton, F. N., Marshall, J.S., Galbraith, R. A. Mc D., Green, R. B., "Helicopter Tail Rotor Orthogonal Blade Vortex Interaction", *Progress in Aerospace Sciences* 40 (2004) 453-486

[34] Blazek, J., "*Computational Fluid Dynamics: Principle and Applications*", Elsevier, ISBN-13: 978-0-08-044506-9, ISBN-10: 0-08-044506-3, United Kingdom, 2005

[35] ANSYS, Inc., "ANSYS CFX-Solver Theory Guide", Release 12.0, ANSYS, Inc. Southpointe, 275 Technology Derive Canonsburg, PA 15317, ansysinfo@ansys.com, http://www.ansys.com, USA, April 2009

[36] Gausz, T., "Légcsavarok", Electronic Lecture Notes, Budapest, 31.01.2015., http://www.ara.bme.hu/oktatas/tantargy/NEPTUN/BMEGEATAKV4/ 2015-2016-I /ea/LEGCSAVAROK.pdf, 21.06.2015

[37] VZLU, ESPOSA Project, "WP6.2.4. BE2 Tractor Configuration: Propeller Characteristics", *Workshop Presentation*, 05.2015

[38] International Organization for Standardization, Standard Atmosphere, ISO

[38] 2533:1975, 1975

[39] Amato, M., Boyle, F., Eaton, J. A., and Gardarein, P., "Euler/Navier-Stokes Simulation for Propulsion-Airframe Integration of Advanced Propeller-Driven Aircraft in the European Research Programs GEMINI/APIAN"; *21st ICAS Congress* 13-18/09/1998, ICAS-98-5,10,2, Australia

[40] EVEKTOR, "Specification of Engine and Airframe Installation Requirements of ACC TR2", NO. ACP1-GA-2011-284859-ESPOSA

[41] Rwigema, M. K., "Propeller Blade Element Momentum Theory with Vortex Wake Deflection", *27th Congress of International Council of the Aeronautical Sciences* (2010) 19-24 September 2010, Nice, France Paper ICAS 2010-2.3.3

[42] Renaud, T., O'Brien, D., Smith, M., Potsdam, M., "Evaluation of Isolated Fuselage and Rotor-Fuselage Interaction using CFD", *American Helicopter Society 60th Forum*, Baltimore, MD, June 7-10, 2004

Detail Diversity Analysis of Novel Visual Database for Digital Image Evaluation

Jure Ahtik*, Deja Muck, Marica Starešinič

Faculty of Natural Sciences and Engineering, Department of Textiles, Graphic Arts and Design, Chair of Information and Graphic Arts Technology, University of Ljubljana
Snežniška 5, SI-1000 Ljubljana, Slovenia; jure.ahtik@ntf.uni-lj.si, deja.muck@ntf.uni-lj.si, marica.staresinic@ntf.uni-lj.si

Abstract: The evaluation of the visual quality of digital images is most commonly performed with various objective and subjective quality assessment methods. To calculate and analyse these methods, usually one of predefined image databases, e.g. TID2008 or TID2013, is used to compare an unmanipulated image with a manipulated one. When comparing quality assessment parameters to the communication value of images, a different, hi-resolution and more detail-oriented image database is required; therefore, a novel database for the evaluation of digital images was developed. Using detail coverage and color difference calculations, the research team designed a series of 30 color images with 28 manipulations that can be successfully used for determining the correlations among various quality assessment parameters, metrics and the communication value (ability to communicate) of digital images. The parameters that were used to manipulate images include sharpness, contrast, noise, compression, resizing and lightness (all were chosen based on real-life photography usage). Using RMSE (root mean square error), PSNR (peak signal to noise ratio) and SSIM index (structural similarity index) assessment methods, the influence of image details on quality parameters was calculated. The calculations demonstrate the importance of each parameter and its influence on the image visual quality. The results show a new way of understanding quality parameters and predicting which quality parameter is more important when the image is more or less complex. Complexity as a mathematical value is closely correlated to the content of an image. Hence, understanding the results of this research can help photographers and editors choose a more suitable digital image for publication. The benefits are not only theoretical, but can be applied instantly in real-life use.

Keywords: photography; image quality assessment; digital image evaluation; image quality parameters; RMSE; PSNR; SSIM; novel image database; visual database

1 Introduction

Images are nowadays the main source of information, as we first observe the image, and then decide if we are going to read the news or not. As a consequence, a large number of images has to be observed, analyzed and tested to decide, which is the best to use. The speed of the so-called image information is constantly on the increase, as more and more images or photographs, respectively, are being taken each second.

Editors, artists and photographers need more time to assess the large amount of image information. To make the process less complicated and time-consuming, the idea of quality assessment was created to determine which images do not apply to the basic quality parameters set by photographers and researchers.

The evaluation of digital images is most commonly performed by different objective and subjective quality assessment methods [1–3] The objective methods used in this research were *RMSE* (root mean square error), *PSNR* (peak signal to noise ratio) and *SSIM* index (structural similarity index). [4]

The *RMSE* (root mean square error) of predicted values, $\hat{y}_i$, for time, i, of a regression dependent variable, y_i, is computed for n different predictions as the square root of the mean of the squares of deviations as shown in Eq. (1).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1}$$

The PSNR (peak signal to noise ratio) of predicted values, $\hat{y}_i$, for time, i, of a regression dependent variable, y_i, is calculated for n different predictions. MAX_I is the maximum possible pixel value of the image and MSE stands for Mean Square Error, as used in Eq. (2).

$$PSNR = 10 \cdot log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 10 \cdot log_{10}\left(\frac{MAX_I^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}\right) \tag{2}$$

The SSIM index (structural similarity index) is calculated on various windows of an image. Eq. (3) shows the measure between two windows x and y of common size $N \times N$, where μ_x is the average of x, μ_y, is the average of y, σ_x^2, is the variance of x, σ_y^2, the variance of y, σ_{xy} is the covariance of x and y, $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ are two variables to stabilize the division with the weak denominator, and L the dynamic range of pixel-values, $k_1 = 0.01$ and $k_1 = 0.03$ by default.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3}$$

To develop and research these methods, one of predefined image databases, e.g. *TID2008* [5], is usually used. When comparing quality assessment parameters to the communication value of images, a different, more detail-oriented image database is required.

The image and video databases used in the quality assessment by Winkler [6] indicate that there are more than a dozen databases available in the public domain that are relevant to quality assessment, and very different research has been conducted with such a procedure. [7–12] A comparison of databases that are publicly available, using the same criterion can be used for testing quality assessment algorithms. The advantages and disadvantages of all tested quality metrics [12] also depend on the viewing conditions, as some researchers believe that controlled lab environment experimental conditions are essential [13], whereas others prefer naturally variable viewing conditions that users experience in their daily life [14], to collect realistic data.

With the overflow of visual information, people are exposed to photographs and images all the time, and as the research indicates, people are good at remembering pictures. [15] SUN dataset [16, 17] images were used in this research to determine the recall of images [18], which is important for advertising, designers and photographers.

In MIT, an algorithm [19] was created to predict the recall of photographs, how memorable or forgettable an image is, to be able to store the information people will most likely remember or forget. This research will help develop better communication systems, teaching resources, social media, as well as advertising and personal health assistant applications to help remember information. There are also researches being conducted on the image quality perception of different devices. [20]

This article focuses on a novel database for the evaluation of digital images that was developed for the purpose of objective evaluation of various image quality parameters. Using detail coverage (percentage of images that is covered with details) and color difference calculations, the research team introduced a series of 30 color images (Figure 1) that can be successfully used for determining the correlations between different quality assessment parameters, metrics and communication values (ability to communicate or successfully transfer message from transmitter to receiver) of digital images. [21] The images in the novel visual database are by about 34% more diverse and also cover a bigger color gamut than the images most commonly used in the Tempere Image Databases *TID2008* and *TID2013*, which contain 25 images distorted at five different levels with 24 types of distortions. [22, 23] The size of images is in both cases 512 × 384 pixels, mainly used for objective visual quality assessment.

Figure 1
All 30 images included in novel image database

2 Experiment

The aim of the experiment was to determine how and which image quality assessment parameters have the greatest influence on image quality.

2.1 Introducing Novel Image Database

First, a novel visual image database of 30 images was introduced. [24] The research team analyzed a new and improved image database of 30 images to investigate the area of image analysis from the photographer's point of view, not merely the mathematical or statistical perspective. *TID2008* and *TID2013* have been used in most research in this field until now; however, when measured and determined, these databases do not have enough detail and color diversity. Furthermore, the images in these databases do not have resolution that would be high enough (512 × 384 pixels) for further subjective testing (a novel visual database has 1920 × 1440 pixels). Considering all of the above, the team is determined that a novel visual image database offers a better foundation for the research.

Detail diversity is one of the most important factors when it comes to the communication value evaluation. Different approaches of image evaluation have been carried out [25, 26], however, for our purpose, detail diversity evaluation

was the most suitable. Image diversity is an attribute that is also important for the content-based image retrieval [27, 28]. A comparison between *TID2008* and the novel image database can be observed in Figure 2. From the average pixel value for each image, the research team calculated that the new visual image database is by 34% more diverse regarding the details. Details were visualized with ImageJ: each first image was converted to an 8-bit greyscale image and then the Threshold with 0–75 setting was applied. Counting the white pixels gave us the detail diversity of each image. The average pixel values ranged from 56 for the least detailed image to 253 for the most detailed image (Table 1). For comparison, the values for *TID2008* spread from 116–227.

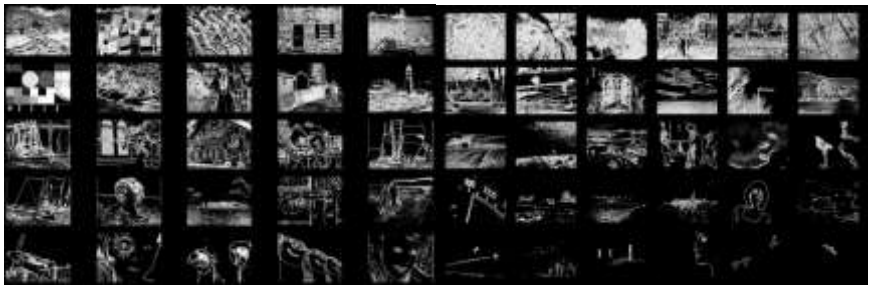


Figure 2
Visualization of details in images included in TID2008 (left) and novel image database (right)

Table 1
Average monochrome pixel value for each image in novel image database.
Higher number means more details.

Image number	Average monochrome pixel value	Image number	Average monochrome pixel value	Image number	Average monochrome pixel value
1	56,348	11	184,742	21	237,098
2	90,111	12	185,917	22	237,98
3	105,057	13	192,963	23	243,943
4	118,425	14	200,337	24	245,169
5	132,95	15	206,845	25	248,511
6	138,244	16	215,736	26	248,519
7	154,052	17	223,593	27	248,528
8	162,232	18	231,808	28	250,496
9	172,681	19	235,347	29	251,694
10	175,431	20	235,747	30	252,866

2.2 Selecting Image Quality Assessment Parameters

The parameters that have the most influence on the image communication value were specified and for each, a mathematical manipulation to simulate the real effect was selected. In this research, the team used the following:

- sharpness (Gaussian blur for decreasing and unsharp mask for increasing – unsharp mask was included as it is commonly used method by photographers: method cannot directly correct sharpness errors caused by the lens or processing, but it includes calculations that give us sharper results),
- contrast (lower contrast for decreasing and higher contrast for increasing),
- noise (poisson, salt & pepper and speckle noise – all for increasing),
- compression (jpeg and jpeg2000 compression levels for increasing),
- size (resizing),
- lightness (lower lightness for decreasing and higher lightness for increasing).

The parameters were chosen based on the direct influence of the digital camera and digital workflow on the digital image visual quality (Table 2).

Table 2
Influence of digital camera on visual quality

	sharpness	contrast	noise	compression	size	lightness
lens	✓	✓	–	–	–	✓
shutter	–	–	–	–	–	✓
sensor	–	✓	✓	–	–	✓
processing	✓	✓	✓	✓	✓	✓

2.2.1 Sharpness

The manipulation of sharpness was conducted in two ways – by decreasing and increasing:

- Decreasing was performed with Gaussian blur in three steps, using Matlab, function “fspecial”, parameter “Gaussian”, radius 5, 10 and 15, and sigma 5, 10 and 15.
- For increasing, sharpness unsharp masking was used in three steps, using Matlab, function “fspecial”, parameter “unsharp”, and radius 0.2, 0.5 and 1.0.

For each of the 30 images, three manipulations with decreased and three with increased sharpness were obtained.

2.2.2 Contrast

The manipulation of contrast was done in two ways – by decreasing and increasing:

- Decreasing was conducted using Matlab, function "imadjust", where the matrix parameter was manipulated with values 0.1, 0.2 and 0.3.
- Increasing was conducted using Matlab, function "imadjust", where the matrix parameter b was manipulated with values 0.4, 0.6 and 0.8.

For each of the 30 images, three manipulations with decreased and three with increased contrast were obtained.

2.2.3 Noise

The manipulation of noise was carried out in three different ways, all increasing noise in the image:

- Salt & pepper noise was applied in three steps using Matlab, function "imnoise", parameter "salt & pepper", and values 0.05, 0.10 and 0.20.
- Speckle noise was applied in three steps using Matlab, function "imnoise", parameter "speckle", and values 0.05, 0.10 and 0.20.
- Poisson noise was applied using Matlab, function "imnoise" and parameter "poisson".

For each of the 30 images, seven noise manipulations were obtained.

2.2.4 Compression

The manipulation of compression was performed in two ways, in both by increasing compression:

- Increasing compression in three steps using JPEG standard, Matlab, function "imwrite", parameter "Quality", values 50, 30 and 10.
- Increasing compression in three steps using JPEG2000 standard, Matlab, function "imwrite", parameter "QualityLayers", values 20, 10 and 5.

For each of the 30 images, we got six manipulations with increased compression.

2.2.5 Size

The manipulation of size was conducted first by decreasing the image size and then by increasing it back to the original size in three steps. Matlab, function "imresize", and values 0.90, 0.75 and 0.50 were used.

For each of the 30 images, we obtained three resized manipulations.

2.2.6 Lightness

The manipulation of lightness was done in two ways – by decreasing and increasing:

- Decreasing was conducted using Matlab, function "imadjust", and by manipulating matrix parameter d with values 0.4, 0.6 and 0.8.
- Increasing was conducted using Matlab, function "imadjust", and by manipulating matrix parameter c with values 0.2, 0.4 and 0.6.

For each of the 30 images, we got three manipulations with decreased contrast and three with increased contrast.

2.3 Database Structure

Applying each of the described parameters and manipulating saturation (which is not presented in this paper, as the team was only researching the complexity of images) in one to six levels in each image, the team developed 1140 different images, altogether called a novel image database. The image manipulation was conducted in Matlab R2014a and all the images were saved in the BMP file format with 1920 × 1440 pixel resolution, suitable for a subjective testing in further research.

2.4 Calculating Objective Image Quality

The next stage was to calculate the objective image quality, using different objective quality assessment methods, e.g. RMSE (root mean square error), PSNR (peak signal to noise ratio) and SSIM index (structural similarity index). These are the most commonly used methods for the visual quality analysis of monochrome images – we were mostly interested in detail diversity, thus, color information was not relevant for this research. The calculations were carried out by comparing the original (reference) or unmanipulated image with the manipulated one (for each of the 30 images in the database, 34 calculations were performed with each method).

3 Results and Discussion

Preliminary research showed significant advantages of the new novel visual image database, which can be used for objective and subjective testing. This database covers a significantly wider color range (34%) and also contains higher resolution images than *TID2008* and *TID2013*. It represents the possibility of testing different aspects of image quality and communication value, using the same image database during the whole process.

The selected images are based on human perception and experiences, and differ in the characteristics such as motive variation and detail coverage. A subjective testing provides accurate results now and hopefully also in the future. The results are practically oriented and the discussion also presents some direct instructions for photographers that are supported with calculations.

3.1 Examples of Image Manipulations

All 30 images in the novel image database were manipulated in different ways. Six examples can be seen in the image number 18, where only the most manipulated samples are presented: sharpness, noise, contrast, JPEG compression and lightness. (Figure 3)

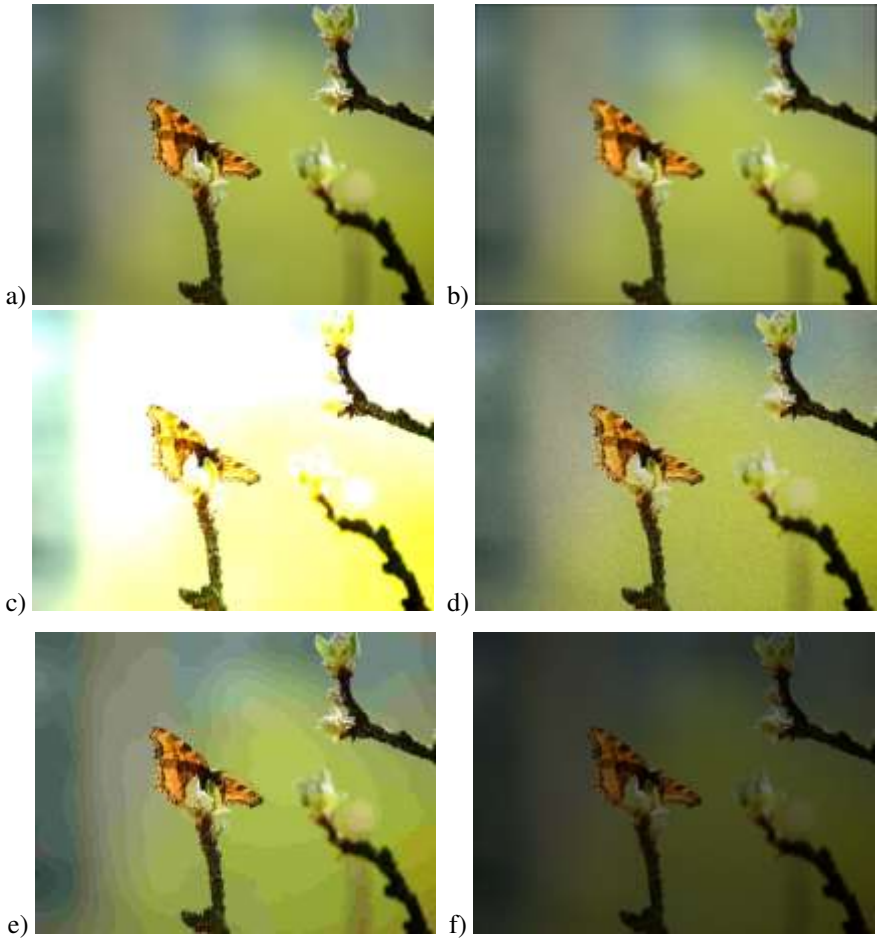


Figure 3
Image 18 from novel image database: a - unmanipulated, b - decreased sharpness, c - highest contrast, d - highest noise, e - highest JPEG compression, f - lowest lightness

3.2 Sharpness

In Figure 4, the relation between SSIM and the average pixel value of images with manipulated sharpness can be observed. A comparison of reference images with manipulated images indicates that a higher level of details in the image has a greater influence on the image quality when manipulating its sharpness (increasing or decreasing). The SSIM index results are spread across the total range, as it was expected. The smaller the details in the image, the greater the influence of blur or unsharp mask on its quality – there are more elements that can be changed according to the original. Therefore, it can be easily concluded that sharpness has a large influence on the image communication value. As a consequence, it is recommended for photographers to use good quality lenses and a short shutter speed.

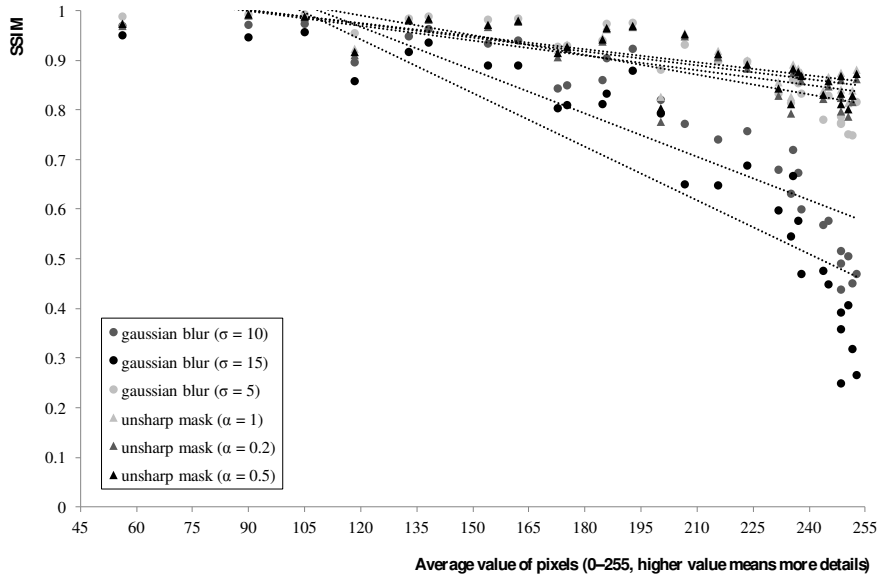


Figure 4
Influence of sharpness manipulation on image quality

3.3 Contrast

The relation between SSIM index and the average pixel value of images, presenting manipulated contrast, is shown in Figure 5, where it is demonstrated that a higher level of details in the image has a greater influence on the image quality when manipulating its contrast. The number of details offers more possibilities for the contrast changes to have a greater effect, which was also

expected. The *SSIM* range is not as wide as in sharpness manipulation; thus, it can be concluded that contrast changes have a smaller effect on the image quality than sharpness. Nevertheless, contrast is very important when it comes to image quality. Photographers are very dependent on their equipment and have a very small influence on the contrast itself in the production phase; the contrast should therefore be corrected in the post-production.

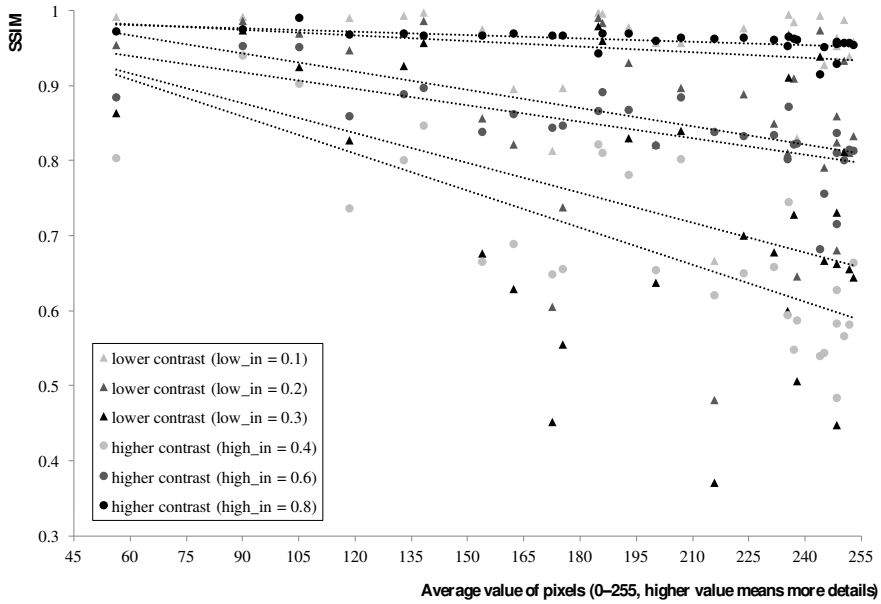


Figure 5
Influence of contrast manipulation on image quality

3.4 Noise

Noise is a common disadvantage of higher ISO sensitivities. The relation between SSIM index and average pixel value of images with manipulated noise is presented in Figure 6. The situation is very different than with sharpness and contrast: it can be seen that a lower level of details in the image has a greater influence on the image quality when manipulating its noise. The reason is that a higher number of details that is constructed out of more different image pixels offers a greater ability to hide noise, whereas noise can be easily seen on flat surfaces with very little pixel differences. To avoid noise, photographers should not use higher ISO sensitivities.

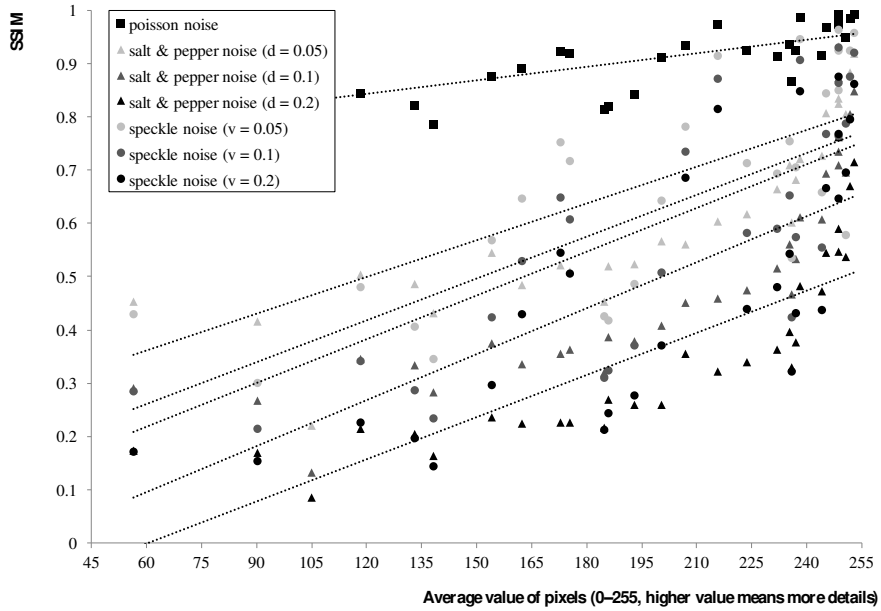


Figure 6
Influence of noise on image quality

3.5 Compression

Regarding compression, the team looked into the JPEG and JPEG2000 compression algorithms. The relation between SSIM index and average pixel value of images with manipulated compression is shown in Figure 7. The team established that the level of details in the image has no significant influence on the image quality when manipulating its compression. That does not mean that compression has no influence on the image quality, it actually has a very high influence. The increase in compression results in a lower image quality, whereas the number of details in the image does not really influence the result. A very low-level of compression is recommended for photographers.

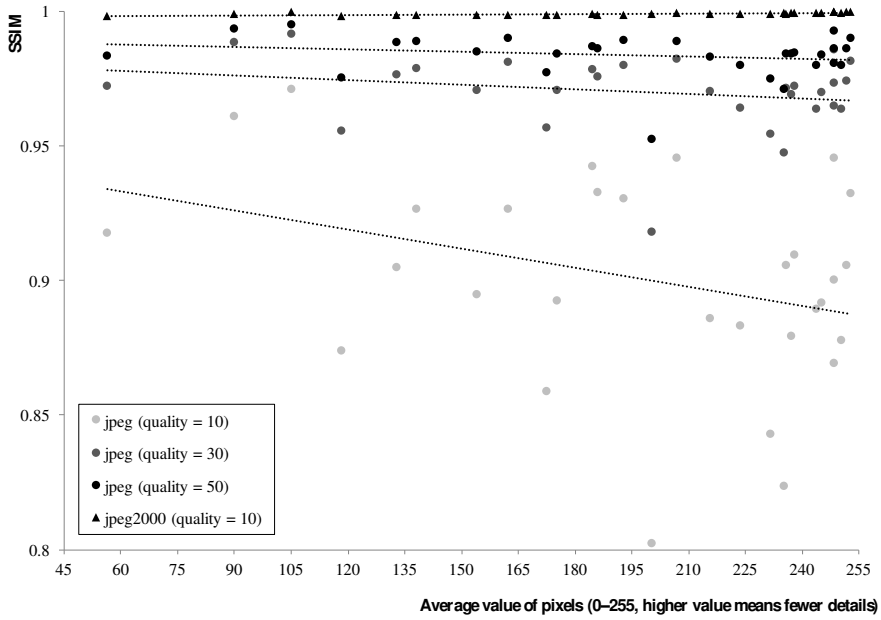


Figure 7
Influence of compression on image quality

3.6 Size

The size manipulation in the images was made by scaling the images down by 10%, 25% and 5%, and then reversed, scaling them back to their original resolution. The image quality drop was expected. The relation between SSIM index and average pixel value of images that have manipulated size is presented in Figure 8. In all cases, even with 10% manipulation, the image quality dropped significantly and a higher level of details in the image, influenced the image quality when manipulating its size. More details lead to more possibilities to losing information and a higher drop in image quality. Scaling up the images is not recommended if high image quality is desired.

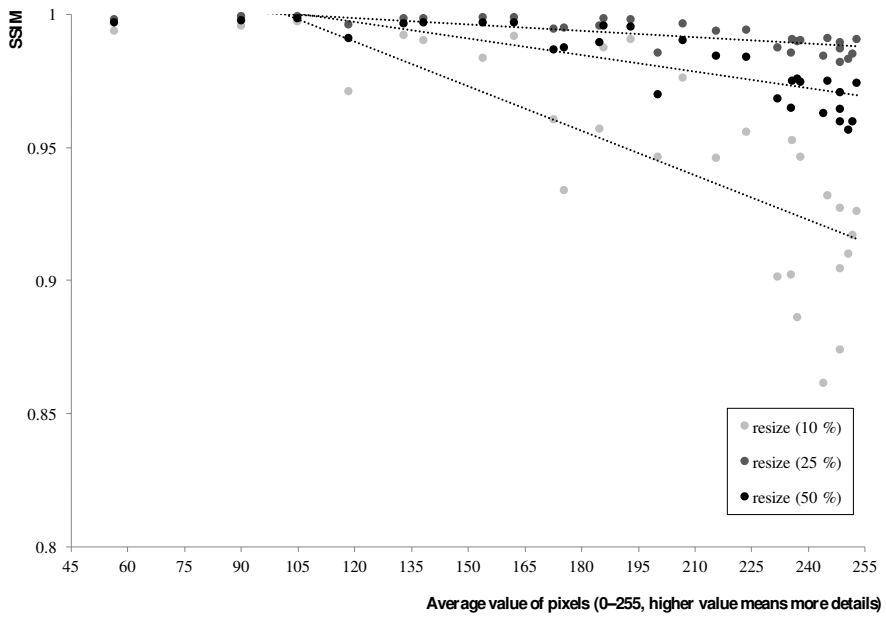


Figure 8
Influence of size manipulation on image quality

3.7 Lightness

There are few photography settings that influence image lightness, e.g. shutter speed, aperture size and ISO sensitivity. In contrast to other cases, a comparison between SSIM index and the level of details has not lead to any noticeable findings. The team therefore compared PSNR to the average image lightness. Figure 9 shows that the image quality drop is higher when lowering brightness on darker images or raising brightness on lighter images. The reason lies in the dynamic range, where the bit depth of the image does not allow the rendering of more details in very dark or very light areas. Photographers should always work with optimal photography settings.

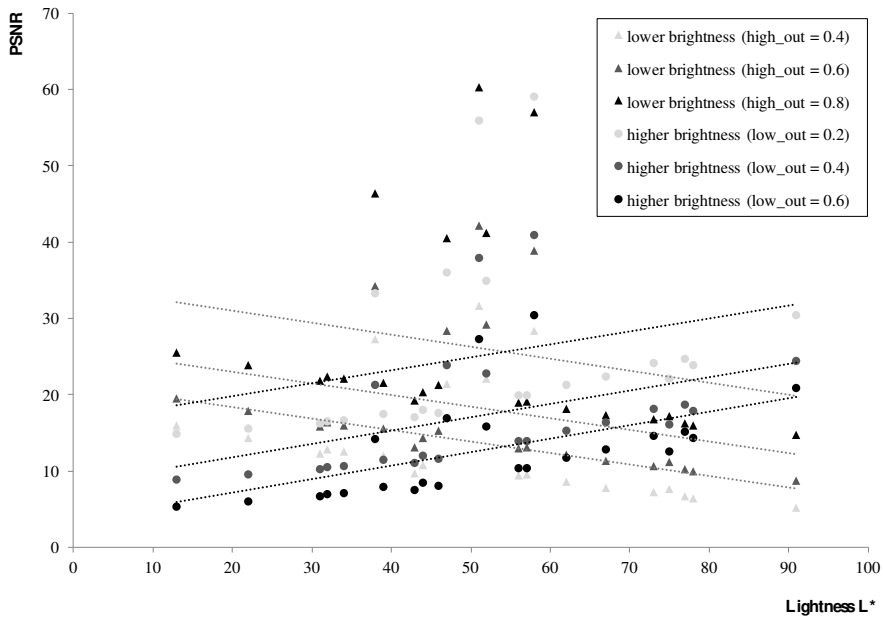


Figure 9

Influence of lightness on image quality:

average image lightness influences image quality when manipulating its brightness

Conclusions

This paper is focused on the presentation of evaluation of different image quality parameters. The influence of image complexity on the image quality parameters has been analysed and the conclusions are as follows:

- sharpness: more details in the image, the greater the influence of sharpness on its quality,
- contrast: more details in the image, the greater the influence of contrast on its quality,
- compression: more details in the image, the greater the influence of compression on its quality,
- size: more details in the image, the greater the influence of resizing on its quality and
- noise: less details in the image, the greater the influence of noise on its quality.

The more complex images are, in most cases more under the influence of the image quality decrease, so working with less complex images can be more flexible. Communication value is preserved when image has less communication elements and has been manipulated in the process. These conclusions are very important not only for researches but also for editors and other communication experts.

This research has also confirmed some of the well-known recommendations for photographers:

- regarding lightness, work in optimal photography settings,
- for increased sharpness, use good quality lenses and short shutter speed,
- to avoid noise, photographers should not use higher ISO sensitivities,
- contrast should be corrected in the post-production,
- scaling up the images is not recommended if high image quality is required; therefore, a very low-level of compression is recommended.

In further research, the novel visual database will be tested on different quality assessment metrics, using subjective testing methods and methods for measuring the image communication value (some methods are still to be developed). Subjective measurements will be performed with observation and eye movement measurement, and the team believes that the results will confirm the present research. The final goal is to have some real objective parameters from which usable results for the communication value prediction could be determined. The exact numbers and a comparison between different quality parameters are important for understanding the real world experience users have when observing images. Knowing that some quality parameters do not have such a substantial influence on the image quality than others can help editors decide what to include into their publications, or which images will have a better communication value. At the end of the research, the novel visual database will be publicly available for other researchers.

References

[1] Z. Wang, L. Qiang, "Information Content Weighting for Perceptual Image Quality Assessment", IEEE Transactions on Image Processing **20**, No. 5, 1185-1198 (2011)

[2] M. Cliff, et al., "Use of Digital Images for Evaluation of Factors Responsible for Visual Preference of Apples by Consumers", HortScience **37**, No. 7, 1127-1131 (2002)

[3] D. M. Chandler, S. S. Hemami, "VSNR: A Wavelet-based Visual Signal-to-Noise Ratio for Natural Image", IEEE Transactions on Image Processing **16**, No. 9, 2284-2298 (2007)

[4] P. Mohammadi, A. Ebrahimi-Moghadam, S. Shirani, "Subjective and Objective Quality Assessment of Image: A Survey", Majlesi journal of Electrical Engineering **9**, No. 1 (2015)

[5] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti, "TID2008 – a Database for Evaluation of Full-Reference Visual Quality Assessment Metrics," Advances of Modern Radioelectronics **10**: pp. 30-45 (2009)

[6] Stefan Winkler, "Analysis of Public Image and Video Databases for Quality Assessment", IEEE Journal on Selected Topics in Signal Processing **6**, No. 6 (2012)

[7] A. Bovik, "Handook of Image and Video Processing", Academic Press

(2000)
[8] Z. Wang, A. Bovik, H. Sheikh and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity", IEEE Transactions on Image Processing **13**, no. 4 (2004)
[9] N. Ponomarenko, F. Battisti, K. Egiazarian, J. Astola, V. Lukin, "Metrics performance comparison for color image database", Proc. of the 4th International Workshop on Video Processing and Quality Metrics for Consumer Electronics (2009)
[10] H. R. Sheikh and A. C. Bovik, "Image Information and Visual Quality", IEEE Transactions on Image Processing **15**, 430–456 (2006)
[11] N. Ponomarenko, M. Carli, V. Lukin, K. Egiazarian, J. Astola, F. Battisti, "Color Image Database for Evaluation of Image Quality Metrics", Proc. of the International Workshop on Multimedia Signal Processing, 403 (2008)
[12] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazariank, J. Astola, "Locally Adaptive Image Filtering based on Learning with Clustering", Proc. of Image Processing: Algorithms and Systems IV **5672**, 94-105 (2005)
[13] C. Keimel, T. Oelbaum, and K. Diepold, "Improving the Verification Process of Video Quality Metrics," in Proc. International Workshop on Quality of Multimedia Experience (QoMEX), San Diego, CA, pp. 121-126 (2009)
[14] N. Staelens et al., "Assessing Quality of Experience of IPTV and Video on Demand Services in Real-Life Environments," IEEE Transactions on Broadcasting **56**, No. 4: pp. 458-466 (2010)
[15] P. Isola, et al. "What Makes an Image Memorable?" Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 145-152 (2011)
[16] SUN Dataset, http://groups.csail.mit.edu/vision/SUN, accessed September 2016
[17] J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba," Sundatabase: Large-Scale Scene Recognition from Abbey to Zoo", In IEEE Conference on Computer Vision and Pattern Recognition (2010)
[18] P. Isola, et al. "Understanding the Intrinsic Memorability of Images", Advances in Neural Information Processing Systems, 2429-2437 (2010)
[19] "Deep-Learning Algorithm Predicts Photos' Memorability at "Ear-Human" Levels", MIT News, online: http://news.mit.edu/2015/csail-deep-learning-algorithm-predicts-photo-memorability-near-human-levels-1215
[20] R. H. Hamid, A. C. Bovik, "Image *Information and Visual Quality",* IEEE Transactions on Image Processing **15**, No. 2, 430-444 (2006)
[21] J.-M. Sung, B.-S. Choi, Y.-H. Ha, "Comparative Display Image Quality Evaluation based on an Analytic Network Process for Mobile Devices", Journal of Imaging Science and Technology **60**, No. 2, 020501 (2016)
[22] R. Iii, "Color Image Database TID2013: Peculiarities and Preliminary Results Tampere", University of Technology, Tampere, Finland, Media Communications Lab, USC Viterbi School of Engineering, USA, 106-111 (2013)

[23] Ponomarenko, et al. "Color Image Database TID2013: Peculiarities and Preliminary Results." Visual Information Processing (EUVIP), 2013 4th European Workshop on. IEEE (2013)

[24] J. Ahtik, T. Muck, M. Starešinič, "A Novel Database for Evaluation of Digital Images", V: URBAS, Raša (ur.). Proceedings, 7th Symposium of Information and Graphic Arts Technology, Ljubljana, 5-6 June 2014, In Ljubljana: Naravoslovnotehniška fakulteta, Oddelek za tekstilstvo, 206-210 (2014)

[25] T. Wei, "An Evaluation of Digital Image Correlation Criteria for Strain Mapping Applications", Strain **41**, No. 4, 167-175 (2005)

[26] Y. Naci, "Documentation of Cultural Heritage using Digital Photogrammetry and Laser Scanning", Journal of Cultural Heritage **8**, No. 4, 423-427 (2007)

[27] M. Azodinia, A. Hajdu, "A Novel Combinational Relevance Feedback Based Method for Content-based Image Retrieval", Acta Polytechnica Hungarica, **13**, No. 5, 121-134 (2016)

[28] S. P. Mathew, V. E. Balas, K. P. Zachariah, "A Content-based Image Retrieval System Based On Convex Hull Geometry", Acta Polytechnica Hungarica, **12**, No. 1, 103-116 (2015)

Configuring Genetic Algorithm to Solve the Inverse Heat Conduction Problem

Sándor Szénási, Imre Felde

Óbuda University, Bécsi út 96/b, H-1034 Budapest, Hungary
E-mail: szenasi.sandor@nik.uni-obuda.hu, felde.imre@nik.uni-obuda.hu

Abstract: Accurate design of heat treatment operations requires the knowledge of the Heat Transfer Coefficients (HTC), which quantity can be determined by performing Inverse Heat Transfer Calculations. The novel approaches for the estimation of HTC are based on heuristic optimisation methods, but the usage of these techniques raises several questions. In the case of genetic algorithms, there are not any rules of thumb for selecting the appropriate population size, mutation rate, stopping condition, and similar. The most efficient way to fine-tune these parameters is to run thousands of experimental tests and evaluate the results. However, in the case of inverse heat conduction, this has not been a viable option because of the high computational demand of fitness calculation which leads to a runtime of dozens of years. This paper presents a solution to this problem using a novel data-parallel direct heat conduction problem solver method implemented on multiple graphics accelerators. The ~100× speed-up achieved by this parallel algorithm made it possible to finish the necessary experimental tests in 15 weeks (instead of 29 years). Data gathered during these experiments are directly useful in practice. Based on these, it is possible to make recommendations for optimal genetic algorithm configuration parameters.

