

POLITECNICO DI TORINO
Mechanical and Aerospace Engineering Department



Master's Degree Thesis

Development and validation of a monitoring model for EMA fault analysis

A. Y. 2018/2019

Relators:

Prof. Paolo Maggiore

Eng. Matteo dalla Vedova

Eng. Piercarlo Berri

Ext. Relators:

Prof. Giovanni Jacazio

Candidate:

Francesco Viglione

Contents

1	Introduction	6
1.1	Brush-less Electrical Motors	7
1.1.1	Clarke-Park	9
2	High Fidelity Model	12
2.1	Model Description	12
2.2	Equivalent Single-Phasic current computation	13
3	Simplified Model	15
3.1	Model Description	15
3.1.1	Control Electronics (PID)	15
3.1.2	Nominal Conditions Electrical Model	18
3.1.3	Electrical Model with Failures Modeling	19
3.1.4	Winding SC Correction	21
3.1.5	Rotor Eccentricity Correction	23
3.1.6	Output	24
3.1.7	Motor-Transmission Dynamical Model	25
3.2	Model Calibration	28
3.2.1	Optimization Toolbox	29
3.2.2	Genetic Algorithms	29
3.2.3	Monitoring Model Current Response Analysis	32
3.3	Model Optimization	38
3.3.1	Single Winding Short Circuit	40
3.3.2	Multiple Windings Short Circuit	50
3.3.3	Multiple Windings Short Circuit and Rotor Eccentricity	69
3.4	Model Validation	89
3.5	Borello-Dalla Vedova Dry Friction Model	90
4	Conclusions and Next Steps	93
	Bibliography	94

List of Figures

1.1	Boeing 777 and 787 FCS architecture	6
1.2	Four poles BLDC motor scheme	7
1.3	Trapezoidal motor wave forms	8
1.4	Sinusoidal motor wave forms	9
1.5	Clarke-Park reference plans	10
2.1	High Fidelity PMSM EMA model	12
2.2	High Fidelity Inverter model	13
2.3	Single-Phase Transformation box	14
2.4	High Fidelity model overall sight	14
3.1	Low Fidelity PMSM EMA model	15
3.2	Control Electronics (PID) box LF EMA model	16
3.3	PID box LF EMA model	16
3.4	Anti-Windup box content	17
3.5	Integrative Branch box content	18
3.6	Electrical Model box content	18
3.7	Electrical Model box content with failures modeling “enabled”	20
3.8	Faults box content	20
3.9	Electrical Model box content, with Faults “deployed”	21
3.10	Winding SC Correction box content	21
3.11	Rotor Eccentricity Correction box content	23
3.12	Static eccentricity scheme: the dotted circle represents the rotor in nominal conditions, the blue one is the stator, the yellow one is the rotor with a static eccentricity	24
3.13	Output box content	25
3.14	Motor-Transmission Dynamical Model box content	25
3.15	Monitoring model overall sight	27
3.16	Chirp command trend	28
3.17	Example of initial population created by a GA: note that all the individuals in the initial population lie in the upper-right quadrant of the picture, that is, their coordinates lie between 0 and 1	30
3.18	First generation children, classified by type of their generation	30
3.19	60th and 80th generations population	31
3.20	95th and 100th generations population	31
3.21	HF and LF current response to a chirp command, with no load	32
3.22	Monitoring system response to a chirp command with no load	33
3.23	First stages of monitoring system response to a chirp command with no load	33
3.24	Detail on speed and position at first stages of monitoring system response to a chirp command with no load	34
3.25	Detail on speed and error position at first command inversion	35
3.26	Detail on positions at first command inversion	36
3.27	Detail on the system response first command inversion	36
3.28	HF and LF current response to a chirp command, with no load, after calibration	38
3.29	Example of a situation, where utilization of normal error makes difference	39
3.30	Current output comparison before and after optimization, with 23% of winding A working	40
3.31	K_{f_s} optimized data interpolation, with winding A SC	42
3.32	K_{f_t} optimized data interpolation, with winding A SC	43

3.33	Current trends with winding A 75% SC, respectively: before, after optimization, all together	45
3.34	Current trends with winding A 61% SC, respectively: before, after optimization, all together	47
3.35	Current trends with winding A 50% SC, respectively: before and after optimization . . .	48
3.36	Current trends with winding A 25% SC, respectively: before and after optimization . . .	49
3.37	Current trends with $[N_A, N_B, N_C] = [0.86, 0.038, 0.99]$, respectively: before, after optimization, all together	52
3.38	Current trends with $[N_A, N_B, N_C] = [0.99, 0.87, 0.71]$, respectively: before, after optimization, all together	54
3.39	Current trends with $[N_A, N_B, N_C] = [0.82, 0.98, 0.86]$, respectively: before, after optimization, all together	56
3.40	Current trends with $[N_A, N_B, N_C] = [1, 0.99, 0.99]$, respectively: before, after optimization, all together	58
3.41	Current trends with $[N_A, N_B, N_C] = [0.54, 0.86, 0.17]$, respectively: before, after optimization, all together	60
3.42	Current trends with $[N_A, N_B, N_C] = [0.38, 0.79, 0.79]$, respectively: before, after optimization, all together	62
3.43	Current trends with $[N_A, N_B, N_C] = [0.22, 0.55, 0.91]$, respectively: before, after optimization, all together	64
3.44	K_{f_i} optimized data interpolation, with A winding SC	65
3.45	K_{f_i} optimized data interpolation, with B winding SC	67
3.46	K_{f_i} optimized data interpolation, with C winding SC	68
3.47	Current trends with $[N_A, N_B, N_C] = [0.99, 0.97, 0.68]$ and $[Z, \phi_Z] = [0.51, 0.0007]$ respectively: before, after optimization, all together	71
3.48	Current trends with $[N_A, N_B, N_C] = [0.99, 0.98, 0.88]$ and $[Z, \phi_Z] = [0.05, 0.42]$ respectively: before, after optimization, all together	73
3.49	Current trends with $[N_A, N_B, N_C] = [0.90, 0.98, 0.99]$ and $[Z, \phi_Z] = [0.09, 0.0003]$ respectively: before, after optimization, all together	75
3.50	Current trends with $[N_A, N_B, N_C] = [0.71, 0.99, 0.64]$ and $[Z, \phi_Z] = [0.06, 0.0006]$ respectively: before, after optimization, all together	77
3.51	Current trends with $[N_A, N_B, N_C] = [1, 0.96, 0.95]$ and $[Z, \phi_Z] = [0.0004, 0.0001]$ respectively: before, after optimization, all together	79
3.52	Current trends with $[N_A, N_B, N_C] = [0.30, 0.96, 0.99]$ and $[Z, \phi_Z] = [0.0025, 0.0006]$ respectively: before, after optimization, all together	81
3.53	Current trends with $[N_A, N_B, N_C] = [0.71, 0.63, 0.27]$ and $[Z, \phi_Z] = [0.02, 0.01]$ respectively: before, after optimization, all together	83
3.54	K_{f_i} optimized data interpolation, with A winding SC + E	84
3.55	K_{f_i} optimized data interpolation, with B winding SC + E	85
3.56	K_{f_i} optimized data interpolation, with C winding SC + E	87
3.57	K_{f_E} optimized data interpolation, with rotor static eccentricity	88
3.58	K_{f_E} optimized data interpolation, with rotor static eccentricity, in a logarithmic horizontal scale	89
3.59	<i>NRMSE</i> histograms graph, with validation data Gaussian ditribution	90
3.60	Borello dry friction Grafic representation	91
3.61	Borello Friction Model box content	92

List of Tables

- 3.1 Before/After calibration data summary 38
- 3.2 Winding A SC data summary 41
- 3.3 Windings A, B, C SC data summary 50
- 3.4 Windings A, B, C SC and E data summary 69

- 4.1 Models simulation time summary 93

1 | Introduction

During the latest years, the necessity to detect failures occurred to any aircraft on-board system in their very first stages, before they potentially become serious (or worst, catastrophic), became very strong. Particularly for those system classified as *primary flight controls* (including the primary moving surfaces and everything related with the propulsion plant), in order to fulfill this need, a large effort has been deploying towards diagnostic and prognostic algorithms development. The role of these latter includes the capacity of understanding, as a first approach, and preventing, as the final objective, some of the most common failures known to occur to certain kind of installations. The ideal application of these studies would be producing something able to directly run on-board and in real time, in order to monitor systems during the missions, provide early warning in case of fault detection and, when possible, take a targeted action to prevent its worsening.

The specific aim of this work, being in perfect harmony with the just explained approach, is to develop an *Electro-Mechanical Actuator (EMA)* monitoring model, able to replicate in a sufficiently reliable way the real system behavior when a motor degradation occurs. The faults we are going to focus on, are the *within-phase* short circuits and rotor *static eccentricity* (see sections 3.1.4 and 3.1.5). Typically, this kind of actuators are employed for mobile surfaces handling by an *all or more-electric* flight control system architecture: basically the input signal given by the pilot or by the autopilot, is processed by the flight control computer and sent to an electric motor, directly responsible of the surface mechanical actuation. As an example, the figure below shows the typical Boeing FCS schematized architecture:

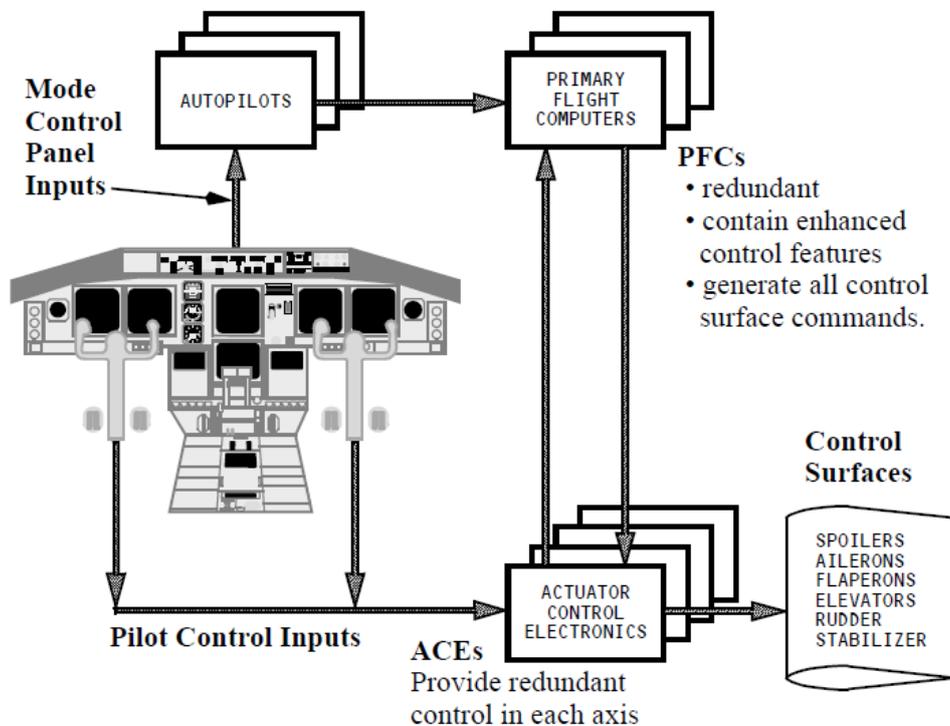


Figure 1.1: Boeing 777 and 787 FCS architecture

Before facing the Matlab-Simulink models utilized within this work, a deeper look at brush-less, sinusoidal electric motors is required, in order to better understand their functioning principles.

1.1 Brush-less Electrical Motors

The *Brush-less Direct Current (BLDC)* motor is a DC electric motor with a permanent magnet rotor and a rotating magnetic field stator.

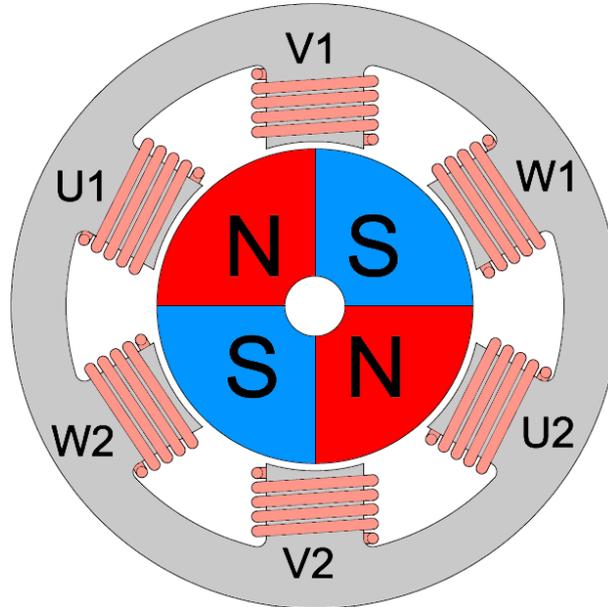


Figure 1.2: Four poles BLDC motor scheme

In a traditional brushed motor, the brushes mechanical contact with the rotating manifold closes the electric circuit between the power supply and the windings on the rotor, periodically reversing the direction of current circulation in the coils. Conversely in a BLDC, the switching of current circulating in the stator windings, and therefore their magnetic field direction changing, needs an electronic control; this way, beyond the reduction of the overall dimensions with the same provided power, big advantages come from the electrical creeping contacts avoidance:

- **Mechanical resistance reduction:** the absence of creeping parts allows to reduce the friction contribution to the resistant torque;
- **Chance of sparks forming elimination:** in a brushed motor, with a huge rotational speed, sparks generation is very probable. By eliminating this components, no more open flames danger on board is ensured;
- **Reduction of periodical maintenance:** no more consuming parts, means less maintenance needed.

Therefore, the motor expected life will be increased, the electromagnetic noise, instead, will be considerably reduced and an overall efficiency growth due to the necessity of generating the rotor magnetic field lack, will be ensured. The permanent magnets on the rotor, are by now realized with special materials allowing to further reduce its inertia and, consequently, to have extremely precise control both in speed and acceleration. Some of these materials, which are now replacing ferrite, both for mass and generated magnetic field density matters, are Neodymium (Nd), Samarium-Cobalt (SmCo) and the Neodymium-Ferrite-Boron alloy (NdFeB). Since the BL motor operates in direct current, an electronic circuit composed by power transistors controlled by a micro-controller is used and, this latter, takes care of current switching, whose variation, in turn, causes the stator generated magnetic field rotation. On the other hand, however, since the controller must know the rotor position with respect to the stator in order to determine the direction to be given to the magnetic field (hence the nickname of synchronous motor), an Hall effect sensor like a *pick-up* or a more precise *resolver* adoption is necessary. Anyway, it's easy to understand that all these advantages cause a device cost increasing: unlike brushed motors, in which the control is made by a potentiometer or a rheostat (inefficient systems, but extremely cheap), the electronic control for speed regulation in BL motors (supplied by the motor manufacturer or by third parties), has an additional cost and must be entirely dedicated to a single motor. Especially in the aeronautical sector,

the increase in reliability and, therefore in safety, is one of the requirements to be complied with in the design phase and in the choice of components. Thus, the consequent cost increasing is a factor playing a secondary role and, often, the most common choice is the more reliable and, in this case, even more efficient.

A further distinguishing, among the BLDC motors, needs to be done between trapezoidal and sinusoidal wave form ones. The main difference between these two configurations deals with the different way of coils supplying: the first one, as shown in figure 1.3, sees a squared current signal which induces a trapezoidal waveform in the back EMF of the single coils.

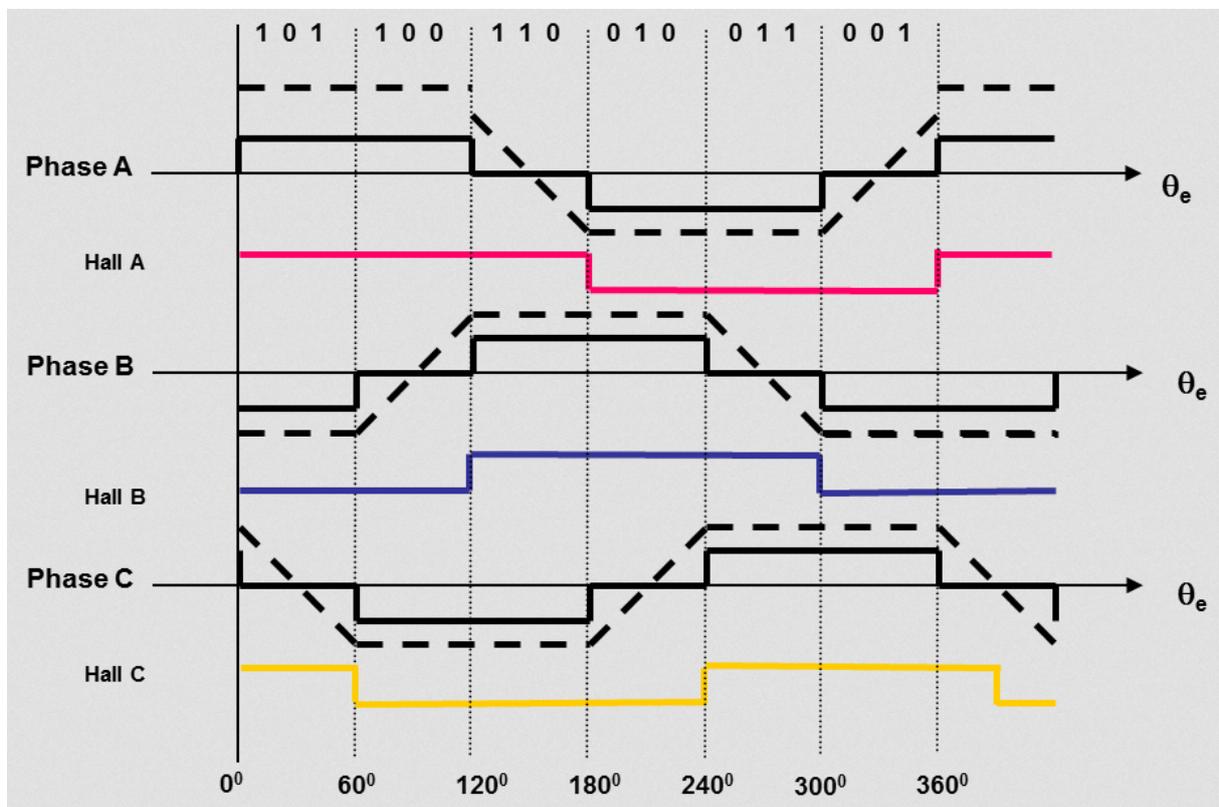


Figure 1.3: Trapezoidal motor wave forms

These latter have a 120° offset between each other in order to ensure a uniform coverage during the complete revolution. Conversely, the sinusoidal motor, supplies the single coils with a such waveform, each with a 120° offset (see figure 1.4), inducing an analogous response in terms of back EMF.

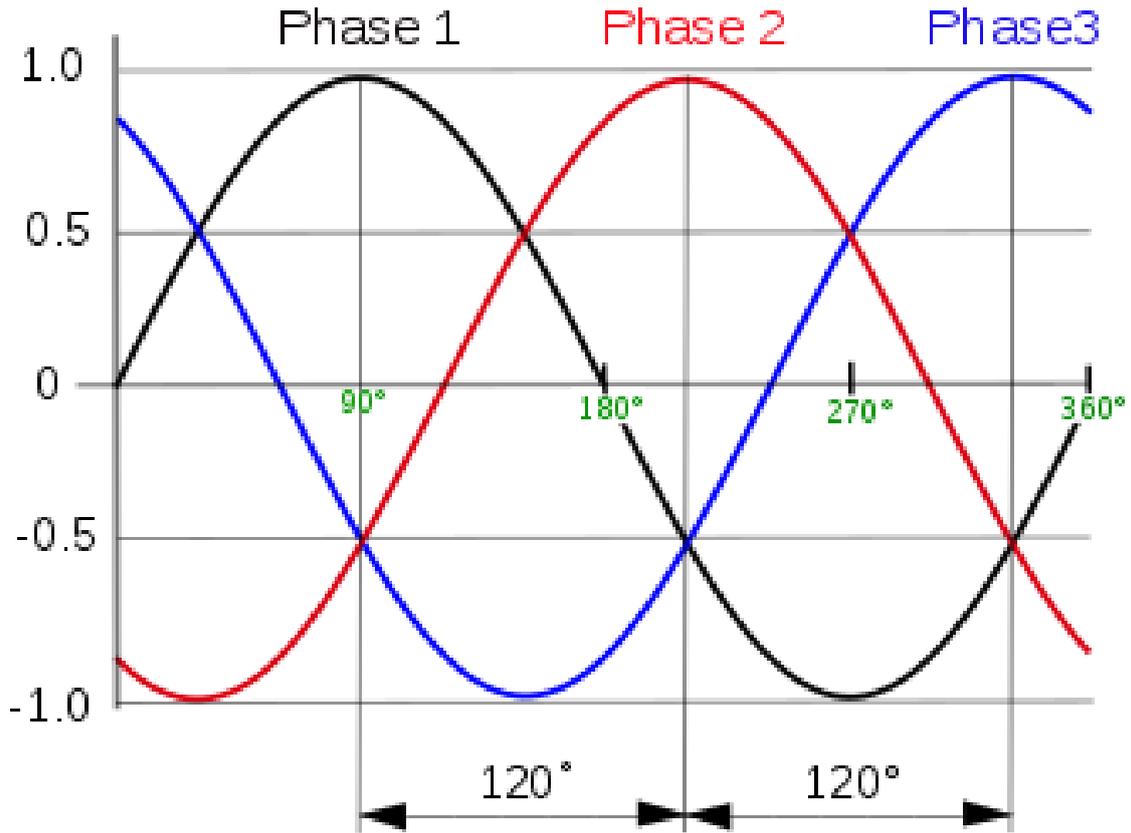


Figure 1.4: Sinusoidal motor wave forms

This latter motor ensures a smoother electro-magnetic field transition during the revolution and, thus, even a smoother rotor motion.

1.1.1 Clarke-Park

As first operation, a reference plan change is required: since we have a rotating object (motor shaft), we need to use a united reference to it in order the latter to be inertial. That's why we use the *Clarke-Park transformation*: the first step is using the *Clarke matrix*, $[C]$ which transforms the fixed coils system coordinates into those of another fixed, right-handed one, given by $\alpha - \beta$ axes ($\alpha \equiv A$ and β perpendicular to it, see figure 1.5), thus having the following current components;

$$\begin{Bmatrix} I_\alpha \\ I_\beta \end{Bmatrix} = [C] \cdot \begin{Bmatrix} I_A \\ I_B \\ I_C \end{Bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{Bmatrix} I_A \\ I_B \\ I_C \end{Bmatrix} \quad (1.1)$$

Causing:

$$\begin{cases} I_\alpha = \frac{2}{3}I_A - \frac{1}{3}(I_B + I_C) \\ I_\beta = \frac{\sqrt{3}}{3}(I_B - I_C) \end{cases} \quad (1.2)$$

Where I_A, I_B, I_C are the phase currents for a star configuration.

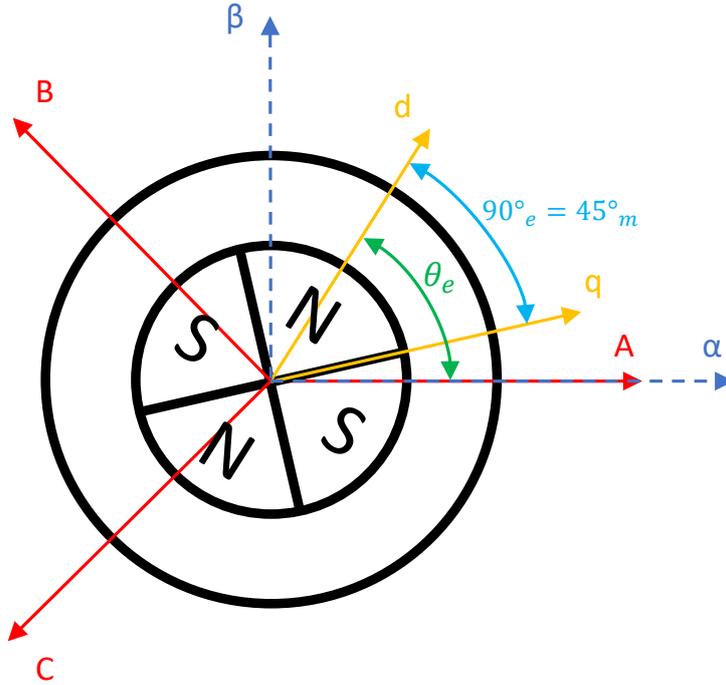


Figure 1.5: Clarke-Park reference plans

The second step is to move into the rotary reference, through the *Park matrix*, $[P]$ which indeed, transforms the Clarke plan coordinates into those of a changing system, integral with the rotor, and composed by *direct* (d) and *quadrature* (q) axes;

$$\begin{Bmatrix} I_d \\ I_q \end{Bmatrix} = [P] \cdot \begin{Bmatrix} I_\alpha \\ I_\beta \end{Bmatrix} = \begin{bmatrix} \cos \theta_e & \sin \theta_e \\ -\sin \theta_e & \cos \theta_e \end{bmatrix} \cdot \begin{Bmatrix} I_\alpha \\ I_\beta \end{Bmatrix} \quad (1.3)$$

The direction \hat{d} , given by the direct axis, points to one of the rotor *north poles*, while \hat{q} indicates the quadrature direction, perpendicular to it in terms of *electrical angle*, in such a way a right-handed system is obtained. Be aware that, since we are using electrical angle, the actual “mechanical” one depends upon the poles pairs number, indeed:

$$\theta_e = N_p \cdot \theta_m \quad (1.4)$$

which in our case, choosing $N_p = 2$, leads to a mechanical angle between \hat{d} and \hat{q} given by:

$$\theta_{mC-P} = \pi/4 = 45^\circ \quad (1.5)$$

From the 1.3 we have:

$$\begin{cases} I_d = I_\alpha \cos \theta_e + I_\beta \sin \theta_e \\ I_q = -I_\alpha \sin \theta_e + I_\beta \cos \theta_e \end{cases} \quad (1.6)$$

Where I_d is the *direct current* and I_q the *quadrature* one. Finally, the full Clarke-Park transformation is given by the following:

$$\begin{aligned} \begin{Bmatrix} I_d \\ I_q \end{Bmatrix} &= [P] \cdot \begin{Bmatrix} I_\alpha \\ I_\beta \end{Bmatrix} = [P] \cdot [C] \cdot \begin{Bmatrix} I_A \\ I_B \\ I_C \end{Bmatrix} = \\ &= \frac{2}{3} \cdot \begin{bmatrix} \cos \theta_e & \sin \theta_e \\ -\sin \theta_e & \cos \theta_e \end{bmatrix} \cdot \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{Bmatrix} I_A \\ I_B \\ I_C \end{Bmatrix} \end{aligned} \quad (1.7)$$

With:

$$\begin{cases} I_d = \left[\frac{2}{3}I_A - \frac{1}{3}(I_B + I_C) \right] \cos \theta_e + \frac{\sqrt{3}}{3}(I_B - I_C) \sin \theta_e \\ I_q = \left[-\frac{2}{3}I_A + \frac{1}{3}(I_B + I_C) \right] \sin \theta_e + \frac{\sqrt{3}}{3}(I_B - I_C) \cos \theta_e \end{cases} \quad (1.8)$$

Where we have the quadrature and direct currents as a function of the phase ones and of rotor electrical position.

It is possible, of course, operating the reverse transformation, by using the *inverse Clarke* and *Park matrices*:

$$[C]^{-1} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \quad (1.9)$$

$$[P]^{-1} = \begin{bmatrix} \cos \theta_e & -\sin \theta_e \\ \sin \theta_e & \cos \theta_e \end{bmatrix} \quad (1.10)$$

$[C]^{-1}$ is obtained through the *Moore-Penrose inverse*, the most diffused way to calculate the matrices *pseudoinverse*, while being $[P]$ an orthogonal matrix, we have:

$$[P]^{-1} = [P]^t \Rightarrow [P] \cdot [P]^t = [I] \quad (1.11)$$

Thus, the overall inverse conversion is given by the following:

$$\begin{aligned} \begin{Bmatrix} I_A \\ I_B \\ I_C \end{Bmatrix} &= [C]^{-1} \cdot \begin{Bmatrix} I_\alpha \\ I_\beta \end{Bmatrix} = [C]^{-1} \cdot [P]^{-1} \cdot \begin{Bmatrix} I_d \\ I_q \end{Bmatrix} = \\ &= \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} \cos \theta_e & -\sin \theta_e \\ \sin \theta_e & \cos \theta_e \end{bmatrix} \cdot \begin{Bmatrix} I_d \\ I_q \end{Bmatrix} \end{aligned} \quad (1.12)$$

Leading to:

$$\begin{cases} I_A = I_d - I_q \sin \theta_e \\ I_B = \frac{1}{2} \left[(\sqrt{3}I_q - I_d) \cos \theta_e + (\sqrt{3}I_d + I_q) \sin \theta_e \right] \\ I_C = \frac{1}{2} \left[(-\sqrt{3}I_q - I_d) \cos \theta_e + (-\sqrt{3}I_d + I_q) \sin \theta_e \right] \end{cases} \quad (1.13)$$

2 | High Fidelity Model

Since the aim of this work is to provide a simplified model for an Electro-Mechanical Actuator (EMA), able to run in real time during the actuation, in order to provide early information about possible failures which are already happening or, in a most suitable case, which are going to occur in the very next future, a sinusoidal EMA high fidelity model, deeply described in [1], has been modified in terms of current output.

2.1 Model Description

As showed by figure 2.1, the Matlab-Simulink model is composed by several subsystems, each of them accomplishing a specific task.

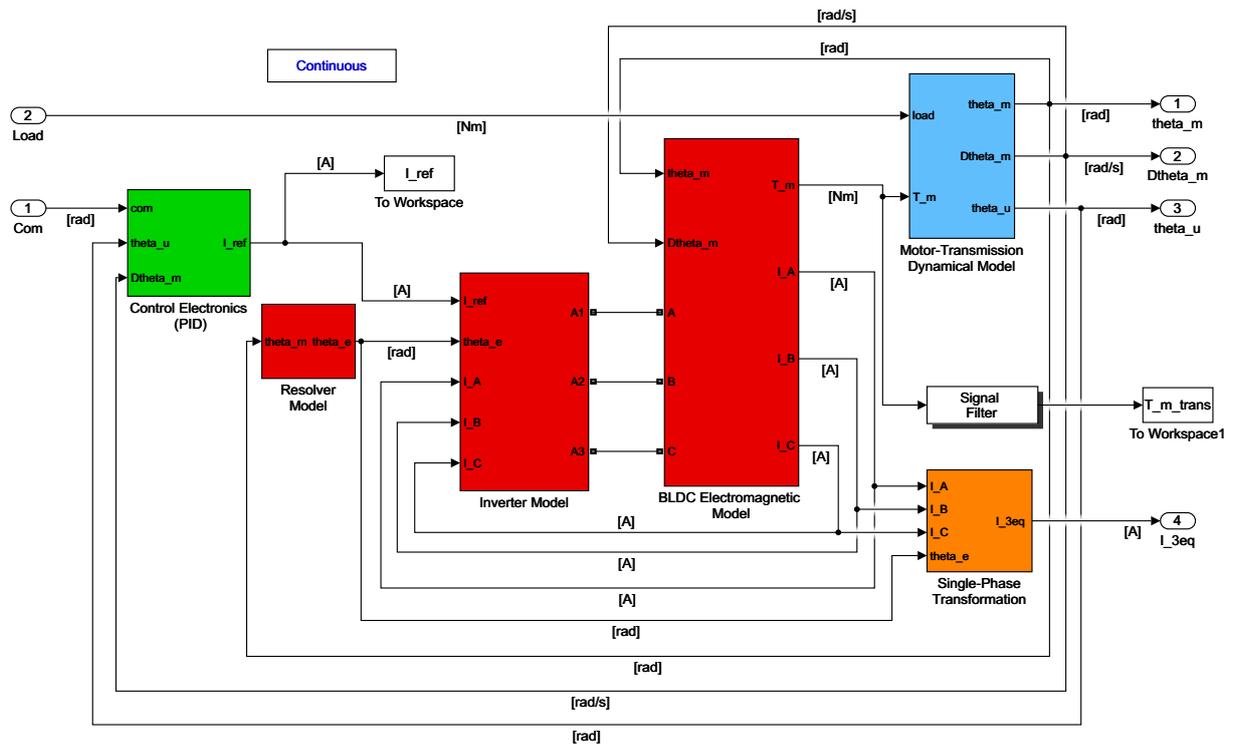


Figure 2.1: High Fidelity PMSM EMA model

- **Control Electronics (PID):** takes the position command, the feedback utilized position and motor angular speed as inputs, and provides the reference current to be given forward;
- **Resolver Model:** simply converts the motor mechanical angular position to the electrical one, through the expression $\theta_e = N_p \cdot \theta_m$, with N_p poles pair number;
- **Inverter Model:** receives the reference current, the motor electrical position, and the three-phased current as feedback; it's physically connected to the *BLDC EM Model* through A, B, C connections.

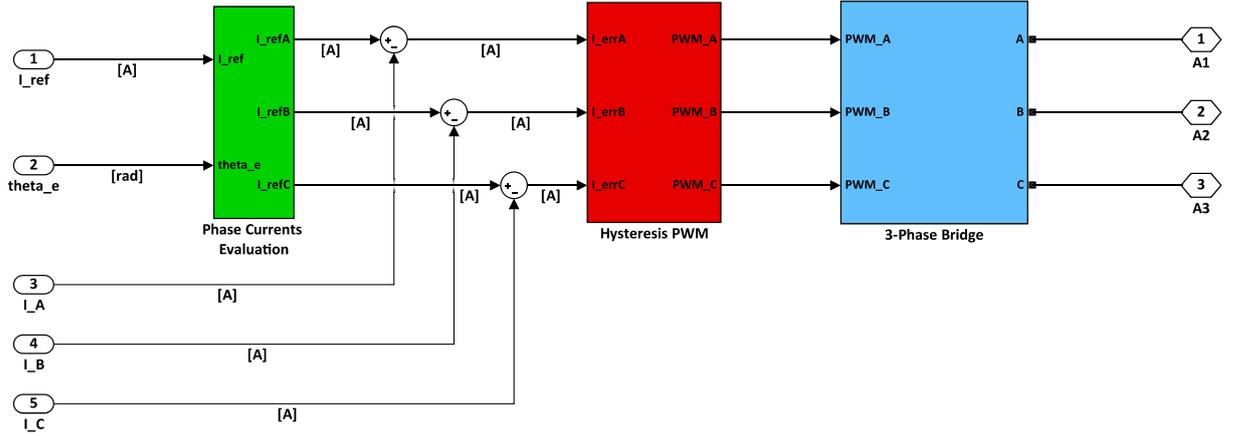


Figure 2.2: High Fidelity Inverter model

The figure 2.2, shows that the *Phase Current Evaluation* box transforms the single-phased reference current into the three-phased one; this box contains the Clarke-Parke transformation described in the dedicated section (1.1.1), while the outputs are subtracted with the current feedback. Afterwards, the *Hysteresis PWM* box contains the pulse width modulation process, whose products feed the *3-phase Bridge* box, containing the three-phase bridge electrical model;

- **BLDC Electromagnetic Model:** receives motor angular position and speed as feedbacks, computes the back EMF coefficients and, through an ohmic-inductive model, calculates the motor torque and phase currents;
- **Motor-Transmission Dynamical Model:** fed by the motor torque and the external load, it contains the mechanical model of the motor-reducer group, including the *Borello's friction model*, is able to provide motor angular speed and position, as much as the utilizer one;
- **Single-Phase Transformation:** computes a single-phasic current “proportional” to the three-phasic one. It’s content is going to be deeply described in the next section;

Notice that the *Control Electronics (PID)* and *Motor-Transmission Dynamical Model* boxes, are going to be widely described in the Simplified Model section (3), given the fact they’re exactly the same.

2.2 Equivalent Single-Phasic current computation

It has been necessary to build a single-phasic equivalent current in pursuance of having a comparable output parameter with the monitor model, which is indeed, single-phasic. This aim has been reached through *Single-Phase Transformation* box, reported in figure 2.3, which uses the Clarke-Park transformation and gives a two components output current:

$$\begin{aligned} \vec{I}_{3_{eq}} &= I_q \cdot \hat{q} + I_d \cdot \hat{d} \\ I_d &= 0 \end{aligned} \quad (2.1)$$

$$I_q = \left[-I_A + \frac{1}{2}(I_B + I_C) \right] \sin \theta_e + \frac{\sqrt{3}}{2}(I_B - I_C) \cos \theta_e \quad (2.2)$$

Where I_q and I_d are respectively the quadrature and direct current (see section 1.1.1, for further information). Since a good permutation sequence requires the resultant coils supply current to be always perpendicular (in terms of electric angle) to the rotor magnetic field, this condition has been used to ensure a proper permutation and explains the fact that I_d is always going to be null, thus causing

$$\vec{I}_{3_{eq}} = I_q \cdot \hat{q} \quad (2.3)$$

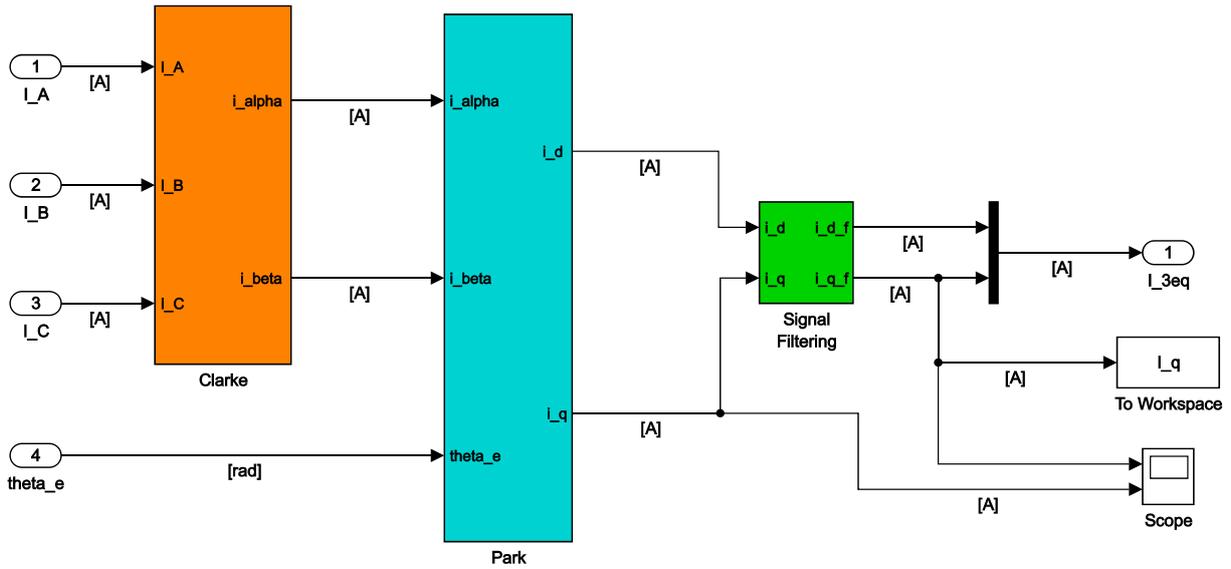


Figure 2.3: Single-Phase Transformation box

Furthermore, this characteristic widely justifies the choosing of quadrature current as the comparison parameter; in the figure just above, we can see the three-phase current contributions of the single coils coming from the *BLDC Electro-Magnetic Model* and being the *Clark* box input, which contains the Clark matrix [C] (equation 1.1); Indeed, its outputs are I_{α} and I_{β} , which are going to enter the Park box together with the rotor electrical angle, containing the Park matrix [P] (equation 1.3). Once obtained the direct and quadrature currents, it's needed a filtering stage cleaning up the signal from higher frequencies given by superior order phenomena such as PWM and undesired numerical noise. Finally, we have the single-phasic equivalent current I_{3eq} , which is going to be used as the comparison parameter with the monitor model we are going to describe in the next section.

Here follows the figure showing the highest level of the model described above, just to give an overall sight and better clarify where the inputs signal comes from, as much as where the output ones go to.

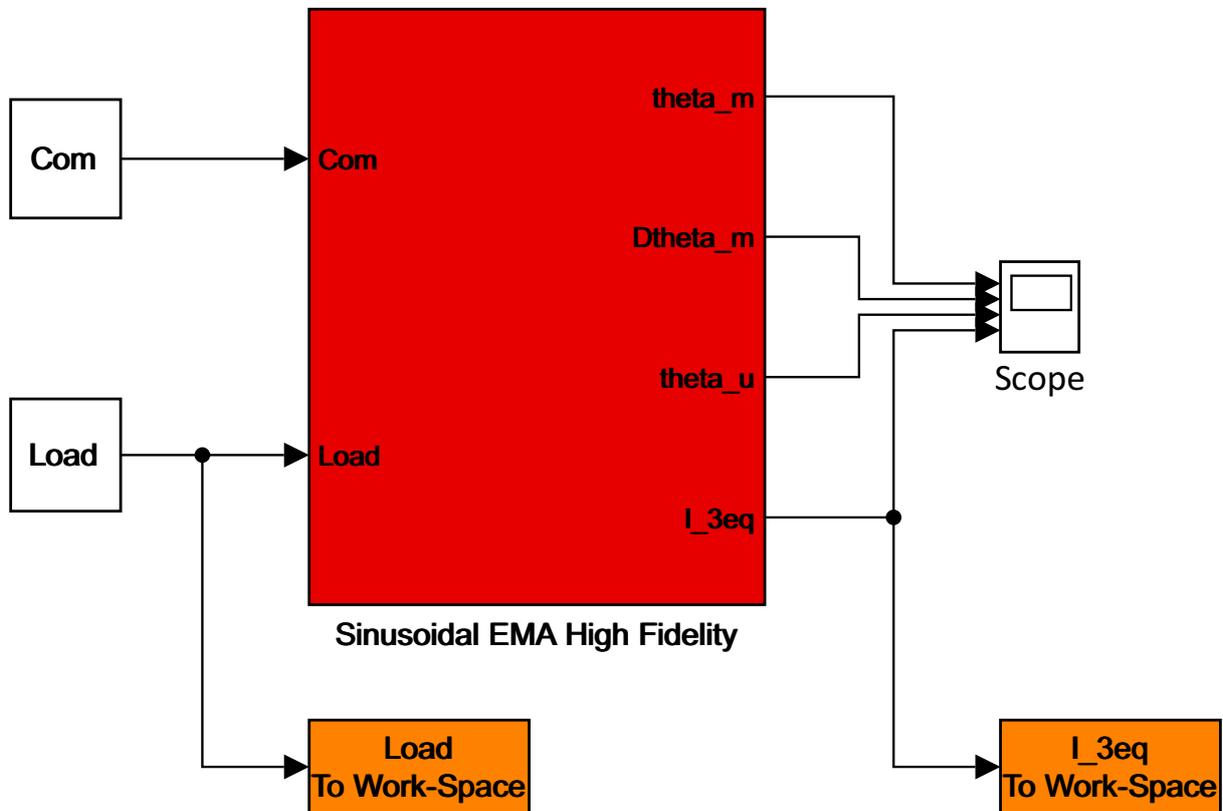


Figure 2.4: High Fidelity model overall sight

3 | Simplified Model

In this section we are going to describe a “low fidelity” model coming from the exemplification of the high fidelity one presented in the previous chapter. The main operation conducted for this purpose, is been “condensing” the red boxes of figure 2.1 into a single one, as shown in figure 3.1:

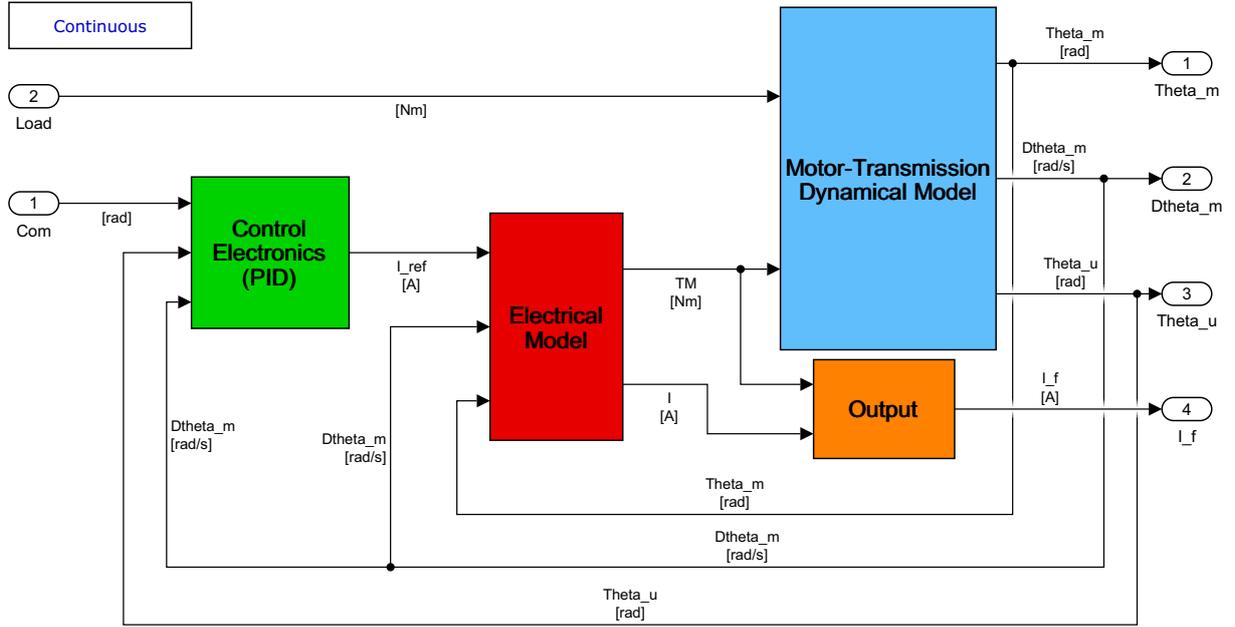


Figure 3.1: Low Fidelity PMSM EMA model

Basically, it’s been skipped the three-phase current conversion and, thus, no Clarke-Park inverse transformation, hysteresis PWM generation and three-phase bridge electrical model (see figure 2.2) are needed anymore. At the same time, the electro-magnetic model doesn’t require computation of three back EMF coefficients, currents, inductances and voltages, allowing to replace everything with a simpler single-phasic ohmic-inductive model (*Electrical Model* in figure 3.1), which is going to be described below.

3.1 Model Description

Let’s see in detail the monitor model, shown in figure 3.1: there is a position command given in radians (*Com*), which is elaborated by *Control Electronics (PID)* box, together with “low speed” shaft angular position (*Theta_u*, due to a reduction stage downstream of the motor) and motor speed loops (*DTheta_m*, related to “high speed” shaft), which produces a reference current *I_ref* as output.

3.1.1 Control Electronics (PID)

The control electronics box, represented in figure 3.2, calculates the position error ($Err_Pos = Com - Theta_u$) and both uses it as input to the proper *PID* box and multiplies it by the controller proportional gain, set to:

$$G_{prop} = 10^{-5} \left[\frac{1}{s} \right] \quad (3.1)$$

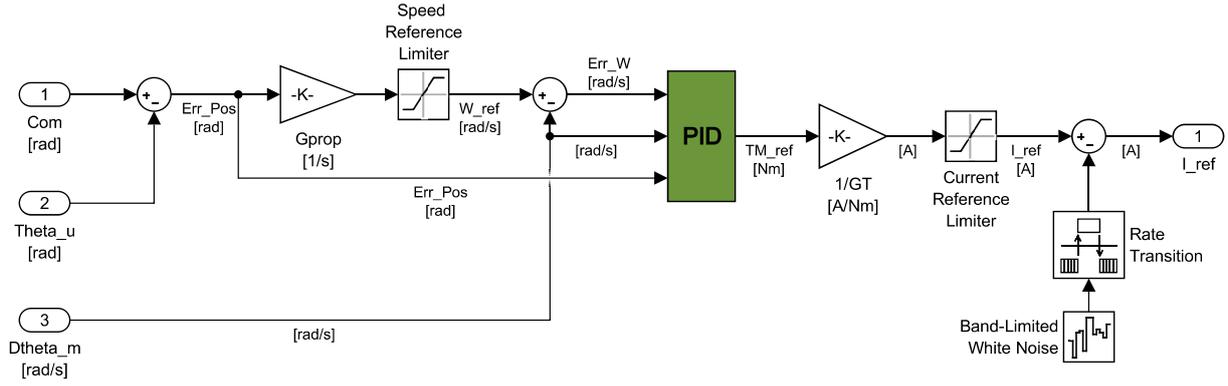


Figure 3.2: Control Electronics (PID) box LF EMA model

Thus, the actuation speed is obtained, followed by its saturation set to:

$$\omega_{reflim} = \pm 8000 [RPM] \cdot \frac{2\pi}{60} = \pm 800 \cdot \frac{\pi}{3} \left[\frac{rad}{s} \right] \quad (3.2)$$

The next step is the speed error calculation ($Err.W = W_{ref} - D\theta_{m}$), which enters along with the motor speed, the *PID* box (see section 3.1.1.1). Its output is the reference motor torque (TM_{ref}) being multiplied by the *gain torque constant* (GT) inverse, contained in the gain box $1/GT$; this way, the reference current I_{ref} is obtained, and enters the *Current Reference Limiter* box, which states its admitted boundary values:

$$I_{reflim} = \pm I_{maxHF} = \pm 22.5 [A] \quad (3.3)$$

Where $I_{maxHF} = 22.5 [A]$, is the maximum allowed current for the high fidelity model. The *Band-Limited White Noise* box, simulates the signals white noise, by adding a certain frequency fluctuation, which in this case is null since not needed for the monitoring model aims.

