



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY



# EDCMAG: Equivalente Dinámico Calculado Mediante Algoritmos Genéticos

Proyecto de fin de carrera de Ingeniería Eléctrica, perfil  
Potencia

**Integrantes:**

Franco Pesce - Ignacio Cibils - Facundo Motta

**Tutor:**

Nicolás Yedrzejewski

15 de agosto de 2017



## *Agradecimientos*

*Queremos expresar, en nombre de los participantes del proyecto, nuestro agradecimiento a todas las personas que lo hicieron posible. En particular a nuestro tutor, Ing. Nicolás Yedrzejewski, por confiar en nosotros para llevar a cabo este estudio y brindarnos su tiempo, apoyo y herramientas necesarias a lo largo de todo este año. También le agradecemos al Ing. Michel Artenstein por compartir su conocimiento sobre la teoría de Equivalentes Dinámicos y por su colaboración en el análisis de los resultados obtenidos.*

*Agradecemos también al Ing. Elías Carnelli, a UTE y al Departamento de Potencia de la Facultad de Ingeniería, por asistirnos con las licencias del software utilizado para la realización de los programas y las simulaciones.*

*Por último, nos gustaría agradecer a nuestras familias y afectos por el apoyo incondicional a lo largo de este proyecto y de nuestra carrera.*



## *Resumen*

*El presente estudio propone la optimización del cálculo de equivalentes dinámicos de sistemas eléctricos de gran potencia, previamente definidos en su topología eléctrica, mediante la aplicación de métodos no tradicionales, específicamente la aplicación de Algoritmos Genéticos, método enmarcado dentro de los llamados Algoritmos Evolutivos. El estudio es realizado desde la red eléctrica uruguaya con el objetivo de obtener un equivalente dinámico de la red eléctrica argentina, para el estudio de fenómenos dinámicos desde el SIN, buscando un equilibrio entre la magnitud del equivalente y la precisión del mismo. Finalmente, se realizaron diversas simulaciones de fallas eléctricas para medir la eficiencia del equivalente y compararlo con el sistema eléctrico original.*



## Índice

<b>1. Introducción</b>	<b>10</b>
1.1. Descripción del problema . . . . .	10
1.2. Objetivo . . . . .	10
1.3. Breve descripción de la solución . . . . .	11
1.4. Sistema Eléctrico Uruguay-Argentina . . . . .	14
1.4.1. Uruguay . . . . .	14
1.4.2. Argentina . . . . .	17
1.4.3. Interconexiones Uruguay-Argentina . . . . .	17
1.4.4. Interconexiones Uruguay-Brasil . . . . .	21
1.5. Antecedentes . . . . .	21
<b>2. Equivalentes Dinámicos</b>	<b>23</b>
2.1. Definición de Equivalente Dinámico . . . . .	23
2.2. Motivación de la búsqueda de Equivalentes Dinámicos . . . . .	23
2.3. Reducción de modelos en sistemas de potencia . . . . .	25
2.4. Métodos tradicionales de obtención de un equivalente . . . . .	27
2.4.1. Equivalentes para el subsistema lineal . . . . .	27
2.4.2. Equivalentes para el subsistema no-lineal . . . . .	28
2.5. Variables dinámicas . . . . .	30
2.6. Modelado de cargas . . . . .	31
2.7. Modelo clásico de sistemas multimáquina . . . . .	32
2.8. EDCMAG: Equivalente Dinámico Calculado Mediante Algoritmos Genéticos . . . . .	35
<b>3. Algoritmos Genéticos</b>	<b>36</b>
3.1. Introducción . . . . .	36
3.2. Función de fitness . . . . .	36
3.3. Operadores . . . . .	38
3.3.1. Selección . . . . .	38
3.3.2. Reproducción . . . . .	39
3.3.3. Mutación . . . . .	39
3.4. Descripción de cómo funciona un AG . . . . .	40
3.4.1. Diagrama de flujo . . . . .	41
3.5. Población e individuos . . . . .	42
3.6. Algunas aplicaciones de los AG . . . . .	43
3.7. Antecedentes de aplicaciones de AG a sistemas eléctricos . . . . .	44
<b>4. Modelado del equivalente</b>	<b>45</b>
4.1. Modelos de generadores del PSS®E . . . . .	46
4.2. Modelos de líneas . . . . .	49
4.3. Variables a optimizar . . . . .	50
4.4. Equivalente Reducido . . . . .	52
4.5. Equivalente NY . . . . .	54

4.6.	Archivos necesarios para cada equivalente . . . . .	56
<b>5.</b>	<b>Implementación del programa de optimización</b>	<b>57</b>
5.1.	Versiones de software . . . . .	57
5.2.	Desarrollo del programa . . . . .	57
5.2.1.	Descripción del programa principal . . . . .	58
5.2.2.	Diagrama de flujo . . . . .	59
5.2.3.	Pseudocódigo . . . . .	60
5.3.	Simulación de faltas . . . . .	61
5.3.1.	Definición de una falta en un sistema eléctrico de potencia . . . . .	61
5.3.2.	Tipos de faltas eléctricas . . . . .	62
5.3.3.	Causas y consecuencias de faltas eléctricas . . . . .	62
5.3.4.	Definición de faltas eléctricas en PSS®E . . . . .	63
5.3.5.	Información relevante sobre una falta genérica . . . . .	64
5.3.6.	Faltas simuladas . . . . .	66
5.4.	Algoritmo de optimización . . . . .	68
5.4.1.	Generación de la población inicial . . . . .	68
5.4.2.	Prueba y elección de la función de fitness . . . . .	69
5.4.3.	Selección y reproducción de la población . . . . .	72
5.4.4.	Mutación de la población . . . . .	74
5.4.5.	Elección del mejor individuo/resultado . . . . .	74
5.5.	Definición de subrutinas . . . . .	74
<b>6.</b>	<b>Pruebas y resultados</b>	<b>82</b>
6.1.	Obstáculos y primeras pruebas . . . . .	82
6.1.1.	Convergencia del AG . . . . .	86
6.2.	Simulaciones . . . . .	88
6.2.1.	Equivalente Reducido . . . . .	89
6.2.2.	Equivalente NY . . . . .	90
6.2.3.	Conclusiones . . . . .	101
6.3.	Análisis cualitativo de fallas . . . . .	106
6.4.	Resultados numéricos . . . . .	113
6.5.	Conclusiones generales . . . . .	120
6.6.	Perspectivas a futuro . . . . .	121
<b>A.</b>	<b>Cartas GENSAL y GENROU</b>	<b>125</b>
<b>B.</b>	<b>Códigos</b>	<b>127</b>
B.1.	Código de programa principal . . . . .	127
B.2.	Código de programa para graficar resultados . . . . .	140
<b>C.</b>	<b>Simulaciones</b>	<b>155</b>
C.1.	Simulación 1: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando frecuencia en MI5 en 4 segundos. Comportamiento en 4 segundos. . . . .	156

C.2. Simulación 2: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando frecuencia en PA5 en 4 segundos. Comportamiento en 4 segundos . .	168
C.3. Simulación 2: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando frecuencia en PA5 en 4 segundos. Comportamiento en 20 segundos . .	180
C.4. Simulación 3: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando ángulo en una máquina de PAL en 4 segundos. Comportamiento en 4 segundos . . . . .	192
C.5. Simulación 3: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando ángulo en una máquina de PAL en 4 segundos. Comportamiento en 20 segundos . . . . .	204
C.6. Simulación 4.1: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando ángulo en una máquina de PTI en 4 segundos. Comportamiento en 4 segundos . . . . .	216
C.7. Simulación 4.2: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando ángulo en una máquina de PTI en 20 segundos. Comportamiento en 20 segundos . . . . .	228
C.8. Simulación 5: Cortocircuito 3F en MA5 con salida de línea en equivalente NY 3, optimizando tensión en MA5 en 4 segundos. Comportamiento en 4 segundos . . .	240
C.9. Simulación 6: Cortocircuito 3F en MA5 con salida de línea en equivalente NY 3, optimizando ángulo en PTI en 4 segundos. Comportamiento en 4 segundos . . . .	252
C.10. Simulación 7: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando frecuencia en PA5 en 4 segundos. Comportamiento en 4 segundos . .	264
C.11. Simulación 7: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando frecuencia en PA5 en 4 segundos. Comportamiento en 20 segundos .	276
C.12. Simulación 8: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando ángulo de un máquina de PAL en 4 segundos. Comportamiento en 4 segundos . . . . .	288
C.13. Simulación 9.1: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando ángulo de un máquina de PTI en 4 segundos. Comportamiento en 4 segundos . . . . .	300
C.14. Simulación 9.2: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando ángulo de un máquina de PTI en 9 segundos. Comportamiento en 20 segundos . . . . .	312
C.15. Simulación 10: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando frecuencia en MI5 en 4 segundos. Comportamiento en 4 segundos . .	324
C.16. Simulación 10: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando frecuencia en MI5 en 4 segundos. Comportamiento en 20 segundos .	336

## 1. Introducción

### 1.1. Descripción del problema

El sistema eléctrico de Uruguay se encuentra fuertemente interconectado con el de Argentina a través de líneas de 500kV con una capacidad de interconexión de 2.000MW. Esto determina que las acciones o perturbaciones que se toman u ocurren en un país afectan al otro. El sistema eléctrico argentino es aproximadamente 10 veces mayor que el uruguayo haciendo que la influencia de Argentina sobre Uruguay sea la predominante. Por este motivo, en mayor o menor medida siempre se debe considerar al país vecino para realizar estudios del sistema eléctrico.

Siempre existe la posibilidad de modelar el sistema eléctrico de ambos países de manera completa. De hecho, UTE cuenta con el modelo completo. Sin embargo, una simulación dinámica con tal grado de detalle insume mucho tiempo de ejecución y una base de datos enorme. Para UTE además implica trabajar con un sistema del que solo conoce en detalle el 10% , el otro 90% corresponde a Argentina. En algunas aplicaciones esto puede ser una desventaja y surge la necesidad de contar con un sistema eléctrico equivalente de Argentina que represente de manera correcta el tipo de fenómenos que se quiere analizar.

Existen algunos métodos propuestos en la literatura para desarrollar equivalentes dinámicos de sistemas eléctricos de potencia, la mayoría de difícil abordaje. Entre ellos se encuentran ejemplos de utilización de métodos de optimización que buscan crear equivalentes dinámicos reducidos.

### 1.2. Objetivo

El objetivo de este proyecto es estudiar y explorar el comportamiento, la utilidad y factibilidad de herramientas de optimización aplicadas a la construcción de equivalentes dinámicos para lograr un modelo equivalente adecuado del sistema eléctrico argentino. Con tal motivo se pretende desarrollar una herramienta basada en procesos iterativos que combine el software de simulación PSS®E con python, que sea flexible y que resuelva el problema en tiempos razonables. En particular se centra en el estudio de Algoritmos Genéticos, dejando abierta la posibilidad para explorar otras metodologías de optimización.

En este proyecto se evalúan tres puntos principales (con márgenes apropiados y a definir): semejanza entre modelo equivalente y modelo real de la red, tiempo de ejecución de los algoritmos y tamaño del equivalente. Desde este punto de vista, se deberá encontrar un equilibrio entre el tamaño del equivalente y la precisión de los resultados. De esta forma, se intentará medir la aplicación de este método de optimización, basado en la teoría de Algoritmos Genéticos, a topologías eléctricas predefinidas que representen la red eléctrica argentina ante fenómenos dinámicos ocurridos en la red eléctrica nacional. La herramienta deberá comparar estos sistemas ante los fenómenos dinámicos y tratará de optimizar estos equivalentes predefinidos, variando constantes dinámicas que deberán también ser estudiadas y probadas, para así asimilarse al sistema eléctrico original en su comportamiento dinámico.

#### **Objetivos específicos del proyecto:**

- Estudiar la factibilidad de la aplicación de algoritmos genéticos para construir el equiva-

lente dinámico de un sistema eléctrico de gran porte (utilizando el sistema argentino como prueba).

- Familiarizarse con métodos no tradicionales de resolución de problemas de optimización, en este caso la implementación de algoritmos genéticos.
- Analizar un sistema eléctrico de gran tamaño y complejidad.
- Adquirir fluidez en el manejo de herramientas informáticas de gran utilidad para la ingeniería de sistemas eléctricos de potencia, como lo son PSSE y Python.
- Familiarizarse con el lenguaje de programación Python.
- Aprender a resolver un problema ingenieril a partir de dos softwares distintos y complementarios trabajando en una misma interfaz.

### 1.3. Breve descripción de la solución

El problema consiste en encontrar un modelo equivalente simplificado a un sistema eléctrico complejo utilizando algoritmos genéticos como método de optimización. En este caso, el sistema eléctrico corresponde al sistema eléctrico argentino y el desafío es simplificar la totalidad de la red a unos pocos elementos (generadores, barras, cargas) para facilitar el análisis de su influencia sobre el sistema eléctrico uruguayo cuando ocurren determinadas fallas. Los resultados consisten en simplificaciones a gran escala de la red argentina.

Por lo tanto, los pasos a seguir para cada equivalente son los siguientes:

- En principio se cuenta con la información total en PSS®E de los sistemas eléctricos argentino y uruguayo.
- A partir del sistema anterior se plantea una determinada falla en el sistema uruguayo, localizada en un determinado punto y se simula en el sistema original.
- A partir de la simulación anterior se elige la variable que se quiere optimizar (sea frecuencia, tensión, etc.) y el tiempo de la simulación (se simula los primeros 4 segundos de la falla) y se guarda el vector como resultado.
- Se elimina en su totalidad la red argentina y se realizan diversas configuraciones eléctricas que puedan representar a la red argentina, sustituyendo a la original. Estas configuraciones son realizadas manualmente, buscando equivalentes de la totalidad de la red.
- Para cada una de las configuraciones, se sortean distintos elementos del sistema y se modifican sus magnitudes mediante el algoritmo genético (constantes de máquinas y de líneas), donde cada vector de magnitudes es un individuo de la población.
- El algoritmo genera una población inicial de individuos de forma aleatoria, donde se sortean las diferentes magnitudes del equivalente, aunque estableciendo márgenes para cada magnitud.

- Con la población inicial generada, se ejecutan n generaciones del algoritmo genético, guardando en cada una de ellas el mejor individuo. La particularidad del algoritmo es que la función de fitness compara la simulación del sistema original con la simulación para cada individuo de la población. La función de fitness calcula la distancia entre ambas curvas con el fin de asignarles un valor y poder evaluar los diferentes individuos de la población.
- Una vez finalizado el algoritmo se tiene como resultado una matriz con n elementos, en la cual se encuentran los mejores individuos de cada generación.
- Como resultado se elige el mejor individuo de la matriz, correspondiente al mejor individuo de las n generaciones.

En la figura 1.1 se muestra la descripción gráfica del proceso, el cual será descrito con detenimiento en los siguientes capítulos.

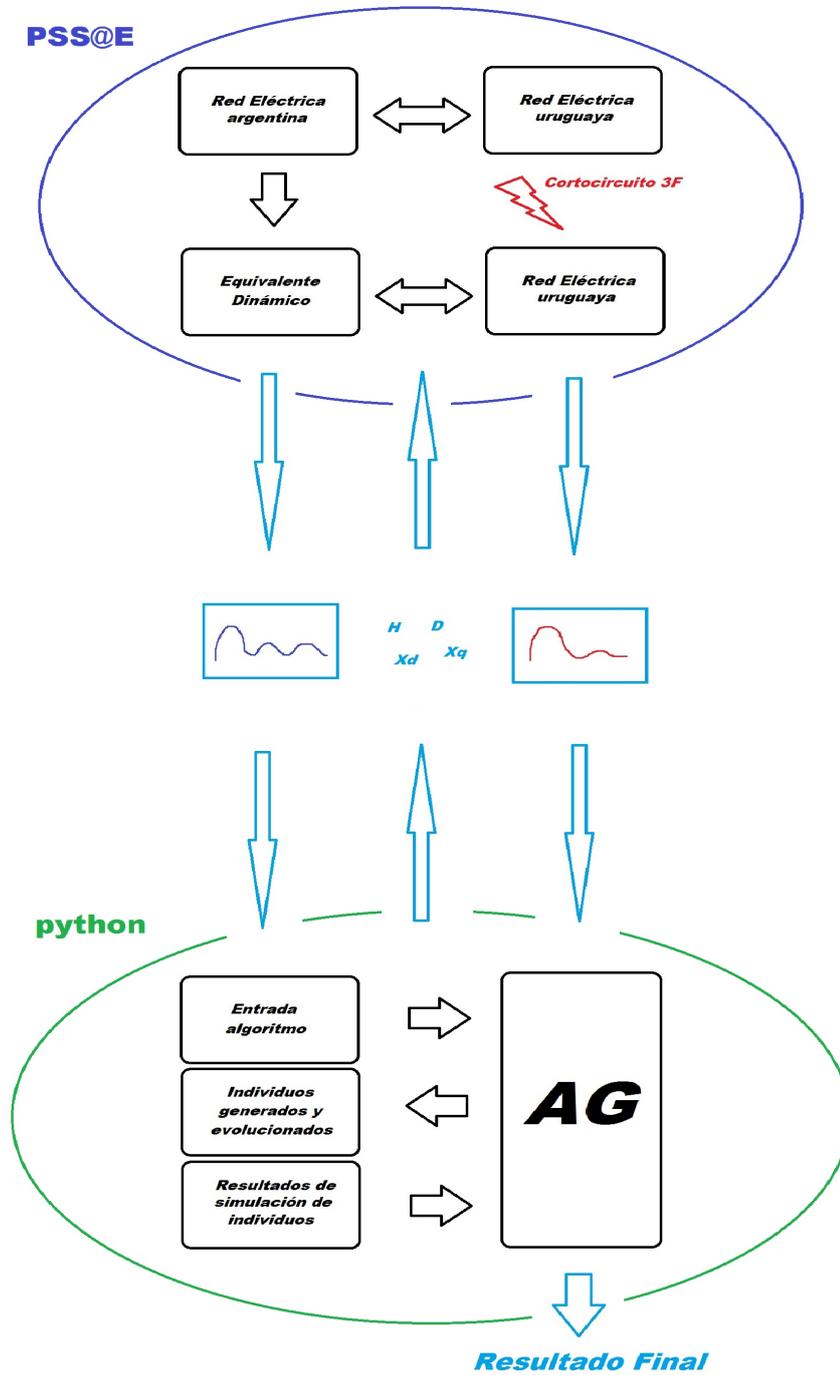


Figura 1.1: Descripción gráfica del proceso.

## 1.4. Sistema Eléctrico Uruguay-Argentina

Para comenzar a entender la magnitud del sistema que se quiere reducir se realiza una breve introducción de los sistemas eléctricos argentino y uruguayo.

### 1.4.1. Uruguay

El sistema eléctrico nacional está formado por un sistema predominantemente de generación hidroeléctrica, aunque en los últimos años ha crecido de forma exponencial la generación de energía eléctrica producida por fuentes de energías renovables, como lo son la eólica, fotovoltaica y biomasa, debido a la política nacional de cambio de la matriz energética. Este cambio llevó al reemplazo en cierta parte de la generación de energía de combustibles fósiles por energías limpias como lo es la energía eólica, la cual ya ha superado los 1000MW de potencia instalada.

La red eléctrica uruguaya consta con un sistema de transmisión en alta tensión de 500kV y 150kV, un sistema de subtransmisión en media tensión de 60kV, un sistema de distribución en media tensión de 31.5kV, 15kV y 6.3kV y un sistema de distribución en baja tensión de 230V/400V.

La red eléctrica de 500kV se extiende en forma de U a través del país, de forma radial, aunque ya se está comenzando a licitar la construcción de la línea Melo-Tacuarembó, para luego realizar otra línea Tacuarembó-Salto, que cerraría el anillo de 500kV, y le daría una mayor estabilidad a la red, así como una mayor facilidad para maniobrar la red.

En el mapa que se detalla en la figura 1.2 se muestra en rojo la red de 500kV y en azul la red de 150kV [1].

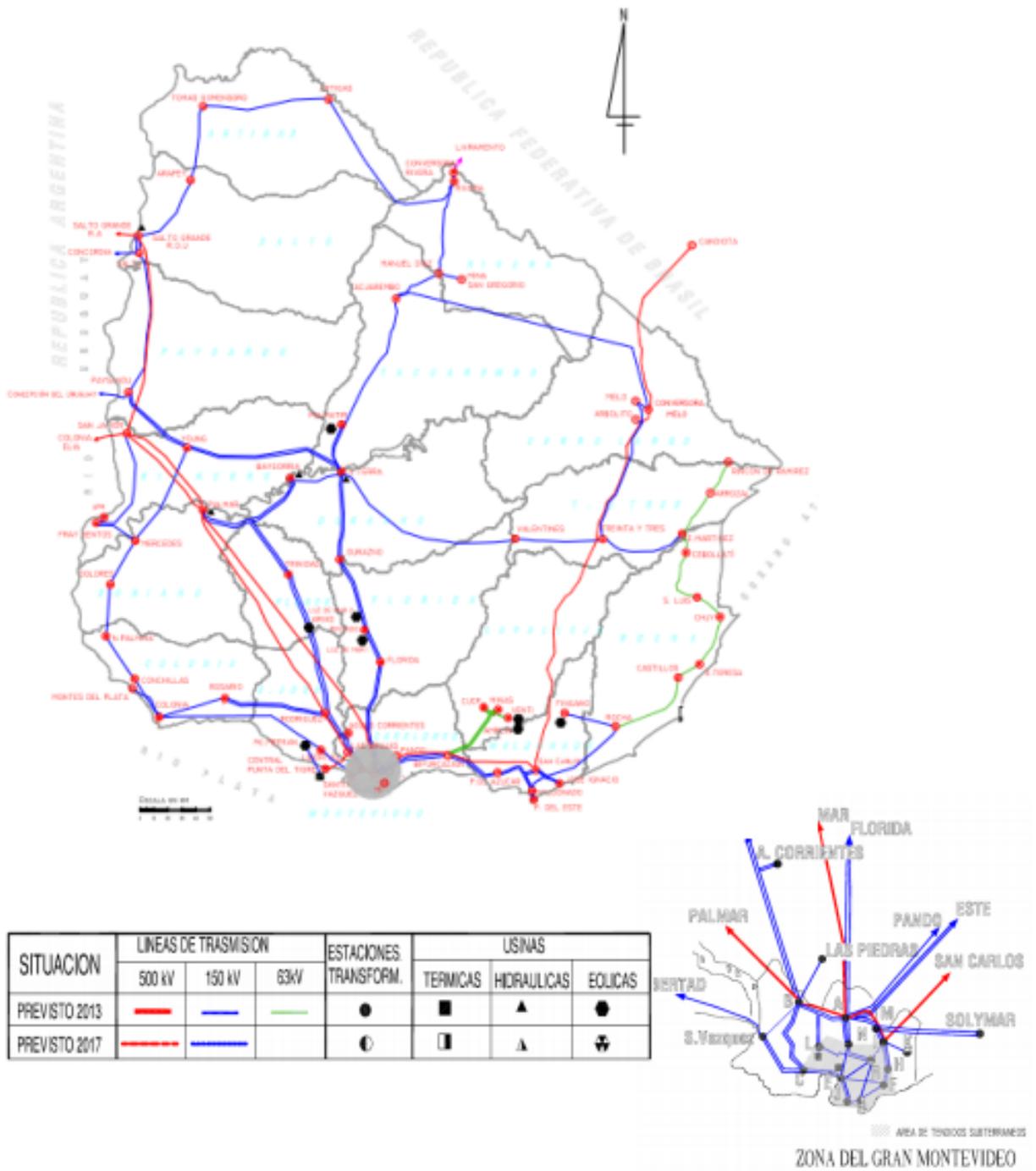


Figura 1.2: Distribución de líneas de 500kV, 150kV y 60kV [2].

La energía producida en las centrales de generación hidroeléctricas y térmicas complementadas por la recibida a través de interconexiones internacionales, se transmite a través de redes de extra alta, alta y media tensión a más de 36.000 subestaciones de distribución con más de 4.000 MVA de potencia instalada [3]. El pico de demanda registrado para el año 2016 se da en invierno y es de 1964 MW lo que se aproxima a 2 GW de potencia demandada [4].

En la figura 1.3 se muestra un resumen de la generación eléctrica del Uruguay.

TECNOLOGÍA	POTENCIA INSTALADA (MW)	PORCENTAJE
Hidroeléctrica	1.538	39%
Eólica	1.294	33%
Térmica	627	16%
Biomasa	413	10%
Solar	79	2%
<b>TOTAL</b>	<b>3.951 MW</b>	<b>100%</b>

Figura 1.3: Potencia instalada por fuente a Diciembre del 2016 [2].

En la figura 1.4 se muestra la distribución de la generación en Uruguay.

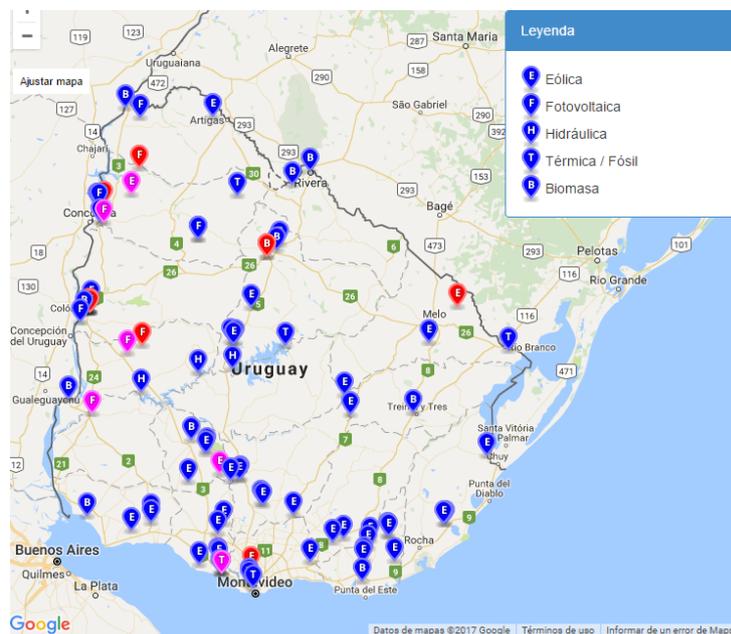


Figura 1.4: Distribución de la generación en Uruguay.

### 1.4.2. Argentina

Por otro lado, el sistema eléctrico argentino cuenta con diversas formas de generación eléctrica, donde predominan la generación térmica y la hidroeléctrica. Esto es propio de una red eléctrica de mucho mayor tamaño a la uruguaya, la cual debe optar por fuentes de generación de energía de mayor envergadura, como lo son las centrales de ciclo combinado o las centrales nucleares.

En la figura 1.5 se muestra un resumen de la generación eléctrica de Argentina.

TECNOLOGÍA		POTENCIA INSTALADA (MW)	PORCENTAJE	
Térmica	Ciclo combinado	9.227	20.083	45,94%
	Turbina de Gas	5.179		25,79%
	Turbo Vapor	4.451		22,16%
	Motor Diesel	1.226		6,10%
Hidroeléctrica		11.108	33,52%	
Nuclear		1.755	5,30%	
Eólica		187	0,56%	
Solar		8	0,20%	
<b>TOTAL</b>		<b>33.141 MW</b>	<b>100%</b>	

Figura 1.5: Potencia eléctrica instalada por fuente a diciembre del 2015 [5].

En la generación térmica el 46 % opera en la modalidad de ciclo combinado, la más eficiente desde el punto de vista operativo y ecológico, basada en el acoplamiento de dos ciclos diferentes, uno de turbina de vapor y otro de turbina de gas; el 22 % corresponde al tipo de turbo-vapor, el 26 % a turbo-gas y el 6 % restante a diésel (datos a Diciembre 2015). El gas es el principal insumo de las centrales térmicas, con un consumo de 12,6 mil millones de metros cúbicos en el año 2009, lo que representa el 78 % del total de combustible utilizado. El fuel oil y el gasoil representaron el 12 % y el 6 % respectivamente, mientras que el uso de carbón mineral fue del 4 % del total de combustibles.

En el área hidroeléctrica las centrales más importantes son Yacypetá (2.280 MW de potencia) y Salto Grande (945 MW), ubicadas en la Mesopotamia, mientras que en el área del Comahue en la Patagonia se destacan Piedra del Águila (1.400 MW), El Chocón (1.260 MW) y Alicurá (1.050 MW). Otra importante central hidroeléctrica es Río Grande, ubicada en Córdoba con 750 MW. La generación nuclear es aportada por las centrales de Embalse (Córdoba) que cuenta con 648 MW de potencia y Atucha (Provincia de Buenos Aires) con 745 MW. [6]

El pico de potencia demandada para el año 2015 es de 23949 MW lo que se aproxima a 23,4 GW de potencia demandada [5].

### 1.4.3. Interconexiones Uruguay-Argentina

La interconexión con Argentina no necesita de convertoras de frecuencia debido a que las dos redes funcionan a la misma frecuencia, 50Hz. Se conecta a través de Salto Grande en 500kV con

una capacidad de 2.000 MVA, en Salto 150kV con una potencia de 150 MVA, y a través de Paysandú en 150kV, con una capacidad de 52 MVA. El sistema de transmisión se encuentra formado por cuatro subestaciones de extra alta tensión (500kV) interconectadas entre sí, conformando el llamado “cuadrilátero” de Salto Grande.

Los cuatro vértices de este sistema de transmisión binacional son las subestaciones, que se encuentran ubicadas dos en el predio de la Central Hidroeléctrica que conforman los vértices superiores, una en Colonia Elía (Argentina) que concuerda con el vértice inferior izquierdo y una en San Javier (Uruguay) que concuerda con el vértice inferior derecho.

Desde el Centro de Control Unificado, ubicado en la margen izquierda del complejo, el anillo cuenta con un sistema de comunicaciones para efectuar las tareas de medición, protección y mando a distancia en las cuatro Subestaciones. Las restantes líneas y subestaciones fueron transferidas para su explotación a la Administración Nacional de Usinas y Transmisiones Eléctricas (UTE), de la República Oriental del Uruguay y a la Compañía de Transporte de Energía Eléctrica en Alta Tensión Transener S.A. (TRANSENER) de la República Argentina. Las salidas del anillo hacia las redes de Argentina y Uruguay, ya sea en 500, 150 o 132kV, constituyen las fronteras físicas entre Salto Grande y sus clientes de Argentina y Uruguay. Allí se realiza el registro de energía para la facturación mediante el Sistema de Medición Comercial (SMEC). Esta es una de las actividades de Salto Grande que en los mercados eléctricos es remunerada [7].

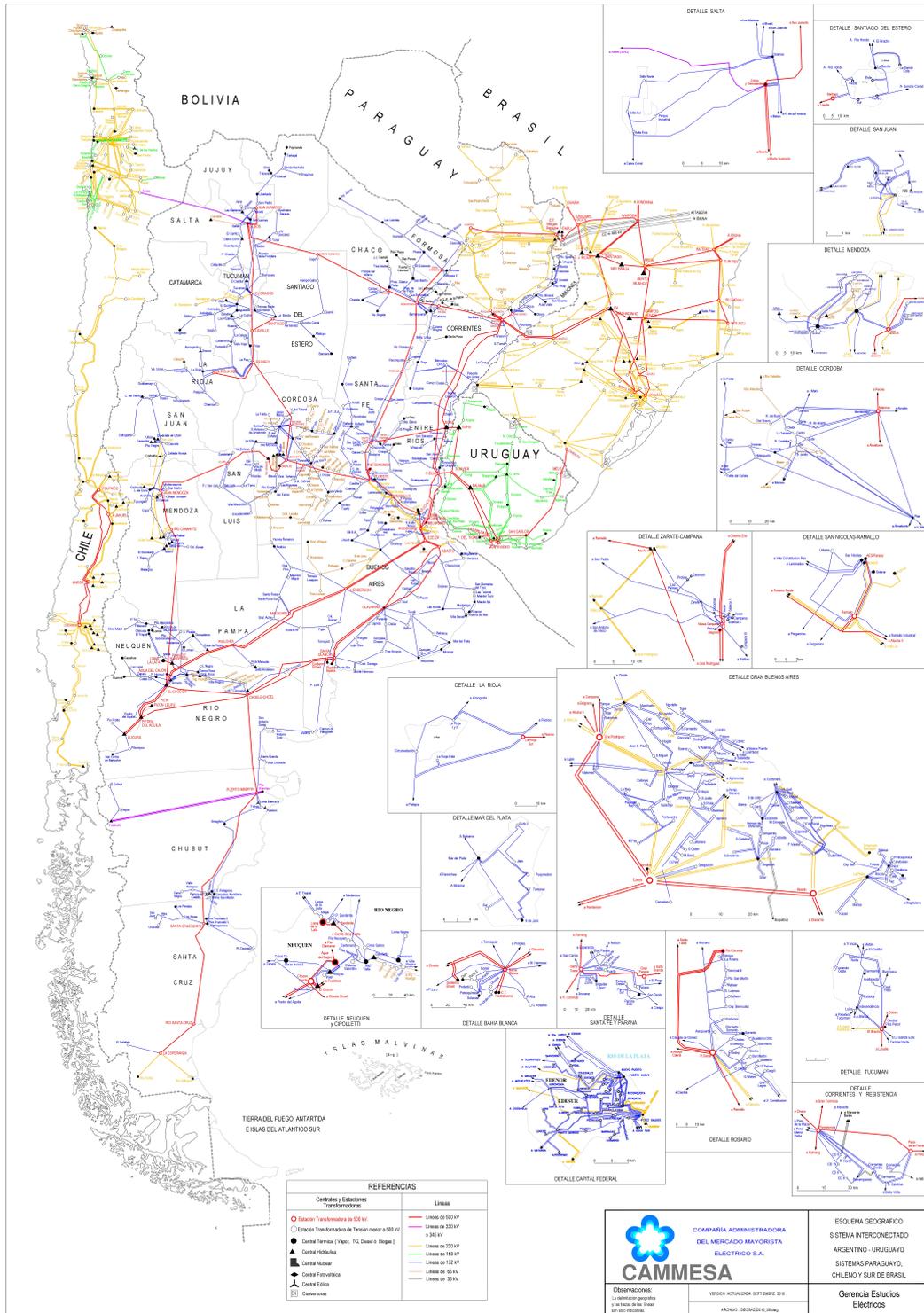


Figura 1.6: Esquema geográfico del sistema interconectado argentino-uruguayo. Sistemas paraguayo, chileno y sur de Brasil [8].

Como se puede observar en el unifilar regional 1.7, la líneas azules corresponden a la red argentina mientras que las líneas verdes corresponden a la red uruguaya. Esto nos da una idea comparativa de ambos sistemas. El sistema eléctrico argentino es aproximadamente 10 veces mas grande que el uruguayo.

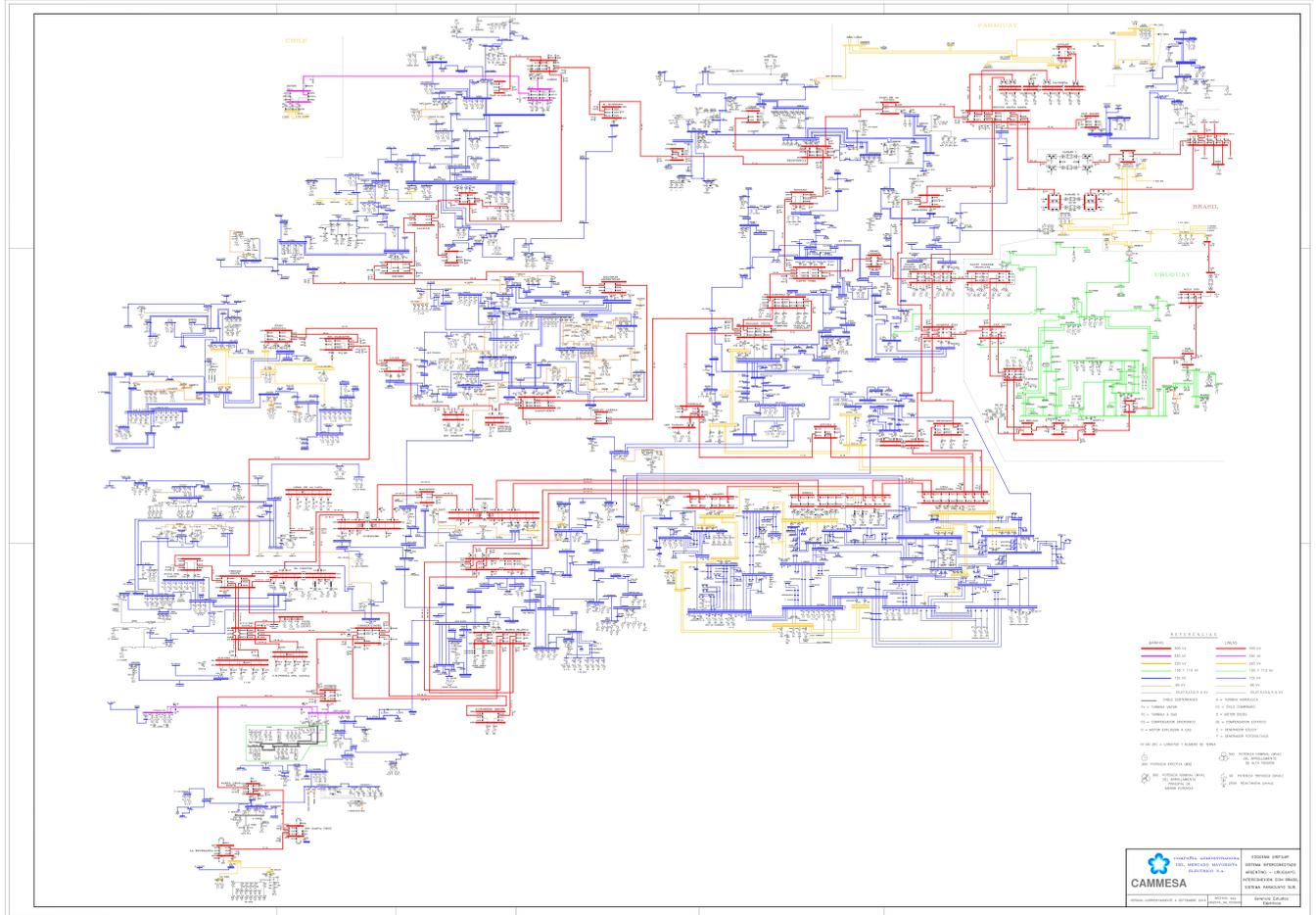


Figura 1.7: Esquema unifilar sistema interconectado argentino-uruguayo. Interconexión con Brasil. Sistema paraguayo sur [9].

#### 1.4.4. Interconexiones Uruguay-Brasil

La interconexión con Brasil se hace por medio de convertoras de frecuencia debido a que la red uruguaya funciona a una frecuencia de 50Hz y la red brasilera a una frecuencia de 60Hz. Se interconecta a través de Rivera-Livramento con una potencia de 70 MW en 150kV y a través de la Convertora de Melo.

La central Melo permite interconectar las redes de extra alta tensión de los dos países (Uruguay y Brasil), un programa considerado de importancia estratégica para Uruguay a través del cual se interconectan las redes de 500kV. La estación Convertora tiene una potencia de interconexión de 500 MW, con posibilidades de ampliarse en el futuro.

Si bien ya existía desde hace tiempo una convertora en Rivera de 70 MW, con la de Melo se ha ampliado a 570 MW, lo cual ubica a Uruguay en una situación mucho más beneficiosa para el intercambio energético [10].

#### 1.5. Antecedentes

Si bien existe un gran número de estudios e investigaciones que abordan el problema de los equivalentes dinámicos para los sistemas eléctricos de potencia y estudios de estabilidad de redes, poco (quizás nada) se ha podido comprobar de la aplicación de algoritmos genéticos a la resolución de un sistema equivalente óptimo.

Por un lado, se encara el problema del equivalente dinámico desde un punto de vista diferente a los métodos tradicionales, y por otro, se decide aplicar la teoría de algoritmos evolutivos para intentar optimizar ese sistema equivalente previamente simplificado. Es por ello que vale la pena nombrar y explicar qué es lo que se ha hecho hasta el momento sobre ambos temas.

En primera instancia, existen métodos tradicionales de obtención de equivalentes dinámicos de sistemas de potencia, los cuales se abarcan brevemente más adelante en este texto, como por ejemplo el método de coherencia.

A continuación se describen algunos de los estudios más relevantes:

- EQUIVALENT MODEL OF THE ARGENTINIAN ELECTRICAL POWER SYSTEM FOR STABILITY ANALYSIS OF THE URUGUAYAN NETWORK. Autores: MSc. Ing. M.Artenstein, Senior Member, IEEE, y Dr. Ing. A.Giusto.

Este trabajo describe la construcción de un modelo equivalente del sistema eléctrico argentino para ser utilizado en el estudio del fenómeno dinámico del sistema uruguayo. Siguiendo la metodología de coherencia lenta, los generadores que oscilan coherentemente a las frecuencias naturales más bajas del sistema interconectado fueron identificados con la ayuda de sus respectivos métodos modales. Los correspondientes modelos lineales fueron construidos con modelos clásicos de generadores de segundo orden. En un segundo paso, los generadores coherentes fueron agrupados al nivel de sus nodos internos. Finalmente, fue reducido de nuevo con la ayuda de las técnicas lineales clásicas para el cálculo de equivalentes estáticos. El resultado final es un modelo equivalente del sistema argentino que incluye aproximadamente el 20 % de las máquinas originales y menos del 5 % de los nodos.

El modelo equivalente fue validado a través de la comparación de la respuesta transitoria del sistema uruguayo a un conjunto de faltas significantes [11].

- ANÁLISIS MODAL DEL SISTEMA ELÉCTRICO URUGUAYO. Autores: Dr. Ing. Alvaro Giusto, y Dr. Ing. Pablo Monzón.

Este trabajo describe el análisis modal del sistema eléctrico uruguayo. Los escenarios bajo estudio comprenden los máximos térmicos e hidráulicos previstos para el año 2010, incluyendo los generadores y grandes consumidores que han entrado en servicio durante los años 2006 y 2007. Abarca, asimismo, el escenario máximo correspondiente al año 2004, con generación hidráulica dominante. Los principales modos de oscilación son estudiados y discutidos en detalle, prestando particular atención a los modos asociados al sistema de generación del Río Negro y a Botnia que exhiben un amortiguamiento insuficiente. Los resultados del análisis modal fueron validados mediante el estudio de las respuestas transitorias frente a faltas significativas. Se describe también, brevemente, el efecto de la colocación de un Power System Stabilizer (PSS) en la central Terra, con el objetivo de amortiguar el denominado modo Río Negro y modos locales asociados [12].

- IDENTIFICACIÓN DEL EQUIVALENTE DINÁMICO DEL SISTEMA COLOMBIANO PARA REALIZAR ANÁLISIS DE ESTABILIDAD EN EL SISTEMA ECUATORIANO. Autores: Dr. Ing. Jaime Cepeda, e Ing. José Aimara.

Este trabajo presenta la metodología aplicada y los resultados obtenidos en la identificación de equivalentes dinámicos de sistemas eléctricos de potencia, aplicado al caso del sistema interconectado Ecuador-Colombia. El método se basa en un proceso interrelacionado que ejecuta simultáneamente rutinas programadas en los ambientes DIgSILENT PowerFactory y Matlab. La identificación paramétrica se aborda como un problema de optimización, resuelto con una variante tipo enjambre de la optimización metaheurística de mapeo media-varianza (MVMOS). Los resultados resaltan tanto el excelente desempeño de la metodología empleada en la identificación paramétrica como la importancia del empleo de equivalentes dinámicos para la disminución considerable del tiempo de simulación, con la menor pérdida de precisión posible [13].

## 2. Equivalentes Dinámicos

### 2.1. Definición de Equivalente Dinámico

El comportamiento de sistemas eléctricos de potencia en transitorios se describe mediante variables de estado y su dinámica mediante ecuaciones diferenciales no lineales que provienen de las ecuaciones que gobiernan el comportamiento de las máquinas eléctricas (principalmente de las máquinas sincrónicas) y de los modelados de algunas cargas (consumidores importantes, motores) o de otros elementos (compensadores estáticos) cuando esto se hace. Para el modelado de un sistema eléctrico se divide la red en dos zonas, la zona de estudio que comprende la red a estudiar donde se aplican las faltas y se trabaja con modelos lo más detallados posibles y la zona externa donde no se aplican perturbaciones y es posible hacer una reducción del modelo dinámico original.

Para el estudio de transitorios en sistemas eléctricos juega un papel importante la dimensión de la red a estudiar ya que realizar simulaciones dinámicas de redes muy extensas insumen tiempos de ejecución muy grandes y bases de datos enormes.

La interconexión de los sistemas eléctricos juega un rol muy importante a la hora de estudiar el comportamiento dinámico de los mismos por lo que no se puede hacer un estudio sin tener en cuenta la interconexión e interacción de los dos sistemas. La solución al problema de las dimensiones de los sistemas es encontrar un método de reducción sistemática del modelado del sistema externo de manera tal que la influencia del mismo sobre el sistema a estudiar se vea inalterada ante eventos que ocurran dentro del área de estudio. Un sistema de orden reducido del sistema externo que posea estas características es lo que se denomina *equivalente dinámico* [14].

### 2.2. Motivación de la búsqueda de Equivalentes Dinámicos

Algunas de las razones por las cuales se busca emplear equivalentes dinámicos son las siguientes:

Simplificar el análisis del sistema. Los modelos completos de sistemas físicos por lo general consisten en un gran número de ecuaciones diferenciales no lineales, en particular en sistemas eléctricos de potencia es común tener cientos de estas ecuaciones en la representación completa del modelo. Esto hace muy complicado de estudiar el comportamiento del sistema, el análisis y el diseño de controladores. Además el manejo de modelos complejos trae un costo computacional enorme asociado al manejo del gran número de ecuaciones. Otra limitante de trabajar con modelos extensos son los problemas numéricos que pueden resultar del mal condicionamiento del sistema de ecuaciones. Una causal frecuente de problemas numéricos es la presencia de patrones dinámicos con escalas de tiempo muy diversos. A esta condición se la conoce como rigidez del modelo.

Desde el punto de vista del análisis, diseño y simulación de los sistemas de control automático, es mucho más sencillo y computacionalmente más eficiente hacerlo en sistemas de orden reducido. Este punto es muy importante ya que cuando se analizan redes complejas, con gran cantidad de generadores y controles, tanto los costos computacionales como el tiempo de cálculo son muy grandes.

De esta manera es de gran ayuda utilizar modelos de orden reducido. Sin embargo, los resultados obtenidos no deben diferir de manera apreciable con respecto a los resultados del sistema

de orden completo. Por ende es importante evaluar la calidad del modelo teniendo en cuenta la desviación con respecto al modelo original.

Es común en ingeniería trabajar con modelos físicos simplificados. Un método común de simplificación es la linealización del sistema en torno a un punto de operación. Es importante tener ciertos recaudos cuando se toman conclusiones del sistema real a partir del modelo linealizado, sin embargo, muchas propiedades importantes del modelo no-lineal pueden inferirse a partir del análisis del modelo linealizado, en particular lo relativo a la estabilidad.

La principal ventaja de trabajar con sistemas de ecuaciones diferenciales lineales invariantes en el tiempo (LIT) es que las herramientas matemáticas disponibles para análisis y control de estos sistemas son más simples y más desarrolladas que para los sistemas no lineales variantes en el tiempo.

Cuando se trata de simplificaciones de modelos LIT frecuentemente se obtiene otro modelo LIT de orden reducido que mantenga ciertas propiedades cualitativas y cuantitativas claves del sistema original. Sin embargo, en muchos casos lo realmente necesario no es una simplificación del modelo en sí, sino un procedimiento simplificado para extraer la información cualitativa y cuantitativa requerida de manera eficiente y exacta para una situación particular.

Otra forma de simplificación explota el hecho de los sistemas lentamente variables en el tiempo. Se obtienen buenas aproximaciones mediante el análisis de las ecuaciones en momentos convenientemente espaciados en el tiempo, asumiendo que los parámetros variantes en el tiempo permanecen fijos en el tiempo dado.

Los enfoques para el análisis de sistemas de gran escala suelen ser basados en la intuición física y la experiencia sobre el sistema y/o dependiente de las propiedades matemáticas y numéricas del modelo. El primer enfoque utiliza nociones de la experiencia ingenieril sobre el modelo de red que se estudiará. El segundo enfoque se basa en conceptos matemáticos formales y manipulaciones numéricas que no dependen de la aplicación.

Los sistemas LIT de parámetros concentrados quedan representados mediante la forma normal de variables de estado:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du,\end{aligned}\tag{2.1}$$

donde las matrices  $A$ ,  $B$ ,  $C$  y  $D$  son de dimensiones  $(n \times n)$ ,  $(n \times m)$ ,  $(p \times n)$  y  $(p \times m)$  respectivamente. Siendo  $n$  el número de estados,  $m$  el número de entradas y  $p$  el número de salidas.

Si el modelo es irreducible, un modelo de orden reducido será:

$$\begin{aligned}\dot{z} &= A_\gamma z + B_\gamma u \\ y &= C_\gamma z + D_\gamma u,\end{aligned}\tag{2.2}$$

siendo  $A_\gamma$ ,  $B_\gamma$ ,  $C_\gamma$  y  $D_\gamma$  de dimensiones  $(r \times r)$ ,  $(r \times m)$ ,  $(p \times r)$  y  $(p \times m)$  respectivamente, donde  $r < n$ . El procedimiento para obtener el sistema reducido  $\{A_\gamma, B_\gamma, C_\gamma, D_\gamma\}$  a partir del modelo original  $\{A, B, C, D\}$  es lo que constituye la particularidad de un modelo [14].

### 2.3. Reducción de modelos en sistemas de potencia

Las redes eléctricas son consideradas como uno de los sistemas de control más grandes y complejos que existen en la actualidad. Están formados por elementos interconectados de distinta naturaleza como eléctricos, electrónicos, térmicos, hidráulicos, eólicos, fotovoltaicos, etc. Cada uno de ellos opera con distinta tecnología por lo que para la planeación, control y operación de dichos sistemas se requieren numerosos estudios.

Para abordar dichos estudios se hace uso de la teoría de equivalentes dinámicos logrando así reducir la complejidad del sistema a estudiar sin perder las características relevantes del modelo original.

El equivalente dinámico de interés debe ser especificado de acuerdo al estudio y necesidad del analista. Las especificaciones definirán el detalle del modelado de los elementos que intervienen en el caso base.

El equivalente dinámico se puede formar de diversas maneras: la obtención de un equivalente de menor orden que la red total, equivalente de un área geográfica de interés, equivalente formado por plantas significativas de interés, equivalente para análisis de las líneas que interconectan el sistema, equivalentes para estudio de sistemas de control automático, o bien una combinación de los mencionados.

Los sistemas eléctricos interconectados son sistemas de gran escala. Para efectuar estudios los cálculos son costosos desde el punto de vista computacional. Se pueden realizar estudios en tiempos razonables de ejecución dados los avances tecnológicos actuales, sin embargo, otro factor que afecta en los estudios de sistemas de gran escala es la enorme cantidad de resultados a analizar para tomar decisiones. Por esto es importante que a la hora de estudiar sistemas de gran escala se maneje la menor cantidad de información posible, y que la misma sea confiable. Con esto, valiéndose de herramientas computacionales, es posible realizar estudios menos trabajosos y muy confiables para la planeación y operación del sistema.

Para los estudios de sistemas de potencia se efectúan reducciones en la formulación a nivel de sistema, al agrupar nodos de carga y nodos de generación, y a nivel de elementos, en los modelados de generadores, sistemas de control y modelos de carga. Para dichas reducciones es importante que el modelo reducido arroje los mismos resultados de tensiones, ángulos de fase y flujo de potencia en líneas entre los nodos que se conservan, lo más cercanos posibles a los obtenidos en la red original.

Los métodos más usados para la reducción de nodos de carga son: *a)* el equivalente de Ward; *b)* el equivalente de REI (Equivalente Radial e Independiente).

Para la agrupación de generadores buscando formar máquinas equivalentes hay que hacer estudios más exhaustivos de la red a analizar, ya que deben tomarse en cuenta diversos factores como la estructura de la red, las condiciones de operación, el efecto de cada falla sobre los generadores, etc. En estudios dinámicos generalmente existe un grupo pequeño de generadores que juegan un rol importante en la respuesta a la perturbación y el resto de ellos con menor influencia. Este comportamiento permite emplear modelos detallados para los generadores cercanos a la falla mientras que para los demás generadores es posible utilizar modelos simplificados.

La reducción del modelo para el estudio de la estabilidad transitoria puede realizarse en varios aspectos, cada uno involucrando características particulares. Aunque el tiempo de ejecución no varía linealmente con la cantidad de ecuaciones a resolver, la reducción de un número considerable

de nodos de carga y de generación en un sistema de gran escala, puede representar un ahorro de alrededor del 75 % del tiempo ejecución con respecto al sistema original.

Para la utilización efectiva de un equivalente es importante definir un área de interés, donde es relevante tener detalles del comportamiento de todas las variables del sistema y los generadores, y un área externa que es la parte del sistema que se puede reducir mediante el uso de equivalentes dinámicos. Una manera común de proceder es considerar en detalle el sistema a estudiar, y representar los sistemas externos mediante equivalentes [14].

## 2.4. Métodos tradicionales de obtención de un equivalente

### 2.4.1. Equivalentes para el subsistema lineal

Los nodos se clasifican como esenciales y no esenciales de acuerdo con su importancia en la red a reducir. Si el equivalente va a ser interconectado con otra red u otro equivalente, entonces se toman los nodos frontera como esenciales. Si el equivalente no estará interconectado, entonces los nodos esenciales serán los generadores más importantes y todos los demás nodos e inyecciones se clasificarán como no esenciales. Esto se ilustra en la figura 2.1 [14].

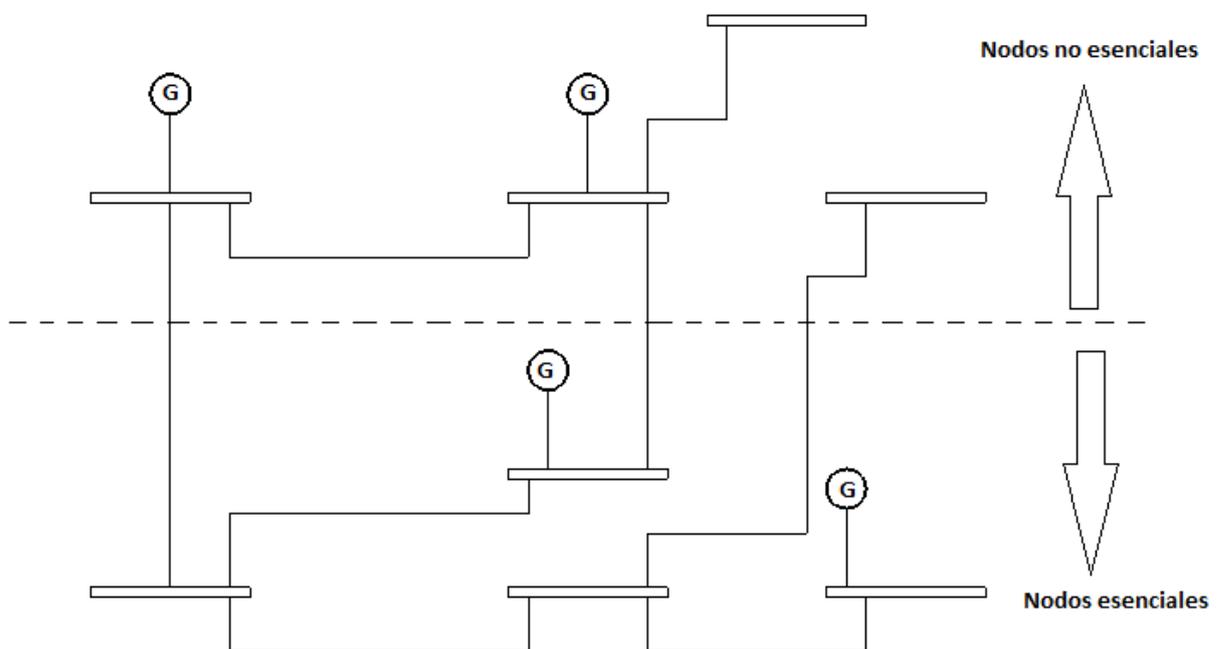


Figura 2.1: Red dividida por nodos esenciales y no esenciales.

### Equivalente Ward

La base para los equivalentes de tipo Ward es el modelado de la generación y las cargas del sistema externo como inyecciones de corriente constante.

Por más información ir a *Técnicas Modernas para el Análisis de Estabilidad Transitoria en Sistemas Eléctricos Interconectados Mediante la Aplicación de Equivalentes Dinámicos*, José Juan Manuel Ramírez Arreondo, Capítulo 2.3.1 [14].

## Equivalente REI

El concepto que se maneja a la hora de realizar el equivalente REI es *la red de balance de potencia cero*.

Estos equivalentes tienen como característica agrupar un conjunto de nodos en un nodo equivalente. En general los nodos son de carga o de generación. Para mostrar la formación de la red de balance de potencia cero, se asume un generador equivalente y una carga equivalente. No se restringirá el número ni la naturaleza de los nodos ficticios a agregar, lo que va a definir la formación del equivalente serán la definición y selección apropiada de estos nodos.

Por más información ir a *Técnicas Modernas para el Análisis de Estabilidad Transitoria en Sistemas Eléctricos Interconectados Mediante la Aplicación de Equivalentes Dinámicos*, José Juan Manuel Ramírez Arreondo, Capítulo 2.3.2 [14].

### 2.4.2. Equivalentes para el subsistema no-lineal

Los métodos tradicionales para obtener equivalentes dinámicos para estudios de estabilidad son tres: el enfoque modal que conserva los principales valores propios del sistema externo; el enfoque de coherencia que separa las máquinas en grupos y agrupa en un equivalente a aquellas que oscilan en fase; el enfoque de estimación que obtiene los equivalentes del sistema externo a partir de la estimación.

El propósito de estos métodos es encontrar generadores coherentes, es decir que se puedan agrupar en uno solo de manera equivalente. Esta coherencia se ve afectada por la configuración de la red, el lugar de la perturbación y las condiciones de operación [14].

### Enfoques modales

Los métodos modales se basan en la característica de que la respuesta dinámica total de un sistema lineal se compone de modos naturales del sistema.

Típicamente ciertos modos naturales dominan las respuestas porque la condición inicial es tal que favorece a ciertos modos más que a otros o algunos modos decaen lentamente debido que el valor propio que les corresponde tiene amplitud pequeña (o en casos de valores propios complejos tienen parte real pequeña).

De esta manera aquellos modos con poco peso en la solución final pueden despreciarse de manera que el conjunto final de soluciones es considerablemente menor que el inicial.

Por más información ir a *Técnicas Modernas para el Análisis de Estabilidad Transitoria en Sistemas Eléctricos Interconectados Mediante la Aplicación de Equivalentes Dinámicos*, José Juan Manuel Ramírez Arreondo, Capítulo 2.4.1 [14].

### Equivalentes basados en análisis de coherencia

Los equivalentes basados en métodos de coherencia han recibido mayor atención que los modales. La coherencia es el término utilizado para describir la tendencia de grupos de máquinas a oscilar en fase en el sistema eléctrico. Los rotores de las máquinas sincrónicas tienen tendencia a comportarse con una dinámica de rotor coherente.

La utilidad del concepto de coherencia se debe al hecho de que pueden identificarse grupos de generadores que tienden a oscilar en fase bajo la mayoría de fallas lejanas eléctricamente. Para agrupar dichos generadores hay que observar en el tiempo el comportamiento de los ángulos de los rotores y decidir si estos son coherentes. Para ello se resuelven ecuaciones linealizadas que representan el comportamiento del rotor de las máquinas. Una vez que se tiene la solución en el tiempo se debe definir qué generadores son coherentes. Para ello se hace un uso de una medida de coherencia, definida fundamentalmente como la diferencia entre los ángulos de rotor de las máquinas que se quiere conocer si oscilan en fase.

De esta manera, una vez localizadas las máquinas coherentes del sistema externo, se agrupan en una única máquina equivalente por grupo reduciendo así la dimensión del sistema externo.

Por más información ir a *Técnicas Modernas para el Análisis de Estabilidad Transitoria en Sistemas Eléctricos Interconectados Mediante la Aplicación de Equivalentes Dinámicos*, José Juan Manuel Ramírez Arreondo, Capítulo 2.4.2 [14].

### **Equivalentes basados en métodos de estimación**

Este enfoque se basa en la utilización de estimación, empleando información sólo del sistema en estudio. En estos casos, a diferencia de los enfoques modales o coherentes, no se necesita información del sistema externo. Esto es muy útil, ya que el sistema externo suele tener muchas máquinas, cargas, líneas y datos difíciles de obtener. Este método se basa en cálculos en línea para estudiar la seguridad del sistema.

Por más información ir a *Técnicas Modernas para el Análisis de Estabilidad Transitoria en Sistemas Eléctricos Interconectados Mediante la Aplicación de Equivalentes Dinámicos*, José Juan Manuel Ramírez Arreondo, Capítulo 2.4.3 [14].

## 2.5. Variables dinámicas

Se definen las variables dinámicas que describen el comportamiento del rotor de la máquina mostrado en la figura 2.2 [15].

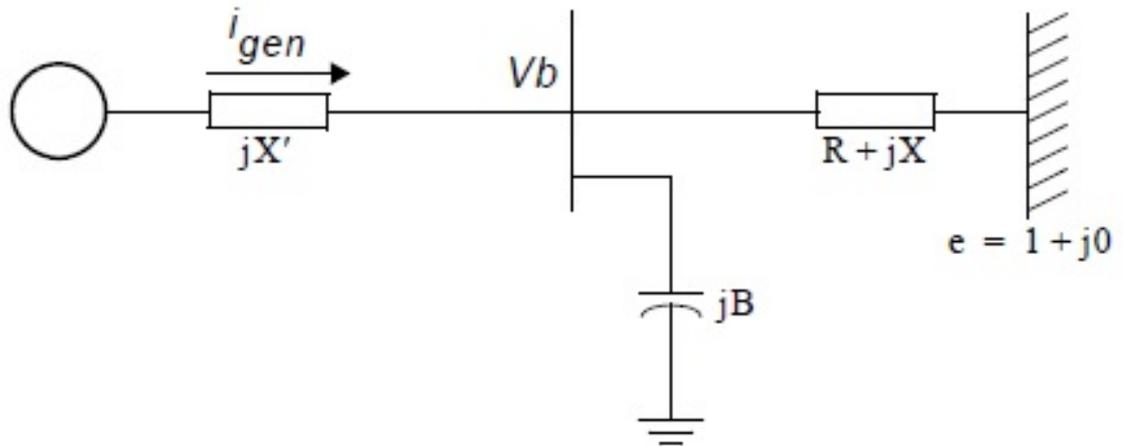


Figura 2.2: Sistema ejemplo.

Las ecuaciones diferenciales en por unidad que describen el comportamiento del rotor son las siguientes:

$$2H \frac{dn}{dt} = P_m - P_e - D_n \quad (2.3)$$

$$\Omega(t) = \Omega_s + \frac{d\delta}{dt} = \omega_0 n, \quad (2.4)$$

donde la tensión de la máquina está dada por  $E \angle \delta$ , la inercia por  $H$  y el damping por  $D$ .

Las ecuaciones descritas anteriormente involucran las constantes  $H$ ,  $D$  y  $\omega_0$  y, adicionalmente, en la ecuación 2.3 la potencia eléctrica del generador  $P_e$  y la potencia mecánica en el eje del generador  $P_m$ . En la ecuación 2.4  $\Omega(t)$  es el valor del ángulo de la máquina para un instante de tiempo determinado. El valor de  $P_e$ , en el instante que las variables  $n$  (velocidad) y  $\delta$  (ángulo del rotor) son dadas, requiere la solución de la red eléctrica para la corriente de generador  $i_{gen}$ . Mientras sea posible escribir una ecuación algebraica de soluciones para la corriente y la potencia, es más práctico manejar la solución de la red de manera numérica. Esto se hace con la siguiente secuencia de cálculos:

1) Calcular  $v_b$  a partir de la siguiente ecuación:

$$v_b = \frac{\left(\frac{E'\angle\delta}{jx'} + \frac{e}{R+jX}\right)}{\left(jB + \frac{1}{jx'} + \frac{1}{R+jX}\right)}. \quad (2.5)$$

Calcular  $i_{gen}$  a partir de:

$$i_{gen} = \frac{E'\angle\delta - v_b}{jx'}. \quad (2.6)$$

2) Calcular  $P_{gen}$  de:

$$P_{gen} = \text{Real}(E'\angle\delta \times i_{gen}). \quad (2.7)$$

3) Calcular las derivadas en el tiempo:

$$\frac{dn}{dt} = \frac{1}{2H}(P_m - P_{gen} - D_n). \quad (2.8)$$

El cálculo de las derivadas en función del tiempo implican el cálculo anterior de un conjunto de magnitudes ( $v_b$ ,  $i_{gen}$  and  $P_{gen}$ ) que están relacionadas algebraicamente con las variables  $n$  y  $\delta$ . Las mismas están dadas por el cálculo numérico. Las variables como  $v_b$ ,  $i_{gen}$  y  $P_{gen}$  son necesarias en la mayoría de las simulaciones dinámicas, por esto es conveniente clasificar las magnitudes involucradas en una simulación dinámica como:

- Constantes: parámetros que no varían durante la simulación
- Variables estáticas: variables que sus valores instantáneos están dados por una ecuación diferencial
- Variables algebraicas: variables que los valores pueden ser determinados si los valores de todas las variables estáticas, constantes y entradas son dados.
- Variables de entrada: cantidades que su valor está especificado para cada instante por una lógica que está afuera de la simulación dinámica

## 2.6. Modelado de cargas

El comportamiento dinámico de los sistemas está relacionado con la respuesta de las cargas, por esto es de suma importancia lograr representar su comportamiento de manera adecuada ya que influirá en los resultados que se obtendrán, y en consecuencia a las conclusiones a las que se llegará a partir del estudio. El comportamiento de las cargas, en términos generales, varía en función a la tensión y la frecuencia. Algunas formas comunes de modelar cargas son mediante potencia constante, corriente constante o impedancia constante.

El tipo de impedancia constante es el más utilizado (aunque no necesariamente la forma más exacta) en los estudios de estabilidad debido a la sencillez en la representación y a las

características de convergencia de los algoritmos de solución. La representación matemática para estas cargas se indica por la expresión

$$Y_c V = I, \quad (2.9)$$

donde  $Y_c$  permanece constante durante el estudio y se calcula con los valores de potencia y tensión complejos específicos previos a la perturbación.

Una manera cualitativa de explicar el efecto de esta manera de modelado es la siguiente. Se considera un nodo con tensión  $V$  y una carga  $P_L + jQ_L$ . Si la carga se representa mediante la admitancia estática  $G_L - jB_L$  se tiene:

$$G_L - jB_L = \frac{P_L}{V^2} - j\frac{Q_L}{V^2}. \quad (2.10)$$

Durante un proceso transitorio la tensión  $V$  y la frecuencia cambiarán, el cambio de tensión se refleja en las potencias activa y reactiva de la carga, mientras el cambio en la frecuencia del nodo de carga no se ve reflejado en las cargas. Este modelo asume  $P_L \propto V^2$ ,  $Q_L \propto V^2$ , donde las potencias son independientes de la frecuencia. Si tomamos que el proceso transitorio se inicia por una falla en el sistema de transmisión, inicialmente la falla causa una reducción de la potencia de salida en la mayoría de los generadores, provocando que sus rotores se aceleren y la frecuencia tienda a crecer. Al mismo tiempo, la falla en la red de transmisión generalmente causa una reducción en las tensiones nodales próximos a la falla. Para el modelo de admitancias constantes, la potencia de carga decrece notoriamente ( $P_L \propto V^2$ ), y el incremento de la frecuencia no causa un aumento en la potencia de carga. En sistemas reales la reducción de potencia no es proporcional a  $V^2$ , sino a un exponente menor a 2. De esta forma, el modelo da una potencia de carga menor que la esperada en el transitorio y mayor que la nominal después de despejada la falla.

Al modelar cargas y generadores equivalentes hay problemas para los cuales no se tiene una respuesta satisfactoria. Uno de los problemas que se encuentran es que las inyecciones equivalentes pueden no representar una realidad física y consecuentemente el manejar los nodos equivalentes como tipo (P-V), tipo (P-Q) o tipo compensador puede no tener una base real. Esto es en particular problemático en la respuesta de reactivos del equivalente para cambios en el sistema interno o en el sistema reducido. Una posible causa de esto es que por lo general la potencia reactiva se genera o absorbe de acuerdo con la ley de variación:

$$Q = KV^2, \quad (2.11)$$

mientras que la potencia real inyectada-tensión es relativamente plana [14].

## 2.7. Modelo clásico de sistemas multimáquina

Para encontrar la coherencia en sistemas multimáquina el modelo de generador usado es el clásico, debido a que las variables de interés son fundamentalmente los desplazamientos angulares de los rotores de las máquinas. La ecuación de oscilación es la siguiente [14]:

$$\frac{2H_i}{\omega_0} \ddot{\delta}_i(t) = P_{m_i} - P_{e_j}, \quad (2.12)$$

donde se desprecia la potencia de amortiguamiento, la potencia mecánica ( $P_m$ ) se asume constante en su valor de prefalla y las cargas se suponen de impedancia constante. Se considerará la velocidad angular  $\omega_0 = 1p.u.$  de tal forma que las potencias son numéricamente iguales a los pares, y se usa  $M = \frac{2H_i}{\omega_0} P_b$  donde  $P_b$  es la potencia base. Para calcular  $P_e$  dada una red de  $n$  máquinas y  $N$  nodos excluyendo el nodo de referencia (tierra) con las inyecciones de corriente de los generadores:

$$\begin{bmatrix} 0 \\ I_g \end{bmatrix} = \begin{bmatrix} Y_{il} & Y_{ig} \\ Y_{gi} & Y_{gg} \end{bmatrix} \begin{bmatrix} V_l \\ E_g \end{bmatrix}, \quad (2.13)$$

donde  $I_g$  es el vector de corrientes de generador de orden  $n \times 1$ ;  $V_l$  es el vector de la tensión terminal de generadores y buses de carga de orden  $N \times 1$ ;  $Y_{gl}$ ,  $Y_{gg}$ ,  $Y_{ig}$ ,  $Y_{ll}$  son submatrices de dimensiones apropiadas.

De la ecuación 2.13, se eliminan todos los nodos que no son de generación aplicando eliminación gaussiana. De esta manera, se obtiene la ecuación que vincula las tensiones de generación del sistema interno con las corrientes:

$$I_g = Y_{nodal} E_g, \quad (2.14)$$

donde  $Y_{nodal} = Y_{gg} - Y_{gl} Y_{ll}^{-1} Y_{lg}$ .

Es importante resaltar lo siguiente:

- La ecuación matricial 2.14 representa  $n$  ecuaciones algebraicas lineales, mientras que la ecuación de la red original 2.13 es de orden  $n + N$ . En general el número  $N$  es mucho mayor que  $n$ , y aquí es donde se ve la gran influencia que tiene la reducción de la red en los cálculos.
- Realmente no se realiza una inversión matricial para obtener  $Y_{nodal}$ , sino que se utiliza la reducción de Kron.
- Los elementos de  $Y_{nodal}$  incluyen las impedancias de carga en derivación y las reactancias transitorias de los generadores.
- En estudios de estabilidad hay tres periodos de interés: periodo de pre falla, periodo de falla y periodo de post falla. Para cada uno de estos periodos debe usarse una  $Y_{nodal}$  diferente que refleje los posibles cambios topológicos en la red.

De la expresión  $P_{e_i} = Re(E_i I_i^*)$  y usando 2.14 para sustituir  $I_i$  se obtiene  $P_{e_i}$ ,

$$P_{e_i} = \sum_{k=1}^n E_i E_k Y_{ik} \cos(\delta_i - \delta_k - \theta_{ik}), \quad (2.15)$$

donde  $E_i = E_i \angle \delta_i$ ,  $Y_{ik} = Y_{ik} \angle \theta$ , y  $Y_{ik}$  es el elemento (i,k) de  $Y_{nodal}$ . Así, la ecuación de oscilación de la  $i$ -ésima máquina tiene la forma:

$$\bar{\delta}_i = \frac{P_{m_i}}{M_i} - \sum_{k=1}^n \frac{E_i E_k Y_{ik}}{M_i} \cos(\delta_i - \delta_k - \theta_{ik}). \quad (2.16)$$

Para un sistema de  $n$  máquinas se requieren  $n$  ecuaciones similares a 2.16. Estas  $n$  ecuaciones son ecuaciones diferenciales no-lineales de segundo orden relacionadas entre sí. No existe una solución analítica para este conjunto de ecuaciones diferenciales. Comúnmente se integran las ecuaciones de oscilación numéricamente utilizando un sistema computacional.

Sería muy difícil y costoso computacionalmente identificar grupos coherente de generadores, mediante el análisis de curvas de oscilación obtenidas al resolver numéricamente el conjunto de ecuaciones diferenciales, la mayoría de las técnicas actuales para la identificación de coherencia hacen uso de la teoría de sistemas lineales y utilizan la versión linealizada de tales ecuaciones, haciendo uso de las siguientes consideraciones.

- a) Los grupos de generadores coherentes son independientes del tamaño del disturbio. Por tanto, la coherencia puede determinarse considerando un modelo linealizado del sistema.
- b) Los grupos coherentes son independientes del grado de detalle en los modelados de las unidades generadoras. Por tanto, se considera un modelo clásico de máquina síncrona y se ignoran los sistemas de control.
- c) El comportamiento coherente de un grupo de generadores no se afecta al cambiar el tiempo de liberación de una falla. Esto es válido para tiempos cortos de falla.
- d) Ya que los coeficientes de amortiguamiento  $D$ , que describen los pares producidos por los devanados del rotor, no afectan significativamente la frecuencia de los modos naturales del sistema, pueden despreciarse en el modelo clásico de la máquina.

## 2.8. EDCMAG: Equivalente Dinámico Calculado Mediante Algoritmos Genéticos

El método para encontrar el equivalente dinámico que se estudia en este proyecto no se encuentra en los que se detallan en la sección anterior, sino que busca el equivalente dinámico del sistema por un camino alternativo. En este caso se hará uso de algoritmos genéticos para la resolución del equivalente.

La red a estudiar es la red argentina y uruguaya que están interconectadas en el cuadrilátero Salto Grande Uruguay-Salto Grande Argentina (SU5-SA5) y San Javier-Colonia Elía (SJ5-CE5). El sistema interno se define como la red uruguaya de la cual se conoce en detalle su topología. El sistema externo, del que se busca un equivalente dinámico, es la red argentina de la que también se conoce en detalle su topología.

Dada una falla en el sistema eléctrico uruguayo, se realiza una simulación de la red completa a estudiar obteniendo el comportamiento de las variables eléctricas de interés (tensión, frecuencia y ángulo de generadores) en el transitorio (entre 5 y 20 segundos) en determinados puntos de la red eléctrica uruguaya. A partir de los resultados se ajustan los parámetros de un equivalente previamente propuesto mediante algoritmos genéticos. Se comparan las variables eléctricas obtenidas del sistema original con las mismas variables eléctricas obtenidas a partir del equivalente y a partir de las diferencias encontradas se ajustan los parámetros del equivalente.

Las variables a ajustar dependen de la topología del equivalente. Los componentes de la red que se ajustan son máquinas síncronas y líneas.

Este enfoque es un método totalmente diferente a los enfoques tradicionales de obtención de equivalentes dinámicos. Este método solo tiene en cuenta los comportamientos ante determinados fenómenos dinámicos, como lo son las faltas, obtenidos en el sistema original. El sistema original sólo se utiliza para obtener una topología adecuada para luego aplicar el algoritmo y comparar únicamente sus resultados.

## 3. Algoritmos Genéticos

### 3.1. Introducción

En los años 1950s y 1960s varios científicos computacionales estudiaron cómo se podrían utilizar los sistemas evolutivos como una herramienta para la optimización de problemas relacionados al campo de la ingeniería. La idea principal de esta técnica es dada una población de candidatos solución, hacer que se desarrollen tratando de emular la evolución genética y la selección natural, con la finalidad de arribar a una población final (o de descendientes) que se ajuste de mejor manera a la solución buscada.

Los algoritmos genéticos (AG) fueron inventados en 1960 por John Holland y desarrollados por sus estudiantes y colegas en la Universidad de Michigan en las décadas de 1960 y 1970. A diferencia de las estrategias de evolución y la programación evolutiva, la meta de Holland no era diseñar algoritmos para resolver problemas específicos, sino de estudiar formalmente el fenómeno de la adaptación tal como ocurre en la naturaleza y desarrollar maneras en las cuales los mecanismos de adaptación natural pueden ser importados a sistemas computacionales.

En 1975 el libro de Holland titulado “Adaptation in Natural and Artificial Systems“ (Adaptación en sistemas artificiales y naturales) presentó a los algoritmos genéticos como una abstracción de la evolución biológica y dio un marco teórico para la adaptación mediante los AG.

Holland presentó un algoritmo basado en una población con cruzamiento, inversión, y mutación la cual fue su mayor innovación.

Actualmente los investigadores utilizan el término algoritmo genético para describir algo un poco alejado de como lo había definido Holland originalmente [16].

### 3.2. Función de fitness

La palabra fitness se utiliza para hacer referencia a qué tan buena o cercana es una solución con respecto a la solución original. Por ejemplo, si se está optimizando la tensión en una barra y gráficamente el resultado, el fitness hace referencia a la cercanía de la gráfica de la tensión en la optimización del equivalente (mediante AG) con respecto a la gráfica de la tensión en la misma barra del sistema original. Lo que se busca es encontrar una función que muestre la proximidad entre lo que se está evaluando y la solución original. En nuestro caso a medida que se mejoran los individuos a través de los AG se mejora el fitness de los mismos (o sea sus valores se aproximan a los valores de los resultados del sistema completo) y eso se traduce en que los individuos se van aproximando a los valores originales (en términos de fitness, el mismo se va aproximando a 1). Uno de los aspectos más importantes de la eficacia del algoritmo es la elección de una función de fitness que se adecue de la mejor manera al problema. La función de fitness cumple el propósito de evaluar y medir el contraste entre cada individuo de la población y el resultado original.

Se debe tener en cuenta que el algoritmo genético maximiza la función de fitness, lo cual debe ser tenido en cuenta para nuestra aplicación, ya que debemos minimizar la distancia entre las curvas obtenidas entre la simulación del sistema original y la simulación del equivalente dinámico con sus diferentes valores. El fitness es función de la diferencia (error) entre estas curvas lo cual debe ser minimizado, es decir, alcanzar el 0.

**Construcción de una función de fitness apropiada para el problema:**

- La función de fitness depende del problema y del criterio de optimización.
- Opera directamente sobre las soluciones del problema.
- Se deben considerar las restricciones del problema.
- Puede definir objetivos múltiples o incorporar sub-objetivos.
- Puede cambiar dinámicamente a medida que el problema se va resolviendo (problemas dinámicos).
- La función de fitness es una caja negra para un AG donde la entrada es una solución del problema y la salida es el Valor del fitness.
- La forma en que se produce la evolución se relaciona con el concepto de presión selectiva.
- La influencia de la función de fitness es fundamental para determinar los individuos a candidatos a sobrevivir (aquellos individuos a los que se aplicaran los operadores evolutivos).

**Escalado del fitness:**

Se suelen utilizar mecanismos avanzados que permitan mejorar la eficacia de la función de fitness. Uno de los mecanismo más utilizados es el escalado. El escalado busca solucionar dos problemas que tradicionalmente surgen cuando se utilizan mecanismos de selección proporcional a los valores de fitness:

1. Dominancia prematura: dominancia de individuos muy adaptados en las etapas tempranas del proceso evolutivo.
2. Caminata aleatoria: entre individuos con valores de fitness similares, que puede ocurrir en las etapas avanzadas de la evolución.

La idea consiste en resolver los problemas transformando la función de fitness, de forma de evitar las grandes o pequeñas diferencias de valores (según el caso).

**Métodos de escalado:**

1. Escalado lineal:

$$F_{escalado} = aF + b. \quad (3.1)$$

- Donde  $a$  es la pendiente de la aproximación lineal y  $b$  el término independiente,  $a$  y  $b$  son parámetros del modelo y dependen del problema.

2. Truncamiento sigma:

$$F' = F - (\bar{F} - c\sigma). \quad (3.2)$$

- Es una propuesta de Forrest (1985) que incorpora información sobre la variación de los valores de fitness de la población (antes de realizar el escalado) mediante la desviación estándar.
- La constante  $c$  se utiliza para definir un múltiplo de la desviación estándar de la población (usualmente  $c$  se utiliza  $c$  entre 1 y 3).
- Los valores negativos son ajustados a 0 en forma arbitraria.
- Después de aplicar el truncamiento sigma, se puede proceder a realizar el escalado, sin el riesgo de obtener valores negativos.

3. Escalado potencia.

$$F_{escalado} = F^k. \quad (3.3)$$

- Originalmente propuesta por Gillies (1985).
- El valor del parámetro  $k$  depende del problema a resolver.
- Se suele aplicar un enfoque de autoadaptación del valor de  $k$  durante la ejecución, con el objetivo de aumentar o disminuir el rango de variación de los valores de fitness, según sea necesario.

### 3.3. Operadores

La forma más simple de AG involucra tres tipos de operadores: selección, reproducción, y mutación.

#### 3.3.1. Selección

El objetivo de la selección va a ser mantener las características de aquellos individuos mejor adaptados, el mecanismo de selección determina en gran medida la manera en que el algoritmo explora y elige a los individuos más aptos para la reproducción.

La presión de selección (estrategias elitistas) es crítica para el funcionamiento del AG. Una presión de selección alta (aquella que elige a los individuos mas adaptados con mayor probabilidad) puede conducir a buenas soluciones mas rápidamente pero puede causar perdida de diversidad y que la búsqueda termine de forma prematura (en un óptimo local). Esta situación se denomina como convergencia prematura (el valor hacia el cual converge no es el óptimo del problema). Una presión de selección baja (no se estaría muestreando adecuadamente el sistema evolutivo) puede conducir a que la búsqueda avance mucho más lento de lo necesario. Lo adecuado es mantener un compromiso entre estas dos características. Una buena opción es utilizar una presión de selección baja para el principio del algoritmo y una vez que se identifican los candidatos más promisorios concentrarse en ellos y aumentar la presión de selección. En nuestro caso la selección se realizó mediante un sorteo probabilístico, teniendo en cuenta el peso relativo

del fitness de cada individuo. O sea a mayor fitness más probabilidad tiene el individuo de ser seleccionado. Luego de que los individuos son seleccionados, estos individuos se reproducen.

Un método común de selección en AG es la selección fitness-proporcional, en la cual el número de veces en que un individuo se espera que se reproduzca (probabilidad de selección) es igual a su fitness dividido la suma del fitness total de la población. (Esto es el equivalente a lo que los biólogos denominan "Selección de viabilidad"). De esta manera se puede definir la probabilidad de selección como:

$$ps_i = \frac{f(i)}{\sum_{k=1}^n f_k}. \quad (3.4)$$

Un método simple de implementación de la selección fitness-proporcional es el "Muestreo de la ruleta" (Goldberg 1989), que conceptualmente es equivalente a darle a cada individuo una porción de área de la ruleta igual al fitness del individuo, siendo el área total de la ruleta la suma de los fitness de toda la población. La ruleta se hace girar, y la bolilla va a frenar en alguna porción en forma de cuña, y el individuo correspondiente es seleccionado.

### 3.3.2. Reproducción

Este operador elige al azar un lugar e intercambia las subsecuencias anteriores y posteriores entre dos individuos para crear descendientes (Por ejemplo, la cadena 10000100 y 11111111 puede cruzarse luego del tercer lugar en cada uno y producir los dos descendientes 10011111 y 11100100). En el caso de este proyecto luego que los individuos son seleccionados se realiza un sorteo para elegir el punto en el cual los individuos (padres) van a realizar el intercambio de parámetros de las distintas maquinas y líneas que entran en juego en la optimización del equivalente, dando origen a dos individuos (hijos). Luego de que esto ocurre se tiene que ver que se hace con los padres debido a que la población debe permanecer con la misma cantidad de individuos de forma permanente. A continuación se sustituye la población inicial con la población de descendientes. Lo que puede ocurrir es que se pierdan de generación a generación individuos muy aptos (individuos de alto fitness). Por ende, otra forma (la cual es la que se utiliza en el método plantado en este proyecto) es la de reemplazar entre un 80 % y 90 % (Se eligió un 90 %) de la población que posea el fitness mas bajo con los descendientes recién creados lo cual mantiene a los individuos más aptos (de mejor fitness). Esta forma de reemplazo se denomina selección elitista. Se pudo observar que en nuestro caso ésta era la mejor forma de reproducción ya que se perdían individuos con elevado fitness luego de varias generaciones y esto repercutía en el fitness de la población y se producía un retroceso en el avance del algoritmo.

### 3.3.3. Mutación

Este operador selecciona aleatoriamente algunos de los parámetros en un individuo (Por ejemplo, la cadena 00000100 puede mutar en su segunda posición, produciendo la cadena 01000100). La mutación puede ocurrir en cualquier parámetro del individuo con cierta probabilidad, en general la probabilidad es muy baja (Por ejemplo: 0,001). En el caso del proyecto que se lleva adelante en este documento, se realiza una pequeña variación a lo propuesto por John Holland. Todos los individuos de la población tienen una posibilidad de mutar de 0,2. Luego de que el

individuo a mutar es elegido se busca de forma aleatoria un punto del mismo y se muta ese punto a su nuevo valor hallado aleatoriamente dentro del rango en donde opera dicho parámetro [16].

### 3.4. Descripción de cómo funciona un AG

Planteado un problema definido a resolver y una cadena de bits que representa el candidato a solución, un simple AG funciona de la siguiente manera:

1. Se comienza con una población generada aleatoriamente de  $m$  individuos (candidatos a solución del problema).
2. Se calcula la función de fitness  $f(x)$  de cada individuo  $x$  de la población.
3. Se repiten los siguientes pasos hasta que  $n$  descendientes hayan sido creados.
  - a. Se selecciona un par de individuos padres de la población actual, la probabilidad de selección es una función creciente del fitness. La selección se realiza con remplazo, esto significa que el mismo individuo puede ser seleccionado más de una vez para ser padre.
  - b. Con una probabilidad  $p_r$  (probabilidad de reproducción o cruzamiento), se realiza la reproducción en el par (de individuos) en un punto elegido aleatoriamente (elegido con probabilidad uniforme) para formar dos descendientes. Si no existiese reproducción se formarían descendientes que serían copias exactas de sus respectivos padres.
  - c. Se muta los dos descendientes en cada punto con una probabilidad  $p_m$  (probabilidad de mutación), y se ubica el individuo resultante en la nueva población.

Si  $n$  es impar, un miembro de la nueva población puede ser descartado de forma aleatoria.

4. Se reemplaza a la población actual con la nueva población.
5. Se vuelve al paso 2.

## 3.4.1. Diagrama de flujo

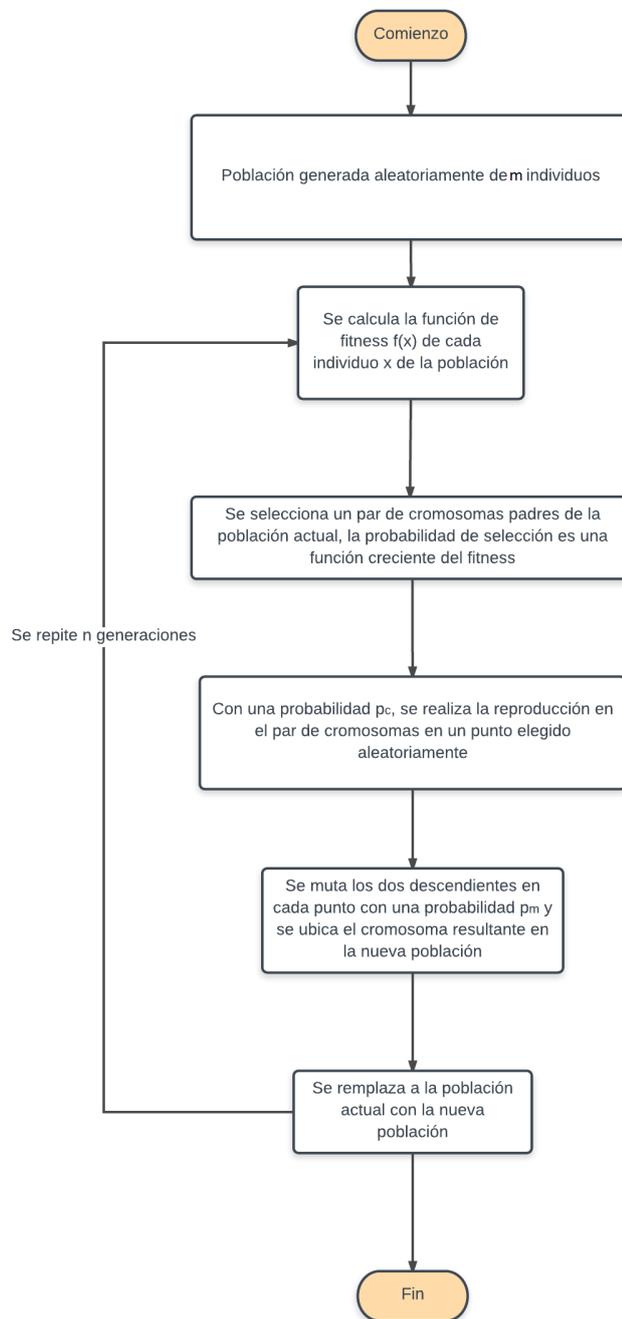


Figura 3.1: Diagrama de flujo para un AG genérico.

Cada iteración en este proceso se denomina generación. Un AG consiste de entre 50 a 500 o más generaciones. El conjunto entero de generaciones se denomina una corrida. Al final de cada corrida generalmente se encuentran individuos más aptos. La aleatoriedad juega un rol muy importante en los AG, cada corrida a partir de padres distintos producen distintos comportamientos. En general los investigadores de AG informan un promedio estadístico (como el mejor fitness hallado en una corrida y la generación en la cual el individuo con mejor fitness fue encontrado) sobre muchas corridas de AG sobre el mismo problema.

Este simple procedimiento recién descrito es la base para la mayoría de los AG. Existen una gran cantidad de detalles a ser completados, como el tamaño de la población, las probabilidades de reproducción y mutación, y el éxito del algoritmo depende en gran medida de estos detalles [16].

### 3.5. Población e individuos

A continuación se detalla en nuestro problema de optimización con AG, a que equivale cada variable que se utiliza.

- Cada máquina y línea que se modelan esta compuesta por variables  $v_{ij}$  y cada conjunto de estas variables (o sea todas las variables expresadas de todas las máquinas y líneas a modelar) corresponden a un individuo. El mismo se puede expresar de la siguiente manera:

$$(v_{11} \quad v_{12} \quad \cdots \quad v_{1l}),$$

donde  $l$  representa el largo total de las variables a optimizar.

- Al conjunto de estos individuos se le denomina población. Para la mayoría de nuestras simulaciones se tomó una población de 500 individuos, pero esto puede ser modificado por el usuario.

La población se representa de la siguiente manera:

$$\begin{bmatrix} (v_{11} & v_{12} & \dots & v_{1l}) \\ (v_{21} & v_{22} & \dots & v_{2l}) \\ (\vdots & \vdots & \ddots & ) \\ (v_{m1} & v_{m2} & \dots & v_{ml}) \end{bmatrix},$$

donde  $m$  representa la cantidad total de individuos en la población.

- La cantidad de generaciones (o iteraciones) que se realizan son también definidas por el usuario. En el caso de este proyecto se eligieron 50 generaciones y 500 individuos (tratando de mantener una relación de 1 generación cada 10 individuos). En la mayoría de los casos

no es necesario ir más allá de este número debido a que no mejoran los descendientes a partir de este punto (lo único que ocurre si se aumentan el número de generaciones es el aumento del tiempo de simulación).

### 3.6. Algunas aplicaciones de los AG

La versión descrita de los AG descrita anteriormente es muy simple, pero variaciones en el tema básico han sido usados por científicos e ingenieros en un gran número de problemas y modelos. Algunos ejemplos son dados a continuación [16]:

- Optimización: Los AG han sido utilizados en una amplia variedad de tareas de optimización, incluyendo optimización numérica, optimización combinatoria en problemas de disposición de circuitos y para programar talleres de trabajo.
- Programación automática: Los AG han sido utilizados para evolucionar programas de computadoras para tareas específicas, y para diseñar otras estructuras computacionales como autómatas celulares y redes de clasificación.
- Aprendizaje automático: Los AG han sido utilizados para varias aplicaciones de aprendizaje automático, incluyendo tareas de clasificación y predicción, como predicción del clima o la estructura proteica. Además han sido utilizados para evolucionar aspectos de sistemas particulares de aprendizaje automático, como el peso de redes neuronales, reglas para aprender sistemas clasificadores o sistemas de producción simbólica, y sensores para robots.
- Economía: Los AG han sido utilizados para modelar procesos de innovación, desarrollo de estrategias de pujas, y en la aparición de mercados económicos.
- Sistema inmune: Los AG han sido utilizados para modelar varios aspectos de sistemas inmunes naturales, incluyendo la mutación somática durante la vida del individuo y el descubrimiento de familias multi gen durante el tiempo evolutivo.
- Ecología: Los AG han sido utilizados para modelar fenómenos ecológicos tales como el desarrollo de armas biológicas, la co-evolución de parásitos huéspedes, simbiosis, y flujo de recursos.
- Genética de poblaciones: Los AG han sido utilizados para estudiar preguntas en la genética de poblaciones, tal como "bajo que condiciones un gen va a ser evolutivamente viable para la recombinación?".
- Evolución y aprendizaje: Los AG han sido utilizados para estudiar como el aprendizaje individual y la evolución de las especies se afectan unos a otros.
- Sistemas sociales: Los AG han sido utilizados para estudiar aspectos evolutivos de sistemas sociales, tales como la evolución en el comportamiento social en colonias de insectos, y, generalmente, la evolución de la cooperación y comunicación en sistemas multi-agentes.

### 3.7. Antecedentes de aplicaciones de AG a sistemas eléctricos

A continuación se nombra uno de los estudios que se han realizado hasta el momento aplicando AG a sistemas eléctricos:

- ALGORITMO EVOLUTIVO EFICIENTE APLICADO A LA PLANEACIÓN DE LA EXPANSIÓN DE SISTEMAS DE DISTRIBUCIÓN. Autores: Carlos García, Edwin García, y Fernando Villada.

En este trabajo se aborda el problema de la optimización de la planeación de la expansión de un sistema de distribución real de la ciudad de Medellín (Colombia). El sistema de distribución en cuestión cuenta con 155 nodos y 154 tramos a un nivel de tensión de 13200 volts. Para este fin se aplican algoritmos evolutivos eficientes, con el objeto de observar las mejoras y el desempeño que estos presentan frente a los algoritmos evolutivos convencionales. Además se plantea un modelo de planeación de la distribución mono-etapa y mono-objetivo, que tiene como función objetivo la optimización de los costos de construcción y las pérdidas técnicas del sistema. Finalmente se analiza el impacto que tienen los operadores eficientes diseñados especialmente para optimización, así como el análisis de los costos obtenidos [17].

## 4. Modelado del equivalente

Un equivalente dinámico, como se definió anteriormente, es una simplificación de un sistema más grande, el cual busca imitar el comportamiento del sistema original ante eventos dinámicos, fuera del régimen permanente.

En el caso de estudio, no se busca un equivalente único del sistema eléctrico argentino, o no es esa la intención en principio. El objetivo de este proyecto es encontrar un equivalente que se comporte de manera similar al sistema argentino ante una cierta perturbación, en un punto de la red y para una determinada magnitud. En la figura 4.1 se muestra un esquema de lo descrito. Resulta claro que si ese equivalente además puede aplicarse para representar otras magnitudes en otros puntos de la red, el objetivo habría sido alcanzado e incluso superado. Todo esto se analizará más adelante en el estudio de los resultados.

Por lo tanto, la idea del proyecto es buscar posibles equivalentes, los cuales son previamente representados en PSS®E, todo esto de forma manual, para luego optimizar sus variables con el algoritmo genético.

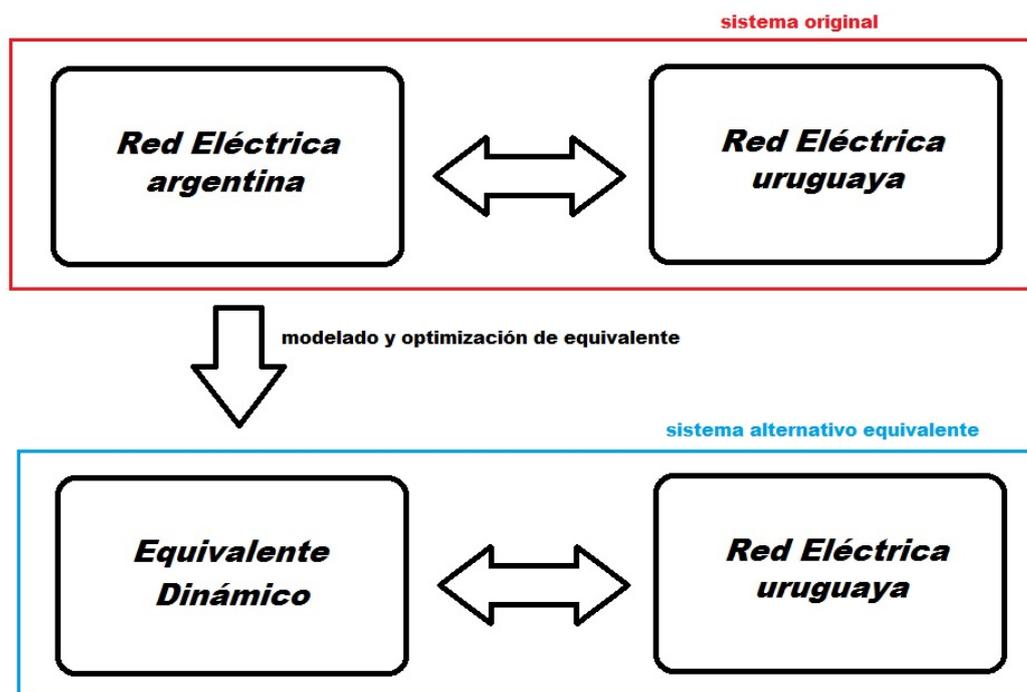


Figura 4.1: Esquema del modelado del equivalente.

De esta manera, se implementarán diferentes equivalentes, con diferentes grados de complejidad, y se evaluará su comportamiento, optimizando sus variables con el algoritmo genético, para luego analizar hasta qué punto podemos simplificar el sistema, manteniendo un margen de error tolerable.

Es importante notar cuáles serán las variables que se modificarán. Sobre esto, hay dos temas principales a considerar: no todas las variables del sistema influyen en las simulaciones dinámicas y no todas las variables del sistema que influyen, lo hacen con el mismo peso. En base a esto se dimensiona el tamaño de los individuos de la población, los cuales serán vectores de valores elegidos randómicamente entre márgenes predeterminados para esas variables el sistema.

#### 4.1. Modelos de generadores del PSS®E

El programa PSS®E cuenta con varios modelos de generadores, a saber:

- GENSAL y GENSAE: generador de polos salientes.
- GENROU y GENROE: generador de rotor liso.
- GENCLS: generador clásico.
- GENTRA: generador de polos salientes de nivel transitorio (Transient Saliency Level Generator Model).
- GENDCO: generador de rotor liso con modelado de efectos de offset DC en el estator (Solid Rotor Generator Model Including Stator dc Offset Effects).
- FRECHG: grupo motor-generador síncronos de polos salientes acoplados mediante un cambiador de frecuencia.
- CGEN1 y TGEN1: generadores síncronos con circuito equivalente de tercer orden (Third-Order Equivalent Circuit Model).
- CIMTR1 y CIMTR3: generadores de inducción (jaula simple y doble jaula).

Las máquinas que se utilizan en este trabajo fueron las de rotor liso, de tipos GENROU y GENROE, y las máquinas de polos salientes, de tipos GENSAL y GENSAE. Los modelos de las GENROE y GENROU tienen 14 constantes, ya que ambas representan una misma máquina, mientras que los modelos GENSAL y GENSAE cuentan con 12 constantes. Estas 12 constantes coinciden en ambos modelos, las de rotor liso y las de polos salientes. El modelo de rotor liso además cuenta con dos constantes adicionales  $T'_{qo}$  y  $X'_q$ .

La diferencia entre una máquina GENROE y GENROU (al igual que entre una GENSAE y una GENSAL) está en la curva de saturación de las reactancias sincrónicas  $X_d$  y  $X_q$  de cada modelo. Las curvas de saturación de las GENSAL y GENROU son del tipo cuadráticas, mientras que las GENROE y GENSAE tienen un tipo de curva exponencial, ambas funciones de  $\psi''$ . En la figura 4.2 se muestra la curva de saturación de máquinas del tipo GENSAL y GENROU.

A continuación se describen las 14 constantes de las máquinas:

- $T'_{do}$  [sec]: constante de tiempo transitoria en el eje d.
- $T''_{do}$  [sec]: constante de tiempo subtransitoria en el eje d.

- $T'_{qo}$  [sec]: constante de tiempo transitoria en el eje q.
- $T''_{qo}$  [sec]: constante de tiempo subtransitoria en el eje q.
- H [pu]: constante de inercia de la máquina
- D [pu]: speed damping, constante de los efectos de amortiguamiento de carga.
- $X_d$  [pu]: reactancia síncrona directa.
- $X_q$  [pu]: reactancia síncrona en cuadratura.
- $X'_d$  [pu]: reactancia transitoria directa.
- $X'_q$  [pu]: reactancia transitoria en cuadratura.
- $X''_d = X''_q$  [pu]: reactancias subtransitoria directa y en cuadratura.
- $X_l$  [pu]: reactancia del bobinado inductor.
- S(1,0): factor de saturación 1.0 de la curva.
- S(1,2): factor de saturación 1.2 de la curva.

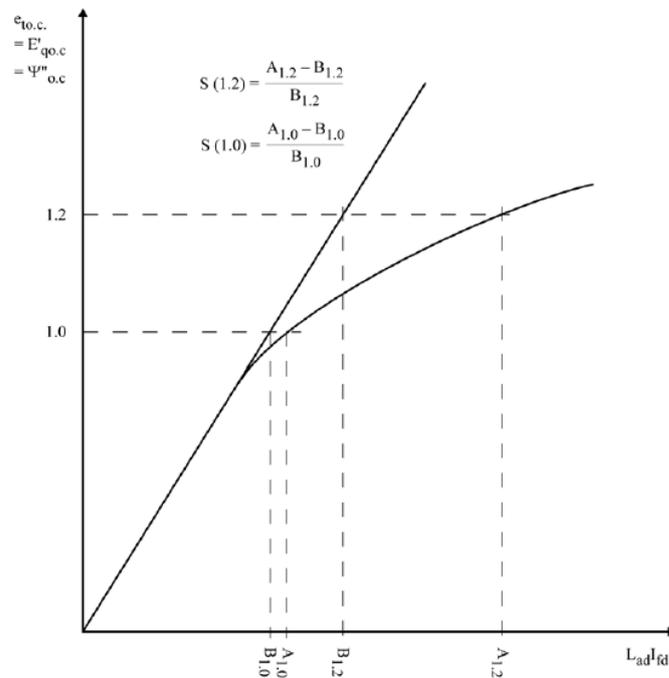


Figura 4.2: Definición de los factores de saturación S.

A su vez, las constantes deben cumplir ciertas restricciones para el correcto funcionamiento de los modelos. Estos también se especifican en la documentación del programa.

**GENROU, GENSAL, GENROE, GENSAE, GENDCO, GENCLS, GENTRA, FRECHG for Applicable Data**

$1 < H < 10$	$0 < S_{1.0}$
$0 \leq D < 3$	$S_{1.0} < S_{1.2}$
$1 < T'_{do} < 10$	$x''_d = \text{IMAG}(\text{ZSORCE})$ for GENROU, GENSAL, GENDCO, FRECHG
$4 \times \text{DELTA} < T''_{do} < 0.2$	$0.2 \leq T'_{qo} \leq 1.5$
$4 \times \text{DELTA} < T''_{qo} < 0.2$	$x'_q < x_q$
$x_d < 2.5$	$x'_d < x'_q$
$x'_d < 0.5 \times x_d$	$x''_d < x'_q$
$x_q < x_d$	$0.025 \leq T_a \leq 0.1$
$x'_d < x_q$ for GENSAL, FRECHG and GENTRA only	$x'_d = \text{IMAG}(\text{ZSORCE})$ for GENTRA
$x''_d < x'_d$	$0 < \text{Acceleration Factor} \leq 1.0$
$x_1 < x''_d$	$H_1 \times \text{MBASE}_1 = H_2 \times \text{MBASE}_2$ for FRECHG

Figura 4.3: Restricción de parámetros en máquinas.

De analizar la figura 4.3, se concluyen las restricciones que deberán cumplir las variables de las máquinas que usaremos:

$$X_l < X''_d < X'_d < X'_q < X_q < X_d \quad (4.1)$$

$$1 < H < 10 \quad (4.2)$$

$$0 \leq D < 3 \quad (4.3)$$

$$1 < T'_{do} < 10 \quad (4.4)$$

$$4 \times \text{PASO} < T''_{do} < 10 \quad (4.5)$$

$$4 \times PASO < T''_{qo} < 10 \quad (4.6)$$

$$0 < S(1,0) < S(1,2) \quad (4.7)$$

$$0,2 \leq T'_{qo} \leq 1,5 \quad (4.8)$$

Cabe mencionar que la mayoría de estas constantes se restringieron a estos entornos. Sin embargo, tanto para la inercia como para el damping, se llegaron a considerar valores máximos de hasta 40 para ambos, con mejores resultados para el equivalente y sin afectar la convergencia de las soluciones.

## 4.2. Modelos de líneas

El modelo de las líneas de transmisión para simulaciones dinámicas del programa PSS®E se realiza fundamentalmente con 3 constantes:

- $R$  [pu]: resistencia de la línea.
- $X$  [pu]: reactancia de la línea.
- $B$  [pu]: susceptancia de la línea, lo que equivale a la parte imaginaria de la admitancia.

En la figura 4.4 se muestran las variables a la hora de modelar una línea de transmisión.

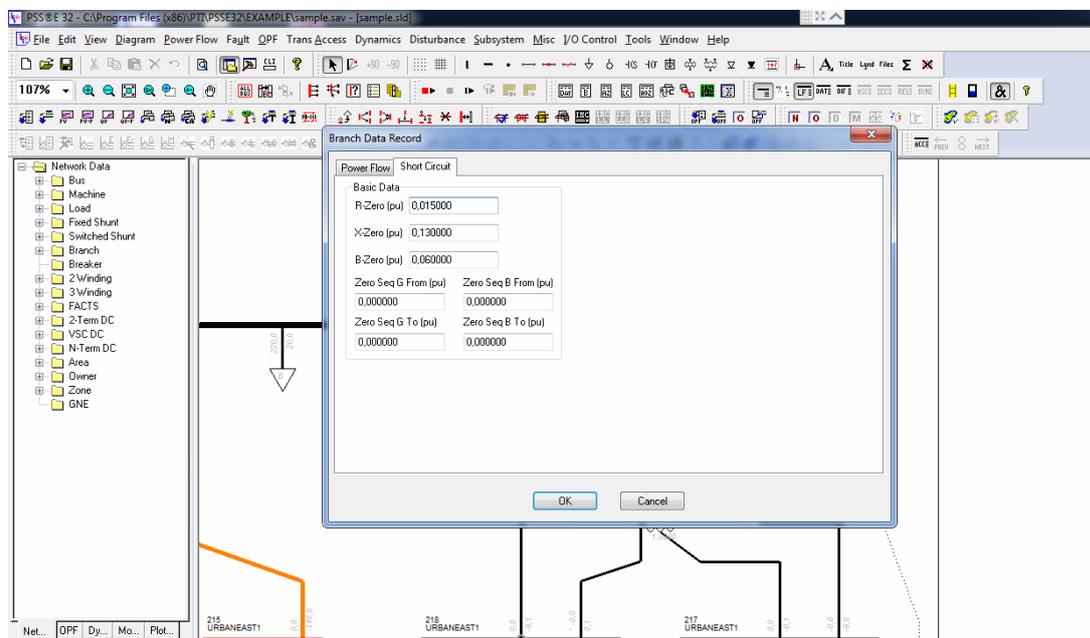


Figura 4.4: Variables de línea en PSS®E.

Por lo tanto, se sabe que la línea se modela como un circuito  $\pi$  mostrado en la figura 4.5.

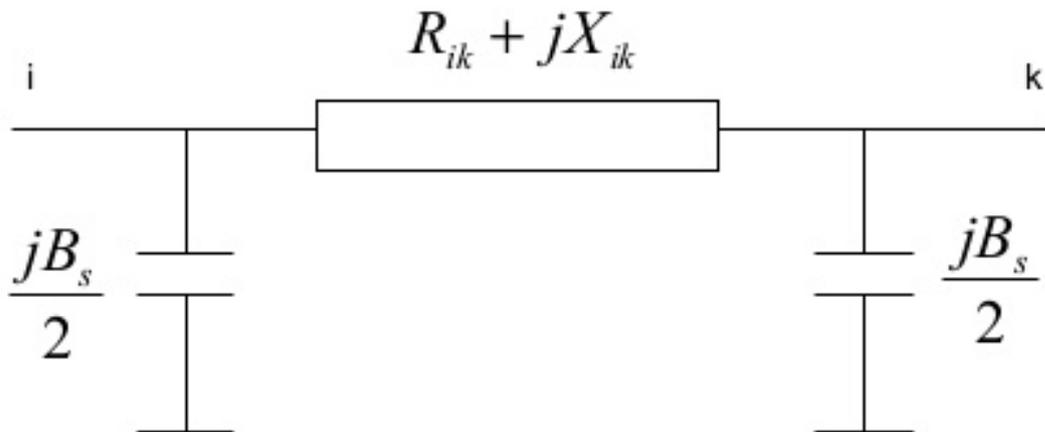


Figura 4.5: Modelo  $\pi$  de línea de transmisión.

Dependiendo del equivalente a optimizar se variarán estos parámetros de las líneas seleccionadas.

### 4.3. Variables a optimizar

Habiendo definido los modelos anteriores, se procede a describir las variables que se optimizarán para cada equivalente dinámico. Se consideran las máquinas y las líneas como los principales elementos del sistema que ayudarán a moldear el equivalente. También se tienen dos modelos de máquinas síncronas a considerar: las máquinas de polos salientes y las de rotor liso.

Es bueno aclarar que nuestras variables en el software de optimización corresponderán a constantes del sistema (máquinas y líneas), no a variables en sí del sistema. Es decir para cada simulación las constantes de las máquinas y líneas estarán dadas, luego de la simulación las constantes de las máquinas y líneas se ajustan en el software donde se corre el AG para correr la próxima simulación.

Por lo tanto, se intentará optimizar la mayor cantidad de constantes del sistema (que son variables en el software). Cada constante del sistema es una variable en el software que corresponde a un valor (o gen) de cada individuo, el cual a su vez es representado por un vector. El largo del vector se determinará por el número de variables que se quieran optimizar. Para el caso de las

líneas de transmisión serán 3 las variables a optimizar, para el caso de las máquinas GENROE serán 14 y para el caso de las GENSAL serán 12.

Uno de los mayores inconvenientes que se presentó al realizar el programa fue el de determinar los rangos asignados para cada variable. Estos se eligieron tomando en cuenta las restricciones descritas anteriormente, así como la puesta en práctica del programa y la búsqueda de valores típicos para cada una.

A continuación se describen los márgenes conseguidos para cada variable:

$$1 \leq H \leq 40 \quad (4.9)$$

$$0 \leq D \leq 40 \quad (4.10)$$

$$1 \leq T'_{do} \leq 10 \quad (4.11)$$

$$0,03 \leq T''_{do} \leq 0,2 \quad (4.12)$$

$$0,2 \leq T''_{qo} \leq 1,5 \quad (4.13)$$

$$0,03 \leq T''_{qo} \leq 0,2 \quad (4.14)$$

$$1,5 \leq X_d \leq 15,0 \quad (4.15)$$

$$0,71 \leq X_q \leq 1,5 \quad (4.16)$$

$$0,26 \leq X'_d \leq 0,35 \quad (4.17)$$

$$0,40 \leq X'_q \leq 0,60 \quad (4.18)$$

$$0,007 \leq X''_d = X''_q \leq 0,010 \quad (4.19)$$

$$0,001 \leq X_t \leq 0,007 \quad (4.20)$$

$$0,03 \leq S(1,0) \leq 0,05 \quad (4.21)$$

$$0,2 \leq S(1,2) \leq 0,9 \quad (4.22)$$

$$0 \leq R \leq 0,01 \quad (4.23)$$

$$0 \leq X \leq 0,01 \quad (4.24)$$

$$0 \leq B \leq 1,0. \quad (4.25)$$

Los márgenes podrán ser modificados por el usuario del programa. No se recomienda cambiar los márgenes descritos para las máquinas de manera considerable, ya que éstos juegan un rol importante en la convergencia del algoritmo y pueden provocar un error en tiempo de ejecución en python. Los márgenes de las líneas pueden cambiarse ya que no influyen en la convergencia del algoritmo, en caso de que se quiera considerar líneas más largas.

#### 4.4. Equivalente Reducido

En primer lugar se hicieron pruebas con un sistema simple. Este sistema cuenta con 2 máquinas GENSAL, 2 barras y 3 líneas, conectadas a la barra de Salto Grande Argentina y Colonia Elía. El equivalente se conecta en la red de  $500kV$  y se muestra en la figura 4.6.

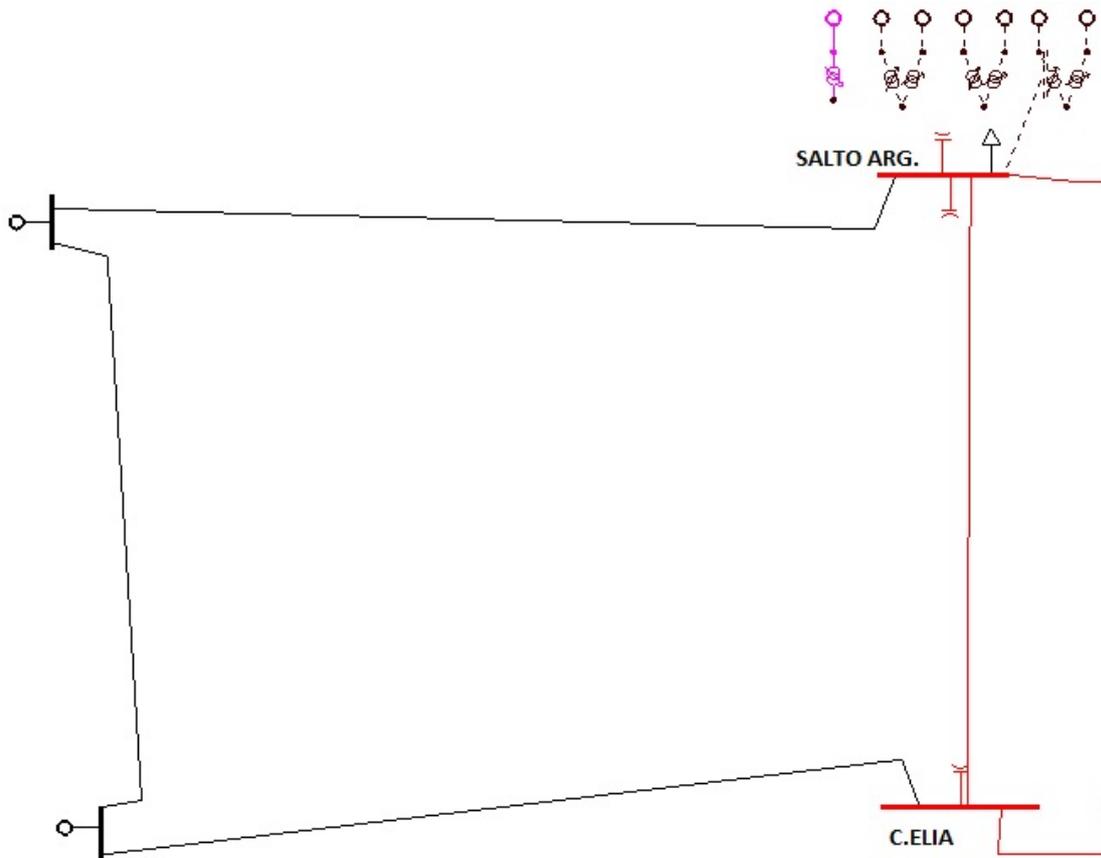


Figura 4.6: Primer equivalente utilizado.

De esta forma el equivalente cuenta con 33 variables: 24 para ambas máquinas y 9 para las 3 líneas.

Este sistema es un sistema muy simple para modelar el sistema argentino. Lo que se busca con este ejemplo es analizar el comportamiento de un sistema de estas características y estudiar hasta que punto se puede modelar el comportamiento del sistema eléctrico argentino mediante este equivalente.

#### 4.5. Equivalente NY

El segundo equivalente analizado se trata de una optimización de un equivalente existente, creado y utilizado por el tutor del presente proyecto para estudios de la red. El equivalente se muestra en la figura 4.7.

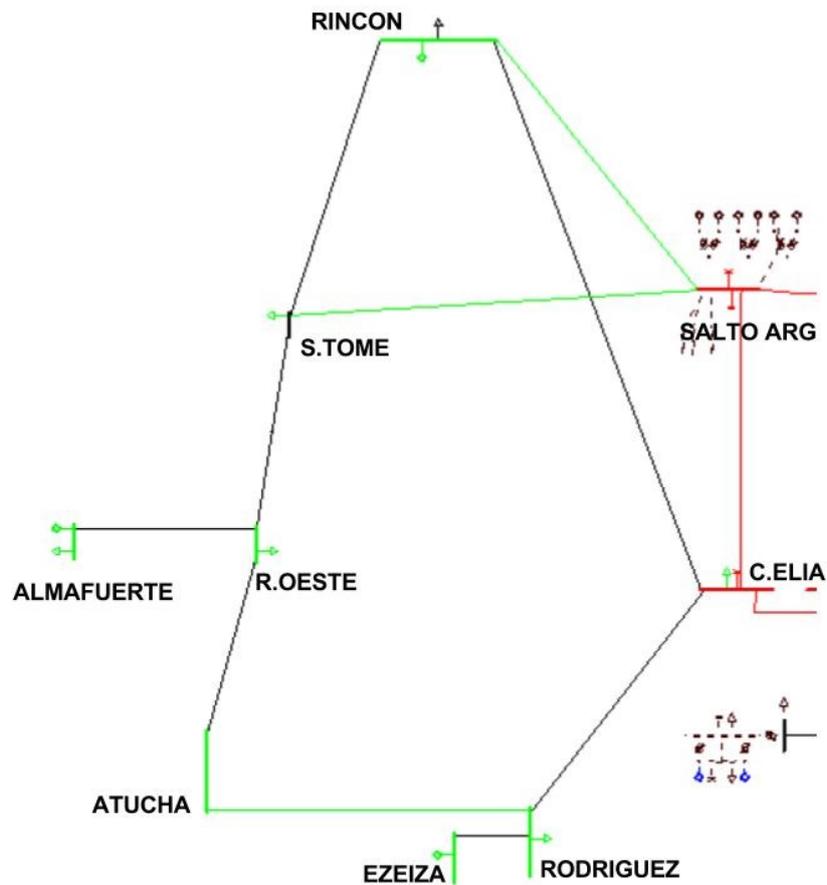


Figura 4.7: Equivalente NY.

De esta manera se estudia este equivalente dinámico buscando modelar el comportamiento del sistema eléctrico argentino dadas determinadas fallas en el sistema eléctrico uruguayo.

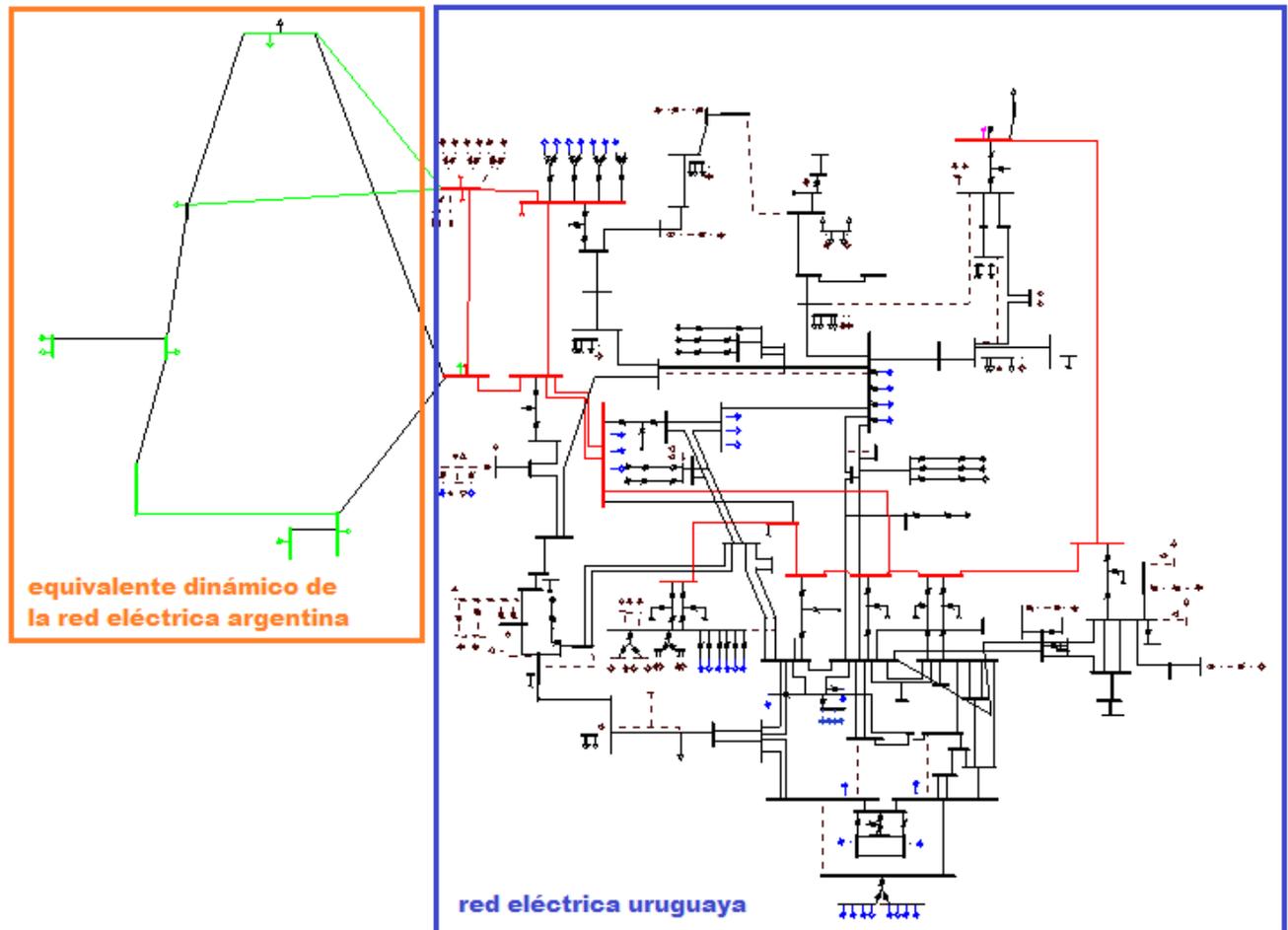


Figura 4.8: Sistema interno y equivalente.

En la figura 4.8 se muestra el equivalente dinámico que se ajusta con el programa de optimización. Se observa que el sistema propuesto es significativamente menor comparado con el sistema original. En el equivalente NY se ajustan 2 máquinas y 10 líneas. En primera instancia se ajustaron las 3 máquinas GENSAL pero, finalmente, se ajustan únicamente 2 máquinas GENSAL (Almafuerte y Ezeiza). En resumen, en términos de variables a ajustar, inicialmente se trabaja con individuos de largo  $12 \times 3 + 3 \times 10 = 66$  y finalmente con individuos de largo  $12 \times 2 + 3 \times 10 = 54$ .

#### 4.6. Archivos necesarios para cada equivalente

En la siguiente sección se describen los archivos necesarios para representar un sistema en el programa PSS®E. Para realizar simulaciones dinámicas se utilizan 3 archivos de cada caso:

- .sav (save case file): archivo binario con modelo del sistema y solución del flujo de carga guardado.
- .sld (slider binary file): archivo binario con el esquema unifilar asociado a un archivo .sav.
- .dyr (dynamics model data raw file): archivo binario con el modelo dinámico del sistema.

El archivo .sld es la representación del unifilar asociado al .sav, este archivo no es necesario para simular el sistema, pero facilita el entendimiento del sistema representándolo de manera gráfica.

El archivo .dyr es necesario para la realización de simulaciones dinámicas, ya que es de donde es el sistema toma los datos dinámicos de cada elemento del sistema. Este archivo no es necesario para realizar flujos de carga. Se debe modificarse manualmente (en un editor de textos) si se agrega un nuevo elemento al sistema para que pueda ser tomado por el programa de optimización.

## 5. Implementación del programa de optimización

En el presente capítulo se expondrá el desarrollo del software implementado para la realización del estudio. Como se explica anteriormente, el programa PSS®E no permite la realización de múltiples simulaciones, ya sea para diferentes modelos, o para iterar sobre uno específico. Asimismo, para poder aplicar la teoría de AG, es necesario realizar cálculos con los resultados obtenidos en las simulaciones. Esto conlleva el manejo de una gran cantidad de información que debe ser procesada y gestionada de manera ordenada. Para satisfacer los puntos anteriores, se debe adoptar una plataforma externa que realice las operaciones correspondientes y que también simule en el PSS®E, y la forma de realizarlo es a través de algoritmos escritos en lenguaje python, el cual cuenta con las características propias de un lenguaje de programación y además permite la utilización de módulos (conjuntos de funciones y rutinas) que pueden manejar externamente el programa PSS®E (específicamente el módulo psspy para python).

A continuación se describe el desarrollo de la herramienta de simulación generada para cumplir con el propósito del proyecto.

### 5.1. Versiones de software

Fue desarrollado en lenguaje python, en la plataforma Eclipse Luna, y ejecuta los modelos eléctricos en el PSS®E 32. Se utilizó la versión de python 2.5, ya que es la que mejor se adaptó a los requerimientos del PSS®E 32, y la cual está incluida en la instalación del programa. A continuación se detallan los programas y las versiones utilizadas:

- PSS®E 32 full commercial version de Siemens
- python 2.5
- Eclipse Standard 4.4 Luna R
- módulo numpy para python 2.5: numpy-1.4.1-win32-superpack-python2.5
- módulo matplotlib para python 2.5: matplotlib-1.1.0.win32-py2.5

Es de una gran relevancia la utilización de las versiones propuestas ya que otras versiones de estos programas han tenido problemas de incompatibilidad.

### 5.2. Desarrollo del programa

El programa principal estará formado por el código principal y el modelo de la red eléctrica en PSS®E. El código debe ser capaz de inicializar el modelo en el PSS®E, modificar sus constantes estáticas y dinámicas, realizar simulaciones de fallas y procesar los resultados en el programa para encontrar un modelo cercano al sistema original, pero de dimensiones mucho menores.

El algoritmo genético deberá evaluar y comparar el sistema original con el resultado de los diferentes modelos generados aleatoriamente, aunque esas aleatoriedades serán manejadas en cierto margen por el usuario. La función de fitness se encarga de minimizar la diferencia entre los resultados de las simulaciones de las fallas para los distintos modelos con el resultado de las fallas

en el sistema original. Se deberán evaluar distintas funciones de fitness y observar su desempeño, para luego elegir la que mejor ajuste las curvas.

### 5.2.1. Descripción del programa principal

El programa principal consiste en la combinación de dos algoritmos: un algoritmo genético general, el cual se adaptó para aplicar la función de fitness al resultado de la simulación de cada individuo generado en la población, y no al individuo en sí, como suele hacerse generalmente, y un algoritmo de simulación de los diferentes casos que se irán probando (una simulación de una falta en PSS®E y un algoritmo genético ordinario). Luego, se modifica el cálculo de fitness del AG, para poder comparar el resultado de la falta en el sistema original con los resultados obtenidos para el equivalente dinámico. Para ello, se intentó buscar una forma de comparar curvas que minimice la distancia entre ellas y le dé mayores probabilidades de "sobrevivir" a aquellas curvas que guardan una mayor similitud con la curva que arrojó la simulación del sistema original. Es un enfoque a resultados, que quizás pueda facilitar y ahorrar la difícil tarea de cálculo que imponen los métodos tradicionales. A su vez, el programa simula sobre dos sistemas eléctricos previamente definidos en PSS®E, por lo que se debe contar con los respectivos modelos y sus archivos para poder ejecutar la simulación.

## 5.2.2. Diagrama de flujo

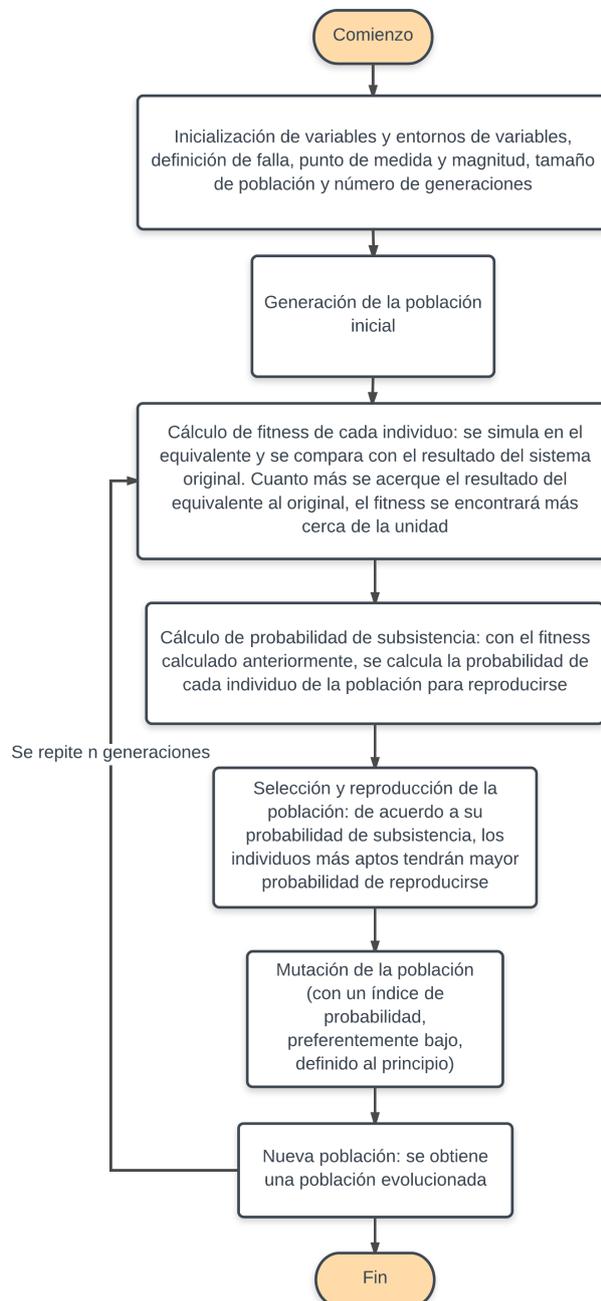


Figura 5.1: Diagrama de flujo del programa.

### 5.2.3. Pseudocódigo

-Inicializo las variables y los entornos. Defino también la magnitud que quiero optimizar y el número de barra.

-Subrutina de falla a probar: se define la falla que se busca optimizar.

-Subrutina de simulación de la falla: simula la falla en el equivalente propuesto con los valores de cada individuo. También simula la falla con otros sistemas que se quieran comparar, como el original u otro.

-Subrutina de generación de un individuo: genera un individuo con los entornos definidos al principio del programa para cada valor (gen).

-Subrutina de generación de la población: genera la población inicial, llamando a la rutina de generación de individuos el número de veces definido para el tamaño de la población.

-Subrutina de cálculo de fitness: calcula el fitness de cada individuo. Para ello, llama a la subrutina de simulación, obtiene la respuesta del equivalente para el individuo y luego calcula el fitness comparándolo con la simulación del sistema original realizada en un principio.

-Subrutina de selección y reproducción: esta subrutina se encarga de sortear los individuos que se van a reproducir. Para esto, primero llama a calcular\_fitness y luego se ordenan los individuos por su valor de fitness y se calcula la probabilidad de subsistencia de cada uno, respecto al resultado de fitness. Con esta probabilidad asociada a cada individuo, se sorteamos dos para su reproducción, teniendo en cuenta que no se repita el mismo. La reproducción resulta de la mezcla del material genético de ambos individuos. El "hijo" de estos dos individuos se incluye en la nueva población, reemplazando al individuo con menor fitness. La cantidad de reproducciones se fija a partir de la cantidad de individuos que quiero reemplazar de la generación. En nuestro caso se resolvió reemplazar el 90 % de los individuos con menor fitness, salvaguardando el 10 % que mejor se adapta. Este concepto se define como "elitismo" en la teoría de algoritmos genéticos. De esta forma, nos aseguramos de no perder el avance en la evolución al reemplazar a los mejores individuos.

-Subrutina de mutación de la población: la subrutina de mutación se encarga de mutar un gen al azar de algunos individuos de la población, elegidos randómicamente con una probabilidad definida. La posibilidad de mutar nos ayuda a encontrar nuevos valores óptimos, generando mayor diversidad a la población, y por ende, mayores chances de tener un caso exitoso.

### 5.3. Simulación de faltas

Como se describió en los puntos anteriores, el programa consta de dos algoritmos fundamentales: el algoritmo genético y un algoritmo de simulación de faltas en PSS®E. En este subcapítulo explicaremos de qué trata la simulación de faltas.

La simulación de faltas se realiza en el programa PSS®E de Siemens, la cual puede ser ejecutada tanto manualmente desde el mismo programa como a partir de un código escrito en lenguaje python, utilizando el módulo psspy, lo que corresponde a una especie de sublenguaje aplicado al PSS®E en python.

De esta forma, se puede automatizar la simulación de múltiples faltas combinando la aplicación del módulo psspy con python, creando programas que iteren sobre una misma simulación y luego procesen el resultado de una manera más práctica, con un mayor número de aplicaciones y resultados. Nuestro programa principal simula la misma falta repetidas veces (alrededor de 650 simulaciones por hora para nuestros equivalentes para fallas de 4 segundos, como las que se utilizaron en la mayoría de las simulaciones) sin la necesidad de estar operando sobre el PSS®E una y otra vez.

Este tipo de estudios son llamadas simulaciones dinámicas y el programa PSS®E ya cuenta con múltiples funciones y bibliotecas para realizar y estudiar este tipo de fenómenos en las redes eléctricas de potencia.

Procederemos a describir una falta genérica para luego pasar a las faltas efectivamente realizadas.

#### 5.3.1. Definición de una falta en un sistema eléctrico de potencia

La concepción clásica del análisis de los sistemas de potencia se basa en el estudio de dos condiciones temporales posibles: un régimen permanente y un régimen dinámico. Una falla es cualquier evento que interfiere con el flujo normal de corriente, colocando al sistema en un punto de operación fuera de lo normal, en un régimen dinámico. El software PSS®E permite el estudio de ambos regímenes.

En este estudio en particular nos interesa el análisis dinámico de un sistema de alta potencia, como lo es la interconexión de 2.000MW de Uruguay y Argentina, y la influencia de este último en el primero, el cual lo sobrepasa en aproximadamente 10 veces su potencia.

La búsqueda de un equivalente dinámico del sistema eléctrico argentino trata justamente la obtención de un sistema eléctrico, de menor porte, que se asimile al comportamiento dinámico del sistema original, es decir, al comportamiento del sistema cuando ocurren eventos anormales que interrumpen el estado permanente.

Generalmente el estudio de los sistemas de potencia en régimen permanente remite solamente a problemas donde se ameritan la resolución de ecuaciones algebraicas, mientras que los problemas de régimen dinámico implican la resolución de ecuaciones diferenciales en el plano temporal. Los grandes y complejos sistemas eléctricos de potencia requieren el planeamiento y resolución de un gran número de ecuaciones ya sean algebraicas o diferenciales, lo cual hace necesario el uso de un software para encontrar una solución [18].

### 5.3.2. Tipos de faltas eléctricas

Como se definió en el punto anterior, una falla, falta o defecto eléctrico es un evento anormal que interrumpen el régimen permanente del sistema eléctrico. Se clasifican las fallas en dos [19]:

- Fallas de aislación (shunt faults): cortocircuitos.
- Fallas de conducción (series fault): línea abierta.

Por lo tanto, se definen los cortocircuitos como todo incidente provocado por un contacto entre un conductor y la tierra o entre un conductor y otra pieza metálica conectada a tierra, o bien entre conductores. En la gran mayoría de los casos de instalaciones de AT, el contacto se realiza por medio de un arco.

### 5.3.3. Causas y consecuencias de faltas eléctricas

Las fallas eléctricas pueden ser ocasionadas debido a diferentes causas [19]:

- Origen puramente eléctrico: debido al deterioro en la aislación de algún equipo, el cual ya no soporta la tensión eléctrica.
- Origen mecánico: ruptura de conductores debido a otros eventos (caída de árboles, rotura de cables subterráneos debido a obras, robos o vandalismo).
- Origen atmosférico: causadas por eventos atmosféricos, como rayos que caen sobre una línea de alta tensión, vientos, humedad)
- Origen por transitorios: transitorios que produzcan sobretensiones elevadas que provoquen la perforación de la aislación.
- Origen por falsas maniobras: apertura accidental de un interruptor, puesta a tierra de un equipo energizado.

Las causas más comunes de las faltas se producen en las LAT (entre un 60 y un 70 %), debido a rayos que provocan sobretensiones y fallas en la aislación, caída de torres y árboles, acercamiento de diferentes fases debido a vientos, incendios o suciedad en aisladores o líneas. También suelen ocurrir fallas con menor frecuencia en otros elementos de la red eléctrica, como pueden ser cables subterráneos (entre un 10 y un 15 %), interruptores (entre 10 y 12 %), generadores, motores y transformadores (entre 10 y 15 %).

Como consecuencia de las diferentes fallas pueden darse los siguientes eventos:

- Circulación de corrientes mucho mayores a las corrientes nominales.
- Recalentamiento debido a las altas corrientes y daños causados por los arcos eléctricos.
- Ruptura de disyuntores u otros equipos.
- Efectos electrodinámicos en barras y bobinados de transformadores.

- Caídas de tensión elevadas.
- Cortes en el suministro de energía.
- Perturbaciones en los sistemas de comunicaciones.

La mayoría de estas consecuencias pueden llegar a ser devastadoras, con grandes pérdidas económicas. Es por ello que las empresas de transmisión de energía como UTE suelen tener partidas anuales predestinadas a cubrir pérdidas económicas debido a este tipo de eventos.

#### 5.3.4. Definición de faltas eléctricas en PSS®E

El programa PSS®E permite el análisis de los sistemas eléctricos de potencia, tanto en régimen permanente como en régimen dinámico. En este texto nos interesa únicamente el análisis dinámico.

A continuación daremos una descripción de las faltas y eventos típicos del PSS®E para la realización de simulaciones dinámicas las cuales se programan en Python con los siguientes comandos:

##### Comandos de faltas:

- `dist_3wind_fault`: falta trifásica en transformador de tres bobinados.
- `dist_branch_fault`: falta trifásica en una línea o un transformador de dos bobinados.
- `dist_bus_fault`: falta trifásica en una barra.
- `dist_scmu_fault`: falta desbalanceada y aplicación de admitancia de falta de secuencia positiva en una barra (falta fase-tierra o bifásica).
- `dist_spcb_fault`: falta desbalanceada y aplicación de modelo pi de secuencia positiva en una línea (falta fase-tierra o bifásica)
- `dist_clear_fault`: limpia una falta previamente ejecutada (`dist_3wind_fault`, `dist_branch_fault`, `dist_bus_fault`, `dist_scmu_fault` o `dist_spcb_fault`).

##### Comandos de apertura:

- `dist_3wind_trip`: saca de servicio un transformador de tres bobinados.
- `dist_branch_trip`: saca de servicio una línea o un transformador de dos bobinados.
- `dist_bus_trip`: saca de servicio una barra.
- `dist_machine_trip`: saca de servicio una máquina.

##### Comandos de cierre:

- `dist_3wind_close`: pone en servicio un transformador de tres bobinados.

- `dist_branch_close`: pone en servicio una línea o un transformador de dos bobinados.

El programa PSS®E trabaja sobre equivalentes monofásicos de elementos trifásicos de la red. Por este motivo los comandos se refieren a elementos trifásicos de la red, debiendo especificar cuando se aplican faltas monofásicas o bifásicas.

Otro punto a tener en cuenta es que cada comando tiene su propia especificación, con datos de entrada específico para cada caso.

Es claro también que estos comandos podrán combinarse para crear diferentes escenarios, creando otras faltas que puedan ocurrir, como es el caso del cortocircuito monofásico con reenganche, falta que se simuló para diferentes equivalentes, la cual se describe más adelante en este texto.

### 5.3.5. Información relevante sobre una falta genérica

En primera instancia, se deben elegir los siguientes puntos, los cuales variarán ligeramente el programa:

- Tipo de falta que deseamos probar: ésta puede ser un cortocircuito trifásico, bifásico, fase-tierra, fase-tierra con reenganche o un disparo de carga.
- Tiempo de medida: el tiempo de medida es el tiempo para el cual se comparan las curvas y se busca minimizar el error. Se debe elegir con en la inicialización del programa.
- Lugar donde ocurrirá la falta: se debe escoger un punto de la red eléctrica uruguaya donde ocurrirá la falta elegida en el punto anterior.
- Variable que deseamos medir y optimizar: frecuencia, tensión, ángulo o ángulo de alguna máquina.
- Lugar donde deseamos medir la variable anterior: este punto corresponde a una barra en la red uruguaya.

Con estos puntos definidos, procederemos a ejecutar la falta en el sistema original y ver el resultado obtenido. Este resultado será comparados con los resultados obtenidos en el equivalente dinámico a probar, optimizando diferentes variables desde el algoritmo genético, hasta encontrar una variante del equivalente dinámico que se comporte lo más parecido posible al original. El resultado numérico es un vector de largo igual al tiempo total de simulación dividido el paso elegido para el programa. El paso elegido dependerá de la aplicación, pero por el tipo de simulación se utilizará un paso de 0.001 segundos, paso utilizado por UTE para sus propias simulaciones dinámicas.

Por lo tanto, se realiza una primer simulación dinámica en el sistema original y luego se realizan  $p \times g$  simulaciones en el equivalente dinámico propuesto, siendo  $p$  el tamaño de la población y  $g$  el número de generaciones. Ambos valores,  $p$  y  $g$ , son inicializados en el programa principal.

En nuestro caso, se decidió implementar el programa para que finalice una vez hayan transcurrido las  $g$  generaciones. Sin embargo, podría haberse escogido un fitness mínimo a alcanzar para

que el programa finalice, la cual es otra forma de funcionamiento de los AG, no siempre aconsejable, ya que el fitness bien podría nunca alcanzar el mínimo y el algoritmo nunca terminaría, perdiendo los resultados hasta el momento obtenidos.