Keywords: inverse heat conduction problem; genetic algorithm; parameter optimisation; population size; mutation rate; stopping condition; graphics accelerators; CUDA

1 Introduction

It is a well-known fundamental experience that material properties are not constantly determined by their chemical composition, but they are influenced by their microstructure [1]. Heat treatment (heating up the object to a specific temperature and cooling it down under strict temperature control) is one of the most efficient methods to produce the desired microstructure of the material.

The proper design of the heat treatment process requires an accurate knowledge of the thermal boundary conditions, including the Heat Flux (HF) or the Heat Transfer Coefficient (HTC). HTC describes heat exchange between the surface of

an object and the surrounding medium. The determination of HTC faces a typical Inverse Heat Conduction Problem (IHCP). The IHCP methods are using the temperature signals recorded and estimated by simulations at given locations of the work-piece in order to predict HTC functions. Several IHCP approaches are based on optimization methods, where the goal function has to be minimised is given as the deviation of the measured and predicted temperature data [2], [3].

It is usual to solve this problem with some kinds of heuristic search algorithms, like Genetic Algorithms (GAs) [4]–[6] or Particle Swarm Optimisation (PSO) [4], [7]–[9]. In the case of GAs, every chromosome encodes one possible HTC function. Using the Direct Heat Conduction Problem (DHCP) calculations, it is feasible to generate the theoretical thermal history based on each HTC variants; and it is also possible to compare this theoretical history to the real measured temperature values. Our goal is to find the HTC function generating the cooling curve, which is the most similar to the measured one.

Beyond the generally known challenges of these heuristic methods (stability, unpredictable convergence, and others), the applications raise several technical issues:

- It is difficult to choose the appropriate variant for a practical task [10].
- Moreover, after selecting the suitable method, there are not any unequivocal rules to set the best working parameters (population size, mutation probability, and stopping condition) [11], [12].

The proposed methodology is based on an evaluation of virtual tests generated by a huge variety of different configurations (including population size, mutation probability/rate, and stopping condition). Due to the high computational requirements of IHCP solvers, parallel processing DHCP is suggested. A novel data-parallel DHCP solver has been developed and implemented as a GPU application. The recommendations for the appropriate genetic algorithm configurations are given by the results of the computational experiments.

The rest of this paper is structured as follows: Section 2 contains details of the problem formulation based on the GA approach, and Section 3 presents the issues raised by fitness calculation and a brief presentation about the already mentioned GPU based DHCP solver implementation. The final section focuses on the methodology followed by the experimental tests and the conclusions.

2 Methodology

2.1 Genetic Algorithms

GA is one of the most popular biologically inspired methods based on the language of natural genetics and biological evolution. As a heuristic search method, it uses a set of individuals (chromosomes) referred to as a population. Every chromosome represents a potential solution for the raised problem. Chromosomes encode the corresponding potential solution in their genes as numbers. After a random initialization, these genes change according to a previously fixed rule set when trying to find the optimal solution for the original problem.

A full GA search needs several iterations, where every iteration consists of the same main steps. Randomly selected pairs (given by the selection operator) of individuals are mated using a process of crossover (using the crossover operator). As a result of these steps, new individuals inherit genes from either parent. To help the algorithm map as large a part of the parameter space as possible, these individuals undergo a random mutation, in which some of their genes change by a random amount (specified by the mutation operator).

An essential part of the method is the fitness calculation. This assigns a comparable value to every chromosome based on their genes. This fitness value plays a major role in the selection phase (individuals with better fitness value usually have a higher probability of being selected). It is also common to use these fitness values to set up some form of stopping condition. For example, the GA stops when the increase of the best fitness value slows down.

2.2 Problem Formulation

2.2.1 Chromosome Representation

Each chromosome represents a possible solution for the problem. There are several methods for chromosome representation (how the information is encoded into the genes) based on the problem itself (input data characteristics – type, amount, structure, and similar.) and the chosen implementation.

In the case of 2D IHCP, the feasible solutions are the potential HTC functions. These are two-dimensional functions, where the actual value depends on the local coordinate and the time (Eq. 1).

$$HTC_{z,t} = ?\left|0 \le z \le L, 0 \le t \le t_{end}\right. \tag{1}$$

Where

- z – local coordinate;
- L – length of the work object;
- t – time;
- t_{end} – time period of the experiment.

Although it is a continuous function, the discrete DHCP process uses some simplifications:

- Local coordinate discretization: if the DHCP process uses an $n \times m$ sized matrix to simulate the heat movement inside the work-piece (see Section 3.1), only the positions 0, L/m, $2L/m$, …, L are used as local coordinates.
- Time discretization: the DHCP process starts the simulation at time point 0 and executes an elementary heat-transfer step for every dt second. Therefore, it will need the HTC values at time coordinates 0, dt, $2dt$, …, t_{end}.

Theoretically, it is possible to encode all of these HTC values as separate floating point values in the genes, but this leads to an unmanageably large number of parameters. To significantly decrease the size of the parameter space, it is enough to store fewer parameters as control points and use bilinear interpolation to estimate the necessary values:

- For local coordinate discretization, we store the HTC values for only Z number of local coordinates (where $Z < m$). These positions are fixed in advance, conveniently according to the positions of thermocouples built inside the real-world object.
- A dynamic resolution model is used to reduce the number of necessary time coordinates. This means that we use K control points (according to different time moments) in each location; and as a novel idea, the number of K varies during the search. In the first phase, K equals to 3 to find a rough estimation. When the genetic algorithm cannot fine-tune the best solution with this setting, it increases the value of K to 5 and continues the search. One search contains 4 phases, using $K_1=3$, $K_2=5$, $K_3=9$, $K_4=17$ values. Consequently, every chromosome encodes Z×K control points; where every control point consists of two floats: time and HTC values (location values are fixed in advance).

2.2.2 Initialization

The first step of any heuristic algorithm is to initialize the elements (chromosomes, particles). As an essential concept, it is assumed that there is no prior information about the characteristics of the searched HTC function.

Therefore, the only possible way is to generate random numbers for the initial gene values based on the following rules:

- genes containing HTC values should have a random value between 0 and 7000 (W/m^2/K);
- genes containing time values should have a random value between 0 and 180 (sec).

To find the ideal number of chromosomes in a population is one of our goals. As a free parameter, we will reference this number as $|P|$. Section 4.2.1 presents the details of the tests related to this value.

2.2.3 Elitism, Selection and Crossover Operators

Because of the large search space, we would like to guarantee that the best individuals will survive. To ensure this, the elitism technique [13] is used to move the top 10% of the chromosomes (having the best fitness values) to the next generation without crossover or mutation. As a side-effect, the fitness values for the best chromosomes become monotonically increasing.

In the next step, the selection operator is responsible for randomly choosing individuals from the population as parents. Individuals with better fitness values will be selected with a greater probability than others (chromosomes involved in elitism are still able to be selected as parents). Our implementation uses fitness proportionate selection (also known as the roulette wheel selection), where the fitness level of chromosomes is used to associate a probability of selection (Eq. 2).

$$p_i = \frac{\left| f_i - f_{best} \right|}{\sum_{j=1}^{|P|} \left| f_j - f_{best} \right|} \tag{2}$$

Where

- p_i = probability to select the i-th chromosome for crossover;
- f_i = fitness value for the i-th chromosome;
- f_{best} = the best fitness value of the population;
- $|P|$ = number of chromosomes.

When preparing the chromosomes to the next generation, it is necessary to select two different ones based on the given probabilities and to use the crossover operation on these to create one offspring for the next generation. The uniform crossover operation [14] is used on the selected parents, which means that each gene (in this particular case, a float value) of the offspring is randomly picked from either of the two parent genes from the same position (the probability of inheriting from either of the parents is 50-50%). This means that there is no linkage between genes, the inheritance of these is independent from the others.

We must repeat the presented selection and crossover steps until the next generation of $|P|$ instances has been created.

2.2.4 Mutation Operator

To ensure exploration, it is necessary to use some kinds of mutation operators. Accordingly, every newly created (by selection and cross-over) chromosome of the next generation goes through an additional random process which can modify its genes. The proposed algorithm uses gene level mutations.

This means that each gene of the new chromosome can be changed with a certain probability. It is hard to determine the appropriate probability rates and extent of the changes. Finding these is one the primary goals of this research.

In all experiments, we use three levels of mutations:

- Large mutation:
  - probability: M_L $(0 \leq M_L \leq 1)$,
  - rate of change: random value between $-R_L \ldots +R_L$.
- Medium mutation:
  - probability: M_M $(0 \leq M_M \leq 1)$,
  - rate of change: random value between $-R_M \ldots +R_M$.
- Small mutation:
  - probability: M_S $(0 \leq M_S \leq 1)$,
  - rate of change: random value between $-R_S \ldots +R_S$.

Section 4.2.2 presents the details of the tests relating to these values.

2.2.5 Stopping Criteria

The stopping criteria give the answer to the question raised at the end of every iteration - whether it is worth continuing the search or not. Obviously, if one of the chromosomes reaches the theoretically best fitness value, then it contains the desired solution, and it is necessary to stop the search. However, due to the nature of heuristic algorithms, there is no guarantee that this is going to happen. Therefore, this criterion is insufficient.

Another approach is to wait until the GA cannot produce any progress with respect to the best fitness of the population. One of the characteristics of this technology is that in most cases, the best fitness value of the population converges to the desired fitness value more and more slowly. Because of the unique dynamic resolution model, it also requires investigation than when it is worth to switch to the next phase.

Setting up these limitations is hard, so, the implementation uses two free parameters to handle this: the search will start the next phase when the achieved improvement of the best fitness value during the last $STOP_{iter}$ number of iterations is less than $STOP_{rate}$.

Section 4.2.3 presents the details of the tests relating to these values.

3 GPU-based Fitness Calculation

3.1 Fitness Definition

The result of the fitness function depends on the difference between the generated thermal history (using the DHCP solver and the HTC value encoded into the chromosome) and the measured temperature signals. This section presents the details of this DHCP solver.

A two-dimensional axis-symmetrical model is considered to estimate the temperature distribution in a cylindrical work-piece. The mathematical formulation of the nonlinear transient heat conduction problem can be described as follows (Eq. 3):

$$\frac{\partial}{\partial r}\left(k\frac{\partial T}{\partial r}\right)+\frac{k}{r}\frac{\partial T}{\partial r}+\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)+q_v=\rho C_p\frac{\partial T}{\partial t} \tag{3}$$

With the following initial and boundary conditions (Eq. 4-5):

$$T(r,z,0)=T_0 \tag{4}$$

$$k\frac{\partial T}{\partial z}\bigg|_{\substack{0\le z\le L\\ r=R}}=HTC(z,t)\left[T_q-T(r,z,t)\right] \tag{5}$$

Where

- r, z – local coordinates;
- t – time;
- R – radius of the workpiece;
- ρ – density of the object;
- T_0 – initial temperature of the workpiece;
- T_q – temperature of the cooling medium;
- $T(r,z,t)$ – temperature of the workpiece at given location/time;

- $k(T)$ – thermal conductivity (varying with temperature);
- $C_p(T)$ – heat capacity (varying with temperature);
- $HTC(z,t)$ – heat conduction (varying with local coordinate and time).

The solution of (Eq. 3) is obtained by a weighted Schmidt explicit finite difference method.

The fitness value of the individual is determined as the deviation between the measured and generated thermal history. Based on these values and the results of the explicit finite difference method, the fitness value for a given HTC is (Eq. 6):

$$F = \sum_{k=1}^{N} \left(T_k^m - T_k^c \right)^2 = \min \tag{6}$$

Where

- N –total number of measured temperatures (the number of points multiplied by the number of measurements at each point);
- T_k^m – measured values;
- T_k^c – calculated values.

Our goal is to find the best HTC with minimal fitness value.

3.2 GPU-based Implementation

By using the finite difference method, it is possible to calculate the heat transfer between two points for a given small time step. To be able to ensure accuracy, a sufficiently small-time interval is necessary ($dt \leq 0.01$ sec). According to real-world measurements, it is usually required to continue the simulation of the cooling process for an extended period ($t_{end} \geq 120$ sec). For this purpose, the algorithm has to run the calculations mentioned above in a loop to specify the heat movement between the finite items for each time steps.

As visible, one DHCP calculation needs a lot of iterations (iteration count $\geq 120 / 0.01 = 12000$). Using a traditional CPU-based sequential algorithm, it takes about 0.24 sec to calculate all the necessary T_k^c values and to specify the fitness value for a given chromosome.

According to the GA operating rules, we have to calculate the fitness value for every chromosome in every iteration. To achieve our goals, we have to launch hundreds of GA searches and to observe the effects of the different parameter configurations. In hindsight, we know that in practice this needs about $3.76 * 10^9$ fitness calculations. For a single core CPU algorithm, the estimated run-time for the whole examination is about 28.7 years. This is obviously unacceptable.

To speed-up this process, a graphics accelerator based implementation has been designed. Nowadays, Graphics Processing Units (GPUs) are highly paralleled devices containing thousands of processing elements and applicable for general purpose numerical computing. This leads to enormous processing power, but it is required to adapt the already existing sequential algorithms to massively parallel ones to utilise these resources fully.

It is easy to see that the proposed method is well-parallelizable and applicable for adaptation to a data-parallel fashion. The DHCP algorithm solves the same differential equation for all finite elements of the grid (with different input data). This data-parallel fashion is ideal for GPU implementation: one GPU thread is assigned to one element of the finite grid, and it is responsible for computing the temperature changes for this referenced location. In the case of a 10×34 sized grid (n = 10, m = 34 because of optimisation reasons), this needs 340 threads running the same function to calculate the heat movement for a given time interval.

3.3 Higher Level Parallelization

Running 340 threads on a modern GPU is not enough to fully utilise its processing power. In the case of NVidia GTX Titan Black cards, the number of cores is 2880. Thereby, executing 340 parallel threads leads only to very low (used cores / all cores = 340 / 2880 = 11.8%) theoretical occupancy (the practical utilisation is even worse).

It is worth noting that the proposed DHCP algorithm is a part of the fitness calculation process. During the GA, it is necessary to calculate the fitness for all chromosomes at the end of all iterations. In the case of 100 or more individuals, the number of parallel threads becomes 340x1000 = 34000 or more. This is enough parallelism to design and implement an efficient GPU-based implementation. We used the CUDA framework [15] and NVidia graphics cards for this purpose.

As a result of further optimisations, a novel data-parallel algorithm had been developed [16] with multi-GPU support [17] to speed-up the fitness calculations, and it is also possible to use all GPU devices and CPU cores together. By using two GPUs and four CPU cores, the run-time is about 100× less than the run-time of the original sequential method.

This implementation makes it possible to run thousands of GA searches within a reasonable period (14.5 weeks instead of 28.7 years) to evaluate all configurations to be examined.

4 Experimental Results

This paper follows the following terminology:

- Heat transfer simulation – using the DHCP solver to generate the 2D temperature history (T_k^c) based on the given parameters (HTC, material, and cooling medium attributes).
- Fitness calculation – calculating the fitness value (F) for a chromosome encoding an HTC. This takes the following steps: 1) running a heat transfer simulation using the given HTC values; 2) comparing the resulting thermal history to the reference curve.
- Iteration – one iteration of the GA. This takes the following steps: 1) selection; 2) crossover; 3) mutation; 4) fitness calculation for each chromosome.
- Search – the execution of a full GA process using a given configuration (population size, mutation rate, stopping condition). Main steps: 1) create initial population; 2) execute iterations; 3) stop the execution if one of the stopping conditions becomes true.
- Session – run several searches using the same configuration and evaluate the results (best-achieved fitness, an average of best fitnesses of all searches, average iteration count).
- Experiment – run several sessions using different configurations. 1st experiment: population size test; 2nd experiment: mutation probability test; 3rd experiment: stopping condition test.

4.1 Methodology

The estimation of the optimal population size, mutation probability and stopping condition for the proposed GA solving the IHCP is outlined. Due to the random behaviour of the genetic algorithms, it is difficult to define the most efficient configuration (the consecutive searches launched with the same parameters would give different results).

Several studies focusing on the evaluation of heuristic search algorithms suggest empirical validation of the parameters [10], [18]–[20]. According to the random behaviour of heuristics, it is usually not enough to run one search per configuration. It is necessary to execute as many examinations as possible and gather the following data:

- Best fitness value – the best-achieved fitness of the entire population;

- Number of iterations – as a secondary objective, we would like to find a parameter set, which is not just accurate but also fast; therefore, the number of iterations taken by the GA is important.
- Number of fitness function calculations – the number of iterations do not determine the required run-time. It is mostly based on the number of fitness function calculations (FFC), which is the multiplication of iteration count and population size.

Unlike many papers [20], we are not dealing with direct run-time. It is an architecture specific measure, and it is very hard to compare the results of different systems. We prefer the analysis of the number of FFC requirements because this is by far the most resource intensive part of the algorithm. The cost of one fitness function evaluation (running the DHCP solver) is independent of the actual HTC values. Therefore, the comparison of the number of these function calls is a good substitution for a platform independent run-time analysis.

The FFC count is also important to determine the end of a session testing a given configuration. It is necessary to run many simulations using the same configuration, but it is essential to set a limit for the number of these. It is common to set up a run-time limit, but as mentioned before it is unstable and platform specific. It is also unfair to limit the number of iterations because GAs with small population size needs fewer computations per iterations.

As a solution, the FFC count becomes the limiting factor. Every session testing a given configuration has a limit for FFC number (actually 50,000,000 calculations per session). The testing framework starts new GA searches one after the other monitoring the number of FFC count. When this accumulated number exceeds the predetermined limit, it starts the next session using the next configuration.

In the literature, there is no consensus on how to evaluate the efficiency of heuristics. Some papers [21] deal only the best fitness values found by a given configuration. Obviously, smaller fitness values mean better results, but according to the random behaviour of GAs, the comparison is based only on the best results found in all configurations is unsatisfactory. It does not give a clear estimation of the expected future performance. We followed the methodology of several papers [12], [22] comparing the average fitness values found by all GA searches of a session. Where required, we performed statistical tests to analyse the raw results.

4.2 Experimental Results

4.2.1 Population Size

We ran several sessions using the following parameters:

- Population size: 100, 200, 300, …, 2000

- Mutation rate: M_L=0.l; R_L=0.001;M_M=0.01;R_M=0.01;M_S=0.001;R_S=0.1
- Stopping condition: $STOP_{iter}$ = 30; $STOP_{rate}$ = 0.01

Table 1 and Fig. 1 show the experimental results.

Table 1

Best and average fitness values and the average iteration count of GAs with given population size. FFC count is equal to population × iteration.

Population	Best fitness	Avg fitness	Avg iteration	Avg FFC count
100	1936.2	4162.3	190	19000
200	699.4	5339.3	510	101959
300	480.5	4289.6	733	219982
400	259.3	3659.0	937	374824
500	307.8	3328.3	976	487932
600	239.1	520.2	1792	1075340
700	296.5	434.2	1927	1349176
800	249.8	374.3	1891	1513153
900	255.9	356.2	1798	1618403
1000	223.7	311.3	1859	1859185
1100	229.4	311.8	1846	2030116
1200	221.8	285.7	1807	2168800
1300	236.2	301.5	1745	2268795
1400	226.8	288.5	1689	2365236
1500	207.0	264.4	1708	2562000
1600	216.7	262.7	1663	2660716
1700	164.5	254.7	1655	2813122
1800	221.1	252.9	1621	2916900
1900	203.9	249.4	1680	3191169
2000	225.1	262.0	1578	3156750

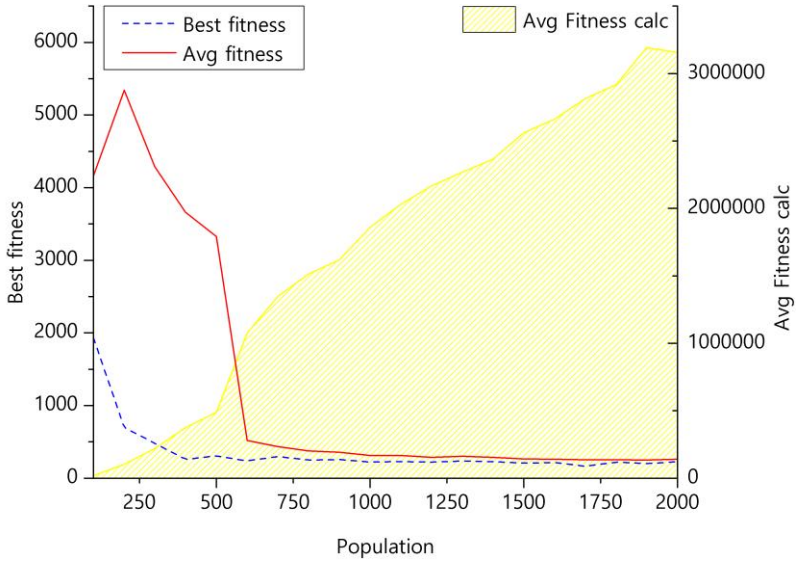


Figure 1

The solid red line shows the average fitness values, and the dashed blue one shows the best fitness values of GAs with the given population size. The yellow area shows the average FFC count.

In the case of small population sizes, the results are not satisfactory. There are not enough chromosomes to ensure convergence to a valid solution. These simulations are usually stopped in an early phase (sometimes without any progress). As can be seen in Fig. 1, where there is a significant improvement near population size 600. From 700 to 1500 the results become even better, but this trend slows down. Both the best fitness and the average fitness values become similar for larger population sizes.

The average number of necessary iterations is decreasing, but because of the higher population size, the number of FFCs (which requires the most computation effort) is increasing. Our first priority is accuracy, but efficiency is also important. Therefore, we should find the point with the smallest population size (and computation effort) after which there is no significant accuracy improvement.

Because of the spread of the results, the naïve comparison of the best fitness and average fitness values are not sufficient. One-way ANOVA tests were run using α=0.05 significance level on the results of all searches [22], [23]. The null hypotheses was that the expected value of the average fitness is the same for all population sizes between P and 2000. In the case of small P values ($P \leq 1300$), it is obviously not true. When P=1400, the result of the ANOVA is: F=2.56 while $F_{critial}$=2.17; therefore, we have to reject the null hypotheses ($F > F_{critial}$) weakly. However, in the case of P=1500: F=0.55 and $F_{critial}$=2.30, the test shows ($F < F_{critial}$) that it is very likely that the expected value of the average fitness is the

same for both groups. In the case of larger P values, the results are similar. Thus, we can state that it is not worthwhile using a larger population size than 1500 because these sessions do not give a significant increase in accuracy, but have higher computational demand.

In the literature, it is also common to use two-tailed t-tests to compare the results of different parameter sets [21], [24]. By using these tests, the final verdict is the same. In comparing the results of the simulation with population size 1400 to the simulation using 2000 chromosomes, the t-test indicated some differences between the expected value of average fitness (t_{stat} = 2.085 and $t_{critical}$=2.028, $t_{stat} > t_{critical}$ - therefore, we have to reject the null hypothesis). Nonetheless, in the case of P=1500 (t_{stat} = 0.184 and $t_{critical}$=2.032, $t_{stat} < t_{critical}$) and larger P values, the t-tests show that the expected average fitness is the same as for population size 2000.

Based on these, our recommendation for population size is $\left| P \right|$=1500.

4.2.2 Mutation Probabilities

The purpose of the second experiment was to find the optimal mutation probabilities. Several sessions were run using the following parameters:

- Population size: 1500 (based on the result of the 1st experiment);
- Mutation rates: M_L=p/1000; R_L=0.01; M_M=p/2000; R_M=0.05; M_S=p/10000; R_S=0.25; where p = 0, 1, 2, … 9;
- Stopping condition: $STOP_{iter}$ = 30; $STOP_{rate}$ = 0.01.

In the case of GAs, it is always required to find the proper balance between exploration and exploitation ability of the search algorithm. Mutation is mostly responsible for the exploration part.

Table 2

Best and average fitness values and the average iteration count of GAs with given mutation probability. FFC count is equal to population × iteration.

p value	Best fitness	Avg fitness	Avg Iteration	Avg FFC count
0	534.0	772.7	292	438491
1	134.7	171.3	1918	2877333
2	119.0	152.9	2114	3170719
3	125.6	151.6	1988	2981559
4	122.9	149.6	1991	2985882
5	124.7	152.5	1862	2792417
6	127.7	155.0	1851	2777000
7	117.5	158.3	1758	2637632
8	137.9	158.6	1710	2565225
9	129.1	155.3	1686	2528775

The experimental results (Table 2) show the expected behaviour based on the literature. In the case of too small mutation rates, the exploitation failed. Chromosomes cannot get away from a local optimum. In the opposite case, too large mutation rates caused an "over randomised" search. These searches were more like a random search than a well-balanced GA. Both extremes led to poor performance.

Due to its nature, the analysis of the optimal mutation rate is simple compared to the population size analysis: p=4 gives the best average fitness values, and both lower and higher mutation rates give worse results.

According to this, the recommended mutation rate is M_L=0.004; R_L=0.01; M_M=0.002; R_M=0.05; M_S=0.0004; R_S=0.25.

4.2.3 Stopping Criteria

The primary aim was to find the parameter set which gives the highest accuracy (lowest fitness value). As an effect of the used elitism technique, the fitness value is monotonically decreasing during the search. Therefore, it is evident that if only accuracy is taken into account, it is worth running the algorithm as long as possible.

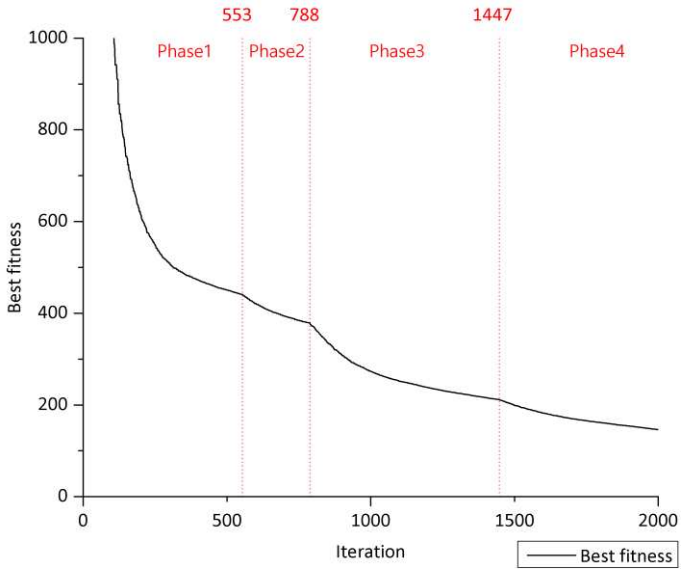


Figure 2

Best fitness values by iterations for a given GA search. Vertical lines show the phase switching points. It is visible that all phases have the same characteristic.

However, a novel technique dynamically changing the number of control points is also used. Fig. 2 shows the fitness values by iteration number for a given GA execution. The common characteristic of these phases is well visible: every phase starts with a fast decreasing period and some iterations later this slows down. This raises an additional question. At which point is it worth stepping over to the next phase (and using more control points)?

If this happens too late, the full GA convergence becomes slow. Early phases use only a few control points, which has a significant limitation in describing the HTC function. When the GA reaches this limit, the decrease of the fitness value becomes almost negligible. It is not worth leaving the algorithm to fine-tune these results because it would be more efficient to change to the next phase which has fewer limitations.

It is also worth avoiding the opposite situation and switching to the next phases too early. First, notice that this dynamic resolution model is an essential part of the algorithm. Without this ($STOP_{iter}$=0 in Table 3) the GA cannot start to converge (in the case of 340 starting parameters, the search space is too large). Too early phase switching leads to similar problems: the algorithm will not be able to converge ($STOP_{iter}$=1 in Table 3), or it converges, but starts the last phase with a relatively poor fitness value, and it takes many iterations to improve this.

Table 3

Fitness values after the given number of iterations using different $STOP_{iter}$ parameters. The first test ($STOP_{iter}$=0) does not use the dynamic resolution method.

$STOP_{iter}$	**10000**	**20000**	**30000**	**40000**	**50000**	**60000**
0	1855.68	1619.75	1444.06	1318.32	1217.46	1122.83
1	4375.95	4002.08	3798.09	3616.72	3475.45	3276.50
10	67.25	61.74	59.76	58.66	57.71	57.09
50	64.72	59.31	57.38	56.29	55.66	55.25
100	62.10	57.30	55.44	54.39	53.76	53.38
200	108.40	53.12	51.33	50.40	49.84	49.31
300	213.28	52.95	50.95	50.00	49.18	48.65
500	213.71	86.29	51.69	49.12	48.19	47.60
600	212.72	100.53	53.37	48.38	47.28	46.61
700	222.54	122.04	63.93	48.87	47.86	47.23
900	221.47	173.51	83.78	53.30	48.61	47.40
1000	228.76	174.47	85.74	52.03	48.83	47.77

Table 3 shows the experimental results. The GA had to finish all phases (except the last one) when the improvement of the best fitness value was less than 1% ($STOP_{rate}$=0.01) in the last $STOP_{iter}$ iterations. There was no similar limitation for the final phase, and the GA was left running for 60000 iterations.

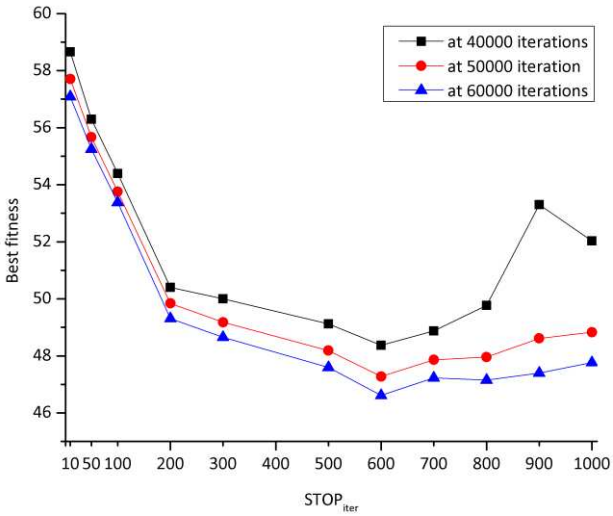


Figure 3

Best achieved fitness values by $STOP_{iter}$ parameter values after given iteration numbers. Blue triangles show the final results (after 60000 iterations), black rectangles show a middle-state after 40000 iterations, red circles show the state between them (50000 iterations).

As is visible from the table and Fig. 3, the optimal value was 600. Lower parameters give poor final fitness. A slight weakening is also visible in the case of higher numbers. It is not significant, but it is worthwhile seeing that without this increment (and considering the fitness values identical for these parameters), we should recommend the same because of the faster convergence. The secondary objective is to find the fastest parameters, and it is visible that GA with $STOP_{iter}$=600 finds the best fitness values earlier.

Based on the results, the recommended stopping criteria is $STOP_{iter}$=600.

Conclusions

Using GA to solve the IHCP is already a known procedure. Nevertheless, the best configuration parameters were unknown to use this method efficiently, and the high computation demand makes it impossible to determine these using experimental test.

The determination of GA's configuration parameters providing the highest efficiency to solve a 2D axis-symmetrical IHCP problem is outlined. A computational framework was developed by which thousands of GA searches have been analysed and the performance of several configurations (population size, mutation probability, stopping condition) has been evaluated.

As a final result, the following recommendations are made:

- Population size: $|P|$=1500

- Mutation rate: M_L=0.004; R_L=0.01; M_M=0.002; R_M=0.05; M_S=0.0004; R_S=0.25
- Stopping condition: $STOP_{iter}$=600

Using these parameters, stable and efficient GA searches can be expected without wasting resources for unnecessary computations.

The most advanced part of the framework is the hybrid (CPU and GPU) parallel DHCP solver module. One of the design concerns was the ability to use this module with other search methods, like PSO, or Fireworks. As a further plan, a request to extend the research with the investigation of these heuristics is made.

It is also possible to improve the efficiency of the already existing DHCP solver. Using more than two graphics cards can linearly improve the computing performance, and the new NVLINK™ technology developed by NVIDIA provides improved GPU-to-GPU link bandwidth and tight integration with IBM Power CPU makes it possible to decrease the memory transfer deficits significantly.

Acknowledgements

We acknowledge the financial support of this work by the Hungarian State and the European Union under the EFOP-3.6.1-16-2016-00010 project and Hungarian-Japanese bilateral Scientific and Technological (TÉT_16-1-2016-0190) project. The authors would like to thank NVIDIA Corporation for providing graphics hardware for the GPU benchmarks through the CUDA Teaching Center program. We also would like to thank IBM Systems for the temporary access to their IBM NVIDIA Acceleration Lab which allows us to run some additional experiments using their new NVLink™ capable Power system.

Supported BY the ÚNKP-17-4/I. New National Excellence Program of the Ministry of Human Capacities.

References

[1] P. Oksman, S. Yu, H. Kytönen, and S. Louhenkilpi, “The Effective Thermal Conductivity Method in Continuous Casting of Steel,” *Acta Polytech. Hungarica*, vol. 11, no. 9, 2014.

[2] O. M. Alifanov, *Inverse Heat Transfer Problems*. Springer, 1994.

[3] J. V. Beck, B. Blackwell, and C. R. J. St. Clair, *Inverse Heat Conduction*. New York: Wiley, 1985.

[4] M. J. Colaço, H. R. B. Orlande, and G. S. Dulikravich, “Inverse and optimization problems in heat transfer,” *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 28, no. 1, pp. 1–24, 2006.

[5] M. N. Özisik and H. R. B. Orlande, *Inverse Heat Transfer: Fundamentals and Applications*. Taylor & Francis, 2000.

[6] I. Felde, *Estimation of thermal boundary conditions by gradient based and genetic algorythms*, vol. 729. Trans Tech Publications, 2013.

[7] S. Vakili and M. S. Gadala, "Effectiveness and Efficiency of Particle Swarm Optimization Technique in Inverse Heat Conduction Analysis," *Numer. HEAT Transf. PART B-FUNDAMENTALS*, vol. 56, no. 2, pp. 119–141, 2009.

[8] I. Felde, S. Szénási, A. Kenéz, S. Wei, and R. Colas, "Determination of complex thermal boundary conditions using a Particle Swarm Optimization method," in *5th International Conference on Distortion Engineering*, 2015, pp. 227–236.

[9] I. Felde and S. Szénási, "Estimation of temporospatial boundary conditions using a particle swarm optimisation technique," *Int. J. Microstruct. Mater. Prop.*, vol. 11, no. 3/4, pp. 288–300, 2016.

[10] M. A. Panduro, C. A. Brizuela, L. I. Balderas, and D. A. Acosta, "A comparison of genetic algorithms, particle swarm optimization and the differential evolution method for the design of scannable circular antenna arrays," *Prog. Electromagn. Res. B*, vol. 13, pp. 171–186, 2009.