3.1.1.1 PID

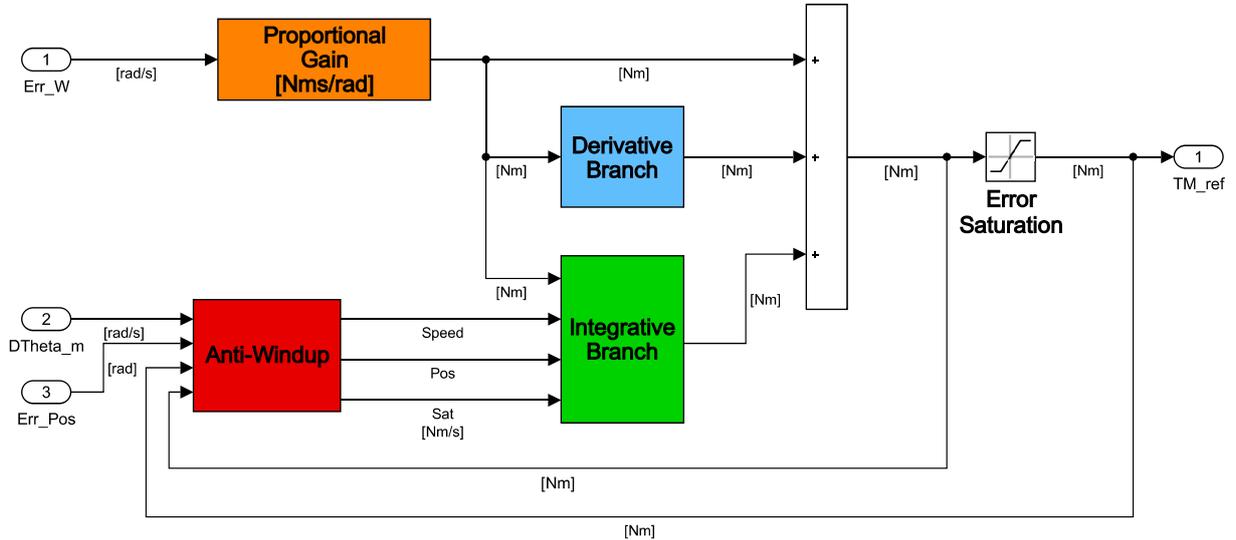


Figure 3.3: PID box LF EMA model

From the figure 3.3, we can see the speed error entering the *Proportional Gain* box and, in there, being multiplied by the:

$$GAP = 0.05 \left[\frac{Nm \cdot s}{rad} \right] \quad (3.4)$$

The output of the orange box has torque dimensions, indeed we have:

$$\left[\frac{\text{rad}}{s} \right] \cdot \left[\frac{\text{Nm} \cdot s}{\text{rad}} \right] = [\text{Nm}] \quad (3.5)$$

and feeds both the *Derivative* and *Integrative* branches; the first one contains the related dynamics, given by the f.d.t.:

$$\frac{T_D \cdot f_b \cdot s}{T_D \cdot s + f_b} = \frac{T_D \cdot s}{\frac{T_D}{f_b} \cdot s + 1} \quad (3.6)$$

Where $T_D = 0$ [s] is the *characteristic time* and $f_b = 1$ [kHz] is the *band pass*; as we can see, his contribution is null, due to a good model attitude, not requiring any further action in this sense, but it can be set to different values in case of necessity by modifying the script parameters.

At the right bottom, we can see the motor shaft speed and its position error entering the *Anti-Windup* box together with motor torque output loops, before and after the integrative *Error Saturation*, set to:

$$E_{sat} = \pm 100 [\text{Nm}] \quad (3.7)$$

All the outputs from the red box converge into the green one and, after the underwent integration in there, we finally have the sum of three branches contribution, producing the reference motor torque TM_{ref} .

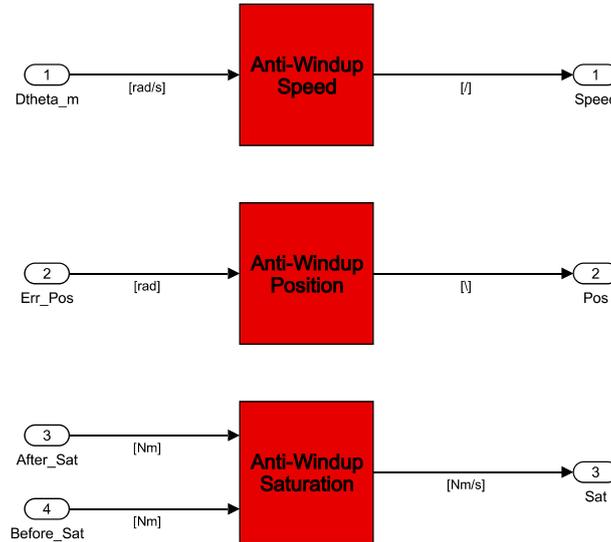


Figure 3.4: *Anti-Windup box content*

The figure 3.4 unveils the anti-windup box content and shows three different kind of filters.

- **Anti-Windup Speed:** inhibits the integrative branch if the rotation speed is null for more than one integration step; it gives a value greater or lower than 0, feeding the switch contained into *Integrative Branch* described below, in order to fulfill this task;
- **Anti-Windup Position:** inhibits the integrative branch if the position error is lower than a given value ($Toll_i = 10^{-3}$ [rad]); if the condition is verified, the output is going to be 0, 1 in the other cases;
- **Anti-Windup Saturation:** compensates for the integrative branch if there's a command saturation. Indeed, when the integrative error saturation occurs, the difference between *After_Sat* and *Before_Sat* is compensated by the gain $\frac{1}{T_t}$ [$\frac{1}{s}$], producing:

$$Sat = (After_Sat - Before_Sat) [\text{Nm}] \cdot \frac{1}{T_t} \left[\frac{1}{s} \right] = (After_Sat - Before_Sat) \left[\frac{\text{Nm}}{s} \right] \quad (3.8)$$

$T_t = 1$ [s], being the filter characteristic time;

Speed, *Pos* and *Sat* signals lead to the *Integrative Branch* box (see figure 3.3), composed as follows:

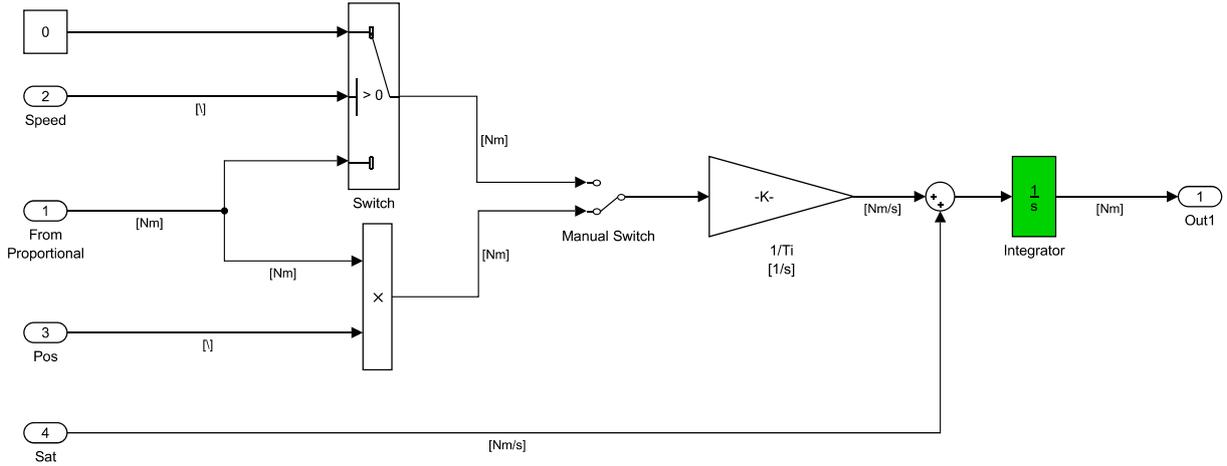


Figure 3.5: Integrative Branch box content

We can see the *Manual Switch* choosing between speed and position filtering and ensuring its output to have torque dimensions; the signal coming *From Proportional* enters the *Switch* box (if speed filtering is enabled), which selects 0 or the signal coming from proportional whether *Speed* is, respectively, greater or lower then 0. On the other hand, if position filtering is selected, *Pos* simply multiplies the input from proportional in such a way it can be 0 or itself. Downstream the manual switch, we find a gain box $\frac{1}{T_i}$ multiplying the signal by:

$$\frac{1}{T_i} \left[\frac{1}{s} \right] = 10^{-4} \left[\frac{1}{s} \right] \quad (3.9)$$

which produces a quantity with the correct dimensions:

$$[Nm] \cdot \left[\frac{1}{s} \right] = \left[\frac{Nm}{s} \right] \quad (3.10)$$

Next step is the sum between *Sat* and the latter gain box output, which can finally be integrated by the green box and provide the integrative torque contribution to the PID controller.

3.1.2 Nominal Conditions Electrical Model

Once obtained the I_{ref} from the *Control Electronics (PID)* box (see figure 3.1), the signal enters *Electrical Model* system, containing the following:

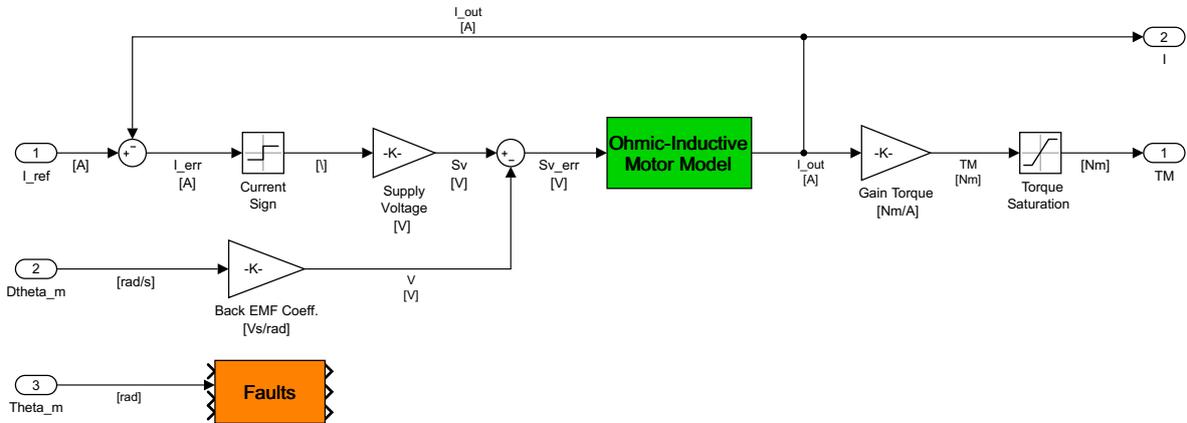


Figure 3.6: Electrical Model box content

The I_{ref} , subtracted with the I_{out} loop, produces the I_{err} :

$$I_{err} = I_{ref} - I_{out} [A] \quad (3.11)$$

Which enters *Current Sign* box, extrapolating its sign in order to assign it to the *Supply Voltage*:

$$S_V = \pm 48 [V] \quad (3.12)$$

Another subtraction stage deprives S_V of the *Back Electro Motive Force* contribution, given by motor angular speed $Dtheta_m$ multiplied by the *Back EMF Coefficient*, equal to:

$$k_{BEMF} = \frac{0.0752}{3} \left[\frac{V \cdot s}{rad} \right] \quad (3.13)$$

Thus having, from a dimensional point of view:

$$V [V] = Dtheta_m \left[\frac{rad}{s} \right] \cdot k_{BEMF} \left[\frac{V \cdot s}{rad} \right] \quad (3.14)$$

Allowing us to write:

$$S_V_err = (S_V - V) [V] \quad (3.15)$$

Which feeds the *Ohmic-Inductive Motor Model*, containing the motor model t.f. given by:

$$\frac{1}{R_m \left[\left(\frac{L_m}{R_m} \cdot \frac{N_{tot}}{3} \right) s + 1 \right]} \quad (3.16)$$

Where, $\frac{N_{tot}}{3}$ is the mean percentage of working coils, being:

$$N_{tot} = N_A + N_B + N_C \quad (3.17)$$

With: $N_{A,B,C}$ percentage of windings A, B, C working coils and $0 \leq N_{A,B,C} \leq 1$, where 1 means completely functional, while 0 means totally damaged. For nominal conditions, we have the following data:

$$\begin{cases} R_m = \frac{2.130}{2} [\Omega] \\ L_m = \frac{720}{2} [\mu F] \\ N_A = N_B = N_C = 1 \\ N_{tot} = 3 \end{cases} \quad (3.18)$$

The L_{out} coming from the green box, goes both to the output 2, as I , and through the *Gain Torque*:

$$GT = 0.0392 \left[\frac{Nm}{A} \right] \quad (3.19)$$

Obtained by the calibration process (see section 3.2).

After this step, we find the *Torque Saturation* box, given by the eventual L_{err} saturation and equal to:

$$TM_{lim} = \pm 1.689 [Nm] \quad (3.20)$$

Finally, we have the TM as output number 1.

3.1.3 Electrical Model with Failures Modeling

Looking at figure 3.6, we can see the orange box *Faults* disconnected; in order to better understand its working principles, in this section we are going to describe the process leading us to build the electrical model, including the failures functions. This is the complete model looking, after connection:

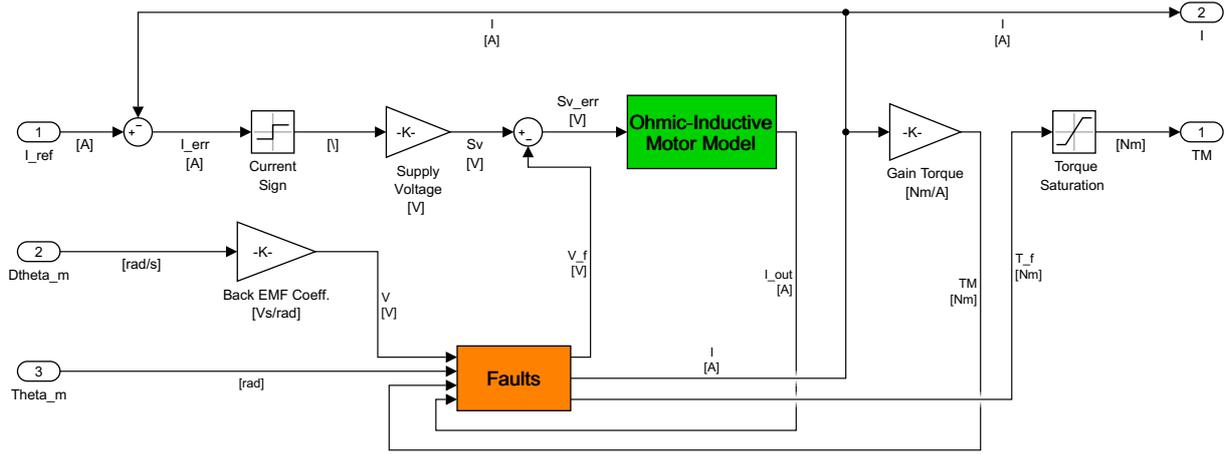


Figure 3.7: Electrical Model box content with failures modeling “enabled”

We can see there is a contribution to the BEM force to the output current I_{out} and, also, to the motor torque one TM , while the last input needed is the motor angular position $Theta_m$. The scheme inside the orange box is the following:

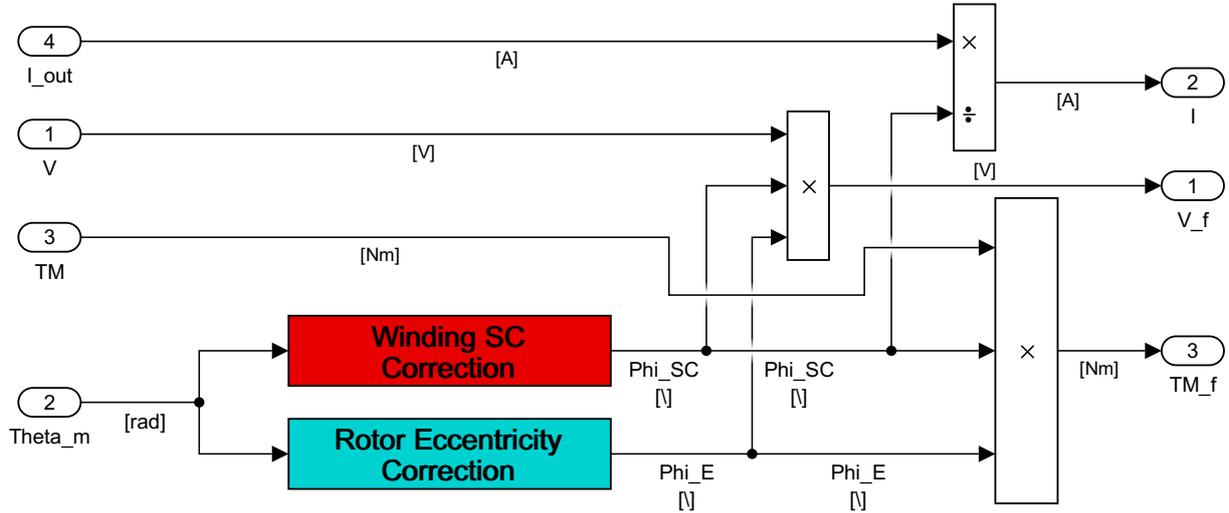


Figure 3.8: Faults box content

Winding SC Correction and *Rotor Eccentricity Correction* take $Theta_m$ input and contain from functions able to model, respectively, the effect of partial windings short circuit and mechanical failures due to rotor deformation or misalignment. In both cases, a non-dimensional multiplicative coefficient is produced and is going to affect V (back EMF contribution), causing $V_f [V]$, I_{out} , giving $I [A]$ and TM , having $TM_f [Nm]$. Of course, the mechanical correction has no direct effects on the ohmic-inductive motor model, that’s why I_{out} is only divided by SC correction: looking at the t.f. of equation 3.16, the most relevant contribution to I_{out} is given by the motor phase-to-phase resistance R_m which, being proportional to the number of working coils, needs to be multiplied by the SC correction coefficient. Moreover, seen its negligible nature compared with R_m , in spite of complicating the model due to having a precise contribution of the inductance L_m multiplying it by SC correction too, a better solution has been considered using the mean working coils number $\frac{N_{tot}}{3}$. This way, the motor t.f. affected by faults, is going to be:

$$\frac{1}{\Phi_{SC} \cdot R_m \left[\left(\frac{L_m}{R_m} \cdot \frac{N_{tot}}{3} \right) s + 1 \right]} \quad (3.21)$$

Where, Φ_{SC} is the SC correction function value; in the same way we have:

$$\begin{cases} V_f = \Phi_{SC} \cdot \Phi_E \cdot V [V] \\ TM_f = \Phi_{SC} \cdot \Phi_E \cdot TM [Nm] \end{cases} \quad (3.22)$$

Φ_E being the eccentricity correction function parameter; for greater clarity, here the scheme of electrical model with faults box “deployed” :

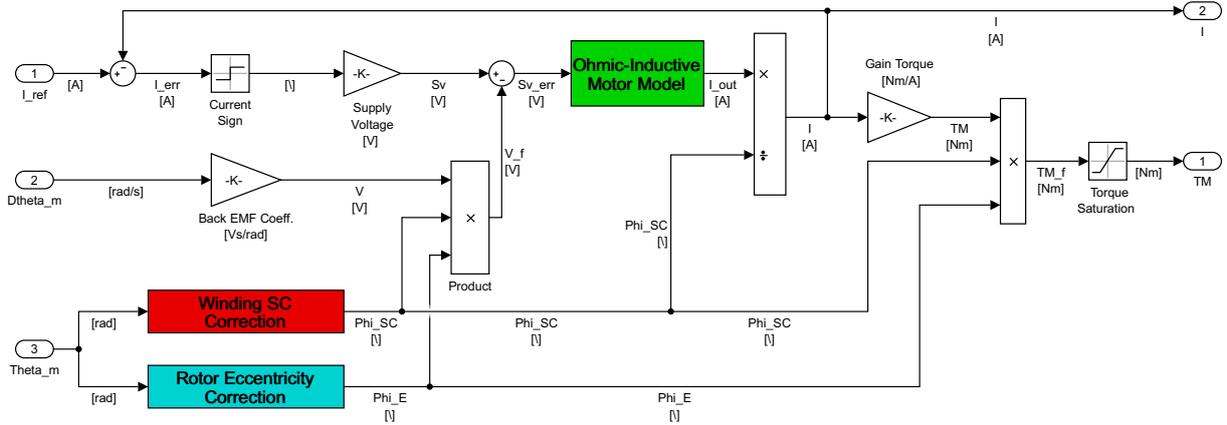


Figure 3.9: Electrical Model box content, with Faults “deployed”

We can see the torque output too needs to be modified by the form functions: that’s because the gain torque, which is basically the same as BEMF coefficient, is affected by the number of working coils and by the air-gap depth, whose contributions are not included into the motor model t.f. modification, as might be thought. In the next sections, we are giving a more in depth look at the correction functions.

3.1.4 Winding SC Correction

The *Winding SC Correction* box is able to provide an “instantaneous” coefficient, proportional to the percentage of working coils $[N_A, N_B, N_C]$ and to the motor angular position.

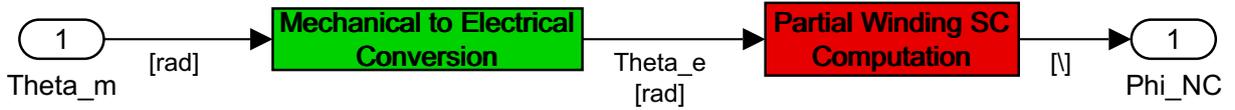


Figure 3.10: Winding SC Correction box content

As first step, a mechanical to electrical motor angular position conversion is needed and achieved by the *Mechanical to Electrical Conversion* box, containing the relation 1.4, while the *Partial Winding SC Computation* contains the actual form function, modeling the eventual short circuits **within the single phase** and having the following feature:

$$k_{f_t} \left\{ N_A [1 + k_{f_s} \cdot \sin^2(\theta_e + \pi)] + N_B \left[1 + k_{f_s} \cdot \sin^2\left(\theta_e + \frac{\pi}{3}\right) \right] + N_C \left[1 + k_{f_s} \cdot \sin^2\left(\theta_e - \frac{\pi}{3}\right) \right] \right\} \quad (3.23)$$

Where, k_{f_s} is the short circuit *single contribution gain* and k_{f_t} is the short circuit *total contribution gain*, which basically are needed for off-set adjusting between the high fidelity and the monitor model, faced into the calibration section 3.2.

Given the fact that a function with the same frequency as the $I_{3_{eq}} = I_q$ is needed and that the percentage of working coil N_i is proportional to the phase current I_i , the structure of equation 3.23, comes from the second of 1.8, indeed:

$$\begin{aligned} I_q &= \left[-\frac{2}{3}I_A + \frac{1}{3}(I_B + I_C) \right] \sin \theta_e + \frac{\sqrt{3}}{3}(I_B - I_C) \cos \theta_e = \\ &= -\frac{2}{3}I_A \cdot \sin \theta_e + \frac{1}{3}I_B(\sin \theta_e + \sqrt{3} \cos \theta_e) + \frac{1}{3}I_C(\sin \theta_e - \sqrt{3} \cos \theta_e) = \\ &= \frac{2}{3} \left[-I_A \cdot \sin \theta_e + I_B \left(\frac{1}{2} \sin \theta_e + \frac{\sqrt{3}}{2} \cos \theta_e \right) + I_C \left(\frac{1}{2} \sin \theta_e - \frac{\sqrt{3}}{2} \cos \theta_e \right) \right] \end{aligned} \quad (3.24)$$

Remembering that:

$$\begin{cases} \frac{\sqrt{3}}{2} = \sin \frac{\pi}{3} \\ \frac{1}{2} = \cos \frac{\pi}{3} \\ \sin(\alpha + \beta) = \sin \alpha \cdot \cos \beta + \cos \alpha \cdot \sin \beta \\ \sin(\alpha - \beta) = \sin \alpha \cdot \cos \beta - \cos \alpha \cdot \sin \beta \\ \sin(\alpha + \pi) = -\sin \alpha \end{cases} \quad (3.25)$$

We obtain:

$$\begin{aligned} I_q &= \frac{2}{3} \left[-I_A \cdot \sin \theta_e + I_B \left(\frac{1}{2} \sin \theta_e + \frac{\sqrt{3}}{2} \cos \theta_e \right) + I_C \left(\frac{1}{2} \sin \theta_e - \frac{\sqrt{3}}{2} \cos \theta_e \right) \right] = \\ &= \frac{2}{3} \left[-I_A \cdot \sin \theta_e + I_B \left(\cos \frac{\pi}{3} \cdot \sin \theta_e + \sin \frac{\pi}{3} \cdot \cos \theta_e \right) + I_C \left(\cos \frac{\pi}{3} \cdot \sin \theta_e - \sin \frac{\pi}{3} \cdot \cos \theta_e \right) \right] = \\ &= \frac{2}{3} \left[I_A \cdot \sin(\theta_e + \pi) + I_B \cdot \sin\left(\theta_e + \frac{\pi}{3}\right) + I_C \cdot \sin\left(\theta_e - \frac{\pi}{3}\right) \right] \propto \\ &\propto \left[N_A \cdot \sin(\theta_e + \pi) + N_B \cdot \sin\left(\theta_e + \frac{\pi}{3}\right) + N_C \cdot \sin\left(\theta_e - \frac{\pi}{3}\right) \right] \end{aligned} \quad (3.26)$$

Next operations take birth from necessity of the form function to be always positive, to have a maximum value equal to one (when $N_A = N_B = N_C = 1$), to have the chance to be calibrated on a given reference model and to be non-dependent by kind (N_A, N_B or N_C) and number of windings (just one, two or all of them) underwent to short circuit. In particular, the fact that a sort of independence and a unitary value for nominal conditions are needed, took us to add 1 to every windings contribution, as follows:

$$\begin{aligned} &\left[N_A \cdot \sin(\theta_e + \pi) + N_B \cdot \sin\left(\theta_e + \frac{\pi}{3}\right) + N_C \cdot \sin\left(\theta_e - \frac{\pi}{3}\right) \right] \rightarrow \\ &\rightarrow N_A [1 + \sin(\theta_e + \pi)] + N_B \left[1 + \sin\left(\theta_e + \frac{\pi}{3}\right) \right] + N_C \left[1 + \sin\left(\theta_e - \frac{\pi}{3}\right) \right] \end{aligned} \quad (3.27)$$

This operation even causes the form function to have a 3 value (for nominal conditions), effect which is going to be compensated by insertion of k_f parameters. Subsequently, during a trial and error process, we had the necessity to increase the form function frequency, fulfilled by square elevating the sinusoidal components:

$$\begin{aligned} &N_A [1 + \sin(\theta_e + \pi)] + N_B \left[1 + \sin\left(\theta_e + \frac{\pi}{3}\right) \right] + N_C \left[1 + \sin\left(\theta_e - \frac{\pi}{3}\right) \right] \rightarrow \\ &\rightarrow N_A [1 + \sin^2(\theta_e + \pi)] + N_B \left[1 + \sin^2\left(\theta_e + \frac{\pi}{3}\right) \right] + N_C \left[1 + \sin^2\left(\theta_e - \frac{\pi}{3}\right) \right] \end{aligned} \quad (3.28)$$

This made the trick for us in terms of frequency, but produced the side effect of rising the function value to 4.5, in such a way the final adding of *single contribution* and *total contribution* gains had to take that into account. Furthermore, the presence of these parameters guaranteed the chance to fit the function output to different reference model configurations, conducting to the final SC form function expression:

$$\Phi_{SC} = k_{f_t} \left\{ N_A [1 + k_{f_s} \cdot \sin^2(\theta_e + \pi)] + N_B \left[1 + k_{f_s} \cdot \sin^2\left(\theta_e + \frac{\pi}{3}\right) \right] + N_C \left[1 + k_{f_s} \cdot \sin^2\left(\theta_e - \frac{\pi}{3}\right) \right] \right\} \quad (3.29)$$

In order to evaluate the nominal k_{f_s} and k_{f_t} values, a “manual fitting” process, with windings conditions set to $N_A = N_B = N_C = 1$, has been conducted and led us to:

$$\begin{cases} k_{f_t NC} = \frac{1}{18} \\ k_{f_s NC} = 9 \end{cases} \quad (3.30)$$

Those being the quantities which guaranteed the minimum (at sight) difference in terms of current output between high fidelity and monitoring models. Indeed, by substituting those values inside the form function expression, we obtain:

$$\Phi_{SC} = \frac{11}{12} \simeq 0.9167 \quad (3.31)$$

No further actions were conducted in order to achieve the unity value, because we knew that a more rigorous method in evaluating these parameters were needed anyway, so that an optimization processes would have taken care of that (see section 3.2).

3.1.5 Rotor Eccentricity Correction

Something very similar to the previous section happens with the *Rotor Eccentricity Correction*, whose content is shown in the figure below:

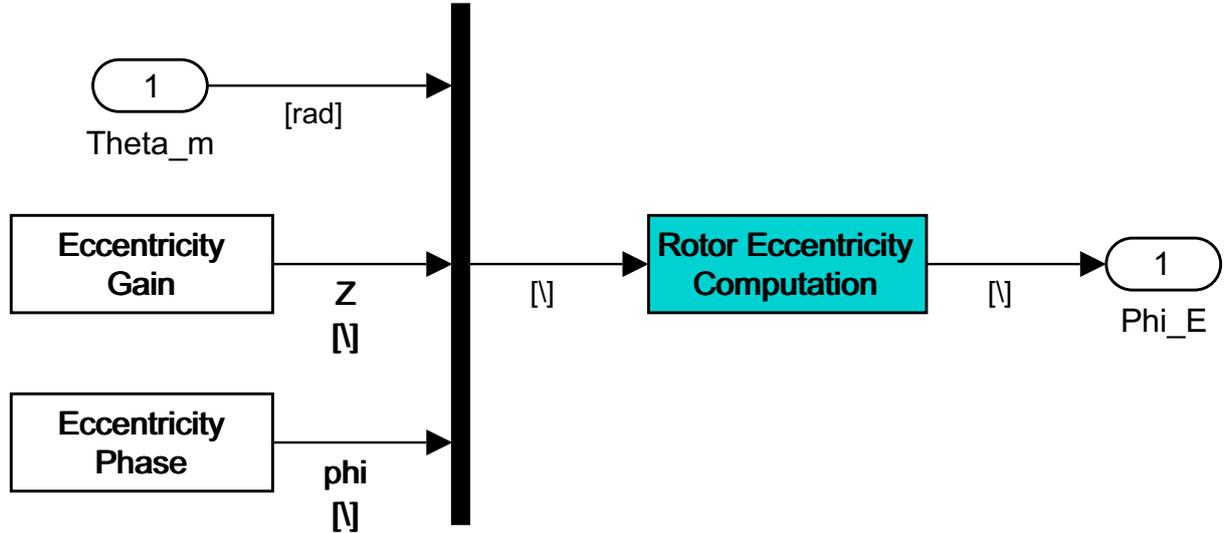


Figure 3.11: Rotor Eccentricity Correction box content

It produces a non-dimensional coefficient Φ_E , affecting the BEMF and the output torque, without causing variations to the current output because of its merely mechanic nature; the way this parameters are influenced by Φ_E , is already reported in equations 3.22. In particular, the *Rotor Eccentricity Computation* aim is representing the **static** misalignment due to the eventual rotor deformed axis, by taking as input a magnitude value (indicating the maximum deformation point) referred as:

$$0 \leq Z \leq 1 \Rightarrow 0 \leq \frac{\Delta}{Z_0} \leq 1 \quad (3.32)$$

Static eccentricity, means that the deformed rotor axis coincides with the rotational one, in such a way the closest and furthest points to the stator, are always in the same angular position. We can see $Z = \frac{\Delta}{Z_0}$ in the equation above, where Δ is the distance between rotational axes in nominal and deformed condition, while Z_0 is the nominal air gap depth[2].

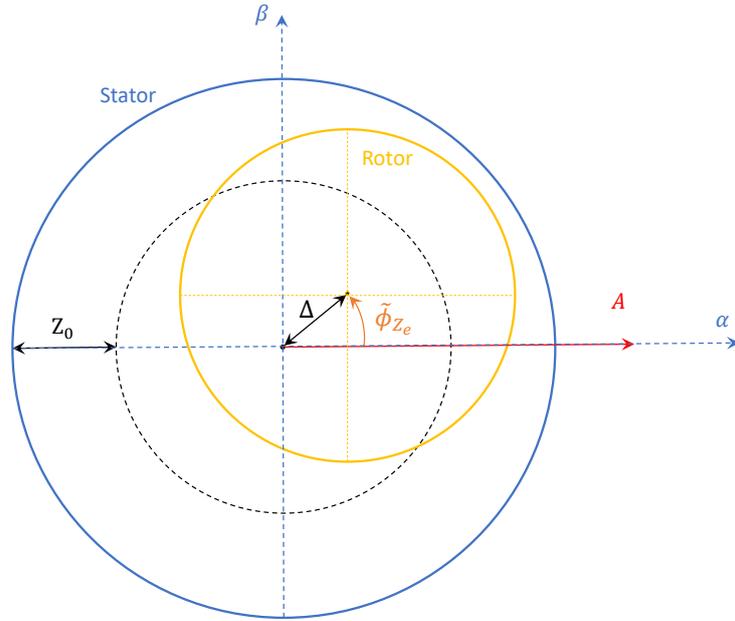


Figure 3.12: Static eccentricity scheme: the dotted circle represents the rotor in nominal conditions, the blue one is the stator, the yellow one is the rotor with a static eccentricity

That's said, it's easy to understand that 0 represents the situation with no axis deformation and 1, the condition which would cause creeping between stator and rotor ($\Delta = Z_0$). A second information, given by the maximum deflection angular position (in respect to the referring system: see figure 3.12 for greater clarity), is stated to:

$$\begin{aligned}
 0 \leq \phi_Z \leq 1 &\Rightarrow 0 \leq \tilde{\phi}_{Z_m} \leq (2\pi)_m \Rightarrow 0 \leq \tilde{\phi}_{Z_e} \leq (4\pi)_e \Rightarrow \\
 &\Rightarrow \phi_Z = \frac{\tilde{\phi}_{Z_m}}{(2\pi)_m} = \frac{\tilde{\phi}_{Z_e}}{(4\pi)_e}
 \end{aligned} \tag{3.33}$$

In this case, 0 indicates alignment with phase A, while 1 represents the interval upper bound of the chosen angular motor position kind (mechanical or electrical). Indeed the subscripts m and e denote respectively mechanical or electrical angle, and ϕ_z is the normalized value of $\tilde{\phi}_Z$ in respect to considered angle.

The remaining input is θ_m , which is converted in electrical angle inside the *Rotor Eccentricity Computation* box, containing the following:

$$\begin{aligned}
 \theta_e &= N_p \cdot \theta_m = 2 \cdot \theta_m \rightarrow \\
 \rightarrow \Phi_E &= 1 - k_{f_E} \cdot Z_E [\cos(\theta_e + \tilde{\phi}_{Z_e})]
 \end{aligned} \tag{3.34}$$

Since Φ_E needs to be ineffective when the input parameters are null, we find $k_{f_E} \cdot Z_E [\cos(\theta_e + \tilde{\phi}_{Z_e})]$ subtraction to 1, while for the SC computation we had to add the related contribution to 1. The parameter k_{f_E} is homologous to k_{f_s} , and is needed in order to have the chance to calibrate the monitoring model upon the high fidelity one. As starting value has been chosen the one well working for a trapezoidal wave form motor, as stated in the article (P. C. Berri, M. D. L. Dalla Vedova, P. Maggiore)[3], and has been set to:

$$k_{f_E} = 0.42 \tag{3.35}$$

Which is going to be optimized as much as the other failure parameters (see section 3.3).

3.1.6 Output

The orange box *Output*, represented in figure 3.1, only contains low pass filters for both I and motor torque TM , and “saving-to-workspace” feature for the latter quantity.

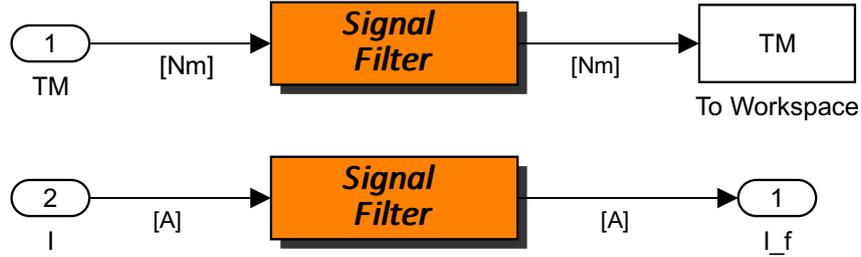


Figure 3.13: Output box content

The filters represented in the figure above are exactly the same between each other and as those used in the high fidelity model; actually their functionality here it's not related with the signal filtering necessity, but they are needed in order to introduce the same delay the HF model is affected by, due to their presence. The t.f. used for this purpose is the following:

$$\frac{1}{(\tau_f \cdot s + 1)^3} = \frac{1}{[(5 \cdot 10^{-5})s + 1]^3} \quad (3.36)$$

Which is a third order t.f., guaranteeing a strong signal attenuation (-60 dB/decade) starting from the cutoff frequency, given by:

$$f_c = \frac{1}{\tau_f} = \frac{1}{5 \cdot 10^{-5}} = \frac{10^5}{5} = 20 \text{ [kHz]} \quad (3.37)$$

3.1.7 Motor-Transmission Dynamical Model

A second order system has been used in the *Motor-Transmission Dynamical Model* (see figure 3.1), represented in the figure below:

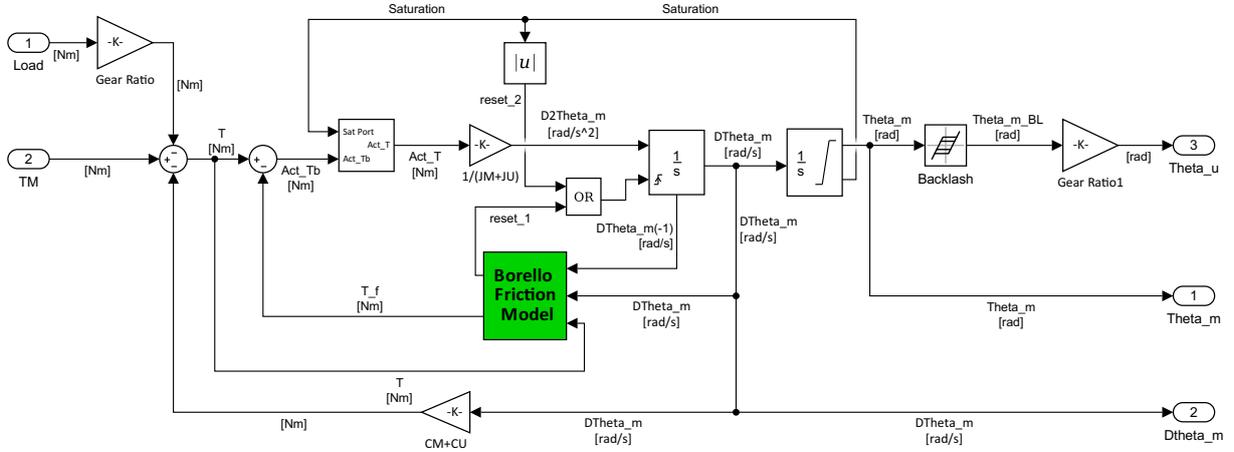


Figure 3.14: Motor-Transmission Dynamical Model box content

The inputs are given by *Load* and motor torque *TM* coming from the *Electrical Model* (section 3.1.3); the first one is multiplied by the *Gear Ratio*:

$$\tau = \frac{1}{500} \quad (3.38)$$

This value can be obtained thanks to what an *epicyclic reducer* could be, but this goes beyond the present work aims. The first operation box encountered downstream of τ is the algebraic sum where we can see *TM* subtracted of the load contribution and of viscous friction factor, given by:

$$(C_m + C_u) \cdot \dot{\theta}_m = C_{tot} \cdot \dot{\theta}_m \quad (3.39)$$

Where, the dumping component of the system is equal to:

$$C_{tot} = C_m + C_u = 5.172 \cdot 10^{-5} \left[\frac{\text{Nm} \cdot \text{s}}{\text{rad}} \right] \quad (3.40)$$

With C_m and C_u being, respectively, the motor and utilizer contribution. This way we have:

$$TM [Nm] - \tau \cdot Load [Nm] - C_{tot} \cdot \dot{\theta}_m \left[\frac{Nm \cdot s}{rad} \right] \cdot \left[\frac{rad}{s} \right] = T [Nm] \quad (3.41)$$

The next step is the *active torque* calculation Act_Tb , where “b” means “before” the withe box producing Act_T as output; we have, indeed:

$$Act_Tb = T - T_f [Nm] \quad (3.42)$$

With T_f being the *Coulomb friction* contribution to the motor torque, calculated by *Borello Friction Model*[8] (see section 3.5), which basically chooses between static and dynamic friction coefficients by evaluating whether the motor angular speed changed sign or not, through a detection point. When this happens, the green box sends a reset signal to the first integrator (*reset_1*), in such a way it can set the speed output to zero; in order to fulfill its task, the friction model receives as inputs two consecutive motor speed steps ($DTheta_m(-1)$ and $DTheta_m$) and the torque T , which is needed for understanding whether we are in aiding or opposing conditions. The Act_Tb enters a switch box which verifies whether mechanical end-stops conditions occurred or not and states which output we are going to have, respectively, between zero and the input active torque. Indeed, this box needs the saturation port *Saturation* signal, coming from the integrator relative to θ_m , which is going to produce the values:

$$\begin{cases} +1 & \text{Upper mechanical end - stop} \\ 0 & \text{Not limited} \\ -1 & \text{Lower mechanical end - stop} \end{cases} \quad (3.43)$$

Thus, if Act_Tb has the same sign as *Saturation*, then the output is going to be null, while in all the other cases, it states $Act_T = Act_Tb$. Notice that, if *Saturation* is other than zero, a second kind of reset signal *reset_2* is sent to the first integrator, allowing it to produce a null speed value for the next integration step. The Act_T encounters the gain:

$$\frac{1}{J_m + J_u} = \frac{1}{J_{tot}} = \frac{1}{2.5^{-5}} \left[\frac{rad}{kg \cdot m^2} \right] \quad (3.44)$$

With J_{tot} being the system total moment of inertia, given by the sum of the motor (J_m) and utilizer (J_u) ones. In such a way, we obtain the motor angular acceleration $D2Theta_m$, applying the fundamental law of rotational dynamics, given by:

$$T = J \cdot \ddot{\theta}_m \Rightarrow \ddot{\theta}_m = \frac{T}{J} \quad (3.45)$$

The following step is the first integration, providing the angular speed $DTheta_m$ and whose box has a *reset signal input port* and a *state port* about which has already been discussed. The second integration box is limited, as could be understood from what was said above, and these limits represent the mechanical end-stops in terms of motor angular position; its output is, indeed, motor mechanical angle $Theta_m$. As last “non-linearity” we have a *Backlash* box, coarsely modeling the reducer mechanical clearance as a simple dead band. For sure this solution will need an upgrade in future developments, but since that’s a conservative assumption, it’s been considered to be enough precise for this application. Before obtaining the utilizer angular position $Theta_u$, the variable $Theta_m_BL$ goes through the reducer gear ratio τ , which decreases by 500 times the motor-shaft angular speed and position. Finally, the dynamical model, has provided the output quantities needed: θ_m , $\dot{\theta}_m$ and $\ddot{\theta}_m$. Just like already done in the previous charter for HF model, here follows the highest level of the LF one, whose sight helps in understanding the involved data flow, by giving an overall picture of the system.

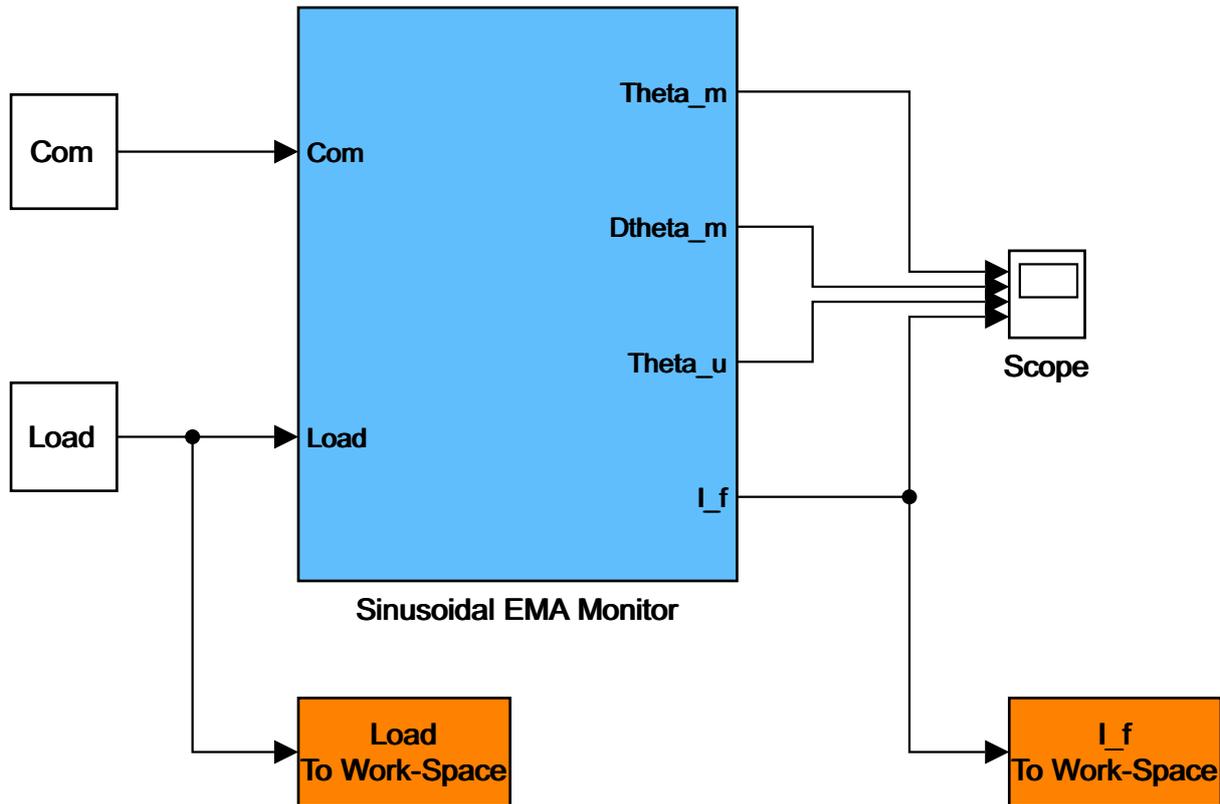


Figure 3.15: Monitoring model overall sight

As we can see, *Com* and *Load* boxes are the inputs, the orange ones save load torque and filtered current to the workspace, which, along with the other output parameters coming from the mechanical model, are plotted through *Scope* box. In the next sections, we are going to *calibrate* the model upon the HF one for nominal conditions, as much as *optimize* the failures parameters in degraded circumstances. Since the kind of faults, aimed to be subjects of the early detection in question, are characterized by medium-to-long performance time frames (we are talking about tens of hours), the “before flight” testing it’s been considered to be a proper contest to run the diagnostic checks for that specific failures. In these conditions, the control surfaces are not invested by the aerodynamic flow, neither subjected to load factors other than 1, making reasonable the choice to conduce calibration and optimization by setting $Load = 0$. Not the same for command signal which, indeed, has been chosen to be a *chirp* kind: sine curve starting with $0.005 \left[\frac{rad}{s} \right]$ amplitude and null frequency, the latter of which linearly increases up to a $15 [Hz]$ maximum value after $0.5 [s]$.

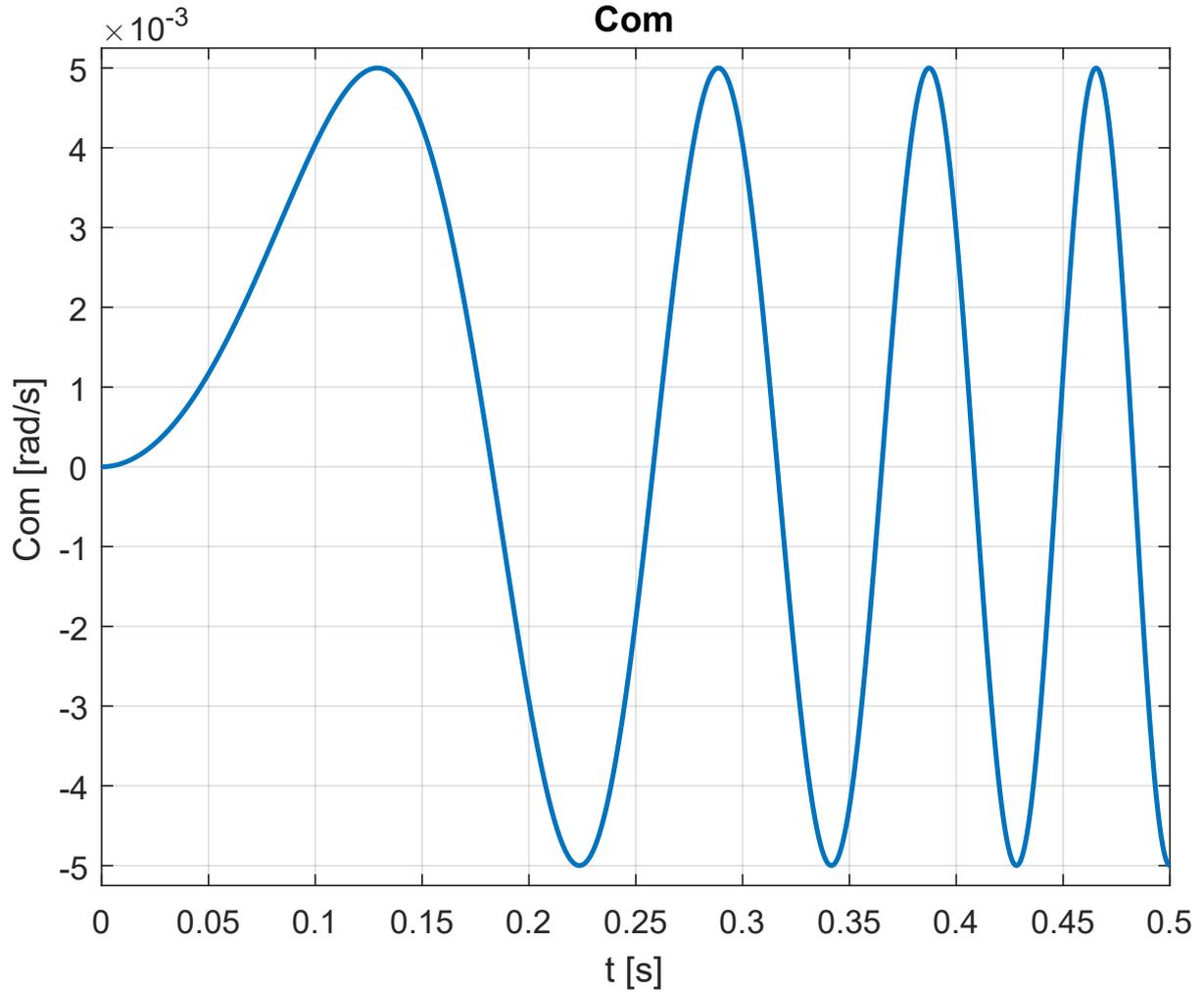


Figure 3.16: Chirp command trend

3.2 Model Calibration

Clarified the command and load conditions at which we're are going to test our monitor model, let's remember that we're going to focus on the current output, calibrating some parameter in order to have the best fitting possible to the HF model equivalent current (see 2.2), because the effects of faults we're examining are visible only there; that's why the response in terms of motor and utilizer angular position, compared to the given command, has not been studied even though, of course, the model provides these informations too.