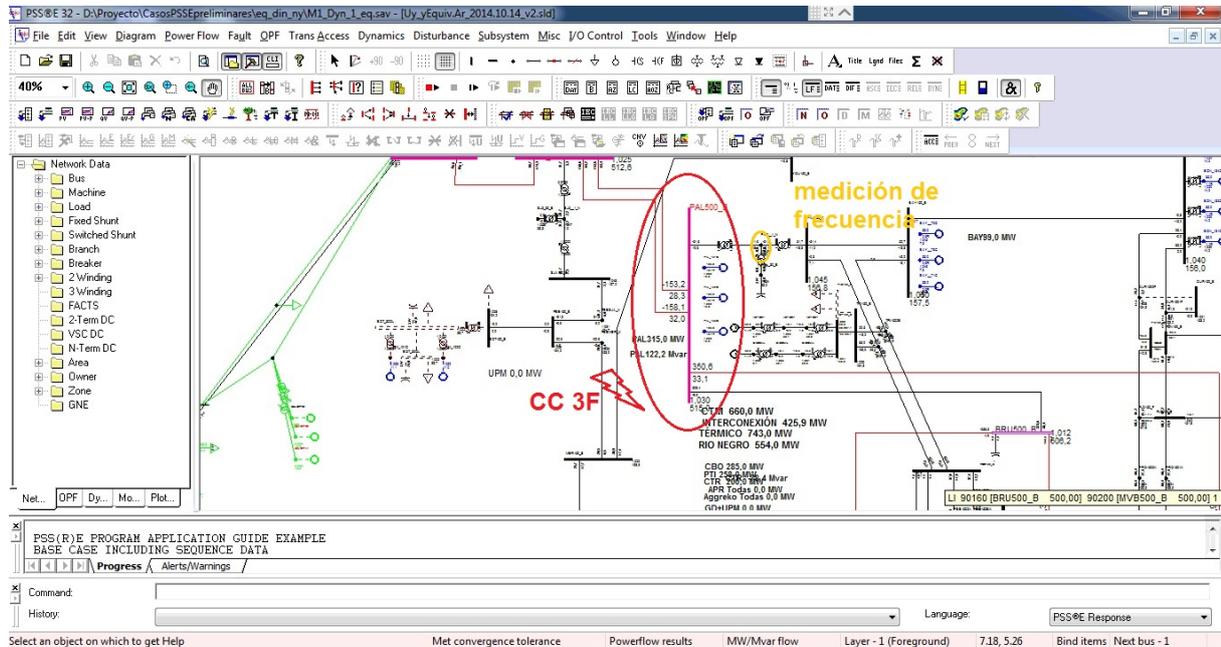


Figura 5.2: Captura del PSS®E indicando la barra donde se aplica la falta y la barra donde se mide la frecuencia.

Para graficar los resultados se utilizaron los módulos matplotlib, numpy y pylab de python.

En la figura 5.3 se puede observar el desvío en la frecuencia en la barra 99500 (en pu) en los primeros 3 segundos de un cortocircito trifásico en la barra 90500.

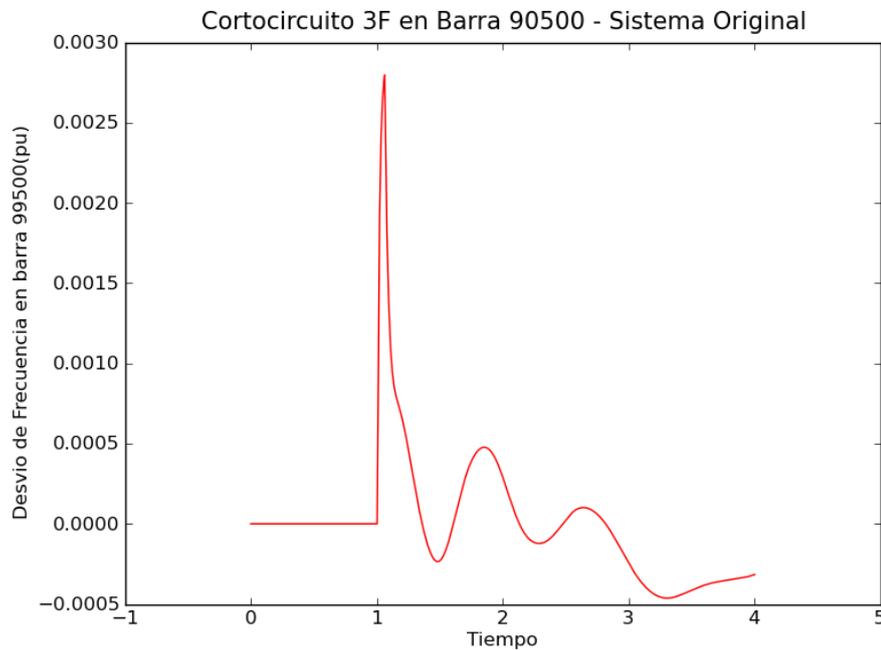


Figura 5.3:

### 5.3.6. Faltas simuladas

Para el estudio dinámico de nuestros equivalentes se eligieron 2 tipos de faltas a probar, en diferentes puntos de la red y optimizando diferentes magnitudes también en diferentes puntos.

- Cortocircuito trifásico con salida de una línea: esta falta simula un cortocircuito trifásico en una línea. La falta se ejecuta por 3 ciclos (60ms) y luego se despeja la falta y se abren las líneas, simulando la apertura de las protecciones.
- Cortocircuito fase a tierra con reenganche: esta falta describe el cierre y recierre luego de una falta monofásica. Se ejecuta la falta y luego se abre una fase durante 800ms. Luego se reengancha sobre la falta, ya que esta no se extinguió, por lo que el reenganche no fue exitoso. Finalmente se abren las tres fases 100ms después y se saca de servicio toda la línea.

Por lo tanto, se seleccionaron dos pruebas para el cortocircuito trifásico y una prueba para el cortocircuito monofásico, a saber:

#### Cortocircuito trifásico en PA5 con salida de línea PA5-MA5

# Falta Trifasica en PA5

```
def falla():
    psspy.dist_bus_fault(ibus=90500)
# defino tiempo de la falta (3 ciclos luego del primer segundo)
    time = 1 + 3.0/50.0
    psspy.run(tpause=time)
# despejo falta
    psspy.dist_clear_fault()
    psspy.dist_branch_trip(ibus=90500, jbus=90100, id='1')
```

### Cortocircuito trifásico en MA5 con salida de línea MA5-MI5

```
# Falta Trifasica en MA5
def falla():
    psspy.dist_bus_fault(ibus=90300)
# defino tiempo de la falta (3 ciclos luego del primer segundo)
    time = 1 + 3.0/50.0
    psspy.run(tpause=time)
# despejo falta
    psspy.dist_clear_fault()
    psspy.dist_branch_trip(ibus=90300, jbus=90100, id='1')
```

### Cortocircuito fase a tierra con reenganche en PA5 y posterior salida de PA5-MA5

```
#Falta Monofasica en PA5
def falla():
    psspy.run(0, 1.0,10,10,10)
    psspy.dist_scmu_fault([0,0,1,90500],[0.0,0.0,0.0,0.0])

    #abre 1fase durante 800ms
    psspy.run(0, 1.1,10,10,10)
#despeja falta
    psspy.dist_clear_fault(1)
#abre una fase (para psse abrir una fase es una falta)
    psspy.dist_spcb_fault(90100,90500,r""1"",[1,0,1,1,0,0],[0.0,0.0,0.0,0.0,0.0])

#cierra fase y reengancha sobre falta
    psspy.run(0, 1.9,10,10,10)
#cierra fase (para psse volver a cerrar la fase es despejar la falta de una
fase abierta)
    psspy.dist_clear_fault(1)
```

```
#aplico otra vez el cortocircuito (reengancha sobre falta, es decir es un
reenganche no exitoso)
    psspy.dist_scmu_fault([0,0,1,90500],[0.0,0.0,0.0,0.0])

#apertuta trifasica definitiva a los 100ms
    t_apertura=2.0
    psspy.run(0, t_apertura,10,10,10)
#despeja el cortocircuito
    psspy.dist_clear_fault(1)
#sale de servicio toda la linea (apertura trifasica definitiva)
    psspy.dist_branch_trip(90100,90500,r""1"")
```

#### 5.4. Algoritmo de optimización

El algoritmo de optimización propuesto se basa en la teoría de algoritmos genéticos, método de optimización que pertenece a la rama de algoritmos evolutivos. Los algoritmos genéticos se basan en la evolución de una población de individuos determinados, los cuales se reproducen y mutan, con determinadas reglas preestablecidas por el usuario. Estas reglas se definen de manera arbitraria de forma de adaptarse lo mejor posible al problema de aplicación.

En el problema actual de optimización se debieron probar diferentes variables del algoritmo, convergiendo a la solución que nos resultó de mejor aplicación, siempre teniendo en cuenta mayormente un enfoque de resultados.

Los principales aspectos que debieron definirse a lo largo del proyecto fueron los siguientes:

1. Generación de la población inicial
2. Prueba y elección de la función de fitness
3. Selección y reproducción de la población
4. Mutación de la población
5. Elección del mejor individuo/resultado

##### 5.4.1. Generación de la población inicial

La generación de la población inicial es realizada por una subrutina del programa para la cual deberá indicarse la cantidad de individuos que se desee simular. Cada individuo es un vector con una longitud igual al valor largo, inicializado al principio del programa. El valor largo se define como la longitud del material genético de cada individuo. A su vez, cada subvalor de cada individuo se sortea randómicamente dentro de un rango también predefinido en la inicialización del programa.

### 5.4.2. Prueba y elección de la función de fitness

Uno de los aspectos más importantes de la eficacia del algoritmo es la elección de una función de fitness que se adecue de la mejor manera al problema. La función de fitness cumple el propósito de evaluar y medir el contraste entre de cada individuo de la población y el resultado original.

Se debe tener en cuenta que el algoritmo genético maximiza la función de fitness, lo cual debe ser tenido en cuenta para nuestra aplicación, ya que debemos minimizar la distancia entre las curvas obtenidas entre la simulación del sistema original y la simulación del equivalente dinámico con sus diferentes valores. El fitness es función de la diferencia (error) entre estas curvas lo cual debe ser minimizada, es decir, alcanzar el 0.

En términos matemáticos, llamamos  $x$  a la suma de los errores entre la curva de la simulación original  $a$  y la curva de simulación de nuestro equivalente  $b$ . De esta forma, se definen matemáticamente las curvas y la variable  $x$ .

Sucesión de puntos de la curva resultante de la simulación de la falta en el sistema original:

$$a_i, i = 1, \dots, n.$$

Sucesión de puntos de la curva resultante de la simulación de la falta en el equivalente que se está probando:

$$b_i, i = 1, \dots, n.$$

Por lo tanto, se define el error (punto a punto) entre ambas curvas:

$$e_i = |a_i - b_i|, i = 1, \dots, n.$$

Y por último definimos la variable  $x$  como la suma de los errores punto a punto entre ambas curvas, variable que debe ser minimizada para obtener una mejor representación de la red argentina.

$$\sum_{i=1}^n e_i = \sum_{i=1}^n |a_i - b_i| = x, x \geq 0.$$

Vale la pena aclarar que el valor  $n$  se refiere al largo del vector resultado de cada simulación, el cual puede calcularse como:

$$n = \frac{t}{p} + 1,$$

donde  $t$ =tiempo de la simulación definido y  $p$ =paso temporal, o de muestreo, entre puntos (el cual fue definido como 1 milisegundo).

Finalmente, habiendo definido la variable que deseamos minimizar, debemos elegir una función que se maximice a medida que minimizamos  $x$ , o lo que es lo mismo, una función decreciente. Esta función también debe tener un máximo absoluto que sea único, para que el algoritmo no converja hacia puntos indeseados, y este máximo debería estar en el valor  $x=0$ , correspondiente al error mínimo que se puede obtener entre las curvas. Por último, también deseamos que cuanto mayor sea el error, más nos alejemos de nuestro punto óptimo, es decir, la función debe tender a 0 cuando  $x$  se hace grande.

Resumiendo las principales características que debemos encontrar para la función:

- Debe ser una función real y decreciente.
- Su máximo absoluto debe darse en  $x=0$  y debe ser único.
- Debe tener un comportamiento estable en torno a 0.
- Debe tender a 0 en infinito.

Se realizaron varias pruebas respecto a la función de fitness y finalmente se alcanzó una función que satisface todos los puntos necesarios para un correcto análisis. Algunas de las funciones probadas fueron:

$$f(x) = e^{-x}, x \geq 0 \quad (5.1)$$

$$f(x) = e^{-x^2}, x \geq 0 \quad (5.2)$$

$$f(x) = e^{-x^3}, x \geq 0 \quad (5.3)$$

$$f(x) = e^{-x^4}, x \geq 0 \quad (5.4)$$

$$f(x) = \frac{1}{1+x}, x \geq 0 \quad (5.5)$$

$$f(x) = \frac{1}{(1+x)^2}, x \geq 0. \quad (5.6)$$

Uno de los principales problemas fue el crecimiento acelerado hacia  $x=0$ , lo cual debió calibrarse casi de forma manual.

Finalmente, tras observarse el comportamiento de las distintas funciones, se llegó a la función:

$$f(x) = e^{-x}, x \geq 0. \quad (5.7)$$

En algunos casos fue necesario introducir un factor de corrección en el exponente (escalado) para que el fitness crezca de manera más lenta y aumentar la diferencia entre los individuos más aptos y los menos. Así fue que se introdujo el factor de fitness  $\alpha$ :

$$f(x) = e^{-\alpha x}, x \geq 0. \quad (5.8)$$

Este factor  $\alpha$  suele ser mayor que 1. Hemos utilizado un factor 10 para las simulaciones de ángulos de máquinas, obteniendo mejores resultados en cuanto al fitness y resultados. El factor es inversamente exponencial al fitness, por lo cual, a medida que este se aumente, el fitness irá decreciendo de manera exponencial.

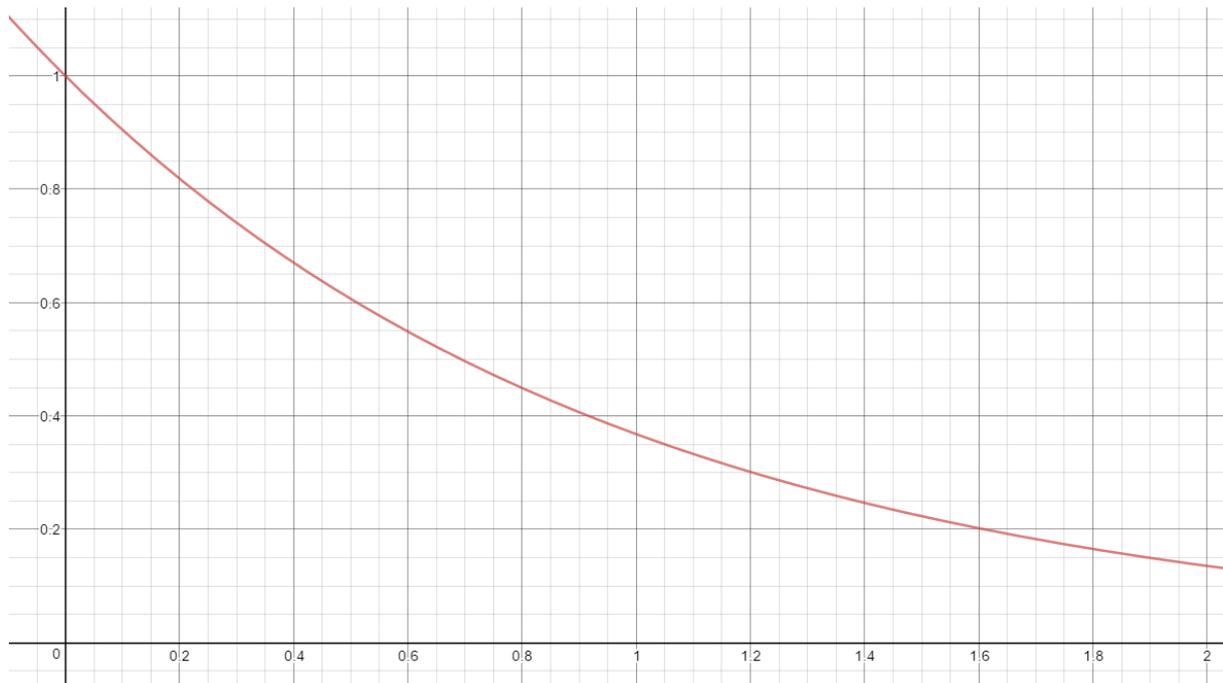


Figura 5.4: Gráfica de la función de fitness  $f(x) = e^{-x}$ .

### 5.4.3. Selección y reproducción de la población

La selección consiste en la elección de dos individuos de la población que se cruzarán, obteniendo otro individuo “hijo” de los anteriores. La selección de cada individuo se logra de manera probabilística, calculada mediante la función de fitness  $f(x)$  definida por el usuario.

La reproducción consiste en mezclar el material genético (los valores numéricos de cada vector) de los individuos seleccionados. Esto se realiza de diferentes formas. La más utilizada suele elegir un punto  $i$  del vector y tomar los valores  $1, \dots, i$  de un padre y los valores  $i + 1, \dots, l$  del otro padre, siendo  $l$  el largo del vector.

El número de individuos se mantiene constante a lo largo de las generaciones. Esto se realiza reemplazando individuos a medida que se va reproduciendo y mutando la población.

Una vez generada la población inicial, ésta se reproduce generación a generación, dando lugar a nuevas generaciones para las cuales se mezcla el material genético de sus padres. Como el número de individuos de la población debe mantenerse, los nuevos individuos creados irán reemplazando a la generación anterior. Tanto el reemplazo como la selección de individuos a reproducirse pueden realizarse de diferentes formas según la bibliografía especializada.

En lo que refiere al reemplazo de la población, se decidió sobreescibir el 90 % de la población con fitness más bajo. Este tipo de reemplazo se denomina de tipo elitista, ya que mantiene los individuos genéticamente superiores, y se recomienda reemplazar entre un 80 % y un 90 % de la población. Se llegó a concluir que esta era la mejor opción luego de perder en múltiples generaciones individuos con un fitness elevado, lo que derivó en una caída repentina del fitness de la población y un retroceso en el avance del algoritmo de más de 20 generaciones en algunos casos.

Respecto a la selección de los individuos a reproducirse, ésta se resolvió mediante un sorteo probabilístico, respecto del peso relativo del fitness de cada individuo de la población.

De esta manera, definimos la población de individuos como la matriz de vectores definido a partir de la población inicial:

$$\begin{bmatrix} (v_{11} & v_{12} & \dots & v_{1l}) \\ (v_{21} & v_{22} & \dots & v_{2l}) \\ (\vdots & \vdots & \ddots & ) \\ (v_{m1} & v_{m2} & \dots & v_{ml}) \end{bmatrix},$$

donde  $m$  representa el número de individuos que tendrá la población y  $l$  representa el número de variables que se buscará optimizar. Por lo tanto, cada vector corresponde a un individuo, formado por  $l$  valores generados aleatoriamente, aunque dentro de cierto margen manejado por el usuario.

A su vez, a cada individuo le va a corresponder una sucesión de valores  $b$  (curva en el tiempo), producto de la simulación del equivalente para los valores de las variables cada vector.

$$\begin{bmatrix} (v_{11} & v_{12} & \dots & v_{1l}) \\ (v_{21} & v_{22} & \dots & v_{2l}) \\ (\vdots & \vdots & \ddots & ) \\ (v_{m1} & v_{m2} & \dots & v_{m-l}) \end{bmatrix} \Rightarrow \begin{bmatrix} b_{11} & \dots & b_{1n} \\ b_{21} & \dots & b_{1n} \\ \vdots & & \\ b_{m1} & \dots & b_{mn} \end{bmatrix}.$$

Nuevamente, a este nuevo conjunto también le corresponde una matriz de valores de la función de fitness asociado a cada uno:

$$\begin{bmatrix} (v_{11} & v_{12} & \dots & v_{1l}) \\ (v_{21} & v_{22} & \dots & v_{2l}) \\ (\vdots & \vdots & \ddots & ) \\ (v_{m1} & v_{m2} & \dots & v_{m-l}) \end{bmatrix} \Rightarrow \begin{bmatrix} b_{11} & \dots & b_{1n} \\ b_{21} & \dots & b_{1n} \\ \vdots & & \\ b_{m1} & \dots & b_{mn} \end{bmatrix} \Rightarrow \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_m) \end{bmatrix}.$$

Y por último, a cada individuo se le asocia una probabilidad de reproducción, calculada respecto del fitness de cada uno:

$$\begin{bmatrix} (v_{11} & v_{12} & \dots & v_{1l}) \\ (v_{21} & v_{22} & \dots & v_{2l}) \\ (\vdots & \vdots & \ddots & ) \\ (v_{m1} & v_{m2} & \dots & v_{m-l}) \end{bmatrix} \Rightarrow \begin{bmatrix} b_{11} & \dots & b_{1n} \\ b_{21} & \dots & b_{1n} \\ \vdots & & \\ b_{m1} & \dots & b_{mn} \end{bmatrix} \Rightarrow \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_m) \end{bmatrix} \Rightarrow \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix},$$

donde la probabilidad de cada individuo se define como

$$p_j = \frac{f(x_j)}{\sum_{i=1}^m f(x_i)}.$$

Esta última probabilidad se utilizará en el sorteo de selección para reproducción. El sorteo de la población se realiza mediante una matriz de sorteo, la cual es generada a partir de la población actual y la probabilidad asociada a cada individuo.

De esta forma, el sorteo probabilístico consiste en armar una matriz  $1 \times k$ , donde  $k$  indica el largo de la matriz, y esta es completada con números que van de 1 a  $m$ , repetido cada uno  $k \times p_j$  veces. A partir de aquí se realiza un sorteo entre los  $k$  elementos de la matriz, y el número elegido corresponderá al número de individuo previamente definido en la matriz de la población. Así fue que se logró realizar un sorteo donde cada individuo tiene diferentes probabilidades de salir sorteado.

#### 5.4.4. Mutación de la población

La mutación de los individuos de la población consiste en la alteración aleatoria de algunos genes de ciertos individuos. Tanto los individuos como los genes a mutar de cada individuo son elegidos de manera randómica. Por lo tanto, al principio del programa se define la constante *mutacion\_chance*, que es la que define la probabilidad de mutar que tiene la población y la rutina *mutacion* se encarga de realizar el sorteo por el cual se van seleccionando los individuos que se mutarán. Para cada individuo seleccionado se sortea un gen (es decir, uno de los valores del vector) y se reemplaza por otro valor randómico, el cual deberá encontrarse entre los límites definidos al principio del programa, teniendo en cuenta la magnitud que se trate.

El sorteo de los individuos se realiza recorriendo la población uno a uno y realizando un sorteo para cada individuo. Para ello se realiza un random entre 0 y 1 para cada uno de los individuos y, si el resultado es menor que *mutacion\_chance*, se selecciona el individuo para mutación. Luego, se sortea el gen a mutar, lo cual se realiza nuevamente con un random entre los índices del vector, y dependiendo del índice sorteado, se reemplazará el gen con un nuevo gen generado a partir de un random entre las cotas definidas al principio del programa. Recordemos que los genes son variables diferentes unas de otras y no pueden mezclarse entre ellas, ya que cada una tiene diferentes cotas.

Una vez realizada la mutación, se reemplaza el individuo original por el individuo mutado y se devuelve a la población.

#### 5.4.5. Elección del mejor individuo/resultado

Los algoritmos genéticos funcionan evolucionando una población a través del tiempo a partir de una función de fitness, previamente definida. De esta manera, la población se irá optimizando de acuerdo al fitness de sus individuos, lo que mejorará tanto los valores de fitness individualmente como en su conjunto. Terminada nuestra simulación, se tendrá una población mejorada respecto a la población inicial.

El programa tiene en cuenta estos detalles y selecciona el mejor individuo de todas las poblaciones como resultado de la simulación. Este individuo coincidirá con el mejor individuo de la población final, ya que estamos evolucionando la población de manera elitista, siempre manteniendo los mejores individuos entre una generación y otra.

Al final de la simulación, se muestra una gráfica evolutiva en función de las generaciones, el fitness de los mejores individuos de cada generación. Esta gráfica nos da una idea del desempeño del programa, convergencia y de cuán cerca estamos del resultado óptimo que queremos obtener para nuestro equivalente.

### 5.5. Definición de subrutinas

#### Programa Principal

El programa principal inicializa las constantes del programa y luego ejecuta el programa invocando las rutinas anteriores.

A continuación se describe la secuencia:

- Se genera la población inicial.

- Se evoluciona la población el número de generaciones preestablecido.
- Se selecciona el mejor individuo de todas las generaciones.
- Se grafican la simulación del sistema original y la simulación del mejor individuo obtenido. Este punto se realiza únicamente para poder apreciar de manera gráfica el resultado obtenido.

### Inicialización de variables

- *time*: tiempo de simulación. Se utilizó 4 segundos para la mayoría de las simulaciones y se hicieron pruebas para 9 y 20 segundos.
- *num*: cantidad de individuos. Se utilizó una población de 500 individuos.
- *generaciones*: número de generaciones. Se eligieron 50 generaciones, tratando de guardar una relación de 10 a 1 entre la población y las generaciones.
- *largo*: longitud del material genético de cada individuo. Este punto depende de la cantidad de variables que se quieran modificar.
- *mutacion\_chance*: probabilidad de que un individuo mute. Se recomienda una probabilidad en entorno de 0.2.
- *XXXmax* y *XXXmin*: márgenes máximo y mínimo de cada constante que se generará para los individuos. Los márgenes recomendados se describen en el capítulo de modelado del equivalente.
- *fact\_fit*: factor de ajuste de fitness. Para algunas magnitudes, como el ángulo de las máquinas, se introdujo un factor de ajuste que mejorara el fitness ( $\alpha = 10$ ).
- *salida*: magnitud que se quiere optimizar. Se puede elegir entre frecuencia, tensión, ángulo, ángulo\_maq.
- *barra\_salida*: barra en que se medirá la magnitud anterior.
- *titulo*: título de la gráfica del resultado.
- *titulo\_ejey*: título de las ordenadas de la gráfica del resultado.
- *falla()*: rutina de definición de fallas. Se debe definir previamente la falla que se simulará.

## Rutinas

◦ *def sim(sim, T1d0\_1, T2d0\_1, T2q0\_1, H\_1, D\_1, Xd\_1, Xq\_1, X1d\_1, X2dq\_1, Xl\_1, S1\_1, S2\_1, T1d0\_2, T2d0\_2, T2q0\_2, H\_2, D\_2, Xd\_2, Xq\_2, X1d\_2, X2dq\_2, Xl\_2, S1\_2, S2\_2, r\_1, x\_1, b\_1, r\_2, x\_2, b\_2, r\_3, x\_3, b\_3, r\_4, x\_4, b\_4, r\_5, x\_5, b\_5, r\_6, x\_6, b\_6, r\_7, x\_7, b\_7, r\_8, x\_8, b\_8, r\_9, x\_9, b\_9, r\_10, x\_10, b\_10, tiempo)*:

La rutina simula la falta definida en *falla()* y devuelve el resultado de la magnitud y la barra definidas en la inicialización. Para ello se cargan los nuevos valores de las distintas variables (constantes de máquinas y líneas), el tiempo de simulación y una constante *sim*, la cual puede tomar tres valores según el sistema que se quiera simular: *orig* para el sistema original, *ny* para el equivalente NY sin optimizar y *eq* para el equivalente NY con las variables ingresadas. Tanto para *orig* como para *ny*, las variables ingresadas no son tenidas en cuenta, ya que ambos sistemas no se modifican para su simulación.

### Parámetros de entrada

- *sim*: sistema que se simulará (*orig*, *ny* o *eq*)
- variables: constantes de máquinas y líneas. Éstas pueden variar dependiendo de los tipos y la cantidad de máquinas y el número de líneas que se quieren optimizar.
- *tiempo*: tiempo de la simulación que se registrará.

### Parámetros de salida

- *chandata\_dict*: devuelve un vector con el resultado de la simulación para la magnitud elegida en *salida*. El largo de este vector dependerá del tiempo de la simulación y de los pasos entre dos puntos consecutivos.

◦ *individuo(T1d0min, T1d0max, T2d0min, T2d0max, T1q0min, T1q0max, T2q0min, T2q0max, Hmin, Hmax, Dmin, Dmax, Xdmin, Xdmax, Xqmin, Xqmax, X1dmin, X1dmax, X1qmin, X1qmax, X2dqmin, X2dqmax, Xlmin, Xlmax, S1min, S1max, S2min, S2max, rmin, rmax, xmin, xmax, bmin, bmax)*

La rutina genera un vector de largo igual a *largo* (definido en el programa principal) con números reales entre un mínimo y un máximo definidos para cada magnitud. Este vector corresponde a un individuo de la población y los valores del vector dependerán de la magnitud que se quiera modificar.

### Parámetros de entrada

- *XXXmin*: valores mínimos que pueden resultar de la aleatorización para cada variable.
- *XXXmax*: valores máximos que pueden resultar de la aleatorización para cada variable.

### Parámetros de salida

- *individuo*: vector de magnitudes de valores randómicos con largo preestablecido.

◦ *generar\_poblacion(T1d0min, T1d0max, T2d0min, T2d0max, T1q0min, T1q0max, T2q0min, T2q0max, Hmin, Hmax, Dmin, Dmax, Xdmin, Xdmax, Xqmin, Xqmax, X1dmin, X1dmax, X1qmin, X1qmax, X2dqmin, X2dqmax, Xlmin, Xlmax, S1min, S1max, S2min, S2max, rmin, rmax, xmin, xmax, bmin, bmax)*

La rutina genera la población inicial de tamaño preestablecido en el programa principal. Para ello invoca la subrutina individuo el número de veces correspondiente al tamaño de la población.

### Parámetros de entrada

- *XXXmin*: valores mínimos que pueden resultar de la aleatorización para cada variable.
- *XXXmax*: valores máximos que pueden resultar de la aleatorización para cada variable.

### Parámetros de salida

- *individuos*: la rutina devuelve la población inicial.

◦ *calcular\_fitness(individuo)*

Calcula el valor de fitness para cada individuo que se le ingrese. La función de fitness se puede modificar, con la salvedad de que deberá utilizarse una función que minimice la distancia entre la simulación entre el caso original y el caso correspondiente al individuo ingresado.

### Parámetros de entrada

- *individuo*: individuo para el cual se calculará el valor de fitness.

### Parámetros de salida

- fitness: devuelve el valor de fitness calculado para el individuo ingresado.

- *seleccion\_y\_reproduccion(poblacion)*

Esta rutina puntúa todos los elementos de la población (población) y se queda con los mejores, guardándolos dentro de 'selección'. Luego mezcla el material genético de los elegidos para crear nuevos individuos y llenar la población (guardando también una copia de los individuos seleccionados sin modificar).

### Parámetros de entrada

- población: población que se seleccionará y luego se reproducirá.

### Parámetros de salida

- población: población resultante de la selección y reproducción.

- *mutacion(poblacion, T1d0min, T1d0max, T2d0min, T2d0max, T1q0min, T1q0max, T2q0min, T2q0max, Hmin, Hmax, Dmin, Dmax, Xdmin, Xdmax, Xqmin, Xqmax, X1dmin, X1dmax, X1qmin, X1qmax, X2dqmin, X2dqmax, Xlmin, Xlmax, S1min, S1max, S2min, S2max, rmin, rmax, xmin, xmax, bmin, bmax)*

Rutina que muta los individuos al azar. La probabilidad de mutación de los individuos se fija en el programa principal.

### Parámetros de entrada

- población: población que se mutará.
- *XXXmin*: valores mínimos que pueden resultar de la aleatorización para cada variable.
- *XXXmax*: valores máximos que pueden resultar de la aleatorización para cada variable.

### Parámetros de salida

- población: población resultante luego de la mutación.

### Invocación de rutinas principales

Se invoca la rutina *generacion\_poblacion* y luego se realiza un for entre 1 y *generaciones*, en el cual se invocan las rutinas *seleccion\_y\_reproduccion* y *mutacion*.

Para cada generación se guardan el mejor vector (el de valor de fitness más alto) y su fitness.

Al final de las iteraciones se tienen los mejores individuos de cada generación, con sus respectivos valores de fitness.

### Gráficas

Finalizada la ejecución del programa principal se realizan dos gráficas. La primera muestra el resultado de la simulación para la magnitud y la barra elegidos, para los tres equivalentes: el original, el equivalente NY sin optimizar y el equivalente a optimizar. La segunda gráfica muestra la evolución del valor de fitness máximo de cada generación.

Se realizó un segundo programa auxiliar para graficar los resultados en distintos puntos de la red, para diferentes magnitudes. A continuación se listan:

<b>Magnitud</b>	<b>Nº de Barra</b>	<b>Nombre de Barra</b>
Frecuencia	4008	SA5
Frecuencia	90000	SU5
Frecuencia	90500	PA5
Frecuencia	90160	BR5
Frecuencia	90200	MB5
Frecuencia	90100	MA5
Frecuencia	90300	MI5
Frecuencia	90700	SC5
Frecuencia	90480	CM5
Tensión	4008	SA5
Tensión	90000	SU5
Tensión	90500	PA5
Tensión	90160	BR5
Tensión	90200	MB5
Tensión	90100	MA5
Tensión	90300	MI5
Tensión	90700	SC5
Tensión	90480	CM5
Ángulo de máquina	98511	PAL
Ángulo de máquina	98551	BAY
Ángulo de máquina	98541	BON
Ángulo de máquina	98171	PTI

Cuadro 1: Listado de gráficas de magnitudes realizadas para cada simulación.

Las gráficas muestran los resultados de las simulaciones de las fallas para frecuencia y tensión de los tres sistemas en los puntos más relevantes de la red de 500kV y los ángulos  $\delta$  de algunos generadores de gran potencia: represas de Palmar, Baygorria y Rincón del Bonete, y para la central térmica de ciclo combinado de Punta del Tigre.

En la figura 5.5 se detalla el unifilar de la red uruguaya obtenida del SCADA de UTE, donde se pueden apreciar las principales barras de la red de 500kV y 150kV.

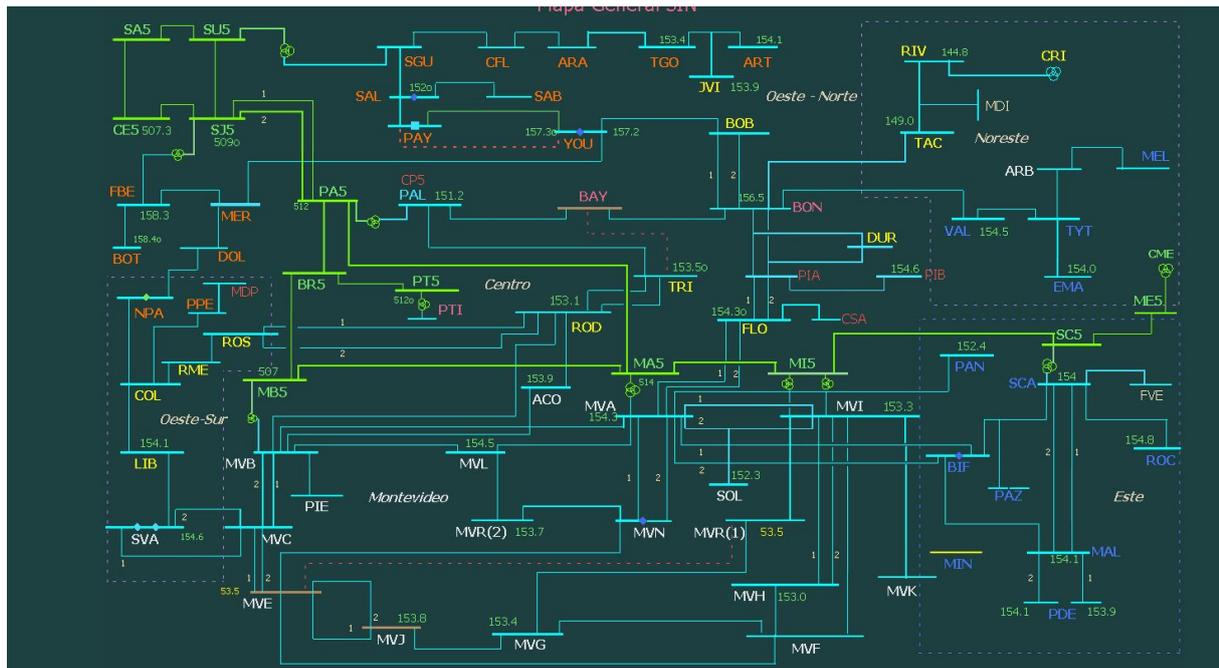


Figura 5.5: Unifilar de la red uruguaya obtenida del SCADA de UTE.

## 6. Pruebas y resultados

Se debió recorrer un largo camino para poder llegar a este punto, y poder finalmente poner a prueba el programa realizado para su evaluación final. En el siguiente capítulo se exponen los principales desafíos que fueron sorteados para luego pasar a la etapa de pruebas, y luego a la etapa de simulaciones finales. Finalmente se analizan los resultados, midiendo el alcance del programa, sus aplicaciones y sus posibles desarrollos y mejoras a futuro.

### 6.1. Obstáculos y primeras pruebas

En esta sección se exponen los principales problemas y obstáculos que se presentaron al realizar las simulaciones y de que manera fueron superados, de forma tal que el futuro usuario del programa desarrollado, o futuro usuario de programas desarrollados en python para PSS®E, tenga en cuenta estas dificultades y pueda sortearlas de la mejor manera.

Los siguientes fueron los problemas más significativos que se presentaron:

- **Instalación:** se sugiere utilizar los programas recomendados, con sus respectivas versiones. Esto se debe a que no todas las versiones son plenamente compatibles entre ellas, en especial python y PSS®E. Se recomienda utilizar la versión python 2.5.
- **Variaciones de alta frecuencia:** se recomienda simular los sistemas por lo menos un segundo antes de realizar la simulación dinámica para que se establezcan las diferentes variables. Si esto no se hace puede suceder variaciones de muy alta frecuencia que afectan los resultados y el desempeño del programa.
- **Canales de salida:** este es un punto de suma importancia cuando se tiene un programa que ejecuta una gran cantidad de simulaciones. Los diferentes comandos de salida de canales para simulaciones dinámicas tienen un vector de entrada llamado “status”, el cual tiene entre dos y cuatro valores, dependiendo de la magnitud. El primer valor corresponde al “starting channel index” y este debe ser establecido como un valor constante (y no dejar el valor por defecto “-1”). Esto es importante ya que si se establece en “-1”, el programa agrega un canal en cada simulación, demorando cada vez más la ejecución del programa, al punto de demorar semanas para terminar la simulación completa de 50 generaciones y 500 individuos. Los siguientes valores del “status” pueden ser el “starting VAR index” y el “starting ICON index”. Estos sí deben ser establecidos en “-1”.
- **Márgenes de las variables:** los márgenes fueron calibrados teniendo en cuenta las restricciones especificadas en la documentación del PSS®E y la evaluación empírica de los sistemas. Debe prestarse atención si se modifican estos márgenes ya que el programa puede dejar de funcionar, provocando errores en tiempo de ejecución en Python.
- **Búsqueda y calibración de función de fitness:** otro de los puntos clave fue la elección y calibración de la función de fitness. Esta debe cumplir los requisitos explicados anteriormente para luego ser evaluada por el programa, de forma que no converja al valor óptimo ni lo suficientemente rápido para que la población pierda su diversidad, ni lo suficientemente lento

para que la población no evolucione. Asimismo, algunos autores recomiendan la utilización de la diferencia de las integrales de ambas curvas (sus áreas) para el cálculo, mientras que en este trabajo se utilizó la diferencia punto a punto entre las curvas, obteniendo una gran diferencia en los resultados a favor de esta última. Los resultados obtenidos para la diferencia de las áreas no fueron auspiciosos.

- Simulación y compilación: los problemas de simulación y compilación son inherentes a los desarrollos de programas. En el presente desarrollo la gran dificultad se encontró en el lenguaje específico del módulo psspy para python. Este lenguaje cuenta con comandos específicos para el manejo del PSS®E desde python. Se recomienda la utilización tanto de PSS®E Application Program Interface (API), la cual se encuentra en la documentación en la carpeta de instalación del programa, y de la ayuda online de páginas especializadas, como lo es <https://psspy.org/>.

Se realizaron diversas pruebas para calibrar el programa, comenzando con sistemas pequeños, sin aplicar fallas dinámicas, con el fin de buscar la mejor forma de unir el algoritmo genético a la simulación en PSS®E y de familiarizarse con el entorno.

Luego de obtenida la base del programa, se procedió a realizar las primeras simulaciones dinámicas y graficar los resultados, por separado del programa principal.

Llegado a este punto se procedió a unificar los programas, es decir, empezar a evaluar el fitness a partir de la diferencia entre la curva resultante de la simulación del sistema completo contra la curva resultante de cada simulación del equivalente NY.

Una vez unificado el programa, debieron determinarse dos puntos: la elección de la función de fitness, para la cual se determinaron sus principales condiciones de forma teórica para luego buscar funciones que satisfagan esas condiciones y evaluar sus resultados empíricamente, y la determinación del elitismo del algoritmo, lo que aseguró que los mejores resultados siempre se mantengan, y no tener retrocesos en la evolución de la población. En principio el programa únicamente optimizaba la inercia (H), el damping (D) y la reactancia de eje directo de las máquinas ( $X_d$ ).

En la figura 6.1 se pueden ver los primeros resultados llegado a este punto.

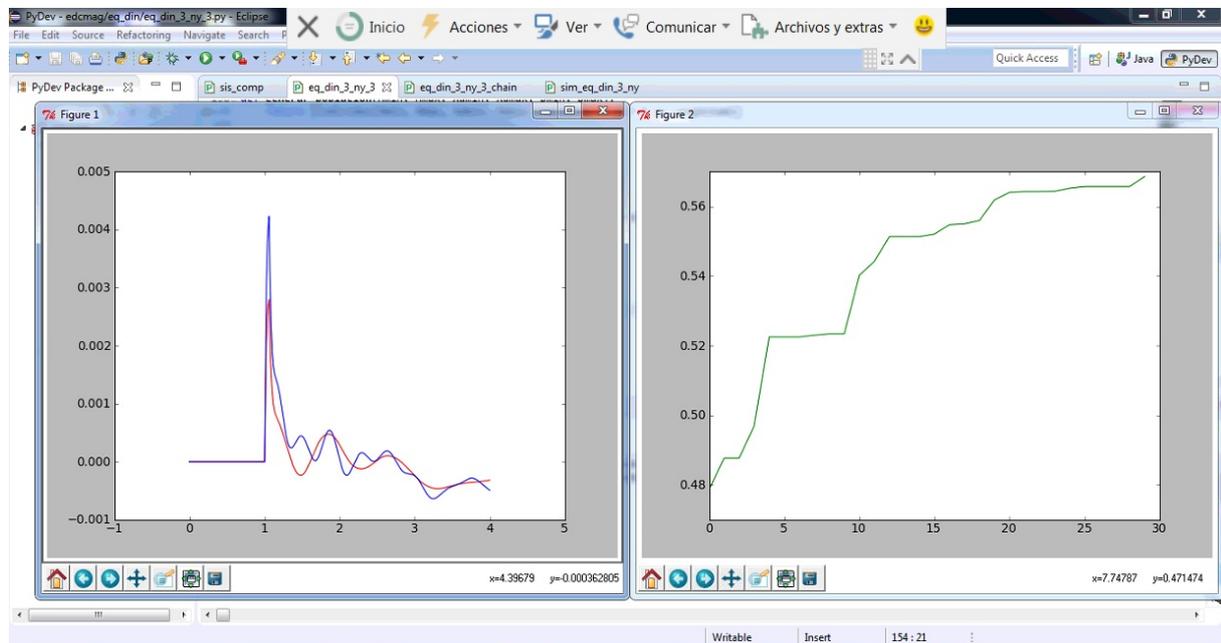


Figura 6.1: Gráfico a la izquierda: en rojo la simulación en sistema original y en azul el equivalente con primera optimización de  $H$ ,  $X_d$  y  $D$ . Gráfico a la derecha: fitness del mejor individuo en función de las generaciones.

Al resultado anterior se le agregó la evaluación del sistema equivalente sin optimizar, para poder visualizar el avance que se obtiene, el cual se puede ver en la figura 6.2 en color verde.

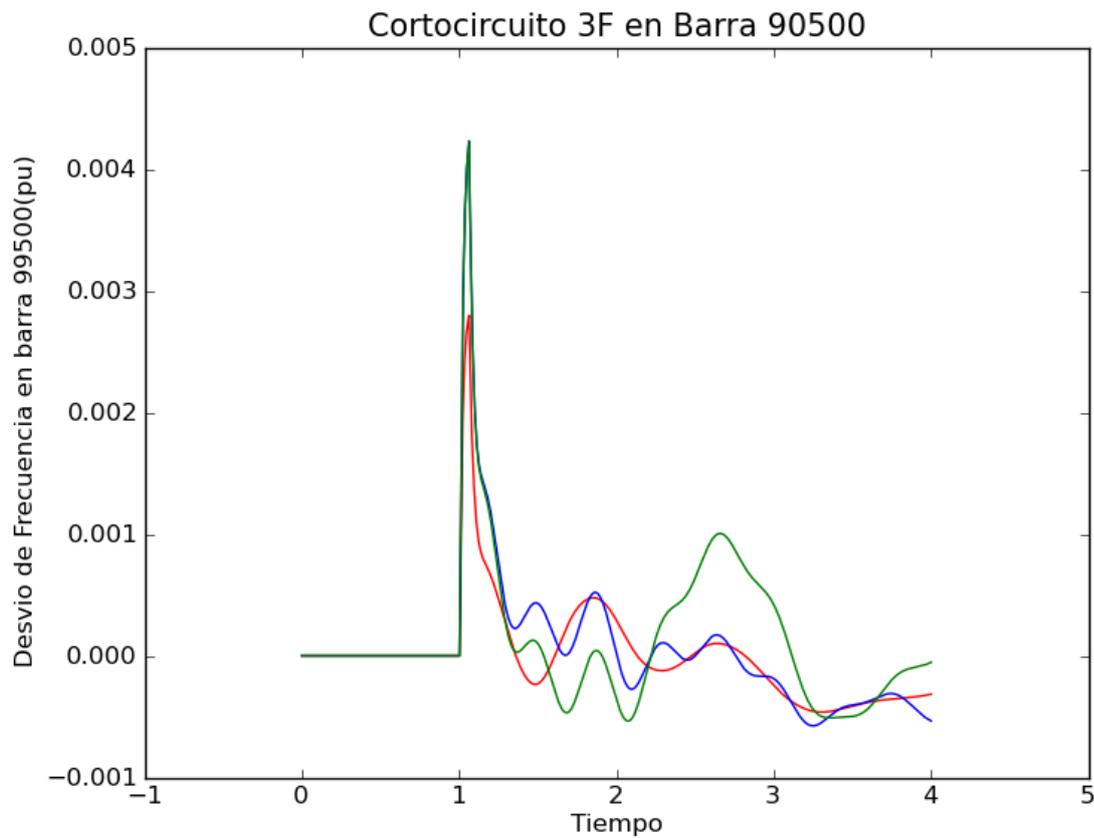


Figura 6.2: Rojo: simulación en sistema original. Verde: Equivalente sin optimizar. Azul: Equivalente con primera optimización de  $H$ ,  $X_d$  y  $D$  y un valor de fitness de 0,5851.

Con el programa en funcionamiento, y habiendo sorteado otros obstáculos como fueron la correcta determinación de los canales de salida, se fueron sumando otras variables a optimizar por el algoritmo.

En primera instancia, se optimizaron las demás variables de las máquinas, sin obtener demasiadas mejoras en el resultado arrojado por el equivalente. Sin embargo, se obtuvo el primer resultado prometedor cuando se agregaron las variables de las líneas de transmisión al programa ( $R$ ,  $X$  y  $B$ ), para el cual se obtuvo el resultado mostrado en la figura 6.3.

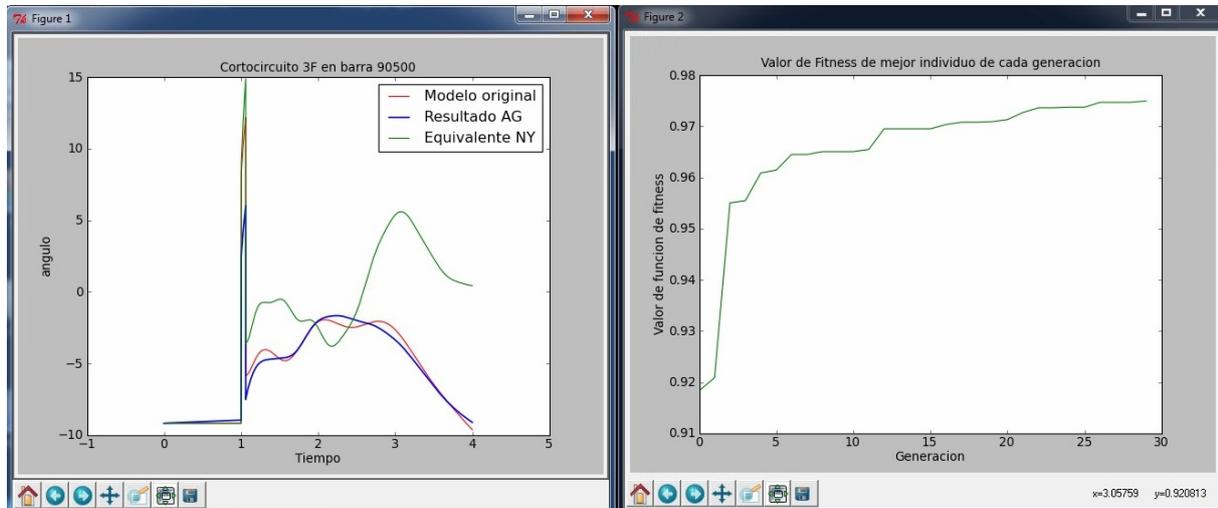


Figura 6.3: Primer resultado después de variar líneas y máquinas

Luego de obtener el resultado anterior se pudo corroborar que los resultados obtenidos podían ser aplicables a la optimización de un equivalente dinámico, con un error aceptable, y a partir de este punto se comenzaron a simular las fallas elegidas, para un mayor número de individuos y generaciones.

### 6.1.1. Convergencia del AG

Es interesante apreciar la convergencia del algoritmo mediante la múltiple ejecución de una misma falla, optimizando en el mismo lugar. Es de suponer que si el programa funciona correctamente, la repetición de la simulación debería arrojar resultados similares. En nuestro caso se debieron repetir algunas pruebas para asegurar los resultados, poniendo a prueba el funcionamiento del algoritmo. En el siguiente ejemplo se puede apreciar la cercanía entre dos resultados obtenidos para una misma simulación. En la figura 6.4 se observan los resultados para la misma falla de dos optimizaciones distintas.

*Cortocircuito 3F en barra PA5*  
*Optimización de ángulo de máquina en PTI*

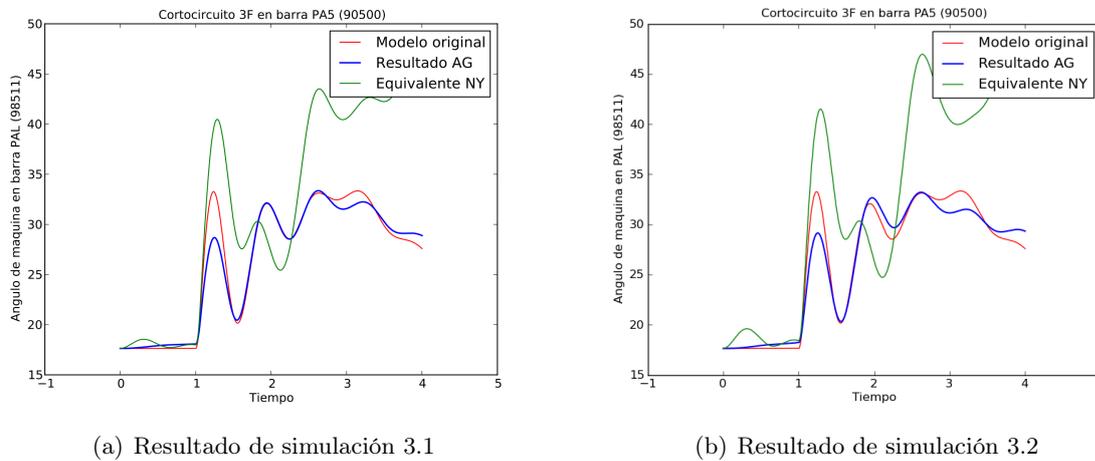


Figura 6.4: Resultados del algoritmo para una misma simulación.

De manera gráfica se pueden notar apenas diferencias en el segundo pico de la curva, mientras que desde el punto de vista numérico se obtuvieron resultados muy similares en algunos de los valores, como en el damping de ambas máquinas, el cual varía en un margen amplio de 0 a 40 pu, pero arrojó valores muy similares en ambos casos. Los valores se pueden apreciar en la figura 6.5.

Ezeiza (3000)												
Nº Simulación	T <sup>d</sup> 0 [seg]	T <sup>i</sup> d0 [seg]	T <sup>i</sup> q0 [seg]	H [pu]	D [pu]	X <sup>d</sup> [pu]	X <sup>q</sup> [pu]	X <sup>d</sup> [pu]	X <sup>i</sup> dq [pu]	X <sup>i</sup> [pu]	S1	S2
3.1	2,34890321	0,06481571	0,05220749	3,08675961	30,97483836	4,31587562	0,83515015	0,32160845	0,00947844	0,00572101	0,04902164	0,82713742
3.2	8,35398912	0,03576401	0,16292574	4,79741561	33,28722968	6,22499494	0,96629176	0,34539966	0,00982918	0,00593134	0,03418620	0,86401254

Almafuerte (106000)												
Nº Simulación	T <sup>d</sup> 0 [seg]	T <sup>i</sup> d0 [seg]	T <sup>i</sup> q0 [seg]	H [pu]	D [pu]	X <sup>d</sup> [pu]	X <sup>q</sup> [pu]	X <sup>d</sup> [pu]	X <sup>i</sup> dq [pu]	X <sup>i</sup> [pu]	S1	S2
3.1	5,71816357	0,03202430	0,18371684	5,65345476	35,18387949	7,36782619	0,79799895	0,27634655	0,00963027	0,00378619	0,03496670	0,69270077
3.2	3,50196691	0,04799237	0,13269133	15,51015696	37,25016582	2,15965773	0,85025236	0,29982567	0,00913729	0,00297511	0,03745254	0,79118434

Nº Simulación	Línea 4000-106000			Línea 4010-4000			Línea 5002-4010			Línea 4004-5002		
	R [pu]	X [pu]	B [pu]	R [pu]	X [pu]	B [pu]	R [pu]	X [pu]	B [pu]	R [pu]	X [pu]	B [pu]
3.1	0,00065675	0,00896536	0,06585081	0,00223383	0,00136883	0,32062156	0,00668051	0,00756379	0,08302496	0,00966031	0,00939857	0,82627991
3.2	0,00046299	0,00754053	0,57664259	0,00257842	0,00068179	0,23619764	0,00821652	0,00569533	0,90394647	0,00747289	0,00914721	0,83932825

Nº Simulación	Línea 3004-4004			Línea 3000-3004			Línea 3004-2018		
	R [pu]	X [pu]	B [pu]	R [pu]	X [pu]	B [pu]	R [pu]	X [pu]	B [pu]
3.1	0,00194567	0,00196719	0,31063698	0,00895893	0,00591583	0,48533004	0,00795234	0,00293546	0,81709701
3.2	0,00479888	0,00019968	0,06839362	0,00776678	0,00845873	0,00348325	0,00801865	0,00951519	0,91695686

Nº Simulación	Línea 4000-2018			Línea 4010-4008			Línea 5002-4008		
	R [pu]	X [pu]	B [pu]	R [pu]	X [pu]	B [pu]	R [pu]	X [pu]	B [pu]
3.1	0,00802138	0,00094482	0,02421729	0,00352528	0,00418101	0,65109566	0,00012241	0,00908909	0,98268801
3.2	0,00725758	0,00575045	0,12713861	0,00109412	0,00024044	0,60442155	0,00194711	0,00921464	0,11425972

Figura 6.5: Valores numéricos de la simulación 3.1 y 3.2.

## 6.2. Simulaciones

Se eligieron 3 fallas para probar en los equivalentes Reducido y NY. A su vez, se varían los puntos donde se aplica la falla y los puntos donde se optimiza el equivalente.

Se simularon en primera instancia 4 segundos de cada falla para poder trabajar con un gran número de individuos y generaciones, y así poder realizar el mayor número de simulaciones posibles. Sobre el final de este capítulo se analizará una falla de un tiempo mayor (9 y 20 segundos). En la figura 6.6 se resumen las pruebas realizadas:

Nº Simulación	Falla	Lugar de falla	Equivalente	Lugar de medida	Magnitud a optimizar
1	<b>CORTOCIRCUITO TRIFASICO</b>	PA5 con salida de PA5-MA5	NY	MI5	frecuencia
2				PA5	frecuencia
3.1				maq_PAL	ángulo
3.2				maq_PAL	ángulo
4.1				maq_PTI	ángulo
4.2				maq_PTI	ángulo
5		MA5 con salida de MA5-MI5	NY	MA5	tensión
6				maq_PTI	ángulo
7	<b>1FT CON REENGANCHE</b>	PA5 con salida de PA5-MA5	NY	PA5	frecuencia
8				maq_PAL	ángulo
9.1				maq_PTI	ángulo
9.2				maq_PTI	ángulo
10				MI5	frecuencia
11					PA5 con salida de PA5-MA5
12	maq_PAL	ángulo			
13	maq_PTI	ángulo			
14	MI5	frecuencia			

Figura 6.6: Tabla de simulaciones realizadas.

Los números correspondientes a cada barra se detallan en la figura 6.7

Nombre de	Nº
MI5	90300
PA5	90500
maq_PAL	98511
maq_PTI	98171
MA5	90100
PTI	90170

Figura 6.7: Número de cada barra.

### 6.2.1. Equivalente Reducido

El equivalente reducido es un sistema eléctrico muy básico que se probó para evaluar el límite del algoritmo y la simplificación del sistema argentino. Por otra parte, este equivalente se optimizó únicamente para la falta monofásica, la cual es la menos “agresiva” de las dos, y los resultados no fueron buenos, como era de esperarse. Es por ello que simplemente se mostrarán los resultados gráficos de estas simulaciones, sin ahondar en el análisis de este sistema.

*Cortocircuito 1FT con reenganche en barra PA5  
Optimización de frecuencia en PA5*

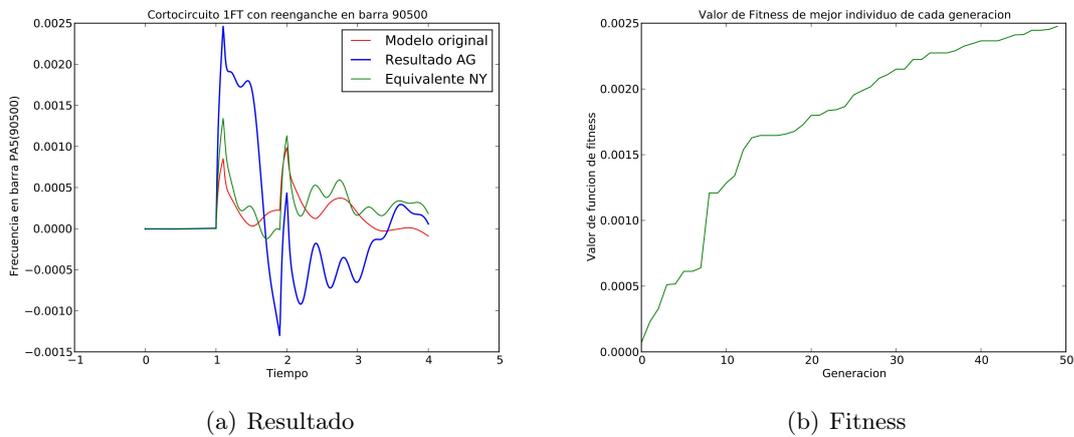


Figura 6.8: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

*Cortocircuito 1FT con reenganche en barra PA5  
Optimización de frecuencia en MI5*

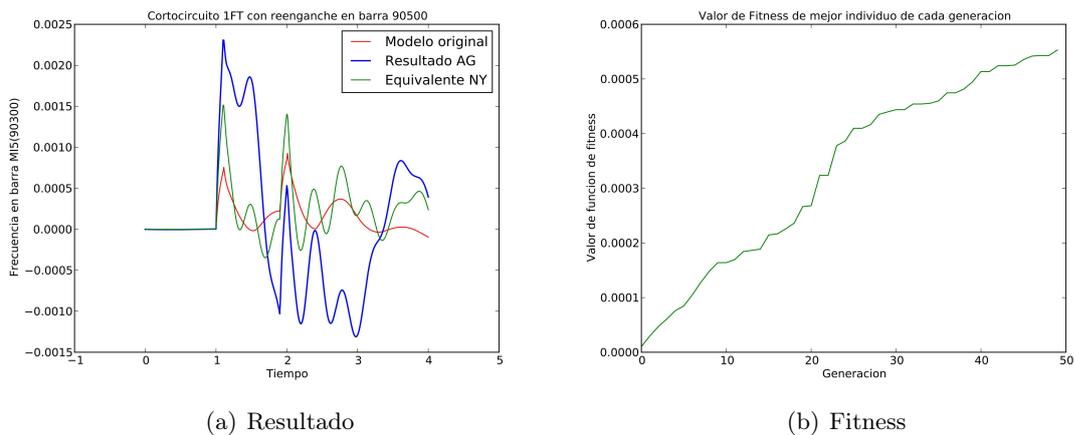


Figura 6.9: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

### 6.2.2. Equivalente NY

El equivalente NY es el resultado de una simplificación de la red argentina, utilizado para estudios de potencia. En este caso, se intenta optimizar este equivalente por medio del algoritmo. Se probaron dos fallas, optimizando diferentes magnitudes en distintos puntos, y se fueron variando algunas de las máquinas a optimizar. Este equivalente cuenta con una parte real del sistema y tres equivalentes: Almafuerite, Ezeiza y Yacyretá. Los dos primeros fueron optimizados en todas las simulaciones, pero se decidió alternar el equivalente de Yacyretá, dejándolo con sus valores originales y optimizar únicamente las máquinas de Almafuerite y Ezeiza. Por lo tanto, se probaron tres alternativas de optimización del mismo equivalente NY:

- Equivalente NY 1: se optimizaron las máquinas de Yacyretá (2 máquinas GENSAL), Almafuerite (GENSAL) y Ezeiza (GENSAL) y las líneas 4000-106000, 4010-4000, 5002-4010, 4004-5002, 3004-4004, 3000-3004, 3004-2018, 4000-2018, 4010-4008, 5002-4008.
- Equivalente NY 2: se optimizaron las máquinas de Yacyretá (GENSAL), Almafuerite (GENSAL) y Ezeiza (GENSAL) y las líneas 4000-106000, 4010-4000, 5002-4010, 4004-5002, 3004-4004, 3000-3004, 3004-2018, 4000-2018, 4010-4008, 5002-4008.
- Equivalente NY 3: se optimizaron las máquinas de Almafuerite (GENSAL) y Ezeiza (GENSAL) y las líneas 4000-106000, 4010-4000, 5002-4010, 4004-5002, 3004-4004, 3000-3004, 3004-2018, 4000-2018, 4010-4008, 5002-4008.

El primer equivalente fue optimizado a partir del equivalente NY original, el cual cuenta con dos máquinas GENSAL similares en Yacyretá. Este primer equivalente dio muchos problemas de estabilidad para los valores usuales de las constantes de las máquinas, provocando los errores en tiempos de ejecución de python que llevaron a la detención del programa en la mayoría de las simulaciones, sin mayor éxito. Sin embargo se pudo correr una falta monofásica, obteniendo buenos resultados.

El problema de estabilidad anterior se solucionó, al menos en parte, ya que se logró ejecutar el programa para todas las fallas, eliminando una de las máquinas de Yacyretá. Este equivalente logró ejecutarse sin problema, pero arrojando condiciones iniciales sospechosas (este es el término utilizado por PSS®E) en la gran mayoría de las simulaciones.

Por lo tanto, se decidió usar la máquina de Yacyretá con sus valores para el equivalente NY original y se procedió a variar únicamente las máquinas de Almafuerite y Ezeiza. Este último equivalente se comportó de buena forma ante las diversas faltas, obteniendo los mejores resultados tanto para los equivalentes como para los comportamientos sospechosos del programa. En los siguientes subcapítulos expondremos los resultados obtenidos.

#### Equivalente NY 1

En principio, se contaban con dos máquinas GENSAL en la central de Yacyretá, las cuales se intentaron optimizar con el programa. Se hizo el intento de correr varias faltas en diferentes puntos y el programa no lograba correr (Python se detenía). Finalmente se pudo correr el programa, pero

únicamente para una sola falta. Debido al poco margen de trabajo que nos daba este equivalente, se decidió modificarlo para corregir los problemas de estabilidad.

## Equivalente NY 2

El equivalente NY 2 es similar al anterior pero se decidió eliminar el segundo generador en Yacyretá. De esta forma, se pudieron simular las distintas faltas, eliminando los problemas de estabilidad del programa. Sin embargo, este nuevo equivalente presentaba problemas en las condiciones iniciales de simulación (simulación estática). Estos problemas se pueden apreciar en las figuras 6.10 y 6.11, corresponden a dos resultados diferentes del AG para la misma simulación.

### *Cortocircuito 3F en barra PA5 Optimización de ángulo en máquina de PAL*

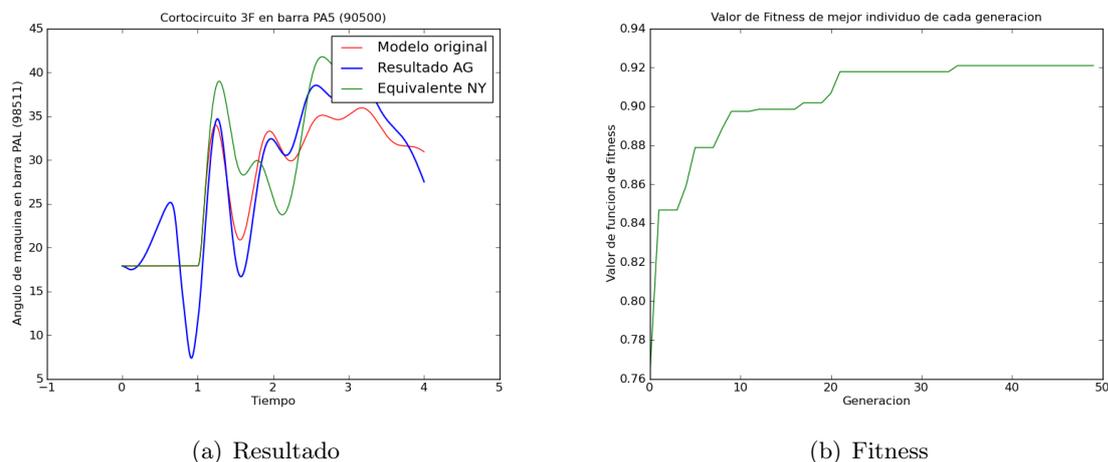


Figura 6.10: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

Esta simulación se repitió para descartar la posibilidad de que la primera haya sido simplemente algún valor de condición inicial particular para alguna de las máquina, pero ejecutando distintas faltas en otros puntos de la red se repitieron las condiciones iniciales sospechosas.

Se puede apreciar también que el fitness este sistema llegó a su máximo aproximadamente en la generación 20, lo que nos puede dar una idea del mal comportamiento del sistema.

*Cortocircuito 3F en barra PA5*  
*Optimización de ángulo en máquina de PAL*

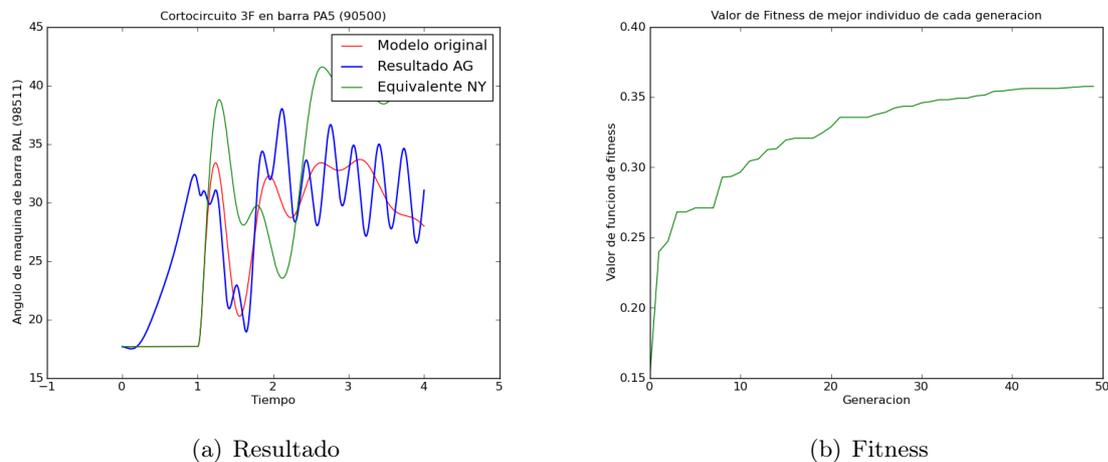


Figura 6.11: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

### Equivalente NY 3

En principio se decidió dejar fijas las constantes de la máquina de Yacyretá para observar si se obtenían mejoras en cuanto a las condiciones iniciales.

Este equivalente fue el que se comportó de mejor manera ante las diferentes fallas, es por ello que se decidió seguir adelante con este equivalente.

### Optimización de frecuencia en fallas trifásicas

En primer lugar se intentó optimizar la frecuencia para fallas trifásicas. Estas fallas suelen ser las más agresivas. Para la primera falla se obtuvo una buena aproximación del pico inicial y un buen desempeño en las oscilaciones posteriores. Esto se muestra en la figura 6.12.

*Simulación 1 - Cortocircuito 3F en barra PA5  
Optimización de frecuencia en MI5*

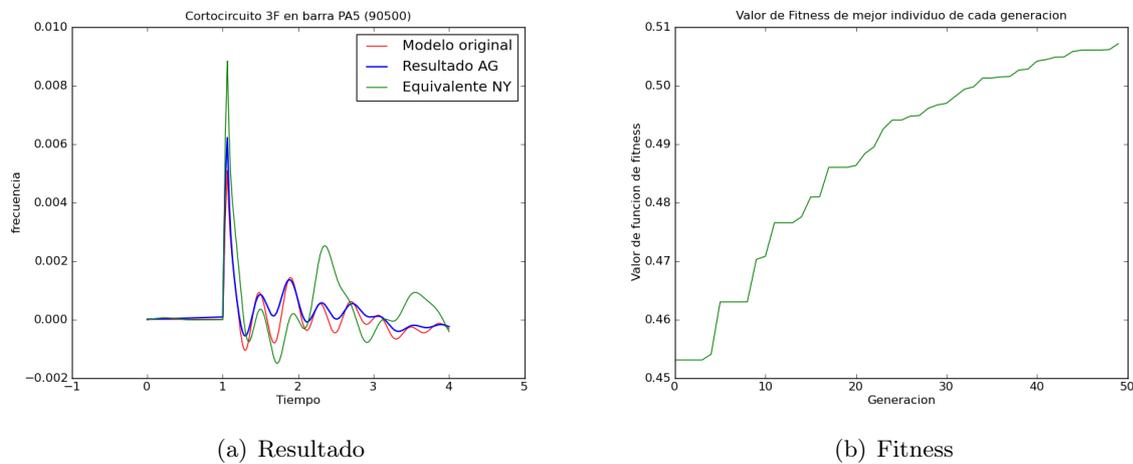


Figura 6.12: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

En la segunda optimización de frecuencia se aproximó el pico inicial pero no se lograron reproducir de buena forma las oscilaciones posteriores como es mostrado en la figura 6.13. Sin embargo, la curva logra una buena aproximación poco después de los 2 segundos de simulación. Es importante notar que la mayor diferencia entre una curva y otra se da alrededor de 1,7 segundos y corresponde a un desvío menor a 0,001 pu en frecuencia, lo que equivale a una diferencia menor a 0,5 Hz.

*Simulación 2 - Cortocircuito 3F en barra PA5  
Optimización de frecuencia en PA5*

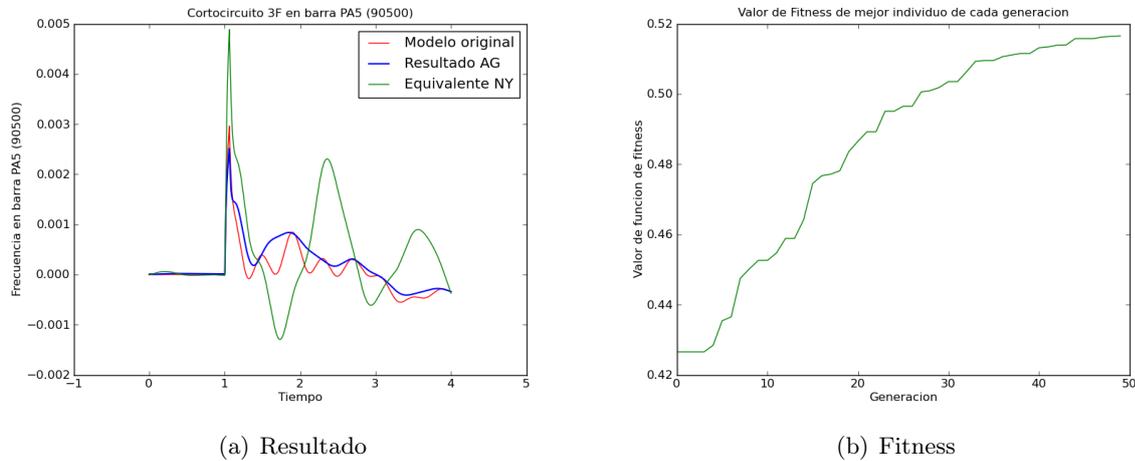


Figura 6.13: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

### Optimización de frecuencia en fallas monofásicas

Los resultados para fallas monofásicas también fueron bastante prometedores y se tuvieron buenos ajustes para los picos de frecuencia.

Para la falla en PA5 optimizando en PA5, cuya simulación es mostrada en la figura 6.14, se obtuvo una respuesta similar que en la falla trifásica, buena aproximación en los picos (en el instante de falta y en la reconexión) pero las oscilaciones posteriores a la falla no se aproximaron como era deseable. Sin embargo, luego de la reconexión, las oscilaciones se asemejan de mejor forma al sistema original.

*Simulación 7 - Cortocircuito 1FT con reenganche en barra PA5  
Optimización de frecuencia en PA5*

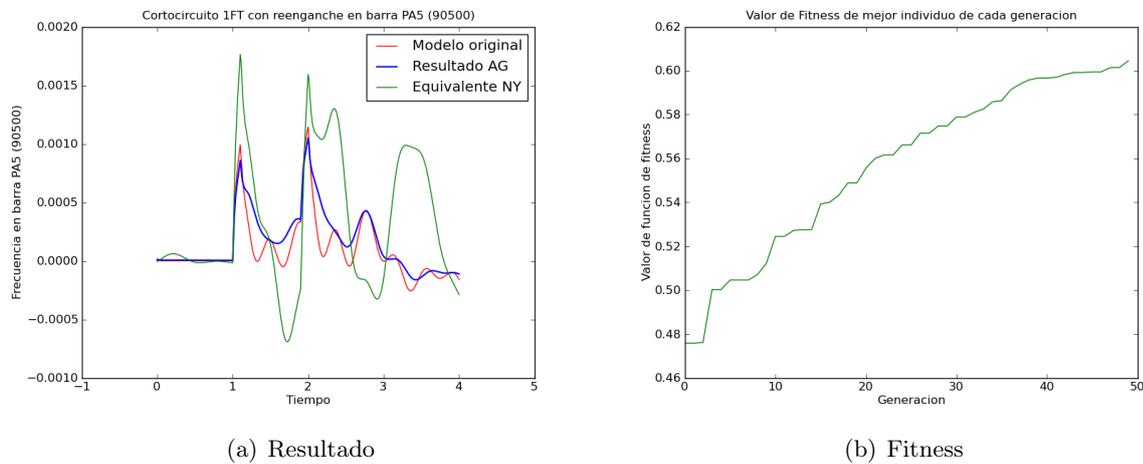


Figura 6.14: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

Analizando la falla monofásica con reconexión en barra de PA5 optimizando en MI5, mostrada en la figura 6.15, se obtiene un buen ajuste en los picos nuevamente. En las oscilaciones posteriores a la falla no se obtiene una mejor aproximación que cuando se optimiza en PA5. Sin embargo, en las oscilaciones posteriores a la reconexión, la oscilación se asemeja rápidamente al sistema original y, si bien las oscilaciones no se ajustan en amplitud (parece que tengan un offset) si se asemejan en forma, es decir oscilan con igual frecuencia pero con un valor medio un poco mayor. De esta manera se obtiene una mejor aproximación en este aspecto.

*Simulación 10 - Cortocircuito 1FT con reenganche en barra PA5  
Optimización de frecuencia en MI5*

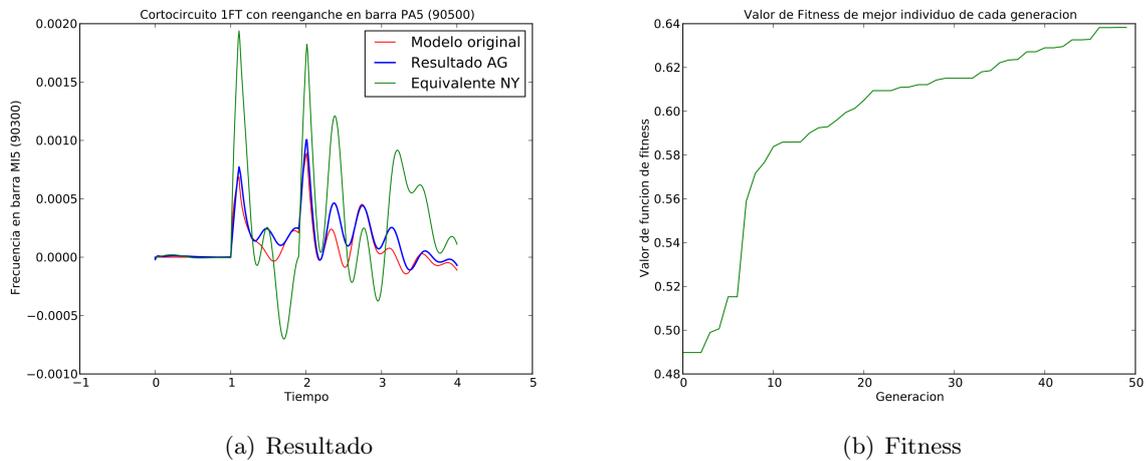


Figura 6.15: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

### Optimización de ángulo de máquina en fallas trifásicas

La optimización de los ángulos también fue exitosa, observando la mayor diferencia en el último segundo graficado. La convergencia de ambas se puede corregir aumentando el tiempo de simulación de la falta a optimizar. Esto está definido en las constantes de inicialización del programa, y se eligió únicamente un tiempo de 4 segundos para poder realizar la mayor cantidad de simulaciones posibles (ya que cada simulación lleva alrededor de 40 horas). También debe tenerse en cuenta que los ángulos fueron puestos en fase tanto para la optimización como para graficar los resultados, por lo que existen diferencias constantes entre unos ángulos y otros. Esto se realizó con la idea de tener mejor apreciación gráfica de los resultados obtenidos.

*Simulación 3.2 - Cortocircuito 3F en barra PA5*  
*Optimización de ángulo de máquina en PAL*

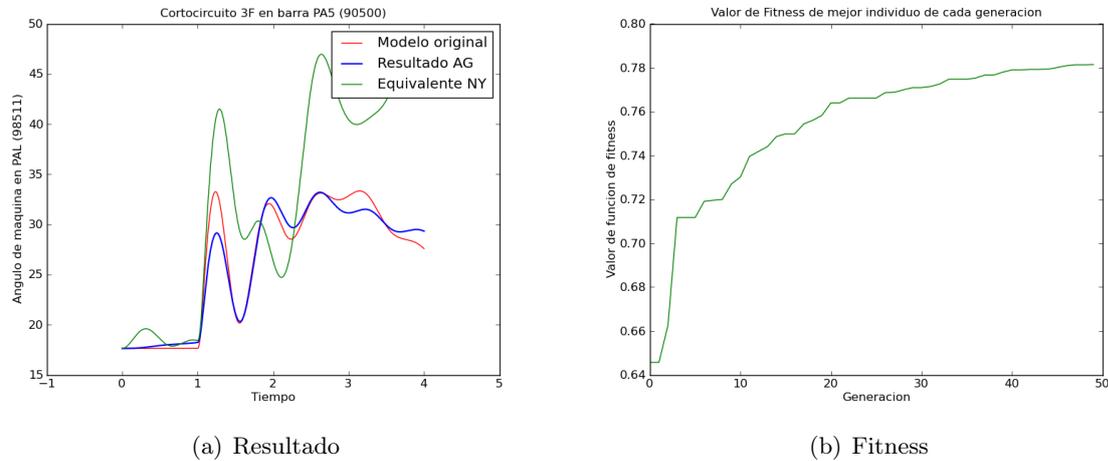


Figura 6.16: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

*Simulación 4.2 - Cortocircuito 3F en barra PA5*  
*Optimización de ángulo de máquina en PTI*

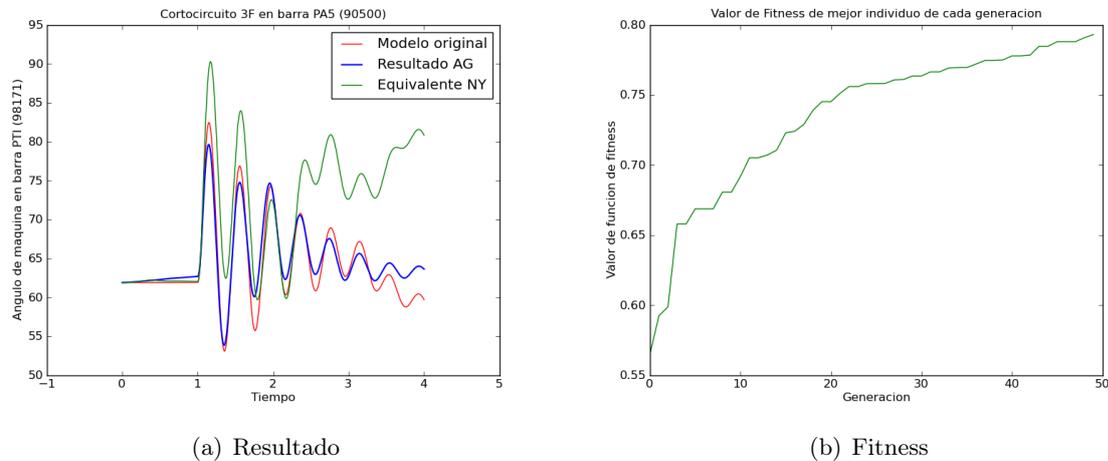


Figura 6.17: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

La optimización del ángulo de máquina en PTI, con un cortocircuito 3F en barra MA5 es muy similar a la misma optimización en la barra PA5. Una diferencia es que la que se está analizando ahora presenta un resultado más distante en el primer pico de la falla. Luego de este pico se aproxima de mejor manera que la anterior y presenta una diferencia mayor en el ultimo segundo, debido a la misma explicación que con la falla en PA5. Además se puede observar que el valor de fitness da un valor muy cerca de la unidad de aproximadamente 0,8, el cual es el casi idéntico

que el valor que se tenía con la optimización del ángulo de máquina en PTI, con un cortocircuito 3F en barra MA5. Esto demuestra de que no varían los valores angulares de máquina PTI si la falla es en PA5 o MA5.

*Simulación 6 - Cortocircuito 3F en barra MA5  
Optimización de ángulo de máquina en PTI*

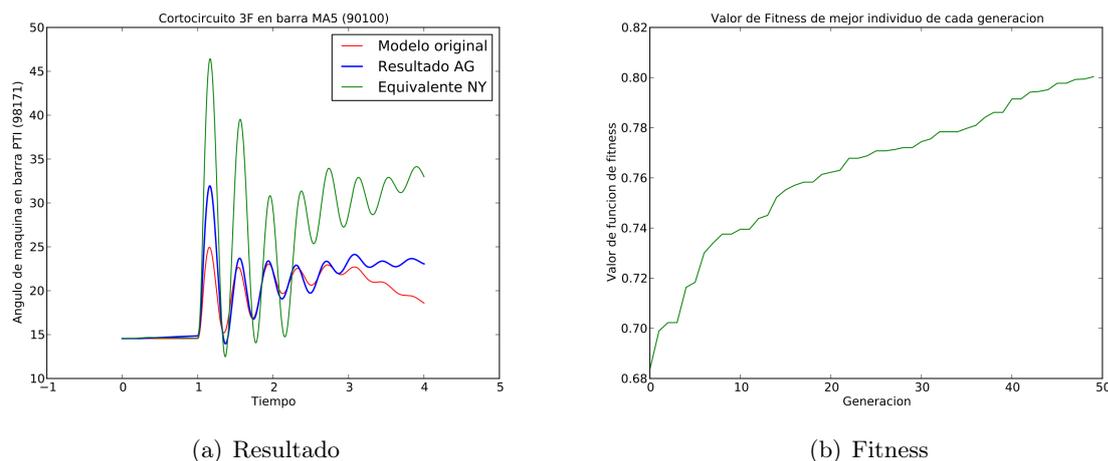


Figura 6.18: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

### Optimización de ángulo de máquina en fallas monofásicas

Al igual que en la optimización para fallas trifásicas, la mayor diferencia entre curvas es en el último segundo de simulación. Se ven diferencias en los picos de oscilación pero esto seguramente se debe a haber puesto los ángulos en fase en el momento inicial. Se observa que el comportamiento oscilatorio es prácticamente de igual frecuencia lo que da un ajuste transitorio muy bueno. En las figuras 6.19 y 6.20 se muestran los ajustes por ángulo para fallas en PA5 optimizando ángulo en PAL y PTI respectivamente. Se aprecia que si bien el comportamiento oscilatorio de las fallas es distinto para los lugares donde se optimiza el equivalente, la simulación se asemeja muy bien en ambos casos al sistema original, logrando valores de fitness que superan el valor 0,88 y el 0,94 cada una.

En un subcapítulo posterior se mostrarán los resultados para una optimización de 9 segundos, lo cual es un tiempo para el cual se observa la estabilización luego de las fallas.

*Simulación 8 - Cortocircuito 1FT con reenganche en barra PA5  
Optimización de ángulo de máquina en PAL*

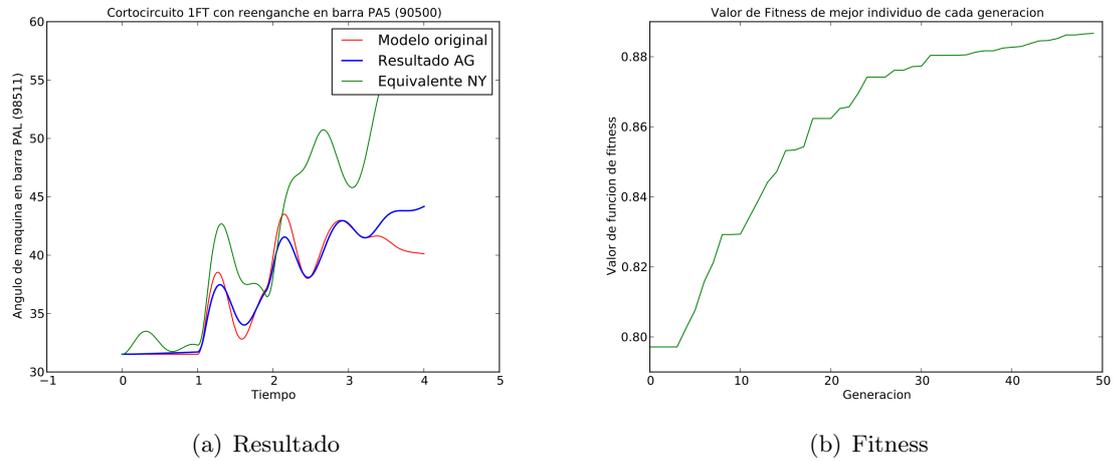


Figura 6.19: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

*Simulación 9 - Cortocircuito 1FT con reenganche en barra PA5  
Optimización de ángulo de máquina en PTI*

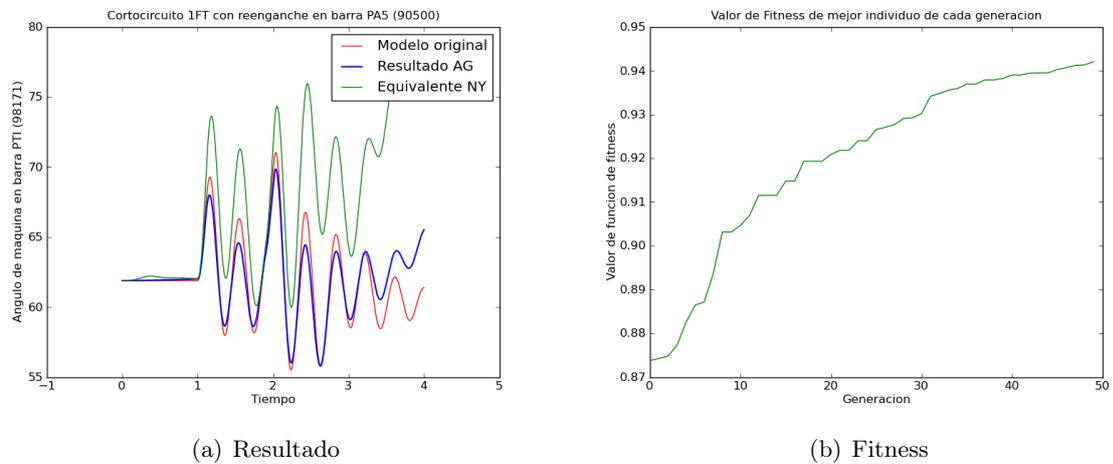


Figura 6.20: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

**Optimización de tensión en fallas trifásicas**

La optimización de la tensión arroja resultados positivos en cuanto a la proximidad de la simulación con el sistema original. Se puede observar también que el fitness da valores muy cercanos a la unidad, por lo que se puede concluir que el comportamiento de la simulación es

casi idéntica a la del sistema real. No fue posible la utilización de reguladores de tensión debido a la falta de tiempo para realizar esa mejora. La inclusión de reguladores de tensión es una de las posibles mejoras del programa a futuro. Estos lograrían mejorar la aproximación de los equivalentes cuando se optimizan otras magnitudes y se observaría una apreciable mejora en tensión.

*Simulación 5 - Cortocircuito 3F en barra MA5  
Optimización de tensión en MA5*

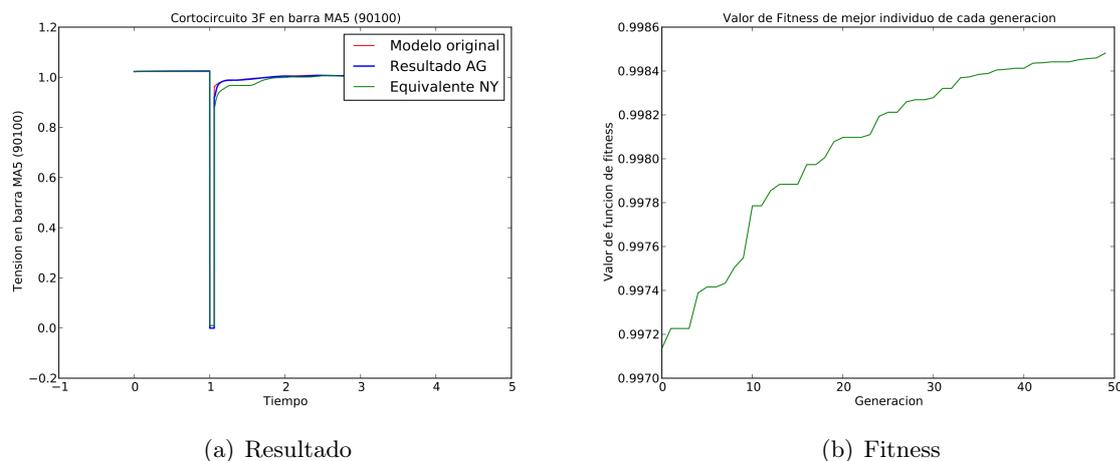


Figura 6.21: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

### Optimización de ángulo de maquina en fallas monofásicas para 9 segundos

La figura 6.22 muestra el comportamiento de la optimización de la falla monofásica con renganche para 9 segundos. Se observa el mismo resultado que en el transitorio de la simulación ya analizada anteriormente. Luego, a medida que se acerca a los 4 segundos, las oscilaciones del equivalente empiezan a caer en amplitud y comienzan a bajar en frecuencia con respecto a las oscilaciones del sistema original. Finalmente, el equivalente converge a un ángulo de 65 grados aproximadamente mientras que el sistema original sigue oscilando. Este comportamiento se puede deber a que el equivalente tiene una constante de amortiguamiento menor que la red original por la simplificación realizada. Esto se explica de manera más detallada en la sección conclusiones.

*Simulación 9.2 - Cortocircuito 1FT con reenganche en barra PA5  
Optimización de ángulo de máquina en PTI*

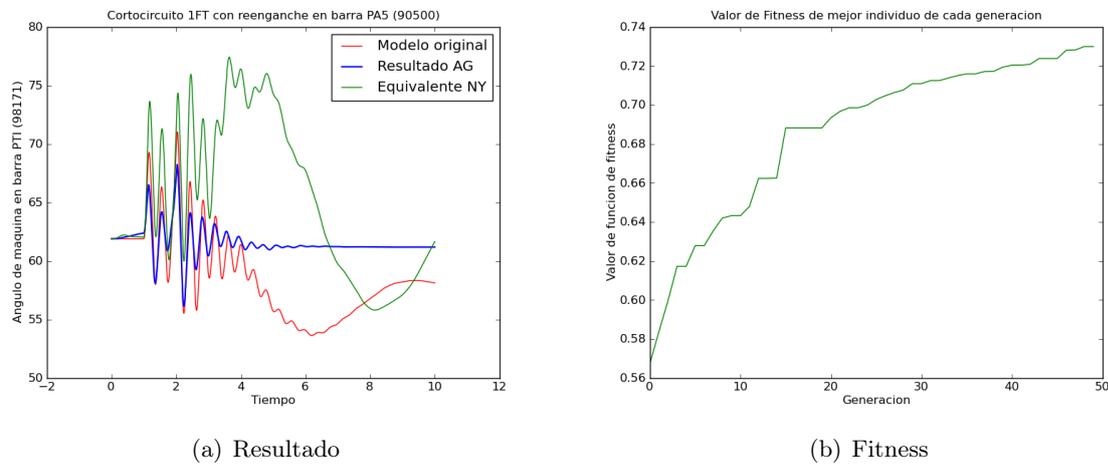


Figura 6.22: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

### 6.2.3. Conclusiones

Se pudieron extraer algunas conclusiones a partir de las simulaciones realizadas. En el anexo C se encuentran todas las gráficas realizadas para cada simulación, en los distintos puntos de la red, las cuales también formaron parte del análisis de este capítulo.

En primer lugar, se encontró que la optimización de más de dos máquinas a la vez no es recomendable y difícilmente se obtengan buenos resultados. Es por esto que se recomienda optimizar de dos en dos las máquinas del sistema equivalente que quiera optimizar.

Se constataron algunos de los resultados más intuitivos. Los resultados más próximos se dieron para la magnitud y el punto que estábamos optimizando. Asimismo, a medida que nos alejamos del punto de optimización y nos acercamos al equivalente, vamos perdiendo precisión en la magnitud que se optimizó. otro punto quizás también intuitivo es que se obtuvieron mejores resultados en general para las faltas monofásicas que para las faltas trifásicas. Esto se desprende de que las faltas monofásicas son las menos agresivas para el sistema, ya que la potencia del cortocircuito es menor, lo que varía en menor medida la velocidad de las máquinas, y por lo tanto la frecuencia y los ángulos  $\delta$  de las máquinas.

Respecto a la optimización de frecuencia, podemos decir que los resultados en la frecuencia en otros puntos de la red se aproxima de buena manera mientras que no sucede lo mismo con la tensión y los ángulos de las máquinas en los otros puntos de la red. También se debe agregar que para este tipo de optimizaciones basta con optimizar la falta para un tiempo entre 4 y 10 segundos, ya que el sistema se estabiliza luego de este punto y vuelve a la frecuencia nominal de la red, 50Hz.

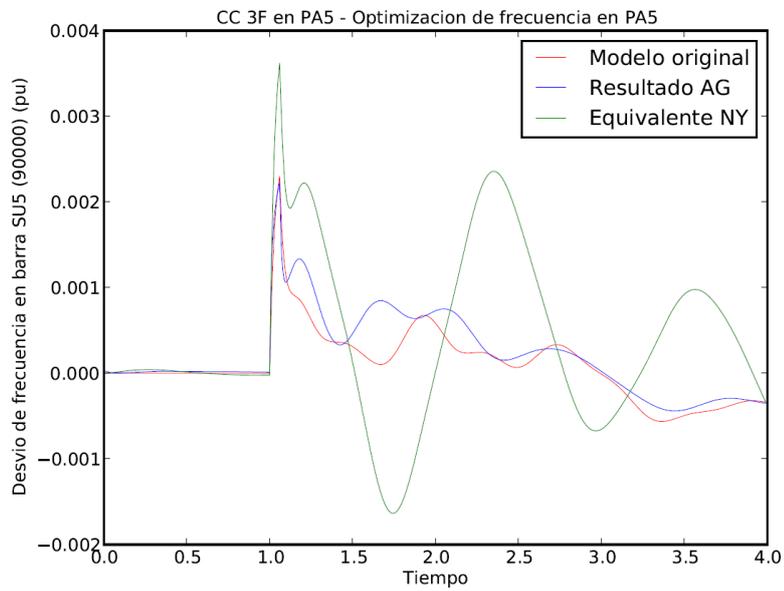


Figura 6.23: Resultado de simulación 2 en frecuencia para 4 segundos.

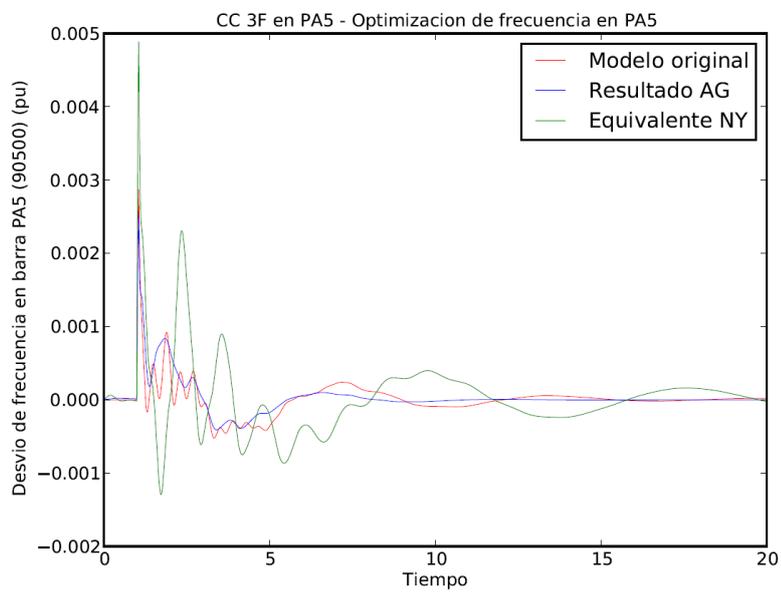


Figura 6.24: Resultado de simulación 2 en frecuencia para 4 segundos. Comportamiento en 20 segundos.

Respecto a la optimización de ángulos, el sistema equivalente se comportó muy similar para los tiempos de optimización utilizados y para la misma magnitud medida en otros puntos de la red. Sin embargo, este resultado obtenido para un tiempo de optimización de 4 segundos no es extrapolable a un tiempo de simulación mucho mayor, ya que ambos sistemas, el original y el equivalente, difícilmente converjan al mismo ángulo luego de ocurrida y despejada la falta.

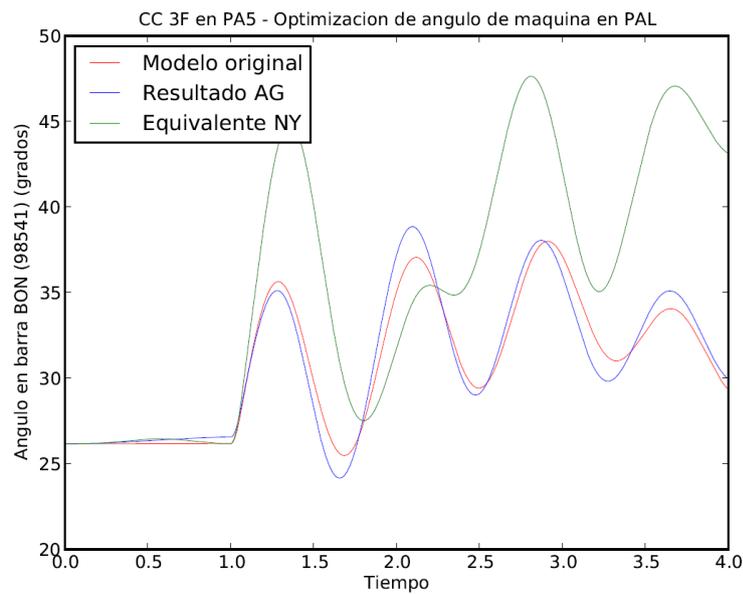


Figura 6.25: Resultado de simulación 3 en ángulo de máquina para 4 segundos.

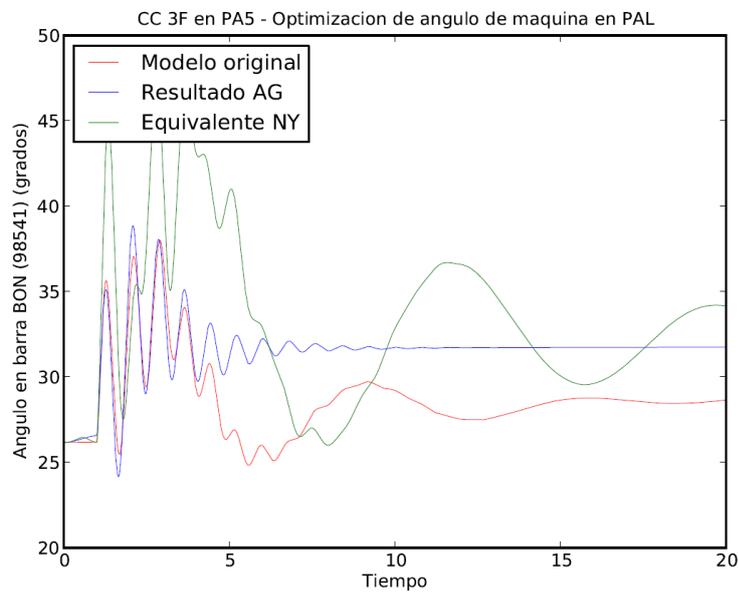


Figura 6.26: Resultado de simulación 3 en ángulo de máquina para 4 segundos. Comportamiento en 20 segundos.

Por esta razón, se recomienda optimizar el ángulo para tiempos mayores a 10 segundos, en el que pueda encontrarse un comportamiento similar para ambos sistemas. Sobre esta optimización también es importante destacar que se obtuvieron mejores resultados en general que optimizando el sistema equivalente por la frecuencia. En pocas palabras, la frecuencia se comportó mejor en la optimización de los ángulos de las máquinas de lo que se comportó el ángulo de las máquinas para las optimizaciones de frecuencia. Esto se puede apreciar de buena manera en las simulaciones 8 y 9 en el anexo C.

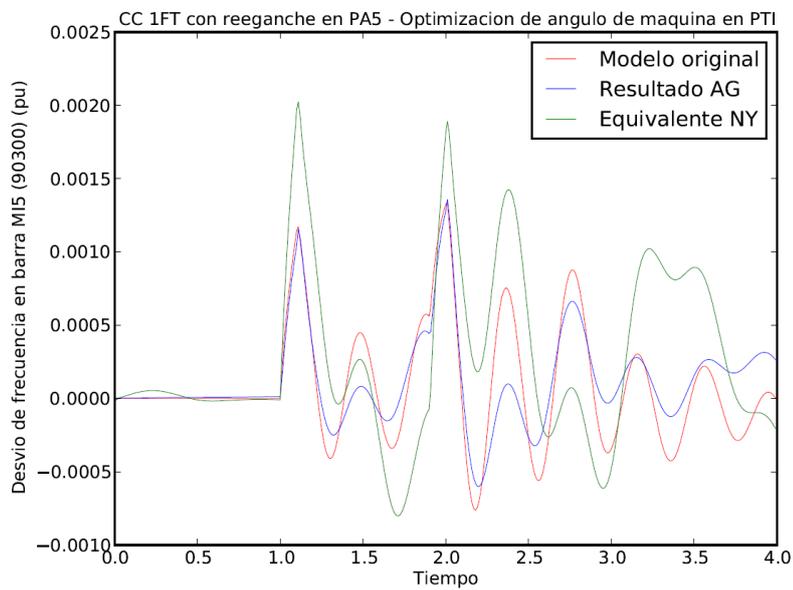


Figura 6.27: Resultado de simulación 9 para frecuencia optimizando ángulo de la máquina.

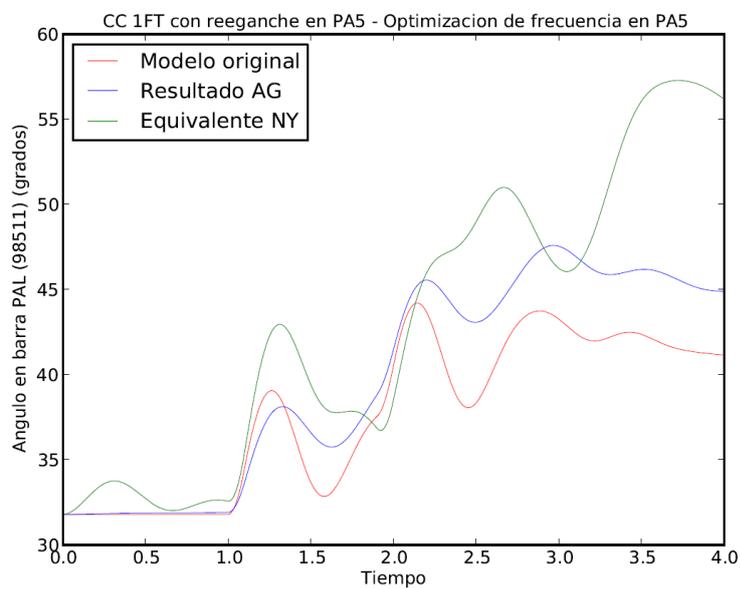


Figura 6.28: Resultado de simulación 7 para ángulo de máquina optimizando frecuencia.

Probablemente la explicación de este resultado se desprende de la relación entre ambas fre-

cuencia y ángulo de máquina:

$$\Omega(t) = \Omega_s + \frac{d\delta}{dt} = \omega_0 n \quad (6.1)$$

$$f = \frac{n}{60p}. \quad (6.2)$$

Entonces, nos queda:

$$\Omega(t) = \Omega_s + \frac{d\delta}{dt} = 60p\omega_0 f. \quad (6.3)$$

Es decir, la frecuencia depende linealmente de la derivada del ángulo  $\delta$  de la máquina. Esto quiere decir que un cambio en la velocidad de la máquina, es decir, un cambio del ángulo  $\delta$ , corresponde a un cambio proporcional en la frecuencia. En otras palabras, la variación del ángulo de la máquina es la causante de la variación de la frecuencia, y al optimizar el sistema desde la magnitud causante del cambio, tenemos un buen resultado también para esta otra magnitud. Esto podría explicar por qué obtenemos buenos resultados en frecuencia al optimizar por el ángulo, y no tanto viceversa.

### 6.3. Análisis cualitativo de fallas

Es interesante analizar las diferentes fallas y el comportamiento del sistema ante cada una de ellas. En este subcapítulo explicaremos brevemente el comportamiento del sistema para cada una de las magnitudes optimizadas, antes de empezar a analizar los resultados propiamente dichos.

#### Tensión

En primer lugar, debemos analizar la tensión ante un cortocircuito. En el caso de un cortocircuito fase-tierra, tendremos un desbalance en las tensiones, una fase se pondrá a 0V mientras las fases “sanas” toman el valor de la tensión compuesta. Por otro lado, en el caso de un cortocircuito trifásico tendremos las 3 fases cortocircuitadas. Por lo tanto, para esta falla tendremos una caída en la tensión en el momento de la falta. En el caso de un cortocircuito trifásico, la tensión caerá a 0V en el punto donde ocurre la falta, mientras que para el cortocircuito fase-tierra habrá una caída de tensión hasta una tensión baja mayor a 0V (alrededor de 0.3 pu) . Es claro que ante esta caída de tensión en un punto de la red, tendremos una caída en la tensión de toda la red.

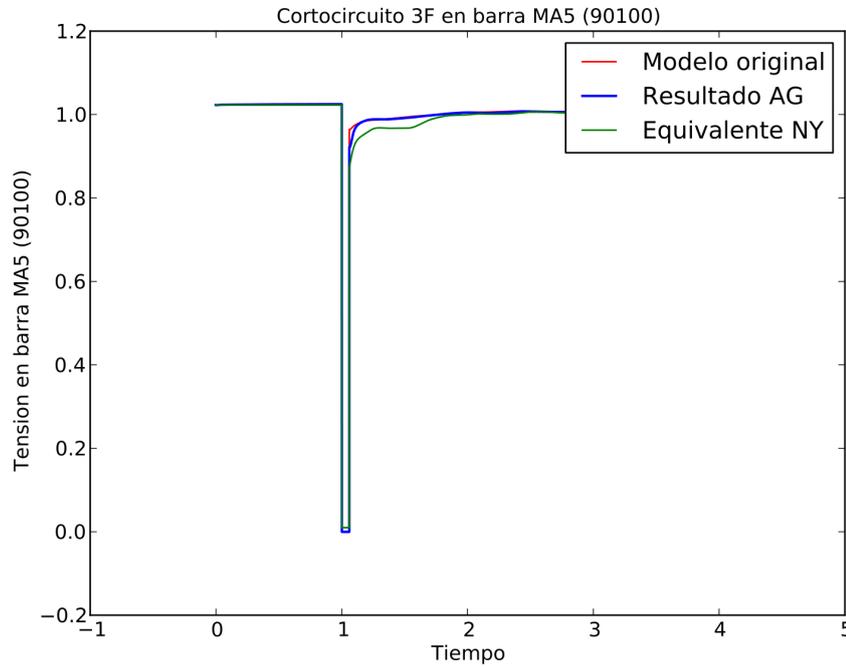


Figura 6.29: Resultado del mejor individuo de todas las generaciones y evolución del fitness.

### Ángulos de máquinas $\delta$ [21]

El ángulo  $\delta$  es el ángulo interno de la máquina, o ángulo entre el campo rotórico y el campo resultante en el EH (entrehierro de la máquina). Este ángulo se relaciona con la potencia y la tensión en bornes de la máquina  $V_S$  con las siguientes ecuaciones.

En máquinas de polos lisos:

$$P = \frac{3V_S E}{X_S} \sin \delta. \quad (6.4)$$

En máquinas de polos salientes:

$$P = \frac{3V_S E}{X_d} \sin \delta + \frac{3V_S^2}{2} \left( \frac{1}{X_q} - \frac{1}{X_d} \right) \sin 2\delta. \quad (6.5)$$

Estas potencias se igualan cuando  $X_d = X_q$ , lo que ocurre para las máquinas de polos lisos.

Por lo tanto, si se simplifica la potencia para una máquina de polos lisos (sólo a efectos de entender este concepto), la podemos graficar como una función del ángulo  $\delta$ .

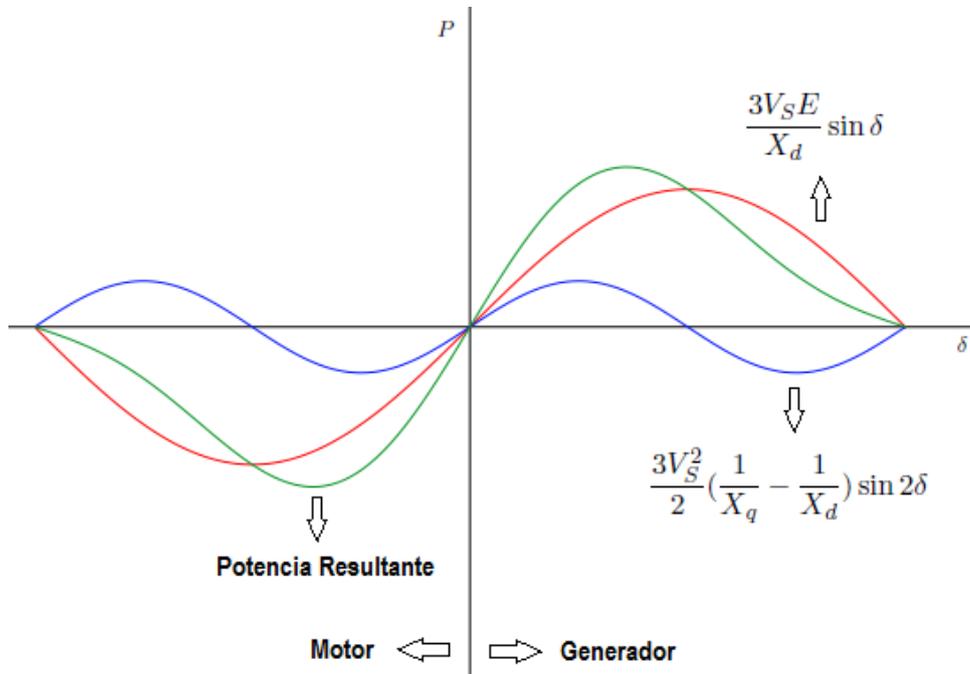


Figura 6.30: Potencia de una máquina síncrona funcionando como motor o generador.

Del capítulo 2.5 sabemos que las ecuaciones de oscilación de la máquina ante las perturbaciones son de la forma:

$$2H \frac{dn}{dt} = P_m - P_e - D_n \quad (6.6)$$

$$\Omega(t) = \Omega_s + \frac{d\delta}{dt} = \omega_0 n. \quad (6.7)$$

Estas ecuaciones las podemos simplificar usando un damping  $D = 0$  (solamente a efectos de explicar este concepto). De esta forma se obtienen las siguientes ecuaciones:

$$2H \frac{dn}{dt} = P_m - P_e \quad (6.8)$$

$$\Omega(t) = \Omega_s + \frac{d\delta}{dt} = \omega_0 n. \quad (6.9)$$

Luego, se deriva 6.9 y se obtiene:

$$\frac{d\Omega(t)}{dt} = 0 + \frac{d(\frac{d\delta}{dt})}{dt} = \omega_0 \frac{dn}{dt}. \quad (6.10)$$

Entonces, con 6.8 y 6.10 se llega a:

$$\frac{2H}{\omega_0} \ddot{\delta}(t) = P_m(\delta) - P_e(\delta). \quad (6.11)$$

Para la cual se puede hallar una preintegral para hallar la velocidad

$$\frac{H}{\omega_0} \dot{\delta}^2(t) = \int_{\delta_0}^{\delta} P_m(\delta) - P_e(\delta) d\delta. \quad (6.12)$$

A partir de la ecuación se puede apreciar el comportamiento del ángulo cuando ocurre una perturbación que lleva de la potencia  $P_{mA}$  a  $P_{mB}$ , siendo  $P_{mA} < P_{mB}$ . Inicialmente se trabaja en el punto  $(\delta_A, P_{mA})$ . Debido a la perturbación la potencia aumenta a  $P_{mB}$ , por lo que  $\delta$  comienza crecer y pasa de  $\delta_A$  a  $\delta_B$ , con  $\dot{\delta} > 0$ . En  $\delta_B$  se da  $P_{mA} = P_{mB}$ , por lo que  $\ddot{\delta}(\delta_B) = 0$  y  $\dot{\delta}(\delta_B) > 0$ . De esta forma,  $\delta$  sigue creciendo aunque  $\ddot{\delta}(\delta) < 0$  cuando se supera  $\delta_B$  y  $\dot{\delta}$  comienza a decrecer.  $\dot{\delta}$  sigue decreciendo hasta que se alcanza el punto  $\delta_C$ , el cual verifica  $\dot{\delta}(\delta_C) = 0$  y  $\ddot{\delta}(\delta_C) < 0$ . Así, el ángulo quedaría oscilando entre los valores  $\delta_A$  y  $\delta_C$  idealmente, pero en la realidad existen bobinados de amortiguamiento que amortiguan la oscilación, extinguiendo la misma hasta estabilizarse en  $\delta_B$ .

También es importante mencionar que un salto muy brusco en la potencia puede provocar que la máquina siga acelerándose ( $\ddot{\delta}(\delta) > 0$ ) hasta superar el límite de estabilidad estática de la máquina síncrona ( $\delta = \frac{\pi}{2}$ ). Aquí se dice que la máquina ha salido de sincronismo.

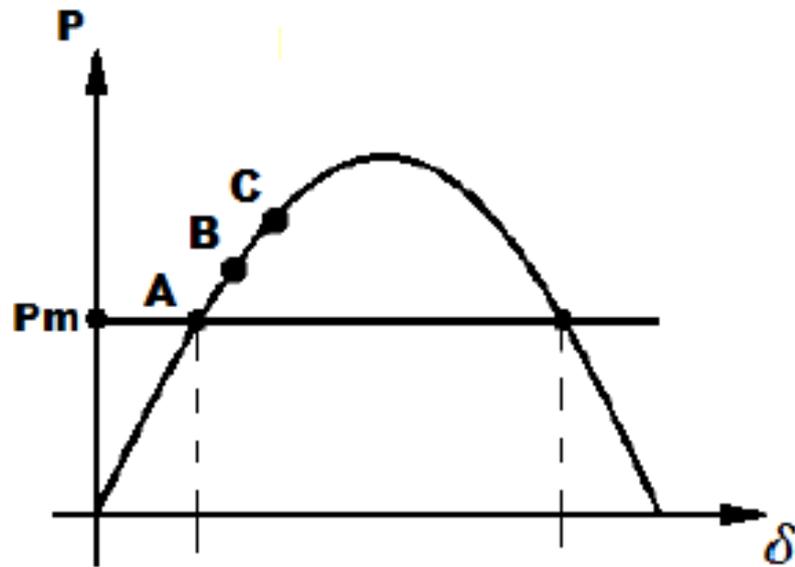


Figura 6.31: Oscilación de potencia.

El aumento del ángulo y la posterior oscilación se pueden ver en los resultados de las simulaciones. Es importante notar que este ejemplo (utilizando  $D = 0$ ) corresponde a una oscilación no amortiguada, la cual no se extinguiría en el tiempo. Es la constante  $D$  de la máquina la que amortigua y extingue la oscilación, y de esta forma también se encuentra un sentido matemático a la constante de amortiguamiento de la máquina.

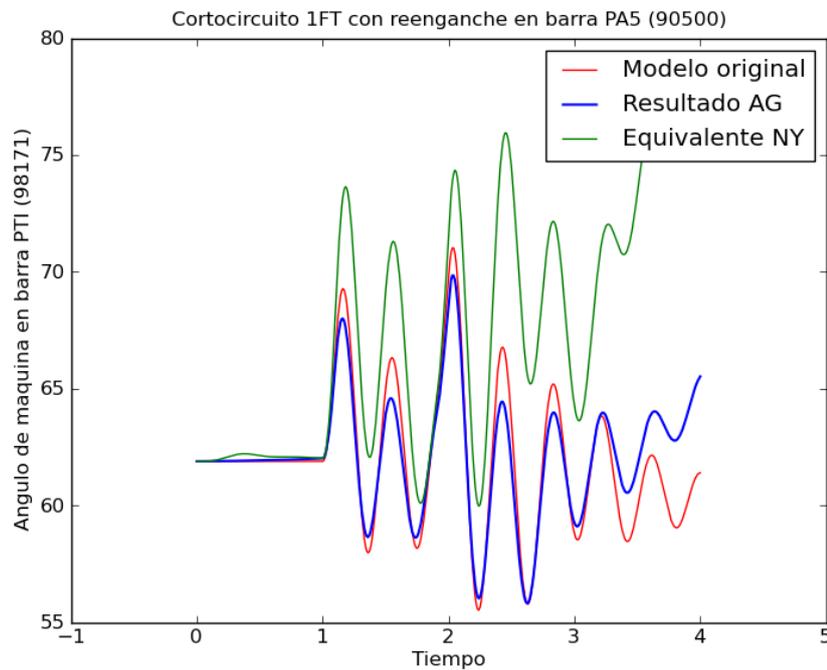


Figura 6.32: Oscilación de los ángulos de las máquinas de Punta del Tigre, resultado de la falla aplicada.

## Frecuencia

En el caso de la frecuencia, se sabe que por tratarse de generadores sincrónicos (alternadores), la frecuencia de salida de la máquina es directamente proporcional a la velocidad del generador:

$$f = \frac{n}{60p}. \quad (6.13)$$

Donde  $n$  es la velocidad del generador en rpm,  $f$  es la frecuencia en Hz y  $p$  es el número de par de polos de la máquina.

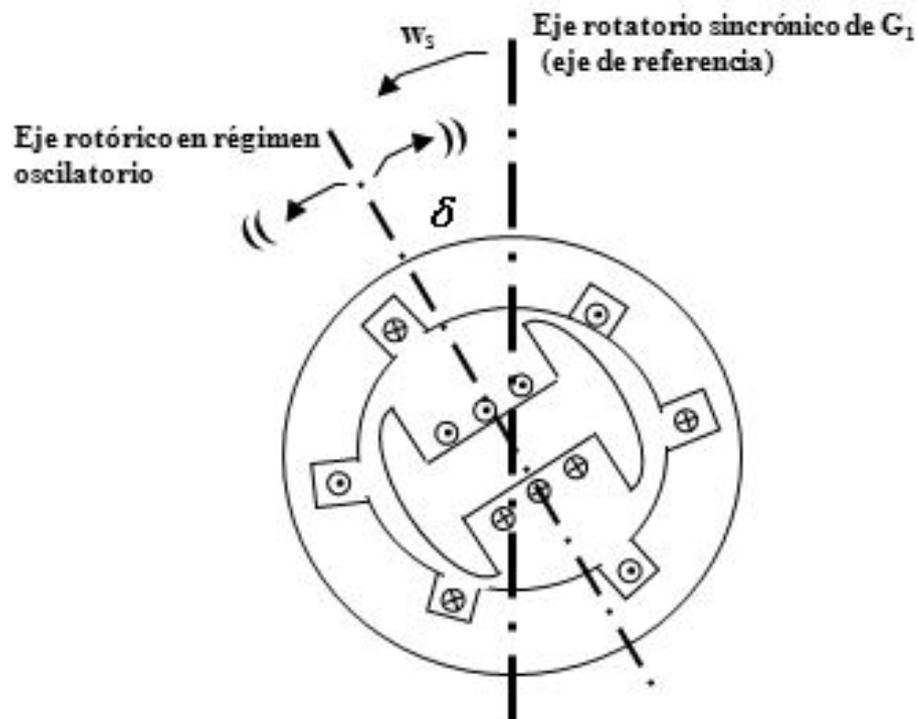


Figura 6.33:

Como se vio en la parte anterior, la falta provoca un aumento en la potencia neta de la máquina que produce un aceleración del eje  $\ddot{\delta} > 0$ , lo que se traduce como el aumento del ángulo  $\delta$  y una posterior oscilación respecto del nuevo punto de equilibrio ( $P_m = P_e$ ), que es amortiguada por arrollamientos internos de la máquina. Este crecimiento y oscilación de  $\delta$ , genera un cambio en la velocidad del eje  $n$ , ya que  $\delta$  es el desplazamiento angular del rotor respecto a un eje que gira a velocidad sincrónica, generando un aumento de la frecuencia de salida de la máquina síncrona y una posterior oscilación hasta volver a la velocidad de sincronismo.

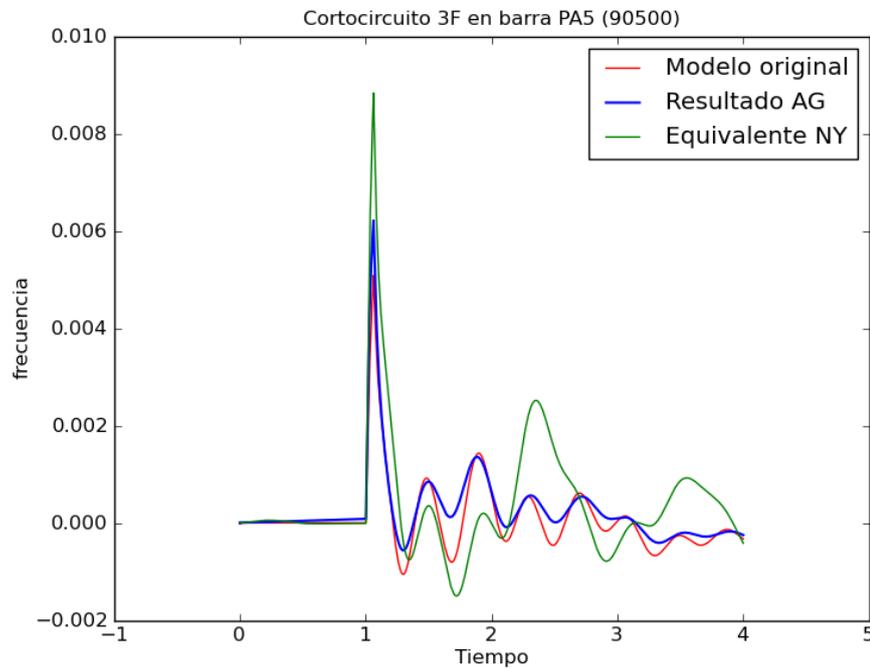


Figura 6.34:

Luego de ocurrido este pico de frecuencia en el que se abre la línea en falta, la frecuencia comienza a oscilar ligeramente hasta llegar nuevamente al punto de equilibrio de la máquina a 50Hz.

#### 6.4. Resultados numéricos

Habiendo analizado las faltas simuladas, se procederá a procesar la información resultado de la simulaciones. De esta forma, se analizarán los datos obtenidos para intentar obtener un equivalente que sea lo más abarcativo posible, ya que en el capítulo anterior se pudo apreciar que es posible obtener un equivalente específico para una falla específica, midiendo una única magnitud en determinado punto de la red, con un margen de error aceptable.

La siguiente tabla muestra un resumen completo de las simulaciones. Las columnas PTI y MVL indican si se pusieron en servicio las respectivas barras en cada una de las simulaciones. La columna alfa corresponde al factor de fitness  $\alpha$  utilizado en la simulación.

Nº Simulación	Falla	Lugar de falla	Equivalente	Lugar de medida	Magnitud a optimizar	Tiempo	Generaciones	Individuos	PTI	MVL	alfa
1	CORTOCIRCUITO TRIFASICO	PA5 con salida de PA5-MA5	NY	M15	frecuencia	4	50	500	SI	NO	1
2				PA5	frecuencia	4	50	500	SI	NO	1
3.1				maq_PAL	ángulo	4	50	500	SI	NO	1
3.2				maq_PAL	ángulo	4	50	500	SI	NO	1
4.1				maq_PTI	ángulo	4	50	500	SI	NO	10
4.2				maq_PTI	ángulo	20	20	120	SI	NO	10
5		MA5 con salida de MA5-M15	NY	MA5	tensión	4	50	500	SI	NO	1
6				maq_PTI	ángulo	4	50	500	SI	NO	10
7	1FT CON REENGANCHE	PA5 con salida de PA5-MA5	NY	PA5	frecuencia	4	50	500	SI	NO	1
8				maq_PAL	ángulo	4	50	500	SI	NO	10
9.1				maq_PTI	ángulo	4	50	500	SI	NO	10
9.2				maq_PTI	ángulo	9	50	500	SI	NO	10
10				M15	frecuencia	4	50	500	SI	NO	1
11				PA5	frecuencia	4	50	500	SI	SI	1
12	PA5 con salida de PA5-MA5	Reducido		maq_PAL	ángulo	4	50	500	SI	SI	1
13				maq_PTI	ángulo	4	50	500	SI	SI	1
14				M15	frecuencia	4	50	500	SI	SI	1

Figura 6.35: Tabla resumen de simulaciones.

Nuestro desafío es procesar estos datos y obtener un equivalente que abarque la mayor cantidad de puntos, sea un equivalente único, un equivalente por falla, o un equivalente por magnitud. En general se encontraron tendencias para las simulaciones en que se optimizaron las mismas magnitudes, aunque también se pudo encontrar una tendencia general en los datos. Esto se aprecia mayormente en la inercia y el damping de las máquinas, las cuales varían ambas en un amplio rango. El damping se varía en un rango de 0 a 40 pu, mientras que la inercia se varía en un rango de 1 a 40 pu. Sobre este punto vale la pena recalcar que los resultados de los equivalentes dieron en general valores muy por encima de los valores límite de las máquinas especificados en la documentación del PSS®E. Como se definió en el capítulo 4.1, la inercia debe variar entre 1 y 10 pu y el damping debe hacerlo entre 0 y 3 pu. De aquí se desprende que con los límites recomendados los resultados no fueron buenos y se decidió ampliar los márgenes para ambas constantes. Respecto a las demás constantes de las máquinas, los márgenes fueron más restrictivos, sea por su sensibilidad y el funcionamiento del programa, así como que la ampliación de los márgenes tampoco arrojaba una mejora sustancial en los resultados, sino que todo lo contrario, resultaba en la caída de python para algunos valores fuera de los márgenes definidos.

A continuación se resumen los datos obtenidos para cada elemento por simulación.

## Ezeiza

- Se encontraron similitudes entre los diferentes resultados para una misma falta y una misma magnitud a optimizar.
- Para el caso de la inercia se encontraron valores del orden para mismas fallas y magnitudes. Incluso la mayoría de los valores anduvieron alrededor de 15.
- En el caso del damping, se encontró que para la mayoría de las simulaciones se tuvo un valor mayor a 30, con una media de aproximadamente 35.

Nº Simulación	Ezeiza (3000)										S1	S2
	T'd0 [seg]	T''d0 [seg]	T''q0 [seg]	H [pu]	D [pu]	Xd [pu]	Xq [pu]	X'd [pu]	X''dq [pu]	Xl [pu]		
1	1,45357850	0,06358579	0,07606269	15,74984346	35,36820891	12,96090107	0,80756300	0,34273671	0,00983584	0,00196034	0,04033351	0,64989807
2	5,43843333	0,03837977	0,17632607	14,97603660	32,52301609	8,45633192	0,79334226	0,30032114	0,00913249	0,00374792	0,04959396	0,29162986
3.1	2,34890321	0,06481571	0,05220749	3,08675961	30,97483836	4,31587562	0,83515015	0,32160845	0,00947844	0,00572101	0,04902164	0,82713742
3.2	8,35398912	0,03576401	0,16292574	4,79741561	33,28722968	6,22499494	0,96629176	0,34539966	0,00982918	0,00593134	0,03418620	0,86401254
4.1	1,79996715	0,07888495	0,03093006	2,72850501	38,03341112	1,59423902	1,29642323	0,32475331	0,00984528	0,00617192	0,03617075	0,85632910
4.2	1,27584786	0,06496685	0,09149025	4,45238842	36,50339157	4,51890562	0,93339913	0,28543884	0,00951007	0,00673470	0,03329975	0,24199342
5	8,42726243	0,03103359	0,06475058	39,52966435	39,77158757	4,05548547	1,44619386	0,26662282	0,00701217	0,00562676	0,04756628	0,52060356
6	2,49245545	0,05673432	0,13206869	20,57996857	39,82039592	2,05875280	1,45781629	0,31805105	0,00976610	0,00158548	0,03131439	0,83118055
7	1,12668239	0,03058381	0,16245596	14,48804131	37,49244925	12,26146873	0,78292716	0,31989990	0,00990110	0,00201514	0,03526608	0,34610152
8	1,10894967	0,03549684	0,17052282	30,38934592	37,37166973	1,68367426	1,29004841	0,34073051	0,00971938	0,00367545	0,03171968	0,84587055
9.1	8,53386999	0,03931999	0,09089478	37,19207492	39,68923881	1,50836416	1,48484564	0,31466414	0,00988368	0,00120803	0,03846039	0,71679183
9.2	1,14868706	0,06217473	0,17858732	3,43496137	36,24378085	1,54589527	0,86189232	0,34793752	0,00990182	0,00154077	0,03809870	0,85904875
10	1,19138218	0,03323133	0,04790921	5,88274130	18,74218199	14,39781923	0,99369162	0,27216366	0,00975303	0,00664489	0,04416953	0,22340324
Promedio	3,43846218	0,04884398	0,11054859	15,17598050	35,06318460	5,81405447	1,07304499	0,31540982	0,00950528	0,00404336	0,03916930	0,62107695
Máximo	8,53386999	0,07888495	0,17858732	39,52966435	39,82039592	14,39781923	1,48484564	0,34793752	0,00990182	0,00673470	0,04959396	0,86401254
Mínimo	1,10894967	0,03058381	0,03093006	2,72850501	18,74218199	1,50836416	0,78292716	0,26662282	0,00701217	0,00120803	0,03131439	0,22340324

Figura 6.36: Valores de máquina de Ezeiza.

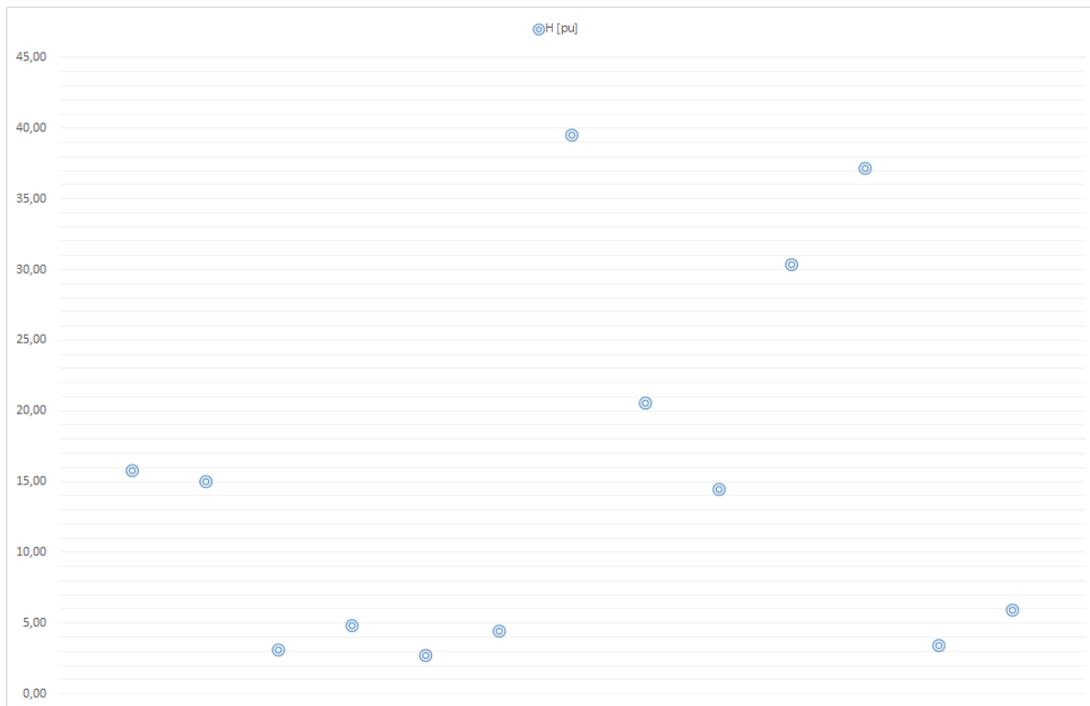


Figura 6.37: Dispersión de la inercia H[pu] de Ezeiza.

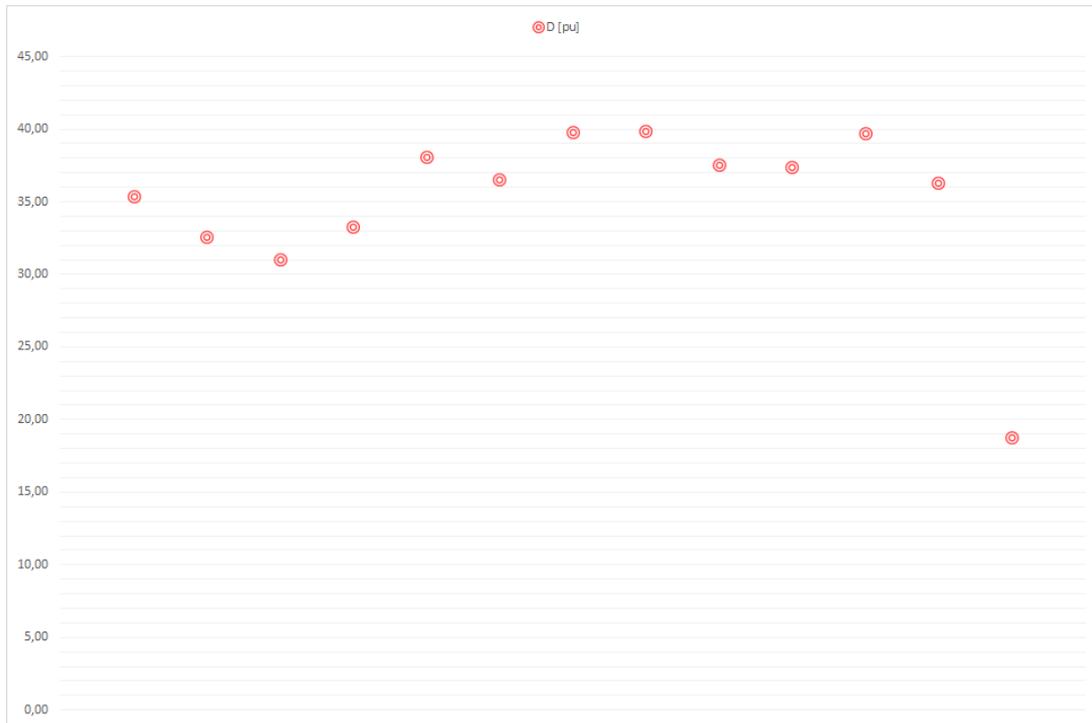


Figura 6.38: Dispersión del damping  $D$ [pu] de Ezeiza.

## Almafuerte

- Se encontraron similitudes entre los diferentes resultados para una misma falta y una misma magnitud a optimizar.
- Para el caso de la inercia se encontraron valores del mismo orden nuevamente para mismas fallas y magnitudes. Incluso la mayoría de los valores anduvieron alrededor de 15.
- Al igual que en el caso de la máquina de Ezeiza, se encontró un damping para la mayoría de las simulaciones mayor a 30, con una media de aproximadamente 32.

Almafuerte (106000)												
Nº Simulación	T <sup>o</sup> d0 [seg]	T <sup>o</sup> d1 [seg]	T <sup>o</sup> q0 [seg]	H [pu]	D [pu]	Xd [pu]	Xq [pu]	X'd [pu]	X''dq [pu]	Xl [pu]	S1	S2
1	2,09158831	0,06587663	0,03751523	17,33428682	20,06204872	7,61484928	1,16613984	0,31853376	0,00999902	0,00536305	0,03137805	0,29401463
2	1,82086038	0,10291435	0,07379769	16,45621139	24,79482318	11,74578202	1,45727374	0,33658735	0,00884665	0,00331431	0,03386537	0,35284983
3.1	5,71816357	0,03202430	0,18371684	5,65345476	35,18387949	7,36782619	0,79799895	0,27634655	0,00963027	0,00378619	0,03496670	0,69270077
3.2	3,50196691	0,04799237	0,13269133	15,51015696	37,25016582	2,15965773	0,85025236	0,29982567	0,00913729	0,00297511	0,03745254	0,79118434
4.1	4,25045649	0,12538566	0,07939864	2,11441717	32,35322772	2,01433720	0,81043987	0,30655223	0,00995969	0,00194877	0,04965236	0,82943593
4.2	9,62686346	0,15439115	0,14231289	13,34472960	27,20820755	3,53614058	1,43958658	0,28005808	0,00972446	0,00401397	0,03893269	0,86844855
5	7,89740437	0,03055716	0,14544744	1,09938429	39,24495932	3,02711637	0,87592301	0,29981879	0,00825276	0,00482321	0,03782328	0,87998275
6	5,31604880	0,18800508	0,07410785	15,20903694	38,55567304	2,31481221	1,00450136	0,29651048	0,00939584	0,00602543	0,03360786	0,89658215
7	5,94311148	0,04618038	0,03429893	10,15160875	26,04565673	4,15174336	1,13855391	0,32295715	0,00993208	0,00512595	0,03403830	0,58605387
8	5,86820064	0,05665893	0,05981905	12,33013535	39,37378385	3,15721950	0,74678038	0,27319470	0,00990323	0,00375690	0,03717301	0,68151694
9.1	2,52067573	0,04603316	0,15264907	38,90378234	37,28668594	11,06771888	0,80687700	0,26459189	0,00993043	0,00395771	0,03369179	0,82568858
9.2	1,48471197	0,13763645	0,17197867	11,24145892	24,64880684	2,04168692	0,83095836	0,26001384	0,00986344	0,00177222	0,04774915	0,88030242
10	1,97937075	0,06522161	0,10067506	29,00210408	35,25077366	6,81513564	1,30592977	0,32129604	0,00998967	0,00679178	0,03454763	0,88493317
Promedio	4,46303253	0,08452902	0,10680067	14,48852057	32,09682245	5,15492507	1,01778578	0,29663743	0,00958191	0,00412728	0,03729836	0,72797646
Máximo	9,62686346	0,18800508	0,18371684	38,90378234	39,37378385	11,74578202	1,45727374	0,33658735	0,00999902	0,00679178	0,04965236	0,89658215
Mínimo	1,48471197	0,03055716	0,03429893	1,09938429	20,06204872	2,01433720	0,74678038	0,26001384	0,00825276	0,00177222	0,03137805	0,29401463

Figura 6.39: Dispersión de la inercia H[pu] de Almafuerte.

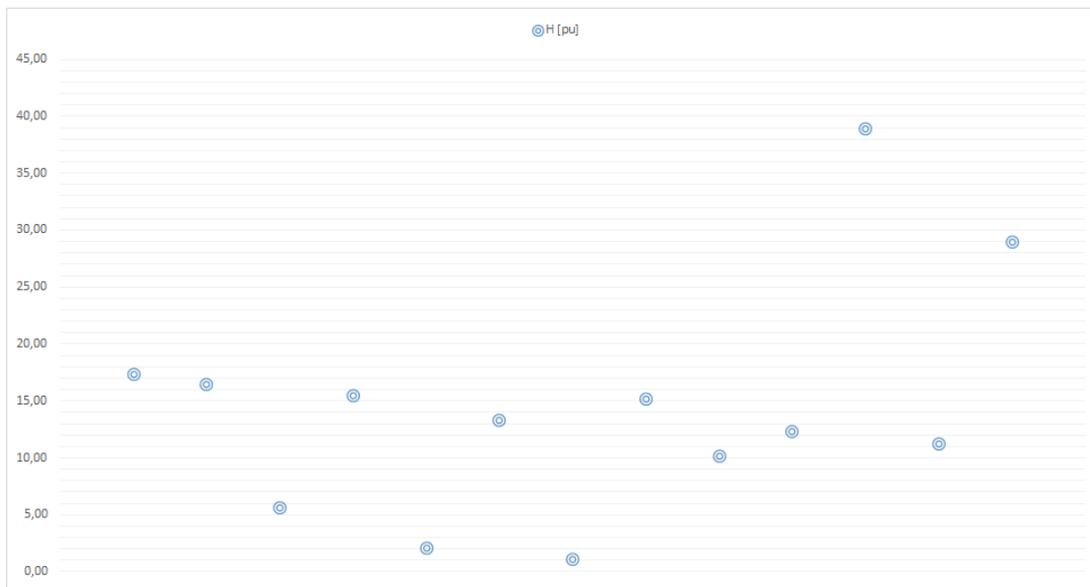


Figura 6.40: Dispersión del damping D[pu] de Almafuerte.



Figura 6.41: Valores de máquina de Ezeiza.

## Líneas de Transmisión

- Los valores de líneas también se comportaron muy similares entre mismas faltas y magnitudes a optimizar. En el caso de la reactancia X de la línea 4004-5002 para las simulaciones 3.1 y 3.2 los valores fueron casi exactos.
- Los valores se encontraron en el promedio de los límites impuestos en la inicialización de las variables. Esto nos da una idea de que los límites impuestos están dentro de un margen aceptable.