[11] D. E. Goldberg and M. Rudnick, "Genetic Algorithms and the Variance of Fitness," *Illinois Genet. Algorithms Lab. Rep.*, vol. 5, no. 91001, pp. 265–278, 1991.

[12] Y. R. Tsoy, "The influence of population size and search time limit on genetic algorithm," *Sci. Technol. 2003. Proc. KORUS 2003. 7th Korea-Russia Int. Symp. (Volume3)*, no. 1, pp. 181–187, 2003.

[13] M. Mitchell, "An introduction to genetic algorithms," *Comput. Math. with Appl.*, vol. 32, no. 6, p. 133, 1996.

[14] G. Syswerda, "Uniform Crossover in Genetic Algorithms," in *Proceedings of the 3rd International Conference on Genetic Algorithms*, 1989, pp. 2–9.

[15] NVIDIA, "CUDA C Programming Guide." 2014.

[16] S. Szénási, I. Felde, and I. Kovács, "Solving One-dimensional IHCP with Particle Swarm Optimization using Graphics Accelerators," in *10th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics*, 2015, pp. 365–369.

[17] S. Szénási and I. Felde, "Heat Transfer Simulation using GPUs," in *20th IEEE Jubilee International Conference on Intelligent Engineering Systems*, 2016, pp. 263–267.

[18] T. Weise, Y. Wu, R. Chiong, K. Tang, and J. Lässig, "Global versus local search: the impact of population sizes on evolutionary algorithm

performance," *J. Glob. Optim.*, no. March, pp. 1–24, 2016.

[19] K. Mills, J. Filliben, and A. Haines, "Determining relative importance and effective settings for genetic algorithm control parameters," *Evol. Comput.*, no. x, 2015.

[20] R. H. Abiyev and M. Tunay, "Experimental Study of Specific Benchmarking Functions for Modified Monkey Algorithm," *Procedia Comput. Sci.*, vol. 102, no. August, pp. 595–602, 2016.

[21] D. Whitley, S. Rana, and R. B. Heckendorn, "The island model genetic algorithm: On separability, population size and convergence," *J. Comput. Inf. Technol.*, vol. 7, pp. 33–47, 1999.

[22] I. Rojas, J. González, H. Pomares, J. J. Merelo, P. a. Castillo, and G. Romero, "Statistical Analysis of the Main Parameters Involved in the Design of a Genetic Algorithm," *Syst. Man, Cybern. Part C Appl. Rev. IEEE Trans.*, vol. 32, no. 1, pp. 31–37, 2002.

[23] P. A. Castillo-Valdivieso, J. J. Merelo, A. Prieto, I. Rojas, and G. Romero, "Statistical analysis of the parameters of a neuro-genetic algorithm," *IEEE Trans. Neural Networks*, vol. 13, no. 6, pp. 1374–1394, 2002.

[24] A. Silva, A. Neves, and E. Costa, "An empirical comparison of particle swarm and predator prey optimisation," *Artif. Intell. Cogn. Sci.*, pp. 1–45, 2002.

Influence of Stress State Conditions on Densification Behavior of Titanium Sponge

Ivan Berezin[1,2], Anton Nesterenko[1], Alexandr Zalazinskii[1], George Kovacs[3]

[1] Institute of Engineering Science, Ural Branch, Russian Academy of Sciences
Komsomolskaya Str. 34, 620049 Ekaterinburg, Russia
berezin@imach.uran.ru, nav@imach.uran.ru, agz@imach.uran.ru
[2] Ural Federal University, Mira Str. 19, 620002 Ekaterinburg, Russia
i.m.berezin@urfu.ru
[3] Computer and Automation Institute, Hungarian Academy of Sciences,
Kende u. 13-17, H-1111 Budapest, Hungary, györgy.kovacs@sztaki.hu

Abstract: The paper discusses the material compaction process under various types of stress conditions based on the simulation of the multiaxial compression process of powdered titanium sponge. The finite element model of the multiaxial compression process made it possible to apply both radial and axial pressure independently from one another. Titanium sponge subjected to reversible thermo-hydrogen processing was used as the research material. For the description of the material plasticity condition, the modified Drucker–Prager Cap plasticity model was used, implemented in Abaqus.

Keywords: stress state; multiaxial compression; titanium sponge; metal powder; modified Drucker–Prager Cap plasticity model; representative unit cell; micromechanical model; finite element simulation

1 Introduction

The intensification of the compaction mechanism of metal powders heavily depends on the choice of an effective deformation scheme, providing the reduction of the compaction forces, at the increase in the density of the compacted work-piece. It is known [1–4] that achieving high densities of a deformable porous solid is only possible in the processes that implement optimal combinations of normal and tangential pressures.

In this connection, one should expect that the problem of the increasing density in a work-piece may be solved based on a complex analysis of stressed state effect on compaction and improvement of existing load distribution schemes. At that, the correlation between the compacting pressure and the material density mostly

relates to those deformation schemes, for which it was obtained. Thus, finding a consistent pattern of changing the material density depending on the value of shear and normal stresses at various deformation schemes is a relevant objective.

The most complete research of the regular pattern of material compaction may be based on the use of triaxal (multiaxial) compression processes [5–7]. This type of experimental equipment must provide different stress conditions in a work-piece by a possibility to apply both radial and axial pressure independently. The complexity of experiments and low effectiveness of triaxial compression installations make this approach expensive and inefficient.

The widespread industrial technology for the production of titanium blanks and articles includes the manufacture of titanium sponge, which is a high-purity noncompact titanium. Further processing of spongy titanium is characterized by large losses of metal in the form of recurrent and irretrievable losses. In addition, the technology of production of titanium blanks and other products is a multistage one, and is characterized by high energy and labor costs. For example, according to an assessment [8] in the cost of a 25 mm titanium sheet the sponge represets only 25 per cent.

Powder metallurgy methods and additive technologies allow to radically increase the utilization of metal raw materials and to reduce production costs. At the same time, the titanium sponge and powder compositions can be the commercially most available base material for such technologies. Manufactured products can be used in medicine, transport engineering, production of consumer goods, architecture, in desalination plants, and it also may be used as pipeline fittings in chemistry, petrochemistry, food industry, etc.

The possibility of manufacturing titanium products with the required properties, obtained by solid-phase consolidation technology of a titanium sponge, is shown in numerous studies [9–16]. The favorable effect of preliminary thermohydrogen treatment for the compactibility of a titanium sponge at certain temperatures has been first studied and shown in [17, 18].

In the present work a finite element (FE) simulation of the process of multiaxial compression of powdered titanium sponge is carried out. The purpose is the identification of material compaction quantitative pattern, where the pattern depends on the combination of equivalent pressure stress and von Mises equivalent stresses.

2 Unit Cell and Yield Criterion

The powdered titanium sponge subjected to reversible thermo-hydrogen treatment was used as material for the study (Figure 1). The hydrogenating was carried out by way of thermal diffusion in a Sieverts type vacuum machine. Titanium

saturation up to the concentration of 0.5 wt% of hydrogen at the compaction process temperature of 325 °C makes it possible to seriously reduce the compaction forces [17]. At that, the hydrogen alloying is temporary, and after the operation of plastic deformation, the hydrogen is removed by vacuum annealing.

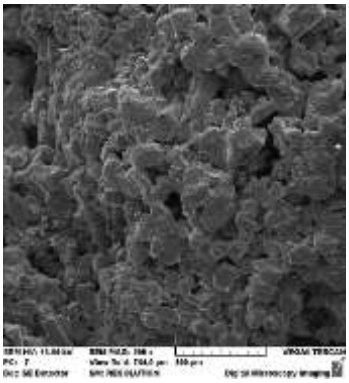


Figure 1
Fragment of titanium sponge structure

The process of material compaction was studied at two scale levels.

At the macrolevel scale, the powdered titanium sponge is entirely homogenous, isotropic, plastically compressible continuum solid. The Drucker–Prager Cap plasticity model (MDPC) has been applied for the description of the titanium sponge yield criterion [19].

$$f_1 = q - d - p\tan\beta = 0,$$

$$f_2 = \sqrt{(p-c)^2 + \left(\frac{R}{1+\alpha-\alpha/\cos\beta}q\right)^2} - m = 0, \tag{1}$$

$$f_3 = \sqrt{(p-c)^2 + \left[q - \left(1-\frac{\alpha}{\cos\beta}\right)\frac{n}{\alpha}\right]^2} - n = 0$$

where q – von Mises equivalent stresses, p – mean normal pressure, d – cohesion of the material, β – angle of friction of the material, R – parameter that controls the shape of the cap, α – coefficient used to define a transition yield surface f_2 (assumed to be equal to 0.05). The functions of the material condition c, m and n may be presented as:

$$c = \frac{p_b - Rd}{1 + R\cdot\tan\beta},$$

$$m = R(d + c\cdot\tan\beta), \tag{2}$$

$$n = \alpha\left(d + c \cdot \tan\beta\right)$$

where p_b is the hydrostatic compression yield stress.

The identification of the parameters of the assumed plasticity model involves a series of experiments [20–23]. The methodological complexity of their implementation leads to the attempts of finding more simple ways of plasticity curve identification. In particular, the MDPC yield curve may be approximated with a function having less internal variables, and this approach may be used for the identification of the initial model parameters. For approximation of the initial yield curve, the following equation has been used:

$$q = \gamma\left[\sqrt{2\left(p^*\right)^2 + 1/4} - \left(p^*\right)^2 - 1/2\right]^{1/2} \tag{3}$$

where p^* – mean normal pressure, presented in dimensionless form ($p^* = p/p_b$); γ – the coefficient, bringing the equation (3) into proximity with the modified Drucker–Prager cap yield curve.

Figure 2 shows a qualitative comparison of the geometric interpretation of the (MDPC) plasticity model and the curve built in accordance with the equation (3) in the q–p plane. It is clear that the curves intersect the following points:

A – isostatic decompression yield stress;

B, – lying on the crossing of the straight-line portion of the MDPC curve and the curve in accordance with the equation (3);

C – evolution limit that represents the volumetric inelastic strain driven hardening/softening;

D – isostatic compression yield stress.

Using an additional curve for the identification of the MDPC plasticity model allows only two experiments: die compaction and isostatic compression. At that, it is assumed that the initial particle cohesion is rather small and the yield strength in isostatic decompression is assumed to be closed to zero. Therefore, the material is loose. It is worth mentioning that this work considers only the compaction process; consequently, the stress condition in the deformed material will correspond to the points lying inside the elliptical portion, where the curve set by equation (3) and the Drucker–Prager curve (1) are close to one another. This makes it possible to conclude that an approximating curve may be used as a technique for determining the MDPC model coefficients.

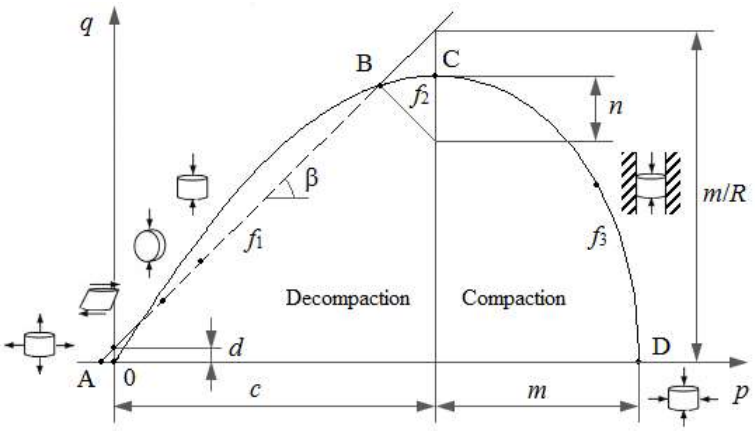


Figure 2

Qualitative comparison of the geometric interpretation of the modified Drucker–Prager cap plasticity model (dashed line) and the approximating curve in accordance with equation (3) (solid line)

Along with the reduction of the number of necessary points for the formig of the plasticity curve, there is a possibility to change the labor consuming natural experiments for the methods based on the theory of micromechanical deformation (microlevel scale) of a representative unit cell [24–26]. The works [27–32] are devoted to choosing a form, particle packing schemes, and to study their interaction and deformation. The results showed a well-founded choice of a cubic array of identical spheres for the definition of metal powder micromechanical model.

In this study, the unit cell is supposed to consist of a single spherical particle with a central pore circumscribed by a cube (Figure 3). The arrangement of a powder has been chosen based on the bulk density of titanium sponge in the initial state.

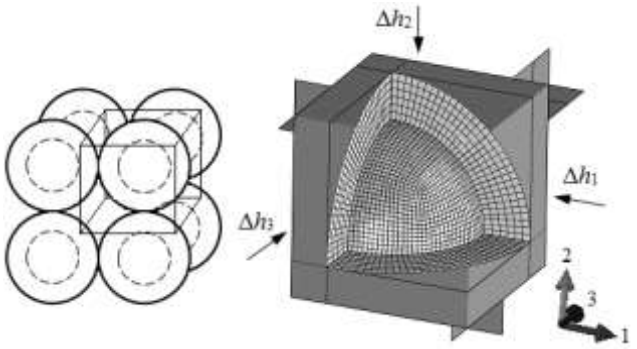


Figure 3a and Figure 3b

Arrangement of powder (3a) and FE model of 1/8 of unit cell (3b)

It is assumed that the considered material has a regular structure and there are no effects related to pulsatory oscillation, rotation and translational motion of the particles. It is important to establish plastic yield stress, shape and void volume changes for the introduced material model. Due to the statistical uniformity of properties, it is sufficient to solve the task only for one representative unit cell. It should be noted that in this work, the material porosity formed by the spherical particle packing, shown in Figure 3a, is understood only as void volume between the particles.

The FE solver Abaqus/Standard was used. The boundary conditions of the particle are the following: nodes at 1–3 plane have U2=UR1=UR3=0 prescribed, nodes at 1–2 plane have U3=UR1=UR2=0 prescribed, nodes at 2–3 plane have U1=UR2=UR3=0 prescribed. The curved surfaces of the particle are traction free. The representative unit cell was deformed by the displacement of boundary condition (Δh_i) fully rigid surfaces, imitating the influence of neighbor particles around the cell. For the isostatic (hydrostatic) compression, $\Delta h_1 = \Delta h_2 = \Delta h_3$ and for the case of die compaction, $\Delta h2 = \Delta h3 = 0$. The particle diameter $d_p = 2$ mm. The choice of particle size for numerical simulation is justified by the position of the statistical mechanics of structurally inhomogeneous materials, according to which the ratio $d_c/d_p \geq 10$, where d_c – container diameter, d_p – particle diameter, must be implemented. For the experimental study, the titanium sponge was screened with an average particle size of 2 mm.The relative density of material at the chosen particle packing diagram is $\rho_{rel} = 0.52$. The Young's modulus $E = 112$ GPa, the Poisson's ratio $\nu = 0.34$. The contact between the particle and the rigid surfaces is modeled with a contact pair. The rigid surfaces are modeled as analytical rigid surfaces. The mechanical interaction between the contact surfaces is assumed to be frictionless. The titanium strain-hardening curve has been taken from [33].

Figure 4 shows the distribution of von Mises equivalent stresses in a particle of the representative unit cell at isostatic compression (Figure 4a) and die compaction (Figure 4b) for porosity $\theta = 17.5\%$.

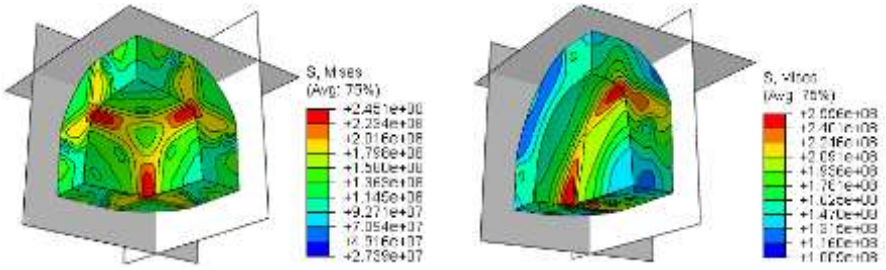


Figure 4a Figure 4b

Distribution of von Mises equivalent stresses in a particle of the representative unit cell at isostatic compression (4a) and die compaction (4b)

In the case of plastic deformation of particles, the material flows into free space (pores) between them and the point contacts under the load develop into contact surfaces. The volumetric plastic deformation (ε_v^{pl}) was established in accordance with the equation

$$\varepsilon_v^{pl} = \ln(V_0/V), \tag{4}$$

where $V_0 = 1$ – representative unit cell initial volume, V – its volume after deformation.

Based on the obtained values of the components of the stress tensor σ_1, σ_2, σ_3 acting on the unit cell contact areas, the values of von Mises equivalent stress (q) and equivalent pressure stress (p) were determined for two cases of stress condition of the representative unit cell.

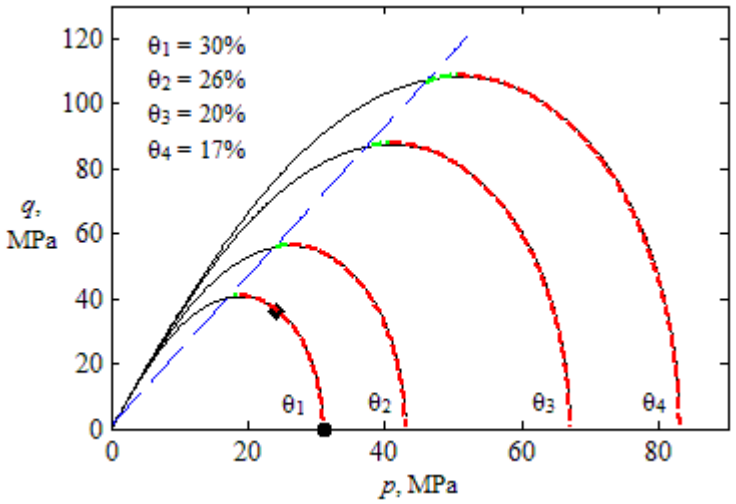


Figure 5
Evolution of the modified Drucker–Prager cap plasticity curve in the p–q plane

In Figure 5, the sign "●" corresponds to the stress condition of isostatic compression, and the sign "♦" – to the stress condition of die compaction. The approximating curve built along the defined points (the solid line in Figure 5) allowed the identification of coefficients of the MDPC model for titanium sponge: $\beta = 66.5°$; $R = 0.276$; $\varepsilon_v^{pl}|_0 = 0$; $\alpha = 0.05$; $d = 1$ MPa.

It should be noted that the hardening of material will inevitably cause changes in the determined coefficients. Nevertheless, the present work uses a default setting of the MDPC plasticity model implemented in Abaqus. Consequently, the coefficients found for specific porosity (θ) will be valid at any level of compaction.

3 Multiaxial Compression Simulation

Figure 6 shows the scheme of the multiaxial compression process.

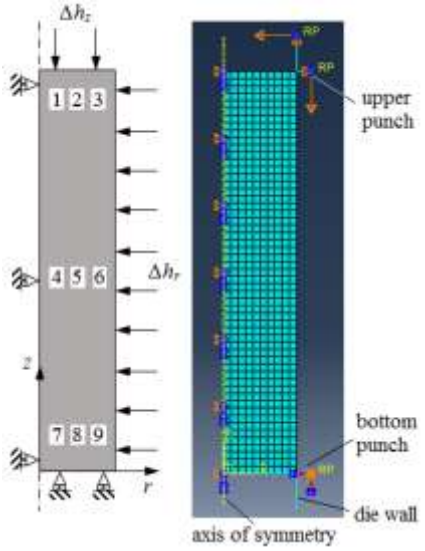


Figure 6
Computational scheme and FE model

The FE model is axisymmetric and includes the half of the billet. In Abaqus/Standard elements of type CAX4R, a 4-node bilinear axisymmetric quadrilateral with reduced integration are used. All parts of the die are fully rigid. The kinematic boundary conditions are symmetric on the axis (nodes at $r = 0$ have $u_r = 0$ prescribed).

The work-piece deformation takes place under hard stress conditions: the upper punch and die wall are displaced by Δh_i using a displacement boundary condition. The bottom punch is fully constrained. The radial pressure is produced on the billet due to the displacement of the matrix wall Δh_r in the radial direction. The height of the billet in initial state $H = 56$ mm, the radius $R = 10$ mm. The given initial porosity of titanium sponge $\theta = 48\%$. The indicated initial porosity of the material has been chosen based on the experimentally established bulk density of the powder used with an average particle size of a titanium sponge ≈ 2 mm. The properties of material are taken at the temperature of 325 °C.

At the first stage, the work-piece was subjected to non-uniform multiaxial compression. The displacement values Δh_z and Δh_r were changed randomly, independently from one another, which made it possible to provide various schemes of the stress state condition.

Moreover, it assists to identify the corresponding values of volumetric plastic deformation (ε_v^{pl}), von Mises equivalent stresses (q) and equivalent pressure stress (p) in areas No. 1-9.

Compression for each scheme of stress state condition continued until porosity (θ) close to zero was achieved in one of the billet areas. In connection with the fact that Abaqus/Standard solver, applied for the FE analysis, by default does not calculate the porosity (θ), the following equation was used:

$$\theta = (1 - \rho_{rel} \cdot \exp(|\varepsilon_v^{pl}|)) \times 100\%. \quad (5)$$

At the second stage of the research, the sample was only subjected to axial compression by way of moving the upper punch to Δh_z value with a fixed position of the matrix wall (Δh_r = const = 0). The deformation was stopped upon achievement of quasimonolythic state in one of the work-piece portions. The material porosity was computed according to the equation (5). The corresponding q and p values in areas No. 1–9 on various stages of deformation were determined with the help of the embedded post-processor Abaqus/Viewer. The third stage of experiments included the radial compression of the billet without the punch axial movement (Δh_z = const = 0). The values of q, p and ε_v^{pl} were determined in the same way as during the experiments of the first and the second stage. A more detailed description of the modeling of the multiaxial compression simulation is given in the paper [18].

4 Model Verifiation and Discussion

As a result of the FE simulation, the following regression equation was obtained

$$\theta = b_1 \cdot \exp(-b_2 \cdot k_q) + b_3 \cdot \exp(-b_4 \cdot k_p). \quad (6)$$

Here $k_q = q/q_s$, $k_p = p/q_s$, where q_s is the yield stress of material in a nonporous condition. The coefficients in equation (6) have the following meaning: $b_1 = 6.222$, $b_2 = 1.585$, $b_3 = 43.21$, $b_4 = 1.142$. The graphical interpretation of equation (6) is given in Figure 7.

It is clear that the value change of equivalent pressure stress, expressed by the coefficient k_p, produces a more substantial effect on the compaction of material as compared with shear deformation. Nevertheless, the absence of shear deformations in case of a stress condition of uniform isostatic compression ($k_q = 0$) does not allow achieving less than 6% porosity. On the other hand, only shear stresses ($k_p = 0$) do not lead to any substantial reduction of material porosity, whereas the change of θ is not more than 8%. Consequently, production of a quasimonolithic material with a minimal porosity is only possible at the synchronizing action of hydrostatical and shear deformations. When the $k_q = 2$ and $k_p = 4$ are achieved, the porosity is close to zero and any further increase of the coefficient values does not lead to any noticeable compaction of the material.

In order to verify the regression equation (6) obtained by the FE simulation, an experimental research of die compaction of titanium sponge was carried out.

The work-pieces were produced at the pressure of 1000 MPa on upper punch and the temperature of 325 °C. For the determination of porosity, the thin plates in various axial section of work-pieces were produced.

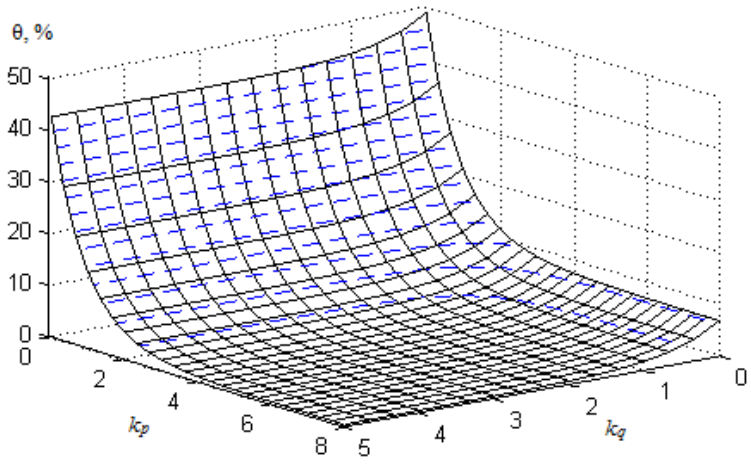


Figure 7

Dependency of porosity on the k_q and k_p

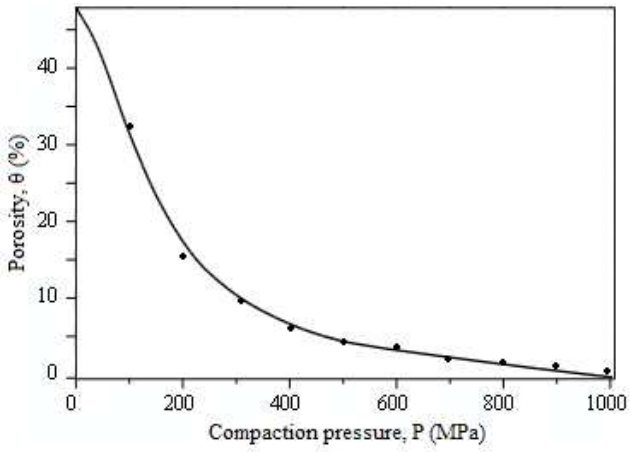


Figure 8

Effect of compaction pressure on the average porosity of green workpiece

Figure 8 shows the titanium sponge compaction diagram, depending on the compaction pressure.

The curve in the form of a solid line has been obtained by computer simulation of the compaction process in a closed mold. The dots denote the results of an experimental measurement of the average volumetric porosity of briquettes in the compacting pressure range from 100 to 1,000 MPa. It can be seen that at compaction pressure of up to 400 MPa the compaction process is quite intensive and the average bulk porosity of the briquette varies from 48% to 7%. Further, the compaction intensity decreases, in the pressure range from 400 to 1,000 MPa, the average volumetric porosity decreases from 7% to 1-2%. The mechanism of compaction of a titanium sponge is similar to the behavior of a metal powder subjected to compaction in a closed mold.

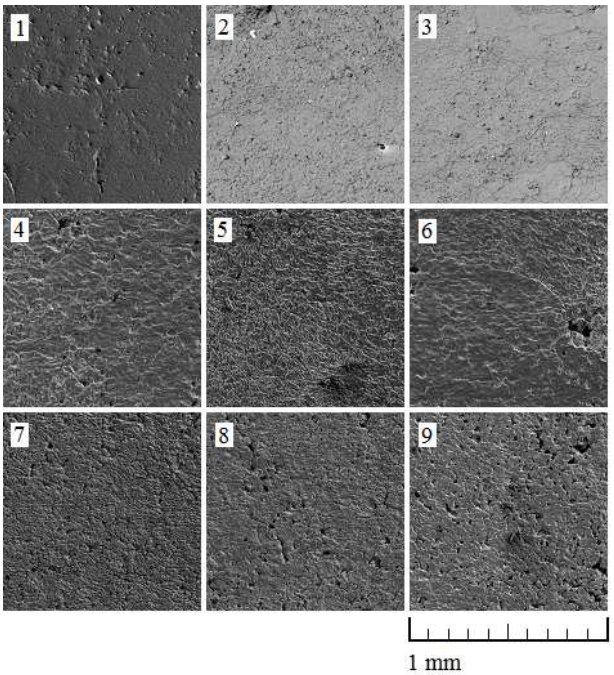


Figure 9
Microstructure in specific areas No. 1–9 of the thin plate surface (×166)

Figure 9 shows the microstructure of one of the thin plates, obtained using the scanning electronic microscope Tescan Vega II XMU. The actual size of areas No. 1–9: height – 1 mm, width – 1 mm. It is obvious that in the horizontal layer of the material, which is in close to the upper punch, the porosity reduces from area 1 towards area 3.

The consistency of porosity distribution pattern in the material lying on the bottom of the container has an inverse dependence, as compaction occurs from area 9 towards area 7.

In the vertical direction, the surface porosity of areas 1–3 is substantially less when compared to areas No. 7–9. The porosity in various areas of the thin plate surfaces was established by the method of quantitative metallography with the application of image processing and analysis with the help of Matlab software. The statistical processing of the experimental data, obtained after the quantitative evaluation of microstructure, was carried out at the confidence level of 0.95. The method of maximal relative deviation computation was used for the verification of the obtained data for normality of distribution and elimination of gross errors.

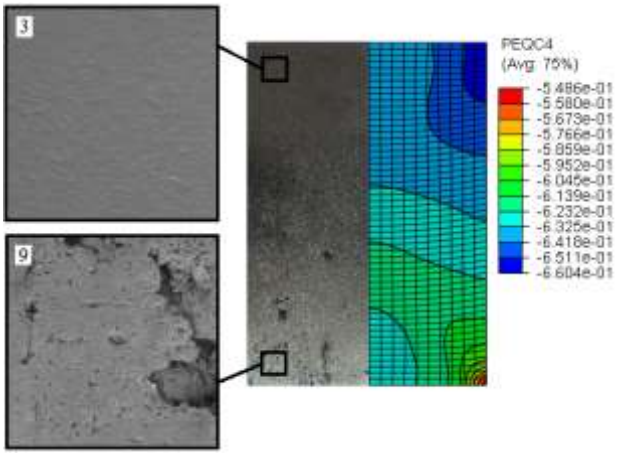


Figure 10
Distribution of total volumetric inelastic strain in the workpiece cross section (right) and microstructure of the thin plate surface (left)

Figure 10 shows the results of the FE simulation die compaction and the microstructure of the work-piece cross-section, obtained by means of electron microscopy. One can see that the distribution of the total volumetric inelastic strain in the work-piece cross-section is in agreement with the material surface porosity distribution on thin plate. The highest value of volume strain is localized in the area adjacent to the upper punch and grows from the center towards periphery; consequently, this area of compacted workpiece will show the lowest porosity value.

In the work-piece center, the level lines of volume strain tend to horizontal position. Coming closer to the edge layers of the work-piece, the horizontal layer compaction is broken. The area of the lowest values of volume strain is located at the junction of the punch and the side surface of the container bottom, here the hydrostatic pressures are less active and their value grows inversely – from the periphery towards the center.

Table 1 shows the comparison of FE simulation and experimental results in specific areas of the briquette cross section.

Table 1
Porosity of the material depending on the tensor stress components (mean stress) in the areas No. 1–9 after die compaction

	Area No.								
	1	2	3	4	5	6	7	8	9
q, MPa	576	617	794	519	539	564	494	444	257
p, MPa	384	413	526	347	356	368	324	290	171
σ_1, MPa	-192	-207	-255	-173	-175	-179	-158	-139	-88
σ_2, MPa	-768	-824	-1054	-692	-709	-732	-653	-585	-342
σ_3, MPa	-192	-207	-255	-173	-175	-179	-158	-139	-88
θ, % (experiment)	1.32	0.82	0.60	1.96	1.73	1.34	2.18	2.84	5.90
θ, % equation (6)	0.80	0.59	0.18	1.18	1.08	0.95	1.50	2.14	7.47

It is obvious that each of the 9 areas (corresponding to the column head numbers of the table) of the work-piece cross-section has unique stress state characteristics. The most effective compaction takes place in area No. 3, where von Mises equivalent stresses value has a maximum value q = 794 MPa and the highest equivalent pressure stress occur p = 526 MPa. Area No. 9 shows the minimal density, characterized by the lowest value of von Mises equivalent stresses q = 257 MPa and by the value of p = 171 MPa.

Conclusions

On the basis of the FE simulation of multiaxial compression process of titanium sponge, the dependence of porosity variation on the value of equivalent pressure stress and von Mises equivalent stresses has been established. It was found that equivalent pressure stress produces a more substantial effect on compaction of material as compared to the effect of shear stress. Nevertheless, in case of a stress condition corresponding to the scheme of isostatic compression, the obtained powdered material porosity has not been less than 6%. It has been shown, that production of quasimonolythic material with a minimal porosity value is only possible at joint action of normal and shear deformations. In order to verify the results of FE simulation, the experiments on compaction of titanium sponge in a closed die were carried out. It was shown that the consistency pattern of the material porosity distribution corresponds to the regularity of changing the field of equivalent pressure stress and the value of von Mises equivalent stresses, obtained by the way of FE analysis.

References

[1] N. A. Shestakov, V. N. Subich, V. A. Demin: Uplotnenie, konsolidatsiya i razrushenie poristykh materialov (in Russian), Moscow, FIZMALIT, 2009

[2] K. L. Nielsen, J. Dahl, V. Tvergaard: Collapse and Coalescence of Spherical Voids Subject to Intense Shearing: Studied in Full 3D. International Journal of Fracture, Vol. 177, Iss. 2, pp. 97-108, 2012

[3] V. Tvergaard: Behaviour of Voids in a Shear Field. International Journal of Fracture, Vol. 158, Iss. 1, pp. 41-49, 2009

[4] G. A. Baglyuk, A. I. Khomenko: Osobennosti deformirovannogo sostoyaniya poristykh zagotovok pri ikh zakrytoi i otkrytoi shtampovke. (in Russsian) Izvestiya vuzov. Tsvetnaya metallurgiya, No. 1, pp. 57-62, 2015

[5] S. C. Lee, K. T. Kim: Densification Behavior of Aluminium Alloy Powder under Cold Compaction. International Journal of Mechanical Sciences, Vol. 44, Iss. 7, pp. 1295-1308, 2002

[6] H. Shin, J. B. Kim, S. J. Kim, K. Y. Rhee: A Simulation-based Determination of Cap Parameters of the Modified Drucker–Prager Cap Model by Considering Specimen Barreling during Conventional Triaxial Testing. Computational Materials Science, Vol. 100, Part A, pp. 31-38, 2015

[7] P. Doremus, C. Geindreau, A. Martin, L. Debove, R. Lecot: High Pressure Triaxial Cell for Metal Powder. Powder Metallurgy, Vol. 38, Iss. 4, pp. 284-287, 1995

[8] A. D. Hartman, S. J. Gerdemann, J. S. Hansen: Producing Lower-Cost Titanium for Automotive Applications. The Journal of The Minerals, Metals & Materials Society, Vol. 50, No. 9, pp. 16-19, 1998

[9] M. P. Bondar, B. V. Voitsekhovskii, E. S. Obodovskii, V. A. Kharchenko: O svoistvakh kompaktnogo titana, poluchennogo obrabotkoi davleniem gubchatogo titana. (in Russia) Tsvetnye metally, Vol. 12, pp. 75-78, 1978

[10] A. M. Laptev, E. S. Obodovskii: Plastic Deformation of Sponge Titanium. Powder Metallurgy and Metal Ceramics, Vol. 25, Iss. 7, pp. 547-552, 1986

[11] E. S. Obodovskii, A. M. Laptev: Effect of Technological Factors on the Properties of High-Density Titanium Sponge Compacts. Powder Metallurgy and Metal Ceramics, Vol. 26, Iss. 4, pp. 295-299, 1987

[12] A. G. Zalazinskii, V. I. Novozhonov, V. L. Kolmykov, M. V. Sokolov: Modelirovanie pressovaniya briketov i vydavlivaniya prutkov iz titanovoi gubki.(I Russia) Metally, No. 6, pp. 64-68, 1997

[13] I. Sh. Trakhtenberg, A. B. Borisov, V. I. Novozhonov, A. P. Rubshtein, A. B. Vladimirov, A. V. Osipenko, V. A. Mukhachev, E. B. Makarova: Mechanical Properties and the Structure of Porous Titanium Obtained by

Sintering Compacted Titanium Sponge. The Physics of Metals and Metallography. Vol. 105, No. 1, pp. 92-97, 2008

[14] A. Hadadzadeha, M. A. Whitney, M. A. Wells, S. F. Corbin: Analysis of Compressibility Behavior and Development of a Plastic Yield Model for Uniaxial Die Compaction of Sponge Titanium Powder. Journal of Materials Processing Technology. Vol. 243, pp. 92-99, 2017

[15] WO 2011049465 A1. Method for Production of Titanium Welding Wire

[16] WO 2012127426 A1. Method for Production of Alloyed Titanium Welding wire

[17] A. V. Nesterenko, V. I. Novozhonov, A. G. Zalazinskii, A. V. Skripov: Influence of Temperature on Compactibility of Briquettes of Titanium Sponge Alloyed with Hydrogen. Russian Journal of Non-Ferrous Metals, Vol. 56, No. 3, pp. 287-292, 2015

[18] I. Berezin, A. Nesterenko, A. Zalazinsky, N. Michurov: "Finite-Element Simulation of Multiaxial Compaction of Sponge Titanium Powder" in Mechanics, Resource and Diagnostics of Materials and Structures (MRDMS–2016) 1785, AIP Conference Proceedings, edited by E. Gorkunov et al. (American Institute of Physics, Melville, NY, 2016), pp. 040008-1-040008-4

[19] ABAQUS 6.10 Theory Manual, Dassault Systemes Simulia Corp., Providence, RI, USA, 2010

[20] L. H. Han, J. A. Elliott, A. C. Bentham, A. Mills, G. E. Amidon, B. C. Hancock: A Modified Drucker–Prager Cap Model for Die Compaction Simulation of Pharmaceutical Powders. International Journal of Solids Structures, Vol. 45, Iss. 10, pp. 3088-3106, 2008

[21] B. S. Zhang, M. Jain, C. H. Zhao, M. Bruhis, R. Lawcock, K. Ly: Experimental Calibration of Density-Dependent Modified Drucker–Prager Cap Model using an Instrumented Cubic Die for Powder Compact. Powder Technology, Vol. 204, Iss. 1, pp. 27-41, 2010

[22] C. Shang, I. C. Sinka, J. Pan: Constitutive Model Calibration for Powder Compaction Using Instrumented Die Testing. Experimental Mechanics, Vol. 52, Iss. 7, pp. 903-916, 2012

[23] S. Garner, J. Strong, A. Zavaliangos: The Extrapolation of the Drucker–Prager/Cap Material Parameters to Low and High Relative Densities. Powder Technology, Vol. 283, pp. 210-226, 2015

[24] C. Pavanachand, R. Krishnakumar: Yield Function Parameters for Metal Powder Compaction based on Unit Cell Studies. Acta Materialia, Vol. 45, Iss. 4, pp. 1425-1444, 1997

[25] V. E. Panin, V. A. Likhachev, Yu. V. Grinyaev: Strukturnye urovni deformatsii tverdykh tel. (in Russia) Novosibirsk: Nauka, 1985

[26] R. J. Henderson, H. W. Chandler, A. R. Akisanya, C. M. Chandler, S. A. Nixon: Micro-Mechanical Modelling of Powder Compaction. Journal of the Mechanics and Physics of Solids, Vol. 49, Iss. 4, pp. 739-759, 2001

[27] N. Ogbonna, N. A. Fleck: Compaction of an Array of Spherical Particles. Acta Metallurgica et Materialia, Vol. 42, Iss. 2, pp. 603-620, 1995

[28] A. Benabbes, L. Dormieux, L. Siad: An Estimate of the Macroscopic Yield Surfaces of Powder Compacts using the Kinematic Approach of the Yield Design Theory. International Journal of Material Forming, Vol. 1, pp. 53-56, 2008

[29] XJ. Xin, P. Jayaraman, G. S. Daehn, R. H. Wagoner: Investigation of Yield Surface of Monolithic and Composite Powders by Explicit Finite Element Simulation. International Journal of Mechanical Sciences, Vol. 45, Iss. 4, pp. 707-723, 2003

[30] S. J. Subramanian, P. Sofronis: Calculation of a Constitutive Potential for Isostatic Powder Compaction. International Journal of Mechanical Sciences, Vol. 44, Iss. 11, pp. 2239-2262, 2002

[31] H. Li, S. Saigal, P. T. Wang: A Solution for the Contact between Two Spherical Particles Undergoing Large Deformation. Acta Materialia, Vol. 44, Iss. 7, pp. 2591-2598, 1996

[32] C. Tsigginos, J. Strong, A. Zavaliangos: On the Force-Displacement Law of Contacts between Spheres Pressed to High Relative Densities. International Journal of Solids and Structures, Vol. 60-61, pp. 17-27, 2015

[33] Yu. A. Aksenov, I. O. Bashkin, V. L. Kolmogorov, E. G. Ponyatovskii, G. G. Taluts, V. K. Kataya, I. V. Levin, Yu. I. Potapenko, A. N. Trubin: Vliyanie vodoroda na plastichnost' i soprotivlenie deformatsii tekhnicheskogo titana VT1-0 pri temperaturakh do 750 °C. FMM, Vol. 67, No. 5, pp. 993-999, 1989

A Method for Determining Roundness and Actual Form of Circular Workpiece Cross Sections

Imre Némedi[1], Milenko Sekulić[2], Vladan Radlovački[2], Janko Hodolič[2], Miodrag Hadžistević[2], Márta Takács[3,4]

[1]Subotica Tech - College of Applied Sciences, Marka Oreškovića 16, 24000 Subotica, Serbia, E-mail: nimre@vts.su.ac.rs

[2] University of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia, E-mail: milenkos@uns.ac.rs, rule@uns.ac.rs, hodolic@uns.ac.rs, miodrags@uns.ac.rs

[3] Óbuda University, Bécsi út 96/b, H-1034 Budapest, Hungary, E-mail: takacs.marta@nik.uni-obuda.hu

[4] University of Novi Sad, Hungarian Language Teacher Training Faculty, Subotica, Stromajerova 11, Serbia, E-mail: takacs.marta@magister.uns.ac.rs

Abstract: Determining a roundness value is essential for enabling fast, easy and smooth assembly of workpieces with circular fitting surfaces. In a number of cases, roundness specifications should be defined if other geometrical specifications are not sufficient for enabling reliable assembly. Various standards treat roundness as a deviation from an ideal circle, whereas, the actual cross section form is not treated at all. It seems this characteristic is not worth examining. Knowing the cross section form, however, enables determining various factors that result in undesirable deviations of workpiece cross sections from an ideal circle. This paper presents a new method which, in addition to determining roundness, can be used to determine the actual form of a circular workpiece cross section, using a coordinate measuring machine.