The reasons why a calibration process is needed are multiples and they all deal with the fact that an extreme exemplification has been made in order to obtain a lighter and faster model, without considering Φ_{SC} in nominal conditions is not exactly unitary (see section 3.1.4). The parameters affected by having completely taken off the three-phase structure are BEMF coefficient k_{BEMF} and the gain torque GT , where the first one strictly deals with the current wave form, while the second one directly affects the motor torque which, through its influence on the motor angular speed calculation (see section 3.1.7), again has impact on the BEMF coefficient and on the current. As starting values, has been assigned, respectively:

$$\begin{cases} k_{BEMF} = \frac{0.0752}{3} \left[\frac{Nm}{A} \right] \\ GT = \frac{0.0752}{2} \left[\frac{Nm}{A} \right] \end{cases} \quad (3.46)$$

From which we can see the parameters are related by the equation:

$$GT = \frac{3}{2} \cdot k_{BEMF} \quad (3.47)$$

The reasons of this relationship is explained by the fact that, if the single-phase transformation decreases the three-phase current by a $\frac{2}{3}$ factor and its taken into account by construction, not the same happens for the torque in the HF model. In order to have a comparable behavior, indeed, we need to multiply the gain torque of the LF model by a $\frac{3}{2}$ factor, which compensates the just explained effect through its influence on the BEMF contribution to the voltage. The process called *calibration*, here, is intended to be an optimization trial whose aim is minimization of the *Mean Squared Error (MSE)* between currents, in **nominal conditions**. This operation has been fulfilled by employment of Matlab *Optimization tool* running a *Genetic Algorithm (GA)* able to generate a certain number of k_{BEMF} and GT casual combinations, to evaluate the *MSE* at each iteration, and to end the process when some *exit criteria* are satisfied.

3.2.1 Optimization Toolbox

Global Optimization Toolbox[4] provides functions searching for global solutions to problems that contain multiple maximum or minimum points. Toolbox solvers include *surrogate*, *pattern search*, *genetic algorithm*, *particle swarm*, *simulated annealing*, *multi-start*, and *global search*. Each of these solvers can be used for optimization problems where the *objective* or *constraint function* is continuous, discontinuous, stochastic, does not possess derivatives and includes simulations or black-box functions. For problems with multiple objectives, a *Pareto front*¹ can be identified by using GA or pattern search solvers. Furthermore, solver effectiveness can be improved by adjusting options and, for applicable solvers, customizing creation, update, and search functions. Custom data types can be used with the GA and simulated annealing solvers, able to represent problems not easily expressed with standard data types. Finally there is even an hybrid function option allowing the improvement of a solution by applying a second solver after the first. In our case, since our problem is a multiple variables one, with unknown evolution pattern, the more effective solution has been considered to be the GA utilization.

3.2.2 Genetic Algorithms

The *GA*[5], by exploiting a natural selection process that mimics biological evolution, is a method for solving both constrained and unconstrained optimization problems. The algorithm repeatedly modifies a population of individual solutions and, at each step, the GA randomly selects individuals from the current population in order to use them as “parents” to produce the “children” for next generation. Over successive generations, the population “evolves” toward an *optimal solution*. The GA can be applied in order to solve problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear. For a better understanding of the process a GA uses in order to generate *children* for next stages, here the rules followed:

- 1) **Selection rules:** select the parents contributing to the population at the next generation;
- 2) **Crossover rules:** two parents combination in order to form children for the next generation;
- 3) **Mutation rules:** random changes application to individual parents before forming children;

The (two) main differences intervening between GA and a classical, derivative-based, optimization algorithm (that we are going to call *CA*) are explained below.

- **Point generation**

- **Classical Algorithm:** a CA generates a single point at each iteration and, if the problem is well-posed, the sequence of points approaches an optimal solution;
- **Genetic Algorithm:** a GA produces a population of points at each iteration, the best among which, approaches an optimal solution.

- **Point selection for next iteration**

- **Classical Algorithm:** a CA selects the next point in the sequence by a deterministic computation;

¹Pareto front: a set of non-dominated solutions, being chosen as optimal, if no objective can be improved without sacrificing at least one other objective.

- **Genetic Algorithm:** a GA chooses the next population by computation using random number generators.

Let's see in detail the steps a GA does, during its utilization[6]: first of all, creates a random initial population, scores each member of it by computing its fitness value (*raw fitness scores*) and scales these latter to convert them into a more usable range of numbers, by creating the *expectation values*.

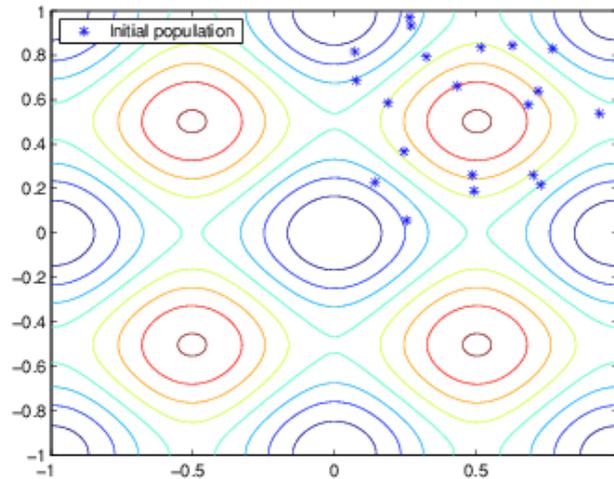


Figure 3.17: Example of initial population created by a GA: note that all the individuals in the initial population lie in the upper-right quadrant of the picture, that is, their coordinates lie between 0 and 1

After that, selects parents, based on their expectation, and classifies them as *elite* individuals, which are directly passed to the next population, if they have the best fitness scores. Anyway, children are produced either by making random changes to a single parent (*mutation*) or by combining the vector entries of a parents pair (*crossover*). The following figure, shows the three types of children generated by the initial population of the figure above:

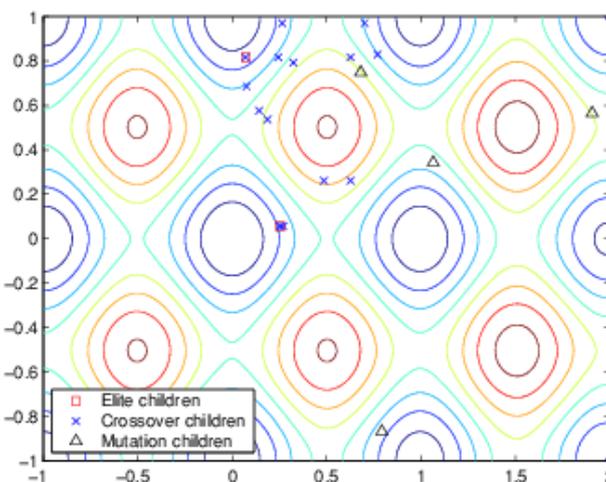


Figure 3.18: First generation children, classified by type of their generation

For mutation children and unconstrained problems, the algorithm adds by default a random vector from a *Gaussian distribution* to the parent, while, for bounded or linearly constrained problems, the child remains feasible. Instead, regarding the crossover children, at each coordinate of the child vector, the default function randomly selects an entry, or *gene*, at the same coordinate from one of the two parents and assigns it to the child. When dealing with problems characterized by linear constraints, the crossover function creates the child as a random weighted average of the parents. Following these tasks generation

by generation, the individuals in the population of our example get closer together and approach the minimum point $([0,0])$; this behavior is well shown by following figures, representing respectively the population at 60th, 80th, 95th and 100th generations.

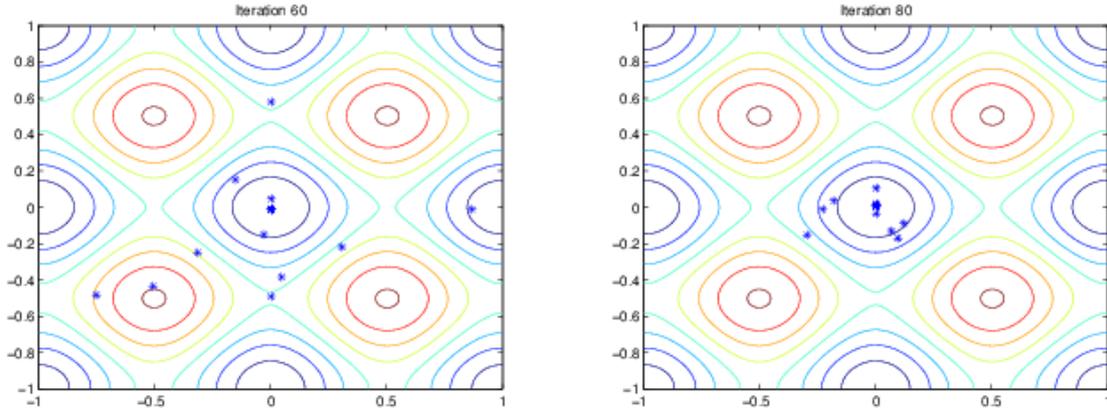


Figure 3.19: 60th and 80th generations population

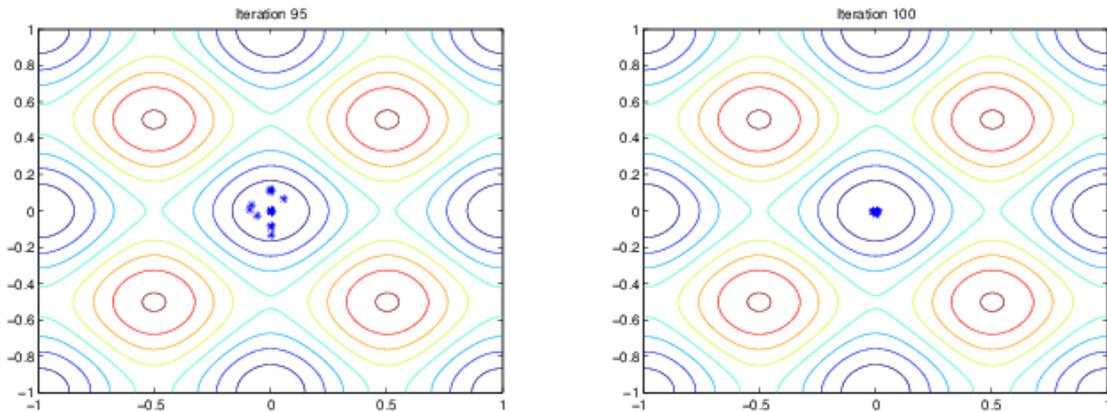


Figure 3.20: 95th and 100th generations population

There are several conditions the GA verifies at each iteration, in order to understand whether it has to stop or not, and are listed below.

- **Generations:** when the number of generations reaches the maximum set value ($100 \cdot Par_{Number}$ by default, which means 200 in our case);
- **Time limit:** if the maximum chosen time is reached (infinite by default and for our application);
- **Fitness limit:** whether the fitness function reaches the minimum imposed value (-infinite by default and for us);
- **Stall generations:** when the fitness function average relative value change is lower then **Function tolerance** when the number of generation takes over the set value (50 by default and for this application);
- **Stall time limit:** if the objective function value doesn't improve for a certain amount of time (infinite by default, no changes);
- **Stall test:** here the kind of test to be run for the *stall generations* conditions verification can be set. The options are average change or geometric weighted, where in case of the latter choice, the weighting function is set to $\frac{1}{2 \cdot n}$, whit n the number of generations prior to the current (average value our choice);
- **Function tolerance:** when the average relative changing of the objective function doesn't exceed the set value (10^{-6} by default, set to 10^{-9} by us);

- **Constraint tolerance:** not used as stopping criterion, but to determine the feasibility with respect to nonlinear constraints. (10^{-3} by default, no changes);

The user has the faculty to change each of these parameters, knowing that GA stops as soon as anyone of these conditions is met. Notice that, if the algorithm stops due to one of these constraints, different by *Stall generations*, *Fitness limit* or *Function tolerance*, there are good possibilities the problem has not been well-posed.

3.2.3 Monitoring Model Current Response Analysis

Clarified the functioning principles of a GA, let's focus on our application and examine the current response, by both monitoring and HF models, before any optimization process is run, to the command presented in figure 3.16:

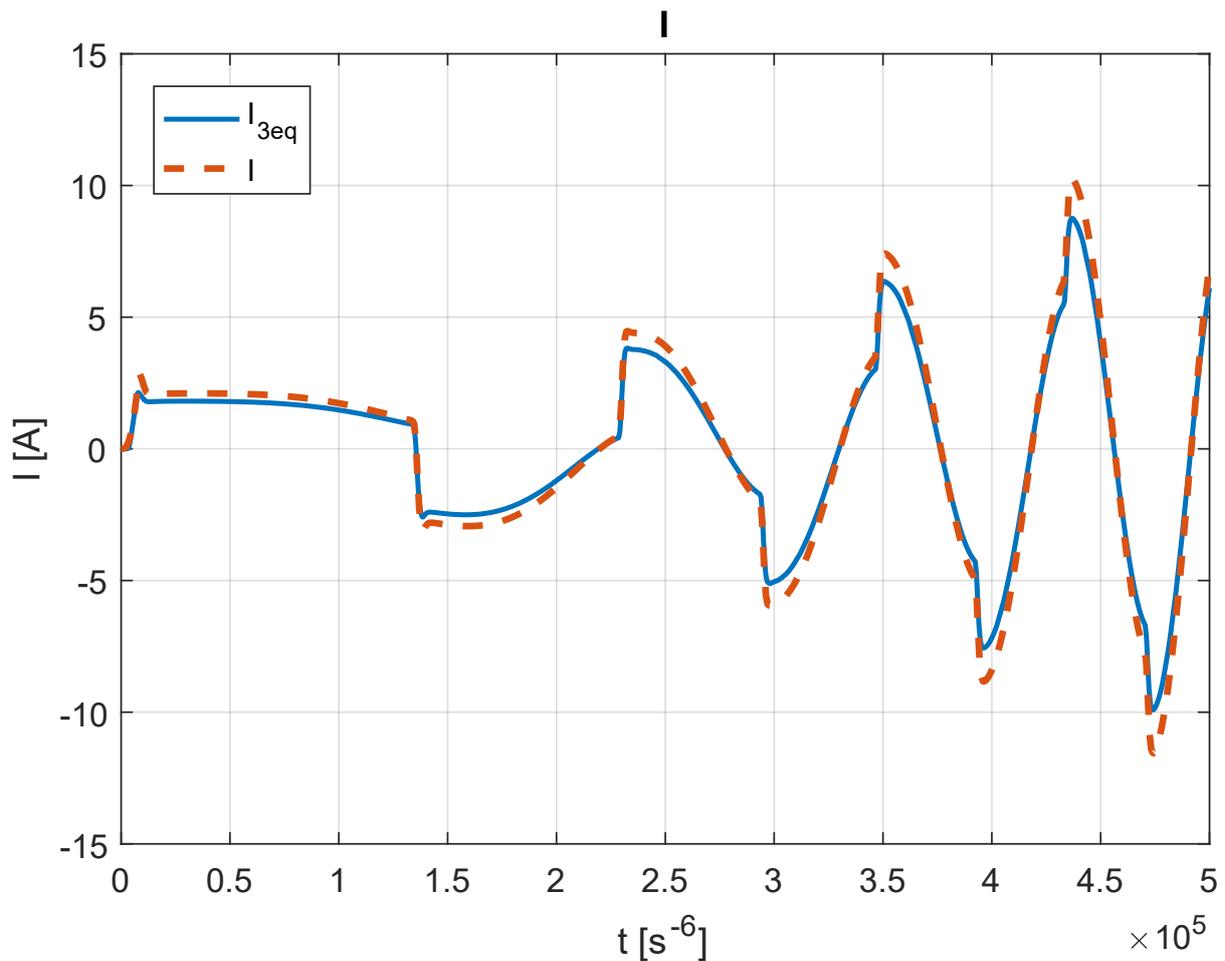


Figure 3.21: HF and LF current response to a chirp command, with no load

We can see the amplitude of both responses increasing as the command frequency grows, even if the latter magnitude remains the same along the whole actuation. That's attributed to inertial forces increasing as much as the frequency does, indeed, when faster direction inversions are required, the system undergoes to greater acceleration, traducing in greater actuation speeds and greater torque required when the next direction changing is imposed. The current is directly proportional to the motor torque, so it grows and decreases in the same way the latter does. Furthermore can be noticed the horizontal stretches gradually become more steep, as much as the required actuation speed increases and the same does position error, till they almost become vertical lines. One last thing to talk about are the real vertical stretches, representing the conditions when the static friction needs to be overcome: the command has changed direction, the system feels it (through the position error evaluation), and reacts with its own dynamics, the actual position speed needs to be inverted, so that at a certain point the actuator has to stop; when this happens, the kind of friction torque to be faced is the static one, which is bigger than the dynamic friction and a sudden current gain is needed in order to start the system up again, producing

those kind of jumps in the current trend.

All this said, here follows the monitor model current response, along with position command θ_{com} , reference current I_{refLF} , utilizer position θ_u , position error θ_{err} and actuation speed $\dot{\theta}_u$:

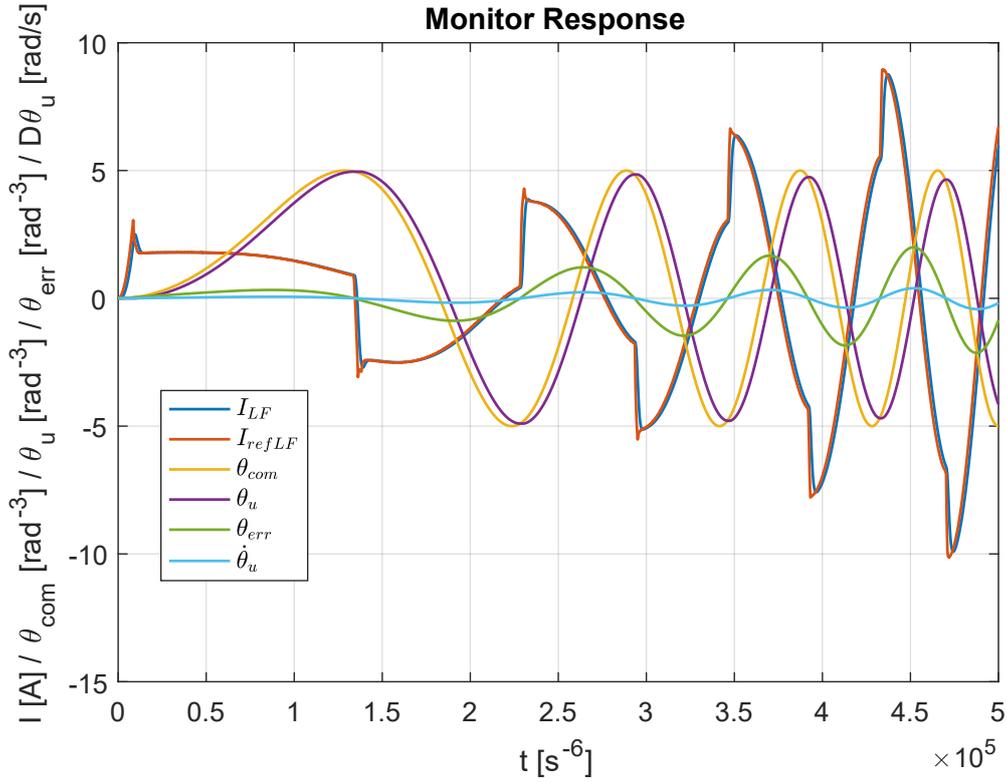


Figure 3.22: Monitoring system response to a chirp command with no load

Let's examine what happens at the very first stages of the actuation:

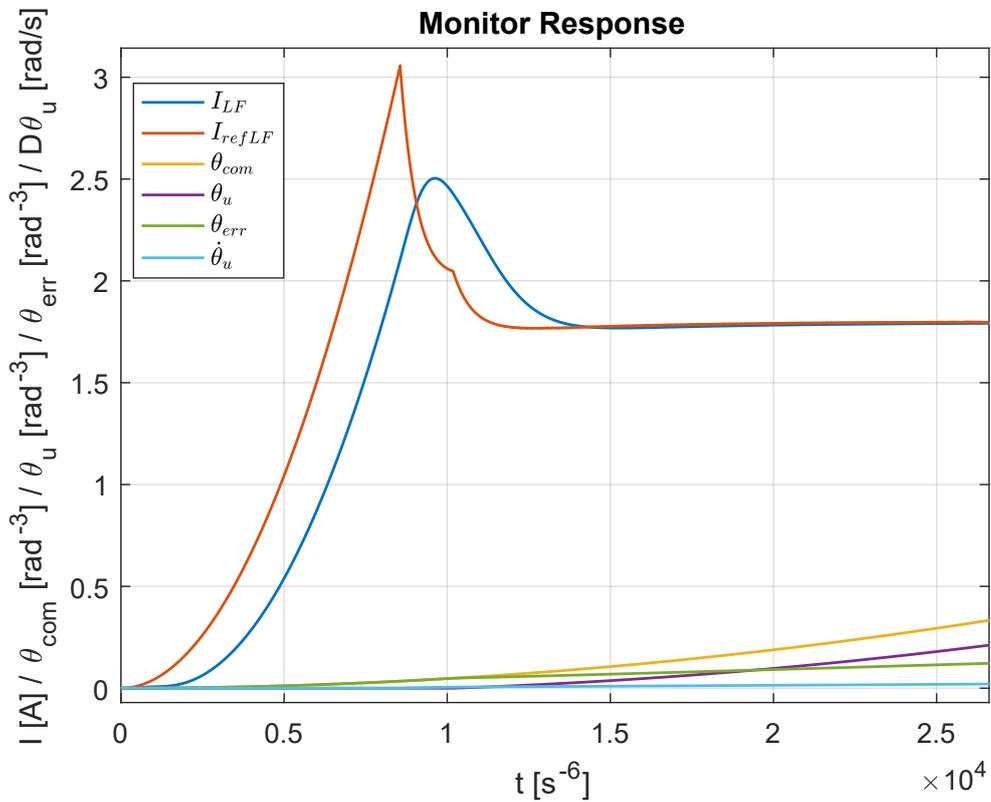


Figure 3.23: First stages of monitoring system response to a chirp command with no load

We can see, the kind of current trend we have, is similar to one typical of the second order systems (even though that's not the case, since we're not studying a variable obtained by double integration of something proportional to the command): an horizontal tangent start and a sort of overshoot before the curves horizontal assessment. Starting from the origin, the position command becomes other than zero, but the system is still stuck because it has to overcome the static friction, so that the reference current grows continuously until the actuation speed differs from zero; when this happens, the error on reference speed decreases and thus the reference current too, producing the cusp visible in the figure above. In the meanwhile, the blue curve follows the red one with the delay characteristic of an RL circuit ($\tau_e = \frac{R}{L} \simeq 0.34 [ms]$) and smooths the cusp thanks to the inductive effect of the motor model. The second point in the red curve, indicates the moment when the utilizer position becomes other than zero, and causes another reference current decreasing due to its positive contribution to the position error. This behavior it's not visible in the blue line, because it's been dumped by the motor RL effect, but let's see better what happens to the speed and position signals:

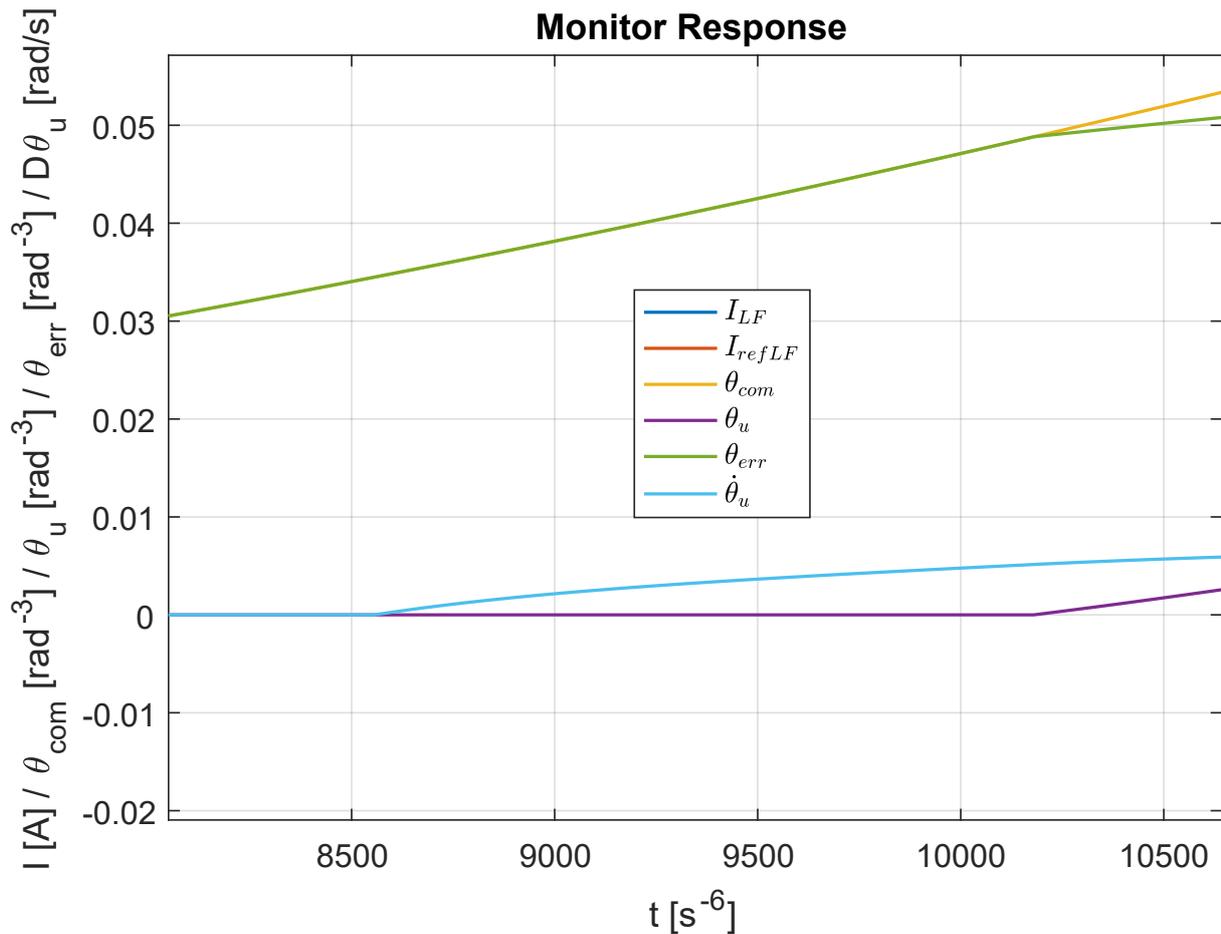


Figure 3.24: Detail on speed and position at first stages of monitoring system response to a chirp command with no load

We sudden notice that the utilizer speed becomes positive, while it's position is still null; of course that's not possible in reality and this happens because of both the backlash effect and the way $\dot{\theta}_u$ has been calculated: the first one imposes to the position to be stuck, but does nothing to the motor speed, from which the utilizer one has been obtained, multiplying it by the gear ratio, and causes this kind of mismatching in certain situations. Indeed, being the backlash downstream of the motor, the fact that rotor moves, but the utilizer does not, is perfectly coherent and, in order to have the real utilizer speed, a model improvement would be required; the correct behavior would have been to have the utilizer speed stuck to zero, as long as the position is, but considered its marginal influence on the model response, this implementation would need an undue complexity enhancement, considered the present application, so much to leave the things the way they are.

Another conduct worthing to be examined is that showing up when the first command direction changing occurs: we can notice the reference current suddenly decreases, sign that the utilizer has stopped during

the direction changing and needs to be restarted towards the opposite way, as the friction force is beaten. The noticeable thing is that the actuation stops before the position error goes through zero, and is clearly visible in the following figure:

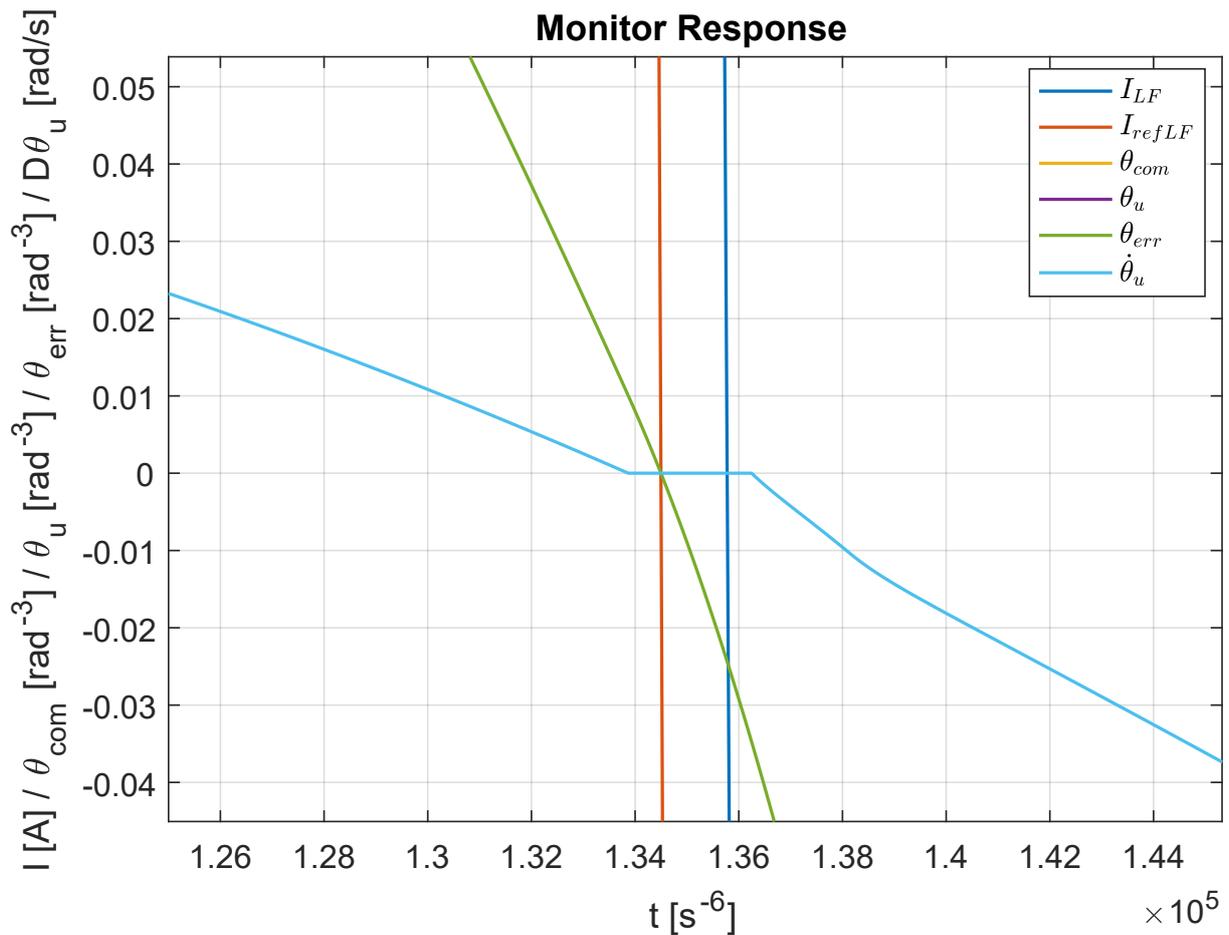


Figure 3.25: Detail on speed and error position at first command inversion

The cyan curve settles to zero, before the position error does, but, this time, everything is correct: the position error drops below the threshold ensuring a torque able to overcome the dynamic friction, before the error becomes null; this provokes the reset signal sending to the acceleration integration box, thus causing the speed to stay null and the position to remain the same, until the error grows again and produces a motor torque able to overcome the static friction one.

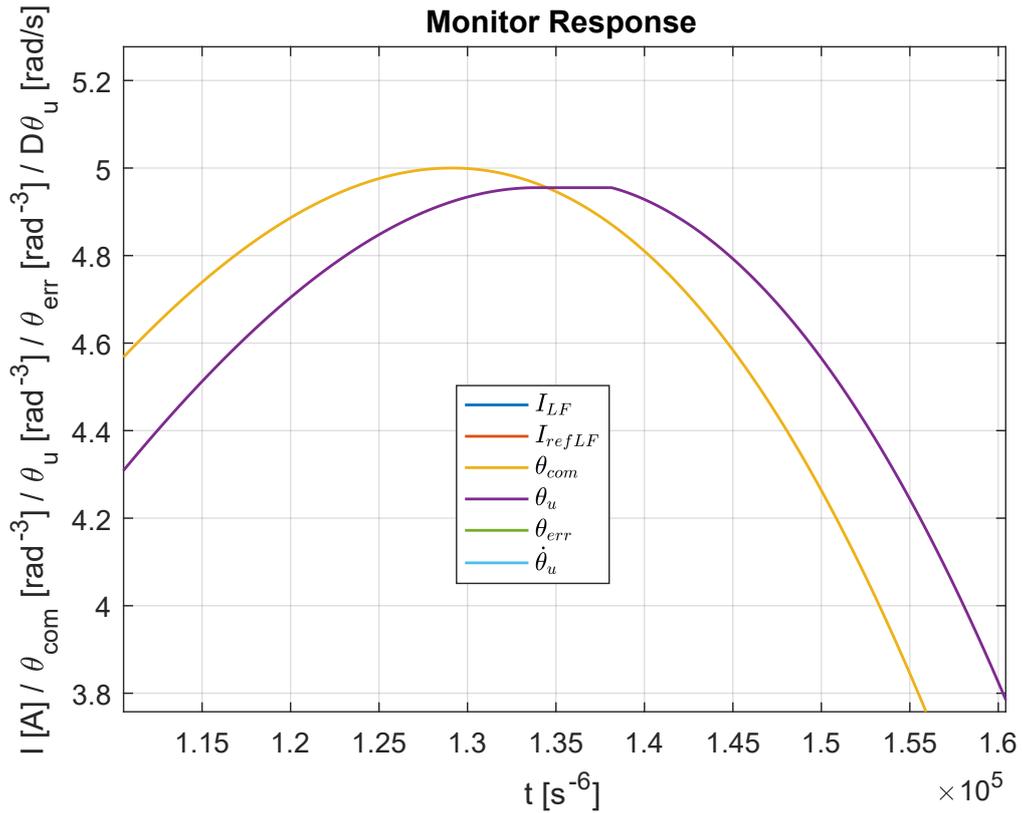


Figure 3.26: Detail on positions at first command inversion

Again, when the latter condition is reached, the system speed becomes negative, but the position doesn't change until the dead band given by the backlash is overtaken, thus causing in the reference current the same behavior had before: a double cusp, as shown in the figure below.

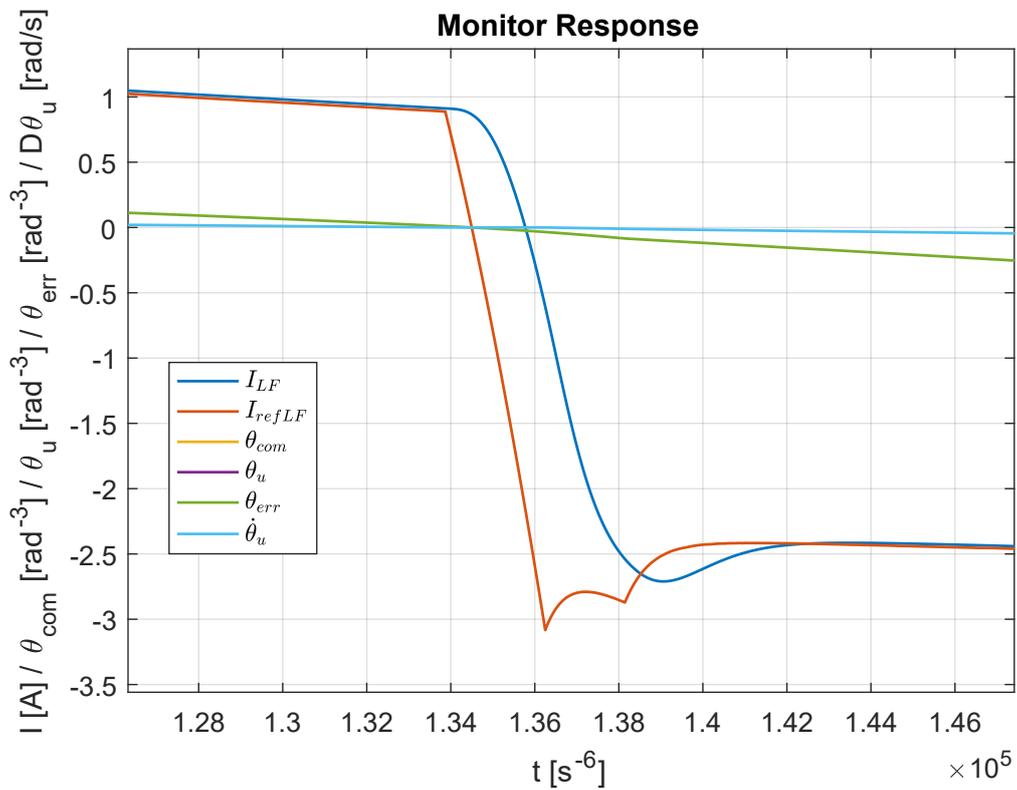


Figure 3.27: Detail on the system response first command inversion

However, once more the current doesn't feel any of these jumps and is able to smooth everything out again. Looking at figure 3.22, it's easy to notice how, on the next command direction changes, the integrative branch can't foresee anymore this behavior, due to the increased actuation speed, indeed, the utilizer always stops after the position error has changed its sign. Furthermore, it can be noticed how the reference current cusps get smoothed out too, as the actuation frequency increases; this is due to the fact that backlash dead band is always faster overtaken, allowing an always smaller amount of time to the reference current to feel it (and thus to react to it), until it simply doesn't show up anymore in the current figure scale.

Looking back at figure 3.21, we can see how the current trends are perfectly comparable, except for an offset in terms of magnitude, that we are going to compensate as best as we can, using the Matlab GA, about which has been widely discussed in the section 3.2.2. As already mentioned, the parameters we are going to change are BEMF coefficient and motor gain torque of the motor model, giving them the possibility to change up to $\pm 20\%$ of the initial values (reported in the equation 3.46). The second thing to do is the choice of the *objective function*, which is going to be the value GA will try to minimize; the current *Mean Squared Error (MSE)* has been chosen and has the following expression:

$$MSE = \frac{\sum_{i=1}^n (I_{3eq_i} - I_i)^2}{n} [A^2] \quad (3.48)$$

I_{3eq_i} and I_i are, respectively, the HF and LF current components at each integration step, while n is the samplings number. Before the optimization process the MSE obtained has been:

$$MSE_{bef} = 0.157 [A^2] \quad (3.49)$$

While, after the GA run, the objective function has decreased of about one *Order Of Magnitude (O.D.M.)* and assessed to:

$$MSE_{opt} = 0.023 [A^2] \quad (3.50)$$

Indeed, the figure comparing currents after the optimization process is the following:

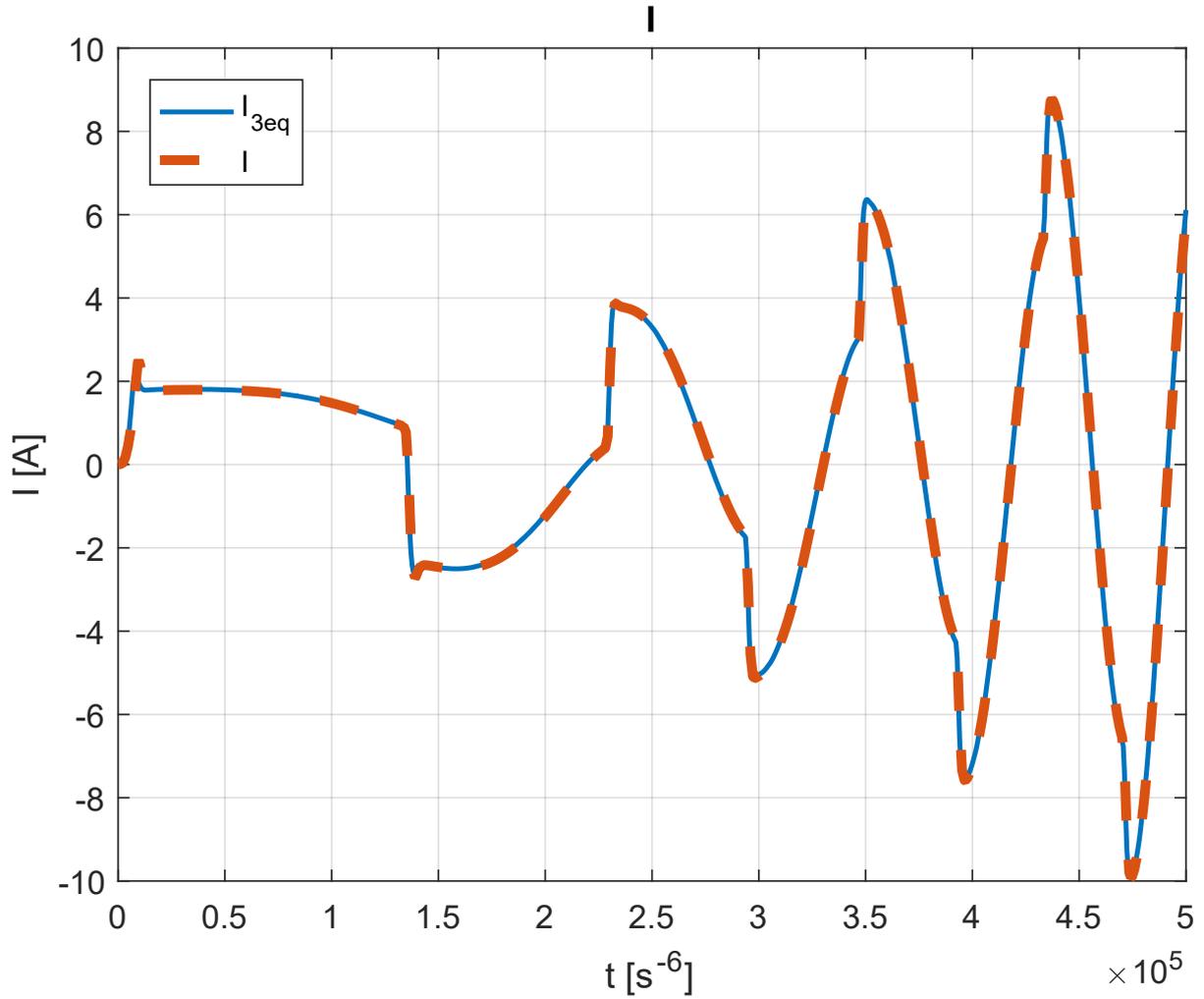


Figure 3.28: HF and LF current response to a chirp command, with no load, after calibration

Where we can see an almost perfect matching between the curves; here follows the table with data recap for before and after calibration situations:

	k_{BEMF}	GT	MSE
Before	0.0251	0.0376	0.157
After	0.0210	0.0392	0.023

Table 3.1: Before/After calibration data summary

3.3 Model Optimization

After being able to match the current trends for nominal condition, through what we called the *calibration* process just described, the same has to be done in degraded occurrences, when the failures form function strongly influence the monitoring system response. In order to distinguish this process from the other one, it will be referred to as *optimization* procedure, even though it will be accomplished by running the same GA, but for different parameters and objective function. The reason why a such process is needed, is to be sure the LF current is able to fit the HF one, with different faults kind and entity and, hopefully, to extrapolate a law relating these latter with failures parameters (whose nominal values are):

$$\begin{cases} k_{f_s} = 9 \\ k_{f_s} = \frac{1}{18} = 0.0556 \\ k_{f_E} = 0.42 \end{cases} \quad (3.51)$$

In order to do that, several optimizations has been run, first by applying different greatness to a single kind of failure and, after, by producing random degraded situations, collecting, of course, parameters and objective function values. The latter has changed in respect to the one used for calibration process, indeed, with the first tests results, we realized it was needed a function able to calculate the discrepancy between curves towards each point *normal direction*. That's because, whether phase displacement occurs, the mean squared error, calculates the difference between currents along the vertical direction (at the same integration step, which means same horizontal axis value), and the GA minimize that discrepancy, focusing its attention to the phase difference. The following figure clarifies in what kind of conditions this detail makes difference:

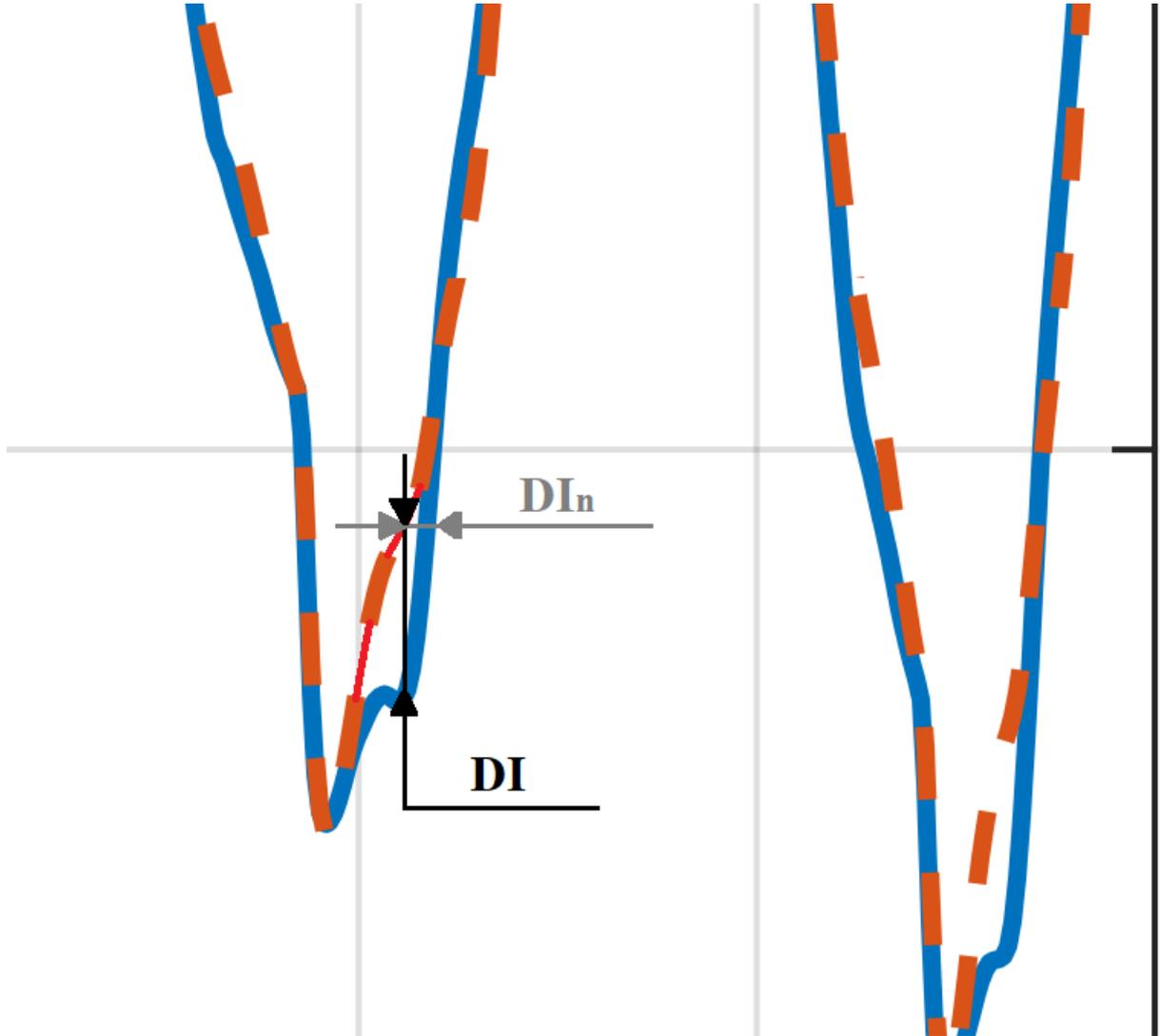


Figure 3.29: Example of a situation, where utilization of normal error makes difference

The *vectorial total mean squared error*[7], instead, by taking into account the currents difference along the normal direction to the reference trend, focuses the target more on the minimization of magnitude than on the phase displacement, which has been considered to be the best solution for this application. In figure 3.29, DI_n stands for *delta i normal*, while DI means *delta i*. Furthermore, another modify has been conducted, is the error normalization in respect to the conditions before the optimization, thus producing the final objective function form given by:

$$MSE_f = \frac{\sum_{i=1}^n \left(\frac{I_{3eq_i} - I_i}{(I_{3eq_i} - I_i)_0 + 1} \right)^2}{n} [A^2] \quad (3.52)$$

Where:

$$(I_{3eq_i} - I_i)'_0 = \frac{d(I_{3eq_i} - I_i)}{dt} \cdot \frac{1}{RMS_0} ['] \quad (3.53)$$

And:

$$RMS_0 = \sqrt{\sum_{i=1}^n \left(\frac{dI_{3eq_i0}}{dt} \right)^2} \left[\frac{A}{s} \right] \quad (3.54)$$

Being root mean square of the equivalent current derivate, calculated before optimization. The contribution of equation 3.52, responsible of the normal calculation, is the squared elevated element denominator, coming from geometrical considerations and leading to the form $1 + 3.53$. Here comes the latter contribution necessity to be normalized in respect to something able to give the same weight to time and amplitude discrepancy, resulted to be the RMS_0 (for further explanation see *P. C. Berri*, [2]).