Nº Simulación	Línea 4000-106000			Línea 4010-4000			Línea 5002-4010			Línea 4004-5002		
	R [pu]	X [pu]	B [pu]	R [pu]	X [pu]	B [pu]	R [pu]	X [pu]	B [pu]	R [pu]	X [pu]	B [pu]
1	0,0001881	0,00234421	0,83001270	0,00870626	0,00928775	0,16460903	0,00964326	0,00331115	0,13440561	0,00854738	0,00797305	0,34872316
2	0,00049485	0,00391930	0,07020474	0,00321631	0,00307562	0,02779005	0,00944055	0,00122598	0,35002231	0,00178741	0,00988495	0,18025431
3.1	0,00065675	0,00896536	0,06585081	0,00223383	0,00136883	0,32062156	0,00668051	0,00756379	0,08302496	0,00966031	0,00939857	0,82627991
3.2	0,00046299	0,00754053	0,57664259	0,00257842	0,00068179	0,23619764	0,00821652	0,00569533	0,90394647	0,00747289	0,00914721	0,83932825
4.1	0,00012876	0,00269064	0,62261727	0,00030007	0,00085145	0,09101498	0,00912501	0,00015774	0,97016273	0,00972533	0,00939854	0,47575646
4.2	0,00059688	0,00571901	0,87750491	0,00327477	0,00122856	0,21007267	0,00989811	0,00631133	0,46417425	0,00046524	0,00891618	0,90907681
5	0,00081449	0,00688799	0,12293859	0,00228009	0,00028607	0,57820454	0,00394262	0,00911786	0,60786358	0,00051816	0,00849648	0,62191342
6	0,00137277	0,00329546	0,20851129	0,00711719	0,00575815	0,25264983	0,00351515	0,00663022	0,46315455	0,00002858	0,00625370	0,89843284
7	0,00097614	0,00932819	0,24548740	0,00152174	0,00206174	0,86627046	0,00521444	0,00998925	0,61112224	0,00982224	0,00978891	0,60164636
8	0,00149834	0,00583732	0,83329737	0,00013162	0,00108628	0,24878399	0,00992689	0,00734924	0,95606990	0,00997882	0,00908233	0,49681565
9.1	0,00039462	0,00197404	0,53555182	0,00011866	0,00148571	0,09723800	0,00871925	0,00723401	0,29918416	0,00984607	0,00962799	0,97083583
9.2	0,00159616	0,00011117	0,47984887	0,00048416	0,00005355	0,14674854	0,00286154	0,00853177	0,34772913	0,00009618	0,00951240	0,72831855
10	0,00048792	0,000730859	0,90485969	0,00408686	0,00405474	0,83809426	0,00554273	0,00769798	0,70222951	0,00109330	0,00582106	0,69941615
Promedio	0,00073073	0,00507091	0,49025600	0,00277308	0,00240617	0,31371504	0,00713281	0,00621659	0,53023765	0,00531050	0,00871549	0,66129213
Máximo	0,00159616	0,00932819	0,90485969	0,00870626	0,00928775	0,86627046	0,00992689	0,00998925	0,97016273	0,00997882	0,00988495	0,97083583
Mínimo	0,00001881	0,00011117	0,06585081	0,00011866	0,00005355	0,02779005	0,00286154	0,00015774	0,08302496	0,00002858	0,00582106	0,18025431

Figura 6.42: Valores de líneas.

Nº Simulación	Línea 3004-4004			Línea 3000-3004			Línea 3004-2018		
	R [pu]	X [pu]	B [pu]	R [pu]	X [pu]	B [pu]	R [pu]	X [pu]	B [pu]
1	0,00223078	0,00095191	0,03581955	0,00812597	0,00972801	0,92132934	0,00076898	0,00960890	0,10358360
2	0,00152018	0,00950297	0,85248955	0,00689555	0,00813616	0,51040624	0,00094094	0,00629301	0,31582865
3.1	0,00194567	0,00196719	0,31063698	0,00895893	0,00591583	0,48533004	0,00795234	0,00293546	0,81709701
3.2	0,00479888	0,00019968	0,06839362	0,00776678	0,00845873	0,00348325	0,00801865	0,00951519	0,91695686
4.1	0,00510445	0,00018024	0,45826121	0,00401854	0,00530763	0,23140501	0,00651172	0,00856489	0,32630166
4.2	0,00709235	0,00142879	0,15516397	0,00232813	0,00128905	0,46447578	0,00773161	0,00075060	0,86414167
5	0,00564267	0,00085416	0,54410395	0,00297742	0,00062716	0,19821299	0,00841076	0,00394292	0,76602127
6	0,00986853	0,00110361	0,44716366	0,00978228	0,00726864	0,95013421	0,00102709	0,00395179	0,40200342
7	0,00176656	0,00997960	0,06736501	0,00333638	0,00869878	0,77339162	0,00021353	0,00762539	0,78115060
8	0,00490423	0,00633213	0,74331475	0,00949803	0,00998011	0,54075146	0,00634925	0,00875242	0,29890396
9.1	0,00860300	0,00072924	0,96047950	0,00957245	0,00963565	0,63576136	0,00079172	0,00454955	0,26463845
9.2	0,00979564	0,00004116	0,25405241	0,00892323	0,00294569	0,93017213	0,00101021	0,00980305	0,77325737
10	0,00032037	0,00942682	0,45925578	0,00947500	0,00012000	0,49369993	0,00000739	0,00114738	0,28995173
Promedio	0,00489179	0,00328442	0,41203846	0,00705067	0,00600857	0,54911949	0,00382571	0,00595697	0,53229510
Máximo	0,00986853	0,00997960	0,96047950	0,00978228	0,00998011	0,95013421	0,00841076	0,00980305	0,91695686
Mínimo	0,00032037	0,00004116	0,03581955	0,00232813	0,00012000	0,00348325	0,00000739	0,00075060	0,10358360

Figura 6.43: Valores de líneas.

Nº Simulación	Línea 4000-2018			Línea 4010-4008			Línea 5002-4008		
	R [pu]	X [pu]	B [pu]	R [pu]	X [pu]	B [pu]	R [pu]	X [pu]	B [pu]
1	0,00109903	0,00197886	0,95575901	0,00994630	0,00137638	0,30974204	0,00935933	0,00934754	0,64544739
2	0,00557754	0,00423426	0,24386990	0,00008288	0,00417812	0,17352537	0,00955768	0,00987076	0,85599214
3.1	0,00802138	0,00094482	0,02421729	0,00352528	0,00418101	0,65109566	0,00012241	0,00908909	0,98268801
3.2	0,00725758	0,00575045	0,12713861	0,00109412	0,00024044	0,60442155	0,00194711	0,00921464	0,11425972
4.1	0,00249421	0,00873290	0,91995550	0,00489931	0,00040944	0,29248577	0,00024542	0,00995886	0,96015291
4.2	0,00654180	0,00859930	0,43565085	0,00467411	0,00058000	0,55995651	0,00238567	0,00569417	0,19161167
5	0,00817375	0,00578012	0,43032181	0,00551670	0,00011400	0,40355965	0,00532079	0,00774147	0,63388318
6	0,00873041	0,00350895	0,53922035	0,00985159	0,00100127	0,02048064	0,00029193	0,00838665	0,58231226
7	0,00953630	0,00959525	0,84995046	0,00011372	0,00110055	0,91276893	0,00423648	0,00979865	0,01653355
8	0,00399811	0,00589969	0,87465899	0,00192076	0,00034670	0,41182018	0,00123276	0,00980120	0,58672093
9.1	0,00043384	0,00759096	0,08975998	0,00330116	0,00213023	0,18675970	0,00885404	0,00963570	0,67287381
9.2	0,00116955	0,00508338	0,68728024	0,00345828	0,00046363	0,85662310	0,00039432	0,00964330	0,89589453
10	0,00246318	0,00000024	0,52471316	0,00741843	0,00876722	0,71392388	0,00849034	0,00331526	0,80215020
Promedio	0,00503821	0,00520763	0,51557663	0,00429251	0,00191454	0,46901254	0,00403371	0,00857671	0,61080925
Máximo	0,00953630	0,00959525	0,95575901	0,00994630	0,00876722	0,91276893	0,00955768	0,00995886	0,98268801
Mínimo	0,00043384	0,00000024	0,02421729	0,00008288	0,00011400	0,02048064	0,00012241	0,00331526	0,01653355

Figura 6.44: Valores de líneas.

## 6.5. Conclusiones generales

El primer punto a concluir es que los algoritmos genéticos tienen un buen desempeño en la optimización de equivalentes dinámicos de estas dimensiones y su aplicación simplifica de gran forma la tarea que proponen los métodos tradicionales, además de simplificar los cálculos enormemente. Es claro que la topología del equivalente dinámico debe ser definida previamente, pero esto se puede realizar fácilmente ubicando los elementos necesarios en puntos estratégicos de la red.

En el caso de la red argentina, se logra una reducción a 3 máquinas, 10 líneas y 7 barras, y se obtienen resultados con un margen de error aceptable. Esta es una reducción muy significativa ya que la red argentina tiene un número de barras que es aproximadamente 10 veces mayor que el de la red eléctrica uruguaya. El hecho de optimizar un equivalente de este tamaño ya nos impone limitantes para representar todo el sistema argentino. Sin embargo, los resultados fueron prometedores, dando lugar a futuras ampliaciones que podrían mejorar sustancialmente los resultados con sólo agregar algunos elementos y optimizarlos con el algoritmo.

Respecto a la estabilidad del sistema equivalente cuando se realiza la optimización, se llegó a la conclusión que el sistema se comporta de manera mucho más estable cuando se optimiza con pocos elementos a la vez. En nuestro caso se optimizaron 2 máquinas y 10 líneas y esto fue suficiente para tener buenos resultados. Cuando se intentó optimizar de a 3 o 4 máquinas a la vez, se daban condiciones iniciales sospechosas y además muchas veces el programa se detenía provocando la caída de python.

En cuanto al análisis de convergencia del algoritmo, los resultados obtenidos nos confirmaron que el algoritmo no solo funciona correctamente en cuanto a los resultados obtenidos, sino que los valores de las variables optimizadas tienden a ciertos valores únicos, o a entornos de valores en los cuales se comporta de buena manera.

Otro punto a tener en cuenta es que la precisión del algoritmo depende de la magnitud que se esté optimizando, y en el punto en el que se está optimizando. Vale la pena destacar que se tuvo una mayor precisión en otras magnitudes al optimizar el sistema por el ángulo de las máquinas.

Del análisis de las gráficas para los distintos tiempos se pudieron comprobar varios puntos:

- Se tuvieron muy buenos resultados en la optimización de frecuencia para las distintas faltas. Incluso los resultados para tiempos de simulación de 4 segundos pueden ser utilizados para tiempos mayores ya que ambos valores logran estabilizarse luego de unos pocos segundos a la frecuencia nominal.
- En el caso de los ángulos de las máquinas, los resultados son muy buenos para tiempos cortos como los que se simuló, pero no logran ajustarse para tiempos mayores, ya que los ángulos del sistema original y el equivalente no logran coincidir luego de estabilizarse el sistema.
- Al ajustar el equivalente dinámico por los ángulos de las máquinas, el sistema tiene un buen comportamiento en frecuencia (simulaciones 8, 9.1). Esto se debe a que la frecuencia debe estabilizarse siempre al mismo valor de la red, así como se explicó anteriormente en 6.2.3 que una variación del ángulo de las máquinas causa una variación en la frecuencia.

- Los resultados en tensión fueron buenos al optimizar justamente en tensión en el lugar de la falta y en lugares cercanos, pero no ocurrió lo mismo cuando se intentó ajustar por otras magnitudes. Esto se debe principalmente a que no se optimizaron reguladores de tensión.

También se ha constatado que las oscilaciones de las máquinas, y por lo tanto el comportamiento dinámico de la red, está regido principalmente por las constantes de inercia  $H$  y el damping  $D$ , como se desprende de la ecuación de oscilación 6.6, y de las constantes de las líneas de transmisión  $R$ ,  $X$  y  $B$ .

De la ecuación de oscilación también podemos inferir que la oscilación de la máquina depende mayoritariamente de estas dos constantes: la inercia de la máquina marcará la frecuencia de las oscilaciones mientras que el damping amortiguará la oscilación, marcando el pico superior y el decrecimiento posterior de los picos.

El hecho de haber obtenido valores de la constante  $H$  mucho mayores a los habituales puede explicarse a partir del método de coherencia lenta de la teoría de equivalentes dinámicos, en el que se deben sumar las inercias de las máquinas coherentes cuando éstas se reducen. Esto tiene sentido ya que las máquinas optimizadas representan un número mucho mayor de máquinas coherentes. Respecto a los resultados del damping  $D$  de las máquinas, los valores altos podrían explicarse por la reducción de la red eléctrica y de su componente resistivo, lo que se compensa con un incremento en la constante de amortiguamiento.

## 6.6. Perspectivas a futuro

Se quiere enfatizar aquellos puntos que quedaron por fuera del proyecto, sea por falta de tiempo o por limitantes del proyecto en sí, así como aquellos puntos en los que se podría continuar la investigación a futuro.

En primer lugar, formaba parte de la idea original la implementación de una interfaz gráfica que facilite el uso del programa. La principal razón por la que no se realizó este punto fue por falta del tiempo necesario para poder realizarla, aunque también se debe a que el programa sufrió modificaciones sustanciales hasta último momento. De todas formas, previendo la posible falta de tiempo para llegar a una interfaz, se implementó el algoritmo de manera tal que el usuario deba modificar la menor cantidad de variables posibles y creando opciones para distintos funcionamientos, como puede ser la elección de magnitud a optimizar ingresando el nombre en la inicialización de "salida". Más allá de los puntos anteriores, debe tenerse en cuenta que se trata de una herramienta de investigación para un usuario avanzado, el cual debería poseer los conocimientos necesarios para poder utilizarla, y no un producto comercial.

En segundo lugar, queremos detenernos en aquellos elementos del equivalente que no se optimizaron, y que quizás hubiese valido la pena ver el desempeño del sistema ante la variación de los mismos. Algunos de los elementos más importantes que no se optimizaron fueron los diferentes reguladores de tensión. Estos podrían haber mejorado aquellas simulaciones de tensión, claramente, pero también podrían influir positivamente en la optimización de la frecuencia y los ángulos de las máquinas.

Otra de las ideas que no logró llevarse a cabo por diferentes motivos fue la optimización de múltiples magnitudes a la vez. En realidad, logró implementarse el programa necesario para

llevar a cabo la optimización de frecuencia, tensión y el ángulo de una máquina, pero este no llegó a calibrarse acertadamente la función de fitness y el “peso” de cada magnitud en el valor de este.

Sobre los tres puntos detallados anteriormente, se obtiene como conclusión que la dificultad más grande estaría en la implementación de una interfaz que funcione de manera más amigable para el usuario. Tanto la optimización de los reguladores de tensión como la posible optimización de varias magnitudes a la vez, son fácilmente abordables llegado a este punto.

## Referencias

- [1] CURSO DE INSTALACIONES ELÉCTRICAS, FACULTAD DE INGENIERÍA, UDELAR, *Tema 1: Introducción al curso de instalaciones eléctricas de baja tensión, página 2.* [https://eva.fing.edu.uy/pluginfile.php/63189/mod\\_resource/content/1/Contenido/0100-Introduccion.pdf](https://eva.fing.edu.uy/pluginfile.php/63189/mod_resource/content/1/Contenido/0100-Introduccion.pdf)
- [2] ADME, *Informe Anual 2016 - Distribución de líneas de 500 kV y 150 kV, página 5* <http://latorre.adme.com.uy/mmee/pdf/informes/anual/InformeAnual2016.pdf>
- [3] UTE, *Energía producida en Uruguay por fuente* <http://portal.ute.com.uy/institucional-nuestro-patrimonio/sistema-eléctrico>
- [4] ADME, *Pico máximo de demanda en Uruguay para el año 2016* <http://adme.com.uy/mmee/maximosdemandahistorico.php>
- [5] CAMESSA, *Potencia eléctrica instalada en Argentina por fuente a Diciembre del 2015* <http://portalweb.cammesa.com/Documentos%20compartidos/Informes/Informe%20Anual%202015.pdf>
- [6] MINISTERIO DE FINANZAS, PRESIDENCIA DE LA NACIÓN DE LA REPÚBLICA ARGENTINA, *Potencia eléctrica instalada en Argentina a Diciembre 2015, se tomo como referencia la pagina que se detalla a continuación pero con los datos actualizados a Diciembre 2015* [http://www.economia.gob.ar/peconomica/informe/notas\\_tecnicas/22%20NOTA%20TECNICA%20Nivel%20de%20Actividad%20%20inf%2070.pdf](http://www.economia.gob.ar/peconomica/informe/notas_tecnicas/22%20NOTA%20TECNICA%20Nivel%20de%20Actividad%20%20inf%2070.pdf)
- [7] REPRESA HIDROELÉCTRICA SALTO GRANDE ARGENTINA-URUGUAY, *Cuadrilátero de Potencia* <https://www.saltogrande.org/transmision.php>
- [8] CAMESSA, *Esquema geográfico sistema interconectado Argentino-Uruguayo. Sistemas Paraguay, Chileno y sur de Brasil* [http://portalweb.cammesa.com/Descargas%20%20archivos/GEOSADI2017\\_06.pdf](http://portalweb.cammesa.com/Descargas%20%20archivos/GEOSADI2017_06.pdf)
- [9] CAMESSA, *Esquema unifilar sistema interconectado Argentino-Uruguayo. Interconexión con Brasil. Sistema Paraguay sur* [http://portalweb.cammesa.com/Descargas%20%20archivos/UNI2017\\_06\\_TENSION.pdf](http://portalweb.cammesa.com/Descargas%20%20archivos/UNI2017_06_TENSION.pdf)
- [10] PRESIDENCIA DE LA REPÚBLICA ORIENTAL DEL URUGUAY, *Estación convertora de Melo* <https://www.presidencia.gub.uy/comunicacion/comunicacionnoticias/ute-sosa-pruebas-convertora-melo-acuerdo-brasil>
- [11] M. ARTENSTEIN, A. GIUSTO, *Equivalent model of the Argentinian Electrical Power System for Stability Analysis of the Uruguayan Network*, 2008 IEEE T&D LA Conference, Agosto de 2008.
- [12] P. MONZÓN, A. GIUSTO, *Análisis Modal del Sistema Eléctrico Uruguayo*, IEEE Uruguay, Octubre de 2008.

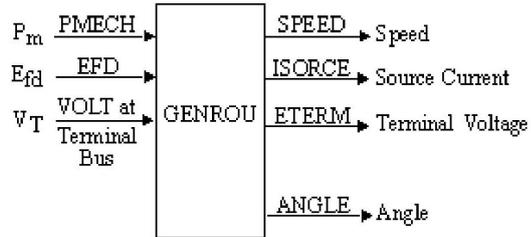
- [13] DR. ING. JAIME CEPEDA, ING. JOSÉ AIMARA, *Identificación del equivalente dinámico del sistema colombiano para realizar análisis de estabilidad en el sistema ecuatoriano*, Corporación Centro Nacional de Control de Energía -CENACE-, 2014.
- [14] JOSÉ JUAN MANUEL RAMÍREZ ARREONDO, *Técnicas Modernas para el Análisis de Estabilidad Transitoria en Sistemas Eléctricos Interconectados Mediante la Aplicación de Equivalentes Dinámicos*, Universidad Autónoma de Nuevo León, Facultad de Ingeniería Mecánica y Eléctrica, Tesis de doctorado, Monterrey N.L. Diciembre de 1992.
- [15] *Siemens PSSE 32.0.5 Volume 2 Program Application Guide, Chapter 12.4 Classification of Variables*, Octubre 2010.
- [16] MELANIE MITCHELL, *An Introduction to Genetics Algorithms*, A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England, Sixth Printing, 1999.
- [17] CARLOS GARCÍA, EDWIN GARCÍA, FERNANDO VILLADA, *Algoritmo evolutivo eficiente aplicado a la planeación de la expansión de sistemas de distribución*, Proyecto de fin de carrera, Universidad Carlos III, Madrid Octubre de 2011. <http://www.scielo.cl/pdf/infotec/v23n4/art02.pdf>
- [18] FRANCISCO M. GONZALEZ-LONGATT, *Sistemas de Potencia II*, Curso ELC 30524, 2007.
- [19] CURSO DE REDES ELÉCTRICAS, FACULTAD DE INGENIERÍA, UDELAR, *Capítulo 11: Cortocircuitos*. [https://eva.fing.edu.uy/pluginfile.php/77638/mod\\_folder/content/0/11-cortocircuitos.pdf?forcedownload=1](https://eva.fing.edu.uy/pluginfile.php/77638/mod_folder/content/0/11-cortocircuitos.pdf?forcedownload=1)
- [20] DAVID E. GOLDBERG, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 21th Printing, August 2000.
- [21] CURSO DE MÁQUINAS ELÉCTRICAS, FACULTAD DE INGENIERÍA, UDELAR, *Capítulo 22: Máquina Síncrona en Régimen Lineal y Capítulo 24: Estabilidad de MS*.

## A. Cartas GENSAL y GENROU

### GENROU

#### Round Rotor Generator Model (Quadratic Saturation)

This model is located at system bus # 201\* IBUS, machine # 1\* I, This model uses CONs starting with #\_\_\_ J, and STATEs starting with #\_\_\_ K. The machine MVA is 10\* for each of 1\* unit = 5\* MBASE. ZSORCE for this machine is 0 + j 0.2 on the above MBASE



\* See power flow data, Figure 13-4.

CONs	#	Value	Description
J		6	T'do (>0) (sec)
J+1		0.05	T''do (>0) (sec)
J+2		1	T'qo (>0) (sec)
J+3		0.05	T''qo (>0) (sec)
J+4		3	Inertia, H
J+5		0	Speed damping, D
J+6		1.4	X <sub>d</sub>
J+7		1.35	X <sub>q</sub>
J+8		0.3	X' <sub>d</sub>
J+9		0.6	X' <sub>q</sub>
J+10		0.2	X'' <sub>d</sub> = X'' <sub>q</sub>
J+11		0.1	X <sub>l</sub>
J+12		0.03	S(1.0)
J+13		0.4	S(1.2)

STATEs	#	Description
K		E'q
K+1		E'd
K+2		ψkd
K+3		ψkq
K+4		Δ speed (pu)
K+5		Angle (radians)

X<sub>d</sub>, X<sub>q</sub>, X'<sub>d</sub>, X'<sub>q</sub>, X''<sub>d</sub>, X''<sub>q</sub>, X<sub>l</sub>, H, and D are in pu, machine MVA base.

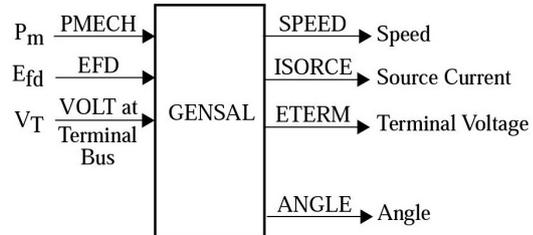
X''<sub>q</sub> must be equal to X''<sub>d</sub>.

IBUS, 'GENROU', I, T'do, T''do, T'qo, T''qo, H, D, X<sub>d</sub>, X<sub>q</sub>, X'<sub>d</sub>, X'<sub>q</sub>, X''<sub>d</sub>, X<sub>l</sub>, S(1.0), S(1.2)/

### GENSAL

#### Salient Pole Generator Model (Quadratic Saturation on d-Axis)

This model is located at system bus # 100\* IBUS, machine # 1\* I. This model uses CONs starting with # \_\_\_ J, and STATEs starting with # \_\_\_ K. The machine MVA is 5\* for each of 1\* units = 5\* MBASE. ZSORCE for this machine is  $0 + j 0.25*$  on the above MBASE.



\* See power flow data, Figure 13-4.

CONs	#	Value	Description
J		5	$T'_{do}$ (>0) (sec)
J+1		0.05	$T''_{do}$ (>0) (sec)
J+2		0.06	$T''_{qo}$ (>0) (sec)
J+3		5.084	Inertia, H
J+4		1	Speed damping, D
J+5		1.5	$X_d$
J+6		1.2	$X_q$
J+7		0.4	$X'_d$
J+8		0.25	$X''_d = X''_q$
J+9		0.12	$X_l$
J+10		0.03	S(1.0)
J+11		0.25	S(1.2)

STATEs	#	Description
K		$E'_q$
K+1		$\psi''_q$
K+2		$\psi_{kd}$
K+3		$\Delta$ speed (pu)
K+4		Angle (radians)

$X_d$ ,  $X_q$ ,  $X'_d$ ,  $X''_d$ ,  $X''_q$ ,  $X_l$ , H, and D are in pu, machine MVA base.

$X''_q$  must be equal to  $X''_d$ .

IBUS, 'GENSAL', I,  $T'_{do}$ ,  $T''_{do}$ ,  $T''_{qo}$ , H, D,  $X_d$ ,  $X_q$ ,  $X'_d$ ,  $X''_d$ ,  $X_l$ , S(1.0), S(1.2)/

## B. Códigos

### B.1. Código de programa principal

---

```
#programa principal
import os, sys
import string
PYTHONPATH = r'C:\Program Files (x86)\PTI\PSSE32\PSSBIN'
MODELFOLDER = r'D:\Proyecto'
sys.path.append(PYTHONPATH)
os.environ['PATH'] += ';' + PYTHONPATH
import psspy
import redirect
import dyntools
redirect.psse2py()
psspy.psseinit(12000)
import random
import math

time = 4.0
num = 500 #La cantidad de individuos que habra en la poblacion
generaciones = 50 #(arranca sim 1/2/17 15:50)
largo = 24 + 30 #La longitud del material genetico de cada individuo
mutacion_chance = 0.2 #La probabilidad de que un individuo mute
contador_gen = 1
mejores_gen = []
num_gen = []
fitness_gen = []
T1d0min= 1.0 #1
T1d0max= 10.0 #10
T2d0min= 0.03 #4xpaso
T2d0max= 0.20 #0.2
T1q0min= 0.2 #0.2
T1q0max= 1.5 #1.5
T2q0min= 0.03 #4xpaso
T2q0max= 0.20 #0.2
Hmin= 1 #1
Hmax= 40 #10
Dmin= 0.0 #0
Dmax= 40 #3
Xadmin= 1.5
Xdmax= 15 #2.5
Xqmin= 0.71
Xqmax= 1.5
X1dmin= 0.26
X1dmax= 0.35
X1qmin= 0.40
X1qmax= 0.60
```

```

X2dqmin= 0.007
X2dqmax= 0.010
Xlmin= 0.001
Xlmax= 0.007
S1min= 0.03 #0
S1max= 0.05 #S2min
S2min= 0.2 #S1max
S2max= 0.9
rmin= 0.0 #
rmax= 0.01 #
xmin= 0.0 #
xmax= 0.01 #
bmin= 0.0 #
bmax= 1.0 #
fact_fit = 1
salida = 'frecuencia' #frecuencia / tension / angulo / angulo_maq
barra_salida = 90300
titulo = 'Cortocircuito 3F en barra PA5 (90500)' #titulo para la grafica
titulo_eje = 'Frecuencia en barra MI5 (90300) '

#MA5=90100
#PA5=90500
#PAL=92500
#PTI_maq=98171 o 98175
#MI5=90300
#PAL_maq=98511

def falla():
# Falta Trifasica en PA5
    psspy.dist_bus_fault(ibus=90500)
# defino tiempo de la falta (3 ciclos luego del primer segundo)
    time = 1 + 3.0/50.0
    psspy.run(tpause=time)
# despejo falta
    psspy.dist_clear_fault()
    psspy.dist_branch_trip(ibus=90500, jbus=90100, id='1')

def falla2():
# Falta Monofasica en PA5
    psspy.run(0, 1.0, 10, 10, 10)
    psspy.dist_scmu_fault([0, 0, 1, 90500], [0.0, 0.0, 0.0, 0.0])

# abre 1 fase durante 800ms
    psspy.run(0, 1.1, 10, 10, 10)
# despeja falta
    psspy.dist_clear_fault(1)
# abre una fase (para psse abrir una fase es una falta)
    psspy.dist_spcb_fault(90100, 90500, r""1"", [1, 0, 1, 1, 0, 0], [0.0, 0.0, 0.0,
        0.0, 0.0])

```

```

# cierra fase y reengancha sobre falta
    psspy.run(0, 1.9, 10, 10, 10)
# cierra fase (para psse volver a cerrar la fase es despejar la falta de una fase
abierta)
    psspy.dist_clear_fault(1)
# aplico otra vez el cortocircuito (reengancha sobre falta, es decir es un reenganche
no exitoso)
    psspy.dist_scmu_fault([0, 0, 1, 90500], [0.0, 0.0, 0.0, 0.0])

# apertura trifasica definitiva a los 100ms
    t_apertura=2.0
    psspy.run(0, t_apertura, 10, 10, 10)
# despeja el cortocircuito
    psspy.dist_clear_fault(1)
# sale de servicio toda la linea (apertura trifasica definitiva)
    psspy.dist_branch_trip(90100, 90500, r""1"")

def sim(sim, T1d0_1, T2d0_1, T2q0_1, H_1, D_1, Xd_1, Xq_1, X1d_1, X2dq_1, Xl_1,
    S1_1, S2_1, T1d0_2, T2d0_2, T2q0_2, H_2, D_2, Xd_2, Xq_2, X1d_2, X2dq_2, Xl_2,
    S1_2, S2_2, r_1, x_1, b_1, r_2, x_2, b_2, r_3, x_3, b_3, r_4, x_4, b_4, r_5,
    x_5, b_5, r_6, x_6, b_6, r_7, x_7, b_7, r_8, x_8, b_8, r_9, x_9, b_9, r_10,
    x_10, b_10, tiempo):
    psspy.psseinit(12000)
# Simulo equivalente NY
    if sim=='ny':
# Modelo a utilizar:
        CASE = r"D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\M1_Dyn_1_eq.sav"
        psspy.case(CASE)
# Redirijo de PSSE a Python:
        psspy.progress_output(2,
            "D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente.txt", [0,
            0])
        psspy.report_output(2,
            r""D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente_r.txt"",
            [0, 0])
# Convierto cargas en 3 pasos
        psspy.conl(0, 1, 1, [0, 0], [0.0, 70.0, 0.0, 100.0])
        psspy.conl(0, 1, 2, [0, 0], [0.0, 70.0, 0.0, 100.0])
        psspy.conl(0, 1, 3, [0, 0], [0.0, 70.0, 0.0, 100.0])
# Convierto generadores
        psspy.cong()
# Resuelvo sistema dinamico
        psspy.ordr()
        psspy.fact()
        psspy.tysl()
        psspy.lines_per_page_one_device(1, 60)

```

```

    psspy.dyre_new([1, 1, 1, 1],
        r"D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\gyr_14_v32_2_eq.dyr",
        "", "", "")
# Cambio paso de simulacion
    _i=psspy.getdefaultint()
    _f=psspy.getdefaultreal()
    psspy.dynamics_solution_param_2([_i, _i, _i, _i, _i, _i, _i, _i], [_f, _f,
        0.001, _f, _f, _f, _f, _f])
# Guardo el caso
    #case_root = os.path.splitext(CASE)[0]
    #psspy.save(case_root + "_C.sav")
# Canales de salida
    if salida=='frecuencia':
        psspy.bus_frequency_channel([1, barra_salida], '')
    if salida=='tension':
        psspy.voltage_channel([1, -1, -1, barra_salida], '')
    if salida=='angulo_maq':
        psspy.machine_array_channel([1, 1, barra_salida], '1', '')
    if salida=='angulo':
        psspy.voltage_and_angle_channel([1, -1, -1, barra_salida], '')
# Corro por 1 seg sin falla
    psspy.strt(outfile="D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente.out")
    psspy.run(tpause=1)
# Llamo a falla()
    falla()
# Corro la falta y deajo corriendo
    time = tiempo
    psspy.run(tpause=time)
    psspy.close_powerflow()
    psspy.pssehalt_2
# Defino datos de salida
    outfile =
        dyntools.CHNF('D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente.out')
    short_title, chanid_dict, chandata_dict = outfile.get_data()
# Simulo sistema original
    if sim=='orig':
# Modelo a utilizar:
        CASE = r"D:\Proyecto\CasosPSSEpreliminares\sis_comp\M1.sav"
        psspy.case(CASE)
# Redirijo de PSSE a Python:
    psspy.progress_output(2,
        "D:\Proyecto\CasosPSSEpreliminares\sis_comp\salida.txt", [0, 0])
    psspy.report_output(2,
        r"D:\Proyecto\CasosPSSEpreliminares\sis_comp\salida_r.txt", [0, 0])
# Convierto cargas en 3 pasos
    psspy.conl(0, 1, 1, [0, 0], [0.0, 70.0, 0.0, 100.0])
    psspy.conl(0, 1, 2, [0, 0], [0.0, 70.0, 0.0, 100.0])
    psspy.conl(0, 1, 3, [0, 0], [0.0, 70.0, 0.0, 100.0])
# Convierto generadores

```

```

        psspy.cong(0)
# Resuelvo sistema dinamico
    psspy.ordr()
    psspy.fact()
    psspy.tysl()
    psspy.lines_per_page_one_device(1, 60)
    psspy.dyre_new([1, 1, 1, 1],
        r"D:\Proyecto\CasosPSSEpreliminares\sis_comp\gyr_14_v32_2.dyr", "",
        "", "")
# Cambio paso de simulacion
    _i=psspy.getdefaultint()
    _f=psspy.getdefaultreal()
    psspy.dynamics_solution_param_2([_i, _i, _i, _i, _i, _i, _i, _i], [_f, _f,
        0.001, _f, _f, _f, _f, _f])
# Guardo el caso
    #case_root = os.path.splitext(CASE)[0]
    #psspy.save(case_root + "_C.sav")
# Canales de salida
    if salida=='frecuencia':
        psspy.bus_frequency_channel([1, barra_salida], '')
    if salida=='tension':
        psspy.voltage_channel([1, -1, -1, barra_salida], '')
    if salida=='angulo_maq':
        psspy.machine_array_channel([1, 1, barra_salida], '1', '')
    if salida=='angulo':
        psspy.voltage_and_angle_channel([1, -1, -1, barra_salida], '')
# Corro por 1 seg sin falla
    psspy.strt(outfile="D:\Proyecto\CasosPSSEpreliminares\sis_comp\salida1.out")
    psspy.run(tpause=1)
# Llamo a falla()
    falla()
# Corro la falta y deajo corriendo
    time = tiempo
    psspy.run(tpause=time)
    psspy.close_powerflow()
    psspy.pssehalt_2
# Defino datos de salida
    outfile =
        dyntools.CHNF('D:\Proyecto\CasosPSSEpreliminares\sis_comp\salida1.out')
    short_title, chanid_dict, chandata_dict = outfile.get_data()
# Simulo equivalente
    if sim=='eq':
# Modelo a utilizar:
        CASE = r"D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\M1_Dyn_1_eq.sav"
        psspy.case(CASE)
# Ingreso las constantes de las lineas
    _i=psspy.getdefaultint()
    _f=psspy.getdefaultreal()

```

```

psspy.branch_data(4000, 106000, r""1"", [_i, _i, _i, _i, _i, _i], [r_1, x_1,
    b_1, _f, _f, _f, _f, _f, _f, _f, _f, _f, _f])
psspy.branch_data(4010, 4000, r""1"", [_i, _i, _i, _i, _i, _i], [r_2, x_2,
    b_2, _f, _f, _f, _f, _f, _f, _f, _f, _f])
psspy.branch_data(5002, 4010, r""1"", [_i, _i, _i, _i, _i, _i], [r_3, x_3,
    b_3, _f, _f, _f, _f, _f, _f, _f, _f, _f])
psspy.branch_data(4004, 5002, r""1"", [_i, _i, _i, _i, _i, _i], [r_4, x_4,
    b_4, _f, _f, _f, _f, _f, _f, _f, _f, _f])
psspy.branch_data(3004, 4004, r""1"", [_i, _i, _i, _i, _i, _i], [r_5, x_5,
    b_5, _f, _f, _f, _f, _f, _f, _f, _f, _f])
psspy.branch_data(3000, 3004, r""1"", [_i, _i, _i, _i, _i, _i], [r_6, x_6,
    b_6, _f, _f, _f, _f, _f, _f, _f, _f, _f])
psspy.branch_data(3004, 2018, r""1"", [_i, _i, _i, _i, _i, _i], [r_7, x_7,
    b_7, _f, _f, _f, _f, _f, _f, _f, _f, _f])
psspy.branch_data(4000, 2018, r""1"", [_i, _i, _i, _i, _i, _i], [r_8, x_8,
    b_8, _f, _f, _f, _f, _f, _f, _f, _f, _f])
psspy.branch_data(4010, 4008, r""1"", [_i, _i, _i, _i, _i, _i], [r_9, x_9,
    b_9, _f, _f, _f, _f, _f, _f, _f, _f, _f])
psspy.branch_data(5002, 4008, r""1"", [_i, _i, _i, _i, _i, _i], [r_10, x_10,
    b_10, _f, _f, _f, _f, _f, _f, _f, _f, _f])
# Redirijo de PSSE a Python:
psspy.progress_output(2,
    "D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente.txt", [0,
    0])
psspy.report_output(2,
    r""D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente_r.txt"",
    [0, 0])
# Convierto cargas en 3 pasos
psspy.conl(0, 1, 1, [0, 0], [0.0, 70.0, 0.0, 100.0])
psspy.conl(0, 1, 2, [0, 0], [0.0, 70.0, 0.0, 100.0])
psspy.conl(0, 1, 3, [0, 0], [0.0, 70.0, 0.0, 100.0])
# Convierto generadores
psspy.cong(0)
# Resuelvo sistema dinamico
psspy.ordr()
psspy.ordr()
psspy.fact()
psspy.tysl()
psspy.lines_per_page_one_device(1, 60)
psspy.dyre_new([1, 1, 1, 1],
    r""D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\gyr_14_v32_2_eq.dyr"",
    "", "", "")
# Cambio paso de simulacion
_i=psspy.getdefaultint()
_f=psspy.getdefaultreal()
psspy.dynamics_solution_param_2([_i, _i, _i, _i, _i, _i, _i, _i], [_f, _f,
    0.001, _f, _f, _f, _f])
# Guardo el caso
#case_root = os.path.splitext(CASE)[0]

```

```

#psspy.save(case_root + "_C.sav")
# Canales de salida
  if salida=='frecuencia':
    psspy.bus_frequency_channel([1, barra_salida], '')
  if salida=='tension':
    psspy.voltage_channel([1, -1, -1, barra_salida], '')
  if salida=='angulo_maq':
    psspy.machine_array_channel([1, 1, barra_salida], '1', '')
  if salida=='angulo':
    psspy.voltage_and_angle_channel([1, -1, -1, barra_salida], '')
# Ingreso las constantes de las maquinas
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 1, T1d0_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 2, T2d0_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 3, T2q0_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 4, H_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 5, D_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 6, Xd_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 7, Xq_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 8, X1d_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 9, X2dq_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 10, X1_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 11, S1_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 12, S2_1)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 1, T1d0_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 2, T2d0_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 3, T2q0_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 4, H_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 5, D_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 6, Xd_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 7, Xq_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 8, X1d_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 9, X2dq_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 10, X1_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 11, S1_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 12, S2_2)
# Corro por 1 seg sin falla
  psspy.strt(1,
    outfile="D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente.out")
  psspy.run(tpause=1)
# Llamo a falla()
  falla()
# Corro la falta y deajo corriendo
  time = tiempo
  psspy.run(tpause=time)
  psspy.close_powerflow()
  psspy.close_report()
  psspy.pssehalt_2
# Defino datos de salida

```



```

        random.uniform(xmin, xmax), random.uniform(bmin, bmax), random.uniform(rmin,
        rmax), random.uniform(xmin, xmax), random.uniform(bmin, bmax)]

# Rutina generadora de poblacion
def generar_poblacion(T1dOmin, T1dOmax, T2dOmin, T2dOmax, T1qOmin, T1qOmax, T2qOmin,
    T2qOmax, Hmin, Hmax, Dmin, Dmax, Xdmin, Xdmax, Xqmin, Xqmax, X1dmin, X1dmax,
    X1qmin, X1qmax, X2dqmin, X2dqmax, Xlmin, Xlmax, S1min, S1max, S2min, S2max,
    rmin, rmax, xmin, xmax, bmin, bmax):
    return [individuo(T1dOmin, T1dOmax, T2dOmin, T2dOmax, T1qOmin, T1qOmax, T2qOmin,
        T2qOmax, Hmin, Hmax, Dmin, Dmax, Xdmin, Xdmax, Xqmin, Xqmax, X1dmin, X1dmax,
        X1qmin, X1qmax, X2dqmin, X2dqmax, Xlmin, Xlmax, S1min, S1max, S2min, S2max,
        rmin, rmax, xmin, xmax, bmin, bmax) for i in range(num)]

# Rutina de calculo de fitness
def calcular_fitness(individuo):
    fitness = 0
    fit = 0
    fit2 = 0
# Simulo la falla para el individuo
    simulacion = sim('eq', individuo[0], individuo[1], individuo[2], individuo[3],
        individuo[4], individuo[5], individuo[6], individuo[7], individuo[8],
        individuo[9], individuo[10], individuo[11], individuo[12], individuo[13],
        individuo[14], individuo[15], individuo[16], individuo[17], individuo[18],
        individuo[19], individuo[20], individuo[21], individuo[22], individuo[23],
        individuo[24], individuo[25], individuo[26], individuo[27], individuo[28],
        individuo[29], individuo[30], individuo[31], individuo[32], individuo[33],
        individuo[34], individuo[35], individuo[36], individuo[37], individuo[38],
        individuo[39], individuo[40], individuo[41], individuo[42], individuo[43],
        individuo[44], individuo[45], individuo[46], individuo[47], individuo[48],
        individuo[49], individuo[50], individuo[51], individuo[52], individuo[53],
        time)
# Elijo el canal correspondiente cuando se mide el angulo
    if salida=='angulo':
        simulacion = simulacion[2]
    else:
        simulacion = simulacion[1]
# Sumo los errores punto a punto
    for i in range(len(simulacion)):
        fit += (abs(simulacion[i] - simulacion[0] - modelo[i] + modelo[0]))
        fit2 += simulacion[i]
# Calculo el fitness
    fitness = abs(math.exp(-abs(fact_fit*fit/fit2)))
    return fitness

# Rutina de seleccion y reproduccion de poblacion
def seleccion_y_reproduccion(poblacion):

```

```

    poblacion_nueva = []
# Calcula el fitness de cada individuo, y lo guarda en pares ordenados de la forma (
    , [])
    puntuados = [ (calcular_fitness(i), i) for i in poblacion]
# Ordeno la poblacion por fitness
    puntuados = (sorted(puntuados), puntuados[0])[0]
    for j in range(len(poblacion)):
        poblacion_nueva.append(puntuados[j][1])
    #print poblacion_nueva
    #print puntuados
    mayor = puntuados[len(poblacion)-1][0]
    mejor = puntuados[len(poblacion)-1]
# Guardo el mejor de cada generacion
    mejores_gen.append(mejor)
    print mayor
    #print mejor

# Sumo los fitness para el denominador de la probabilidad
    total_fitness = 0
    for k in range(len(poblacion)):
        total_fitness=total_fitness + puntuados[k][0]

# Armo matriz de probabilidades
    vector_prob = []
    fin = 0
    for l in range(len(poblacion)):
        vector_prob.append([1, round(10000*puntuados[l][0]/total_fitness, 0),
            puntuados[l][1]])
        fin = fin + round(10000*puntuados[l][0]/total_fitness, 0)
    print vector_prob

# Armo matriz de sorteo con probabilidad
    vector_sorteo = []
    for m in range(len(poblacion)):
        for n in range(int(vector_prob[m][1])):
            vector_sorteo.append([vector_prob[m][0], vector_prob[m][2]])
            n=n+1

#Se reproducen los individuos, sorteando los padres de la matriz anterior,
    reemplazando el 90% de la poblacion con menor fitness
    for i in range(int(round(0.9*len(poblacion)))):
        a = random.randint(0, fin-1)
        b = random.randint(0, fin-1)
        while vector_sorteo[a][0]==vector_sorteo[b][0]:
            b = random.randint(0, fin-1)
        padre = (vector_sorteo[a][1], vector_sorteo[b][1])
# Se elige un punto para hacer el intercambio
    punto = random.randint(0, largo-1)
# Se mezcla el material genetico de los padres en cada nuevo individuo

```

```

        poblacion_nueva[i][:punto] = padre[0][:punto]
        poblacion_nueva[i][punto:] = padre[1][punto:]
# Se reemplaza la poblacion con la nueva poblacion
    poblacion = poblacion_nueva
    return poblacion

# Rutina de mutacion de poblacion
def mutacion(poblacion, T1d0min, T1d0max, T2d0min, T2d0max, T1q0min, T1q0max,
            T2q0min, T2q0max, Hmin, Hmax, Dmin, Dmax, Xdmin, Xdmax, Xqmin, Xqmax, X1dmin,
            X1dmax, X1qmin, X1qmax, X2dqmin, X2dqmax, Xlmin, Xlmax, S1min, S1max, S2min,
            S2max, rmin, rmax, xmin, xmax, bmin, bmax):
# Cada individuo de la poblacion tiene una probabilidad de mutar
    for i in range(len(poblacion)-1):
        if random.random() <= mutacion_chance:
# Se elgie un punto al azar
            punto = random.randint(0, largo-1)
# Se reemplaza el punto por un nuevo valor aleatorio, pero dentro de cada margen
            if punto == 0 or punto == 12:
                nuevo_valor = random.uniform(T1d0min, T1d0max)
            if punto == 1 or punto == 13:
                nuevo_valor = random.uniform(T2d0min, T2d0max)
            if punto == 2 or punto == 14:
                nuevo_valor = random.uniform(T2q0min, T2q0max)
            if punto == 3 or punto == 15:
                nuevo_valor = random.uniform(Hmin, Hmax)
            if punto == 4 or punto == 16:
                nuevo_valor = random.uniform(Dmin, Dmax)
            if punto == 5 or punto == 17:
                nuevo_valor = random.uniform(Xdmin, Xdmax)
            if punto == 6 or punto == 18:
                nuevo_valor = random.uniform(Xqmin, Xqmax)
            if punto == 7 or punto == 19:
                nuevo_valor = random.uniform(X1dmin, X1dmax)
            if punto == 8 or punto == 20:
                nuevo_valor = random.uniform(X2dqmin, X2dqmax)
            if punto == 9 or punto == 21:
                nuevo_valor = random.uniform(Xlmin, Xlmax)
            if punto == 10 or punto == 22:
                nuevo_valor = random.uniform(S1min, S1max)
            if punto == 11 or punto == 23:
                nuevo_valor = random.uniform(S2min, S2max)
            if punto == 24 or punto == 27 or punto == 30 or punto == 33 or punto == 36
                or punto == 39 or punto == 42 or punto == 45 or punto == 48 or punto ==
                51:
                nuevo_valor = random.uniform(rmin, rmax)
            if punto == 25 or punto == 28 or punto == 31 or punto == 34 or punto == 37
                or punto == 40 or punto == 43 or punto == 46 or punto == 49 or punto ==
                52:

```

```

        nuevo_valor = random.uniform(xmin, xmax)
    if punto == 26 or punto == 29 or punto == 32 or punto == 35 or punto == 38
        or punto == 41 or punto == 44 or punto == 47 or punto == 50 or punto ==
        53:
        nuevo_valor = random.uniform(bmin, bmax)
# Se aplica la mutacion
    poblacion[i][punto] = nuevo_valor
    return poblacion

# Muestro el modelo
print("\n\nModelo: %s\n"%(modelo))

# Genero la poblacion inicial
poblacion = generar_poblacion(T1d0min, T1d0max, T2d0min, T2d0max, T1q0min, T1q0max,
    T2q0min, T2q0max, Hmin, Hmax, Dmin, Dmax, Xdmin, Xdmax, Xqmin, Xqmax, X1dmin,
    X1dmax, X1qmin, X1qmax, X2dqmin, X2dqmax, Xlmin, Xlmax, S1min, S1max, S2min,
    S2max, rmin, rmax, xmin, xmax, bmin, bmax)#Inicializar una poblacion
# Se muestra la poblacion inicial
print("Poblacion Inicial:\n%s"%(poblacion))

# Se evoluciona la poblacion
for i in range(0, generaciones):
    print("\nGeneracion:\n%s"%(contador_gen))
    contador_gen = contador_gen + 1
    poblacion = seleccion_y_reproduccion(poblacion)
    poblacion = mutacion(poblacion, T1d0min, T1d0max, T2d0min, T2d0max, T1q0min,
        T1q0max, T2q0min, T2q0max, Hmin, Hmax, Dmin, Dmax, Xdmin, Xdmax, Xqmin, Xqmax,
        X1dmin, X1dmax, X1qmin, X1qmax, X2dqmin, X2dqmax, Xlmin, Xlmax, S1min, S1max,
        S2min, S2max, rmin, rmax, xmin, xmax, bmin, bmax)
    print("\nMejor solucion:\n%s"%(mejores_gen))

# Se muestra la poblacion evolucionada
print("\n\nPoblacion Final:\n%s"%(poblacion))
print("\n\n")

may = mejores_gen[0][0]
mejor_sol = mejores_gen[0][1]
for j in range(generaciones):
    num_gen.append(j)
    fitness_gen.append(mejores_gen[j][0])
    if mejores_gen[j][0] > may:
        may=mejores_gen[j][0]
        mejor_sol=mejores_gen[j][1]

# Grafico resultados
import matplotlib.pyplot as plt
import numpy as np

```

```
import pylab
plt.ion()
plt.figure(1)
vector=sim('eq', mejor_sol[0], mejor_sol[1], mejor_sol[2], mejor_sol[3],
mejor_sol[4], mejor_sol[5], mejor_sol[6], mejor_sol[7], mejor_sol[8],
mejor_sol[9], mejor_sol[10], mejor_sol[11], mejor_sol[12], mejor_sol[13],
mejor_sol[14], mejor_sol[15], mejor_sol[16], mejor_sol[17], mejor_sol[18],
mejor_sol[19], mejor_sol[20], mejor_sol[21], mejor_sol[22], mejor_sol[23],
mejor_sol[24], mejor_sol[25], mejor_sol[26], mejor_sol[27], mejor_sol[28],
mejor_sol[29], mejor_sol[30], mejor_sol[31], mejor_sol[32], mejor_sol[33],
mejor_sol[34], mejor_sol[35], mejor_sol[36], mejor_sol[37], mejor_sol[38],
mejor_sol[39], mejor_sol[40], mejor_sol[41], mejor_sol[42], mejor_sol[43],
mejor_sol[44], mejor_sol[45], mejor_sol[46], mejor_sol[47], mejor_sol[48],
mejor_sol[49], mejor_sol[50], mejor_sol[51], mejor_sol[52], mejor_sol[53], time)
x=vector['time']
if salida=='angulo':
    y = vector[2]
else: y = vector[1]
plt.plot(x, modelo, color="red", linewidth=0.5, linestyle="-", label="Modelo
original")
print("\nMejor solucion:\n%s"%(y))
print mejor_sol
a=modelo[0]
b=y[0]
c=modelo_ny[0]
for i in range(0, len(modelo)):
    y[i]=y[i]+a-b
    modelo_ny[i]=modelo_ny[i]+a-c
plt.plot(x, y, color="blue", linewidth=0.5, linestyle="-", label="Resultado AG")
plt.plot(x, modelo_ny, color="green", linewidth=0.5, linestyle="-",
label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel(titulo_ejey)
#plt.legend(loc='upper right')
plt.legend(loc=1)
plt.figure(2)
plt.plot(num_gen, fitness_gen, 'g')
plt.title("Valor de Fitness de mejor individuo de cada generacion", fontsize = 12)
plt.xlabel("Generacion")
plt.ylabel("Valor de funcion de fitness")
plt.show()
pylab.show()
```

---

## B.2. Código de programa para graficar resultados

---

```
#programa principal
import os, sys
import string
PYTHONPATH = r'C:\Program Files (x86)\PTI\PSSE32\PSSBIN'
MODELFOLDER = r'D:\Proyecto'
sys.path.append(PYTHONPATH)
os.environ['PATH'] += ';' + PYTHONPATH
import psspy
import redirect
import dyntools
redirect.psse2py()
psspy.psseinit(12000)
import random
import math

#Defino tiempo de simulacion y titulo de graficas
time = 19.99
titulo = 'CC 3F en MA5 - Optimizacion de angulo de maquina en PTI'

#MA5=90100
#PA5=90500
#PAL=92500
#PTI_maq=98171 o 98175
#MI5=90300
#PAL_maq=98511

def falla():
# Falta Trifasica en PA5
    psspy.dist_bus_fault(ibus=90100)
# defino tiempo de la falta (3 ciclos luego del primer segundo)
    time = 1 + 3.0/50.0
    psspy.run(tpause=time)
# despejo falta
    psspy.dist_clear_fault()
    psspy.dist_branch_trip(ibus=90300, jbus=90100, id='1')

def falla2():
# Falta Monofasica en PA5
    psspy.run(0, 1.0, 10, 10, 10)
    psspy.dist_scmu_fault([0, 0, 1, 90500], [0.0, 0.0, 0.0, 0.0])

# abre 1 fase durante 800ms
    psspy.run(0, 1.1, 10, 10, 10)
# despeja falta
    psspy.dist_clear_fault(1)
# abre una fase (para psse abrir una fase es una falta)
```

```

    psspy.dist_spcb_fault(90100, 90500, r""1"", [1, 0, 1, 1, 0, 0], [0.0, 0.0, 0.0,
        0.0, 0.0])

# cierra fase y reengancha sobre falta
    psspy.run(0, 1.9, 10, 10, 10)
# cierra fase (para psse volver a cerrar la fase es despejar la falta de una fase
    abierta)
    psspy.dist_clear_fault(1)
# aplico otra vez el cortocircuito (reengancha sobre falta, es decir es un reenganche
    no exitoso)
    psspy.dist_scmu_fault([0, 0, 1, 90500], [0.0, 0.0, 0.0, 0.0])

# apertura trifasica definitiva a los 100ms
    t_apertura=2.0
    psspy.run(0, t_apertura, 10, 10, 10)
# despeja el cortocircuito
    psspy.dist_clear_fault(1)
# sale de servicio toda la linea (apertura trifasica definitiva)
    psspy.dist_branch_trip(90100, 90500, r""1"")

def sim(sim, T1d0_1, T2d0_1, T2q0_1, H_1, D_1, Xd_1, Xq_1, X1d_1, X2dq_1, X1_1,
    S1_1, S2_1, T1d0_2, T2d0_2, T2q0_2, H_2, D_2, Xd_2, Xq_2, X1d_2, X2dq_2, X1_2,
    S1_2, S2_2, r_1, x_1, b_1, r_2, x_2, b_2, r_3, x_3, b_3, r_4, x_4, b_4, r_5,
    x_5, b_5, r_6, x_6, b_6, r_7, x_7, b_7, r_8, x_8, b_8, r_9, x_9, b_9, r_10,
    x_10, b_10, tiempo):
    psspy.psseinit(12000)
# Simulo equivalente NY
    if sim=='ny':
# Modelo a utilizar:
        CASE = r"D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\M1_Dyn_1_eq.sav"
        psspy.case(CASE)
# Redirijo de PSSE a Python:
        psspy.progress_output(2,
            "D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente.txt", [0,
                0])
        psspy.report_output(2,
            r""D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente_r.txt"",
                [0, 0])
# Convierto cargas en 3 pasos
        psspy.conl(0, 1, 1, [0, 0], [0.0, 70.0, 0.0, 100.0])
        psspy.conl(0, 1, 2, [0, 0], [0.0, 70.0, 0.0, 100.0])
        psspy.conl(0, 1, 3, [0, 0], [0.0, 70.0, 0.0, 100.0])
# Convierto generadores
        psspy.cong()
# Resuelvo sistema dinamico
        psspy.ordr()
        psspy.fact()
        psspy.tysl()

```

```

psspy.lines_per_page_one_device(1, 60)
psspy.dyre_new([1, 1, 1, 1],
    r"D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\gyr_14_v32_2_eq.dyr",
    "", "", "")
# Cambio paso de simulacion
_i=psspy.getdefaultint()
_f=psspy.getdefaultreal()
psspy.dynamics_solution_param_2([_i, _i, _i, _i, _i, _i, _i, _i], [_f, _f,
    0.001, _f, _f, _f, _f])
# Guardo el caso
#case_root = os.path.splitext(CASE)[0]
#psspy.save(case_root + "_C.sav")
# Canales de salida
psspy.bus_frequency_channel([1, 4004], '')
psspy.bus_frequency_channel([2, 90000], '')
psspy.bus_frequency_channel([3, 90500], '')
psspy.bus_frequency_channel([4, 90160], '')
psspy.bus_frequency_channel([5, 90200], '')
psspy.bus_frequency_channel([6, 92100], '')
psspy.bus_frequency_channel([7, 90300], '')
psspy.bus_frequency_channel([8, 90700], '')
psspy.bus_frequency_channel([9, 90480], '')
psspy.voltage_channel([10, -1, -1, 4004], '')
psspy.voltage_channel([11, -1, -1, 90000], '')
psspy.voltage_channel([12, -1, -1, 90500], '')
psspy.voltage_channel([13, -1, -1, 90160], '')
psspy.voltage_channel([14, -1, -1, 90200], '')
psspy.voltage_channel([15, -1, -1, 92100], '')
psspy.voltage_channel([16, -1, -1, 90300], '')
psspy.voltage_channel([17, -1, -1, 90700], '')
psspy.voltage_channel([18, -1, -1, 90480], '')
psspy.machine_array_channel([19, 1, 98511], '1', '')
psspy.machine_array_channel([20, 1, 98551], '1', '')
psspy.machine_array_channel([21, 1, 98541], '1', '')
psspy.machine_array_channel([22, 1, 98121], '1', '')
psspy.machine_array_channel([23, 1, 98171], '1', '')
# Corro por 1 seg sin falla
psspy.strt(outfile="D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente.out")
psspy.run(tpause=1)
# Llamo a falla()
falla()
# Corro la falta y deajo corriendo
time = tiempo
psspy.run(tpause=time)
psspy.close_powerflow()
psspy.pssehalt_2
# Defino datos de salida
outfile =
    dyntools.CHNF('D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente.out')

```

```

        short_title, chanid_dict, chandata_dict = outfile.get_data()
# Simulo sistema original
    if sim=='orig':
# Modelo a utilizar:
        CASE = r"D:\Proyecto\CasosPSSEpreliminares\sis_comp\M1.sav"
        psspy.case(CASE)
# Redirijo de PSSE a Python:
        psspy.progress_output(2,
            "D:\Proyecto\CasosPSSEpreliminares\sis_comp\salida.txt", [0, 0])
        psspy.report_output(2,
            r"D:\Proyecto\CasosPSSEpreliminares\sis_comp\salida_r.txt", [0, 0])
# Convierto cargas en 3 pasos
        psspy.conl(0, 1, 1, [0, 0], [0.0, 70.0, 0.0, 100.0])
        psspy.conl(0, 1, 2, [0, 0], [0.0, 70.0, 0.0, 100.0])
        psspy.conl(0, 1, 3, [0, 0], [0.0, 70.0, 0.0, 100.0])
# Convierto generadores
        psspy.cong(0)
# Resuelvo sistema dinamico
        psspy.ordr()
        psspy.fact()
        psspy.tysl()
        psspy.lines_per_page_one_device(1, 60)
        psspy.dyre_new([1, 1, 1, 1],
            r"D:\Proyecto\CasosPSSEpreliminares\sis_comp\gyr_14_v32_2.dyr", "",
            "", "")
# Cambio paso de simulacion
        _i=psspy.getdefaultint()
        _f=psspy.getdefaultreal()
        psspy.dynamics_solution_param_2([_i, _i, _i, _i, _i, _i, _i, _i], [_f, _f,
            0.001, _f, _f, _f, _f, _f])
# Guardo el caso
        #case_root = os.path.splitext(CASE)[0]
        #psspy.save(case_root + "_C.sav")
# Canales de salida
        psspy.bus_frequency_channel([1, 4004], '')
        psspy.bus_frequency_channel([2, 90000], '')
        psspy.bus_frequency_channel([3, 90500], '')
        psspy.bus_frequency_channel([4, 90160], '')
        psspy.bus_frequency_channel([5, 90200], '')
        psspy.bus_frequency_channel([6, 92100], '')
        psspy.bus_frequency_channel([7, 90300], '')
        psspy.bus_frequency_channel([8, 90700], '')
        psspy.bus_frequency_channel([9, 90480], '')
        psspy.voltage_channel([10, -1, -1, 4004], '')
        psspy.voltage_channel([11, -1, -1, 90000], '')
        psspy.voltage_channel([12, -1, -1, 90500], '')
        psspy.voltage_channel([13, -1, -1, 90160], '')
        psspy.voltage_channel([14, -1, -1, 90200], '')
        psspy.voltage_channel([15, -1, -1, 92100], '')

```

```

psspy.voltage_channel([16, -1, -1, 90300], '')
psspy.voltage_channel([17, -1, -1, 90700], '')
psspy.voltage_channel([18, -1, -1, 90480], '')
psspy.machine_array_channel([19, 1, 98511], '1', '')
psspy.machine_array_channel([20, 1, 98551], '1', '')
psspy.machine_array_channel([21, 1, 98541], '1', '')
psspy.machine_array_channel([22, 1, 98121], '1', '')
psspy.machine_array_channel([23, 1, 98171], '1', '')
# Corro por 1 seg sin falla
psspy.strt(outfile="D:\Proyecto\CasosPSSEpreliminares\sis_comp\salida1.out")
psspy.run(tpause=1)
# Llamo a falla()
falla()
# Corro la falta y deajo corriendo
time = tiempo
psspy.run(tpause=time)
psspy.close_powerflow()
psspy.pssehalt_2
# Defino datos de salida
outfile =
    dyntools.CHNF('D:\Proyecto\CasosPSSEpreliminares\sis_comp\salida1.out')
short_title, chanid_dict, chandata_dict = outfile.get_data()
# Simulo equivalente
if sim=='eq':
# Modelo a utilizar:
CASE = r"D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\M1_Dyn_1_eq.sav"
psspy.case(CASE)
# Ingreso las constantes de las lineas
_i=psspy.getdefaultint()
_f=psspy.getdefaultreal()
psspy.branch_data(4000, 106000, r""1"", [_i, _i, _i, _i, _i, _i], [r_1, x_1,
    b_1, _f, _f])
psspy.branch_data(4010, 4000, r""1"", [_i, _i, _i, _i, _i, _i], [r_2, x_2,
    b_2, _f, _f])
psspy.branch_data(5002, 4010, r""1"", [_i, _i, _i, _i, _i, _i], [r_3, x_3,
    b_3, _f, _f])
psspy.branch_data(4004, 5002, r""1"", [_i, _i, _i, _i, _i, _i], [r_4, x_4,
    b_4, _f, _f])
psspy.branch_data(3004, 4004, r""1"", [_i, _i, _i, _i, _i, _i], [r_5, x_5,
    b_5, _f, _f])
psspy.branch_data(3000, 3004, r""1"", [_i, _i, _i, _i, _i, _i], [r_6, x_6,
    b_6, _f, _f])
psspy.branch_data(3004, 2018, r""1"", [_i, _i, _i, _i, _i, _i], [r_7, x_7,
    b_7, _f, _f])
psspy.branch_data(4000, 2018, r""1"", [_i, _i, _i, _i, _i, _i], [r_8, x_8,
    b_8, _f, _f])
psspy.branch_data(4010, 4008, r""1"", [_i, _i, _i, _i, _i, _i], [r_9, x_9,
    b_9, _f, _f])

```

```

psspy.branch_data(5002, 4008, r"""1""", [_i, _i, _i, _i, _i, _i], [r_10, x_10,
    b_10, _f, _f])
# Redirijo de PSSE a Python:
psspy.progress_output(2,
    "D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente.txt", [0,
    0])
psspy.report_output(2,
    r"""D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente_r.txt""",
    [0, 0])
# Convierto cargas en 3 pasos
psspy.conl(0, 1, 1, [0, 0], [0.0, 70.0, 0.0, 100.0])
psspy.conl(0, 1, 2, [0, 0], [0.0, 70.0, 0.0, 100.0])
psspy.conl(0, 1, 3, [0, 0], [0.0, 70.0, 0.0, 100.0])
# Convierto generadores
psspy.cong()
# Resuelvo sistema dinamico
psspy.ordr()
psspy.fact()
psspy.tysl()
psspy.lines_per_page_one_device(1, 60)
psspy.dyre_new([1, 1, 1, 1],
    r"""D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\gyr_14_v32_2_eq.dyr""",
    "", "", "")
# Cambio paso de simulacion
_i=psspy.getdefaultint()
_f=psspy.getdefaultreal()
psspy.dynamics_solution_param_2([_i, _i, _i, _i, _i, _i, _i, _i], [_f, _f,
    0.001, _f, _f, _f, _f, _f])
# Guardo el caso
#case_root = os.path.splitext(CASE)[0]
#psspy.save(case_root + "_C.sav")
# Canales de salida
psspy.bus_frequency_channel([1, 4004], '')
psspy.bus_frequency_channel([2, 90000], '')
psspy.bus_frequency_channel([3, 90500], '')
psspy.bus_frequency_channel([4, 90160], '')
psspy.bus_frequency_channel([5, 90200], '')
psspy.bus_frequency_channel([6, 92100], '')
psspy.bus_frequency_channel([7, 90300], '')
psspy.bus_frequency_channel([8, 90700], '')
psspy.bus_frequency_channel([9, 90480], '')
psspy.voltage_channel([10, -1, -1, 4004], '')
psspy.voltage_channel([11, -1, -1, 90000], '')
psspy.voltage_channel([12, -1, -1, 90500], '')
psspy.voltage_channel([13, -1, -1, 90160], '')
psspy.voltage_channel([14, -1, -1, 90200], '')
psspy.voltage_channel([15, -1, -1, 92100], '')
psspy.voltage_channel([16, -1, -1, 90300], '')
psspy.voltage_channel([17, -1, -1, 90700], '')

```

```

psspy.voltage_channel([18, -1, -1, 90480], '')
psspy.machine_array_channel([19, 1, 98511], '1', '')
psspy.machine_array_channel([20, 1, 98551], '1', '')
psspy.machine_array_channel([21, 1, 98541], '1', '')
psspy.machine_array_channel([22, 1, 98121], '1', '')
psspy.machine_array_channel([23, 1, 98171], '1', '')
# Ingreso las constantes de las maquinas
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 1, T1d0_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 2, T2d0_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 3, T2q0_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 4, H_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 5, D_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 6, Xd_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 7, Xq_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 8, X1d_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 9, X2dq_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 10, X1_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 11, S1_1)
psspy.change_plmod_con(106000, r""1"", r""GENSAL"", 12, S2_1)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 1, T1d0_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 2, T2d0_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 3, T2q0_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 4, H_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 5, D_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 6, Xd_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 7, Xq_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 8, X1d_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 9, X2dq_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 10, X1_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 11, S1_2)
psspy.change_plmod_con(3000, r""1"", r""GENSAL"", 12, S2_2)
# Corro por 1 seg sin falla
psspy.strt(1,
    outfile="D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente.out")
psspy.run(tpause=1)
# Llamo a falla()
falla()
# Corro la falta y deajo corriendo
time = tiempo
psspy.run(tpause=time)
psspy.close_powerflow()
psspy.close_report()
psspy.pssehalt_2
# Defino datos de salida
outfile =
    dyntools.CHNF("D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\salida_equivalente.out")
short_title, chanid_dict, chandata_dict = outfile.get_data()
return chandata_dict

```



```
plt.plot(x, vector_eq[3], color="blue", linewidth=grosor_linea, linestyle="-",
        label="Resultado AG")
plt.plot(x, vector_ny[3], color="green", linewidth=grosor_linea, linestyle="-",
        label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Desvio de frecuencia en barra PA5 (90500) (pu)")
plt.legend(loc='upper right')
plt.figure(4)
plt.plot(x, vector_orig[4], color="red", linewidth=grosor_linea, linestyle="-",
        label="Modelo original")
plt.plot(x, vector_eq[4], color="blue", linewidth=grosor_linea, linestyle="-",
        label="Resultado AG")
plt.plot(x, vector_ny[4], color="green", linewidth=grosor_linea, linestyle="-",
        label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Desvio de frecuencia en barra BR5 (90160) (pu)")
plt.legend(loc='upper right')
plt.figure(5)
plt.plot(x, vector_orig[5], color="red", linewidth=grosor_linea, linestyle="-",
        label="Modelo original")
plt.plot(x, vector_eq[5], color="blue", linewidth=grosor_linea, linestyle="-",
        label="Resultado AG")
plt.plot(x, vector_ny[5], color="green", linewidth=grosor_linea, linestyle="-",
        label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Desvio de frecuencia en barra MB5 (90200) (pu)")
plt.legend(loc='upper right')
plt.figure(6)
plt.plot(x, vector_orig[6], color="red", linewidth=grosor_linea, linestyle="-",
        label="Modelo original")
plt.plot(x, vector_eq[6], color="blue", linewidth=grosor_linea, linestyle="-",
        label="Resultado AG")
plt.plot(x, vector_ny[6], color="green", linewidth=grosor_linea, linestyle="-",
        label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Desvio de frecuencia en barra MA5 (99101) (pu)")
plt.legend(loc='upper right')
plt.figure(7)
plt.plot(x, vector_orig[7], color="red", linewidth=grosor_linea, linestyle="-",
        label="Modelo original")
plt.plot(x, vector_eq[7], color="blue", linewidth=grosor_linea, linestyle="-",
        label="Resultado AG")
plt.plot(x, vector_ny[7], color="green", linewidth=grosor_linea, linestyle="-",
        label="Equivalente NY")
plt.title(titulo, fontsize = 12)
```

```
plt.xlabel("Tiempo")
plt.ylabel("Desvio de frecuencia en barra MI5 (90300) (pu)")
plt.legend(loc='upper right')
plt.figure(8)
plt.plot(x, vector_orig[8], color="red", linewidth=grosor_linea, linestyle="-",
         label="Modelo original")
plt.plot(x, vector_eq[8], color="blue", linewidth=grosor_linea, linestyle="-",
         label="Resultado AG")
plt.plot(x, vector_ny[8], color="green", linewidth=grosor_linea, linestyle="-",
         label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Desvio de frecuencia en barra SC5 (90700) (pu)")
plt.legend(loc='upper right')
plt.figure(9)
plt.plot(x, vector_orig[9], color="red", linewidth=grosor_linea, linestyle="-",
         label="Modelo original")
plt.plot(x, vector_eq[9], color="blue", linewidth=grosor_linea, linestyle="-",
         label="Resultado AG")
plt.plot(x, vector_ny[9], color="green", linewidth=grosor_linea, linestyle="-",
         label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Desvio de frecuencia en barra CM5 (90480) (pu)")
plt.legend(loc='upper right')
plt.figure(10)
#a=vector_orig[10][0]
#b=vector_eq[10][0]
#c=vector_ny[10][0]
#for i in range(0, len(vector_orig[10])):
#    vector_eq[10][i]=vector_eq[10][i]+a-b
#    vector_ny[10][i]=vector_ny[10][i]+a-c
plt.plot(x, vector_orig[10], color="red", linewidth=grosor_linea, linestyle="-",
         label="Modelo original")
plt.plot(x, vector_eq[10], color="blue", linewidth=grosor_linea, linestyle="-",
         label="Resultado AG")
plt.plot(x, vector_ny[10], color="green", linewidth=grosor_linea, linestyle="-",
         label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Tension en barra SA5 (4008) (pu)")
plt.legend(loc='lower right')
plt.figure(11)
#a=vector_orig[11][0]
#b=vector_eq[11][0]
#c=vector_ny[11][0]
#for i in range(0, len(vector_orig[11])):
#    vector_eq[11][i]=vector_eq[11][i]+a-b
#    vector_ny[11][i]=vector_ny[11][i]+a-c
```

```

plt.plot(x, vector_orig[11], color="red", linewidth=grosor_linea, linestyle="-",
         label="Modelo original")
plt.plot(x, vector_eq[11], color="blue", linewidth=grosor_linea, linestyle="-",
         label="Resultado AG")
plt.plot(x, vector_ny[11], color="green", linewidth=grosor_linea, linestyle="-",
         label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Tension en barra SU5 (90000) (pu)")
plt.legend(loc='lower right')
plt.figure(12)
#a=vector_orig[12][0]
#b=vector_eq[12][0]
#c=vector_ny[12][0]
#for i in range(0, len(vector_orig[12])):
#    vector_eq[12][i]=vector_eq[12][i]+a-b
#    vector_ny[12][i]=vector_ny[12][i]+a-c
plt.plot(x, vector_orig[12], color="red", linewidth=grosor_linea, linestyle="-",
         label="Modelo original")
plt.plot(x, vector_eq[12], color="blue", linewidth=grosor_linea, linestyle="-",
         label="Resultado AG")
plt.plot(x, vector_ny[12], color="green", linewidth=grosor_linea, linestyle="-",
         label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Tension en barra PA5 (90500) (pu)")
plt.legend(loc='lower right')
plt.figure(13)
#a=vector_orig[13][0]
#b=vector_eq[13][0]
#c=vector_ny[13][0]
#for i in range(0, len(vector_orig[13])):
#    vector_eq[13][i]=vector_eq[13][i]+a-b
#    vector_ny[13][i]=vector_ny[13][i]+a-c
plt.plot(x, vector_orig[13], color="red", linewidth=grosor_linea, linestyle="-",
         label="Modelo original")
plt.plot(x, vector_eq[13], color="blue", linewidth=grosor_linea, linestyle="-",
         label="Resultado AG")
plt.plot(x, vector_ny[13], color="green", linewidth=grosor_linea, linestyle="-",
         label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Tension en barra BR5 (90160) (pu)")
plt.legend(loc='lower right')
plt.figure(14)
#a=vector_orig[14][0]
#b=vector_eq[14][0]
#c=vector_ny[14][0]
#for i in range(0, len(vector_orig[14])):

```

```

#   vector_eq[14][i]=vector_eq[14][i]+a-b
#   vector_ny[14][i]=vector_ny[14][i]+a-c
plt.plot(x, vector_orig[14], color="red", linewidth=grosor_linea, linestyle="-",
         label="Modelo original")
plt.plot(x, vector_eq[14], color="blue", linewidth=grosor_linea, linestyle="-",
         label="Resultado AG")
plt.plot(x, vector_ny[14], color="green", linewidth=grosor_linea, linestyle="-",
         label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Tension en barra MB5 (90200) (pu)")
plt.legend(loc='lower right')
plt.figure(15)
#a=vector_orig[15][0]
#b=vector_eq[15][0]
#c=vector_ny[15][0]
#for i in range(0, len(vector_orig[15])):
#   vector_eq[15][i]=vector_eq[15][i]+a-b
#   vector_ny[15][i]=vector_ny[15][i]+a-c
plt.plot(x, vector_orig[15], color="red", linewidth=grosor_linea, linestyle="-",
         label="Modelo original")
plt.plot(x, vector_eq[15], color="blue", linewidth=grosor_linea, linestyle="-",
         label="Resultado AG")
plt.plot(x, vector_ny[15], color="green", linewidth=grosor_linea, linestyle="-",
         label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Tension en barra MA5 (92100) (pu)")
plt.legend(loc='lower right')
plt.figure(16)
#a=vector_orig[16][0]
#b=vector_eq[16][0]
#c=vector_ny[16][0]
#for i in range(0, len(vector_orig[16])):
#   vector_eq[16][i]=vector_eq[16][i]+a-b
#   vector_ny[16][i]=vector_ny[16][i]+a-c
plt.plot(x, vector_orig[16], color="red", linewidth=grosor_linea, linestyle="-",
         label="Modelo original")
plt.plot(x, vector_eq[16], color="blue", linewidth=grosor_linea, linestyle="-",
         label="Resultado AG")
plt.plot(x, vector_ny[16], color="green", linewidth=grosor_linea, linestyle="-",
         label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Tension en barra MI5 (90300) (pu)")
plt.legend(loc='lower right')
plt.figure(17)
#a=vector_orig[17][0]
#b=vector_eq[17][0]

```

```

#c=vector_ny[17][0]
#for i in range(0, len(vector_orig[17])):
#    vector_eq[17][i]=vector_eq[17][i]+a-b
#    vector_ny[17][i]=vector_ny[17][i]+a-c
plt.plot(x, vector_orig[17], color="red", linewidth=grosor_linea, linestyle="-",
        label="Modelo original")
plt.plot(x, vector_eq[17], color="blue", linewidth=grosor_linea, linestyle="-",
        label="Resultado AG")
plt.plot(x, vector_ny[17], color="green", linewidth=grosor_linea, linestyle="-",
        label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Tension en barra SC5 (90700) (pu)")
plt.legend(loc='lower right')
plt.figure(18)
#a=vector_orig[18][0]
#b=vector_eq[18][0]
#c=vector_ny[18][0]
#for i in range(0, len(vector_orig[18])):
#    vector_eq[18][i]=vector_eq[18][i]+a-b
#    vector_ny[18][i]=vector_ny[18][i]+a-c
plt.plot(x, vector_orig[18], color="red", linewidth=grosor_linea, linestyle="-",
        label="Modelo original")
plt.plot(x, vector_eq[18], color="blue", linewidth=grosor_linea, linestyle="-",
        label="Resultado AG")
plt.plot(x, vector_ny[18], color="green", linewidth=grosor_linea, linestyle="-",
        label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Tension en barra CM5 (90480) (pu)")
plt.legend(loc='lower right')
plt.figure(19)
a=vector_orig[19][0]
b=vector_eq[19][0]
c=vector_ny[19][0]
for i in range(0, len(vector_orig[19])):
    vector_eq[19][i]=vector_eq[19][i]+a-b
    vector_ny[19][i]=vector_ny[19][i]+a-c
plt.plot(x, vector_orig[19], color="red", linewidth=grosor_linea, linestyle="-",
        label="Modelo original")
plt.plot(x, vector_eq[19], color="blue", linewidth=grosor_linea, linestyle="-",
        label="Resultado AG")
plt.plot(x, vector_ny[19], color="green", linewidth=grosor_linea, linestyle="-",
        label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Angulo en barra PAL (98511) (grados)")
plt.legend(loc='upper left')
plt.figure(20)

```

```
a=vector_orig[20][0]
b=vector_eq[20][0]
c=vector_ny[20][0]
for i in range(0, len(vector_orig[20])):
    vector_eq[20][i]=vector_eq[20][i]+a-b
    vector_ny[20][i]=vector_ny[20][i]+a-c
plt.plot(x, vector_orig[20], color="red", linewidth=grosor_linea, linestyle="-",
        label="Modelo original")
plt.plot(x, vector_eq[20], color="blue", linewidth=grosor_linea, linestyle="-",
        label="Resultado AG")
plt.plot(x, vector_ny[20], color="green", linewidth=grosor_linea, linestyle="-",
        label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Angulo en barra BAY (98551) (grados)")
plt.legend(loc='upper left')
plt.figure(21)
a=vector_orig[21][0]
b=vector_eq[21][0]
c=vector_ny[21][0]
for i in range(0, len(vector_orig[21])):
    vector_eq[21][i]=vector_eq[21][i]+a-b
    vector_ny[21][i]=vector_ny[21][i]+a-c
plt.plot(x, vector_orig[21], color="red", linewidth=grosor_linea, linestyle="-",
        label="Modelo original")
plt.plot(x, vector_eq[21], color="blue", linewidth=grosor_linea, linestyle="-",
        label="Resultado AG")
plt.plot(x, vector_ny[21], color="green", linewidth=grosor_linea, linestyle="-",
        label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Angulo en barra BON (98541) (grados)")
plt.legend(loc='upper left')
plt.figure(22)
a=vector_orig[22][0]
b=vector_eq[22][0]
c=vector_ny[22][0]
for i in range(0, len(vector_orig[22])):
    vector_eq[22][i]=vector_eq[22][i]+a-b
    vector_ny[22][i]=vector_ny[22][i]+a-c
plt.plot(x, vector_orig[22], color="red", linewidth=grosor_linea, linestyle="-",
        label="Modelo original")
plt.plot(x, vector_eq[22], color="blue", linewidth=grosor_linea, linestyle="-",
        label="Resultado AG")
plt.plot(x, vector_ny[22], color="green", linewidth=grosor_linea, linestyle="-",
        label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Angulo en barra MVL (98121) (grados)")
```

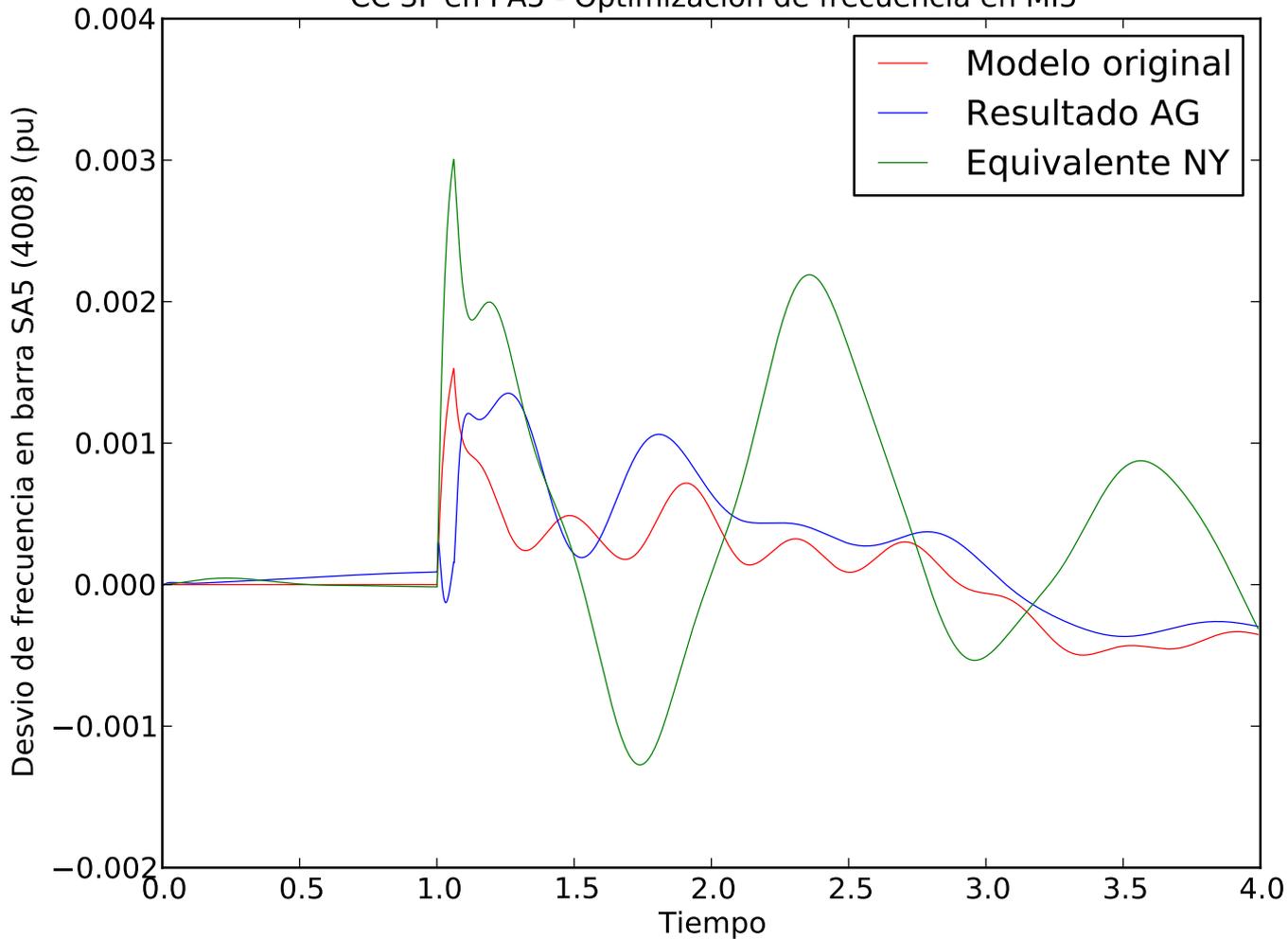
```
plt.legend(loc='upper left')
plt.figure(23)
a=vector_orig[23][0]
b=vector_eq[23][0]
c=vector_ny[23][0]
for i in range(0, len(vector_orig[23])):
    vector_eq[23][i]=vector_eq[23][i]+a-b
    vector_ny[23][i]=vector_ny[23][i]+a-c
plt.plot(x, vector_orig[23], color="red", linewidth=grosor_linea, linestyle="-",
        label="Modelo original")
plt.plot(x, vector_eq[23], color="blue", linewidth=grosor_linea, linestyle="-",
        label="Resultado AG")
plt.plot(x, vector_ny[23], color="green", linewidth=grosor_linea, linestyle="-",
        label="Equivalente NY")
plt.title(titulo, fontsize = 12)
plt.xlabel("Tiempo")
plt.ylabel("Angulo en barra PTI (98171) (grados)")
plt.legend(loc='upper left')
from matplotlib.backends.backend_pdf import PdfPages
pp = PdfPages('D:\Proyecto\CasosPSSEpreliminares\eq_din_ny\graficas.pdf')
pp.savefig(1)
pp.savefig(2)
pp.savefig(3)
pp.savefig(4)
pp.savefig(5)
pp.savefig(6)
pp.savefig(7)
pp.savefig(8)
pp.savefig(9)
pp.savefig(10)
pp.savefig(11)
pp.savefig(12)
pp.savefig(13)
pp.savefig(14)
pp.savefig(15)
pp.savefig(16)
pp.savefig(17)
pp.savefig(18)
pp.savefig(19)
pp.savefig(20)
pp.savefig(21)
#pp.savefig(22)
pp.savefig(23)
pp.close()
plt.show()
pylab.show()
plt.close()
pylab.close()
```

---

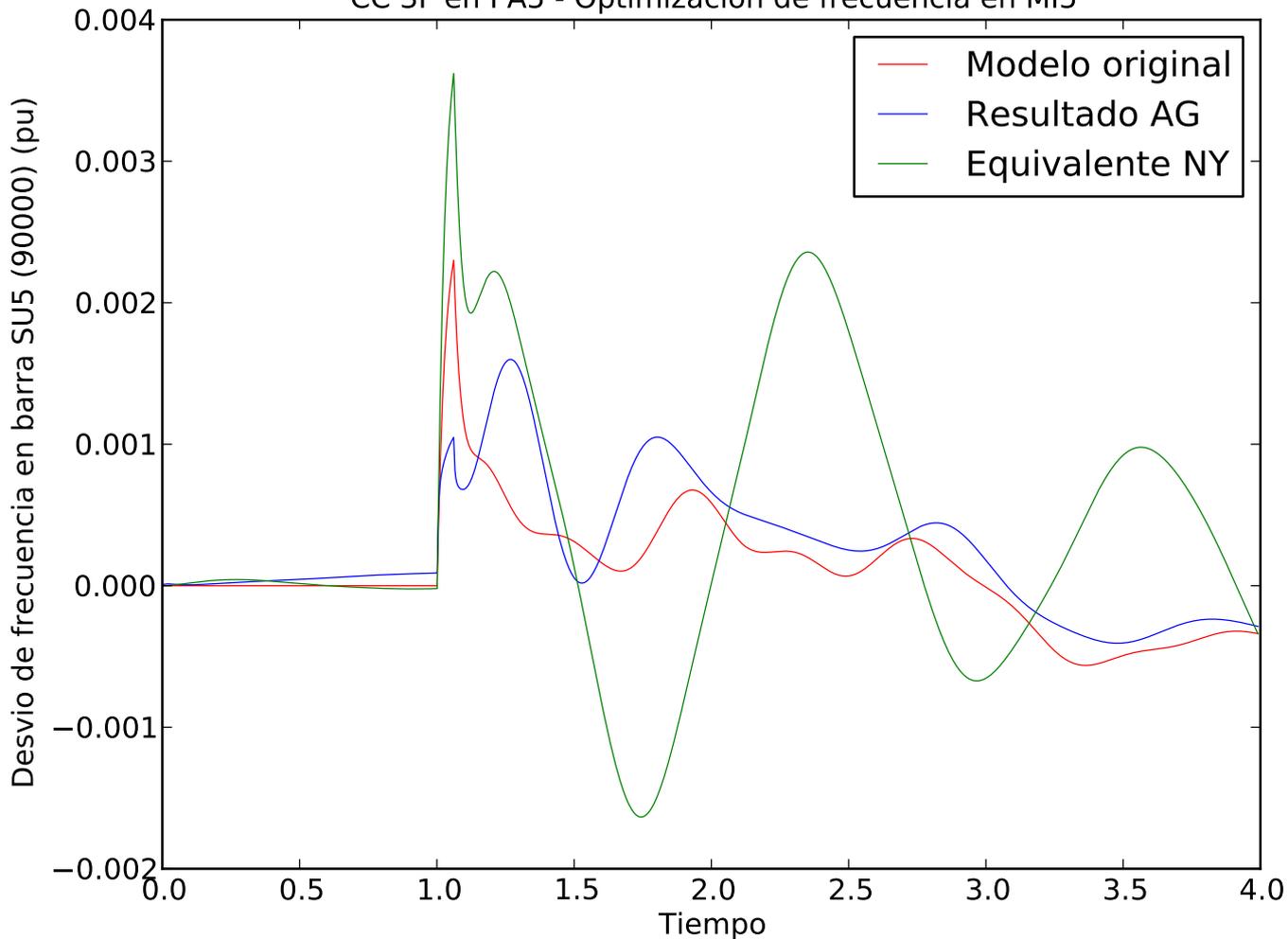
## **C. Simulaciones**

- C.1. Simulación 1: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando frecuencia en MI5 en 4 segundos. Comportamiento en 4 segundos.

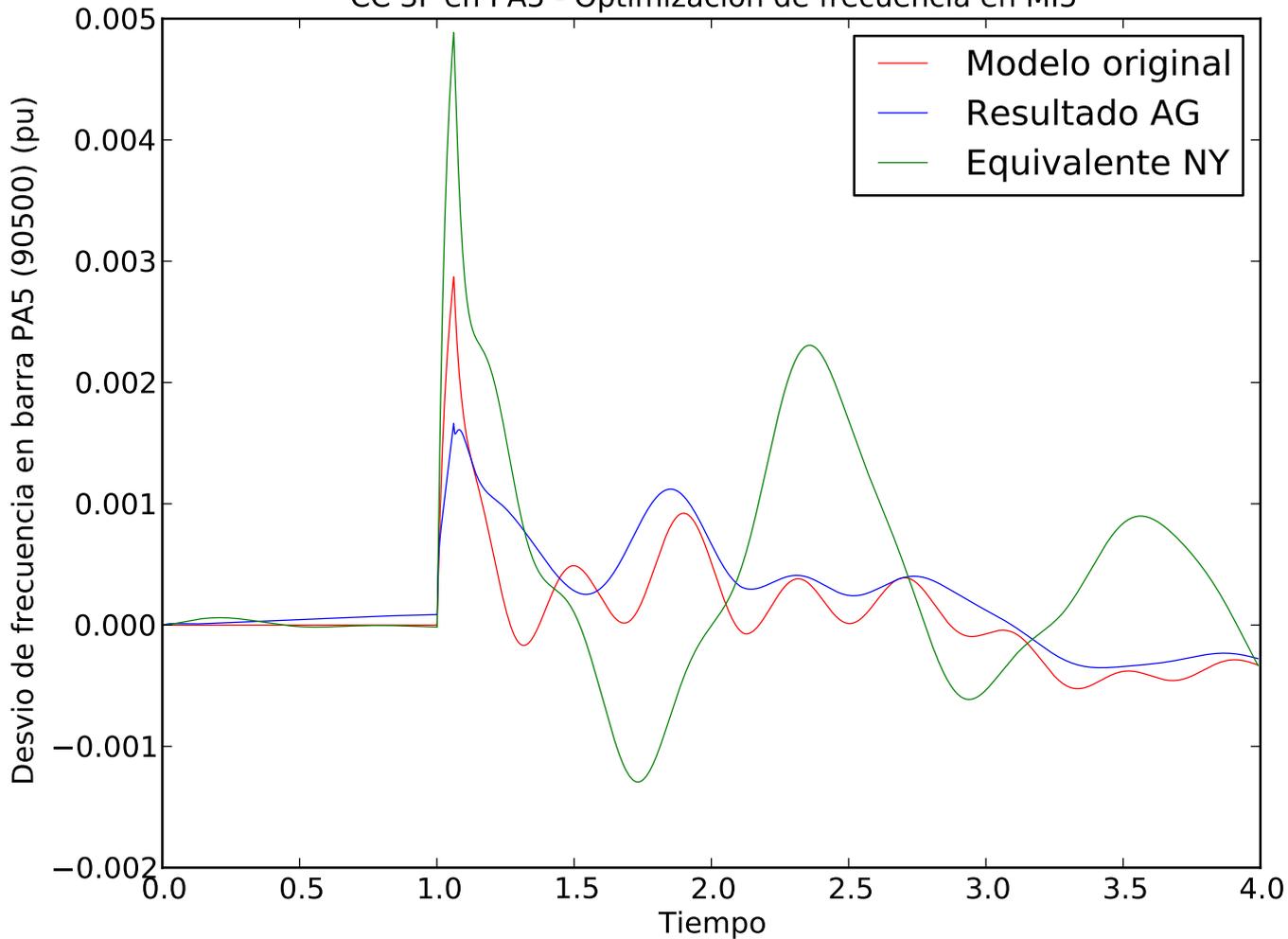
CC 3F en PA5 - Optimizacion de frecuencia en MI5



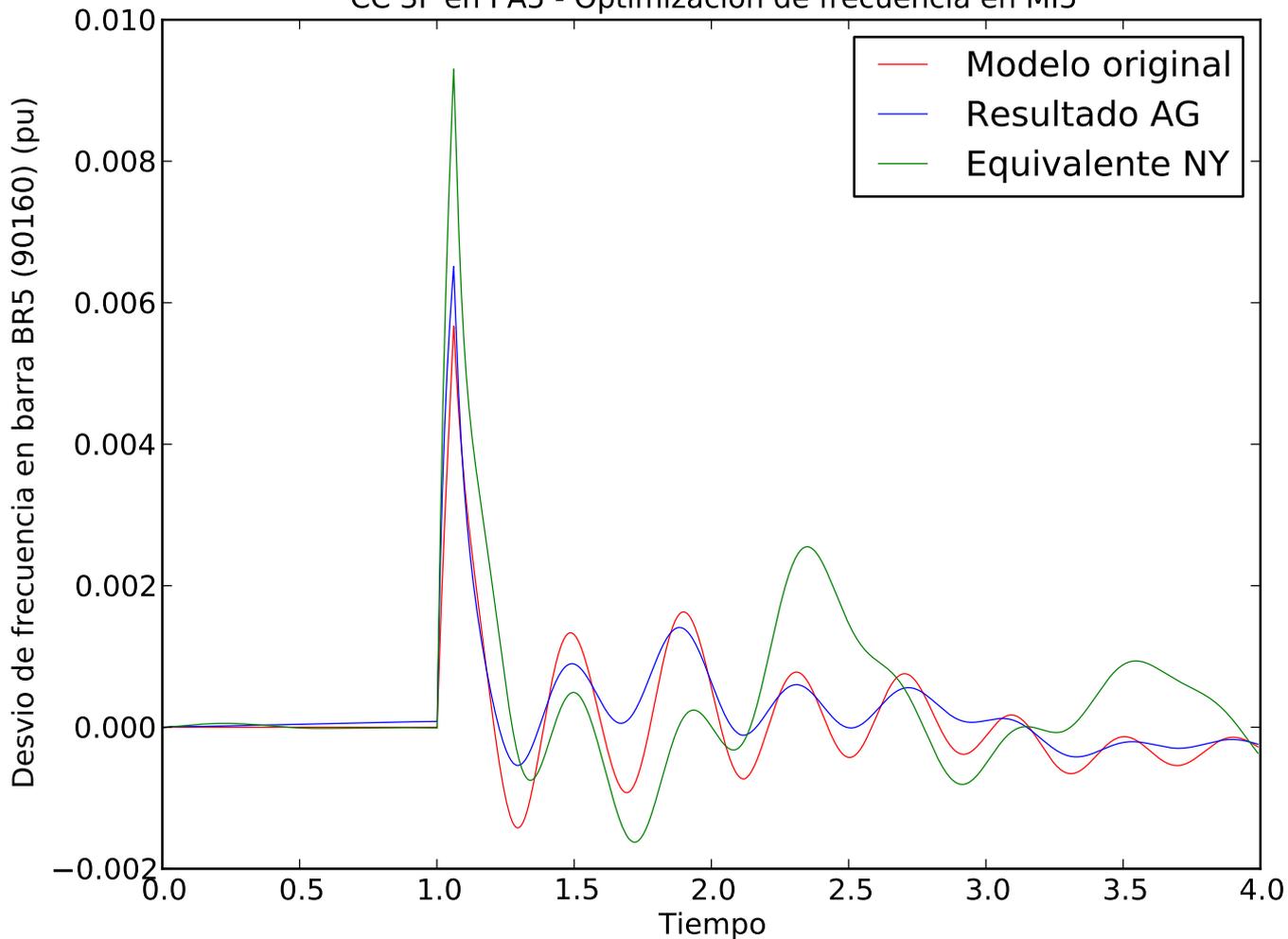
CC 3F en PA5 - Optimizacion de frecuencia en MI5



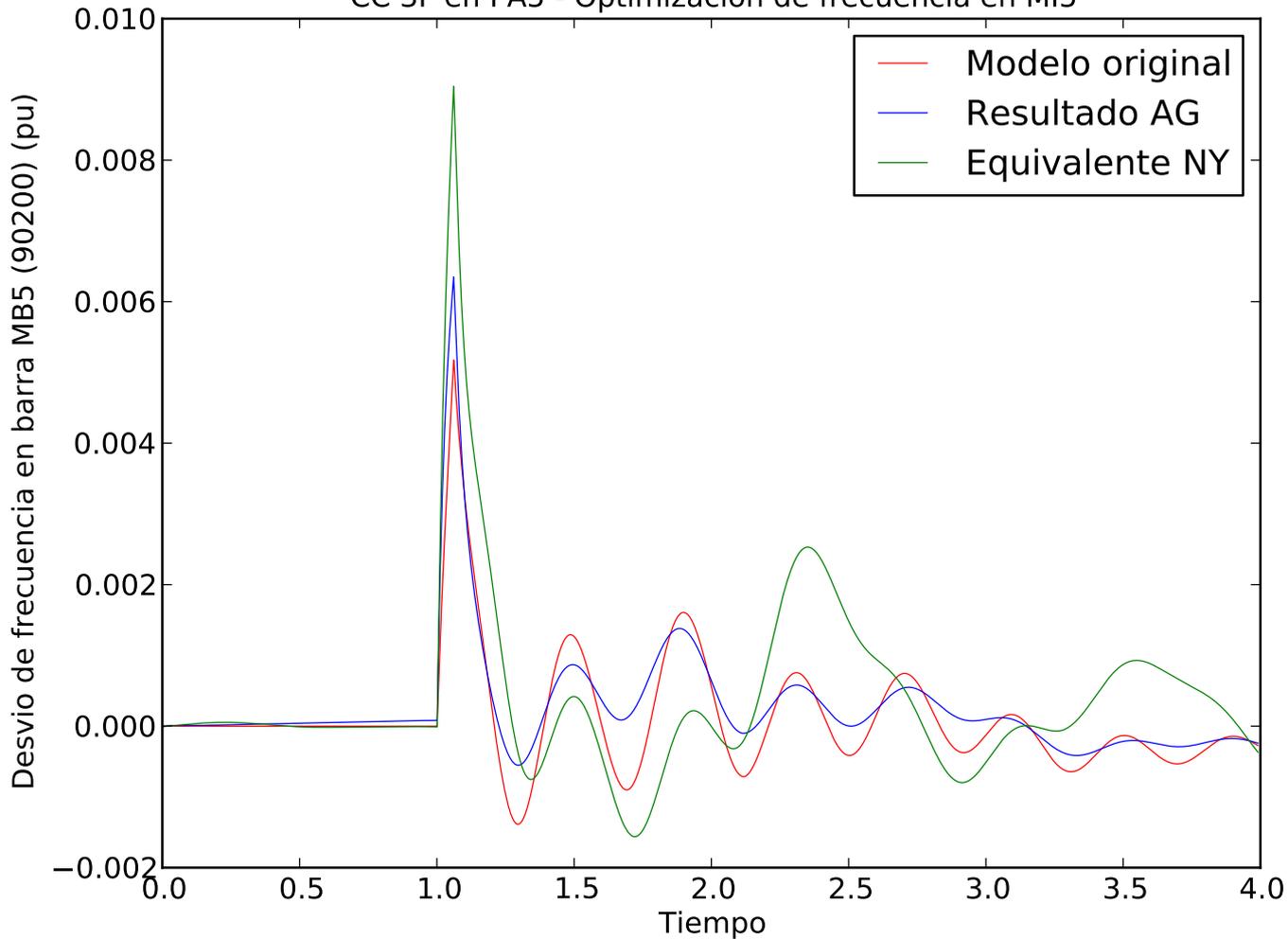
CC 3F en PA5 - Optimizacion de frecuencia en MI5



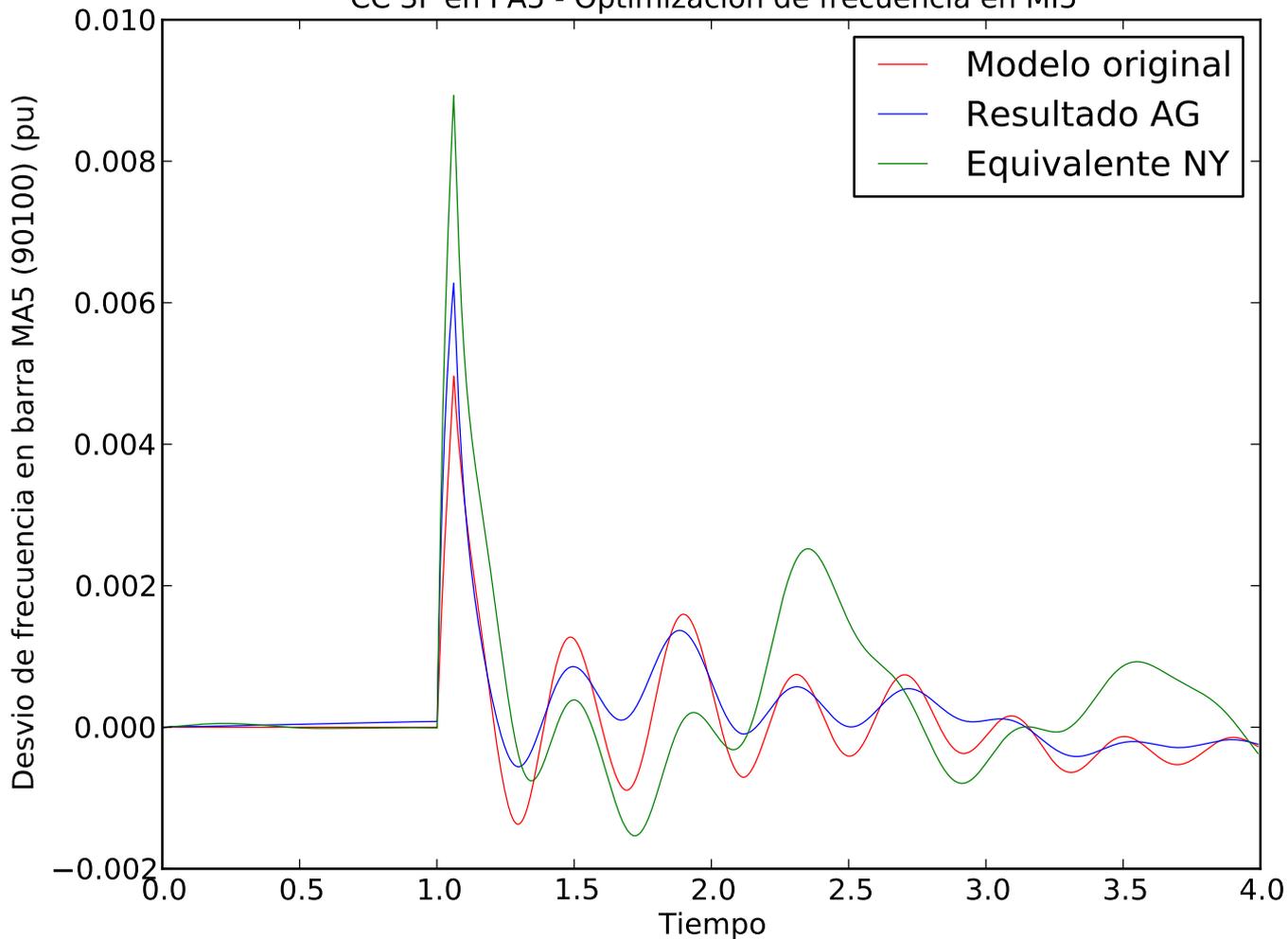
CC 3F en PA5 - Optimizacion de frecuencia en MI5



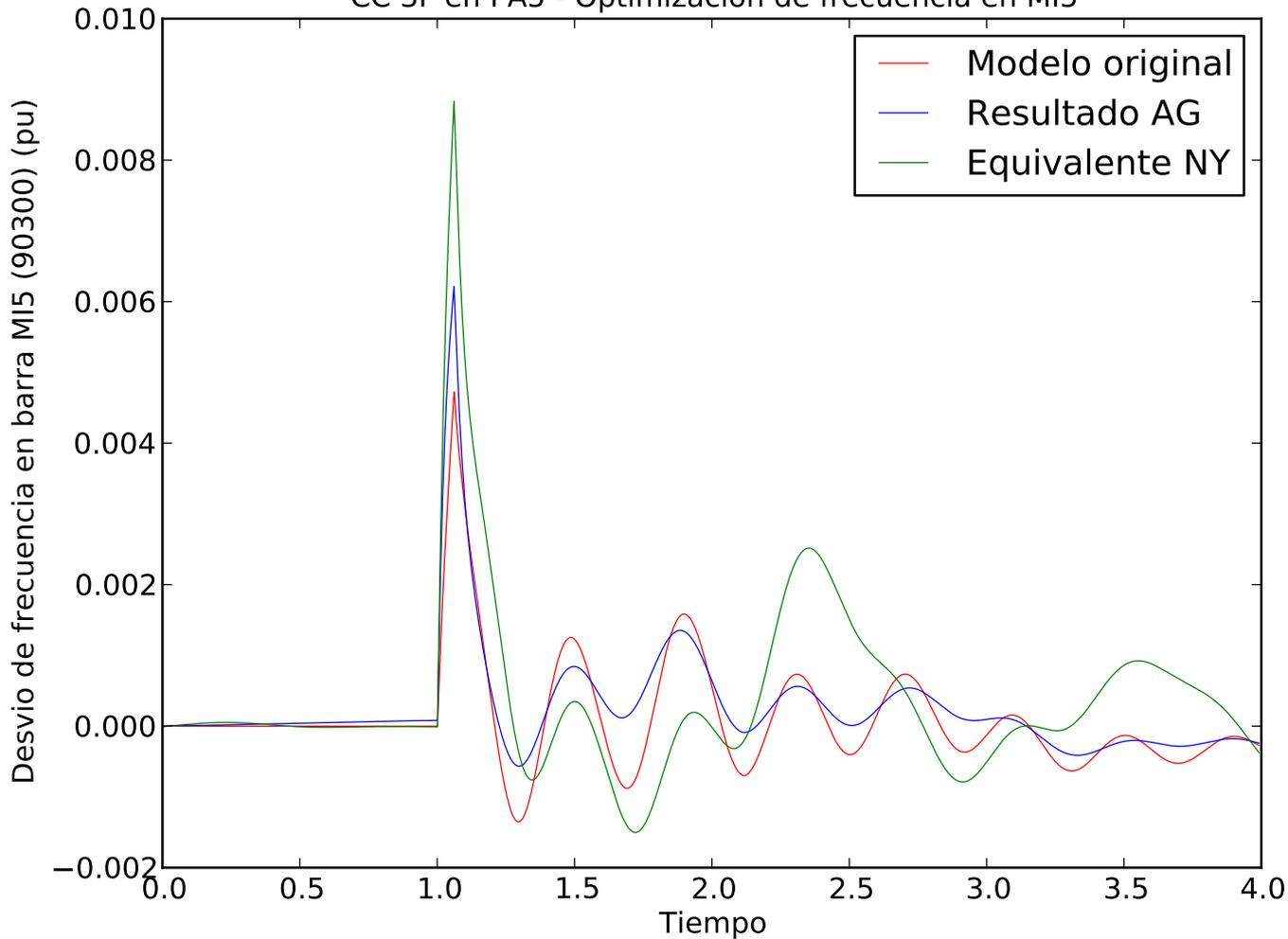
CC 3F en PA5 - Optimizacion de frecuencia en MI5



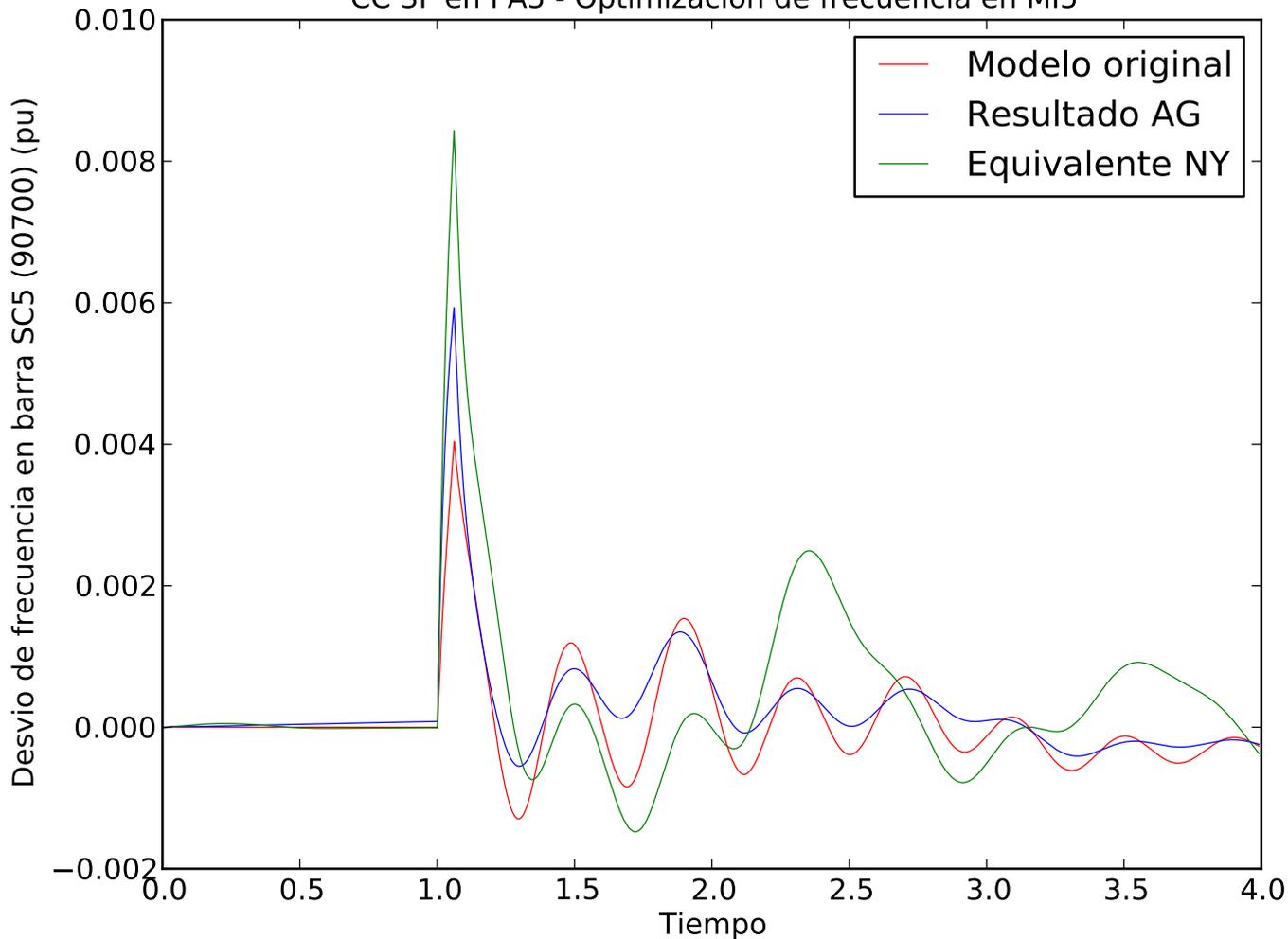
CC 3F en PA5 - Optimizacion de frecuencia en MI5



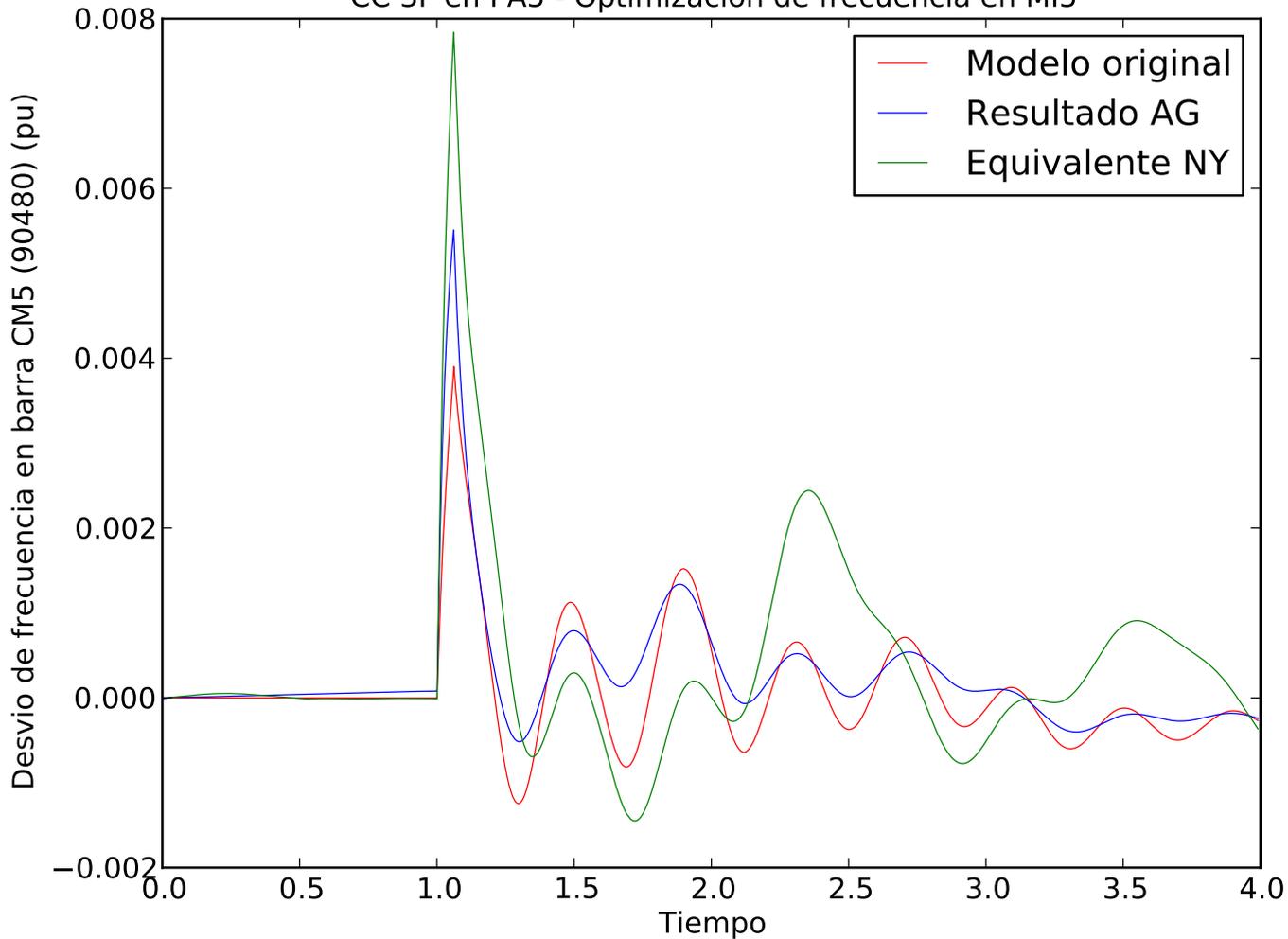
CC 3F en PA5 - Optimizacion de frecuencia en MI5



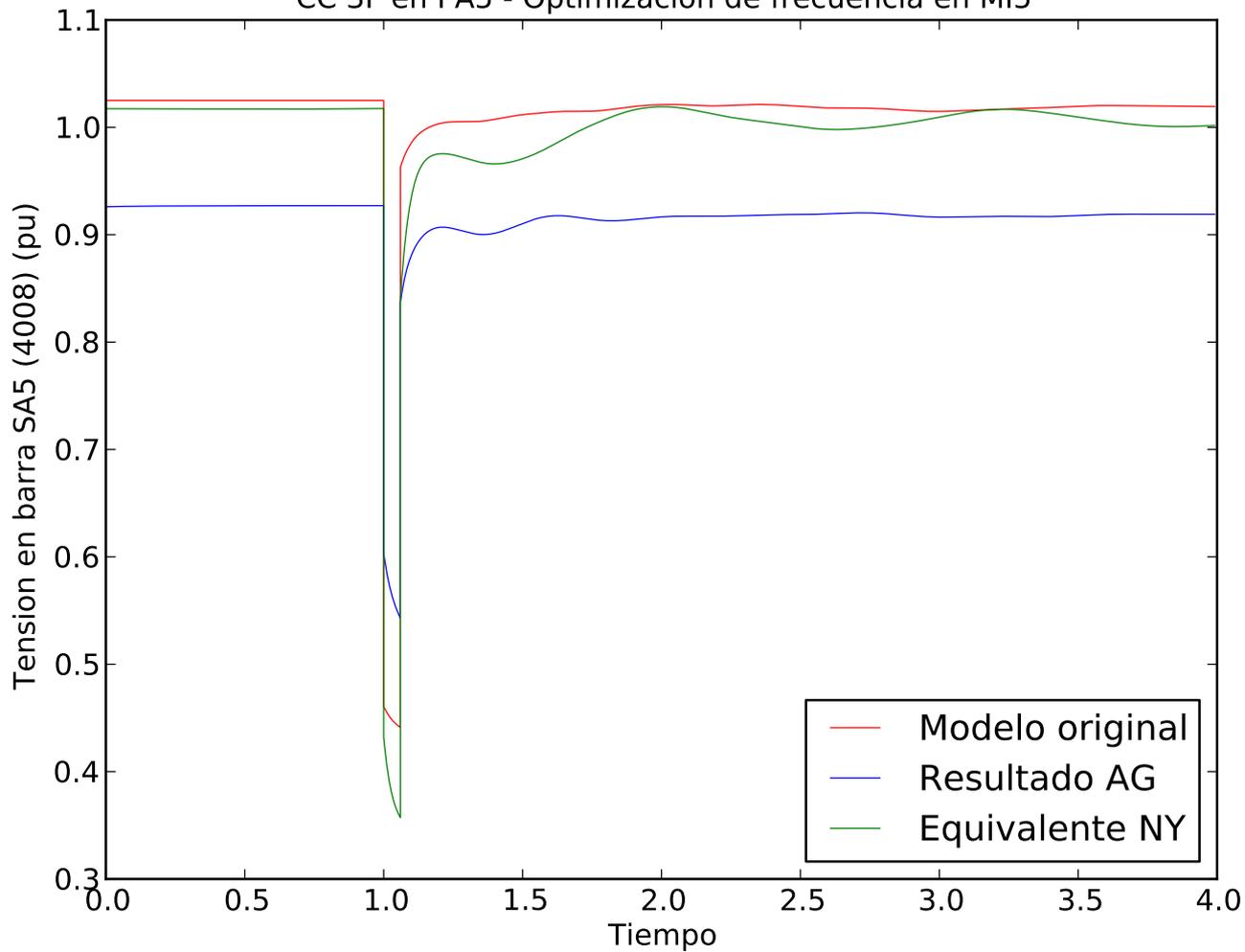
CC 3F en PA5 - Optimizacion de frecuencia en MI5



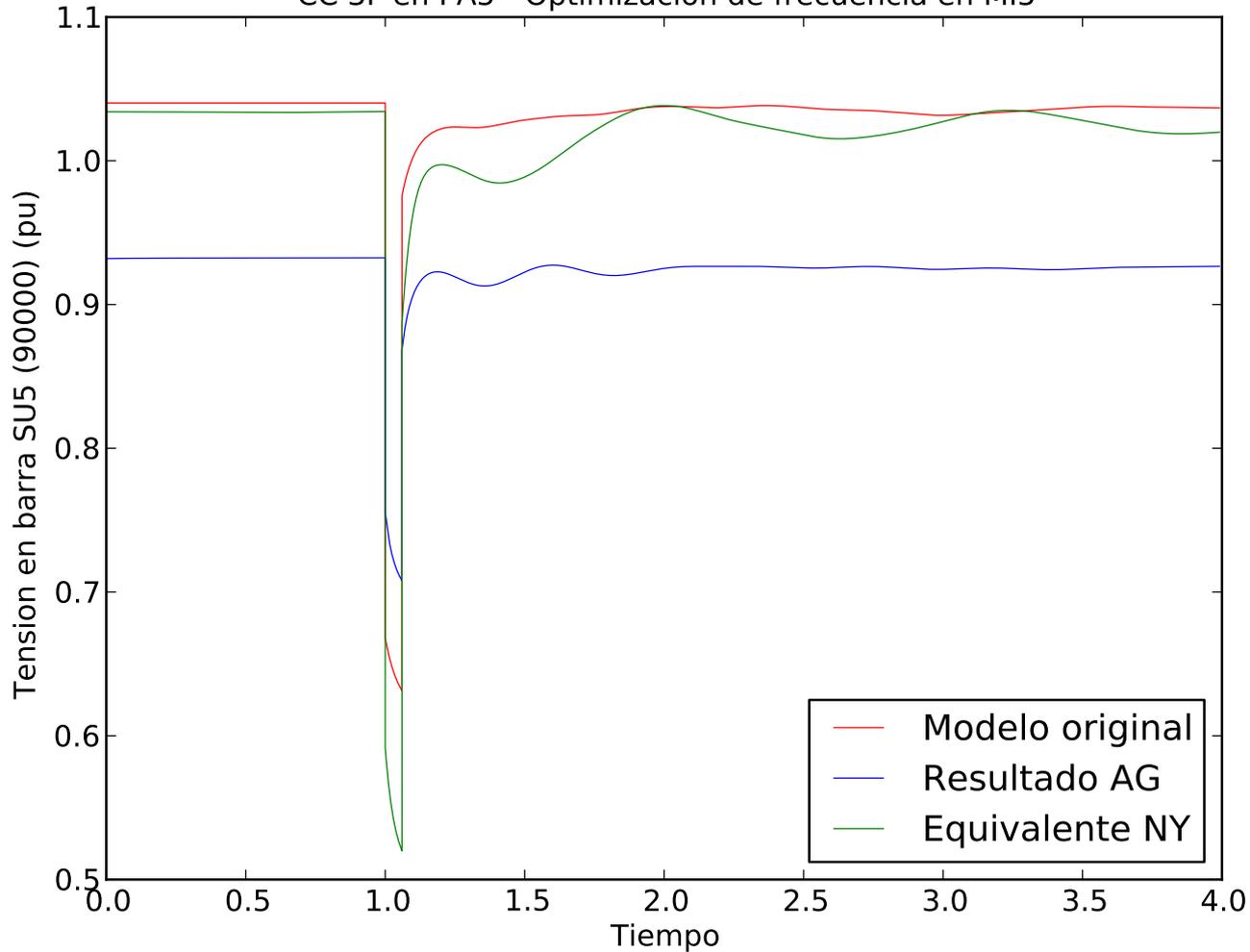
CC 3F en PA5 - Optimizacion de frecuencia en MI5



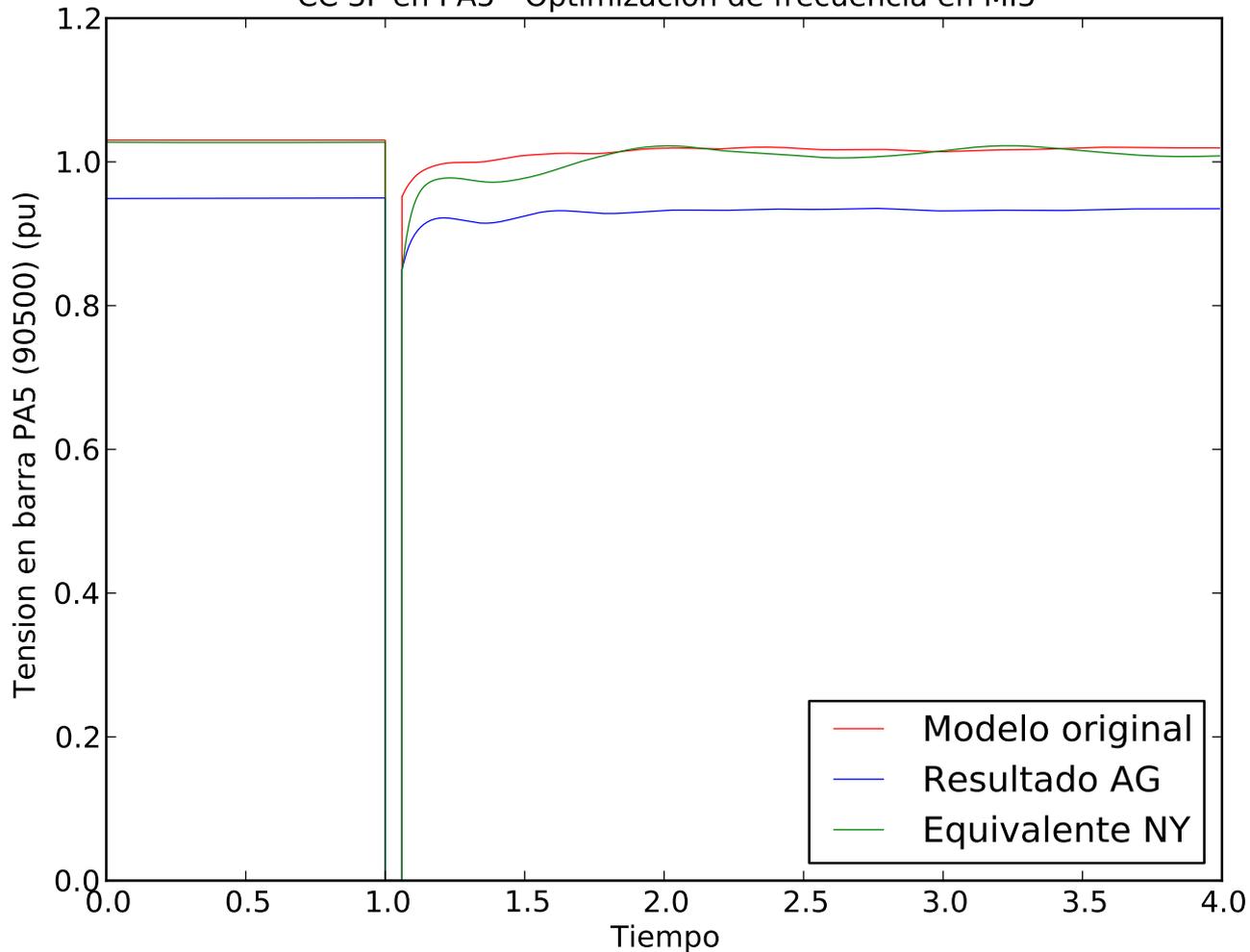
CC 3F en PA5 - Optimizacion de frecuencia en MI5



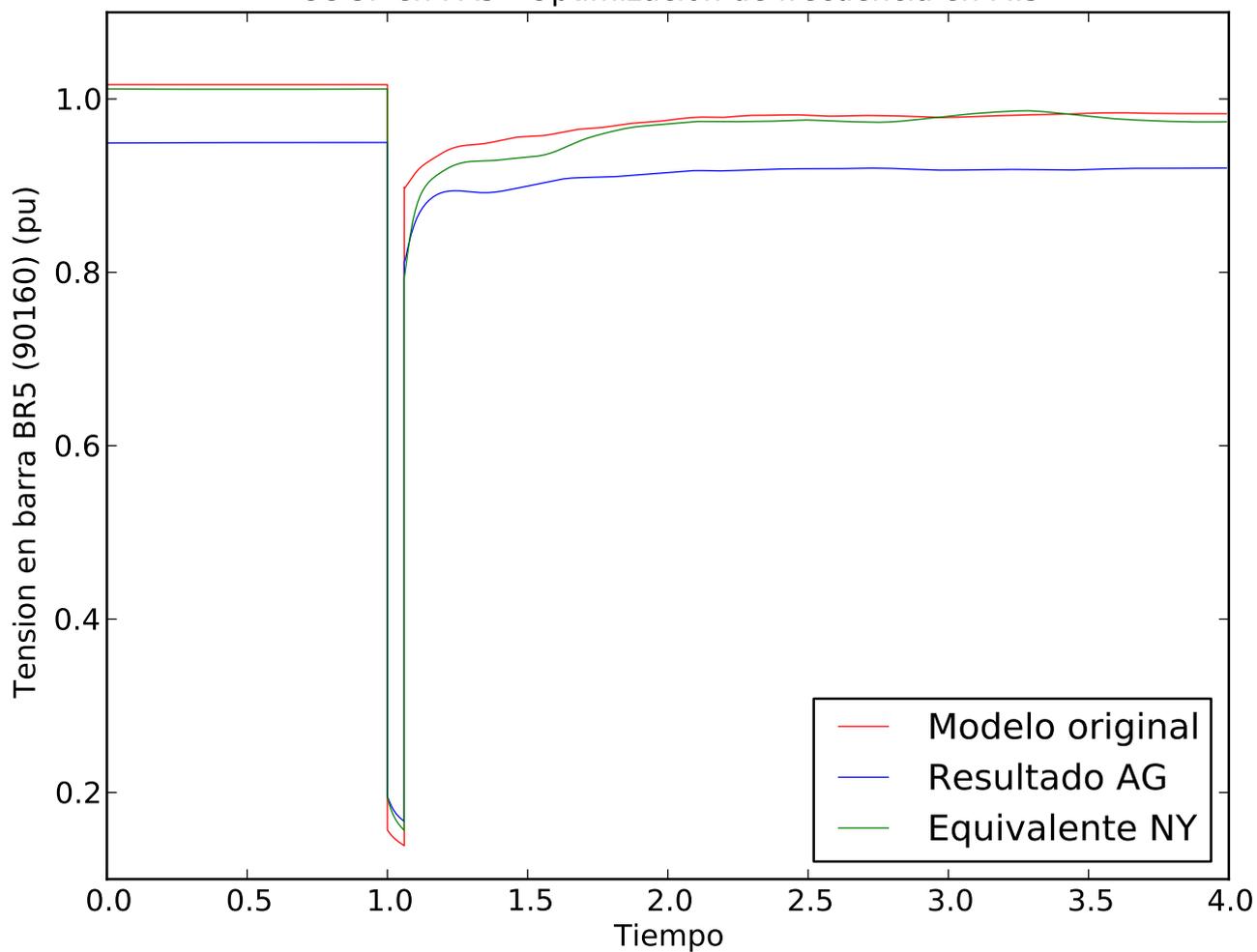
CC 3F en PA5 - Optimizacion de frecuencia en MI5



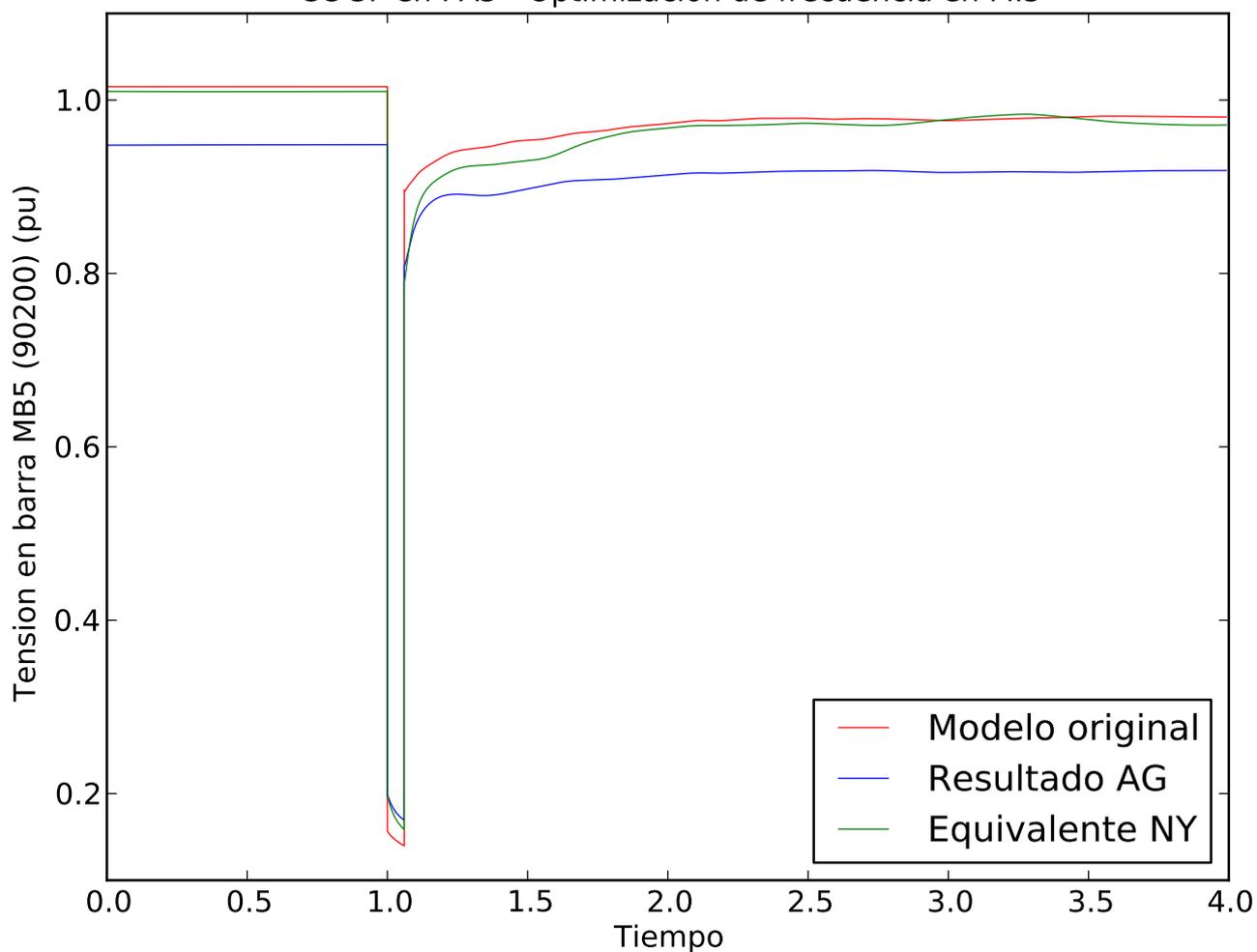
CC 3F en PA5 - Optimizacion de frecuencia en MI5



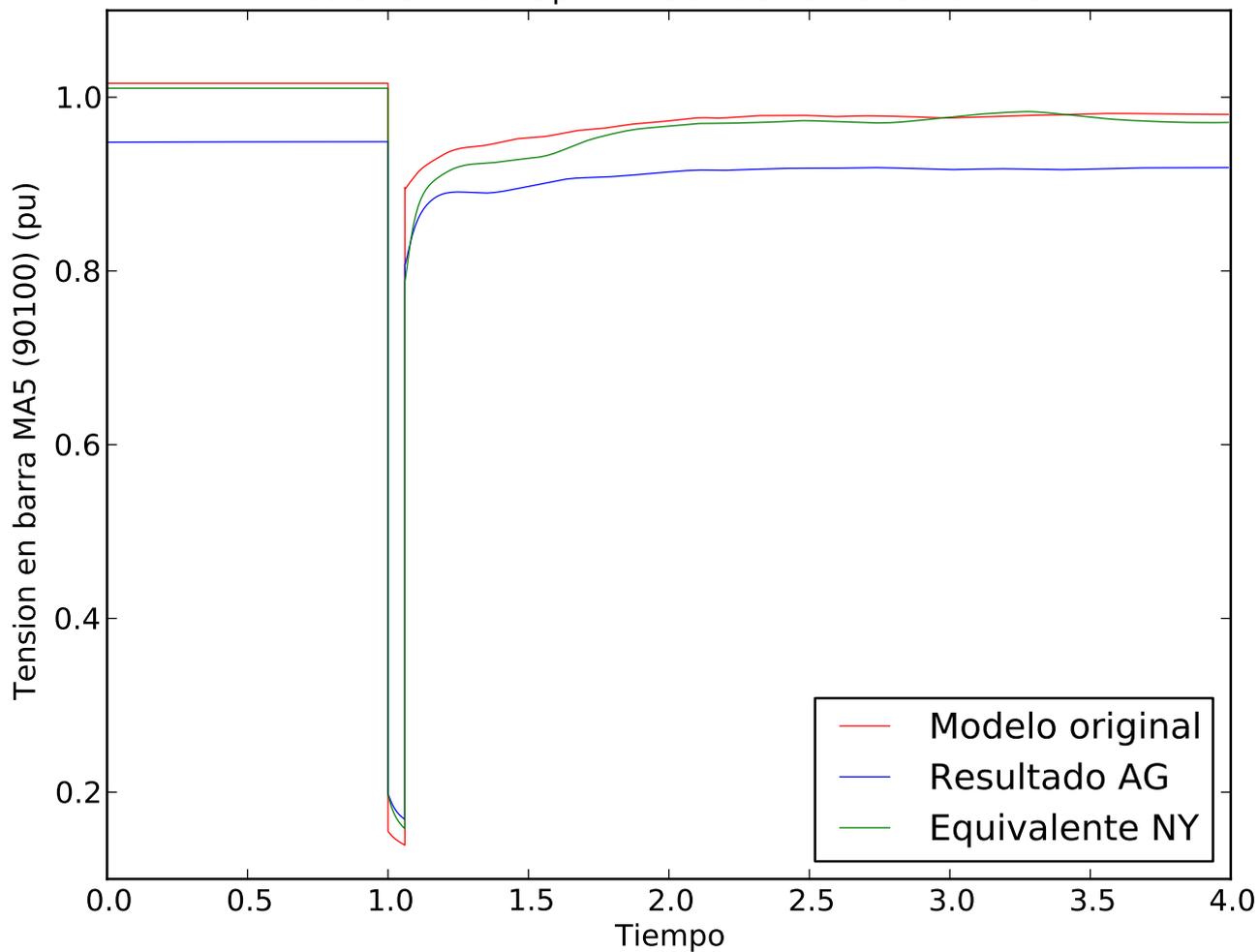
CC 3F en PA5 - Optimizacion de frecuencia en MI5



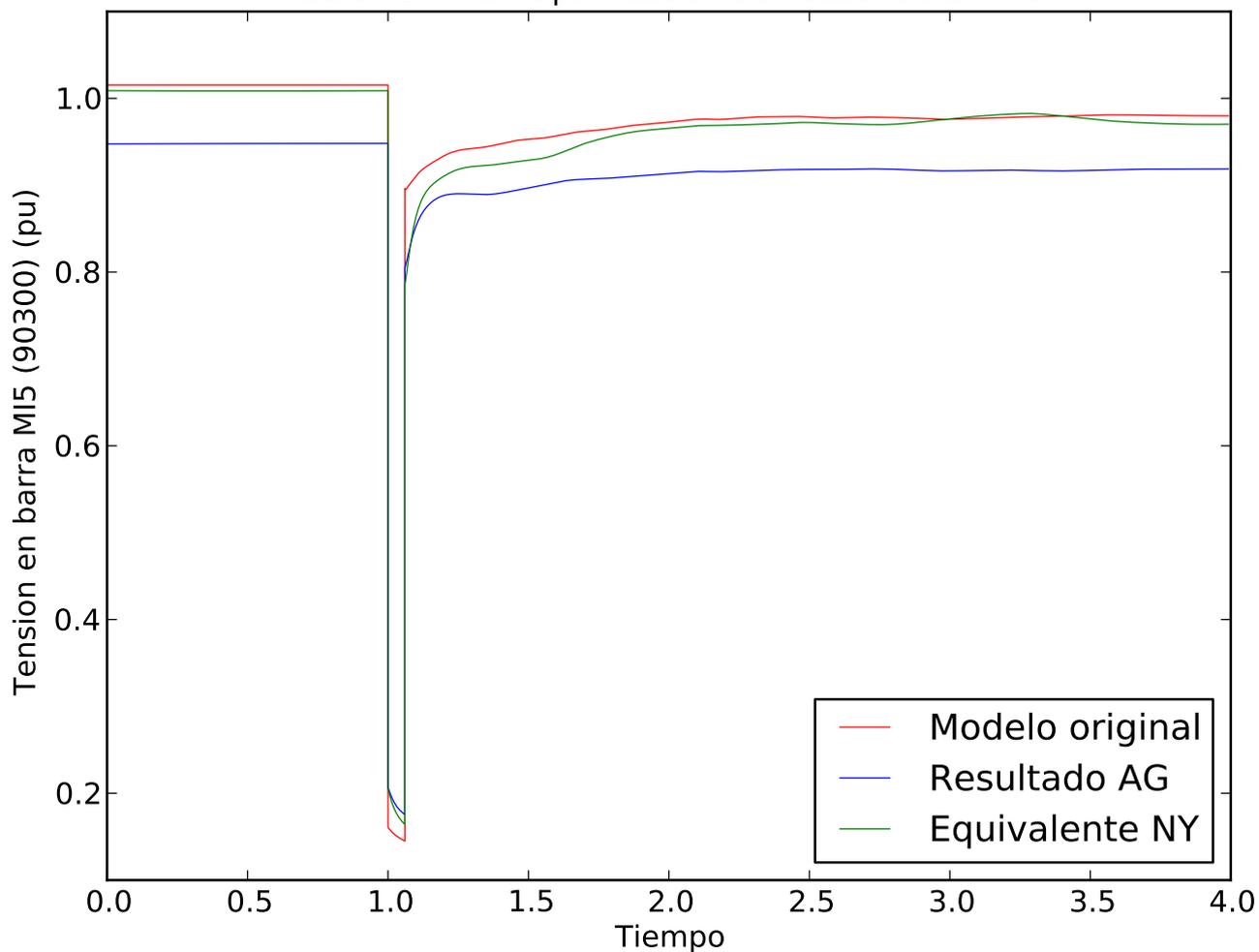
CC 3F en PA5 - Optimizacion de frecuencia en MI5