Keywords: Cross section actual form; roundness; CMM; V-block method

1 Introduction

This paper presents a new method for determining the actual form of a circular workpiece cross section and determining roundness. By knowing the actual form of a cross section, some important information for detecting production errors may

become clear much before they cause unnecessary wastes and costs. To the knowledge of the authors, no method has yet been proposed to solve problems of determining the actual form of a cross section and roundness at the same time. This method may be of a great help in a process where timely detecting form errors of workpieces with circular cross sections is needed. The method is based on analyzing so called "Core of centers", a set of points obtained by calculations described in further text.

Roundness is observed on the basis of internationally accepted and applicable parameters of roundness and specification operators [1]. In their study, Shunmugam and Venkaiah report the results of comparing different methods for processing the results of measuring roundness and cylindricity, and the review thereof [2, 3, 4].

Roundness can be measured in several ways, using different methods. They range from simple diameter measurement and measurement in prism (*V-block method*) [5, 6], then measuring using computer controlled Coordinate Measuring Machines (*CMM*), to using special equipment for measuring roundness, which operates on the basis of the Rotational Datum Method (*RDM*).

These methods are constantly being revised with the aim of improving measurement accuracy [5, 7]. It should be noted that efforts are made to improve conventional methods as well, such as V-block method, which is widely accepted for measuring roundness. However, significant problems exist in analyzing results. These problems are being solved by using inverse matrices [7]. Thus, data processing is being developed, but the measurement principle remains the same, with all its drawbacks [8, 9, 10, 11]. These drawbacks are an integral part of a measurement process. In the V-block method they include permanent movement of the rotation center during the rotation of the measured object along a leaning surface.

2 Description

2.1 Roundness

Roundness tolerance zone in a plane is limited by two concentric circles at a distance t [12].

Roundness can be defined in four ways [13, 14]. In this paper, the definition of roundness using Circular zone with minimum radial separation (Minimum Zone Circles, MZC) is applied.

2.2 Actual Form of a Cross Section

Actual form of circular work pieces deviates from the ideal one. These deviations depend on a number of factors, primarily on the rigidity of a work piece material and machine components, as well as on clamping. Actual forms of circular work pieces' cross sections (both the shaft and holes) most often appear as: a) triangular or polygonal, b) oval or c) multiangular.

Common methods used to determine roundness do not generally enable determining any type of actual form of a circular work piece cross section. Assuming the most probable actual form is usually needed before the measurement. This requires considerable experience. Using CMM easily provides roundness values, but actual form of a cross section still usually remains unknown.

For example, a triangular actual form is common with thin-walled pipes if the clamping is made at three points.

3 Core of Centers Method

Common measuring methods used with ring-shaped cross sections and some other workpieces, require gauges that establish contacts with the measured surface at three points. These are workpieces with triangular and pentagonal cross section error forms. However, oval and square error forms cannot be determined with the required accuracy using three-point contacts. With such shapes contact needs to be made at two points. The most important problem when measuring roundness using common methods is the unknown position of the center and the axis of the measured cross section.

If we want to uncover the form of roundness error (or the actual form of a workpiece cross section), it is necessary to follow positions of those centers. The core of centers method was named after the positions of these centers which it thoroughly analyzes.

The method assumes the use of a coordinate measuring machine. The first step of the method is collecting the coordinates of all cross section contour points using the standard probe.

The easiest way to present and explain the method is by providing examples of ideal oval and pentagonal cross section error forms, using graph presentations.

The second step is calculating the centers of circumscribed auxiliary circles, drawn on pairs or triplets of points from the contour.

Preliminary empirical evidence tells us that data from oval, square and other cross sections with an even number of vertices should be calculated using three-point calculations, while data from other cross sections (triangular, pentagonal etc.) should be subject to two-point calculations, which is actually opposite to the common methods. The reason is obvious: uncovering the positions of the auxiliary circles' centers makes the method. Triplets are contour points on the lines drawn from the center, making angles of 120° each, while the two-points calculation is based on selecting opposite points of the contour.

A scheme of three-point data calculations of the actual, oval contour is presented in Figure 1.

An auxiliary circle is constructed using the selected contour points. The center of this circle is easy to determine.

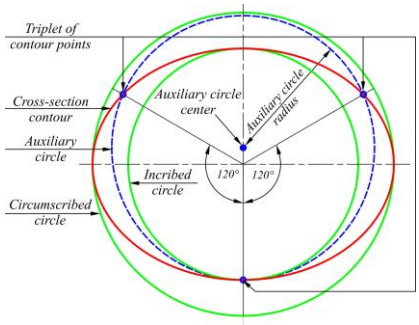


Figure 1
A circle defined by three selected points with its center

After a rotation for a desired angle, the procedure is repeated: another auxiliary circle is obtained along with its center. This step is presented in Figure 2.

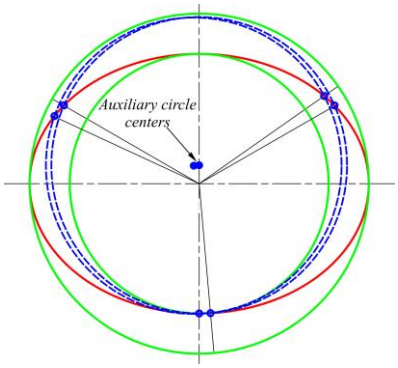


Figure 2
Two auxiliary circles with their centers

The described step is repeated for n times until the contour is entirely rounded.

The centers of the auxiliary circles are distributed according to a certain pattern. This pattern actually presents the "path" of the cross section center "movement" during the rotation and the calculation. The set of centers (core of centers) indicates a roundness error form of the cross section.

At the same time, empirical evidence shows that the "core of centers" points to an actual form of a work piece cross section. Thus, the form of the core of centers is obtained using the "movement" of auxiliary circle centers. If the calculations had been done using pairs of points, the auxiliary circles would have been nearly concentric, and their centers would have coincided providing an unusable image.

All types of roundness error forms can be analyzed using the described method, e.g. ideal pentagonal form presented in Figure 3.

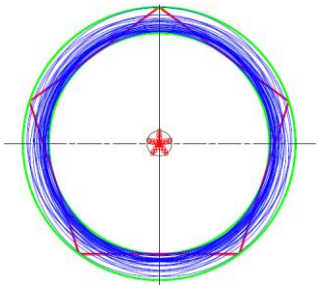


Figure 3
The results of two-point data calculations around the entire contour on the pentagonal error form

Appropriate magnification provides clarity - Figure 4 shows an enlarged core of centers.

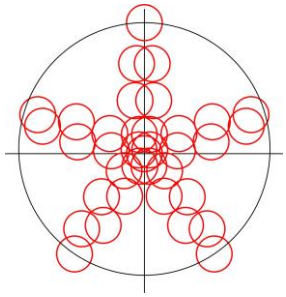


Figure 4
An enlarged core for the pentagonal error form as a result of two-point calculations

Unlike the oval error form, in the case of the pentagonal error form, two-points data calculations must be used. Two points used for drawing auxiliary circles are located on the opposite sides of a cross section contour. The result obtained undoubtedly points to the pentagonal error form.

Calculations using triplets of points result in an unusable distribution of centers - they are scattered along the core of centers, see Figure 5.

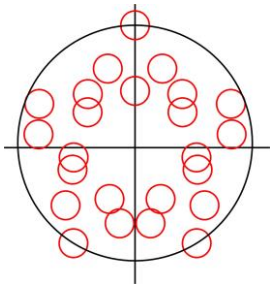


Figure 5
An enlarged picture of a core for pentagonal deviation with three-point calculations

The obtained results show that choosing the number of points for calculation depends on the number of sides of the roundness error form. For an even number of sides of the form (including oval form) three points should be used, while an odd number of sides (vertices) requires the use of pairs of points.

Practical measurements imply that the error form type is not known in advance. To determine the form and enable further processing, calculations need to be performed with both two and three calculation points, and, based on the obtained shape of the core of centers, an unambiguous conclusion should be made about the actual form of the measured object's cross section.

Determining roundness is explained later as a part of experiment.

4 Experiment, Results and Discussion

4.1 Measurement and Workpieces Used

The measurements were performed in the Metrological laboratory at the Department of Metrology, quality, equipment, tools and ecological engineering aspects at the Faculty of Technical Sciences in Novi Sad. The coordinate measuring machine Carl Zeiss CONTURA G2 was used to complete the measurements. Measurement uncertainty of these machine is MPE_E=1.9 μm.

The measurement conditions were standard – working environment temperature was 21°C and humidity was within the standard limits (40 – 60%).

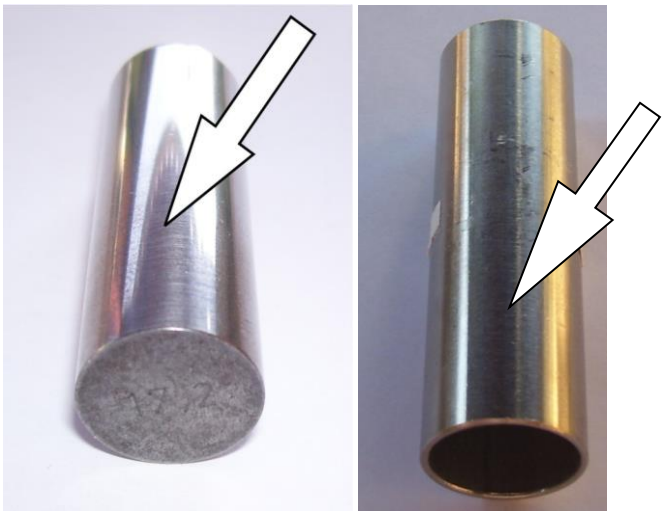
Work pieces were also measured in Mitutoyo Hungaria Kft Testing Laboratory in Budapest, using a special measuring machine for determining roundness, Ra-2100 (RDM), width mmeasurement uncertainty MPE_E=0.02 μm!

Environmental conditions in the laboratory were within required limits (t=21.2°C, humidity was 46%).

For the provision of comparable methodological background with the CMM measurements, the MZC method without filtering was selected.

Work pieces used for the measurements were cylinder d=17.2 mm, and thin-walled pipe d=30 mm (Fig. 6).

Figure 6 presents work pieces used for measurement - measuring surfaces are marked with arrows.



a) b)

Figure 6
The measured workpieces – a) master cylinder d=17.2 mm, and b) thin-walled pipe d=30 mm

4.2 Data Processing

In Section 3, for the purpose of providing appropriate explanations, cross-section center was assumed fixed. The situation with the actual work pieces is significantly different. There is no rotation center point defined in advance, it yet needs to be determined. The rotation center corresponds to the so-called Best Fit circle. Many researchers use a well-known Least Square method for its determination. However, Maisonobe had a different approach [15]. To find the center he used the bisectors' intersections of paired line segments which connect randomly selected triplets from the set of measuring points.

Now, by selecting triplets from the set of the measured points adequate circles are determined. Their centers have to be calculated.

The equations of bisectors of adjacent line segments $P_i(x_i;y_i) \div P_j(x_j;y_j)$ and $P_j(x_j;y_j) \div P_k(x_k;y_k)$ are given by formulas 1 to 4: In Section 3, for the purpose of providing appropriate explanations, cross-section center was assumed fixed. The situation with the actual work pieces is significantly different. There is no rotation center point defined in advance, it yet needs to be determined. The rotation center corresponds to the so-called Best Fit circle. Many researchers use a well-known Least Square method for its determination. However, Maisonobe had a different approach [15]. To find the center he used the bisectors' intersections of paired line

segments which connect randomly selected triplets from the set of measuring points.

Now, by selecting triplets from the set of the measured points adequate circles are determined. Their centers have to be calculated.

The equations of bisectors of adjacent line segments $P_i(x_i;y_i) \div P_j(x_j;y_j)$ and $P_j(x_j;y_j) \div P_k(x_k;y_k)$ are given by formulas 1 to 4:

$$x_{c_{i,j,k}} = \frac{(x_i + x_j) + \alpha_{i,j}(y_j - y_i)}{2} \tag{1}$$

$$y_{c_{i,j,k}} = \frac{(y_i + y_j) - \alpha_{i,j}(x_j - x_i)}{2} \tag{2}$$

$$x_{c_{i,j,k}} = \frac{(x_j + x_k) + \alpha_{j,k}(y_k - y_j)}{2} \tag{3}$$

$$y_{c_{i,j,k}} = \frac{(y_j + y_k) - \alpha_{j,k}(x_k - x_j)}{2} \tag{4}$$

Solving this system of linear equations results in (expressions 5-7):

$$\alpha_{i,j} = \frac{(x_k - x_i)(x_k - x_j) + (y_k - y_i)(y_k - y_j)}{\Delta} \tag{5}$$

$$\alpha_{j,k} = \frac{(x_k - x_i)(x_j - x_i) + (y_k - y_i)(y_j - y_i)}{\Delta} \tag{6}$$

where:

$$\Delta = (x_k - x_j)(y_j - y_i) - (x_j - x_i)(y_k - y_j) \tag{7}$$

The coordinates of the circle center expressed as a function of the selected three coordinates are then (8 and 9):

$$x_{c_{i,j,k}} = \frac{(y_k - y_j)(x_i^2 + y_i^2) + (y_i - y_k)(x_j^2 + y_j^2) + (y_j - y_i)(x_k^2 + y_k^2)}{2\Delta} \tag{8}$$

$$y_{c_{i,j,k}} = \frac{(x_k - x_j)(x_i^2 + y_i^2) + (x_i - x_k)(x_j^2 + y_j^2) + (x_j - x_i)(x_k^2 + y_k^2)}{2\Delta} \tag{9}$$

This procedure has to be repeated using all possible triplets from the contour.

Using the described algorithm, Maisonobe obtained a cloud of intersection points. This is followed by finding the focus point within this cloud. For that purpose Maisonobe used a special type of iterations [15]. The result of these iterations is the position of the Best Fit circle center. Based on the nature of this method, this point was named a triangular center.

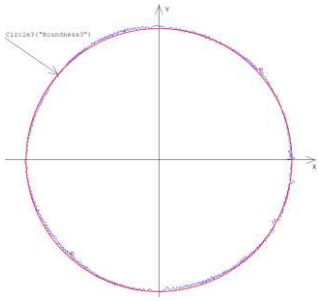
After the triangular center is known, the core of centers is determined by selecting pairs or triplets of contour points and carrying out calculations as explained earlier. The number of calculations needed here is significantly smaller. Cases where pairs of points were used are referred to in the following text as two-point calculations, while triplets of points are referred to three-point calculations.

4.2 Results of Measurements and Measured Data Processing

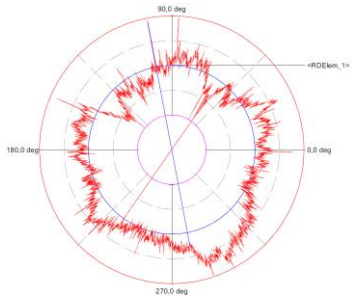
Results for master cylinder d=17.2 mm

Provided review of results includes CMM reports with $n=1600$ measuring points along with measured roundness values (t). This is followed by results of measuring roundness using RDM and Roundtest Ra-2100 special measuring machine, with roundness value determined on the basis of $n=7200$ measuring contour points (the machine had this number of points as a minimum). Next is a review of results of measured points coordinates processing by using the method proposed in this study. Circular diagrams are given, together with enlarged core centers. Circular diagrams are very similar to these obtained directly from the CMM. As previously emphasized, cores are different depending on whether two or three-point calculations were used. For the purpose of comparison, both cases are presented (see Fig. 7).

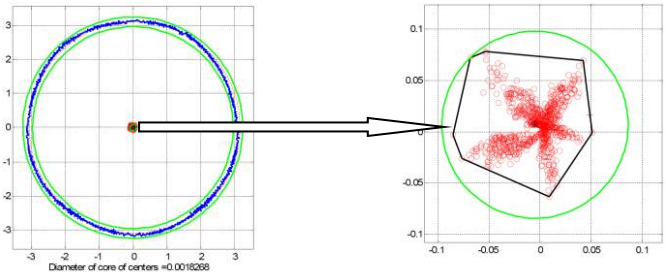
Under circular diagrams obtained by the proposed method, values t as diameters of cores of centers are also designated. These values are determined using the core of centers method. Namely, by drawing a circle as an envelope around the core of centers with the triangular center and by determining its diameter (see enlarged cores and Figure 7c and 7d), values very close to CMM determined roundness error values are obtained. By analyzing several workpieces of the same type (these analyses are not presented herein for understandable reasons), described method is found very useful for estimating roundness error. Envelope circle is determined by a triangular center and the position of the furthest point in the core (these two points make the radius of an envelope circle).



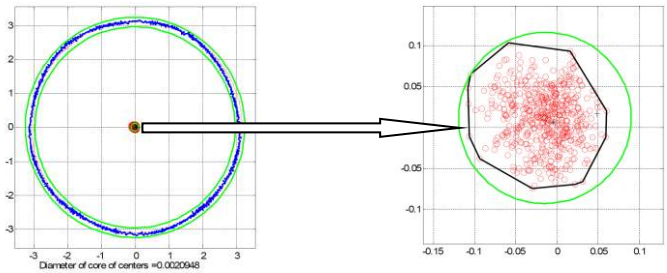
a) Circular diagram obtained from CMM (*t=0.0027mm*)



b) Circular diagram provided by the RDM (*t=0.001225mm*)



c) Core of centers results with two-points calculations (*t=0.0018268mm*)



d) Core of centers results with three-points calculations (*t=0.0020948mm*)

Figure 7
Measuring results for cylinder d=17.2 mm

Figure 7 makes obvious the difference between results of two and three-point calculations. All important values are shown separately, under the figures. The actual form of a cross-section is impossible to determine by observing only CMM or core of centers circular diagrams (Figures 7a, 7c and 7d). Special Roundtest machine provides circular diagram which shows the real form of a cross section.

However, core of centers shape from Figure 7c unambiguously points out to a pentagonal cross section error form (also see Figure 3). This core is obtained using two-point calculations. On the other hand, if three-point calculations are used (see Fig. 7d), a core of centers is shaped as a cloud of points which, at the best, makes determining the real cross section form impossible. So, for pentagonal cross section error forms, two points calculations should be used, as previously explained.

Differences between CMM (*t=0.0027 mm*) and RDM *(t=0.001225 mm)* results are expected. Namely, measured cross sections for these two measurements cannot be identical. During the measurement, all possible measures were undertaken to prevent measuring different cross-sections. There also exists a difference between numbers of measuring contour points as previously explained. This cannot imply extreme result differences, but also can have significant impact to evaluating roundness error. On the other hand, it appeared that core of centers method provided better evaluation of roundness error (*t=0.0018268 mm*) then the CMM, based on same set of contour point coordinates (see Table 1).

Table 1

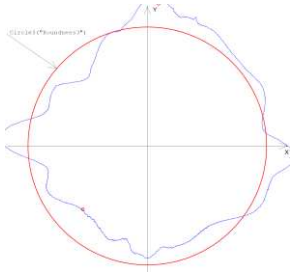
Estimates of roundness errors *t* for cylinder d=17.2 mm (values in mm)

CMM	Core of centers method	RDM
0.0027	0.0018268	0.001225

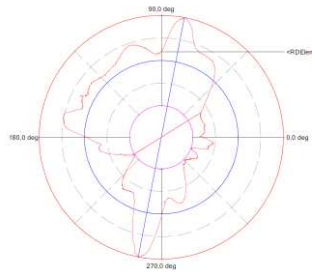
Results for thin-walled pipe d=30 mm

Second example is a thin-walled pipe deformed for unknown reasons to the squared roundness deviation. Deformation is even visible from circular diagrams (see Fig. 8). This example also points out to the practical usability of a proposed method.

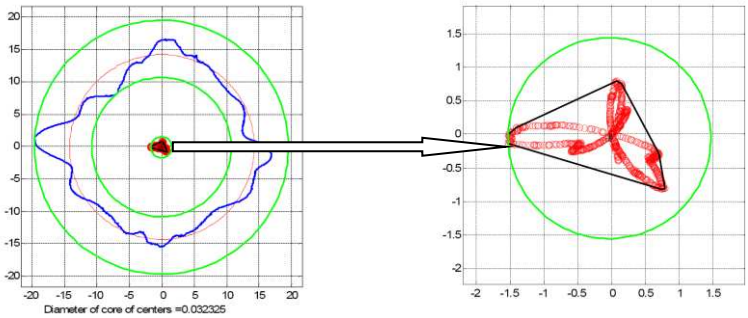
Determining real form of a cross-section does not require using core of centers method in this case. Core of centers from Fig. 8c is obtained using two-point calculations, while three-point calculations produce core of centers given in Fig. 8d. By analyzing several workpieces of the same type (these analyses are also not presented herein) it is proven that the core shape from Fig. 8c is impossible to obtain in any other way. It shows that using two-point calculations is not to be used in this case. So, using three-point calculations is the only remaining alternative.



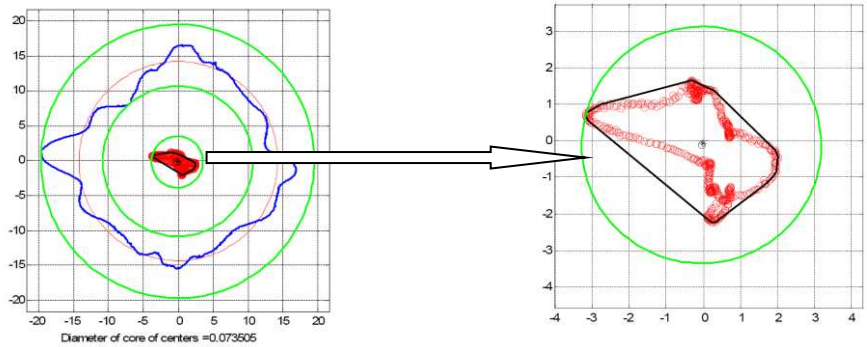
a) Circular diagram obtained from CMM (t=0.0732mm)



b) Circular diagram provided by the RDM (t=0.08628mm)



c) Core of centers results with two-points calculations (t=0.032325mm)



d) Core of centers results with three-points calculations (t=0.073505mm)

Figure 8
Measuring results for thin-walled pipe d=30 mm

Table 2
Estimates of roundness errors t for thin-walled pipe d=30 mm (values in mm)

CMM	Core of centers method	RDM
0.0732	0.073505	0.08628

Results are very similar to the ones in the previous case. The RDM roundness estimate is again different from the estimate provided by CMM, while core of centers method estimate is close to CMM estimate, yet between the rest two values.

In case of larger scale deviation, as the results prove it, the Core of Centers Method does not provide greater accuracy of the roundness value. Knowing that the CMM and the Core of Centers Method works with the same entering data, the only acceptable explanation is that the mathematical apparatus in such cases produces similar results. The reason of this may be found in the detailed mathematical analysis of both methods which could be an interesting theme of a new research. In any case the aim of the real form of the part's cross section did not fail to be attained either.

Based on described results of measurement and measured points coordinates processing results, the following conclusions can be drawn:

1. Core of centers method provides an unambiguous and reliable estimate of a rotating workpiece's measured cross-section's real form, regardless of a roundness error value. For small and even for large deviations from the ideal circle, the method is stable and always provides useful results.

2. Completed measurements showed that the number of points to be used in calculations for determining auxiliary circles in core of centers method has to be carefully determined. If an expected cross-section real form is a polygon with even number of sides (vertices), 3 points should be used for calculations. For an odd number of expected polygon sides, 2 points calculations should be used.

   This reasoning reminds us of the processing of polygonal holes by means of a special drilling tool. This time the number of necessary cutting edges if the number is odd (ex. 3, 5, etc.) will be 2 (two), but if the number is even (ex. 4, 6, etc.) will be 3 (three). Everything else is achieved by the complex moving of the axis of the drilling tool.

   So the logical conclusion is the same, only the application is reversed. Instead of the number of cutting edges the number of measuring points (2 or 3 points calculations) comes into consideration,, while the moving of the tool's axis is done by the position of the centers of auxiliary circles.

   The practical application of the method as it is imagined can be represented by thought of 7c and d as well as 8c and d figures. A set of coordinates of the measuring points, that we got from the CMM is processing by 2 and 3 point touch (method with 2 or 3 points calculations). On basis of the cores that we got in this way we can have an unambiguous decision on the real form.

3. Core of centers obtained by applying the method can be used not only for determining a real form of a circular workpiece cross-section, but for estimating a roundness error as well. The diameter of a core envelope circle can be used for that purpose. There are indications that in cases of small deviations from ideal circle (i.e. small roundness error values) this method results are more accurate then methods used in modern CMMs.

Conclusions

Inspecting workpiece form errors, during production is often a key, to achieving smooth and fast assembly. Inspecting form errors is very important when narrow tolerances exist. In such cases real forms of fitting surfaces have significant impact on a likelihood of successful and effective assembly of paired workpieces.

On the other hand, by recognizing the actual form of a cross-section may help revealing systematic and random external effects implying unfavorable variations. This information can be used as a guideline for committing appropriate adjustments and corrective actions to the production system, i.e. reducing or eliminating any unfavorable external effects.

The results of this study support the significant practical value of the proposed method. However, these results can also be treated as preliminary, for more extensive research is required to improve the method and bring it into practical use. It is authors' opinion that using this method, that reveals both the roundness error and actual form of a workpiece cross-section, may contribute a great deal to increasing the efficiency of some measuring and inspection processes. Its application might help revealing causes of the nonconformities that emerge during production, management of assembly process and therefore, decrease production costs, by providing important information about nonconformity causes in real time.

References

[1] International Organization for Standardization, Geneva, Switzerland, ISO/TS 12181–1 and 12181–2, Geometrical product specifications (GPS) – Roundness; Part 1: Terms, definitions and parameters of roundness and Part 2: Specification operators, international standards, 2003

[2] M. S. Shunmugam, N. Venkaiah: Establishing circle and circular-cylinder references using computational geometric techniques, Int J Adv Manuf Technol 51 (1-4), 2010, pp. 261-275

[3] M. S. Shunmugam, N. Venkaiah: Evaluation of form data using computational geometric techniques—Part I: Circularity error, Int J Mach Tool Manu 47 (7-8), 2007, pp. 1229-1236

[4] N. Venkaiah, M. S. Shunmugam: Evaluation of form data using computational geometric techniques—Part II: Cylindricity error, Int J Mach Tool Manu 47 (7-8), 2007, pp. 1237-1245

[5] S. Adamczak, D. Janecki, K. Stepien: Cylindricity measurement by the V–block method – Theoretical and practical problems, Measurement 44 (1), 2011, pp. 164-173

[6] K. Stepien, D. Janecki, S. Adamczak: Investigating the influence of selected factors on results of V–block cylindricity measurements, Measurement, 44 (4), 2011, pp. 767-777

[7] E. Okuyama, K. Goho, K. Mitsui: New analytical method for V–block three–point method, Precis Eng 27 (3), 2003, pp. 234-244

[8] J. Gi–Bum, H. K. Dong, Y. J. Dong: Real time monitoring and diagnosis system development in turning through measuring a roundness error based on three–point method, Int J Mach Tool Manu, 45 (12-13), 2005, pp. 1494-1503

[9] G. Wei, K. Satoshi, S. Takamitu: High–accuracy roundness measurement by a new error separation method, Precis Eng 21 (2-3), 1997, pp. 123-133

[10] G. X. Zhang: A Study on the Abbe Principle and Abbe Error, CIRP Ann-Manuf Techn 38 (1), 1989, pp. 525-528

[11] R. Thalmann: Basics of highest accuracy roundness measurement, Simposio de Metrologia 2006 of CENAM (Centro Nacional de Metrología), Queretaro, Mexico, 2006, pp. 1-6

[12] International Organization for Standardization, Geneva, Switzerland, ISO 1101, Geometrical product specifications (GPS) - Geometrical tolerancing - Tolerances of form, orientation, location and run-out, 2012

[13] R.W. Berger: The Certified Quality Engineer Handbook, ASQ Quality Press, 2007, pp. 253-255

[14] M. Hadžistević, I. Nemedi, M. Sekulić, M. Bosak, J. Hodolič: Multi–Aspect Value of Measuring Systems and Methods Based on the Results of Roundness Measurements, Journal of Mechanics Engineering and Automation 2 (8), 2012, pp. 514-530

[15] L. Maisonobe (2007) Finding the circle that best fits a set of points, <http://www.spaceroots.org/documents/circle/circle-fitting.pdf /> (accessed 12 Mar 2012)

Control of Physiological Systems through Linear Parameter Varying Framework

György Eigner

Physiological Controls Research Center
Research and Innovation Center of Óbuda University
H-1032, Budapest, Kiscelli street 82
eigner.gyorgy@nik.uni-obuda.hu

Abstract: The paper presents a novel controller and observer design methodology for nonlinear systems based on the Linear Parameter Varying (LPV) framework. The introduced techniques effectively combine the classical state feedback methodology with matrix similarity theorems. The presented solutions are analyzed in order to assess their benefits, drawbacks and limitations. The possible selection of scheduling variables is investigated and dyadic structures are used to strengthen the eigenvalue equality from a mathematical point of view. The connection between the controller and observer side is presented and a solution is given for occurring matrix invertibility issues. The method is tested for a control of nonlinear physiological system, more specifically, for the control of innate immune system. The results show that the developed complementary LPV controller and observer are able to satisfy the predefined criteria.

Keywords: Linear Parameter Varying, Control of immune system, LPV-based control technique, Physiological control

1 Introduction

During controller design, today's control engineers have to face many challenges. On the one hand, application of nonlinear controller design techniques requires an experienced designer, use of advanced mathematical tools and unique approaches in each case. On the other hand, the application of well-known linear design methods provides controllers operating only under specific circumstances. Since the real world processes are not linear, to catch that specific proper operating environment in which the nonlinear process can be handled as linear is difficult [1].
In the last two decades, several applications appeared – mostly on LPV basis – that aim to deal with the effective combination of the linear controller design methods on nonlinear systems under given restrictions and requirements. Besides, such innovative methods appeared that effectively exploit the iterative and adaptive techniques, even without the use of the Lyapunov laws or other techniques.

A good example for the latter is the Robust Fixed Point Transformation (RFPT) based techniques. The RFPT-based observer design formalizes the control problem as a fixed point problem. By using adaptive iterative mathematical tools the solution of the fixed point problem also becomes the desired control action which satisfies the predefined criteria [2, 3].
An important direction in the combination of linear design methods on a Lyapunov basis and nonlinear control tasks is the application of the LPV framework [4]. All state space models can be described as LPV models in which the most crucial properties are represented by the so-called parameter vector. A LPV model consists of finite or infinite Linear Time Invariant (LTI) systems. That is, which LTI system's properties reflect in the LPV system depends on the varying parameter vector. If the parameter vector is constant then the LPV system reduces to a LTI system.
Many application possibilities appeared recently which effectively exploited the benefits of the LPV model descriptions [5, 6]. One of these is the Gain Scheduling (GS) controller design [7, 8]. In this case the parameter space of the LPV system is divided into sections and it is possible to design such controllers on the basis of linear design techniques that can deal with the control of a given sector. In this way there are plenty different, but similar controllers designed one-by-one for each sector. The change between the designed sector controllers is taking place accordingly the variation of the parameter vector. Another direction is the controller design for polytopic LPV models via Linear Matrix Inequalities (LMI). In this case, the resulting controller can be designed for a given parameter domain – defined as a hyperbox in the parameter space – by using LMI techniques. The control tasks can be formalized as a LMI, which satisfies given prescriptions and true for all vertices of the defined polytope. Via optimization it is possible to design such a LMI-LPV controller, which is the convex combination of the designed subcontrollers (one for each vertex) and, which is able to control the system, if the parameter vector is inside the given domain [9, 10]. There are other possibilities as well which aim to catch all possible occurring LTI systems during the operation, like the frequency domain based methods [11]. Although, all of them have many benefits, but they do have drawbacks as well. One main limitation is that these methods use only a particular region of the parameter space and do not provide a solution for the whole.
In this work we focus on another direction, namely, we aim to provide a controller design solution that is able to handle the whole parameter space beside the appropriate control action and global stability. The introduced method uses the mathematical properties of the parameter space of the LPV systems and linear controller design techniques. Furthermore, it does not require LMIs or other computational costly methods.
The paper is organized as follows: first we introduce the LPV based design method, mathematical tools, limitations, applicability and developed control structure. Afterwards, the application of the method in case a physiological system is shown. Finally, we conclude with the results and give a short outline of the future work.

2 The LPV-based Design Method

2.1 LPV Systems in General

In this section the developed state feedback based complementary LPV controller and observer designs are detailed. The procedure allows the controller design for the nonlinear system to be controlled – given by its state-space representation – through the so-called LPV framework. The method combines modern state feedback design, LPV methods and the matrix similarity theorems in order to realize the complementary LPV controller and observer.