3.3.1 Single Winding Short Circuit

All of this clarified, the first optimization iterations has dealt with the winding A partial SC, by setting the following working coil percentages: 75%, 50%, 39%25%, 23%, where 39 and 23 values were randomly generated. Here we report the case that produced the best improvement throughout the optimization, allowing to see better the difference in terms of current trend, which is the last one:

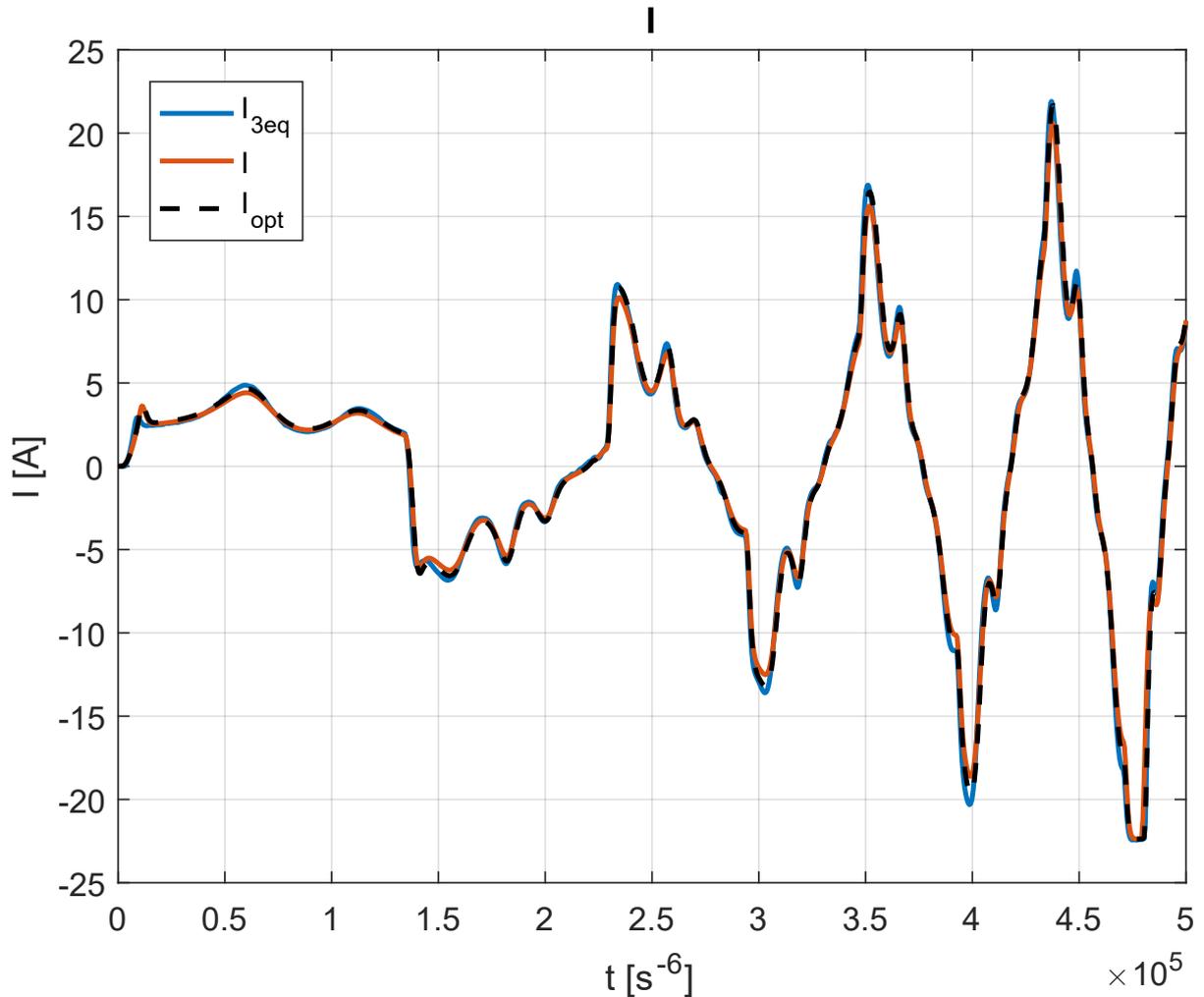


Figure 3.30: Current output comparison before and after optimization, with 23% of winding A working

The first thing to be noticed is the curves amplitude, definitely increased in respect to that shown in figure 3.28; the reason for this, can be found in the fact that, according to the Ohm law $V = RI$, when a short circuit occurs, the resistance decreases, causing a growth in the absorbed current, at the same

level of voltage supply, indeed: $I = \frac{V}{R}$. The second effect to put in evidence are the oscillations along the signal, caused by the SC form function effect on the motor model, affecting more the trend, when the phase A contribution has greater influence (when the motor angular position is “aligned” to winding A). However, be aware that this fault entity already produces effects on the actuation, in such a way the EMA going through this, is to be considered irreversibly damaged and to be dismissed. Furthermore, the fact the curves have a good fitting already before optimization, means that a good job in the assumptions of failure functions form and parameters has been done (see section 3.1.4). The optimized failure parameters values, in this case, producing a $MSE_f \simeq 1,05 [A^2]$, are the following:

$$\begin{cases} k_{f_{s_{23}}} = 11.24 \\ k_{f_t} = \frac{1}{18} = 0.0444 \end{cases} \quad (3.55)$$

While the MSE_f value before optimization was $2.05 [A^2]$. All the other parameter values for the remaining studied conditions, are reported in the table below:

N_A [%]	k_{f_s} [/]	k_{f_t} [/]	$MSE_{f_{bef}}$ [A^2]	$MSE_{f_{opt}}$ [A^2]
100	9	0.0556	0.157	0.023
75	10.52	0.0487	0.0872	0.0816
50	11.15	0.0458	0.2421	0.1668
39	10.03	0.0500	0.4525	0.3514
25	10.93	0.0484	3.0710	2.8889
23	11.24	0.0444	2.0526	1.0495

Table 3.2: Winding A SC data summary

To be noticed the fact that values of MSE_f increase as much as the working coils percentage decreases, since mismatching between high fidelity and monitoring models becomes greater. Anyway, that’s not a problem, since the application range for which the monitoring model has been thought, is that dealing with very small amount of failed coils, which is the kind of situations relevant for early warning and fault detection; moreover, the entity of “bad” matchings is confined, in the words case, within something similar to the 6% of maximum current value, which can be considered acceptable. Below the interpolation charts produced by data of table 3.2:

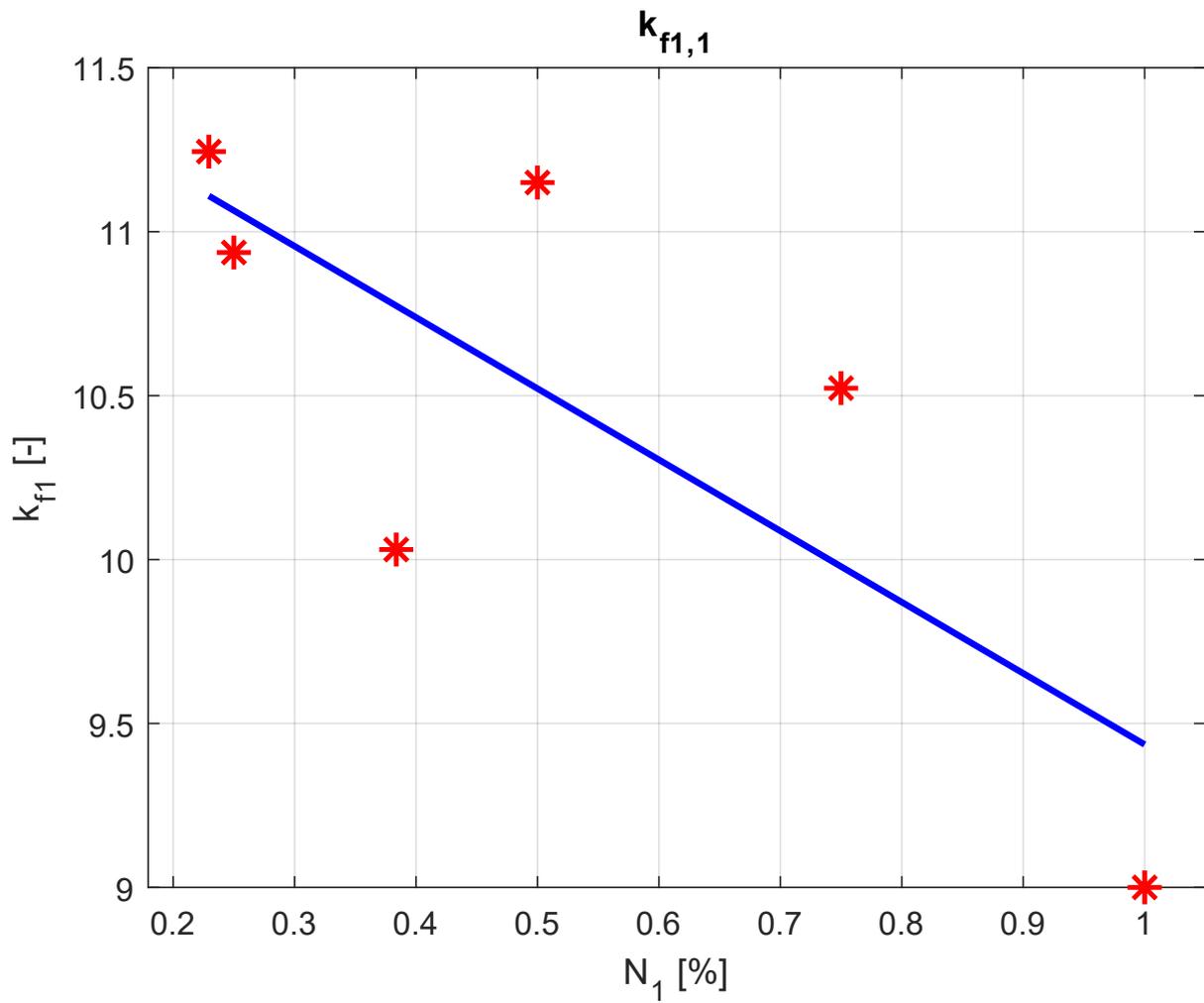


Figure 3.31: K_{fs} optimized data interpolation, with winding A SC

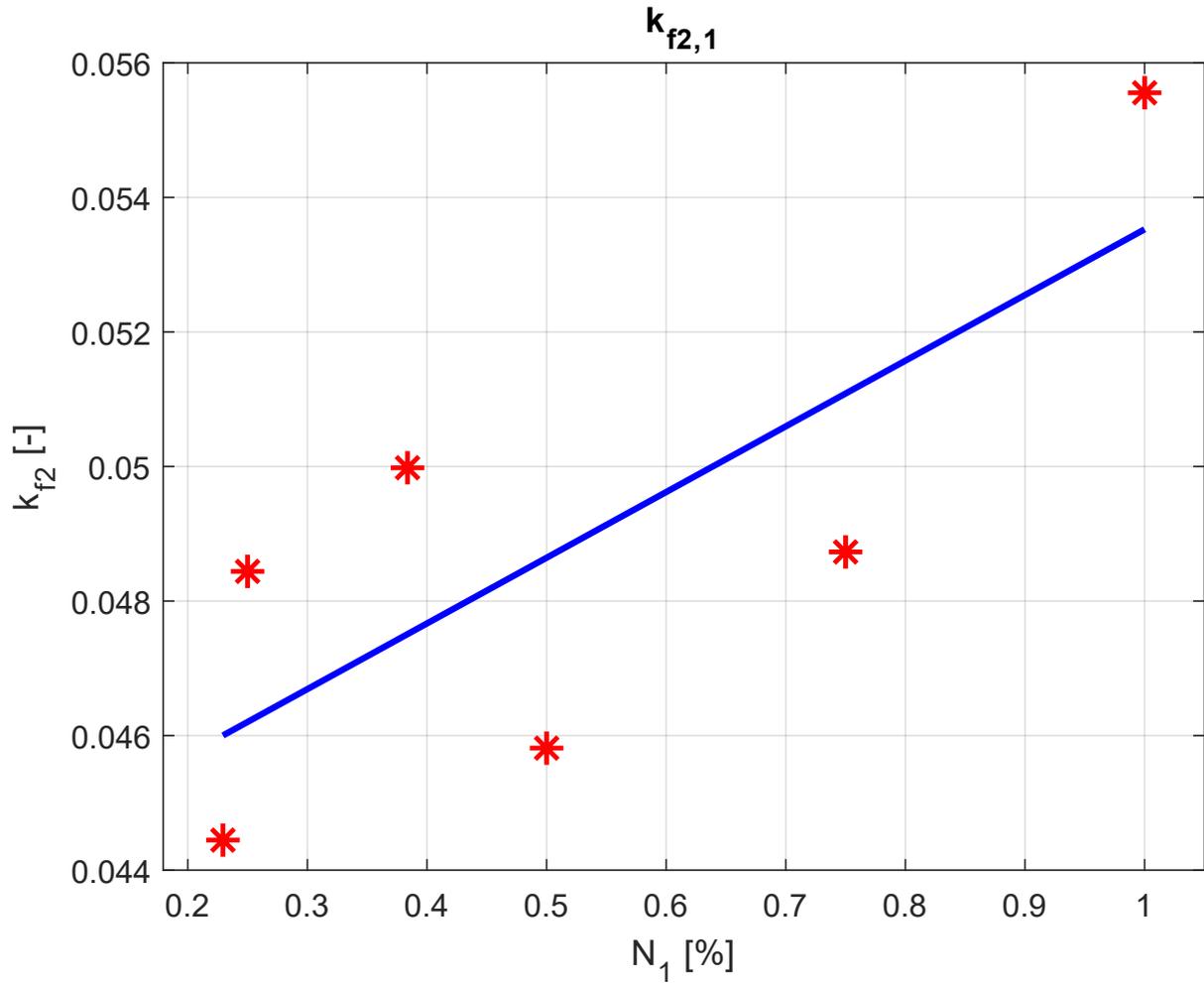
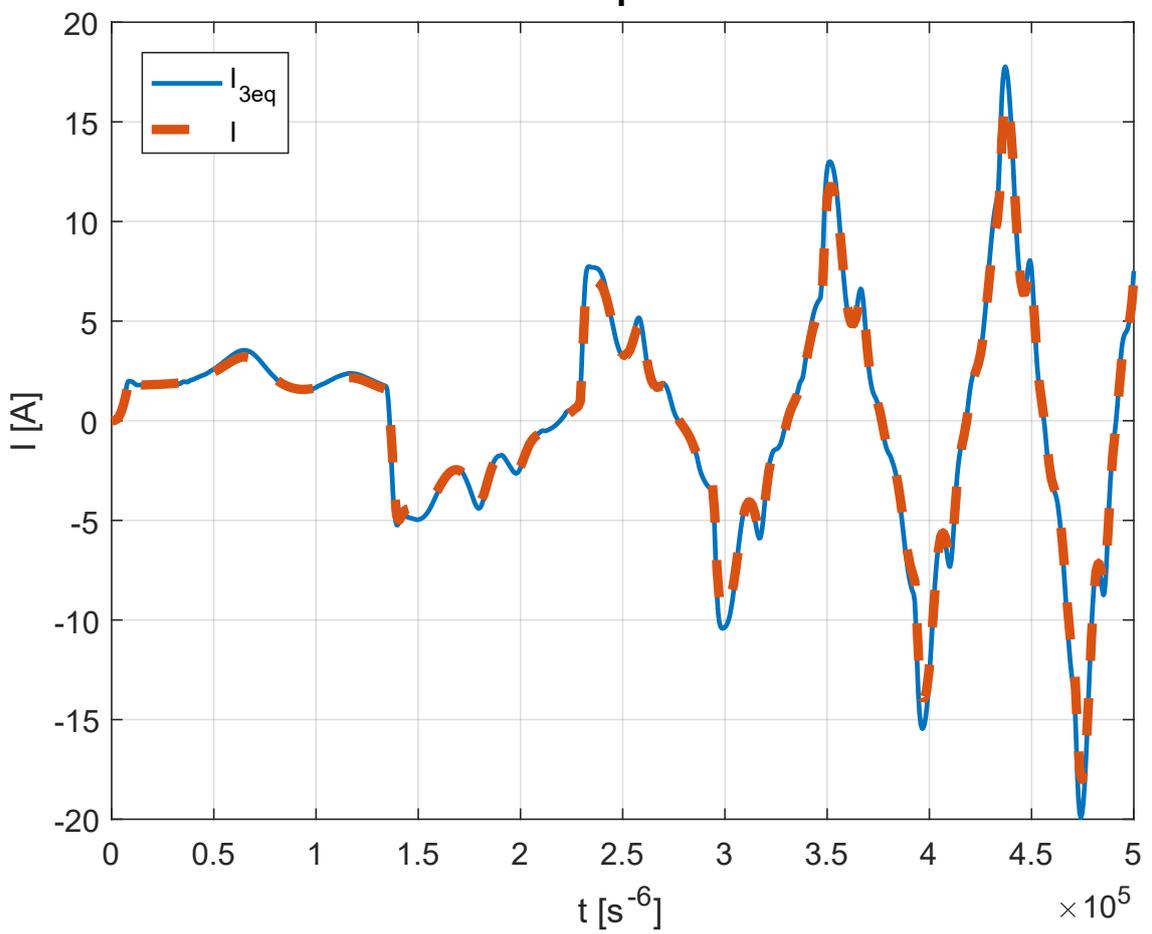
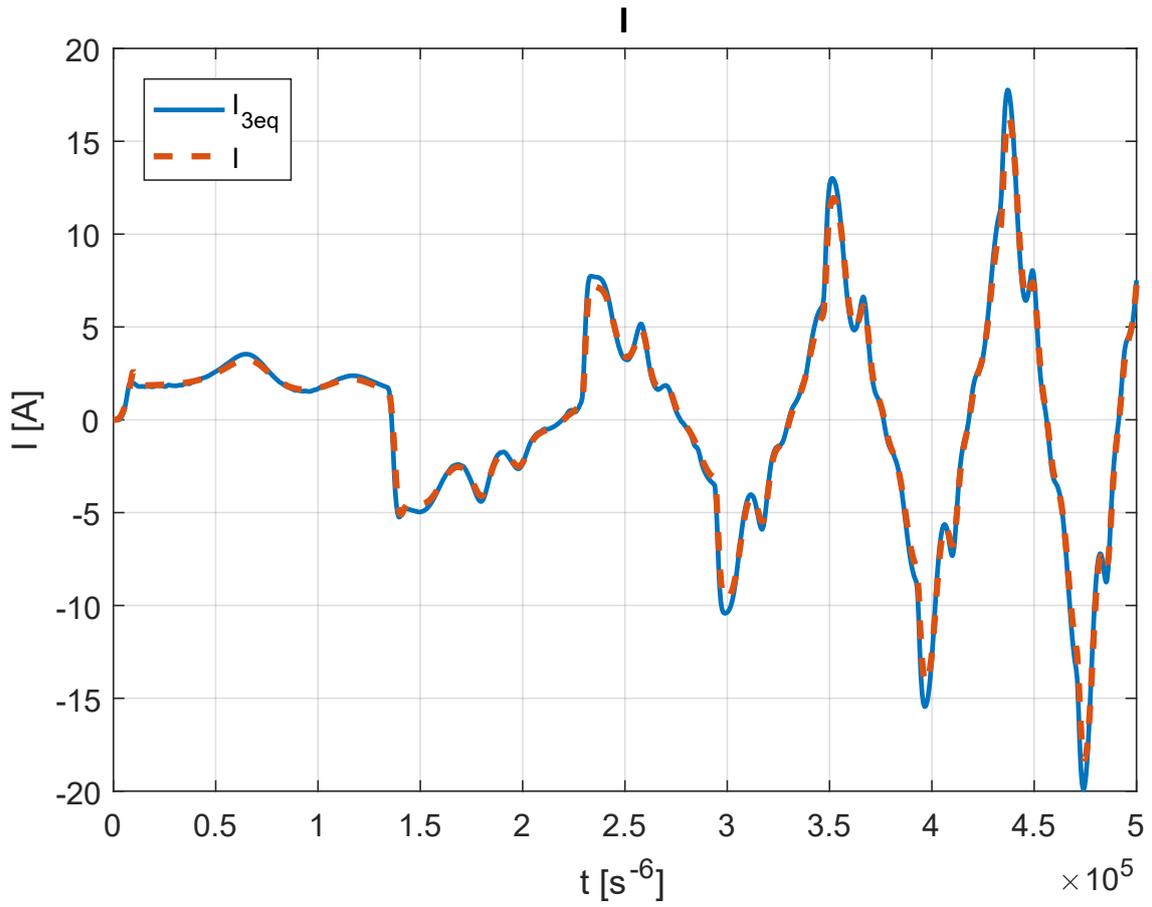


Figure 3.32: K_{f_t} optimized data interpolation, with winding A SC

N.B.: In all the interpolation graphs the following applies:

$$\begin{cases} k_{f_s} = k_{f_1} \\ k_{f_t} = k_{f_2} \\ k_{f_E} = k_{f_3} \\ N_A = N_1, N_B = N_2, N_C = N_3 \\ Z = E_1, \phi_Z = E_2 \end{cases} \quad (3.56)$$

We can see from the figure above, both k_{f_s} and k_{f_t} are well fitted by a linear interpolation, in which has been included the nominal conditions value for both parameters. This gives us the chance to enhance the monitoring model in the future, by implementing something similar to a *look-up table*, able to choose the right parameters value in function of the simulated faults. Here we report all the other test current trends, before and after the optimization, even though there will be difficult to see any differences between them, due to very small entity of the MSE_f variation.



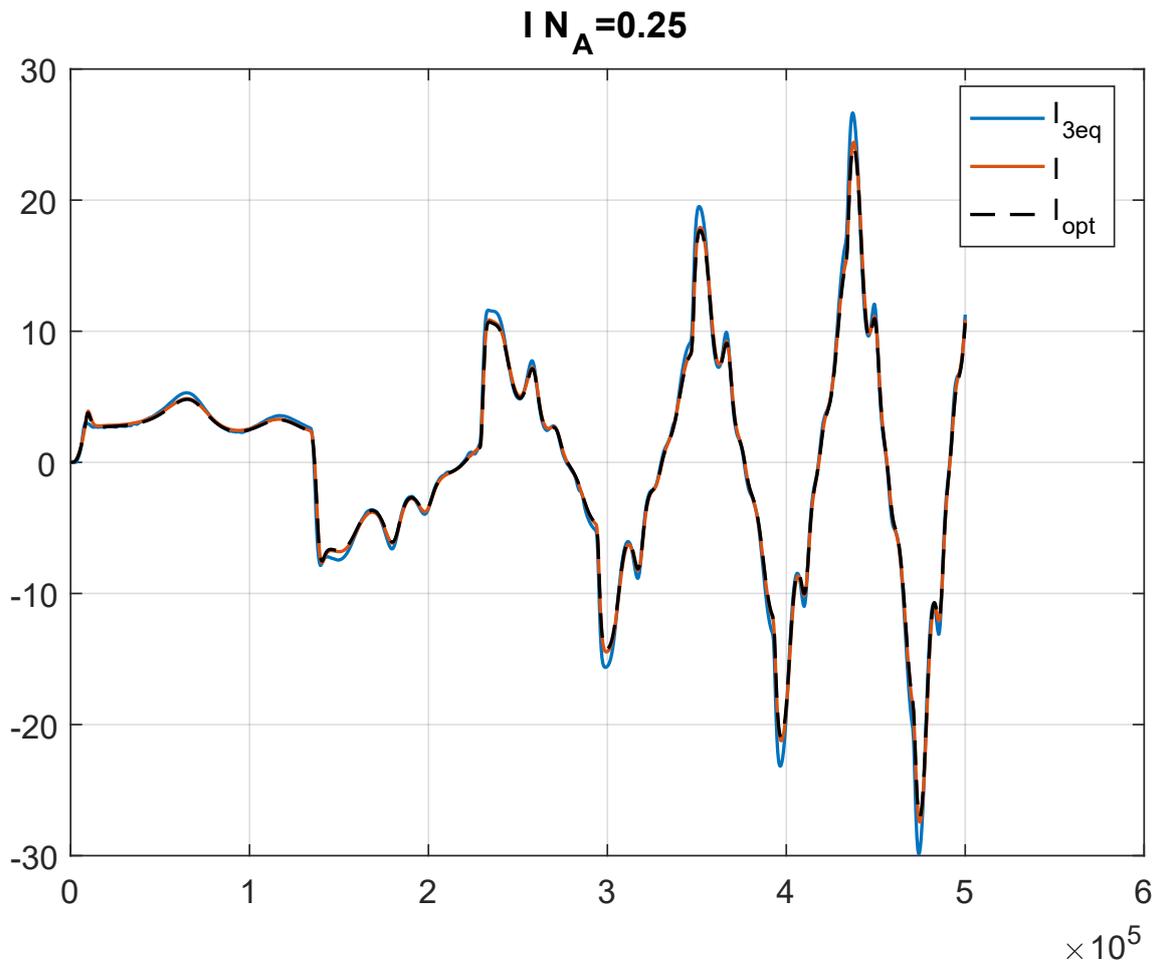
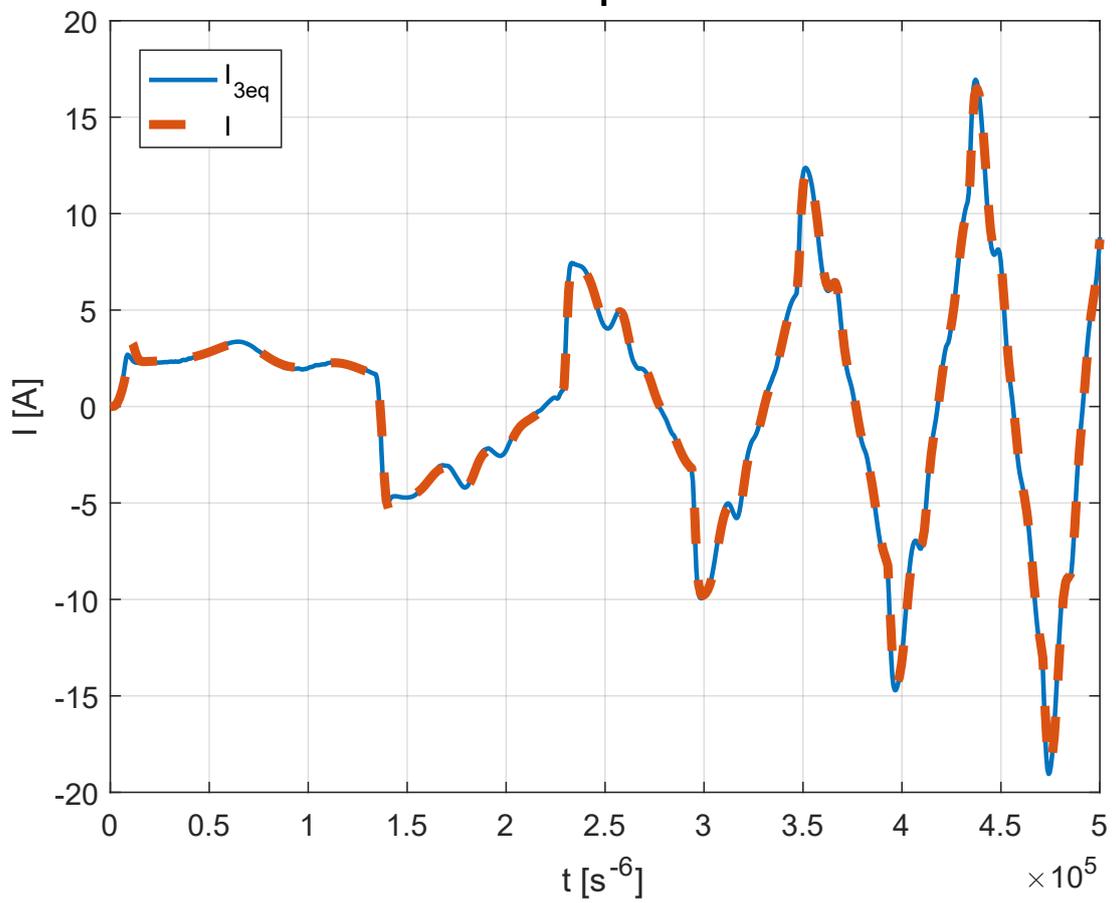
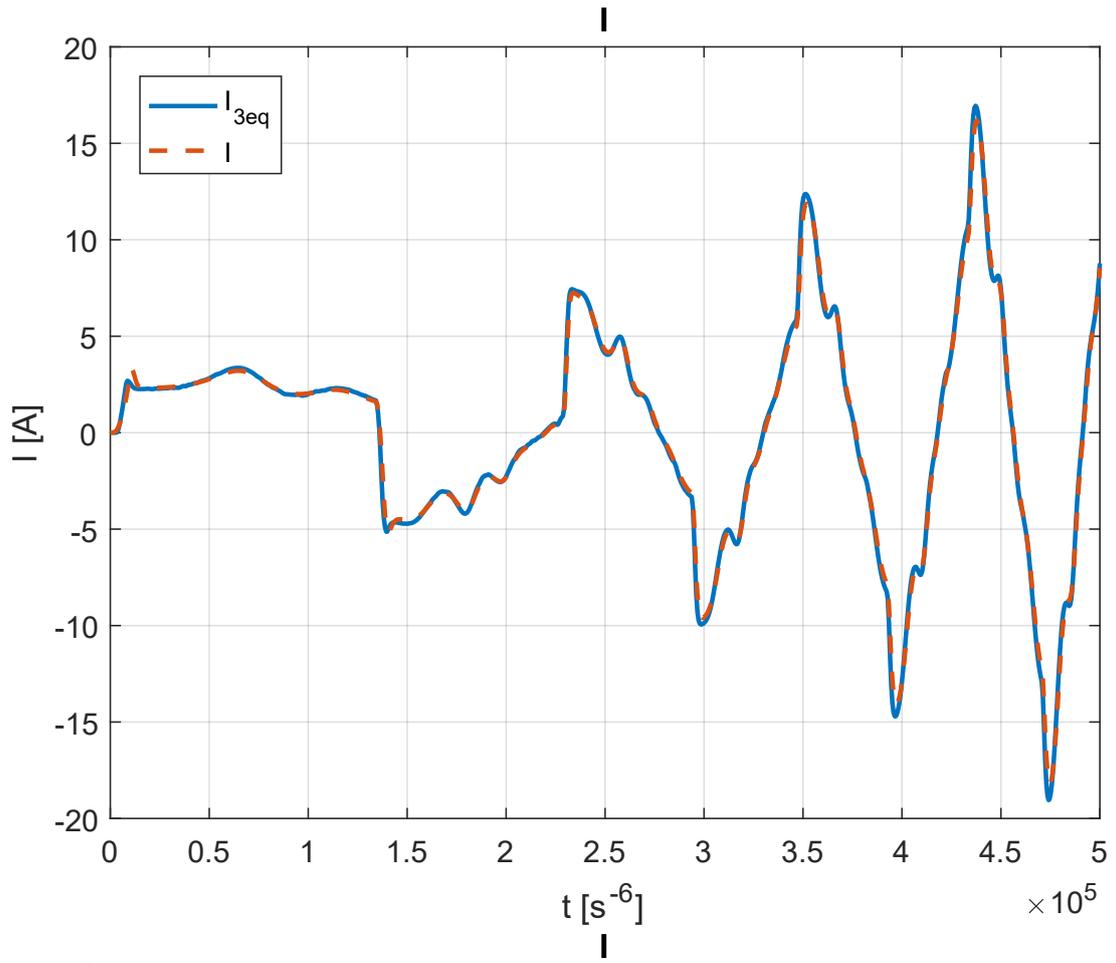


Figure 3.33: Current trends with winding A 75% SC, respectively: before, after optimization, all together



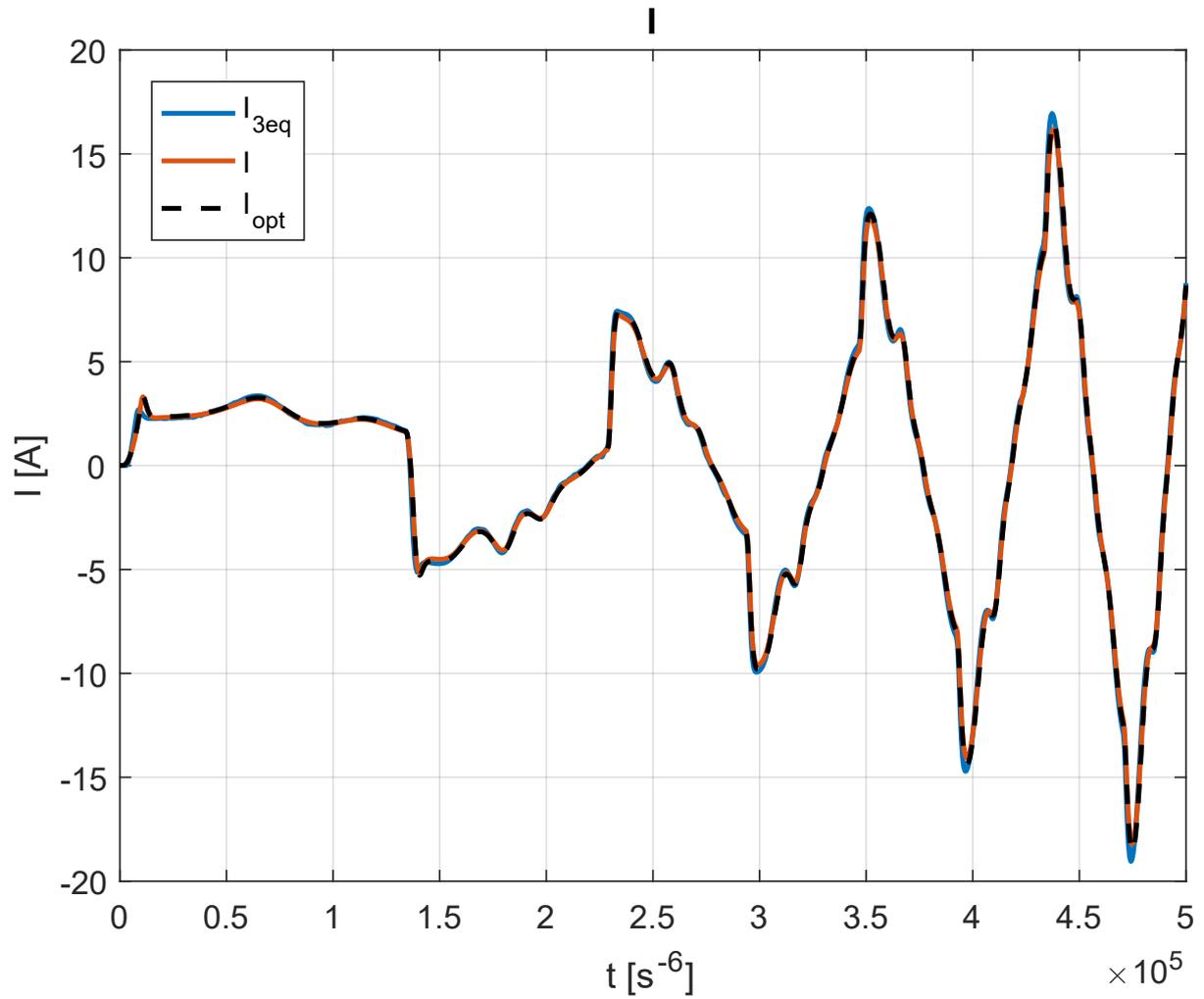


Figure 3.34: Current trends with winding A 61% SC, respectively: before, after optimization, all together

In all these figures, we are going to see a gradual decreasing in current magnitude and oscillation entity due to increasing of the working coils percentage.

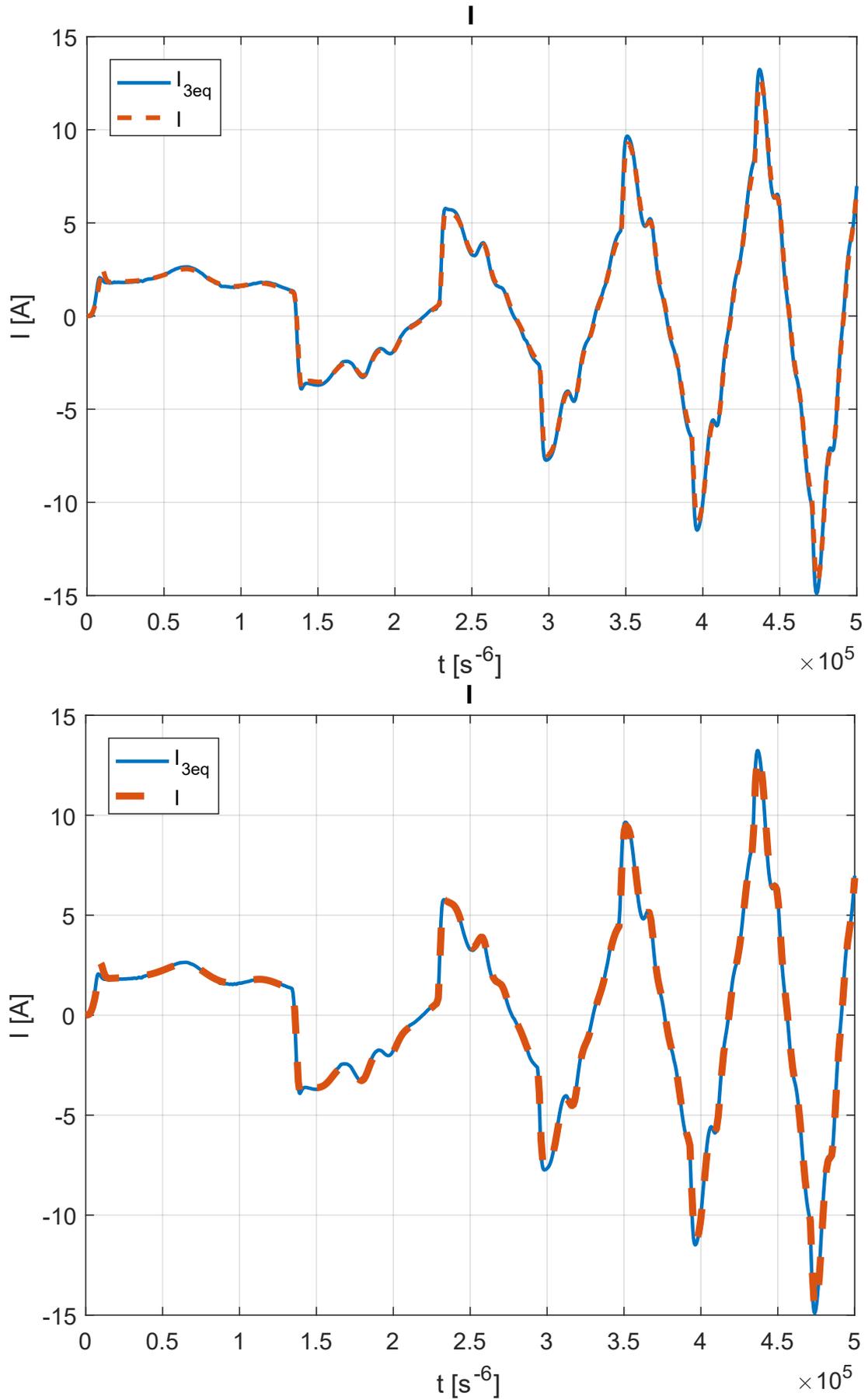


Figure 3.35: Current trends with winding A 50% SC, respectively: before and after optimization

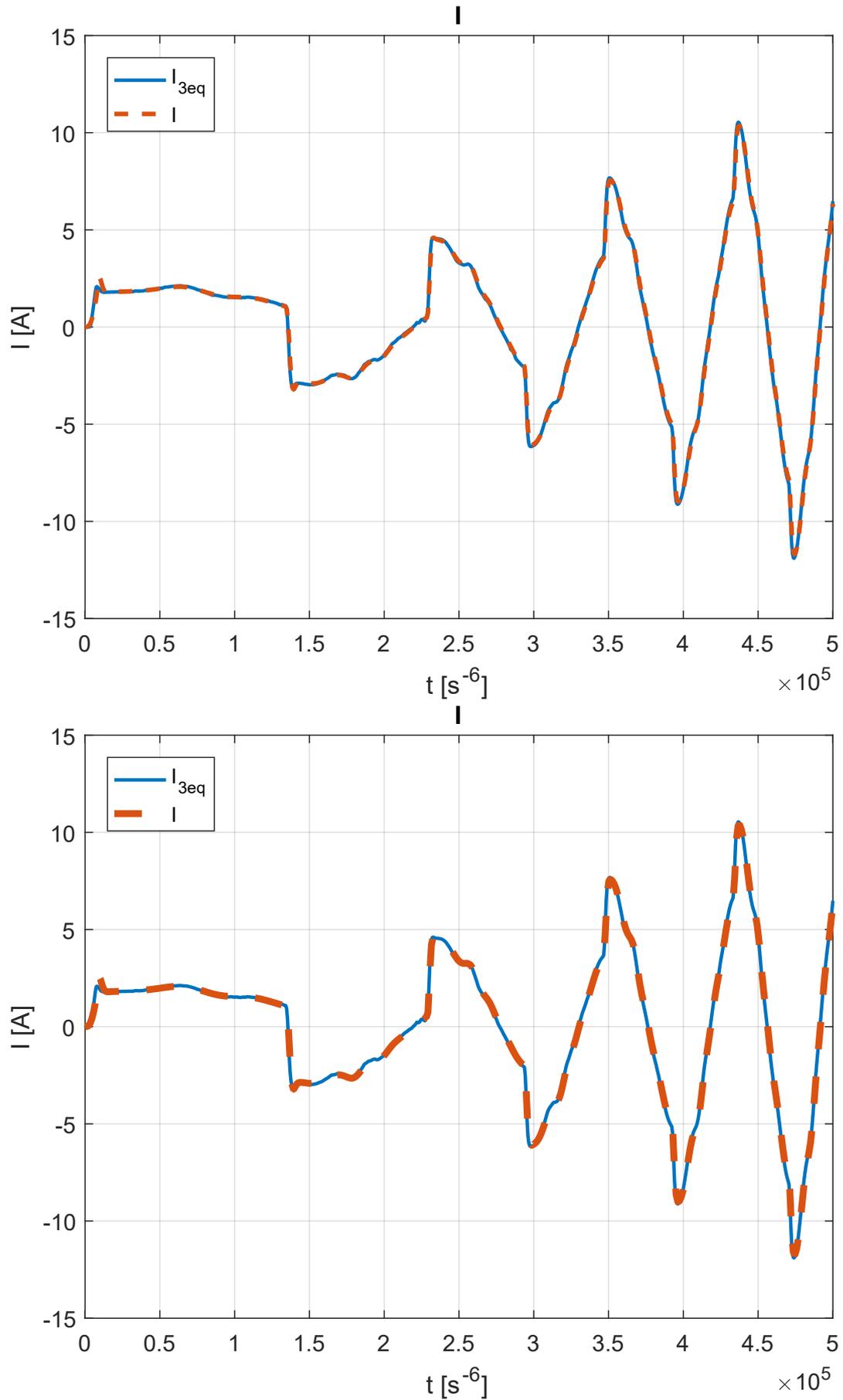


Figure 3.36: Current trends with winding A 25% SC, respectively: before and after optimization

3.3.2 Multiple Windings Short Circuit

The second kind of tests conducted, involved the SC failure of all the windings contemporary, and randomly generated. Anyway a further action has been accomplished: in order to have a uniform samplings distribution with the same interval amplitude, regardless of the distance from the coordinate origin, the randomly generated faults, has been elevated with the number of variables involved. This operation results useful to the fact that, being the main values range aimed for this application really near to 1 (meaning small percentage of failed coils), the number of samplings remains the same as that near to the origin. Thus, in this specific instance, the random generation involved the working coils percentage $[N_A, N_B, N_C]$, in such a way the latter data have been set to:

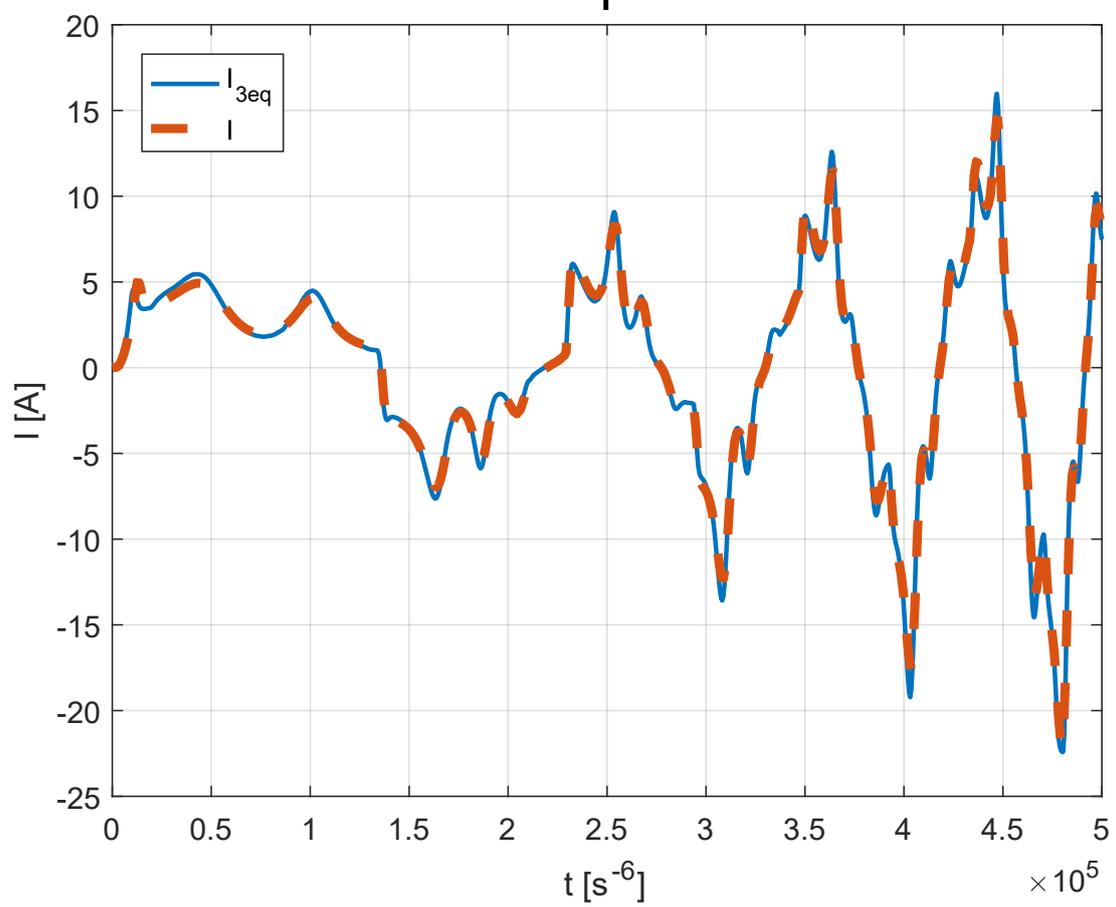
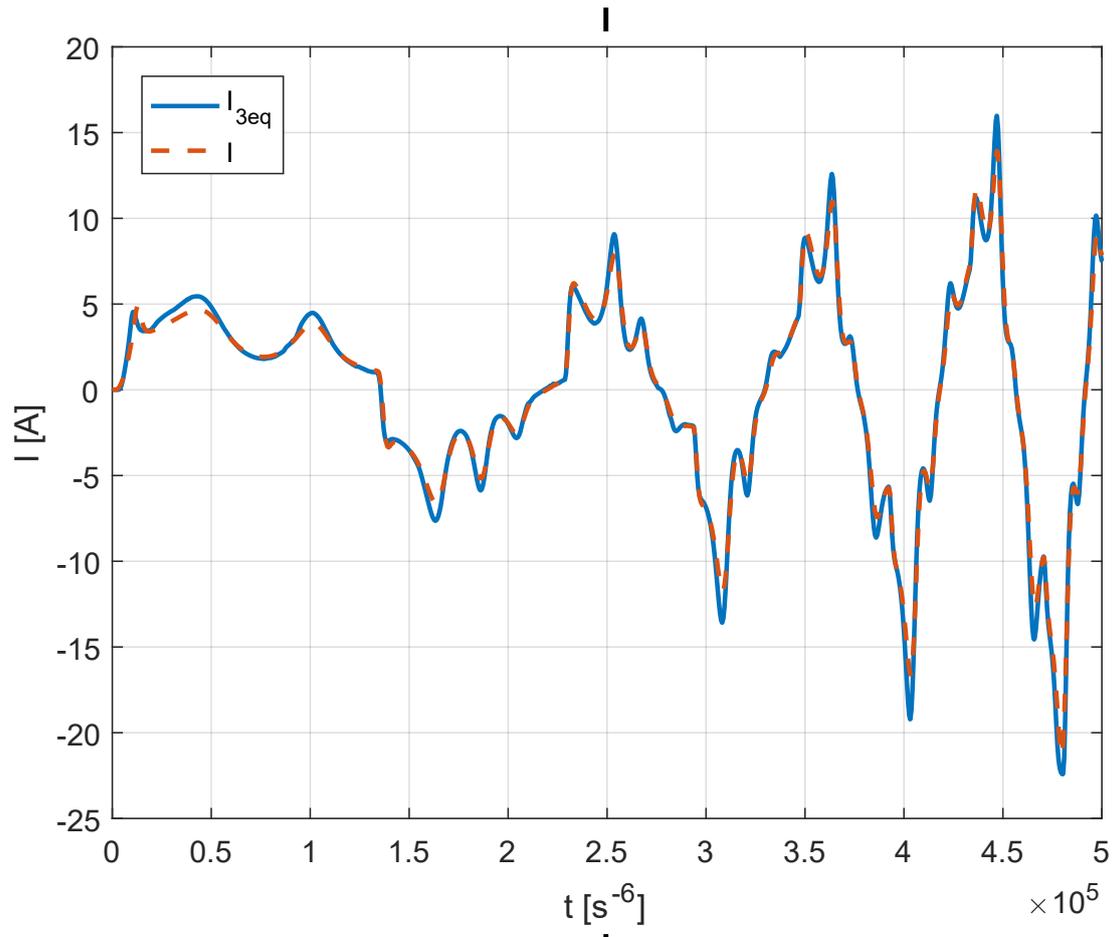
$$[N_A, N_B, N_C] = 1 - rand(3)^3 \quad (3.57)$$

Where, the Matlab command $rand(3)$, generates a vector with three casual components having values between 0 and 1. Actually, 21 different data were generated, for a total of 7 diverse faults combinations, to which our monitor model was subjected in order to evaluate its response with multiple low entity SC failures. At the same time, MSE_f has been evaluated before and after optimization, as well as the failure parameters, which have all been reported in the following table:

N_A [%]	N_B [%]	N_C [%]	k_{fs} [\backslash]	k_{ft} [\backslash]	$MSE_{f_{bef}}$ [A^2]	$MSE_{f_{opt}}$ [A^2]
86.18	3.86	99.94	10.90	0.0458	2.0259	1.3882
99.71	87.11	71.04	9.58	0.0531	0.0863	0.0804
82.50	98.00	86.18	8.19	0.0600	0.0745	0.0742
100	99.90	99.50	8.99	0.0558	0.0490	0.0481
54.93	86.90	17.32	10.94	0.0466	1.5289	1.2485
38.87	79.92	79.41	10.03	0.0500	0.4525	0.3514
22.94	55.60	91.44	11.24	0.0444	2.0525	1.0495

Table 3.3: Windings A, B, C SC data summary

Subsequently, the failure parameters have been interpolated in respect to each windings faults. Since, every k_{f_i} value has been influenced by all windings SC, we expect to see a grater data dispersion, best fitting the winding failures which has better uniform coverage of the faults interval $([0, 1])$. However, looking at MSE_f , again the differences between before and after optimization in therms of current matching are going to be very difficult to see, without considering the fact that the values are not bad even before the GA does its trick.