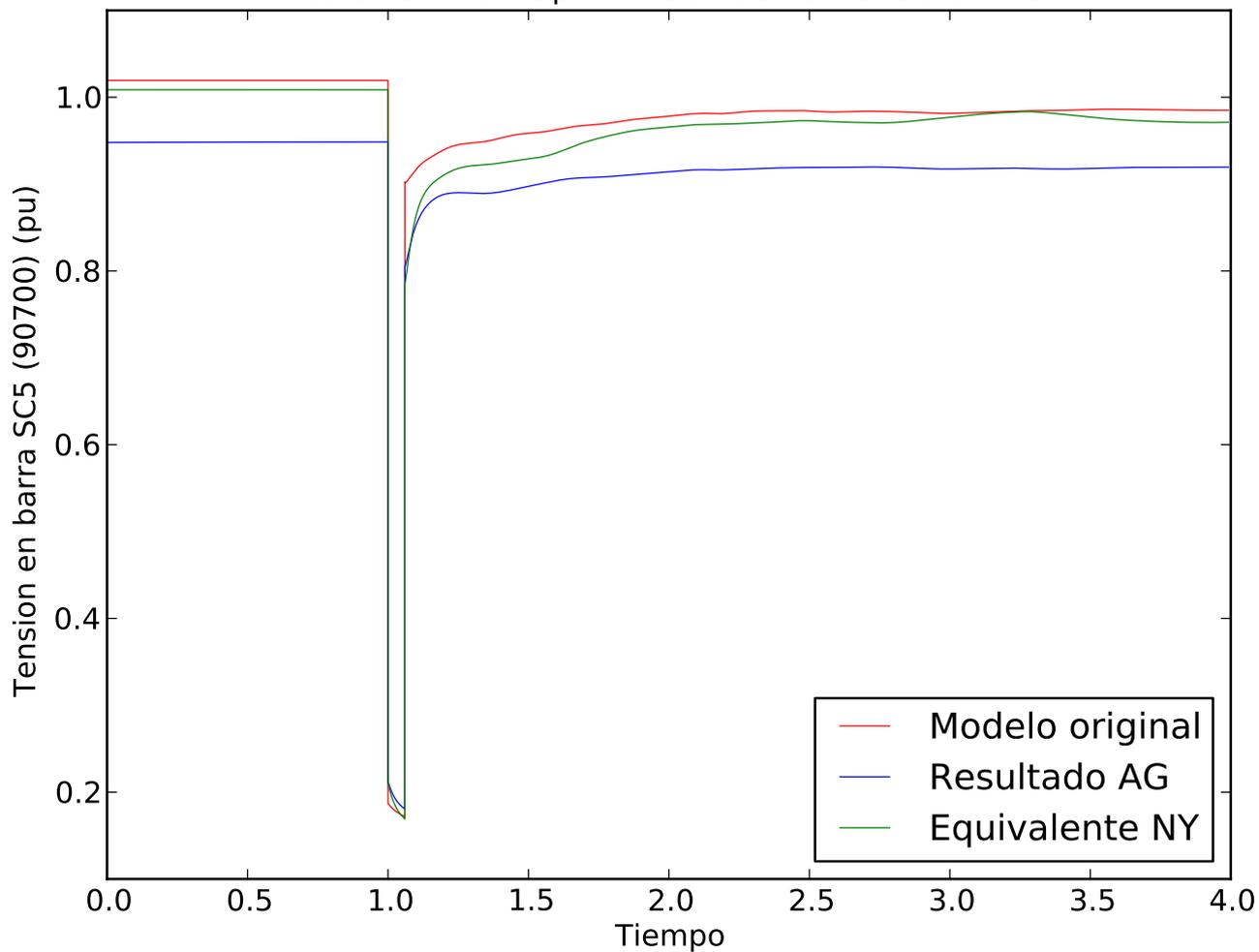
CC 3F en PA5 - Optimizacion de frecuencia en MI5



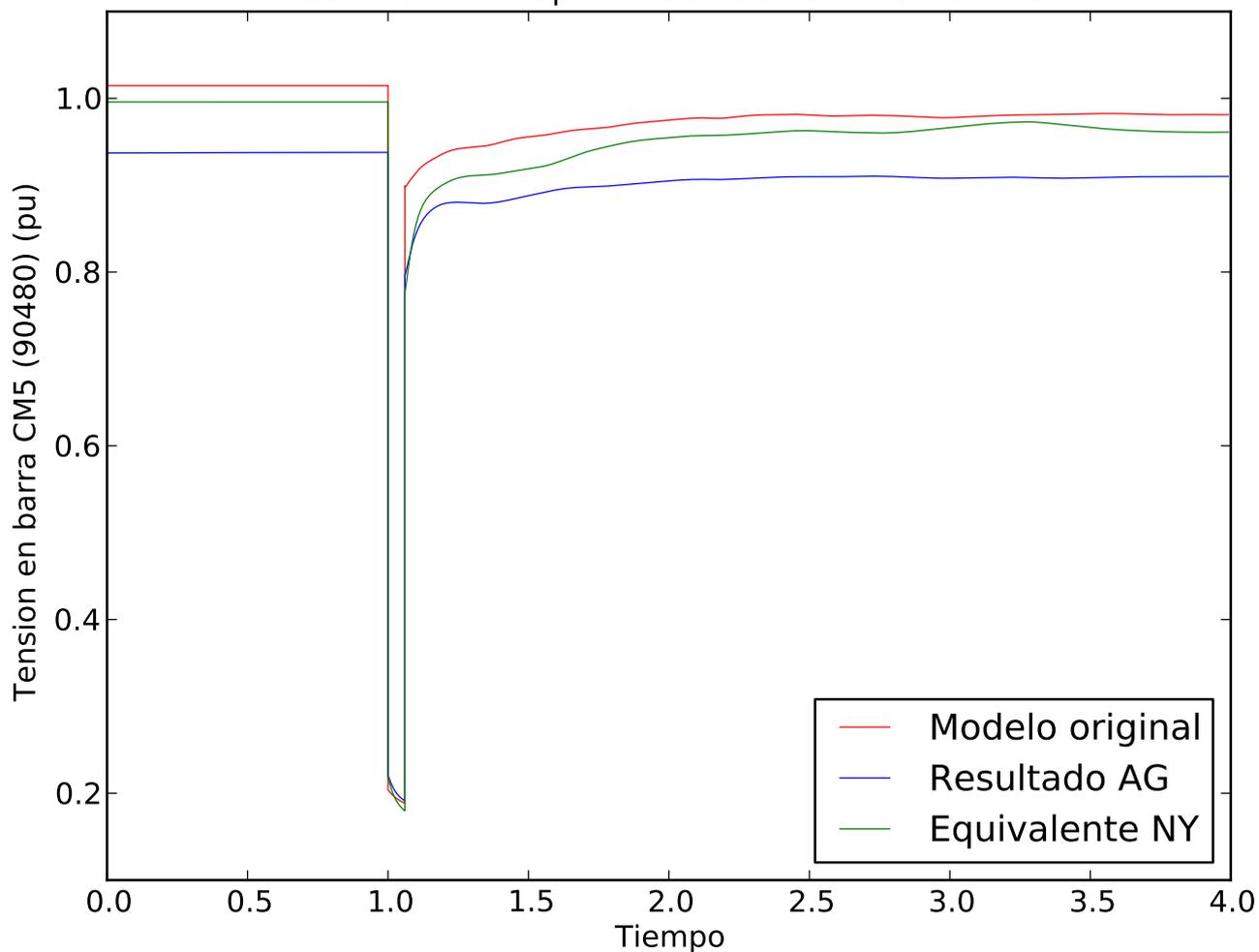
CC 3F en PA5 - Optimizacion de frecuencia en MI5



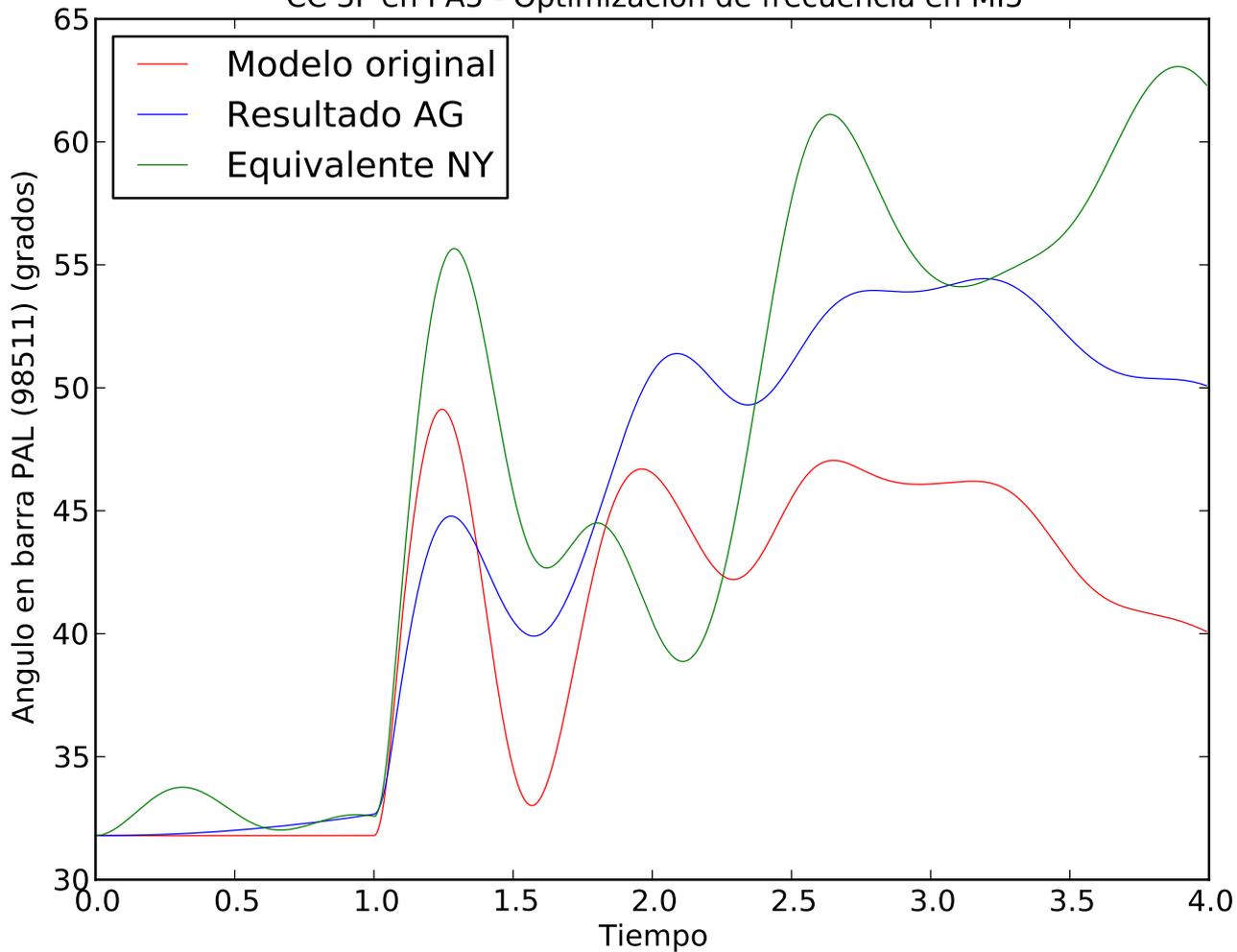
CC 3F en PA5 - Optimizacion de frecuencia en MI5



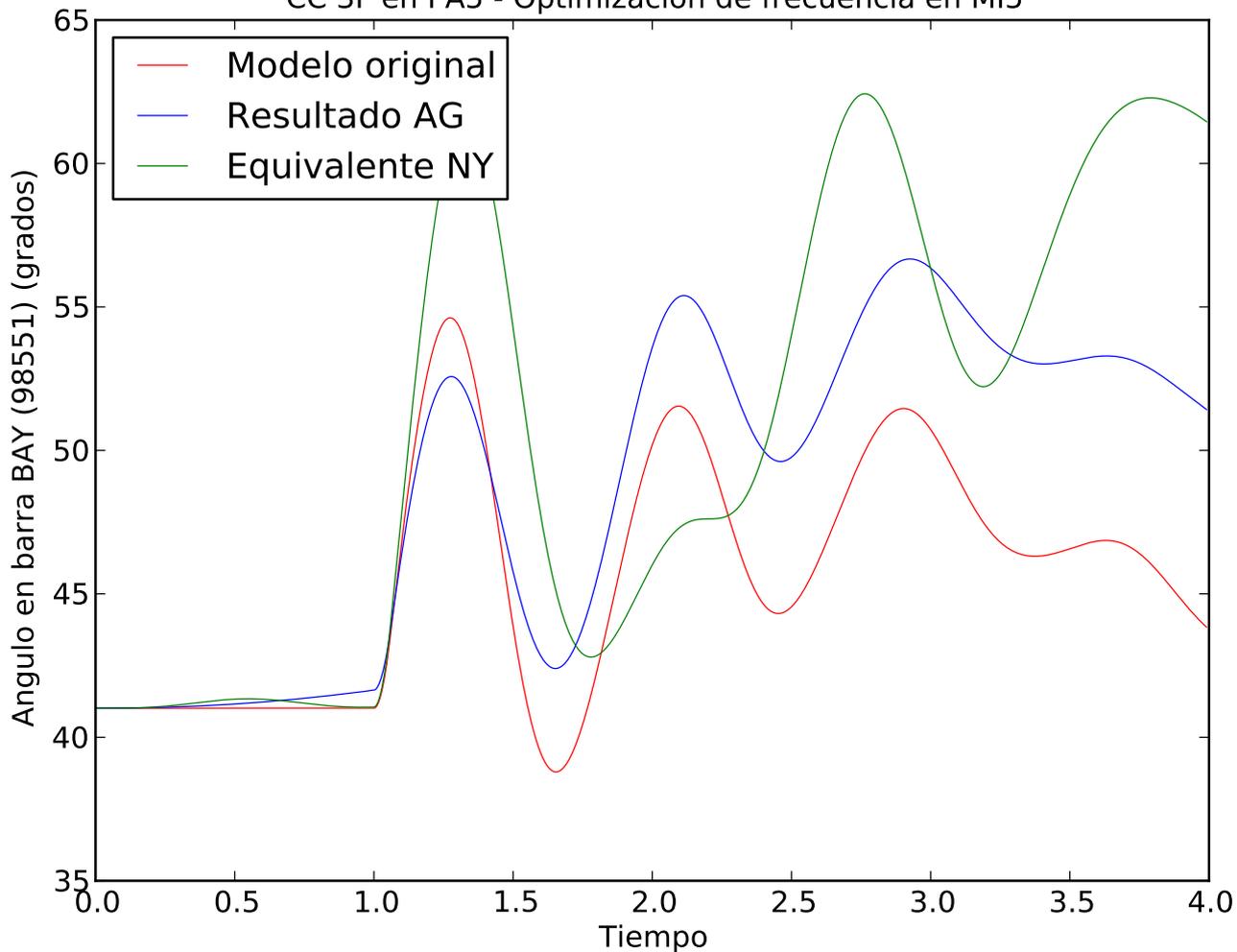
CC 3F en PA5 - Optimizacion de frecuencia en MI5



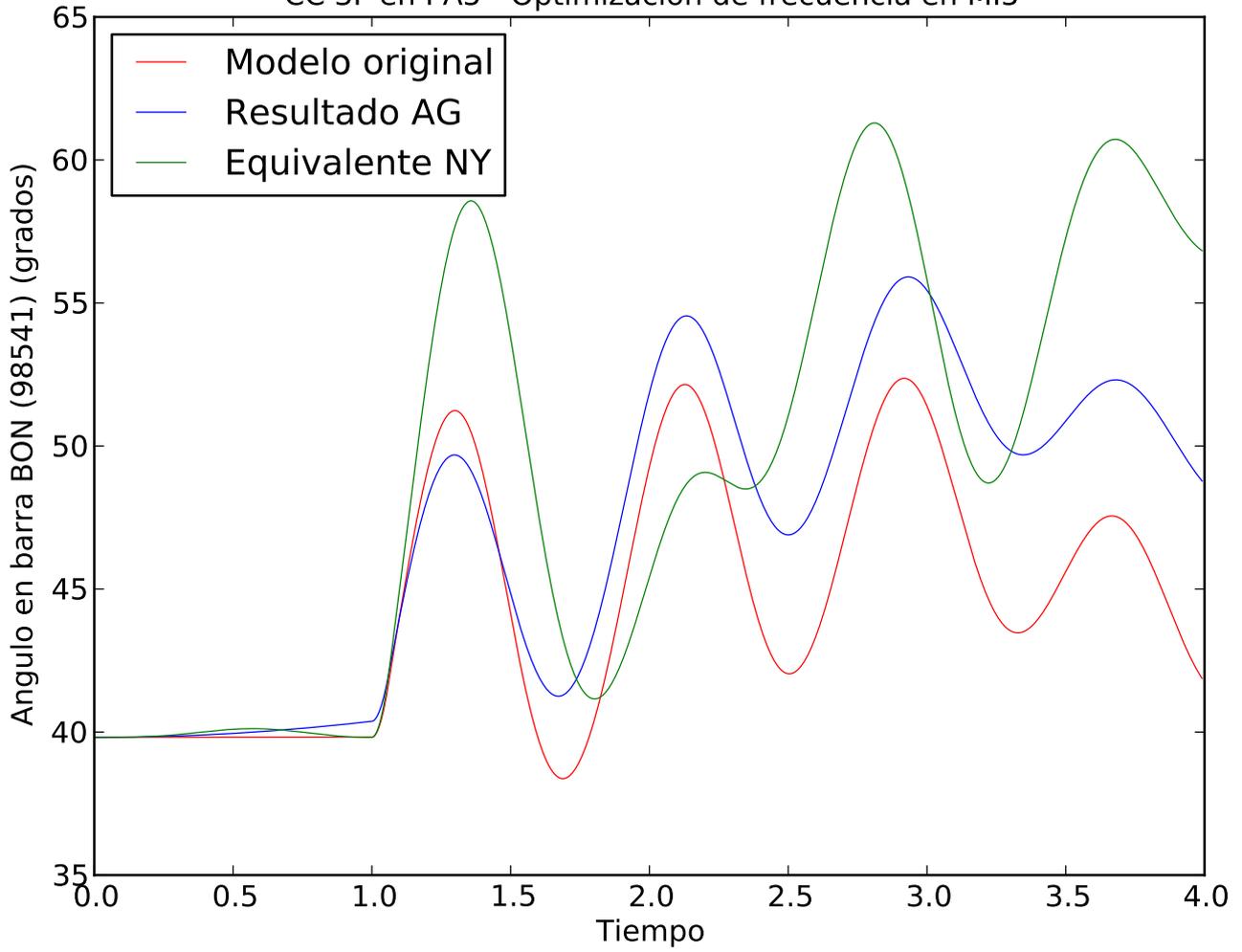
CC 3F en PA5 - Optimizacion de frecuencia en MI5



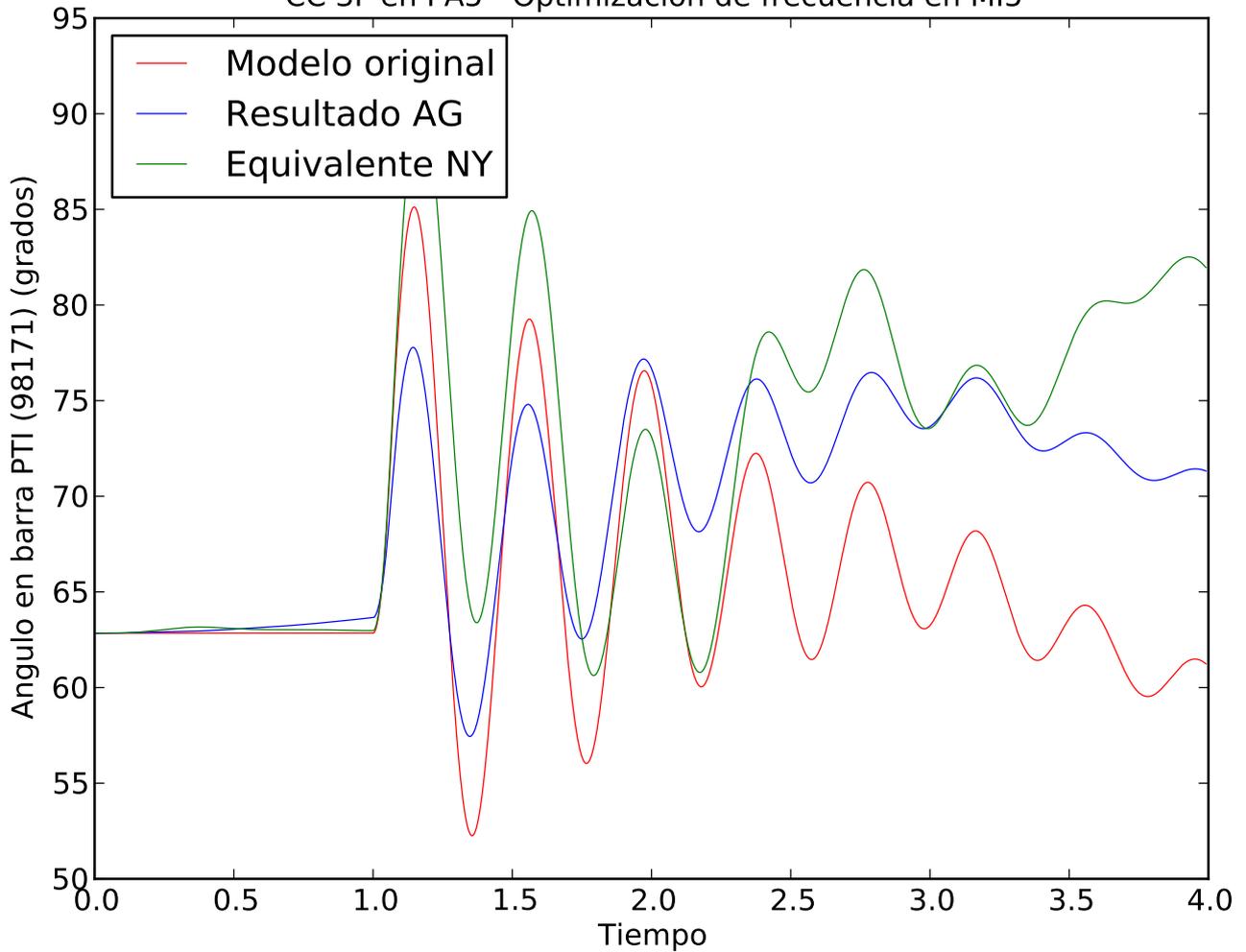
CC 3F en PA5 - Optimizacion de frecuencia en MI5



CC 3F en PA5 - Optimizacion de frecuencia en MI5

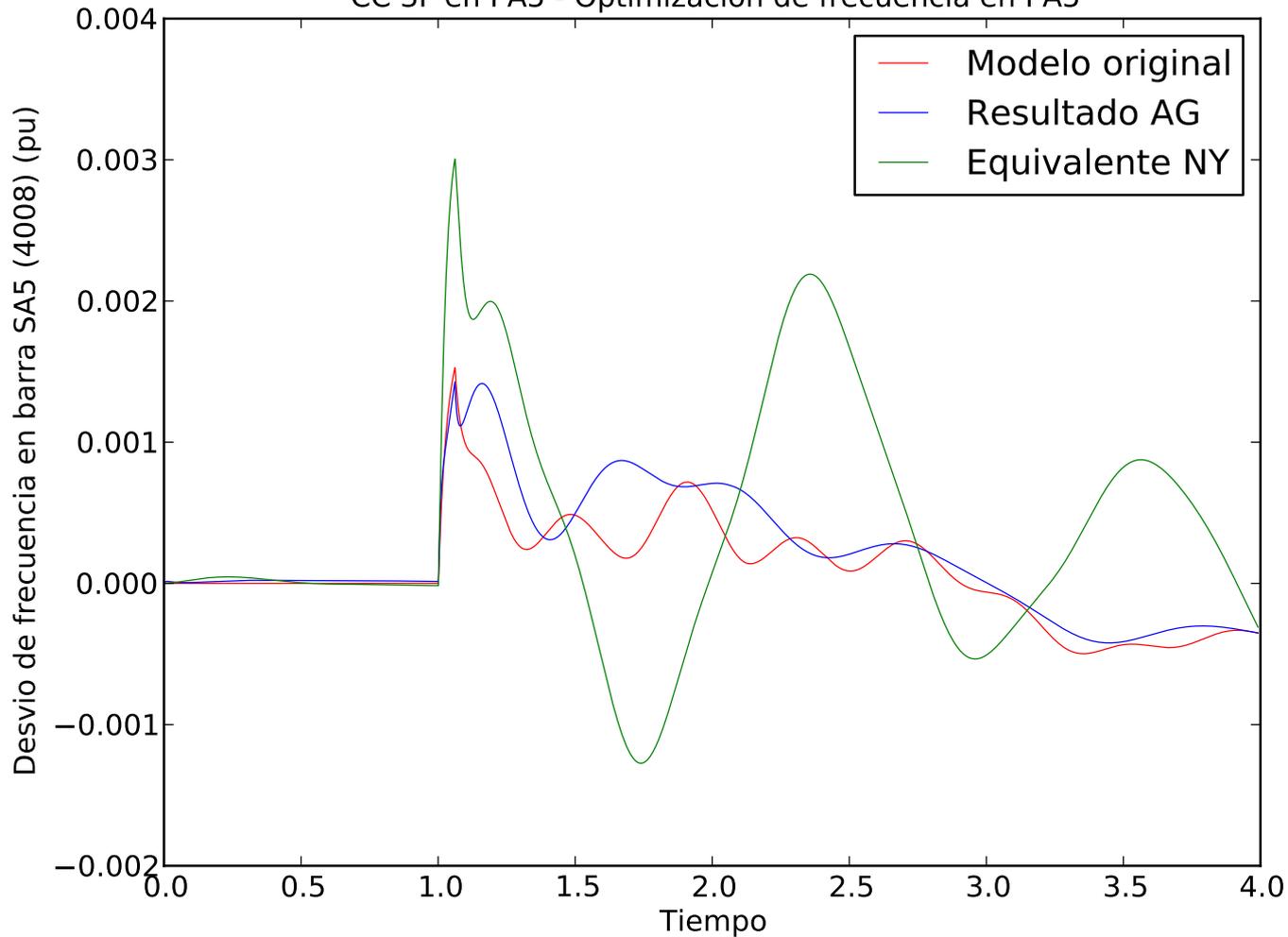


CC 3F en PA5 - Optimizacion de frecuencia en MI5

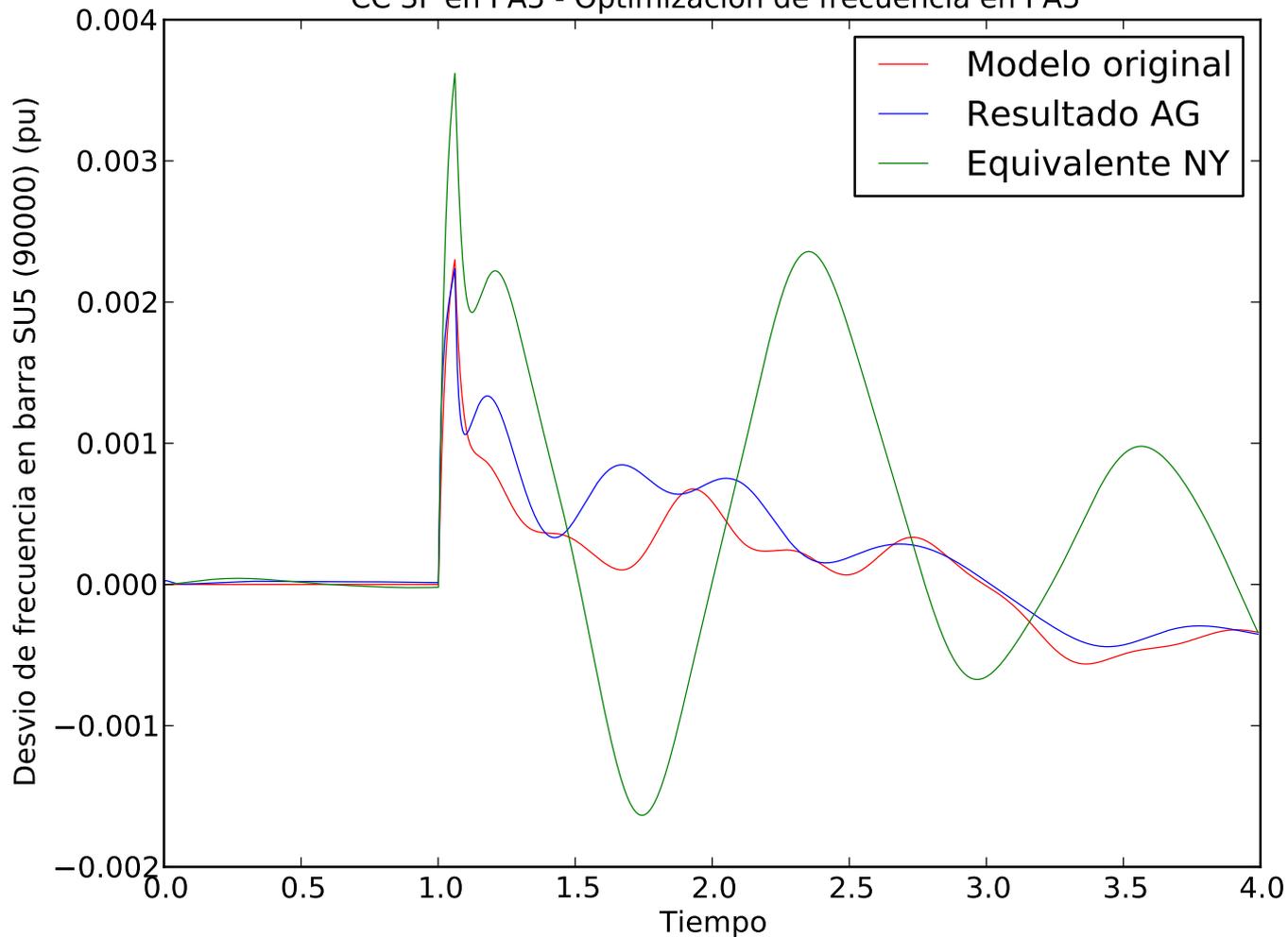


**C.2. Simulación 2: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando frecuencia en PA5 en 4 segundos. Comportamiento en 4 segundos**

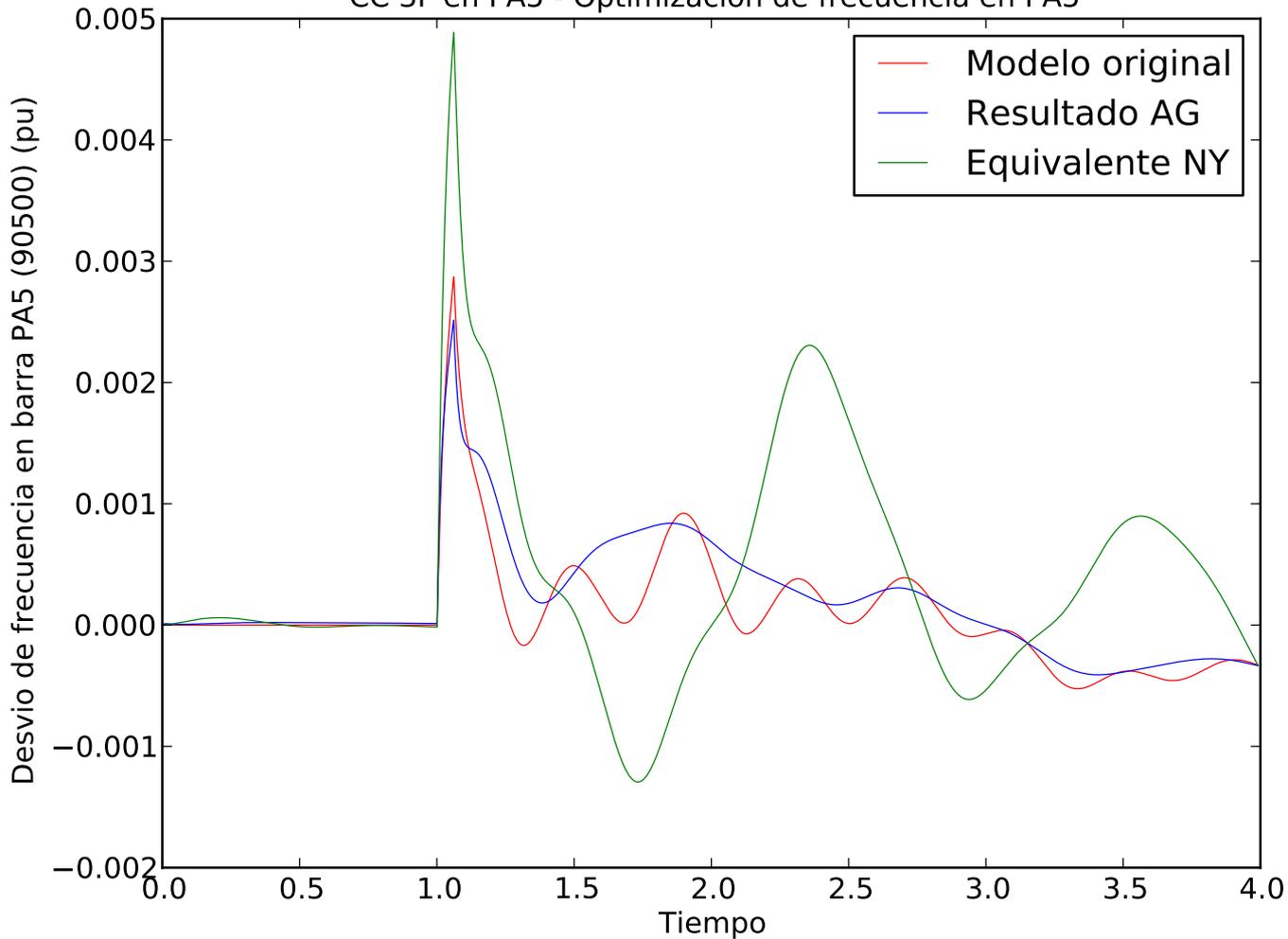
CC 3F en PA5 - Optimizacion de frecuencia en PA5



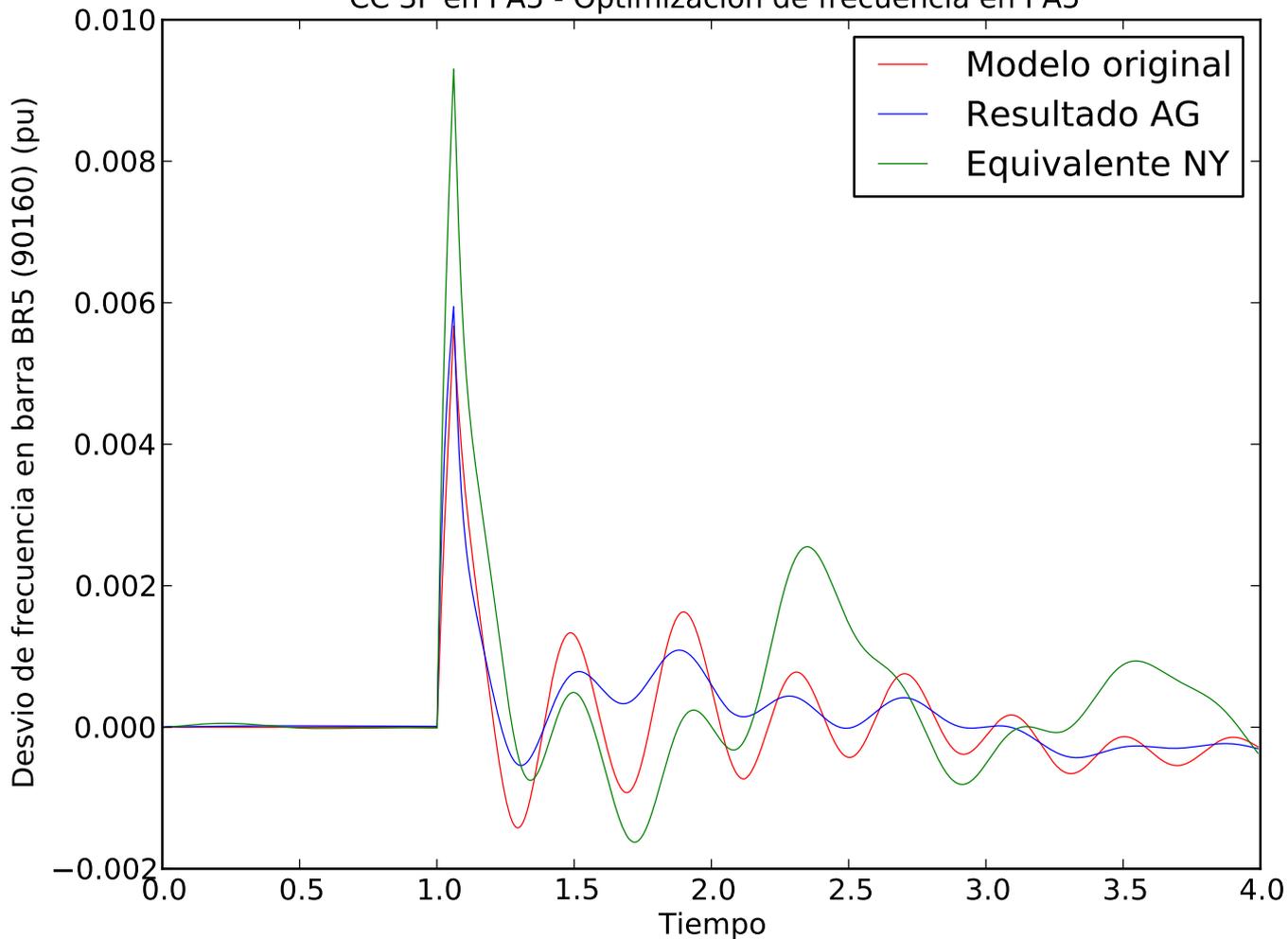
CC 3F en PA5 - Optimizacion de frecuencia en PA5



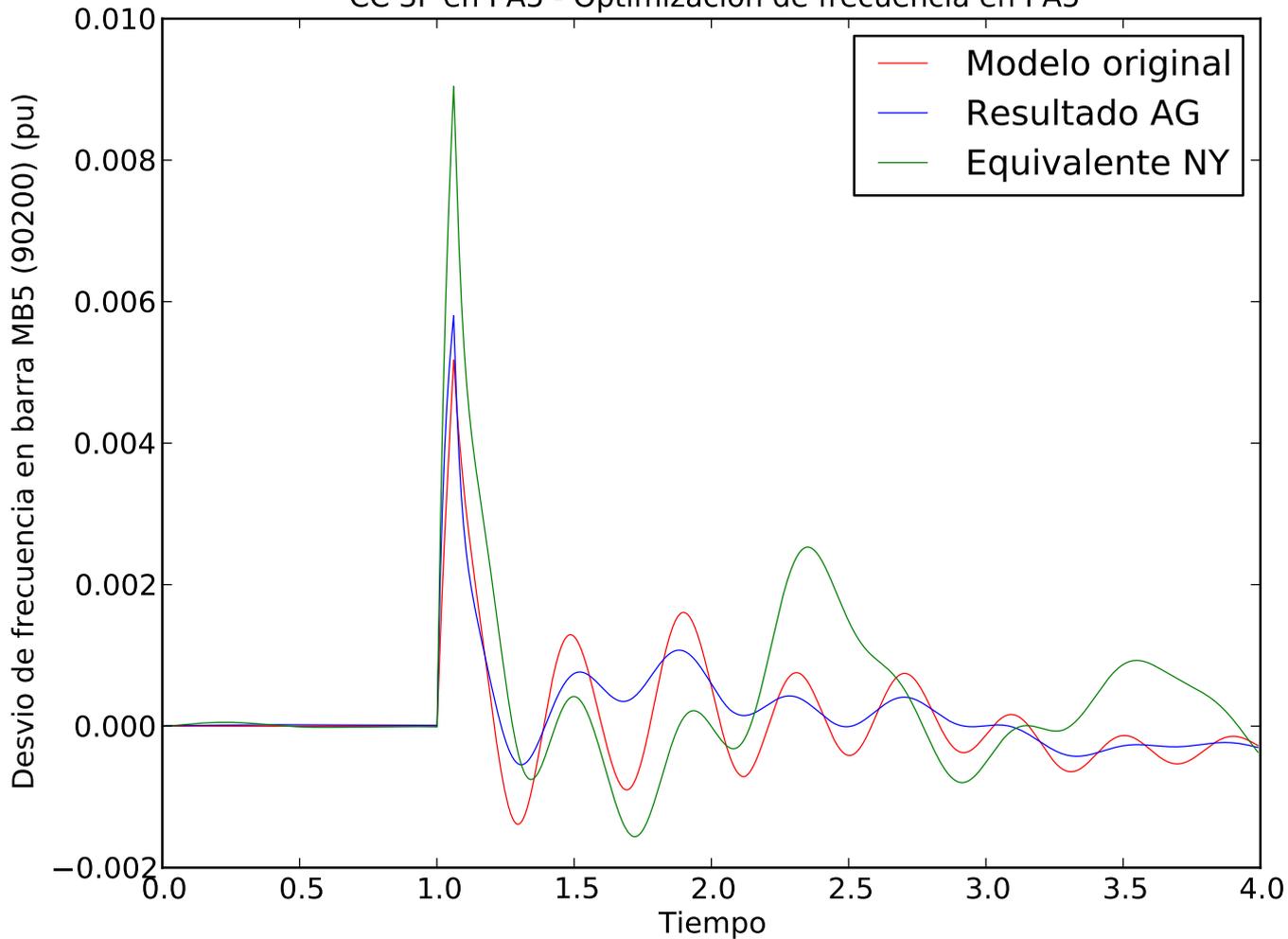
CC 3F en PA5 - Optimizacion de frecuencia en PA5



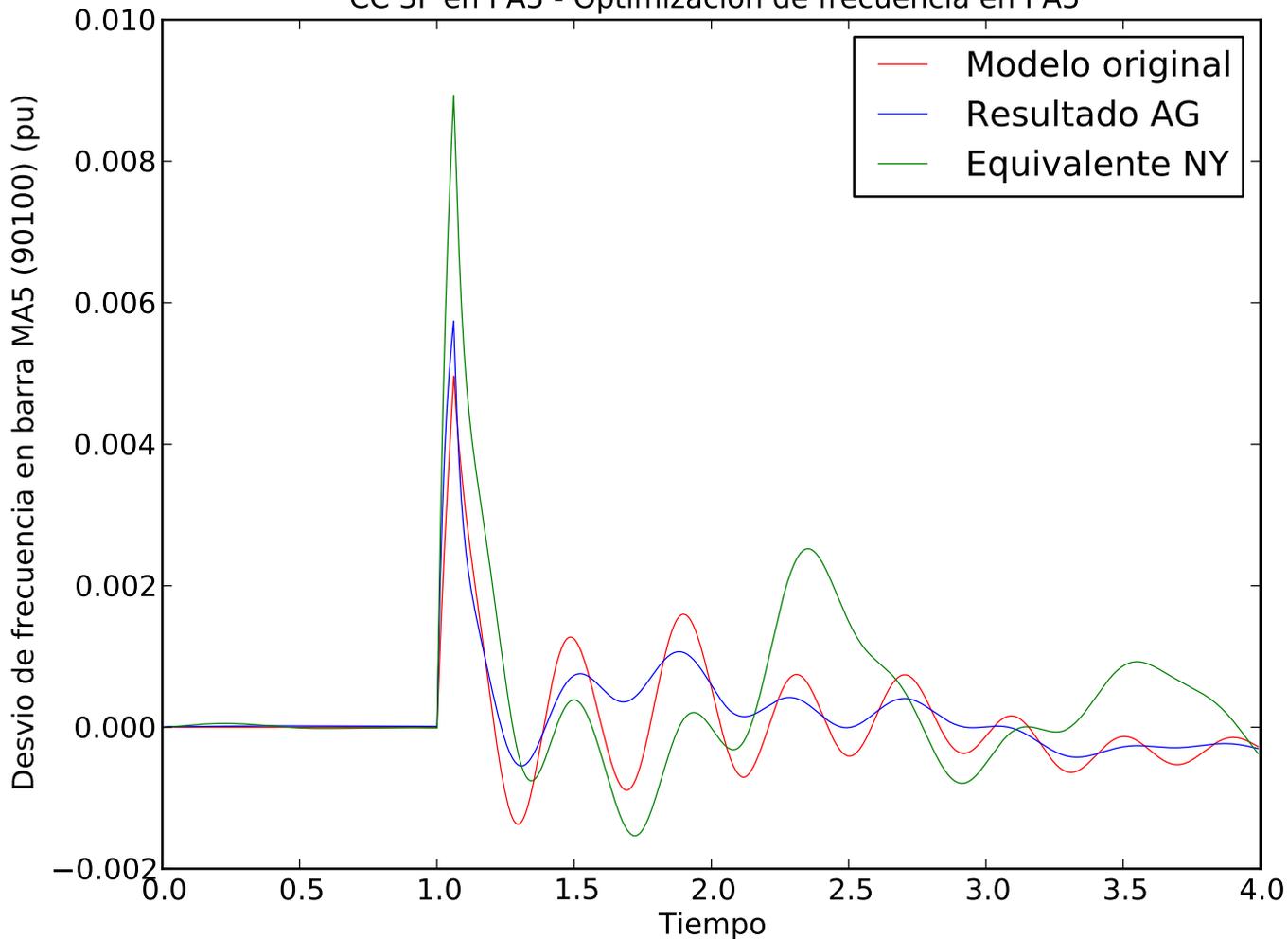
CC 3F en PA5 - Optimizacion de frecuencia en PA5



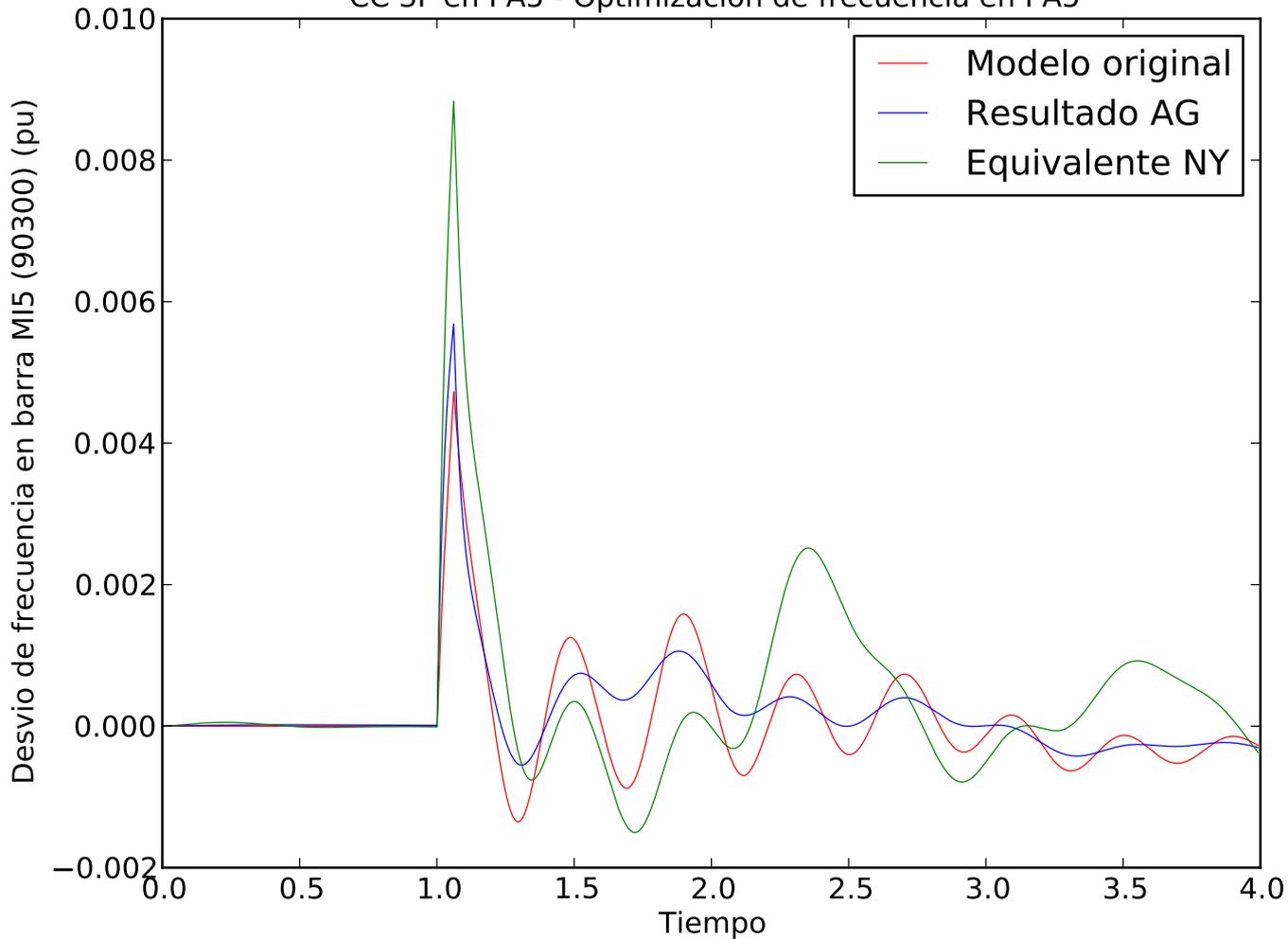
CC 3F en PA5 - Optimizacion de frecuencia en PA5



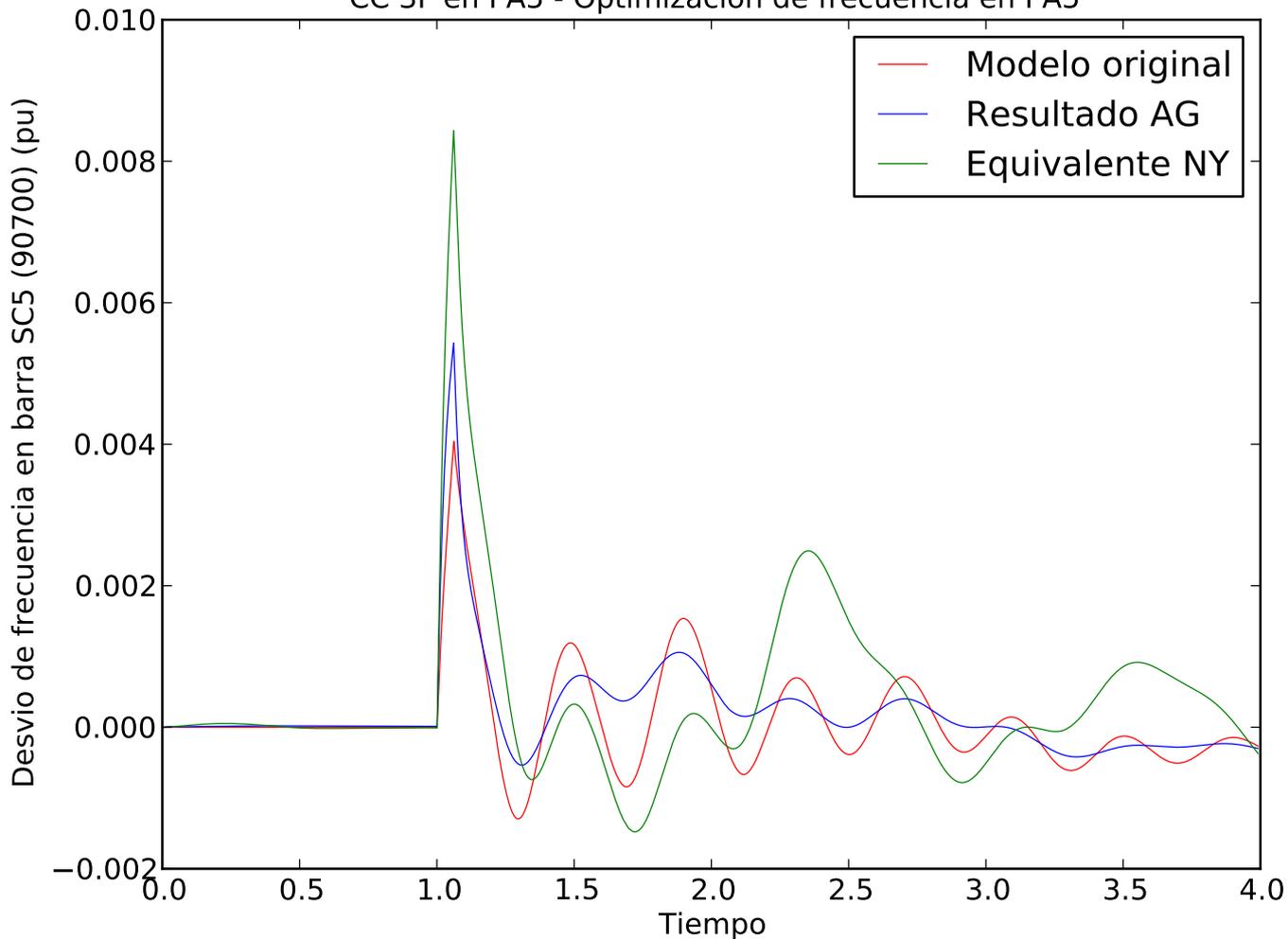
CC 3F en PA5 - Optimizacion de frecuencia en PA5



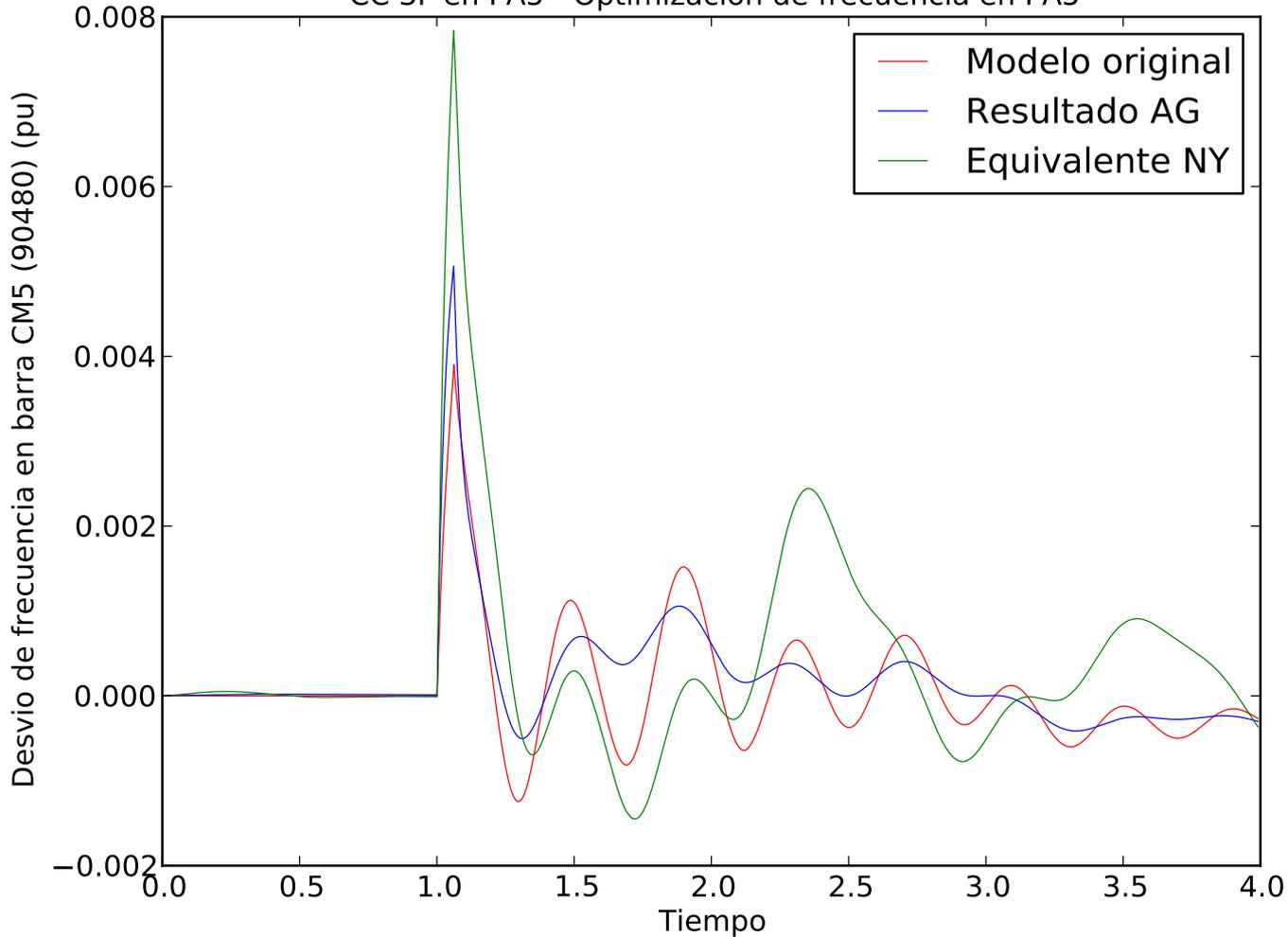
CC 3F en PA5 - Optimizacion de frecuencia en PA5



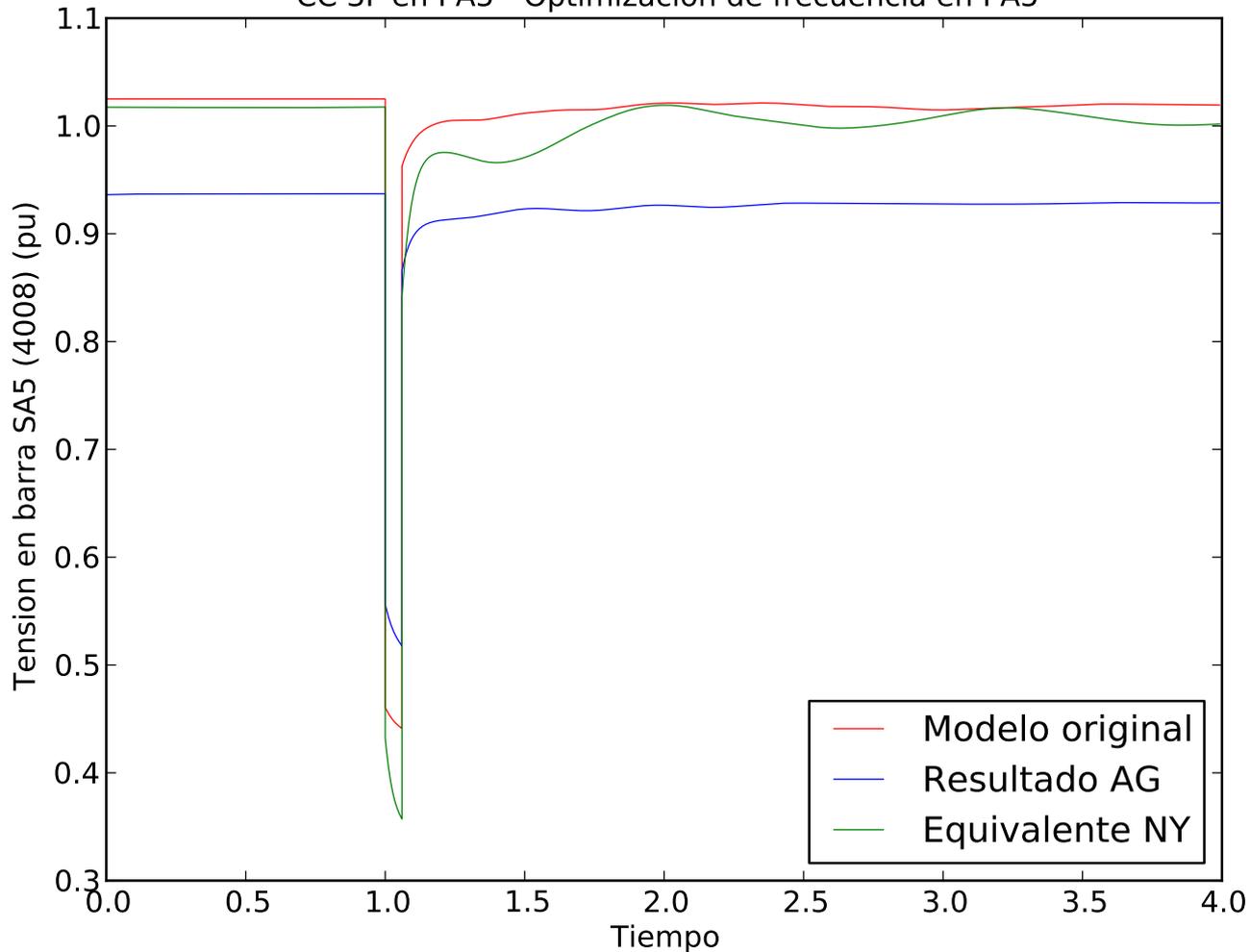
CC 3F en PA5 - Optimizacion de frecuencia en PA5



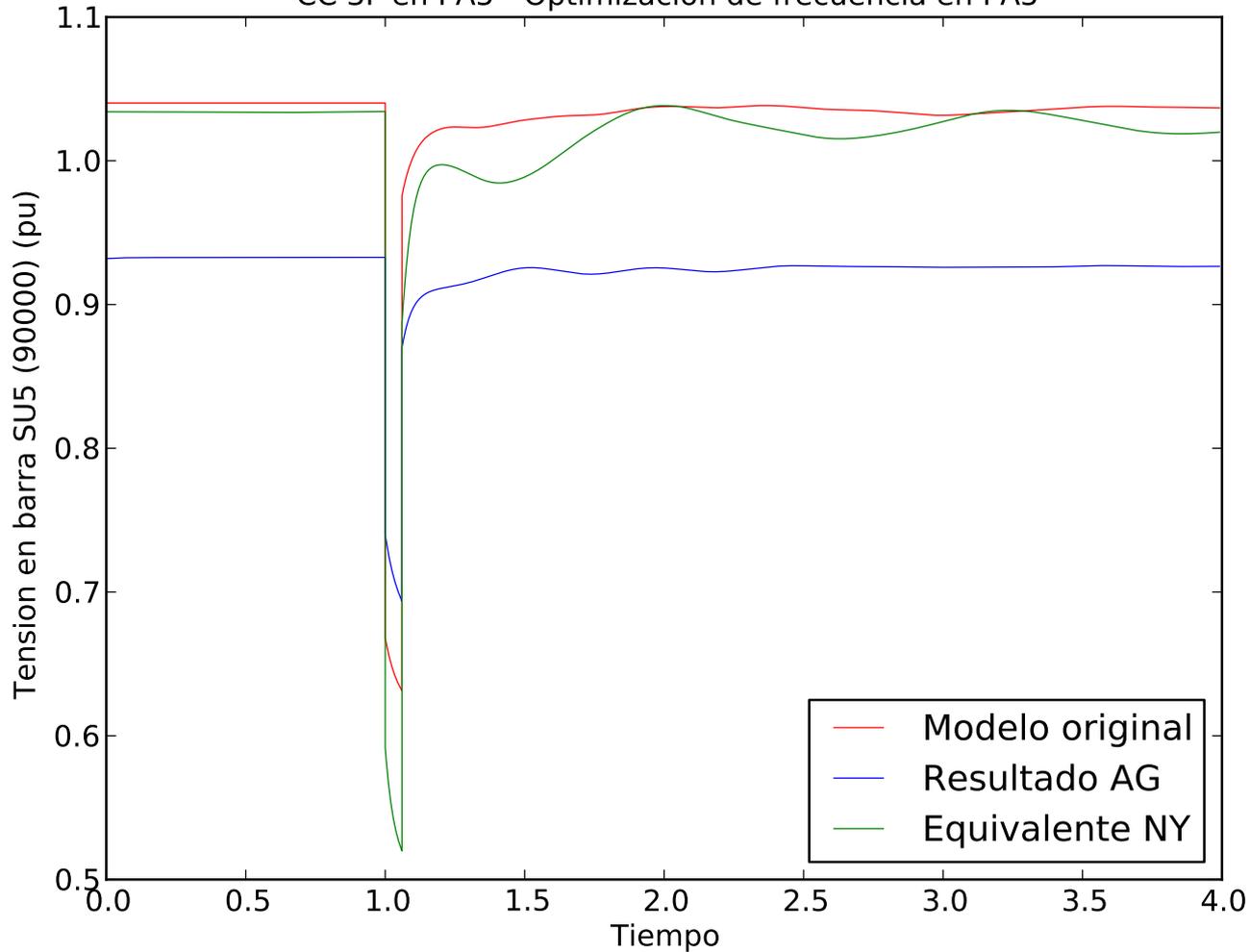
CC 3F en PA5 - Optimizacion de frecuencia en PA5



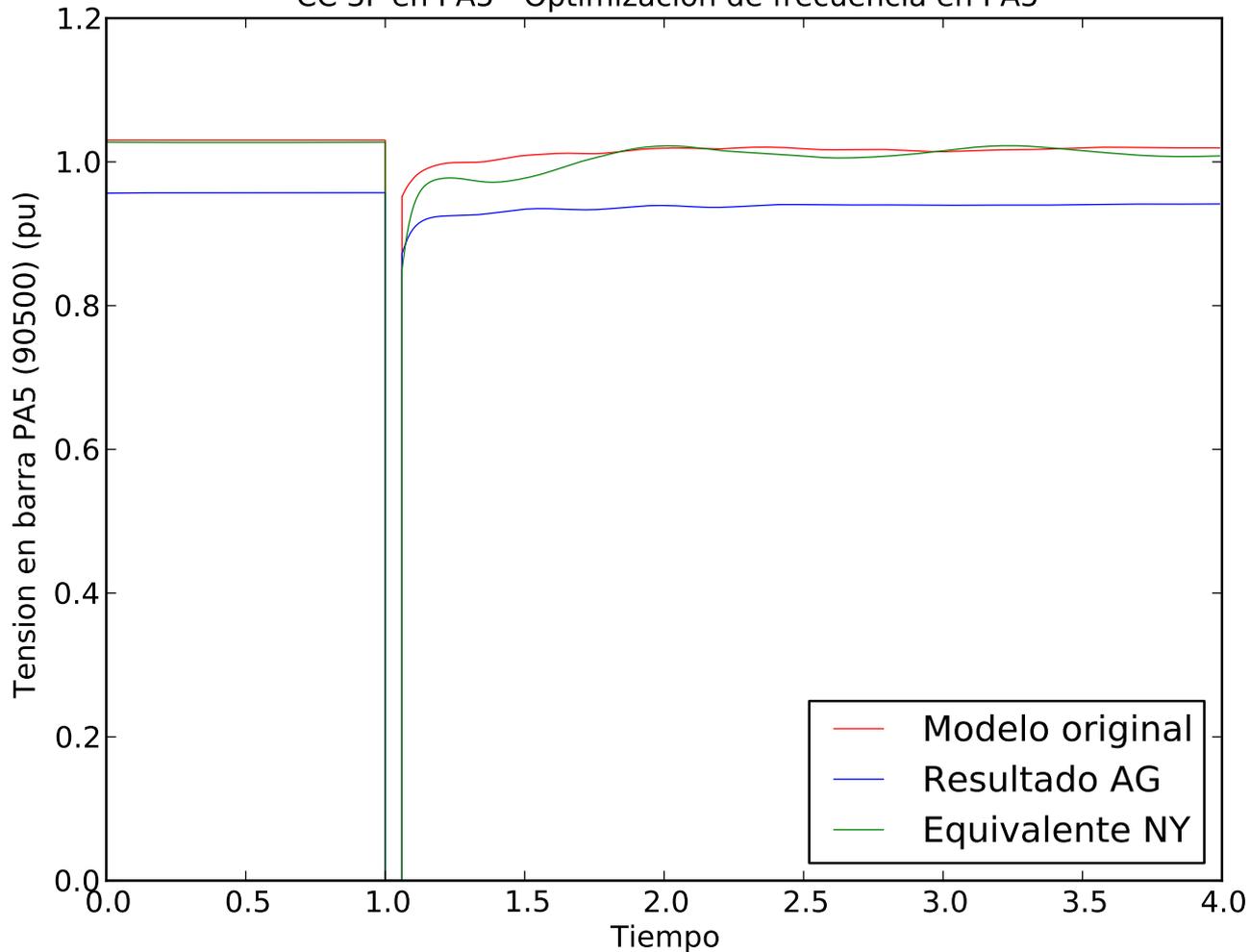
CC 3F en PA5 - Optimizacion de frecuencia en PA5



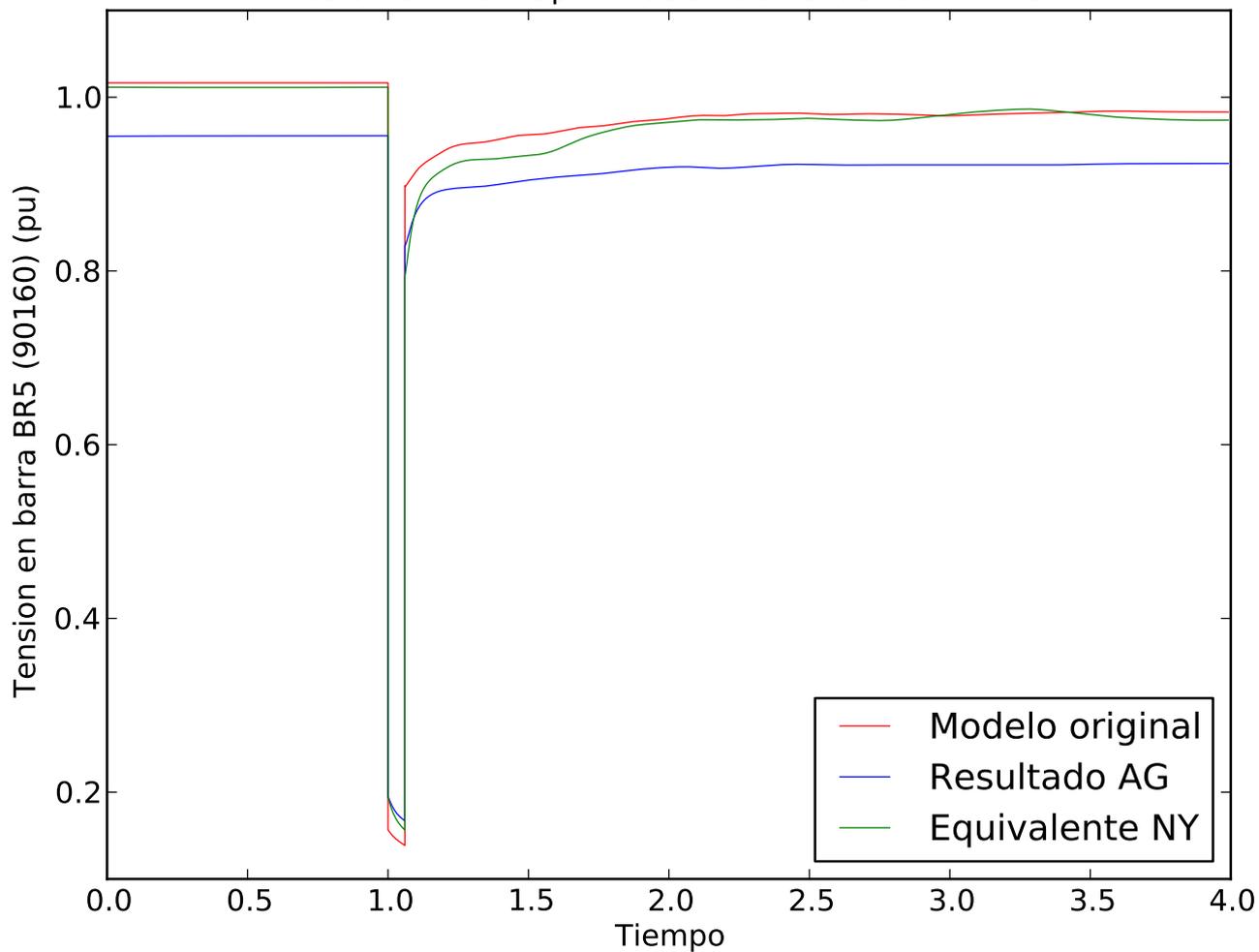
CC 3F en PA5 - Optimizacion de frecuencia en PA5



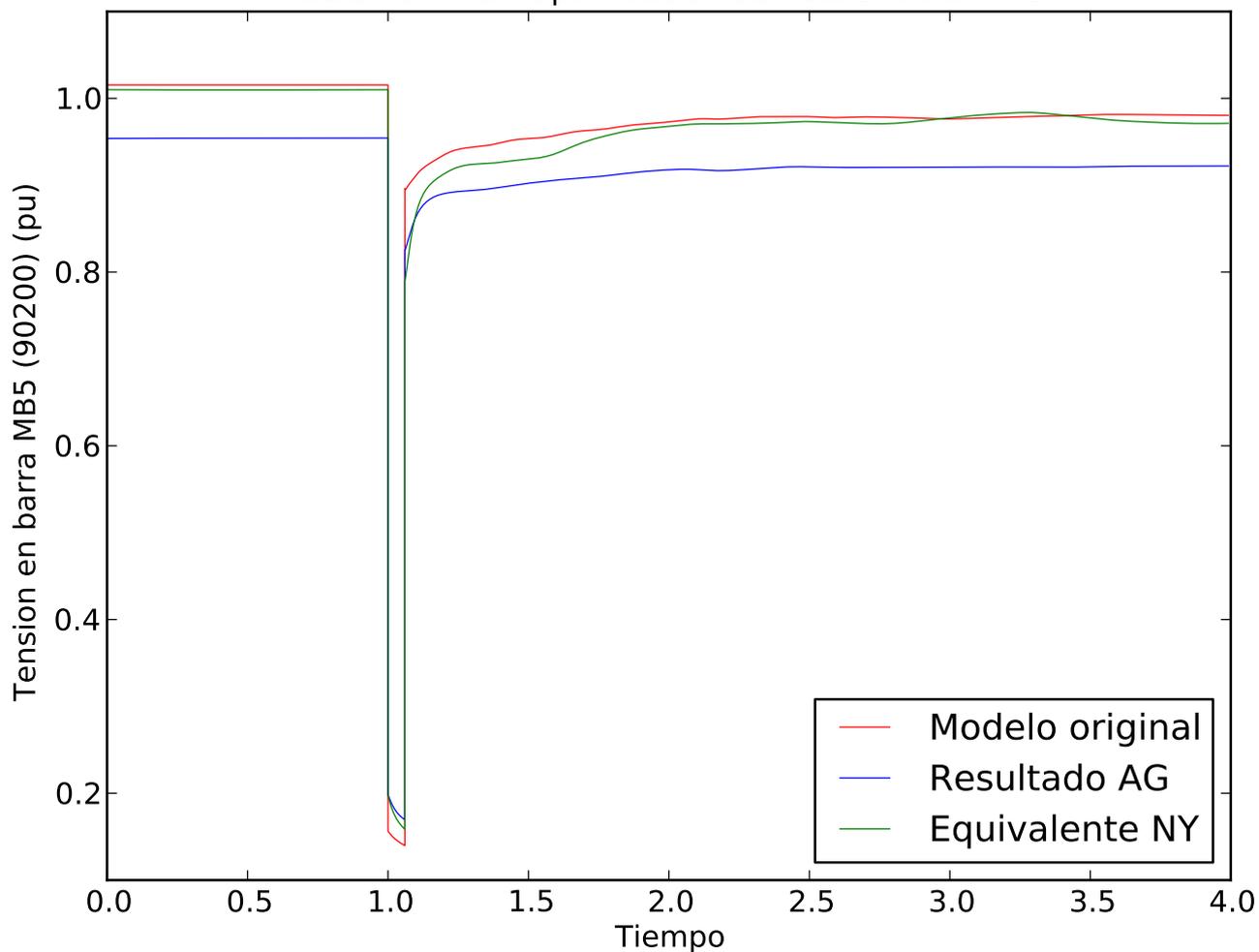
CC 3F en PA5 - Optimizacion de frecuencia en PA5



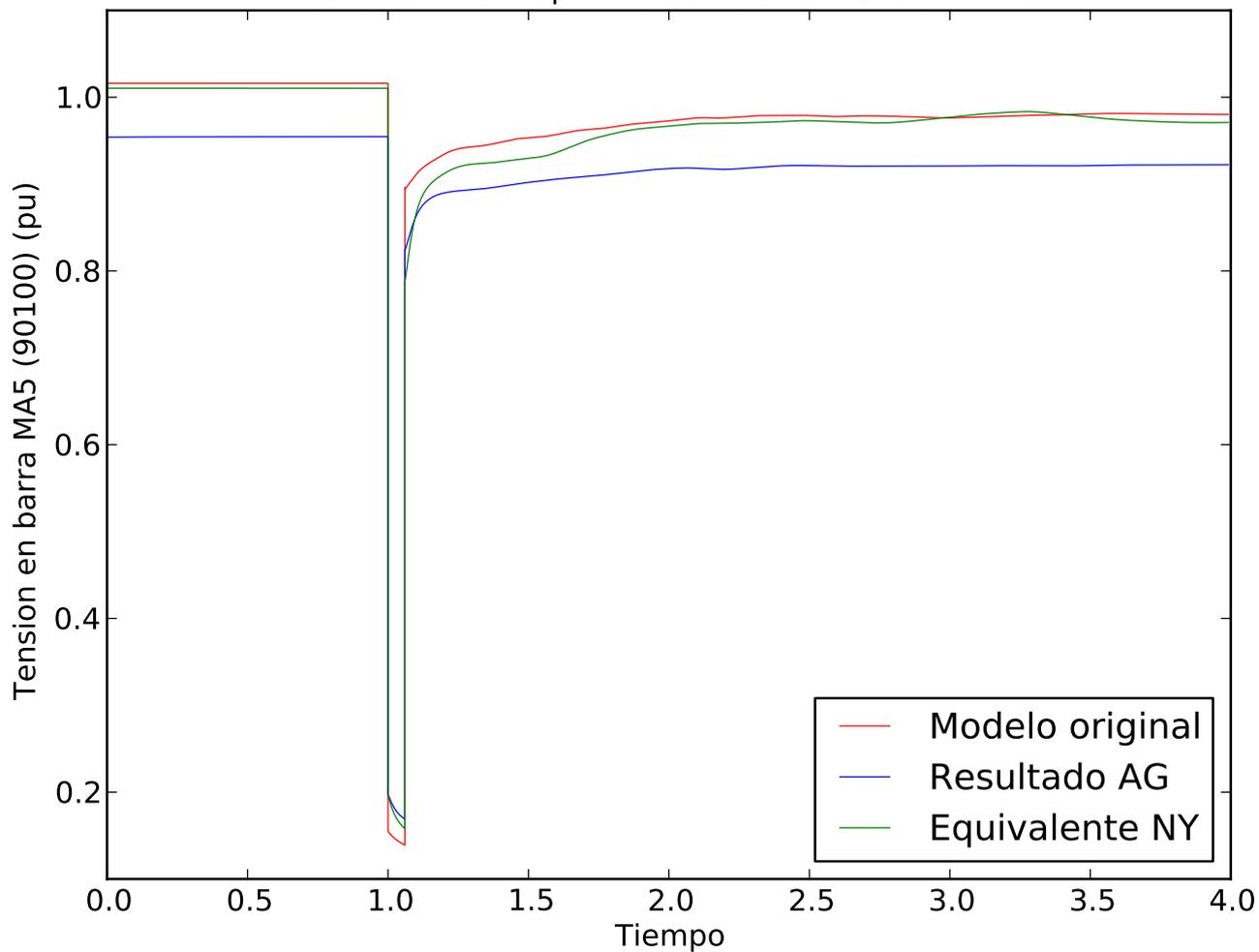
CC 3F en PA5 - Optimizacion de frecuencia en PA5



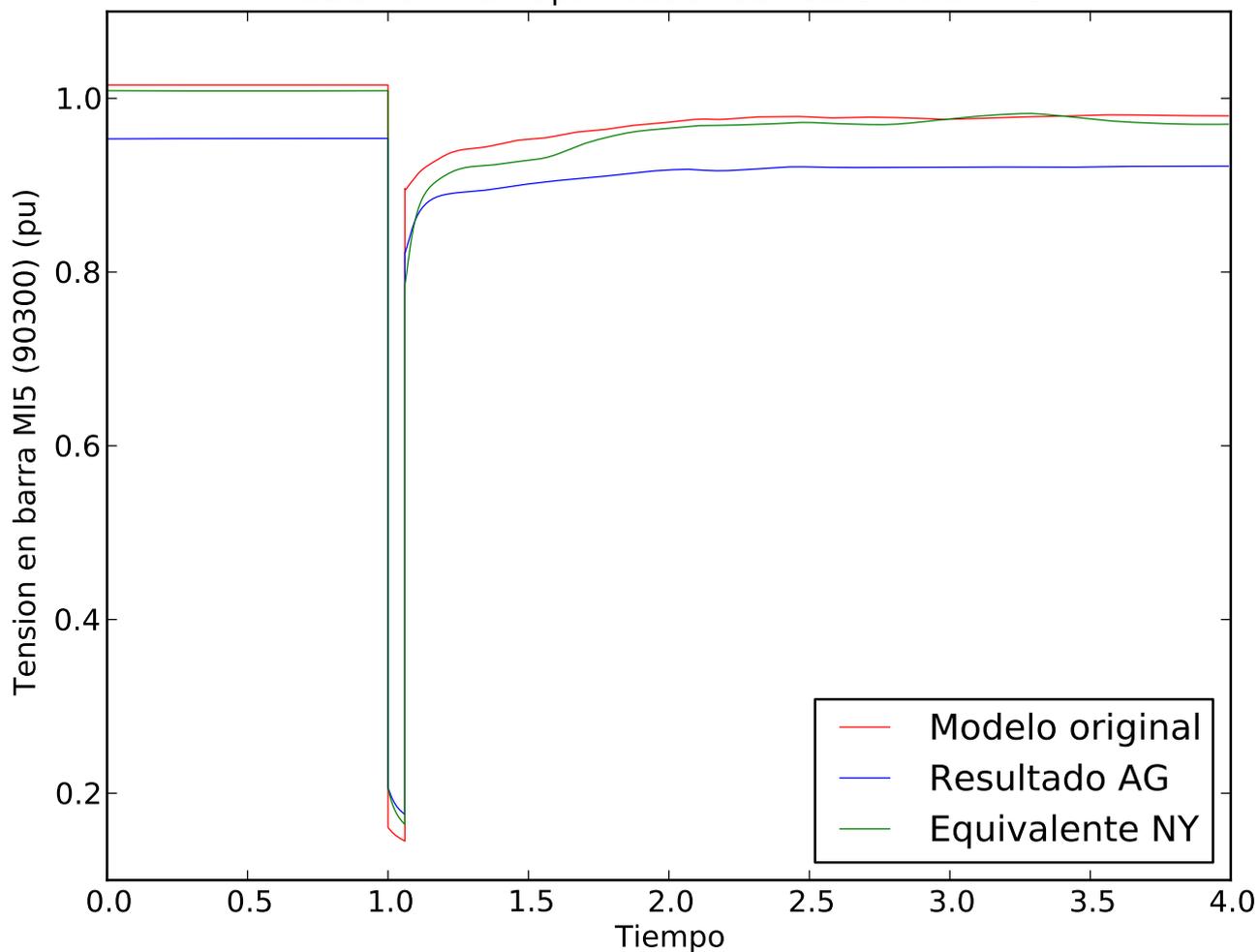
CC 3F en PA5 - Optimizacion de frecuencia en PA5



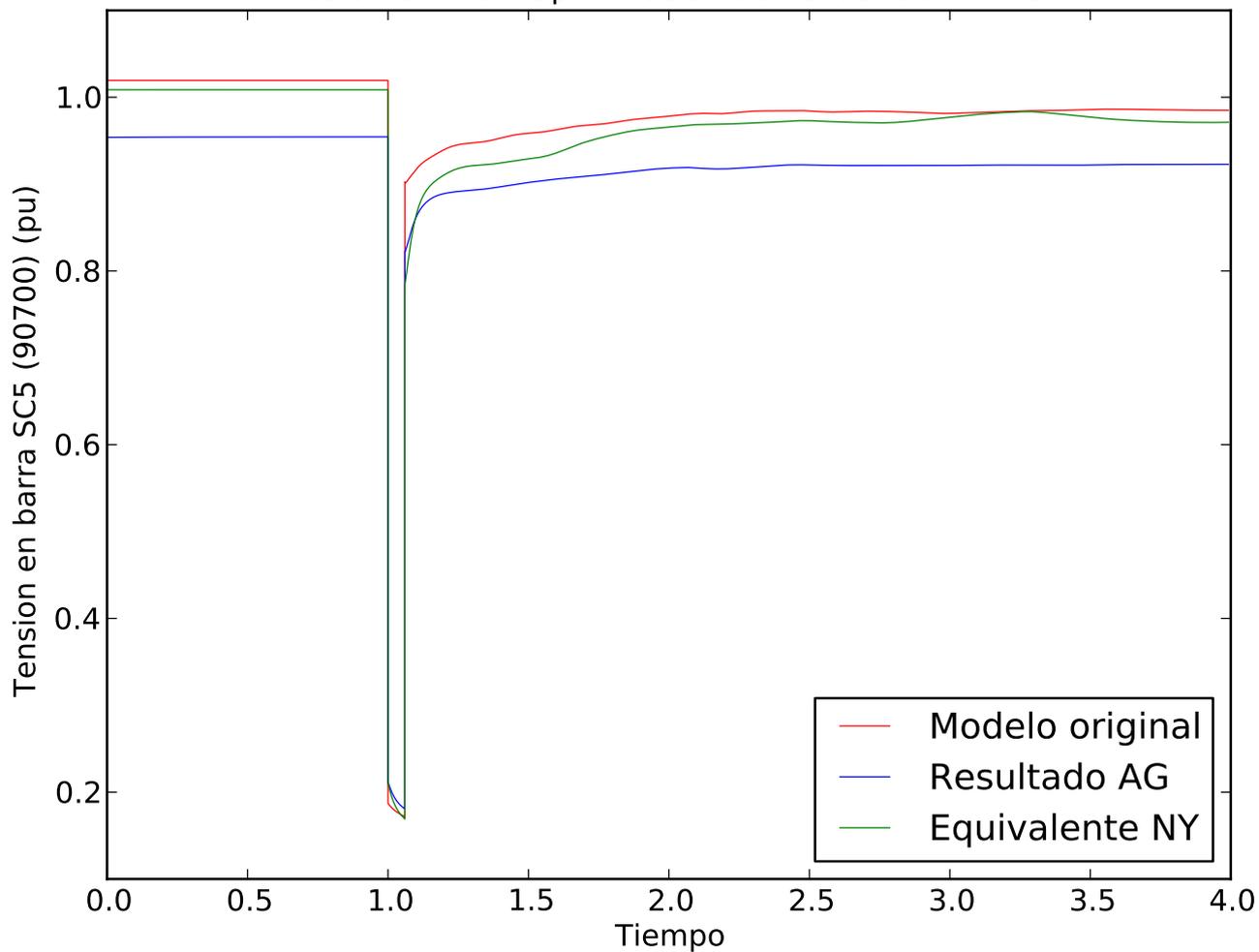
CC 3F en PA5 - Optimizacion de frecuencia en PA5



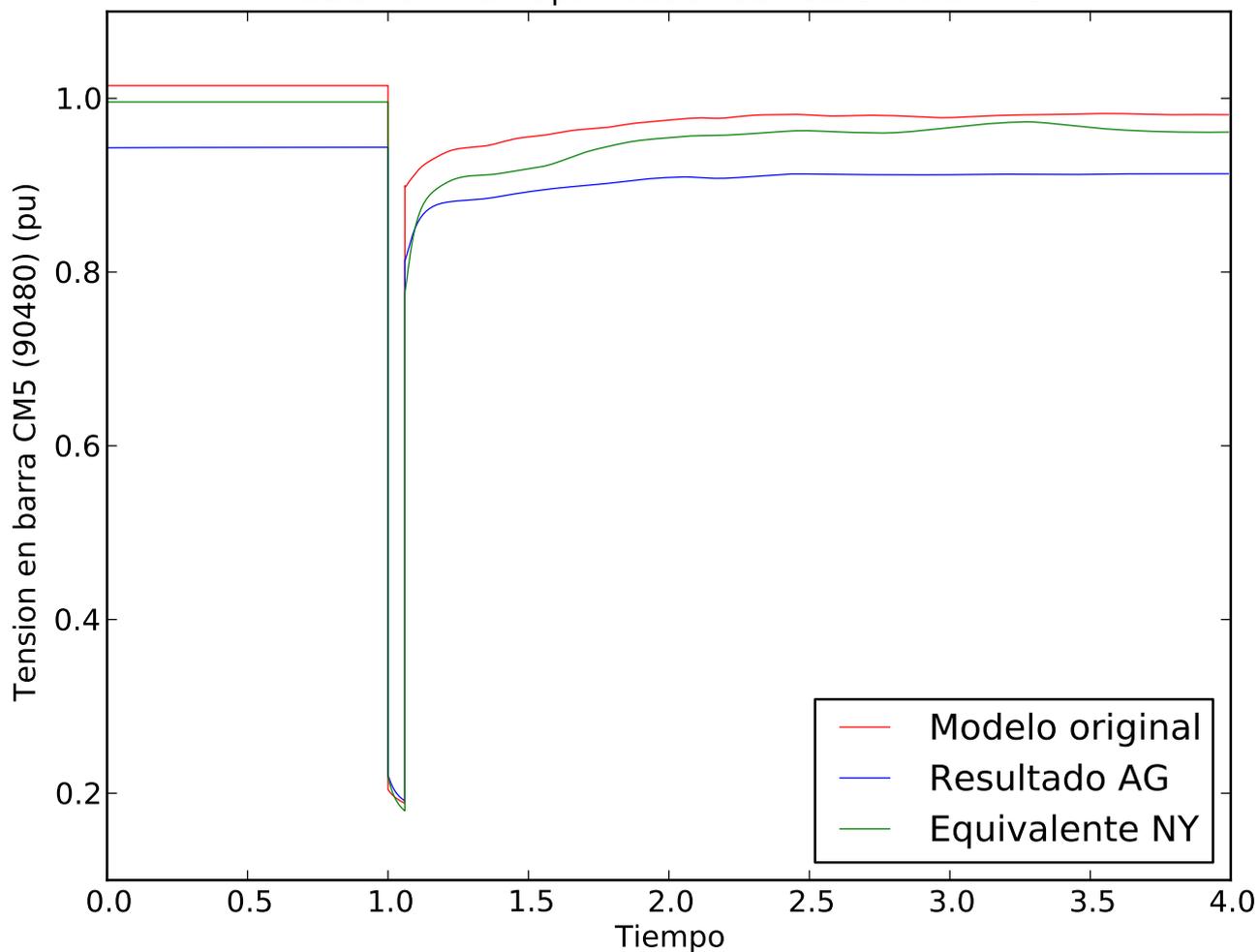
CC 3F en PA5 - Optimizacion de frecuencia en PA5



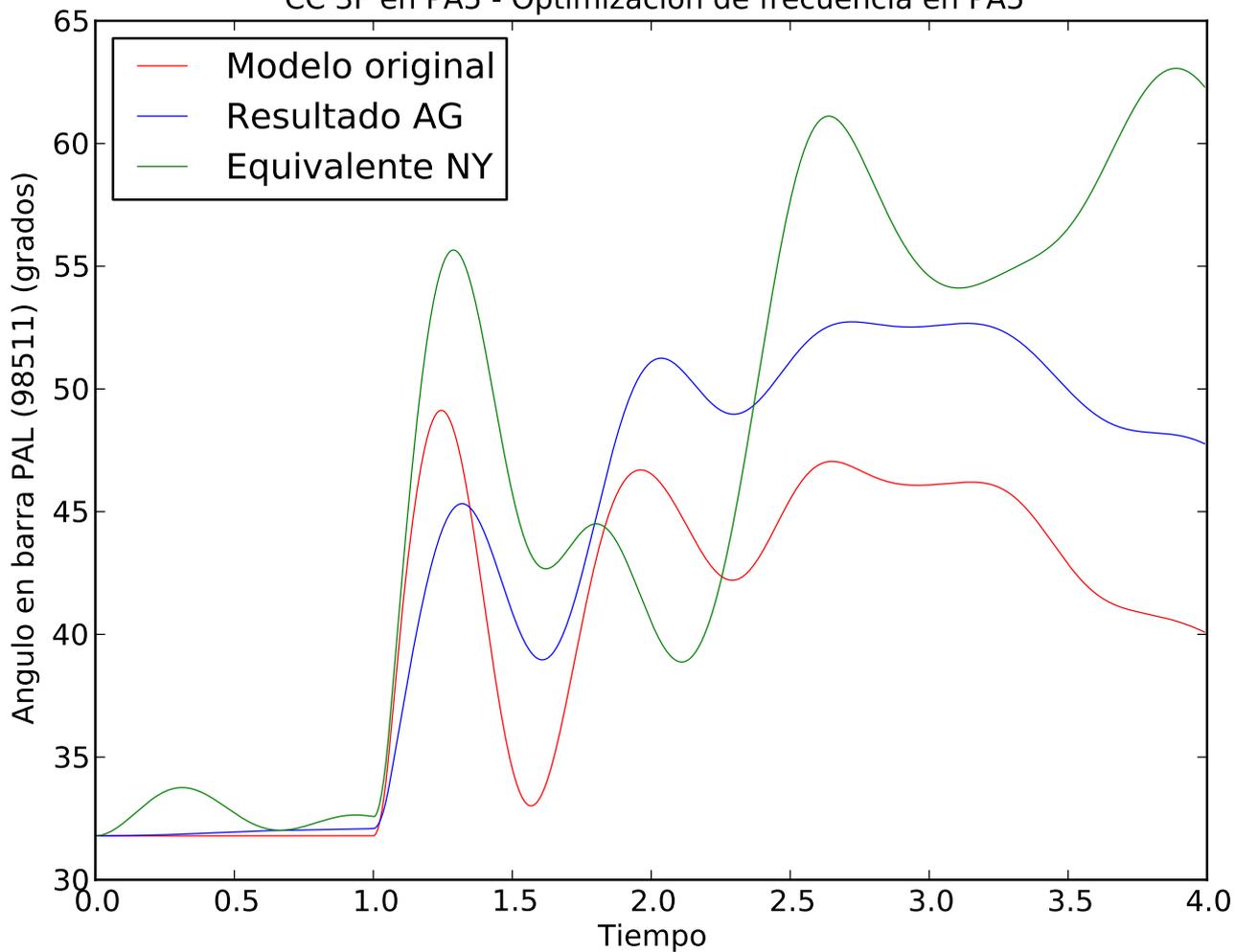
CC 3F en PA5 - Optimizacion de frecuencia en PA5



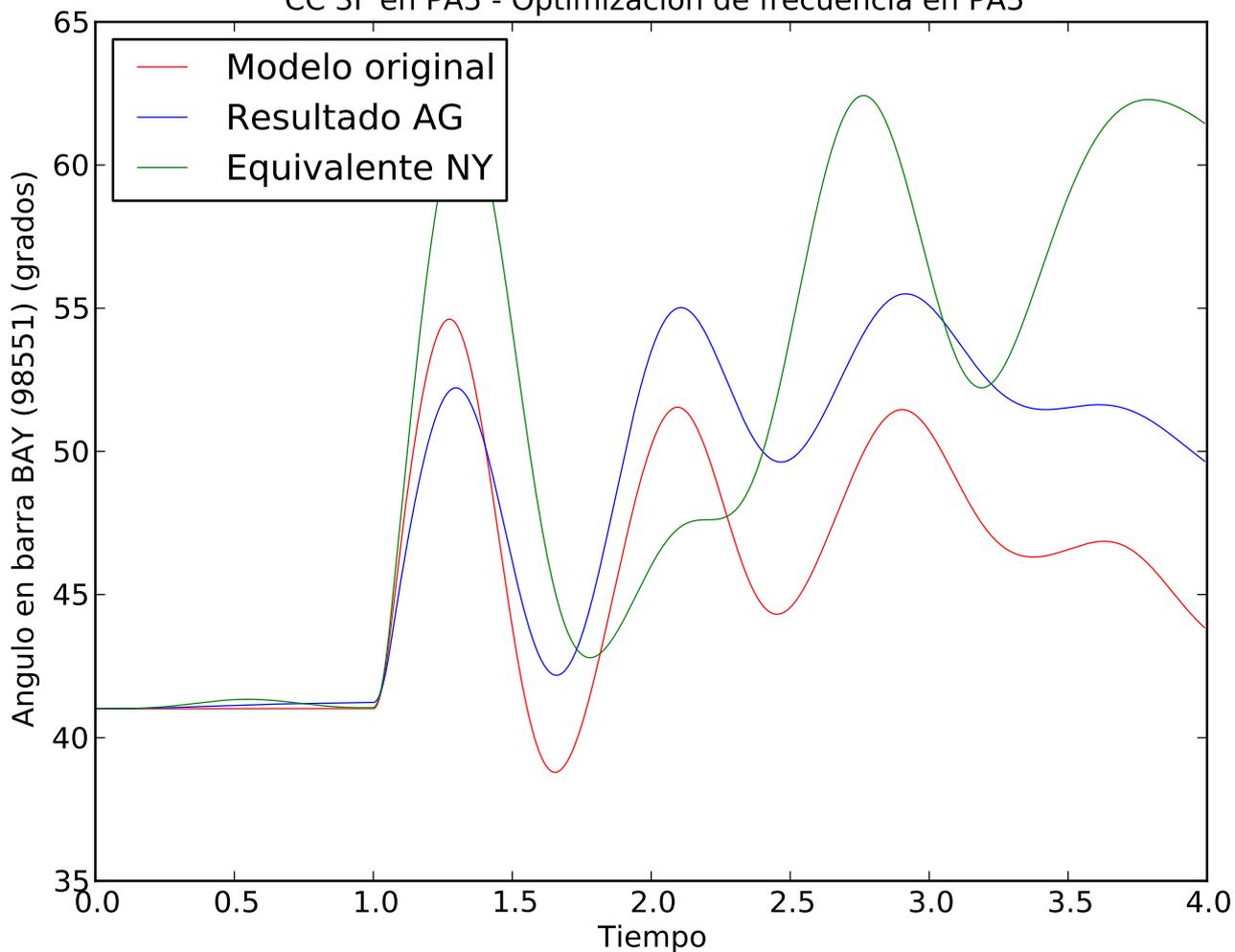
CC 3F en PA5 - Optimizacion de frecuencia en PA5



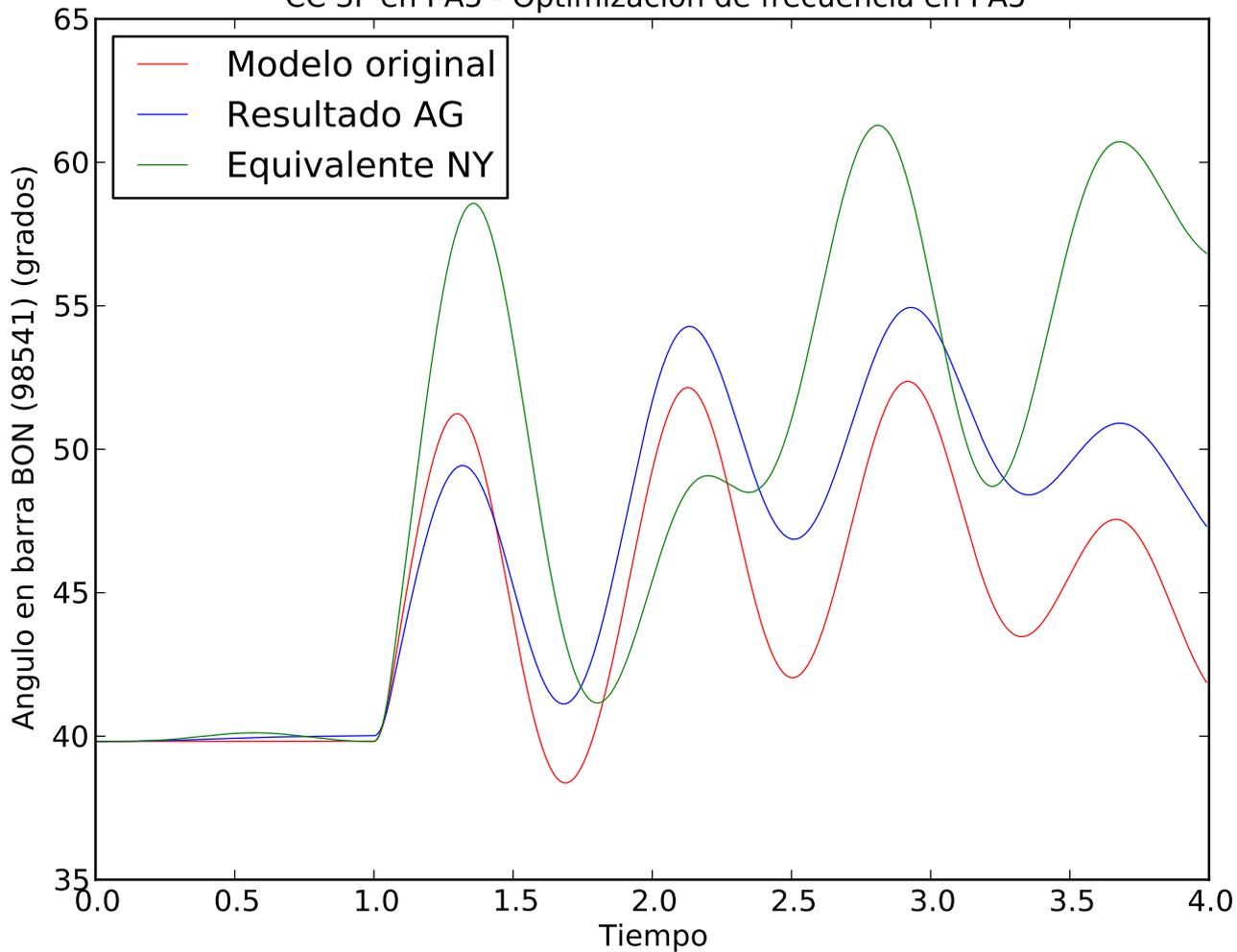
CC 3F en PA5 - Optimizacion de frecuencia en PA5



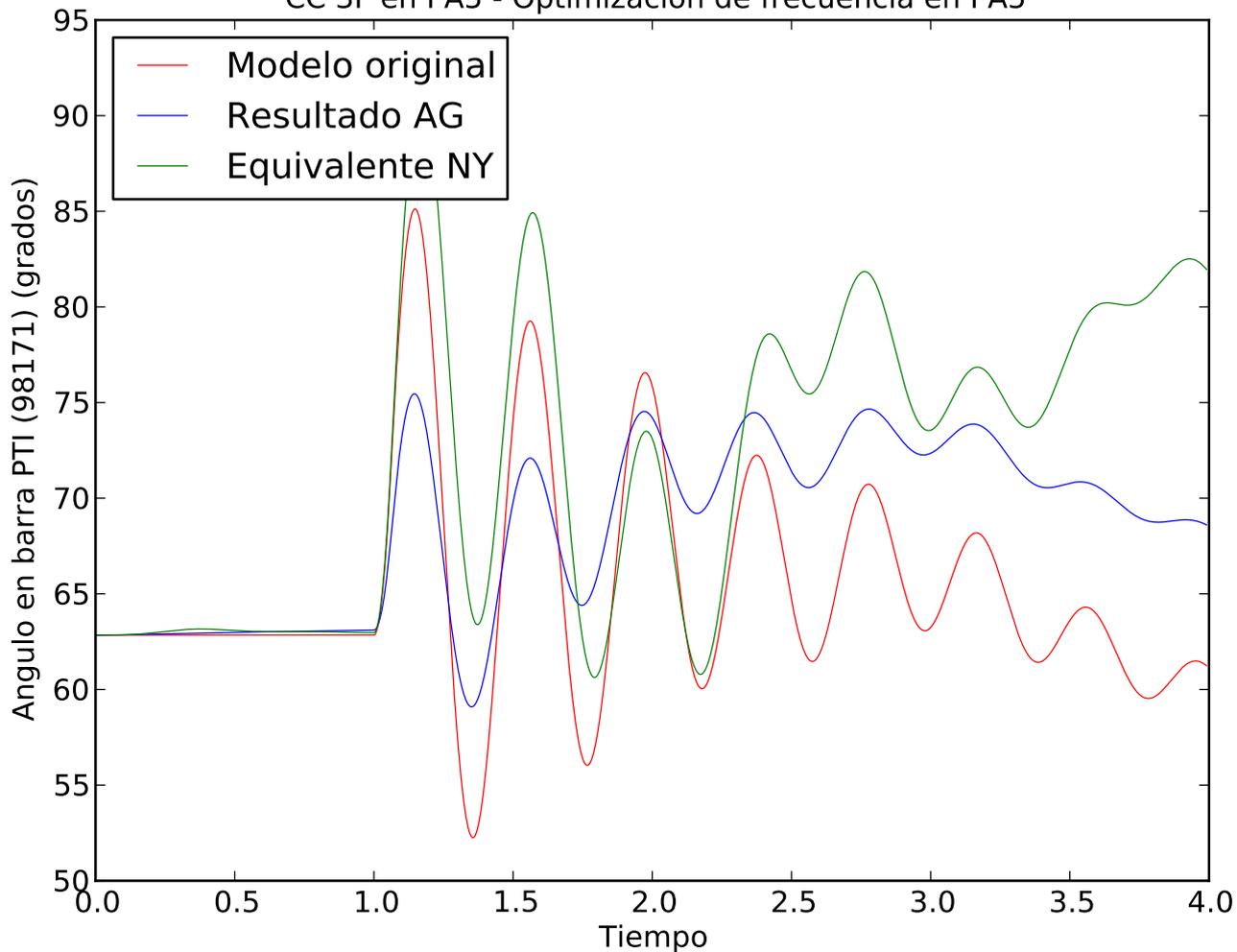
CC 3F en PA5 - Optimizacion de frecuencia en PA5



CC 3F en PA5 - Optimizacion de frecuencia en PA5

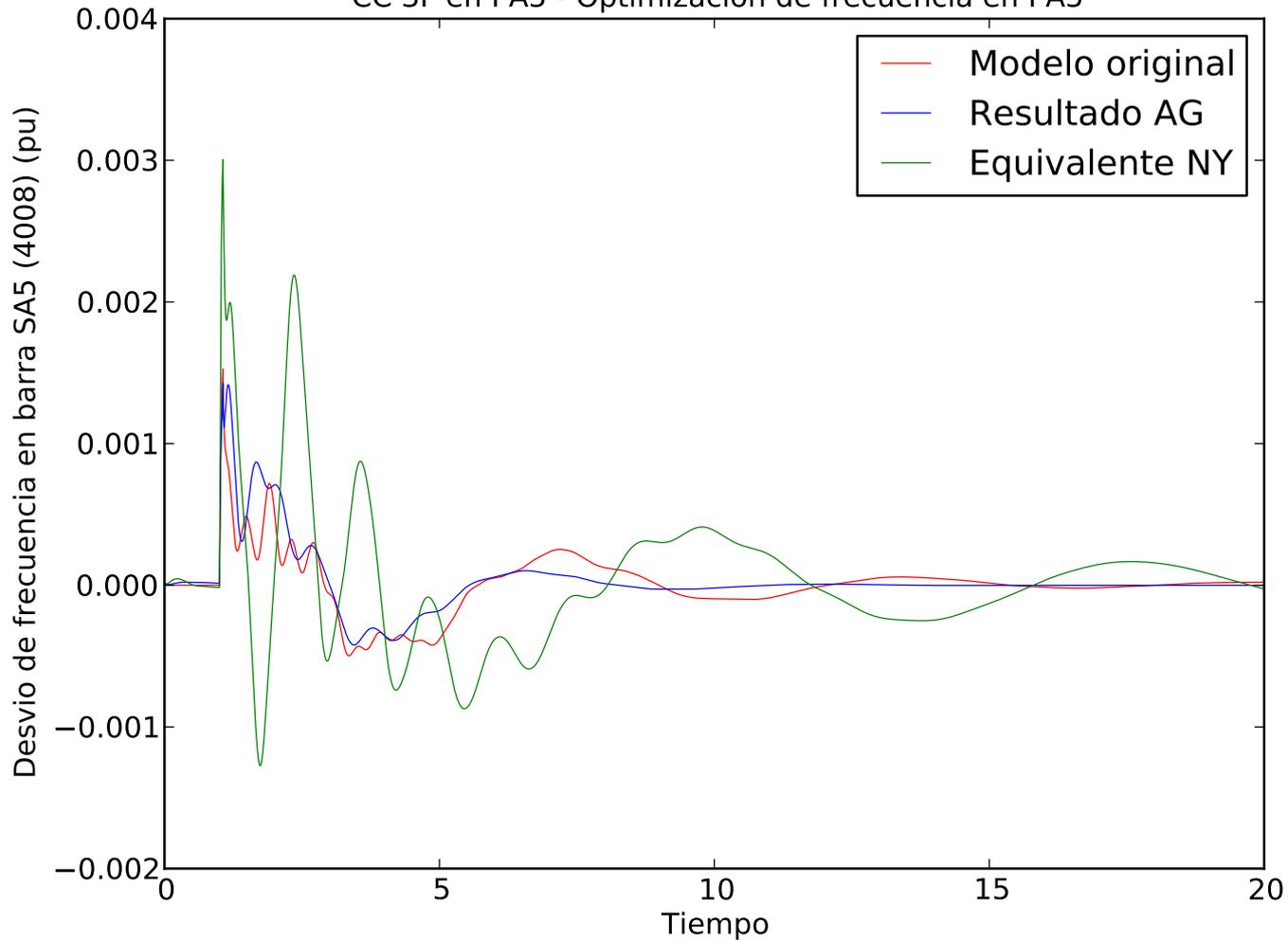


CC 3F en PA5 - Optimizacion de frecuencia en PA5

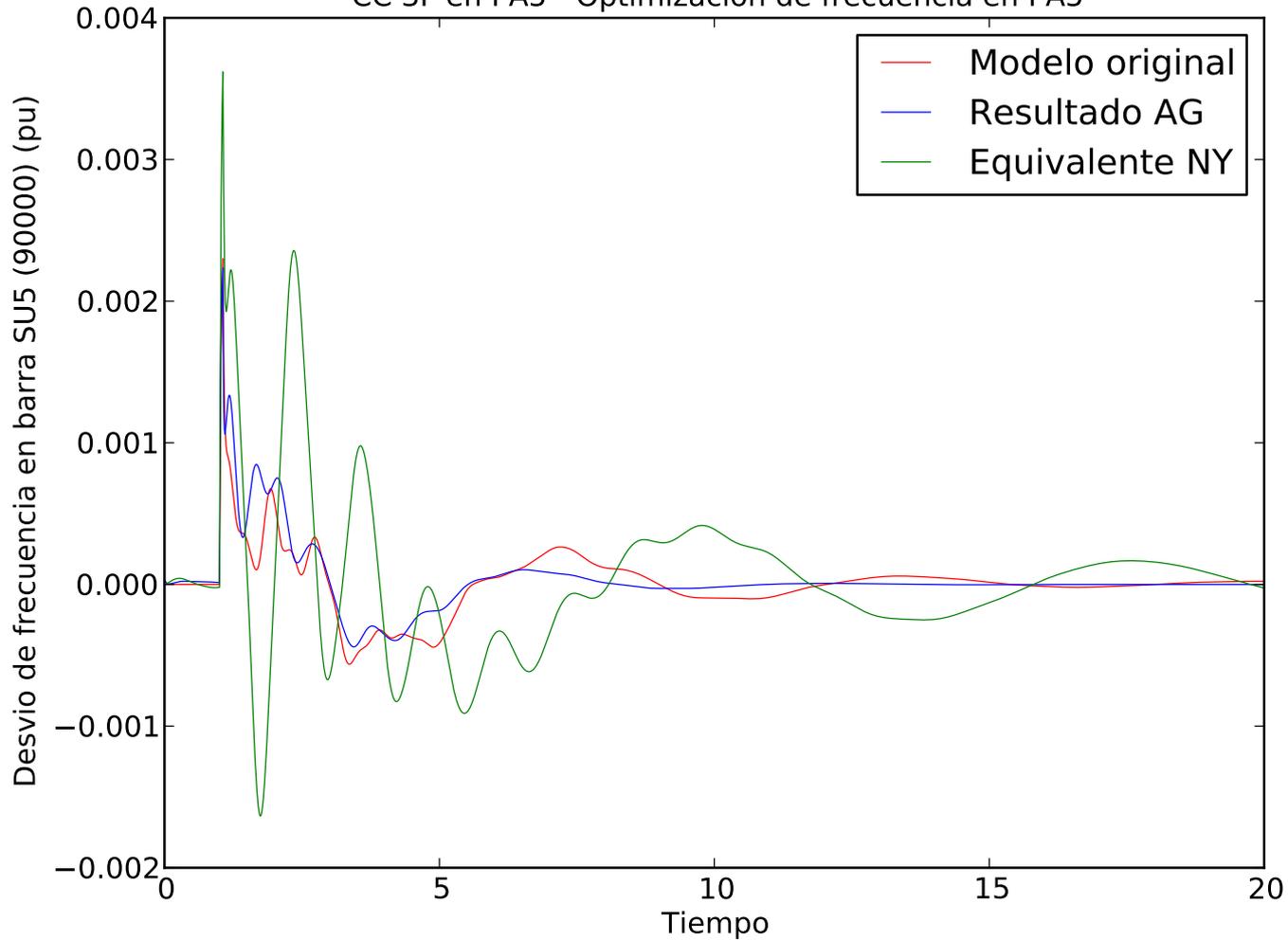


**C.3. Simulación 2: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando frecuencia en PA5 en 4 segundos. Comportamiento en 20 segundos**

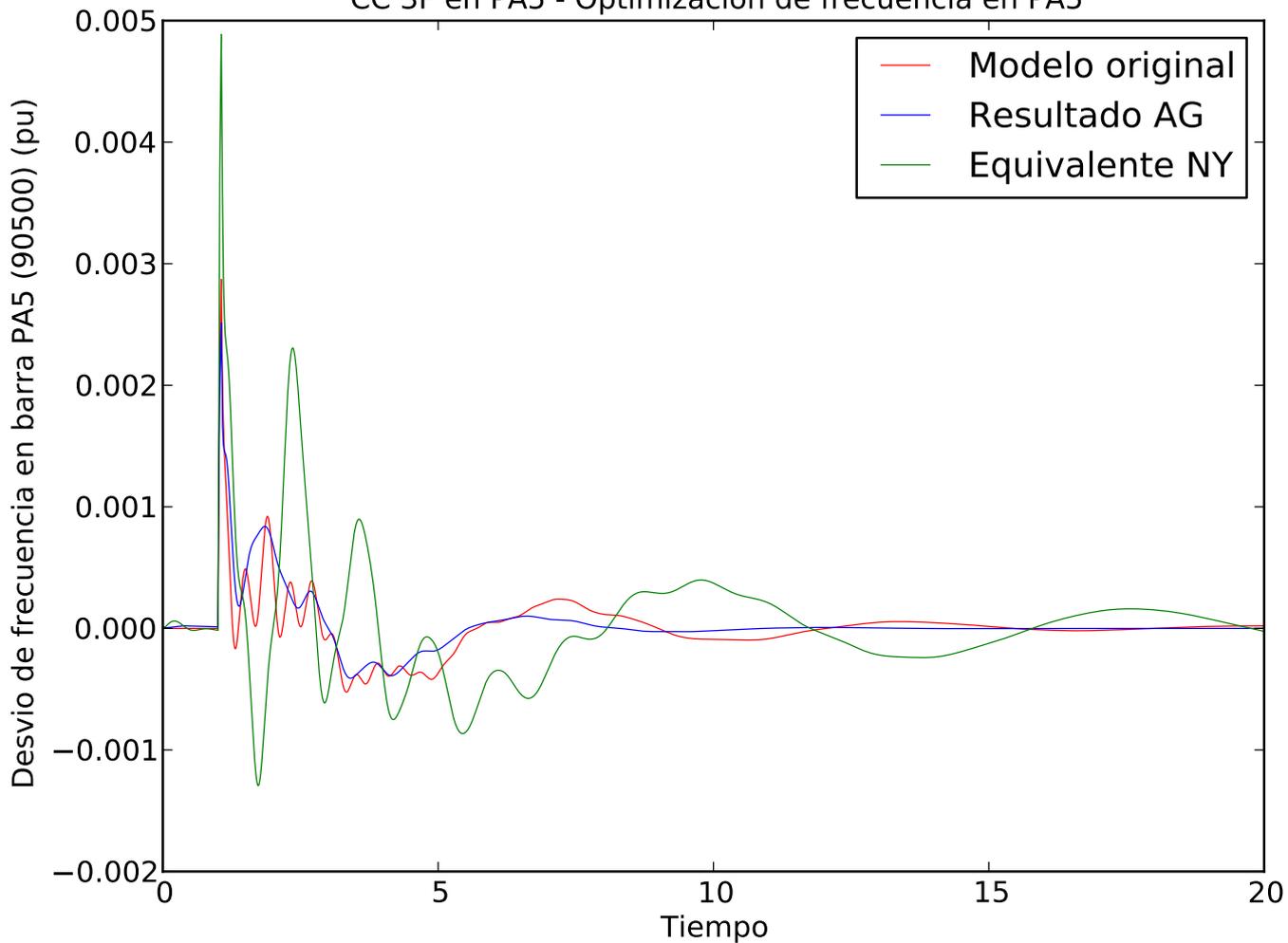
CC 3F en PA5 - Optimizacion de frecuencia en PA5



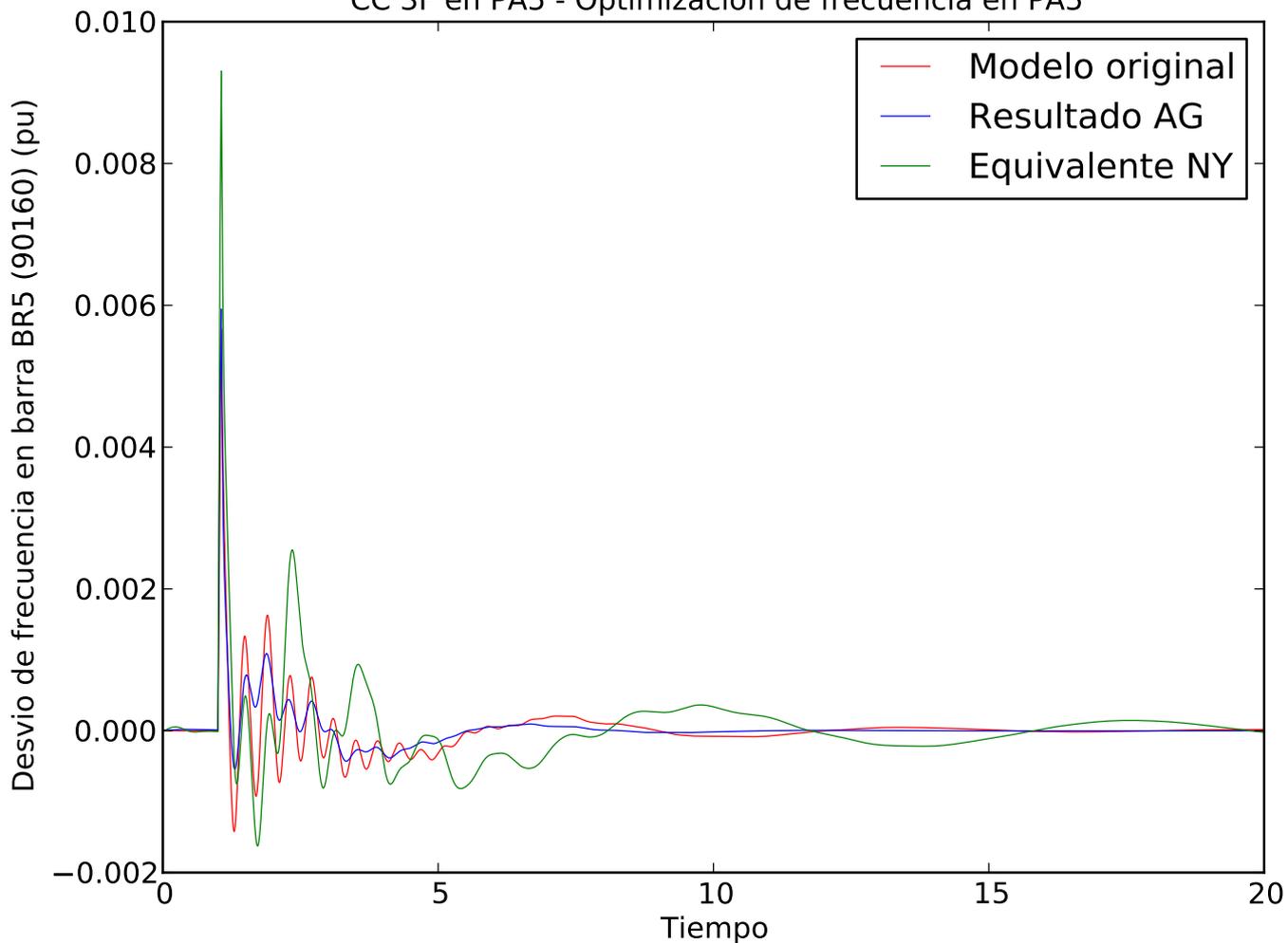
CC 3F en PA5 - Optimizacion de frecuencia en PA5



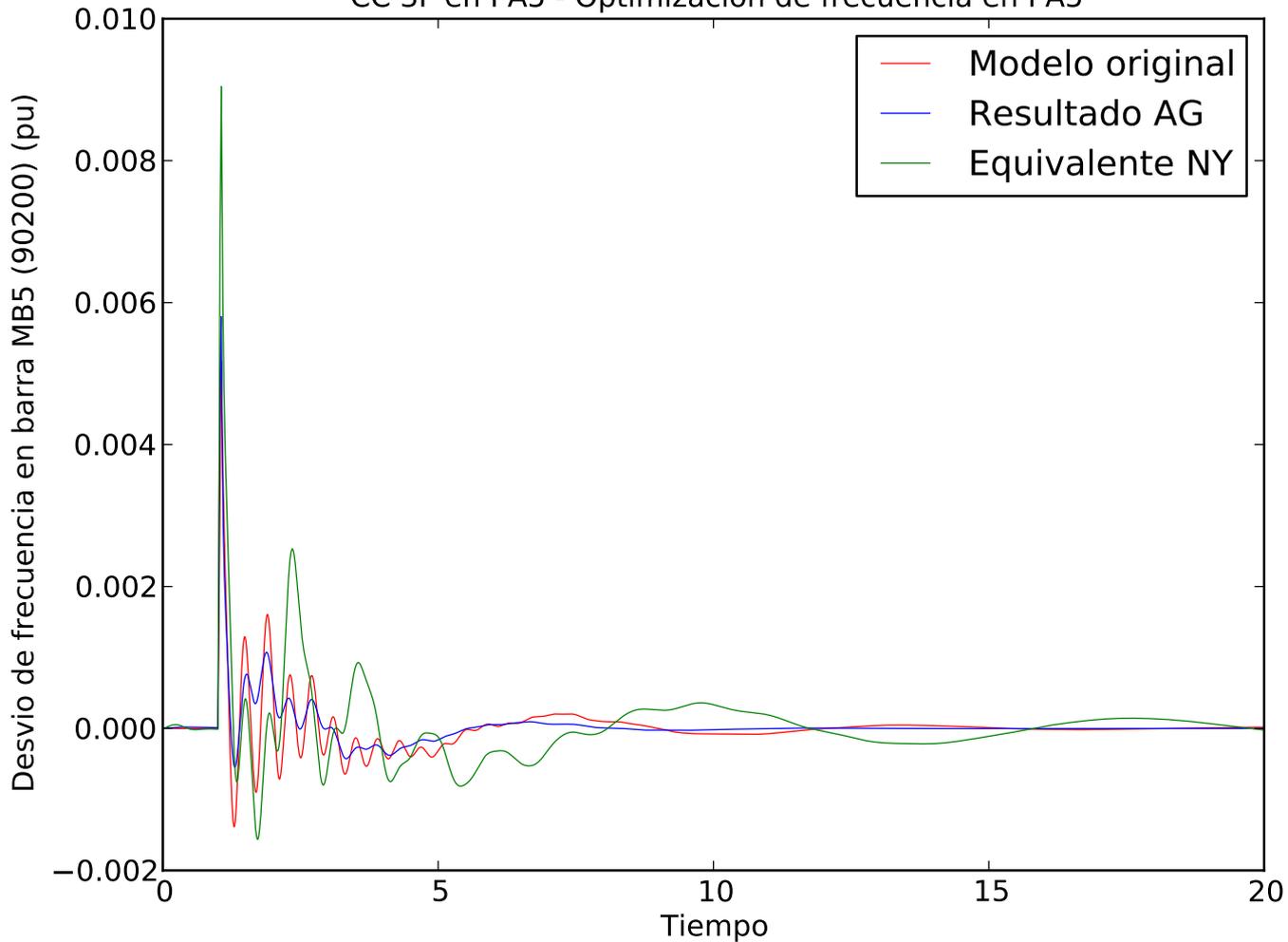
CC 3F en PA5 - Optimizacion de frecuencia en PA5



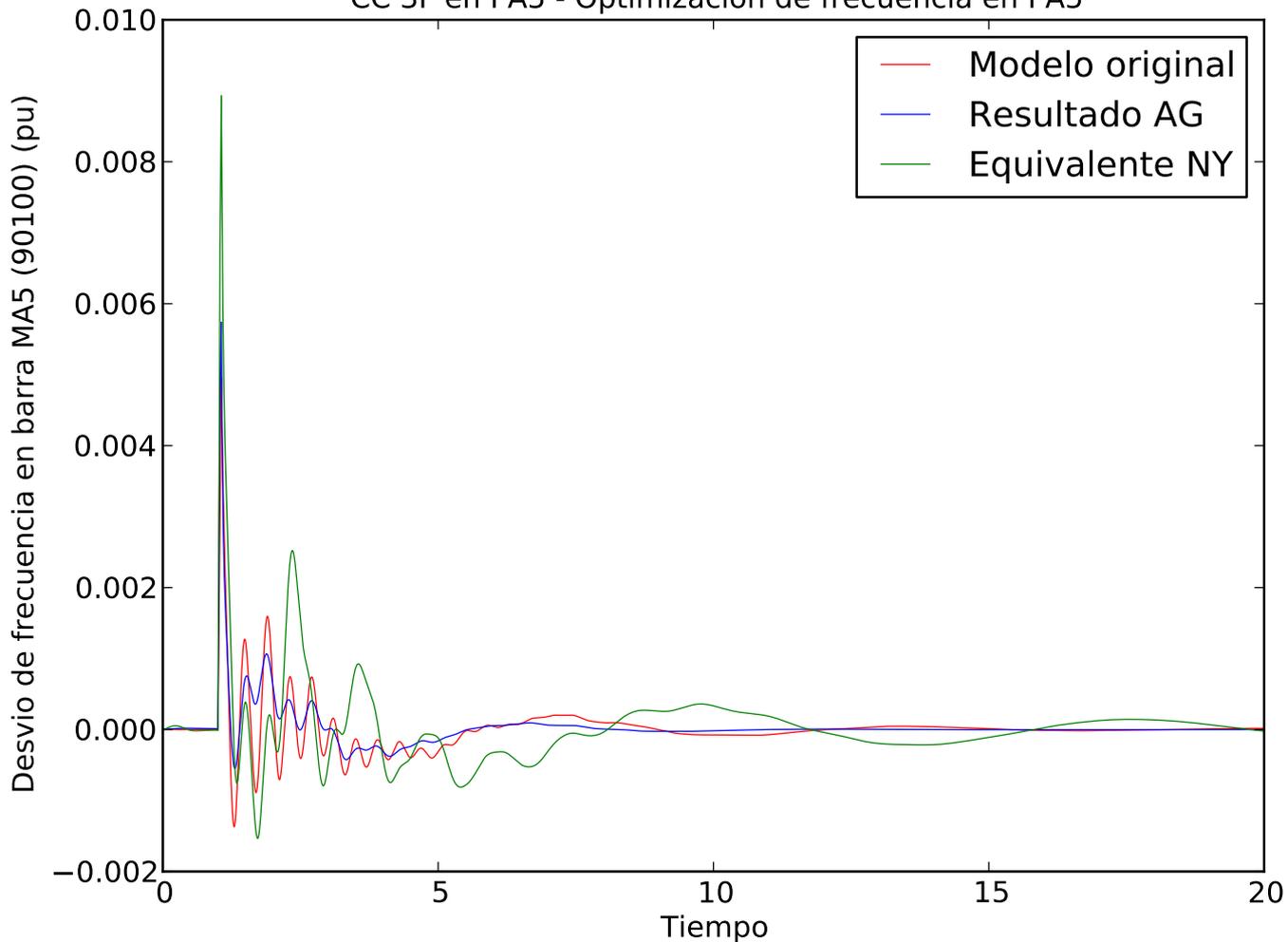
CC 3F en PA5 - Optimizacion de frecuencia en PA5



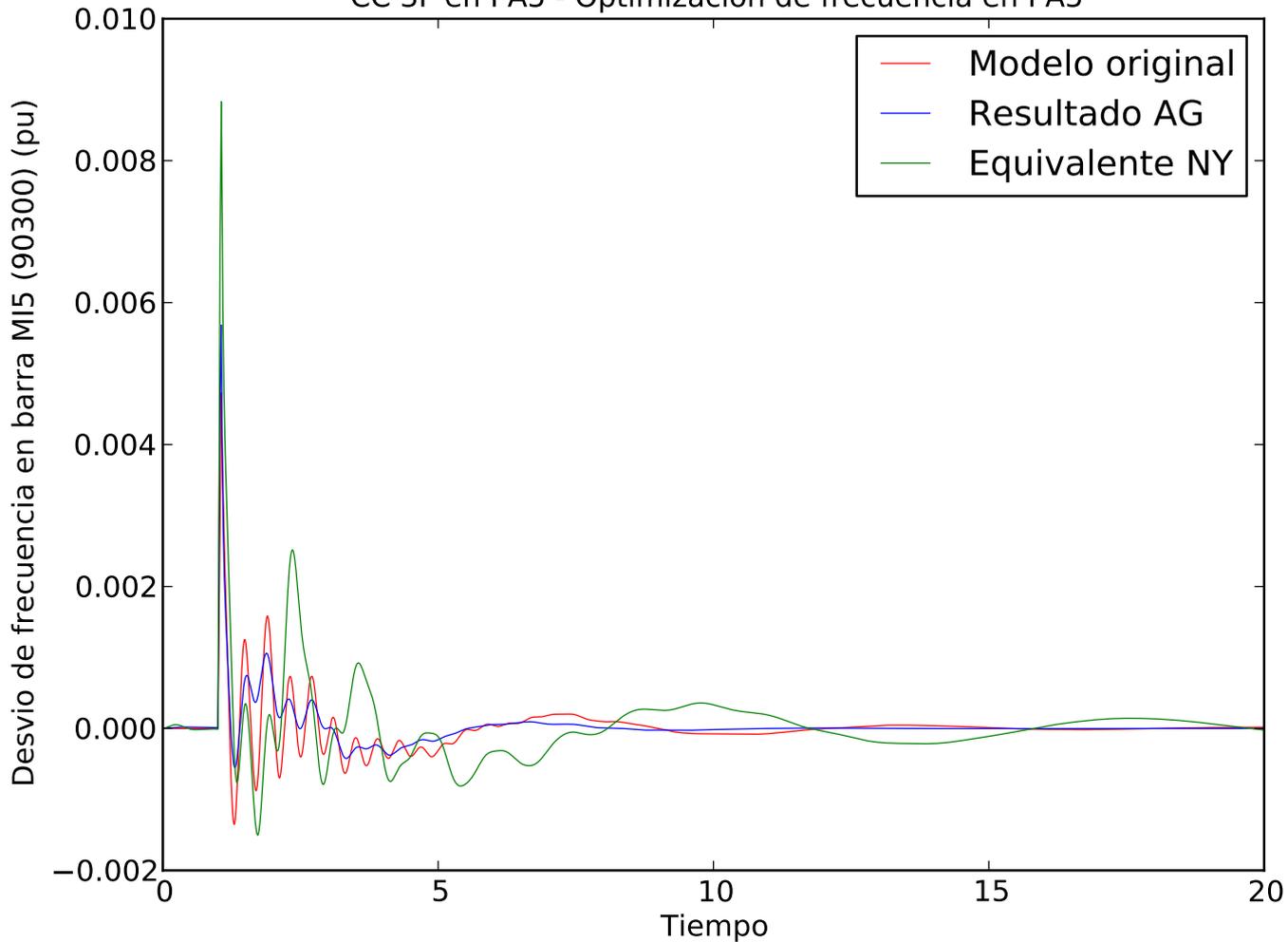
CC 3F en PA5 - Optimizacion de frecuencia en PA5



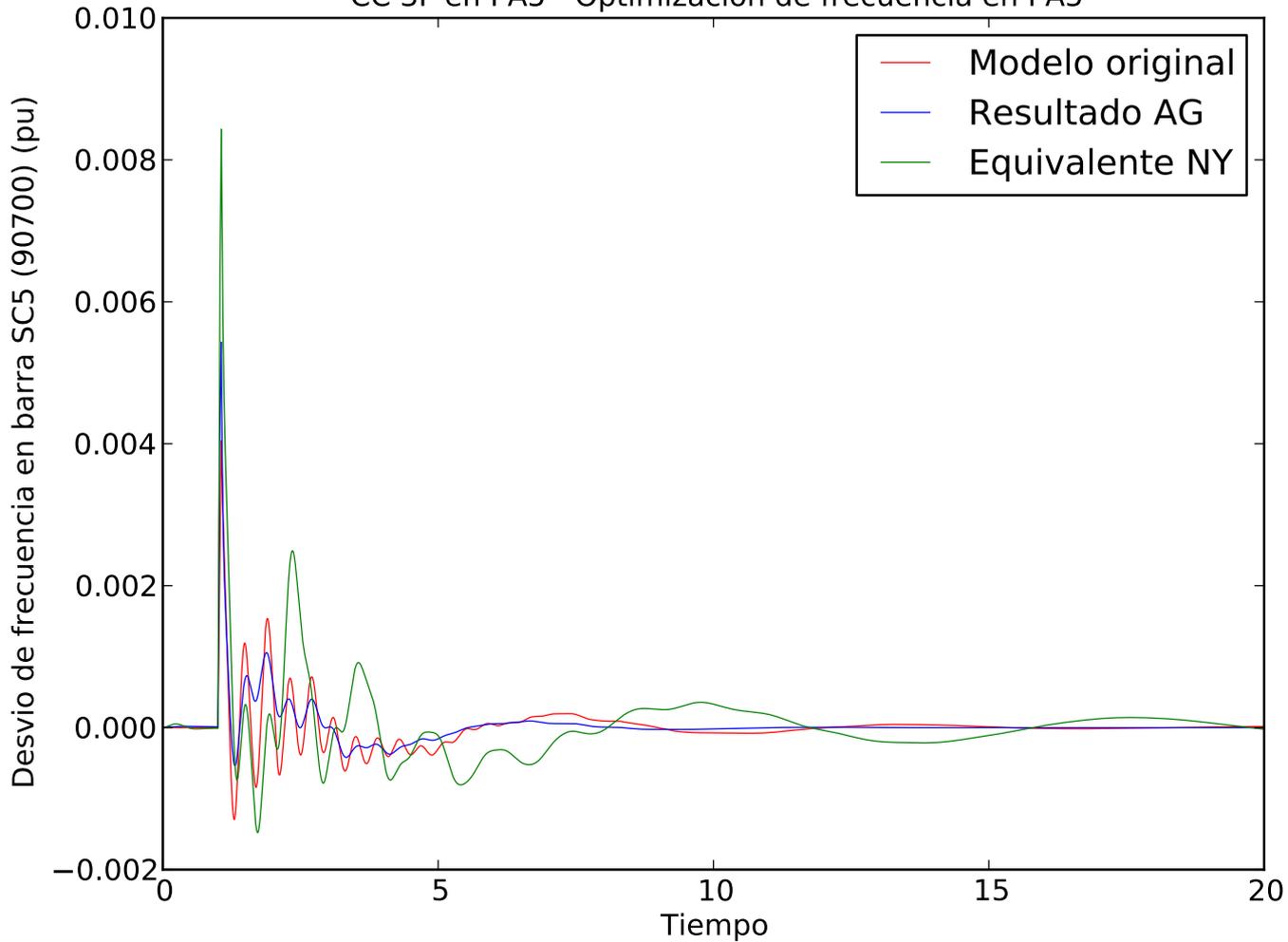
CC 3F en PA5 - Optimizacion de frecuencia en PA5



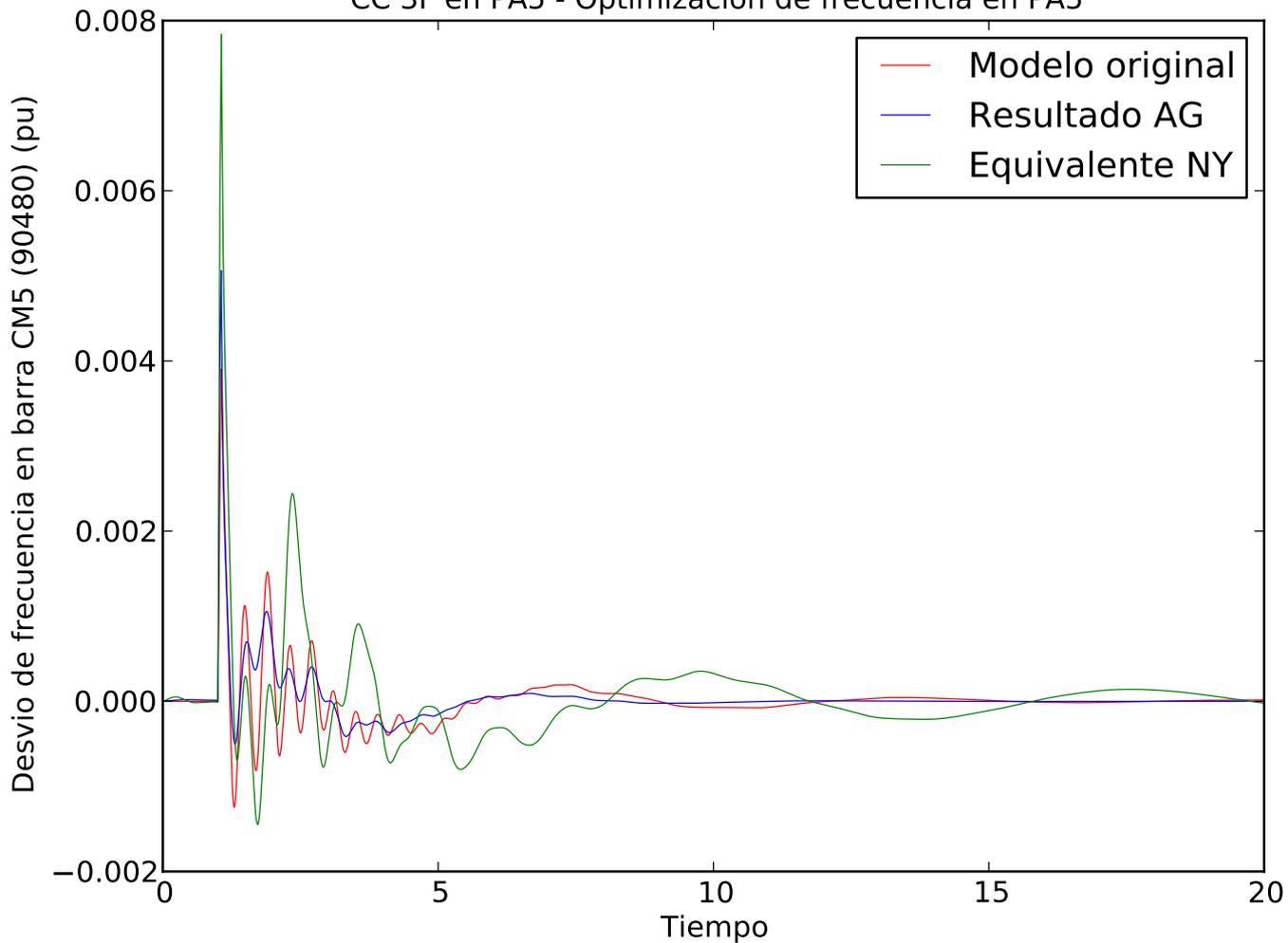
CC 3F en PA5 - Optimizacion de frecuencia en PA5



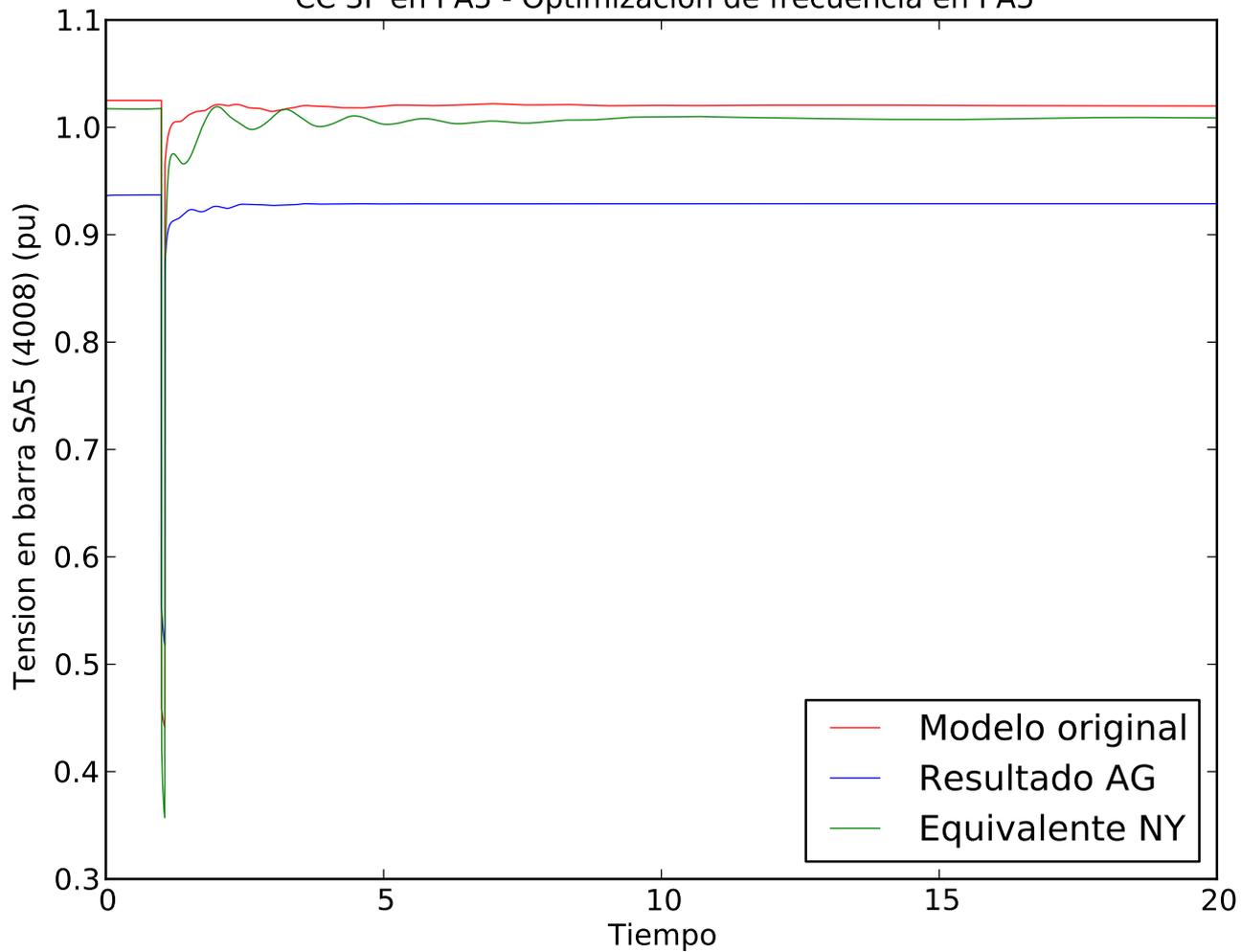
CC 3F en PA5 - Optimizacion de frecuencia en PA5