Definition 1. *LPV model in state space form*
A LPV model can be described in state space representation, and the compact form of it is:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}(\mathbf{p}(t))\mathbf{x}(t)+\mathbf{B}(\mathbf{p}(t))\mathbf{u}(t)+\mathbf{E}(\mathbf{p}(t))\mathbf{d}(t)\\ \mathbf{y}(t) &= \mathbf{C}(\mathbf{p}(t))\mathbf{x}(t)+\mathbf{D}(\mathbf{p}(t))\mathbf{u}(t)+\mathbf{D_2}(\mathbf{p}(t))\mathbf{n}(t)\end{aligned} \tag{1a}$$

$$\mathbf{S}(\mathbf{p}(t)) = \begin{pmatrix}\mathbf{A}(\mathbf{p}(t)) & \mathbf{B}(\mathbf{p}(t)) & \mathbf{E}(\mathbf{p}(t)\\ \mathbf{C}(\mathbf{p}(t)) & \mathbf{D}(\mathbf{p}(t)) & \mathbf{D_2}(\mathbf{p}(t))\end{pmatrix}, \tag{1b}$$

$$\begin{pmatrix}\dot{\mathbf{x}}(t)\\ \mathbf{y}(t)\end{pmatrix} = \mathbf{S}(\mathbf{p}(t))\begin{pmatrix}\mathbf{x}(t)\\ \mathbf{u}(t)\\ \mathbf{d}(t)\\ \mathbf{n}(t)\end{pmatrix}, \tag{1c}$$

where $\mathbf{A}(\mathbf{p}(t)) \in \mathbb{R}^{n\times n}$ *is the state matrix,* $\mathbf{B}(\mathbf{p}(t)) \in \mathbb{R}^{n\times m}$ *is the control input matrix,* $\mathbf{E}(\mathbf{p}(t)) \in \mathbb{R}^{n\times h}$ *is the disturbance input matrix,* $\mathbf{C}(\mathbf{p}(t)) \in \mathbb{R}^{k\times n}$ *is the output matrix,* $\mathbf{D}(\mathbf{p}(t)) \in \mathbb{R}^{k\times m}$ *is the control input forward matrix and* $\mathbf{D_2}(\mathbf{p}(t)) \in \mathbb{R}^{k\times h}$ *disturbance input forward matrix. Moreover,* $\mathbf{u}(t) \in \mathbb{R}^m$, $\mathbf{d}(t) \in \mathbb{R}^h$, $\mathbf{n}(t) \in \mathbb{R}^h$, $\mathbf{y}(t) \in \mathbb{R}^k$ *and* $\mathbf{x}(t) \in \mathbb{R}^n$ *vectors are the control, disturbance and noise inputs, output and state vector, respectively.*
$\mathbf{S}(\mathbf{p}(t)) \in \mathbb{R}^{(n+k)\times(n+m+h)}$ *is the parameter dependent system matrix, which equivocally determines the LPV system. Further, the* $\mathbf{p}(t) \in \Omega \in \mathbb{R}^q$ *is the time dependent parameter vector.*

Evidently, if a LPV system does not contain or model the noise and disturbance then only $\mathbf{A}(\mathbf{p}(t))$, $\mathbf{B}(\mathbf{p}(t))$, $\mathbf{C}(\mathbf{p}(t))$ and $\mathbf{D}(\mathbf{p}(t))$ matrices occur and $\mathbf{S}(\mathbf{p}(t))$ consists of these matrices in appropriate dimensions.

Definition 2. *Parameter vector and parameter space*
The $\mathbf{p}(t) \in \Omega \in \mathbb{R}^q$ *real parameter vector consists of the so-called scheduling variables* $p_i(t)$ $i = 1,2,\ldots,q$, *which are selected terms of the original nonlinear model. The* $\mathbf{p}(t)$ *spans the* $\mathbb{R}^q$ *real parameter space (which is a real Euclidean vector space) in which the dimension q is equal to the number of the selected scheduling variables (dimension of the parameter vector). The* $\Omega \in \mathbb{R}^q$ *is a bounded subspace (hypercube) of the parameter space that is determined by the interpreted (reasonable/possible) extremes of the scheduling variables, i.e.* $\mathbf{p}(t)$*:* $\Omega = [p_{1,min}, p_{1,max}] \times$

$[p_{2,min}, p_{2,max}] \times \ldots \times [p_{q,min}, p_{q,max}] \in \mathbb{R}^q$. *The variation of the* $\mathbf{p}(t)$ *must be inside* Ω*, if* Ω *is defined.*

Remark 1. qLPV model
If the $\mathbf{p}(t)$ does contain not only scalers or functions from the original nonlinear model but state variables as well it is called quasi-LPV (qLPV) model.

Usually, the LPV system can be generally described only in affine and polytopic forms [9]. In this study, the general LPV form is used, which means the $\mathbf{p}(t)$ is embedded directly into the system matrices.

Remark 2. Selection of the $p_i(t)$ $i = 1, 2, \ldots, q$ scheduling variables
The $p_i(t)$ scheduling variables should be the nonlinearity inducing terms in the original nonlinear system. In this way, the $\mathbf{p}(t)$ contains each nonlinearity inducing elements from the system equation, thus the $\mathbf{S}(\mathbf{p}(t))$ LPV description is able to hide the nonlinear terms and handle them as scalars (if $\mathbf{p}$ is fixed) or time varying parameters (if $\mathbf{p}(t)$ varies in time). Appropriate selection of $p_i(t)$ is a key condition for controllability and observability of the later defined reference LTI system.

2.2 State Feedback, Controllability and Observability

The applicability of a state feedback based controller depends on the controllability (stabilizability) and observability (detectability) of the given system. These criteria – due to Kalman [12] – are determined by the structures of the given system representation. More precisely, the eigenvalues of $\mathbf{A}$ (modes of the system), the $\mathbf{B}$ input matrix and the $\mathbf{C}$ output matrix determine these key properties, if disturbance, noise and direct coupling between the input and output are not considered [13, 14].
Consider the following dynamical LTI system:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned} . \tag{2}$$

The system (2) is controllable, if the $\mathbf{Co} = [\mathbf{B} \;\; \mathbf{AB} \;\; \mathbf{A}^2\mathbf{B} \;\; \ldots \;\; \mathbf{A}^{n-1}\mathbf{B}]$ controllability matrix has full row rank, or equivalently, if all modes of $\mathbf{A}$ ($\lambda(\mathbf{A})$ eigenvalues) are accessible through $\mathbf{B}$, namely $v^*\mathbf{A} = \lambda(\mathbf{A})v^*$ and $v^*\mathbf{B} \neq 0$ (the latter criteria is the so-called Popov-Belevitch-Hautus (PBH) test) [14]. In this case, it is possible to design such $\mathbf{K}$ feedback gain through which the $\mathbf{A} - \mathbf{BK}$ closed-loop poles $\lambda(\mathbf{A} - \mathbf{BK})$ can be freely assigned on the complex plain and the unstable modes can be stabilized, i.e. $\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t)$ and

$$\begin{aligned} \dot{\mathbf{x}}(t) &= (\mathbf{A} - \mathbf{BK})\mathbf{x}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned} . \tag{3}$$

The system (2) is observable, if the $\mathbf{Ob} = [\mathbf{C} \;\; \mathbf{CA} \;\; \mathbf{CA}^2 \;\; \ldots \;\; \mathbf{CA}^{n-1}]^\top$ observability matrix has full column rank, or equivalently, if all modes of $\mathbf{A}$ ($\lambda(\mathbf{A})$ eigenvalues) are detectable through $\mathbf{C}$, namely $\mathbf{A}w = \lambda(\mathbf{A})w$ and $\mathbf{C}w \neq 0$ [14].
In this case, it is possible to design such $\mathbf{G}$ observer gain through which the $\mathbf{A} -$

$\mathbf{GC}$ closed-loop poles $\lambda(\mathbf{A}-\mathbf{GC})$ can be freely assigned on the complex plain and $e(t) := \mathbf{x}(t) - \hat{\mathbf{x}}(t)$ observation error $e(t) \to 0$, $t \to \infty$.

$$\dot{\hat{\mathbf{x}}}(t) = (\mathbf{A}-\mathbf{GC})\hat{\mathbf{x}}(t) + \mathbf{G}\mathbf{y}(t) + \mathbf{H}\mathbf{u}(t) \ , \tag{4}$$

where $\mathbf{H} := \mathbf{B}$.
Assume that the $\mathbf{p}(t)$ parameter vector is fixed (does not vary in time) and named as $\mathbf{p}_{ref}$ reference parameter vector. In this case, the $\mathbf{S}(\mathbf{p}(t))$ general LPV system simplifies to a $\mathbf{S}_{ref} := \mathbf{S}(\mathbf{p}_{ref})$ reference LTI system. Controllability and observability of the reference LTI system is a key property according to the preliminary assumptions.

Remark 3. Before we go further two limitations should be noted regarding this study:

- Only fully controllable and observable $\mathbf{S}_{ref}$ reference LTI systems are investigated – investigation of only stabilizable and detectable systems will be the part of the future work;
- Parameter dependency can occur only in the $\mathbf{A}(\mathbf{p}(t))$ system matrix – thus, other matrices cannot contain parameter dependent terms.

The latter restriction can be easily relaxed. If an input (output) contains nonlinearity causing element this input (output) should be handled as new "state variable" and with the extension of $\mathbf{A}$ and reduction of $\mathbf{B}$ ($\mathbf{C}$) the term can be linked to a state (e.g. input case: $\dot{x}_1(t) = x_1(t)u_1(t) \to \dot{x}_1(t) = x_1(t)x_2(t)$ and $\dot{x}_2(t) = u(t)$; output case: $y(t) = x_1(t)x_2(t) \to x_3(t) = x_1(t)x_2(t)$ and $y(t) = x_3(t)$). The price is the extra dynamics, which has to be handled.
The controllability depends on the $(\mathbf{A}(\mathbf{p}_{ref}), \mathbf{B})$ complex. Assume that the reference controllability matrix $\mathbf{Co}_{ref} = [\mathbf{B} \ \mathbf{A}(\mathbf{p}_{ref})\mathbf{B} \ \mathbf{A}(\mathbf{p}_{ref})^2\mathbf{B} \ \dots \ \mathbf{A}(\mathbf{p}_{ref})^{n-1}\mathbf{B}]$. If $\text{rank}(\mathbf{Co}_{ref}) = n$ then the $(\mathbf{A}(\mathbf{p}_{ref}), \mathbf{B})$ (thus, the reference LTI system) is controllable.
The observability depends on the $(\mathbf{A}(\mathbf{p}_{ref}), \mathbf{C})$ complex. Assume that the reference observability matrix $\mathbf{Ob}_{ref} := [\mathbf{C} \ \mathbf{CA}(\mathbf{p}_{ref}) \ \mathbf{CA}(\mathbf{p}_{ref})^2 \ \dots \ \mathbf{CA}(\mathbf{p}_{ref})^{n-1}]^\top$. If $\text{rank}(\mathbf{Ob}_{ref}) = n$ then the $(\mathbf{A}(\mathbf{p}_{ref}), \mathbf{B})$ (thus, the reference LTI system) is observable.

Remark 4. It is important to realize that $\mathbf{S}(\mathbf{p}(t))$ LPV system cannot be controlled and/or observed at every $\mathbf{p}(t)$ (everywhere in the parameter domain). This can occur when $\mathbf{p}(t)$ or given elements of it become equal to zero. In this case, the rank of $\mathbf{Co}$ and/or $\mathbf{Ob}$ can be lower than n. Moreover, $\mathbf{p}(t)$ or $p_i(t)$ can cause linear dependencies in $\mathbf{Co}$ and/or $\mathbf{Ob}$ which reduces the rank of the matrices as well and reduces the controllability and/or observability.

Appropriate selection of $p_i(t)$ scheduling variables and the $\mathbf{p}_{ref}$ is critical and determines the controllability and observability properties of $\mathbf{S}_{ref}$. On the one hand, only those states can be embedded into the $\mathbf{p}(t)$ which can be measured or estimated – since the $\mathbf{p}(t)$ is directly used in the complementary controller and observer structures. On the other hand, the "positions" of the $p_i(t)$ scheduling variables in $\mathbf{A}(\mathbf{p}(t))$ are also crucial as we see later and determines which states can be linked to the scheduling variables.

2.3 Matrix similarity theorems and dyadic structures

The core of the developed complementary LPV controller and observer structures are based on the special properties of the similarity theorems. Further, dyadic structures are useful for the generalization of the methods as well. The following definitions, theorems and proofs can be found in [15, 16, 17, 18].

Definition 3. *Similarity of matrices:*
A quadratic, $n \times n$ matrix $\mathbf{Q}$ is similar to a matrix $\mathbf{W}$, if exists an invertible $\mathbf{R}$ matrix that is $\mathbf{Q} = \mathbf{R}^{-1}\mathbf{W}\mathbf{R}$. Notation: $\mathbf{Q} \sim \mathbf{W}$.

Theorem 1. *Similarity invariance of the determinants of matrices: If $\mathbf{Q} \sim \mathbf{W}$, then $|\mathbf{Q}| = |\mathbf{W}|$.*

Proof. Let $\mathbf{Q} \sim \mathbf{W}$, namely, $\mathbf{Q} = \mathbf{R}^{-1}\mathbf{W}\mathbf{R}$. Then $|\mathbf{Q}| = |\mathbf{R}^{-1}\mathbf{W}\mathbf{R}| = |\mathbf{R}^{-1}||\mathbf{W}||\mathbf{R}| = |\mathbf{W}|$, since $|\mathbf{R}||\mathbf{R}^{-1}| = 1$. [15, 17]. ■

Theorem 2. *If $\mathbf{Q} \sim \mathbf{W}$, then the characteristic polynomials of the matrices and thus, the eigenvalues and the geometric and algebraic multiplicities of the eigenvalues of the matrices are the same (i.e. $\lambda(\mathbf{Q}) = \lambda(\mathbf{W})$)*

Proof. Let $\mathbf{Q} \sim \mathbf{W}$, namely, $\mathbf{Q} = \mathbf{R}^{-1}\mathbf{W}\mathbf{R}$. Then $\mathbf{Q} - \lambda\mathbf{I} = \mathbf{R}^{-1}\mathbf{W}\mathbf{R} - \lambda\mathbf{R}^{-1}\mathbf{I}\mathbf{R} = \mathbf{R}^{-1}(\mathbf{W}\mathbf{R} - \lambda\mathbf{I}\mathbf{R}) = \mathbf{R}^{-1}(\mathbf{W} - \lambda\mathbf{I})\mathbf{R}$, namely, $\mathbf{Q} - \lambda\mathbf{I} \sim \mathbf{W} - \lambda\mathbf{I}$, where $\mathbf{I}$ is the unity matrix in appropriate dimension [15, 16]. ■

Definition 4. *Dyadic product (or shortly dyad):*
The product of $\mathbf{q}_{n\times 1}$ and $\mathbf{w}^{\top}_{1\times m}$ vectors results a $\mathbf{q}_{n\times 1}\ \mathbf{w}^{\top}_{1\times m} := \mathbf{X}_{n\times m}$ matrix [18].

Definition 5. *Sum of dyadic series:*
The sum of a dyadic series can be described with a product two matrices and the opposite is also true.

$$\begin{bmatrix} \mathbf{q}_1 \end{bmatrix} \begin{bmatrix} \quad \mathbf{w}_1^{\top} \quad \end{bmatrix} + \vdots + \begin{bmatrix} \mathbf{q}_k \end{bmatrix} \begin{bmatrix} \quad \mathbf{w}_k^{\top} \quad \end{bmatrix} = \sum_{i=1}^{k} \mathbf{q}_i \mathbf{w}_i^{\top} = \mathbf{Q}\mathbf{W}^{\top}$$

$$\mathbf{Q}\mathbf{W}^{\top} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_k \end{bmatrix} \begin{bmatrix} \mathbf{w}_1^{\top} \\ \mathbf{w}_2^{\top} \\ \vdots \\ \mathbf{w}_k^{\top} \end{bmatrix} \tag{5}$$

Definition 6. *Minimal dyadic decomposition:*
If we realize a matrix as the sum of the minimum of dyads as possible [18].

Definition 7. *Rank of a matrix*
The rank of a matrix is equal to the number of dyads which are represented in the minimal dyadic decomposition [18].

On one hand, these mathematical tools can be used to define eigenvalues equality rules for state feedback systems. Further, the useful properties of dyadic representation – especially the rank criteria – can be used for generalization purposes for the developed method.

2.4 Design of Complementary LPV Controller

Assume that a state feedback controller for the reference $\mathbf{S}_{ref}$ can be designed which requires the satisfaction of the controllability criteria detailed above. In this case, the control law can be described as $\mathbf{u}(t) = -\mathbf{K}_{ref}\mathbf{x}(t)$. As we see in (3) the closed-loop system matrix becomes $\mathbf{A}_{ref} - \mathbf{B}\mathbf{K}_{ref}$ whose eigenvalues $\lambda(\mathbf{A}_{ref} - \mathbf{B}\mathbf{K}_{ref})$ define the dynamics of the reference system. In that case, if we want to use the state feedback control concerning a given LPV system the parameter dependency has to be represented in the state feedback controller as well. Thus, only $\mathbf{A}(\mathbf{p}(t))$ can be parameter dependent – as it was declared above – a given LPV system under control can be described as:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \big(\mathbf{A}(\mathbf{p}(t)) - \mathbf{B}\mathbf{K}(t)\big)\mathbf{x}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned} \quad . \tag{6}$$

Definition 8. *Complementary $\mathbf{p}(t)$ dependent feedback gain*
A LPV feedback gain $\mathbf{K}(t)$ consists of a $\mathbf{K}_{ref}$ reference and $\mathbf{K}(t)$ varying feedback gain as follows: $\mathbf{K}(t) := \mathbf{K}_{ref} + \mathbf{K}(\mathbf{p}(t))$. Therefore, from (6) $\big(\mathbf{A}(\mathbf{p}(t)) - \mathbf{B}\mathbf{K}(t)\big) = \big(\mathbf{A}(\mathbf{p}(t)) - \mathbf{B}(\mathbf{K}_{ref} + \mathbf{K}(\mathbf{p}(t)))\big)$.

Assume that $\mathbf{A}_{ref} - \mathbf{B}\mathbf{K}_{ref} \sim \mathbf{A}(\mathbf{p}(t)) - \mathbf{B}(\mathbf{K}_{ref} + \mathbf{K}(\mathbf{p}(t)))\ \ \forall \mathbf{p}(t)$. The eigenvalues (poles) of the closed-loop reference LTI system are $\lambda_{ref} := \lambda(\mathbf{A}_{ref} - \mathbf{B}\mathbf{K}_{ref})$ and the eigenvalues (poles) of the closed-loop LPV system are $\lambda(\mathbf{p}(t)) := \lambda(\mathbf{A}(\mathbf{p}(t)) - \mathbf{B}(\mathbf{K}_{ref} + \mathbf{K}(\mathbf{p}(t))))$.
Theorem 2 consequences that $\lambda_{ref} = \lambda(\mathbf{p}(t))\ \ \forall \mathbf{p}(t)\ t \geq 0$ due to the similarity. From control perspective, this means that the controlled reference LTI system and the controlled LPV system will have the same eigenvalues (poles) everywhere in the parameter domain – which entails that they will have the same dynamics and behavior.
When the similarity transformation matrix is the $\mathbf{I}$ unity matrix in appropriate dimension, then similarity described above occurs as $\mathbf{A}_{ref} - \mathbf{B}\mathbf{K}_{ref} = \mathbf{I}^{-1}\big(\mathbf{A}(\mathbf{p}(t)) - \mathbf{B}(\mathbf{K}_{ref} + \mathbf{K}(\mathbf{p}(t)))\big)\mathbf{I}$. This equality provides not just the similar dynamical behavior, but also the possibility to compute the parameter dependent $\mathbf{K}(\mathbf{p}(t))$ on the Ω parameter domain at every $\mathbf{p}(t)\ t \geq 0$.

$$\begin{aligned} &\mathbf{A}_{ref} - \mathbf{B}\mathbf{K}_{ref} = \mathbf{I}^{-1}\big(\mathbf{A}(\mathbf{p}(t)) - \mathbf{B}(\mathbf{K}_{ref} + \mathbf{K}(\mathbf{p}(t)))\big)\mathbf{I} = \\ &= \mathbf{A}(\mathbf{p}(t)) - \mathbf{B}(\mathbf{K}_{ref} + \mathbf{K}(\mathbf{p}(t))) \\ &\mathbf{K}(\mathbf{p}(t)) = -\mathbf{B}^{-1}(\mathbf{A}_{ref} - \mathbf{B}\mathbf{K}_{ref} - \mathbf{A}(\mathbf{p}(t)) + \mathbf{B}\mathbf{K}_{ref}) = -\mathbf{B}^{-1}(\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))) \end{aligned} \quad . \tag{7}$$

Hence, the control law becomes:

$$\mathbf{u}(t) = -(\mathbf{K}_{ref} + \mathbf{B}^{-1}(\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))))\mathbf{x}(t) \quad . \tag{8}$$

Therefore, the (6) should be modified accordingly to (7) which leads back to (3) as

we can see below due to the equality criteria.

$$\begin{aligned}\dot{\mathbf{x}}(t) = &\big(\mathbf{A}(\mathbf{p}(t)) - \mathbf{B}(\mathbf{K}_{ref} + \mathbf{K}(\mathbf{p}(t)))\big)\mathbf{x}(t) = \\ &= \big(\mathbf{A}(\mathbf{p}(t)) - \mathbf{B}(\mathbf{K}_{ref} - \mathbf{B}^{-1}(\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))))\big)\mathbf{x}(t) = \\ &= \big(\mathbf{A}(\mathbf{p}(t)) - \mathbf{B}\mathbf{K}_{ref} + \mathbf{B}\mathbf{B}^{-1}\mathbf{A}_{ref} - \mathbf{B}\mathbf{B}^{-1}\mathbf{A}(\mathbf{p}(t)))\big)\mathbf{x}(t) = \\ &= \big(\mathbf{A}(\mathbf{p}(t)) - \mathbf{B}\mathbf{K}_{ref} + \mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))\big)\mathbf{x}(t) = \\ &= \big(\mathbf{A}_{ref} - \mathbf{B}\mathbf{K}_{ref}\big)\mathbf{x}(t) \\ \mathbf{y}(t) = &\ \mathbf{C}\mathbf{x}(t)\end{aligned} \quad . \tag{9}$$

2.5 Design of Complementary LPV Observer

Assume that a state observer for the reference $\mathbf{S}_{ref}$ can be designed which requires the satisfaction of the observability criteria detailed above. According to (4) the closed-loop system matrix becomes $\mathbf{A}_{ref} - \mathbf{G}_{ref}\mathbf{C}$. The eigenvalues $\lambda(\mathbf{A}_{ref} - \mathbf{G}_{ref}\mathbf{C})$ define the dynamics of the reference observer. Similar to the control perspective, if we want to use the state observer regarding a given LPV system the parameter dependency has to be represented in the state observer as well. As previously, it is considered that only $\mathbf{A}(\mathbf{p}(t))$ can be parameter dependent and a given observed LPV system can be described as:

$$\dot{\hat{\mathbf{x}}}(t) = \big(\mathbf{A}(\mathbf{p}(t)) - \mathbf{G}(t)\mathbf{C}\big)\hat{\mathbf{x}}(t) + \mathbf{G}(t)\mathbf{y}(t) + \mathbf{H}\mathbf{u}(t) \quad . \tag{10}$$

Definition 9. *Complementary $\mathbf{p}(t)$ dependent observer gain*
A LPV observer gain $\mathbf{G}(t)$ consists of a $\mathbf{G}_{ref}$ reference and $\mathbf{G}(\mathbf{p}(t))$ varying observer gain as follows: $\mathbf{G}(t) := \mathbf{G}_{ref} + \mathbf{G}(\mathbf{p}(t))$. Therefore, from (10) $\big(\mathbf{A}(\mathbf{p}(t)) - \mathbf{G}(t)\mathbf{C}\big) = \big(\mathbf{A}(\mathbf{p}(t)) - (\mathbf{G}_{ref} + \mathbf{G}(\mathbf{p}(t))\mathbf{C}\big)$.

Assume that $\mathbf{A}_{ref} - \mathbf{G}_{ref}\mathbf{C} \sim \mathbf{A}(\mathbf{p}(t)) - (\mathbf{G}_{ref} + \mathbf{G}(\mathbf{p}(t)))\mathbf{C} \ \ \forall \mathbf{p}(t)$. The eigenvalues (poles) of the closed-loop reference LTI system are $\lambda_{ref} := \lambda(\mathbf{A}_{ref} - \mathbf{G}_{ref}\mathbf{C})$ and the eigenvalues (poles) of the closed-loop LPV system are $\lambda(\mathbf{p}(t)) := \lambda(\mathbf{A}(\mathbf{p}(t)) - (\mathbf{G}_{ref} + \mathbf{G}(\mathbf{p}(t)))\mathbf{C})$.
According to Theorem 2 the consequence of the similarity the $\lambda_{ref} = \lambda(\mathbf{p}(t)) \ \ \forall \mathbf{p}(t)$ $t \geq 0$. From control perspective point of view that means the reference LTI and the LPV observers will have the same eigenvalues (poles) everywhere in the parameter domain – which entails the they will have the same dynamics and behavior.
The similarity above requires that the transformation matrix be the $\mathbf{I}$ unity matrix in appropriate dimension. That is, $\mathbf{A}_{ref} - \mathbf{G}_{ref}\mathbf{C} = \mathbf{I}^{-1}\big(\mathbf{A}(\mathbf{p}(t)) - (\mathbf{G}_{ref} + \mathbf{G}(\mathbf{p}(t)))\mathbf{C}\big)\mathbf{I}$, thus $\mathbf{G}(\mathbf{p}(t))$ can be calculated on the Ω parameter domain at every $\mathbf{p}(t)$.

$$\begin{aligned}&\mathbf{A}_{ref} - \mathbf{G}_{ref}\mathbf{C} = \mathbf{I}^{-1}\big(\mathbf{A}(\mathbf{p}(t)) - (\mathbf{G}_{ref} + \mathbf{G}(\mathbf{p}(t)))\mathbf{C}\big)\mathbf{I} = \\ &= \mathbf{A}(\mathbf{p}(t)) - (\mathbf{G}_{ref} + \mathbf{G}(\mathbf{p}(t)))\mathbf{C} \\ &\mathbf{G}(\mathbf{p}(t)) = -(\mathbf{A}_{ref} - \mathbf{G}_{ref}\mathbf{C} - \mathbf{A}(\mathbf{p}(t)) + \mathbf{G}_{ref}\mathbf{C})\mathbf{C}^{-1} = -(\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)))\mathbf{C}^{-1}\end{aligned} \quad . \tag{11}$$

Therefore, the (10) should be modified accordingly to (11) which leads back to (4):

$$\begin{aligned}\dot{\hat{\mathbf{x}}}(t) &= \big(\mathbf{A}(\mathbf{p}(t)) - \mathbf{G}(\mathbf{p}(t))\mathbf{C}\big)\hat{\mathbf{x}}(t) + \mathbf{G}(\mathbf{p}(t))\mathbf{y}(t) + \mathbf{H}\mathbf{u}(t) = \\ &= \big(\mathbf{A}(\mathbf{p}(t)) - (\mathbf{G}_{ref} + \mathbf{G}(t))\mathbf{C}\big)\hat{\mathbf{x}}(t) + (\mathbf{G}_{ref} + \mathbf{G}(t))\mathbf{y}(t) + \mathbf{H}\mathbf{u}(t) = \\ &= \big(\mathbf{A}(\mathbf{p}(t)) - (\mathbf{G}_{ref} - (\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)))\mathbf{C}^{-1})\mathbf{C}\big)\hat{\mathbf{x}}(t) + \\ &+ (\mathbf{G}_{ref} - (\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)))\mathbf{C}^{-1})\mathbf{y}(t) + \mathbf{H}\mathbf{u}(t) = \\ &= \big(\mathbf{A}(\mathbf{p}(t)) - \mathbf{G}_{ref}\mathbf{C} + \mathbf{A}_{ref}\mathbf{C}^{-1}\mathbf{C} - \mathbf{A}(\mathbf{p}(t))\mathbf{C}^{-1}\mathbf{C}\big)\hat{\mathbf{x}}(t) + \\ &+ (\mathbf{G}_{ref} - (\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)))\mathbf{C}^{-1})\mathbf{y}(t) + \mathbf{H}\mathbf{u}(t) = \\ &= \big(\mathbf{A}_{ref} - \mathbf{G}_{ref}\mathbf{C}\big)\hat{\mathbf{x}}(t) + (\mathbf{G}_{ref} - (\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)))\mathbf{C}^{-1})\mathbf{y}(t) + \mathbf{H}\mathbf{u}(t)\end{aligned} \quad . \tag{12}$$

2.6 Consequences and limitations

As it was declared above, the $p_i(t)$ scheduling parameters has to be measured or estimated since these are directly used for tuning the developed controller and observer structures. The main limitations are the invertibility of **B** and **C** matrices – which are key properties regarding the applicability. Thus, if **B** and **C** are fully invertable, then (9) and (12) can be applied and complementary LPV controller and observer design is possible.
However, there are generalization possibilities based on Definition 4 - 7 which can be utilized in order to make the developed solutions more flexible.

2.6.1 Controller Side

1. *Appropriate selection of $p_i(t)$ scheduling parameters.*
It is possible to calculate the $\mathbf{K}(\mathbf{p}(t))$ matrix in element by element way without inversion of **B**. The key components are the selection and linking of $p_i(t)$.

Remark 5. The expression "linking" should be explained at this point. If we select a nonlinearity inducing term from a given equation, we can "link" it to a given state variable in a natural or forced way depending on the structure of the equation and the requirements detailed below. For example, assume the following simple equations:

$$\begin{aligned}\dot{x}_1(t) &= k_1 x_1(t) x_2(t) + k_2\sqrt{x_2(t)} + k_3 x_2(t) \\ \dot{x}_2(t) &= -k_2\sqrt{x_2(t)} - k_3 x_2(t) + u_1(t)\end{aligned} \quad . \tag{13}$$

In (13) two nonlinearity inducing elements can be found, $x_1(t)x_2(t)$ and $\sqrt{x_2(t)}$.
Natural linking: we can select $p_1(t) = k_1 x_2(t)$, which means we link $p_1(t)$ to $x_1(t)$ in this equation as $\dot{x}_1(t) = p_1(t)x_1(t) + k_2\sqrt{x_2(t)} + k_3 x_2(t)$. This linking is a natural choice and come from the structure of the equation.
Forced linking: we have to select $p_2(t) = k_2\sqrt{x_2(t)}$ and link to a state. It is possible by using simple manipulations, eg. multiplication by $1 = \dfrac{x_1(t)}{x_1(t)}$. Therefore, $k_2\sqrt{x_2(t)}\dfrac{x_1(t)}{x_1(t)}$ occurs and $p_2(t) = k_2\dfrac{\sqrt{x_2(t)}}{x_1(t)}$ will be the selected scheduling variable. Thus, $p_2(t)$ can be linked in a forced way to $x_1(t)$ as $\dot{x}_1(t) = p_1(t)x_1(t) +$

$p_2(t)x_1(t) + k_3x_2(t)$. Strong limitation is that $x_1(t) \neq 0 \ \forall t \geq 0$ in order to avoid singularity. With forced linking we can arbitrarily bound given terms as scheduling variables to selected states beside the mentioned limitation.
It has to be mentioned that in case of the forced linking, we have to be sure that denominator of the newly realized scheduling variable – the state to which the scheduling variable is linked – not only cannot be zero during operation. However, it also has to be measurable, since we use it as an external "input". The other solution is to estimate this state via nonlinear state estimators (such as the Kalman filter [12]).

From the control input side, the structure of $\mathbf{B}$ determines the selection and linking of $p_i(t)$. From (7) it is clear that $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ difference matrix does only contain elements in its structure where scheduling variables can be found – since the other elements of the matrices are the same and by the $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ subtraction become zero. However, in those entries which contain scheduling variables $p_{i,j,ref} - p_{i,j}(t)$ difference occur.
The structure of $\mathbf{B}$ will be also important. Suppose that every column and row of $\mathbf{B}$ does contain at most one non zero element regardless the position (entry) of the element in the structure of $\mathbf{B}$. That means that every state could have at most one different control input – which is reasonable in most of the physical and physiological systems. For example, in case of a system with three states which have three inputs, $\mathbf{B}$ can only contain one elements in each row, e.g..: $\mathbf{B} = \begin{bmatrix} 0 & b_2 & 0 \\ b_1 & 0 & 0 \\ 0 & 0 & b_3 \end{bmatrix}$.
Assume that the structure of $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ is such that it contains $p_{i,j,ref} - p_{i,j}(t)$ elements in only those rows where the rows of $\mathbf{B}$ does have non zero $b_{i,j}$ elements and the previous statement for $\mathbf{B}$ is true (columns and rows regardless the position does contain only one element). In this case, the elements of $\mathbf{K}(t)$, namely $k_{i,j}(t)$ can be calculated in an inverse way from the corresponding $p_{i,j,ref} - p_{i,j}(t)$ and $b_{i,j}$.
Thus, we know that $p_{i,j,ref} - p_{i,j}(t) = b_{i,j}k_{i,j}(t) \rightarrow k_{i,j}(t) = \dfrac{p_{i,j,ref} - p_{i,j}(t)}{b_{i,j}}$.
The last missing piece in this regard is to establish the equality of $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \mathbf{BK}(t)$, which is true when $\operatorname{rank}(\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))) = \operatorname{rank}(\mathbf{BK}(t))$. This rank criteria can be covered by the Definitions 4 - 7.
Assume that $\mathbf{BK}(t)$ can be decomposed to a dyad as $\mathbf{BK}(t) = \sum_{i=1}^{g} \mathbf{b}_i \mathbf{k}(\mathbf{p}(t))_i^\top$ and $\sum_{i=1}^{g} \mathbf{b}_i \mathbf{k}(\mathbf{p}(t))_i^\top$ is a minimal dyadic decomposition of $\mathbf{BK}(t)$. In this case, $\operatorname{rank}(\mathbf{BK}(t)) = g$. Having regard to this fact we have to select the scheduling variables in such a way that $\operatorname{rank}(\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))) = g$ as well. It is only possible, if the structure of $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ does contain g linearly independent columns (or rows). Then, the rank criteria automatically satisfies and $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \sum_{i=1}^{g} \mathbf{b}_i \mathbf{k}(\mathbf{p}(t))_i^\top$ which means that $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ can be described as the sum of g piece of dyadic products.
However, the number of restrictions seems high, but in practice most of them is automatically satisfied and with forced linking we can link the scheduling variables to

a given state arbitrarily.
For example, in case of a system with three states, two control inputs and two scheduling variables:

$$\mathbf{B} = \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \begin{bmatrix} p_{1,ref} - p_1(t) & 0 & 0 \\ 0 & p_{2,ref} - p_2(t) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \sum_{i=1}^{2} \mathbf{b}_i \mathbf{k}(\mathbf{p}(t))_i^{\top} = \begin{bmatrix} p_{1,ref} - p_1(t) & 0 & 0 \\ 0 & p_{2,ref} - p_2(t) & 0 \\ 0 & 0 & 0 \end{bmatrix} =$$

$$= \begin{bmatrix} b_1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \frac{p_{1,ref} - p_1(t)}{b_1} & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & \frac{p_{2,ref} - p_2(t)}{b_2} & 0 \end{bmatrix} \tag{14}$$

$$\mathbf{K}(t) = \begin{bmatrix} \frac{p_{1,ref} - p_1(t)}{b_1} & 0 & 0 \\ 0 & \frac{p_{2,ref} - p_2(t)}{b_2} & 0 \end{bmatrix}$$

$$\operatorname{rank}(\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))) = 2$$

2. *Control input virtualization.*
There are special opportunities to "virtually" increase the number of control input signals, if there is only one control input. Thus, the structure of **B** can be extended with additional columns with appropriate entries. From the control input side, it means that all state equations can be completed additionally with $u_{virt,i}(t)$ "virtual" inputs via the duplication of the real control input. In this case, the $u_{virt,i}(t)$ virtual input signals have to be equal to the real control input, namely $u_{virt,i}(t) = u_{real}(t)$ regardless of how many $u_{virt,i}(t)$ virtual inputs are considered. The main restriction will be that all of the rows of the realized $\mathbf{K}(t)$ have to be equal, which is the only way to reach the equality of $u_{virt,i}(t) = u_{real}(t)$. The usage of this technique requires the assumptions from the previous section regarding the structure of **B** and $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$.
The input virtualization technique will be introduced via a practical example. Assume a three state system – with $x_1(t)$, $x_2(t)$ and $x_3(t)$ states – which contains a control input signal in its third state equation and a selected scheduling variable can be found only in the third equation linked to the first state as follows:

$$\begin{aligned} \dot{x}_1(t) &= -a_1 x_1(t) + a_2 x_2(t) \\ \dot{x}_2(t) &= -a_2 x_2(t) + a_3 x_3(t) \\ \dot{x}_3(t) &= -x_1(t)\sqrt{x_3(t)} - a_3 x_3(t) + b_1 u_{real}(t) \end{aligned}, \tag{15}$$

where $p_1(t) = -\sqrt{x_3(t)}$ is selected as scheduling variable. In (15) $\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ b_1 \end{bmatrix}$ and

$\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ p_1(t) & 0 & 0 \end{bmatrix}$. Introduce two new virtual inputs into the previous equation:

$$\begin{aligned} \dot{x}_1(t) &= -a_1 x_1(t) + a_2 x_2(t) + c_1 u_{virt,1}(t) - c_1 u_{virt,1}(t) \\ \dot{x}_2(t) &= -a_2 x_2(t) + a_3 x_3(t) + c_2 u_{virt,2}(t) - c_2 u_{virt,2}(t) \\ \dot{x}_3(t) &= p_1(t) x_1(t) - a_3 x_3(t) + b_1 u_{real}(t) \end{aligned} , \tag{16}$$

In this case, an extended input matrix can be introduced: $\mathbf{B}_{ex} = \begin{bmatrix} c_1 b_1 & -c_1 b_1 & 0 \\ c_2 b_1 & 0 & -c_2 b_1 \\ b_1 & 0 & 0 \end{bmatrix}$, where $c_1 := 1[\dot{x}_1(t)/\dot{x}_3(t)]$ and $c_2 := 1[\dot{x}_2(t)/\dot{x}_3(t)]$ converter scalers take care of the appropriate units and $u_{virt,1}(t) = u_{virt,2}(t) = u_{real}(t)$. Since the concrete values of c_1 and c_2 are equal to 1, they will not be indicated in the followings.The extended $\mathbf{B}_{ex}$ is invertible, namely $\mathbf{B}_{ex}^{-1} = \begin{bmatrix} 0 & 0 & b_1 \\ -b_1 & 0 & b_1 \\ 0 & -b_1 & b_1 \end{bmatrix}$. In this case, $\mathbf{K}(t)$ can be calculated by using (7) such as:

$$\begin{aligned} &\mathbf{K}(t) = -\mathbf{B}^{-1}(\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))) = \\ &= -\begin{bmatrix} 0 & 0 & b_1 \\ -b_1 & 0 & b_1 \\ 0 & -b_1 & b_1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ p_{1,ref} - p_1(t) & 0 & 0 \end{bmatrix} = \\ &= -\begin{bmatrix} b_1(p_{1,ref} - p_1(t)) & 0 & 0 \\ b_1(p_{1,ref} - p_1(t)) & 0 & 0 \\ b_1(p_{1,ref} - p_1(t)) & 0 & 0 \end{bmatrix} \end{aligned} . \tag{17}$$

The mentioned key component is that $u_{virt,i}(t) = u_{real}(t)$ and the configuration of (15) will provide this restriction.
In general, the states feedback design does not modify in case of the reference LTI system, namely, the state feedback designing process have to be done by using the original $\mathbf{B} = \begin{bmatrix} 0 & 0 & b_1 \end{bmatrix}^\top$. The $\mathbf{K}_{ref}$ feedback gain will be a row matrix as $\mathbf{K}_{ref} = \begin{bmatrix} k_{1,ref} & k_{2,ref} & k_{3,ref} \end{bmatrix}$. In this given case to reach $u_{virt,i}(t) = u_{real}(t)$, we have to duplicate the rows of $\mathbf{K}_{ref}$ and realize an extended feedback gain matrix, such as $\mathbf{K}_{ref,ex} = \begin{bmatrix} k_{1,ref} & k_{2,ref} & k_{3,ref} \\ k_{1,ref} & k_{2,ref} & k_{3,ref} \\ k_{1,ref} & k_{2,ref} & k_{3,ref} \end{bmatrix}$. By using the extended $\mathbf{B}_{ex}$ in the control law description the virtual inputs will drop out from the given state equations and will be represented as an addition of zero in these (e.g. $+0 := +u_{virt} - u_{virt}$), which is a

realizable configuration by state feedback.