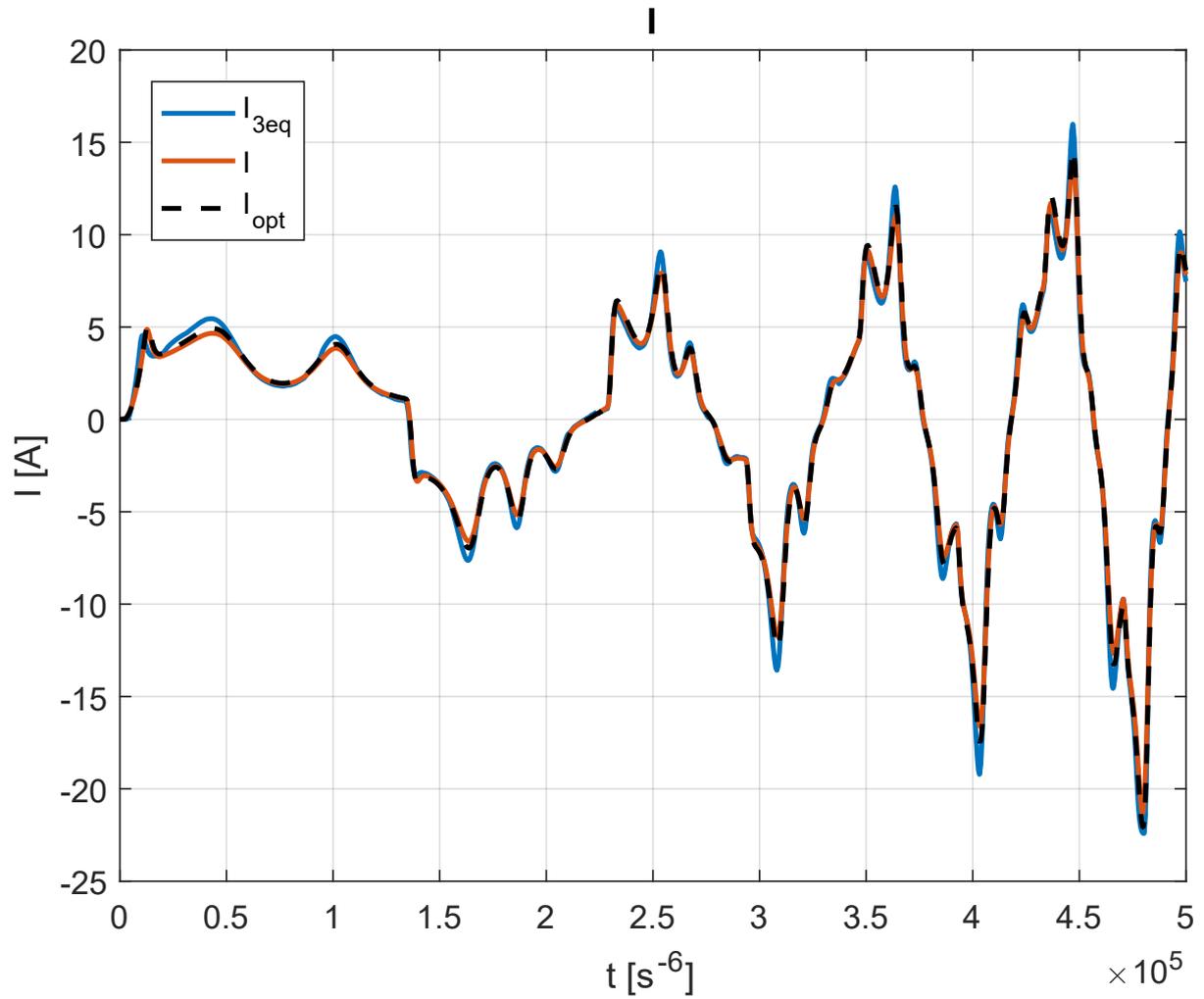
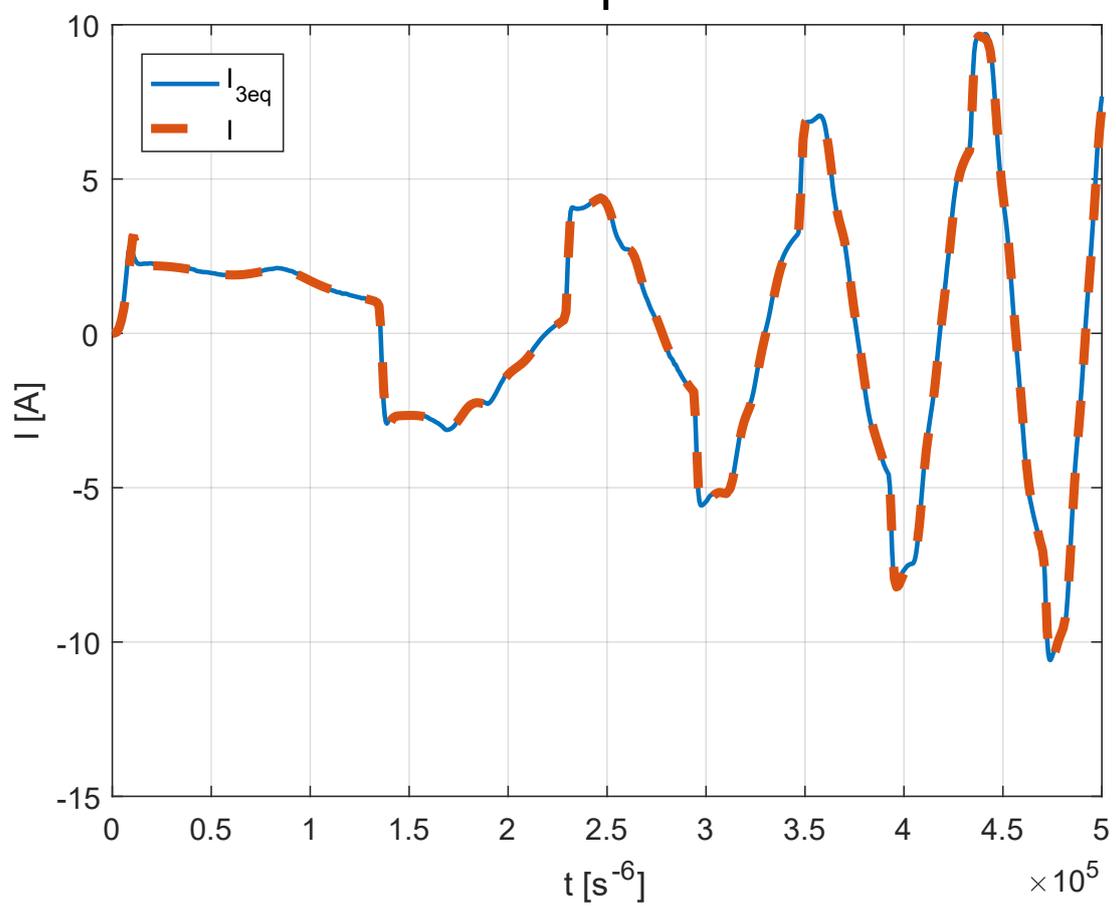
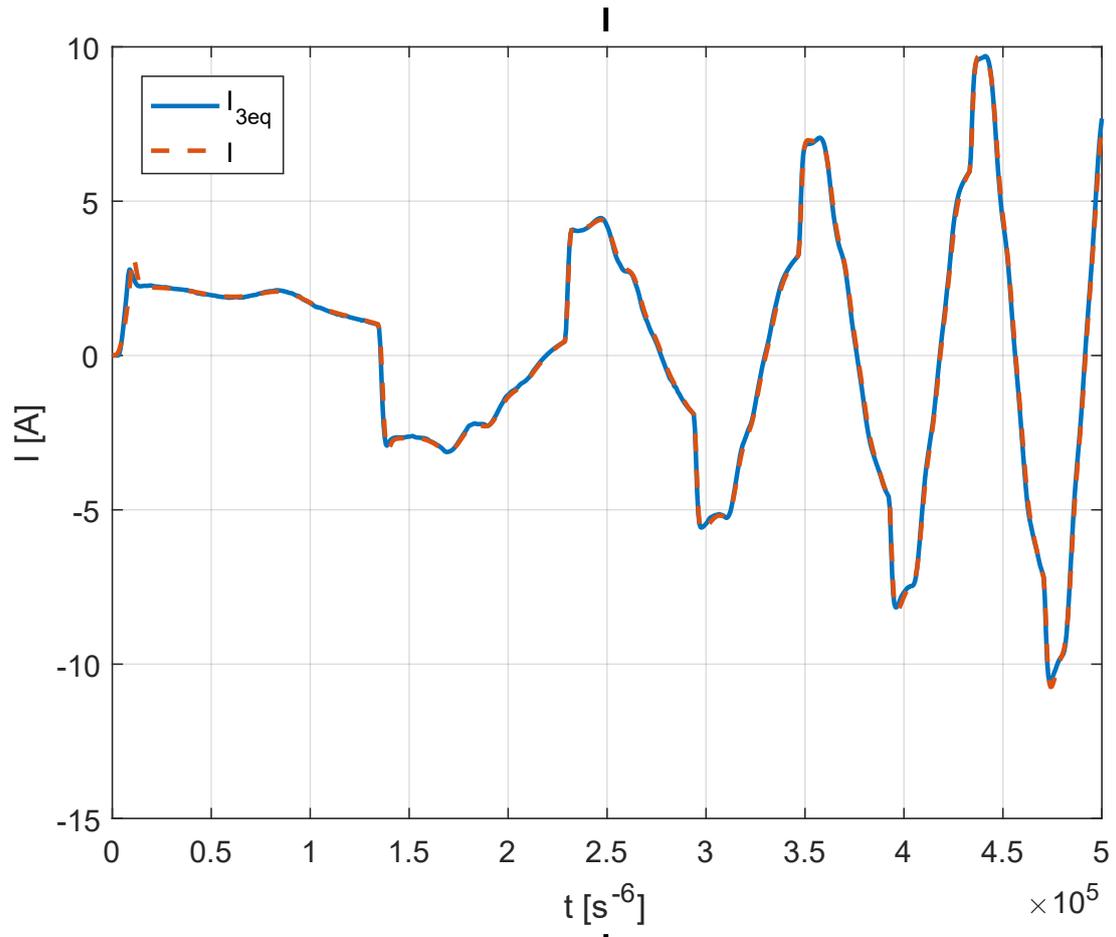


Figure 3.37: Current trends with $[N_A, N_B, N_C] = [0.86, 0.038, 0.99]$, respectively: before, after optimization, all together

Here, the trend is similar to the one reported in figure 3.30, except for location, number and amplitude of the oscillations: since the relevant failures are those imposed to windings A and B, is reasonable attributing these differences to the fact that, in this case the current suffers modifications more times per motor revolution, and with more strength.



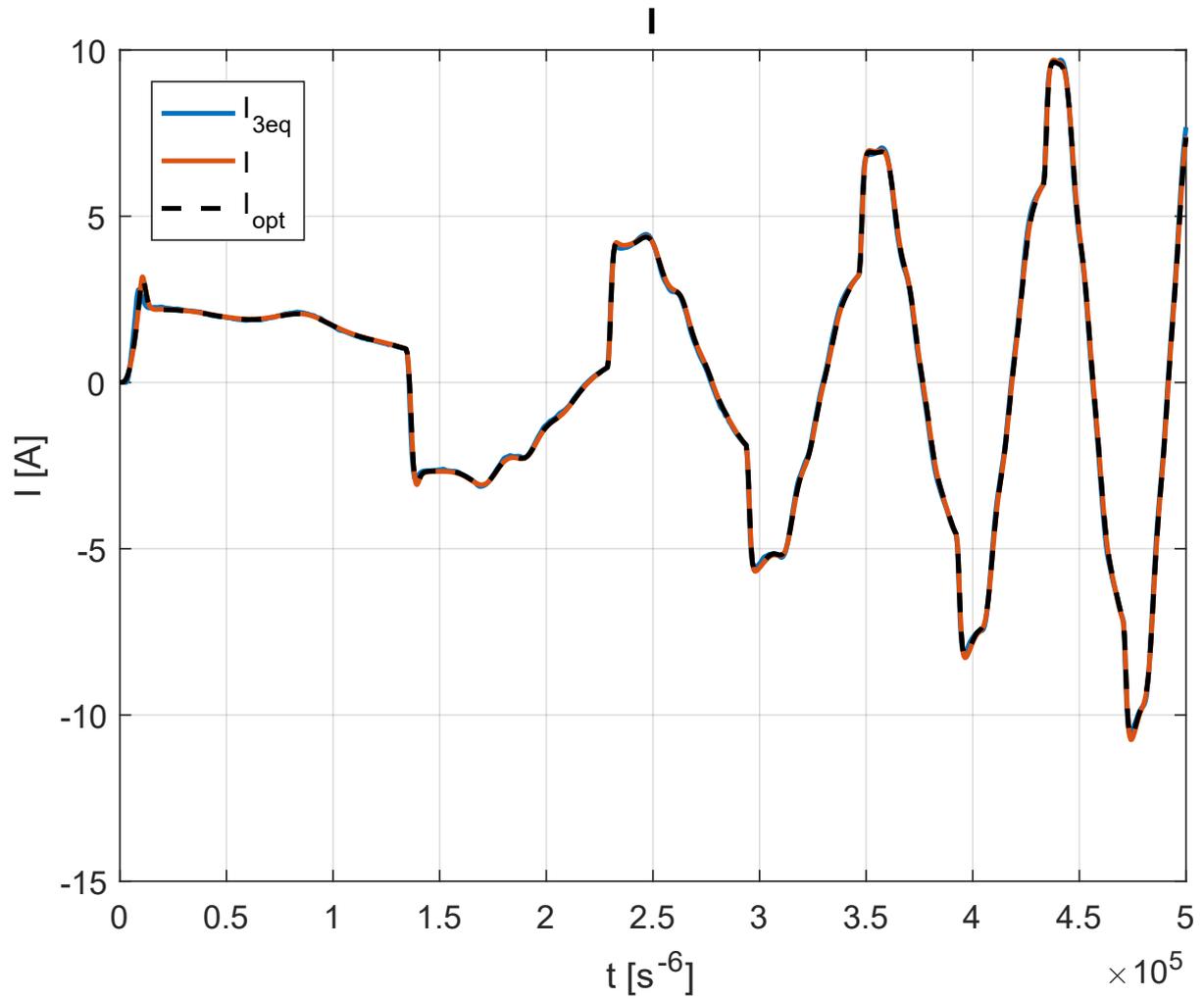
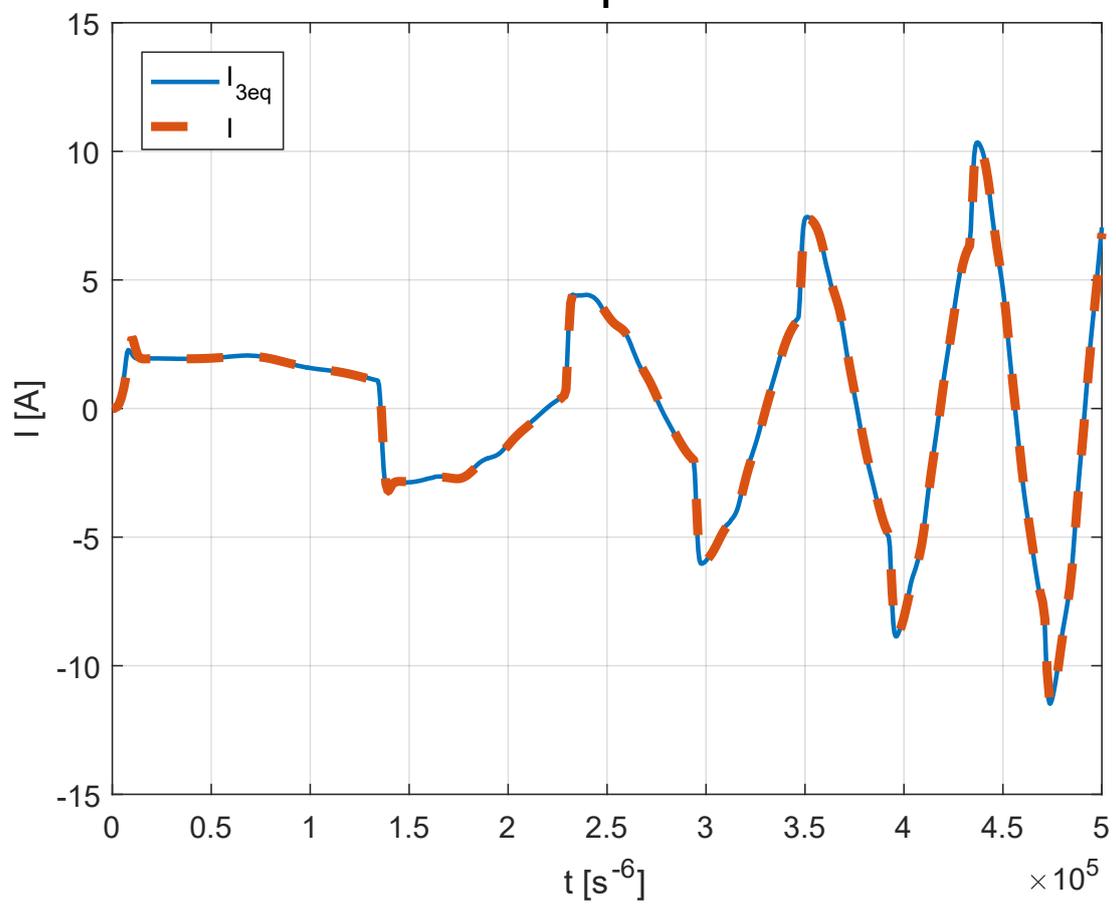
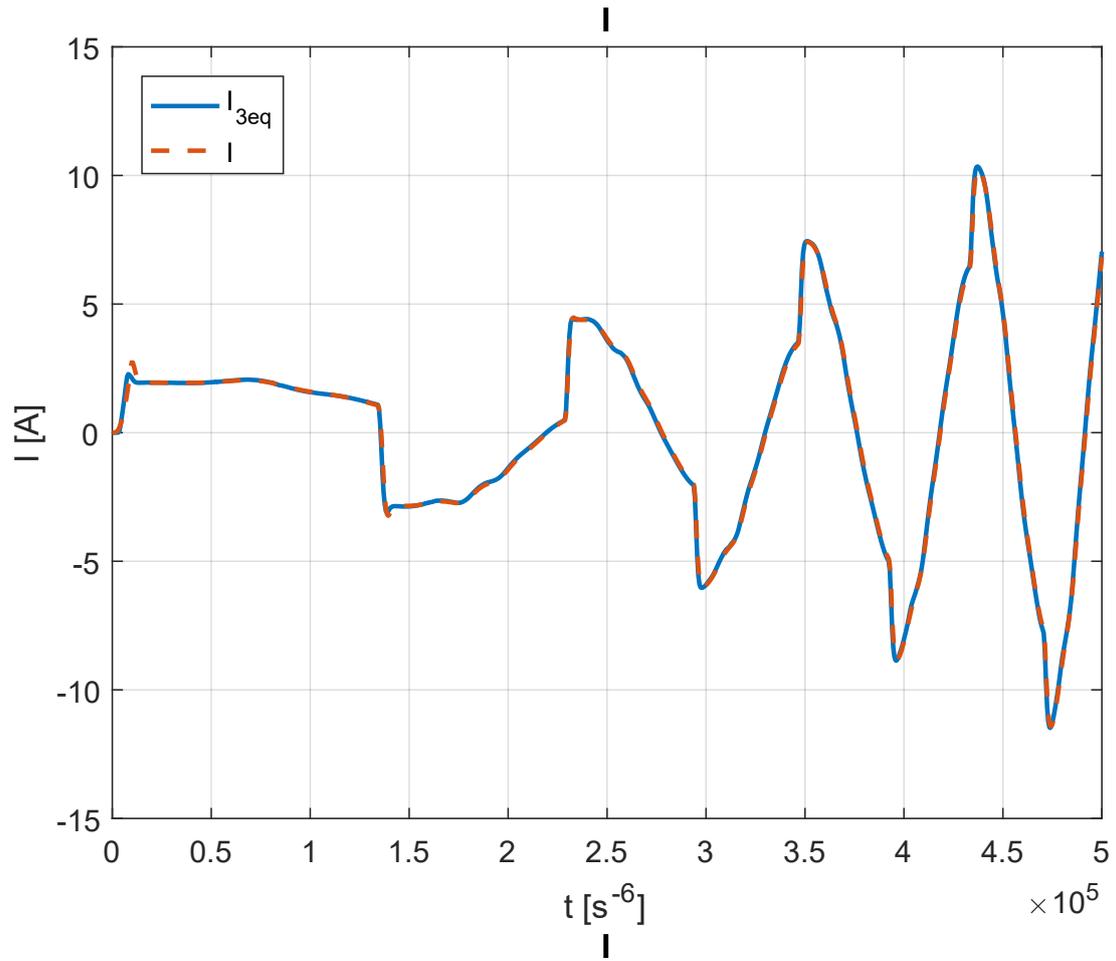


Figure 3.38: Current trends with $[N_A, N_B, N_C] = [0.99, 0.87, 0.71]$, respectively: before, after optimization, all together

In the figures above, instead, the faults combination causes the “chopping” of the current peaks seen until now, and a strong oscillations reduction. Furthermore, seen the minor SCs entity, the magnitude has definitely decreased.



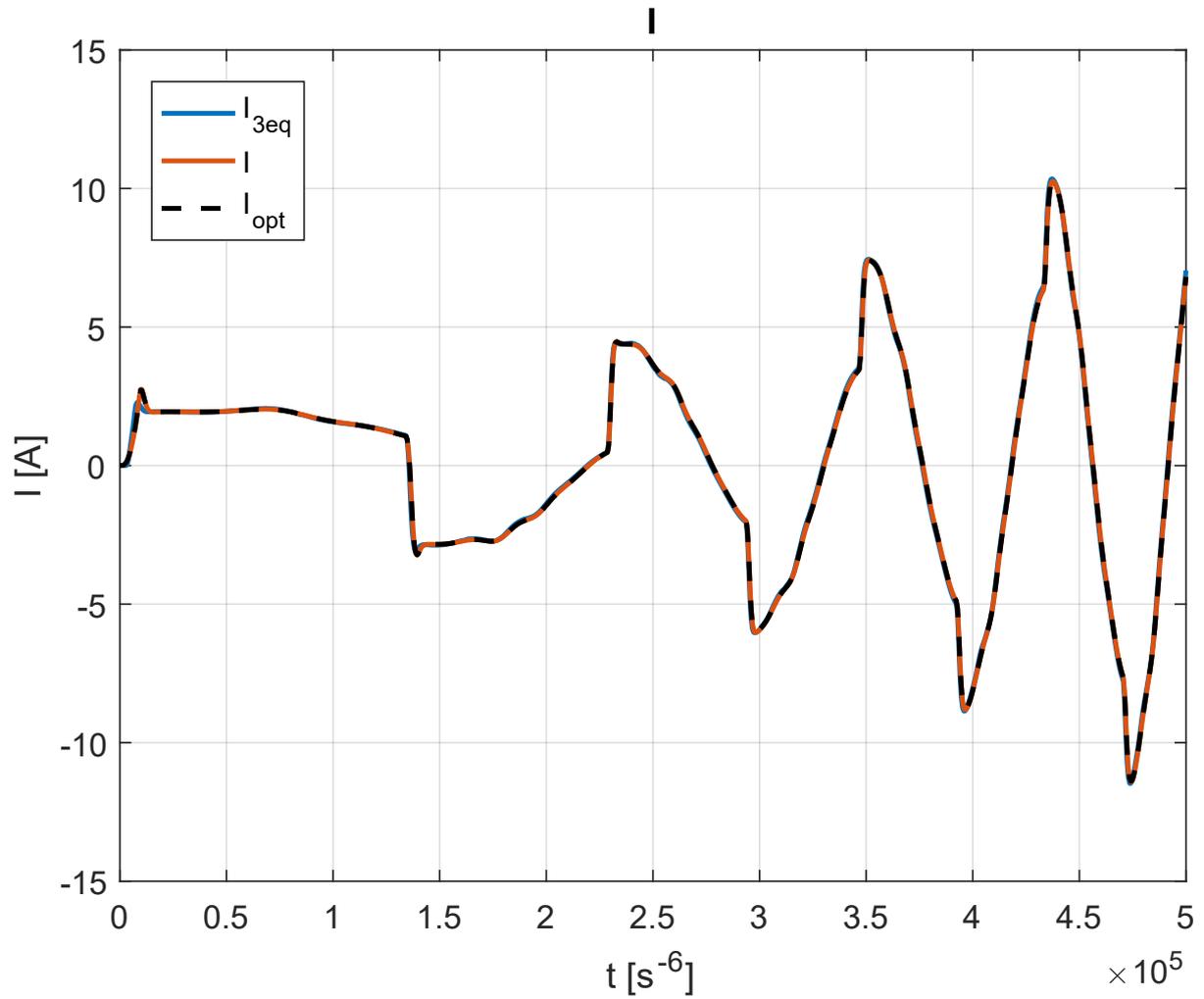
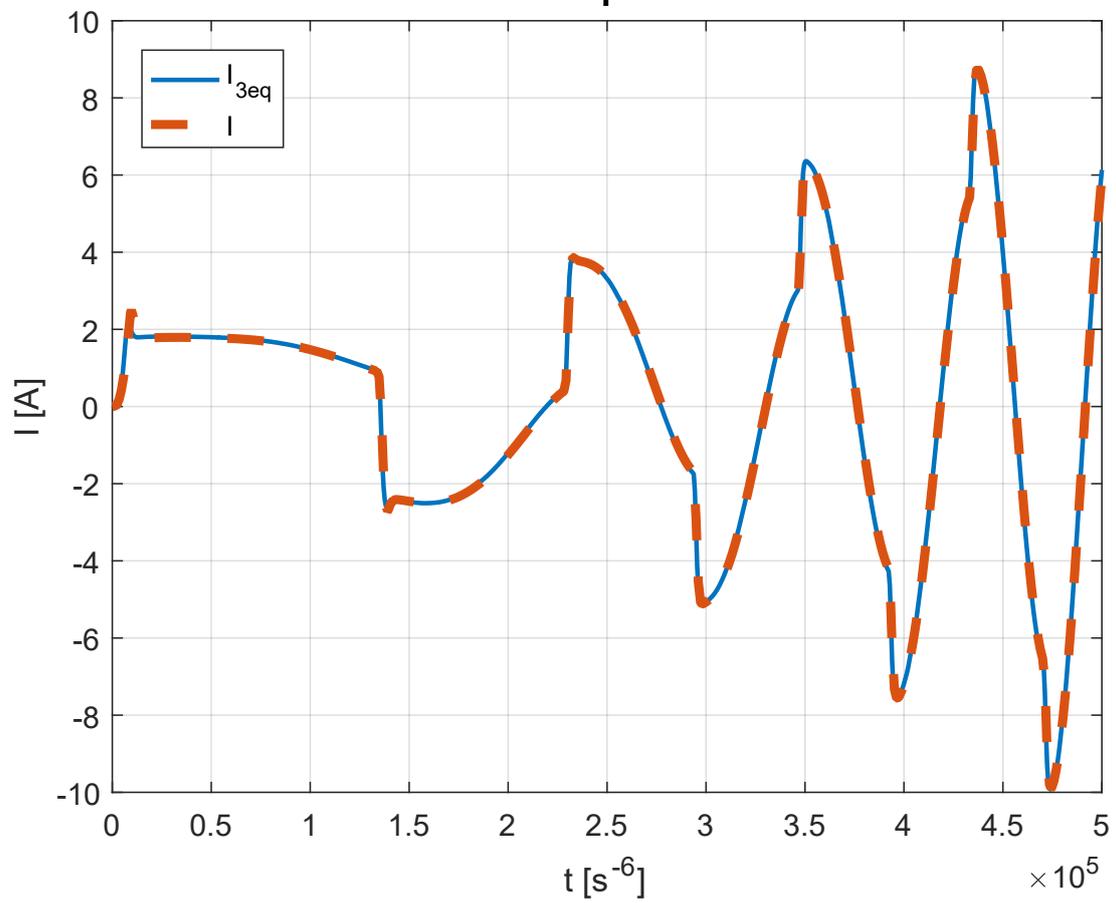
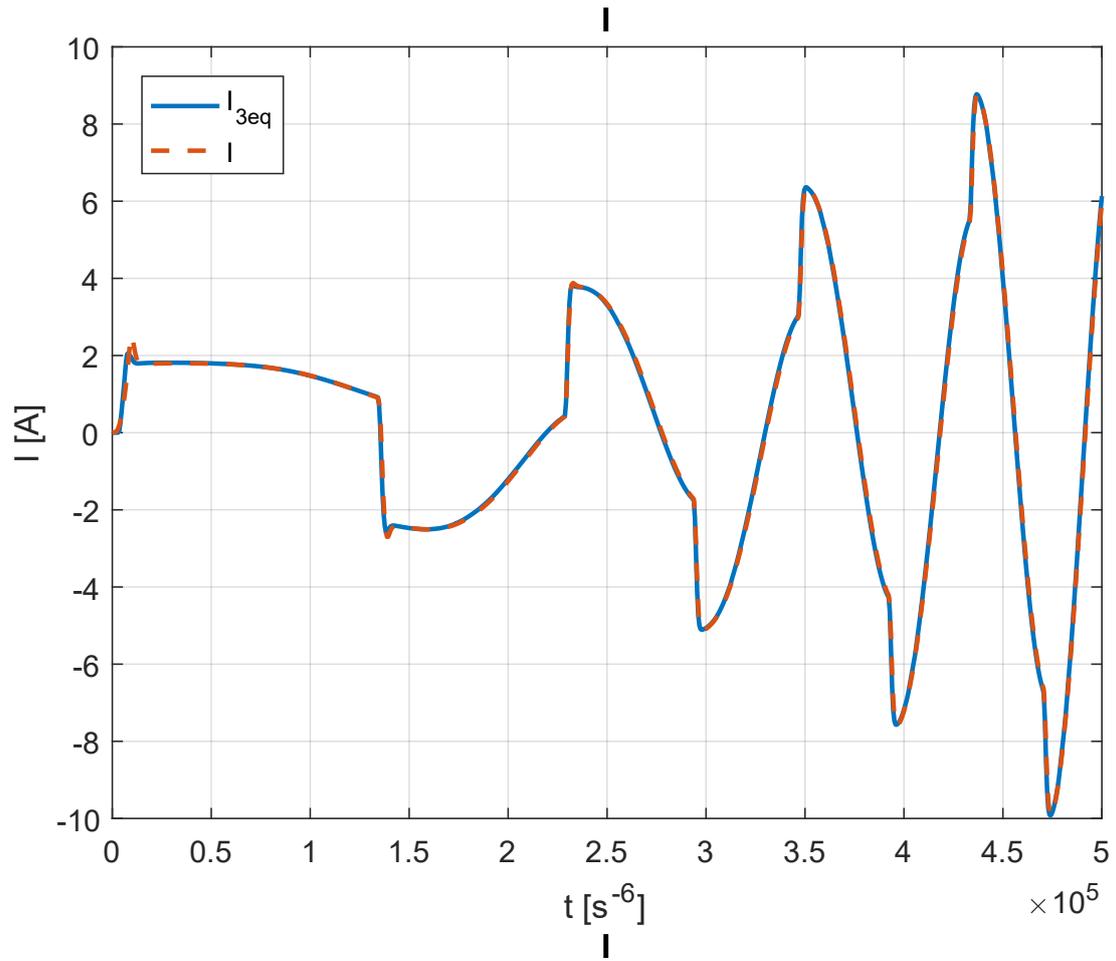


Figure 3.39: Current trends with $[N_A, N_B, N_C] = [0.82, 0.98, 0.86]$, respectively: before, after optimization, all together

The figures 3.39, show a trend very similar to the nominal one, due to the fact that the windings A and C SCs are very similar between each other, while the winding B SC is almost null.



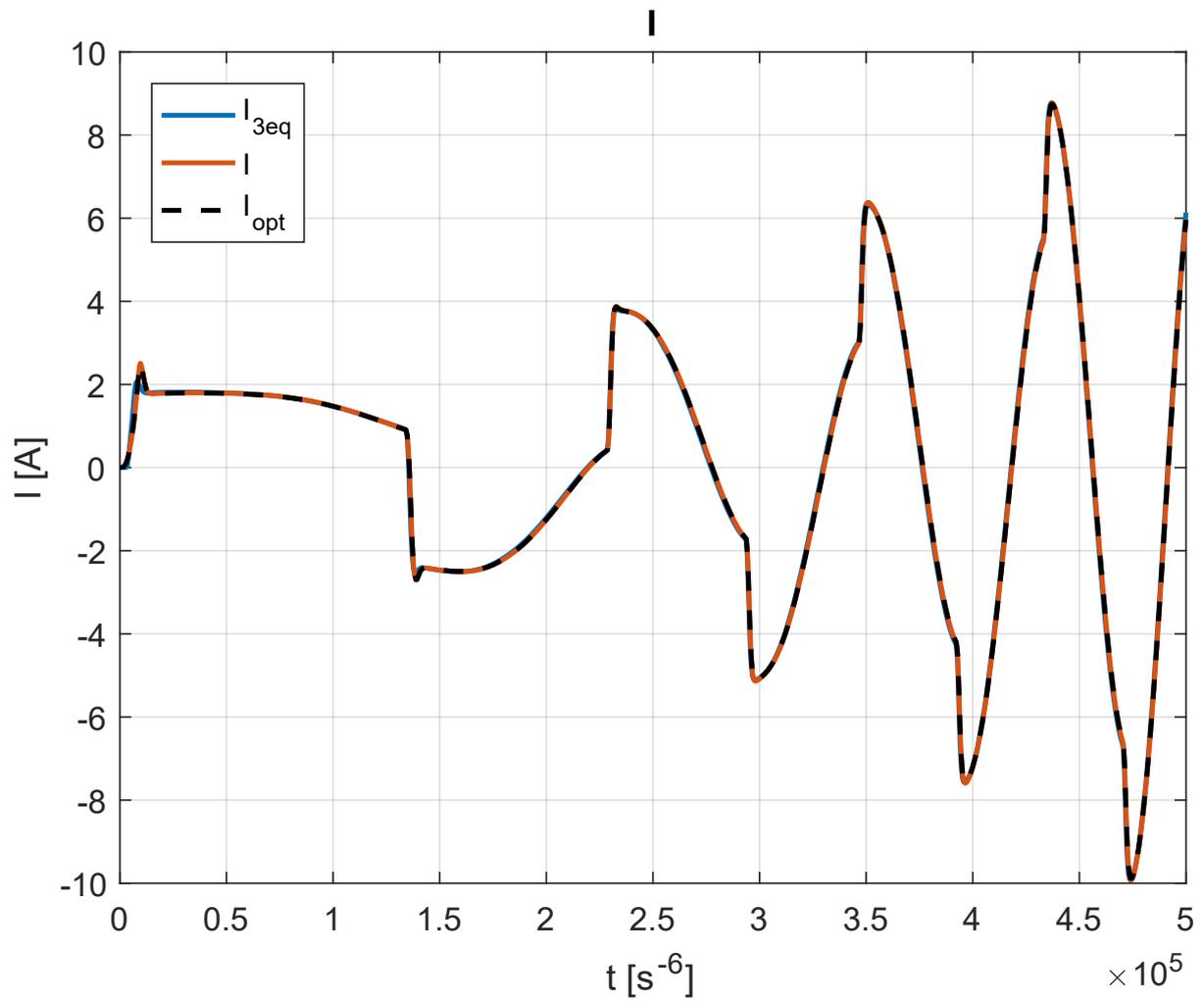
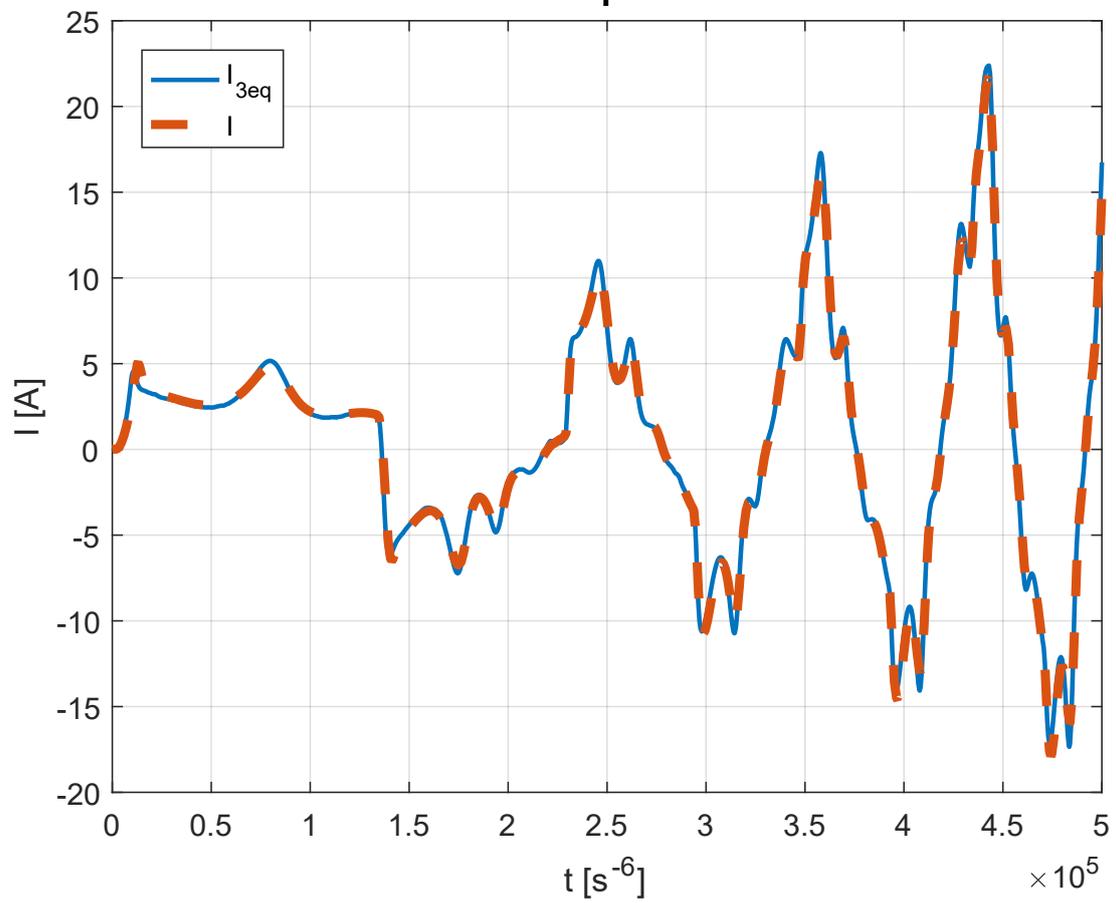
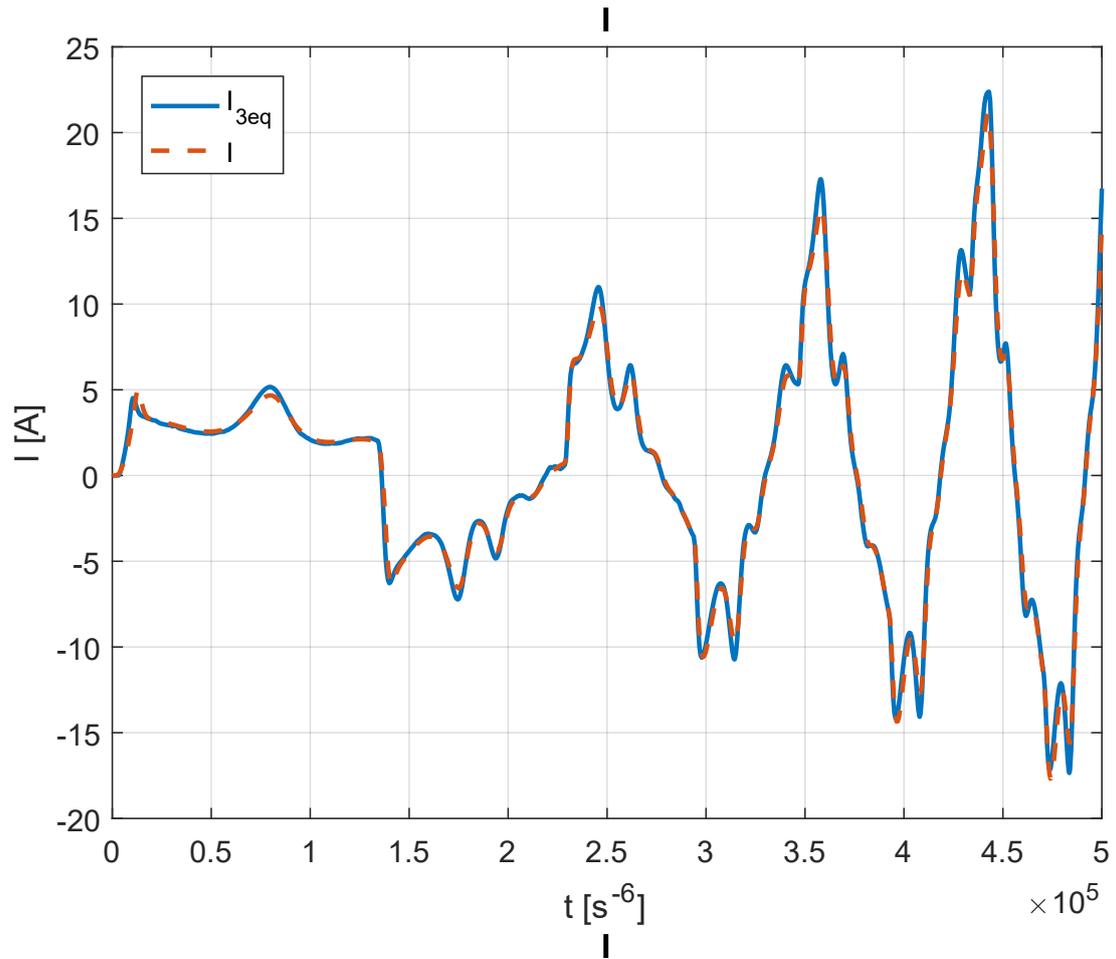


Figure 3.40: Current trends with $[N_A, N_B, N_C] = [1, 0.99, 0.99]$, respectively: before, after optimization, all together

Here the current is basically the same as the one shown into figure 3.28, due to the faults combination very small amount.