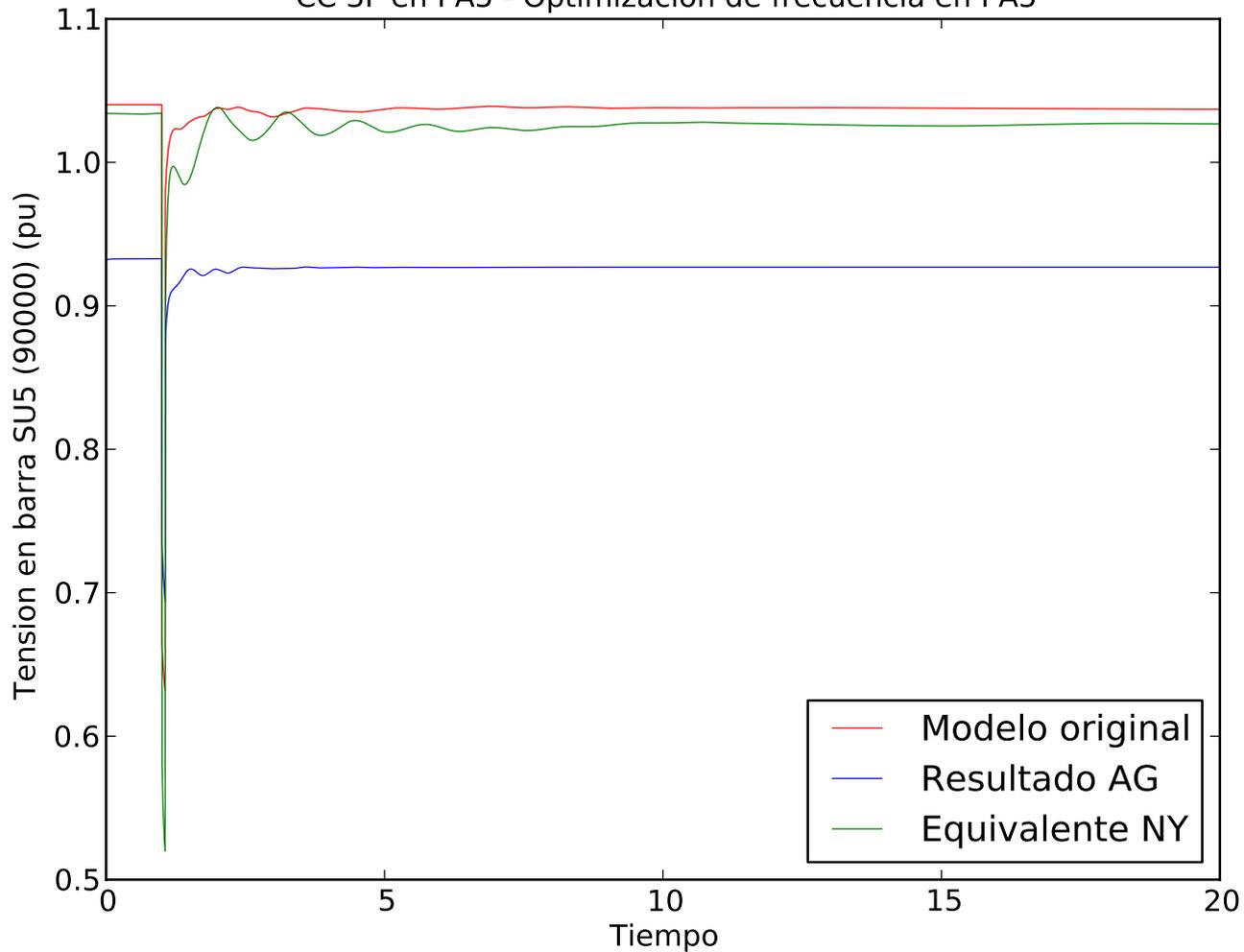
CC 3F en PA5 - Optimizacion de frecuencia en PA5



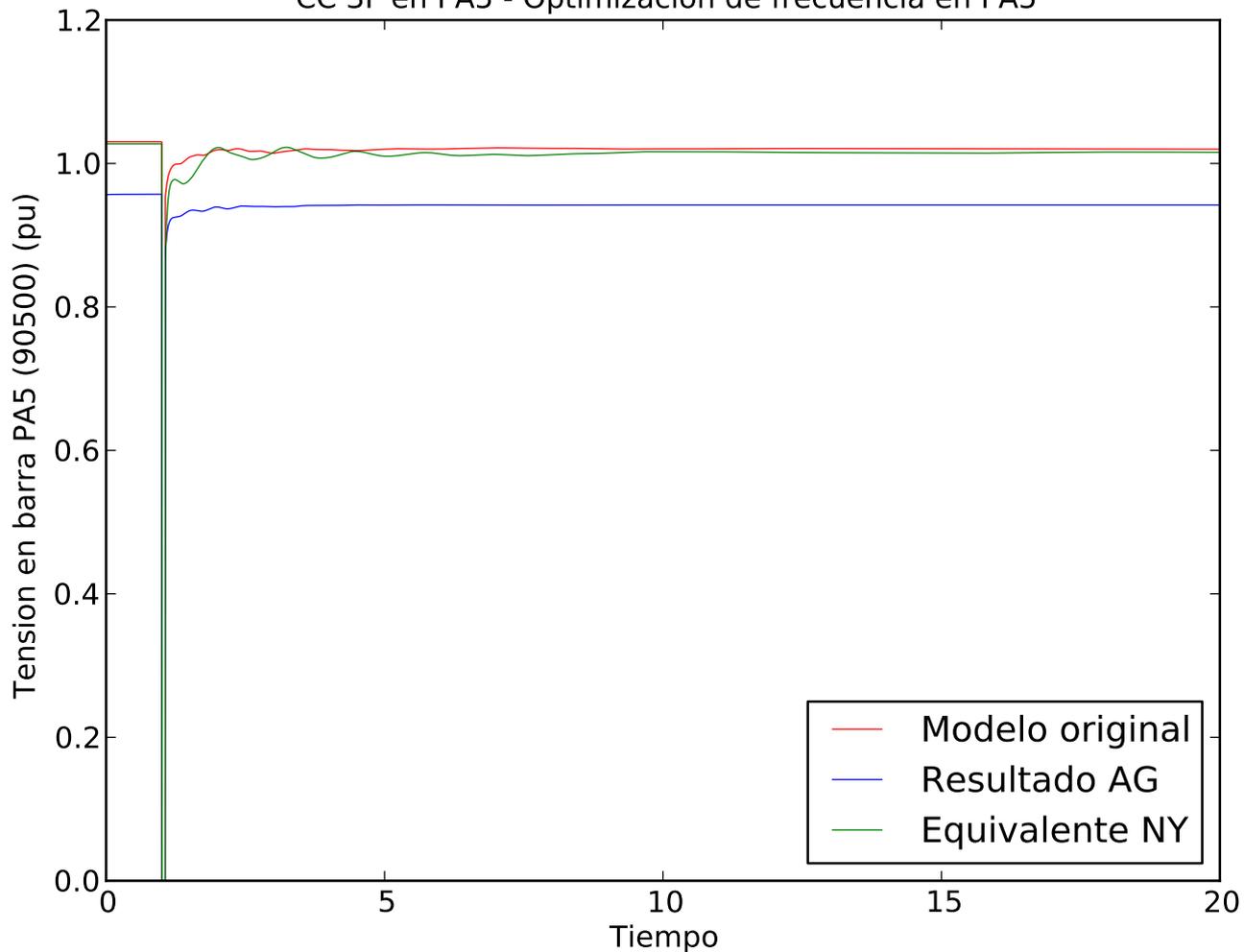
CC 3F en PA5 - Optimizacion de frecuencia en PA5



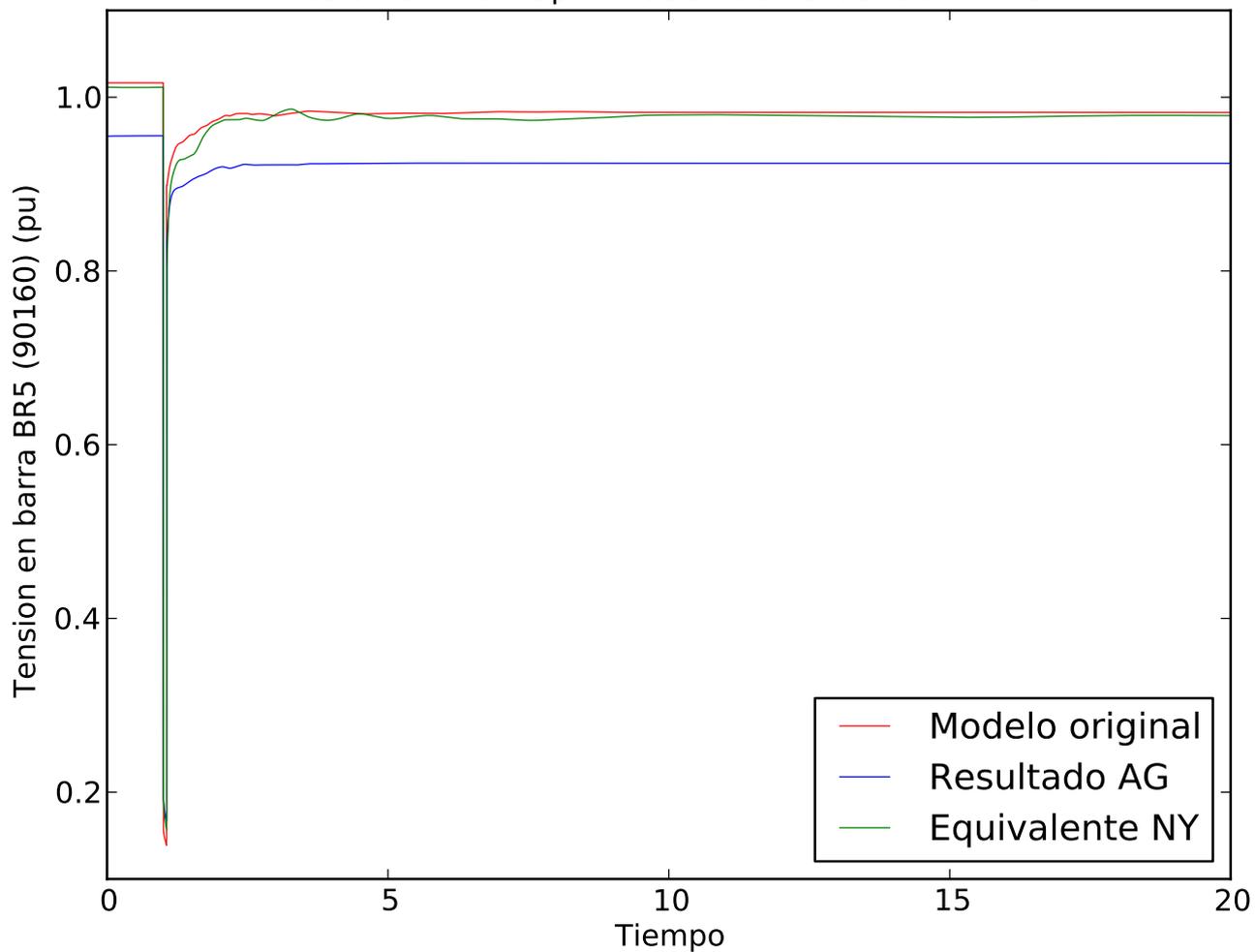
CC 3F en PA5 - Optimizacion de frecuencia en PA5



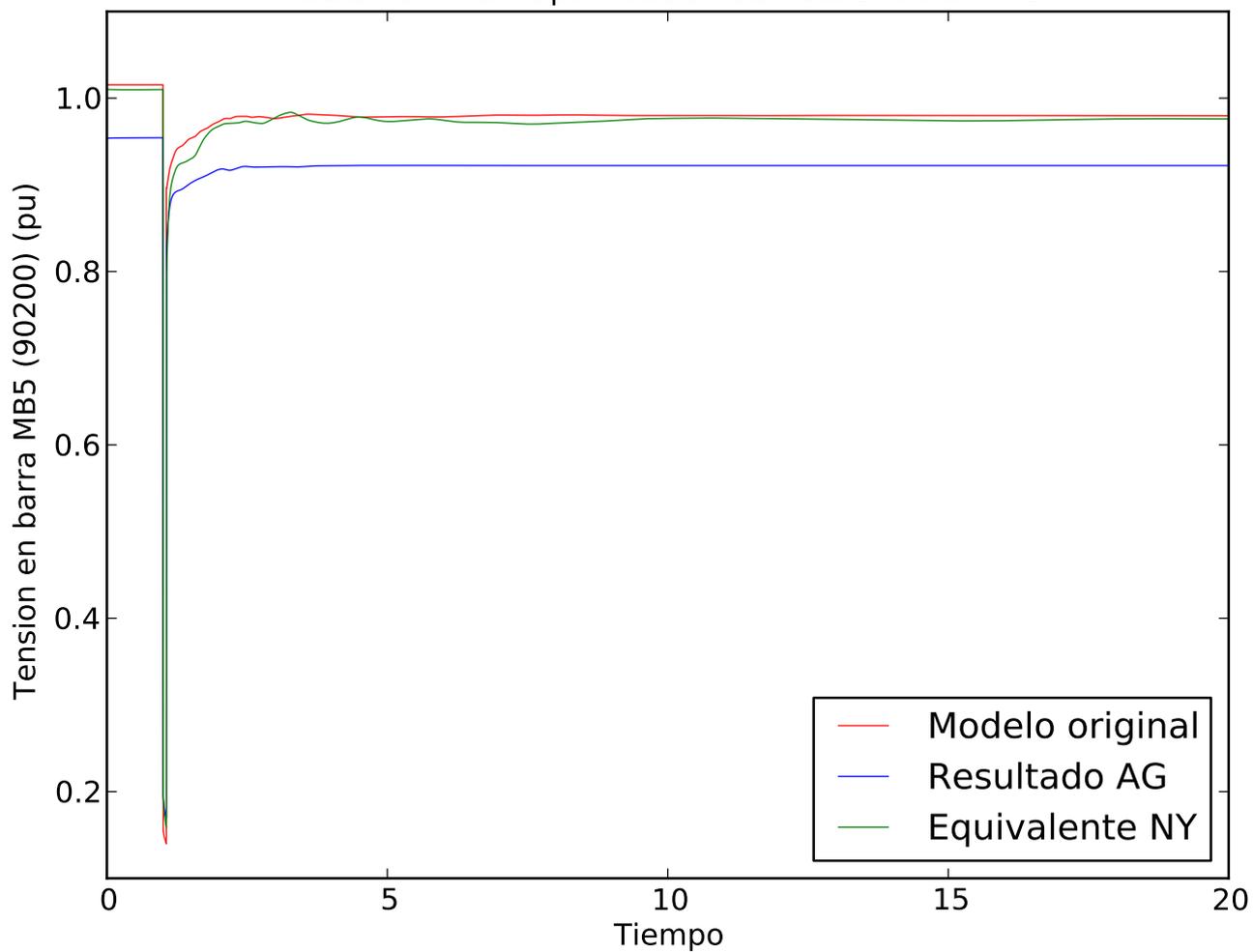
CC 3F en PA5 - Optimizacion de frecuencia en PA5



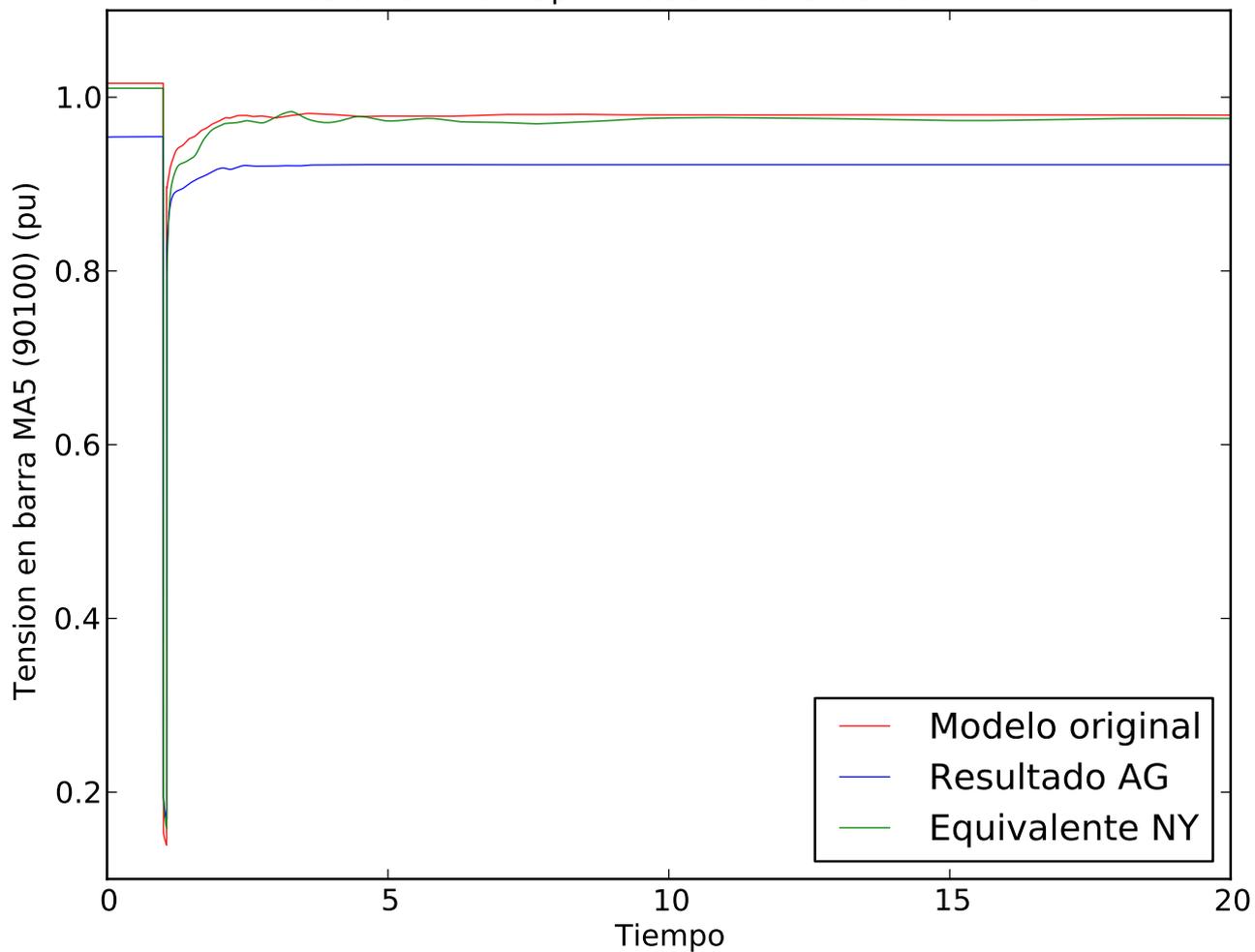
CC 3F en PA5 - Optimizacion de frecuencia en PA5



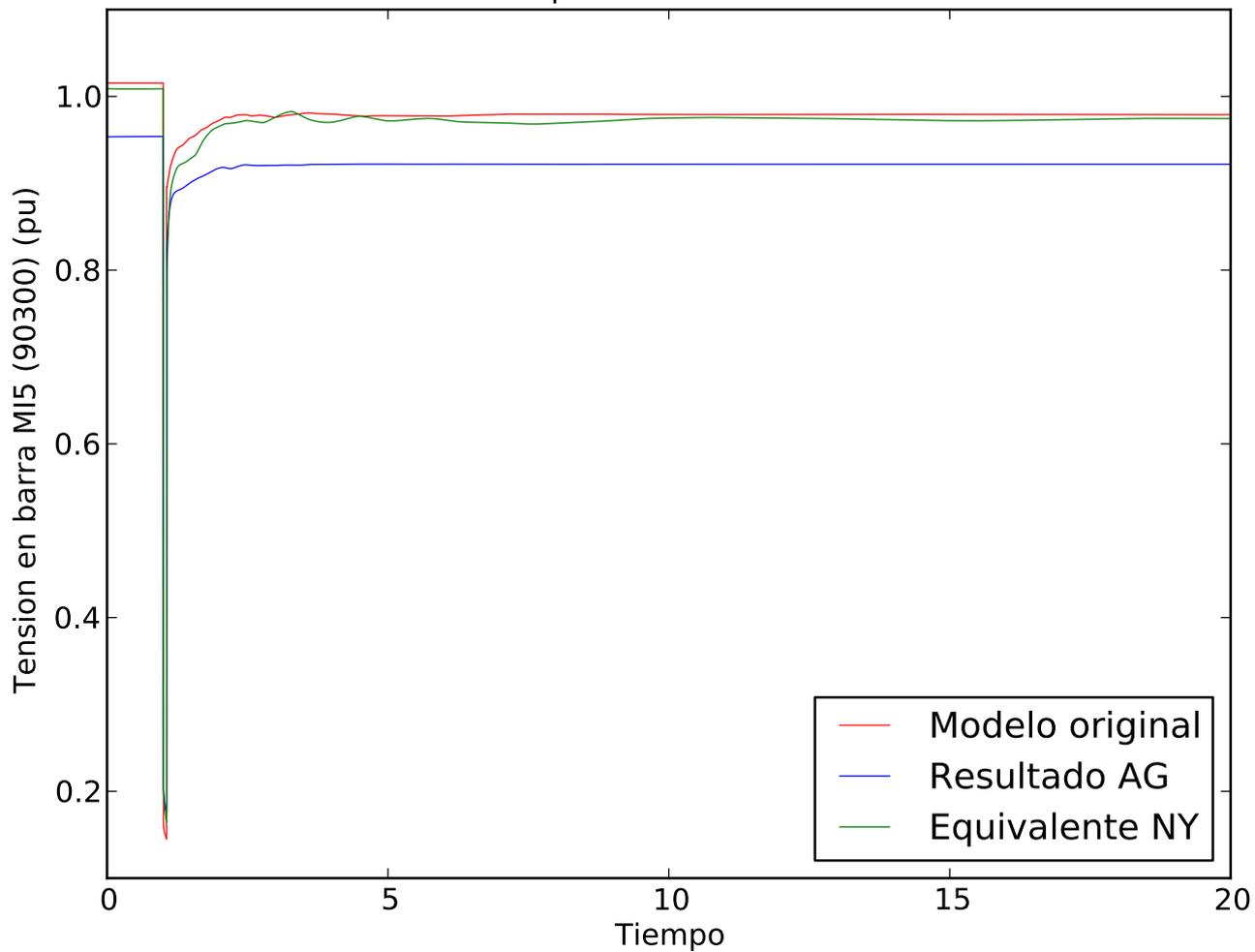
CC 3F en PA5 - Optimizacion de frecuencia en PA5



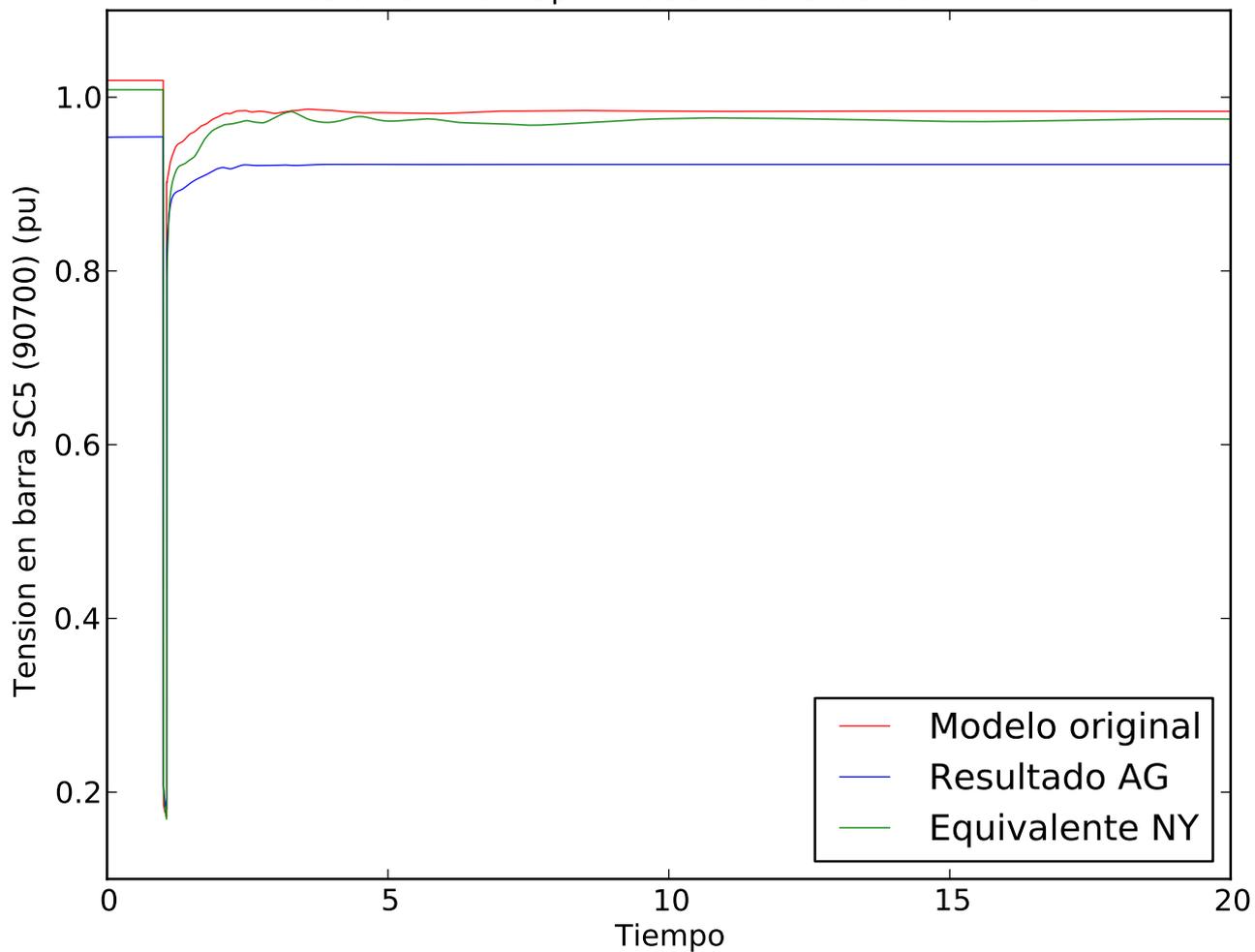
CC 3F en PA5 - Optimizacion de frecuencia en PA5



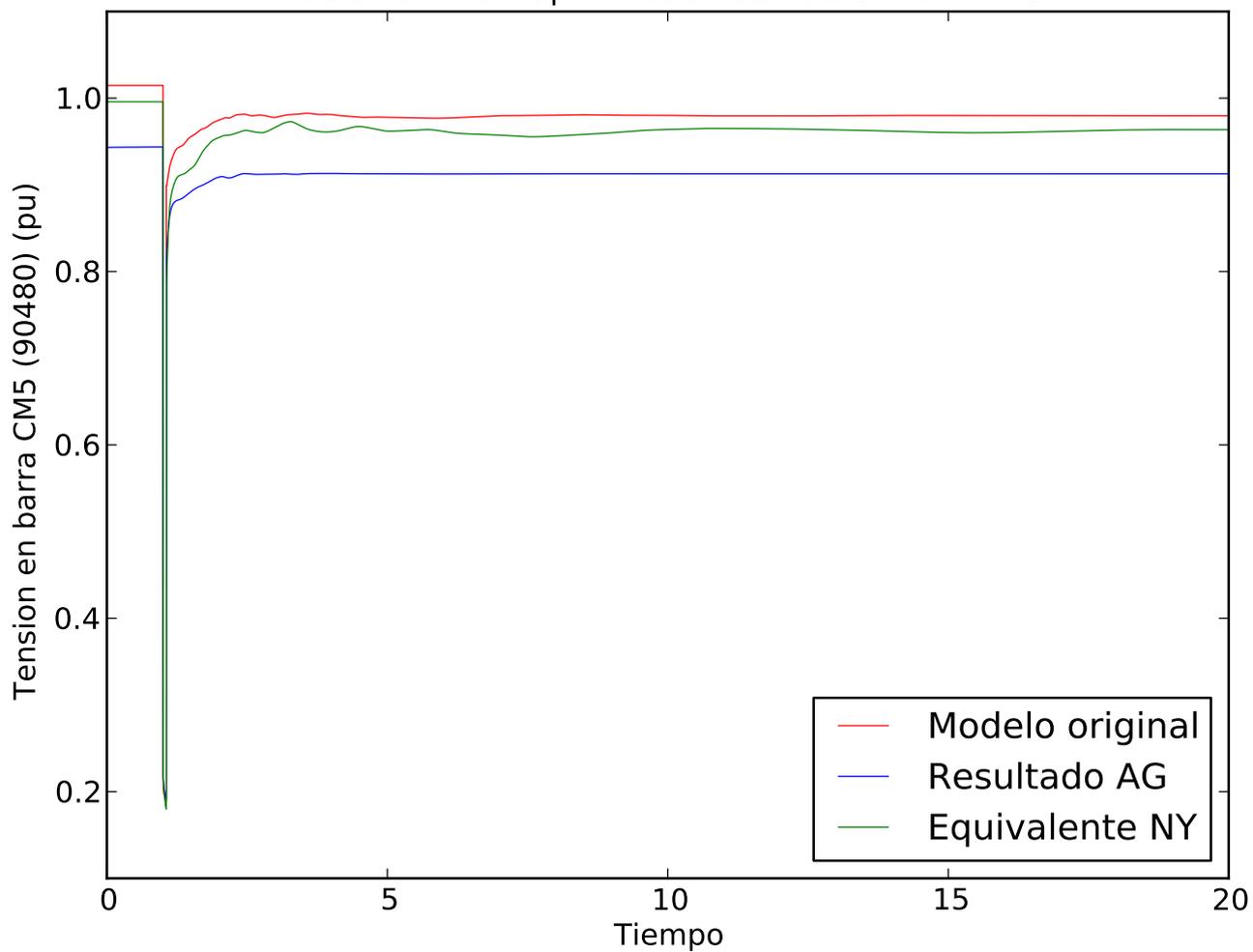
CC 3F en PA5 - Optimizacion de frecuencia en PA5



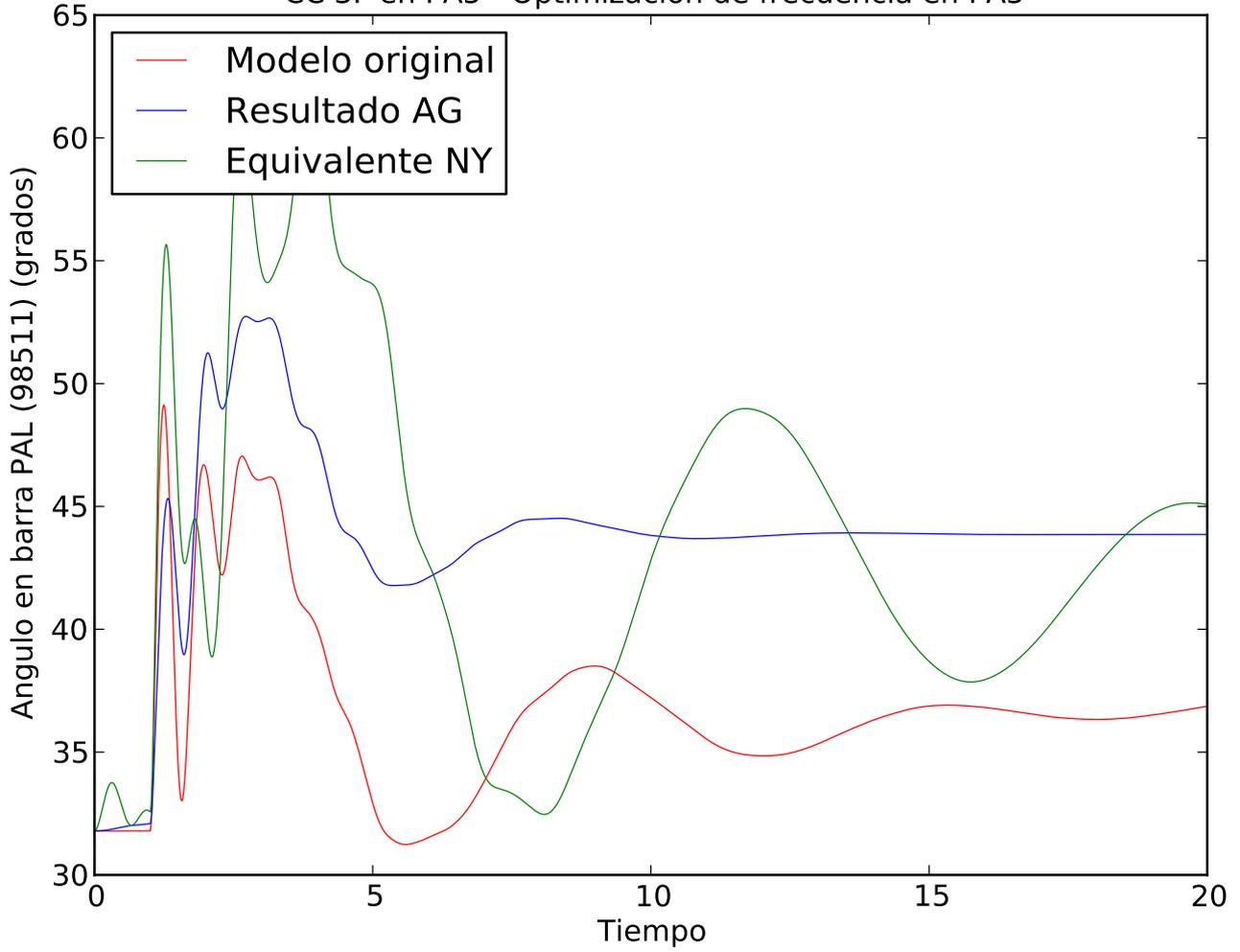
CC 3F en PA5 - Optimizacion de frecuencia en PA5



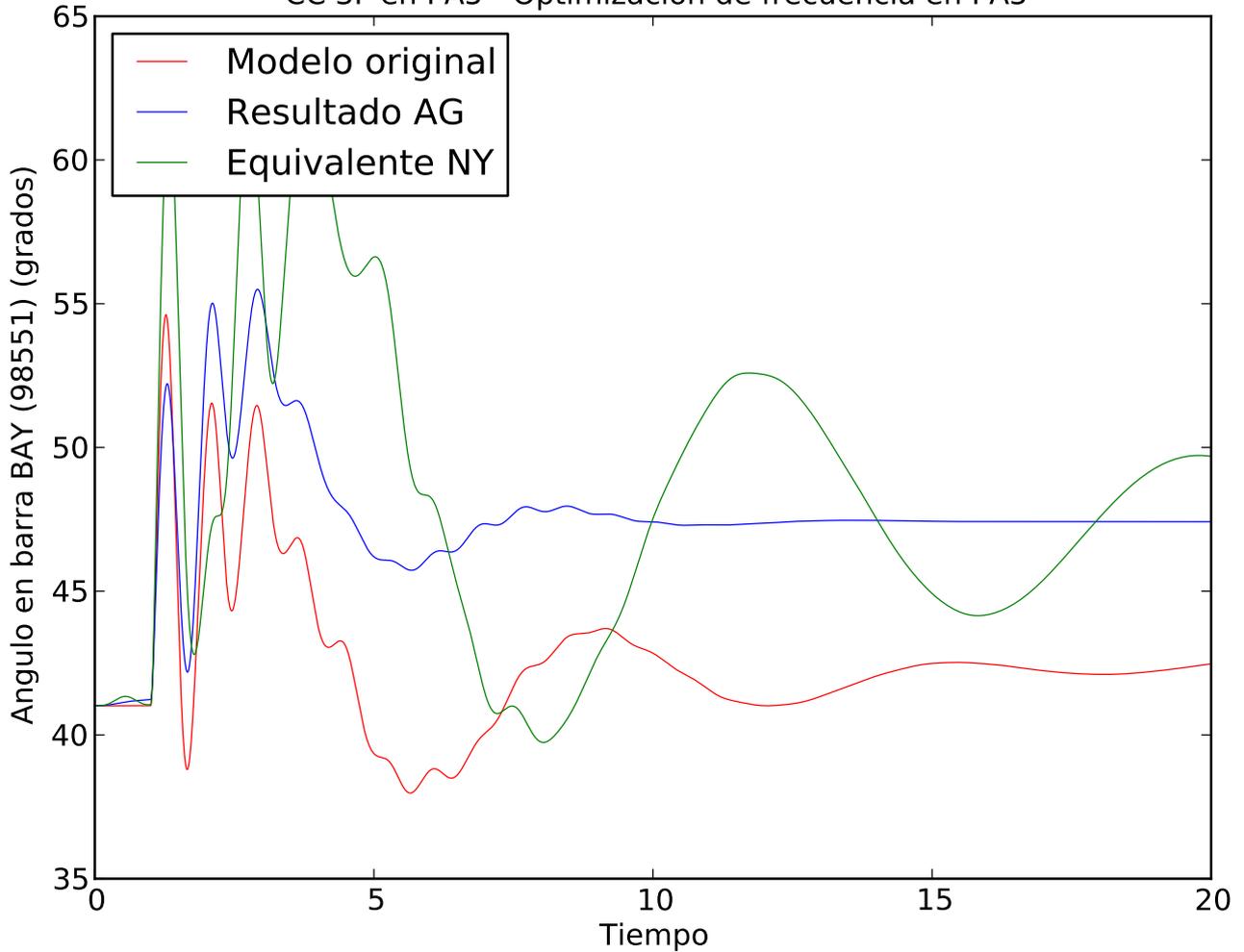
CC 3F en PA5 - Optimizacion de frecuencia en PA5



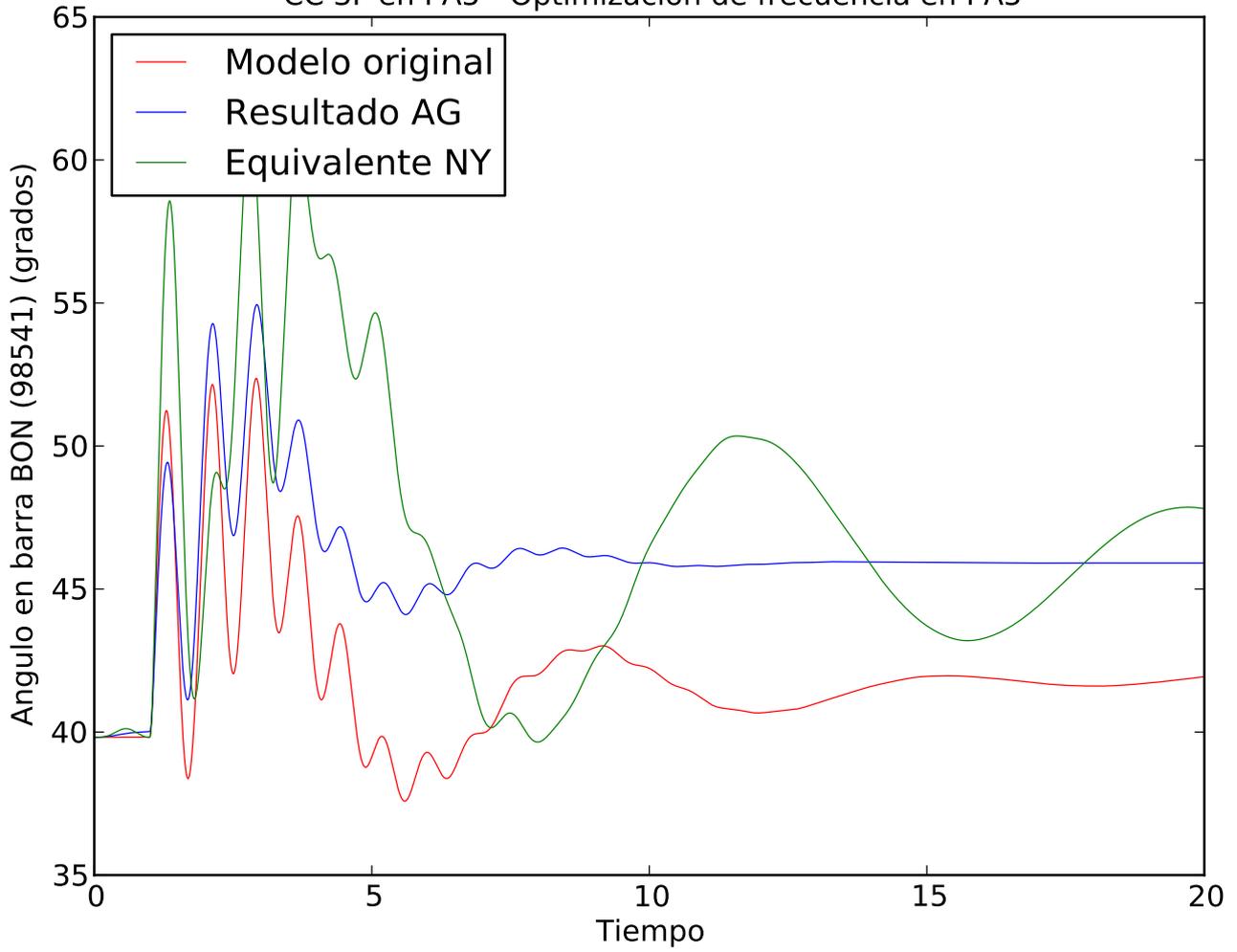
CC 3F en PA5 - Optimizacion de frecuencia en PA5



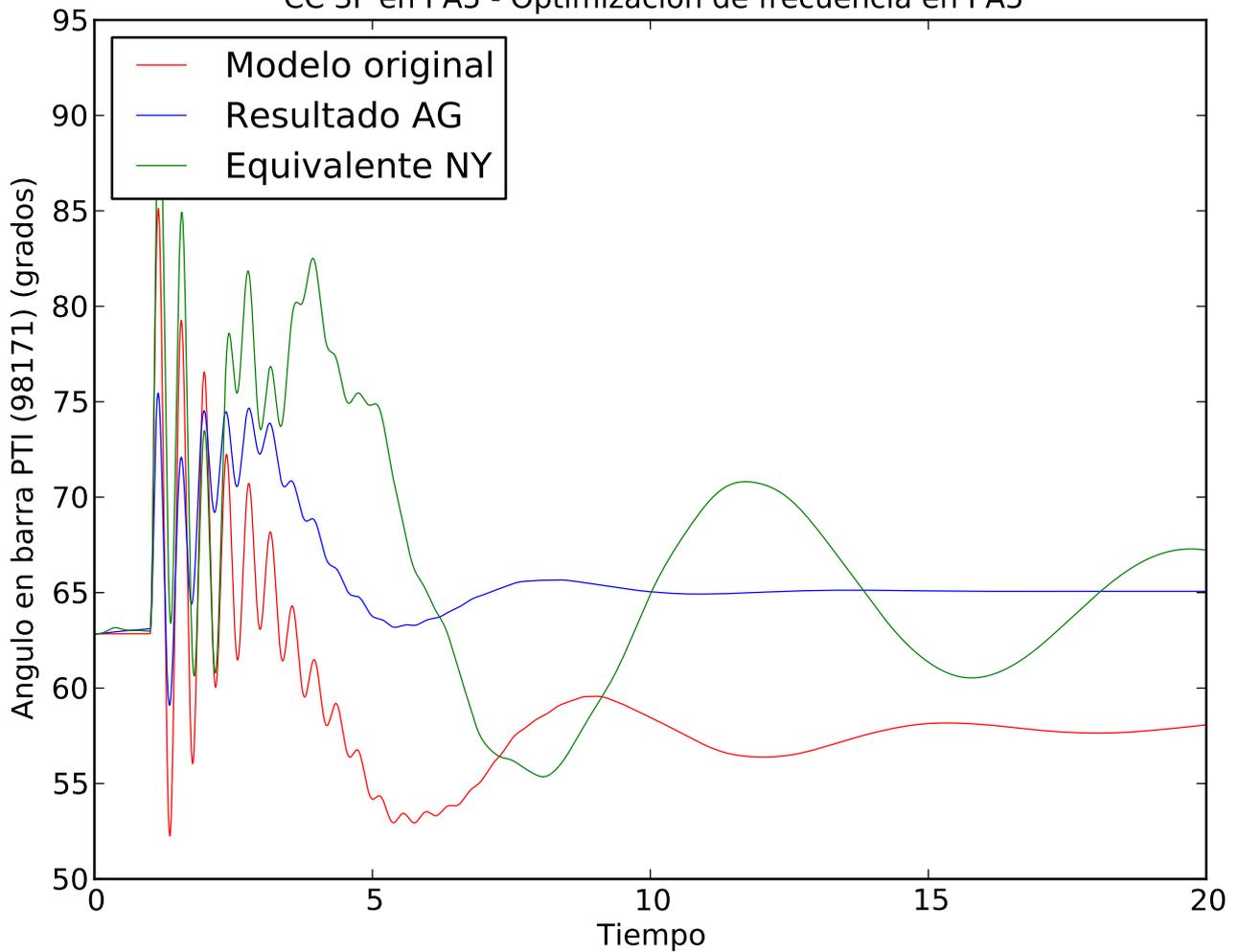
CC 3F en PA5 - Optimizacion de frecuencia en PA5



CC 3F en PA5 - Optimizacion de frecuencia en PA5

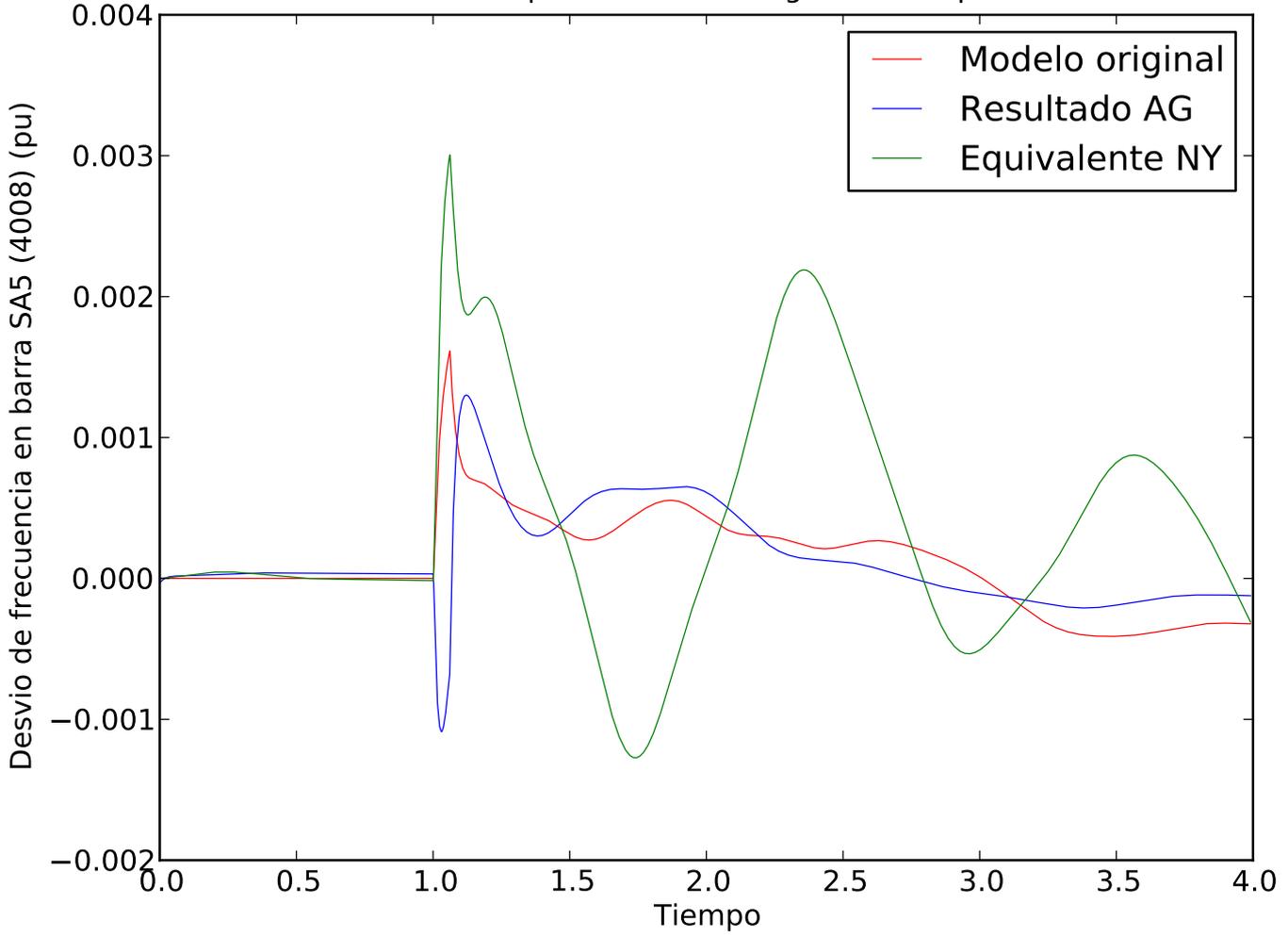


CC 3F en PA5 - Optimizacion de frecuencia en PA5

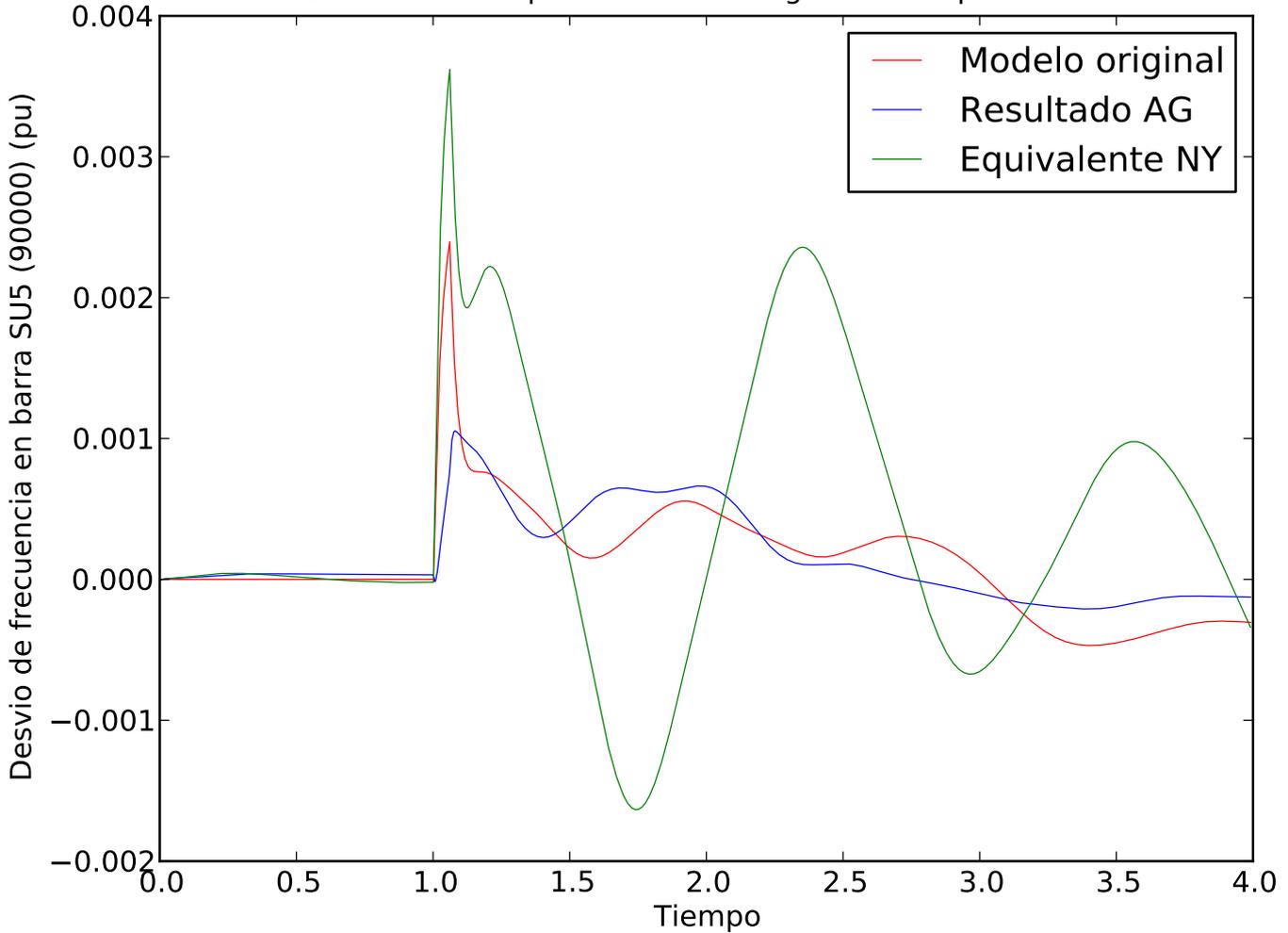


- C.4. Simulación 3: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando ángulo en una máquina de PAL en 4 segundos. Comportamiento en 4 segundos**

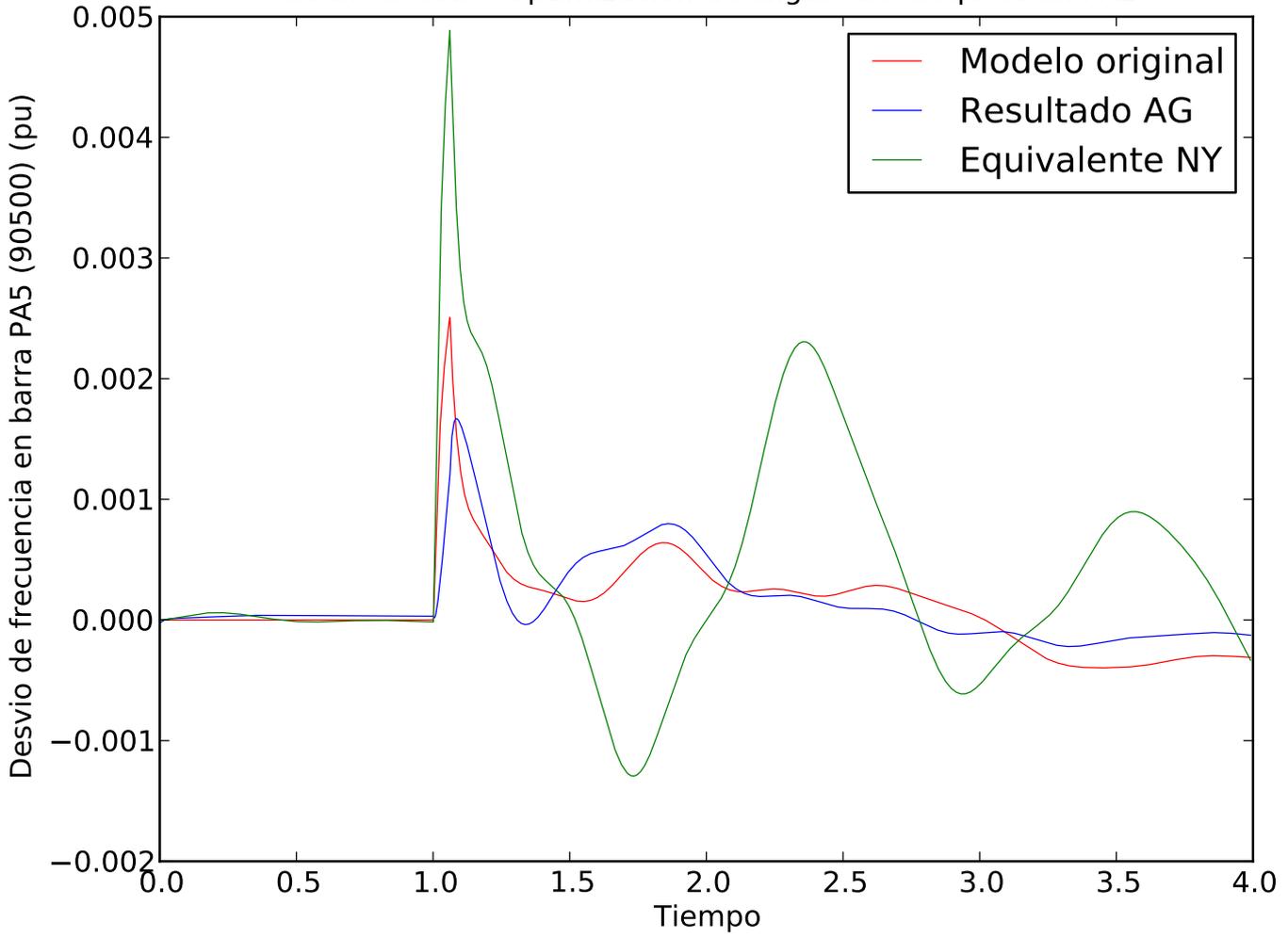
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



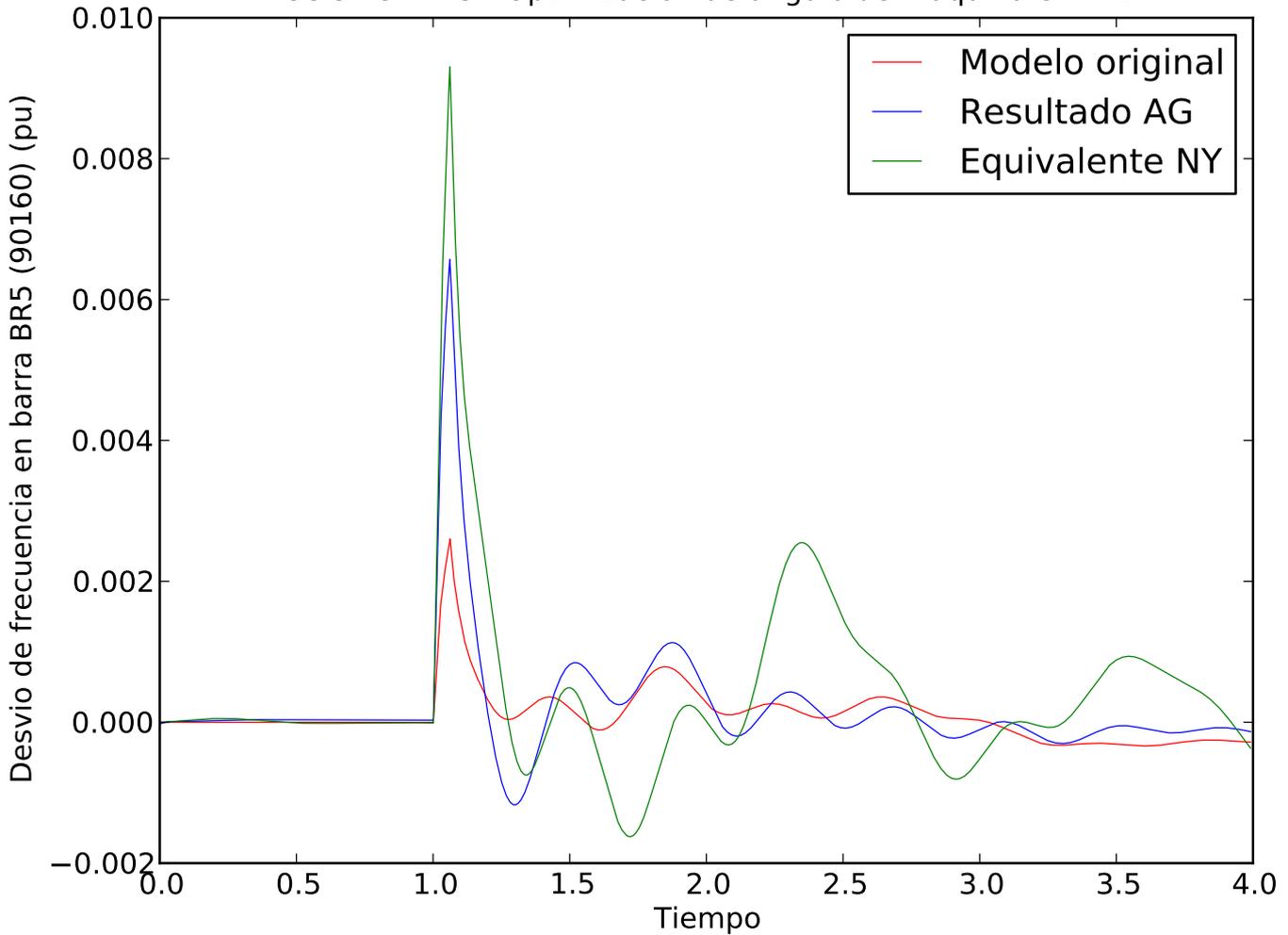
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



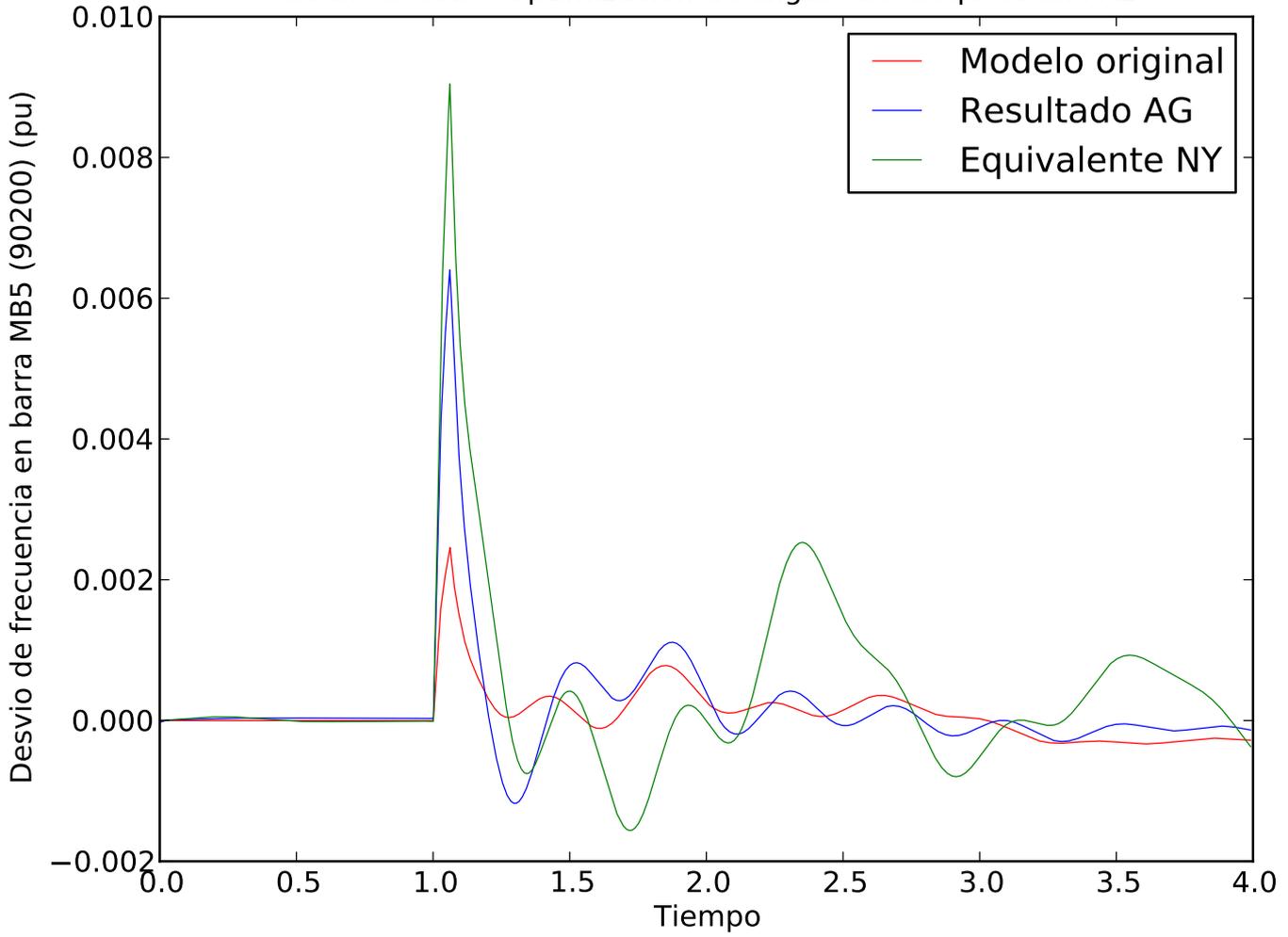
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



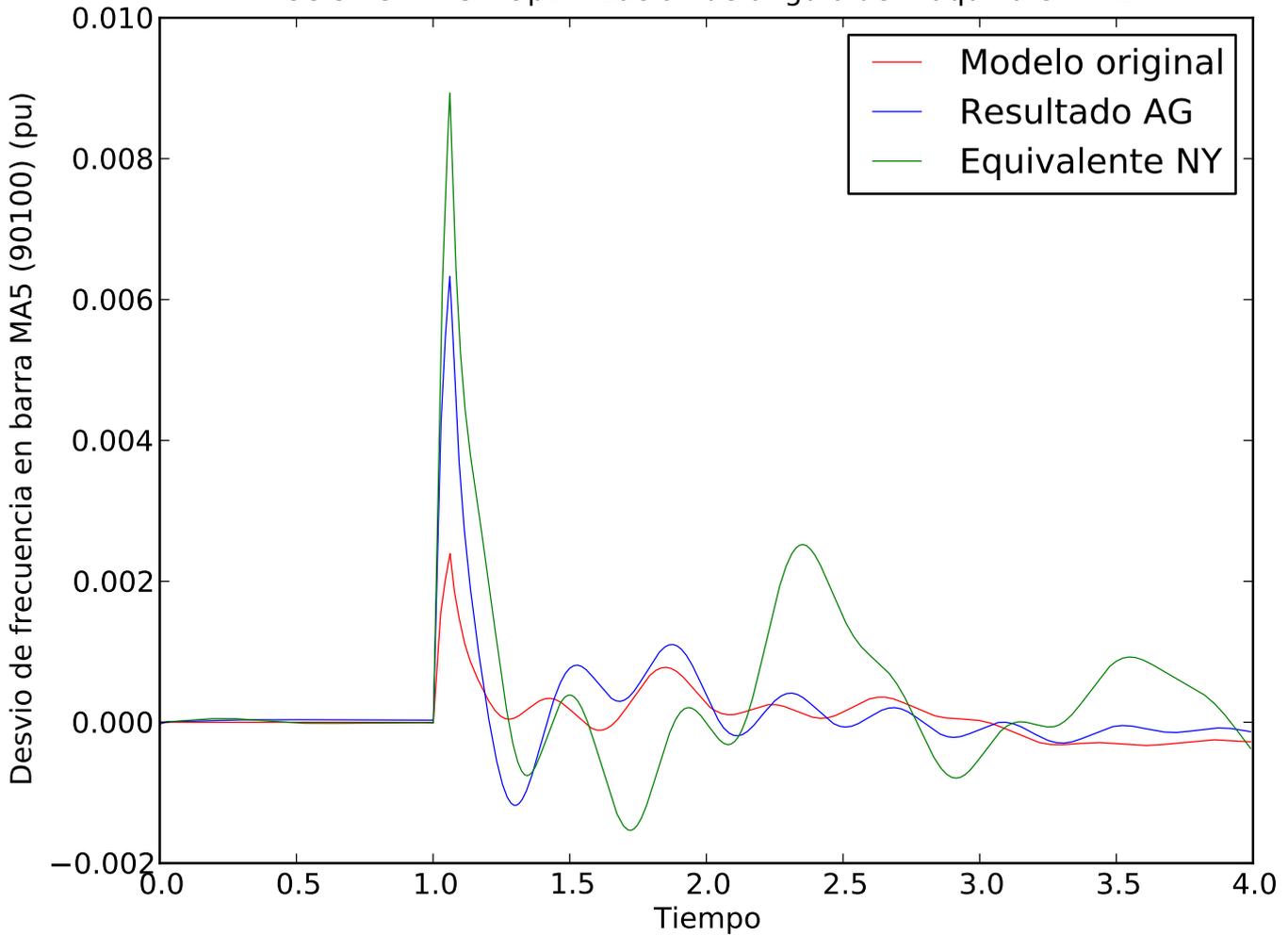
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



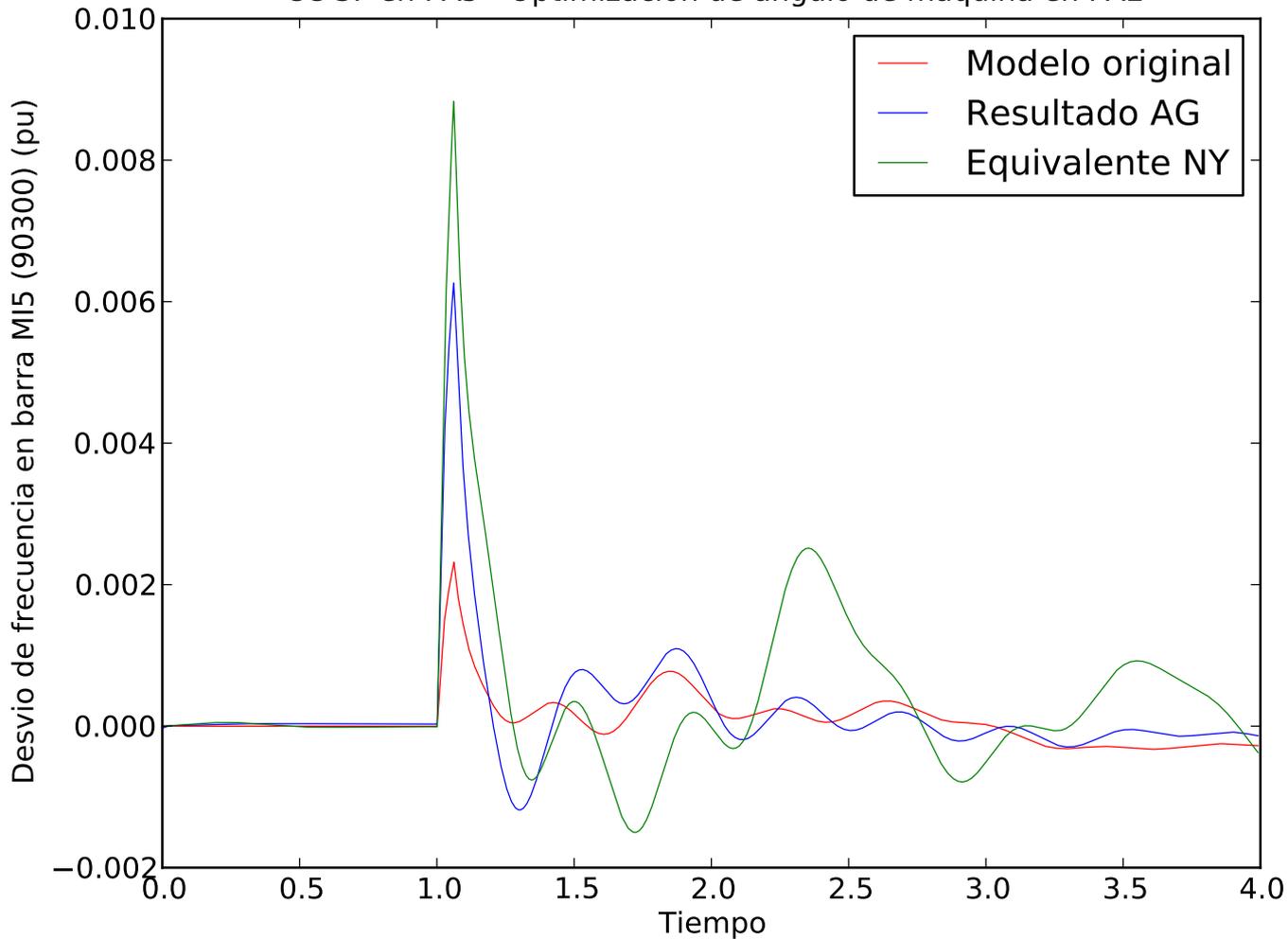
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



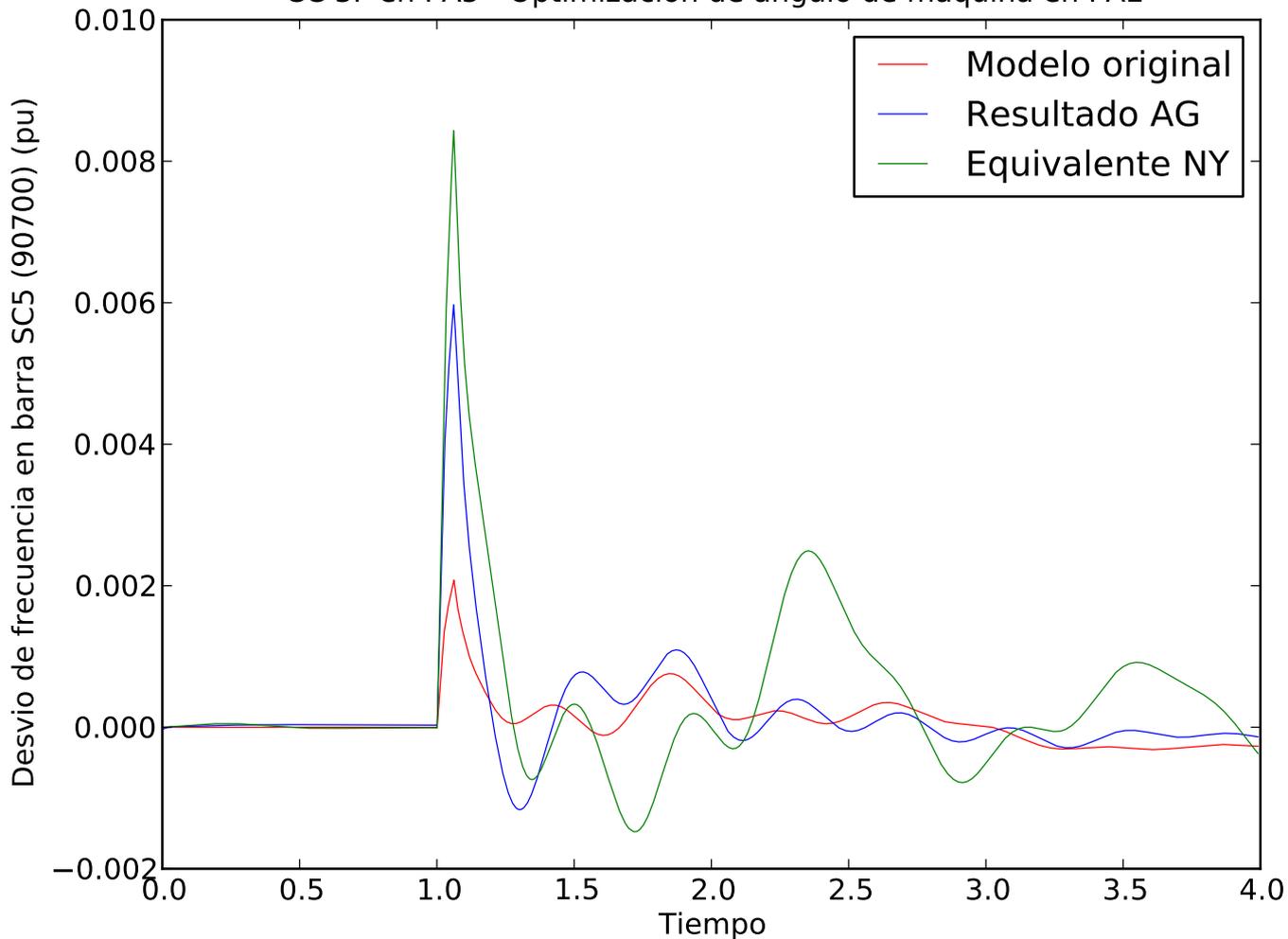
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



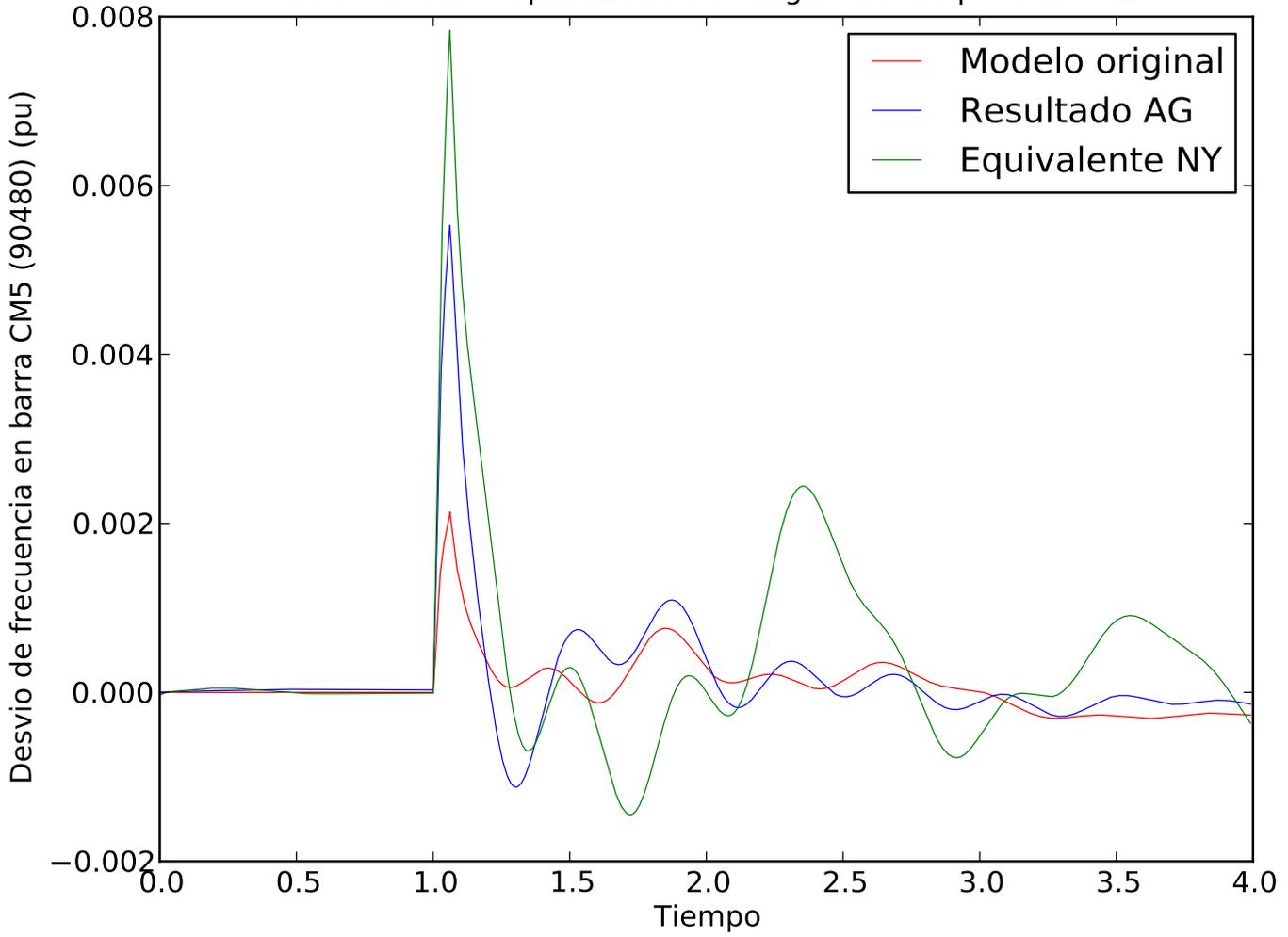
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



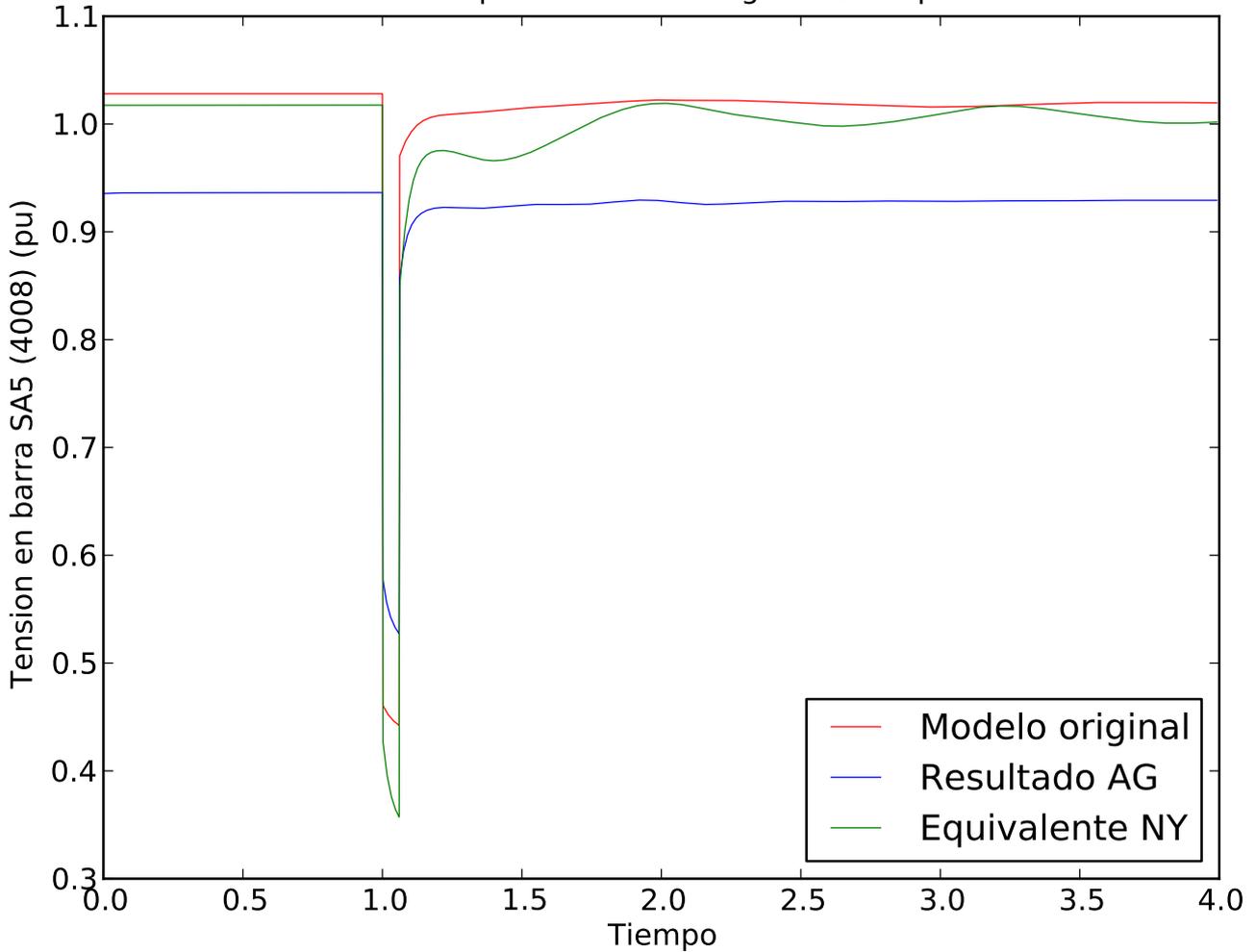
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



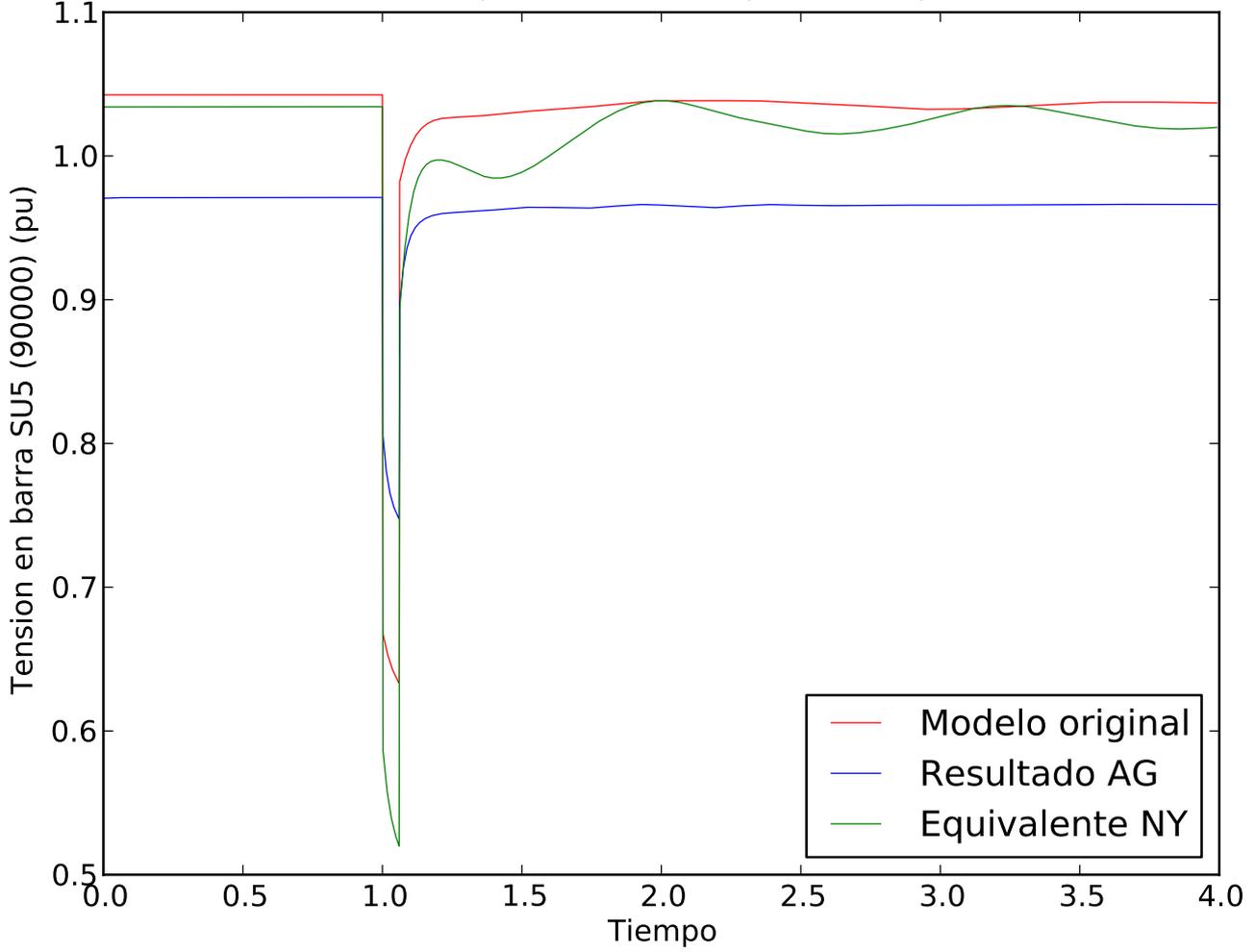
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



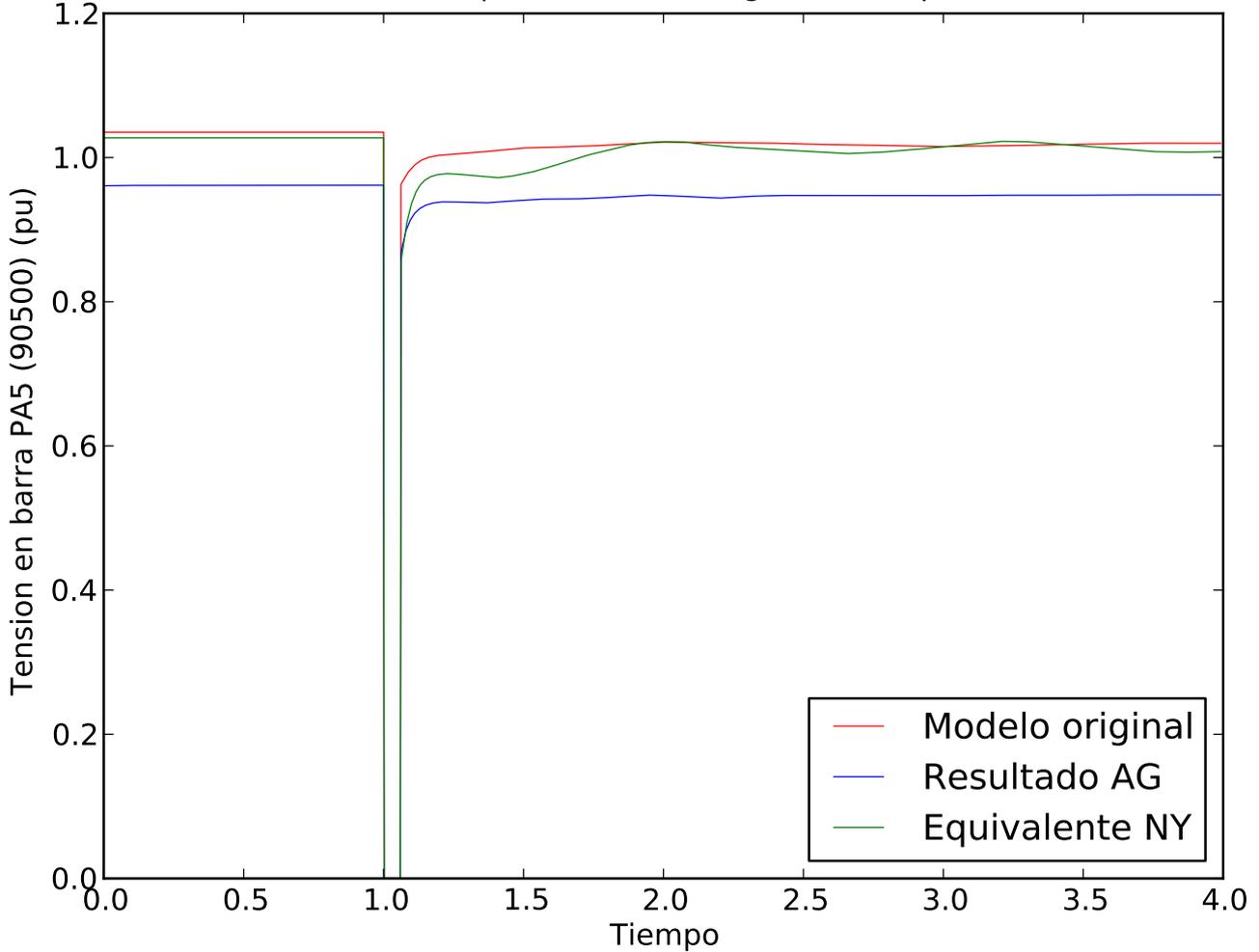
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



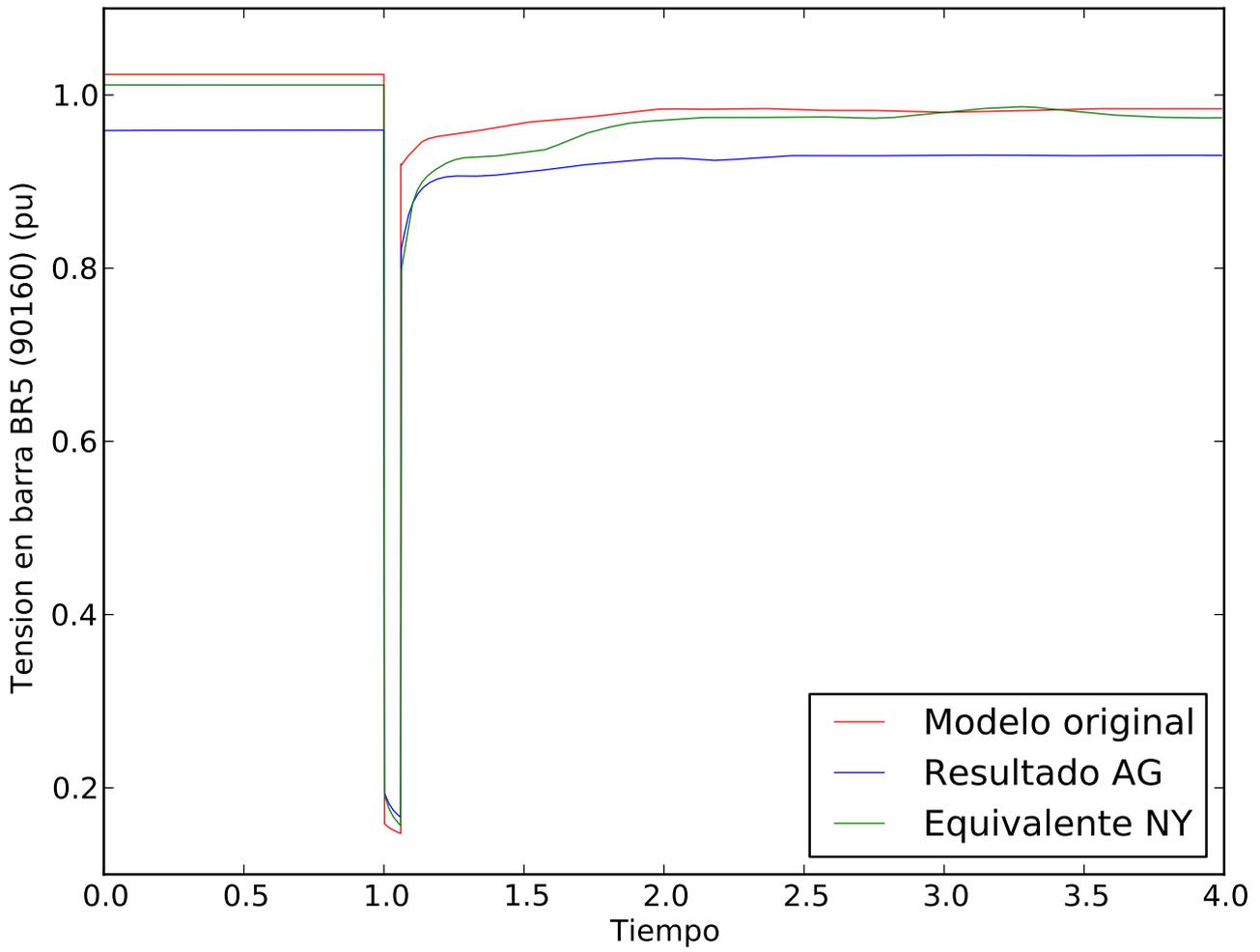
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



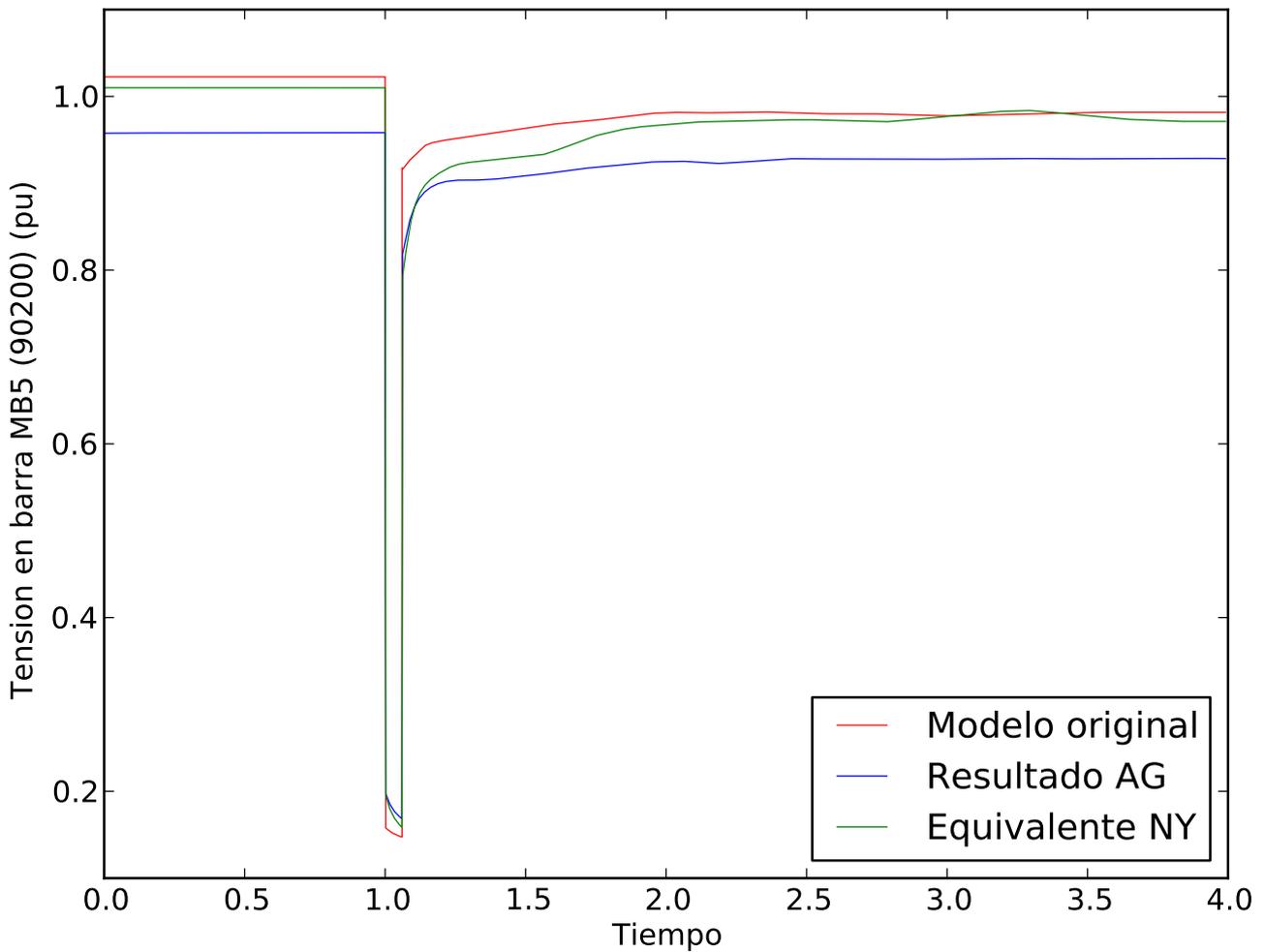
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



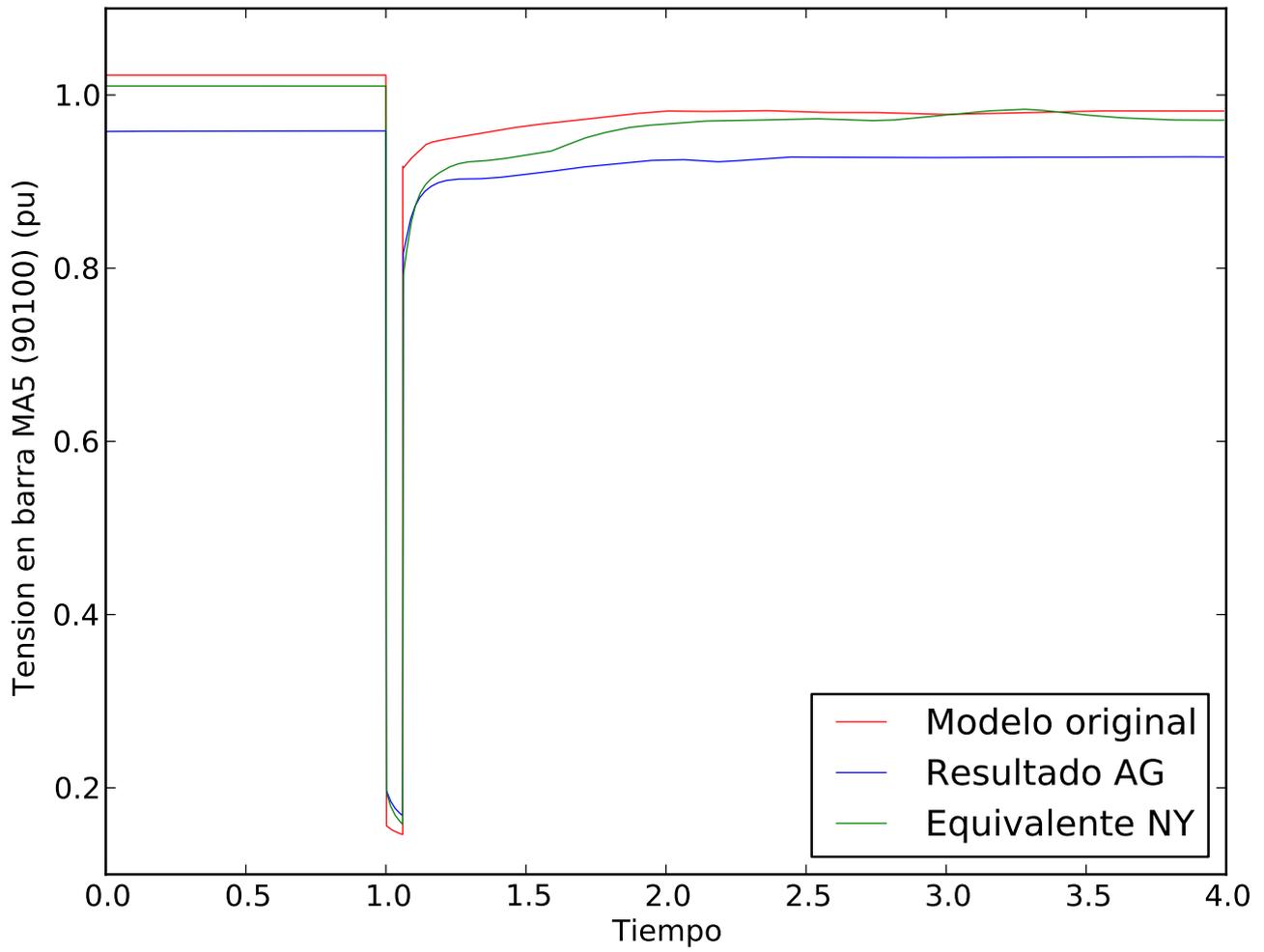
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



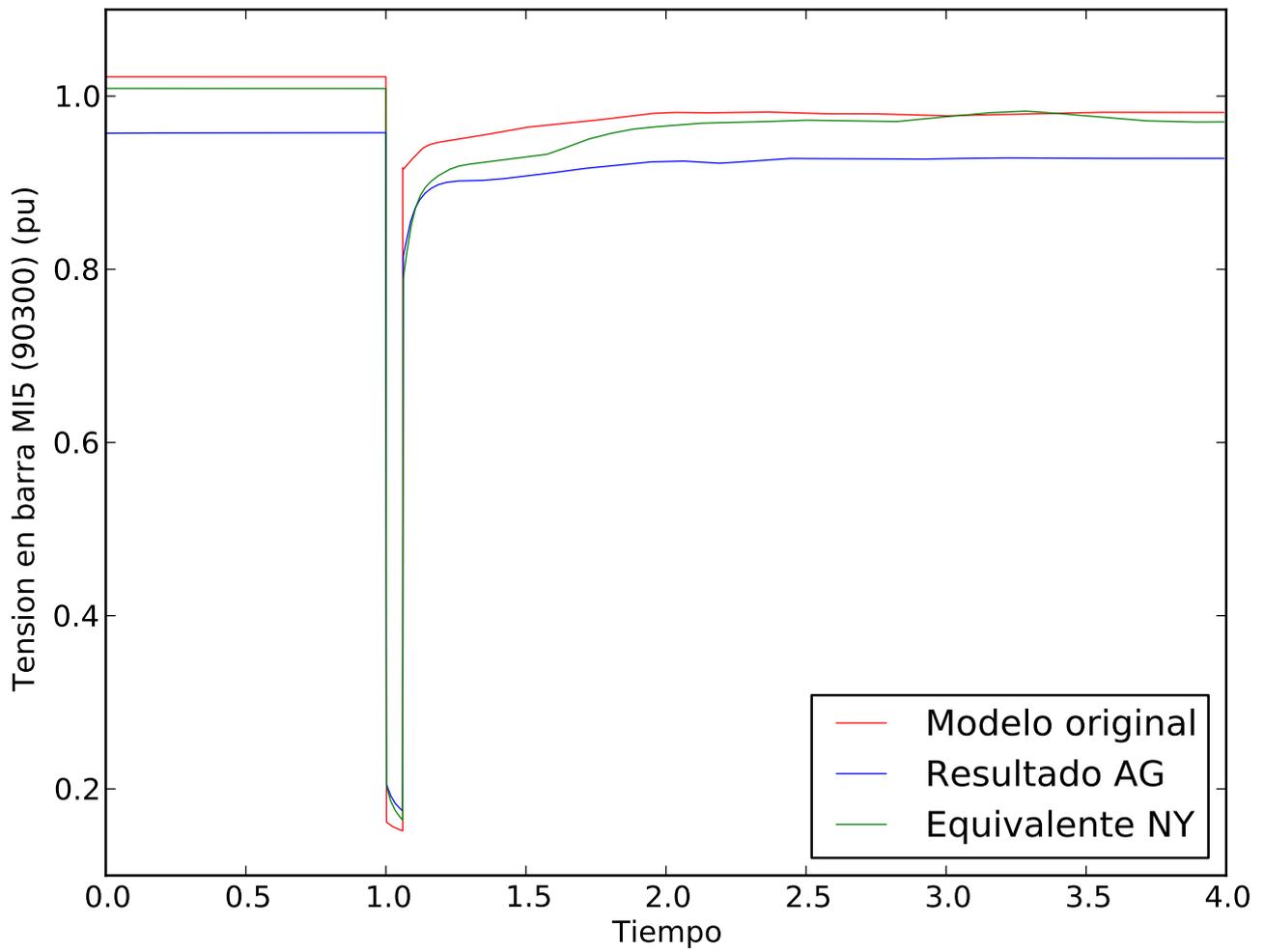
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



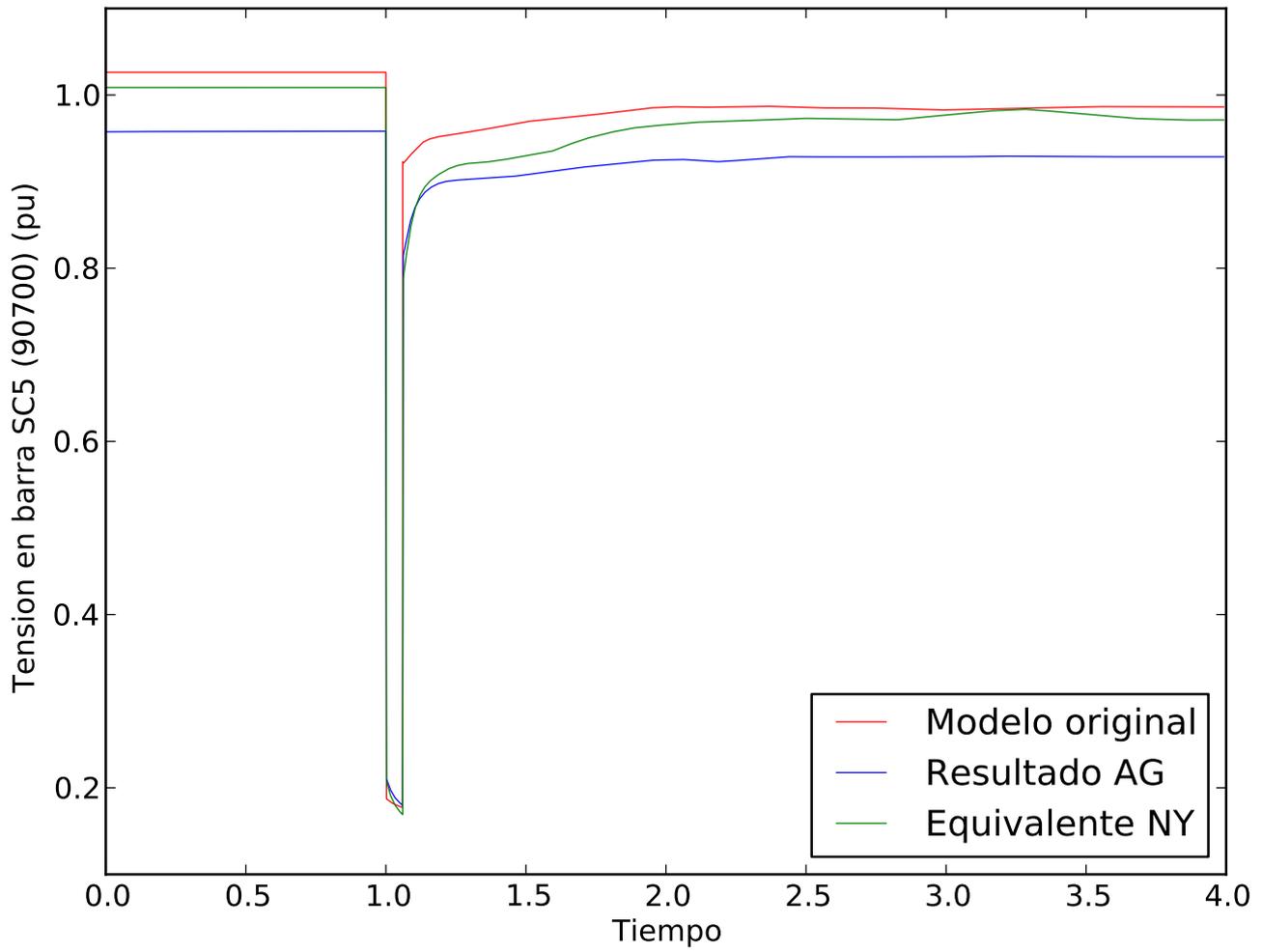
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



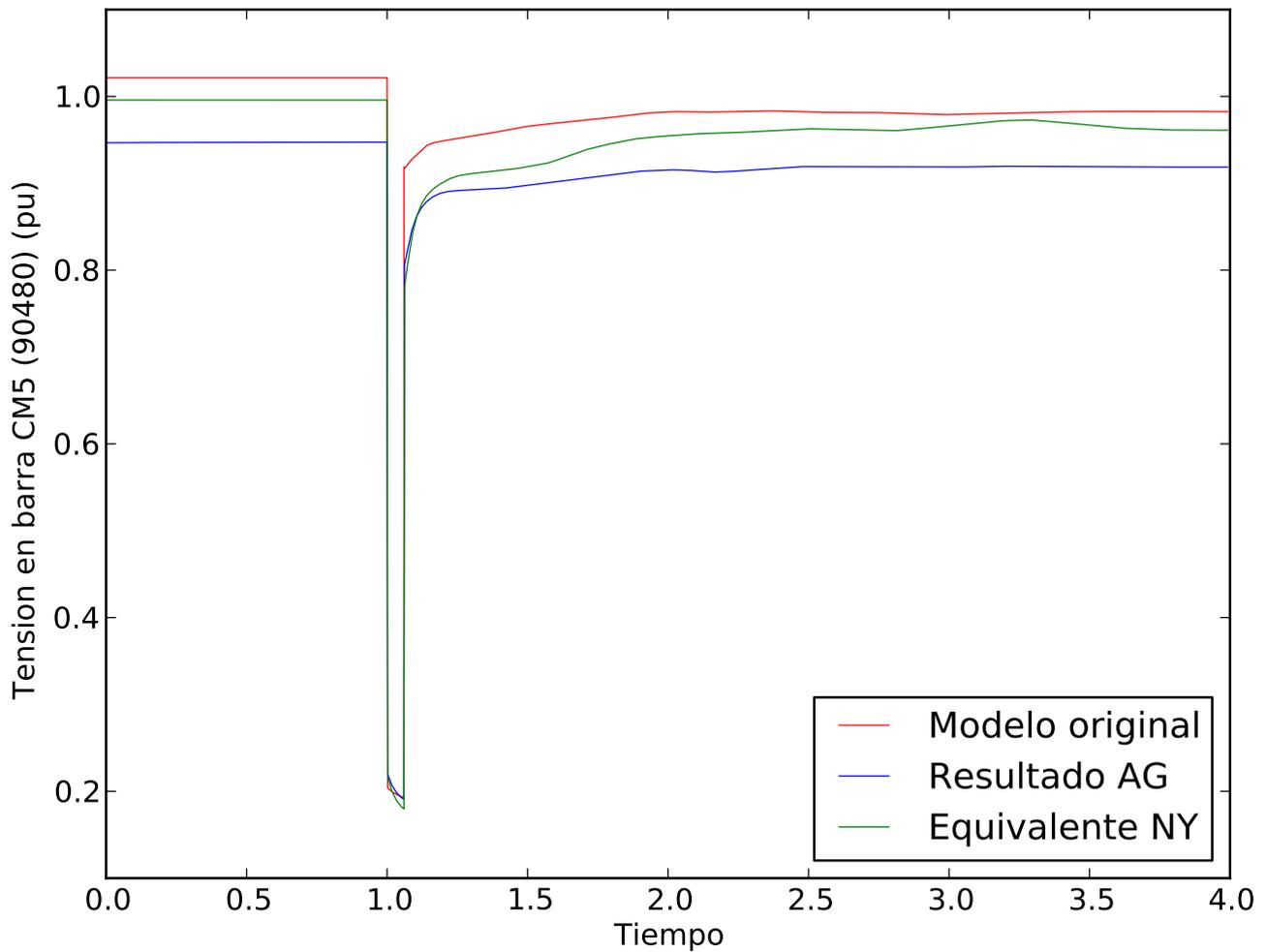
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



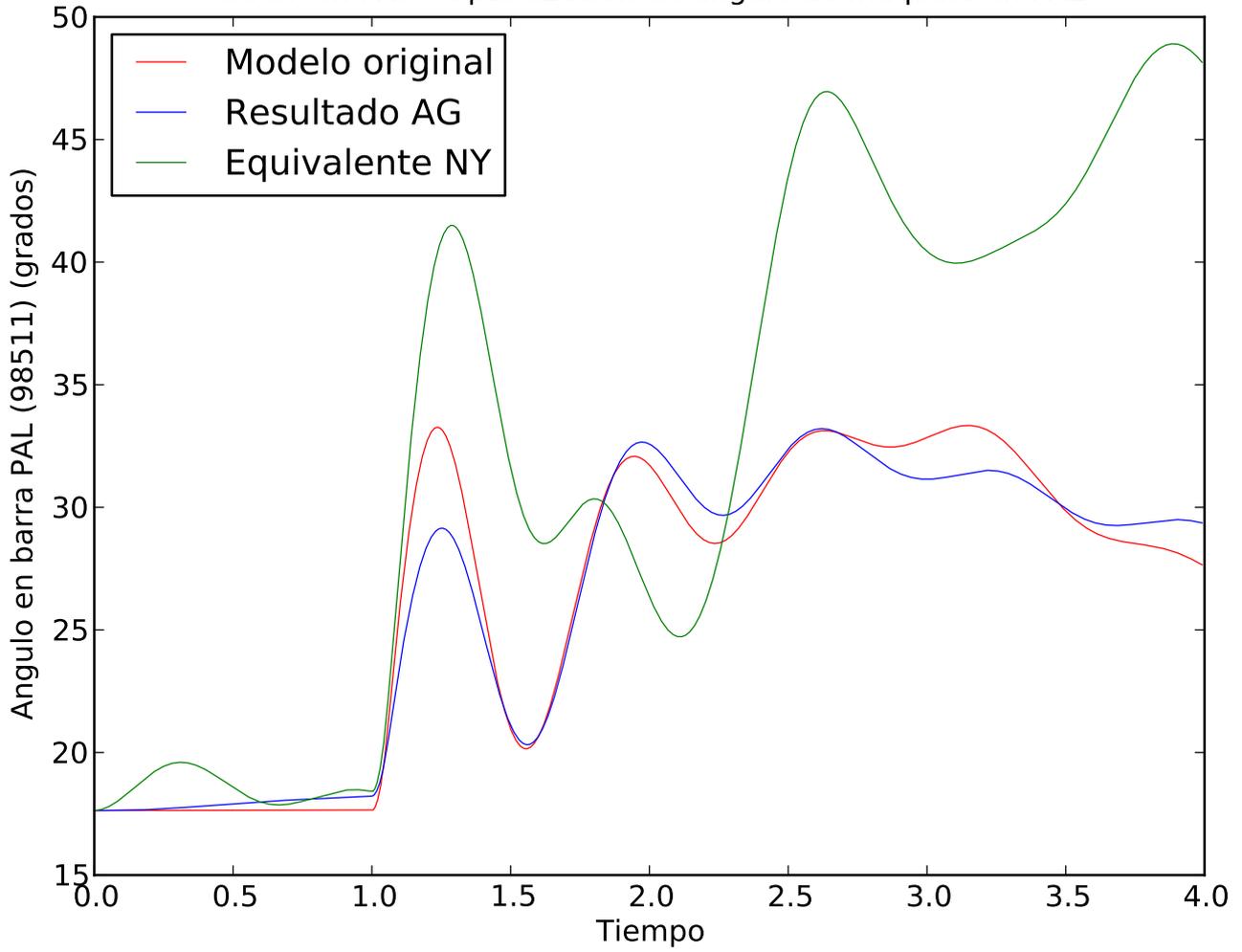
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



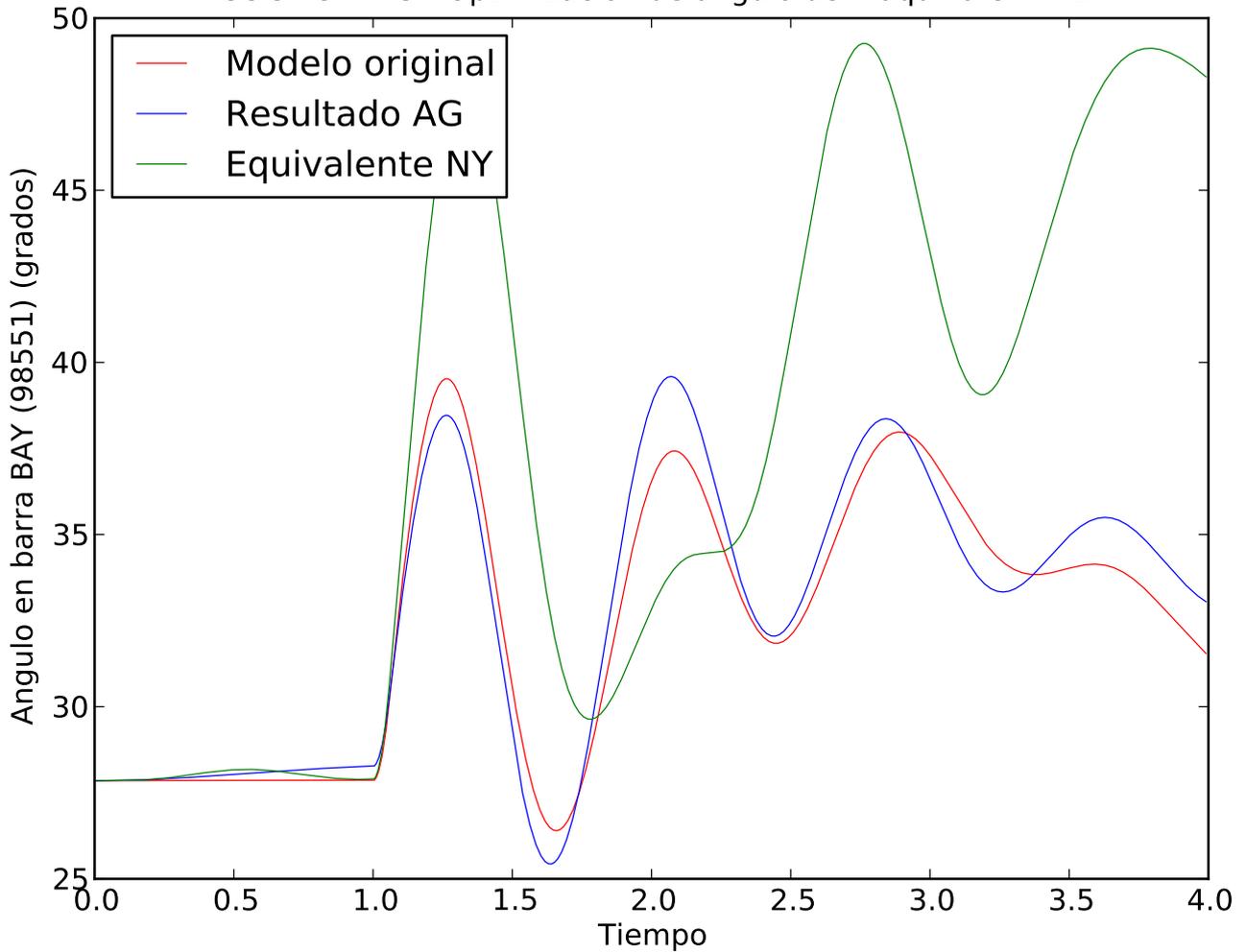
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



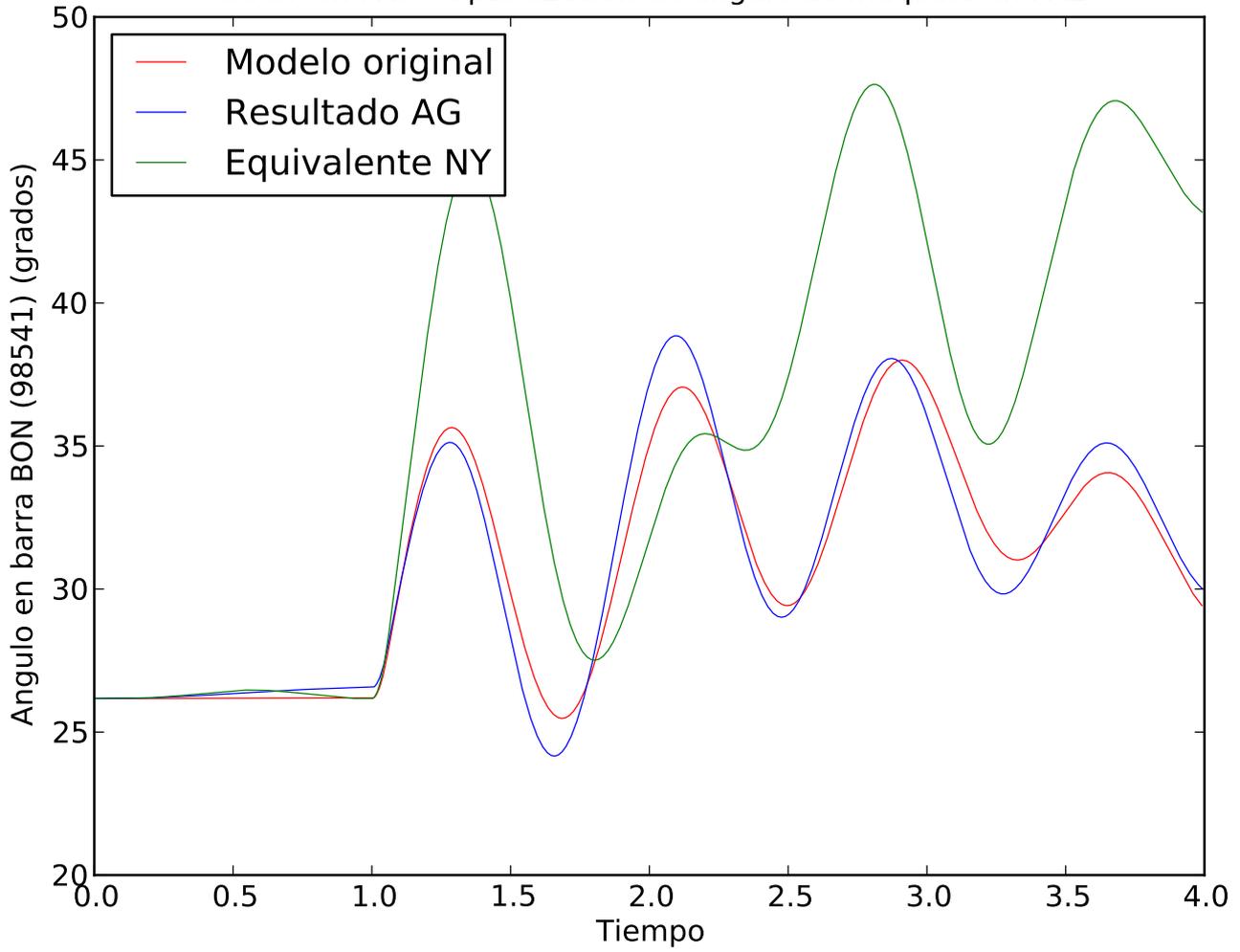
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



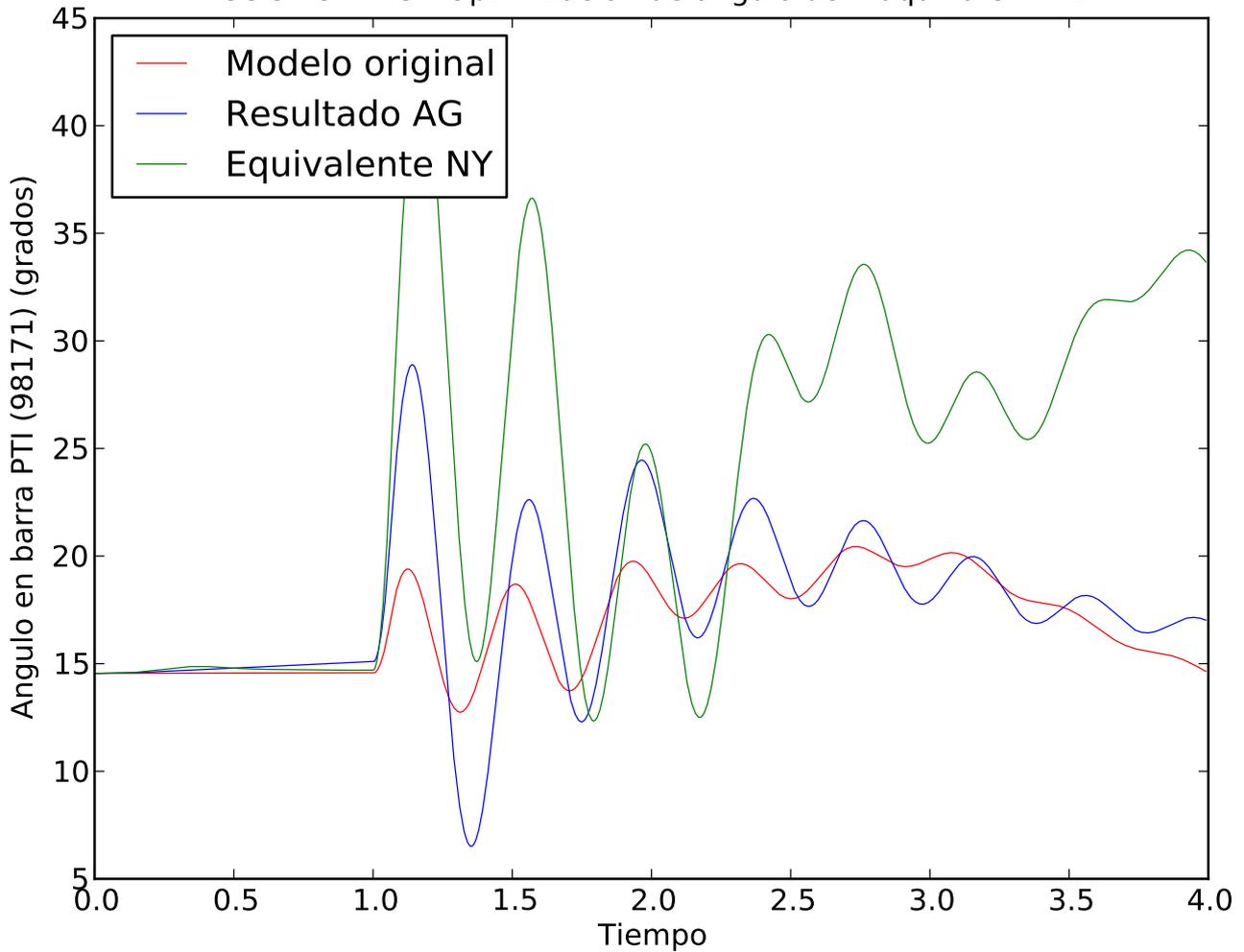
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



CC 3F en PA5 - Optimizacion de angulo de maquina en PAL

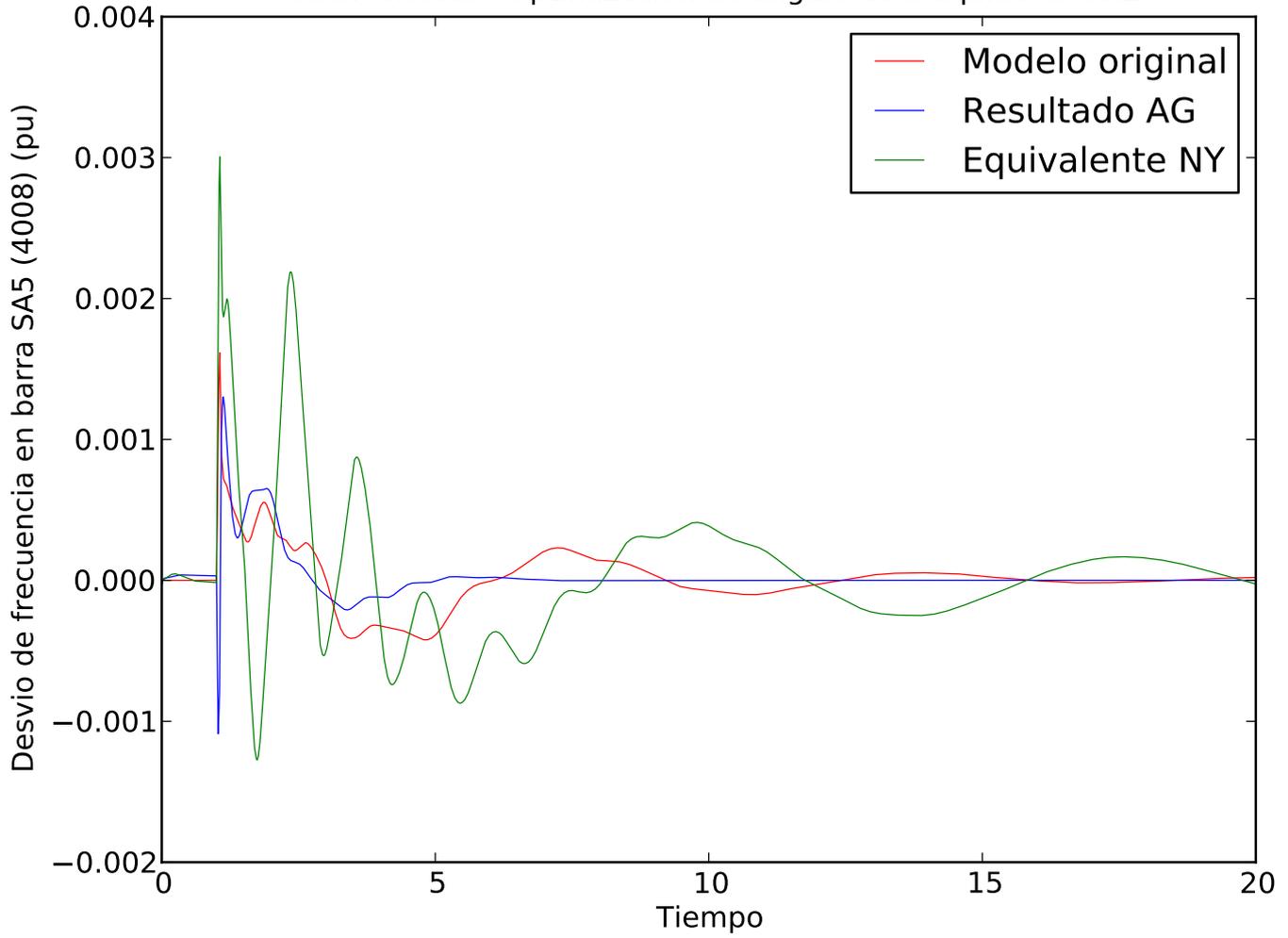


CC 3F en PA5 - Optimizacion de angulo de maquina en PAL

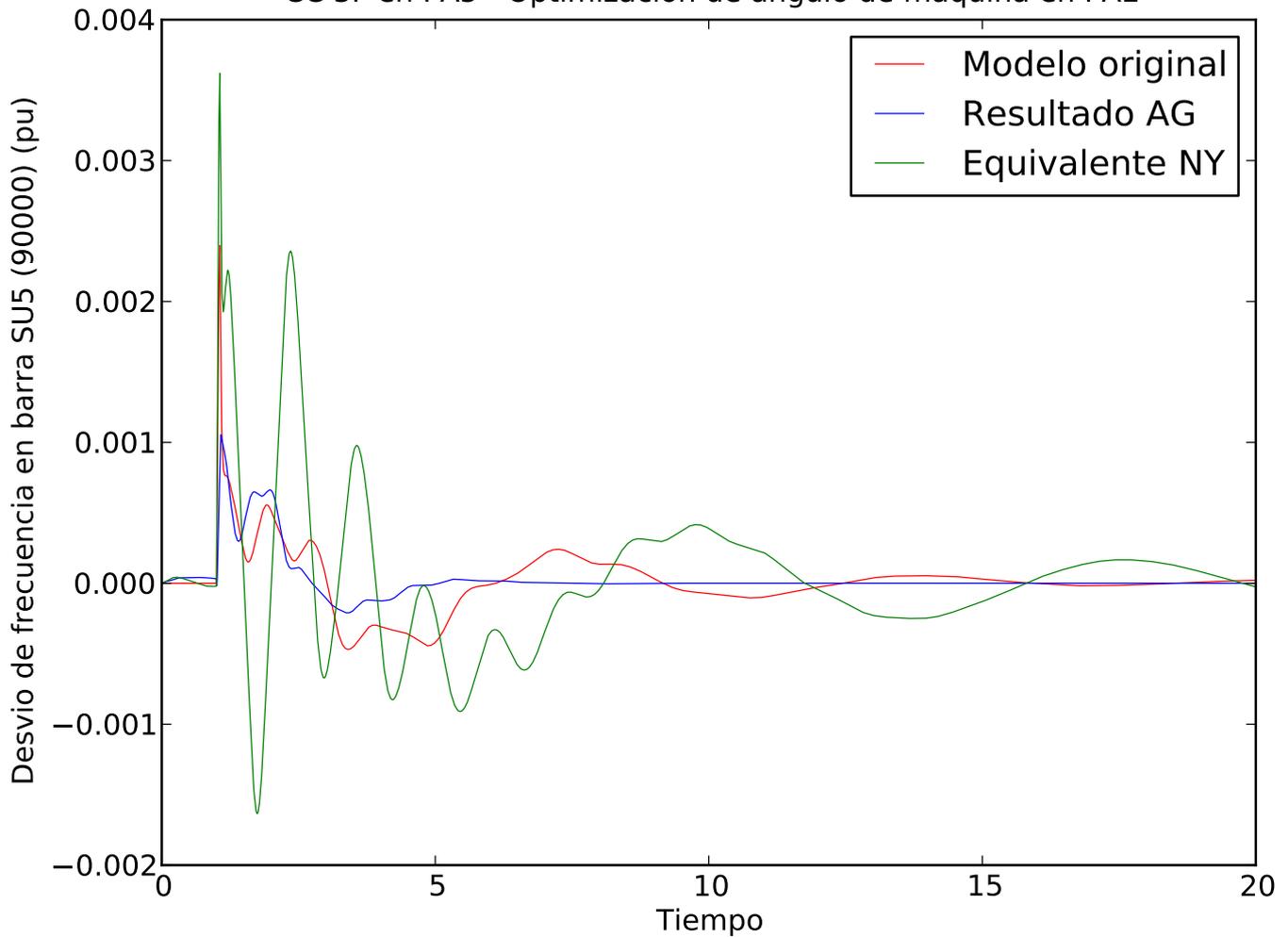


- C.5. Simulación 3: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando ángulo en una máquina de PAL en 4 segundos. Comportamiento en 20 segundos**

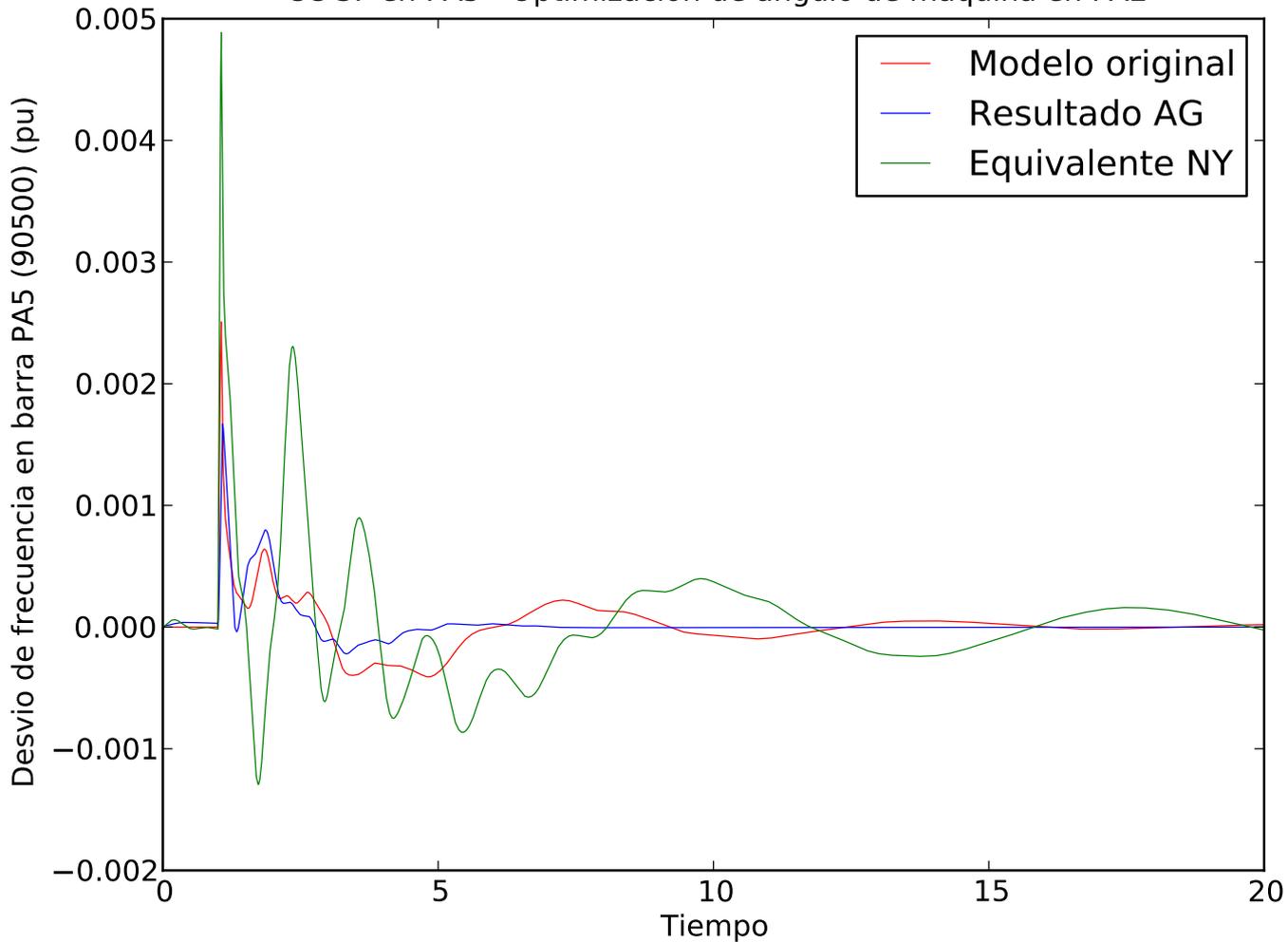
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



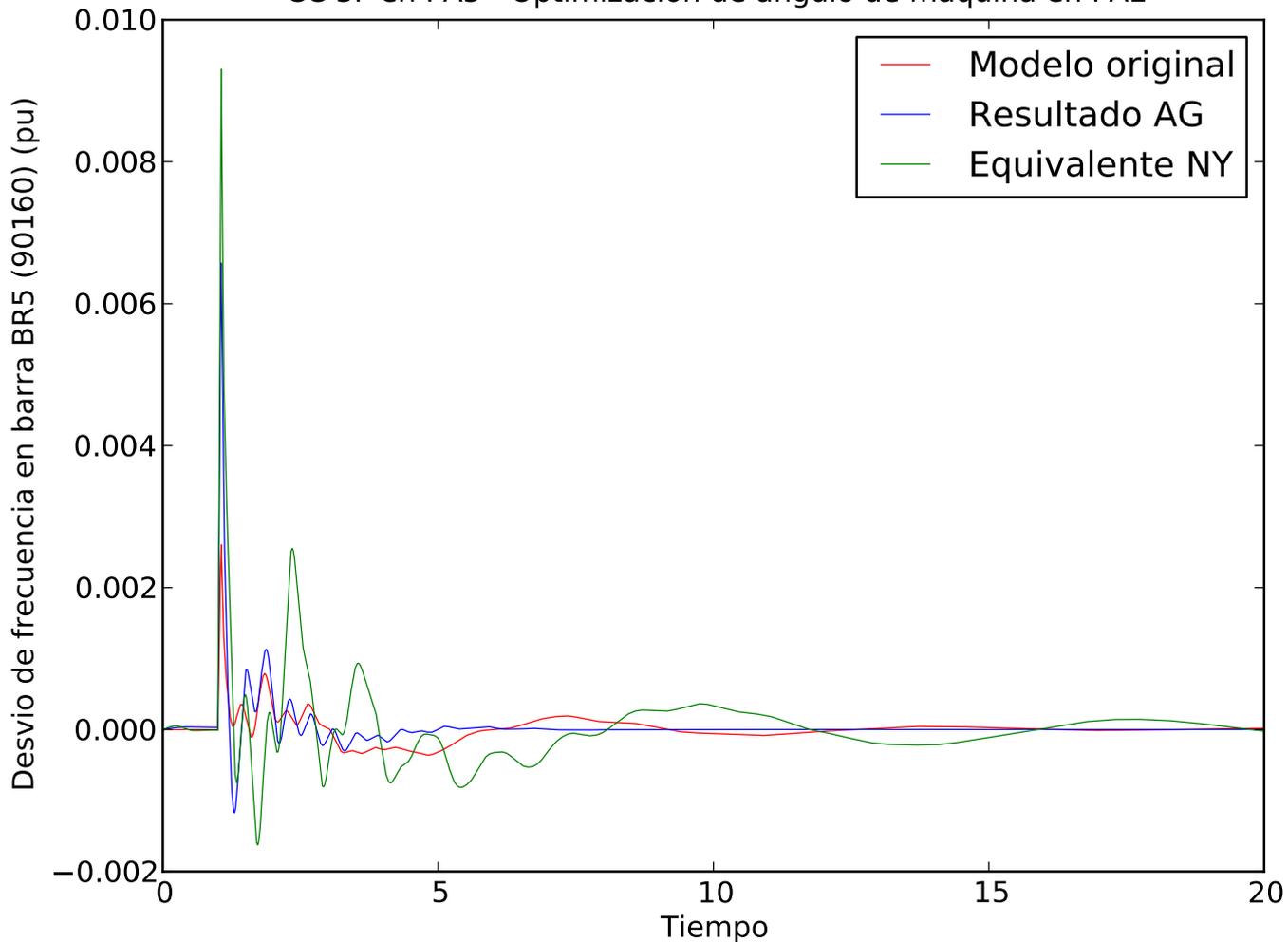
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



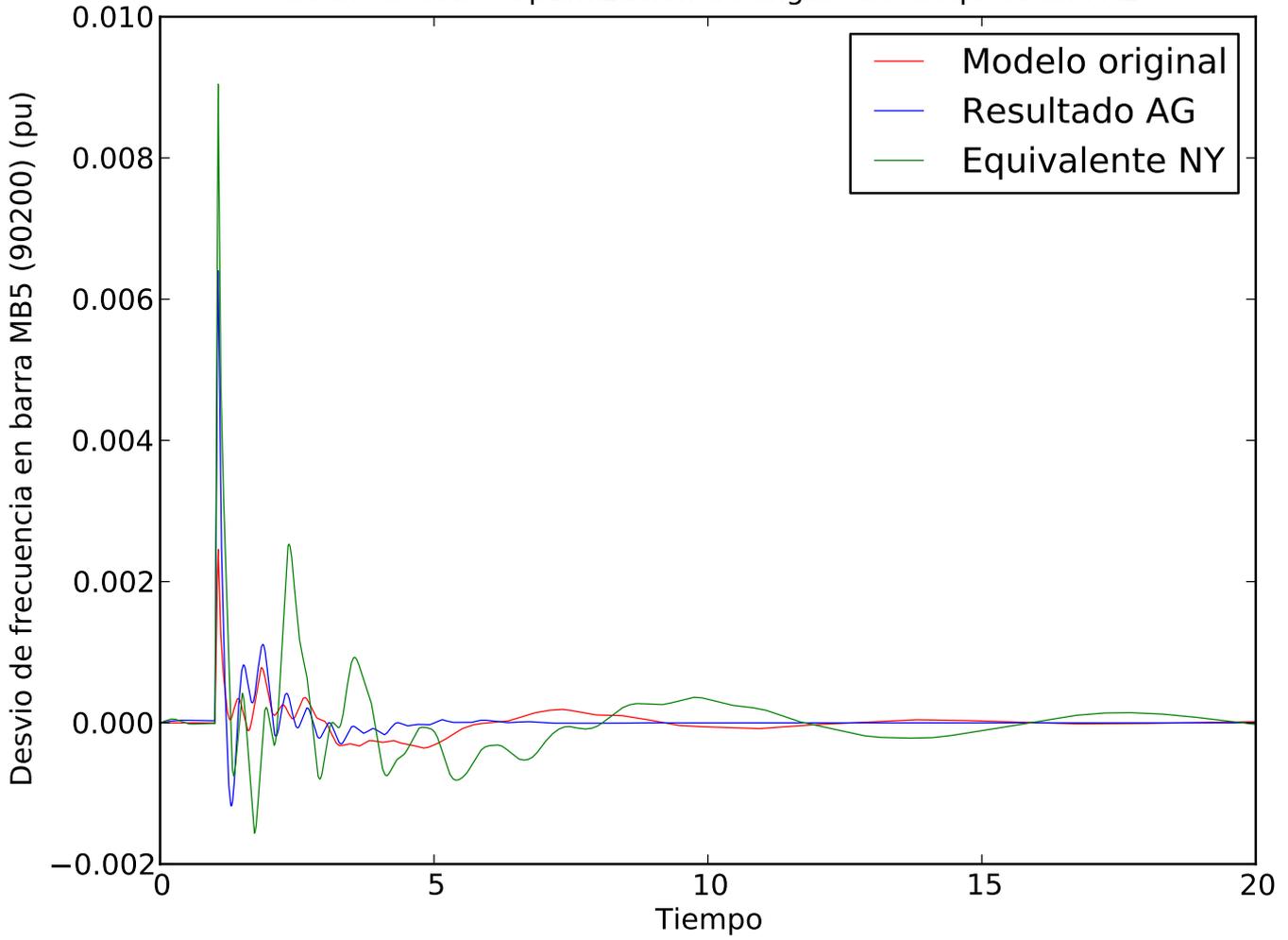
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