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{B}_{ex}\mathbf{u}_{ex}(t) =$$
$$= \begin{bmatrix} -a_1 & a_2 & 0 \\ 0 & -a_2 & a_3 \\ p_1(t) & 0 & -a_3 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} b_1 & -b_1 & 0 \\ b_1 & 0 & -b_1 \\ b_1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_{virt,1}(t) \\ u_{virt,2}(t) \\ u_{real}(t) \end{bmatrix} =$$

$$= (\mathbf{A}(\mathbf{p}(t)) - \mathbf{B}_{ex}(\mathbf{K}_{ref} + \mathbf{K}(t)))\mathbf{x}(t) =$$

$$= \left(\begin{bmatrix} -a_1 & a_2 & 0 \\ 0 & -a_2 & a_3 \\ p_1(t) & 0 & -a_3 \end{bmatrix} - \begin{bmatrix} b_1 & -b_1 & 0 \\ b_1 & 0 & -b_1 \\ b_1 & 0 & 0 \end{bmatrix} \right.$$
$$\left. \left(\begin{bmatrix} k_{1,ref} & k_{2,ref} & k_{3,ref} \\ k_{1,ref} & k_{2,ref} & k_{3,ref} \\ k_{1,ref} & k_{2,ref} & k_{3,ref} \end{bmatrix} - \begin{bmatrix} b_1(p_{1,ref} - p_1(t)) & 0 & 0 \\ b_1(p_{1,ref} - p_1(t)) & 0 & 0 \\ b_1(p_{1,ref} - p_1(t)) & 0 & 0 \end{bmatrix} \right) \right) \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} =$$

$$= \begin{bmatrix} -a_1 & a_2 & 0 \\ 0 & -a_2 & a_3 \\ p_1(t) - b_1(k_{1,ref} - b_1(p_{1,ref} - p_1(t))) & 0 - b_1 k_{2,ref} & -a_3 - b_1 k_{3,ref} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \tag{18}$$

From (18) it is clear that the input virtualization does not modify the first and second state equation via the state feedback, however, directly affects the third equation in which the control input occurs. At the same time, this construction provides the restriction from above in general (eigenvalue equality, rank criteria, etc.).
Naturally, other constructions can be imagined, but each case requires unique construction of $\mathbf{B}_{ex}$ and careful selection of $p_i(t)$. Regarding the given case, the same result occurs if all of the states have linked scheduling parameters in the third state equation (beside keeping in mind the limitations of selection of them). Namely, $\mathbf{A}(\mathbf{p}(t)) = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ p_1(t) + \mathrm{AD} & p_2(t) + \mathrm{AD} & p_3(t) + \mathrm{AD} \end{bmatrix}$, where AD means those additional coefficients of the states which are not embedded into a scheduling variable. Moreover, the difference matrix becomes

$$\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ p_{1,ref} - p_1(t) & p_{2,ref} - p_2(t) & p_{3,ref} - p_3(t) \end{bmatrix}. \tag{19}$$

It can be observed that the $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ difference matrix contains elements only in the third row which corresponds to that row in $\mathbf{B}$ which contains the real input coefficient.

Remark 6. It has to be noticed that the mentioned techniques which help to get around the invertibility issue of $\mathbf{B}$ strongly coupled to the complementary observer

design. The structure of $\mathbf{C}$ is similarly important and determines the usage of same techniques regarding the observer design.

In multi input case the input virtualization technique may not be applicable. It depends on the structure of $\mathbf{B}$ and $\mathbf{C}$ matrices – however, further generalization from this point of view is ongoing.

2.6.2 Observer Side

From (11) it is clear that the key point is the invertibility of $\mathbf{C}$. This is only possible if all of the states are represented in the output, so directly measurable or calculable. Otherwise, if $\mathbf{C}$ is not invertible, we have to face the same problem as in case of the invertiblity of $\mathbf{B}$. Although the same solution – appropriate selection of $p_i(t)$ from the output point of view – can be used for element by element calculation of $\mathbf{G}(t)$ observer gain. Virtualization of the output is meaningless from the output point of view.
The structure of $\mathbf{C}$ determines the selection and linking of $p_i(t)$. Equation (11) shows that $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ difference matrix does only contain elements in those entries where scheduling variables can be found which are equal to $p_{i,j,ref} - p_{i,j}(t)$ difference.
The other component to be investigated is the structure of $\mathbf{C}$. Assume that every rows and columns of $\mathbf{C}$ contain at most one non zero element regardless the position (entry) of the element in the structure of $\mathbf{C}$. From system point of view this assumption is reasonable, since in most of the physical or physiological systems each output connects to one state. For example, in case of a system with three states which have two outputs connected to $x_1(t)$ and $x_2(t)$ states, $\mathbf{C}$ can only be $\mathbf{C} = \begin{bmatrix} c_1 & 0 & 0 \\ 0 & c_2 & 0 \end{bmatrix}$. Assume that the structure of $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ such that it contains $p_{i,j,ref} - p_{i,j}(t)$ elements in only those columns where the columns of $\mathbf{C}$ does have non zero $c_{i,j}$ elements and the previous statement for $\mathbf{C}$ is true (columns and rows regardless the position does contain only one element). In this case, the elements of $\mathbf{G}(t)$, namely $g_{i,j}(t)$ can be calculated in the same inverse way as $\mathbf{K}(t)$ from the corresponding $p_{i,j,ref} - p_{i,j}(t)$ and $c_{i,j}$. Thus, we know that $p_{i,j,ref} - p_{i,j}(t) = g_{i,j}(t)c_{i,j} \rightarrow g_{i,j}(t) = \dfrac{p_{i,j,ref} - p_{i,j}(t)}{c_{i,j}}$.
The equality $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \mathbf{G}(t)\mathbf{C}$ holds if $\operatorname{rank}(\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))) = \operatorname{rank}(\mathbf{G}(t)\mathbf{C})$. Again, by using the consequences of Definitions 4 - 7 this rank criteria can be proven.

Assume that $\mathbf{G}(t)\mathbf{C}$ can be decomposed to a dyad as $\mathbf{G}(t)\mathbf{C} = \sum_{i=1}^{f} \mathbf{g}(\mathbf{p}(t))_i \mathbf{c}_i^\top$ and $\sum_{i=1}^{f} \mathbf{g}(\mathbf{p}(t))_i \mathbf{c}_i^\top$ is a minimal dyadic decomposition of $\mathbf{G}(t)\mathbf{C}$. In this case $\operatorname{rank}(\mathbf{G}(t)\mathbf{C}) = f$. Due to this fact, we have to select the scheduling variables in such a way that $\operatorname{rank}(\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))) = f$ as well. It is only possible if the structure of $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ does contain f linearly independent columns (or rows). Then the rank cri-

teria is automatically satisfied and $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \sum_{i=1}^{f} \mathbf{g}(\mathbf{p}(t))_i \mathbf{c}_i^{\top}$ which means that $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ can be described as the sum of f piece of dyadic products.
Similar to the previous case by forced linking of the scheduling variables and simple mathematical manipulations it can be achieved that the described conditions are automatically satisfied in practice.
For example, in case of a system with three states, two outputs and two scheduling variables:

$$\mathbf{C} = \begin{bmatrix} c_1 & 0 & 0 \\ 0 & c_2 & 0 \end{bmatrix}, \quad \mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \begin{bmatrix} p_{1,ref} - p_1(t) & 0 & 0 \\ 0 & p_{2,ref} - p_2(t) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \sum_{i=1}^{2} \mathbf{g}(\mathbf{p}(t))_i \mathbf{c}_i^{\top} = \begin{bmatrix} p_{1,ref} - p_1(t) & 0 & 0 \\ 0 & p_{2,ref} - p_2(t) & 0 \\ 0 & 0 & 0 \end{bmatrix} =$$

$$= \begin{bmatrix} \dfrac{p_{1,ref} - p_1(t)}{c_1} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} c_1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{p_{2,ref} - p_2(t)}{c_2} \\ 0 \end{bmatrix} \begin{bmatrix} 0 & c_2 & 0 \end{bmatrix} \tag{20}$$

$$\mathbf{G}(t) = \begin{bmatrix} \dfrac{p_{1,ref} - p_1(t)}{c_1} & 0 \\ 0 & \dfrac{p_{2,ref} - p_2(t)}{c_2} \\ 0 & 0 \end{bmatrix}$$

$$\operatorname{rank}(\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))) = 2$$

2.6.3 Connection between the Controller and Observer side

If the $\mathbf{B}$ and $\mathbf{C}$ is ivertible then the calculated $\mathbf{K}(t)$ and $\mathbf{G}(t)$ practically separated from each other.
By using the mentioned element-by-element calculation, the connection between the $\mathbf{K}(t)$ and $\mathbf{G}(t)$, furthermore between $\mathbf{B}$ and $\mathbf{C}$ is straightforward. The structure of $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ have to satisfy the requirements defined by the structures of $\mathbf{B}$ and $\mathbf{C}$. Namely, non zero elements can be in only those rows of $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ where the corresponding rows of $\mathbf{B}$ have non zero elements. Moreover, non zero elements can be in only those columns of $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ where the corresponding columns of $\mathbf{C}$ have non zero elements. Furthermore, the measurability of $\mathbf{p}(t)$ parameter vector has to be kept in mind all the time. If $\mathbf{p}(t)$ cannot be measured directly, estimation of it is needed, for example via Kalman filtering.
From realization point of view this means that the complementary controller and observer design have to be investigated in a strong conjunction and forced linking should be applied in order to reach the appropriate structure for $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ to

find the trade-off between the control and observer requirements come from the **B** and **C**.
To provide a full picture, the following practical example demonstrates this balancing between the requirements. Assume a given four state system with two inputs (connected to $x_3(t)$ and $x_4(t)$) and two outputs (connected to $x_1(t)$ and $x_3(t)$). First, we have to investigate where the $p_{i,ref} - p_i(t)$ differences can occur in the $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ embedded in the system matrix to test the applicability of the method. In this case, the investigation will be extended to the controller and observer parts as well from **B** and **C** point of view. Be $\mathbf{C} = \begin{bmatrix} c_1 & 0 & 0 & 0 \\ 0 & 0 & c_2 & 0 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ b_1 & 0 \\ 0 & b_2 \end{bmatrix}$.
Denote the entries of $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ with $\Delta a_{i,j}(t)$. According to the prescriptions regarding **B** and **C** non zero $\Delta p_i(t)$ elements in $\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))$ can be occurred only in $\Delta a_{3,1}(t), \Delta a_{3,3}(t), \Delta a_{4,1}(t)$ and $\Delta a_{4,3}(t)$. At this given case for calculate $\mathbf{K}(t)$ and $\mathbf{G}(t)$ the following equations can be written:

$$\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \mathbf{B}\mathbf{K}(t) \rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ b_1 & 0 \\ 0 & b_2 \end{bmatrix} \begin{bmatrix} \frac{\Delta a_{3,1}(t)}{b_1} & 0 & \frac{\Delta a_{3,3}(t)}{b_2} & 0 \\ \frac{\Delta a_{4,1}(t)}{b_1} & 0 & \frac{\Delta a_{4,3}(t)}{b_2} & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \Delta a_{3,1}(t) & 0 & \Delta a_{3,3}(t) & 0 \\ \Delta a_{4,1}(t) & 0 & \Delta a_{4,3}(t) & 0 \end{bmatrix}$$

$$\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \mathbf{G}(t)\mathbf{C} \rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{\Delta a_{3,1}(t)}{c_1} & \frac{\Delta a_{3,3}(t)}{c_2} \\ \frac{\Delta a_{4,1}(t)}{c_1} & \frac{\Delta a_{4,3}(t)}{c_2} \end{bmatrix} \begin{bmatrix} c_1 & 0 & 0 & 0 \\ 0 & 0 & c_2 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \Delta a_{3,1}(t) & 0 & \Delta a_{3,3}(t) & 0 \\ \Delta a_{4,1}(t) & 0 & \Delta a_{4,3}(t) & 0 \end{bmatrix} \tag{21}$$

where the complementary feedback and observer gains can be calculated as

$$\mathbf{K}(t) = \begin{bmatrix} \frac{\Delta a_{3,1}(t)}{b_1} & 0 & \frac{\Delta a_{3,3}(t)}{b_2} & 0 \\ \frac{\Delta a_{4,1}(t)}{b_1} & 0 & \frac{\Delta a_{4,3}(t)}{b_2} & 0 \end{bmatrix}, \quad \mathbf{G}(t) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{\Delta a_{3,1}(t)}{c_1} & \frac{\Delta a_{3,3}(t)}{c_2} \\ \frac{\Delta a_{4,1}(t)}{c_1} & \frac{\Delta a_{4,3}(t)}{c_2} \end{bmatrix}. \tag{22}$$

If the aforementioned restrictions and requirements are held for the calculation of the gains, then the connection between them is obvious from (20) and (21):

$$\begin{aligned} &\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \mathbf{BK}(t) = \mathbf{G}(t)\mathbf{C} \\ &\operatorname{rank}(\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \operatorname{rank}(\mathbf{BK}(t)) = \operatorname{rank}(\mathbf{G}(t)\mathbf{C}) \end{aligned} . \tag{23}$$

The suggested element-by-element calculation cannot be used all the time – it depends on the given system to be controlled and usability requires deep investigation of the possible LPV structures of the system.

2.7 Feed Forward Compensator

Due to the basic properties of classical state feedback control an additional feed forward compensator has to be embedded in the control loop. Without it the state feedback controller enforces the states (and though the outputs) to reach zero values over time during operation. In order to reach the desired steady state values of the output this $\mathbf{N}(\mathbf{p}(t)) = \begin{pmatrix} \mathbf{N}_x(\mathbf{p}(t)) \\ \mathbf{N}_u(\mathbf{p}(t)) \end{pmatrix}$ feed forward compensator should be $\mathbf{p}(t)$-dependent as well [13, 19, 20].
The $\mathbf{p}(t)$ dependent compensator matrices can be calculated as follows [13, 4]:

$$\begin{aligned} &\begin{bmatrix} \mathbf{A}(\mathbf{p}(t)) & \mathbf{B} \\ \mathbf{I}_n & \mathbf{0}_{n\times m} \end{bmatrix} \begin{bmatrix} \mathbf{N}_x(\mathbf{p}(t)) \\ \mathbf{N}_u(\mathbf{p}(t)) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{n\times m} \\ \mathbf{I}_m \end{bmatrix} \\ &\begin{bmatrix} \mathbf{N}_x(\mathbf{p}(t)) \\ \mathbf{N}_u(\mathbf{p}(t)) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(\mathbf{p}(t)) & \mathbf{B} \\ \mathbf{I}_n & \mathbf{0}_{n\times m} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0}_{n\times m} \\ \mathbf{I}_m \end{bmatrix} \end{aligned} , \tag{24}$$

where $\mathbf{I}_n$ is the feedback "selector" matrix (here is a unity matrix), $\mathbf{0}_{n\times m}$ is zero matrix and $\mathbf{I}_m$ is unity matrix.
The compensator does modify the state vector by substracting the desired steady-state from the actual state of the system, the steady-state is calculated as $\mathbf{N}_x(\mathbf{p}(t))r(t)$, where $r(t)$ is the reference signal in the time instant t, and it modifies the control input by adding the steady-state control input calculated as $\mathbf{N}_u(\mathbf{p}(t))r(t)$. Therefore, the controller is governed by the equations:

$$\begin{aligned} &\dot{\hat{\mathbf{x}}}(t) = \mathbf{F}\hat{\mathbf{x}}(t) + (\mathbf{G}_{ref} + (\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)))\mathbf{C}^{-1})\mathbf{y}(t) + \mathbf{Hu}(t) \\ &\mathbf{u}(t) = (\mathbf{K}_{ref} + \mathbf{B}^{-1}(\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t))))(\hat{\mathbf{x}}(t) - \mathbf{N}_x(\mathbf{p}(t))\mathbf{r}(t)) + \mathbf{N}_u(\mathbf{p}(t))\mathbf{r}(t) \end{aligned} . \tag{25}$$

2.8 Particular steps to realize complementary LPV controller and observer in practice

Here we have collected the main steps of the realization of the complementary LPV controller and observer structure.

- Realization of the appropriate $\mathbf{S}(\mathbf{p}(t))$ LPV model form from the original nonlinear model,

- Selection of the $\mathbf{S}(\mathbf{p}_{ref})$ reference LTI system (which is an underlying LTI system as well),
- Design of the $\mathbf{K}_{ref}$ reference state feedback controller with an arbitrary method, which is appropriate to handle the $\mathbf{S}(\mathbf{p}_{ref})$ reference LTI system,
- Design of the $\mathbf{G}_{ref}$ reference observer gain with an arbitrary method which is appropriate to observes the $\mathbf{S}(\mathbf{p}_{ref})$ reference LTI system,
- Realization of the complementary LPV controller and observer structure based on Fig. 1.

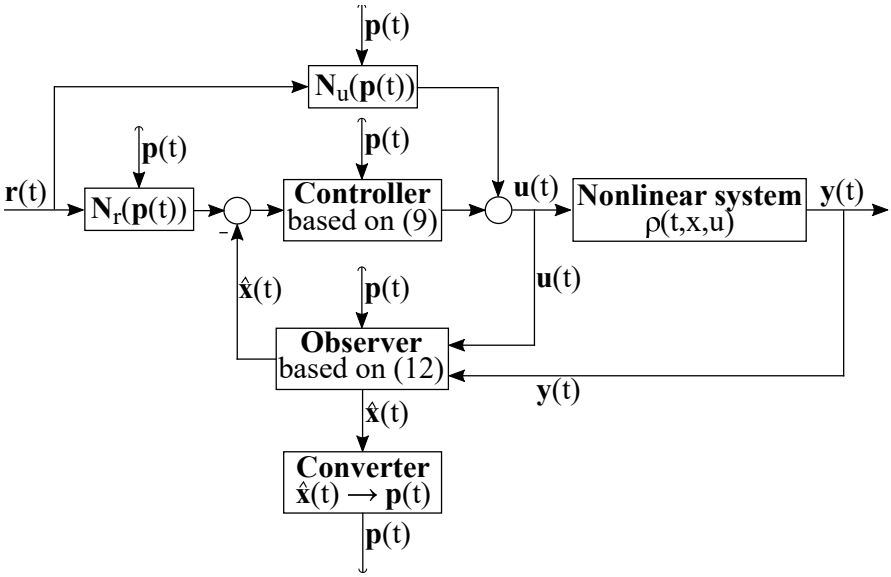


Figure 1
General feedback control loop with completed controller and observer.

3 Control of Innate Immune Response

In this section a spectacular physiological control example will be demonstrated by using the aforementioned methods.
Control of the response of innate immune system for given loads is crucial in many cases these days, especially when the patient's quality of life depend on it. Organ transplantation as final solution in case of organ disorders and malfunctions requires strict immunosuppression in order to prevent the rejection of the transplanted organ [21]. Furthermore, in case of many autoimmune diseases the effective immunosuppression is the only way to avoid the self-destruction of the human body by automated mechanisms [22]. On the other hand, suppression of internal defense system could lead ot unwanted states, e.g. activates of carried, but inactive viruses or bacteria and less effective immune protection against cancer [23, 24]. For example, the resting cytomegalovirus infection – which does not cause problems for a

healthy human – may causes serious problems for people with transplanted kidney or liver. The virus, even if it comes together with a donor organ or belongs to the recipient may lead to massive inflammation and critical state of different organs, if the immune suppression is strong [21].
Therefore, the accurate description of the immune response by mathematical models which can be basis for control design is very useful in these cases. In the following, we show a general model which can be adopted for various instances in order to describe the dynamics of infections and the response of the immune system for that.

3.1 Applied Model

The mathematical description of the used general theoretical model appeared in [25]. This model has beneficial properties, because it is able to describe the dynamics of several different diseases and its structure can be dynamically transformed in order adapt to the particular disease to be modeled and/or controlled.
$x_1(t)$ is the concentration of a pathogen, however, this can be measured by the concentration of associated antigen, $x_2(t)$ is the concentration of plasma the cells carrying and producing the antibodies, $x_3(t)$ is the concentration of antibodies which kill the pathogen and $x_4(t)$ is the relative characteristics of the damaged organ (where $x_4 = 0$ and $x_4 = 1$ mean the "healthy" and "dead" conditions, respectively). In general, the values of the states cannot be lower than zero ($x_i(t) \geq 0$, $i = [1,2,3,4]$, $\forall t \geq 0$).
An important property of the model has to be highlighted, namely, the $x_i(t)$ states are concentrations, however, the concrete units are not given due to the model is general in the given form and can be adopted to a wide range of cases. The same is true for the time span as well, namely, it can be arbitrarily determined to make the applicability of the model more flexible. Because of these facts, in this study the concrete type of concentration and time span are handled as "general units", without specification – similarly as [25].
The model consists of the following ordinary and delayed differential equations:

$$\dot{x}_1(t) = (a_{11} - a_{12}x_3(t))x_1(t) + b_1u_1(t), \tag{26a}$$

$$\dot{x}_2(t) = a_{21}(x_4(t))a_{22}x_1(t-\tau)x_3(t-\tau) - a_{23}(x_2(t) - x_2^*) + b_2u_2(t), \tag{26b}$$

$$\dot{x}_3(t) = a_{31}x_2(t) - (a_{32} + a_{33}x_1(t))x_3(t) + b_3u_3(t), \tag{26c}$$

$$\dot{x}_4(t) = a_{41}x_1(t) - a_{42}x_4(t) + b_4u_4(t), \tag{26d}$$

where $a_{11} = 1$, $a_{12} = 1$, $a_{22} = 3$, $a_{23} = 1$, $a_{31} = 1$, $a_{32} = 1.5$, $a_{33} = 0.5$, $a_{41} = 1$, $a_{42} = 1$, $b_1 = -1$, $b_2 = 1$, $b_3 = 1$ and $b_4 = -1$ are constant parameters of the model. In this study, the applied values of the parameters were the same as in [25]. The model does contain a saturation as follows:

$$a_{21}(x_4(t)) = \begin{cases} cos\pi x_4(t), & 0 \leq x_4(t) \leq 0.5 \\ 0, & \text{otherwise} \end{cases}. \tag{27}$$

In this case – similarly to [25] – the τ constant time delays are not taken into consideration in states $x_1(t)$ and $x_3(t)$.
In order to highlight what are the critical parts which shall be handled as scheduling variables (25) can be described in extended and completed form.

$$\dot{x}_1(t) = (a_{11} - a_{12}x_3(t))x_1(t) + b_1u_1(t) = p_1(t)x_1(t) + b_1u_1(t), \tag{28a}$$

$$\begin{aligned} \dot{x}_2(t) = \quad & a_{21}(x_4(t))a_{22}x_1(t)x_3(t) - a_{23}(x_2(t) - x_2^*) + b_2u_2(t) = \\ & a_{21}(x_4(t))a_{22}x_1(t)x_3(t) - a_{23}x_2(t) + a_{23}x_2^*1 + b_2u_2(t) = \\ & a_{21}(x_4(t))a_{22}x_3(t)x_1(t) - a_{23}x_2(t) + \frac{a_{23}x_2^*}{x_3(t)}x_3(t) + b_2u_2(t) = \\ & p_2(t)x_1(t) - a_{23}x_2(t) + p_3(t)x_3(t) + b_2u_2(t) \end{aligned} \tag{28b}$$

$$\begin{aligned} \dot{x}_3(t) = \quad & a_{31}x_2(t) - a_{32}x_3(t) - a_{33}x_3(t)x_1(t) + b_3u_3(t) = \\ & a_{31}x_2(t) - a_{32}x_3(t) + p_4(t)x_1(t) + b_3u_3(t), \end{aligned} \tag{28c}$$

$$\dot{x}_4(t) = a_{41}x_1(t) - a_{42}x_4(t) + b_4u_4(t), \tag{28d}$$

where $p_1(t) = a_{11} - a_{12}x_3(t)$, $p_2(t) = a_{21}(x_4(t))a_{22}x_3(t)$, $p_3(t) = \dfrac{a_{23}x_2^*}{x_3(t)}$ and $p_4(t) = -a_{33}x_3(t)$ are the selected scheduling variables, respectively. Hence, the parameter vector becomes $\mathbf{p}(t) = [p_1(t), p_2(t), p_3(t), p_4(t)]^\top$. Therefore, a $4D$ parameter space occurs.

The outputs of such a theoretical system is not predefined, but also depend on the given application. In this study, the followings are considered: $x_1(t)$, $x_3(t)$ and $x_4(t)$ are selected as outputs, namely these are measurable. The concentration of possible pathogens are usually higher than the concentration of associated antigens [26, 21] – this is taken into account with a scaler c_1 at the output side of $x_1(t)$ – therefore, $c_1x_1(t)$ term is handled as measurable outputs. In this way, the outputs of the system are $y_1(t) = c_1x_1(t)$, $y_2(t) = c_2x_3(t)$ and $y_3(t) = c_3x_4(t)$ where $c_1 = 1.5$, $c_2 = 1$ and $c_3 = 1$, respectively. Now, the system matrices of the LPV system arises as follows:

$$\mathbf{A}(\mathbf{p}(t)) = \begin{bmatrix} p_1(t) & 0 & 0 & 0 \\ p_2(t) & -a_{23} & p_3(t) & 0 \\ p_4(t) & a_{31} & -a_{32} & 0 \\ a_{41} & 0 & 0 & -a_{42} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} b_1 & 0 & 0 & 0 \\ 0 & b_2 & 0 & 0 \\ 0 & 0 & b_3 & 0 \\ 0 & 0 & 0 & b_4 \end{bmatrix}.$$
$$\mathbf{C} = \begin{bmatrix} c_1 & 0 & 0 & 0 \\ 0 & 0 & c_2 & 0 \\ 0 & 0 & 0 & c_3 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{29}$$

The LPV system can be written in compact form:

$$\begin{pmatrix}\dot{\mathbf{x}}(t)\\ \mathbf{y}(t)\end{pmatrix} = \mathbf{S}(\mathbf{p}(t))\begin{pmatrix}\mathbf{x}(t)\\ \mathbf{u}(t)\end{pmatrix} = \begin{bmatrix}\mathbf{A}(\mathbf{p})(t) & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}\begin{pmatrix}\mathbf{x}(t)\\ \mathbf{u}(t)\end{pmatrix} = \\ = \begin{bmatrix} p_1(t) & 0 & 0 & 0 & b_1 & 0 & 0 & 0\\ p_2(t) & -a_{23} & p_3(t) & 0 & 0 & b_2 & 0 & 0\\ p_4(t) & a_{31} & -a_{32} & 0 & 0 & 0 & b_3 & 0\\ a_{41} & 0 & 0 & -a_{42} & 0 & 0 & 0 & b_4\\ c_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & c_2 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & c_3 & 0 & 0 & 0 & 0 \end{bmatrix}\begin{pmatrix}\mathbf{x}(t)\\ \mathbf{u}(t)\end{pmatrix} . \quad (30)$$

3.2 Design of the Complementary LPV Controller

The first step is to determine the reference LTI model. Assume that a set of favorable states is $\mathbf{x}_{favorable} = [0, x_2^*, (2x_2^*/3), 0]^\top$ which describes healthy condition without presence of pathogens [25]. Through $\mathbf{x}_{favorable}$ the $\mathbf{p}_{ref} = [-0.3333, 0, 1.5, -0.6667]$ can be used. Moreover, the $\mathbf{r}$ reference vector can be selected to be equal to $\mathbf{x}_{favorable}$, as the desired values which have to be reached by the states over time ($\mathbf{r} = \mathbf{x}_{favorable} = [0, x_2^*, (2x_2^*/3), 0]^\top$ over $t \to \infty$). The $\mathbf{A}(\mathbf{p}_{ref})$ becomes

$$\mathbf{A}(\mathbf{p}_{ref}) = \begin{bmatrix} -0.3333 & 0 & 0 & 0\\ 0 & -1 & 1.5 & 0\\ -0.6667 & 1 & -1.5 & 0\\ 1 & 0 & 0 & -1 \end{bmatrix} . \quad (31)$$

The eigenvalues of the $\mathbf{A}(\mathbf{p}_{ref})$ are $\lambda(\mathbf{A}(\mathbf{p}_{ref})) = [-2.5, 0, -1, -0.3333]^\top$, so the reference LTI system is close to the border of stability because of its pole at 0.
The rank of the **Co** controllability matrix is equal to 4, namely $rank(\mathbf{Co}) = 4 \equiv n$, i.e. the reference LTI system is controllable. Therefore, it is possible to design a reference states feedback controller $\mathbf{K}_{ref}$ so that $\mathbf{u}(t) = -\mathbf{K}_{ref}\mathbf{x}(t)$.
The $\mathbf{K}_{ref}$ optimal gain (for LQ optimal state feedback controller) can be designed by using the MATLAB^{TM} *care* command. The design parameters in this given case are assumed to be: $\mathbf{Q} = \mathrm{diag}(20, 10, 10, 10)$ and $\mathbf{R} = \mathrm{diag}(1/20, 1/5, 1/5, 1/5)$, which provide fast poles and higher feedback gain at the critical state $x_1(t)$. The *care* command does calculate $\mathbf{X}$ as the unique solution of the control algebraic Ricatti equation (in the continuous-time domain) [27] as follows

$$\mathbf{A}^\top\mathbf{X}\mathbf{E} + \mathbf{E}^\top\mathbf{X}\mathbf{A} - (\mathbf{E}^\top\mathbf{X}\mathbf{B} + \mathbf{S})\mathbf{R}^{-1}(\mathbf{B}^\top\mathbf{X}\mathbf{E} + \mathbf{S}^\top) + \mathbf{Q} = \mathbf{O} , \quad (32)$$

where the calculated optimal gain – besides $\mathbf{S} = \mathbf{0}$ and $\mathbf{E} = \mathbf{I}$ – is equal to $\mathbf{K}_{ref} = \mathbf{R}^{-1}(\mathbf{B}^\top\mathbf{X}\mathbf{E} + \mathbf{S}^\top)$. The obtained optimal gain for the reference LTI system is:

$$\mathbf{K}_{ref} = \begin{bmatrix} -19.7359 & 0.4638 & 0.6369 & -0.9001\\ -0.1159 & 6.2073 & 1.0364 & 0.0073\\ -0.1592 & 1.0364 & 5.8609 & 0.0101\\ -0.2250 & -0.0073 & -0.0101 & -6.1272 \end{bmatrix} . \quad (33)$$

The $\mathbf{K}_{ref}$ feedback gain does provide that the eigenvalues of the closed loop become $\lambda_{ref,closed} = [-19.9798, -7.1725, -7.3062 + 0.1332i, -7.3062 - 0.1332i]^\top$ via $\mathbf{A}(\mathbf{p}_{ref}) - \mathbf{B}\mathbf{K}_{ref}$. The higher negative real parts of the eigenvalues provide fast transient part and stability, moreover, the occurring small complex parts do not cause high transient excursion.
Since $\mathbf{B}$ is invertible, the (9) can be used directly to calculate $\mathbf{K}(t)$ and realize the complementary LPV controller structure. Through the developed control law the structure of the complementary controller does provide that the strict equality of (7) will be satisfied over the operation. Namely, the $\lambda_{LPV,closed} = \lambda_{ref,closed}\ \forall t(t \geq 0)$ everywhere in the state space (and parameter space) regardless the actual value of $\mathbf{p}(t)$ parameter vector.

3.3 Design of the Complementary LPV Observer

The design of the complementary LPV observer is similar to the previous controller case and the calculation steps follow the same straightforward path.
First, the observability of the reference LTI system has to be investigated as the critical point of the design. The rank of the observability matrix determines whether the system is observable or not. In this particular case, the rank of the **Ob** observability matrix $rank(\mathbf{Ob}) = 4 \equiv n$, i.e. the reference LTI system is observable.
The $\mathbf{G}_{ref}$ reference observer gain can be calculated by using the MATLAB's *place* command [27]. In practice, the eigenvalues of $\mathbf{A} - \mathbf{L}\mathbf{C}$ should have more negative real parts (should be lower) than the system matrix in the $\lambda_{ref,closed}$ closed loop the eigenvalues in order to reach good observer dynamics (fast response and adaptivity). Consider, that $\lambda_{obs} = [-41, -43, -45, -47]\top$, where $\lambda_{obs,i} > \lambda_{ref,closed,i}$ $i = [1,2,3,4]$.
The resulting $\mathbf{G}_{ref}$ becomes:

$$\mathbf{G}_{ref} = 10^3 \begin{bmatrix} 0.0298 & -0.0000 & 0 \\ -0.2951 & 1.9354 & 0 \\ -0.0071 & 0.0875 & 0 \\ 0.0007 & 0 & 0.0400 \end{bmatrix}. \tag{34}$$

Since, $\mathbf{C}$ is not invertible and (12) cannot be used directly, we have to apply the design process from Sec. 2.6.2. In this case $\mathbf{G}(t)$ can be calculated based on (21) and (22).

$$\mathbf{A}_{ref} - \mathbf{A}(\mathbf{p}(t)) = \mathbf{G}(t)\mathbf{C} = \sum_{i=1}^{2} \mathbf{g}(\mathbf{p}(t))_i \mathbf{c}_i^\top \rightarrow \mathbf{G}(t)$$

$$\mathbf{G}(t) = \begin{bmatrix} \frac{\Delta p_1(t)}{c_1} & 0 & 0 \\ \frac{\Delta p_2(t)}{c_1} & \frac{\Delta p_3(t)}{c_2} & 0 \\ \frac{\Delta p_4(t)}{c_1} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{-0.3333 - p_1(t)}{c_1} & 0 & 0 \\ \frac{0 - p_2(t)}{c_1} & \frac{-1 - p_3(t)}{c_2} & 0 \\ \frac{-0.6667 - p_4(t)}{c_1} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{35}$$

The $rank(\mathbf{A}_{ref} - \mathbf{A}(t)) = 2$ which is equal to the number of dyads in the minimal dyadic structure. After calculating $\mathbf{G}(t)$ the (12) can be used to realize the complementary observer structure and because of the above described similarity the eigenvalue equality will be satisfied.The last missing piece is the feed forward compensation from Sec. 2.7. By using (24) the $\mathbf{N}_x(\mathbf{p}(t))$ and $\mathbf{N}_u(\mathbf{p}(t))$ can be calculated continuously during operation.
In this way the final control loop is realizable in accordance with Fig. 1.

3.4 Results

In this section the results of the simulations are detailed. Since the aim of the complementary LPV controller and observer is to enforce a particular LPV system and through the original nonlinear system to behave as the given LTI reference system, the focus during the presentation of the results is to highlight this property. In order to do that – beside keeping in mind the constraints – the corresponding signals of the nonlinear and LTI reference system will be presented and compared to each other. The simulations are carried out with the following system models:

1. Reference LTI system $\mathbf{S}_{ref}$: state vector $\mathbf{x}_{LTI}(t)$, observed reference state vector $\hat{\mathbf{x}}_{LTI}(t)$, output vector $\mathbf{y}_{LTI}(t)$, the permanent parameter vector $\mathbf{p}_{ref}$.

2. Original nonlinear system with complementary LPV controller and observer: state vector $\mathbf{x}_{orig}(t)$, observed state vector $\hat{\mathbf{x}}_{LPV}(t)$ coming from the complementary LPV observer (12), output vector of the nonlinear system $\mathbf{y}_{orig}(t)$, parameter vector $\mathbf{p}_{LPV}(t)$ generated by the observed states $\hat{\mathbf{x}}_{LPV}(t)$.