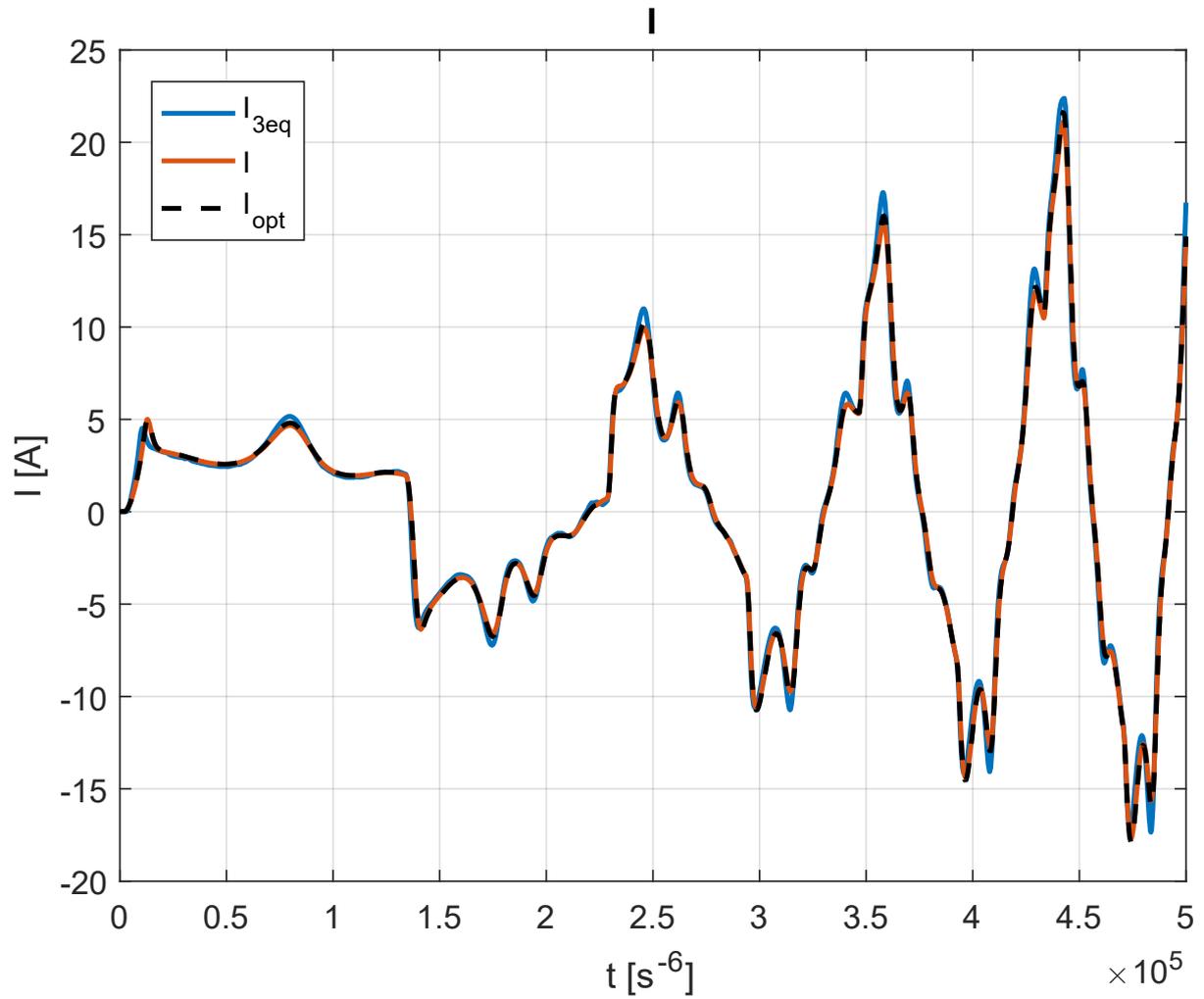
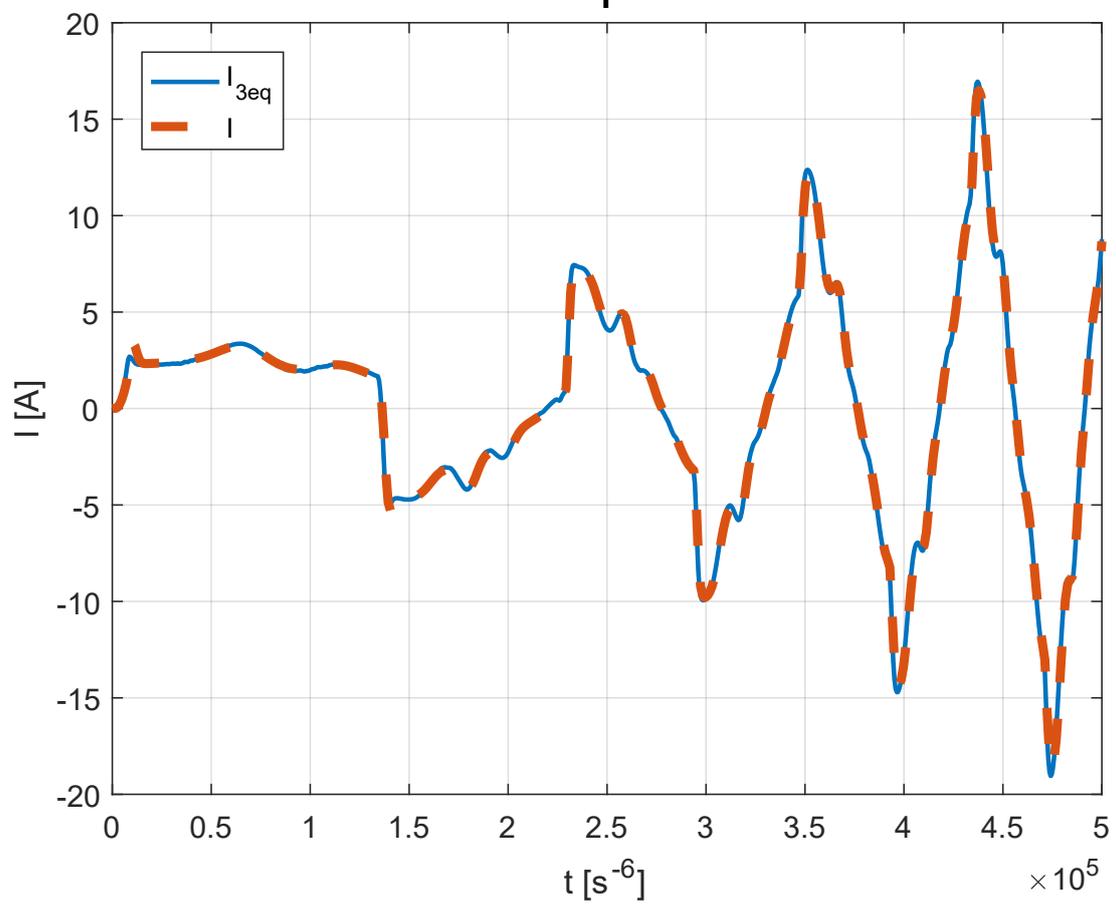
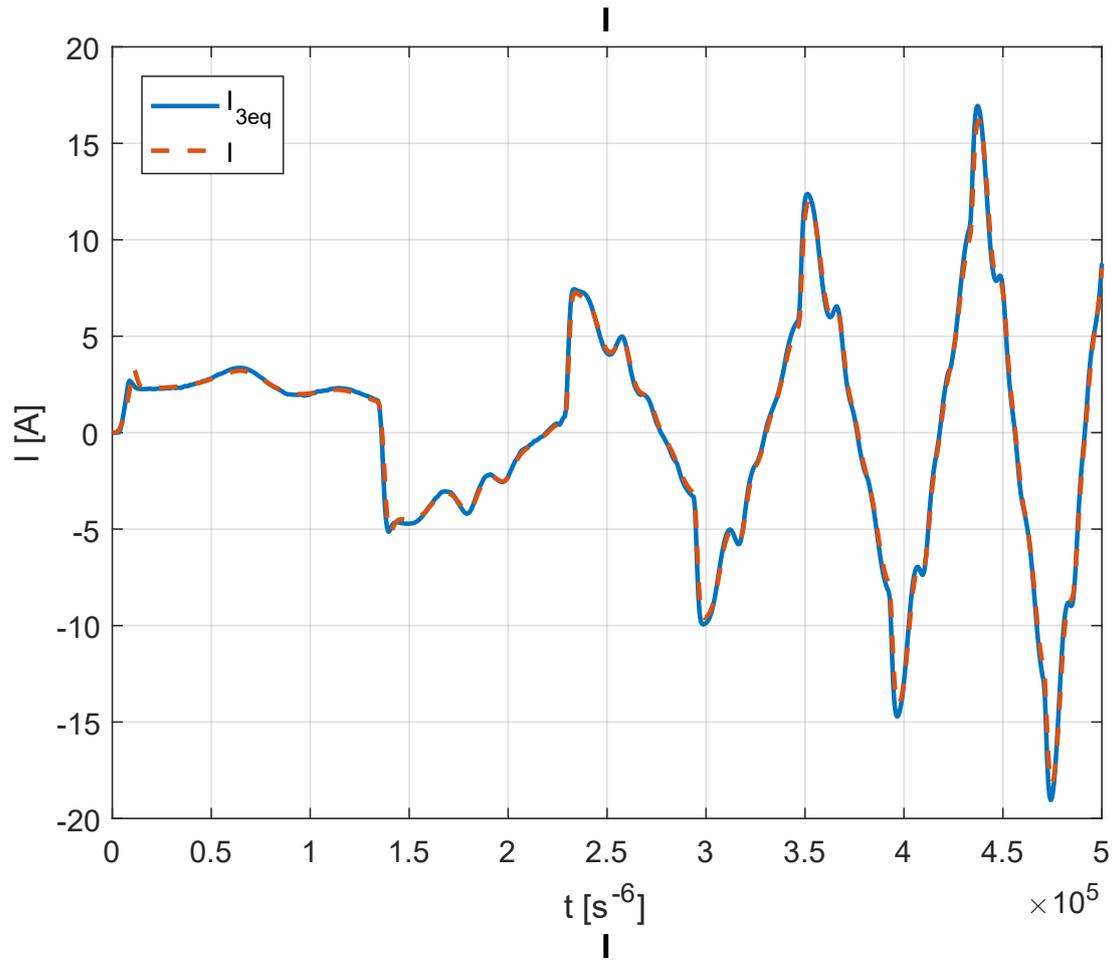


Figure 3.41: Current trends with $[N_A, N_B, N_C] = [0.54, 0.86, 0.17]$, respectively: before, after optimization, all together

That's the current response to fifth faults combination: again, very strong oscillations due to the bad interaction between damaged windings and rotor angular position, even causing the growth in absorbed current amplitude.



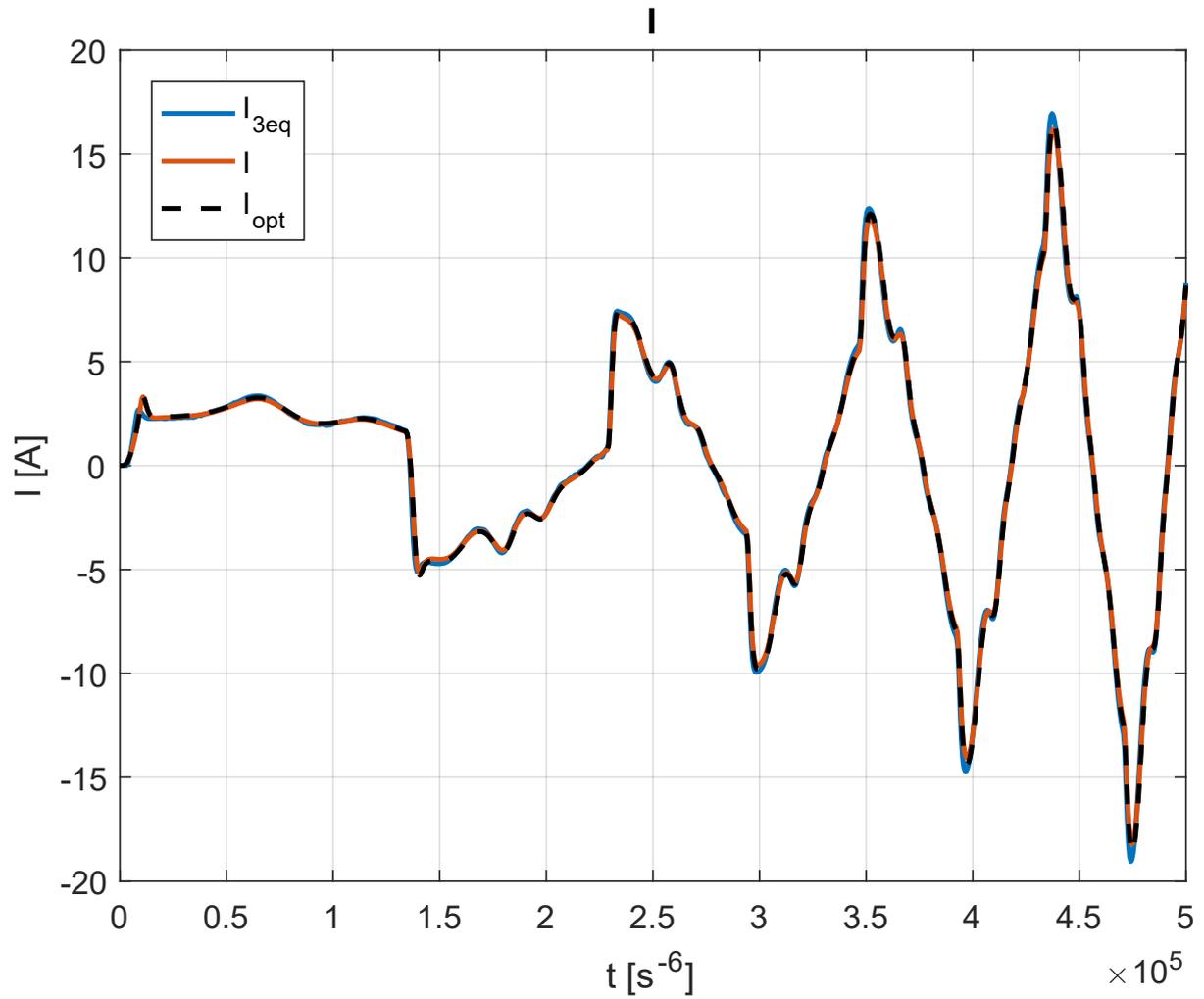
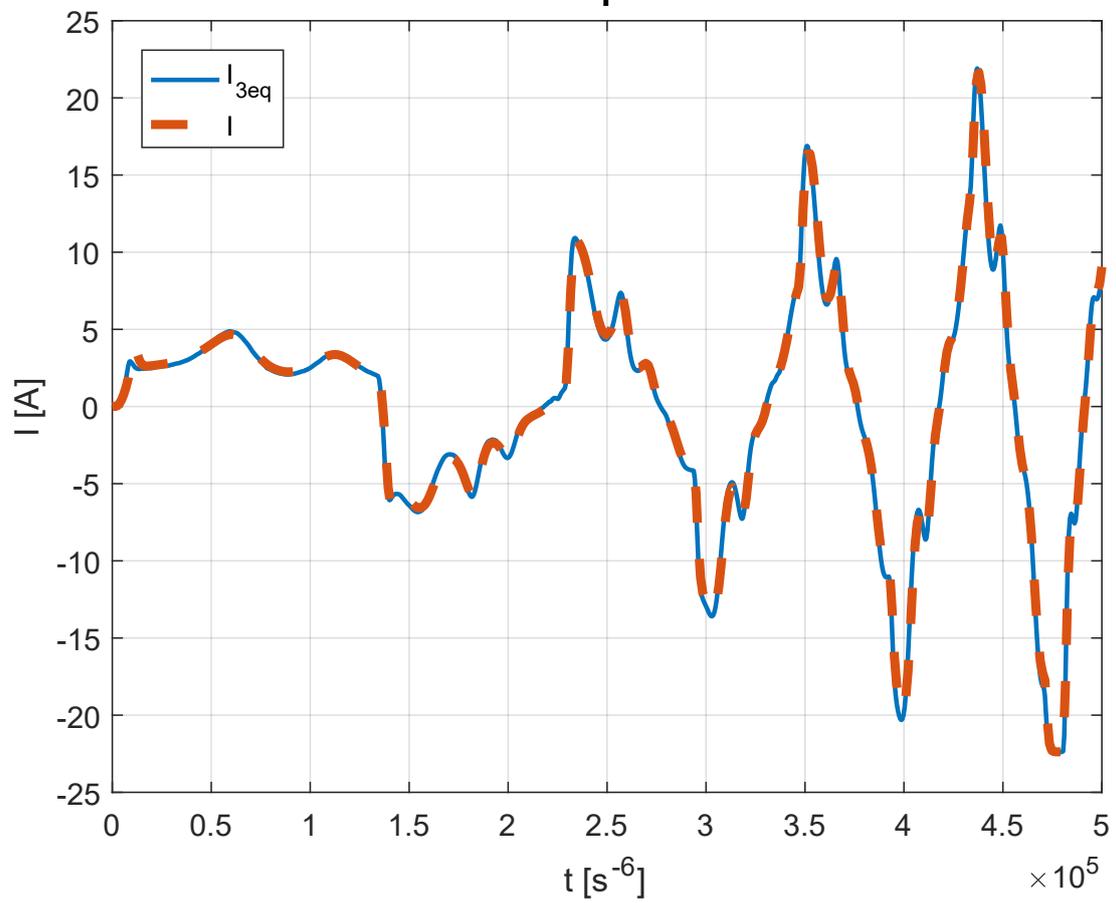
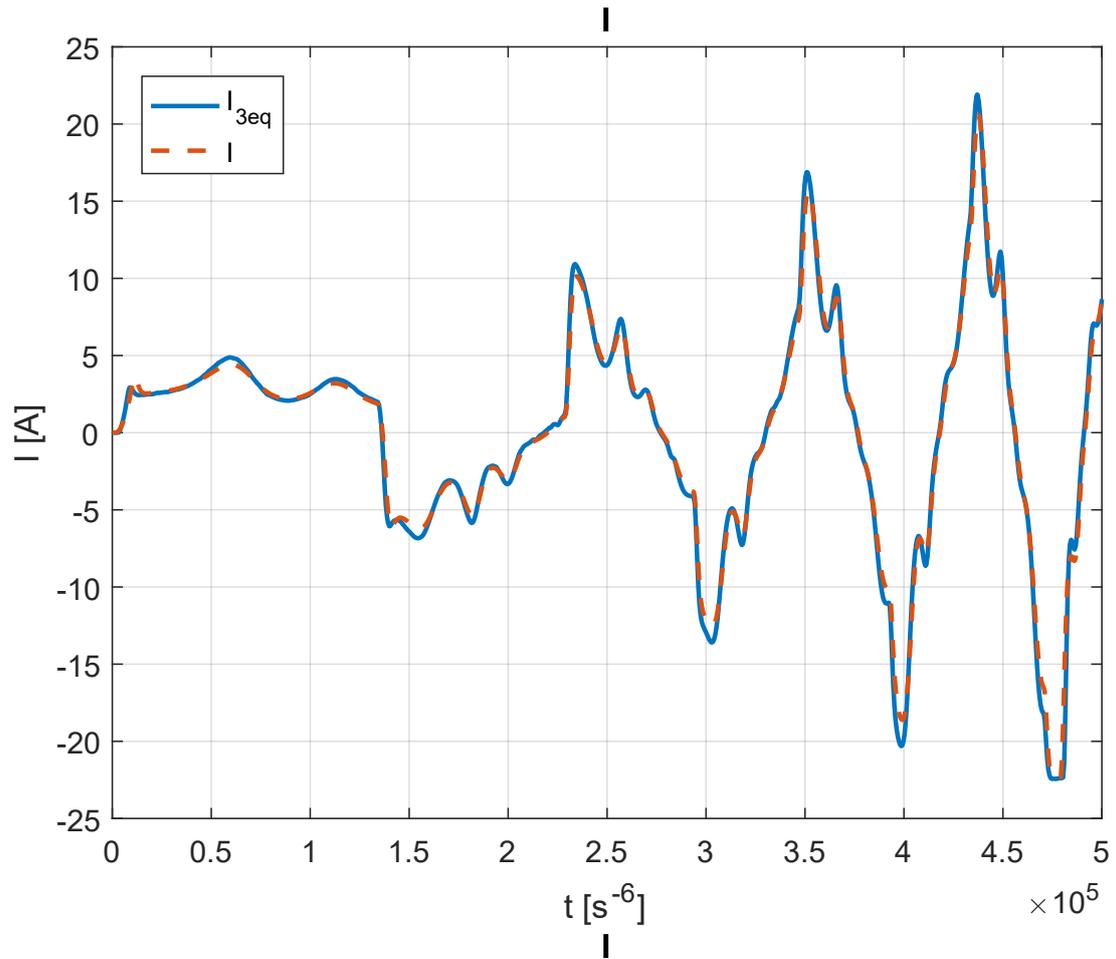


Figure 3.42: Current trends with $[N_A, N_B, N_C] = [0.38, 0.79, 0.79]$, respectively: before, after optimization, all together

These figures show the current wave-form, responding to the set of faults indicated in the sixth row of table 3.3. Due to identical values of the SCs on windings B and C, the trend is very similar to the one reported in figure 3.30.



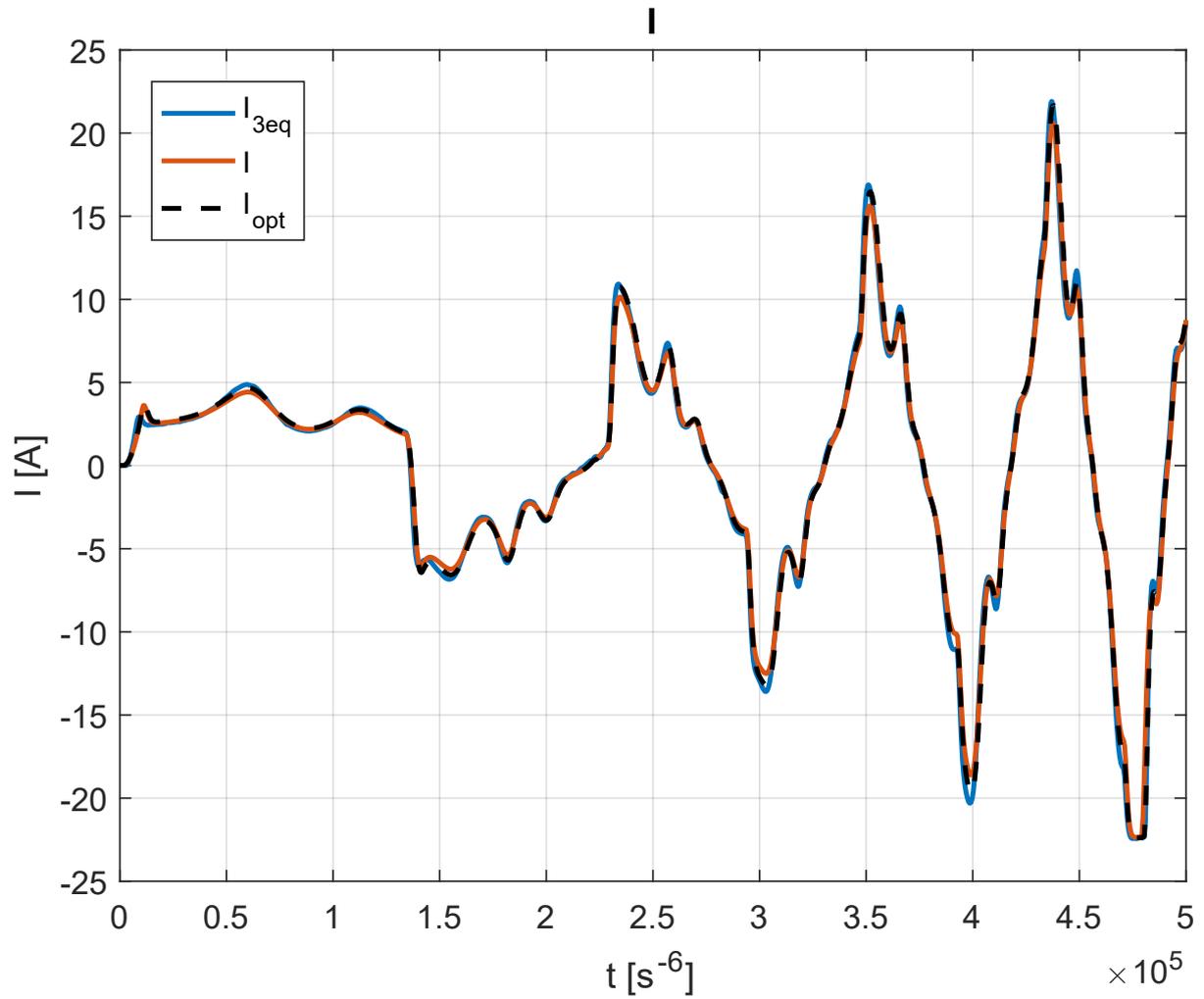


Figure 3.43: Current trends with $[N_A, N_B, N_C] = [0.22, 0.55, 0.91]$, respectively: before, after optimization, all together

For the last faults generation, the strong combination between windings A and B SCs, causes the great growth in terms of currents amplitude shown in figures above, so much to cause the saturation in the very last peak, before the end of simulation. After examining all the responses to the randomly generated short circuits faults combinations, here comes the turn of failure parameters interpolation in respect to each winding faults evolution, remembering the equivalence stated in the equations 3.56:

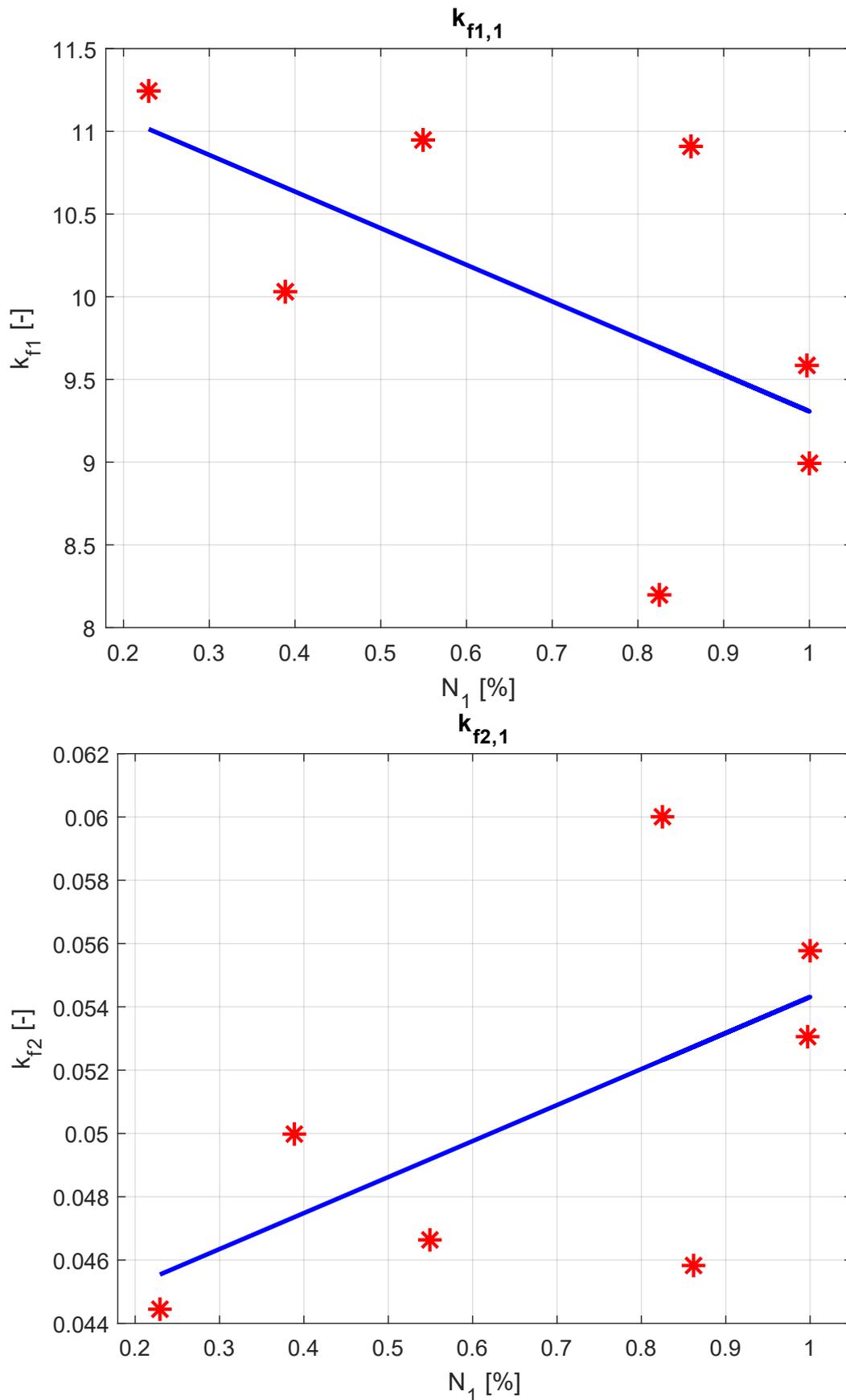
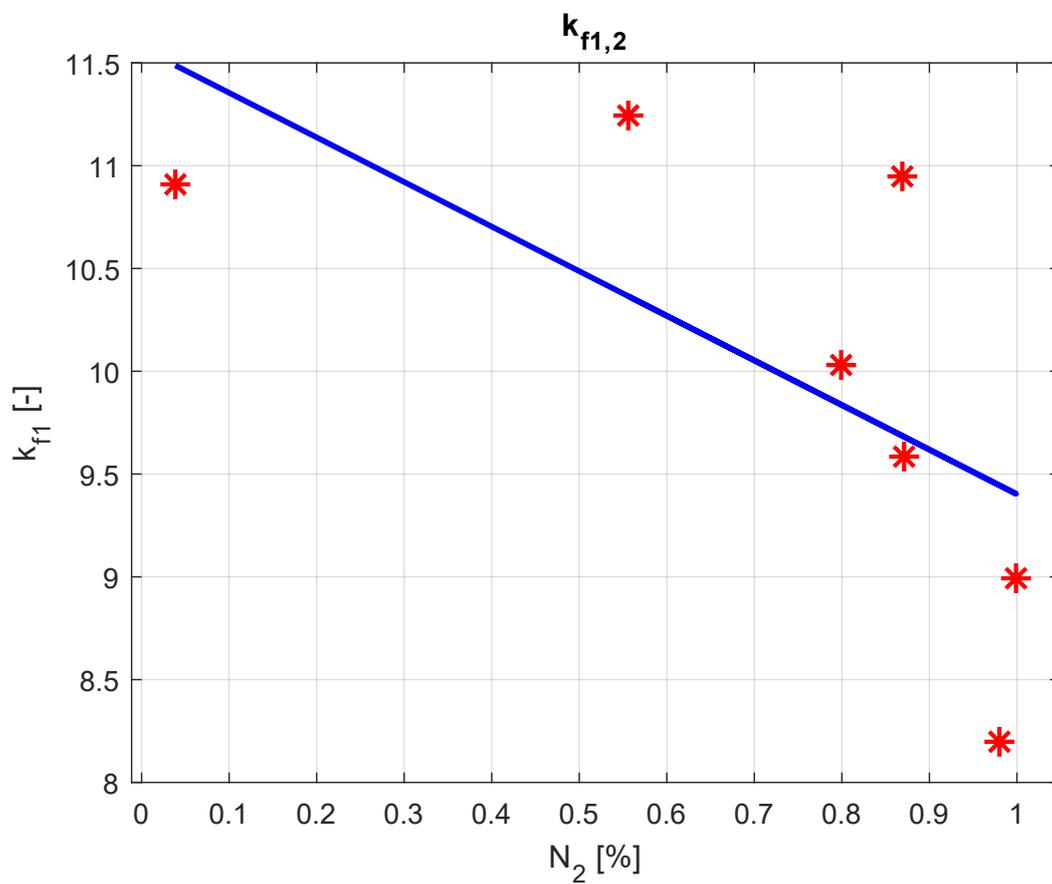


Figure 3.44: K_{f_i} optimized data interpolation, with A winding SC

Here the data fitting with the interpolation line is good, due to a pretty uniform points distribution; to be noticed two values between 0.8 and 0.9 of N_A , being completely different between each other, but

almost not affecting the interpolation line because of their symmetry with respect to it. This behavior, as already mentioned at the section beginning, is due to the influence the other windings faults have on the parameters. However, we can see the trends are perfectly comparable with those given by figures 3.31 and 3.32.



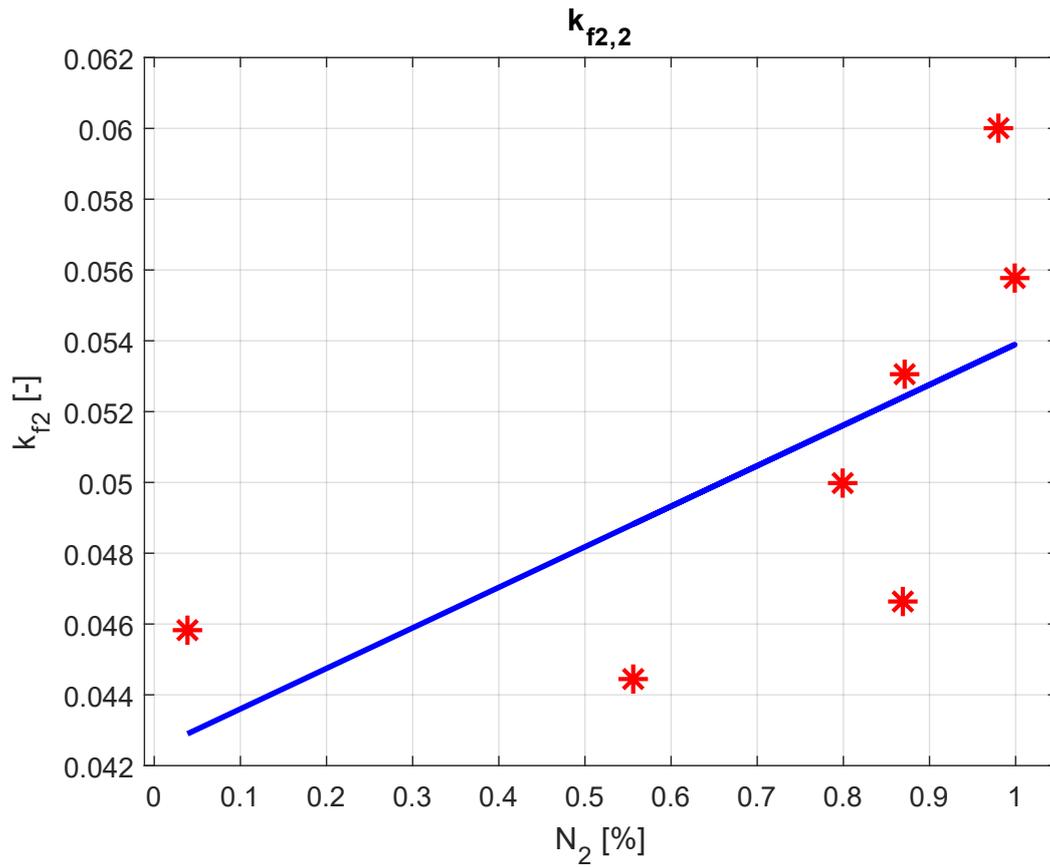


Figure 3.45: K_{f_i} optimized data interpolation, with B winding SC

With the B winding faults data, instead, we can see a certain points concentration near to the nominal conditions, causing a great dispersion, whose cause is to be found, beside the other windings faults conditions, in the fact that the k_{f_i} only have influence when failures occurs, meaning that is reasonable to have different values minimizing the objective function, for very small damages. Nevertheless, the fitting lines are very similar to the others found before.

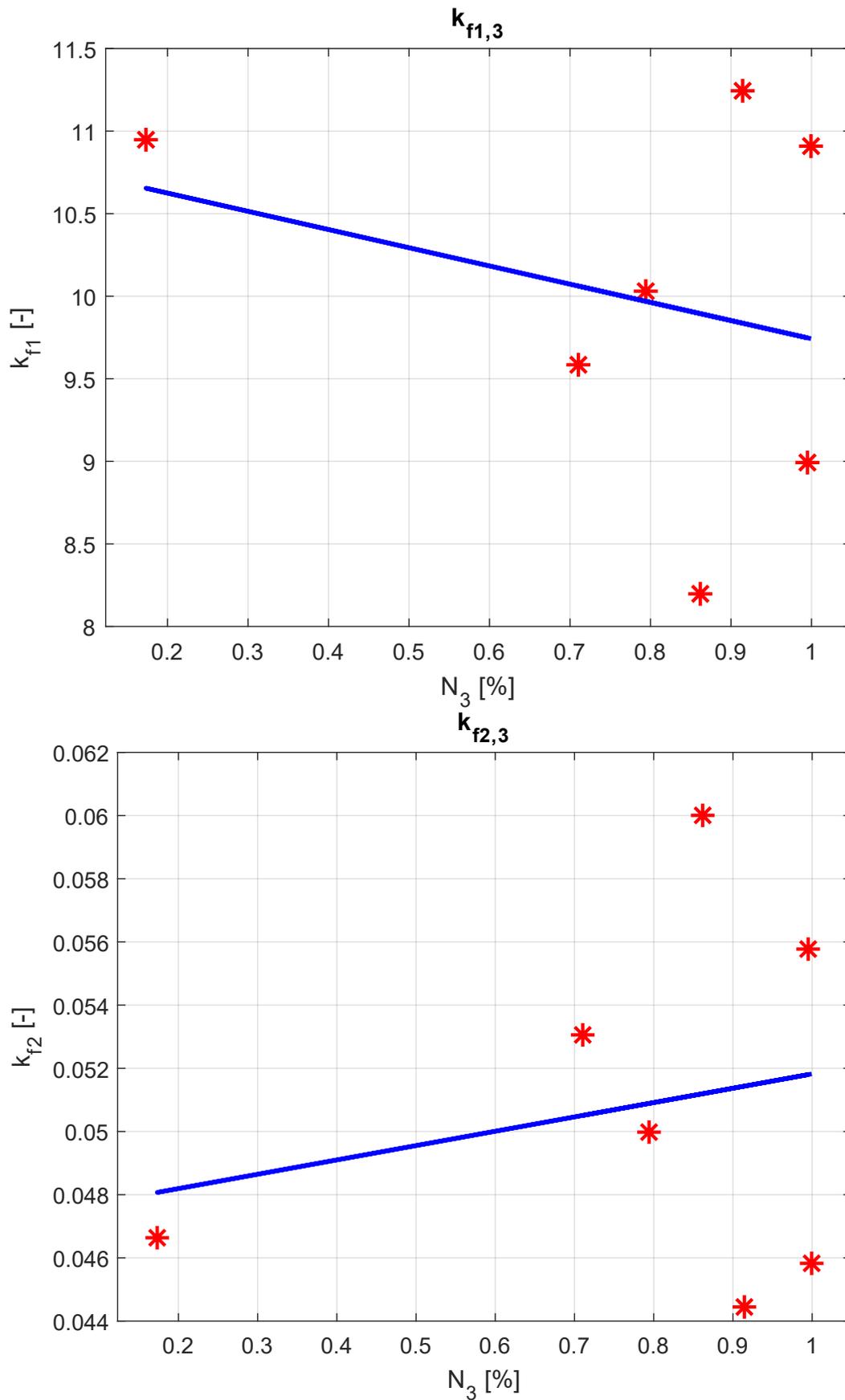


Figure 3.46: K_{f_i} optimized data interpolation, with C winding SC

Here happens the same as with winding B; Furthermore to be considered the number of samplings it's not statistically relevant: a greater number of tests couldn't be run due to the huge amount of time the

GA optimization required to be accomplished. Just think that every iteration took about 8 hours, for a total of around 56 straight hours of running code. That said, this is definitely one of the thing to be done in the future: enhance the number of tests and, if necessary, ensure an isolation between the windings affected by faults.

3.3.3 Multiple Windings Short Circuit and Rotor Eccentricity

The last kind of tests carried out, included multiple windings short circuit along with mechanical failures given by the occurrence of rotor statical eccentricity (see section 3.1.5). In this case, the faults randomly generated, have been treated as follows:

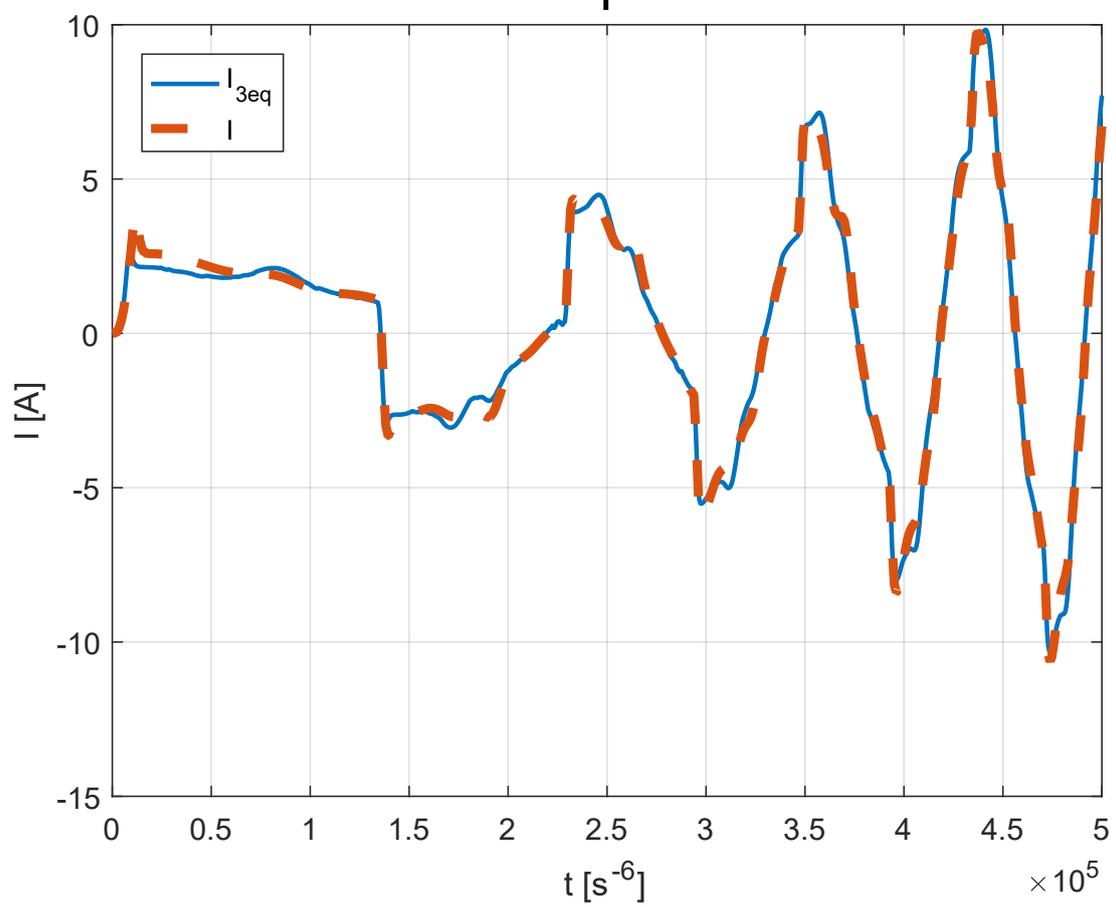
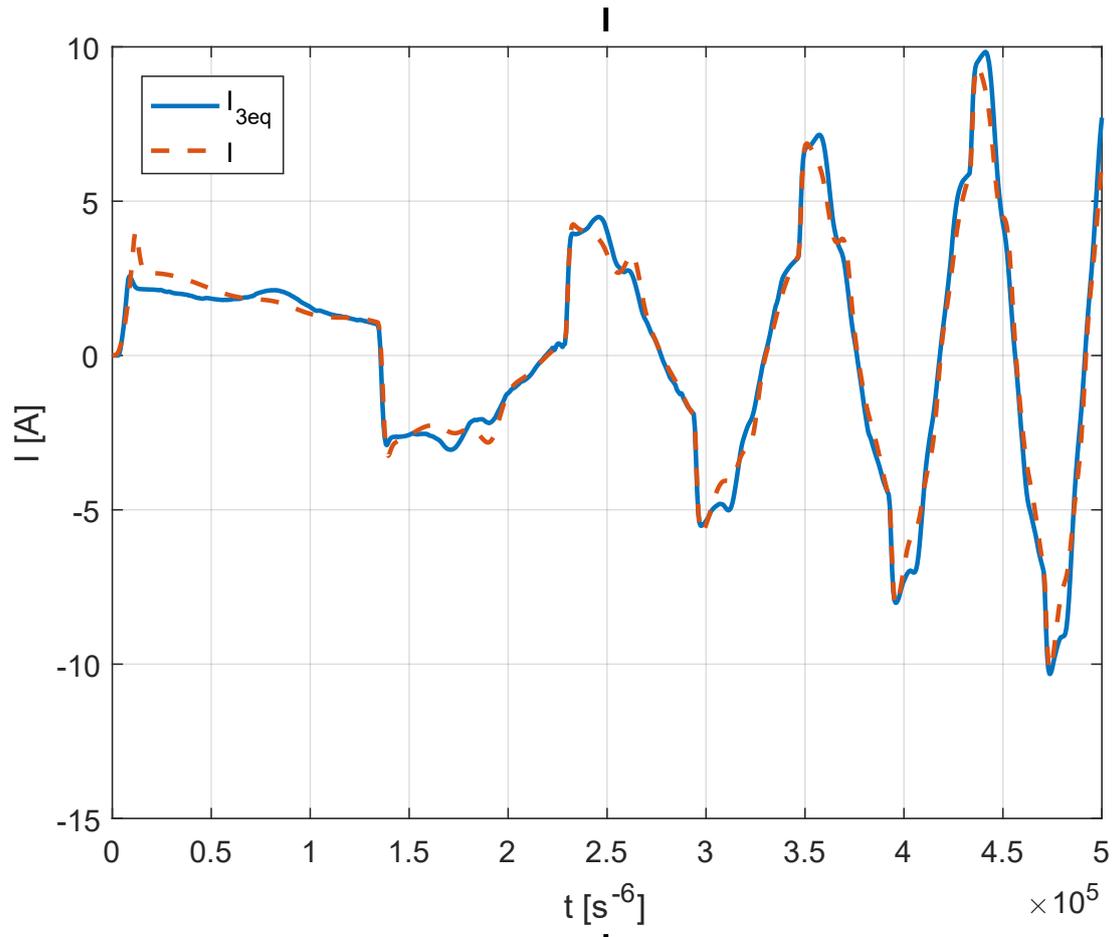
$$\begin{cases} faults = rand(5)^5 \\ [N_A, N_B, N_C] = 1 - faults(1 : 3) \\ [Z, \phi_Z] = faults(4, 5) \end{cases} \quad (3.58)$$

And we have one more degree of freedom to add to minimization problem of the objective function; indeed, if for the previous tests the parameters to be evaluated have been just those related with the SC (k_{f_s} and k_{f_t}), now even k_{f_E} needs to be involved. This will for sure put under stress the GA, which has to handle three parameters variation and, due to this, we would not be surprised if some results would not produce a perfect current matching, or a perfect interpolation path. Indeed, seen the larger variables number involved, we expect to find a data dispersion at least equal to that seen in the previous section. Let's take a look at following table, containing the relevant data dealing with the simulations in question:

N_A [%]	N_B [%]	N_C [%]	Z [%]	ϕ_Z [%]	k_{f_s} [/]	k_{f_t} [/]	k_{f_E} [/]	$MSE_{f_{bef}}$ [A ²]	$MSE_{f_{opt}}$ [A ²]
99.98	97.27	68.28	51.57	0.07	8.8925	0.0543	0.3150	2.0401	1.1189
99.32	98.43	88.90	5.04	42.90	6.8198	0.0693	0.3150	0.0803	0.0717
90.42	98.22	99.22	9.35	0.03	6.7648	0.0692	0.3150	0.1100	0.0847
71.09	99.73	64.79	6.97	0.06	11.0662	0.0465	0.3155	0.1986	0.1605
100.00	96.60	95.71	0.04	0.01	7.0863	0.0675	0.3469	0.0514	0.0496
30.66	96.52	99.47	0.25	0.06	10.7502	0.0462	0.5250	0.7435	0.3896
71.91	63.46	27.00	2.32	1.57	9.7322	0.0522	0.3152	0.7939	0.7369

Table 3.4: Windings A, B, C SC and E data summary

Even in these tests, the MSE_f values, except for few cases, always remains pretty low and with a small variation between before and after optimization. Here follow the currents graphs, referred to data reported in the table above.



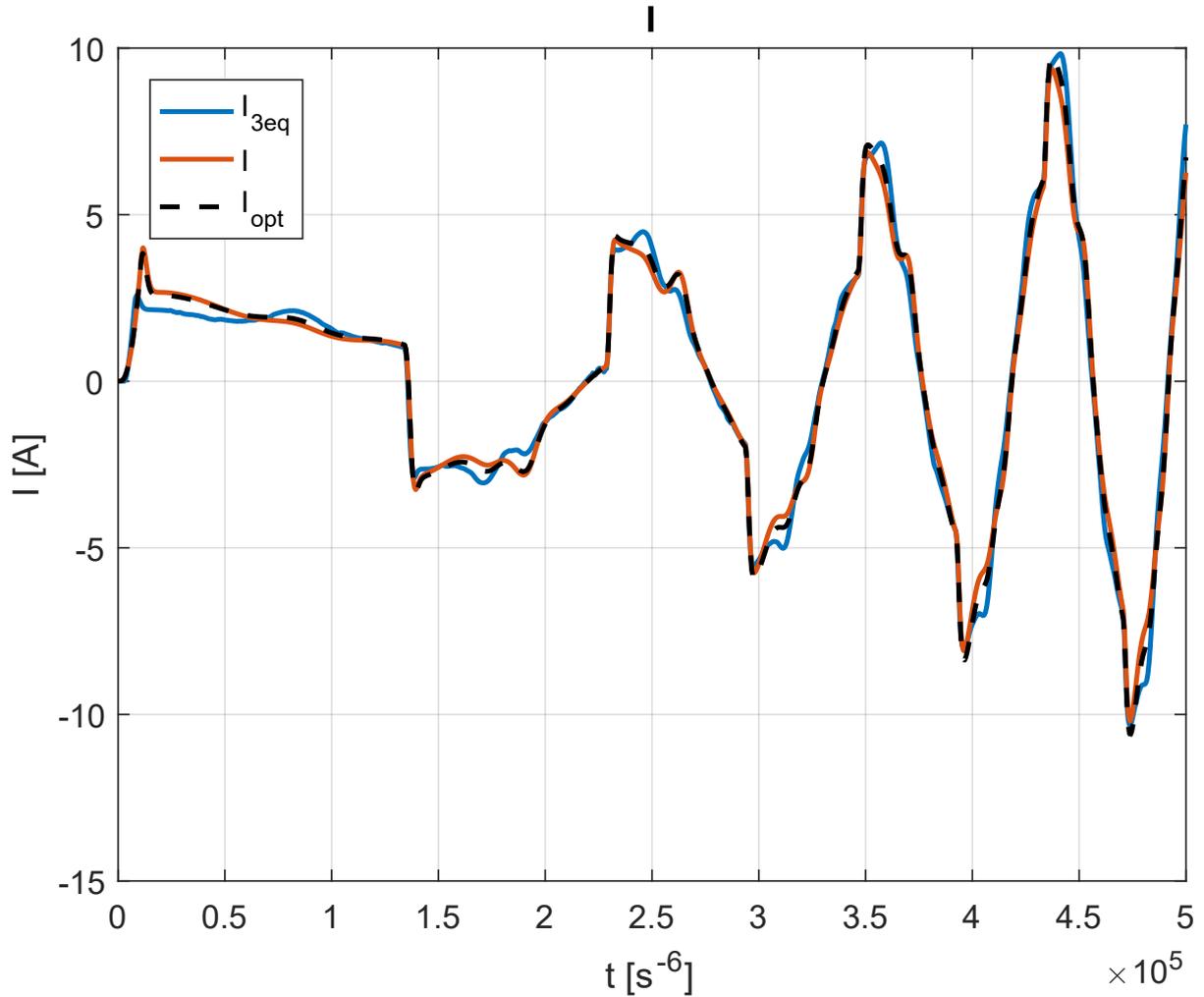
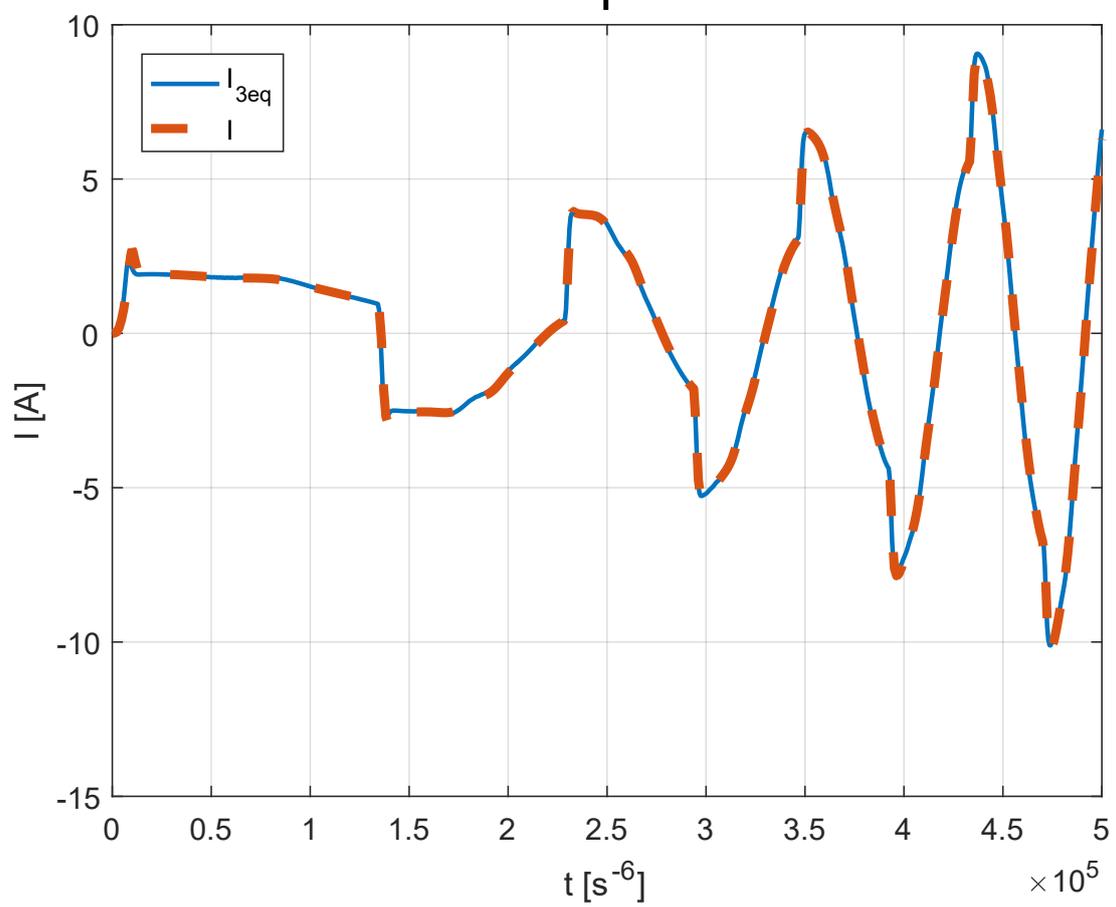
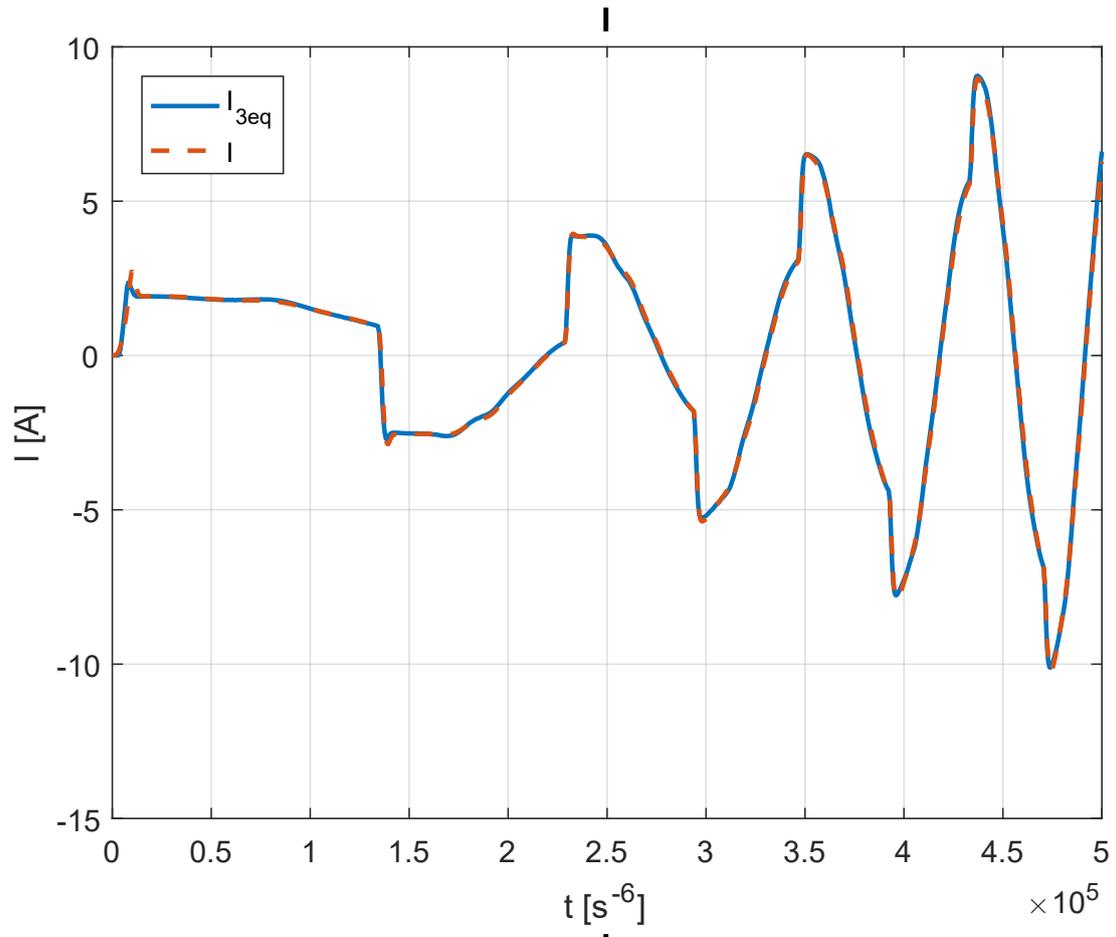


Figure 3.47: Current trends with $[N_A, N_B, N_C] = [0.99, 0.97, 0.68]$ and $[Z, \phi_Z] = [0.51, 0.0007]$ respectively: before, after optimization, all together

The one reported in the figures above, is definitely the worst matching we've been through among all the cases. We can see that, in terms of amplitude there is a good correspondence, not the same in terms of oscillations. This could depend upon many factors: the most relevant is the huge eccentricity entity, indeed, we can see $Z = 0.51$; in this conditions, the actuation is completely compromised, therefore this event goes beyond the field of applicability this work deals with. Anyway it's important to know where the monitoring model limits are, thus making relevant this kind of results too. The second reason that could have caused this kind of mismatching is a bad optimization result: for example the GA could have slipped into a local minimum, or simply, for the reasons explained before, the interaction between the parameters is so strong that that's the best result possible. All of this said, in order to further investigate this situation, more than one optimization cycle on the same failures conditions would have been required, but, due to a lack of time, this is one of the things to be done in the future developments. Anyway, there was a great improvement in terms of MSE_f , as can be seen in the table 3.4.