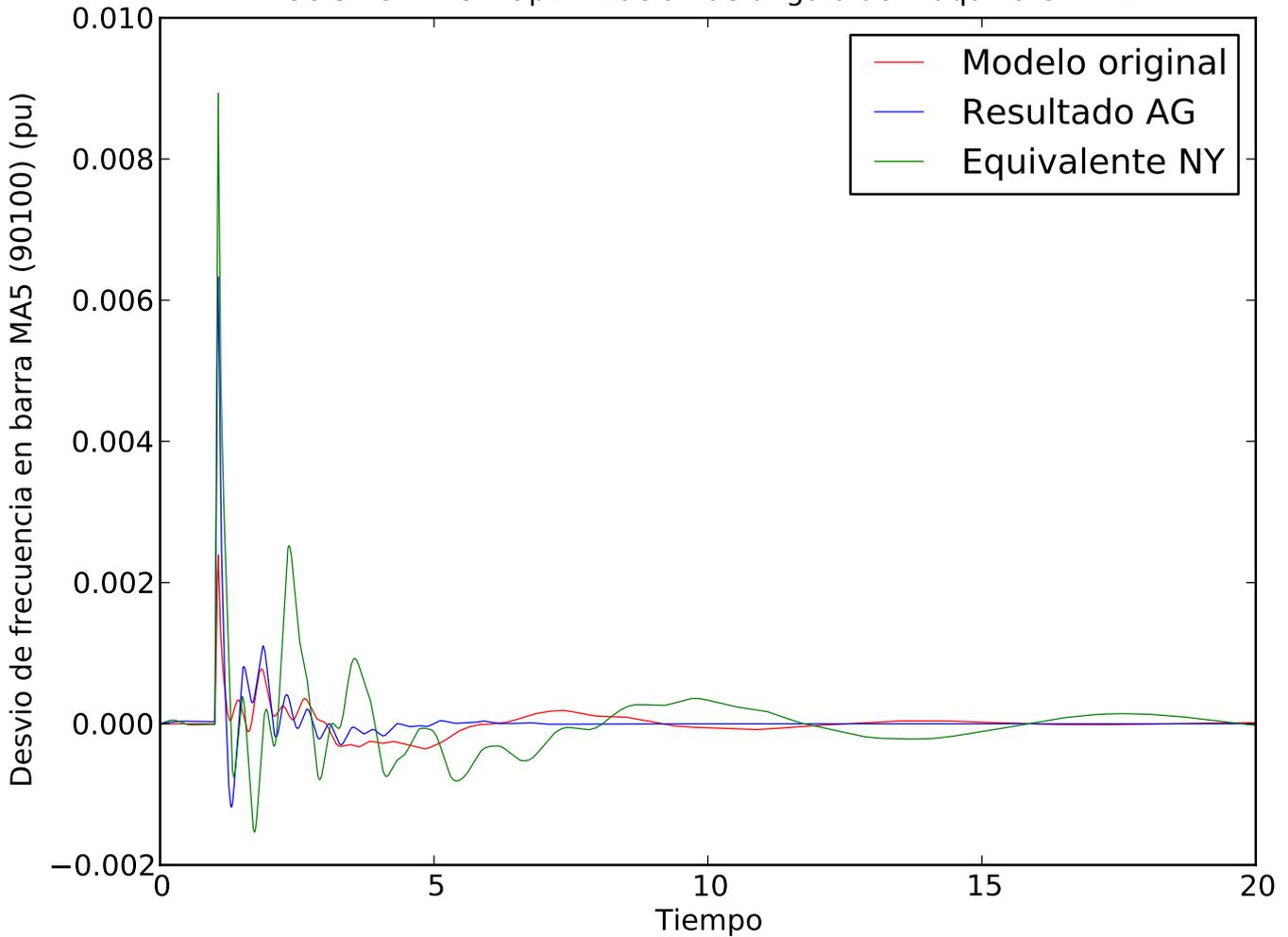
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



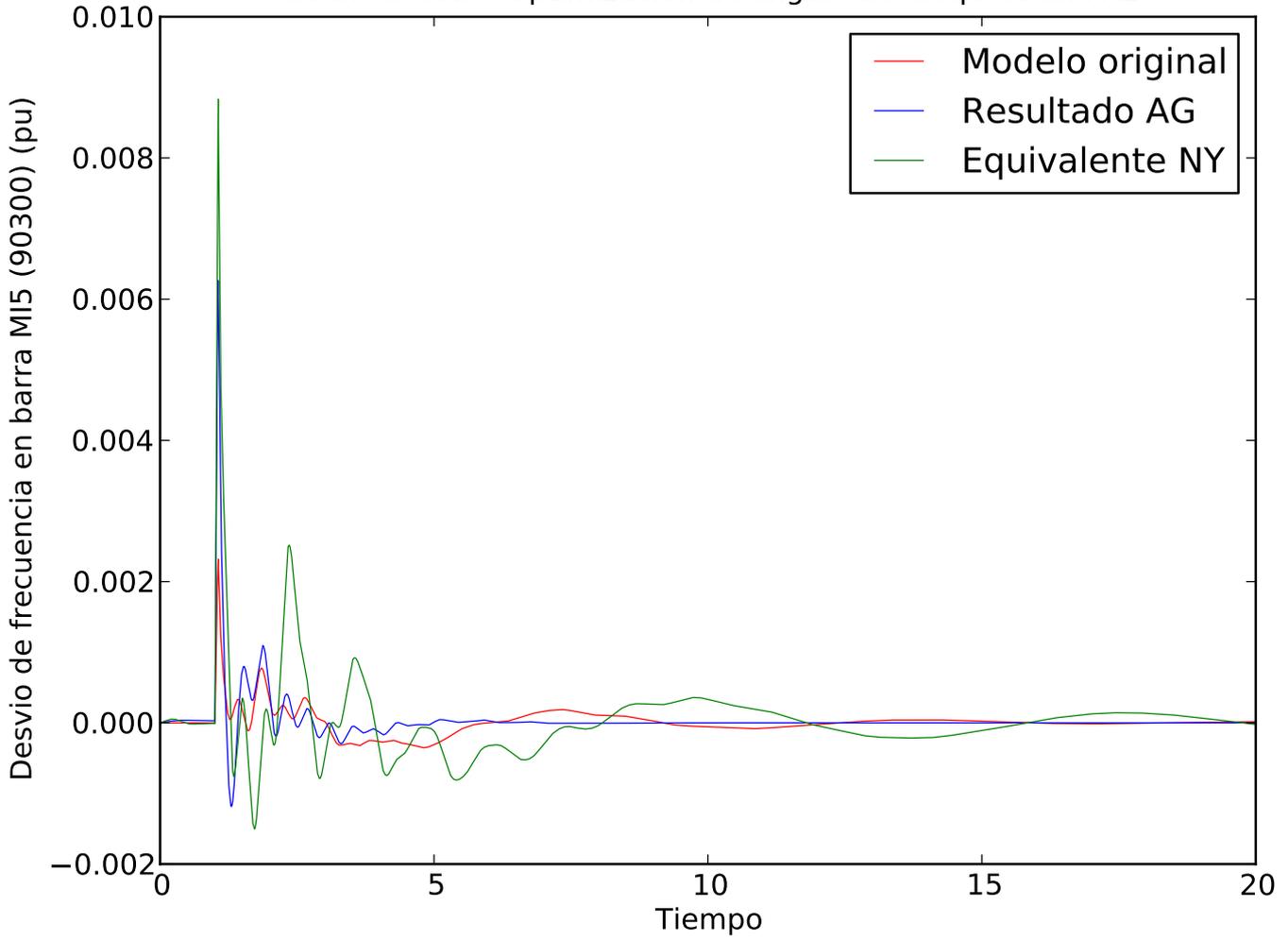
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



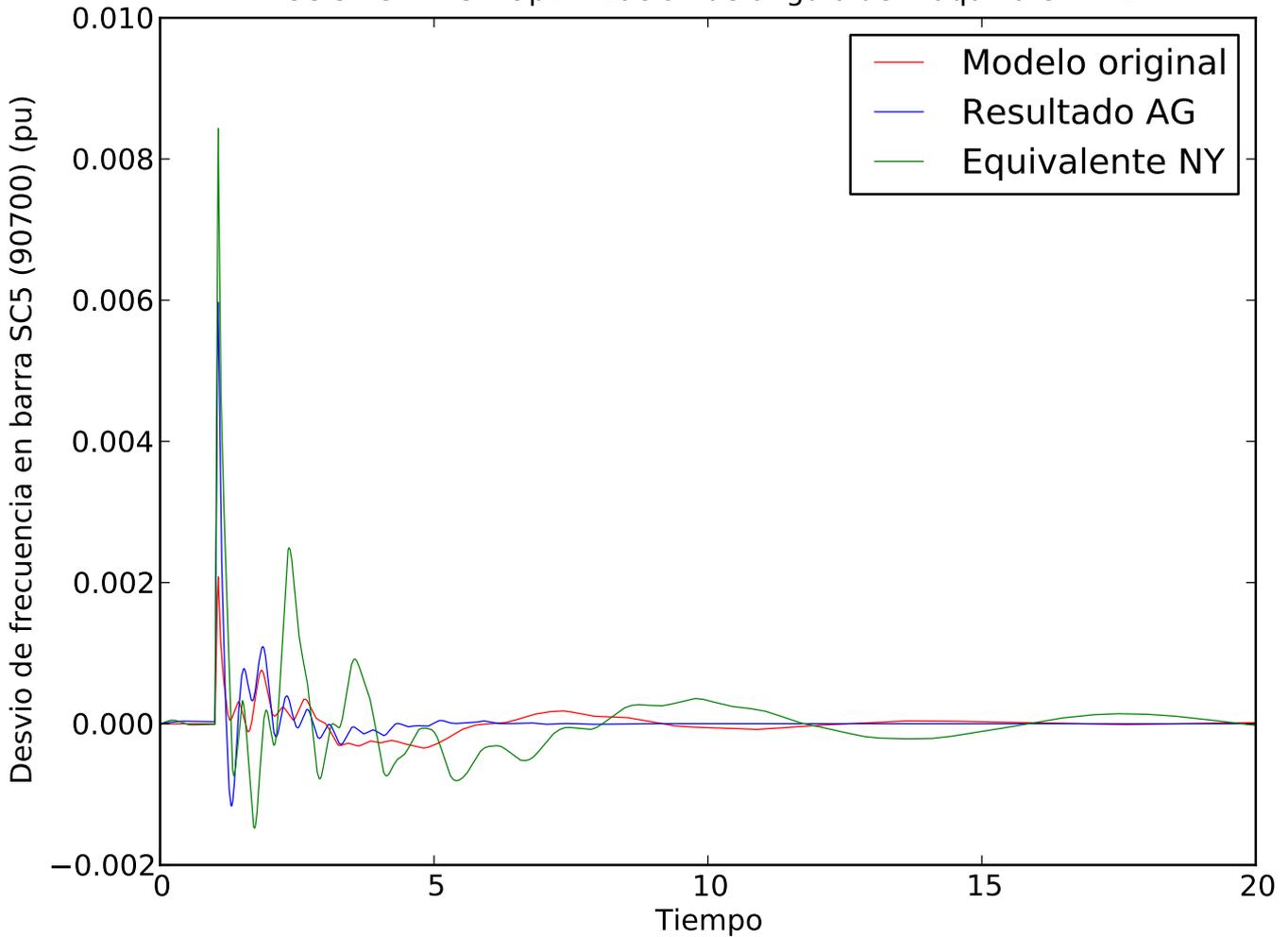
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



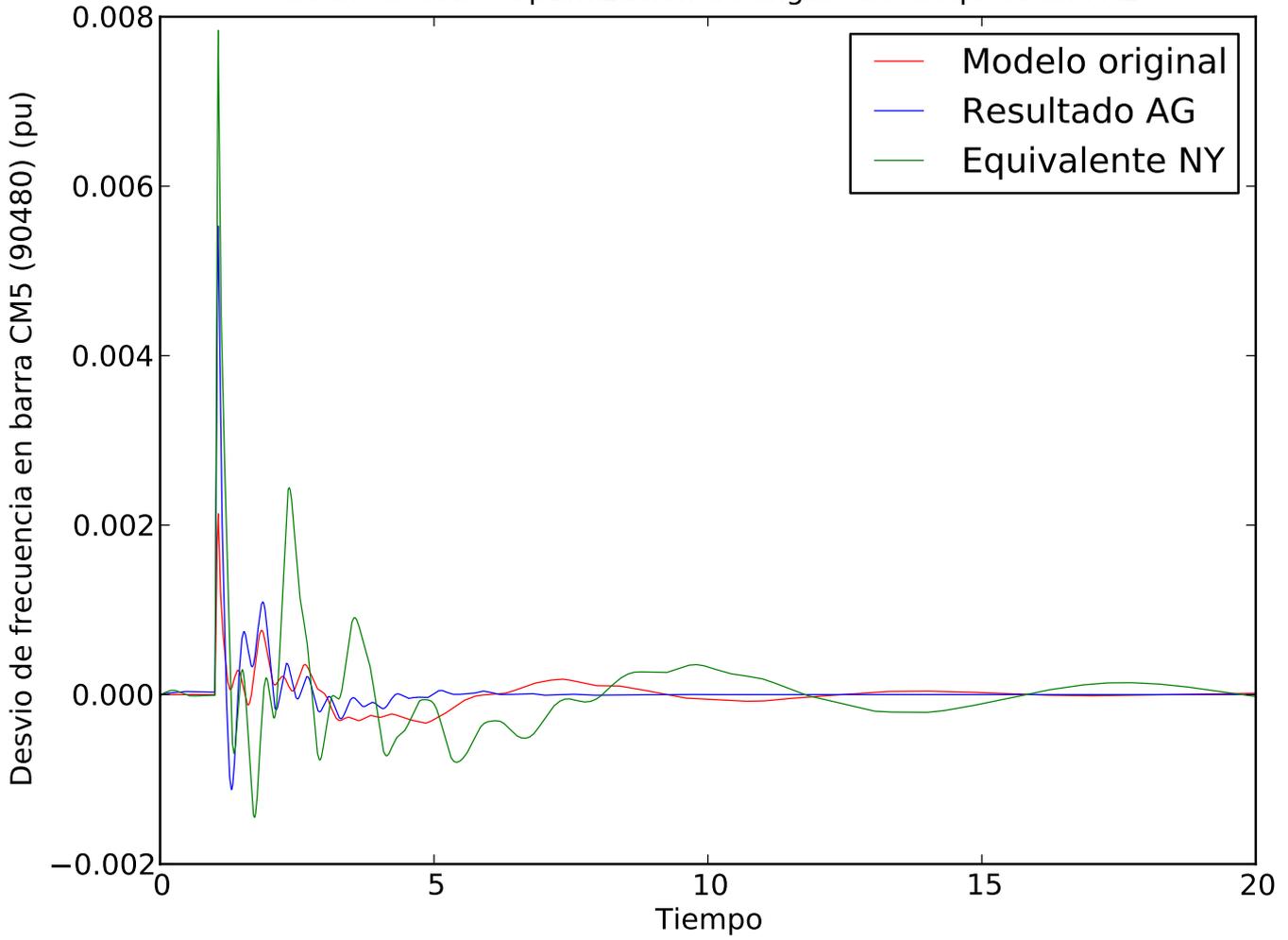
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



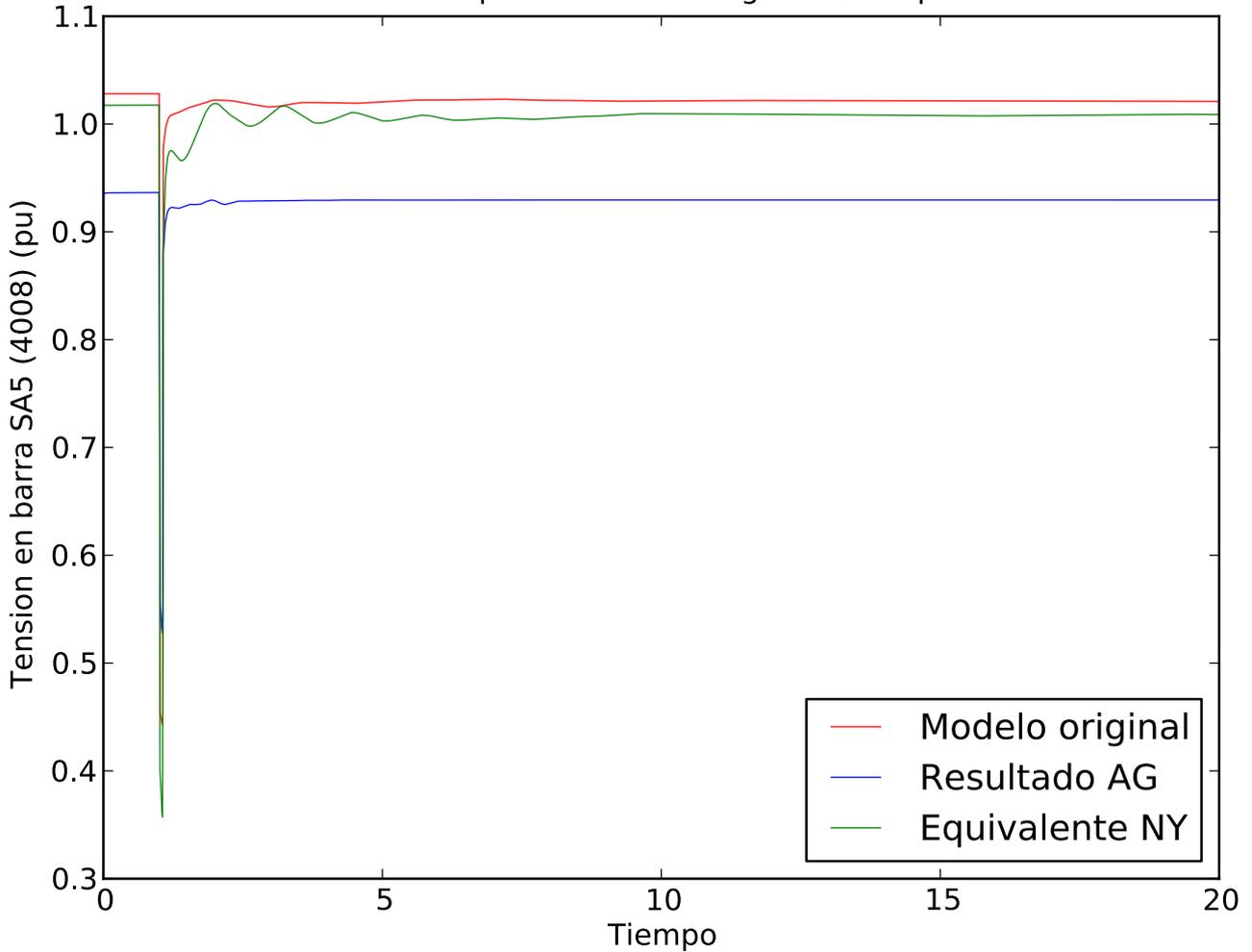
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



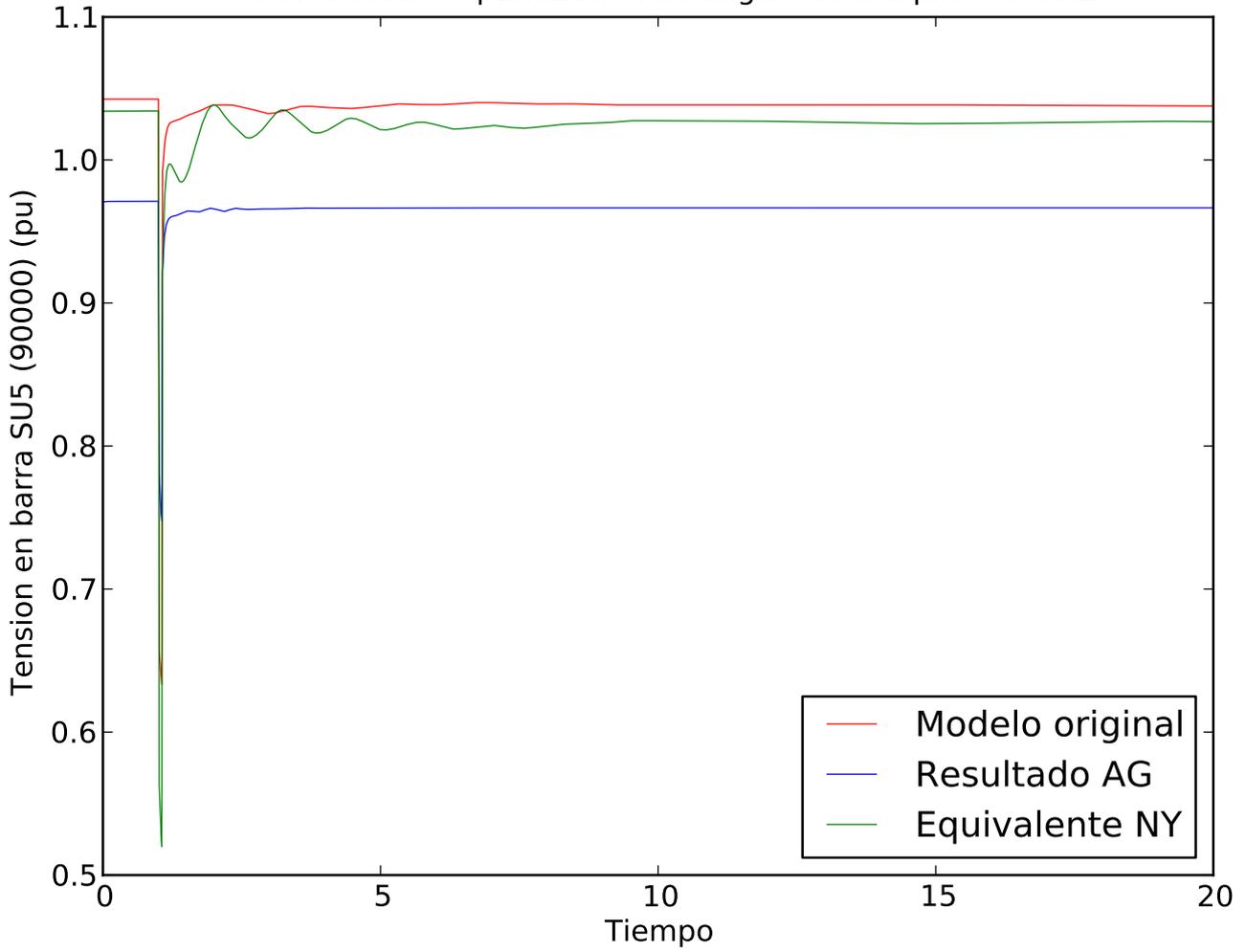
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



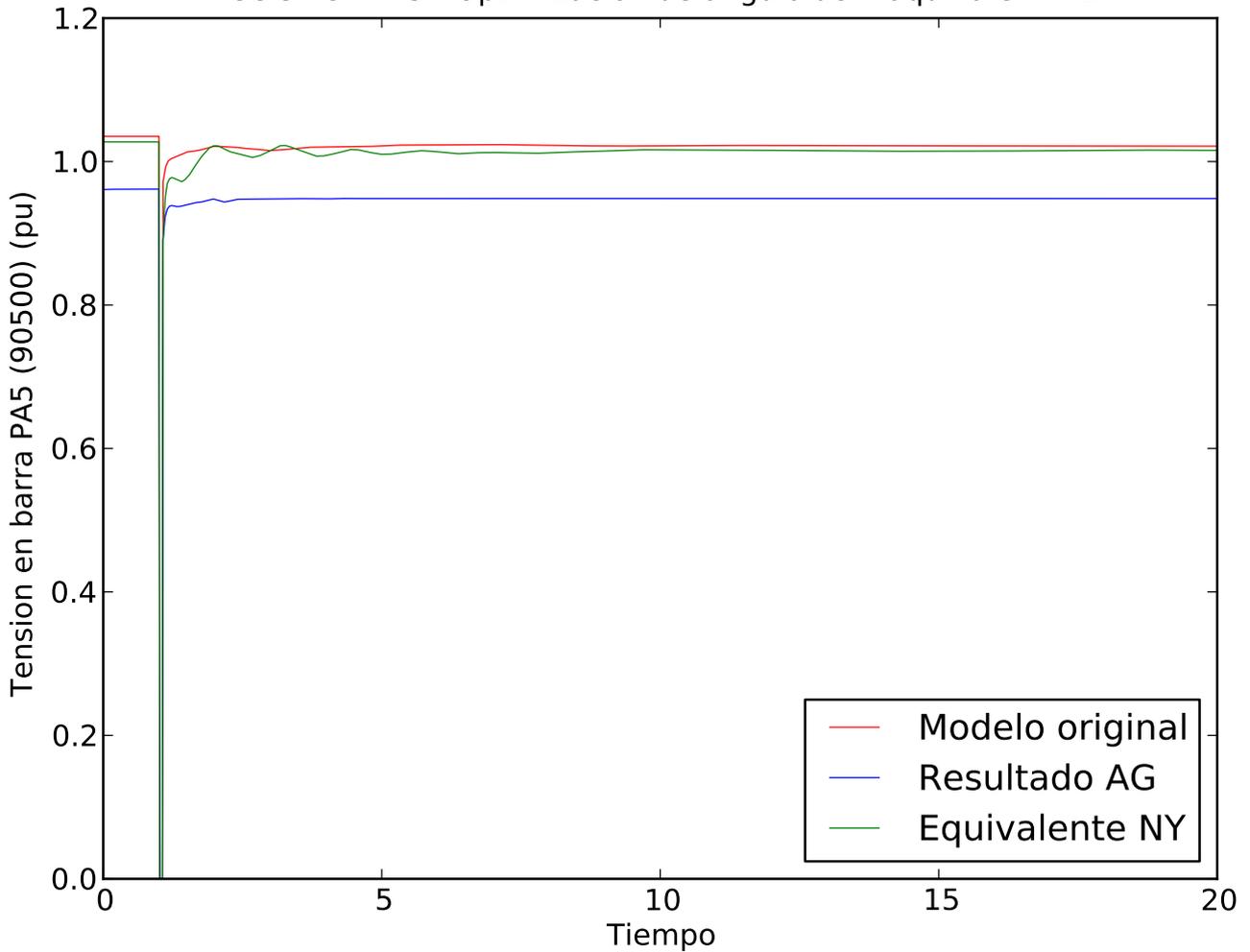
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



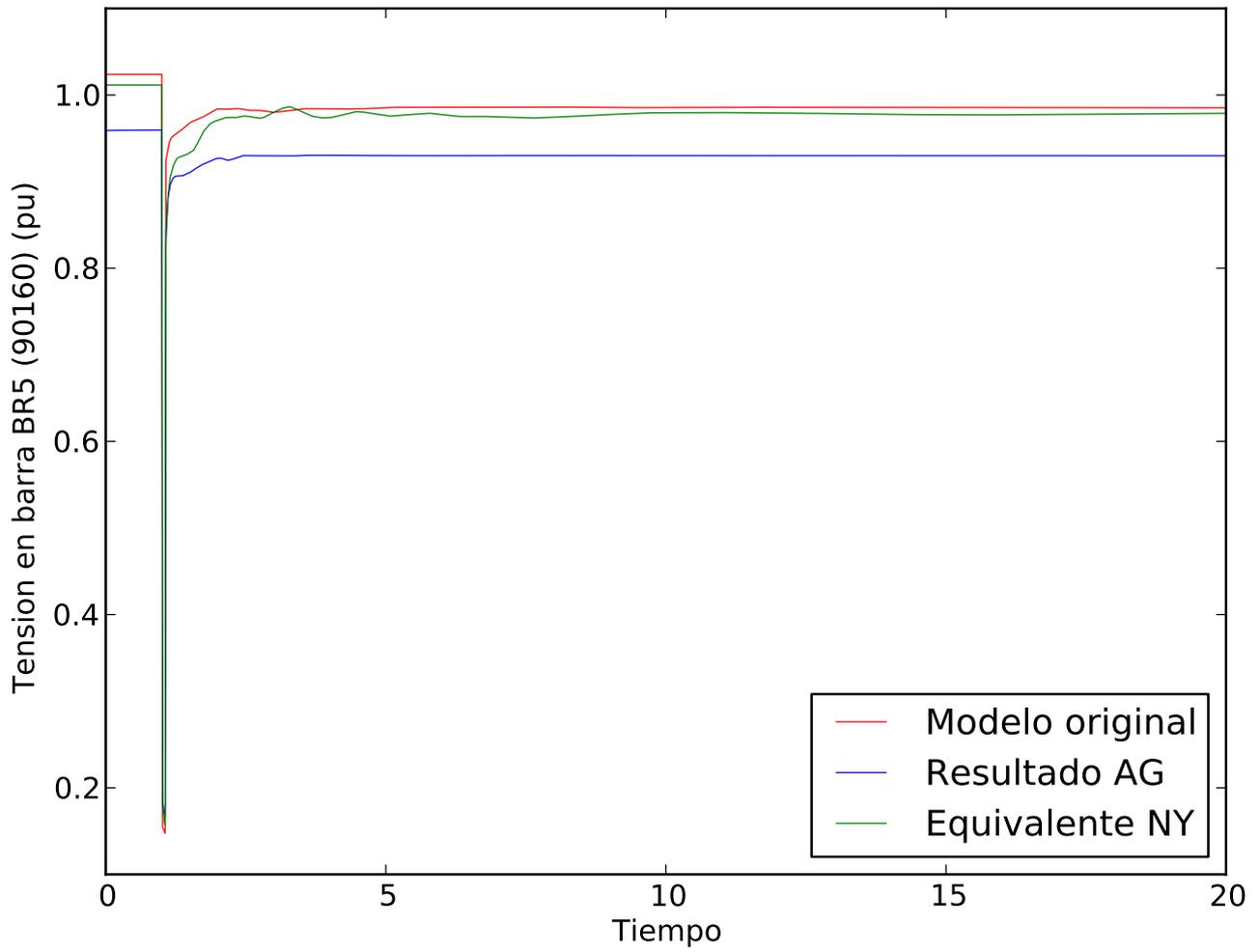
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



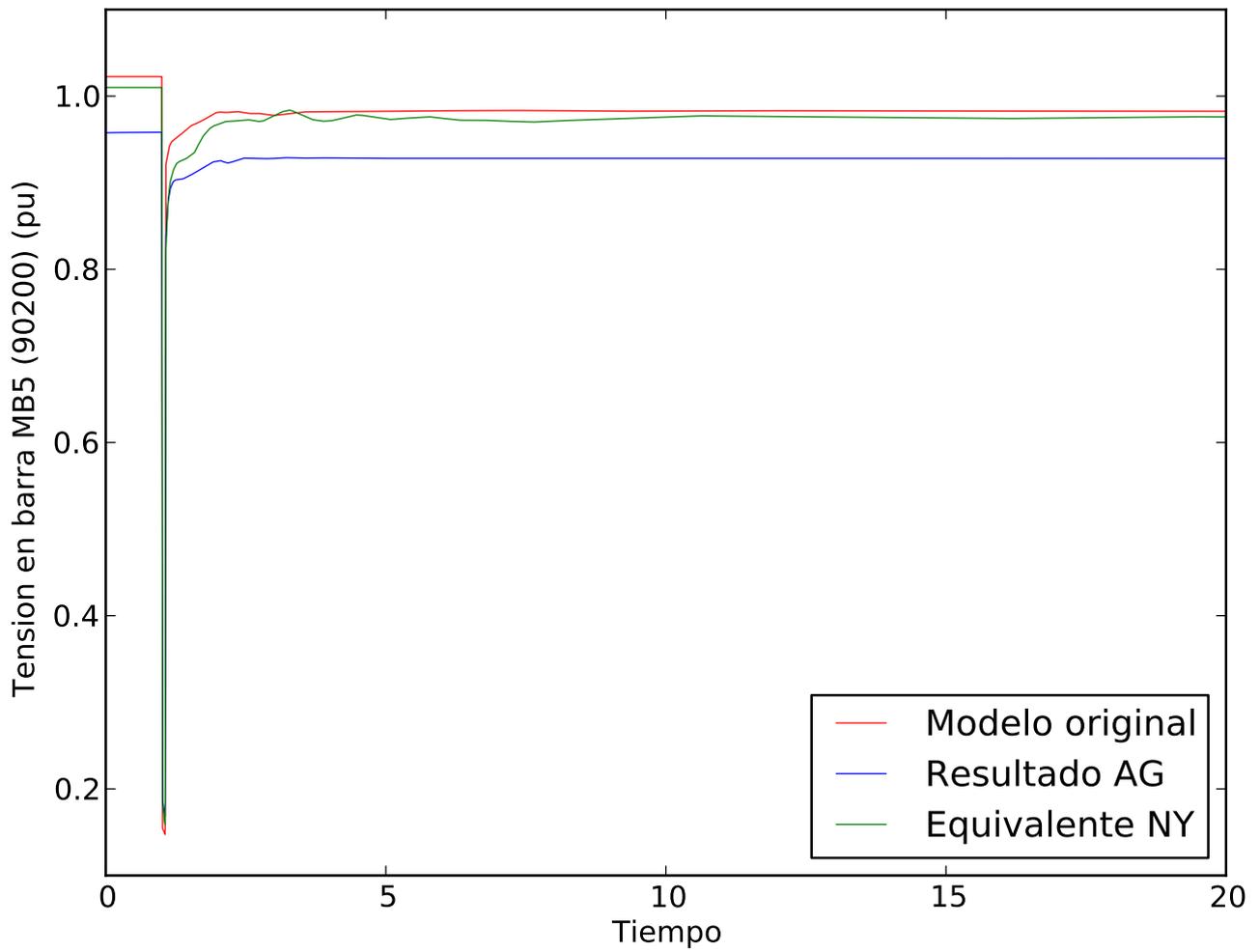
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



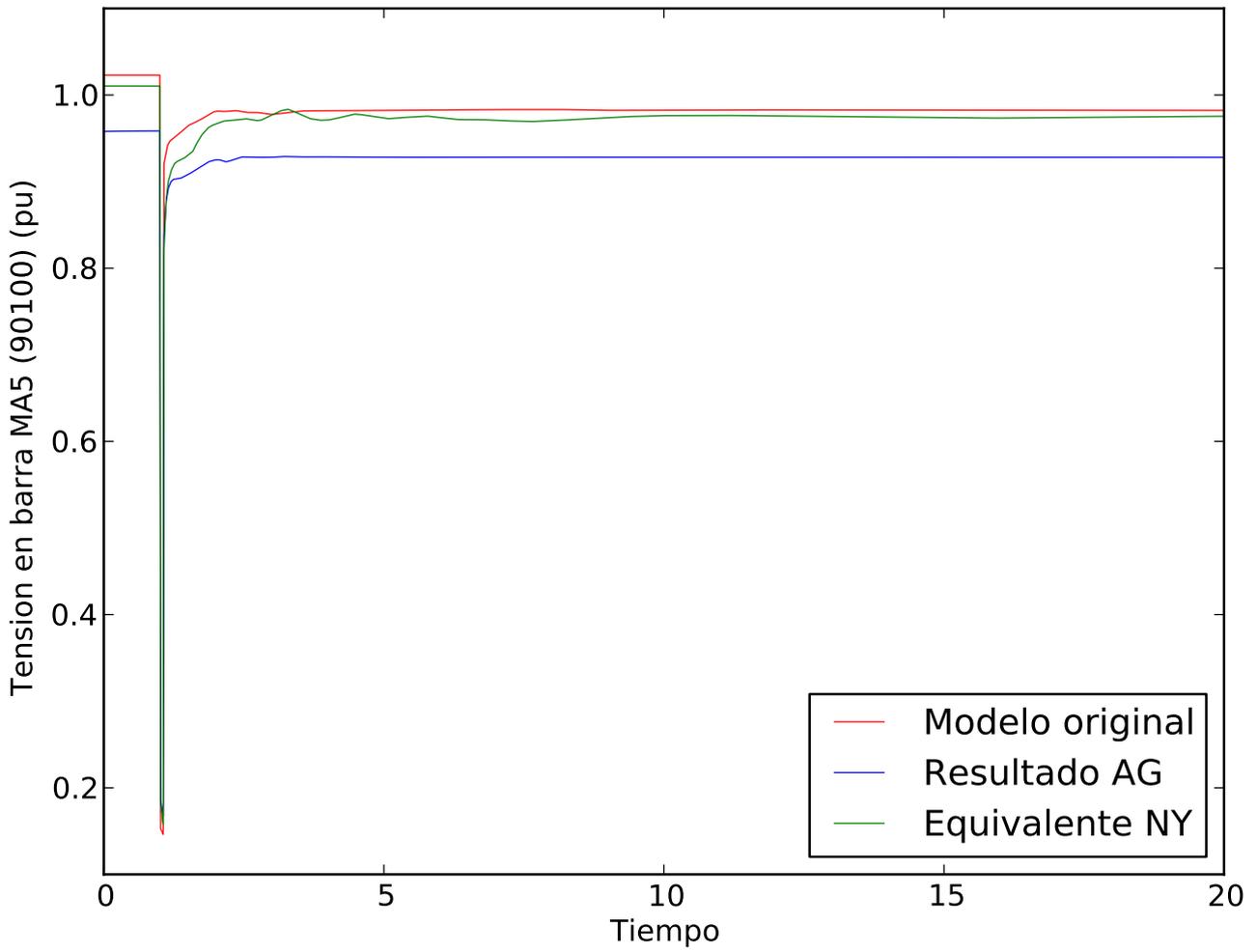
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



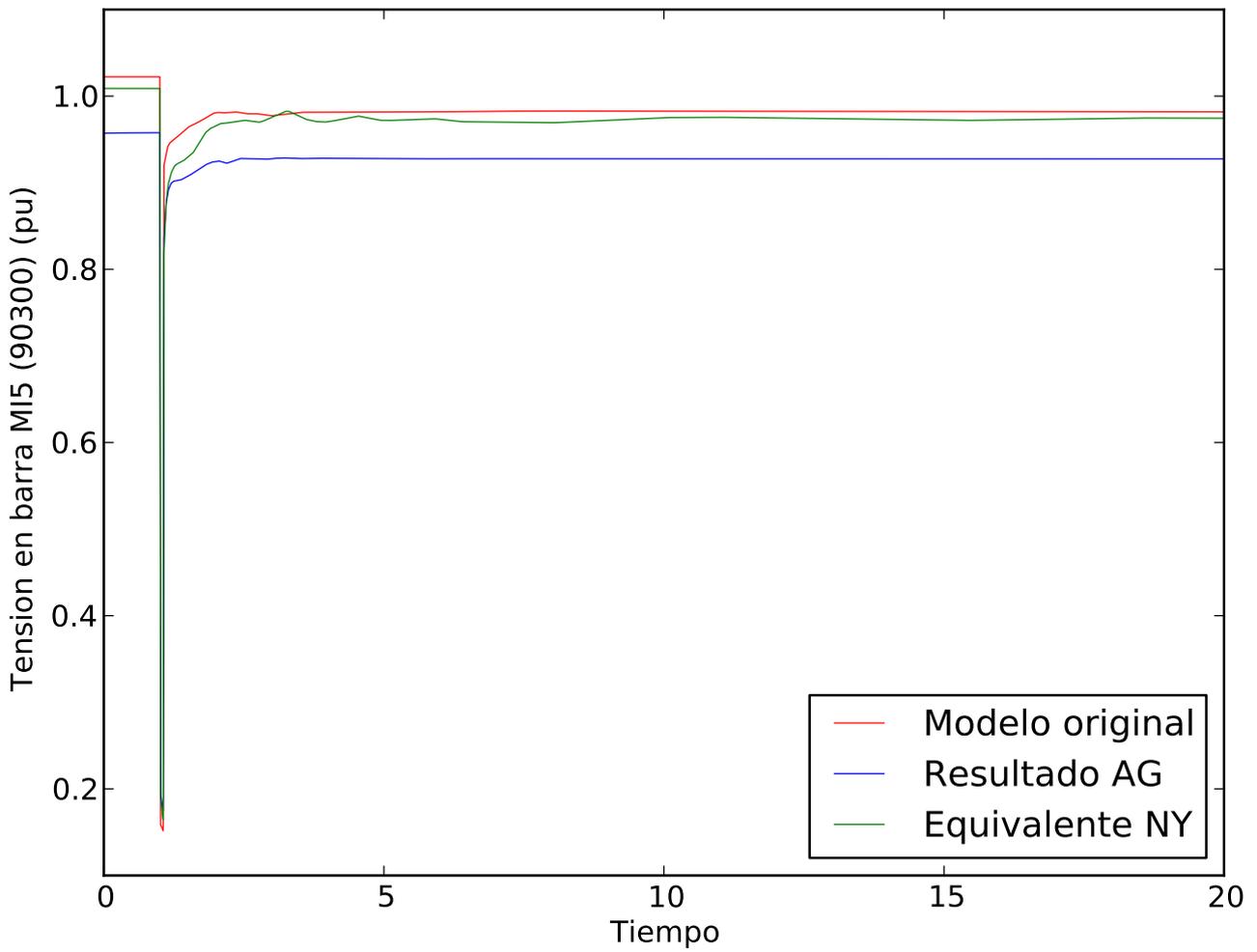
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



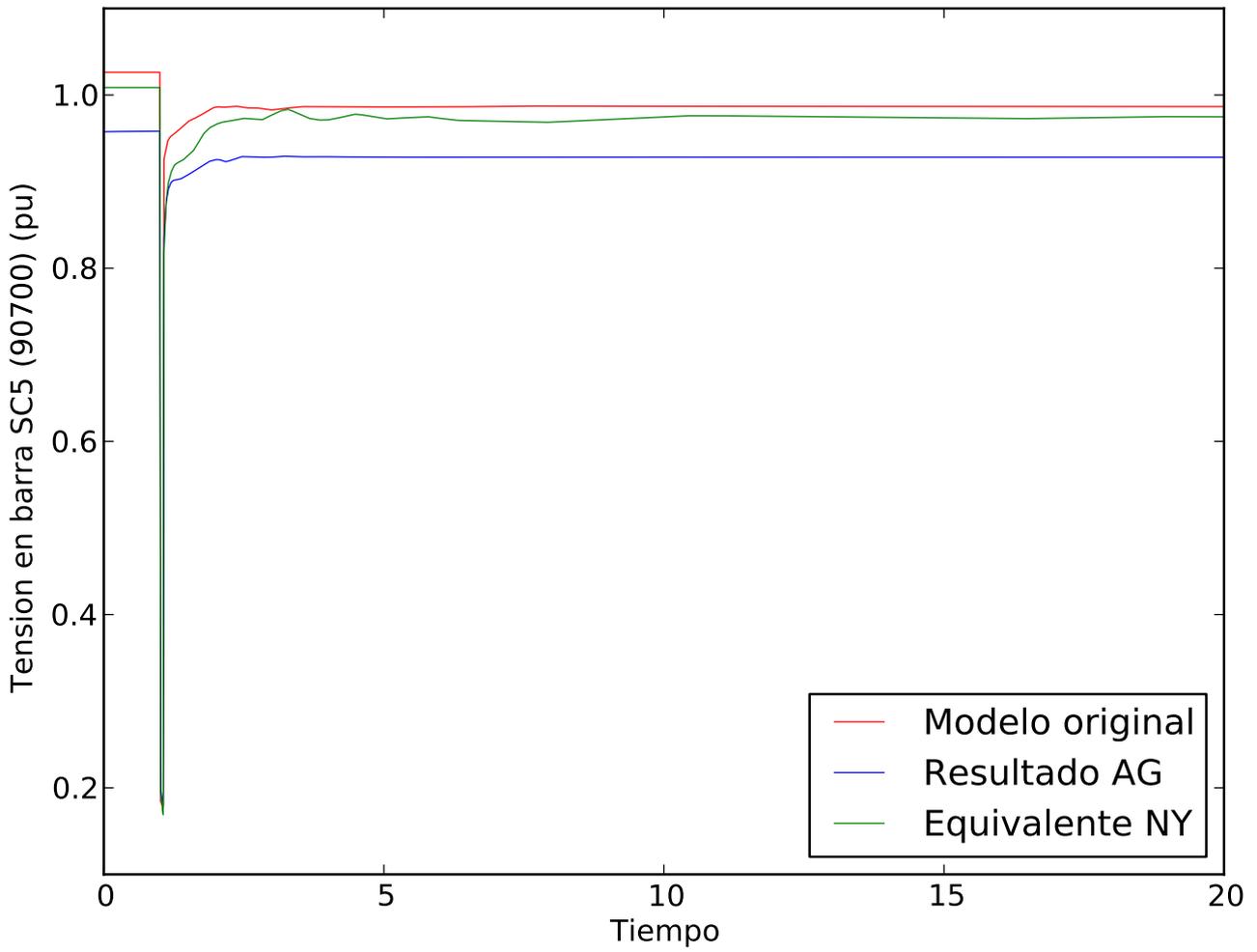
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



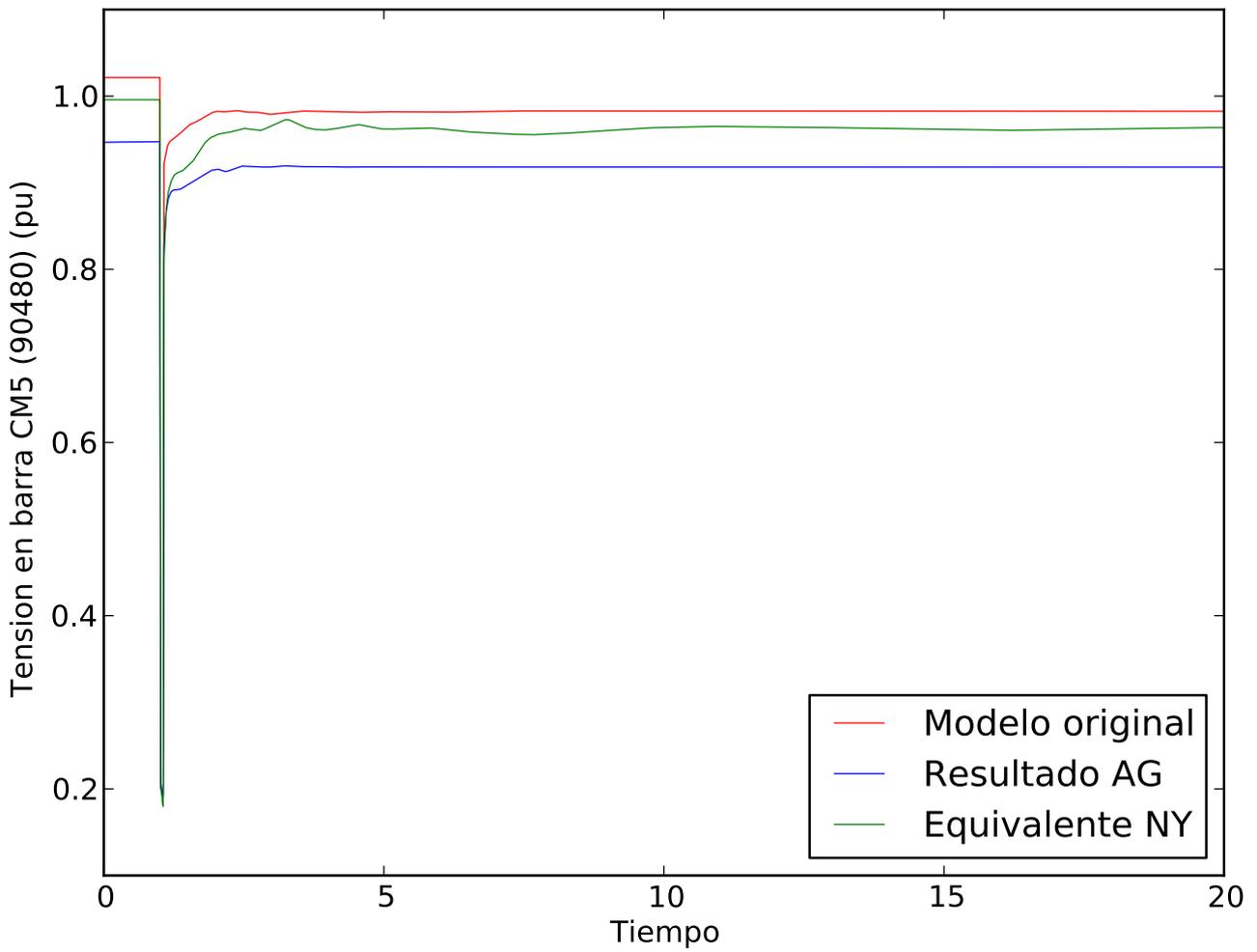
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



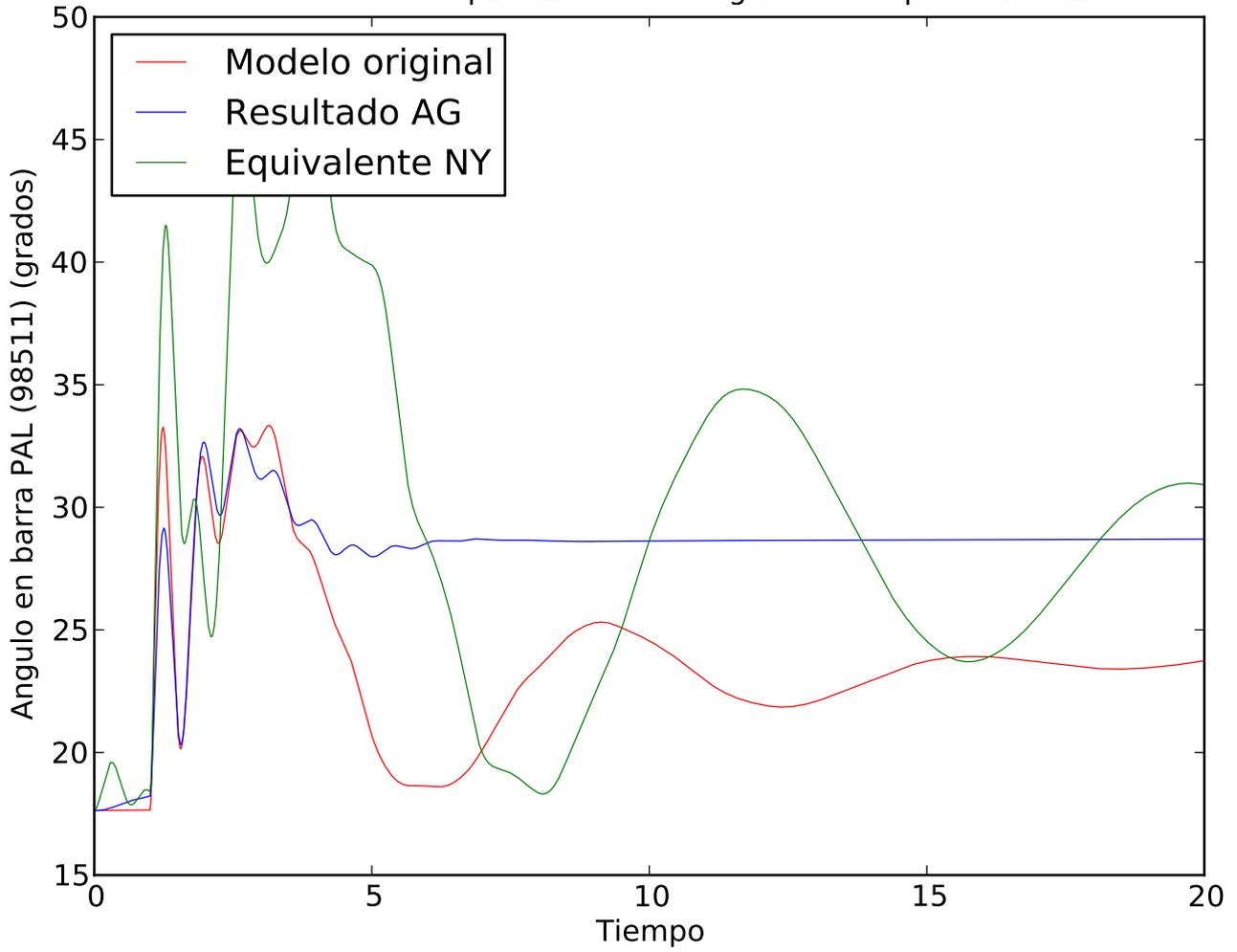
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



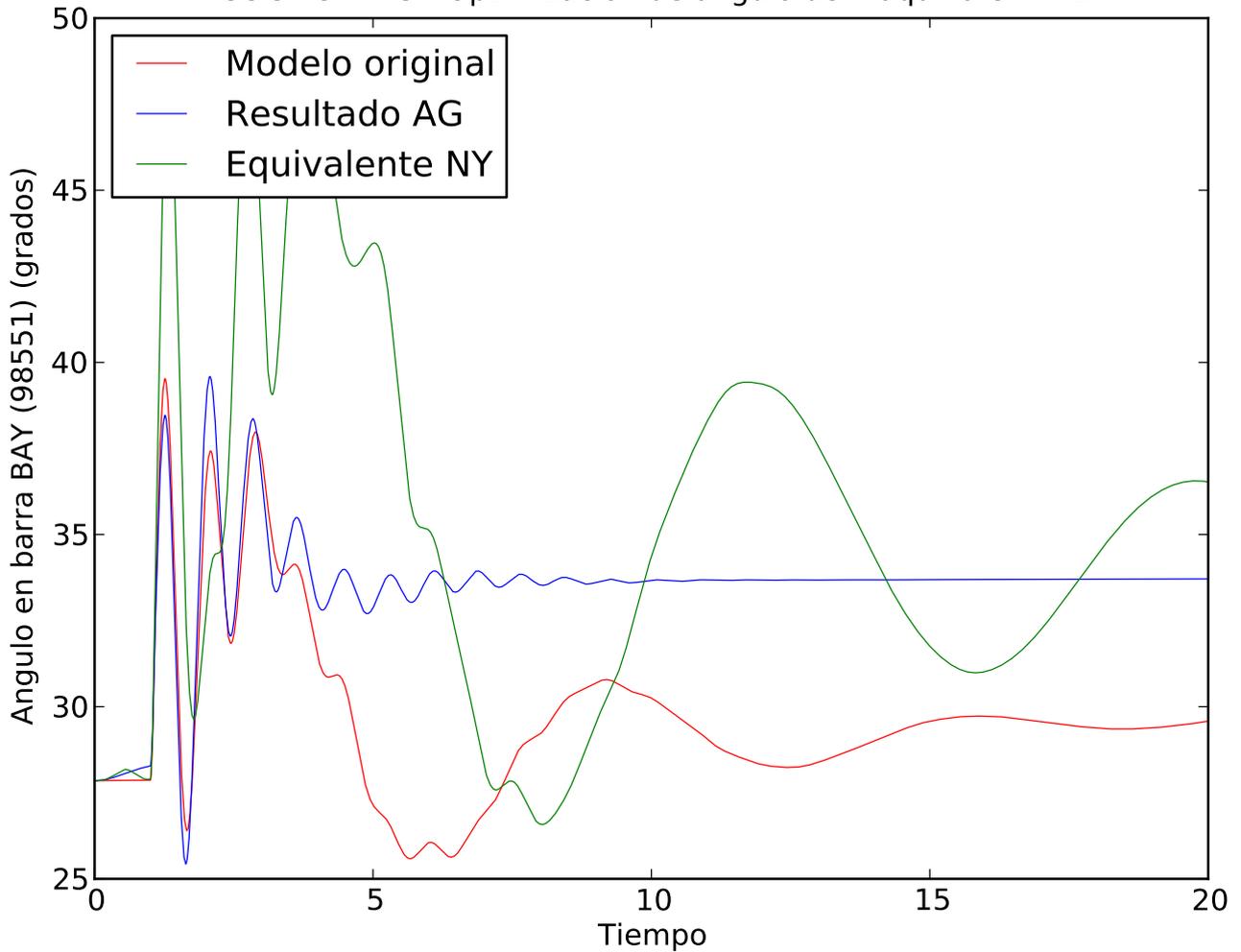
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



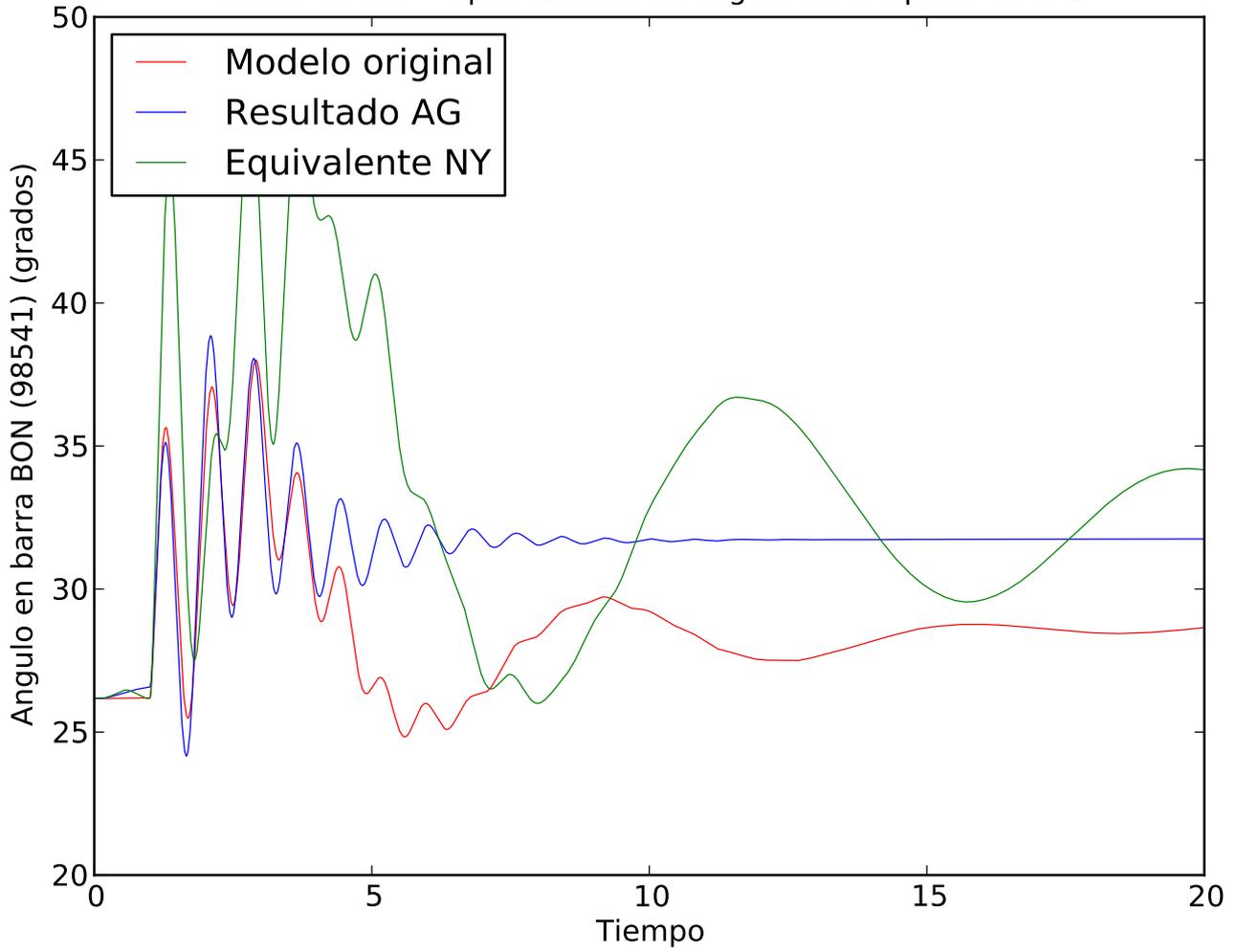
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



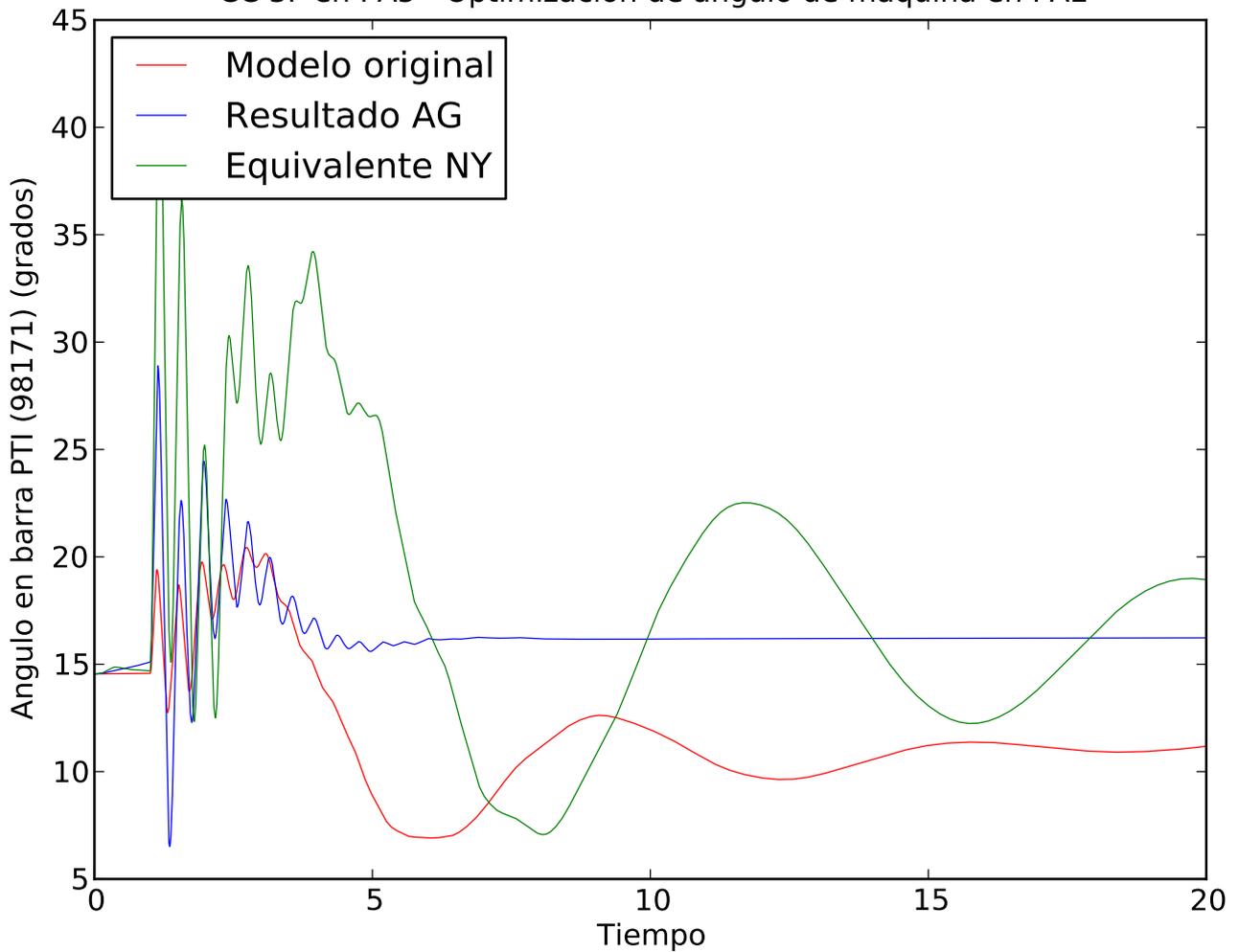
CC 3F en PA5 - Optimizacion de angulo de maquina en PAL



CC 3F en PA5 - Optimizacion de angulo de maquina en PAL

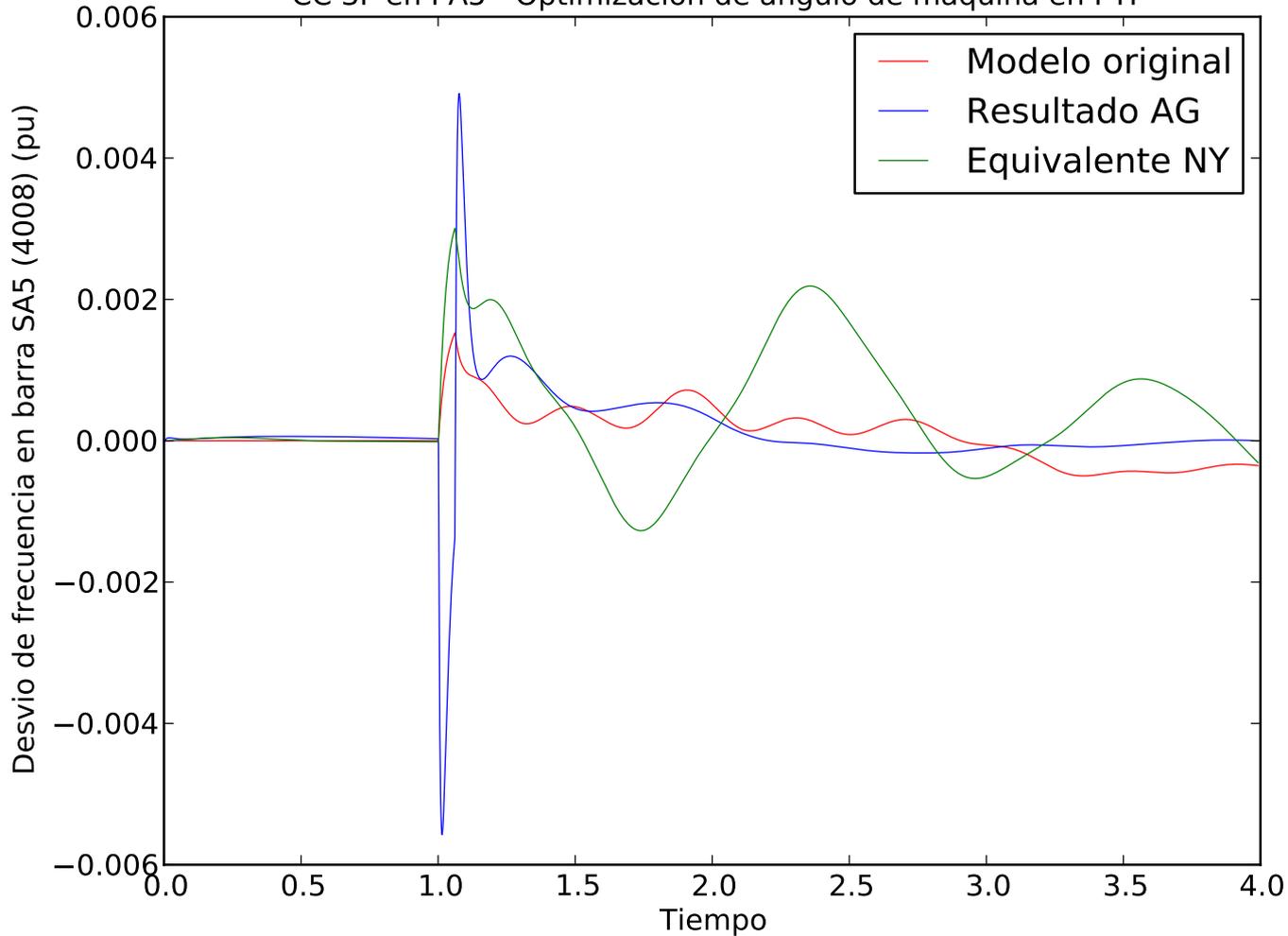


CC 3F en PA5 - Optimizacion de angulo de maquina en PAL

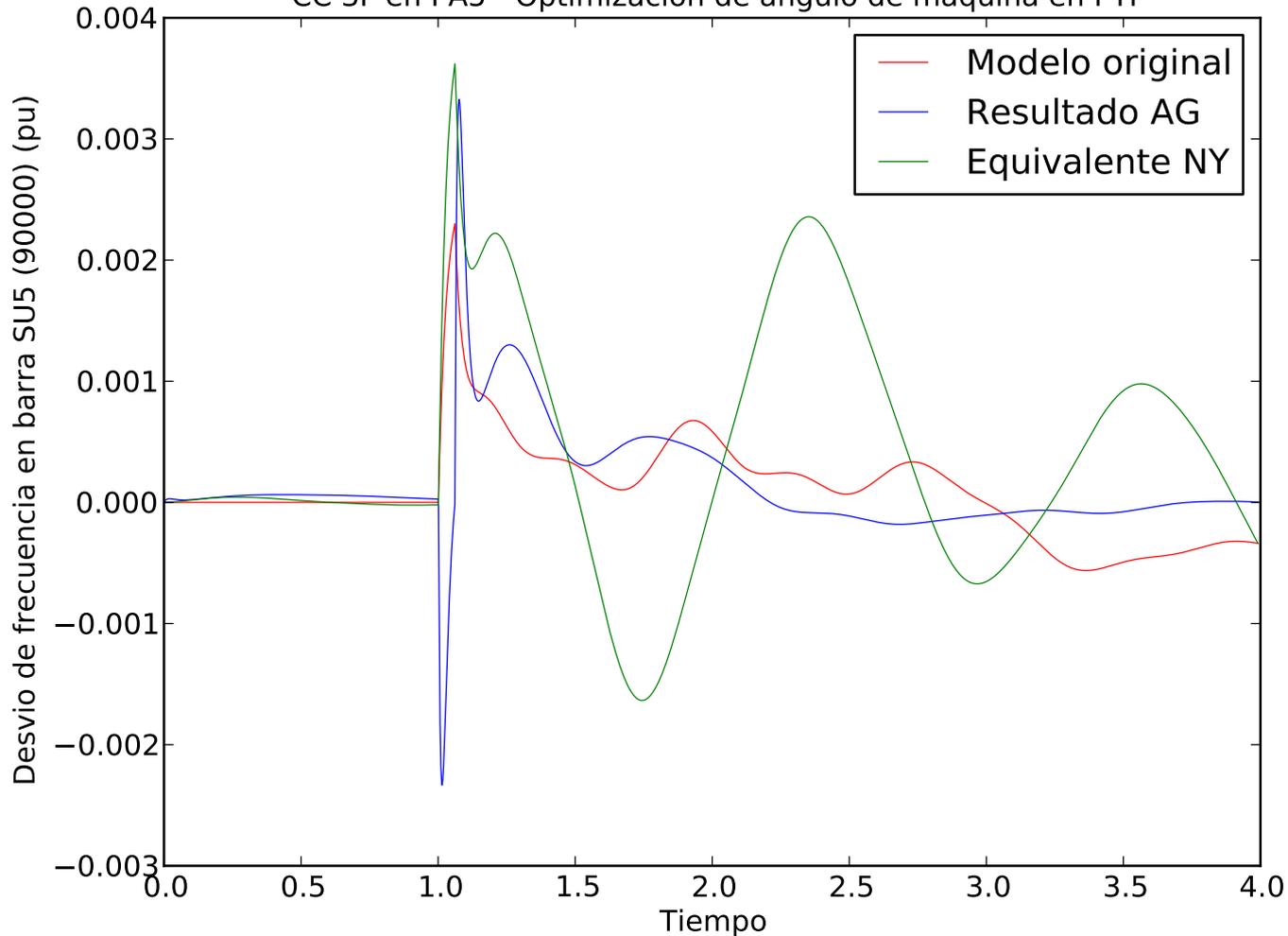


- C.6. Simulación 4.1: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando ángulo en una máquina de PTI en 4 segundos. Comportamiento en 4 segundos**

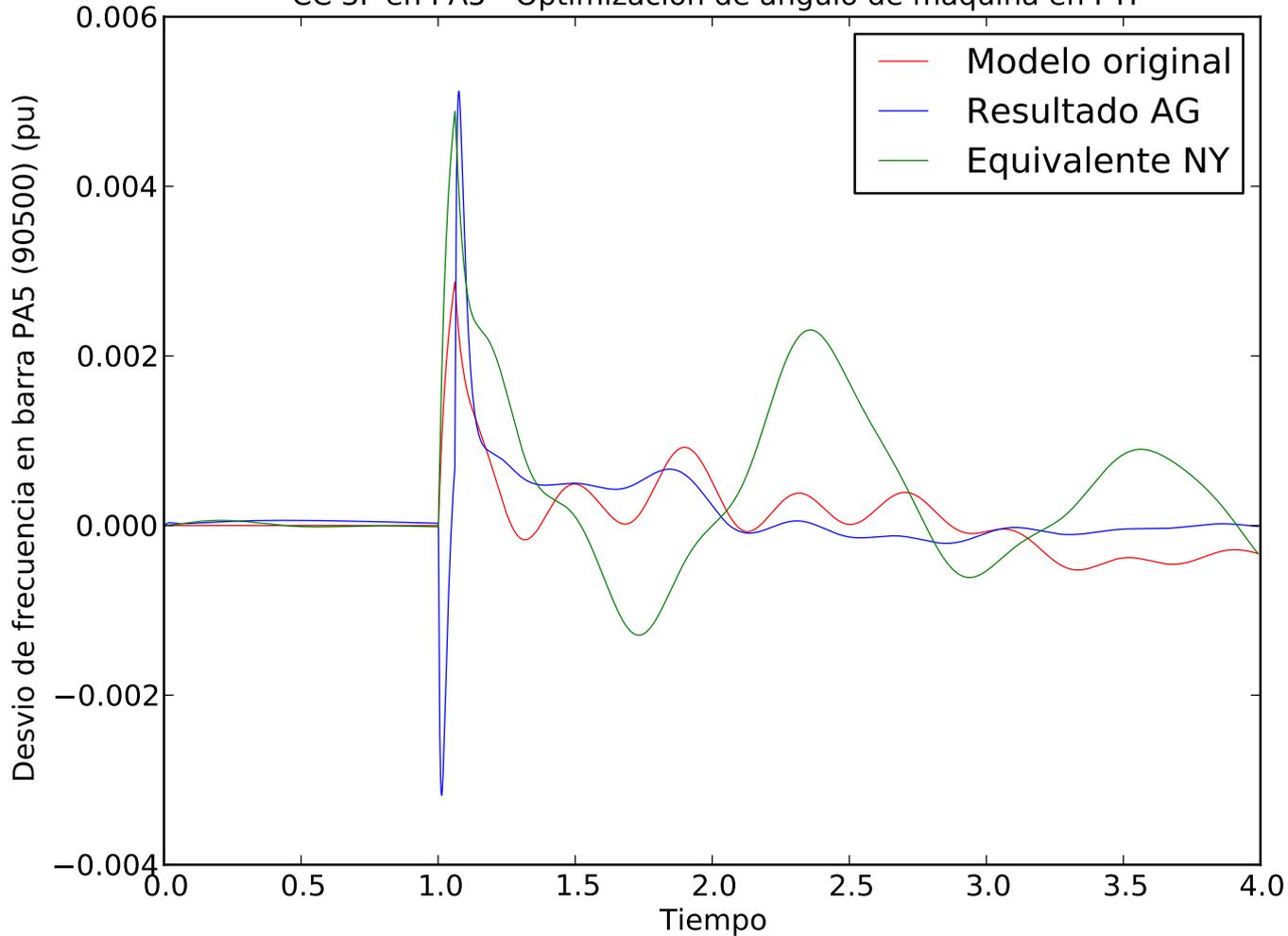
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



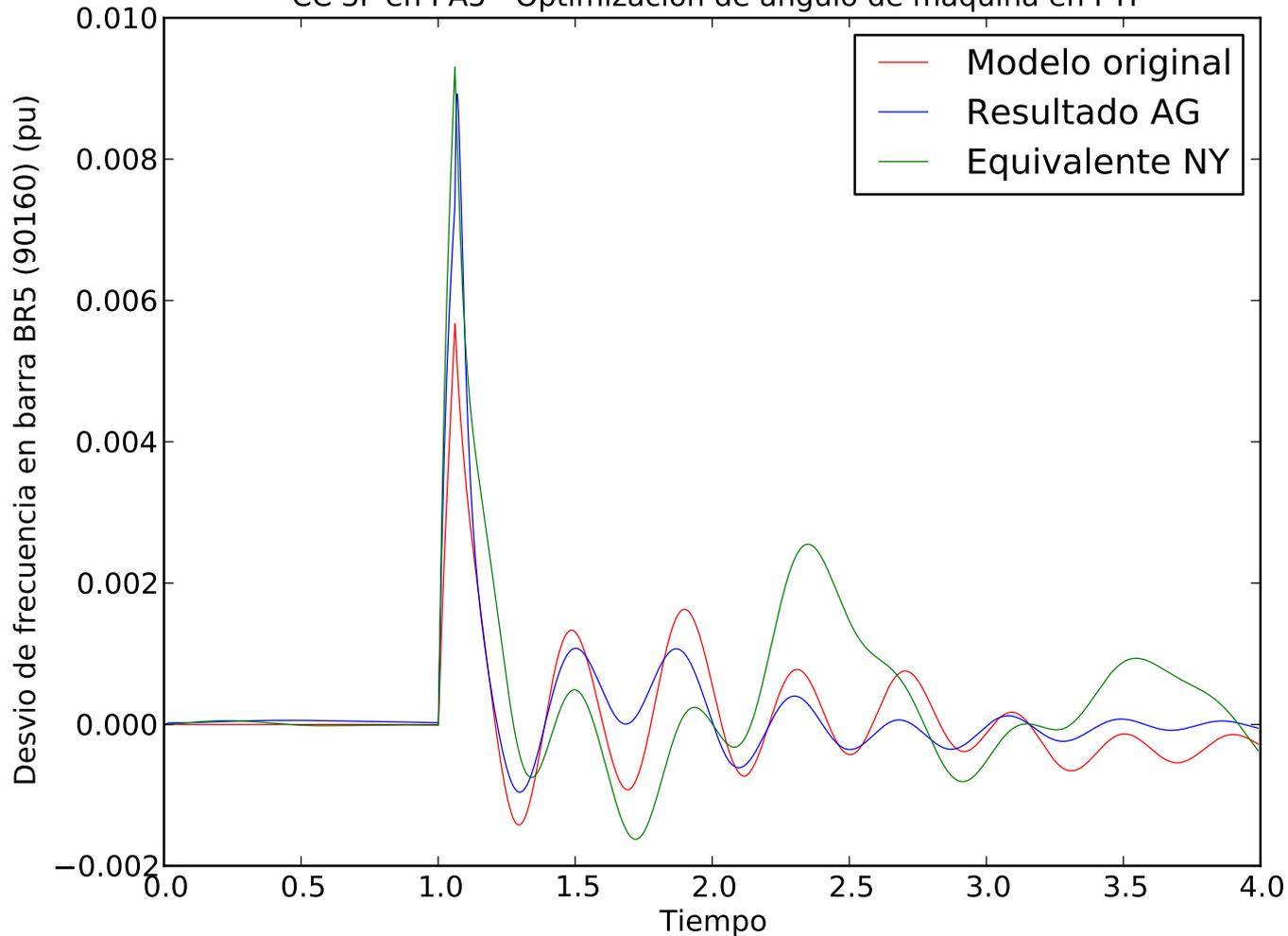
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



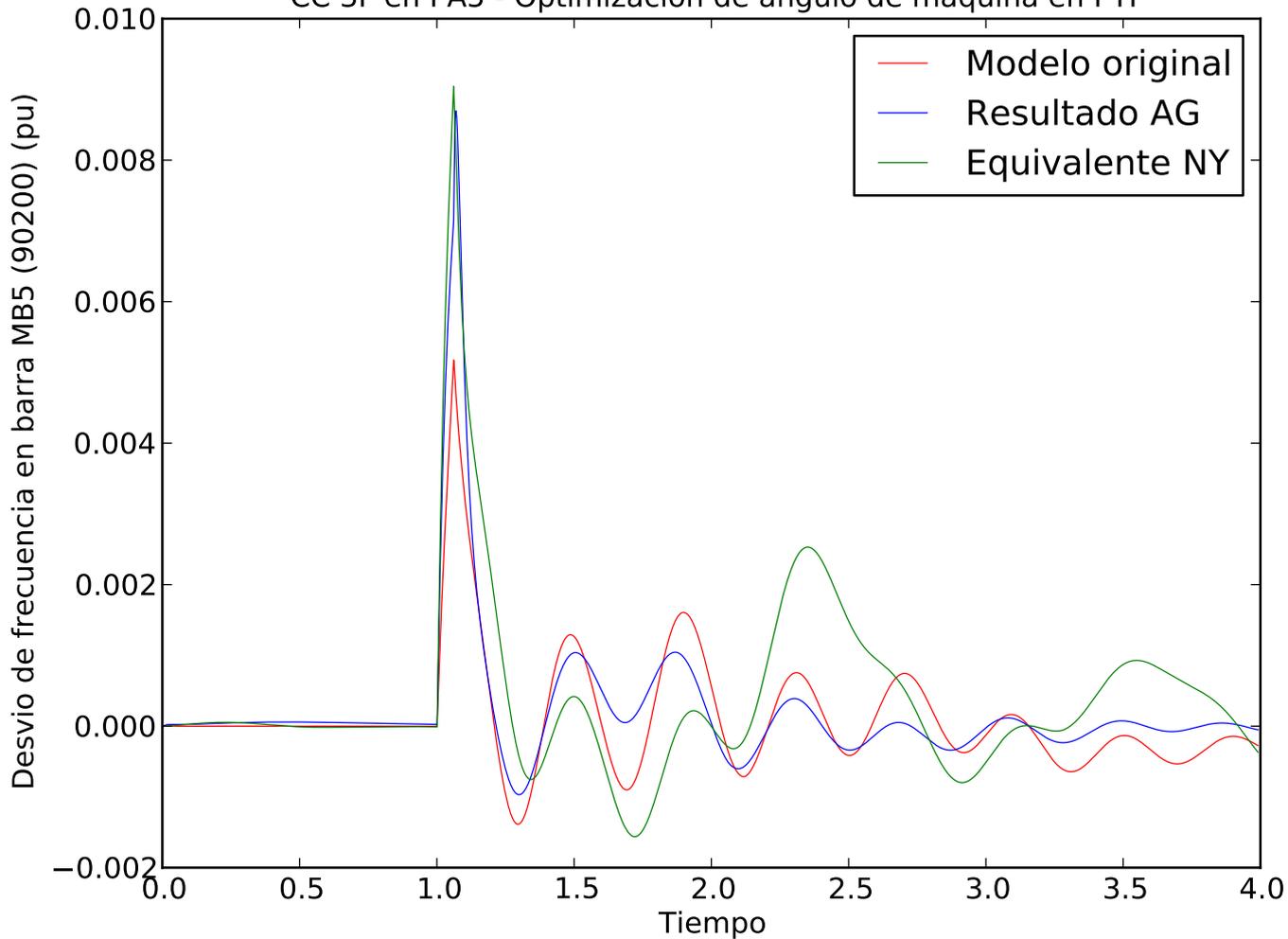
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



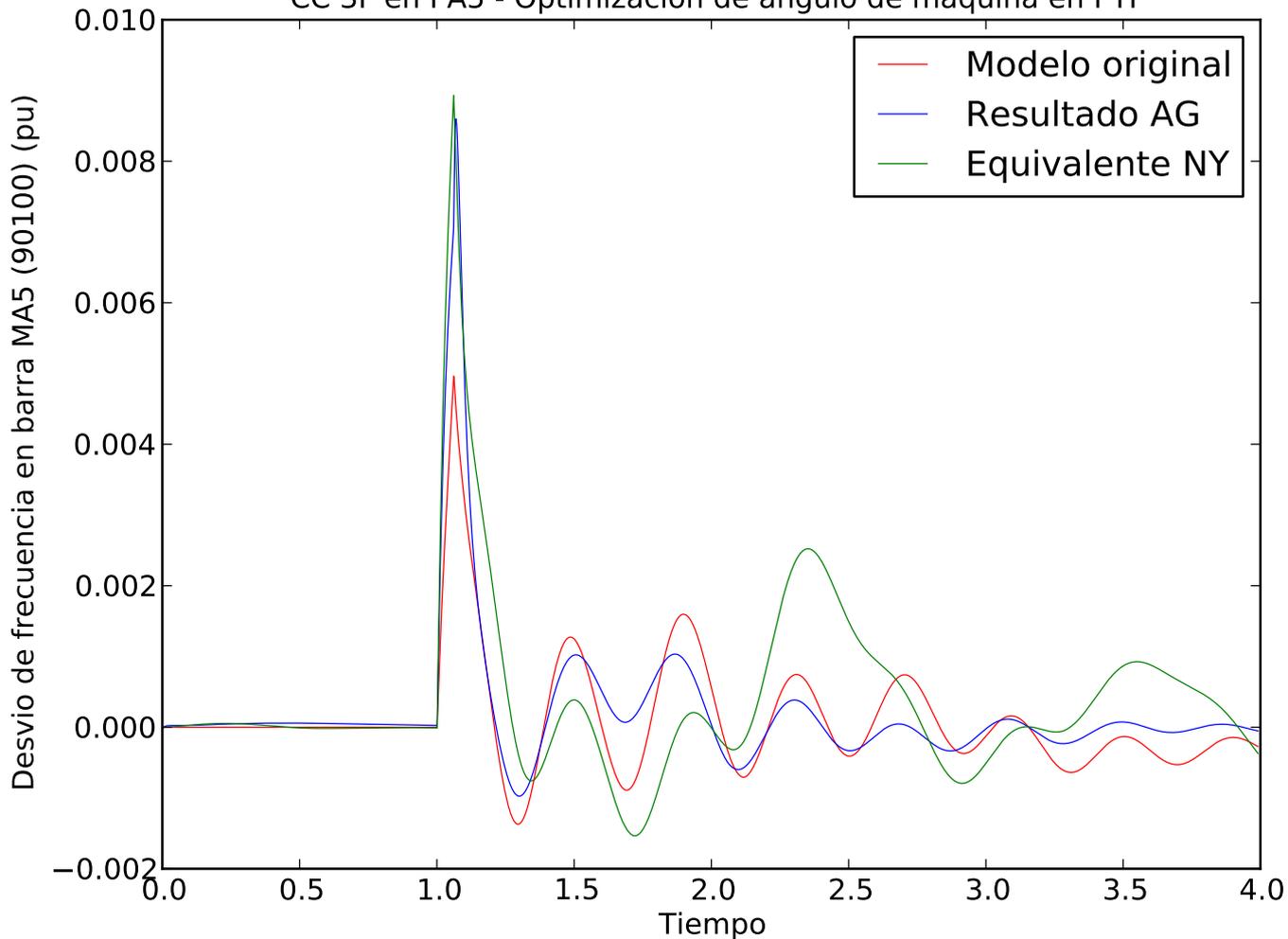
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



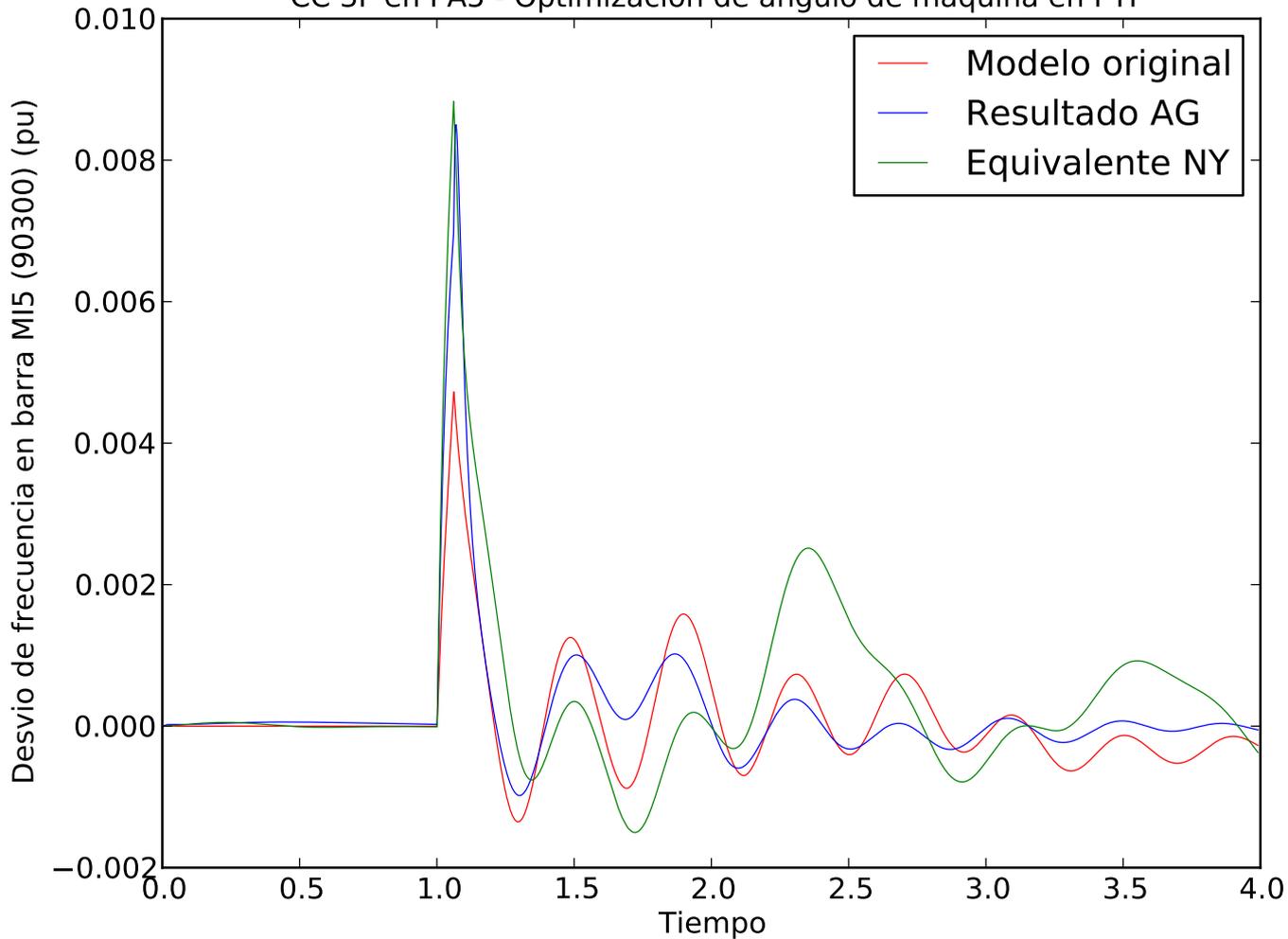
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



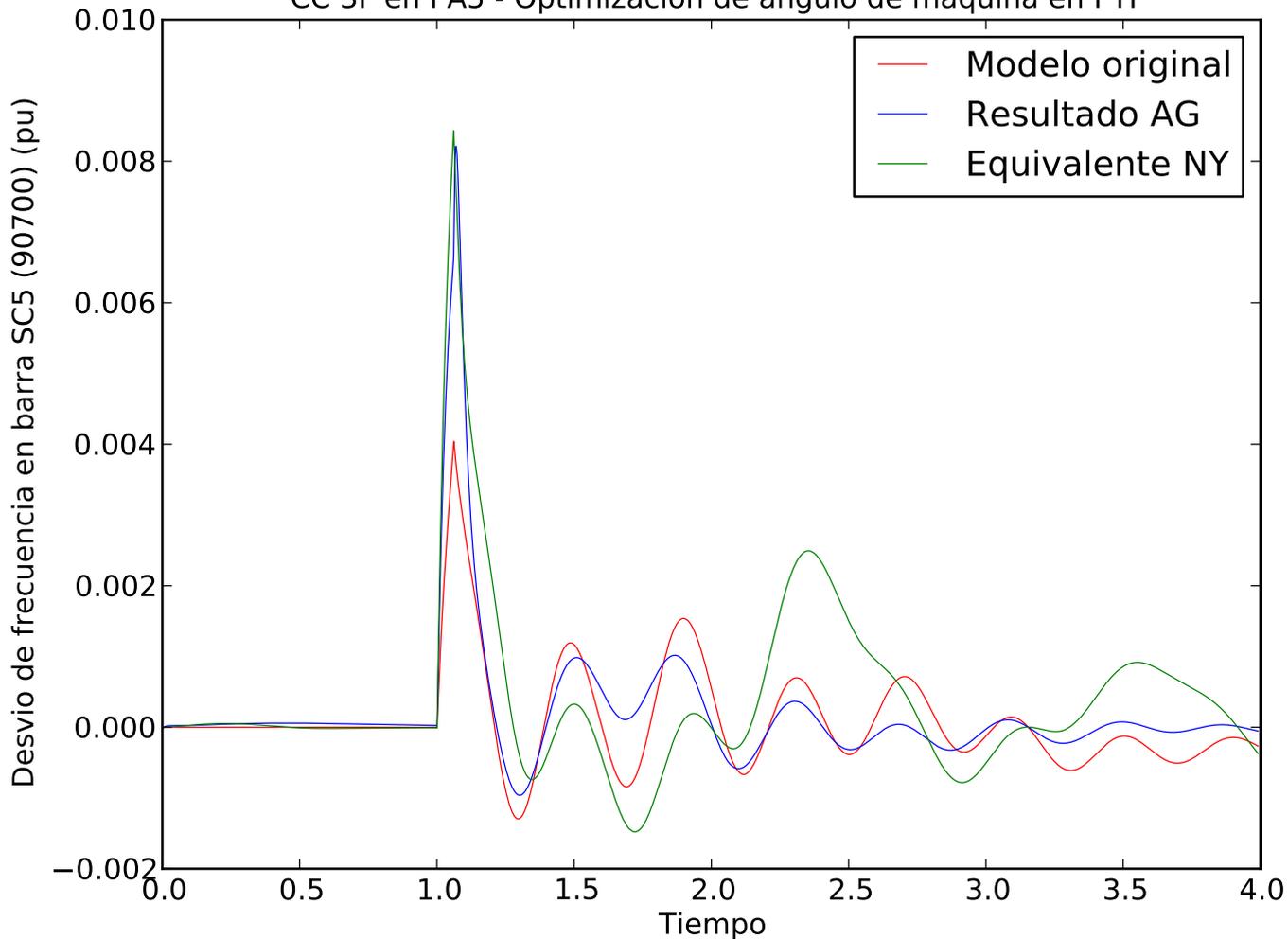
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



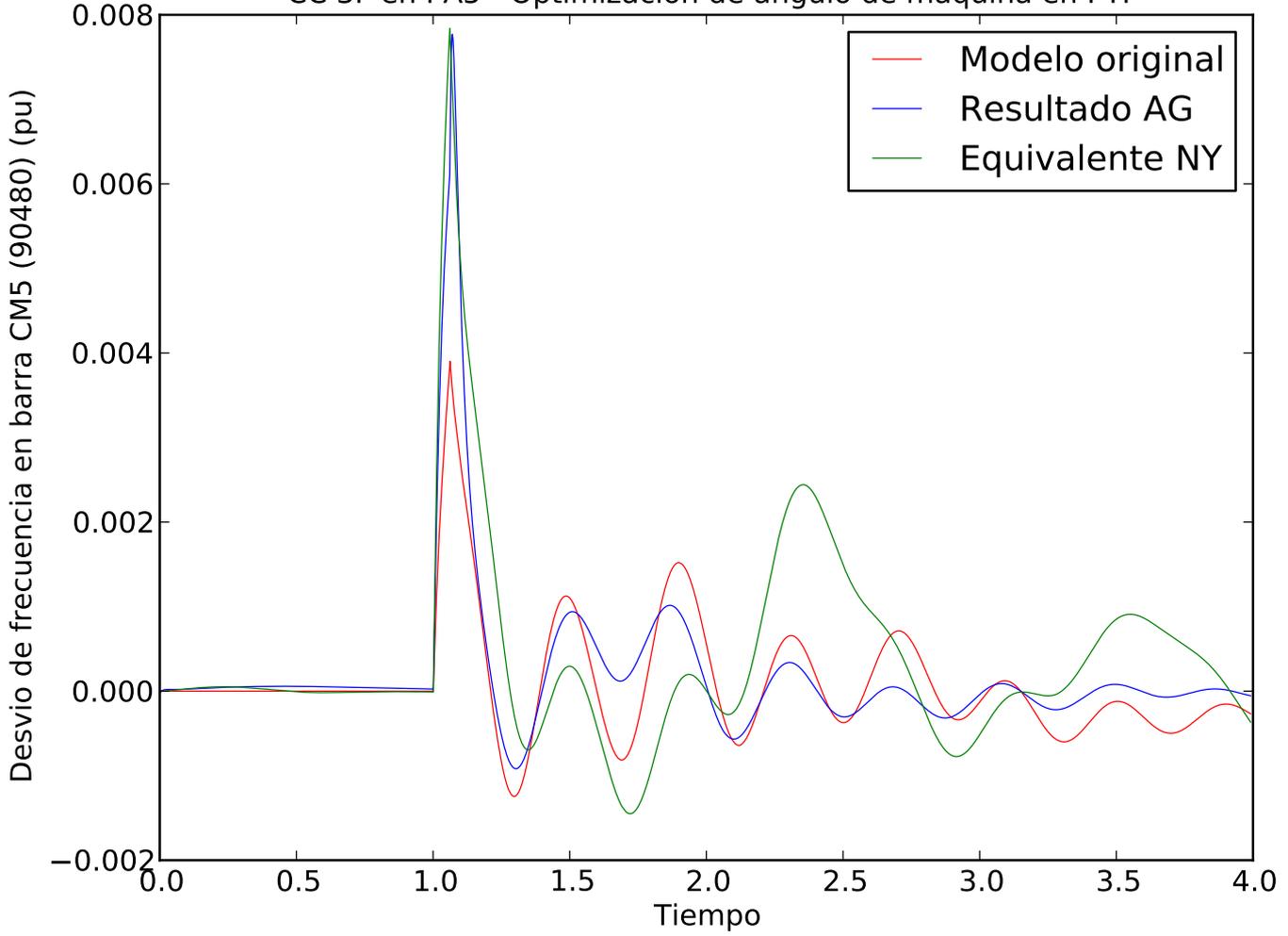
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



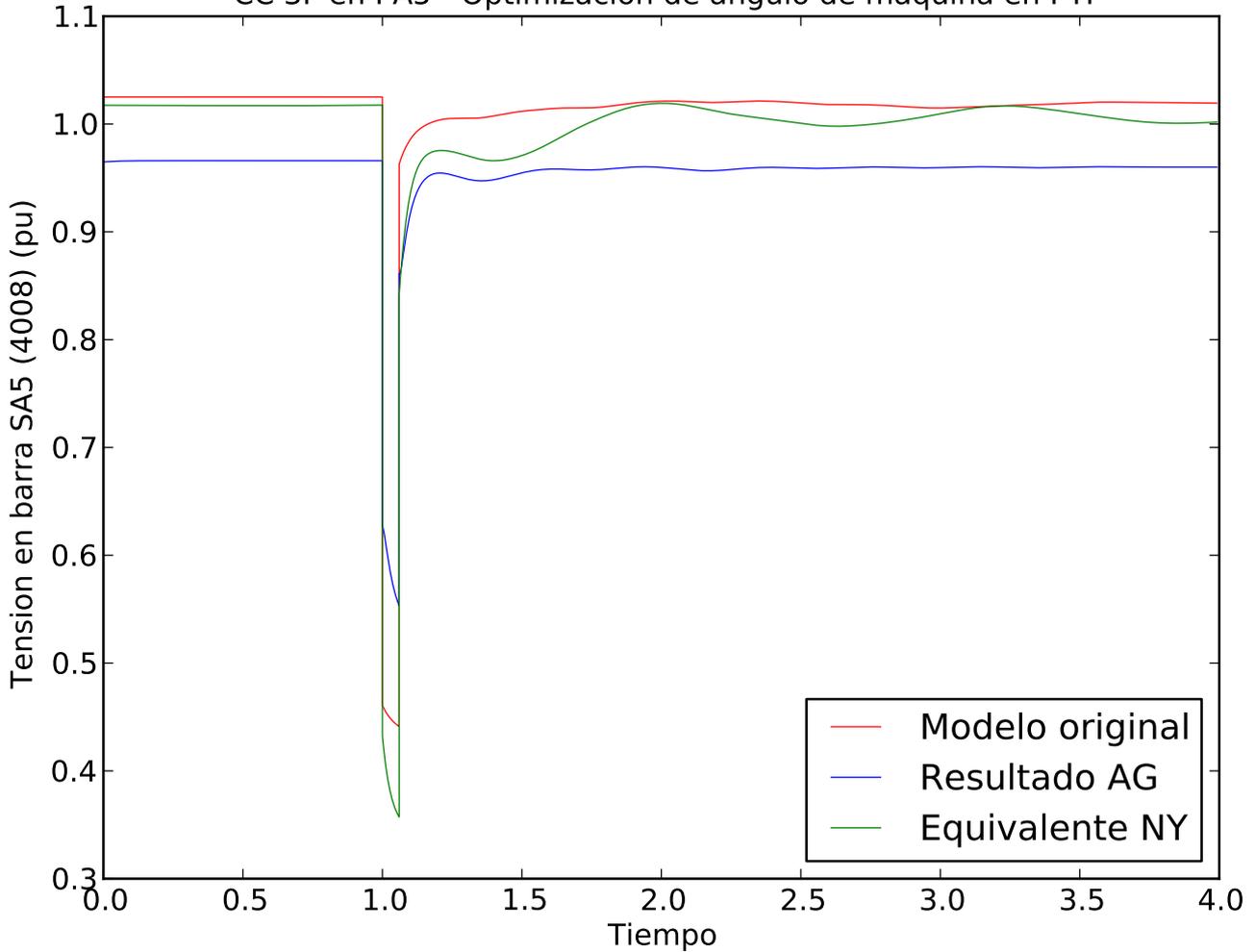
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



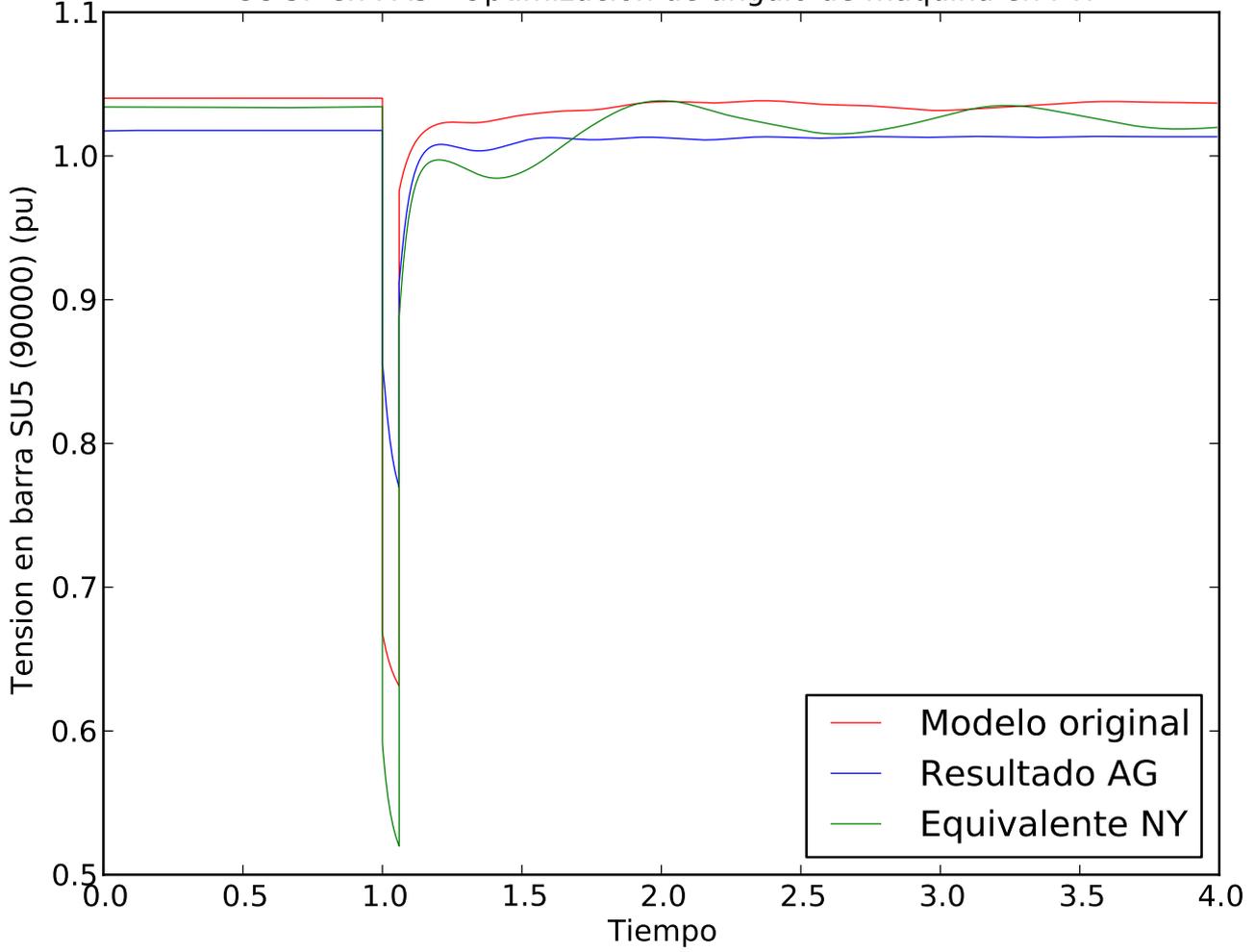
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



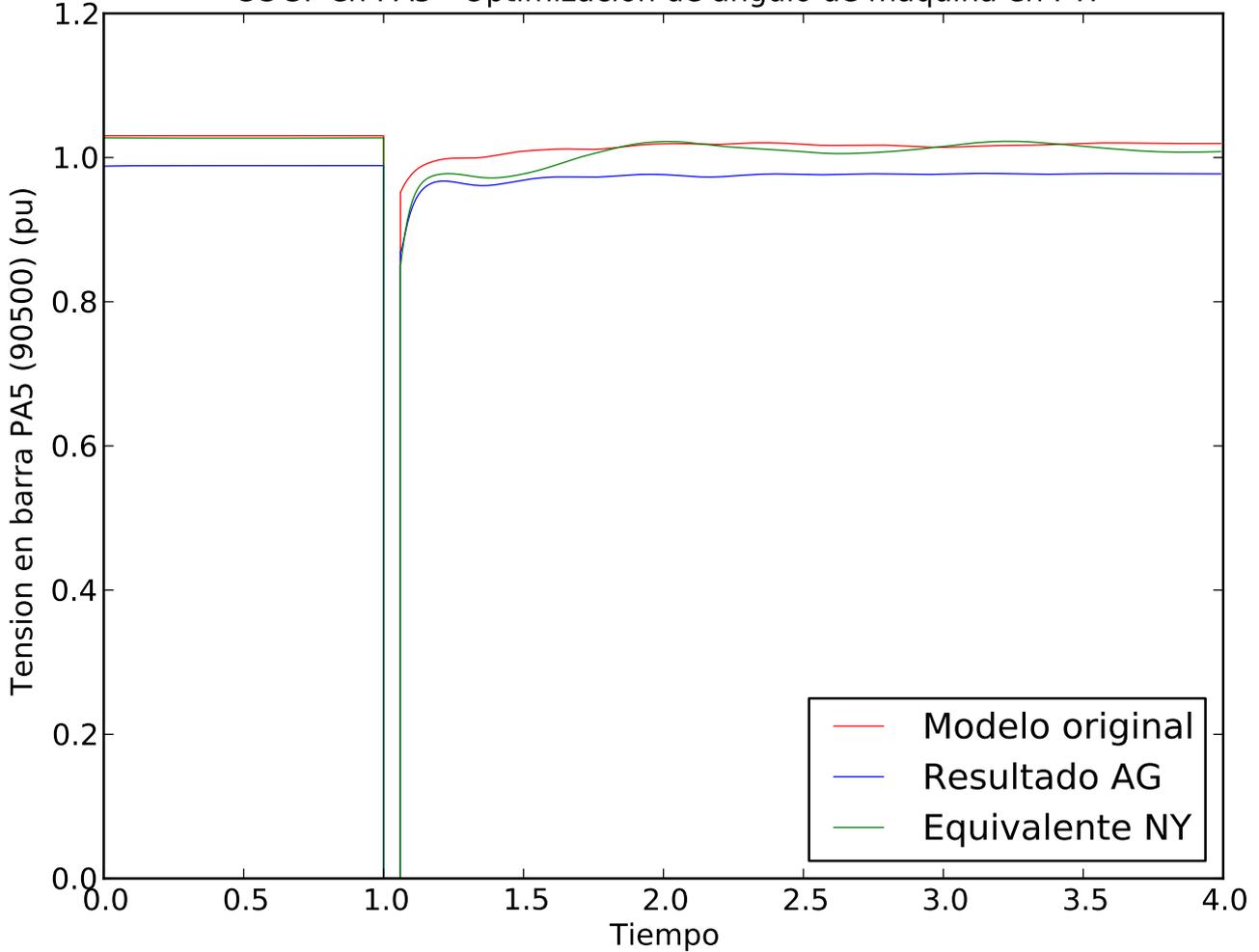
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



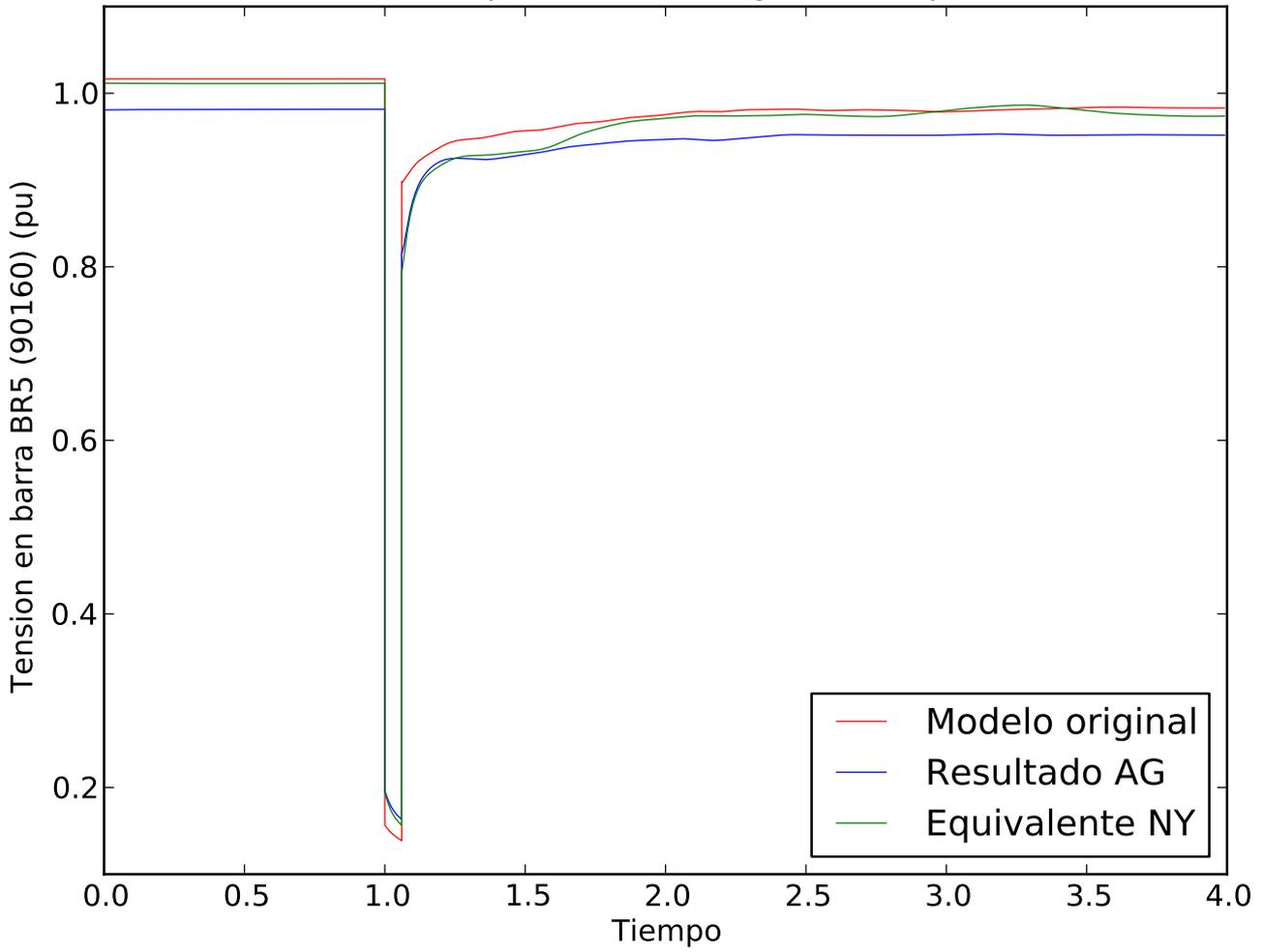
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



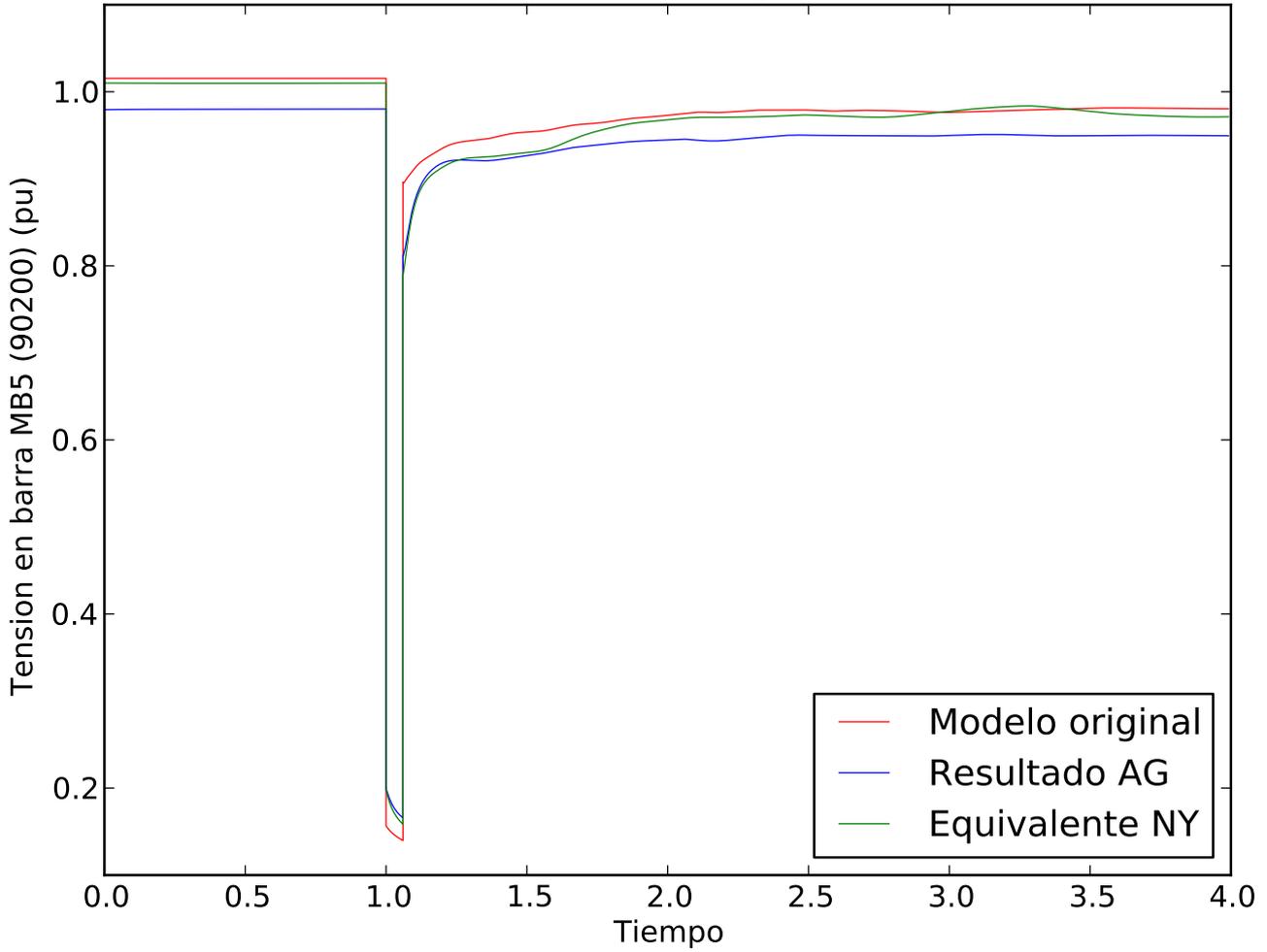
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



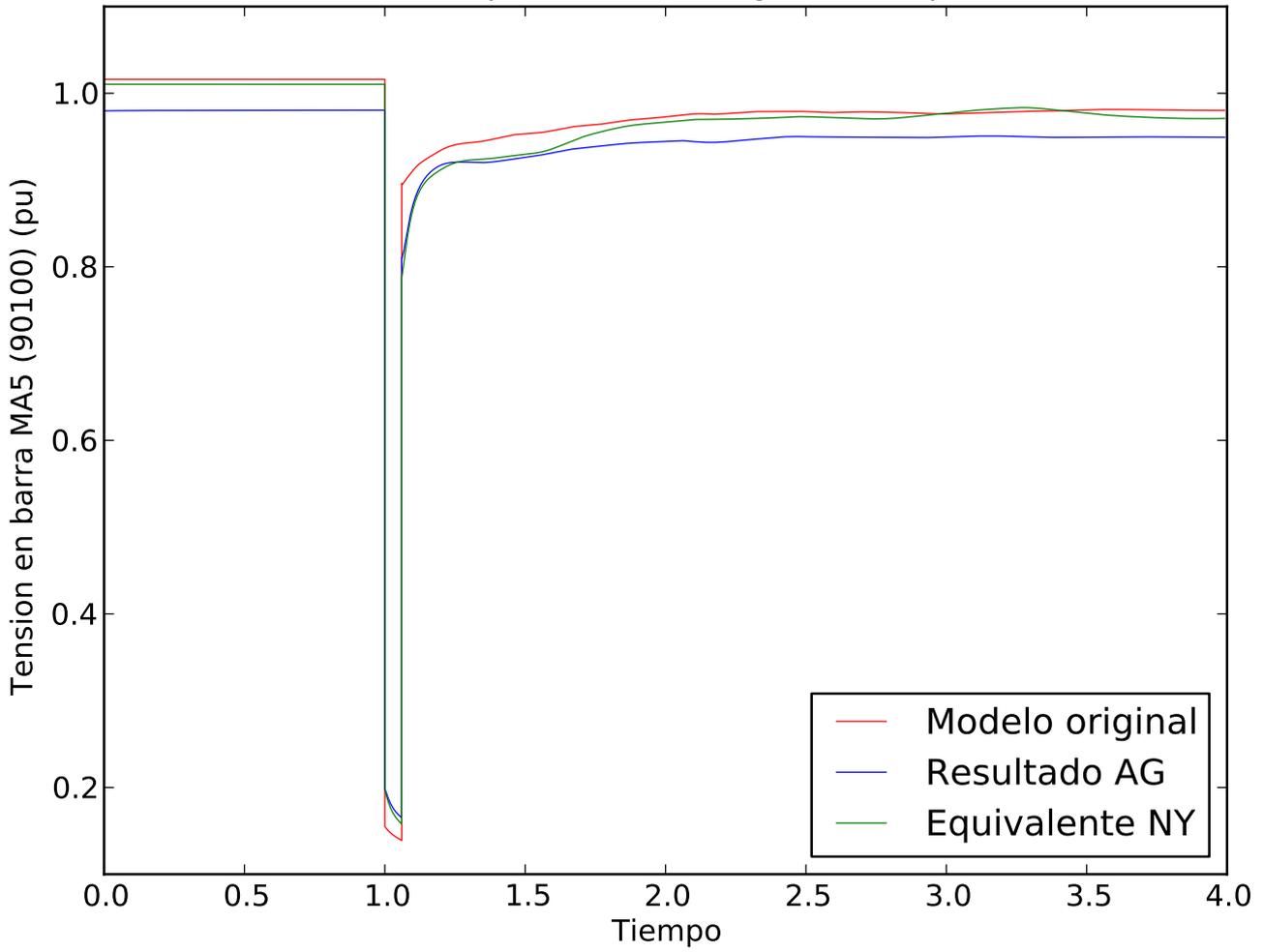
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



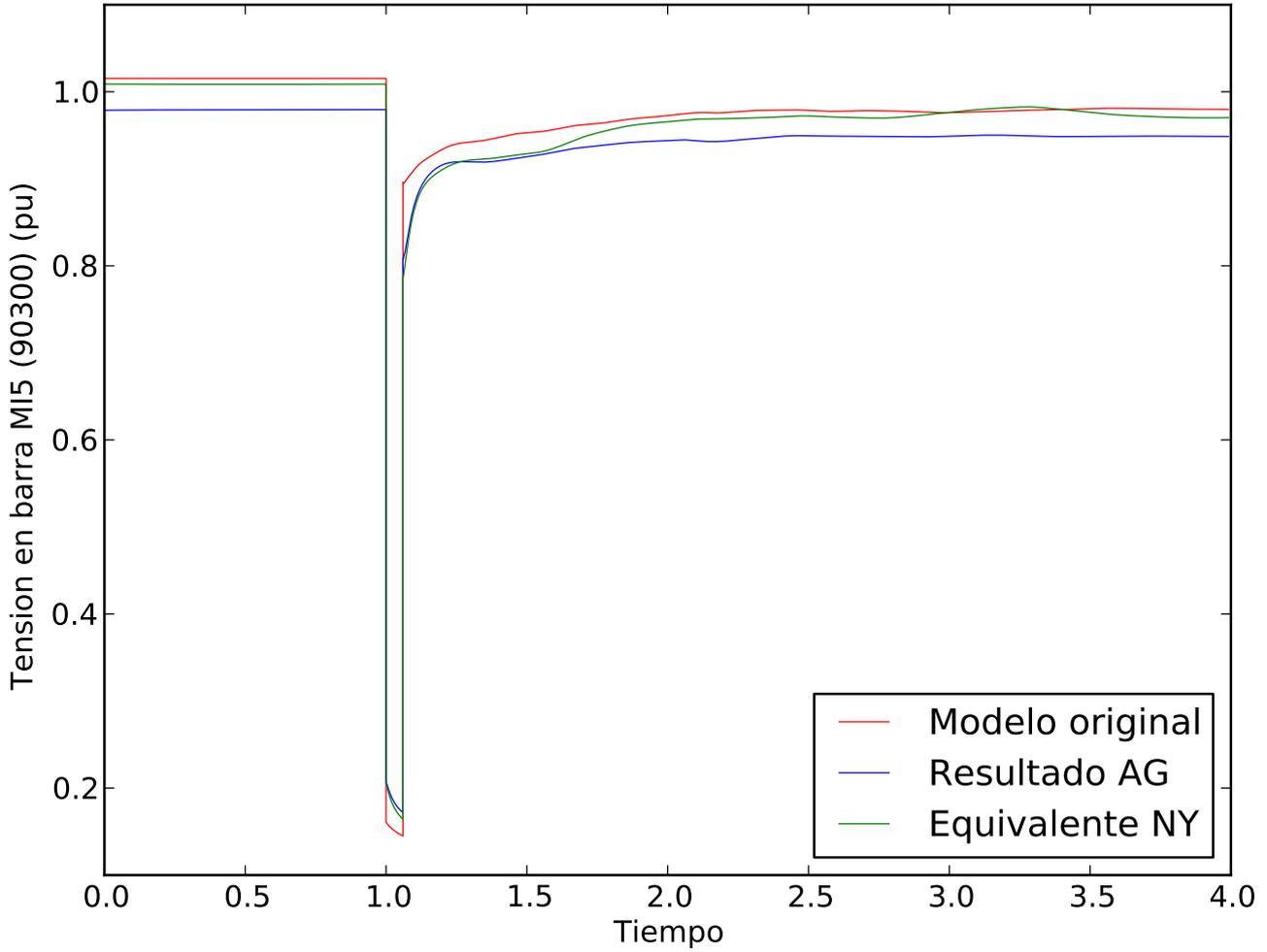
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



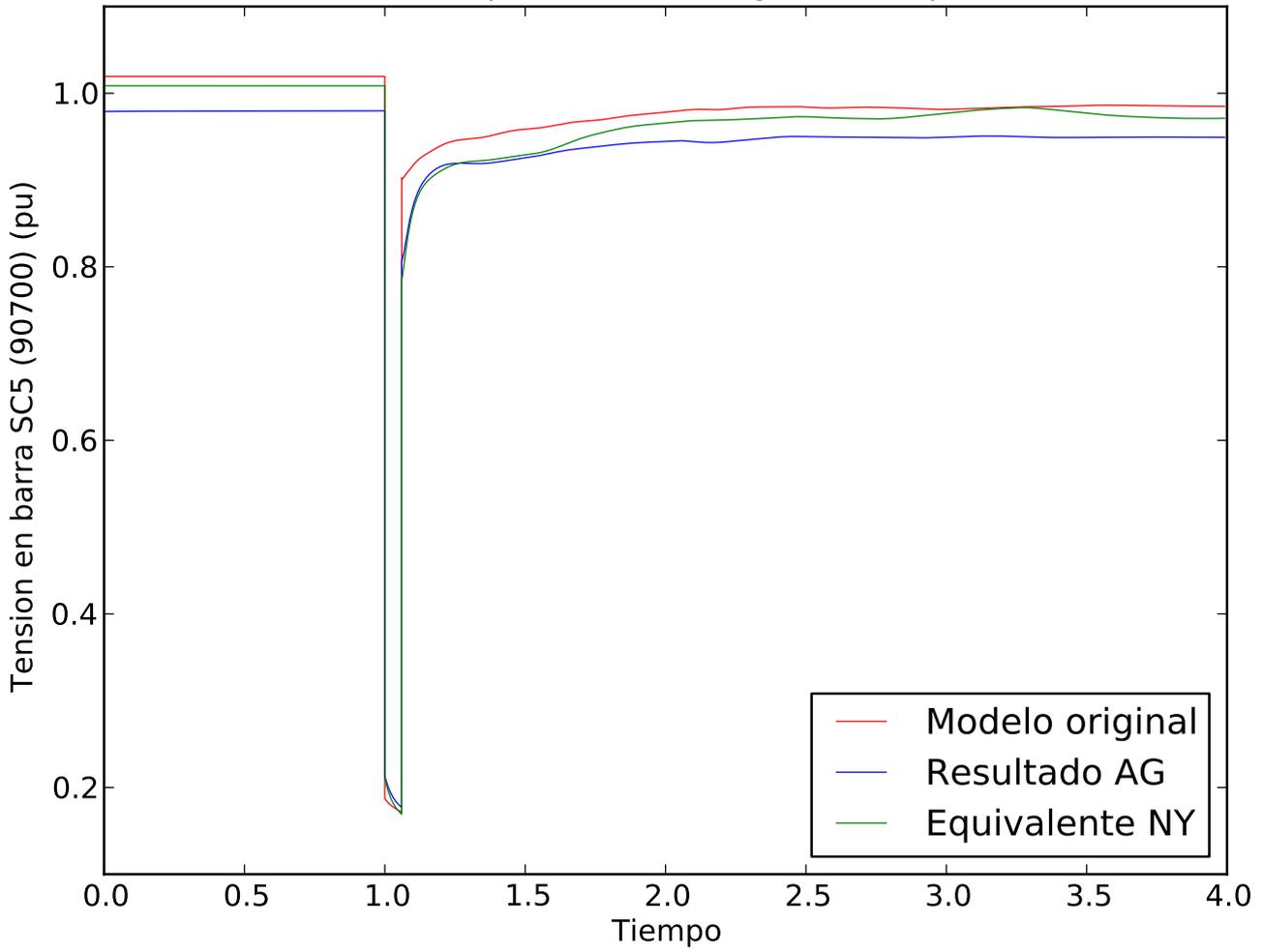
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



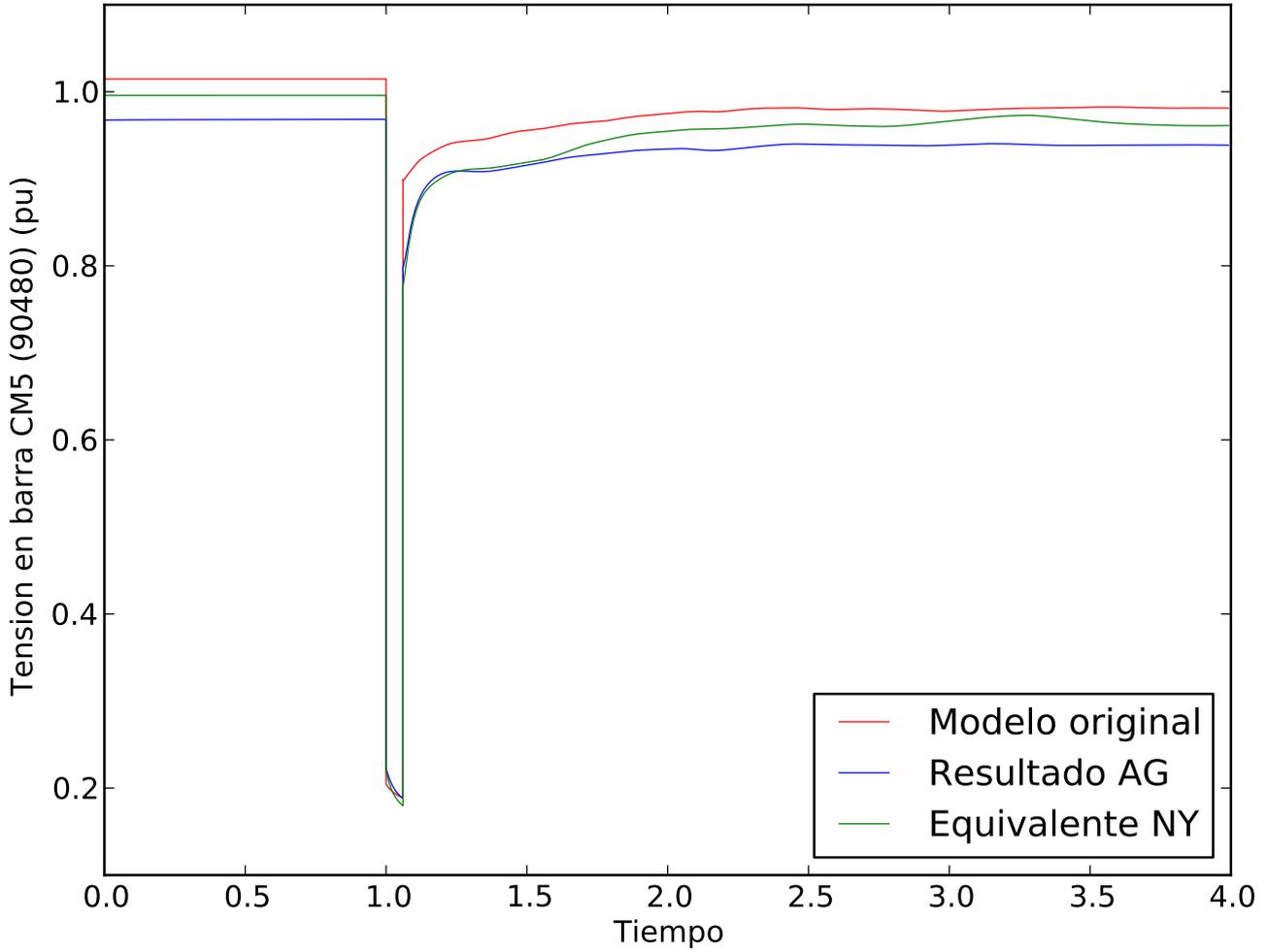
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



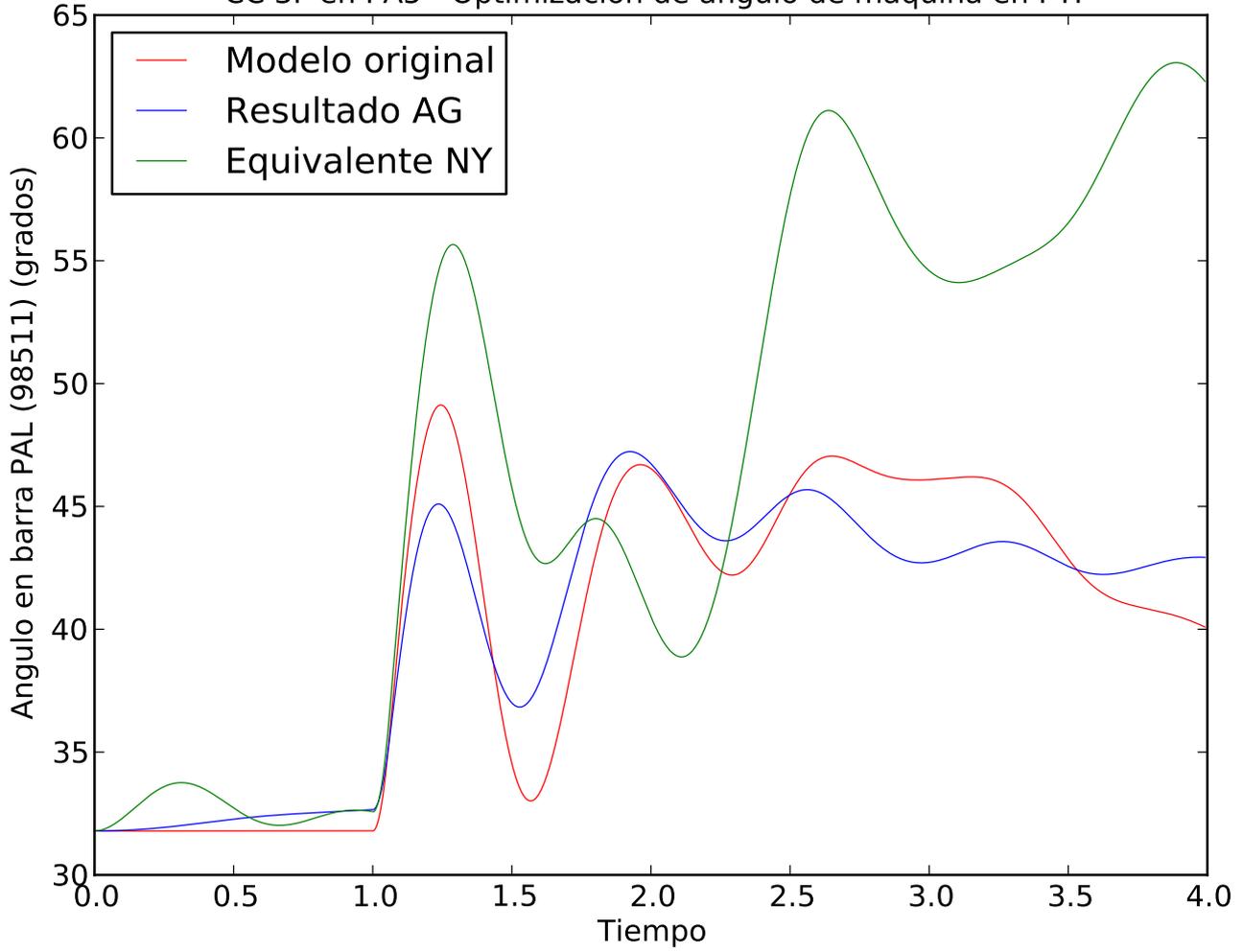
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



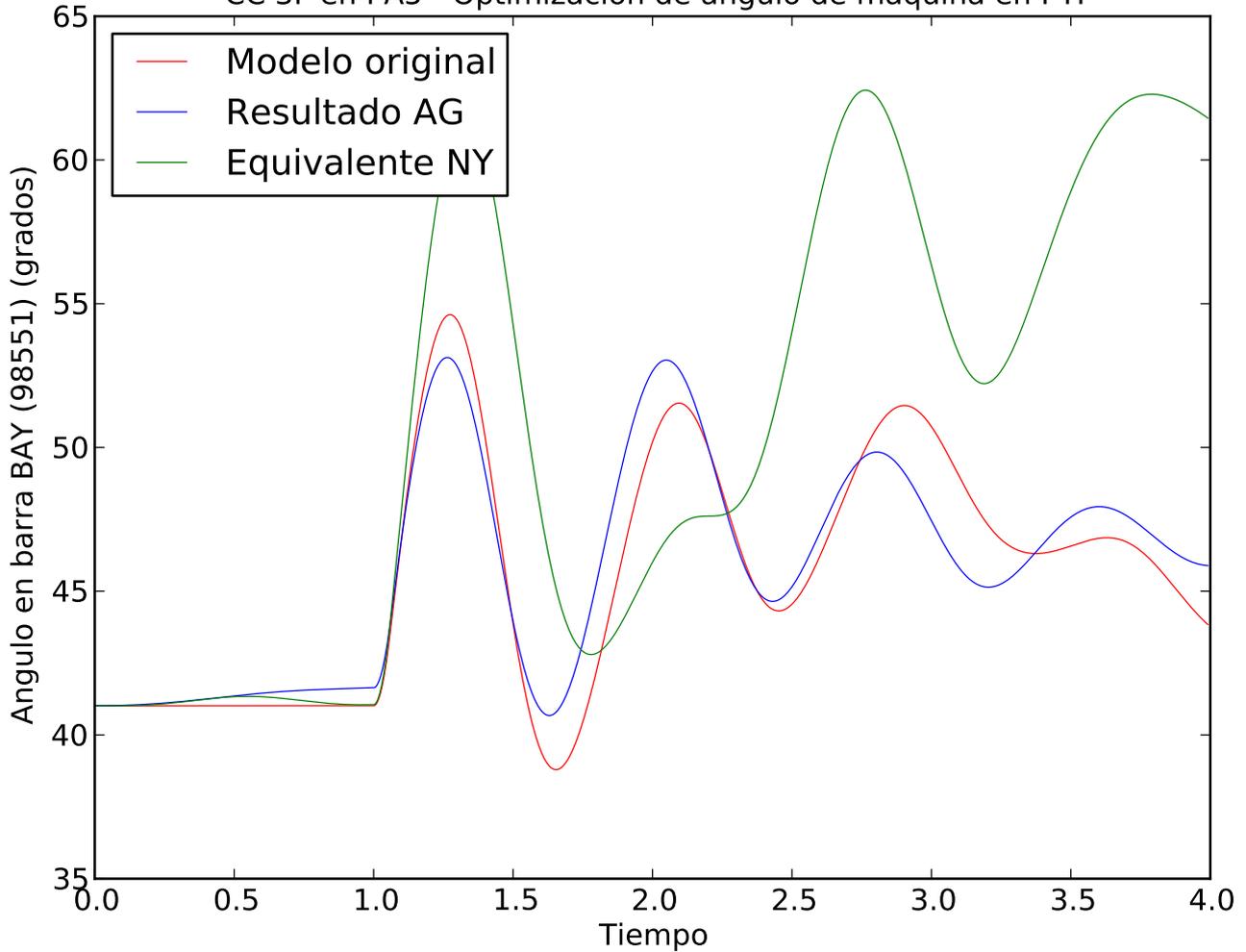
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



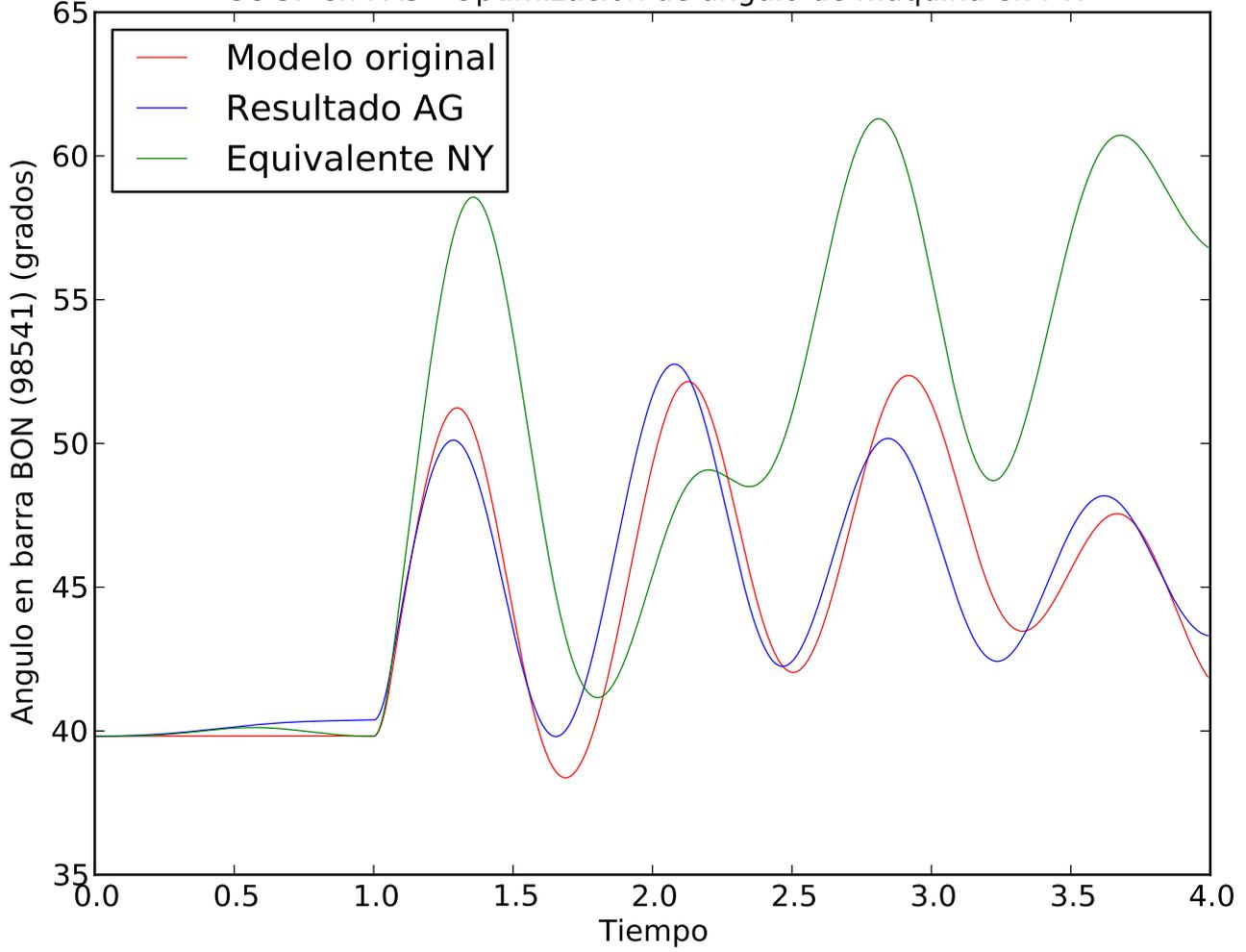
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