During the simulation the same settings were used in every case. The reference signal for the system states is the mentioned favorable steady values $\mathbf{r} = \mathbf{x}_{favorable} = [0, x_2^*, (2x_2^*/3), 0]^\top = [0, 2, 1.3333, 0]^\top$, hence the desired steady-state of the system to be $\mathbf{x}_\infty = \mathbf{r}$. The corresponding steady-state output is $\mathbf{y} = [0, 1.333, 0]^\top$. The initial state vector for every system was $\mathbf{x}(0) = [1.5, 3, 2, 0]^\top$. The initial state vector for the observers was equal to the desired steady-state values (since, this is known and determined) $\hat{\mathbf{x}}(0) = \mathbf{r}$. However, in this way there is an initial observation error, thus the dynamics of the observer can be analyzed. The simulation time was 1 time unit. This time span is enough to study the behavior of the system since all of the transients disappear under this time frame because of the fast operation and dynamics.
On Fig. 2. the variation of the states are presented over the simulated time span. Naturally, not all of the states are measurable ($x_1(t)$ and $x_2(t)$ cannot be measured directly). Albeit, – in order to reach a better understanding of the developed method – all of the corresponding states can be found on the diagram. The figure contains the following signals from the top to the bottom (started with the left column):

a) Vary of the $\mathbf{x}_{LTI}(t)$ states of the selected reference LTI system $\mathbf{S}_{ref}$ belongs to $\mathbf{p}_{ref}$

b) Vary of the $\mathbf{x}_{orig}(t)$ states of the original nonlinear time varying model

c) Comparison of the difference of the observed states based on the $\mathscr{L}_1(t)$ vector norm as follows: $\mathscr{L}_1(t) := \|\mathbf{x}_{LTI}(t) - \mathbf{x}_{orig}(t)\|_1$

d) Vary of the $\hat{\mathbf{x}}_{LTI}(t)$ observed states of the selected reference LTI system by the reference LTI observer

e) Vary of the $\hat{\mathbf{x}}_{LPV}(t)$ observed states of original nonlinear system coming from the complementary LPV observer

f) Comparison of the difference of the observed states based on the $\mathscr{L}_1(t)$ vector norm as follows: $\mathscr{L}_1(t) := \|\hat{\mathbf{x}}_{LTI}(t) - \hat{\mathbf{x}}_{LPV}(t)\|_1$

In practice, only the signals from Subfigs. d), e) and f) are available (accessible), since these are produced by the observers. However, the simulations can tell us useful information regarding the original states as well.

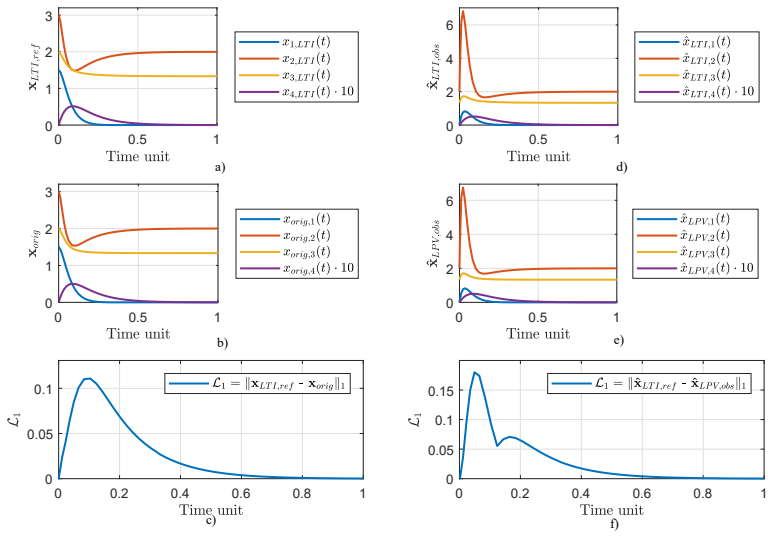


Figure 2
Comparison of the states during simulation. Due to the magnitude of x_4 was too low compared to other states it was multiplied by 10 to make it comparable, thus the order of magnitude became 10^1.

According to the simulations the developed complementary controller and observer structures performed well. Subfigs. a) and b) present the variation of the states of the reference LTI system ($\mathbf{S}_{ref}$) and the original nonlinear system, respectively. As it was stated, the $\mathbf{x}(0)$ initial values are the same in case of these systems. According to Subfig. c) there is a small deviation between the states based on the $\mathscr{L}_1(t)$ norm at the beginning which disappears over time. Furthermore, the magnitude of difference is small and can be neglected (since $\mathscr{L}_1(t) := \|\mathbf{x}_{LTI}(t) - \mathbf{x}_{orig}(t)\|_1$ which means there was only a small numerical difference between the states of the LTI reference system and the original nonlinear system).

Subfigs. d) and e) show that the same results regard to the reference observer and complementary LPV observer structure. The initial values of the observers were the same (and equal to the **r** reference). The Subfig. f) shows that the complementary observer structure performs well, thus the difference between the states of the observers were small and disappeared over time.
Finally, all of the states – with respect to the reference LTI system, the original nonlinear system, the reference LTI observer and the complementary LPV controlled system – reached the same final value what was the main target during operation regardless the variation of the parameter vector and the different initial conditions according to Fig. 2.

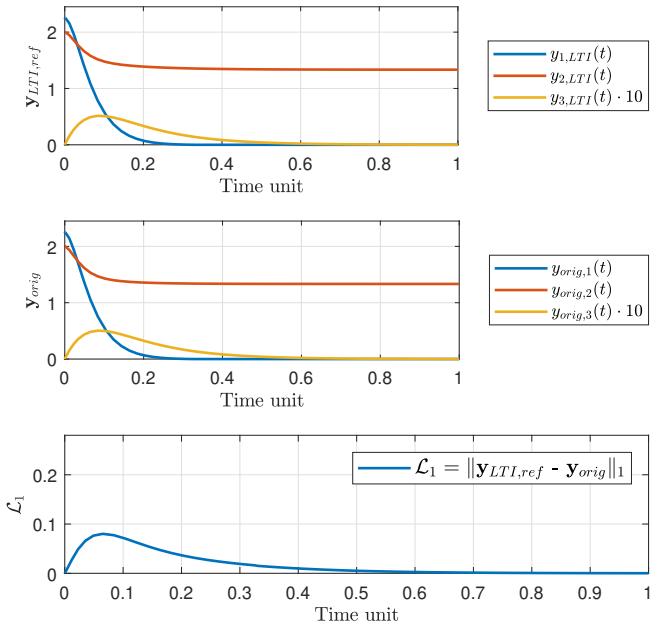


Figure 3
Comparison of the outputs during simulation. Due to the magnitude of y_3 was too low compared to other states it was multiplied by 10 to make it comparable, thus the order of magnitude became 10^1.

Figure 3 shows the variation of the output of the original nonlinear system ($\mathbf{y}_orig$) and the reference LTI system ($\mathbf{y}_{LTI,ref}$). As it can be seen the results correspond to the previous findings and the outputs behave as expected. At the beginning there is a small difference between the outputs (according to the defined $\mathscr{L}_1$) norm, but the deviation ceases over time. The results reflect that the complementary LPV controller and observer structure works well, thus it enforces that original nonlinear system to behave as the reference LTI system – and the numerical values of the outputs became equal over time as well.

The variation of the $\mathbf{p}(t)$ parameter vector converted from the observed states can be seen on Fig. 4. The figure strengthened the previous results, namely, regardless the variation of the parameter vector the completed LPV controller and observer structure is able to enforce the original nonlinear system to behave as the reference LTI system.

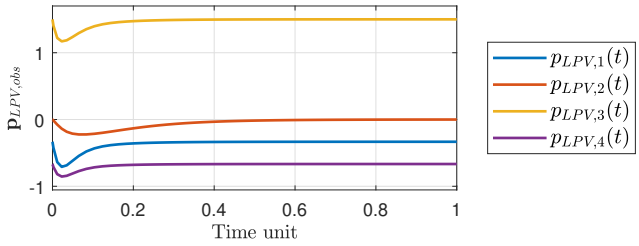


Figure 4
Vary of the parameter vector during simulation.

Conclusions

In this paper we presented a novel complementary LPV controller and observer design approach. The proposed method combines the classical state feedback with matrix similarity theorems, respectively. We analyzed the drawbacks, limitations and benefits of the introduced method.
The main advantages of this method is that it is able to provide appropriate, stable LPV controller and observer for the whole parameter domain by using a given reference LTI system as basis. Through the completed LPV controller and observer structure it is possible to enforce the nonlinear system to behave as the given LTI reference system.
We provided a practical example, namely, control of innate immune response. The results were satisfying since the completed LPV controller and observer structure was able to provide good control action and during operation the states of the reference LTI system and the original nonlinear system behaved similarly.
In our future work we are going to investigate the further generalization possibilities of the proposed techniques and we will try the methods in case of physical systems as well.

Acknowledgement

Gy. Eigner was supported by the ÚNKP-16-3/IV. New National Excellence Program of the Ministry of Human Capacities.
The author also thanks the support of the Robotics Special College of Óbuda University and the Applied Informatics and Applied Mathematics Doctoral School of Óbuda University.

References

[1] H.K. Khalil. *Nonlinear Control*. Prentice Hall, 2014.

[2] J.K. Tar, J.F. Bitó, and I. Rudas. Contradiction Resolution in the Adaptive Control of Underactuated Mechanical Systems Evading the Framework of Optimal Controllers . *ACTA Pol Hung*, 13(1):97 – 121, 2016.

[3] J.K. Tar, L. Nadai, and I.J. Rudas, editors. *System and Control Theory with Especial Emphasis on Nonlinear Systems*. Typotex, Budapest, Hungary, 1st edition, 2012.

[4] W.S. Levine. *The Control Engineering Handbook*. CRC Press, Taylor and Francis Group, Boca Raton, 2nd edition, 2011.

[5] L. Kovács. Linear parameter varying (LPV) based robust control of type-I diabetes driven for real patient data. *Knowl-Based Syst*, 122:199–213, 2017.

[6] Gy. Eigner. Novel LPV-based Control Approach for Nonlinear Physiological Systems . *ACTA Pol Hung*, 14(1):45–61, 2017.

[7] F. Bruzelius. *Linear parameter-varying systems: an approach to gain scheduling*. PhD thesis, Chalmers University of Technology, 2004.

[8] J. Mohammadpour and C.W. Scherer. *Control of Linear Parameter Varying Systems with Applications*. Springer New York, 2012.

[9] A.P. White, G. Zhu, and J. Choi. *Linear Parameter Varying Control for Engineering Applicaitons*. Springer, London, 1st edition, 2013.

[10] P. Baranyi, Y. Yam, and P. Varlaki. *Tensor Product Model Transformation in Polytopic Model-Based Control*. Series: Automation and Control Engineering. CRC Press, Boca Raton, USA, 1st edition, 2013.

[11] H. S. Abbas, R. Tóth, N. Meskin, J. Mohammadpour, and J. Hanema. A robust mpc for input-output lpv models. *IEEE Transactions on Automatic Control*, 61(12):4183–4188, 2016.

[12] K. Ogata. *The Biomedical Engineering Handbook*. Prentice Hall, Upper Saddle River, NJ, USA, 5th edition, 2010.

[13] B. Lantos. *Theory and design of control systems [in Hungarian]*. Akademia Press, Budapest, Hungary, 2nd edition, 2005.

[14] K. Zhou and J.C. Doyle. *Essentials of robust control*, volume 104. Prentice Hall, 1998.

[15] F. Wettl. *Linear Algebra [in Hungarian]*. Budapest University of Technology and Economy, Faculty of Natural Sciences, Budapest, Hungary, 1nd edition, 2011.

[16] R.A. Beezer. *A First Course in Linear Algebra*. Congruent Press, Washington, USA, version 3.40 edition, 2014.

[17] C.D. Meyer. *Matrix analysis and applied linear algebra*, volume 2. SIAM, Philadelphia, PA, USA, 2000.

[18] P. Rózsa. *Introduction to Matrix theorems [in Hungarian]*. Typotex, Budapest, Hungary, 2009.

[19] Gy. Eigner, P. Pausits, and L. Kovács. A novel completed lpv controller and observer scheme in order to control nonlinear compartmental systems. In *SISY 2016 – IEEE 14th International Symposium on Intelligent Systems and Informatics*, pages 85 – 92. IEEE Hungary Section, 2016.

[20] Gy. Eigner. *Working Title: Closed-Loop Control of Physiological Systems*. PhD thesis, Applied Informatics and Applied Mathemathics Doctoral School, Óbuda University, Budapest, Hungary, 2017. Manuscript. Planned defense: 2017.

[21] A.C. Guyton and J.E. Hall. *Textbook of Medical Physiology*. Elsevier, Philadelphia, USA, 2006.

[22] A.K. Abbas, A.H. Lichtman, and P. Shiv. *Basic Immunology: Functions and Disorders of the Immune System*. Elsevier, Philadelphia, USA, 4th edition, 2014.

[23] A.J. McMichael, P. Borrow, G.D. Tomaras, N. Goonetilleke, and B.F. Haynes. The immune response during acute HIV-1 infection: clues for vaccine development. *Nat Rev Immunol*, 10(1):11 – 23, 2009.

[24] M. Ravaioli, F. Neri, T. Lazzarotto, V.R. Bertuzzo, P. Di Gioia, G. Stacchini, M.C. Morelli, G. Ercolani, M. Cescon, A. Chiereghin, M. Del Gaudio, A. Cucchetti, and A.D. Pinna. Immunosuppression Modifications Based on an Immune Response Assay: Results of a Randomized, Controlled Trial. *Transplant*, 99(8):1625 – 1632, 2015.

[25] R.F. Stengel, R. Ghigliazza, N. Kulkarni, and O. Laplace. Optimal control of innate immune response. *Optim Contr Appl Met*, 23:91 – 104, 2002.

[26] A. Fonyó and E. Ligeti. *Physiology (in Hungarian)*. Medicina, Budapest, Hungary, 3rd edition, 2008.

[27] MATLAB. *Control System Toolbox Getting Started Guide*. The MathWorks, Inc, 2016.

A Heuristic Active Fault Tolerant Controller for the Stabilization of Spacecraft

Rouzbeh Moradi[1], Mohsen Fathi Jegarkandi[2] and Alireza Alikhani[3]

[1] Aerospace Research Institute (Ministry of Science, Research and Technology), P. o. Box: 14665-834, Tehran, Iran, rouzbeh_moradi@ari.ac.ir

[2] Department of Aerospace Engineering, Sharif University of Technology, P. o. Box: 11365-11155, Tehran, Iran, corresponding author, fathi@sharif.edu

[3] Aerospace Research Institute (Ministry of Science, Research and Technology), P. o. Box: 14665-834, Tehran, Iran, aalikhani@ari.ac.ir

Abstract: A heuristic active fault tolerant controller is designed based on the model of elections in a two-party democratic society. The goal of the proposed controller is to modify reference trajectories to maintain the stability of the faulty spacecraft. The elections are assumed to be first order Markovian. Final state constraints are used to ensure that the angular velocities asymptotically converge to the origin. A simulation shows that the proposed method makes the origin an asymptotically stable equilibrium point for the considered spacecraft. Because of its computational efficiency, the proposed method can be effectively used on-line in real-time, an important feature of any active fault tolerant controller. The present paper shows that socio-political models have a great potential to solve complex engineering problems and opens a new window for further developments in this field.

Keywords: Active fault tolerant control, two-party democratic society, reference trajectory management, under-actuated spacecraft

1 Introduction

Fault Tolerant Control (FTC) is an active research area in automatic control theory. The importance of FTC comes from the fact that the conventional feedback control systems are not capable of handling component malfunctions [12]. An almost countless number of books, review papers, research papers and theses have been published in the literature to cover an aspect of this important problem. [11] is a review paper that studies recent developments in the spacecraft attitude fault tolerant control system.

FTC is divided into two main parts: Active FTC (AFTC) and Passive FTC (PFTC). AFTC uses the on-line information provided by fault detection and diagnosis (FDD) to reconfigure the controller after the occurrence of a fault/failure in the system. In PFTC, a robust control method is used to make the closed-loop system as insensitive as possible to a range of anticipated faults and contrary to AFTC, there are no FDD and reconfiguration mechanisms. Therefore, the goal of designing PFTC is to provide a fixed structure controller such that the closed-loop system shows the least sensitivity to anticipated faults in the design stage. According to this classification, the presence of FDD and a mechanism to reconfigure the controller are two main features distinguishing AFTC from PFTC [12].

Reference trajectory management (RTM) is one of the components of general active fault tolerant controller [12]. The responsibility of RTM [2] is to adjust the reference trajectories, to make the post-fault model of the system stable, even after the occurrence of multiple actuator faults [4].

This paper proposes a heuristic AFTC that is based on the model of elections in a two-party democratic society [8]. Then, the proposed method is used to design an RTM block for a faulty spacecraft. The RTM produces desired reference trajectories to make the origin an asymptotically stable equilibrium point for the post-fault model. The main assumption of the paper is that two-party democratic societies are more stable than the other political systems [8].

A combination of global and local search optimization is used to satisfy the final state constraints. Due to its simple structure, the proposed method has a very low computational complexity and is suitable for on-line and real-time purposes.

The main contribution of this paper is to use socio-political models to solve engineering control problems. To the authors' best knowledge, previous studies have not considered such a methodology. The results show that the main idea has great potential for further research in the future.

This paper consists of the following sections: Section 2 presents the general scheme of the closed-loop system, an introduction to the election process in two-party democratic societies and finally, modeling the election process. Section 3 discusses the rotational dynamics of a rigid spacecraft and the controller structure. Asymptotic stability is discussed in section 4. Section 5 presents numerical results, and finally, the paper ends with a conclusion.

2 RTM Structure

2.1 The Closed-Loop System

The goal of the proposed RTM is to generate the desired reference trajectories ($\mathbf{r}_d$) to make the origin an asymptotically stable equilibrium point for the post-fault system. Fig. 1 shows the general scheme of the closed-loop system:

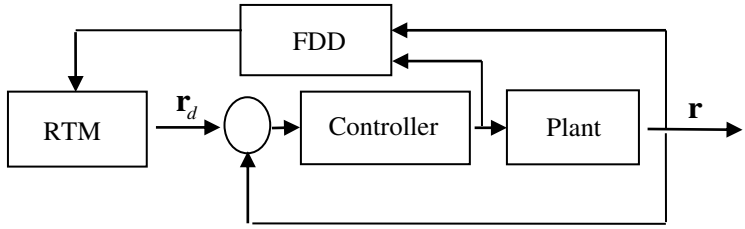


Figure 1
Closed-loop system

It is assumed that the post-fault model of the system is provided by the FDD. The desired reference trajectory vector should steer the post-fault model to the origin, even after occurrence of severe actuator faults/failures.

2.2 An Introduction to the Election Process in Two-Party Democratic Societies

The election system in two-party democratic societies is the basis of the proposed RTM. [8] states that two-party democracies tend to be more stable than the other political systems. On the other hand, there is historical evidence that the countries with multi-party democracies are also converging to two-party political systems [1].

One of the main features of a two-party system is as follows: if a destabilizing effect occurs in the society, e.g. a war or economic crisis, the opposition party will be selected by the public [8].

According to [10], it is possible to map any political spectrum to the left-right axes. This paper suggests Fig. 2 as a possible mapping:

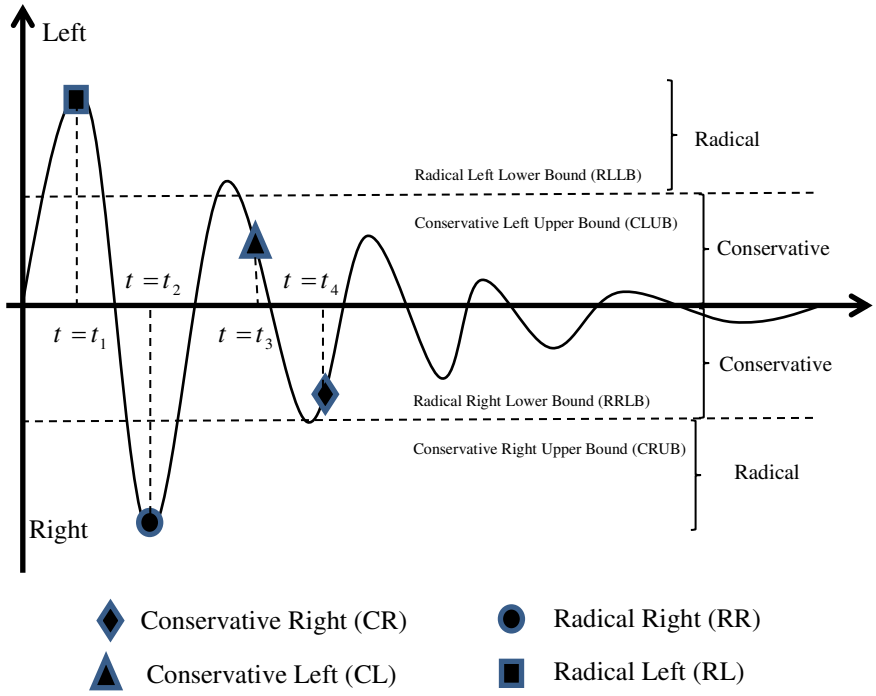


Figure 2
Political spectrum in the response diagram

Fig. 2 shows the upper and lower bounds of the conservative and radical policies. There is a total of four bounds that define the boundaries of the political spectrum. However, since RLLB = CLUB and RRLB = CRUB (Fig. 2), the total number of bounds for a state is reduced to two.

A conservative policy adheres to tradition and opposes any radical social change [7]. According to this definition, a conservative policy (whether it belongs to the left or right parties) favors being close to the equilibrium. The opposite is true about radical policies.

These facts are the basis of three important claims in this paper:

a) In a fast converging society, the public is satisfied with the ruling party and votes for the *conservative* policies.

b) In a moderately converging society, the public votes for the current policies.

c) In a slowly converging/diverging society, the public votes for the *opposition* party.

Assumption 1: The elections are assumed to be first order Markovian. In other words, the result of the current election is only dependent on the result of the previous election. This assumption is directly adopted from [8].

2.2 Modeling the Election Process

Fig. 3 shows the general scheme of the election process:

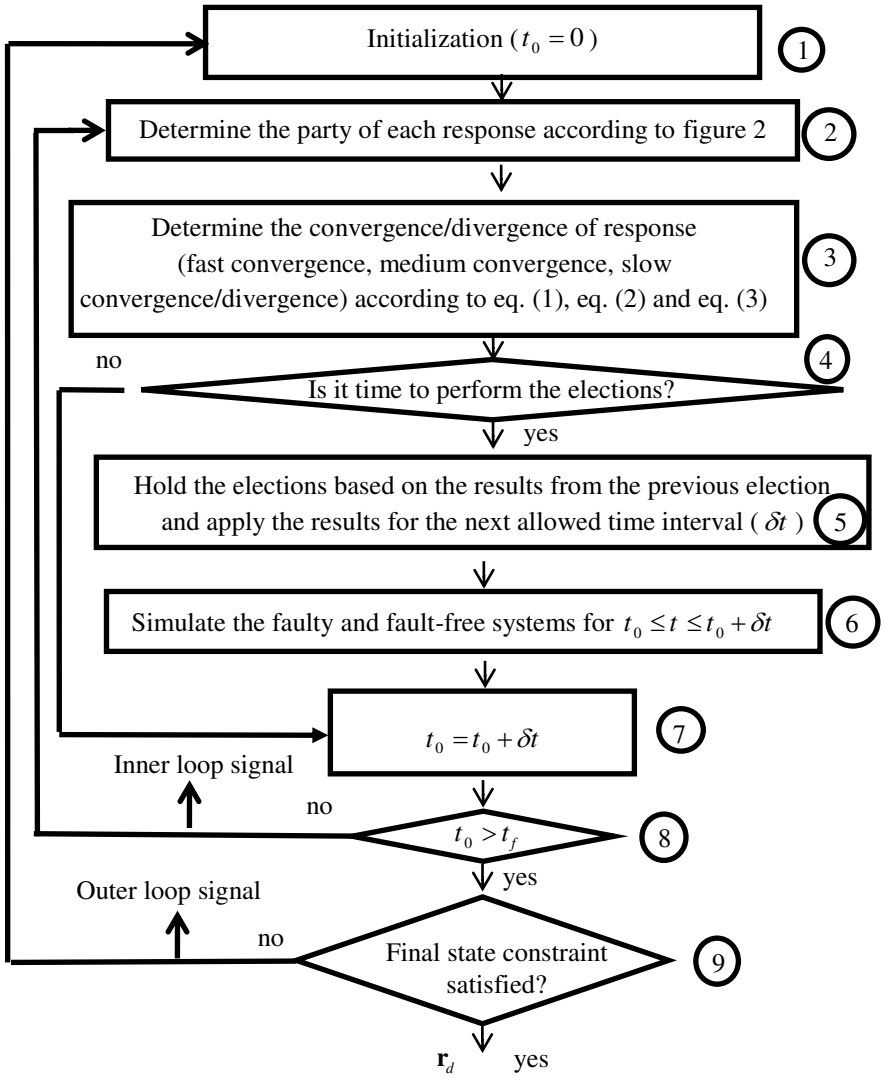


Figure 3
General scheme of the proposed method

This flowchart consists of two loops: an inner loop and an outer loop. According to Fig. 3, a description of the procedure is as follows:

1-The initial conditions, the election parameters (to be defined in this section) and the controller parameters are defined, and t_0 is set to zero.

2-The party of each state is determined, according to Fig. 2.

3-After determining parties, the regime of states is evaluated. Three main regimes are considered: fast convergence, medium convergence, slow convergence/divergence.

These regimes are determined according to the following criteria (where x_i and x_{n_i} are the *i-th* components of the faulty and nominal (fault-free) state vectors, respectively):

Fast convergence:

$$\sum_{i=1}^{n}\left|x_i\left(t_0+\delta t\right)-x_{n_i}\left(t_0+\delta t\right)\right| \le \beta_1 \sum_{i=1}^{n}\left|x_i\left(t_0\right)-x_{n_i}\left(t_0\right)\right| \qquad 0 \le \beta_1 < 1 \tag{1}$$

β_1 is the *fast convergence coefficient* and *n* is the number of states.

Divergence or slow convergence:

$$\sum_{i=1}^{n}\left|x_i\left(t_0+\delta t\right)-x_{n_i}\left(t_0+\delta t\right)\right| \ge \beta_2 \sum_{i=1}^{n}\left|x_i\left(t_0\right)-x_{n_i}\left(t_0\right)\right| \qquad 0 \le \beta_2 < 1 \tag{2}$$

β_2 is the *slow convergence/divergence* coefficient and $\beta_2 > \beta_1$.

Medium convergence:

$$\begin{aligned} &\beta_1 \sum_{i=1}^{n}\left|x_i\left(t_0\right)-x_{n_i}\left(t_0\right)\right| \le \\ &\sum_{i=1}^{n}\left|x_i\left(t_0+\delta t\right)-x_{n_i}\left(t_0+\delta t\right)\right| \le \\ &\beta_2 \sum_{i=1}^{n}\left|x_i\left(t_0\right)-x_{n_i}\left(t_0\right)\right| \end{aligned} \tag{3}$$

4-Is it time to hold an election?

To hold an election, the current time should be smaller than t^* (this parameter will be defined in section 4).

5-If it is allowed, the election is held. The results of the previous election are used to determine the results of the current election (*assumption* 1). The procedure is as follows:

If the regime of states is *"fast convergence"*; the desired trajectories become the origin or $\mathbf{r}_d = 0$ (the public votes for conservative policies). If the regime of states is *"slow convergence/divergence"*; the opposition party is elected. If the regime of

states is *"medium convergence"*; the previous reference trajectories are selected again (the public votes for the current policies).

To make the election process clearer, an example is presented:

Example:

ω_1, ω_2 and ω_3 are the spacecraft angular velocities (refer to section 3). Assume that ω_1 is the least qualified state.

Note 1: A state will be the least qualified if it deviates more than the other states from the fault-free system. Obviously, for under-actuated systems, the un-actuated state will be the least qualified.

In order to stabilize the system, the public votes for the following reforms:

If the policy of ω_2 is RR, the party of $\omega_{d,2}$ changes to CR (more details will be presented in the following)

If the policy of ω_2 is CR, the party of $\omega_{d,2}$ changes to CL

If the policy of ω_2 is CL, the party of $\omega_{d,2}$ changes to CR

If the policy of ω_2 is RL, the party of $\omega_{d,2}$ changes to CL

Note 2: If the reforms are carried through rapidly, false conclusions will be made about the performance of the ruling party. Therefore, it is assumed that when the policy of a state is radical, it first switches to conservative and then the opposition party is elected. This is equivalent to *gradual reforms* in the society.

Note 3: ω_2 is selected to carry through the reform and ω_3 is used as an assistant for ω_2 to perform its policy making. If ω_3 does not help ω_2, ω_2 will be forced to act alone and therefore, the performance will not be acceptable. The role given to the assistant state (ω_3 in this case), has its roots in the spacecraft dynamics (eq. (6)).

According to *note* 3, the following policy makings are also performed by ω_3:

If the policy of ω_3 is RR, it remains there (more details will be given in the following paragraphs)

If the policy of ω_3 is CR, the party of $\omega_{d,3}$ changes to RR

If the policy of ω_3 is CL, the party of $\omega_{d,3}$ changes to RL

If the policy of ω_3 is RL, it remains there

In order to quantify the outcome of elections, the following procedures will be considered:

If the next policy is CL/CR (CL or CR), $\mathbf{r}_d$ will be the average of zero (origin) and CLUB/CRUB and if the next policy is RL/RR, $\mathbf{r}_d$ will be RLLB/RRLB. Therefore, $\mathbf{r}_d$ will be constant in δt intervals (refer to the simulation section).

Since the upper and lower bounds are selected as the election parameters (section 4), the whole political spectrum will be covered, and no *political censorship* will occur.

6-Faulty[1] and nominal (fault-free) systems are simulated for $t_0 \le t \le t_0 + \delta t$.

7-t_0 is updated to $t_0 + \delta t$.

8-A condition to check whether $t_0 > t_f$, where t_0 and t_f are the current and final times, respectively. If this condition is not satisfied, the inner loop continues from the beginning.

9-The final state constraint (eq. (16)) is checked. If this condition is not satisfied, the outer loop continues from the beginning.

Modeling the election process has now been completed. Spacecraft dynamics and controller structure are the subjects of the next section.

3 Spacecraft Dynamics and Control

3.1 Spacecraft Dynamic Equations

The rigid body spacecraft rotational dynamics in the principal coordinate system is described by the following equations [9]:

$$\begin{aligned}
&\dot{\omega}_1 = \alpha_1 \omega_2 \omega_3 + u_1' \quad \alpha_1 = \left(\frac{J_2 - J_3}{J_1} \right) \\
&\dot{\omega}_2 = \alpha_2 \omega_1 \omega_3 + u_2' \quad \alpha_2 = \left(\frac{J_3 - J_1}{J_2} \right) \\
&\dot{\omega}_3 = \alpha_3 \omega_1 \omega_2 + u_3' \quad \alpha_3 = \left(\frac{J_1 - J_2}{J_3} \right)
\end{aligned} \tag{4}$$

[1] - It is assumed that the FDD block provides the post-fault information

$(\omega_1, \omega_2, \omega_3)$ are the angular velocities, (u_1', u_2', u_3') are the normalized control inputs and finally, (J_1, J_2, J_3) are the principal moments of inertia of the rigid body along the principal body axis. The relation between control torques and inputs are given by the following equations:

$$\begin{aligned} u_1' &= u_1 / J_1 \\ u_2' &= u_2 / J_2 \\ u_3' &= u_3 / J_3 \end{aligned} \tag{5}$$

(u_1, u_2, u_3) are the three control moments acting on the spacecraft. The upper and lower bounds of the control inputs are restricted according to the following saturation function:

$$\text{sat}(u_i) = \begin{cases} u_i & \text{if} \quad -u_{\max} \le u_i \le u_{\max} \\ u_{\max} & \text{if} \quad u_i > u_{\max} \\ -u_{\max} & \text{if} \quad u_i < -u_{\max} \end{cases} \tag{6}$$

$u_{\max}$ is the maximum torque that can be produced by the actuators.

3.2 Controller Structure

The error signal is defined as follows:

$$\boldsymbol{\omega}_e = \boldsymbol{\omega} - \boldsymbol{\omega}_d \tag{7}$$

$\boldsymbol{\omega}_d$ and $\boldsymbol{\omega}_e$ are the desired and error angular velocity vectors, respectively.

Rewriting the spacecraft dynamics in the form of error dynamics will result in the following set of equations:

$$\begin{aligned} \dot{\omega}_{1e} &= \dot{\omega}_1 - \dot{\omega}_{1d} = \alpha_1 (\omega_{2e} + \omega_{2d})(\omega_{3e} + \omega_{3d}) + u_1' - \dot{\omega}_{1d} = u_1'' \\ \dot{\omega}_{2e} &= \dot{\omega}_2 - \dot{\omega}_{2d} = \alpha_2 (\omega_{1e} + \omega_{1d})(\omega_{3e} + \omega_{3d}) + u_2' - \dot{\omega}_{2d} = u_2'' \\ \dot{\omega}_{3e} &= \dot{\omega}_3 - \dot{\omega}_{3d} = \alpha_3 (\omega_{1e} + \omega_{1d})(\omega_{2e} + \omega_{2d}) + u_3' - \dot{\omega}_{3d} = u_3'' \end{aligned} \tag{8}$$

The nonlinear terms are canceled using feedback linearization. Consequently, the closed-loop system will be transformed into the following simple linear time invariant form:

$$\begin{aligned} \dot{\omega}_{1e} &= u_1'' \\ \dot{\omega}_{2e} &= u_2'' \\ \dot{\omega}_{3e} &= u_3'' \end{aligned} \tag{9}$$

and the following form of feedbacks will lead to the exponential stabilization of $\boldsymbol{\omega}_e$ to zero:

$$\begin{aligned} u_1'' &= -k_1(\omega_{1e}) \quad k_1 \in R^+ \\ u_2'' &= -k_2(\omega_{2e}) \quad k_2 \in R^+ \\ u_3'' &= -k_3(\omega_{3e}) \quad k_3 \in R^+ \end{aligned} \tag{10}$$

This will result in the exponential convergence of $\boldsymbol{\omega}$ to $\boldsymbol{\omega}_d$.

Considering the set of eq. (8) and eq. (10), (u_1', u_2', u_3') will be obtained as follows:

$$\begin{aligned} u_1' &= -\alpha_1(\omega_{2e}+\omega_{2d})(\omega_{3e}+\omega_{3d}) + \dot{\omega}_{1d} - k_1\omega_{1e} \\ u_2' &= -\alpha_2(\omega_{1e}+\omega_{1d})(\omega_{3e}+\omega_{3d}) + \dot{\omega}_{2d} - k_2\omega_{2e} \\ u_3' &= -\alpha_3(\omega_{1e}+\omega_{1d})(\omega_{2e}+\omega_{2d}) + \dot{\omega}_{3d} - k_3\omega_{3e} \end{aligned} \tag{11}$$

For feedback purposes, it is more suitable to rewrite (u_1', u_2', u_3') in terms of the original variables:

$$\begin{aligned} u_1' &= -\alpha_1(\omega_2)(\omega_3) + \dot{\omega}_{1d} - k_1(\omega_1 - \omega_{1d}) \\ u_2' &= -\alpha_2(\omega_1)(\omega_3) + \dot{\omega}_{2d} - k_2(\omega_2 - \omega_{2d}) \\ u_3' &= -\alpha_3(\omega_1)(\omega_2) + \dot{\omega}_{3d} - k_3(\omega_3 - \omega_{3d}) \end{aligned} \tag{12}$$

These are the desired control inputs that will lead to the exponential convergence of $\boldsymbol{\omega}$ to $\boldsymbol{\omega}_d$.

Now imagine $\boldsymbol{\omega}_d = 0$. The equations of the closed-loop system will be:

$$\begin{aligned} \dot{\omega}_1 &= -k_1(\omega_1) \\ \dot{\omega}_2 &= -k_2(\omega_2) \\ \dot{\omega}_3 &= -k_3(\omega_3) \end{aligned} \tag{13}$$

Clearly, as long as there is no saturation and the actuators can produce the required control inputs, the closed-loop system remains globally exponentially stable (GES). However, after the occurrence of actuator failures, GES will not be guaranteed. As will be seen in the simulation section, the proposed method will be able to steer the faulty system towards the origin. This happens even when severe actuator failures occur, and the system becomes under-actuated.

The next section will provide a stability analysis to prove that under a certain condition, the proposed method can make the origin an asymptotically stable equilibrium point for the faulty system.

4 Stability Analysis

As shown in Fig. 1, the RTM block receives the post-fault model of the system and produces the desired reference trajectory vector ($\mathbf{r}_d$). The qualitative and quantitative procedures outlined in Section 2 are the main structures of the RTM block. However, in order to tune the *election parameters*, the following problem should be solved:

Determine the election parameters ($\beta_1, \beta_2,$ δt and political spectrum) such that the final state vector becomes zero, i.e. $\boldsymbol{\omega}(t_f) = 0$. Such a final state constraint is well-known in the literature and is introduced to ensure asymptotic stability [3].

Note 4: In order to make sure $\boldsymbol{\omega}_d$ approaches the origin before $t = t_f$, its value is set to zero as *t* passes t^*.

In other words:

$$\boldsymbol{\omega}_d = 0 \ \forall \ t \geq t^* \tag{14}$$

In order to give the solver more flexibility to solve the problem, another variable (k_s) is introduced:

$$t^* = k_s t_f \quad k_s \in (0.5 \ 1) \tag{15}$$

The role of k_s is to determine t^* as a function of the final time. This will give the solver more flexibility to solve the problem.

Remark 1: The proposed method includes some issues related to the convergence (eq. (1), eq. (2) and eq. (3)). Therefore, as will be seen in the simulation section, the final state constraint can be easily satisfied through a few simulations. Consequently, the convergence speed will be high (a very important feature of any AFTC design).

5 Simulation

The system/controller parameters and initial conditions are given in Table 1:

Table 1
System/Controller parameters and initial conditions

Controller parameters		Initial conditions	(degree/sec)	Moments of Inertia	($kg.m^2$)
k_1	0.1	$\omega_1(0)$	10	J_1	449.5
k_2	0.1	$\omega_2(0)$	10	J_2	264.6
k_3	0.1	$\omega_3(0)$	-15	J_3	312.5

It is assumed that the fault occurs at $t_{fault} = 10$ second and the final time is $t_f = 200$ second.