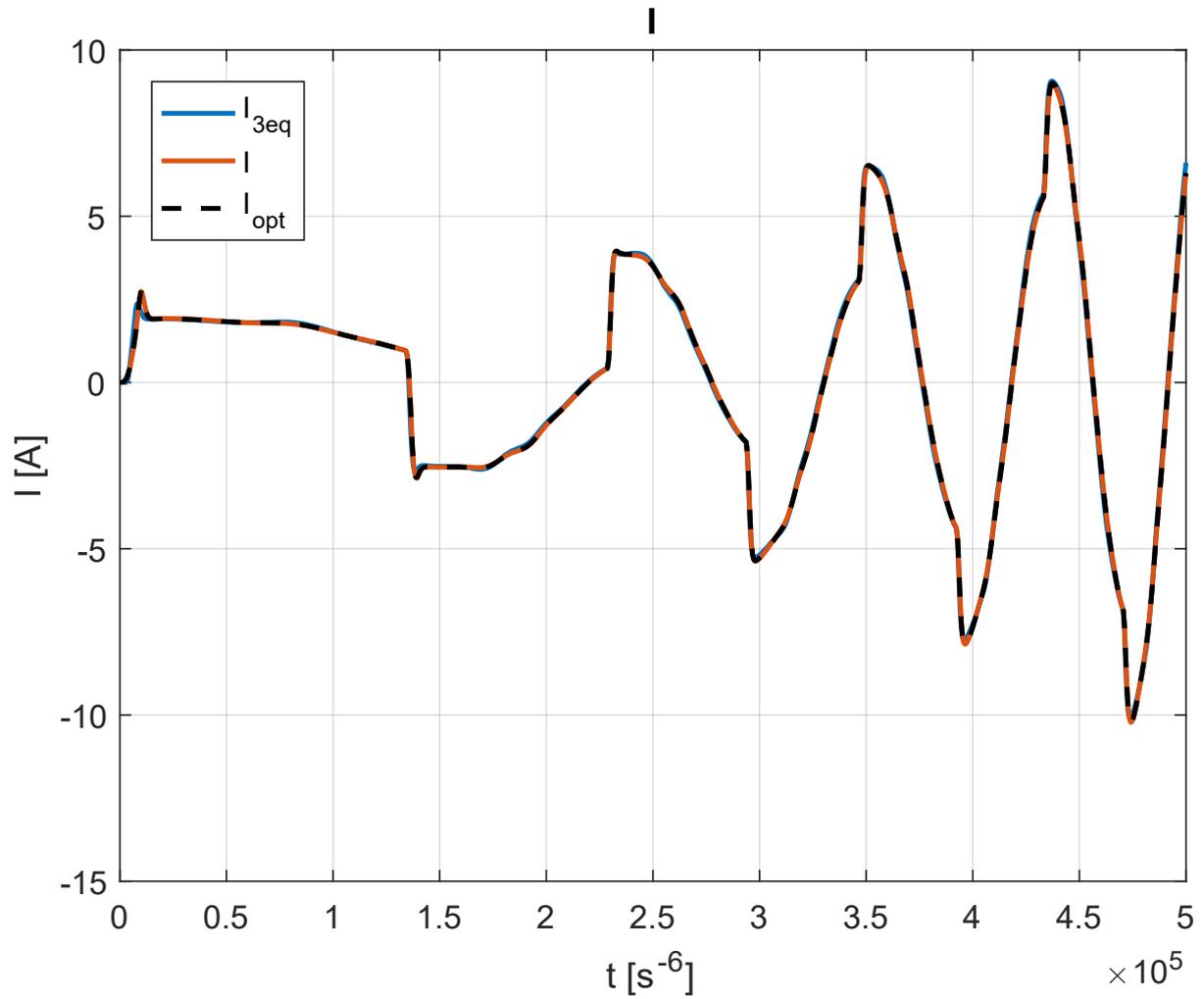
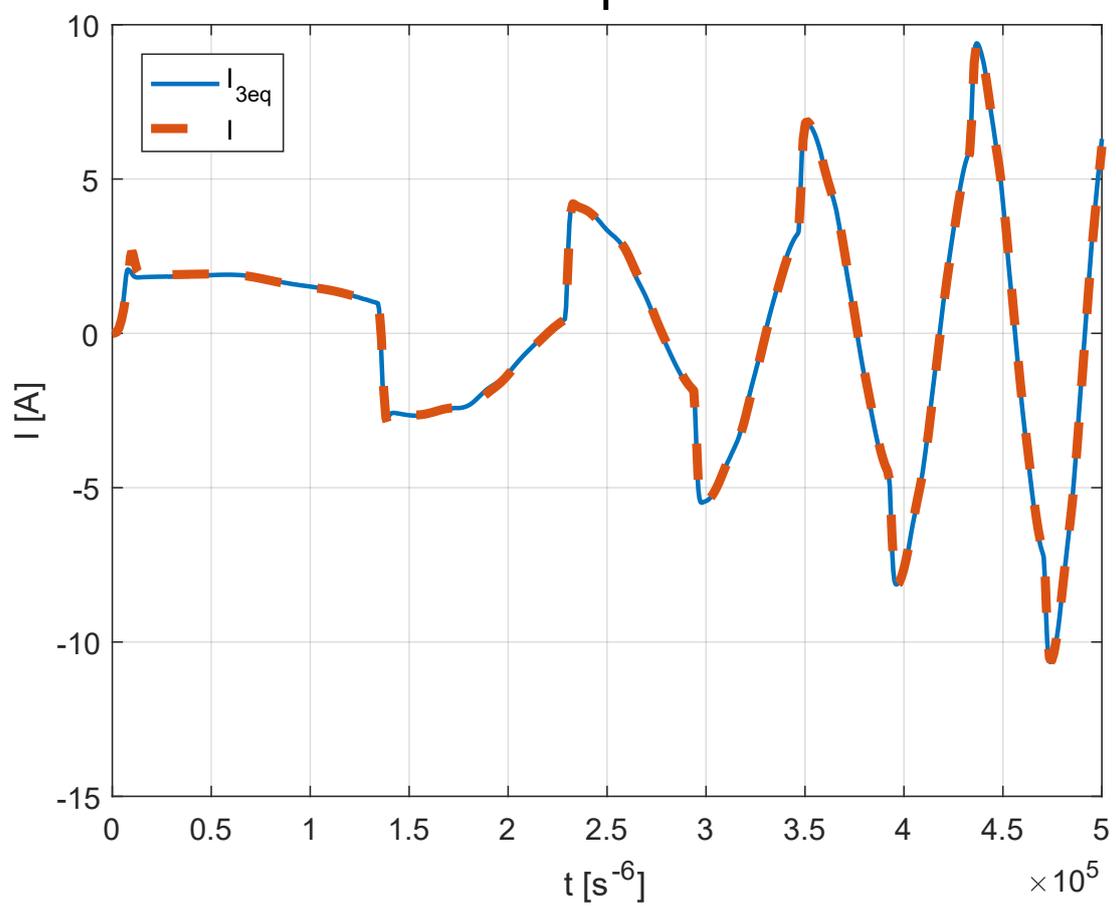
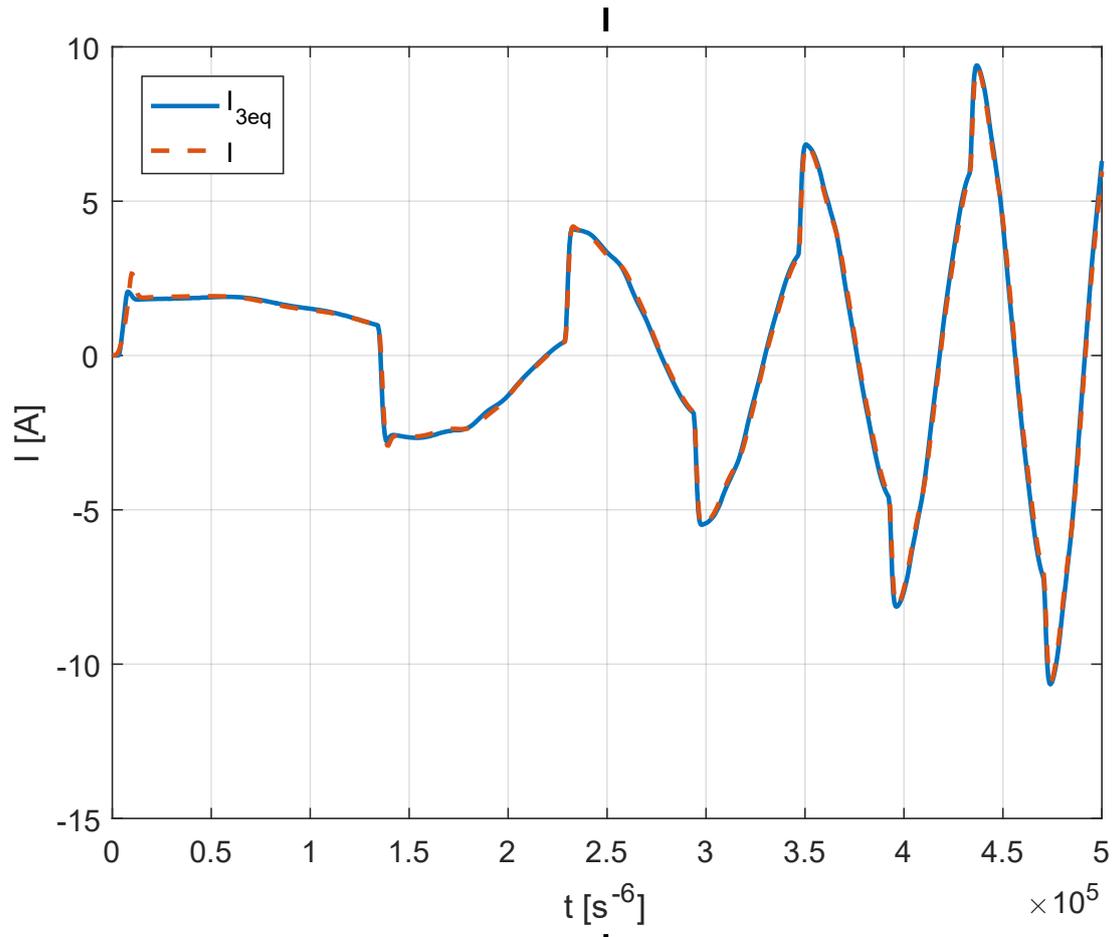


Figure 3.48: Current trends with $[N_A, N_B, N_C] = [0.99, 0.98, 0.88]$ and $[Z, \phi_Z] = [0.05, 0.42]$ respectively: before, after optimization, all together

The second set of faults, due to their very small entity, really little affects the current trend, thus giving the chance to a perfect matching. Anyway, there are a few oscillations visible and, that, exactly represents the kind of failures to be relevant for this work: for sure, nothing is going to be notable in terms of actuation, but the faults are there and, through the current output evaluations, we are able to detect them and, eventually, make the situation safe.



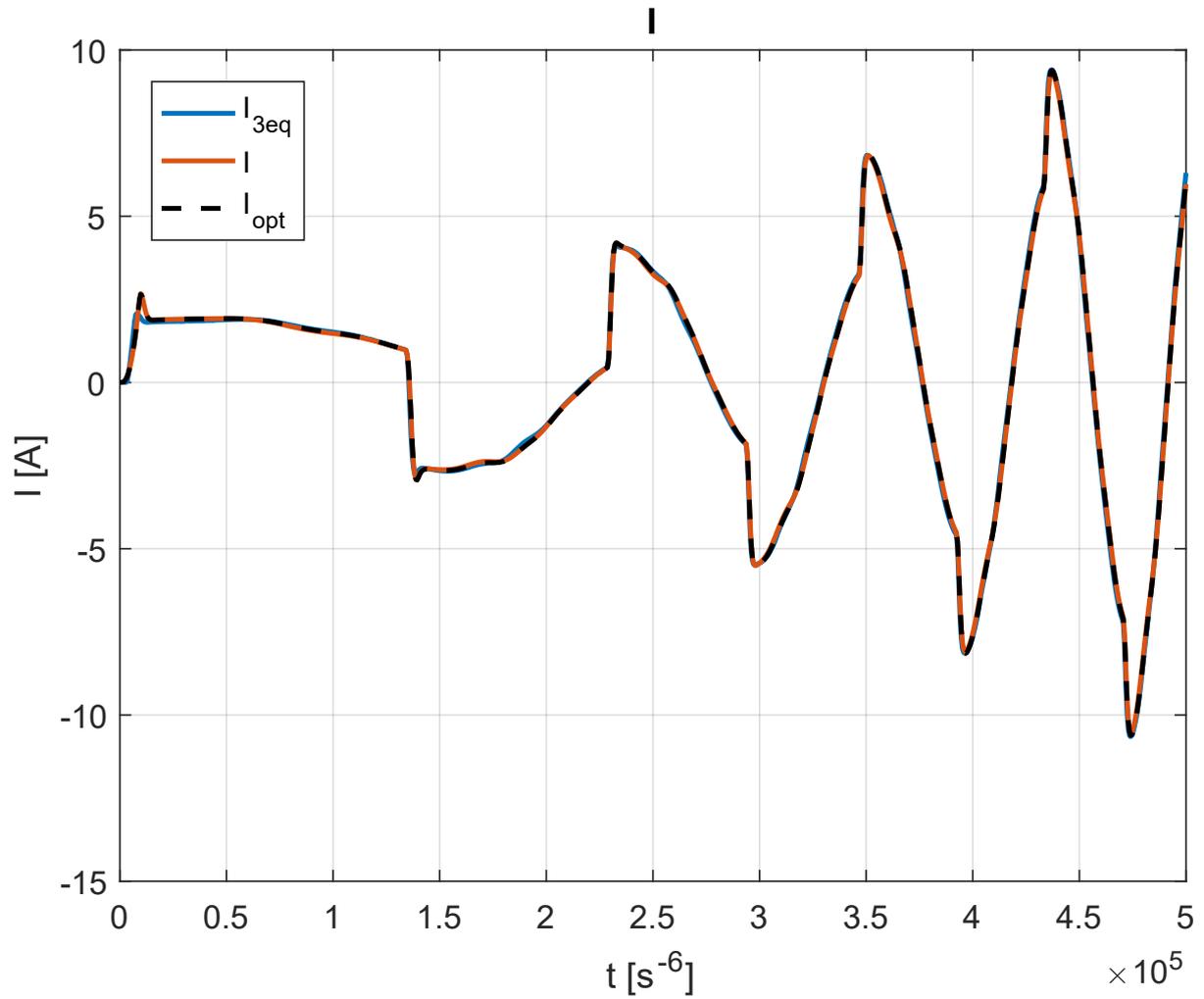
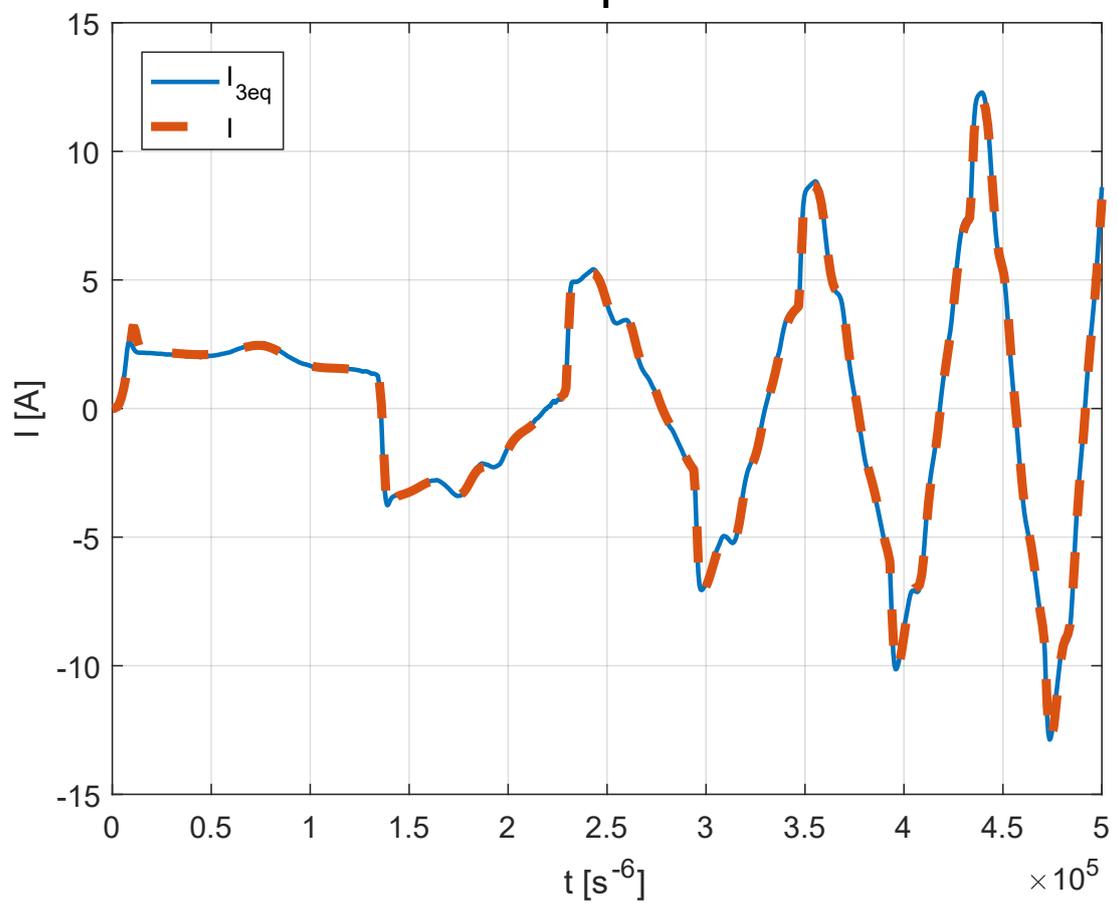
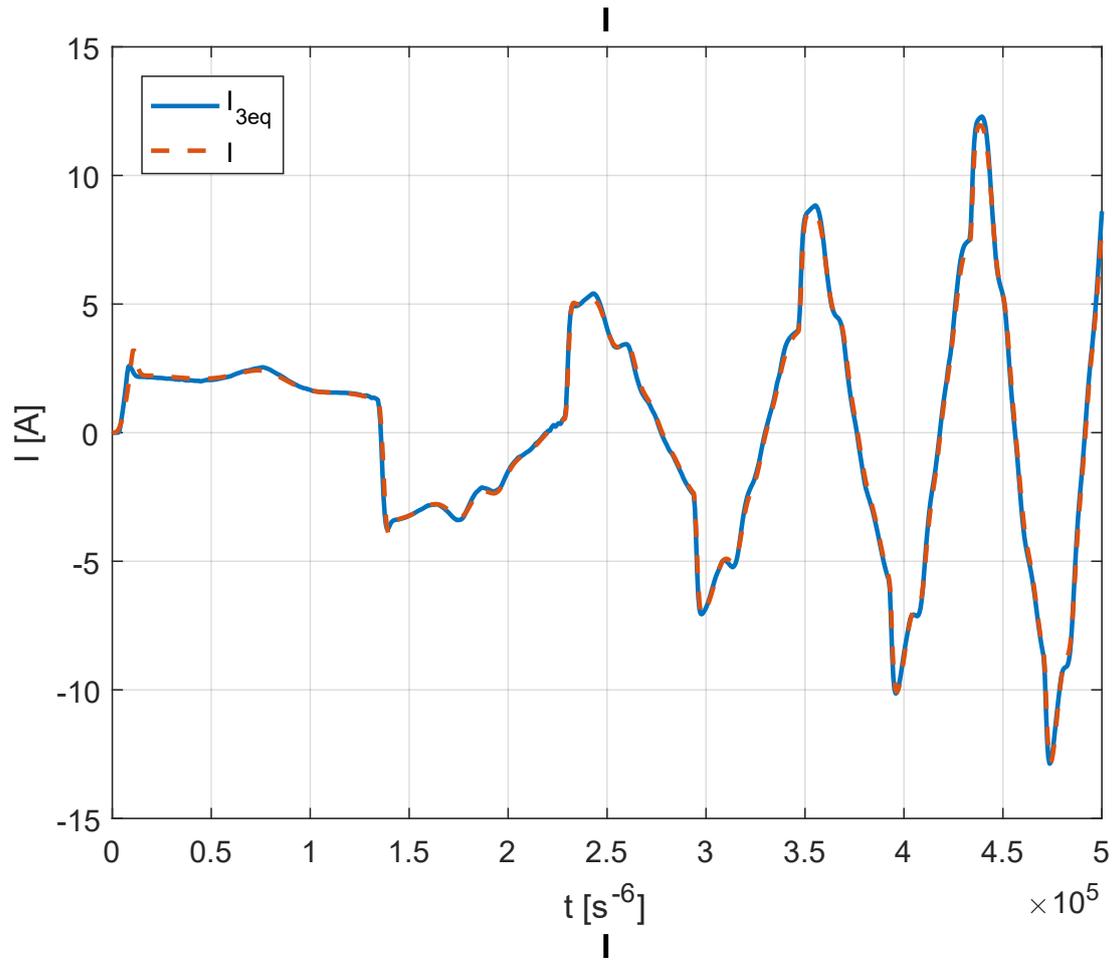


Figure 3.49: Current trends with $[N_A, N_B, N_C] = [0.90, 0.98, 0.99]$ and $[Z, \phi_Z] = [0.09, 0.0003]$ respectively: before, after optimization, all together

Here the situation is basically the same as the previous one, given the almost identical faults values, as we can see from the figures notes: very slight oscillations and perfect matching between the curves.



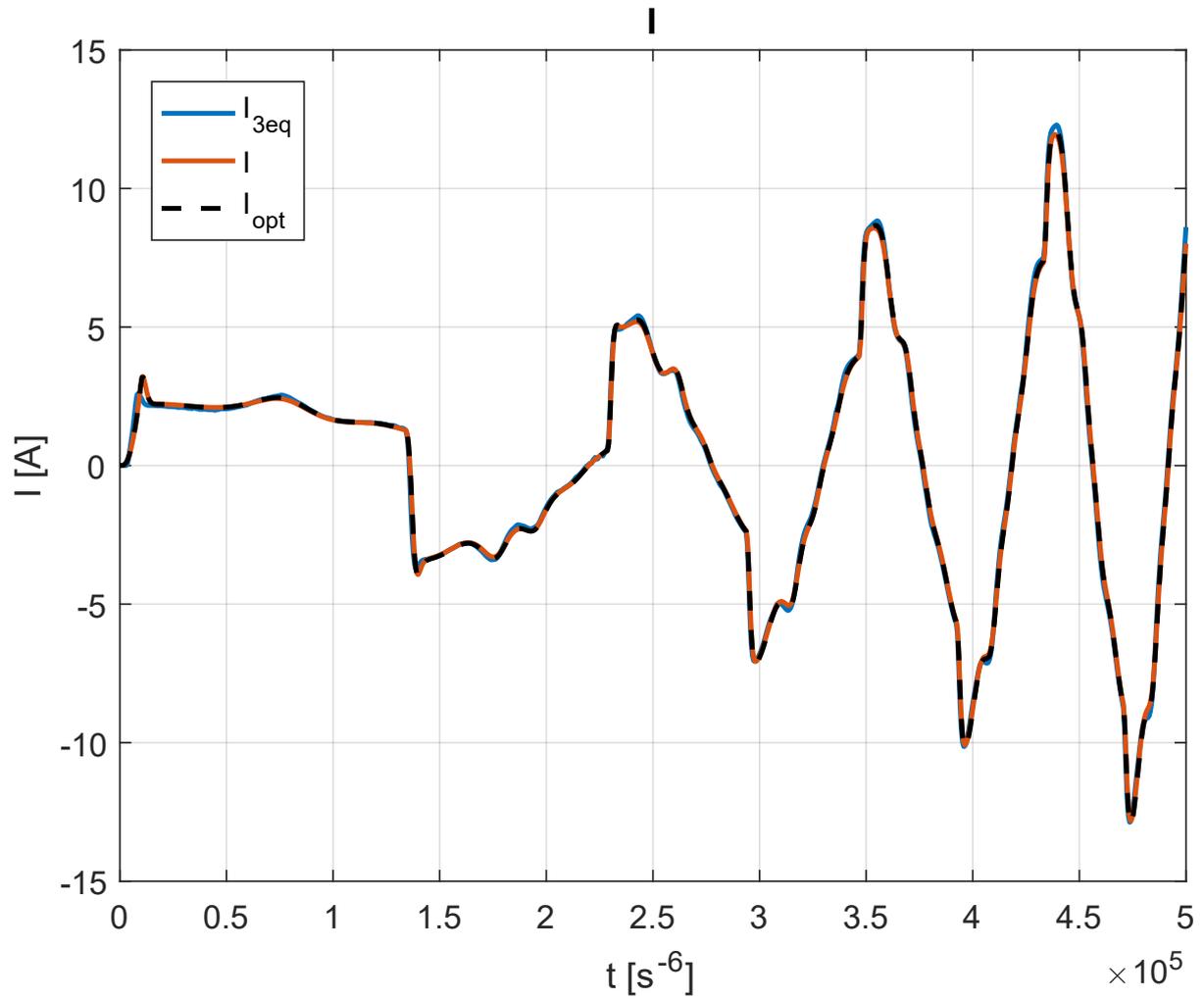
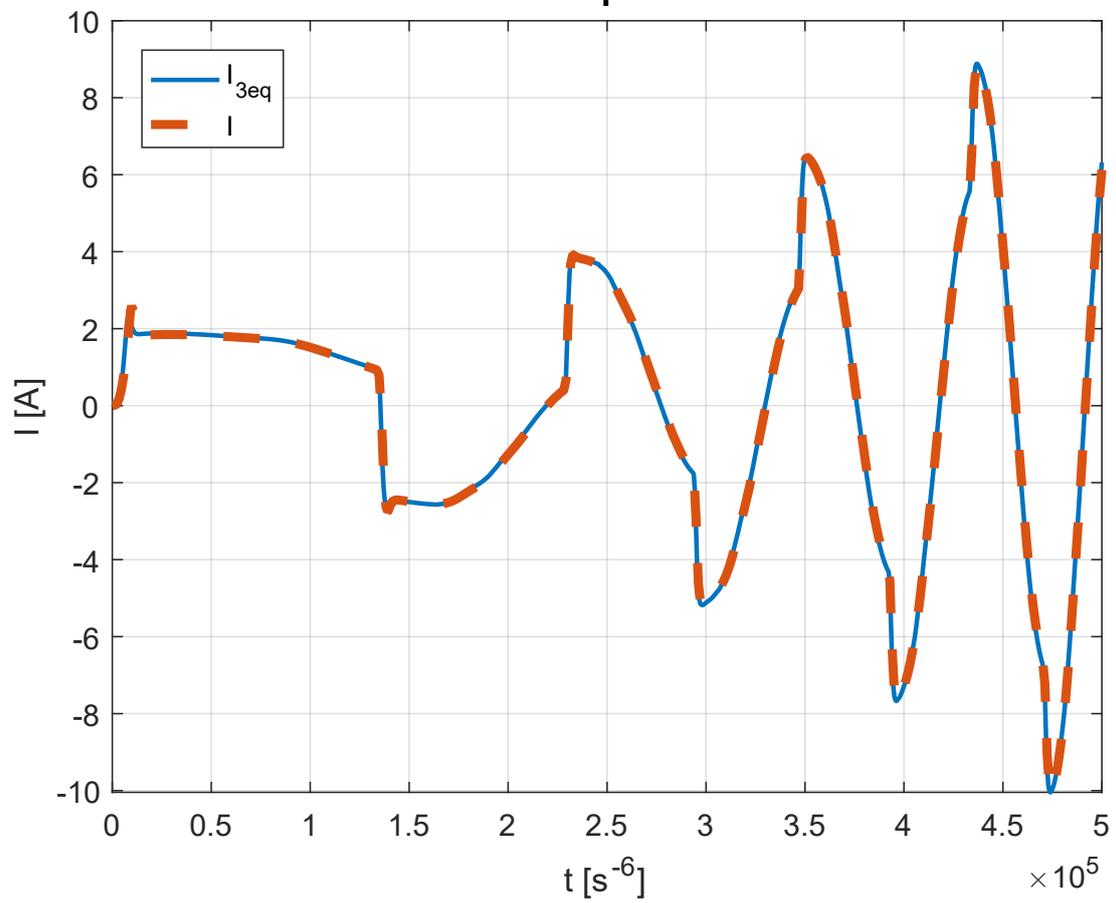
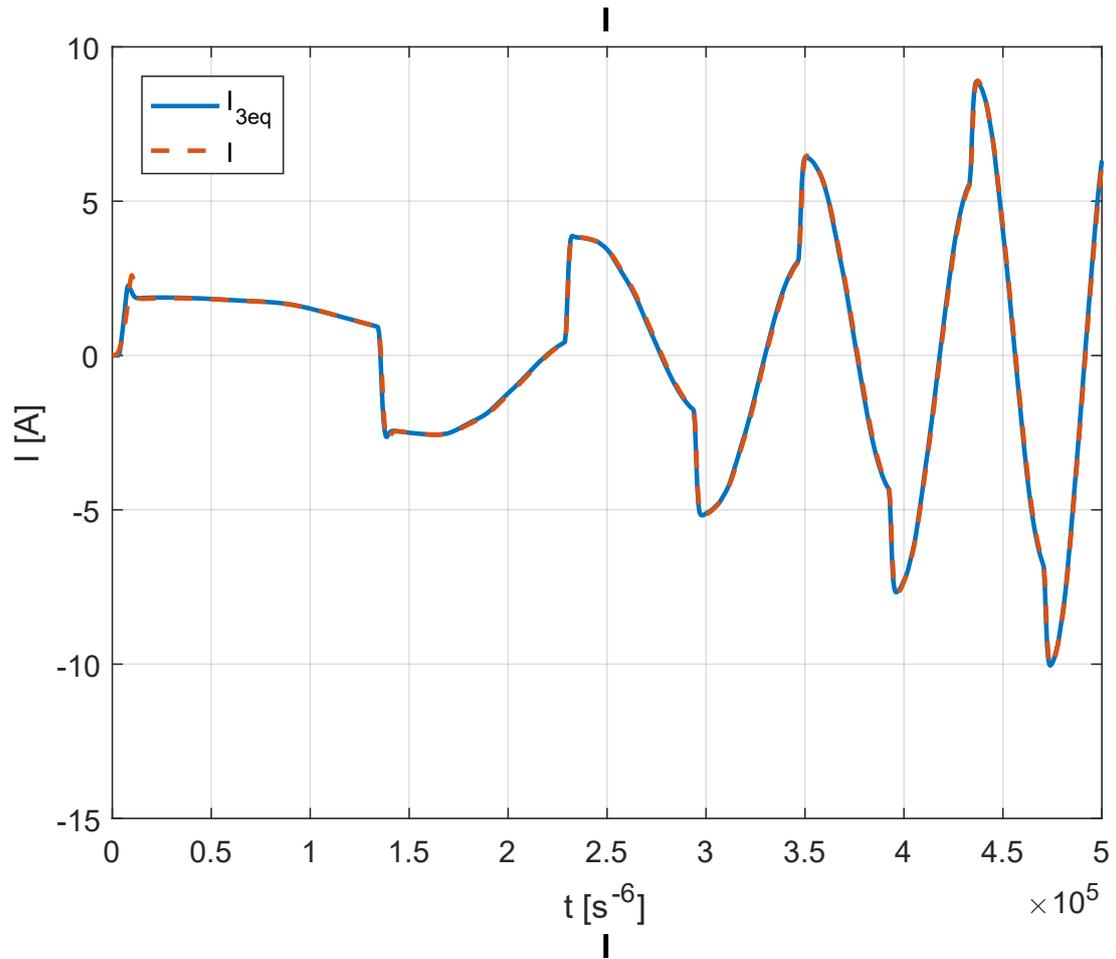


Figure 3.50: Current trends with $[N_A, N_B, N_C] = [0.71, 0.99, 0.64]$ and $[Z, \phi_Z] = [0.06, 0.0006]$ respectively: before, after optimization, all together

With the fourth set of faults, seen the higher SCs percentage, we find the usual oscillation and notice the growth of current magnitude; even in this case, the matching between the curves is almost perfect.



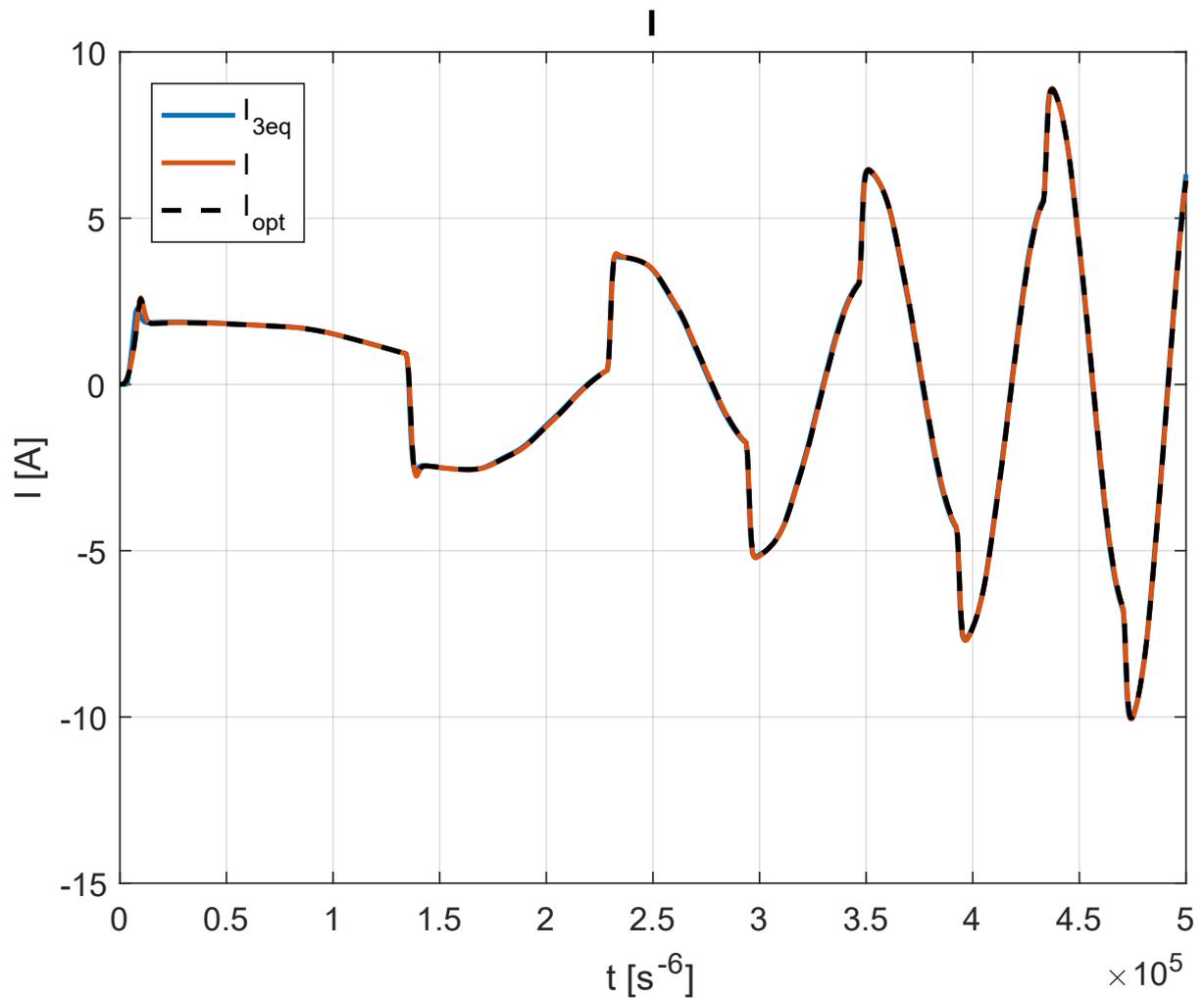
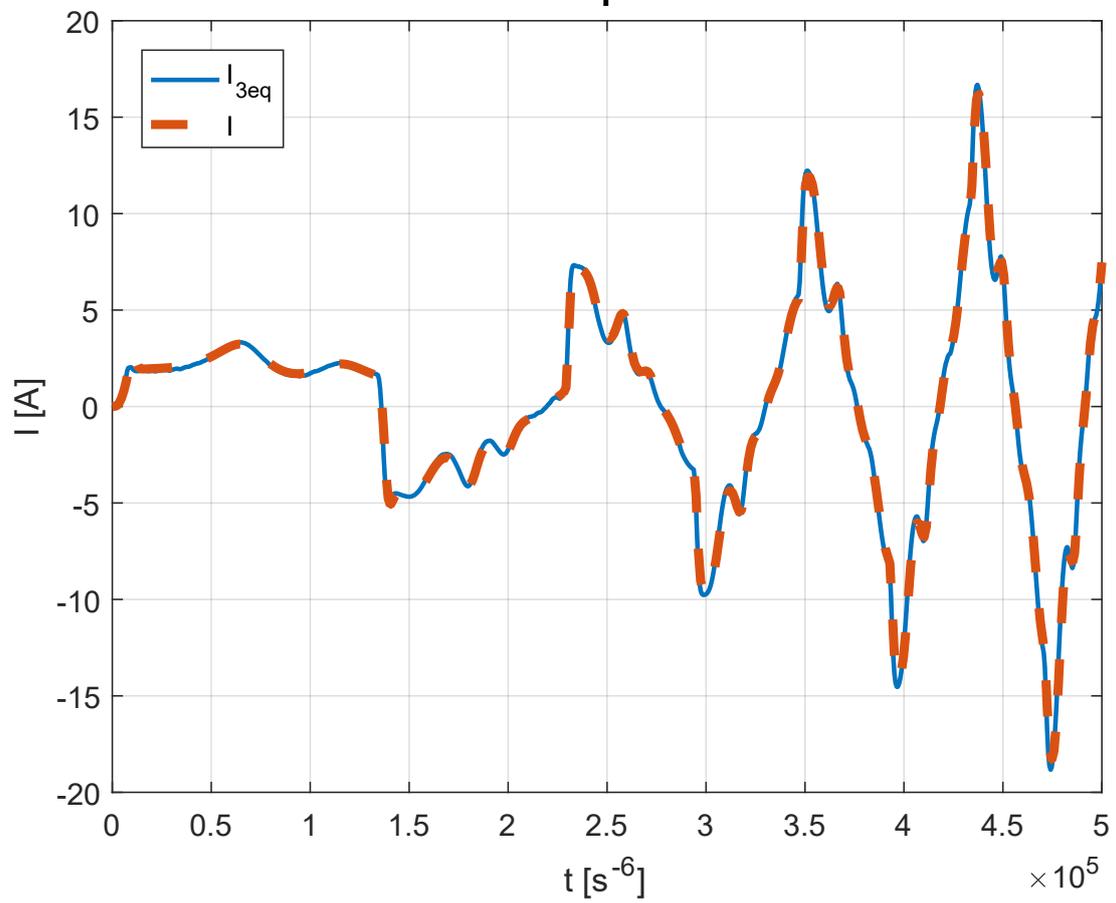
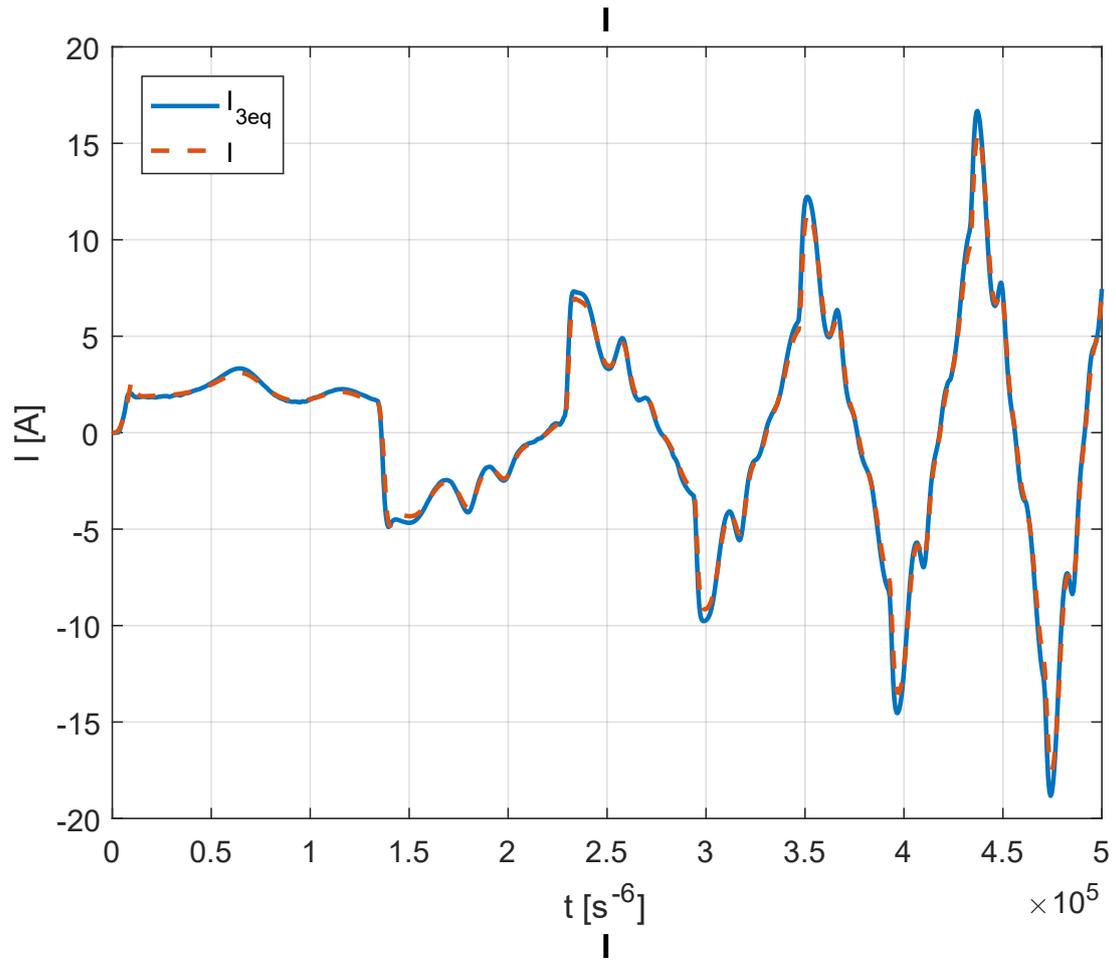


Figure 3.51: Current trends with $[N_A, N_B, N_C] = [1, 0.96, 0.95]$ and $[Z, \phi_Z] = [0.0004, 0.0001]$ respectively: before, after optimization, all together

The conditions reported above are basically the nominal ones. Indeed, beyond for the fact that the objective function is defined in a different way, even the MSE_f has a value very similar to the one obtained with the calibration process (see section 3.2).