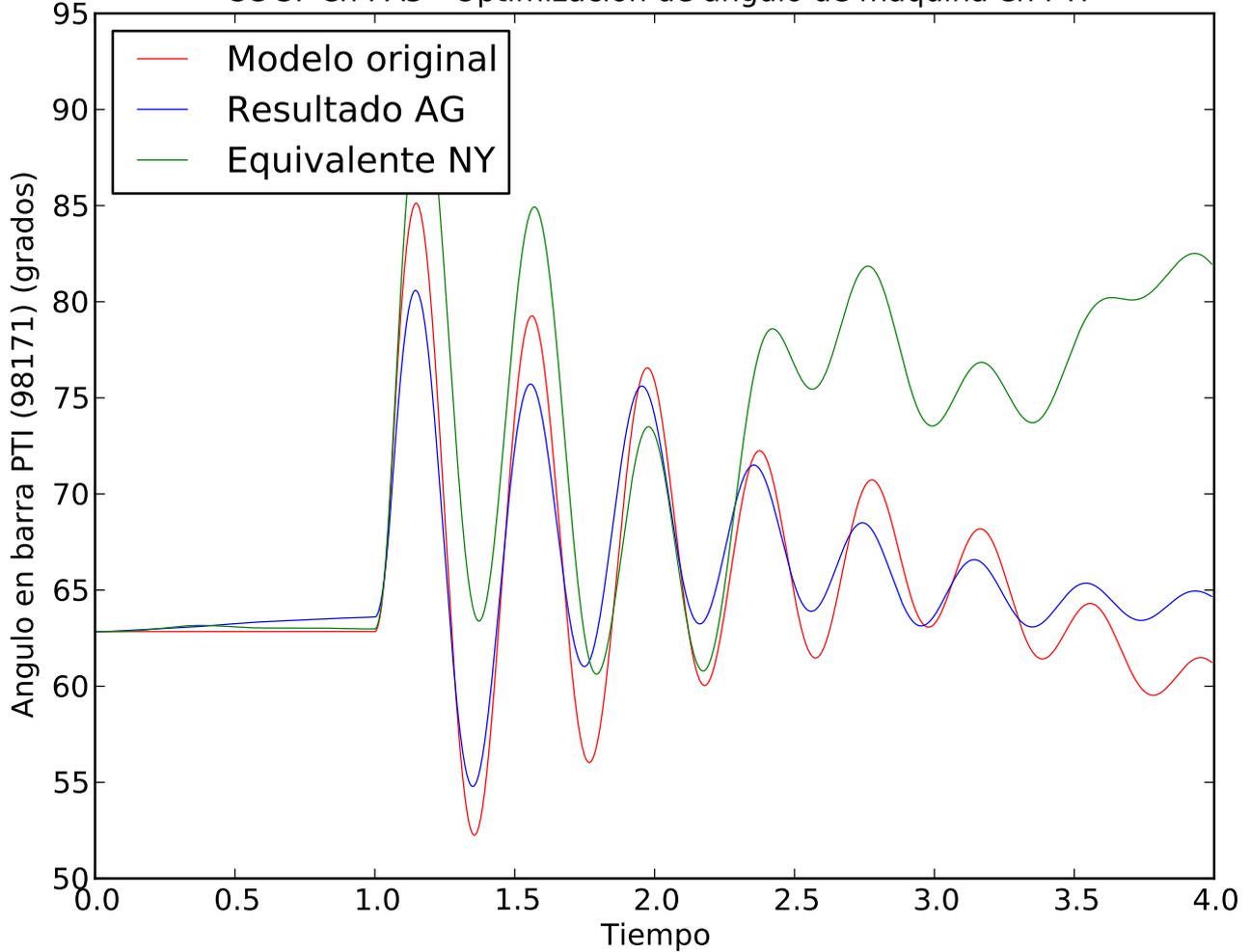
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



CC 3F en PA5 - Optimizacion de angulo de maquina en PTI

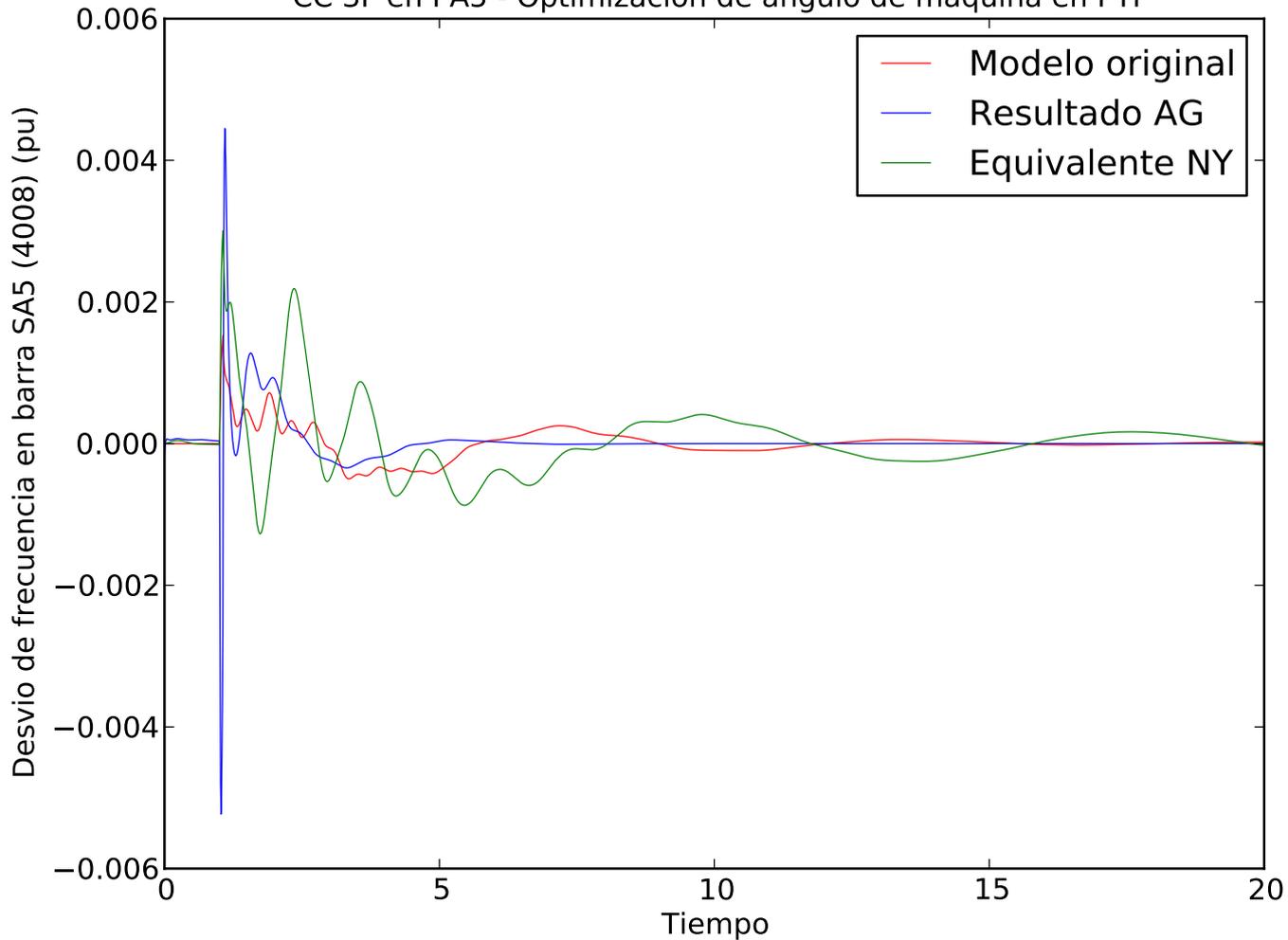


CC 3F en PA5 - Optimizacion de angulo de maquina en PTI

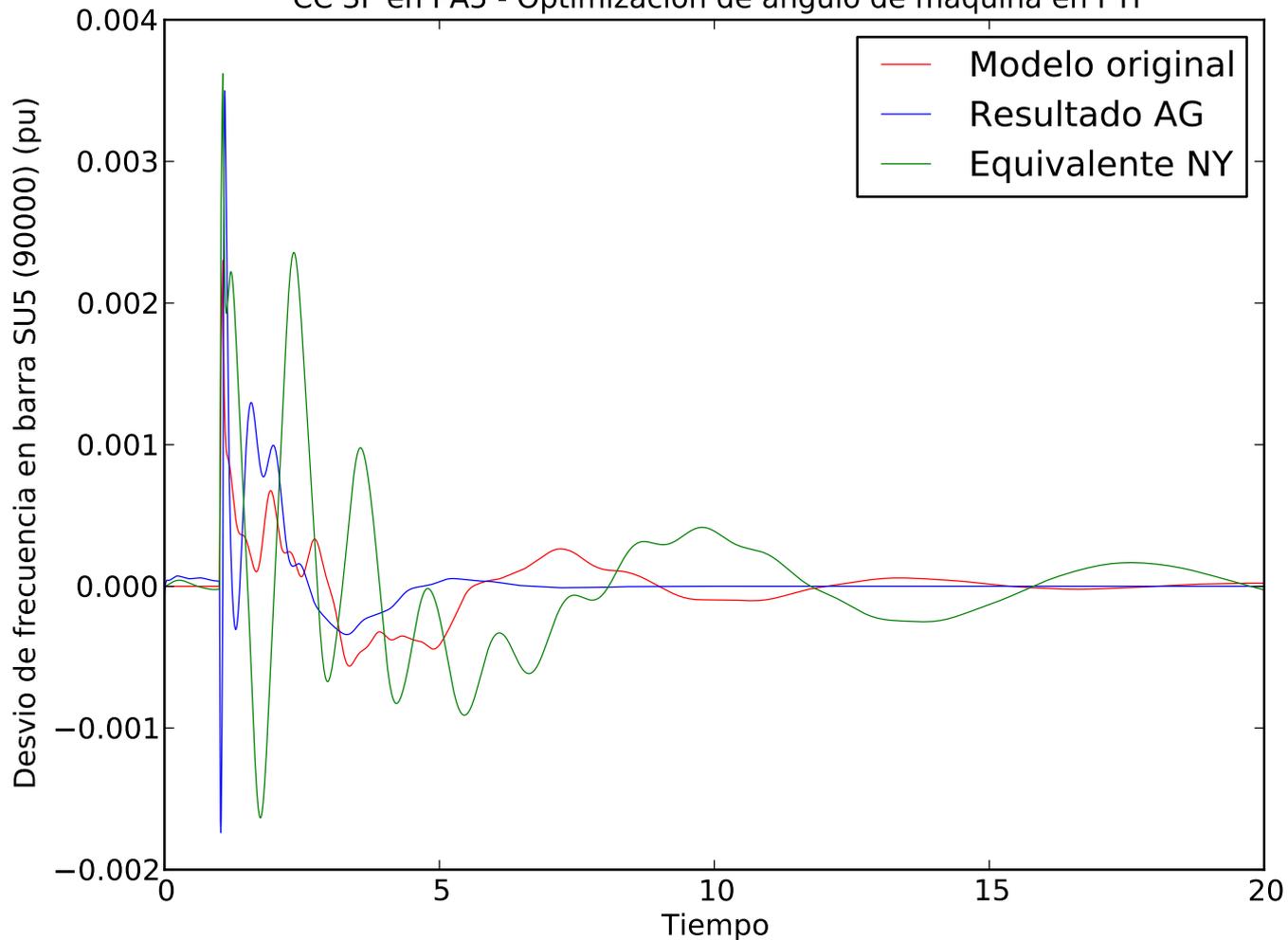


- C.7. **Simulación 4.2: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando ángulo en una máquina de PTI en 20 segundos. Comportamiento en 20 segundos**

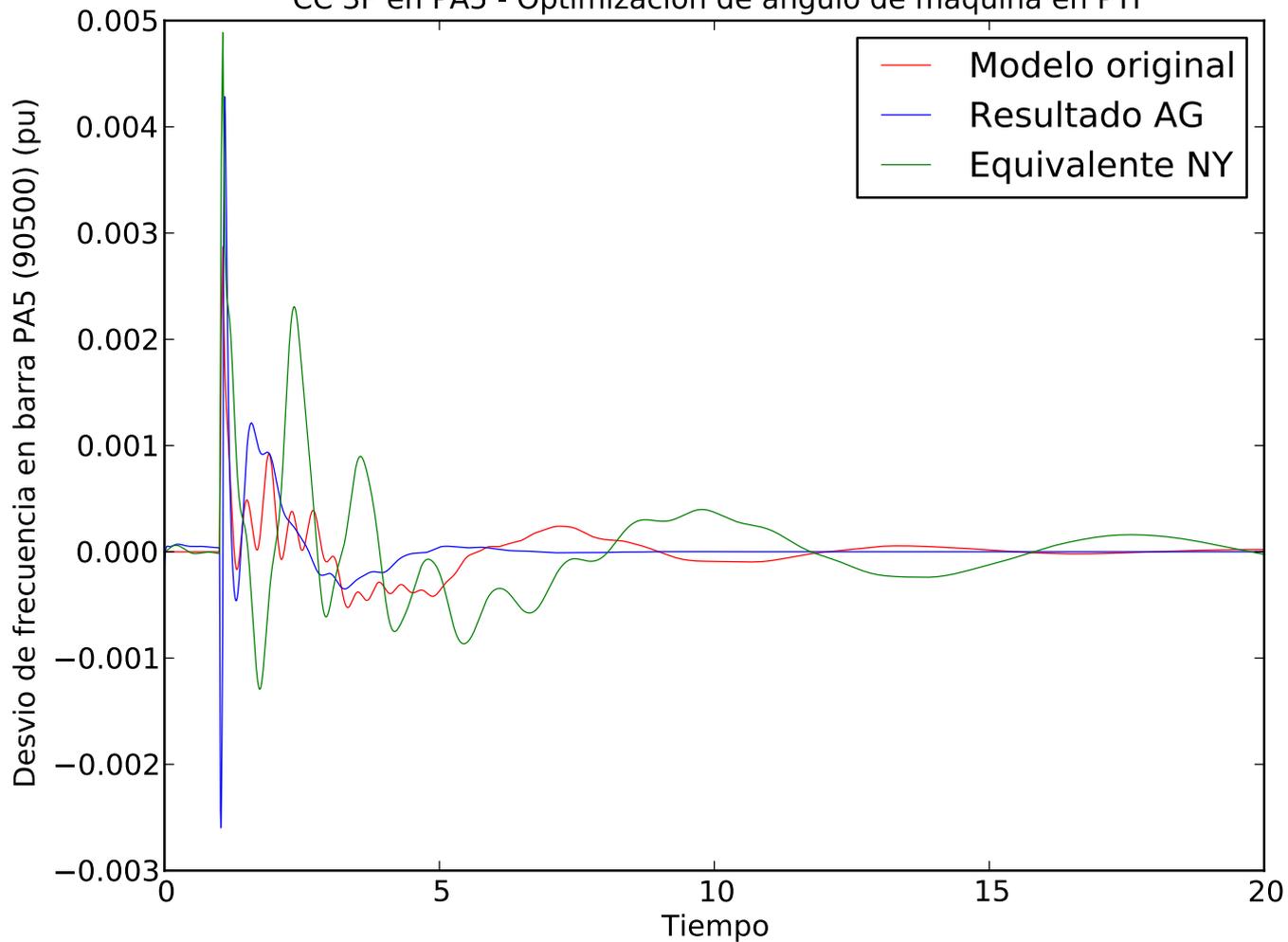
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



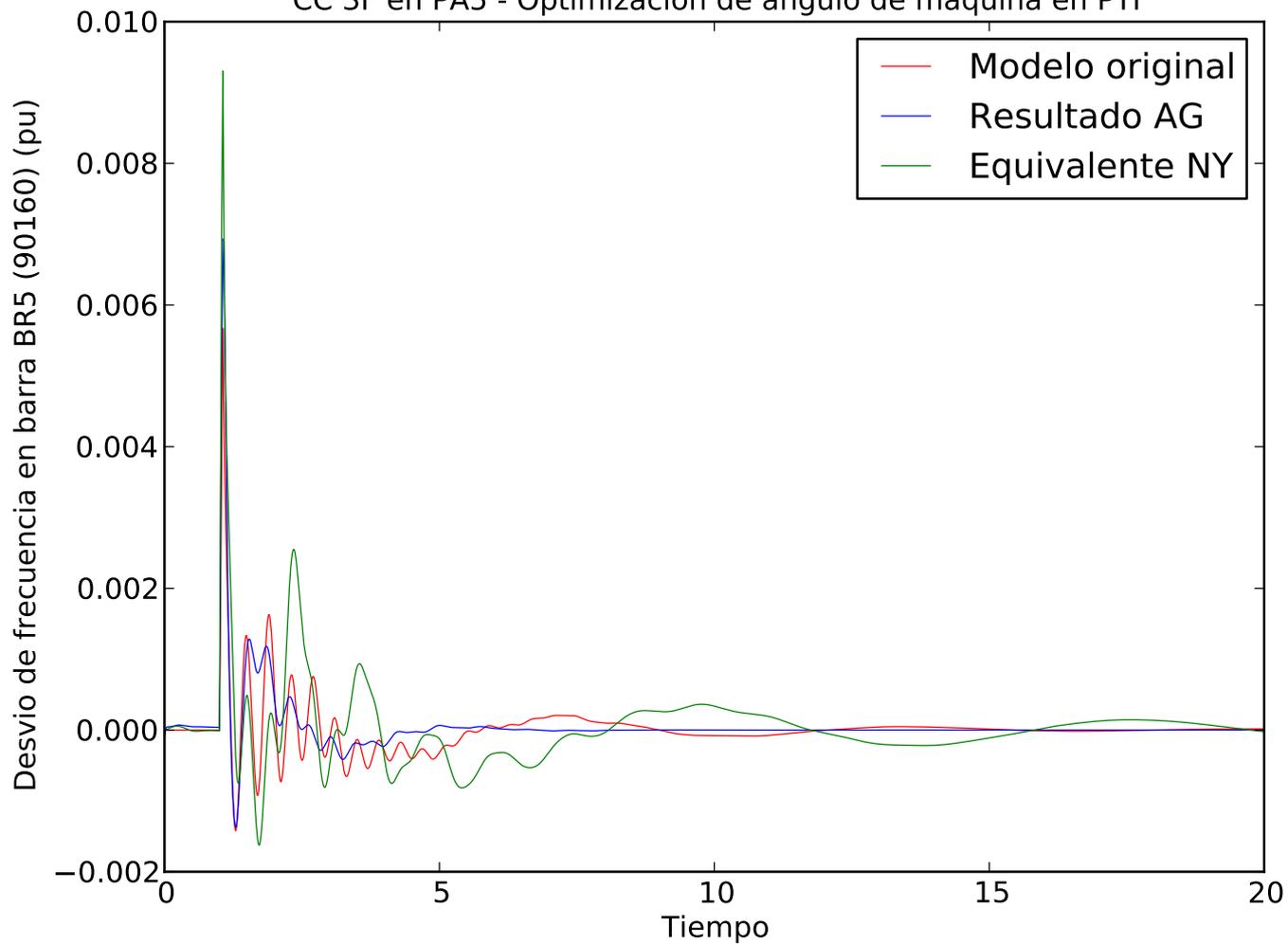
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



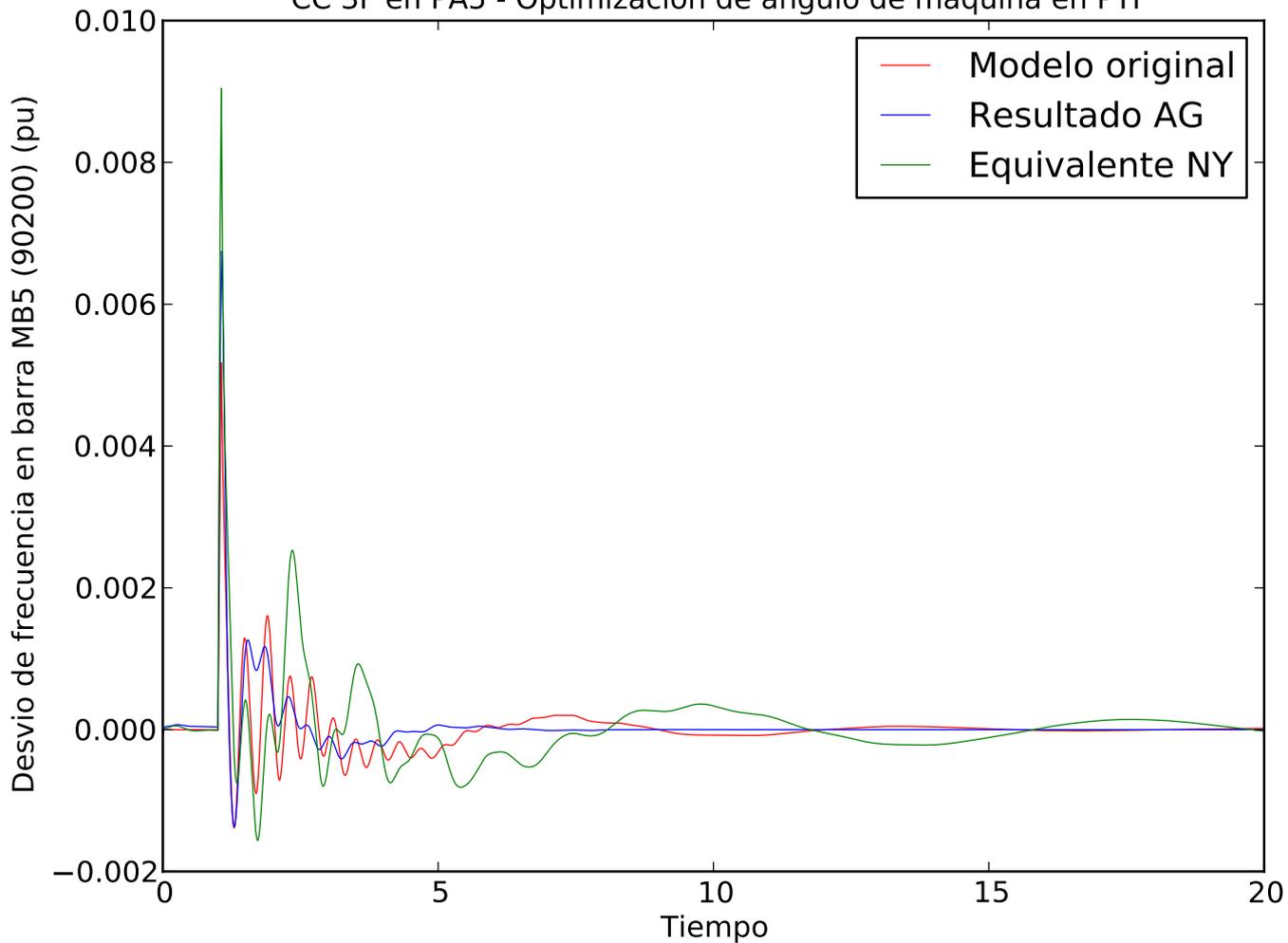
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



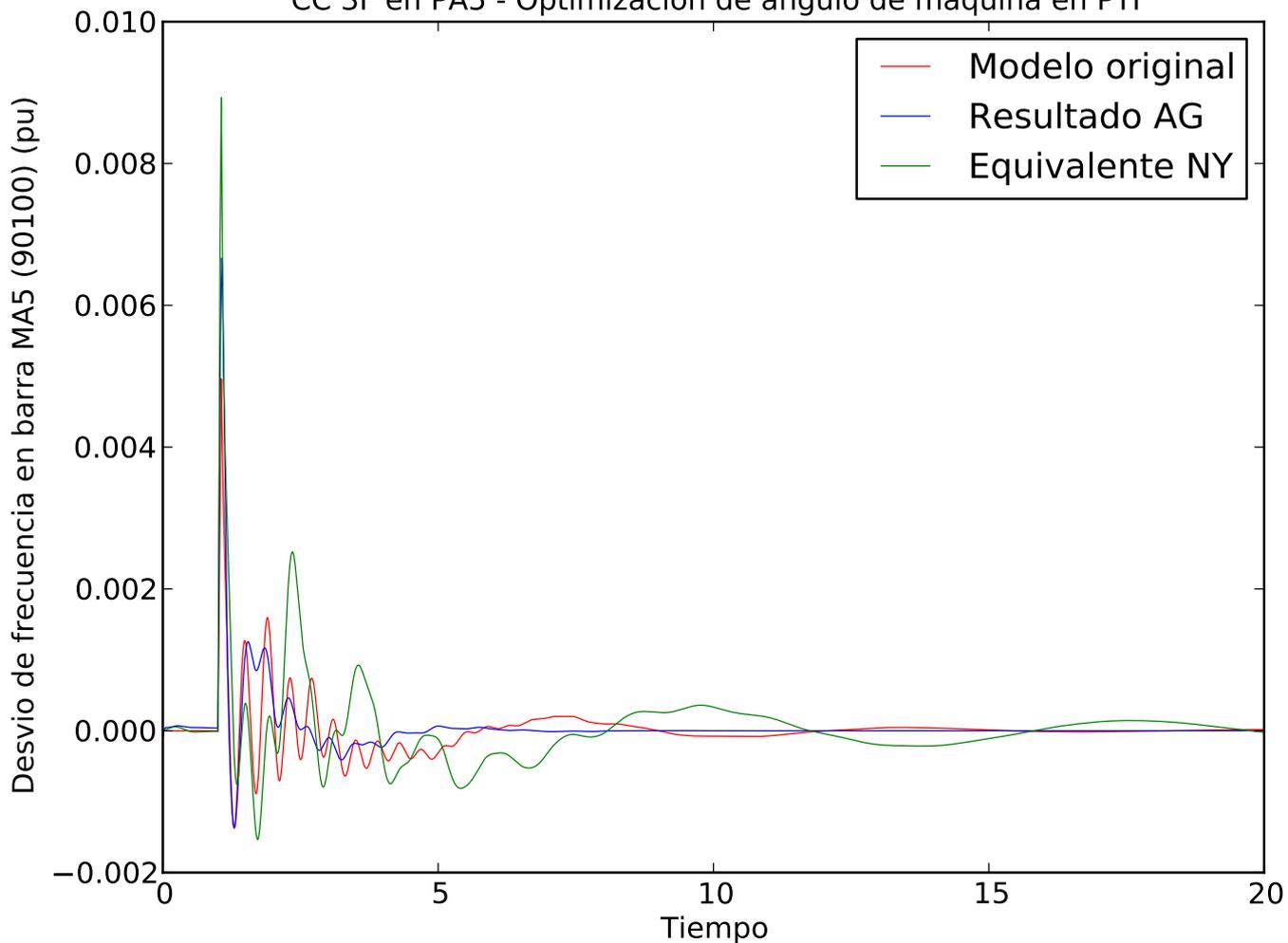
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



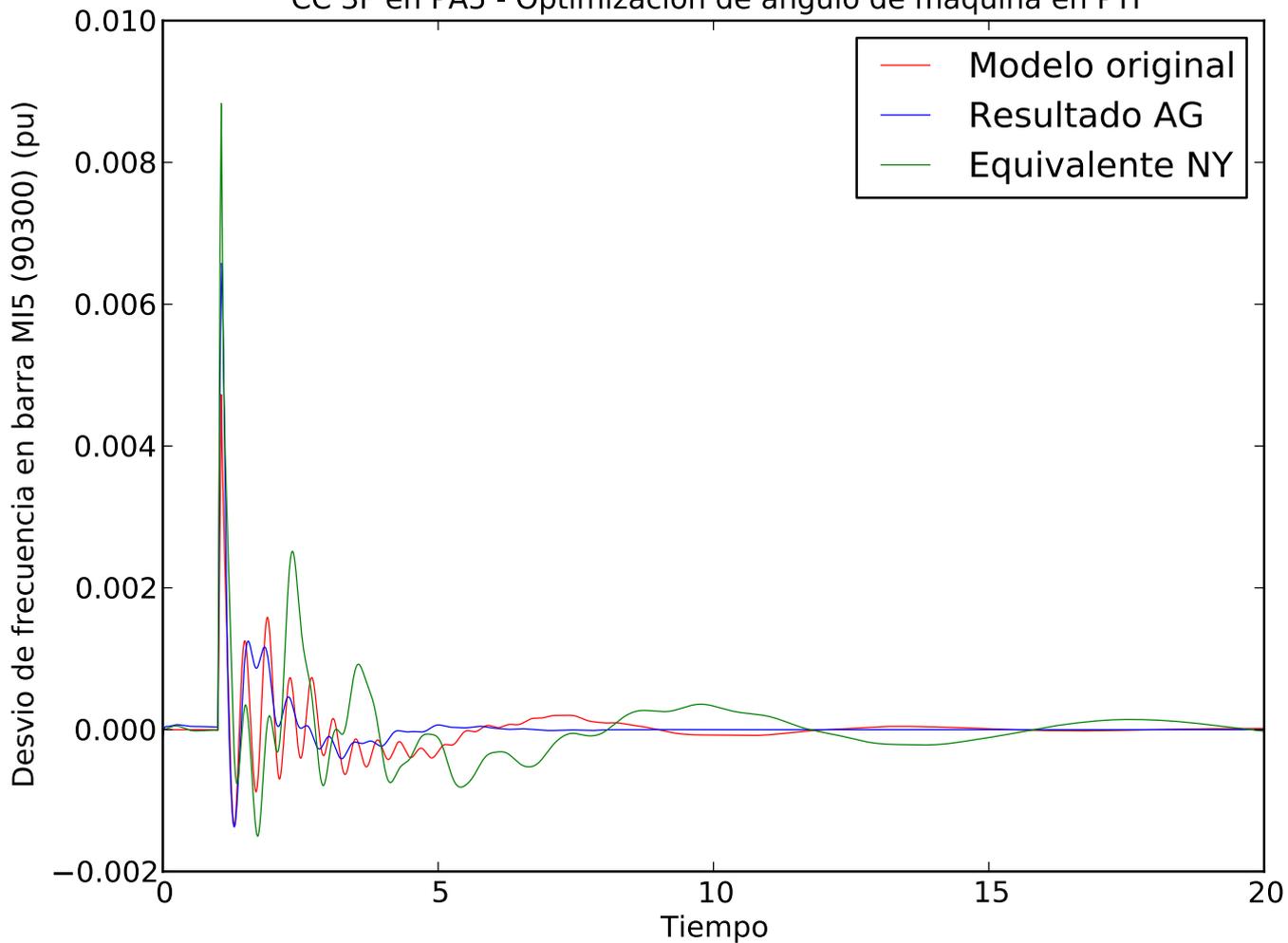
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



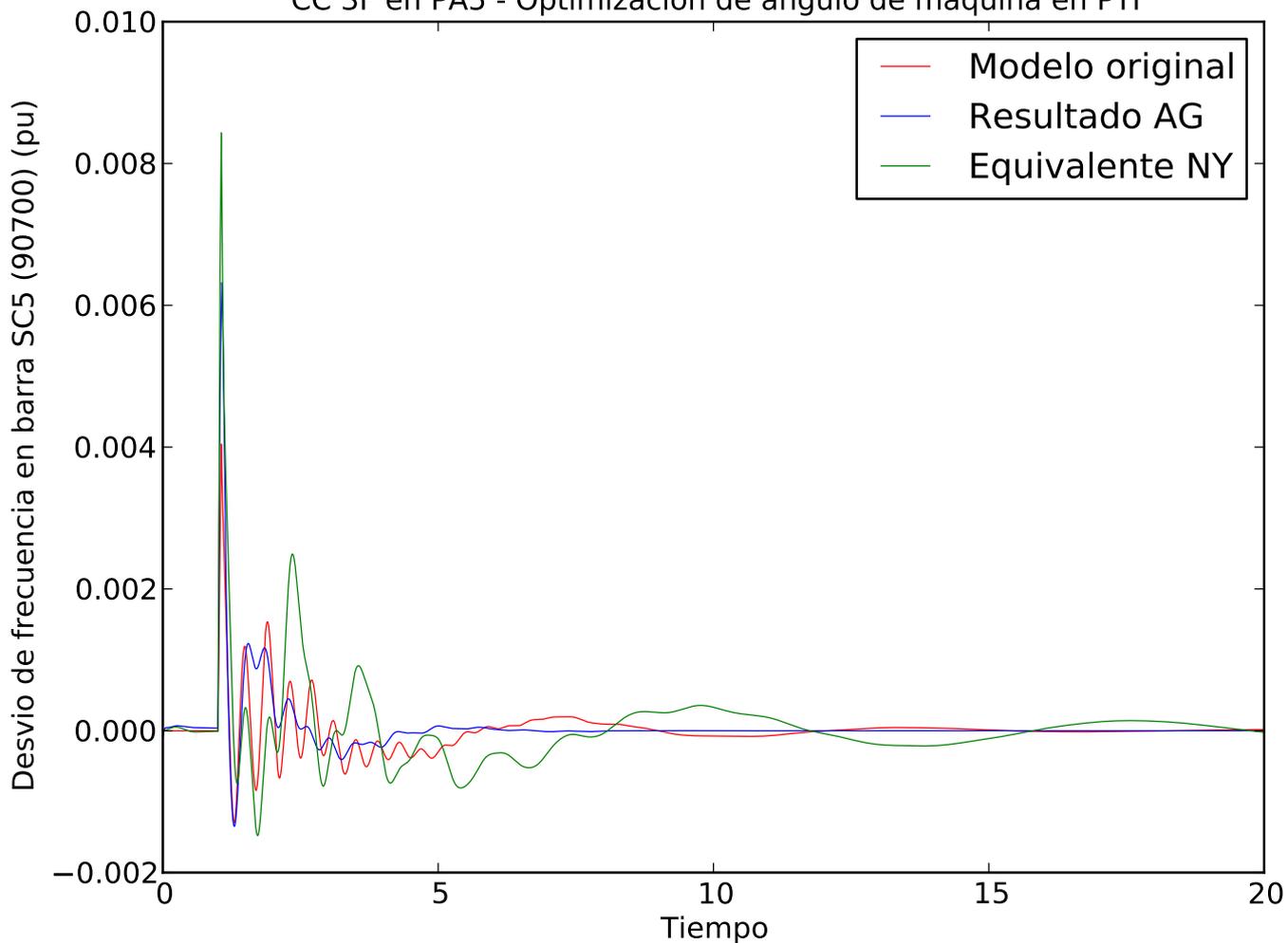
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



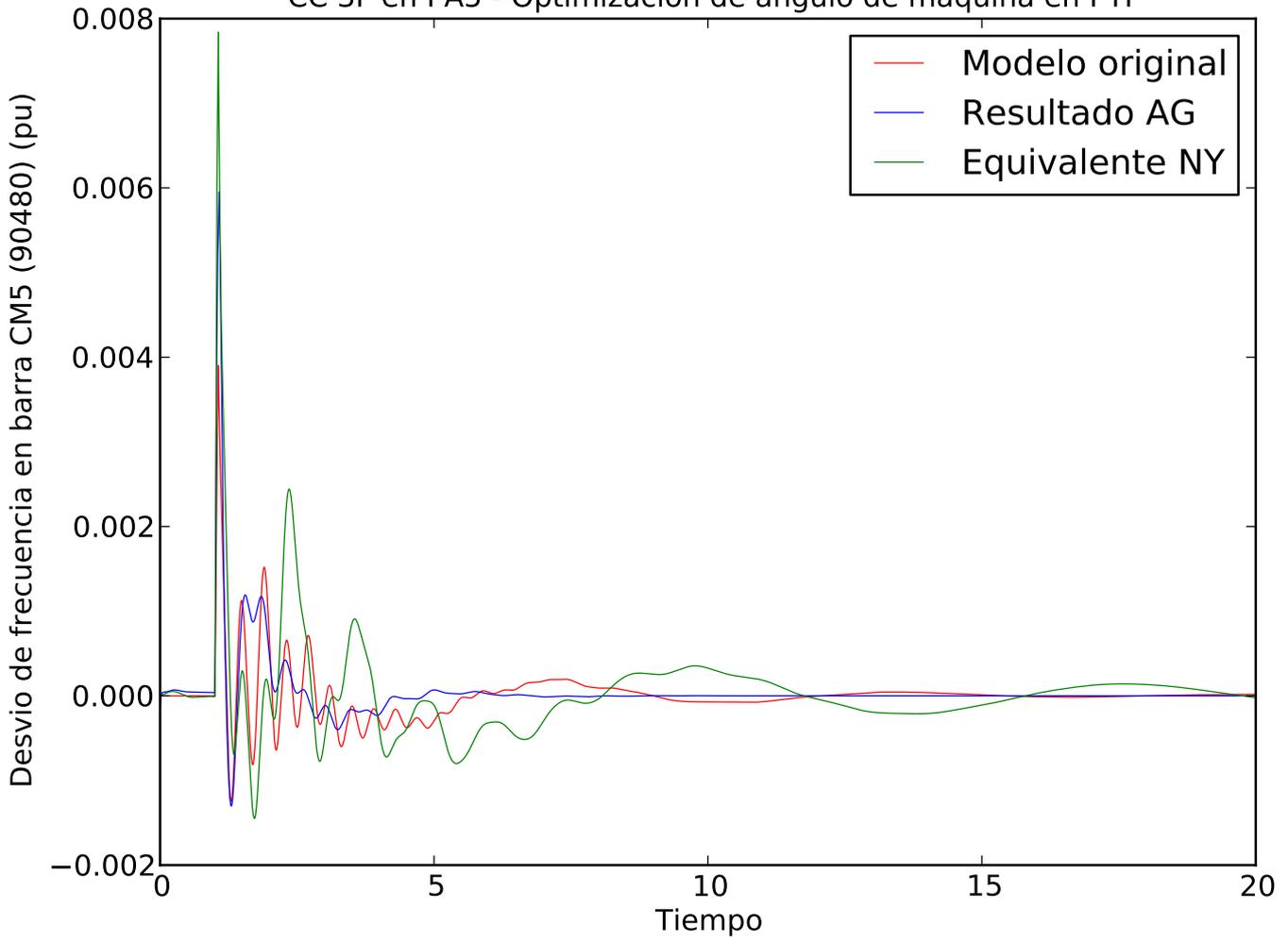
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



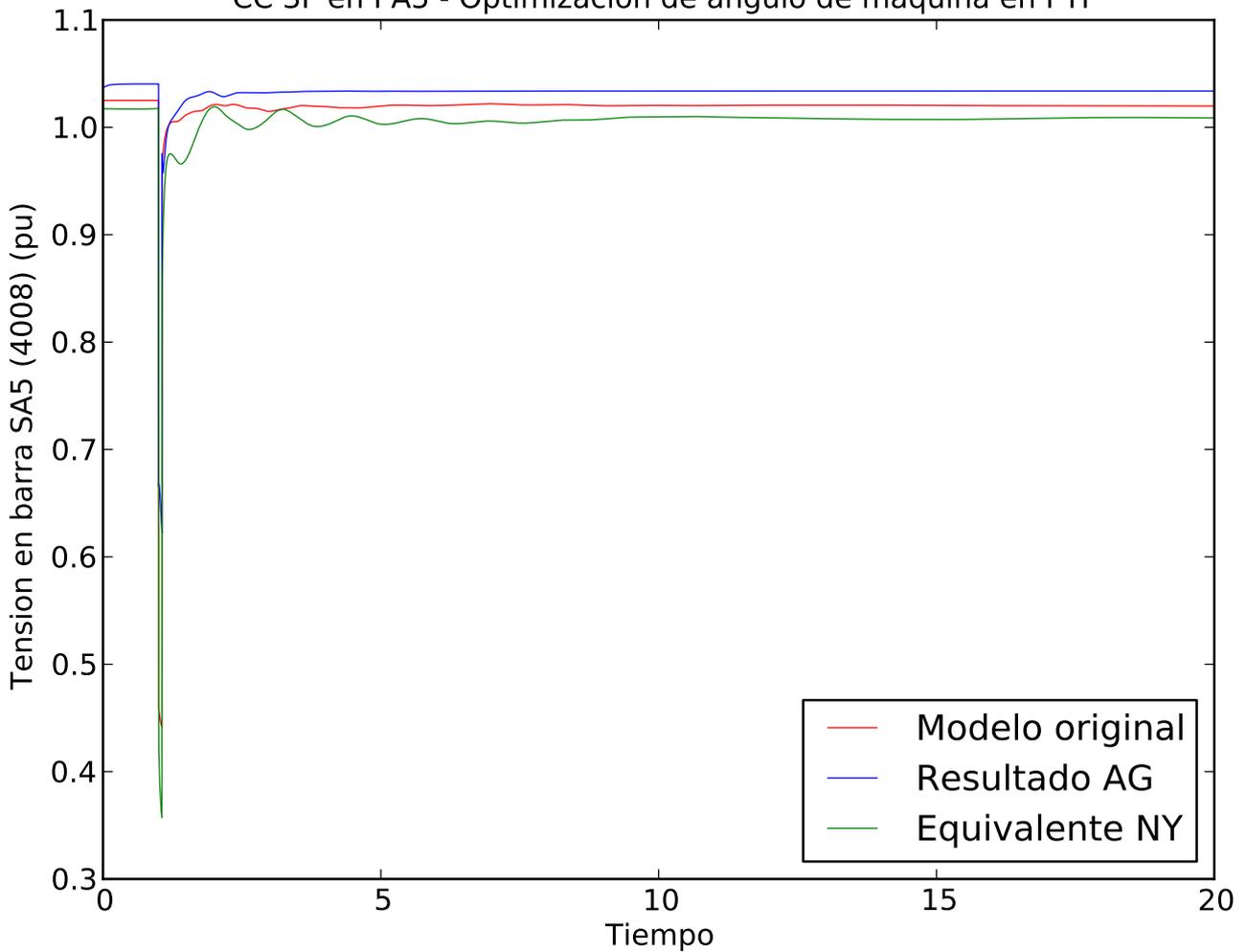
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



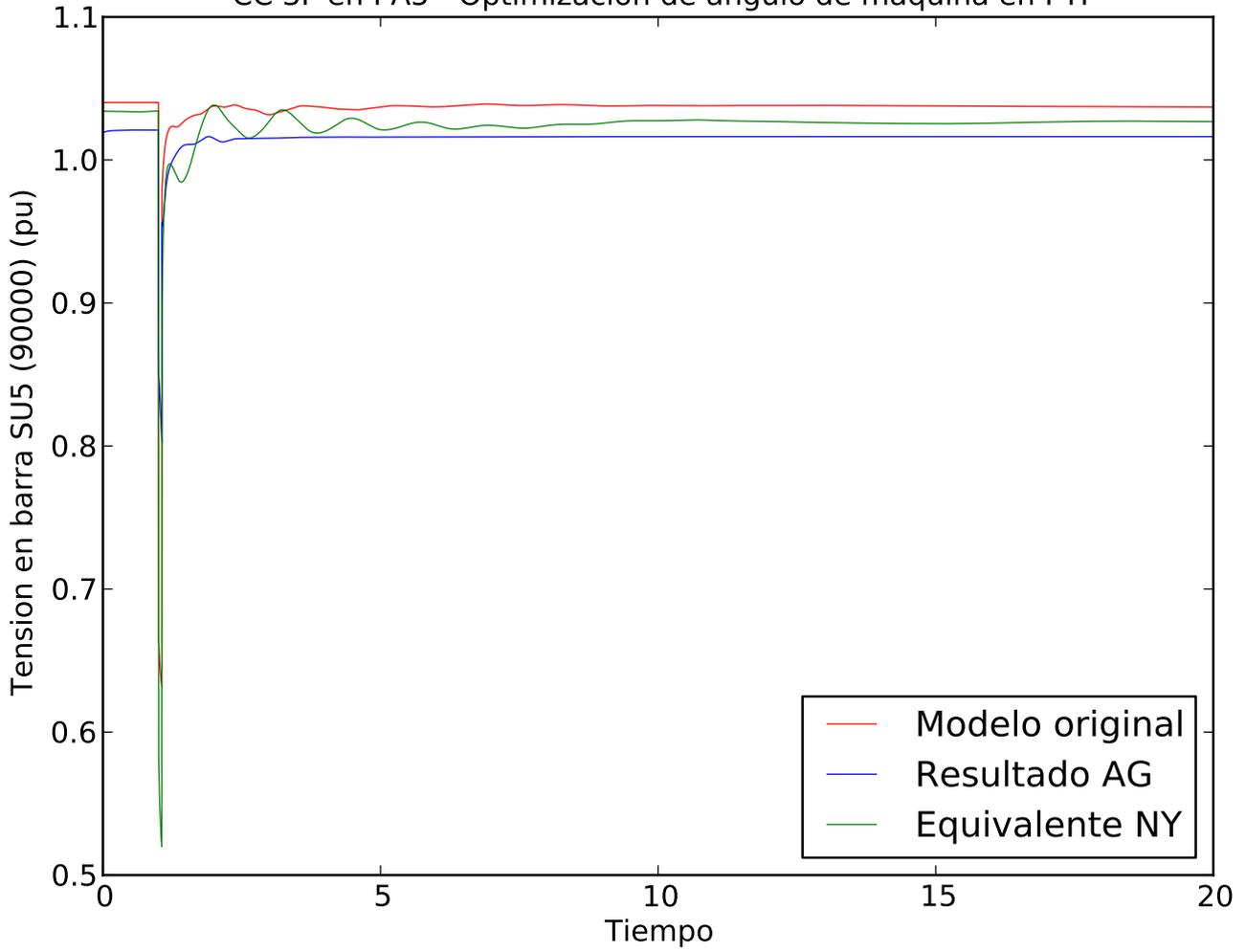
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



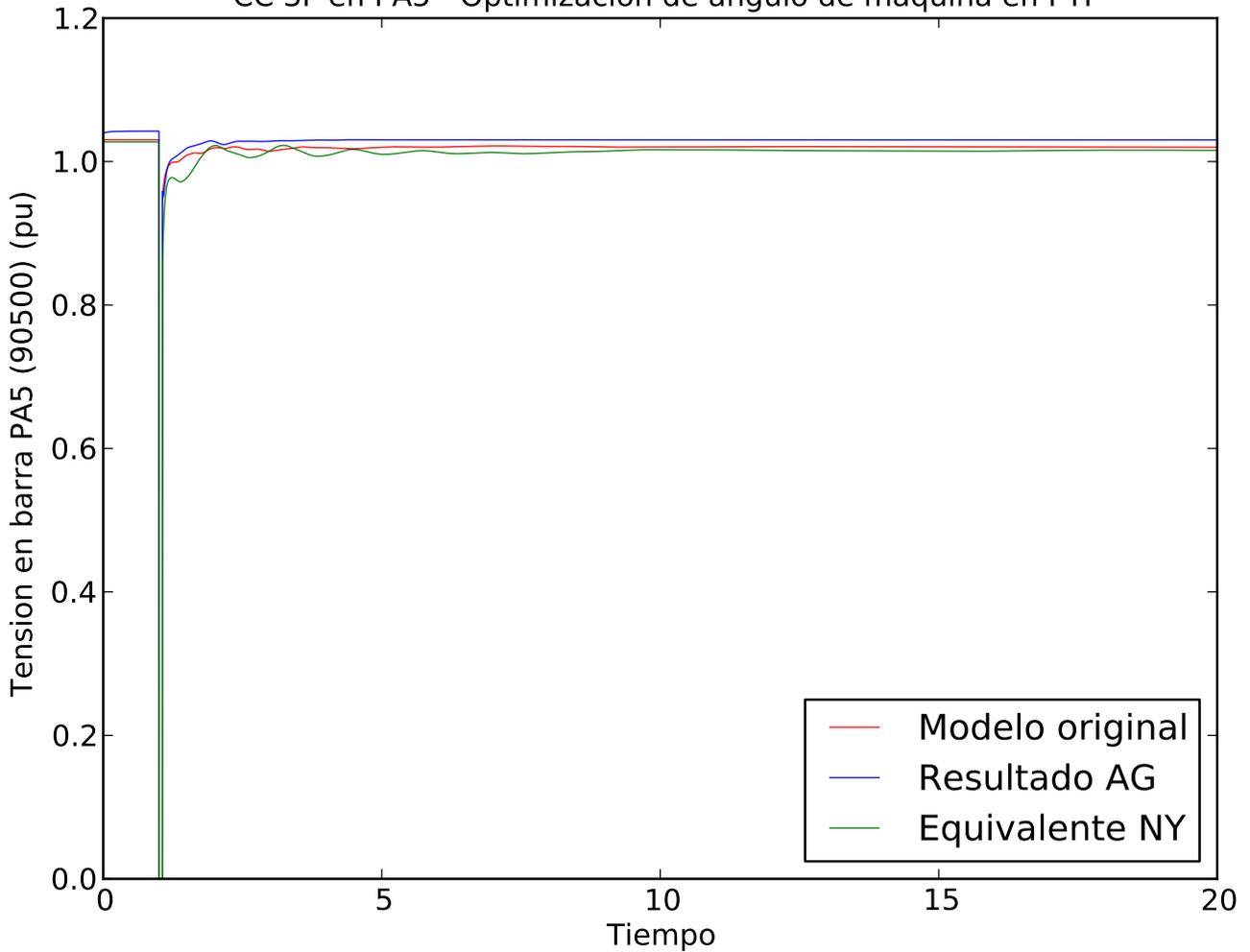
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



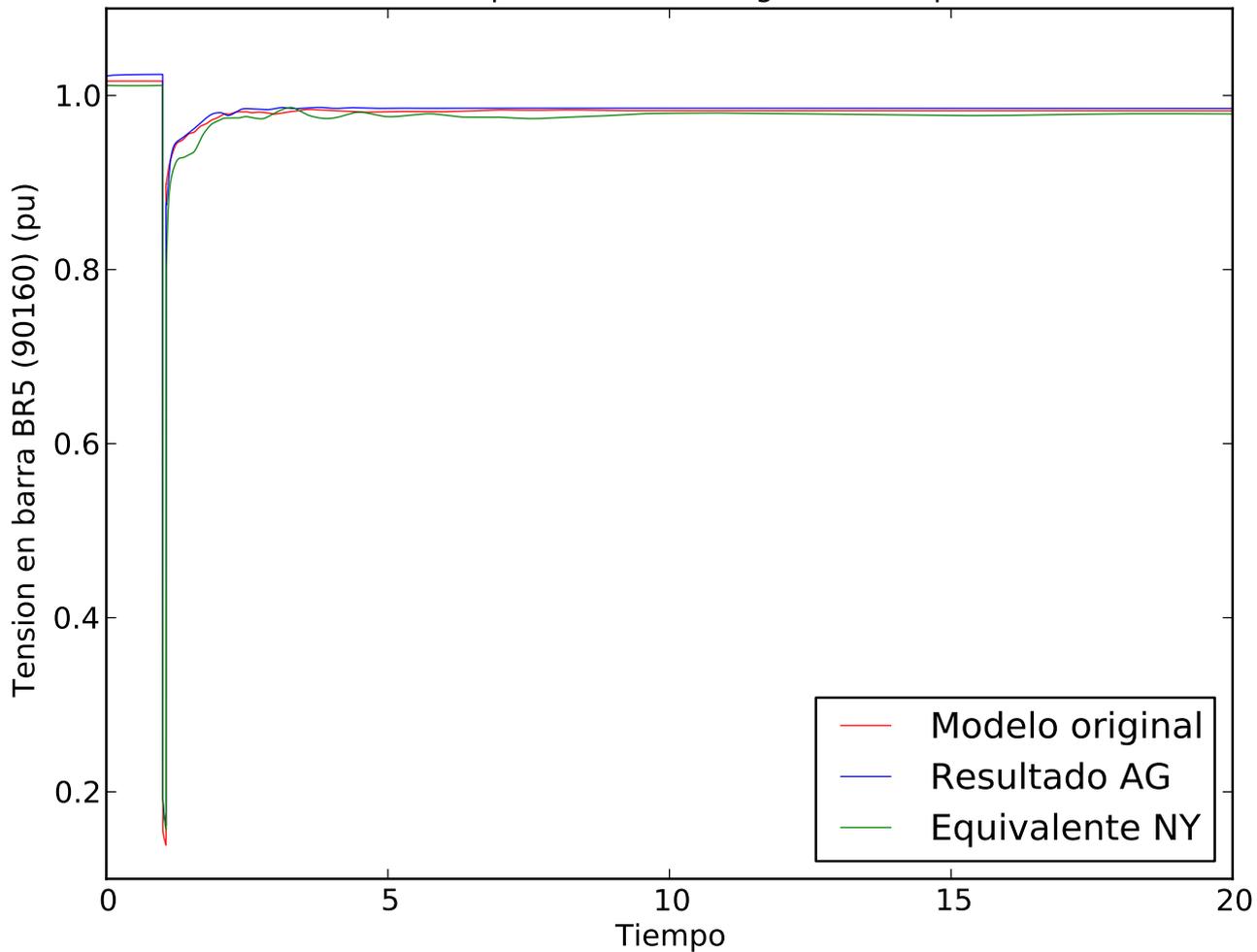
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



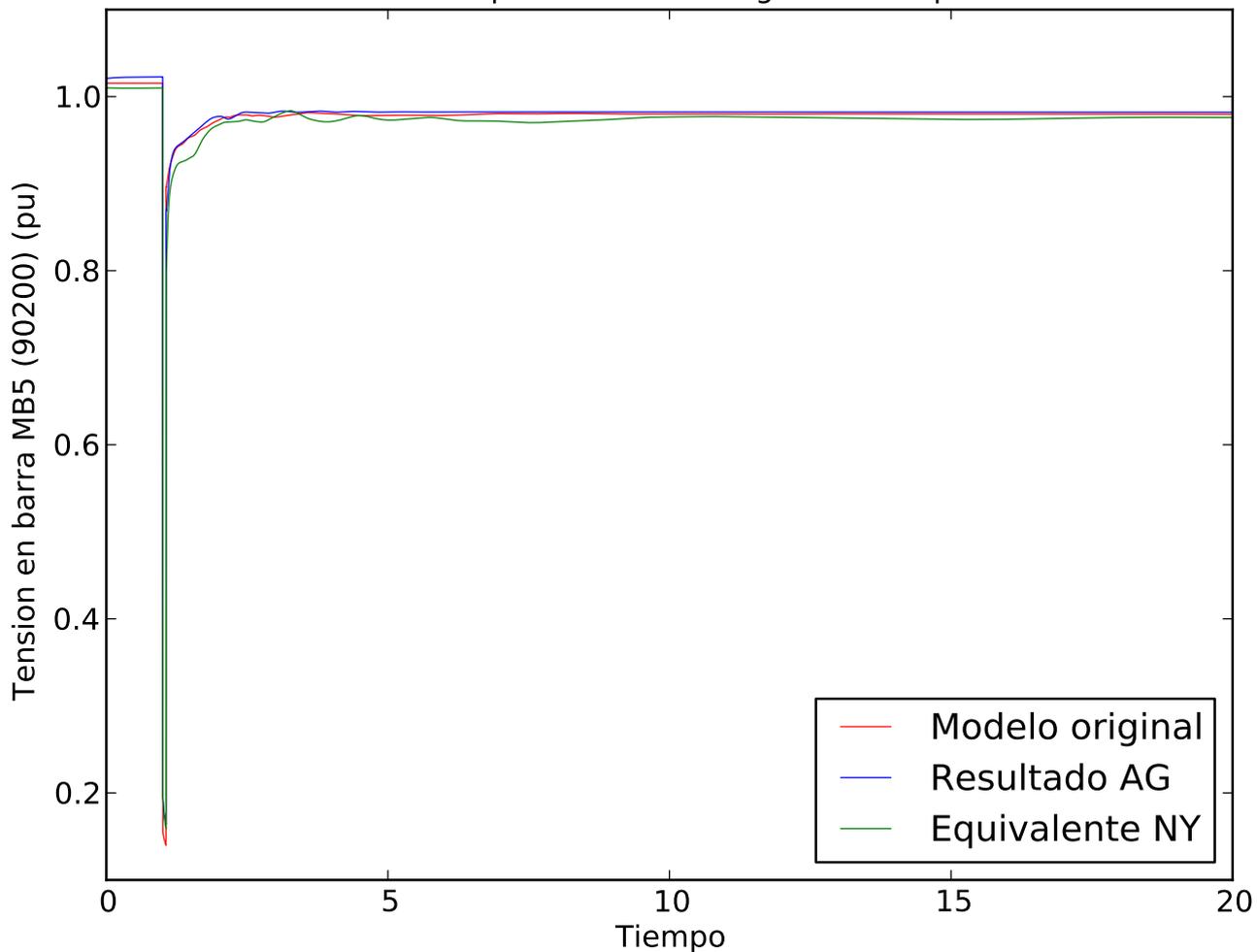
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



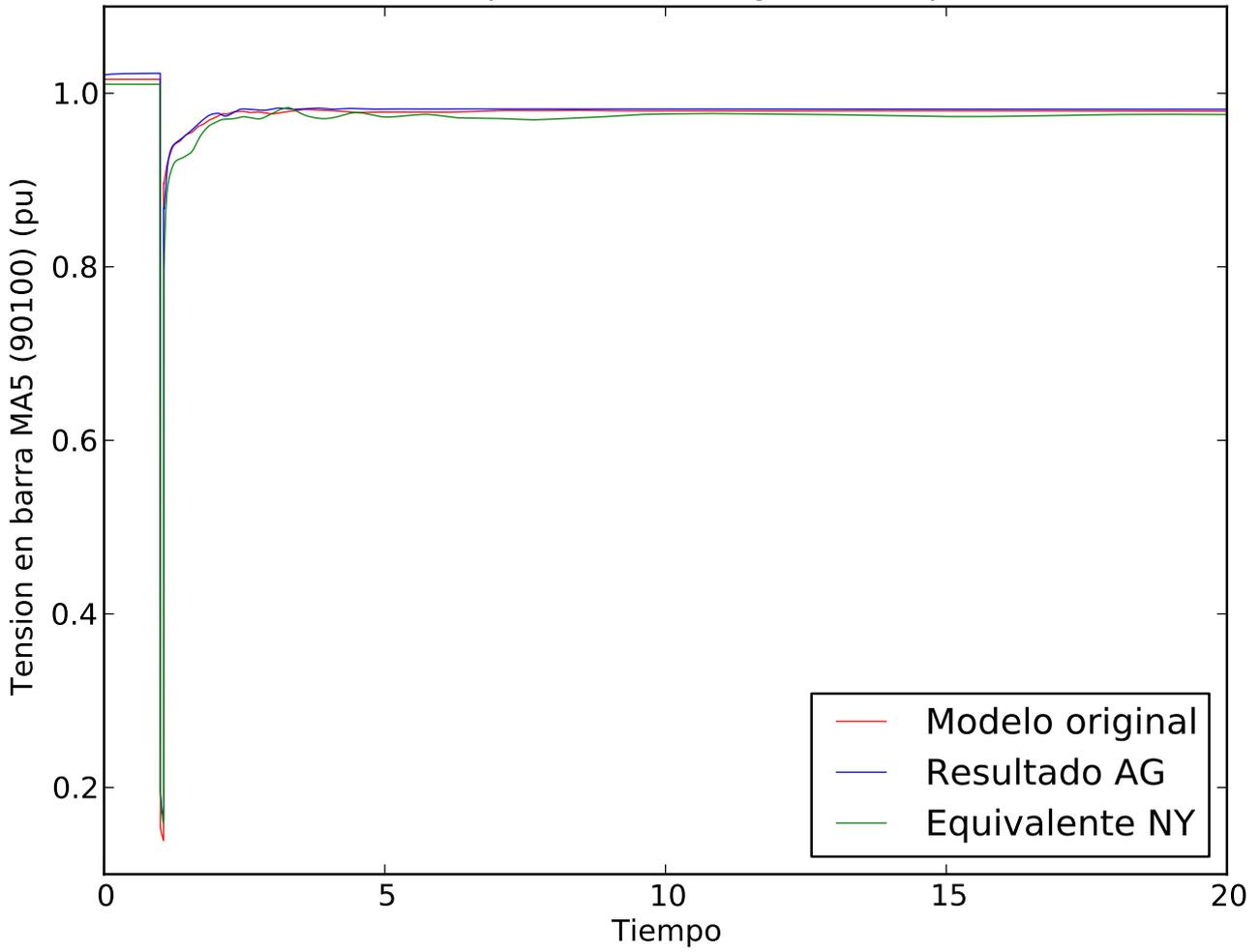
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



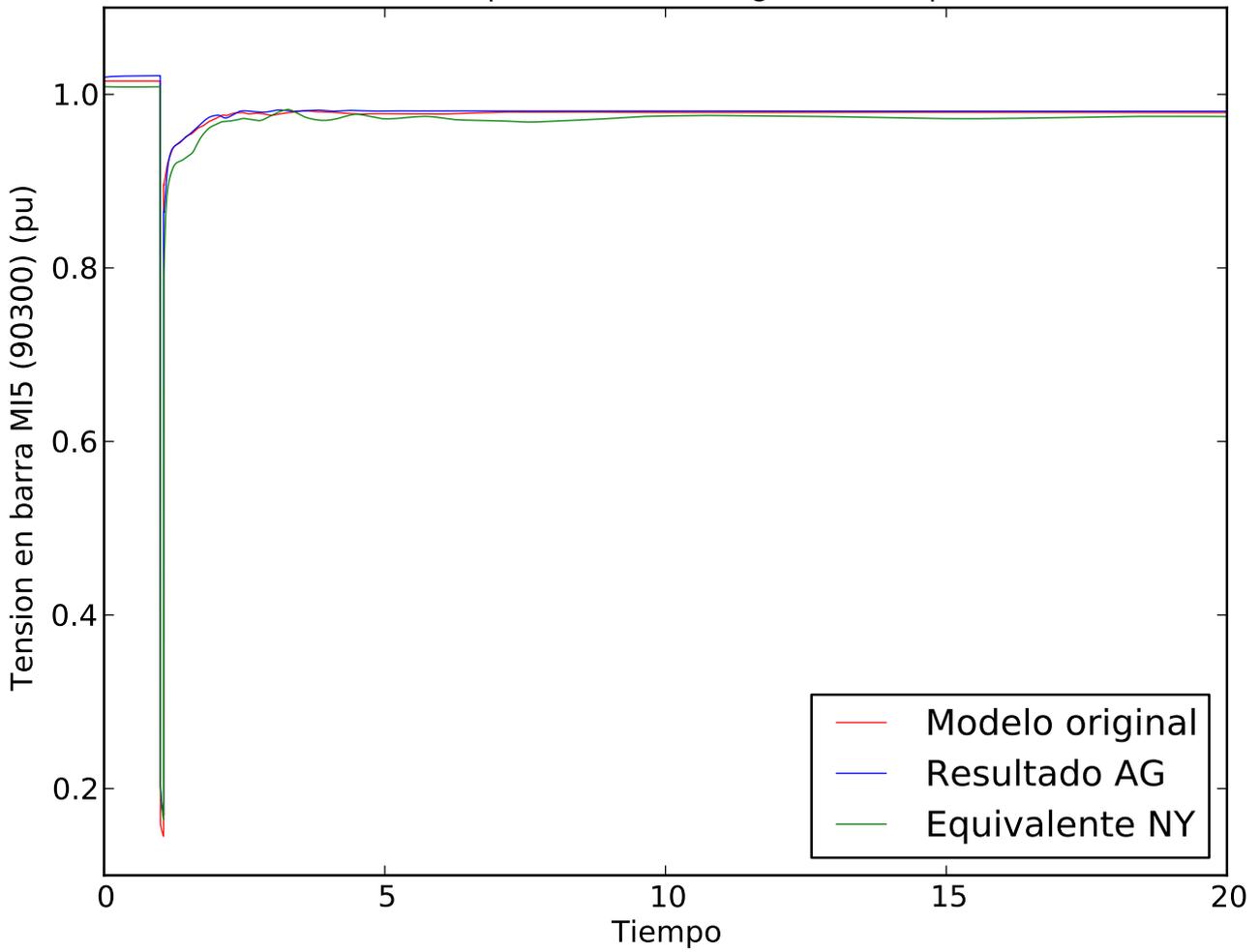
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



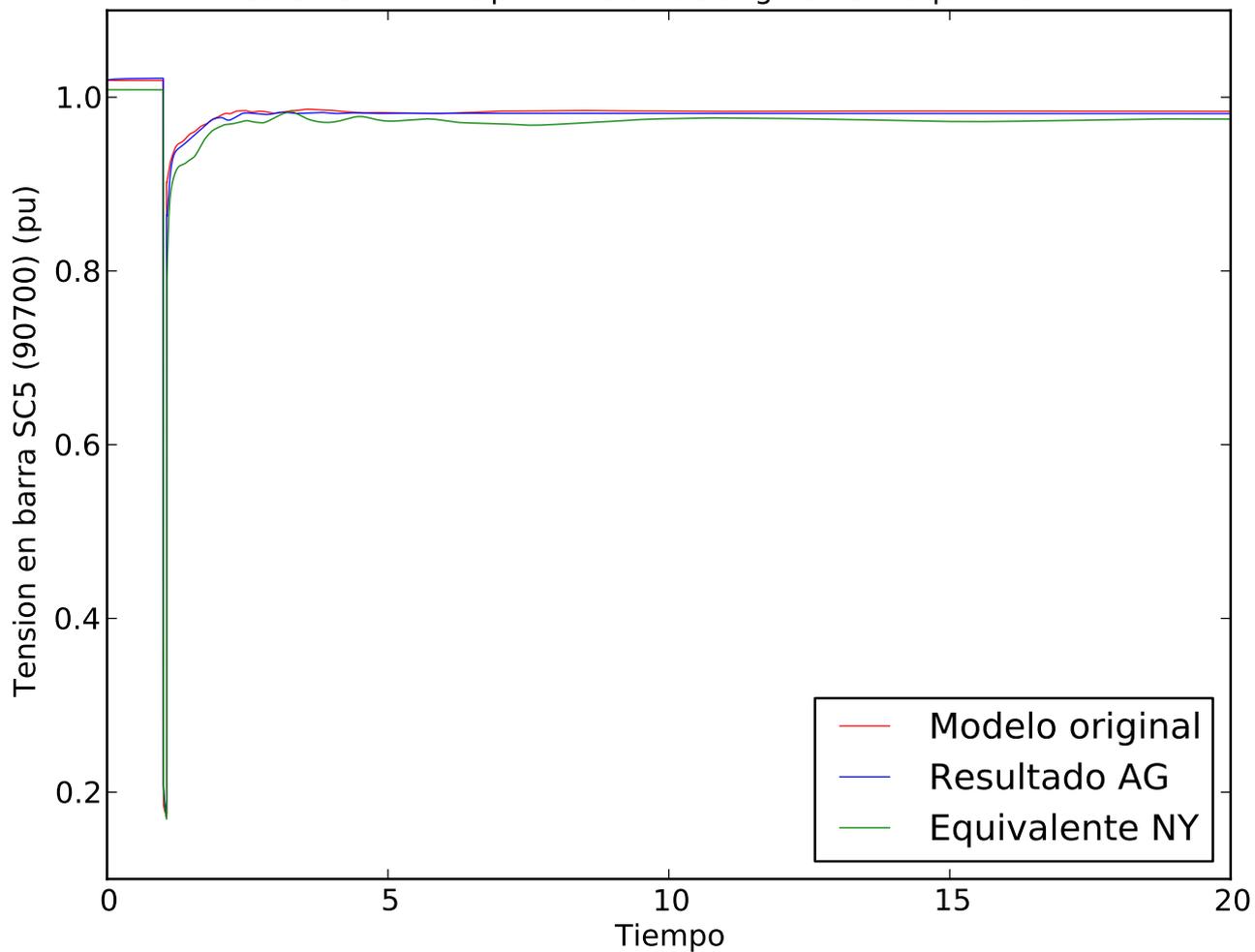
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



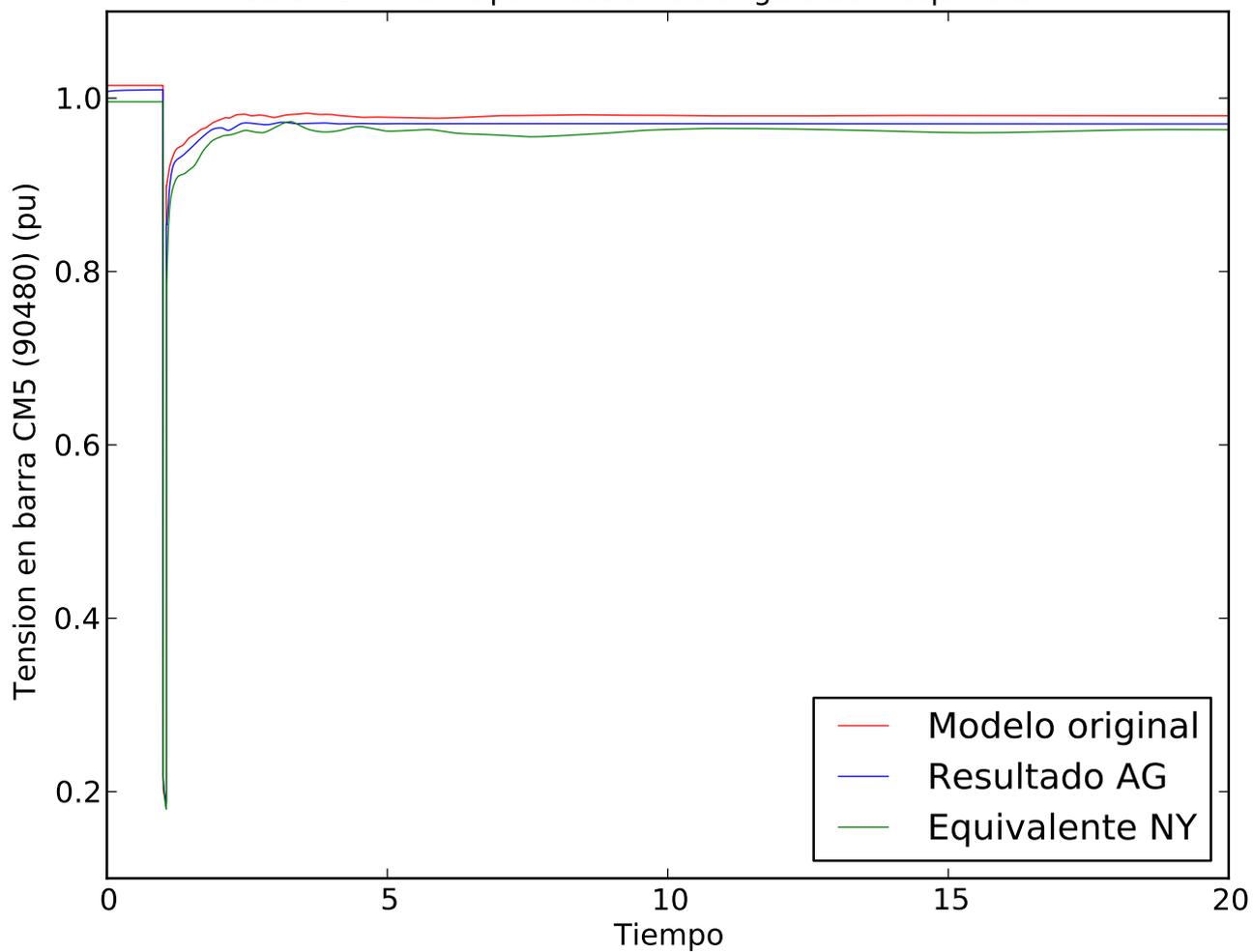
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



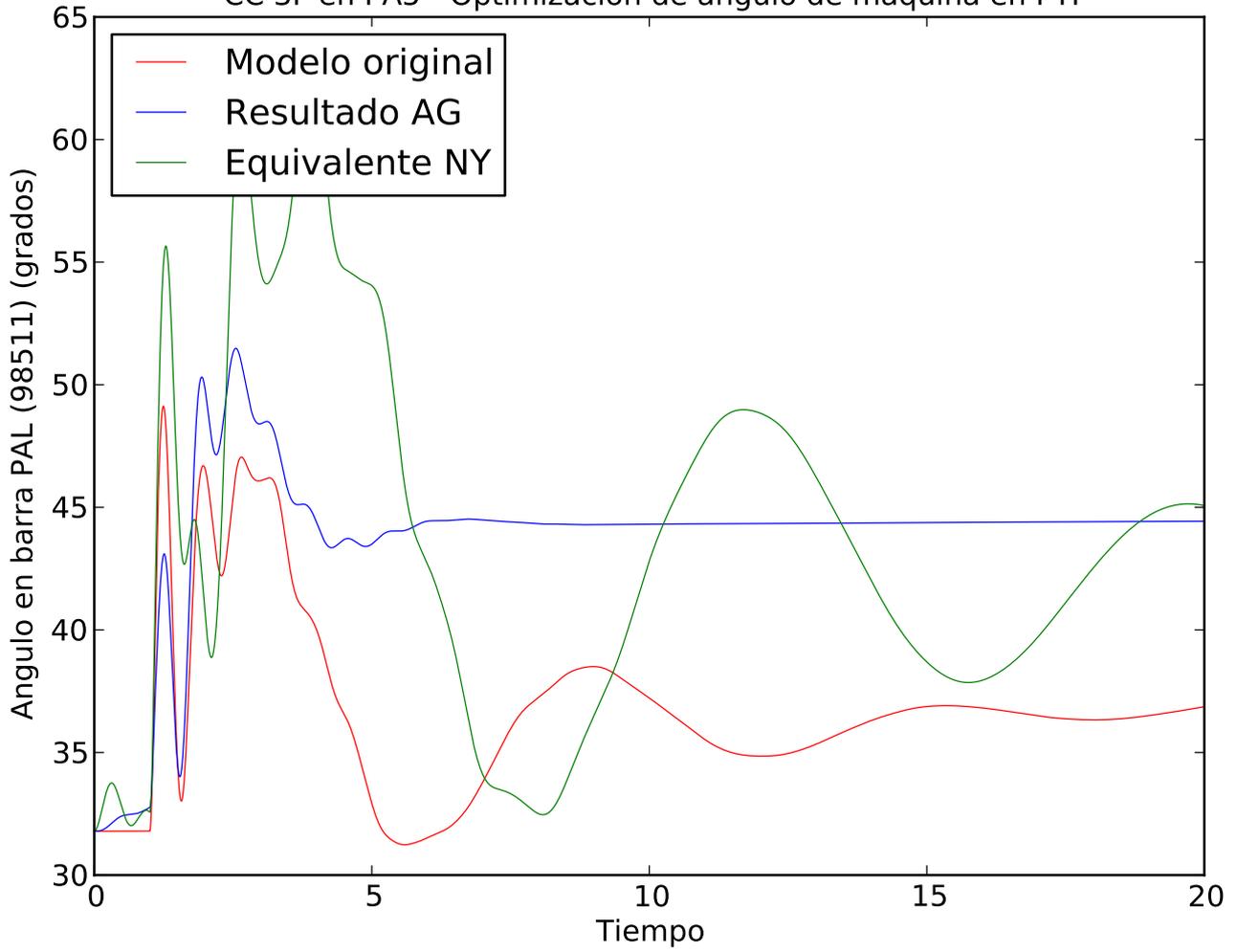
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



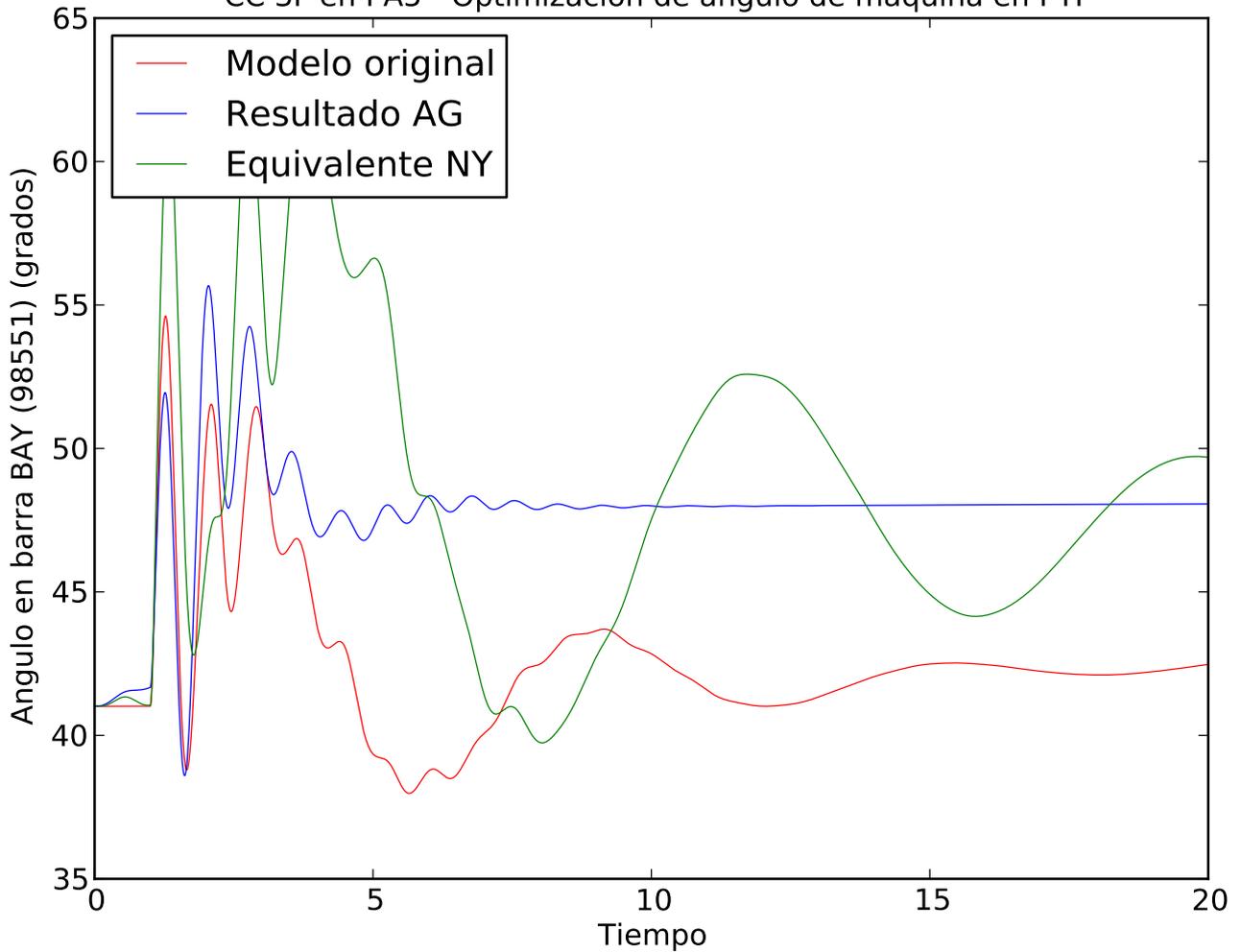
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



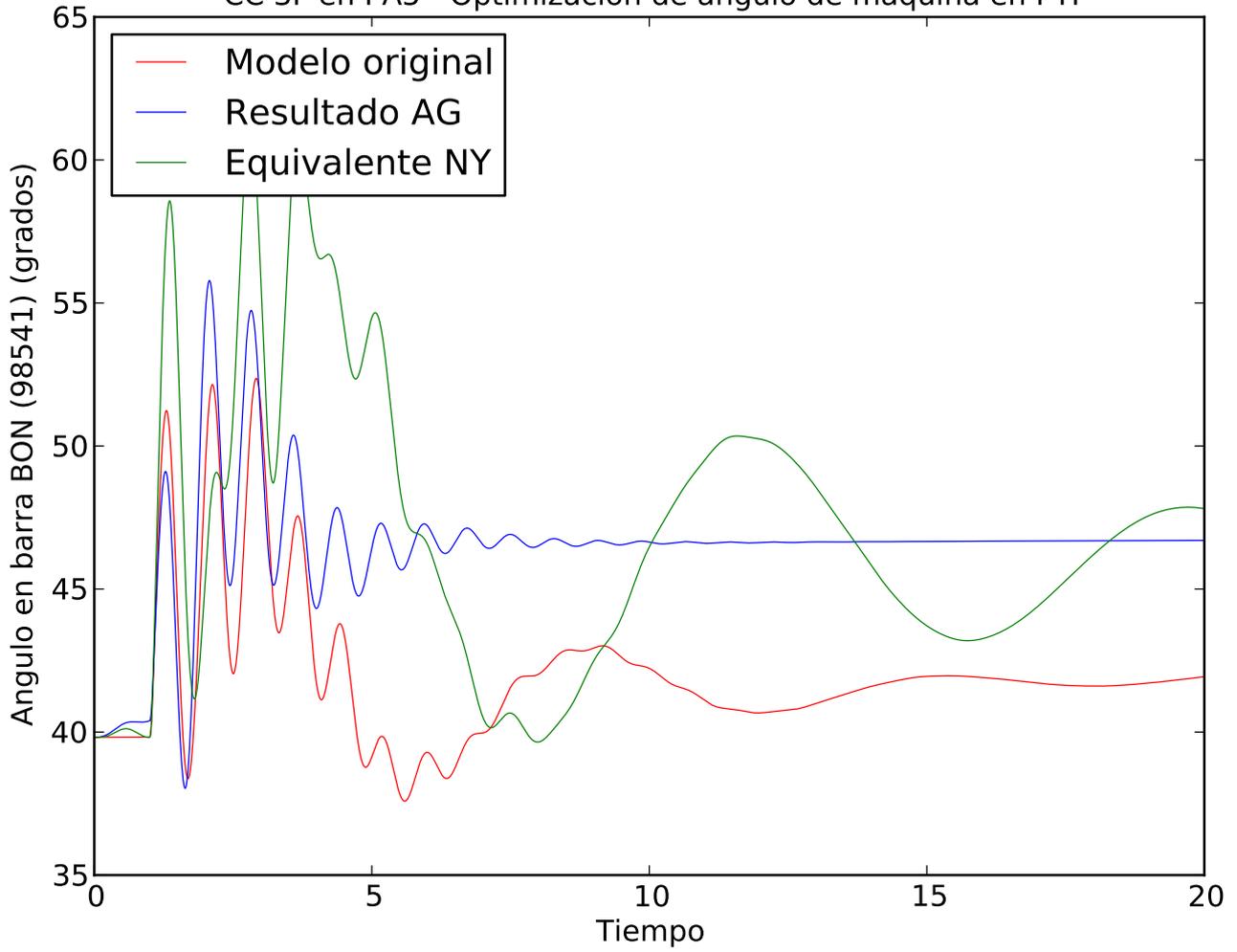
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



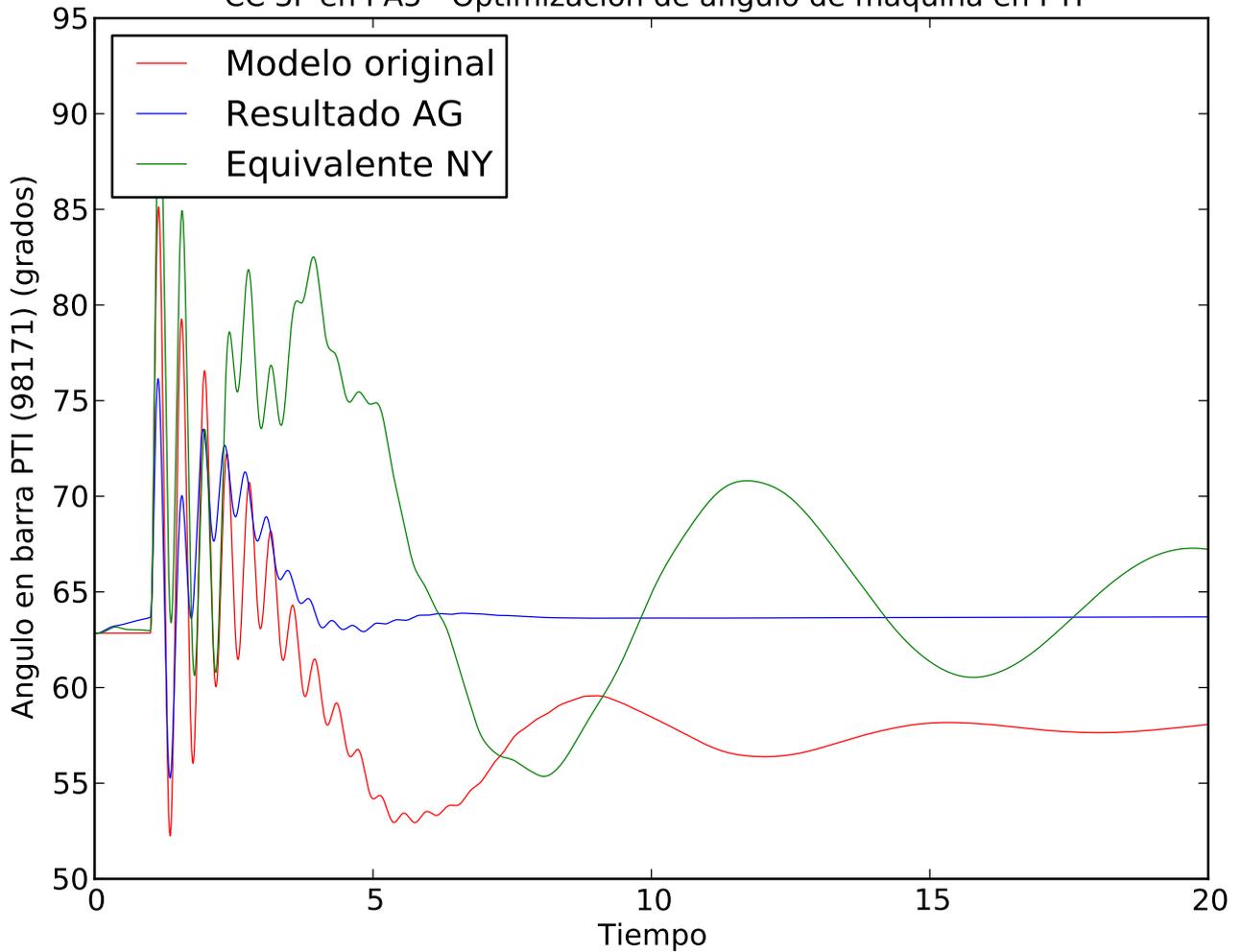
CC 3F en PA5 - Optimizacion de angulo de maquina en PTI



CC 3F en PA5 - Optimizacion de angulo de maquina en PTI

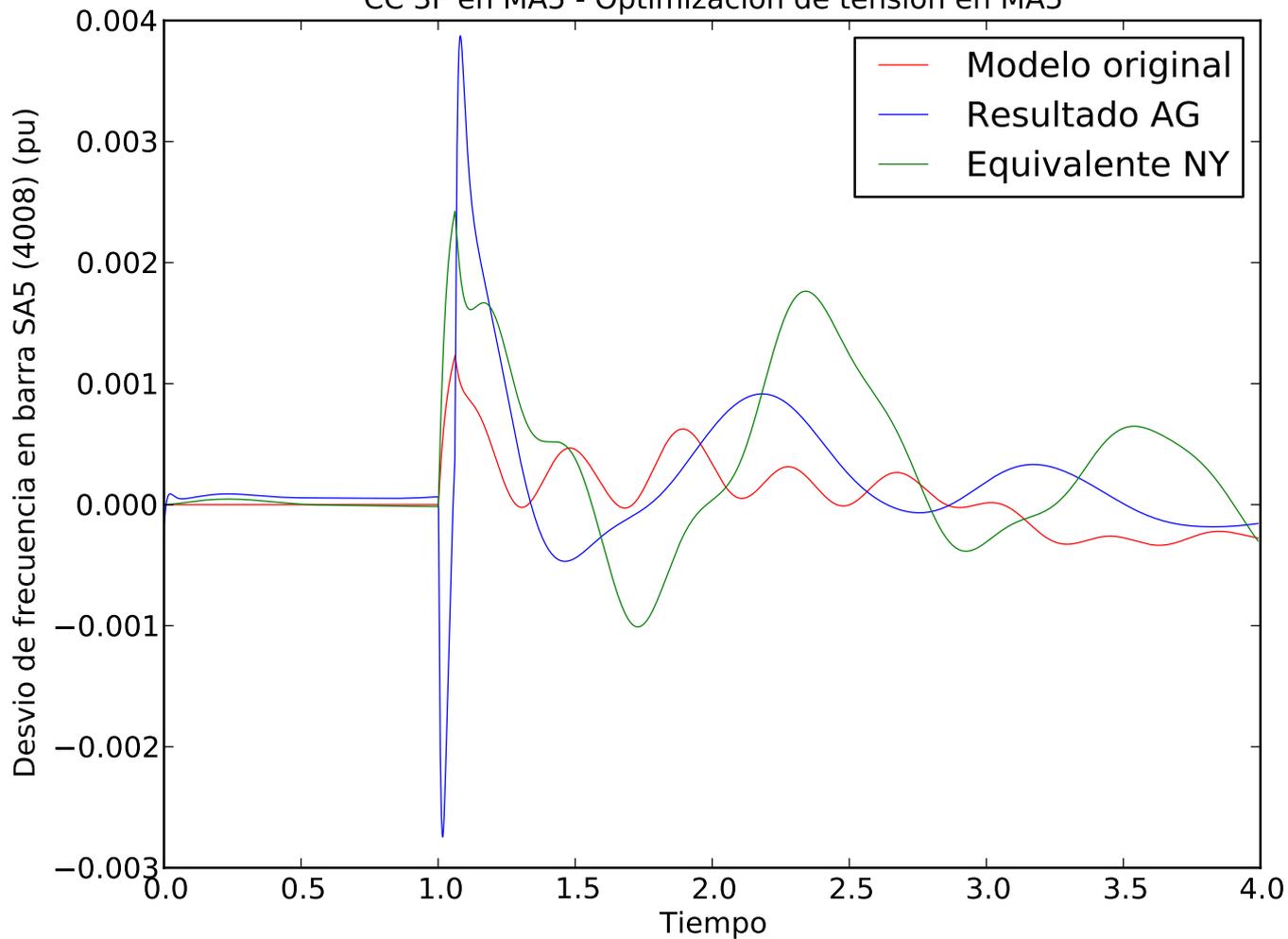


CC 3F en PA5 - Optimizacion de angulo de maquina en PTI

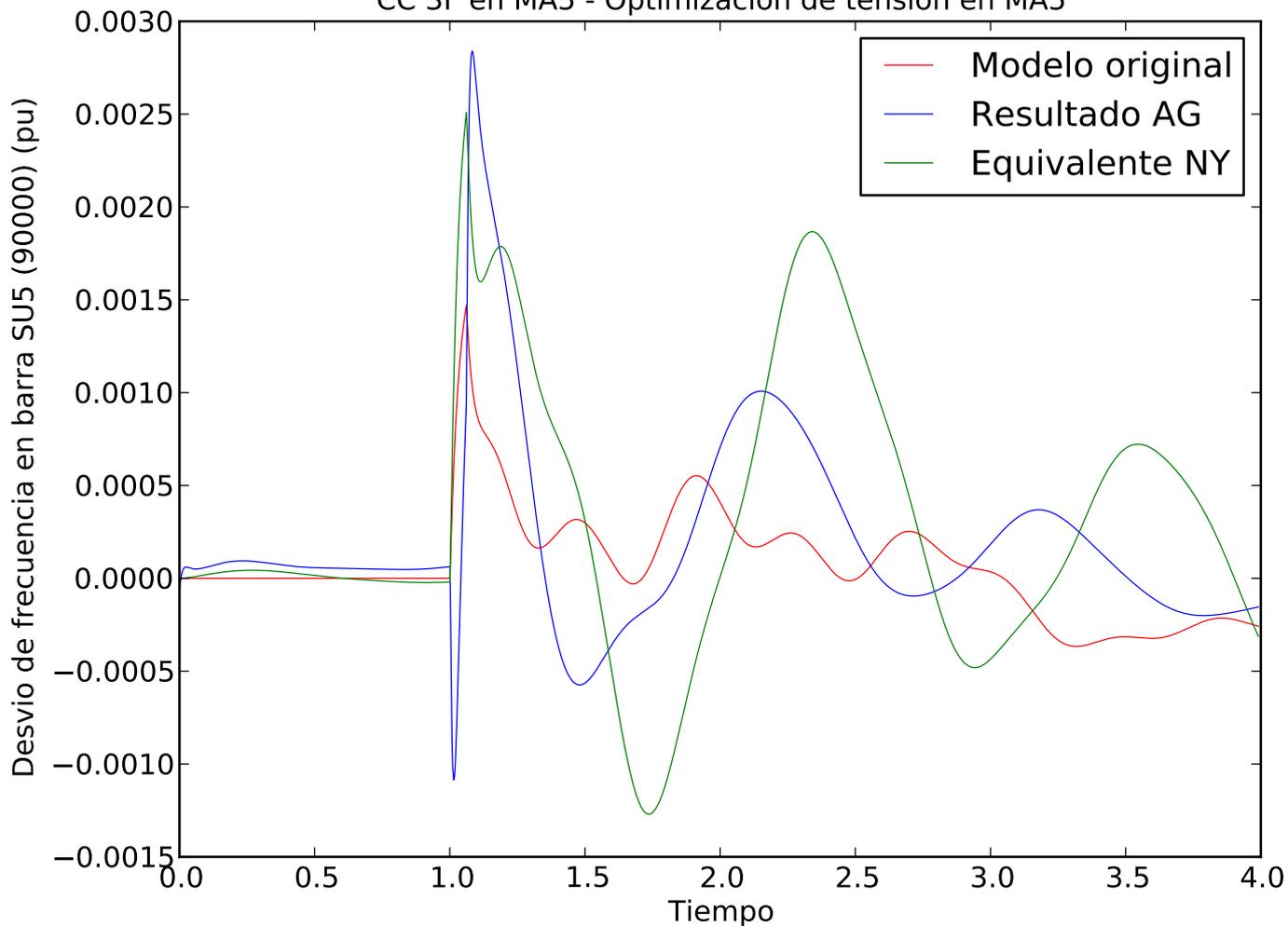


**C.8. Simulación 5: Cortocircuito 3F en MA5 con salida de línea en equivalente NY 3, optimizando tensión en MA5 en 4 segundos. Comportamiento en 4 segundos**

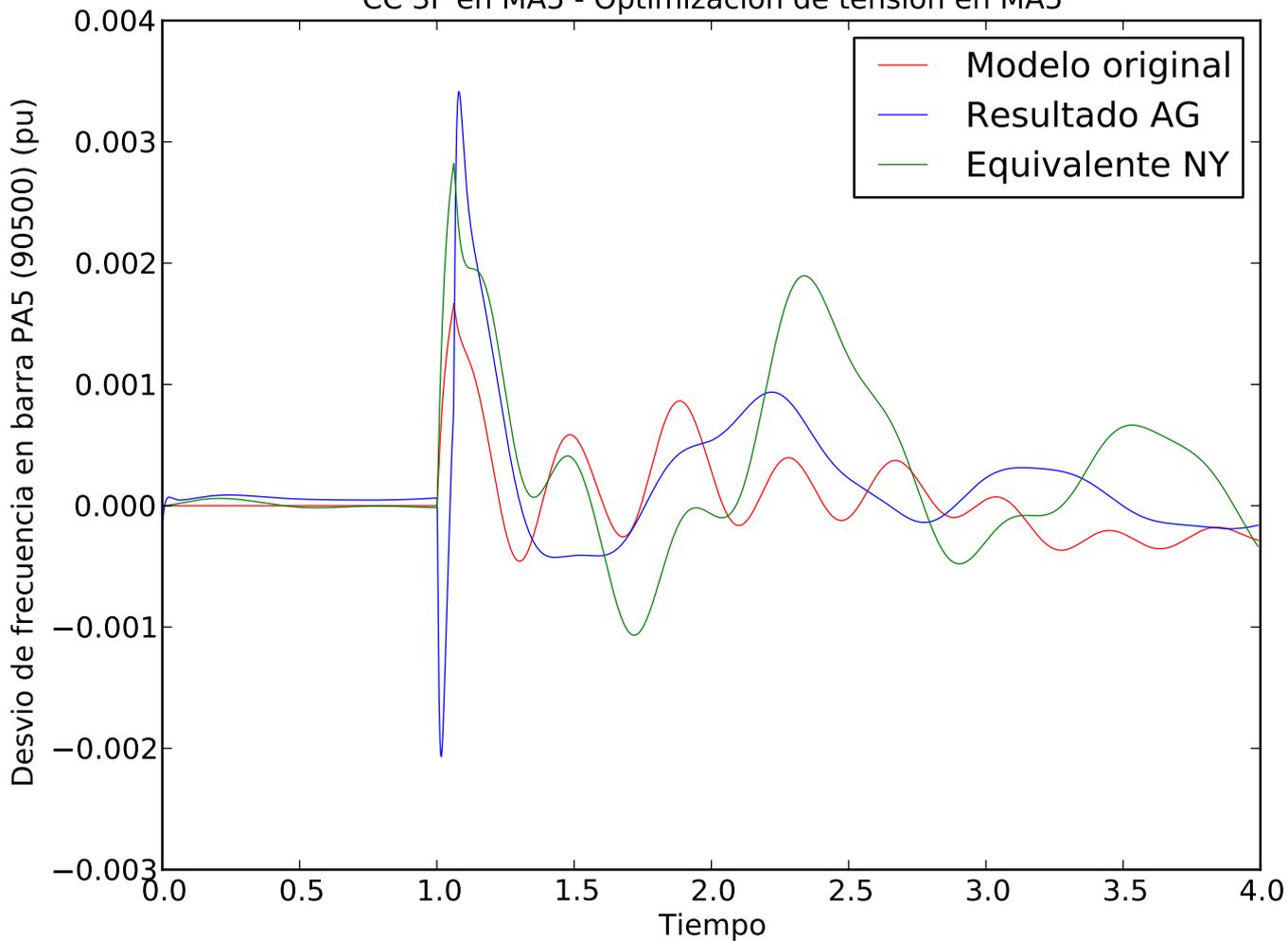
CC 3F en MA5 - Optimizacion de tension en MA5



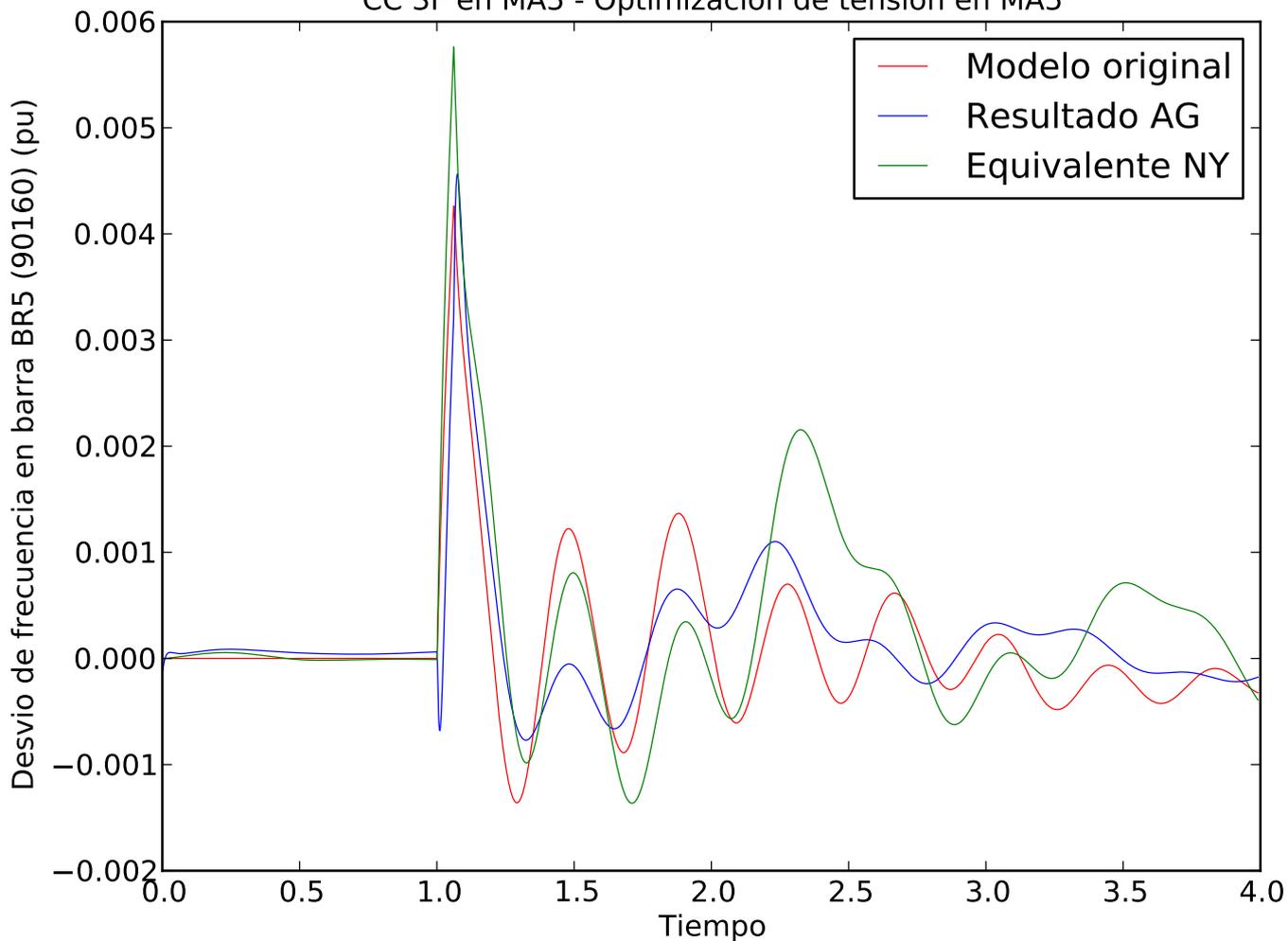
CC 3F en MA5 - Optimizacion de tension en MA5



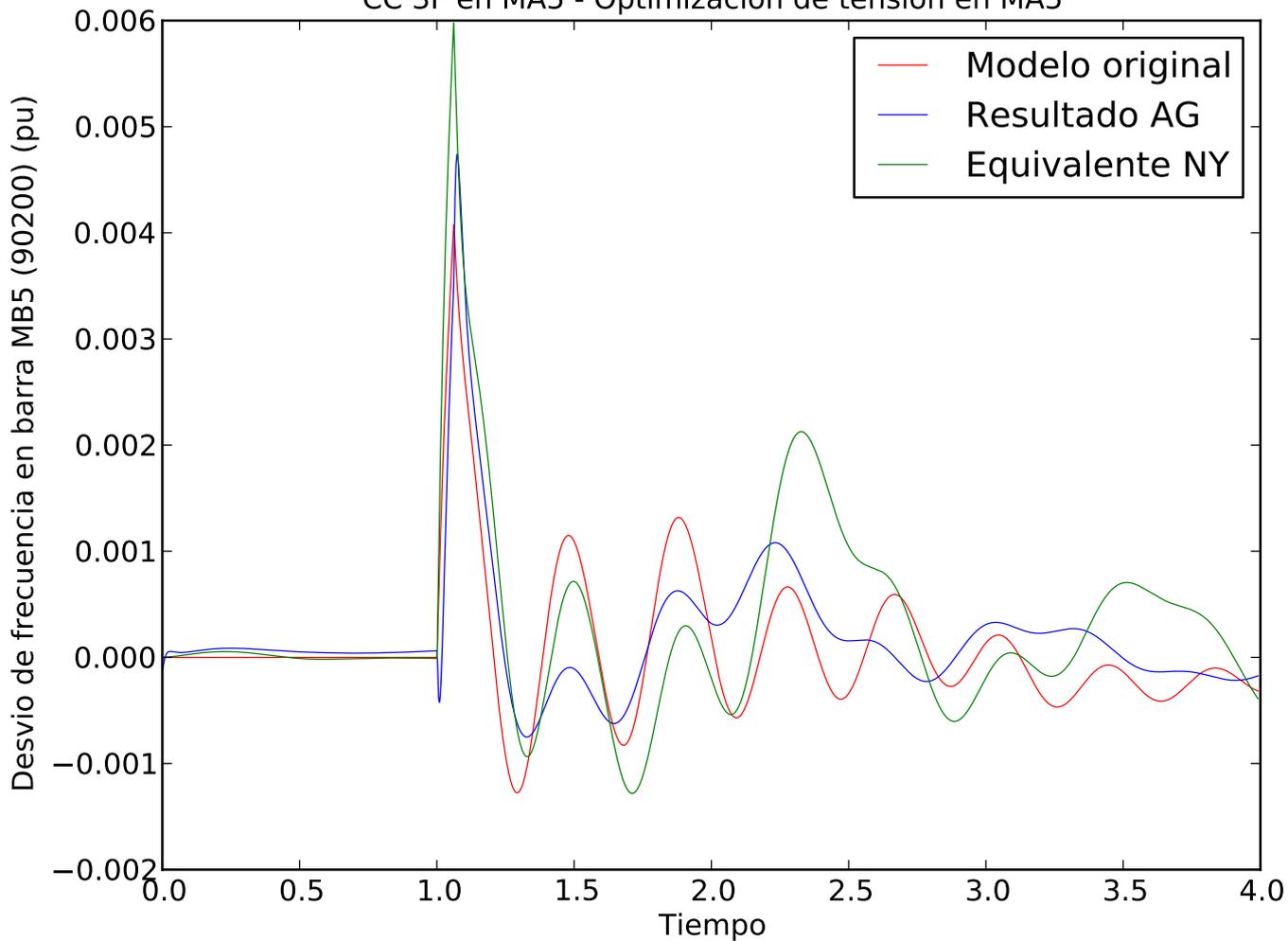
CC 3F en MA5 - Optimizacion de tension en MA5



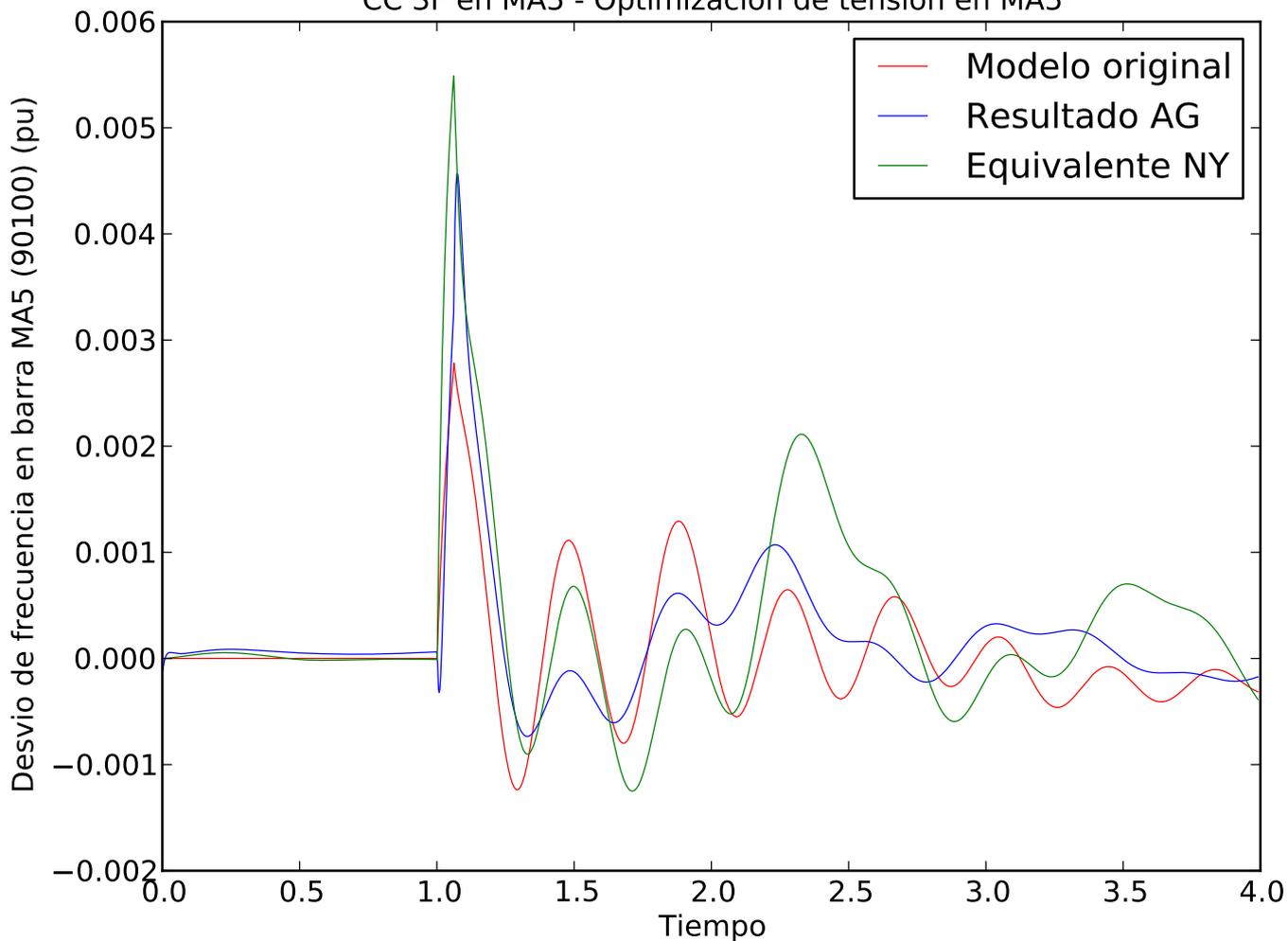
CC 3F en MA5 - Optimizacion de tension en MA5



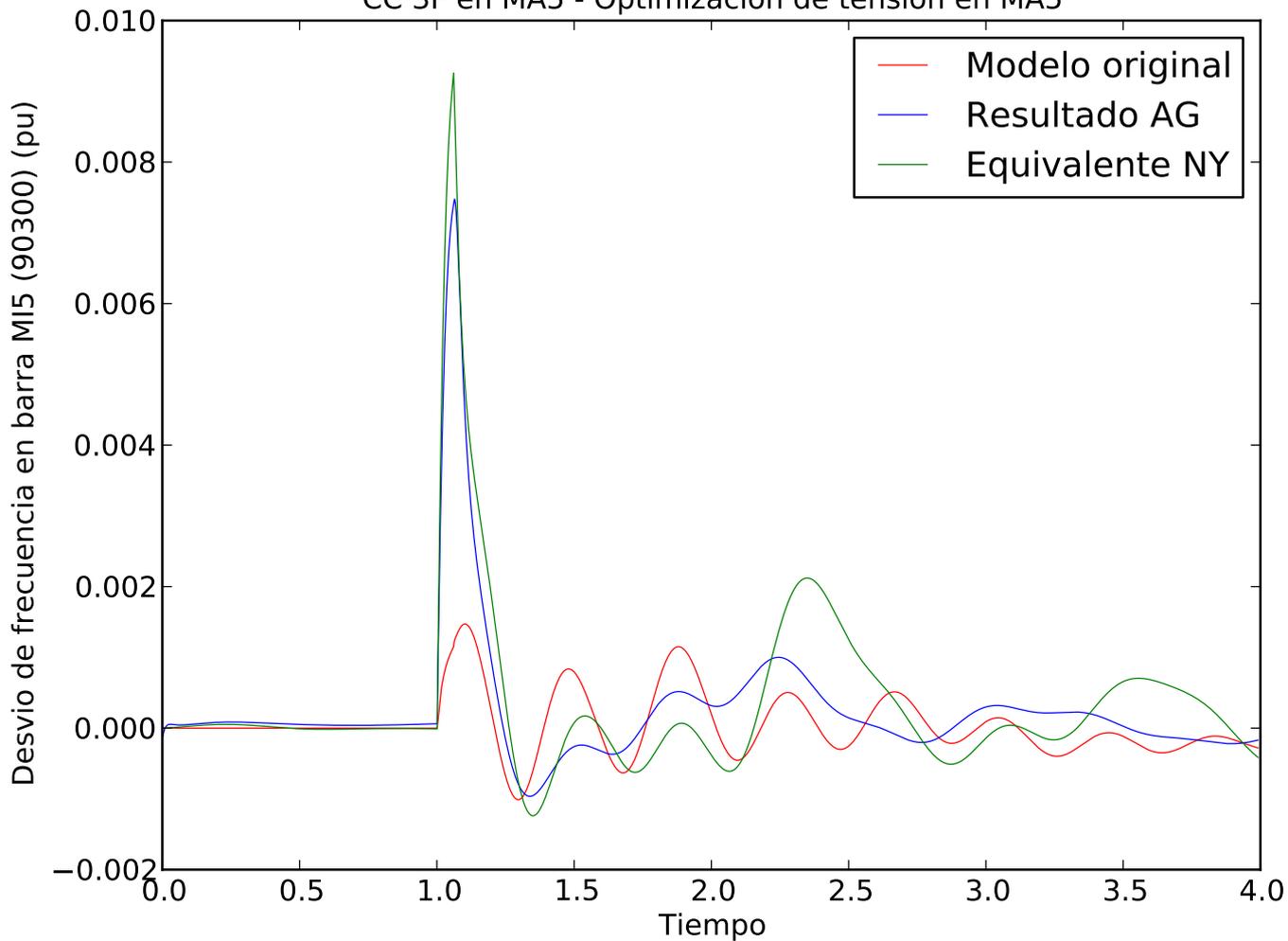
CC 3F en MA5 - Optimizacion de tension en MA5



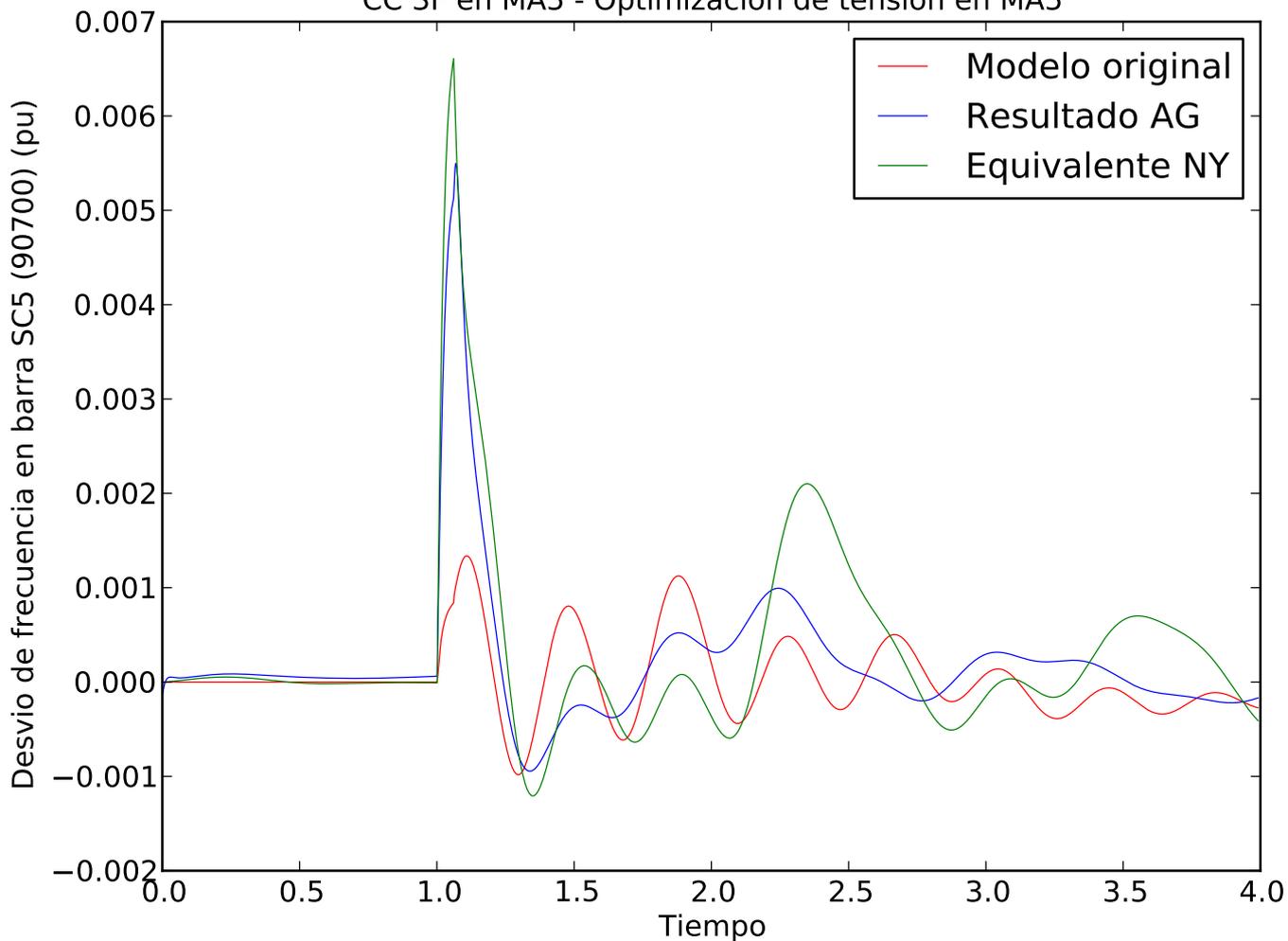
CC 3F en MA5 - Optimizacion de tension en MA5



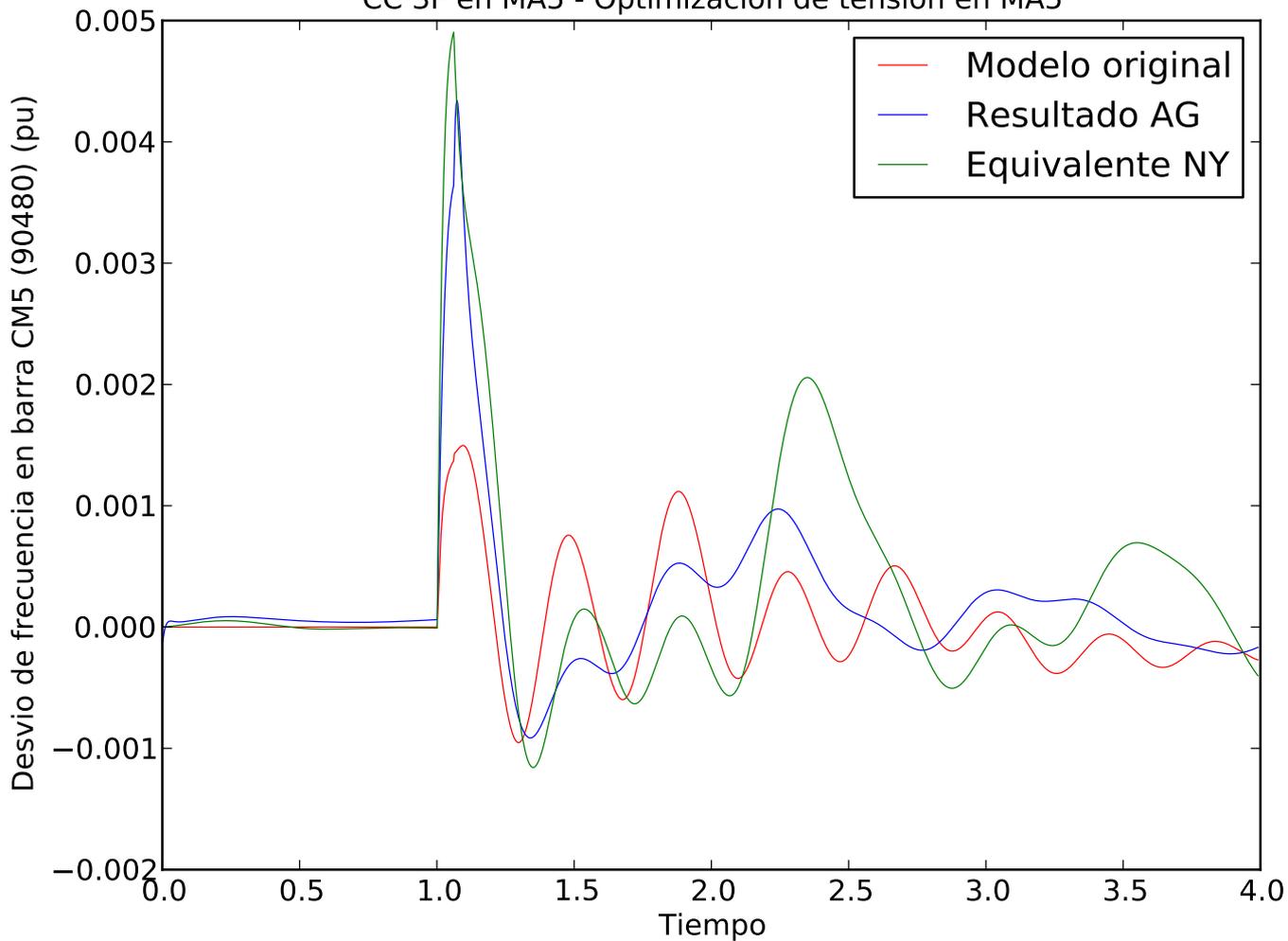
CC 3F en MA5 - Optimizacion de tension en MA5



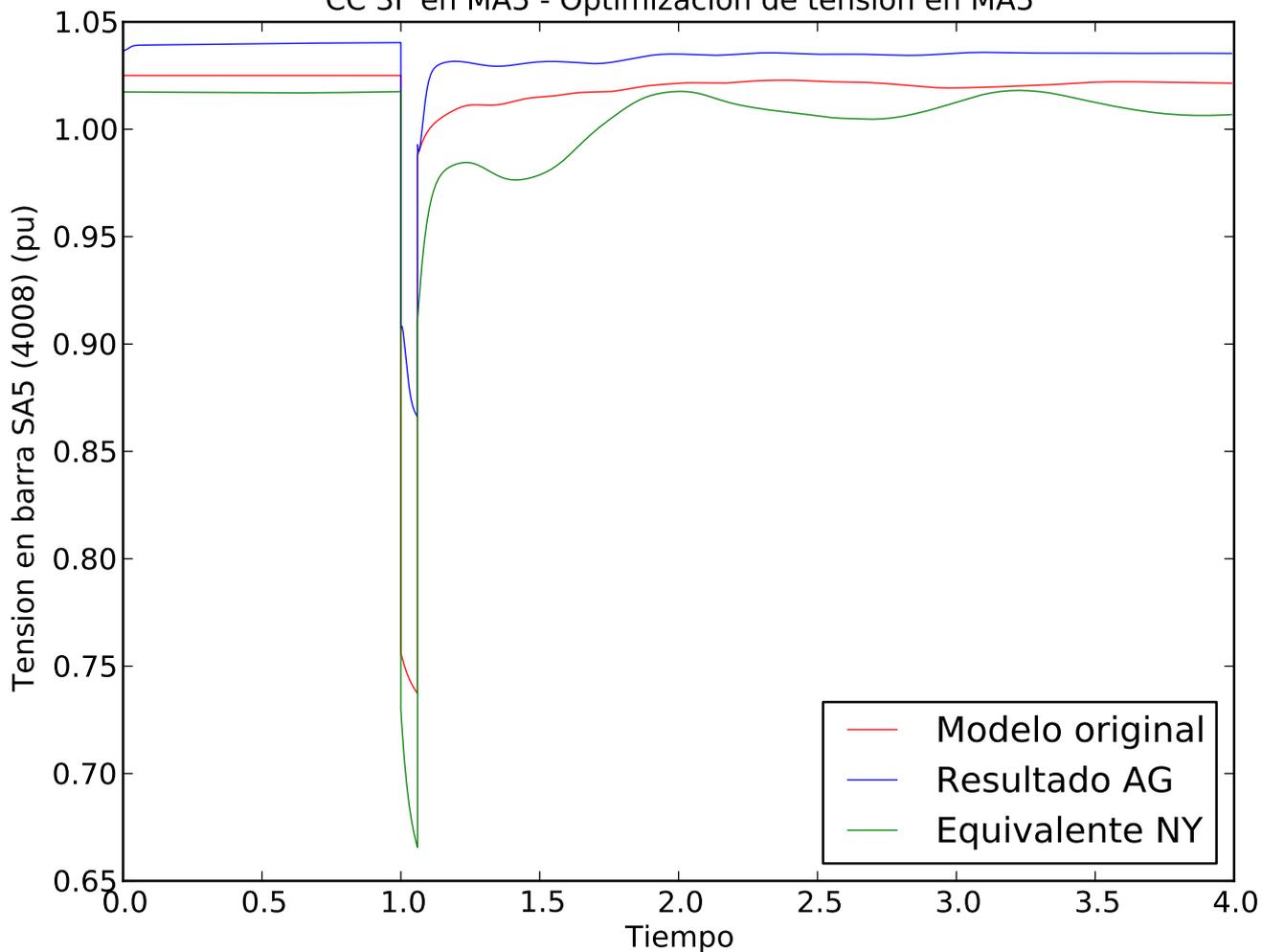
CC 3F en MA5 - Optimizacion de tension en MA5



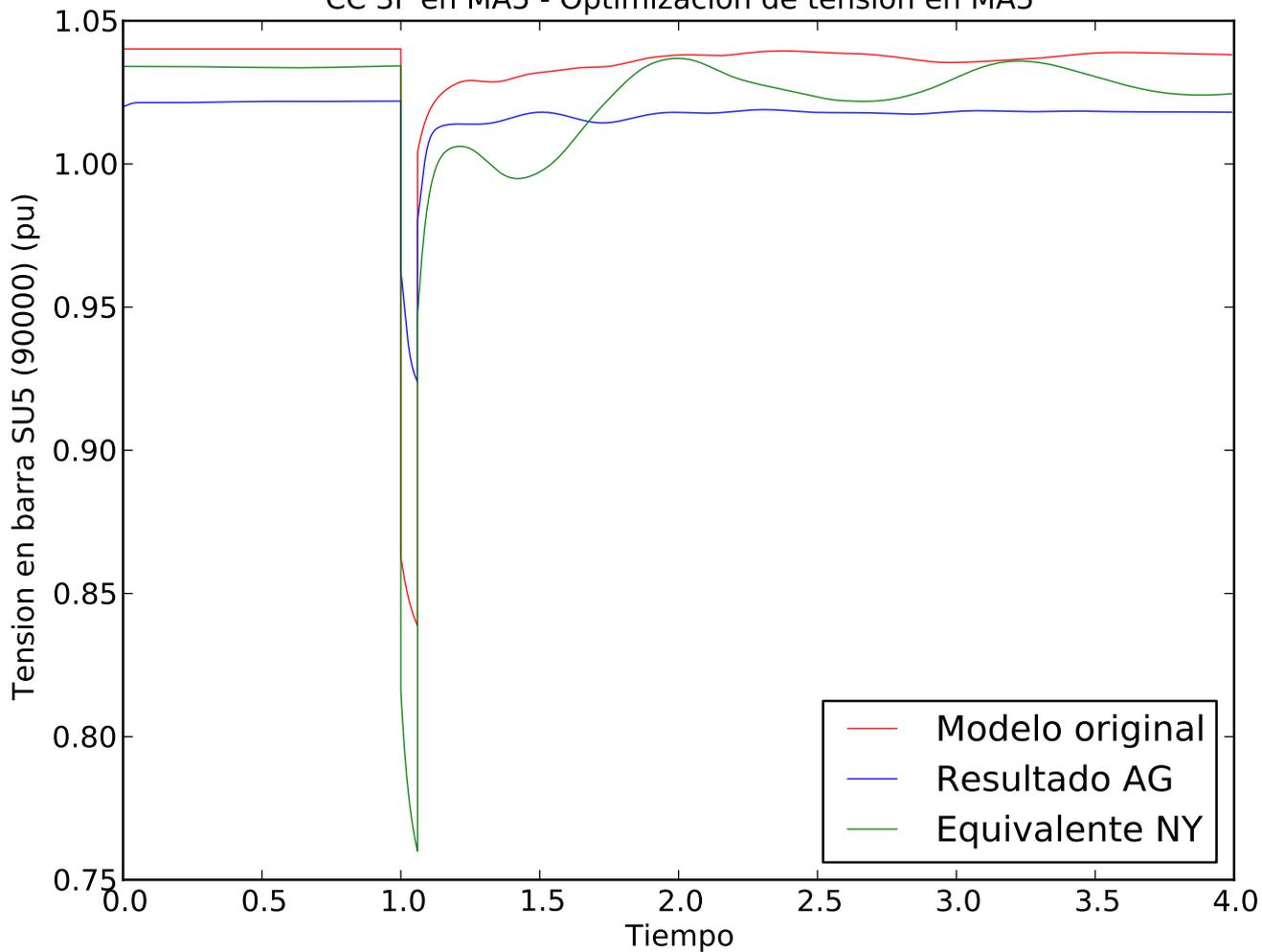
CC 3F en MA5 - Optimizacion de tension en MA5



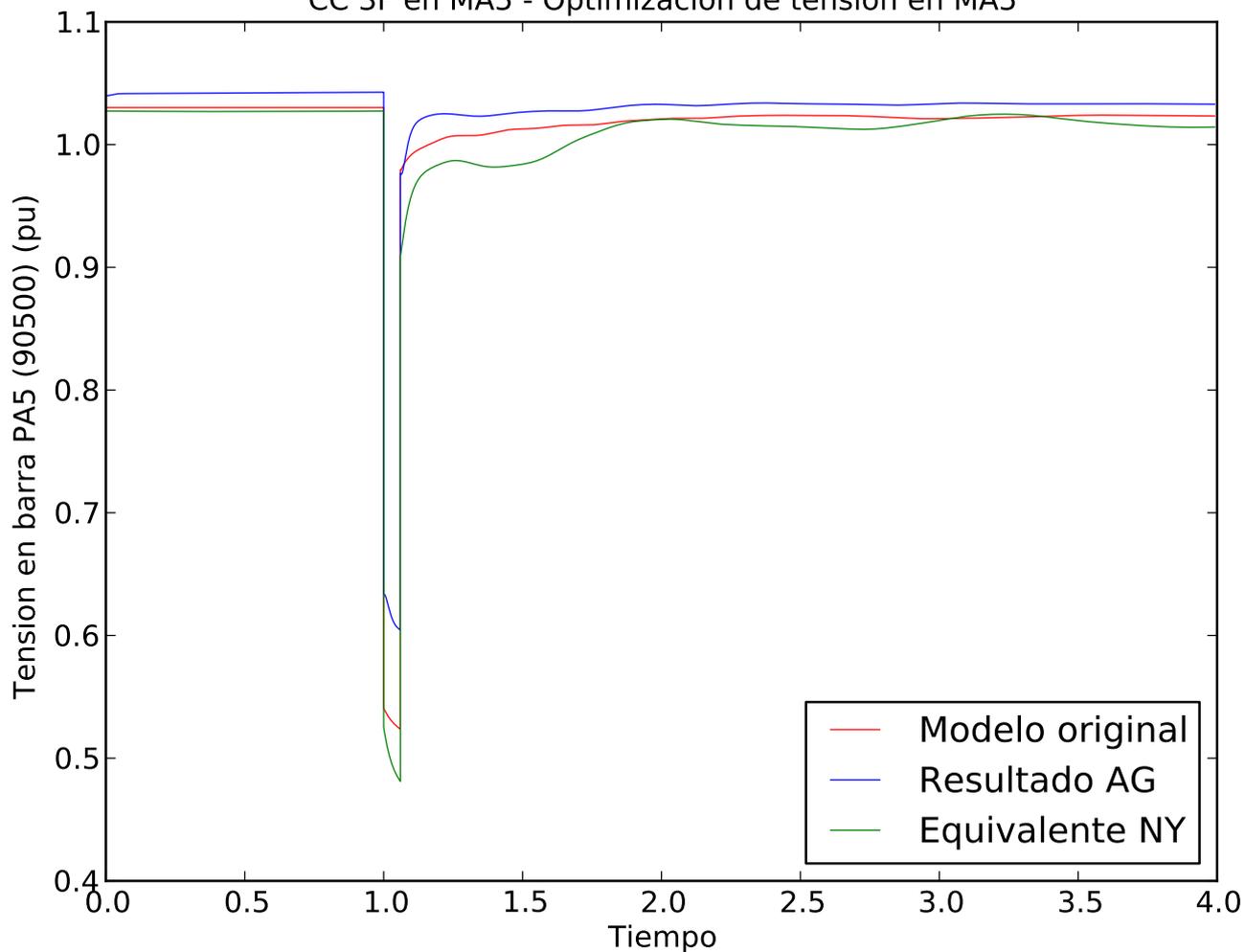
CC 3F en MA5 - Optimizacion de tension en MA5



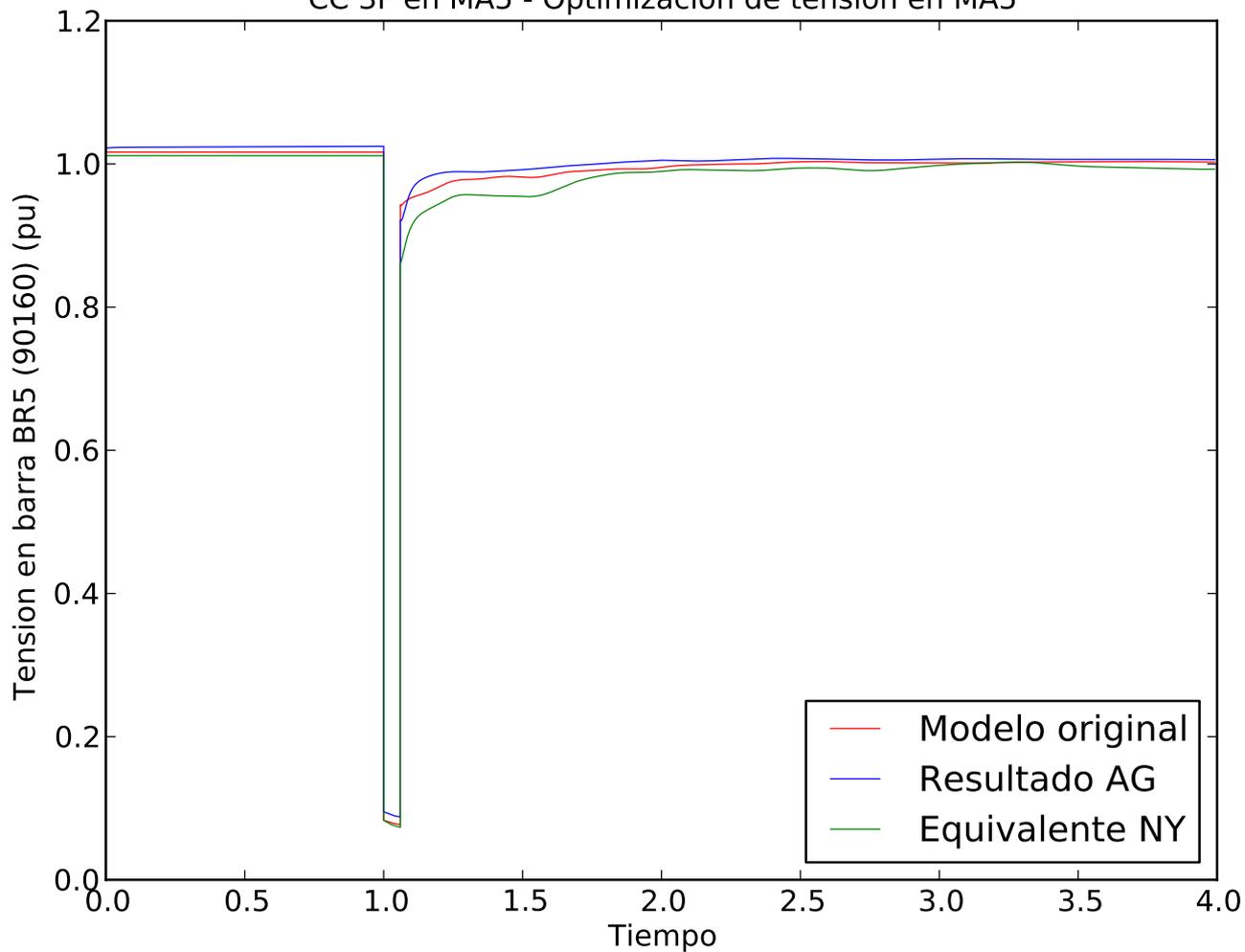
CC 3F en MA5 - Optimizacion de tension en MA5



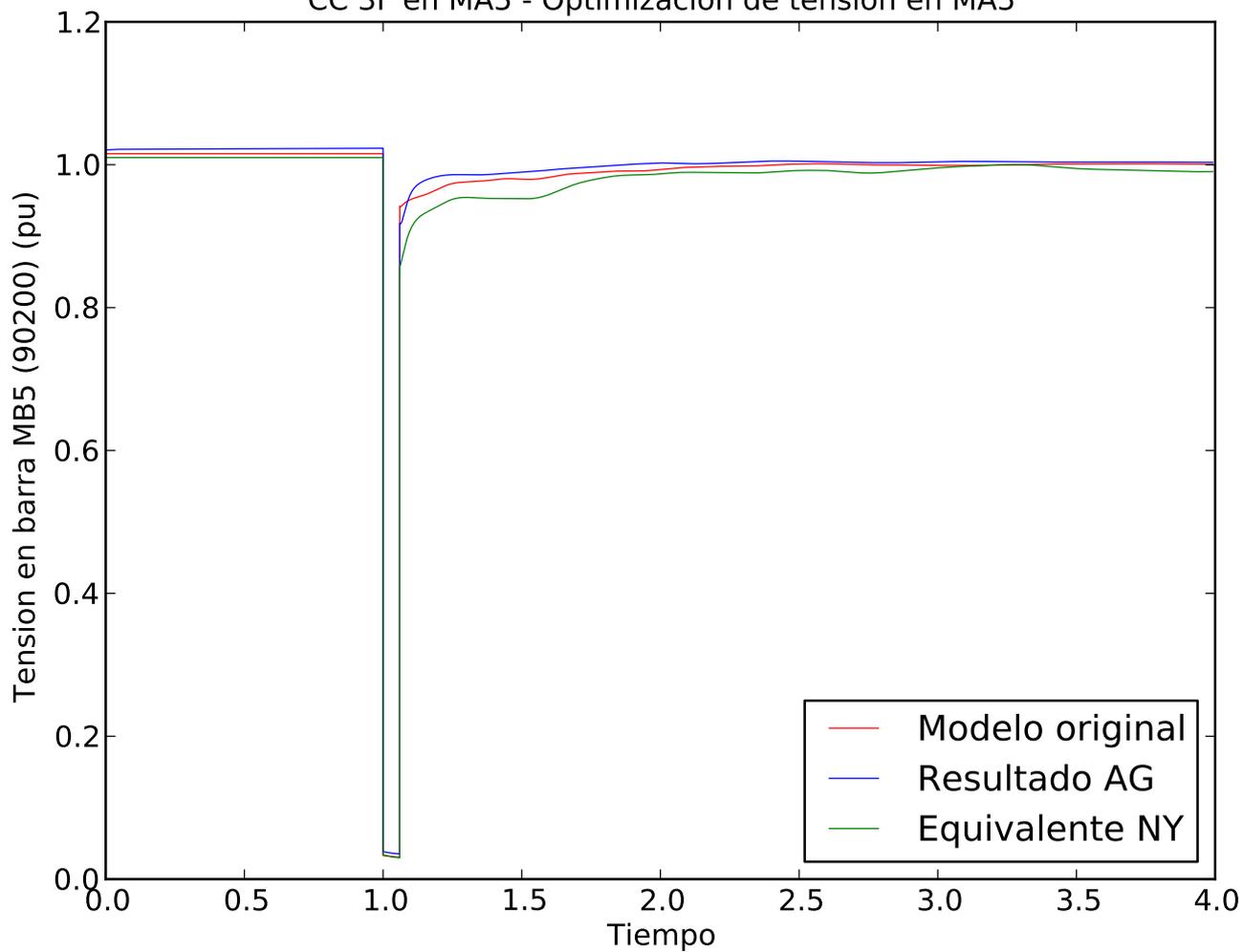
CC 3F en MA5 - Optimizacion de tension en MA5



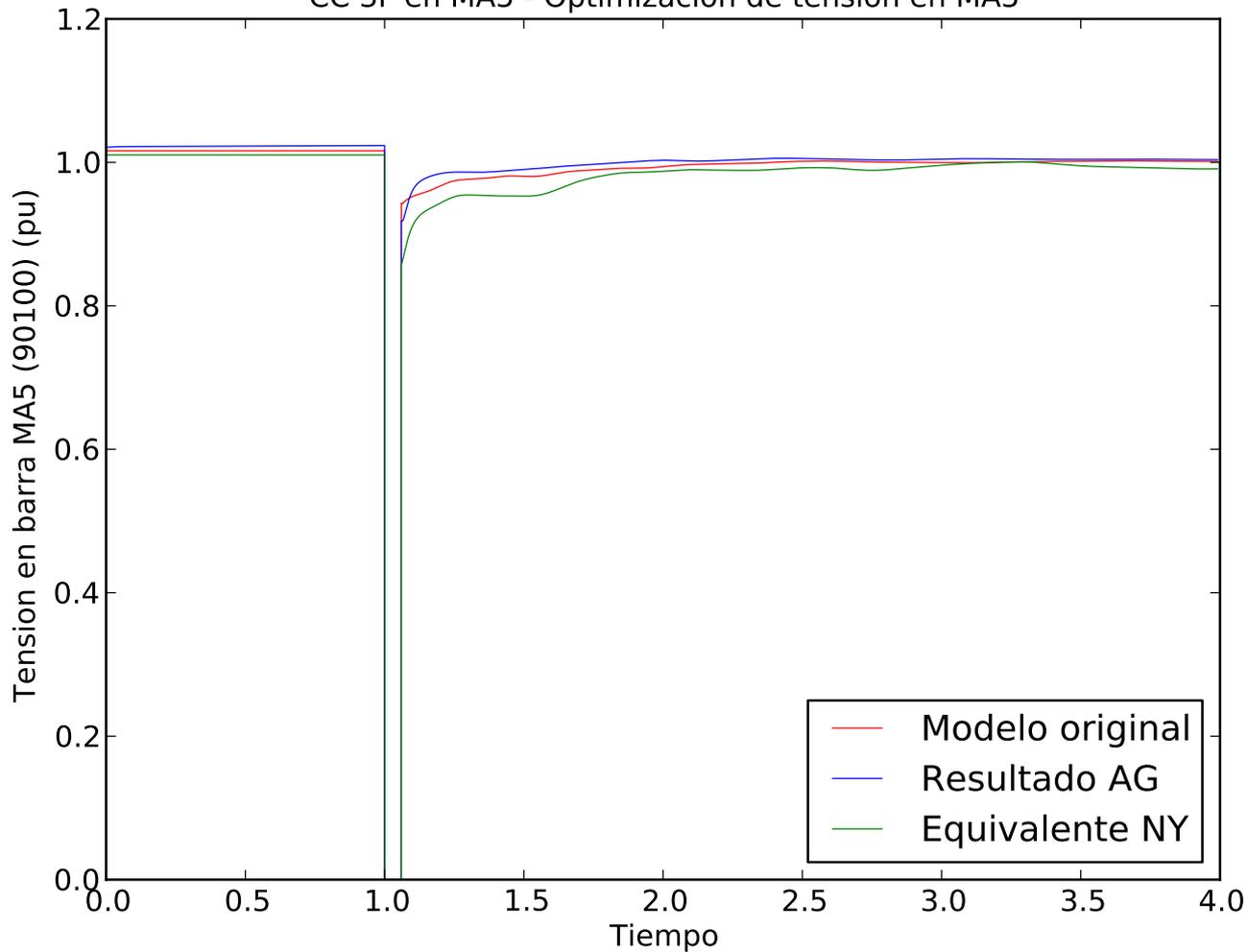
CC 3F en MA5 - Optimizacion de tension en MA5



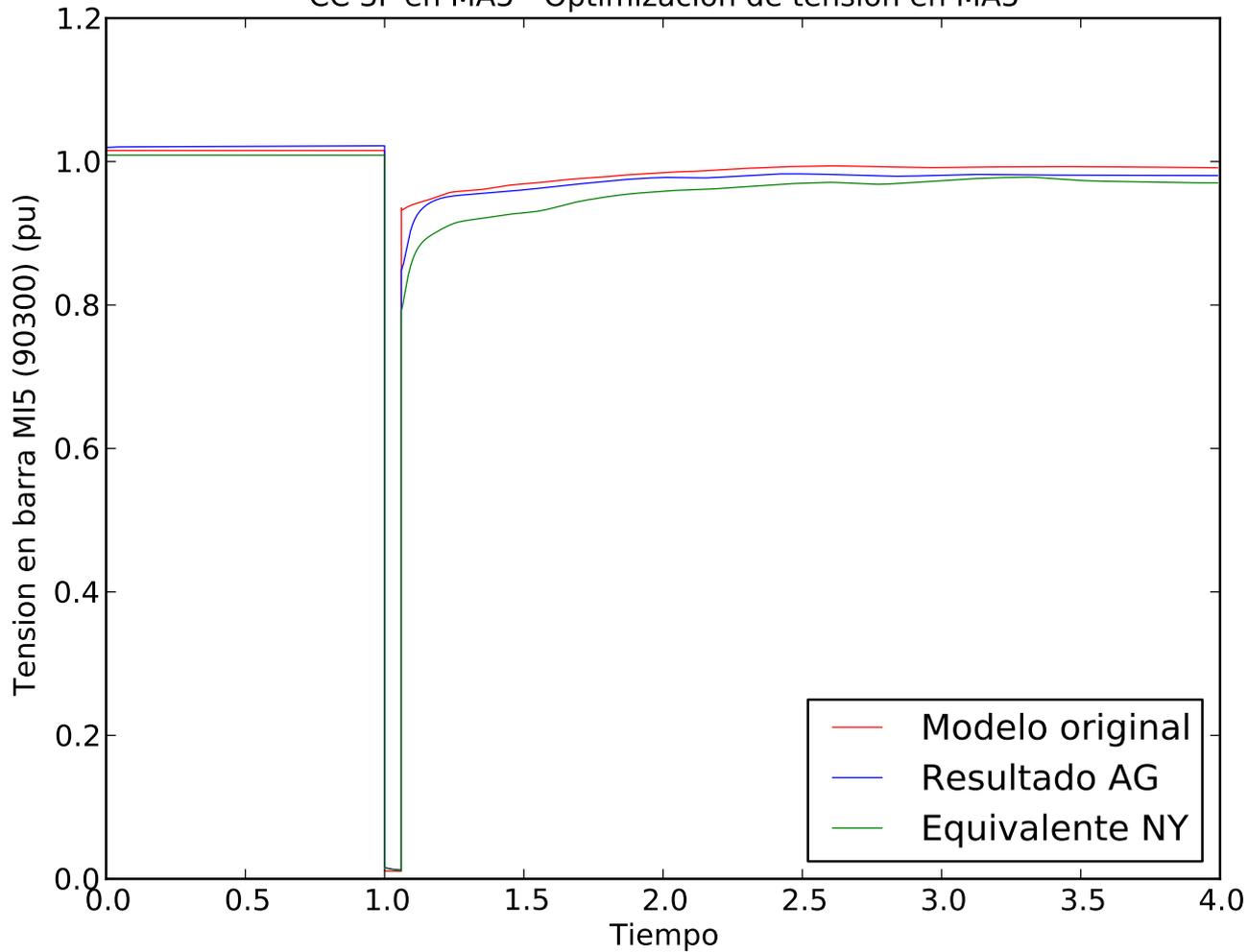
CC 3F en MA5 - Optimizacion de tension en MA5