Table 2 presents the range of the election parameters:

Table 2
Range of the election parameters

Parameter	Range
β_1	$[0.1\ 0.6]$
β_2	$[0.6\ 1]$
δt	$[10\ 50]$ second
RLLB	$[1\ 20]$ degree/ second
RRLB	$[-20\ \ -1]$ degree/ second
k_s	$[0.5\ 1]$

In order to satisfy the final state constraint $\left(\boldsymbol{\omega}(t_f) = 0\right)$, eq. (16) is defined:

$$\sum_{i=1}^{3} \omega_i^2(t_f) \le 0.01(\text{deg/ sec})^2 \tag{16}$$

A combination of global (Genetic Algorithm[1] [6]) and local (Sequential Quadratic Programming[2] [5]) optimization is used to satisfy eq. (16). First, GA explores the search space to find the promising region and then SQP exploits the region to satisfy eq. (16). The population size of GA is selected as 20. The other parameters of GA and SQP are the default values considered in MATLAB 2011a. The stopping criteria for GA and SQP are illustrated in Table 3:

[1] -GA (ga command)

[2] -SQP (fmincon command)

Table 3
Stopping criteria for GA and SQP

GA	500 seconds elapsed time
SQP	Satisfying eq. (16)

The actuation system consists of six thrusters (without considering hardware redundancy), that are placed in opposite directions and each thruster is capable of producing a maximum of 50 N variable thrust. The effective moment arm of all thrusters is one meter along the principal body axis. However, the configuration of the thrusters is such that (T_1-T_2), (T_3-T_4) and (T_5-T_6) produce net moments about the first, second and third principal axes, respectively (fig. 4). (Direction of the arrows = Direction of the forces).

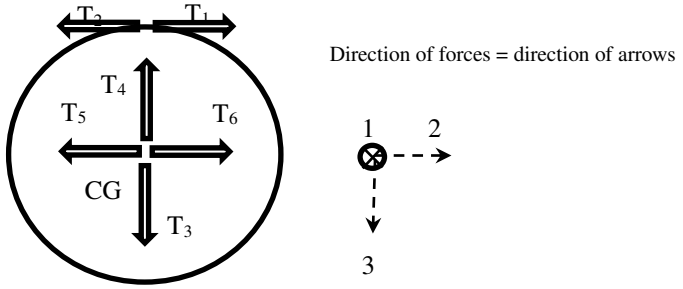


Figure 4
Thruster configuration

It seems that the thrusters T_3, T_4, T_5 and T_6 pass through CG. However as stated in the previous paragraph, they have a moment arm of one meter along the first body axis.

The considered failure scenario is presented in Table 4:

Table 4
Failure scenario

Failure scenario	Failure of T_1 and T_2 (first body axis is under-actuated)

This failure scenario makes the spacecraft under-actuated, a challenging issue for control purposes.

5.1 Failure Scenario

5.1.1 Without RTM

Fig. 5a and Fig. 5b illustrate the nominal (fault-free) and faulty (without RTM) systems response and control input:

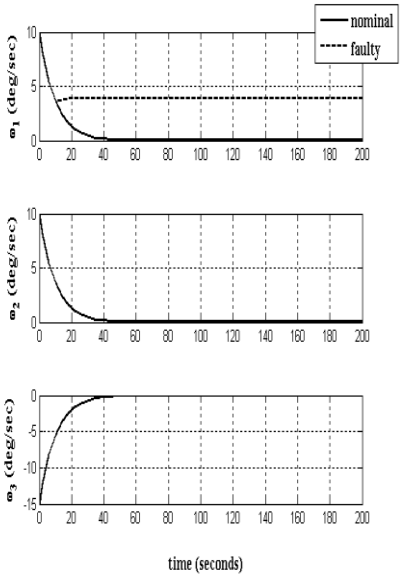


Figure 5a
Nominal and faulty systems response-
first fault scenario (without RTM)

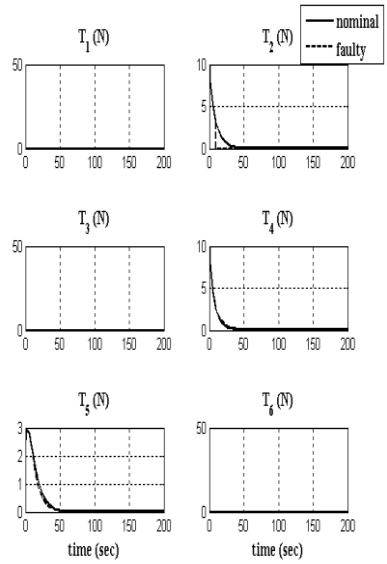


Figure 5b
Nominal and faulty systems control input-
first fault scenario (without RTM)

It is observed that if the faulty system is not recovered, ω_1 will not converge to the origin (a previously predicted result).

5.1.2 With RTM

The corrected faulty system response and control input are illustrated in Fig. 6a and Fig. 6b[1]:

[1] -Intel(R) Core™2 CPU, T7200@2.00GHz, MATLAB® 2011a

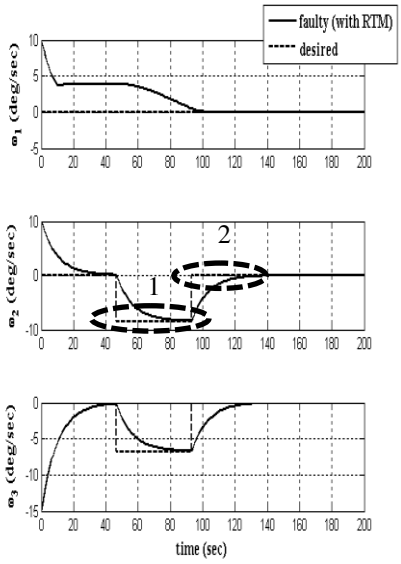


Figure 6a
Faulty system response (with RTM)-first fault scenario (elapsed time=671sec)

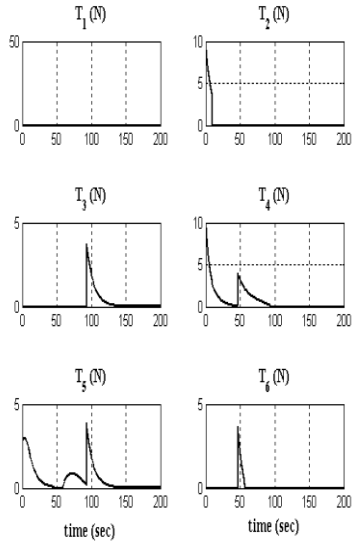


Figure 6b
Faulty system control input (with RTM)-first fault scenario

Remark 2: "desired" signal shown by the dashed line is the modified $\boldsymbol{\omega}_d$, assumed to be constant in δt intervals.

A comparison of Fig. 5a and Fig. 6a shows the ability of the proposed method to steer the post-fault system to the origin. Table 5 presents the obtained results for the election parameters:

Table 5
Election parameters obtained for the first fault scenario

β_1	0.53
β_2	0.61
δt	46.53
RLLB for ω_1	16.43 degree /sec
RRLB for ω_1	-8.11 degree /sec
RLLB for ω_2	14.42 degree /sec
RRLB for ω_2	-16.97 degree /sec
RLLB for ω_3	11.29 degree /sec
RRLB for ω_3	-6.76 degree /sec
k_s	0.63

In order to explain the election process, two dashed ellipses are shown in Fig. 6a. Circle 1 demonstrates a reform. As mentioned in the text, the reason for such an election is the slow convergence/divergence of states. Circle 2 shows the election of conservative policies. This election shows that the public is satisfied with the fast rate of convergence.

Note 5: According to the simulation results and the elapsed time, the proposed method is real-time (refer to http://stackoverflow.com/questions/20513071/ for more information).

Remark 3: One of the limitations of the proposed RTM is that the desired reference trajectories are considered to be constant in δt intervals. Future work will concentrate on more flexible trajectory generation tools, like cubic splines. This idea will make the proposed RTM more applicable to a wider class of systems.

One of the main contributions of this paper is to use political theories to solve control problems. To the authors' best knowledge, no previous work has considered such methodology. The results show that the proposed method has a great potential for further developments.

Note 6: The present paper shows that socio-political models have a great potential to solve complex engineering problems. On the other hand, implementation of such ideas in engineering applications can be used as a method for their evaluation. Such a bilateral relationship will be helpful for both socio-political and engineering related problems.

Conclusions

Based on the model of elections in two-party democratic societies, a heuristic RTM for AFTC was proposed. The proposed method was implemented on a spacecraft, and it was observed that the faulty system was able to reach the origin, an event that would not occur if no actions were taken. The present paper shows the ability of socio-political models to solve complex engineering problems and opens a new window for further developments in this field.

References

[1] Aloisi, S: Election pushes Italy towards two-party system, Reuters, http://www.reuters.com/article/us-italy-election-system-dUSL1580537620080415, accessed 1 November 2016

[2] Nemes, A: Continuous Periodic Fuzzy-Logic Systems and Smooth Trajectory Planning for Multi-Rotor Dynamic Modeling, *Acta Polytechnica Hungarica*, Vol. 13, No. 6 (2016)

[3] Fontes, FACC: A general framework to design stabilizing nonlinear model predictive controllers, *Systems and Control Letters*, doi:10.1016/S0167-6911(00)00084-0 (2001)

[4] Garone, E., Cairano, S. Di and Kolmanovsky, I. V: Reference and command governors for systems with constraints: A survey on theory and applications, *Mitsubishi Electric Research Laboratories*, http://www.merl.com (2016)

[5] Gill, PE., Murray, W and Wright, MH: Practical Optimization, *Emerald Group Publishing Limited*, ISBN-13: 978-0122839528 (1982)

[6] Goldberg, DE: Genetic Algorithms in Search, Optimization & Machine Learning, *Addison-Wesley Professional*, 1 edition, ISBN-13: 978-0201157673 (1989)

[7] McLean, I and McMillan, A: Conservatism: Concise Oxford Dictionary of Politics, *Oxford University Press*, Third Edition, ISBN: 978-0-19-920516-5 (2009)

[8] Midlarsky, MI: Political stability of two-party and multi-party systems: probabilistic bases for the comparison of party systems, *The American Political Science Review*, doi: 10.2307/1955799 (1984)

[9] Sidi, MJ: Spacecraft Dynamics and Control: A Practical Engineering Approach, *Cambridge University Press*, Revised edition, ISBN-13: 978-0521787802 (2000)

[10] Ware, A: Political Parties and Party Systems, *Oxford University Press*, ISBN: 9780198780779 (1995)

[11] Yin, S., Xiao, B., Ding, S and Zhou, D: A review on recent development of spacecraft attitude fault tolerant control system, *IEEE Transactions on Industrial Electronics*, 63(5): 3311-3320, doi: 10.1109/TIE.2016.2530789 (2016)

[12] Zhang, Y and Jiang, J: Bibliographical review on reconfigurable fault-tolerant control, *Annual Reviews in Control*, doi:10.1016/j.arcontrol.2008.03.008 (2008)

Dependable Peer-to-Peer SCADA Architecture

Mihály Sági

University of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia, sagi@uns.ac.rs

Ervin Varga

University of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia, evarga@uns.ac.rs, e.varga@ieee.org

Abstract: SCADA solutions are under a high flux to shift their focus from process control of a limited set of industrial plants to the control of large-scale system of systems. This is in par with the recent proliferation of ubiquitous/pervasive computing paradigm mostly embodied as Internet of Things (IoT). In a traditional setup, a whole system is only partially covered by SCADA data points; therefore, complex simulation is required to fit the missing measurements, hence, buttress decision support scenarios. This usually entails a fully integrated and centralized approach, where SCADA infrastructure needs to hold and distribute data, both collected and calculated. It leads to the increase of load on a supporting real-time database, which hosts millions of data points. It is a challenge that a traditional SCADA design (based on a shared memory database and competing processes), cannot fulfill in real-time. This paper proposes an alternative approach of an architecture and basic functionality of a SCADA system. The proposed architecture targets distributed SCADA systems that can be used to supervise and control large-scale distributed industrial or infrastructural systems. Strategic data organization and segmentation are introduced, so that the acquired data can be efficiently distributed throughout the system. The proposed architecture pushes forward a peer-to-peer node structuring scheme, where an autonomous node supervises and controls only subsets of the system. Nodes collaborate to establish a unified view of the entire system. The proof of the concept implementation has proven to be able to manage significantly more data points in a distributed fashion than a centralized variant.

Keywords: SCADA; smart city; distributed system; peer-to-peer architecture; smart grid

1 Problem Statements and Objectives

As industrial processes became more complex and computing power became cheaper and more robust, these processes started to be supervised and controlled by programmable logic controllers (PLC) to induce more precision and reliability into the process itself. With the growth of the industry and the complexity within the industrial processes, the need for a larger scale process supervision and control was needed, so SCADA systems were developed as a universal mean of access local control modules such as PLCs. They soon became the most commonly used industrial control systems. After the "automation revolution" in industrial systems, SCADAs started being applied to infrastructural management systems as well (e.g. electricity, gas, water, waste-water). Since infrastructure systems are of a more complex nature than industrial systems, the move to this field introduced new challenges into SCADA development. This paper will give a brief historic overview of currently available solutions and will propose a solution for the newly introduced challenges.

1.1 Current Work

SCADA systems were initially used in industrial processes that were located in a single processing plant with well defined geographical boundaries. The geographical and industrial process based limitations had many conveniences such as limited number of sensors and actuators, no or slow expension of number of sensors (telemetered data) over time. These systems usually had only a few operator terminals that showed the overall state of the plant to the operators. Decision support systems in manufacturing process were rare. [1] [2]

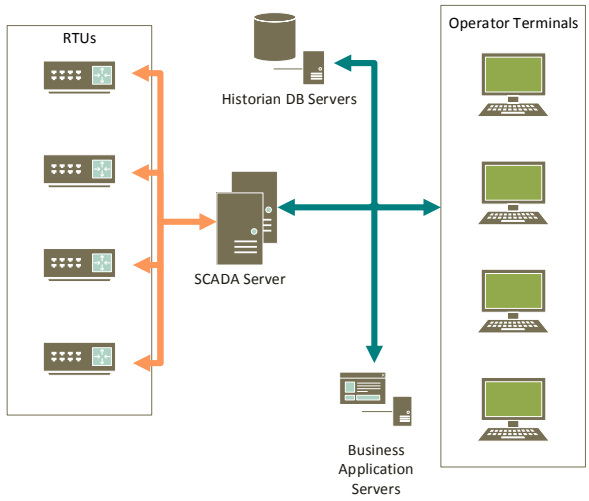


Figure 1. Building Blocks of a Traditional Clustered SCADA System

One of the most popular SCADA designs was building multiple processes that execute segments of the overall SCADA functionality (data processing, communication, supervisory control, automated control procedures, historization, decision support, etc.) (Figure 1). Such designs provided easy integration with the SCADA real-time database and simple extension of basic SCADA functionalities driven requirements specific for the controlled industrial process. Traces of such an architecture are still found in some of the high-end SCADA solutions available on the market.

Current developments in supervised systems generally go towards further merging and increase in size, which has to be followed by the provision of adequate control systems, both in size and acceptable performance. In addition, control requirements are far more demanding than before, in order to run technological processes in more precise and safer manner.

This is particularly obvious in infrastructural SCADA applications (from power distribution, through oil, gas and water transport and distribution systems to even smart cities), that requires integrated control system with acquisition of process data that is placed to the centralized real-time database with several million data points. [3]

In these systems, the decision support is integrated with the SCADA so the real-time database stores telemetered SCADA data together with derived values (e.g. calculations, estimations, predictions, simulations, etc.). The SCADA is expected to feed data to decision support components, treat the collected derived data in the same manner as the telemetered and to distribute all data to client (operator or business) applications. In addition, the scale of such systems introduces a need for multiple client application. Even though contemporary computers scaled through time, the bottleneck when applying traditional SCADA architectures to industrial systems is the communication infrastructures, that is far more complex and expensive to change or upgrade. Additionally, infrastructural systems usually are geographically scattered, they have a large number of sensing and actuating points (up to several million) and the configuration is changing (mostly growing) over time. Such systems also rely on complex decision support and tens of operator and maintenance terminals. [4]

The challenges coming from infrastructural systems were the main driving force of the design of a new SCADA architecture. However, with the advances in both hardware and software over time, there are additional requirements important for its applicability of contemporary SCADA systems: extensive and elaborate SCADA model, reusability, extensiblity, easy adaptation to user demands, cross-platform and secured operation. [5] [6] [7]

Contemporary SCADA systems on the modern market need to support the handling of millions of data points (part telemetered, part derived) in a networked, real-time environment. Data acquisition and primary data processing is always done in a centralized fashion due to architectural limitations, and then the

concentrated values in the real-time database counting several million data points are distributed to achieve high-availability. The complex operations in the decision support systems are also performed where the data acquisition is taking place. An example is, an infrastructural system where SCADAs are being introduced are Distribution Management System implementations for power distributions, where it is a usual market requirement that the SCADA handles more than 10 million data points in real-time. To achieve high availability, multiple copies of the SCADA software is running and data replication mechanism is used. Due to the size of the database and the need that the data needs to be transferred as quickly as possible to the backup SCADA system, the communication subsystem must be optimized to the highest level. [7] [16]

Other contemporary approaches include WEB based SCADA systems. [8] Such systems leverage the network infrastructure already available and while they are easier to implement than "traditional" SCADA systems, the required level of security and real-time operation capability are not achievable for big systems under supervision. [9] [10]

1.2 Proposal

Core requirements for a SCADA system (regardless if industrial or infrastructure) are real-time operation, reliability and high availability. The minimum response time is defined by the supervised system and is measured in seconds. Other requirements that are mandatory for SCADA systems are reliability availability as for most of the real-time systems.

The architecture presented in this paper relies on a distributed storage system that provides data distribution with configurable bandwidth and performance ratio where the communication between SCADA nodes can be optimized per needs. [14] The data distribution is built into the real-time database enabling the distribution of telemetry and decision support subroutines.

The paper describes the architecture and basic functionality of a SCADA system developed in line with this concept, providing efficient real-time execution of complex supervisory and control procedures in a distributed environment.

The proposed architecture can be classified as a third generation SCADA architecture. The SCADA core is based on an earlier study [11], where a monolithic architecture is used. Basic principles are inherited and the architecture is upgraded to adopt the "modern", networked nature.

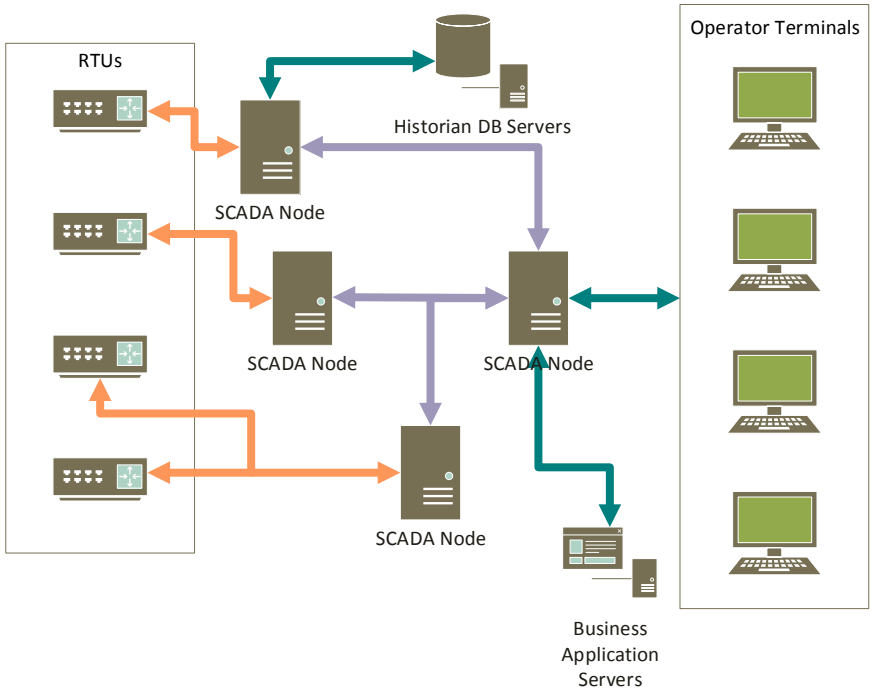


Figure 2
Proposed Peer-to-Peer Structured SCADA System

Since the proposed architecture is considered to be of a generic purpose, so it can drive all infrastructural decision support systems found in a smart city, the most complex and response critical infrastructure is taken as a target. The new architecture must be fit to feed a power system. Based on the research of Kang L. and Yang L. [4] the acquisition and control tasks are being parallelized throughout homogenous SCADA nodes.

Dietrich B. et al. [12] confirmed the usability of the object-oriented paradigm as a tool to reach full extensibility and customizability even in real-time application, thus it is adopted as the core concept of the new SCADA.

Ramdan F [8] and Qiang Z [13] show, that multi tier SCADA systems can be designed and preserve the full functionality of conventional SCADAs.

2 System Architecture

Taken into consideration the distributed nature of most common infrastructural management systems, a target SCADA should be distributed with homogenous

nodes that can execute field communication and that can share data with all other "interested" nodes.

Each node is responsible for handling operations regarding the subset of telemetry equipment it is directly connected to through a computer network (e.g. ethernet)All other operations are ignored by the operation logic of the note, but is stored and distributed for redundancy purposes.

Client terminals are also connected to nodes. Each client receives data and is able to execute commands for its area of responsibility (AOR) that is configurable Commands and telemetry data that is outside of the nodes AOR is stored for redundancy purposes.

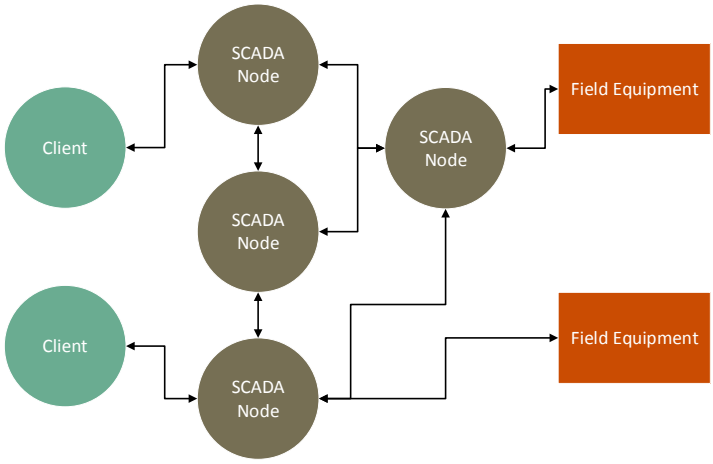


Figure 3
Simplifyed System Architecture

With such an architecture, the traditional monolithic SCADA is broken up into SCADA nodes. Since clients are not aware of the communication between these nodes, they are not aware of infrastructure or hierarchy change within thin the SCADA itself (Figure 3).

Data processing and conversion from the inner model to an industry specific model is also done in a distributed fashion. Once the data enters the system from the field equipment, it is immediately processed and then distributed. To allow the distribution of this task once the configuration changes and new field equipment is added to the system, the node that is least loaded will connect to the new equipment. On a node failure, all connected external devices are re-routed to other, healthy nodes. Load for each of the nodes is calculated based on the number

of expected value changes in one second incoming from the connected field equipment.

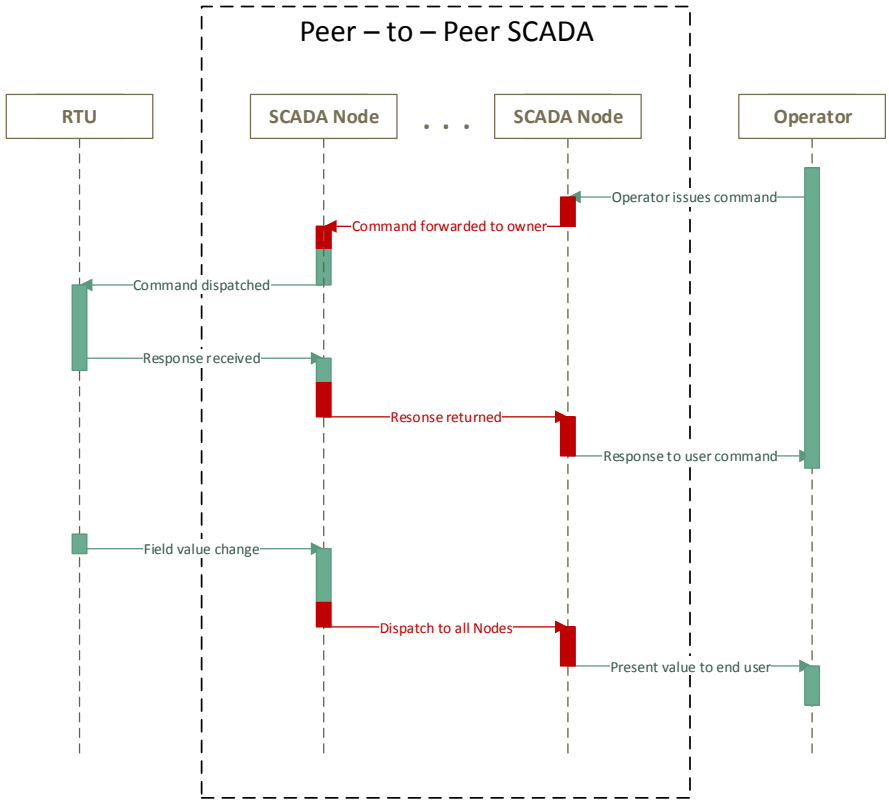


Figure 4
Sequence diagram of basic SCADA operations

The figure above (Figure 4) shows the sequence of major SCADA use cases: supervisory control and data acquisition. The sequence for both operations is decomposed on the domain specific part (colored green) that needs to be done in every SCADA system, no matter what architecture it has. These operations are: primary (front-end) and secondary (back-end) processing and data, and input validations. The architecture specific part of the sequence diagram (colored red) represents the additional steps (time) that is the overhead the distributed architecture includes.

During a configuration change it is of high importance that the entire cluster works on the same configuration. Configuration changes are distributed as

changes in a distributed version control system ensuring the fulfillment of the above-mentioned requirement. [15]

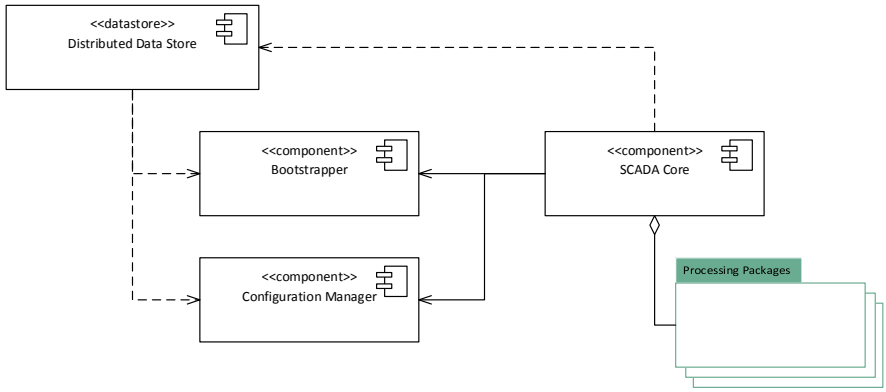


Figure 5
Components of a SCADA Node

A key required feature for the distributed data stores analyzed for this paper was an embedded VCS (*Versioning Control System*) support that allows versioning of the system configuration. The *Configuration Manager* component controls the configuration version using the VCS interface of the DDS.

The VCS that is embedded into the distributed data storage that the SCADA nodes are built on top of, enables that the configuration can be changed from any of the nodes the configuration manager client software connects to, as long as the operator using the software has sufficient privileges for the AOR the configuration change is intended for. Once the operator changes the configuration on one node, the changes are propagated through the DDS and trigger the configuration manager component to apply the changes to the SCADA core.

The internal SCADA model is designed to support one real-time, and multiple history/simulation data contexts and are accessible to all authorized users. Each context supports loading of different configuration version, current or any valid previous configurations.

Each node in the system, in spite of different roles it may play, always executes a single software executable, which holds the real-time database and performs the data acquisition, control and communication tasks. Since decision support routines need to be handled in the same (real-time) manner as telemetry, there is no need to differentiate them. A node is based on a common software infrastructure implemented within a core library, and a set of additional dynamic libraries implementing process variables, various protocols, etc. that are distributed with

the configuration (Figure 5). Each library implements or extends an entity (class) using the predefined API. Using OOP paradigm, it is easy to extend or replace any of the classes, or add completely new.

Addressing entities within the system uses a simple, but effective schema where four logical sections of one unique global key are assigned to each entity: context, node and variable names is unique, providing a primary key to each for every configurable component in the system:

- *Context* – unique identifier of the context where the entity is available. Default 0 is the identifyer of the real-time context. When a real-time entity is copied to a simulation context only this portion of the primary key is changed.
- *Node* – unique identifier of the node owning the entity. Only nodes with this ID are making true changes in the entity (from the field or decision support routines). The rest of the nodes get these entities using data distribution.
- *Type*– entity type set from all the available types in the system configuration. The system has a pre-defined set of entity types, but can be extended to achieve customizability.
- *Sequence* – a simple ordinal number within the owning node.

To run the solution only two libraries are needed besides the executable – *Bootstrapper* and *Core*. With highly modular architecture, the rest of the libraries providing support for various industrial protocols, decision support or customizability are not the crucial. Thus, the system can function without them in a "store-and-forward" mode when only stores partial configuration and real-time data for redundancy purposes. [14]

Bootstrapper is the coordinator of the local node (initialization / de-initialization, configuration loading / unloading, starting / stopping routines, etc.). It is composed of two essential components:

- *Communication manager* – interface to the SCADA administrator to reach the configuration manager to take actions on a node.
- *Configuration manager* – runs both the initialization and de-initialization process of the local SCADA node core on configuration version change.

The core of the system is a communication engine that does the telemetry data acquisition and command dispatching based on the node configuration. All data reaching a node is treated the same way regardless if it is acquired directly from an RTU or through standardized industrial protocol or is distributed by the DDS itself. It also provides a means for local and remote user interfaces for system supervision and control.

4 Proof of the Concept Implementation

Key difference in the currently described architecture, compared to the traditional solutions, is the introduction of a high-level program model of the physical process based on catalogues and process variables (Figure 6). [16]

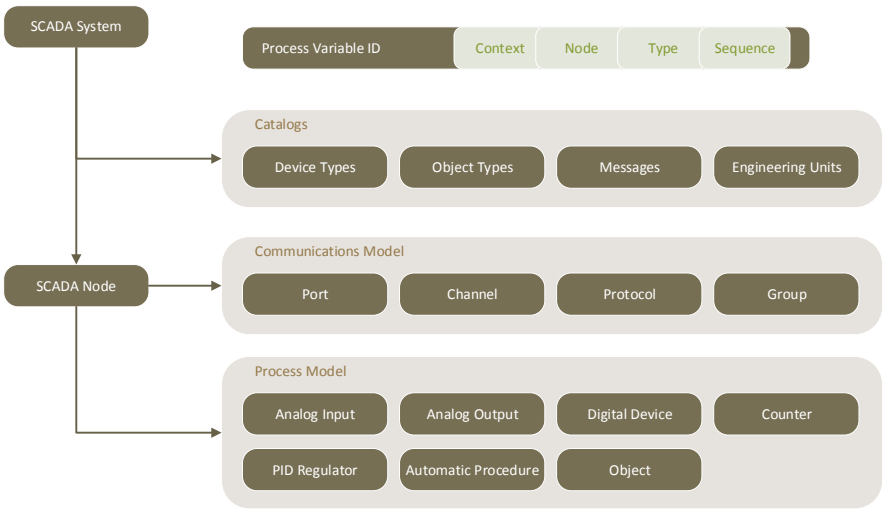


Figure 6
Process model and real-time database organization

Catalogs define set of application-specific values and data-types, referenced later by configured entities. As an example, digital devices catalog defines types of used control modules (e.g. on-off valve with two limit switches), including legal set of states and commands associated to the module.

Process variables as described in the process model section on Figure 6, are divided into levels by their complexity, represent various measuring and actuating equipment and attributes in the physical process itself. The current state or value of the process variable is always expressed in its engineering unit (EU) value, or through the associated state/command pair if the variable is digital in its nature. There are conversion capabilities provided so that external values can be converted to the internal data types.

To achieve extensibility and customizability, each type of process variables is, following OOP paradigm, providing a convenient way to extend or modify original definition or functionality according to the application-specific requirements.

4.1 Tools Used to Build the Solution

Real-time software is often built in C++ to leverage the performance this programming language provides and to us the object-oriented paradigm for easier implementation of the high level design. As one of the requirements set for the solution is cross-platform operation, the aforementioned programming language is also fits that need.

For the distributed storage system, two solutions were analyzed, the IBM Cloud Object Storage and IPFS [15]. After a detailed analysis of both solutions, IPFS was chosen as a platform to build the SCADA system on top of. [17]

To unify the C++ libraries that Visual Studio C++ compiler and G++ provide, Boost library is used.

Conclusions and Future Work

This paper analyzes the bottlenecks of centralized SCADA architectures when used on large scale CIS. In addition, it presents a possible architecture of a distributed industrial grade SCADA system that fulfills the demands of a smart city and/or smart grid, such as distribution of the data, processing and control, system scalability, real time efficiency and cross-platform operation.

The proposed architecture enables homogenization of the SCADA nodes that can be arbitrarily extended and configured due to the high modularity of the new concept. With these characteristics, both SCADA operators and business information system applications are enabled to connect to any node and acquire the data they demand, or control the process supervised by the node.

This architecture also provides the capability of having an arbitrary number of nodes in a system to share processing and storage of data. With a distributed data store as the basis of a SCADA system, requirements for high availability can be reached without additional mechanisms (e.g. replication). Both configuration and real-time data is shared across the system as soon as changes occur.

Configuration versioning as an integral part of the majority of distributed data stores, adds value to this solution since without a need of additional applications or components, version control of the configuration is achieved.

Further research can be done on an efficient way of managing the configuration of a distributed SCADA system using the proposed architecture and also measuring the performance and scalability of the solution as well as the minimum system requirements.

Some of the additional topics that need to be answered before reaching a commercially viable architecture are: separation of configuration and real-time data, determining the impact of persisting all changes in a normal work of the system.

As presented on Figure 3 and mentioned in Section 5, the current work relies on 3rd party implementation of distributed data storage system with proven history. This DDS is not optimized for real-time operation and some SCADA specific needs, once a stable system is built on top of this library, additional improvements will be needed on the DDS itself to minimize the overhead that the proposed architecture introduces to the SCADA system.

References

[1] D. Bailey and E. Wright, "Practical SCADA for Industry", Elsevier, 2003

[2] S. Lishev, R. Popov and A. Georgiev, "Laboratory SCADA Systems – the State of Art and the Challenges," Balkan Journal of Electrical & Computer Engineering, Vol. 3, No. 3, p. 164, 2015

[3] J. M. Black, C. Rawie and M. Mattson, "High-Performance SCADA System Provides an Integrated Information System," in Water Environment Federation, Anaheim, California, USA, 2000

[4] L. Kang and L. Yang, "A Distributed and Parallell Computing Framwork for SCADA Application in Power System," in International Conference on Electrical and Control Engineering, Tuxtla Gutierrez, Mexico, 2010

[5] O. Rysavy, J. Rab, P. Halfar and M. Sveda, "A Formal Authorization Framework for Networked SCADA Systems," in 19th IEEE International Conference and Workshops on Engineering of Computer-Based Systems, Novi Sad, Serbia, 2012

[6] l. Yang, X. Geng and X. Cao, "A Supervisory Control and Data Acquisition Network Security Attack Recognition Method Based on Multi-Agent," Journal of Computational and Theoretical Nanoscience, Vol. 13, No. 4, pp. 2504-2511, 2016

[7] S. Ju, J. Lee, J. Park and J. Lee, "Secure Concept of SCADA Communication for Offshore Wind Energy," in Advances in Parallel and Distributed Computing and Ubiquitous Services, Springer, 2016, pp. 91-97

[8] R. Fan, L. Cheded and O. Toker, "Designing a SCADA system powered by Java and XML," Computing and Control Engineering, Vol. 16, No. 5, pp. 31-39, 2005

[9] D. Li, Y. Serizawa and M. Kiuchi, "Concept design for a Web-based supervisory control and data-acquisition (SCADA) system," IEEE/PES Transmission and Distribution Conference and Exhibition, Vol. 1, pp. 32-36, 2002

[10] M. Shahidehpour and Y. Wang, Control in Electric Power Systems, Hoboken, New Jersey: Wiley-Interscience, 2003

[11] B. Atlagic, D. Kukolj, V. Kovacevic and M. Popovic, "Application development environment of an integrated SCADA system," in The IEEE Region 8 EUROCON 2003. Computer as a Tool, Ljubljana, 2003

[12] D. Beck, H. Brand, C. Karagiannis and C. Rauth, "A new approach to object oriented programming for real-time targets," in 14th IEEE-NPSS Real Time Conference, Stockholm, Sweden, 2005

[13] Z. Qiang and C. Danyan, "The Research to Power SCADA Based on J2EE Framework," in WASE International Conference on Information Engineering, Taiyuan, China, 2009

[14] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright and K. Ramchandran, "Network Coding for Distributed Storage Systems" in IEEE Transactions On Information Theory, Vol. 56, No. 9, pp. 4539-4551, 2010

[15] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System (DRAFT 3)", Whitepaper, https://ipfs.io

[16] S. McCrady, "Designing SCADA Application Software", Elsevier, 2013

[17] A. Patil et al, "Cloud Object Storage as a Service: IBM Cloud Object Storage from Theory to Practice", IBM Corp, 2017