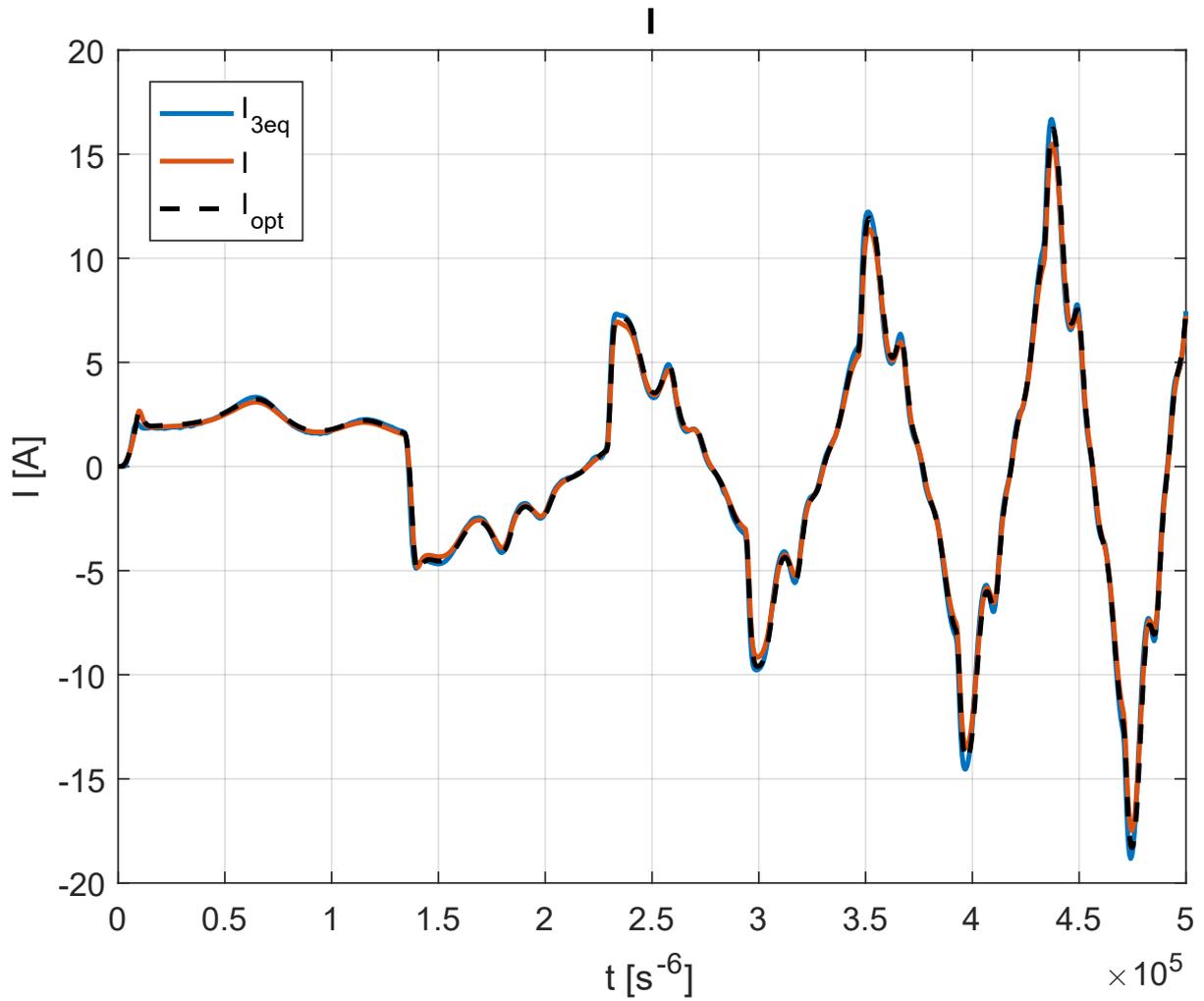
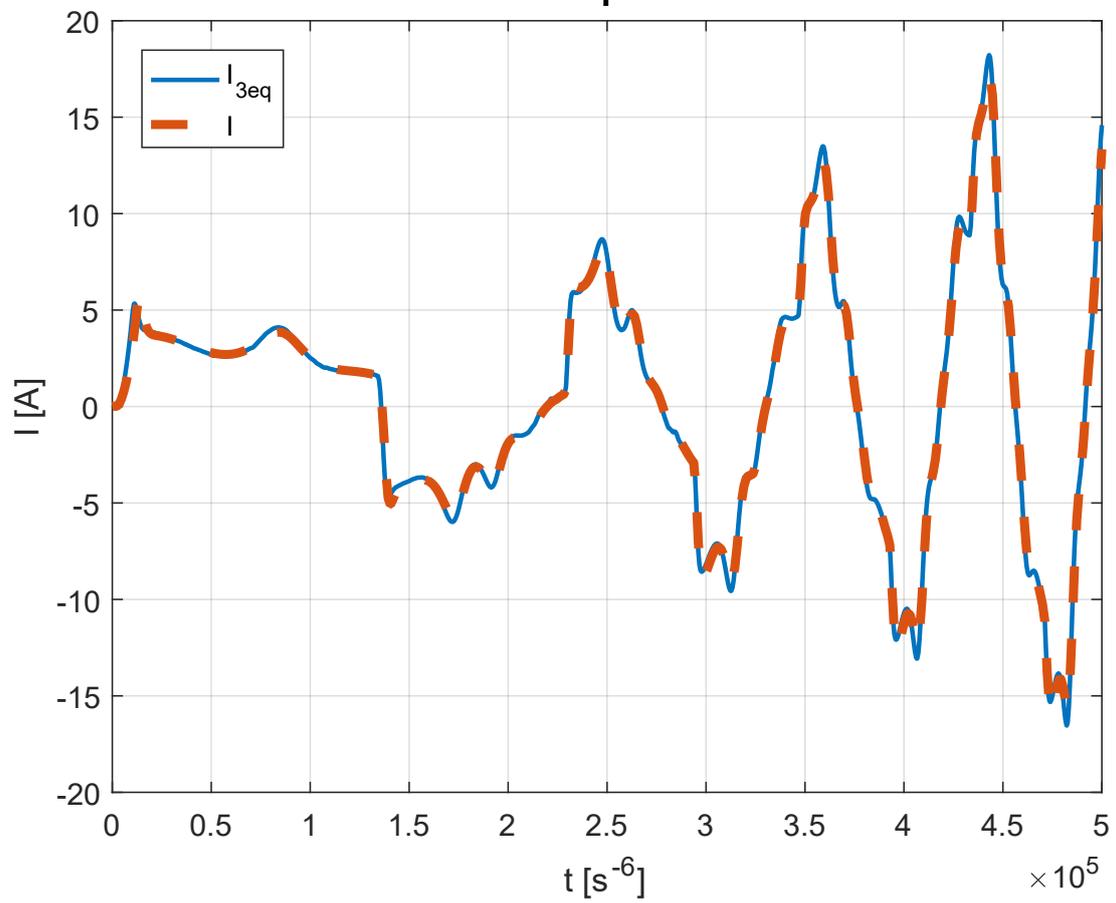
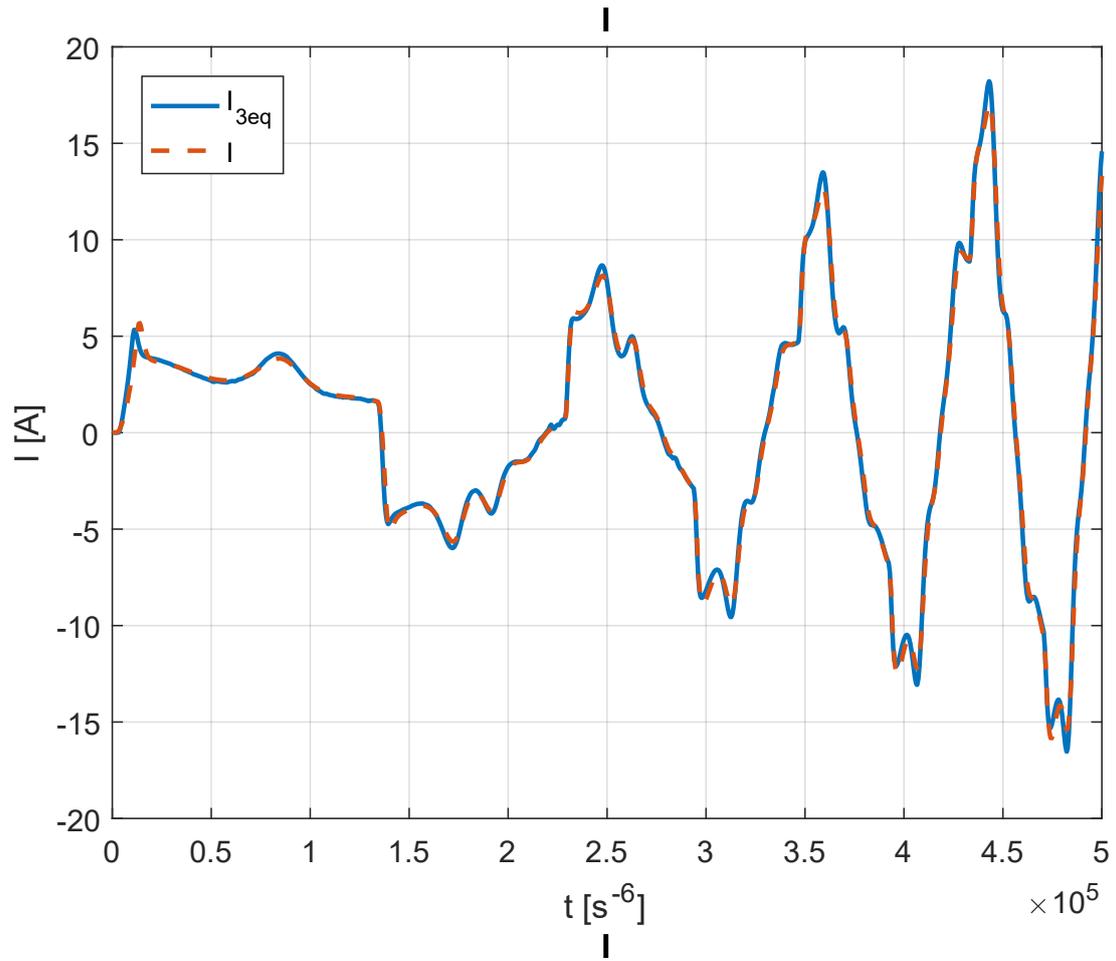


Figure 3.52: Current trends with $[N_A, N_B, N_C] = [0.30, 0.96, 0.99]$ and $[Z, \phi_Z] = [0.0025, 0.0006]$ respectively: before, after optimization, all together

The sixth set of faults, allows us to see a little better the improvement the GA optimization brings: indeed The MSE_f almost gets halved, even though its value stands under $1 [A^2]$. The prevalent effect is given by the SCs, so much that the current response has the same characteristics seen in those kind of failures. As always, when the amount of working coils percentage is low, the current amplitude significantly grows.



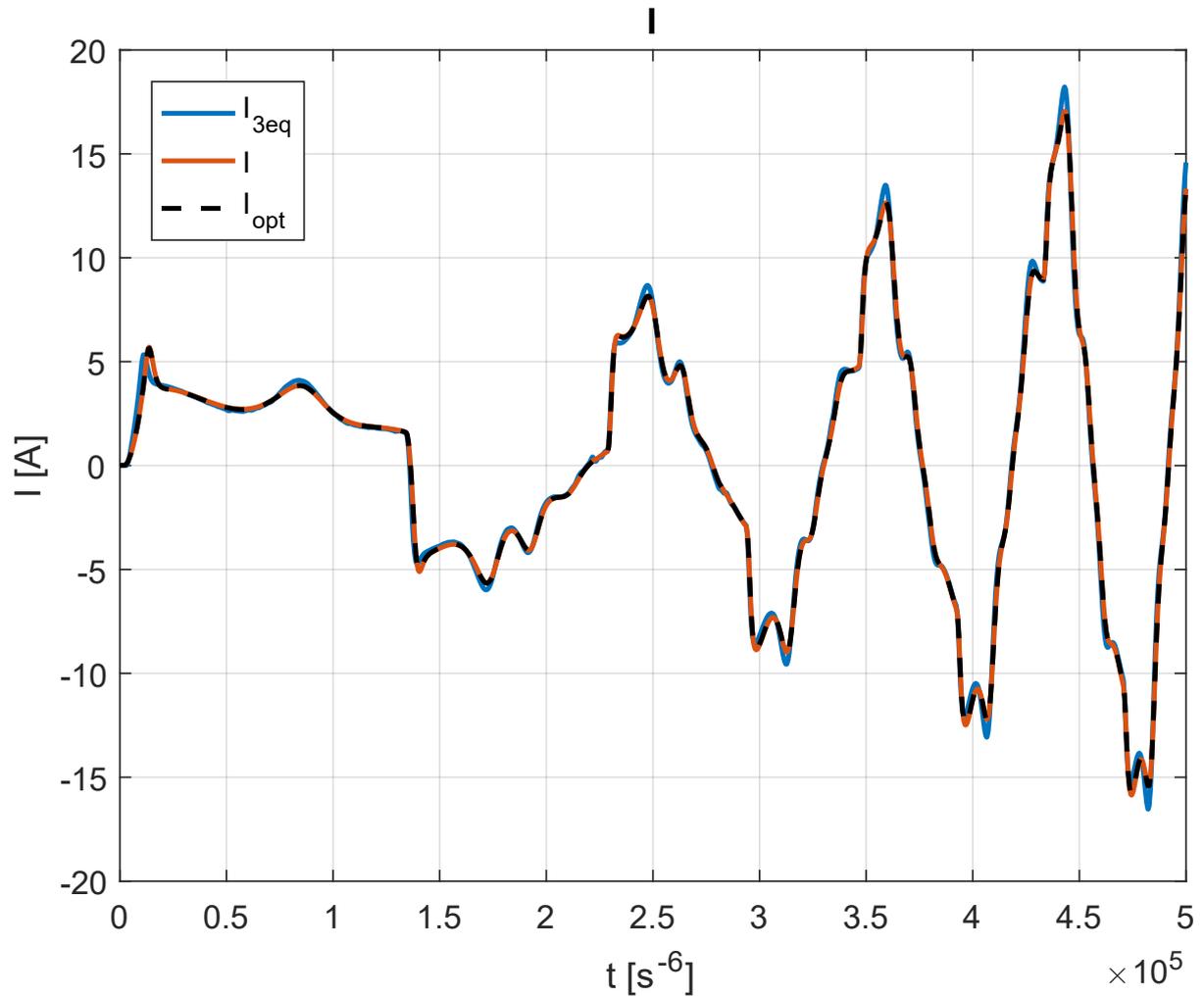


Figure 3.53: Current trends with $[N_A, N_B, N_C] = [0.71, 0.63, 0.27]$ and $[Z, \phi_Z] = [0.02, 0.01]$ respectively: before, after optimization, all together

With the last randomly generated failures, again the most relevant effect is that produced by the windings SCs. The matching between curves is very good, but the improvement conducted by GA it's definitely not visible, except for the MSE_f data reported in table 3.4.

Next, we are going to interpolate the failure parameters, with respect to the single kind of faults. The same has been done in the previous section, but this time we will even have k_{fE} charts, related with Z and ϕ_Z , so keep in mind the equivalences stated in equation 3.56.

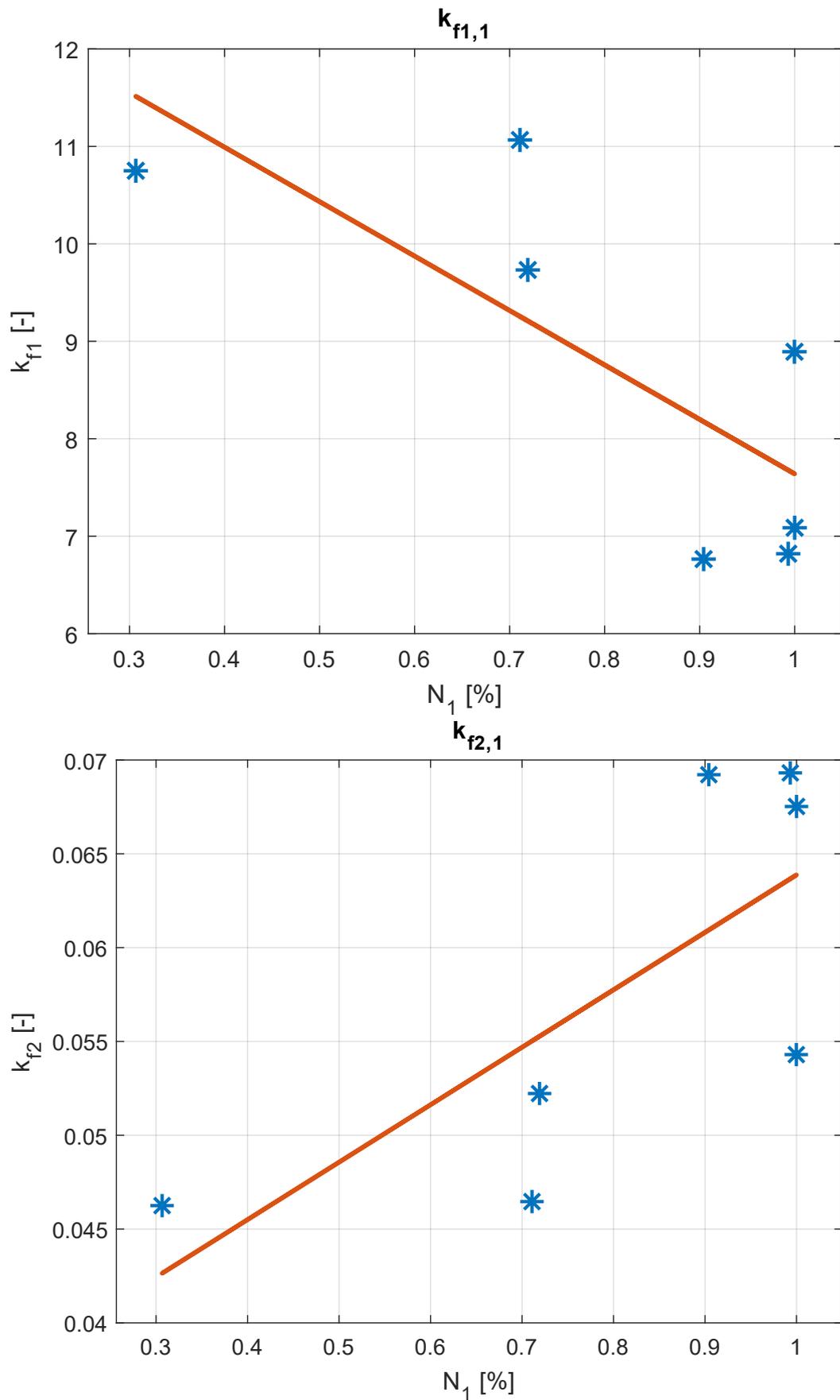


Figure 3.54: K_{f_i} optimized data interpolation, with A winding SC + E

We can see, as expected, there is a strong dispersion among data to be attributed to the influence the

faults have between each other and to the small amount of points; indeed the interpolation lines slope has increased compared with those of figure 3.44.

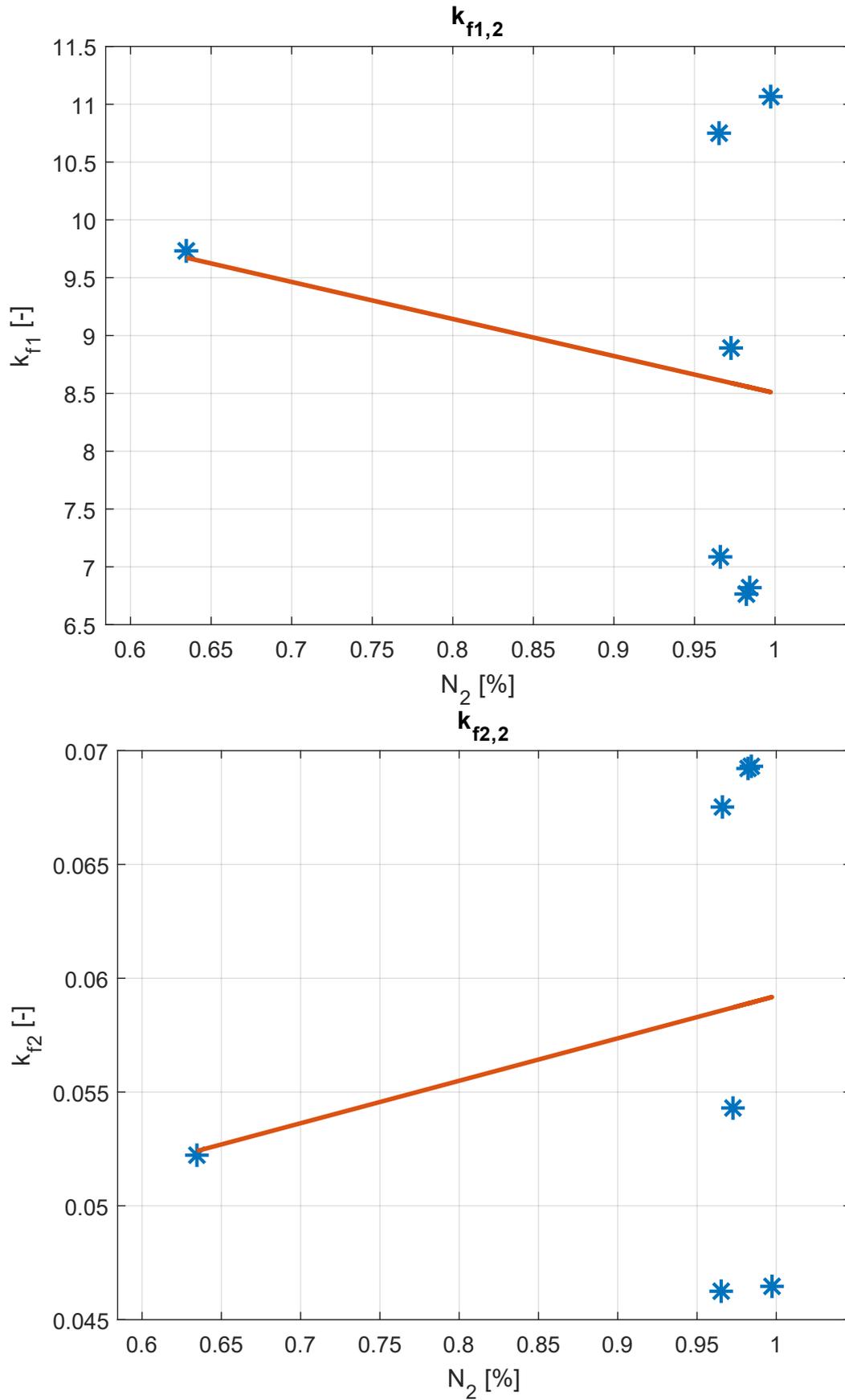
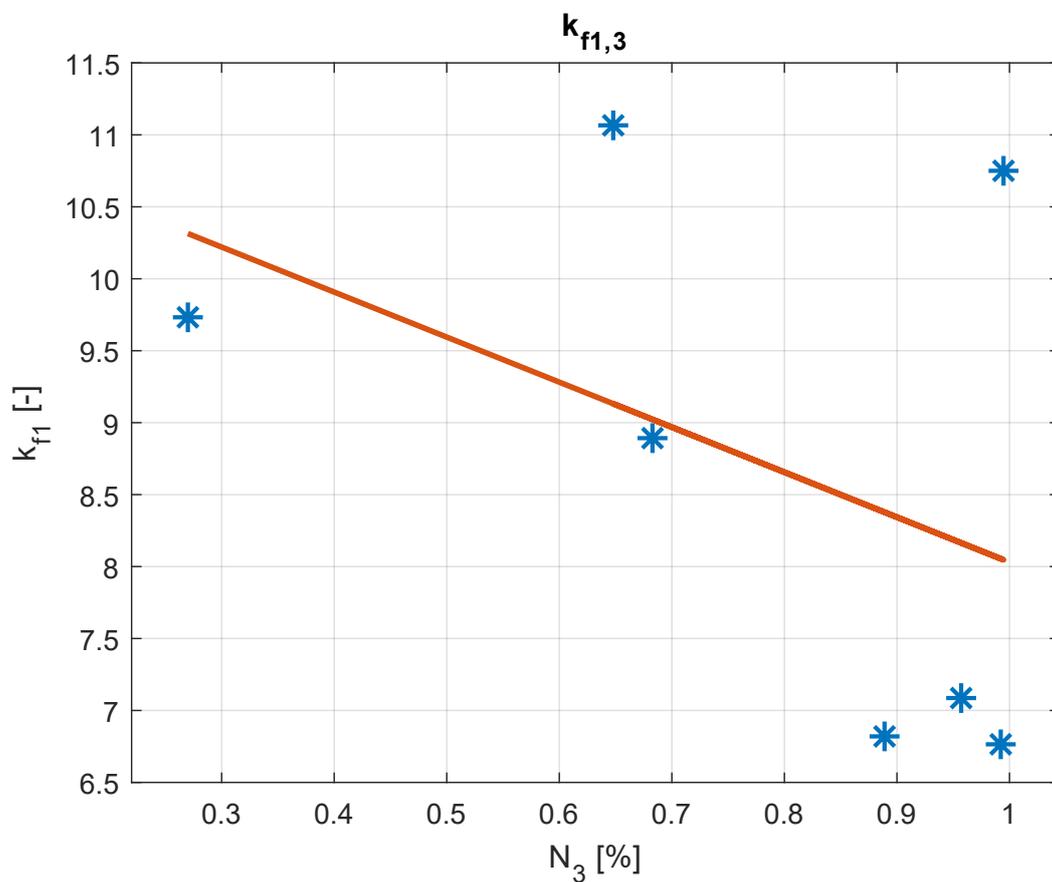


Figure 3.55: K_{f_i} optimized data interpolation, with B winding SC + E

In the figures above, the situation is even worse, but here, one more factor has a role: the fact 6 of 7 points are between 95% and 100% of B working coils. Indeed, as already explained, in those conditions, the influence of k_{f_i} parameters on the objective function is very slight, causing the spread shown in figures 3.55, along with the other reasons already presented.



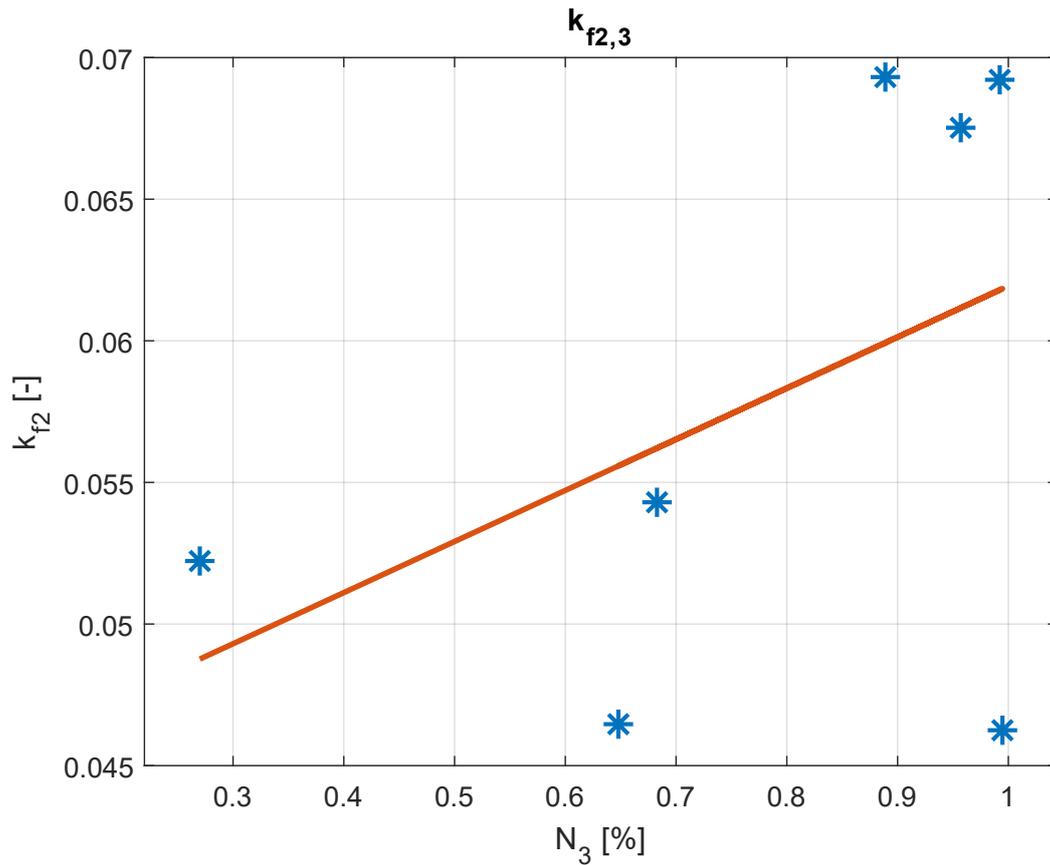


Figure 3.56: K_{f_i} optimized data interpolation, with C winding $SC + E$

Nothing, that hasn't already been said, to add: strong data dispersion. It's evident that, this kind of tests (with this non statistically relevant samplings), are not suitable for data interpolation, at least, for those related with short circuits, being apparently, strongly connected. For sure, the fact that each windings component is multiplied by the same factor, doesn't play a good role in this game, but, the eventuality of utilizing three different coefficients must be carefully taken into account. Indeed, the parameters to be optimized, in this case, would become 5, and with this quantity of d.o.f. a neural network employment would be needed, with all the complexity this would involve. Seen the goodness of the model, even without parameters optimization, that simply doesn't worth it.

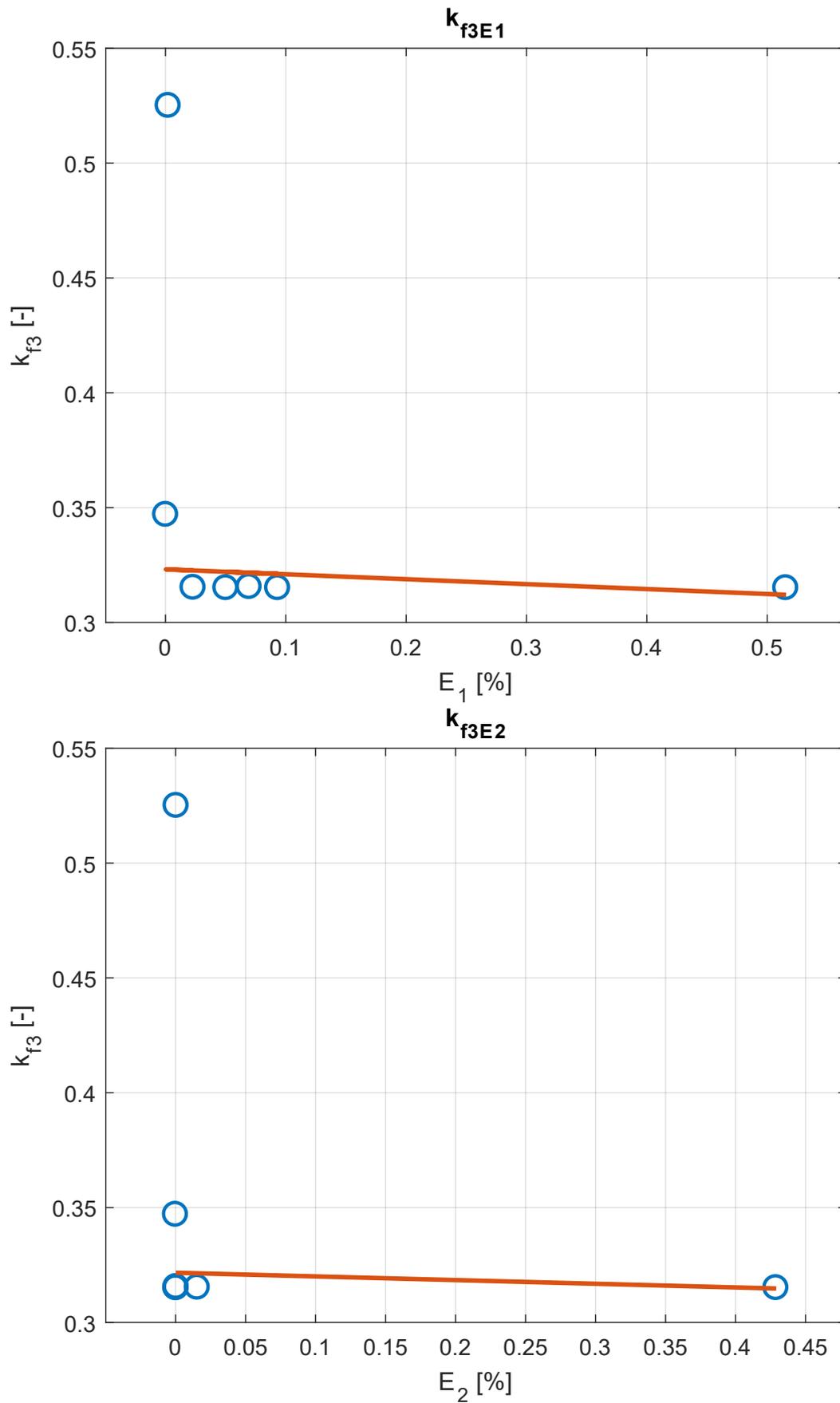


Figure 3.57: K_{fE} optimized data interpolation, with rotor static eccentricity

Regarding the eccentricity parameter k_{fE} , instead, it looks like it's not influenced by short circuits

failures, indeed its value stays almost constant, if we exclude the outstanding value greater than 0.5 from the interpolation. That's a reasonable thing to do because of the fact that, when Z is really close to zero, whatever parameter multiplies it, the results is always going to be zero, and that justifies the huge gap between those values and the exclusion of the bigger one. In the second of the 3.57, because of their proximity, two points are hidden by others, but become visible whether a logarithmic scale is used for the horizontal axis:

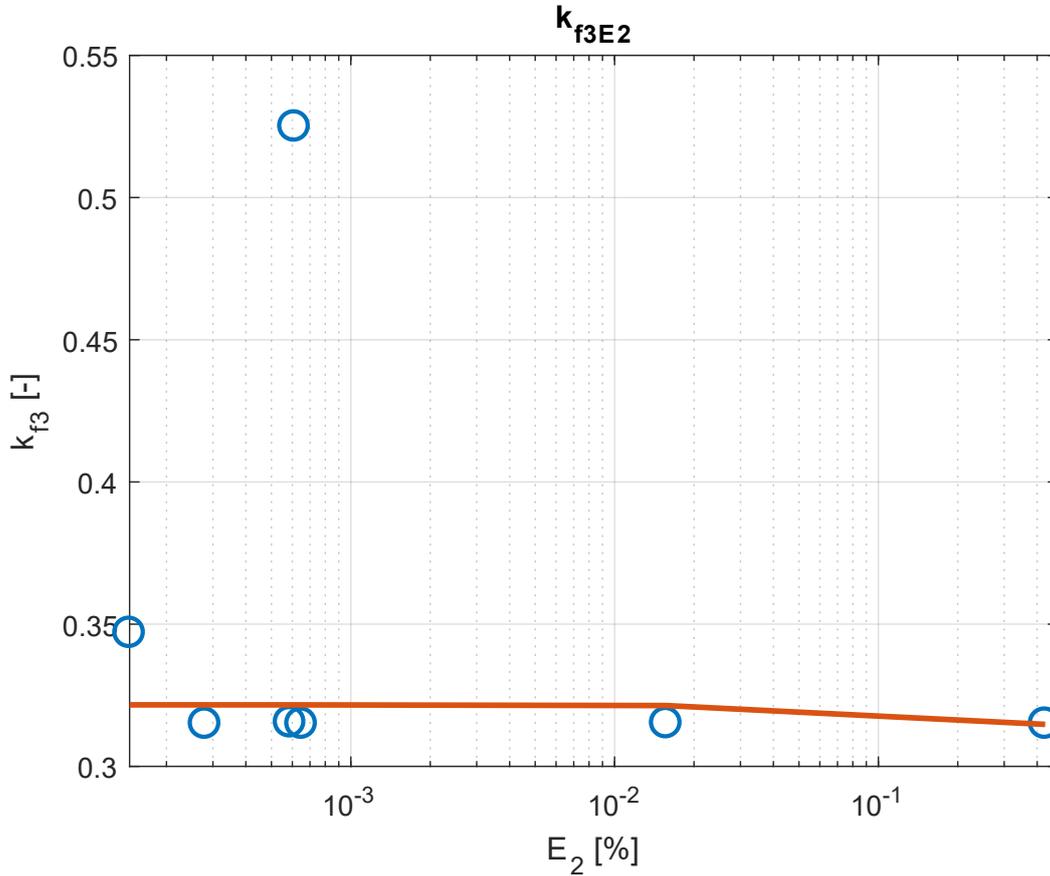


Figure 3.58: K_{fE} optimized data interpolation, with rotor static eccentricity, in a logarithmic horizontal scale

To be noticed that, due to the scale changing, the interpolating line looks no more straight, but be aware that is just a “visual” effect.

3.4 Model Validation

In order to have a more rigorous confirmation of the monitoring model goodness, it was underwent to a hundred casual faults combinations, calculating at each test the *Normalized Root Mean Squared Error (NRMSE)* defined as follows:

$$NRMSE = \frac{\sqrt{\frac{\sum_{i=1}^n (I_{3eq_i} - I_i)^2}{n}}}{\max(I_{3eq_i}) - \min(I_{3eq_i})} [\%] \quad (3.59)$$

The choice of this parameter has been given by the fact that, in literature, is one the most diffused way to estimate the error between curves, and by the need of tearing down the calculation time. Indeed, the *NRMSE* is way faster to estimate than the MSE_f given by 3.52.

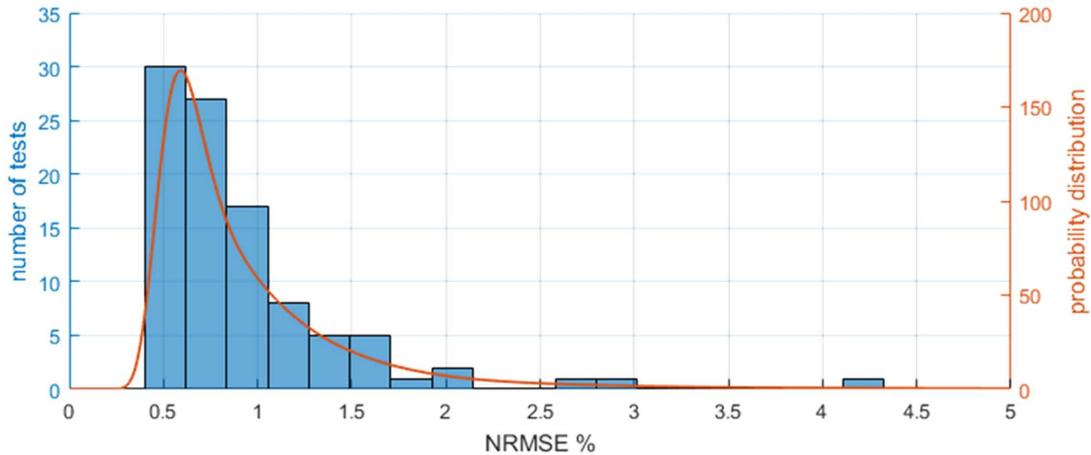


Figure 3.59: NRMSE histograms graph, with validation data Gaussian distribution

As we can see, the Gaussian distribution curve center stands around a *NRMSE* value of 0.6%, which is considered to be small enough for our applications. Therefore, according to what can be done without having an experimental feedback, the model can be considered validated. To be noticed that this tests have been run with no parameters optimization; The model, indeed, would require an upgrade in order to implement the “failure parameters linear interpolation” which is going to be part of the next steps to do (see section 4).

3.5 Borello-Dalla Vedova Dry Friction Model

In this section, we are going to examine the mathematical model related to a dry friction innovative numerical simulation routine (developed by Prof. Lorenzo Borello ed implemented in Matlab-Simulink environment by Eng. Matteo D. L. Dalla Vedova); the proposed numerical model is considered to be enough consistent with the classical Coulomb friction model (as long as necessary data are available) ed it’s easily integrable with more complex calculation programs (related to. for example, servomechanisms or whole actuation systems), proving, furthermore, a particular robustness towards the specific simulated conditions. The model proposed overcomes the limits showed by other solutions and reflects the physic model behavior assumed as target by this work; indeed its main tasks are listed below.

- **Distinguishes the friction torque sign** in function of the speed direction;
- **Discriminates adherence from dynamic conditions** (indeed two different values can be assigned to the friction torque: *FSJ* for statical condition and *FDJ* for dynamical ones);
- **Evaluates the eventual stoppage of the mechanical component**, moving at the beginning;
- **Assesses the eventual element restart**, still at the beginning;
- **Correctly keeps arrested (or in motion)** the mechanical component;
- **Takes into account the presence of eventual mechanical end-stops**, against which a completely inelastic shock is supposed to occur.

The considered model, starting from the classical Coulomb formulation, describes the dry friction effects as a function of speed and active torque according to the following modeling:

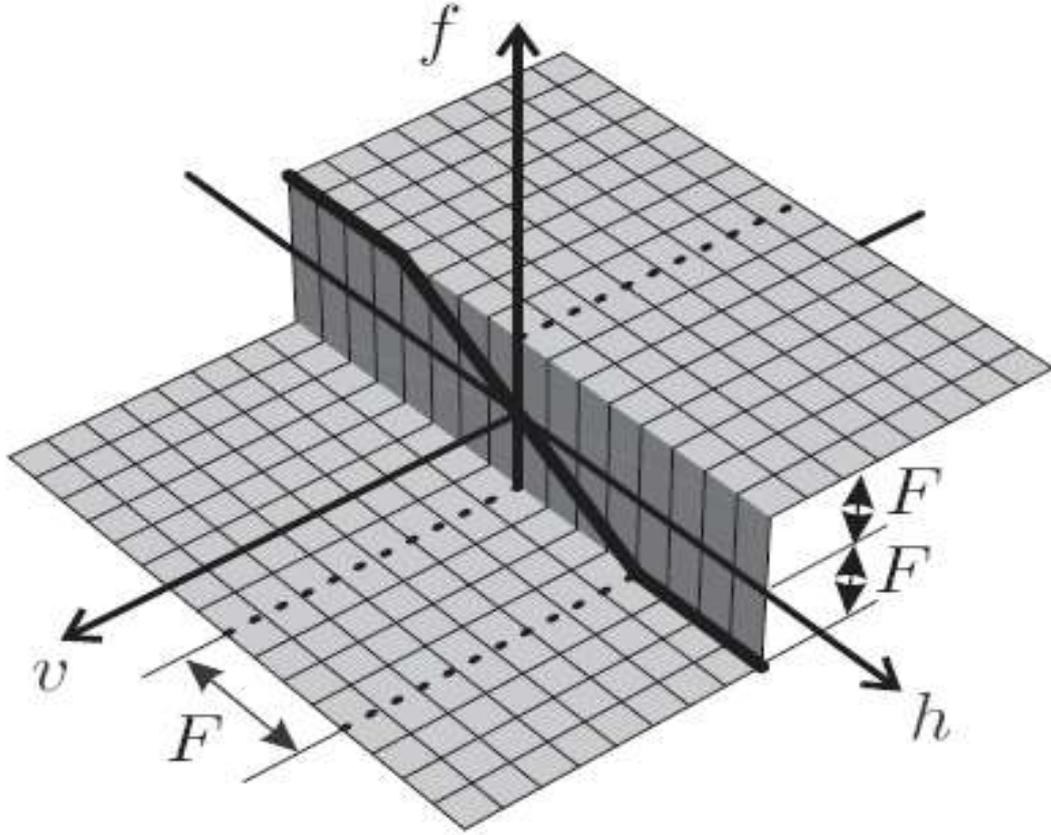


Figure 3.60: Borello dry friction Grafic representation

Whose mathematical model can be formulated as follows:

$$FF \equiv T_f = \begin{cases} F_{att} \equiv T & \text{if } v \equiv \dot{\theta}_m = 0 \wedge |T| \leq FSJ \\ \text{sgn}(T) \cdot FSJ & \text{if } \dot{\theta}_m = 0 \wedge |T| > FSJ \\ FDJ & \text{if } \dot{\theta}_m \neq 0 \end{cases} \quad (3.60)$$

Where FF is our equivalent T_f , the torque coming as model output, and represents the calculated friction torque; FSJ is the *static friction torque*, FDJ the *dynamic friction* one and F_{att} is our T . Numerical simulation programs, build upon the mathematical model described just above, in order to avoid numerical instability phenomena similar to those manifested, for example, in the *Karnopp* model, are made in such a way the mechanical system stop is imposed whether angular speed changes sign during the integration step, that is:

$$\dot{\theta}_m(t_{i+1}) = 0 \text{ if } \dot{\theta}_m(t_{i+1}) \cdot \dot{\theta}_m(t_i) \leq 0 \quad (3.61)$$

In case this kind of imposed stop would not appear to be correct, due to active torque exceeding the passive one (friction), the imbalance between forces acting on the system, would cause its proper restart at the next calculation step. It can be said that the conditions related to the speed, represents one among the fundamental innovations in respect to the models reported in literature and, in its apparent obviousness, dwell the method robustness and accuracy. Moreover, the model we're talking about, is able to distinguish aiding and opposing conditions, by comparing the load sign with the actuation speed one and choosing, time after time, the right parameters to use. Anyway, since the monitor model described in section 3, has been tested in *zero-load* conditions, the necessity of operating the distinction between the two cases above fails, allowing us to streamline the numerical calculation, by avoiding the load input furniture to the Borello model box.

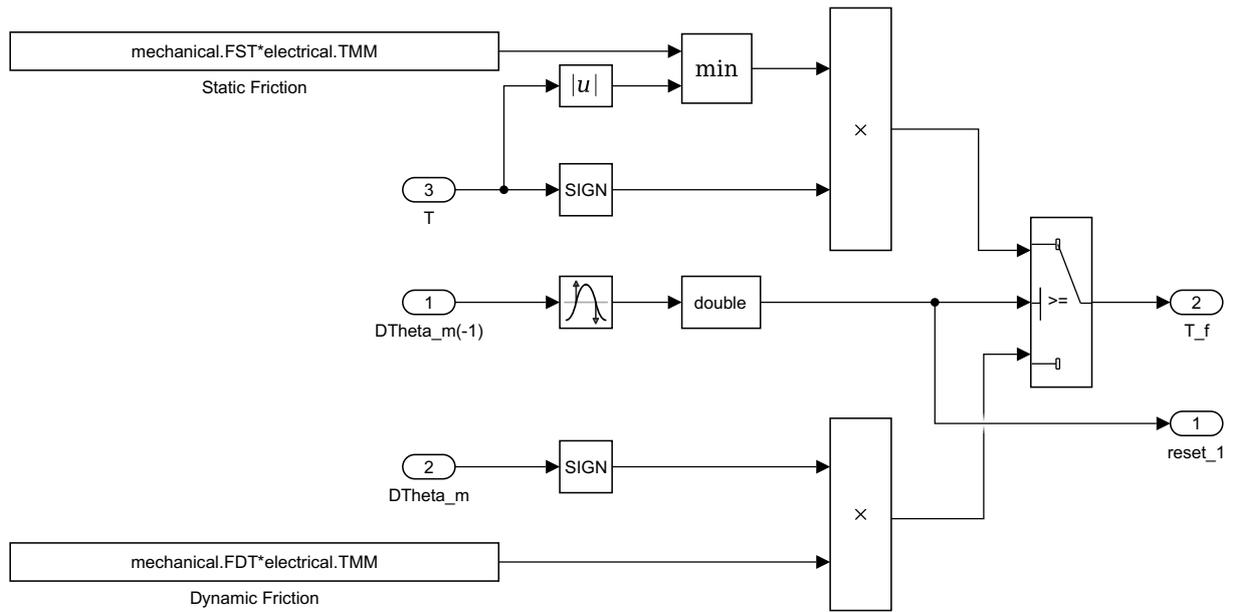


Figure 3.61: Borello Friction Model box content

The produced output is the friction torque T_f performing in those conditions, which is subtracted (or could be added, whether aiding conditions would be considered) to the torque T .

4 | Conclusions and Next Steps

It can be said the aim purposed at the beginning of the present work has been reached: build up a light model enough precise in terms of current output, able to well reproduce partial windings short circuits and rotor mechanical failures, with the future target of real time run, in order to perform diagnostic and prognostic features for flight control EMAs. At the current state, the monitoring model is able to run a 0.5 [s] simulation in about 1.8 [s], with Simulink *acceleration* feature set on, while in *normal mode*, it takes around 7.8 [s]. The HF model, instead, takes around 3.8 [s] with acceleration mode and 11.7 [s] with the normal one. To be noticed, these estimations have been done considering only the simulation time, excluding all the data loading, and accessories operations. Furthermore, the acceleration mode gave several problems during the calibration and optimization processes, so much that it required to be deactivated, justifying that big amount of time taken by the operations reported in section 3.3.

MODE	HF [s]	LF [s]	TR [%]
Norm	11.7	7.8	34
Acc	3.8	1.8	53

Table 4.1: Models simulation time summary

In the table above is reported a run-time summary, where *TR* stands for *Time Reduction* and is expressed in percentage related to the HF model simulation time. We can notice a 34% of reduction in normal mode and a 53% one in accelerated conditions; both are considered good improvements, even though, for sure more can be done. It's been considered to be a good idea to make a point of the current state of the work and to give some cues for future developments. Even though all the "weak points" have already been putted in evidence as they were found along the present study, here follows a list of things "to be done" :

- **Backlash model improvement:** a smart solution needs to be found, in order to let the model consider the backlash effects in all possible occurrences, without weighting it down too much;
- **Φ_E function improvement(?):** it could be evaluated whether or not, this failure function needs to be modified in order to well reproduce medium-to-high entity of static eccentricity faults;
- **Running statistically consistent tests for better parameters interpolation:** in order to have a more precise reply in terms of failures parameters variation with different kind of faults, the number of tests conducted must be increased. Furthermore, a better faults category isolation should be taken into account;
- **Run-time tear down:** since the final aim of the monitoring model is to be run in real-time, during the actuation, the simulation time should undergo to a further cut-out. A good result could be reached, for example, by simplifying the control logics and removing the anti wind-up filters; of course, all of this to be done after deep consequences evaluation in terms of model fidelity;
- **Experimental validation:** this step, would have been part of the present work, but since the test bench assembly requires way more time than that given for this study, it will be passed to the "next generation". In particular, the failures will be reproduced in laboratory, in order to verify the current matching goodness and some other "home-made" prognostic algorithms effectiveness.

Bibliography

- [1] M. Scanavino: *Numerical methods for the electromechanical modeling of aerospace actuators based on sinusoidal brush-less motors*, M.D. Thesis, A.A. 2015/16
- [2] P. C. Berri: *Genetic Algorithms for Prognostics of Electromechanical Actuators*, M.D. Thesis, A.A. 2015/16
- [3] P. C. Berri, M. D. L. Dalla Vedova, P. Maggiore: *A smart electromechanical actuator monitor for new model-based prognostics algorithms*, International Journal of Mechanics and Control, Vol. 17, No. 02, 2016
- [4] [MathWorks](#), *Global Optimization Toolbox*
- [5] [MathWorks](#), *Genetic Algorithm presentation*
- [6] [MathWorks](#), *Genetic Algorithm working description*
- [7] P. C. Berri, M. D.L. Dalla Vedova, P. Maggiore: *On-board electromechanical servomechanisms affected by progressive faults: Proposal of a smart GA model-based prognostic approach*, 27th European Safety and Reliability Conference (ESREL 2017), Portoroz, Slovenia (2018)
- [8] L. Borello, M. D. L. Dalla Vedova: *A dry friction model and robust computational algorithm for reversible or irreversible motion transmission*, International Journal of Mechanics and Control (JoMaC) 13 (02), 37-48