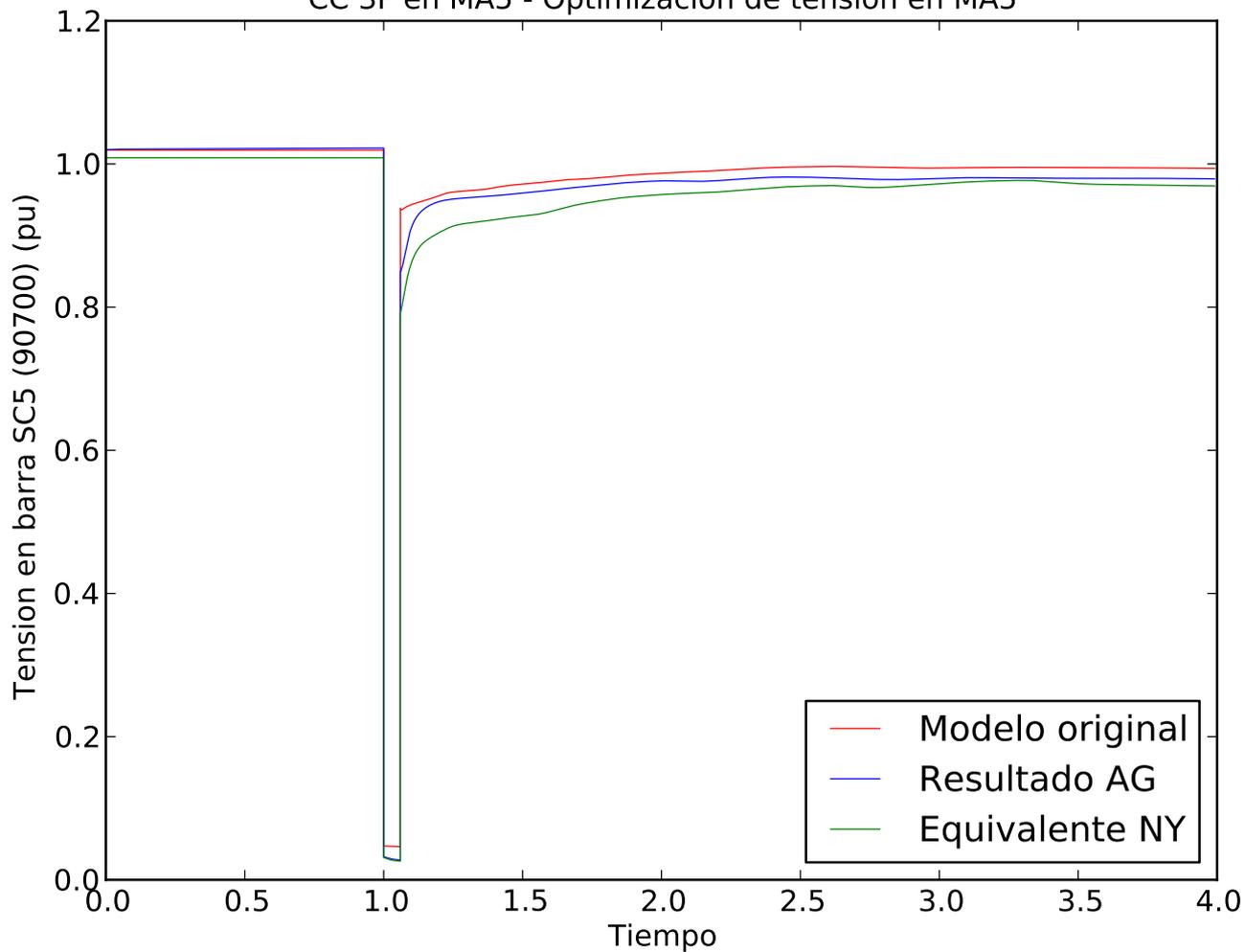
CC 3F en MA5 - Optimizacion de tension en MA5



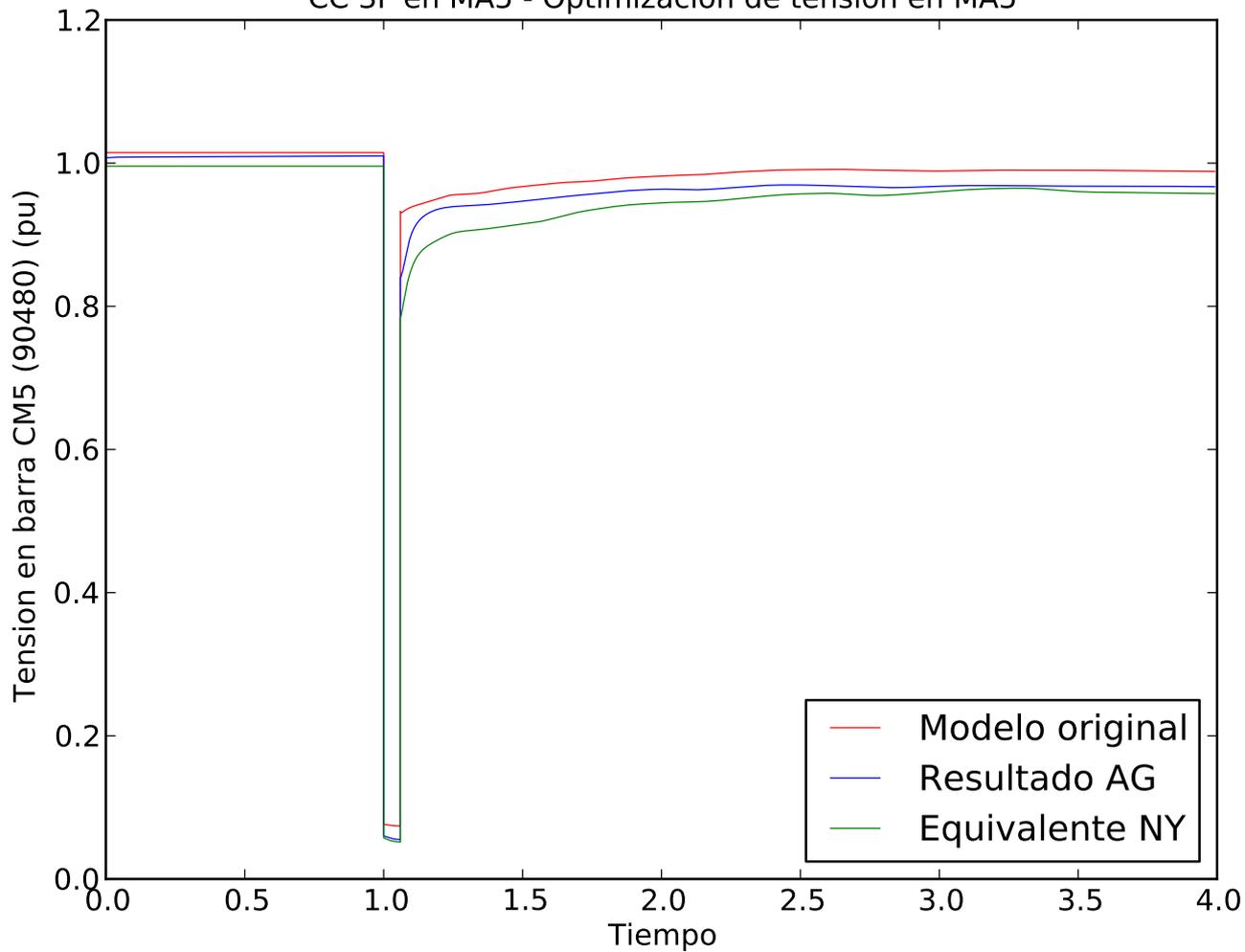
CC 3F en MA5 - Optimizacion de tension en MA5



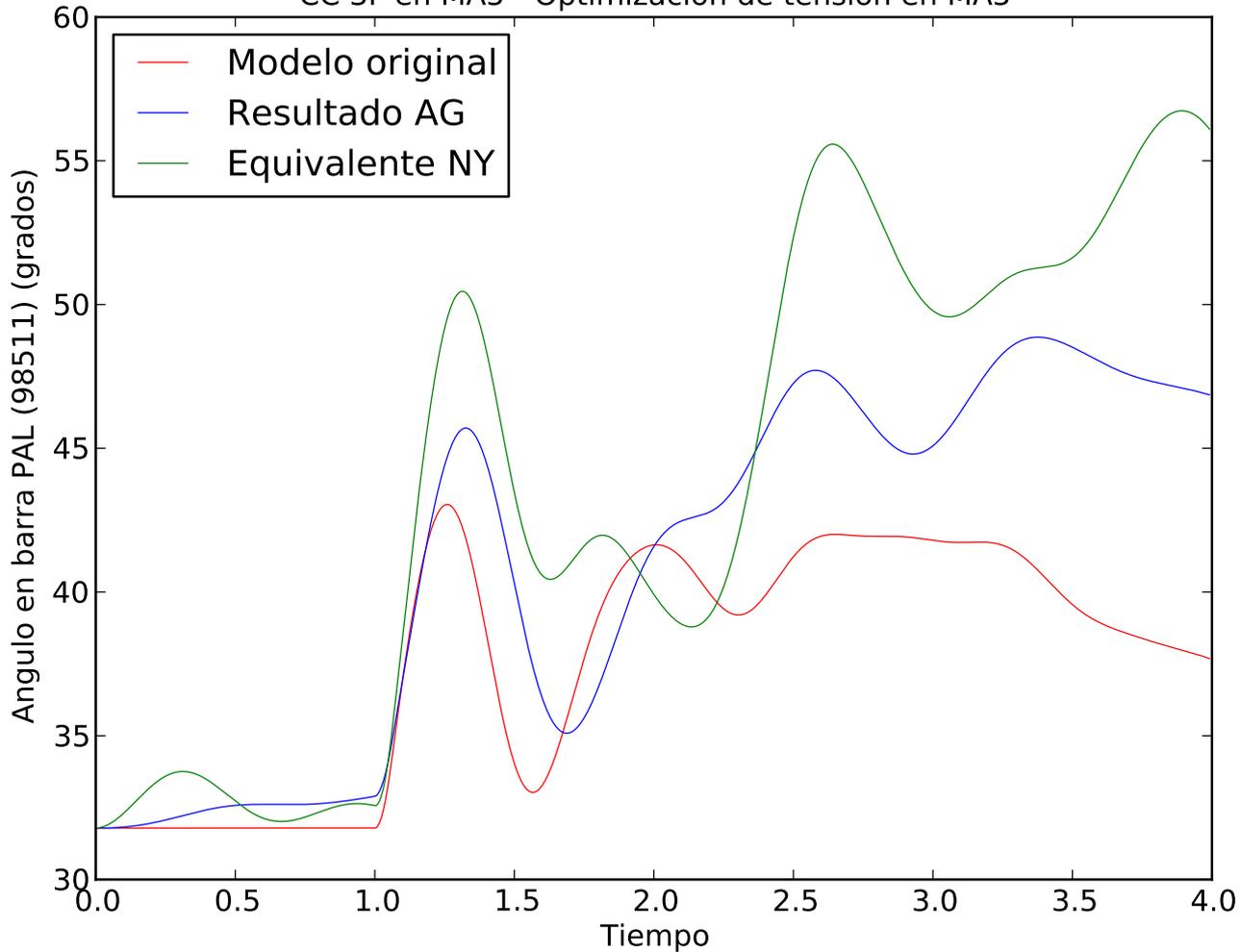
CC 3F en MA5 - Optimizacion de tension en MA5



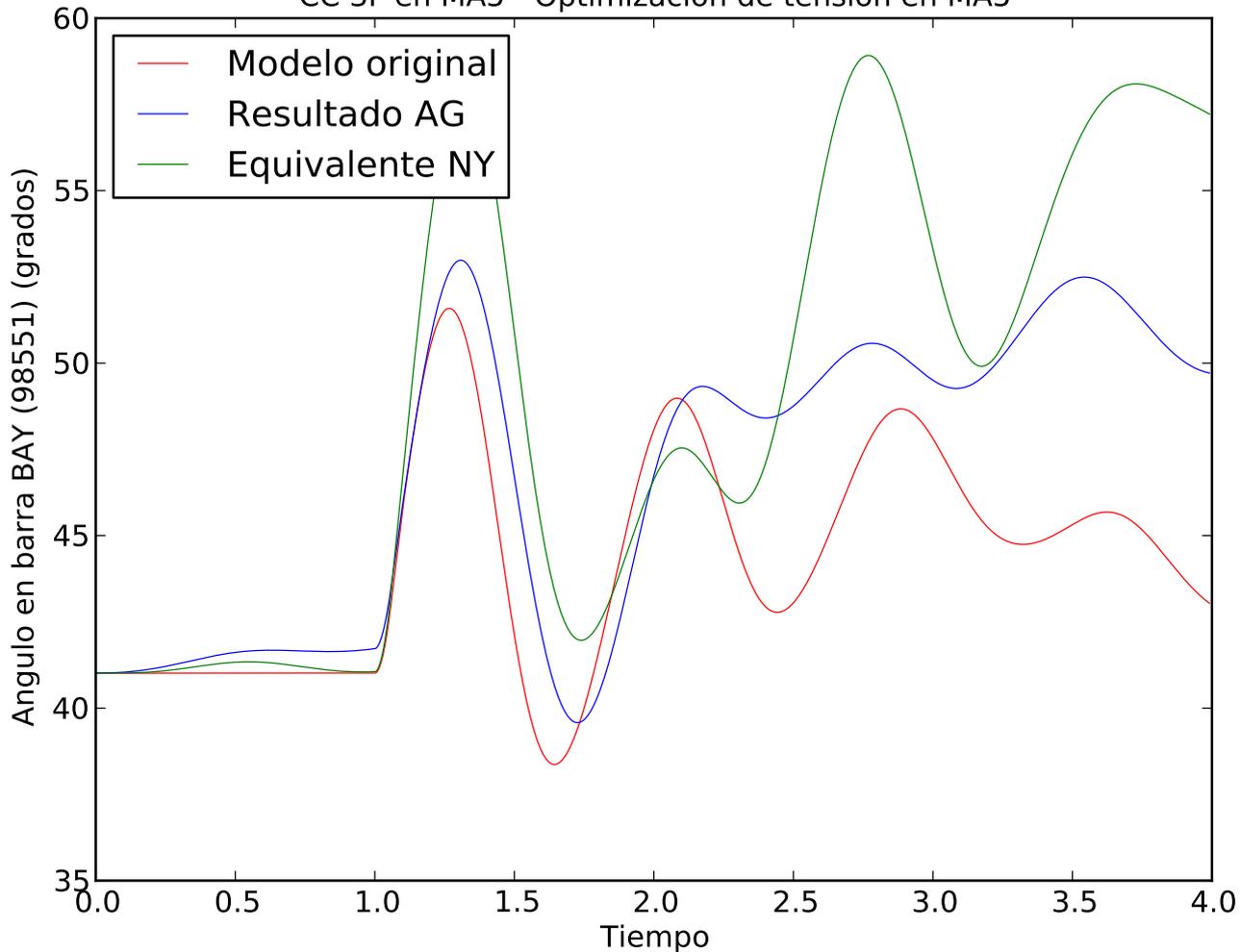
CC 3F en MA5 - Optimizacion de tension en MA5



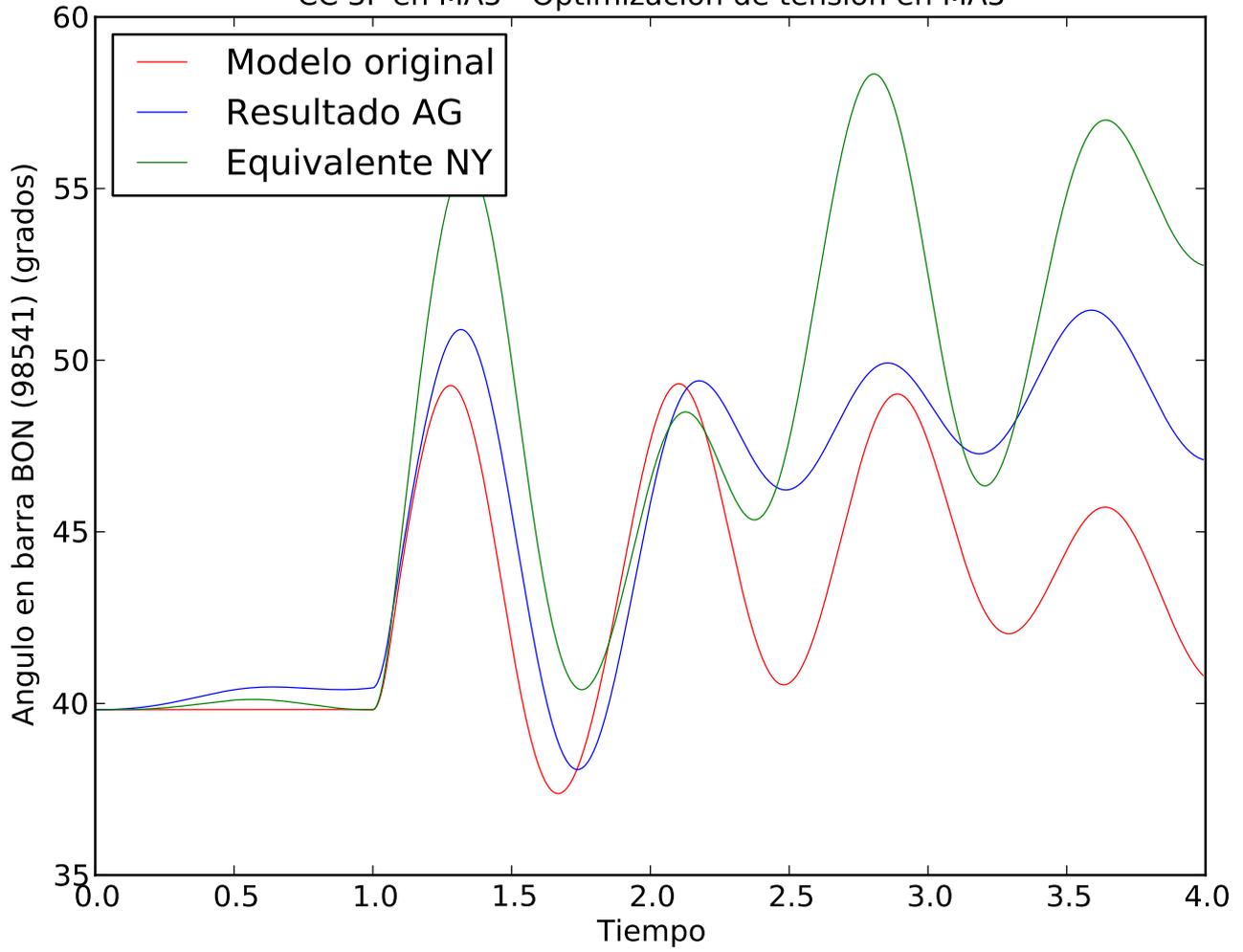
CC 3F en MA5 - Optimizacion de tension en MA5



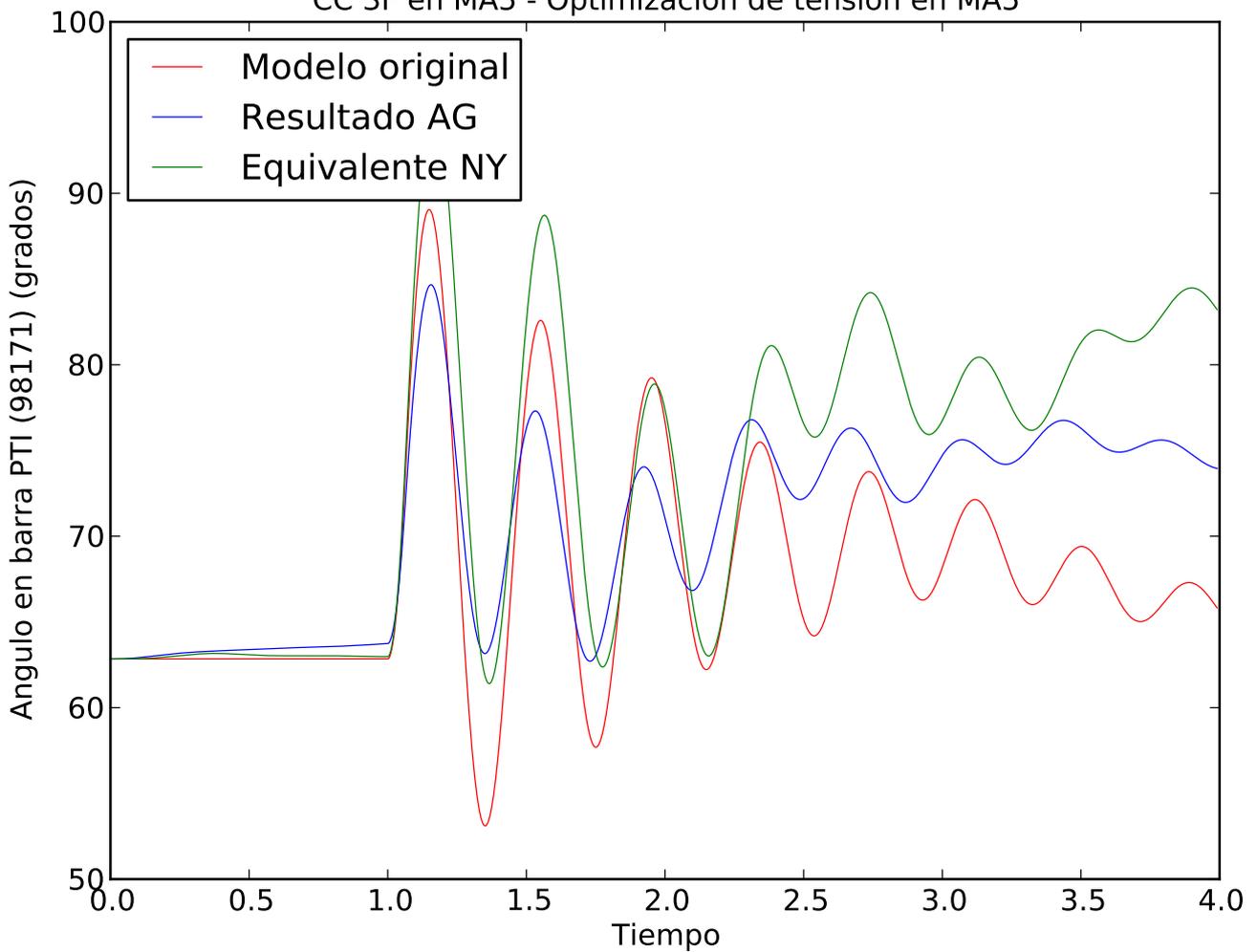
CC 3F en MA5 - Optimizacion de tension en MA5



CC 3F en MA5 - Optimizacion de tension en MA5

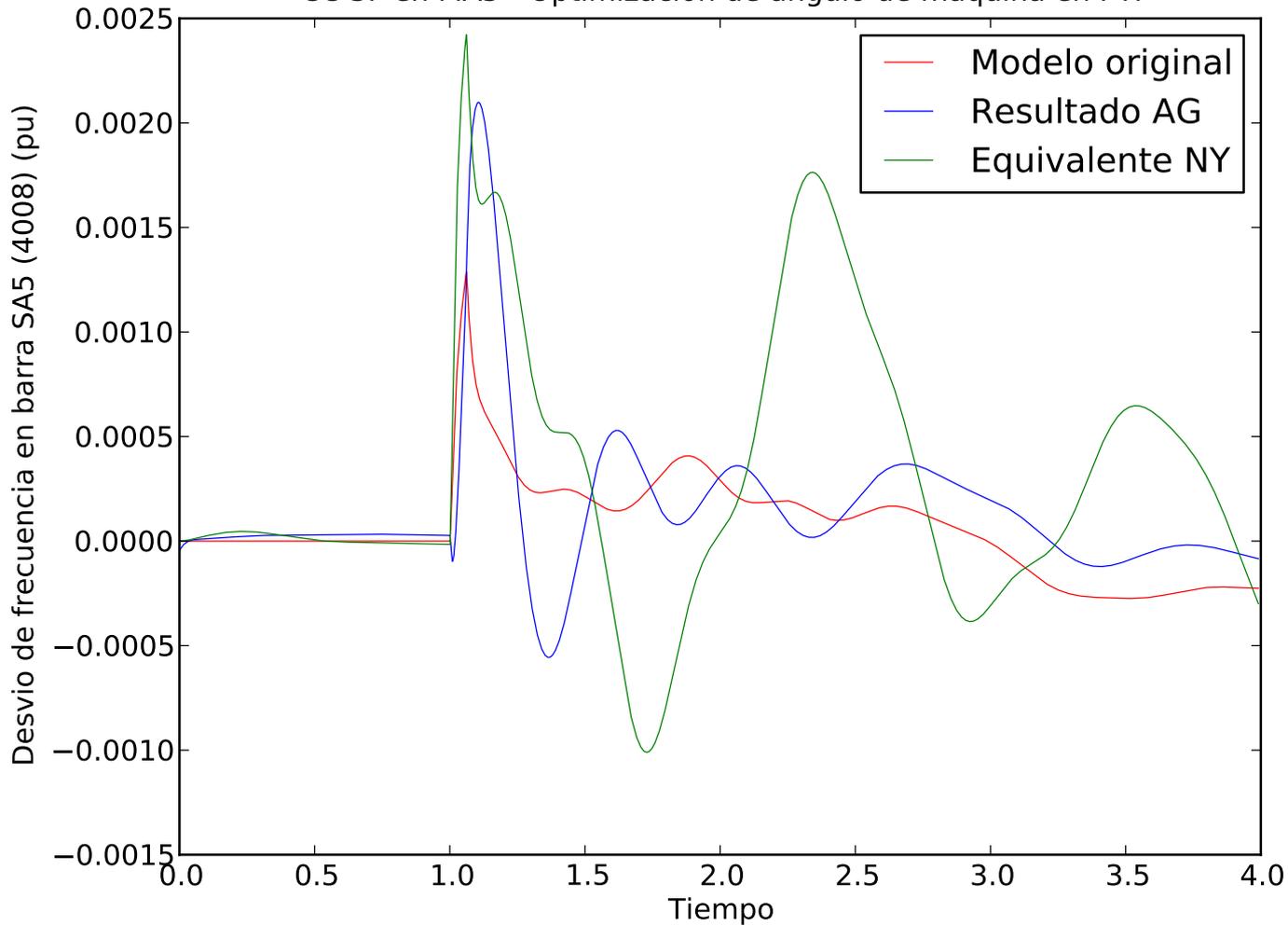


CC 3F en MA5 - Optimizacion de tension en MA5

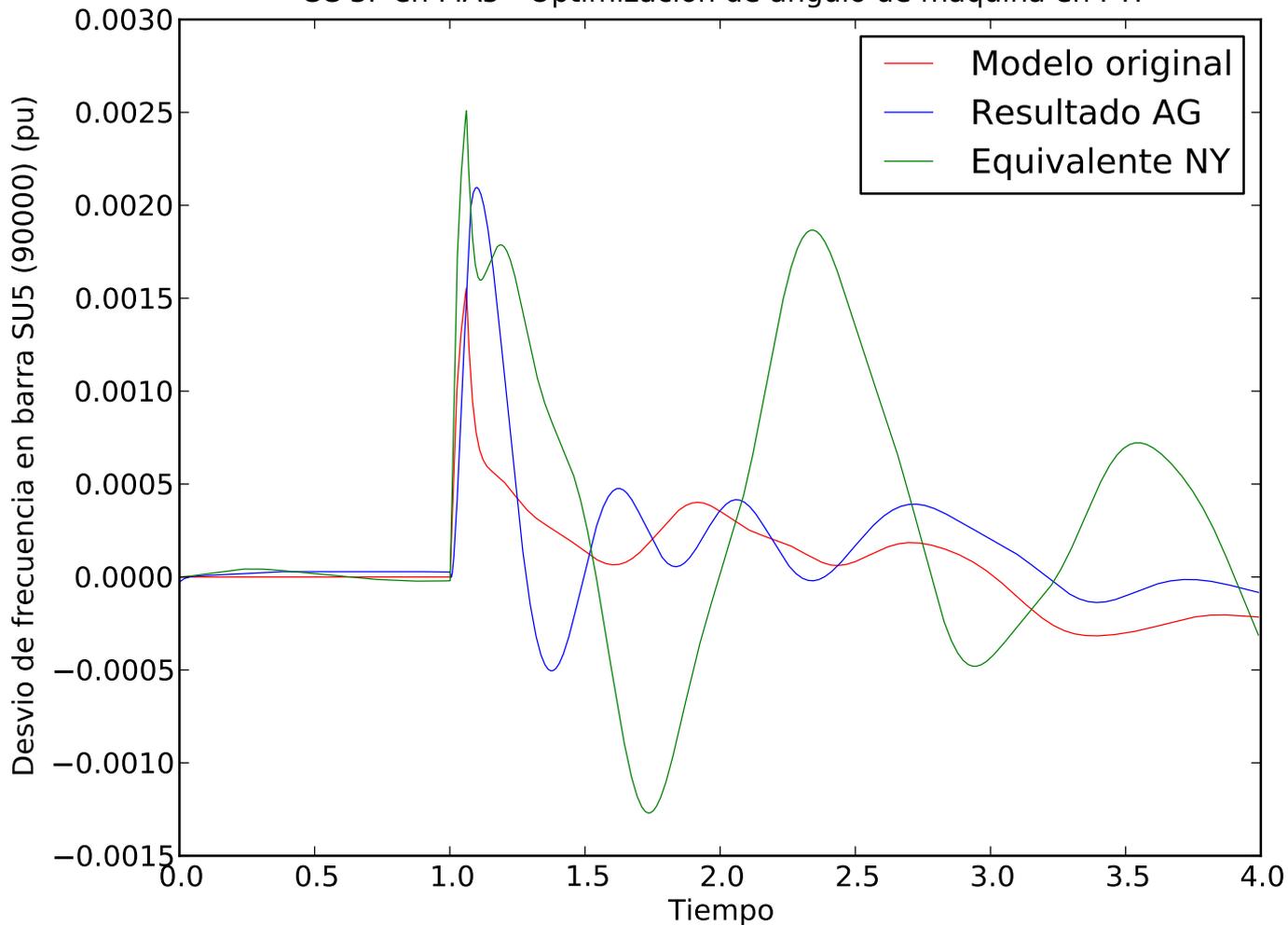


**C.9. Simulación 6: Cortocircuito 3F en MA5 con salida de línea en equivalente NY 3, optimizando ángulo en PTI en 4 segundos. Comportamiento en 4 segundos**

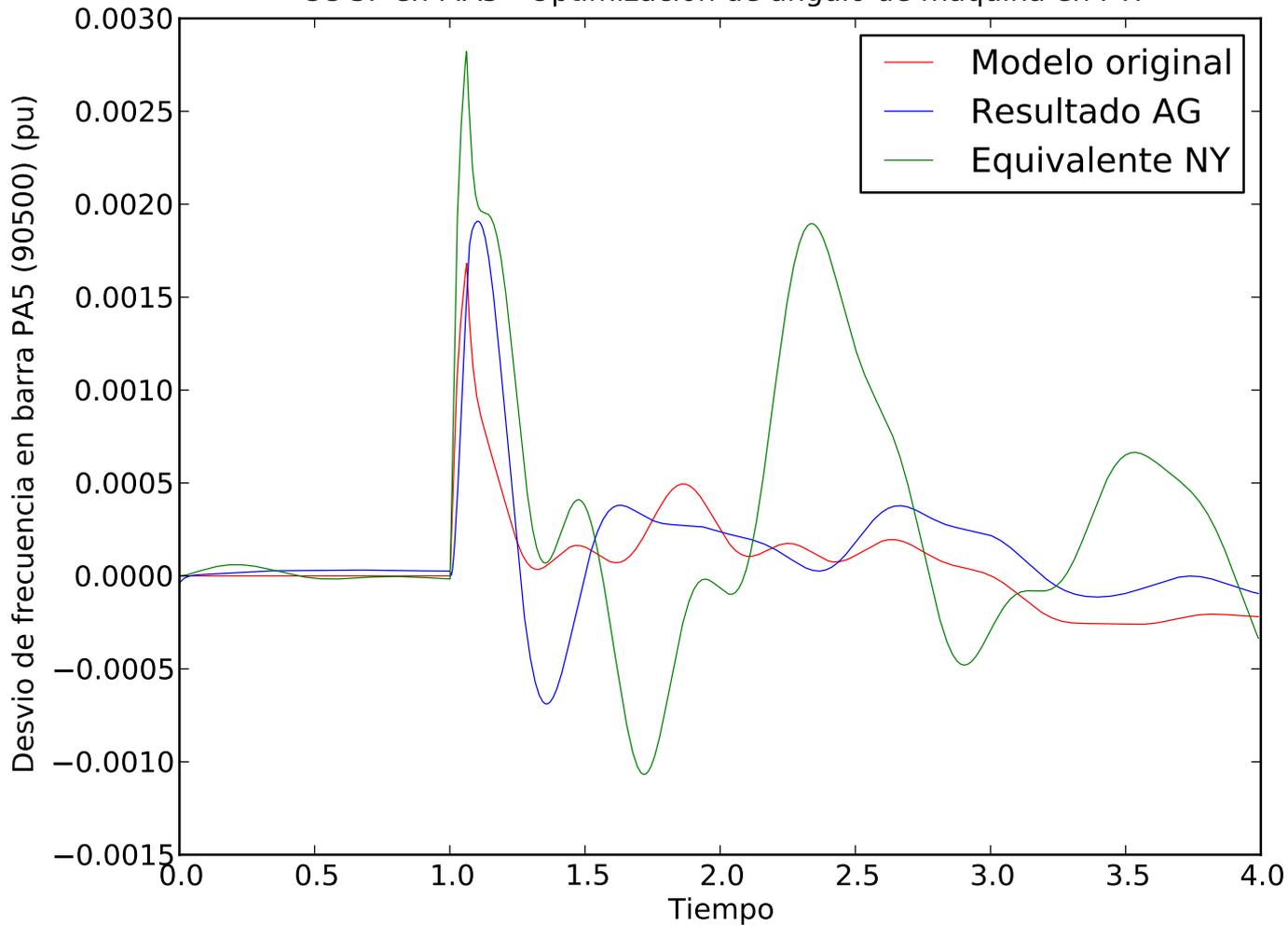
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



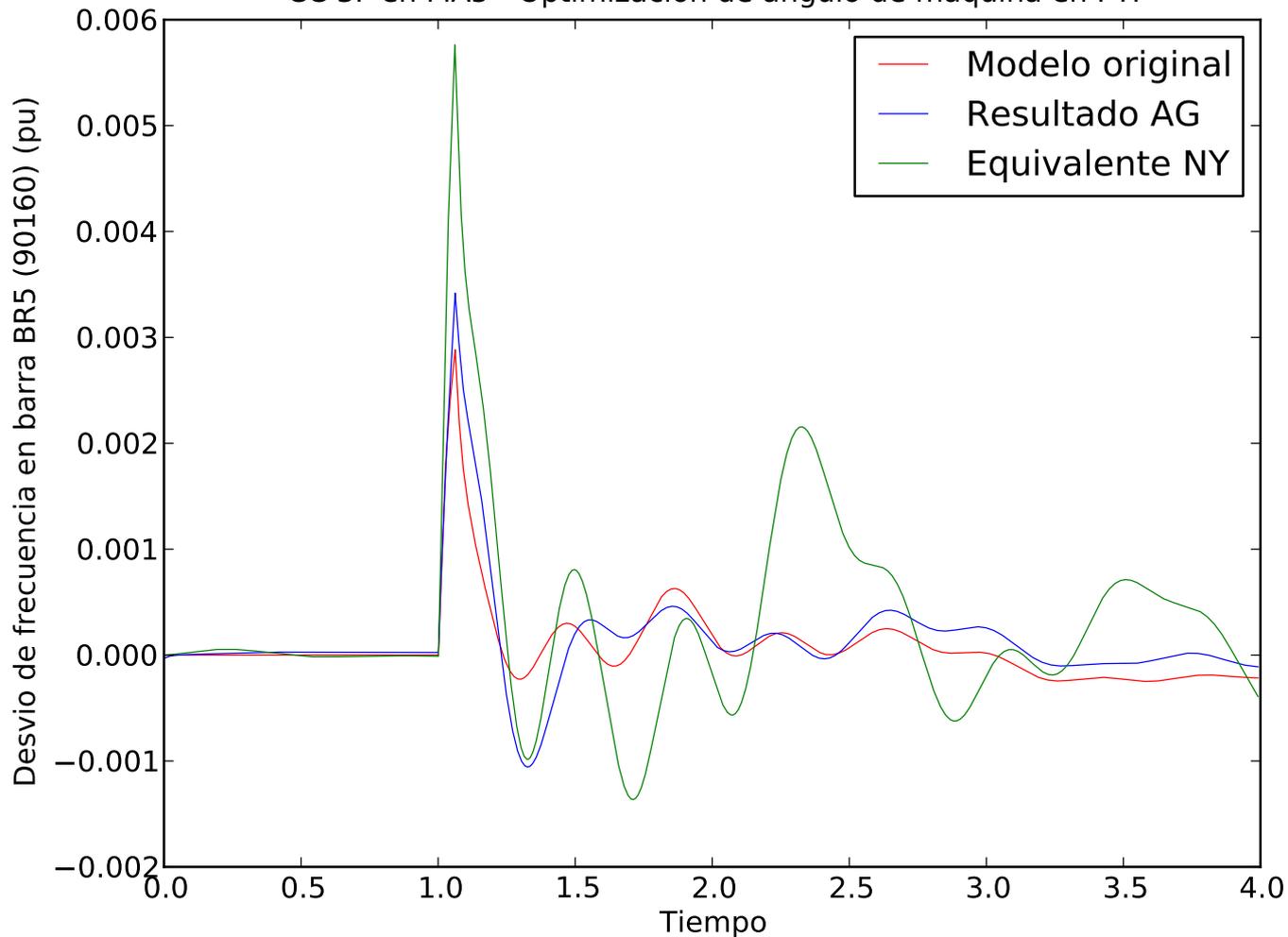
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



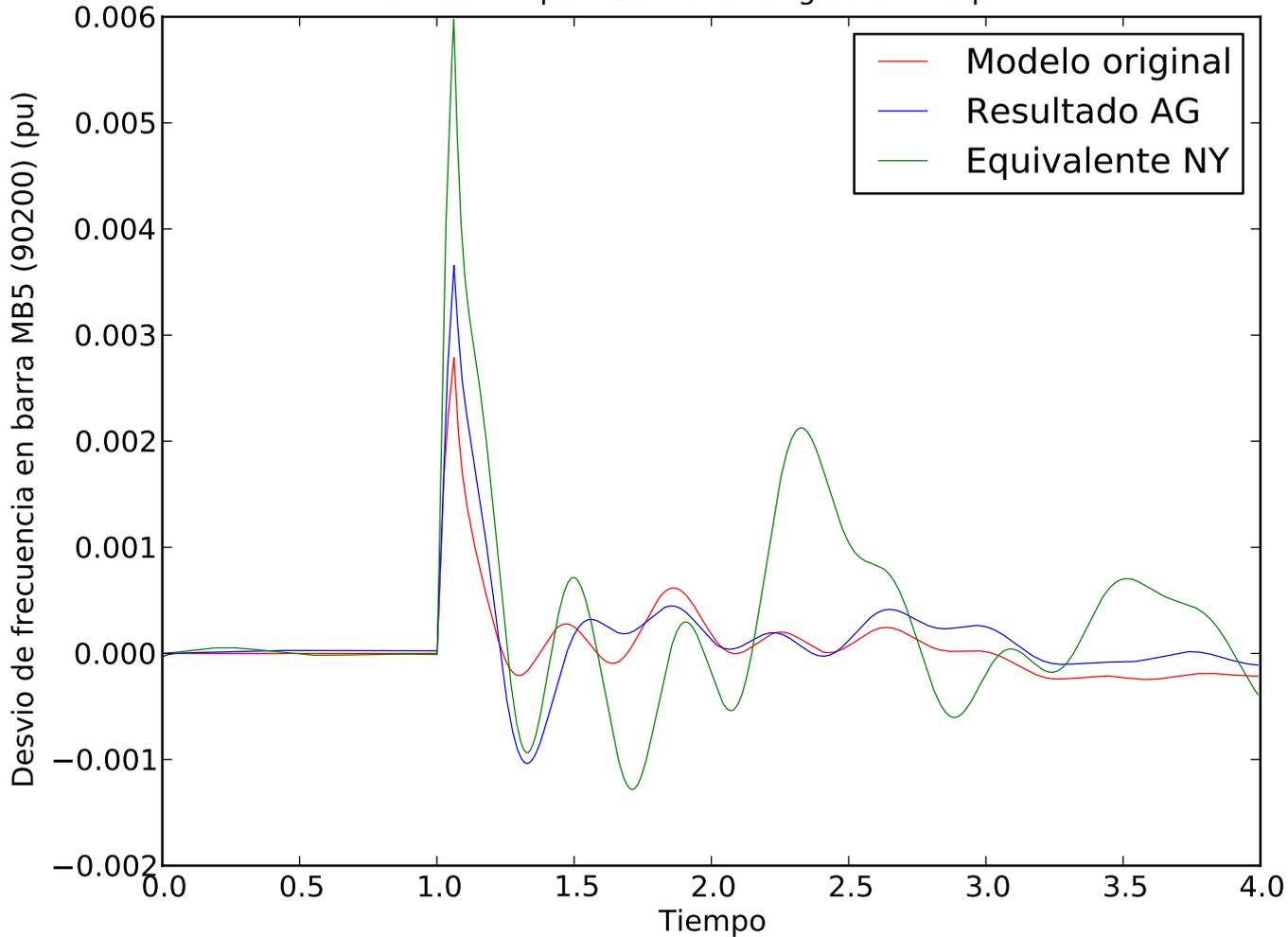
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



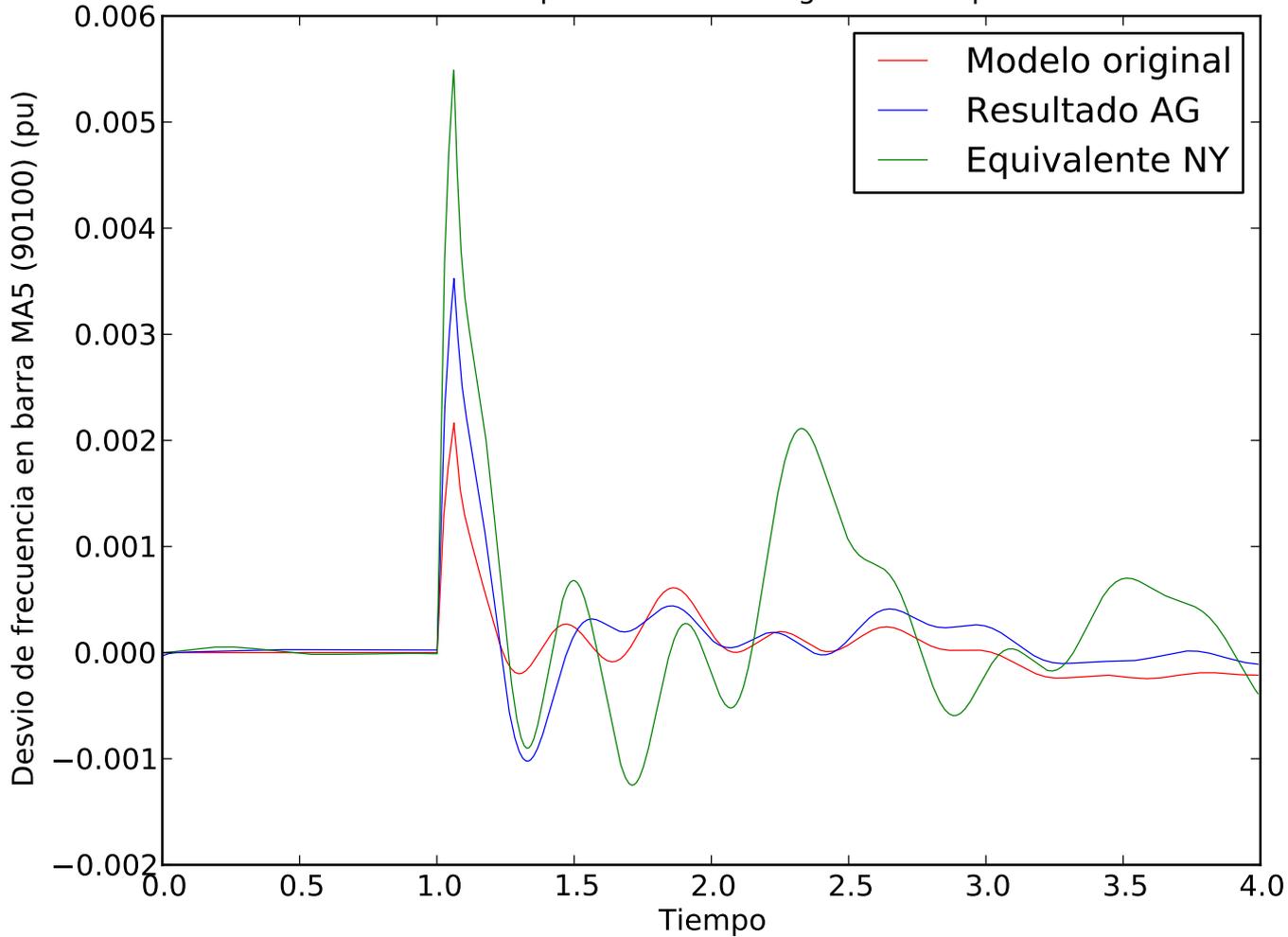
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



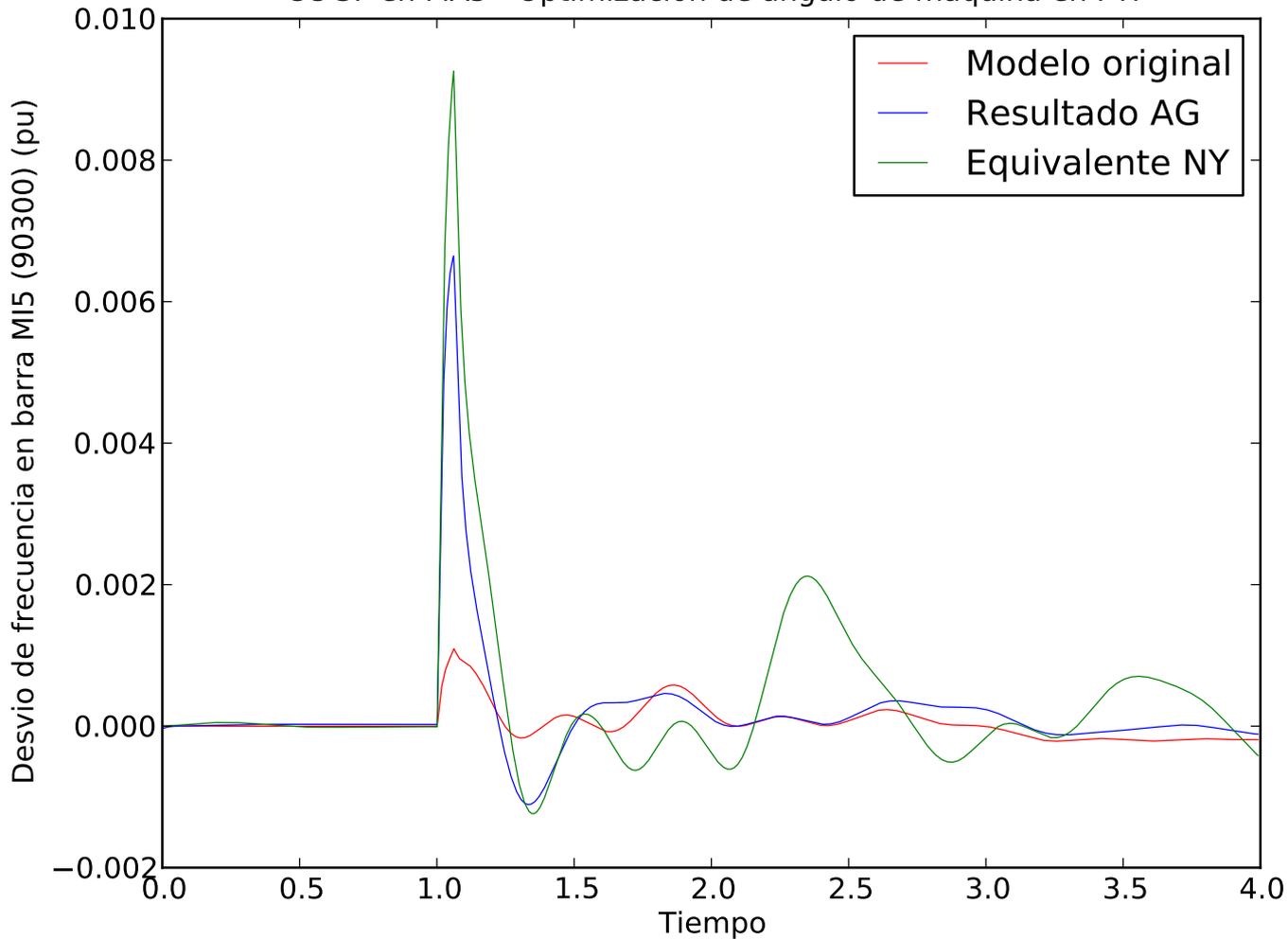
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



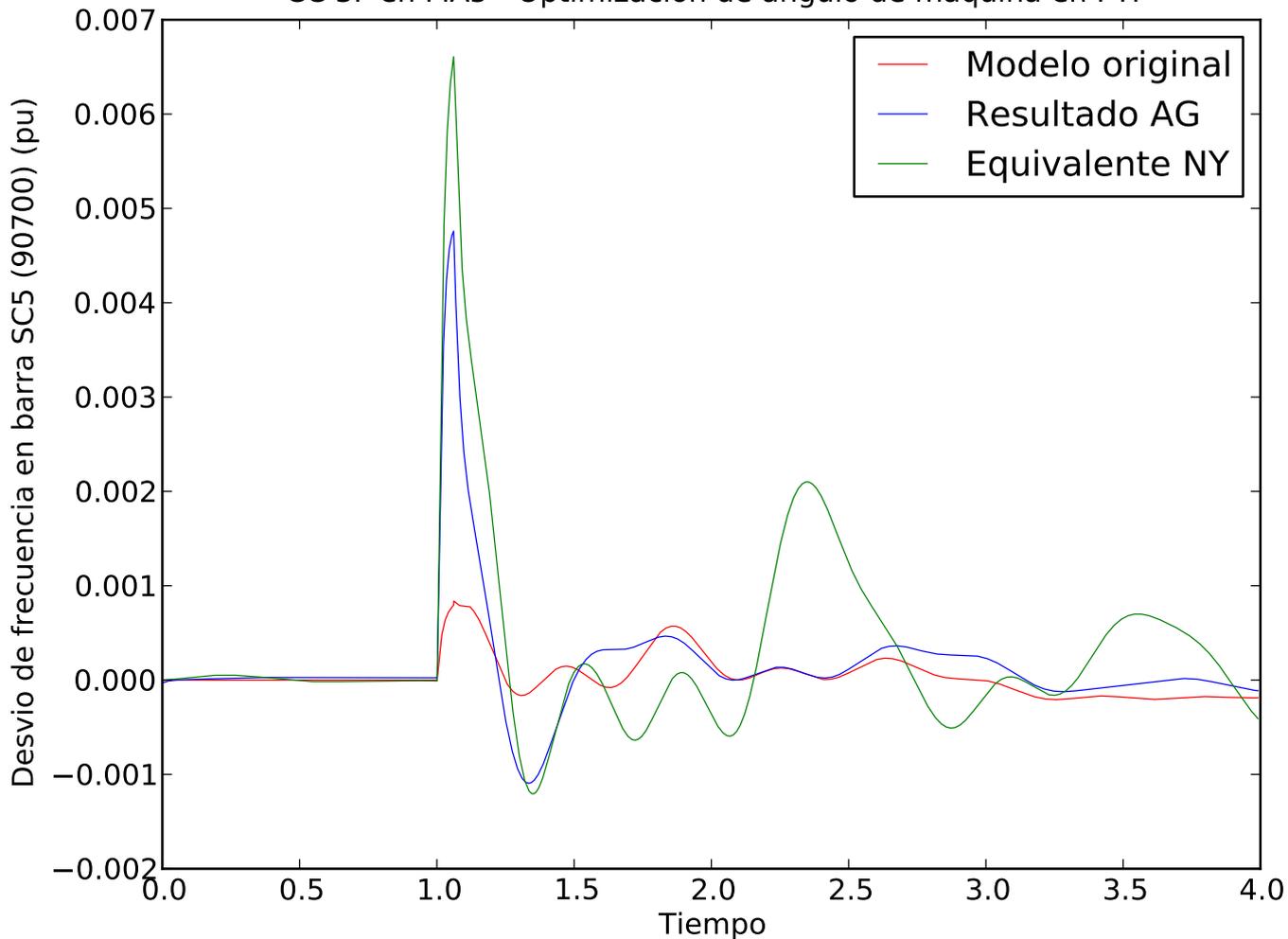
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



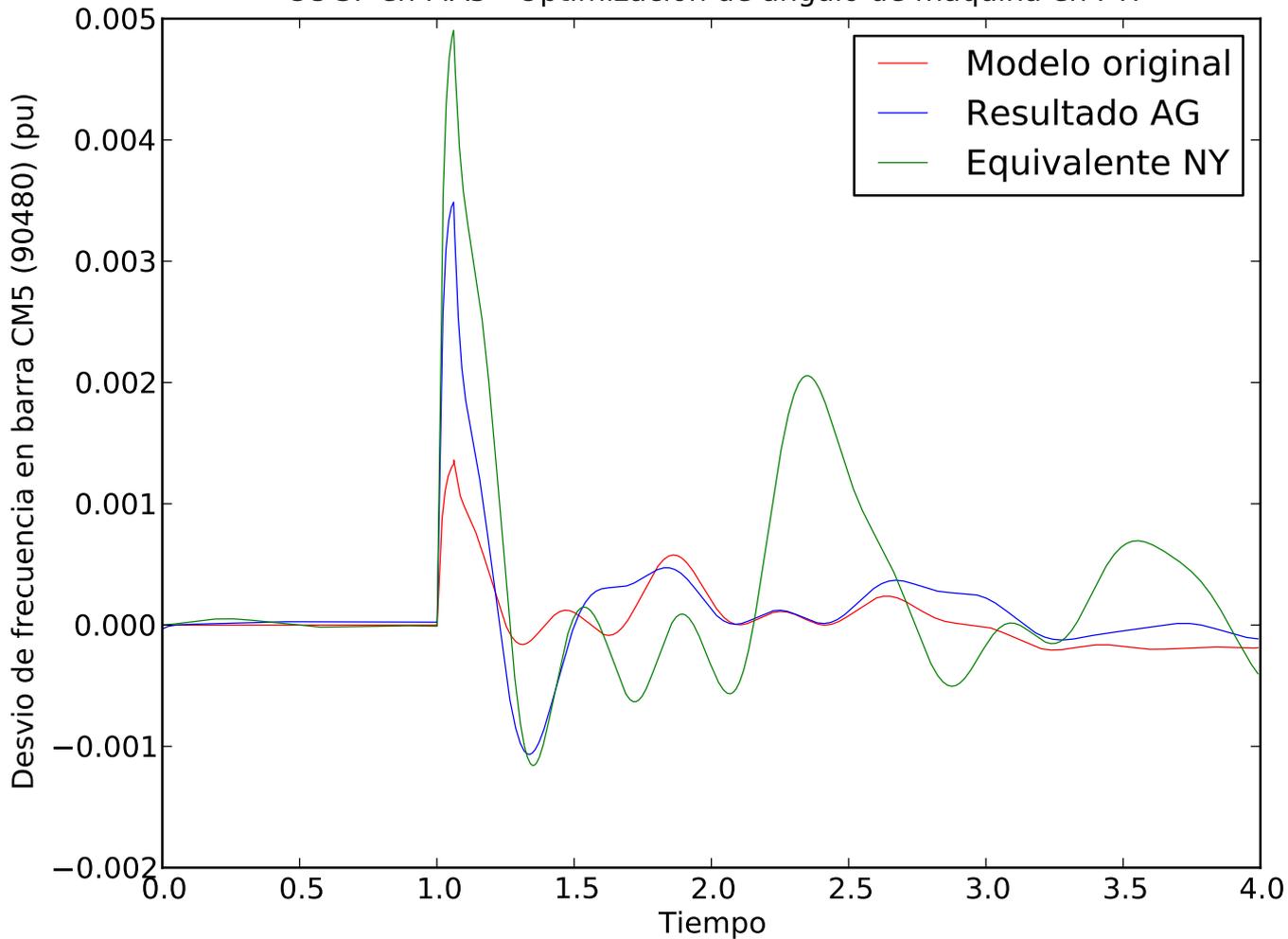
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



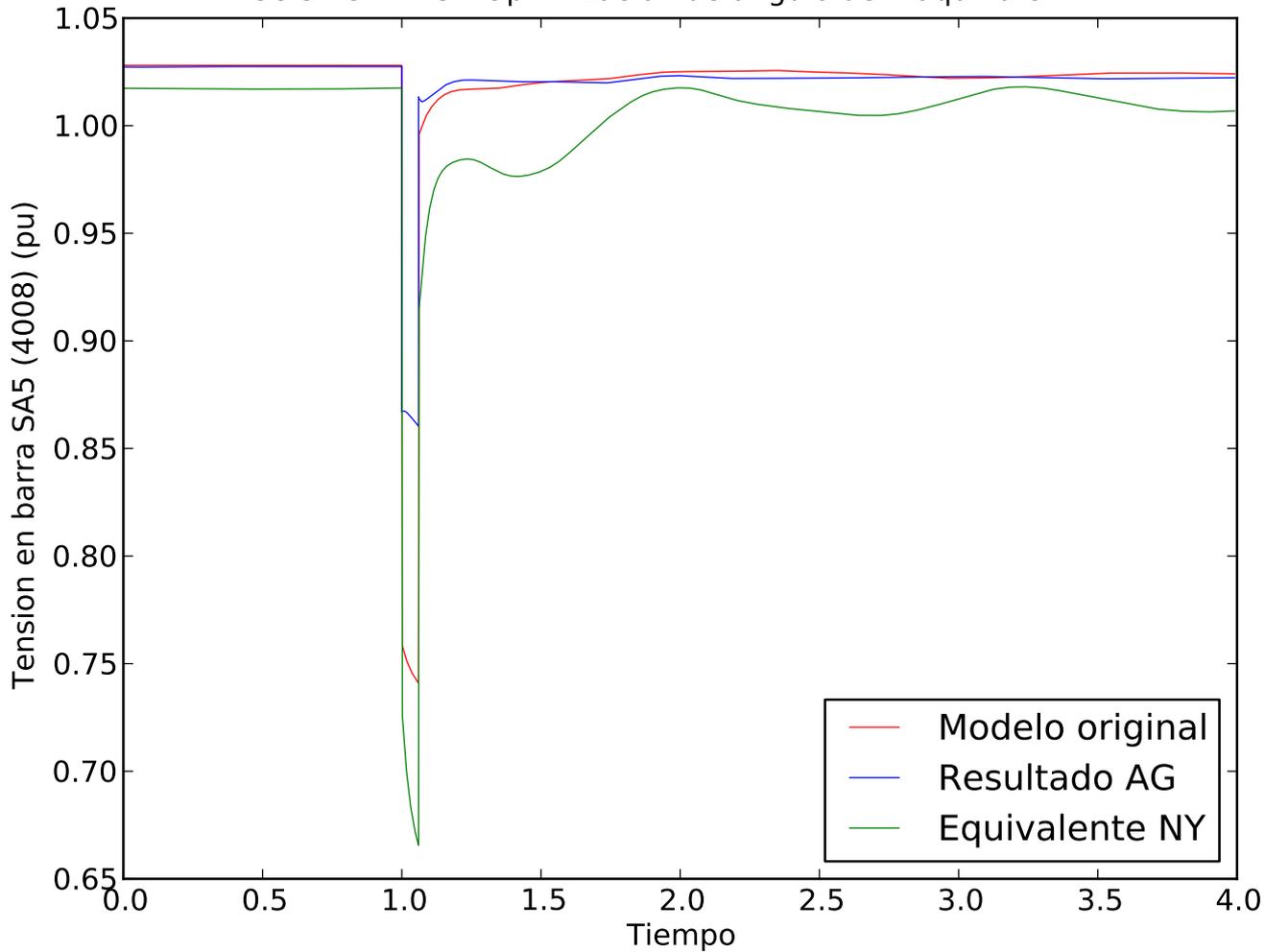
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



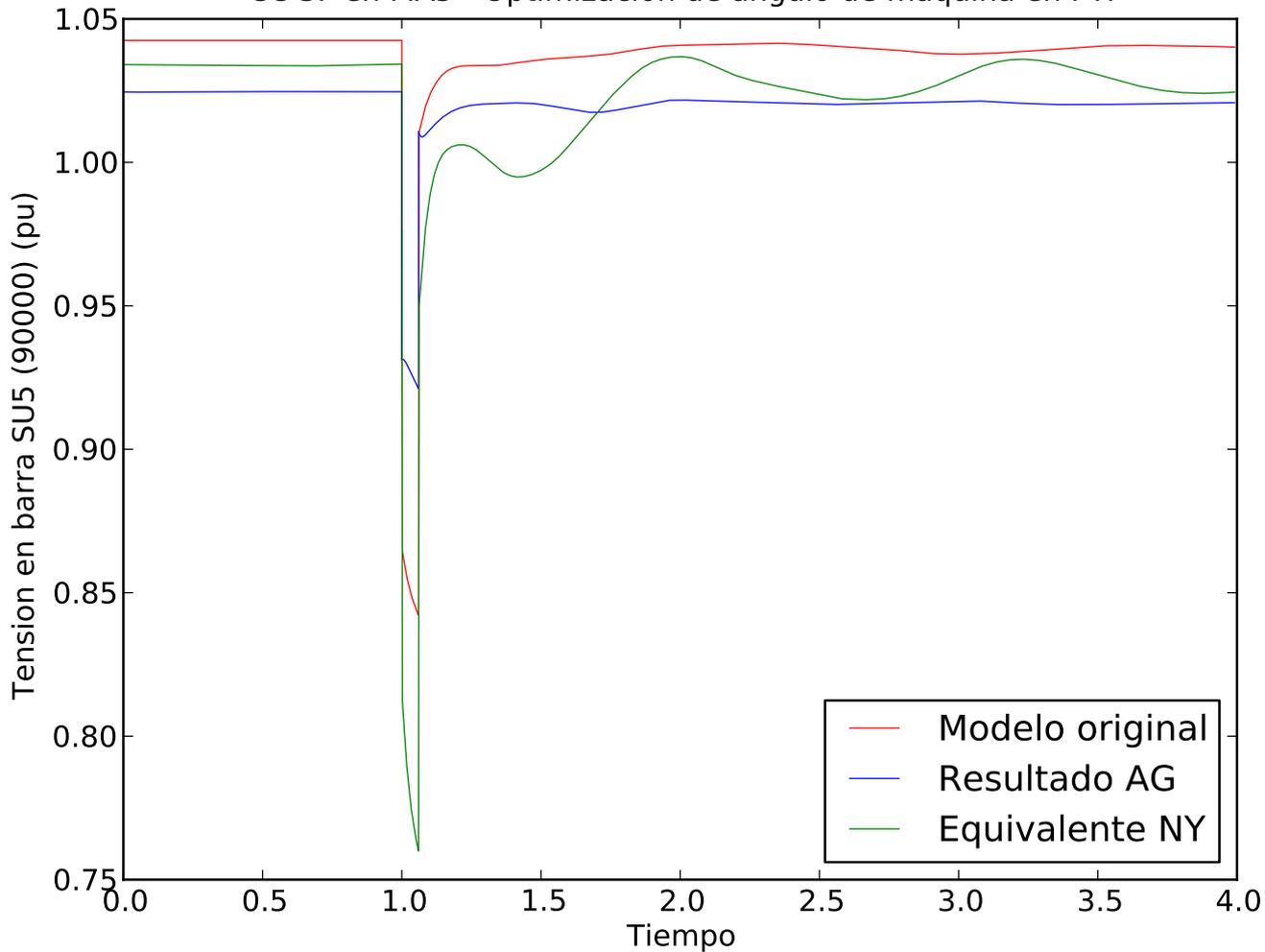
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



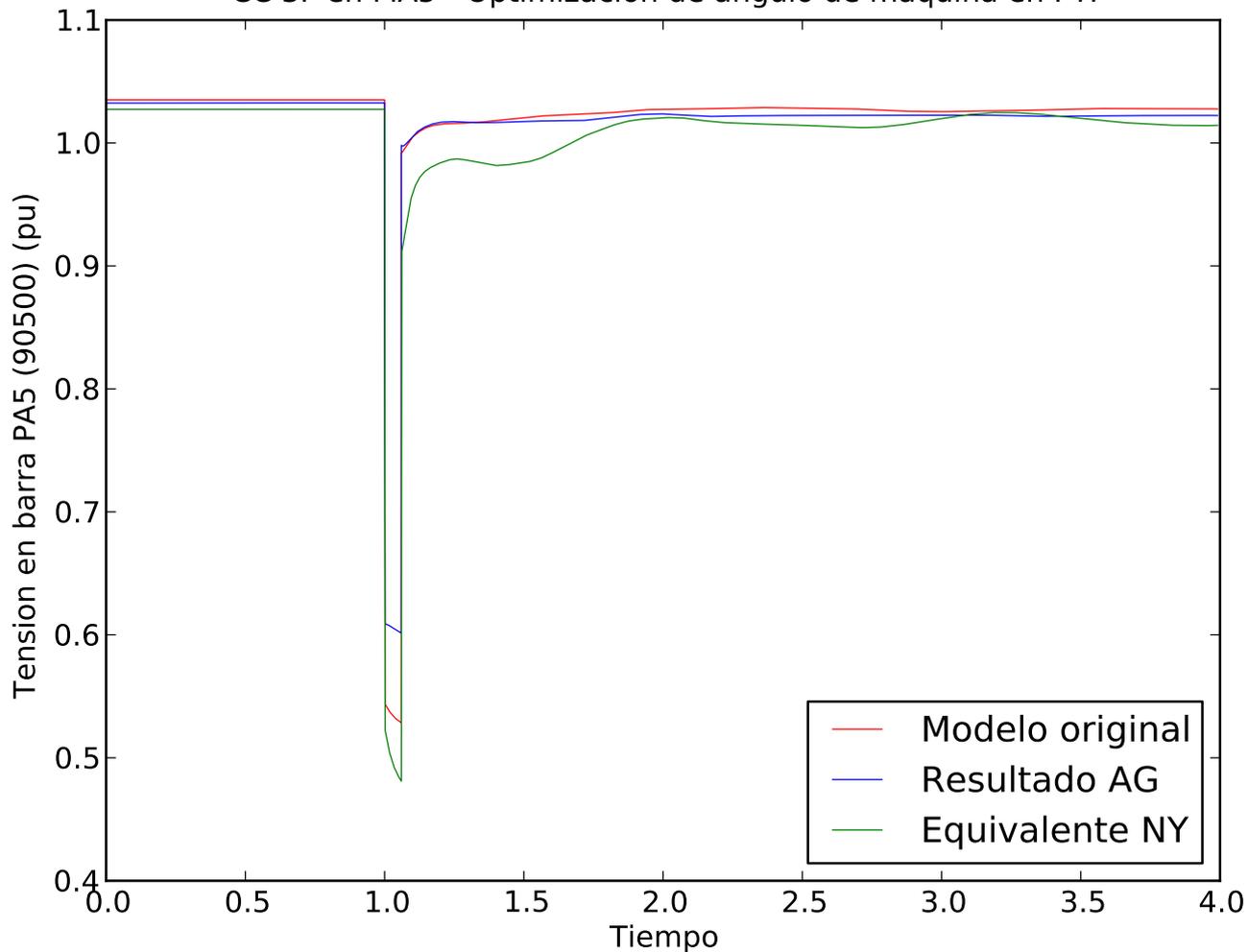
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



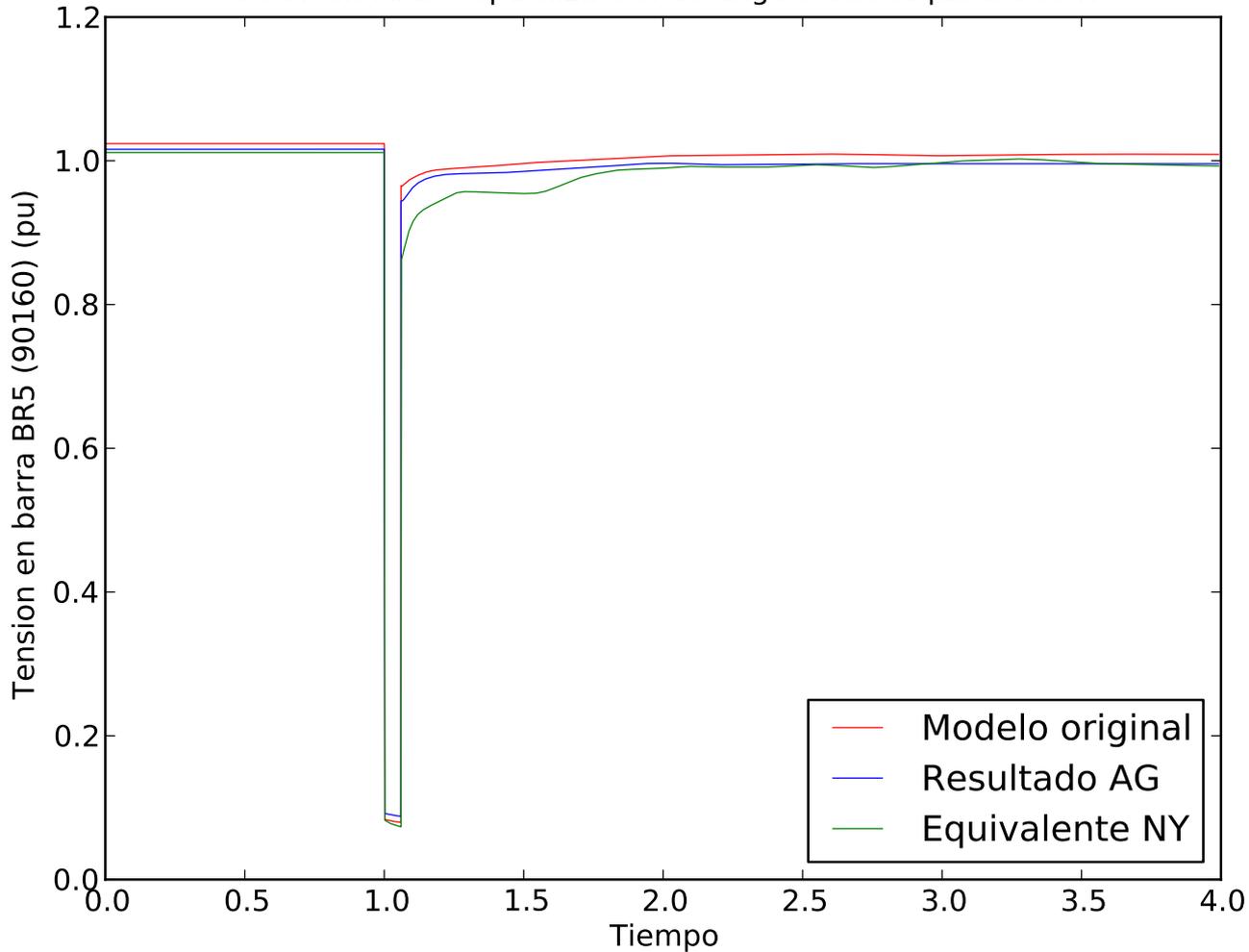
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



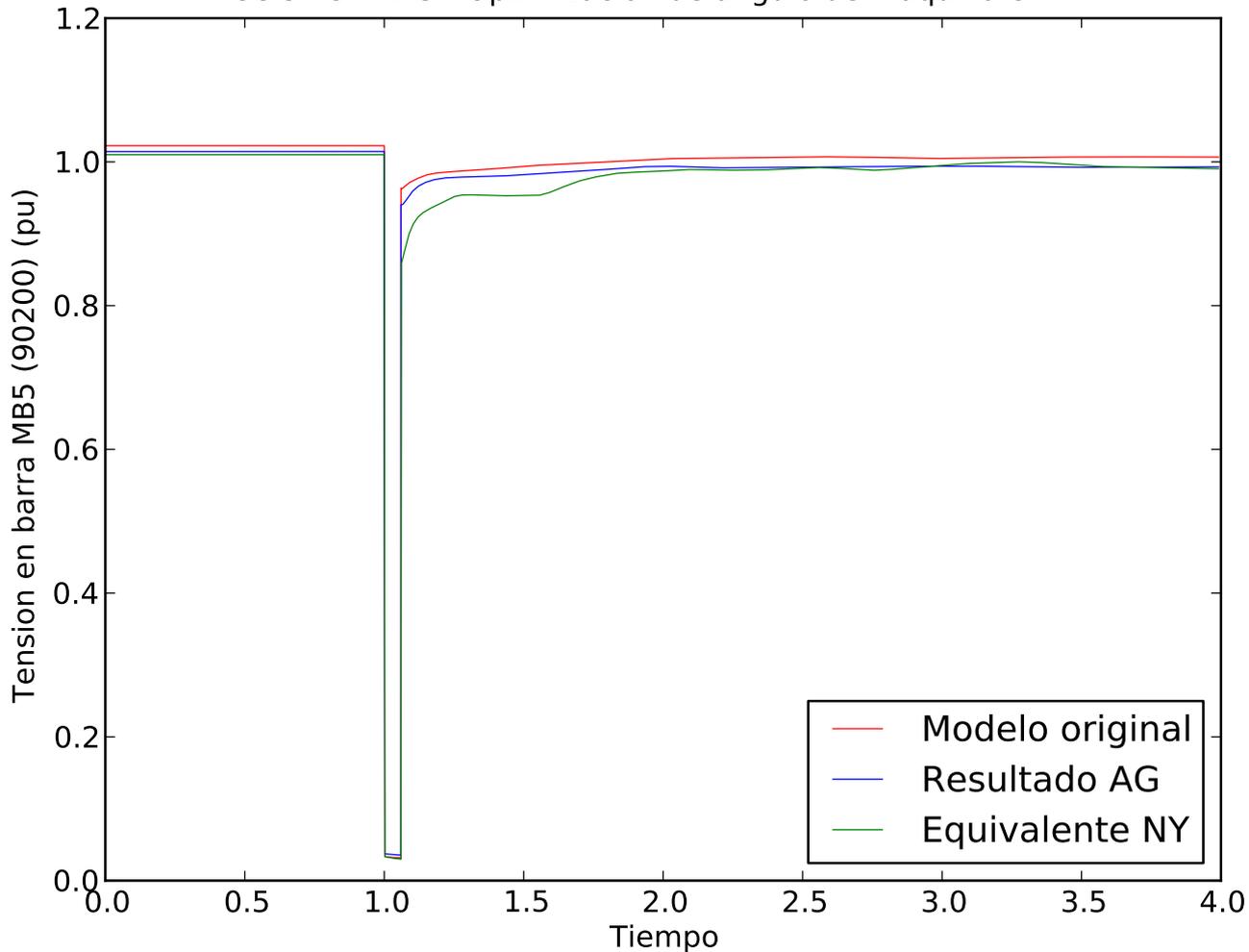
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



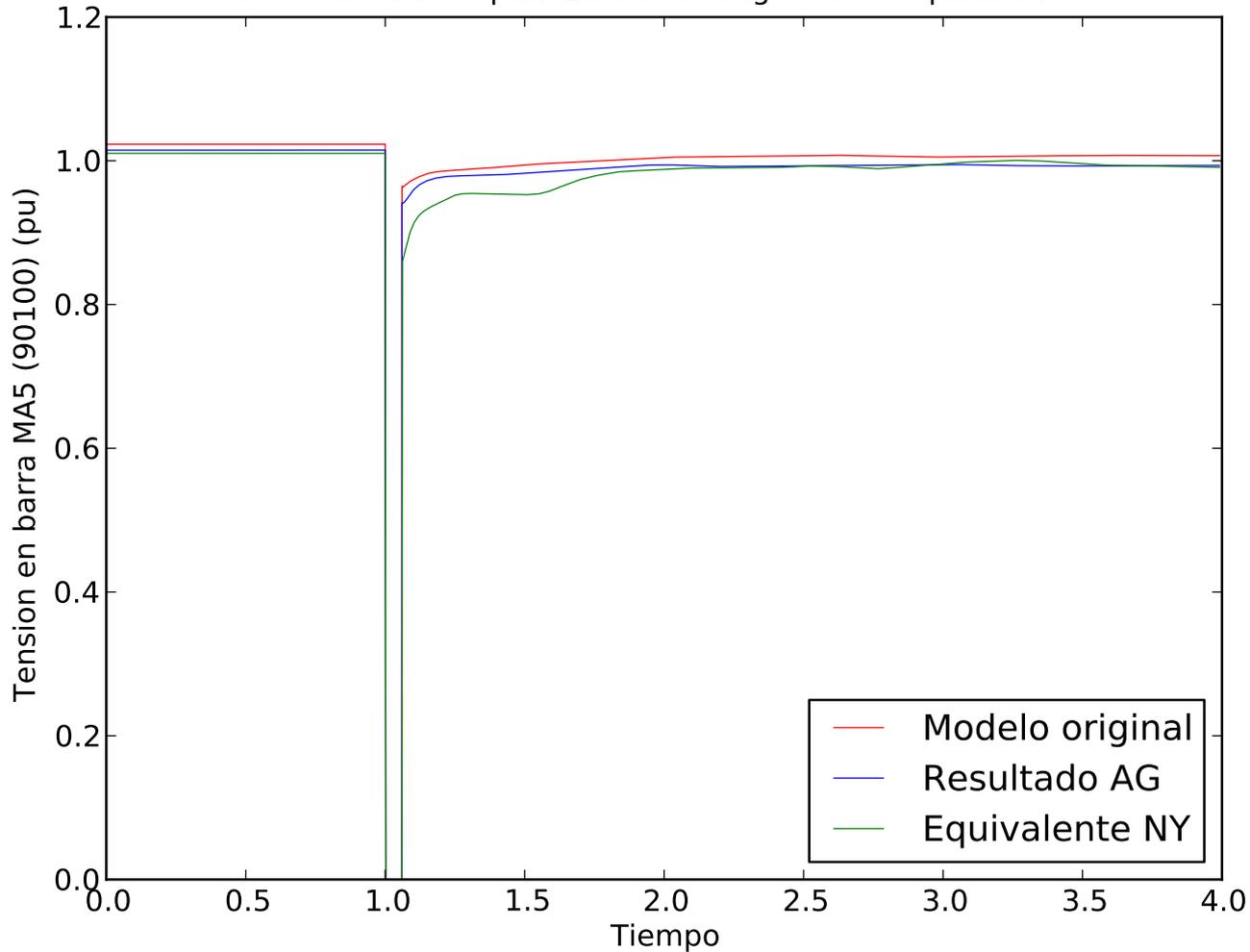
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



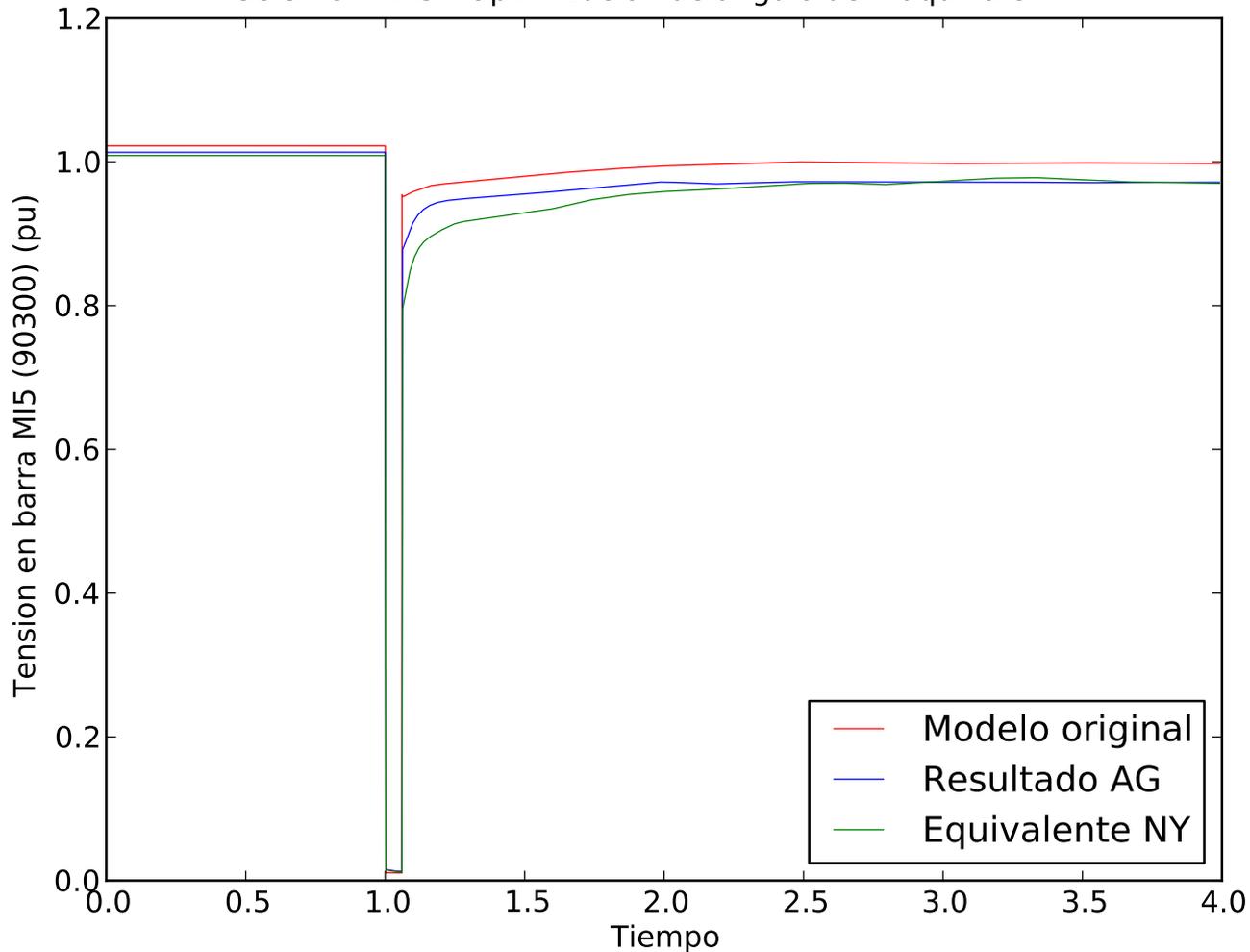
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



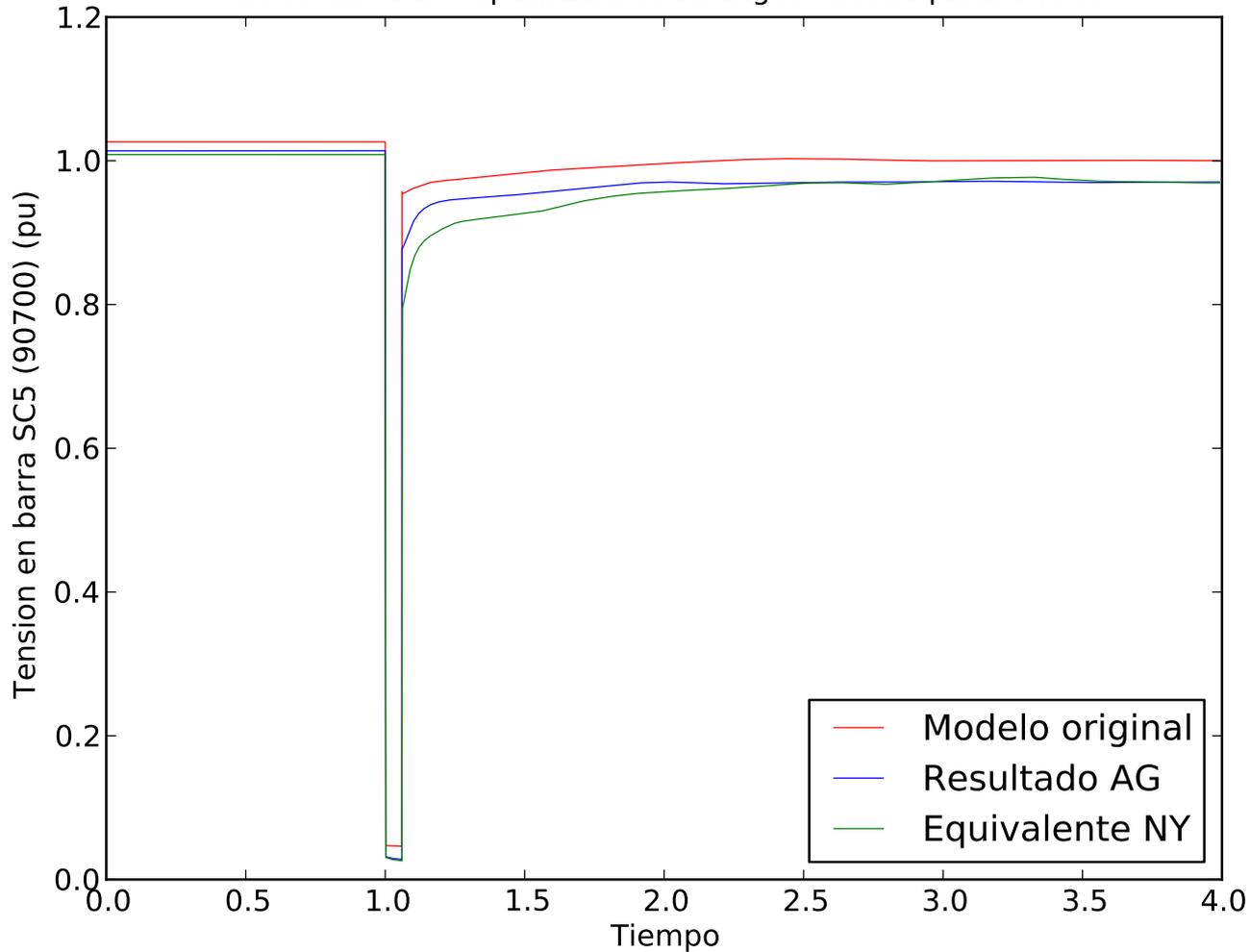
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



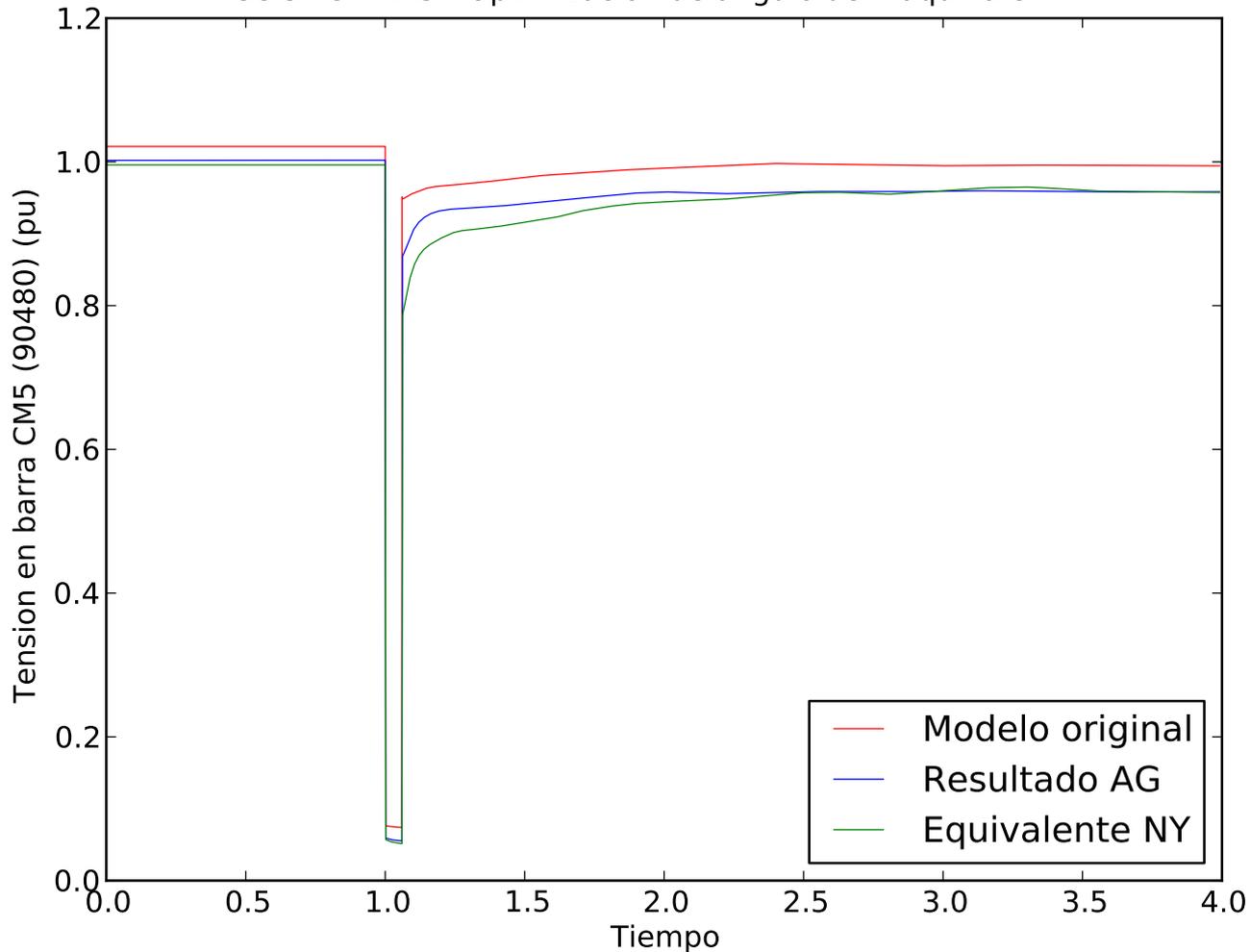
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



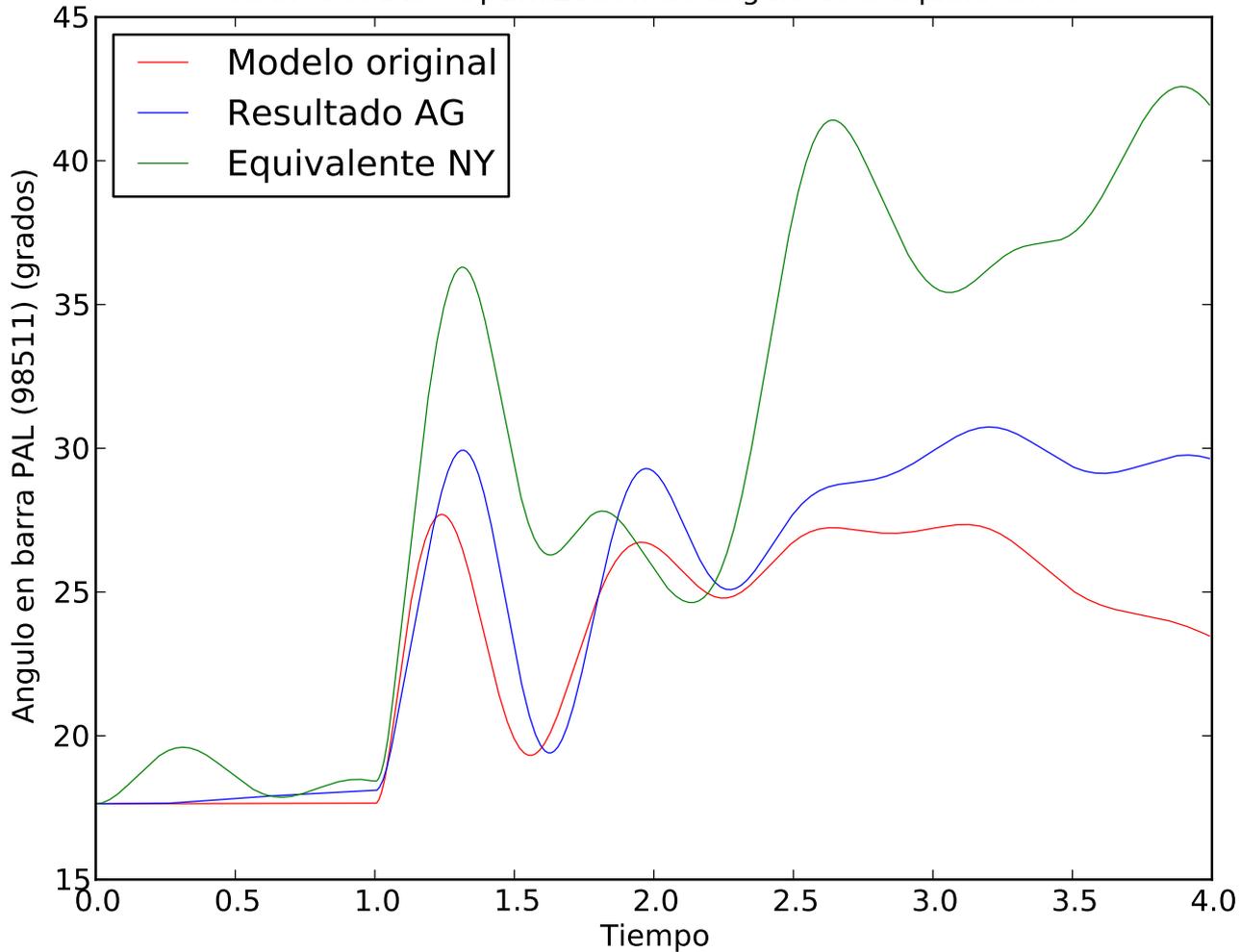
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



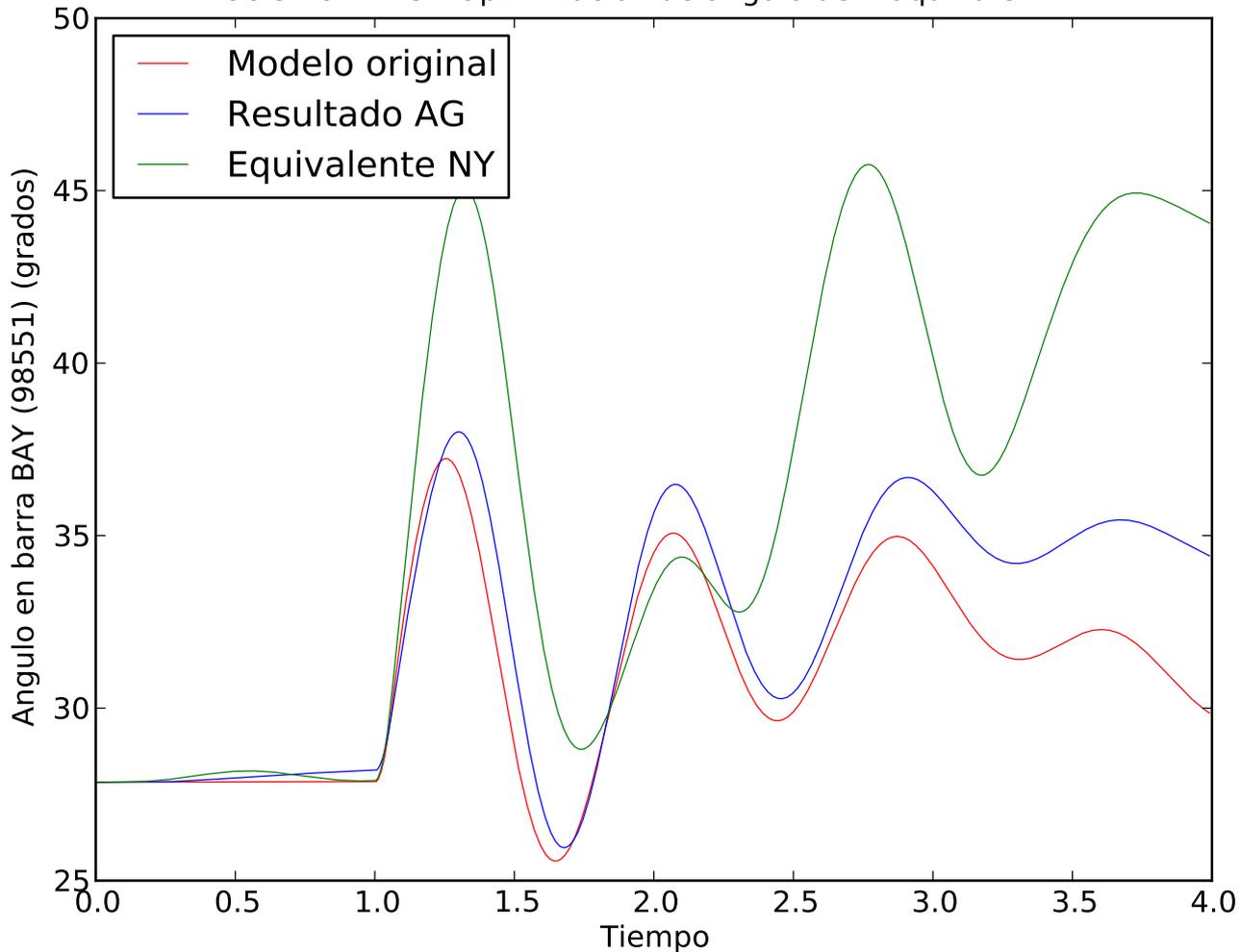
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



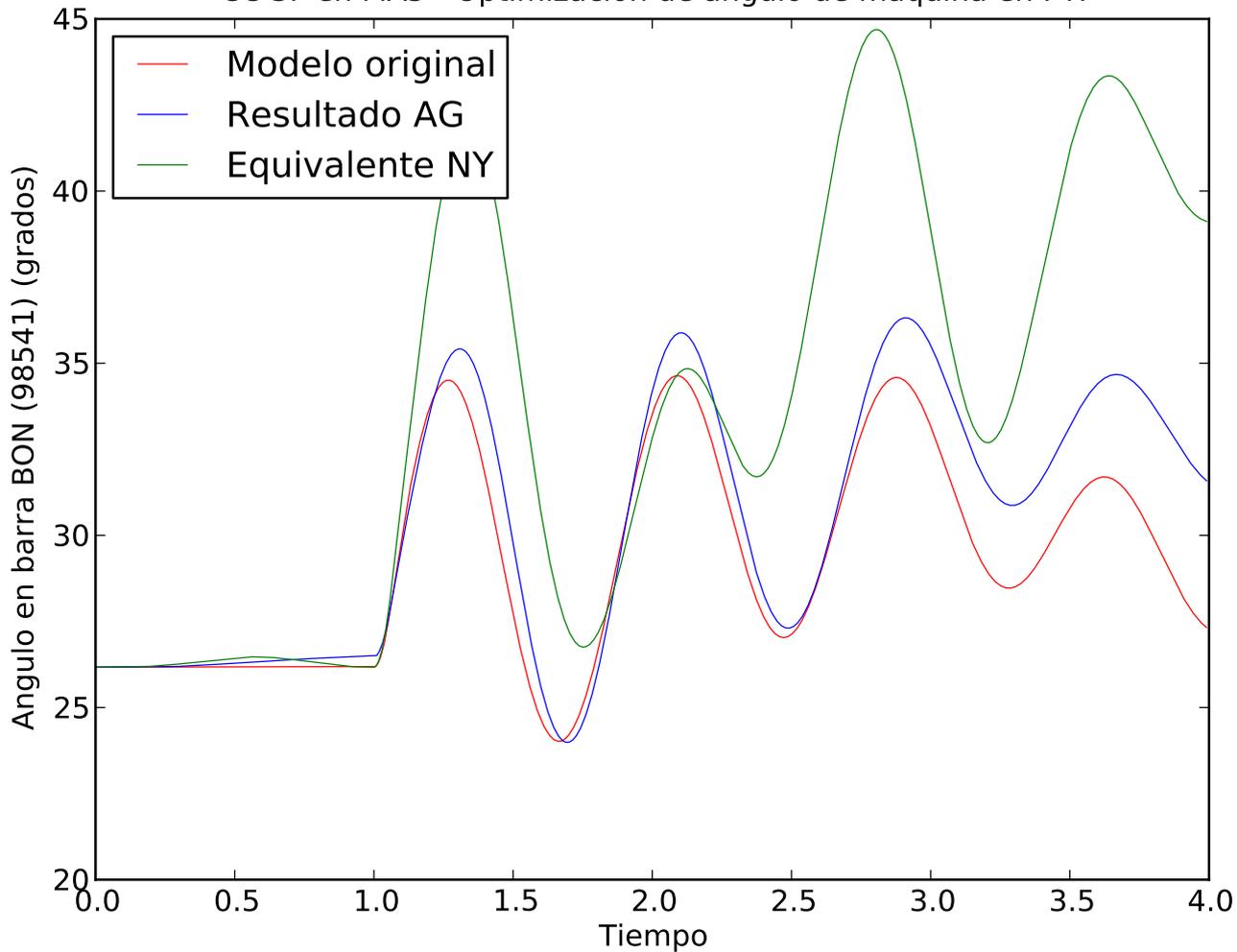
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



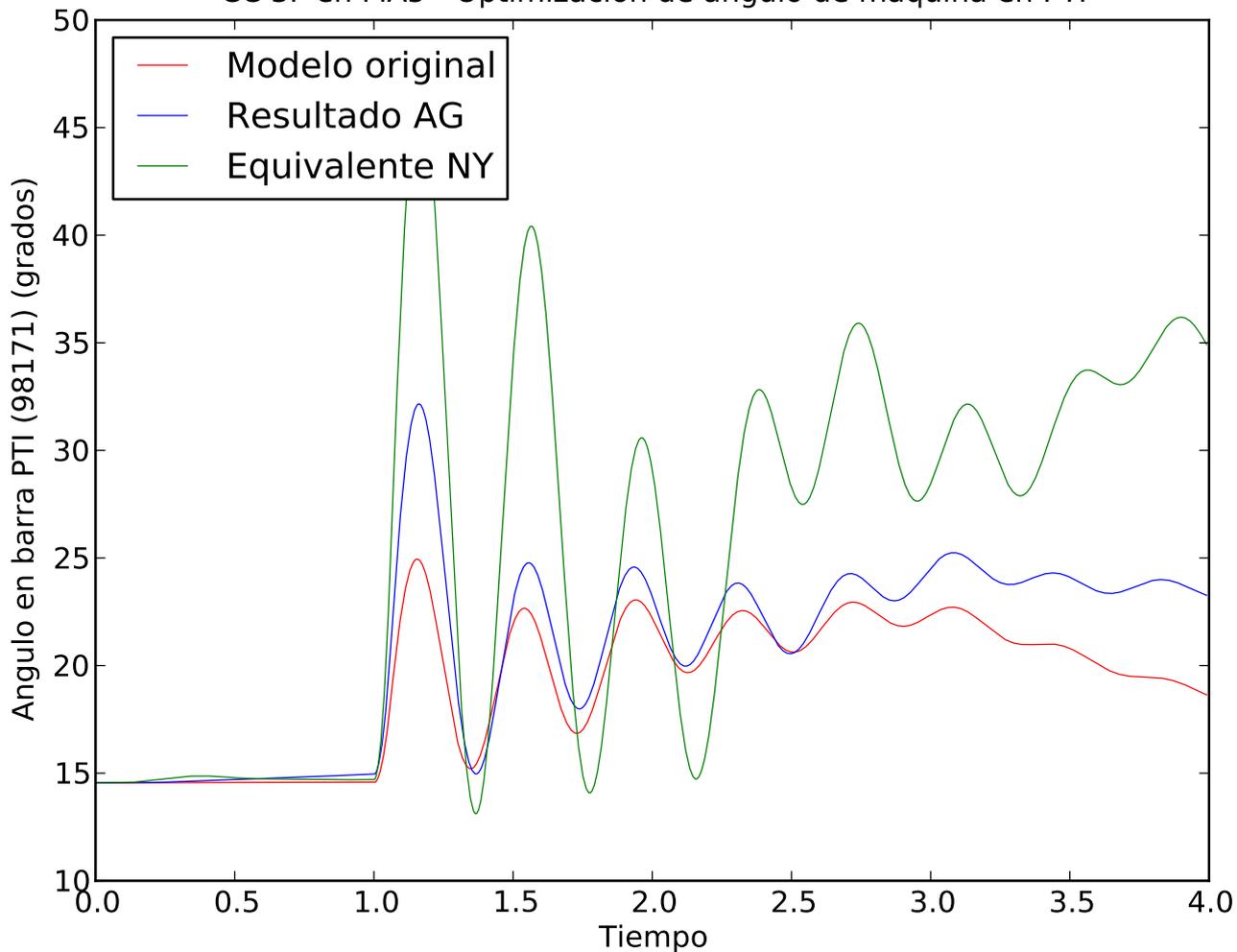
CC 3F en MA5 - Optimizacion de angulo de maquina en PTI



CC 3F en MA5 - Optimizacion de angulo de maquina en PTI

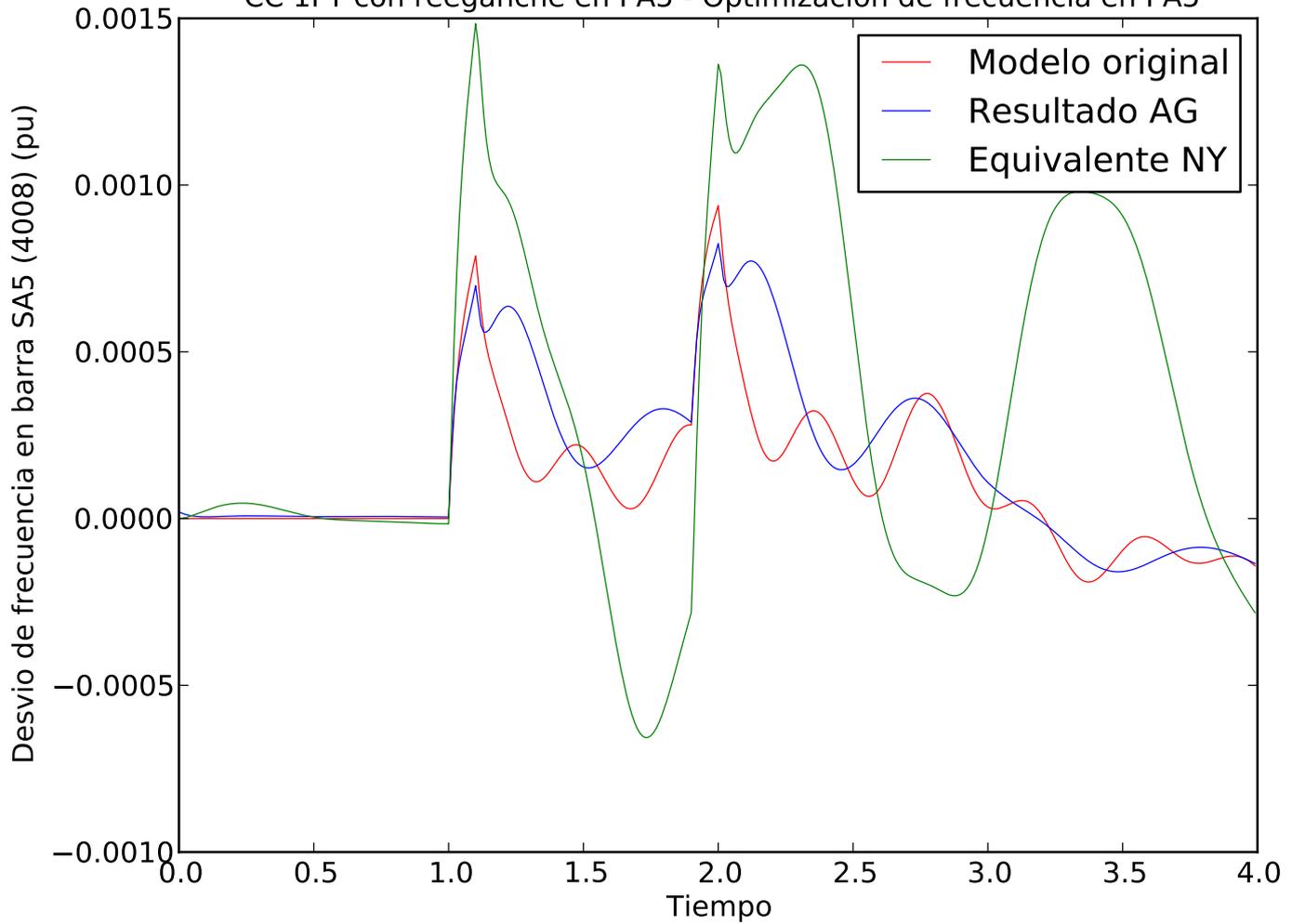


CC 3F en MA5 - Optimizacion de angulo de maquina en PTI

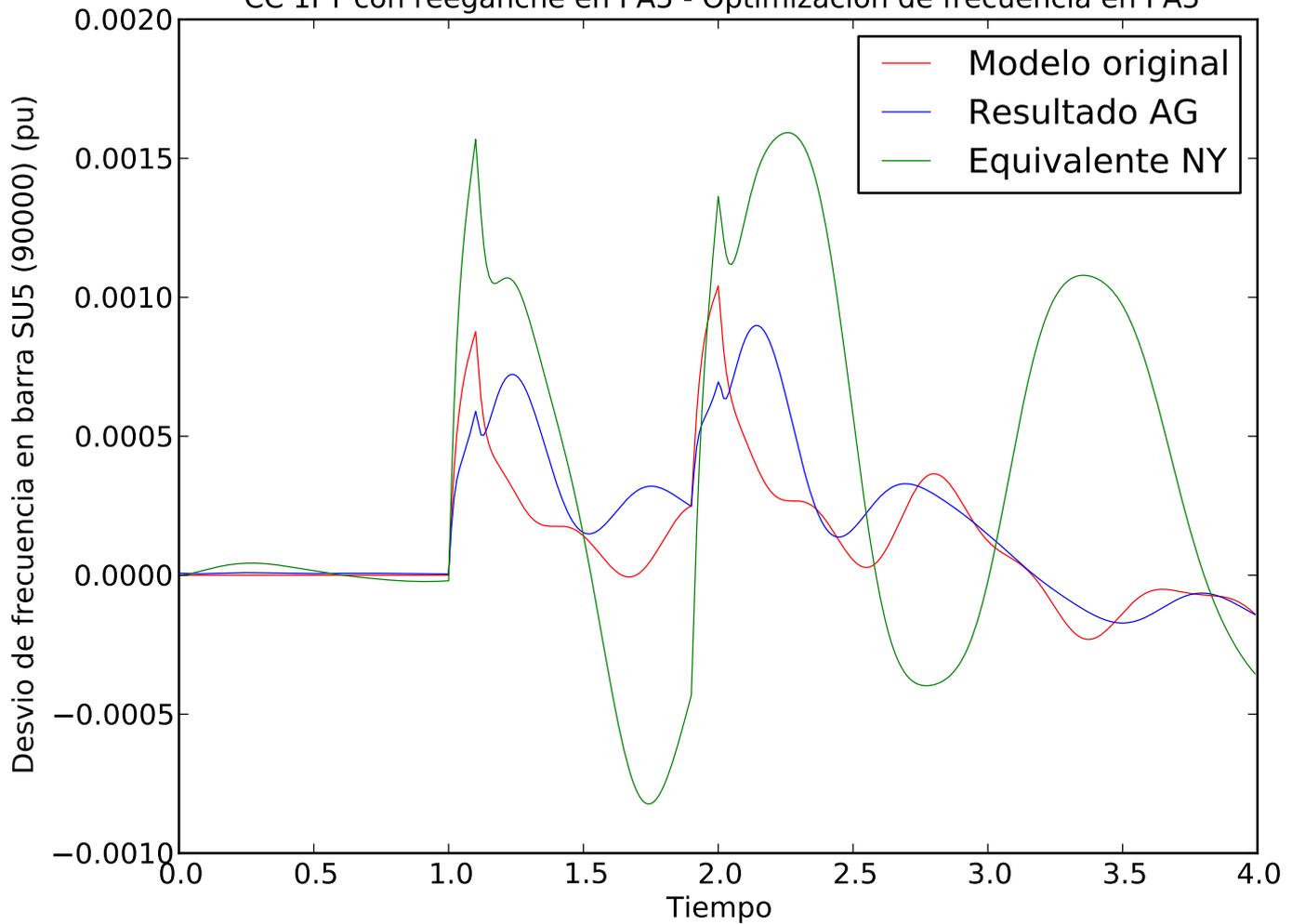


- C.10. Simulación 7: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando frecuencia en PA5 en 4 segundos. Comportamiento en 4 segundos

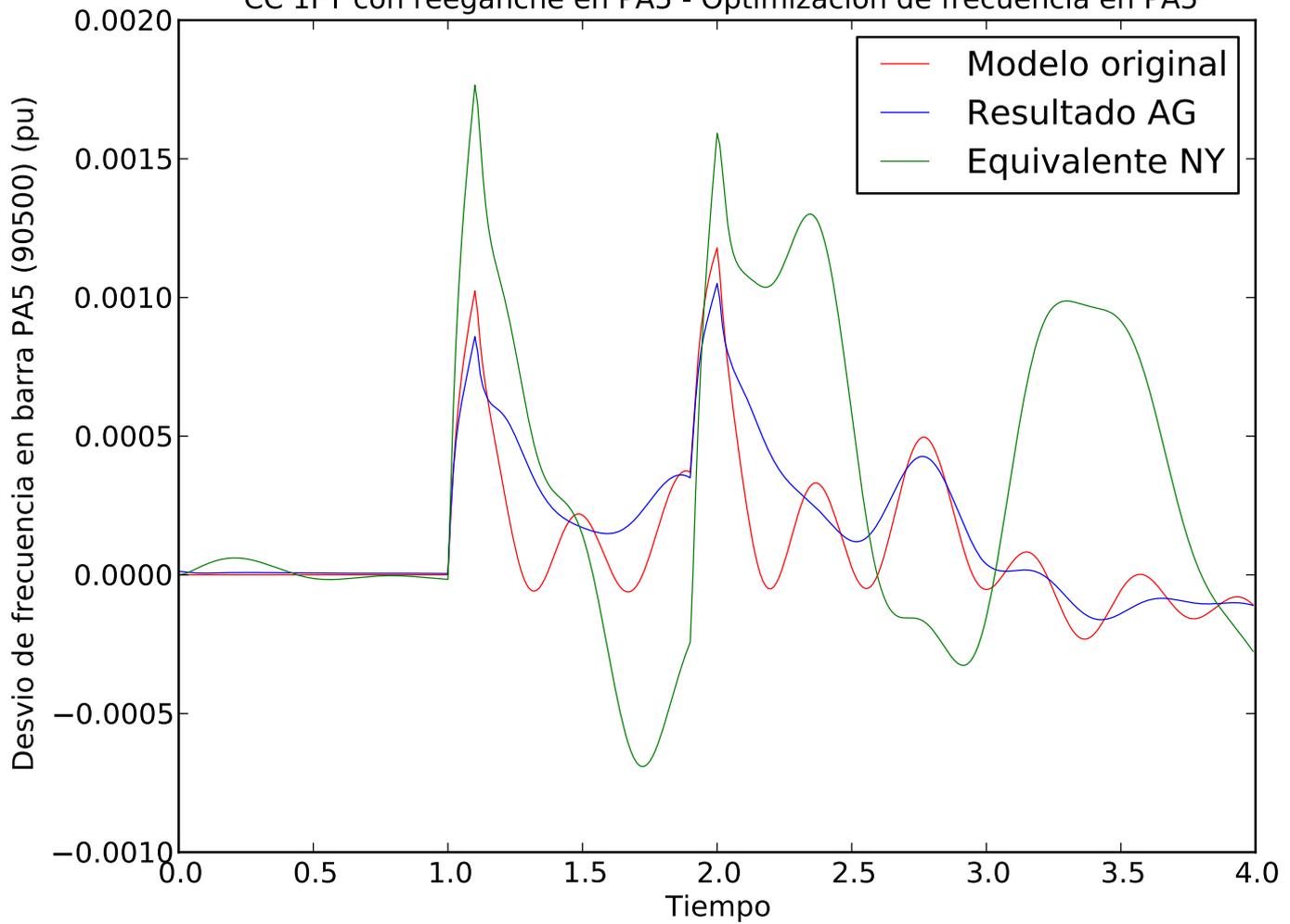
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



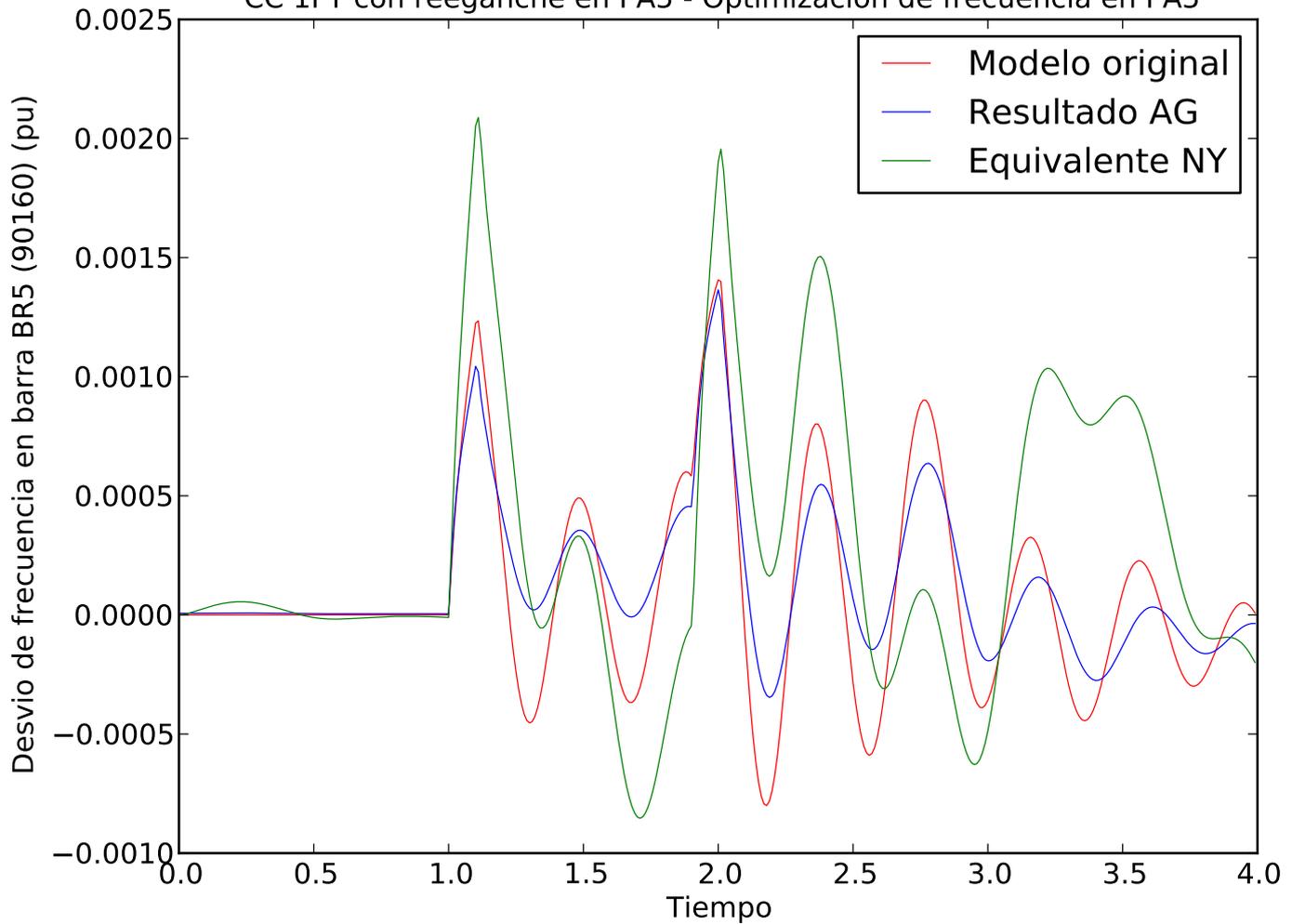
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



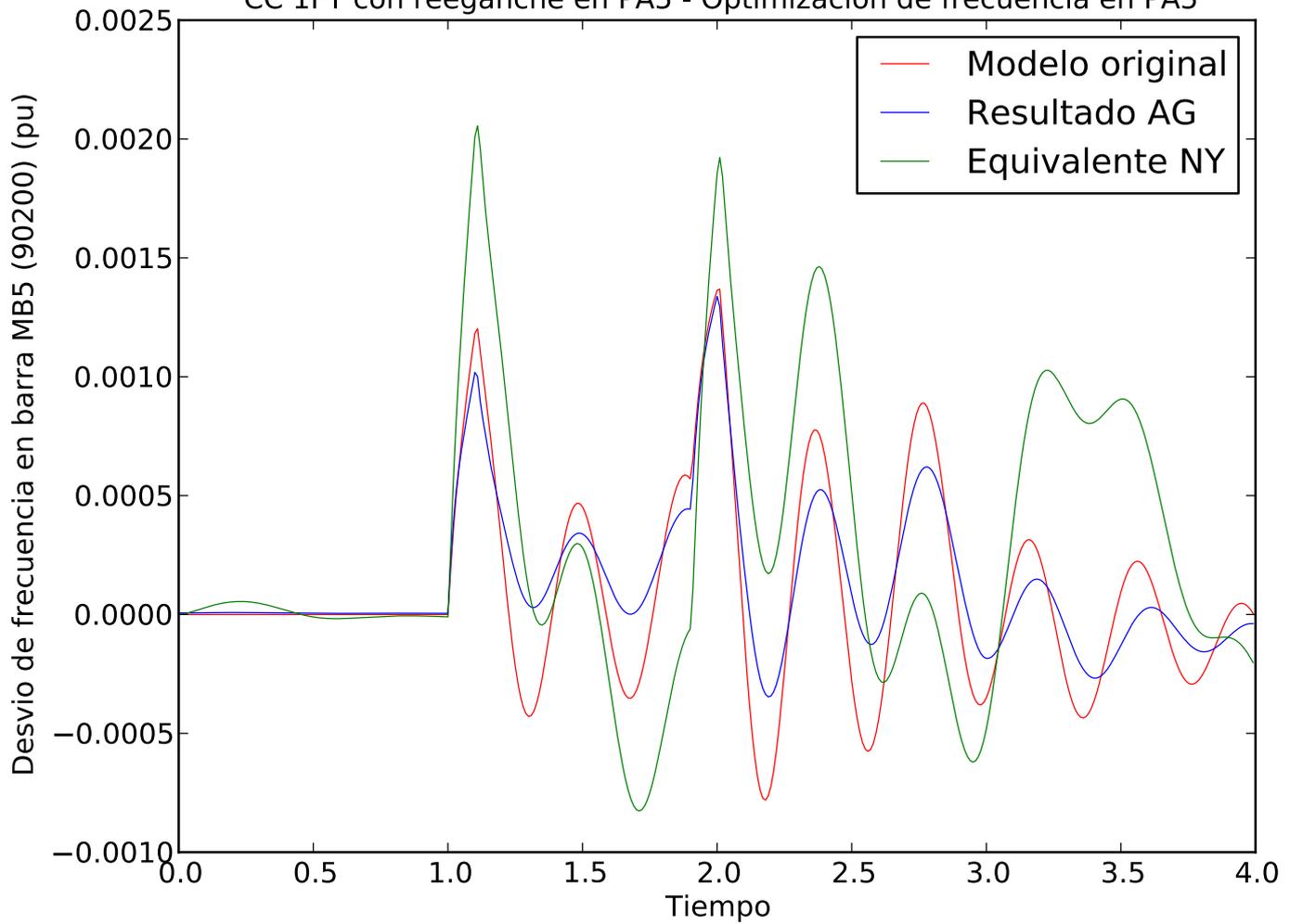
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



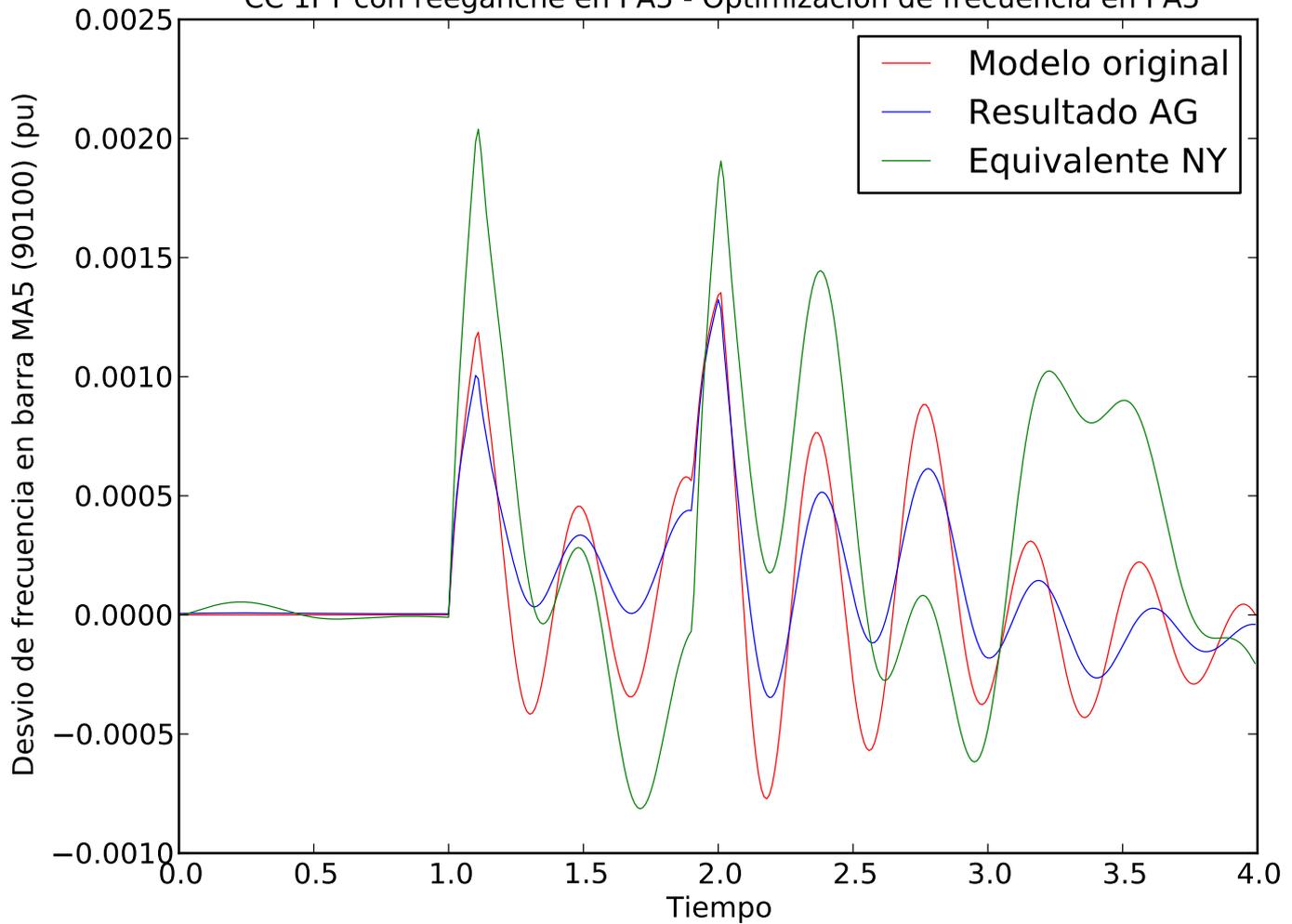
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



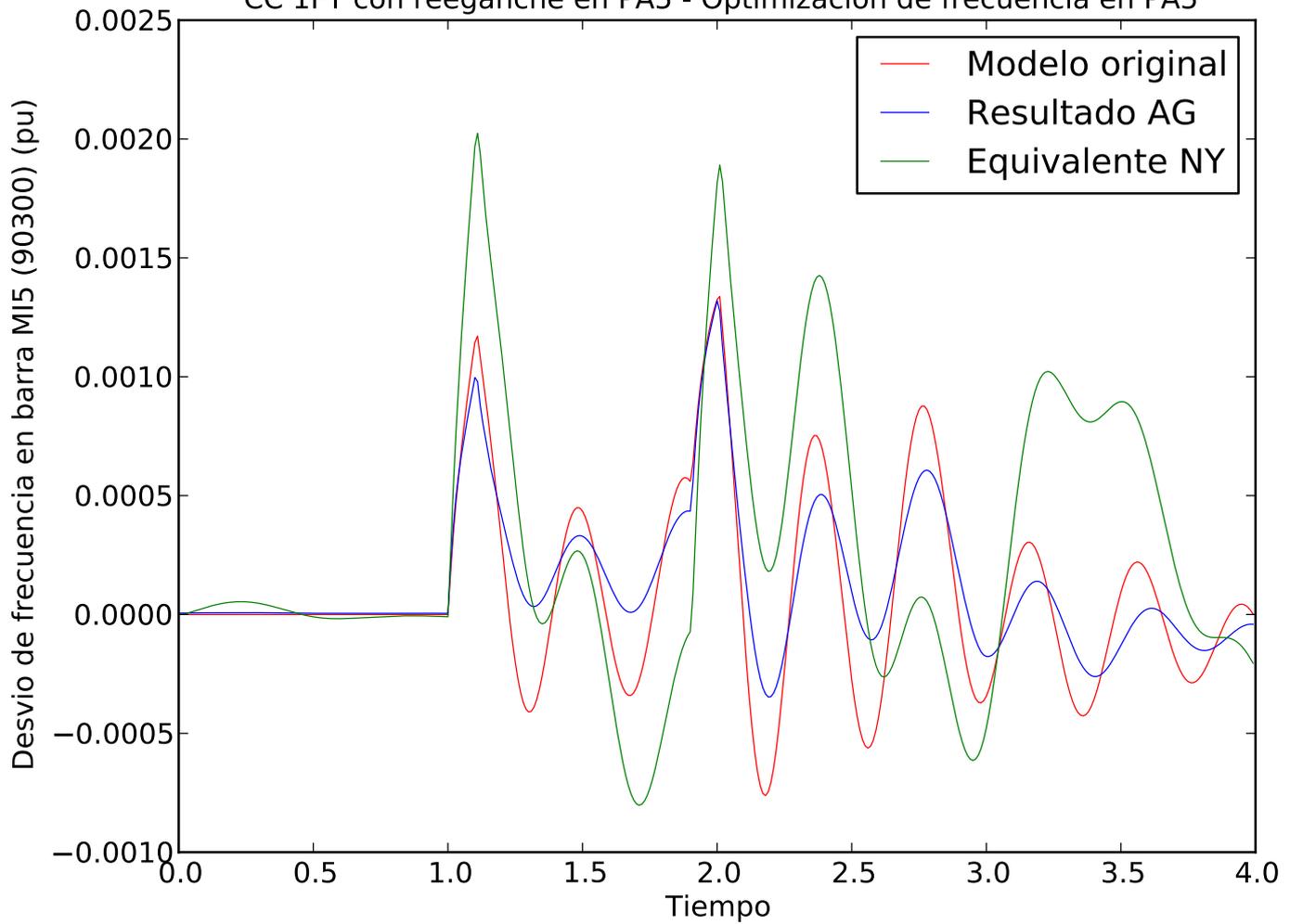
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



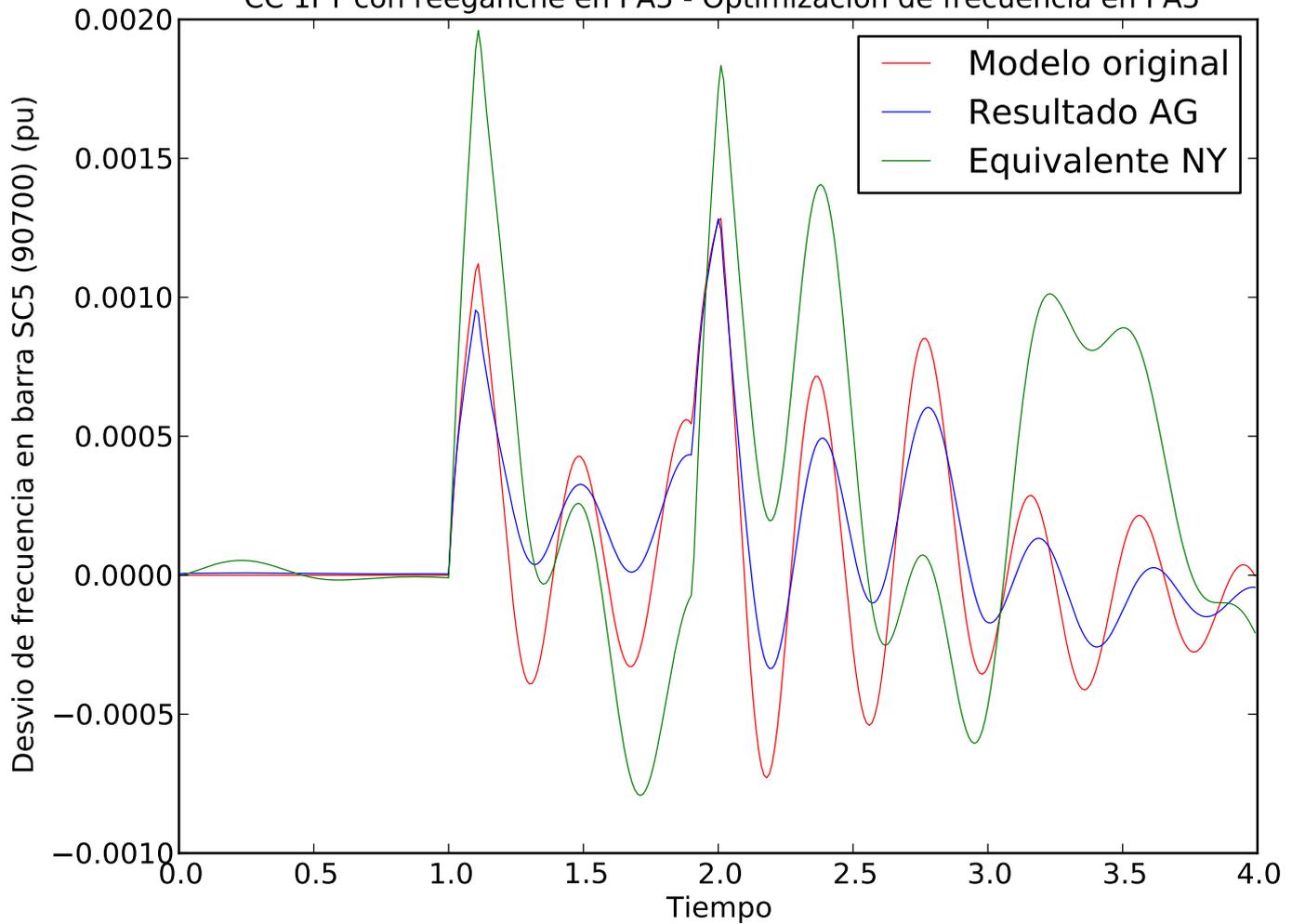
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



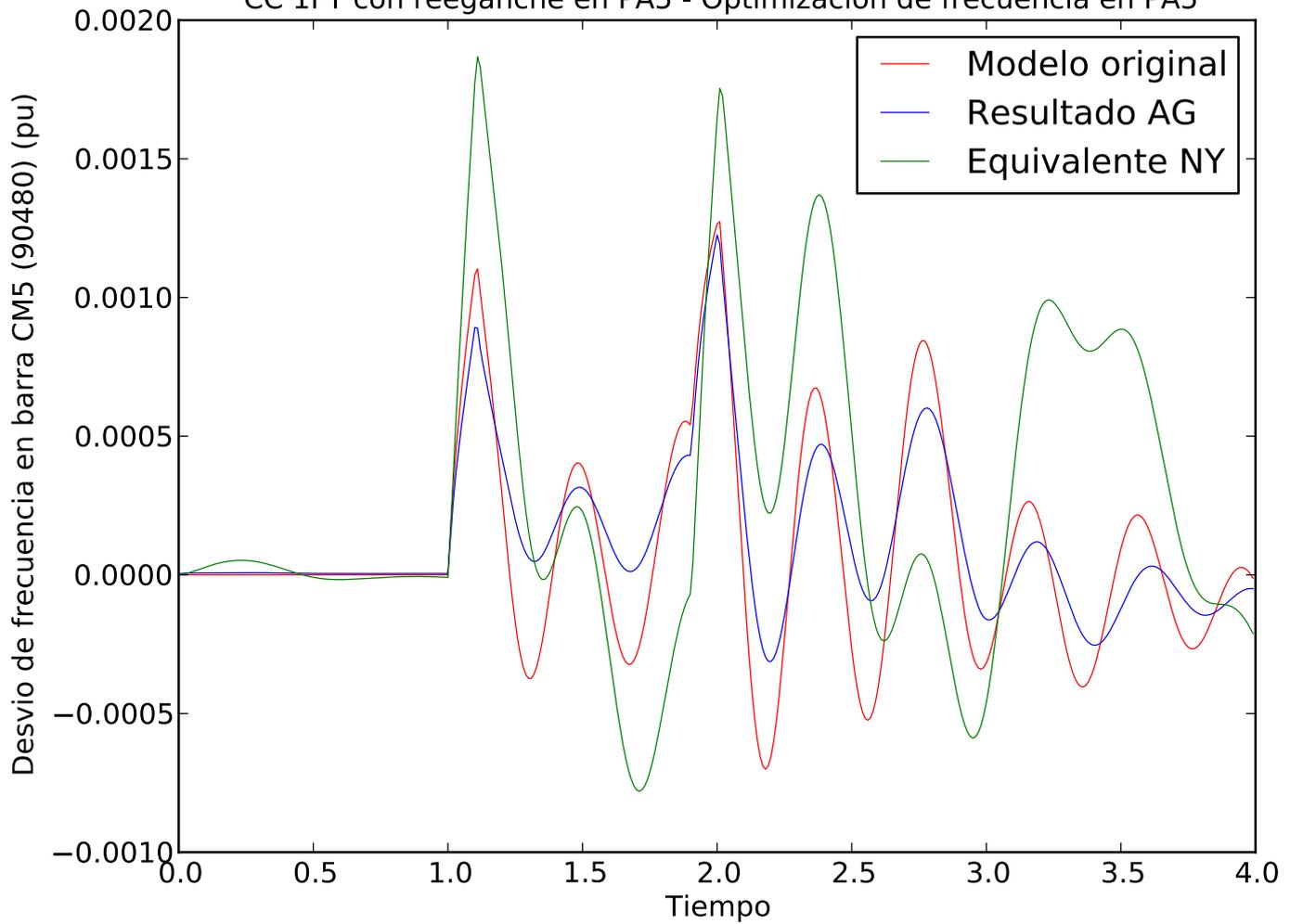
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



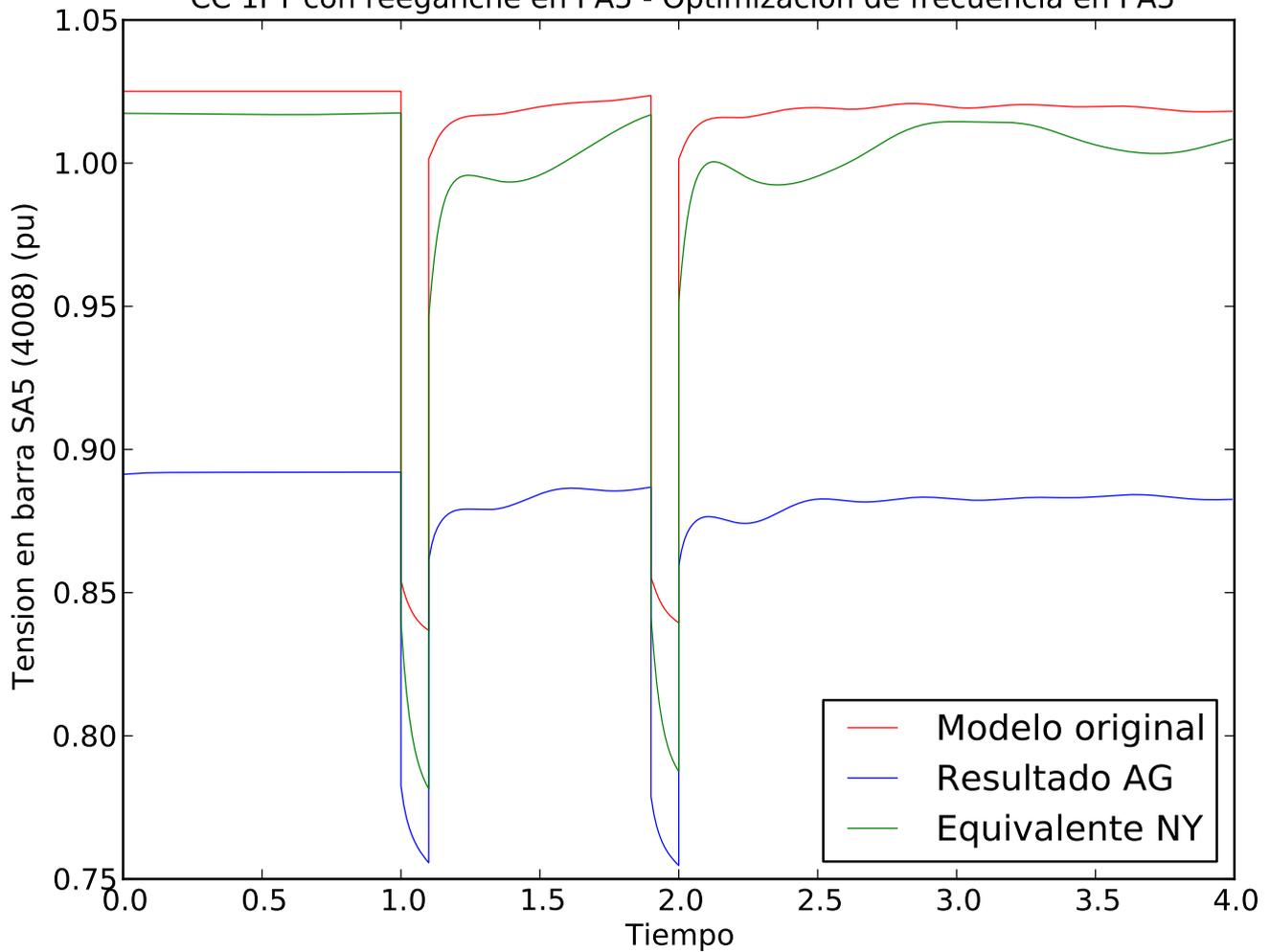
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



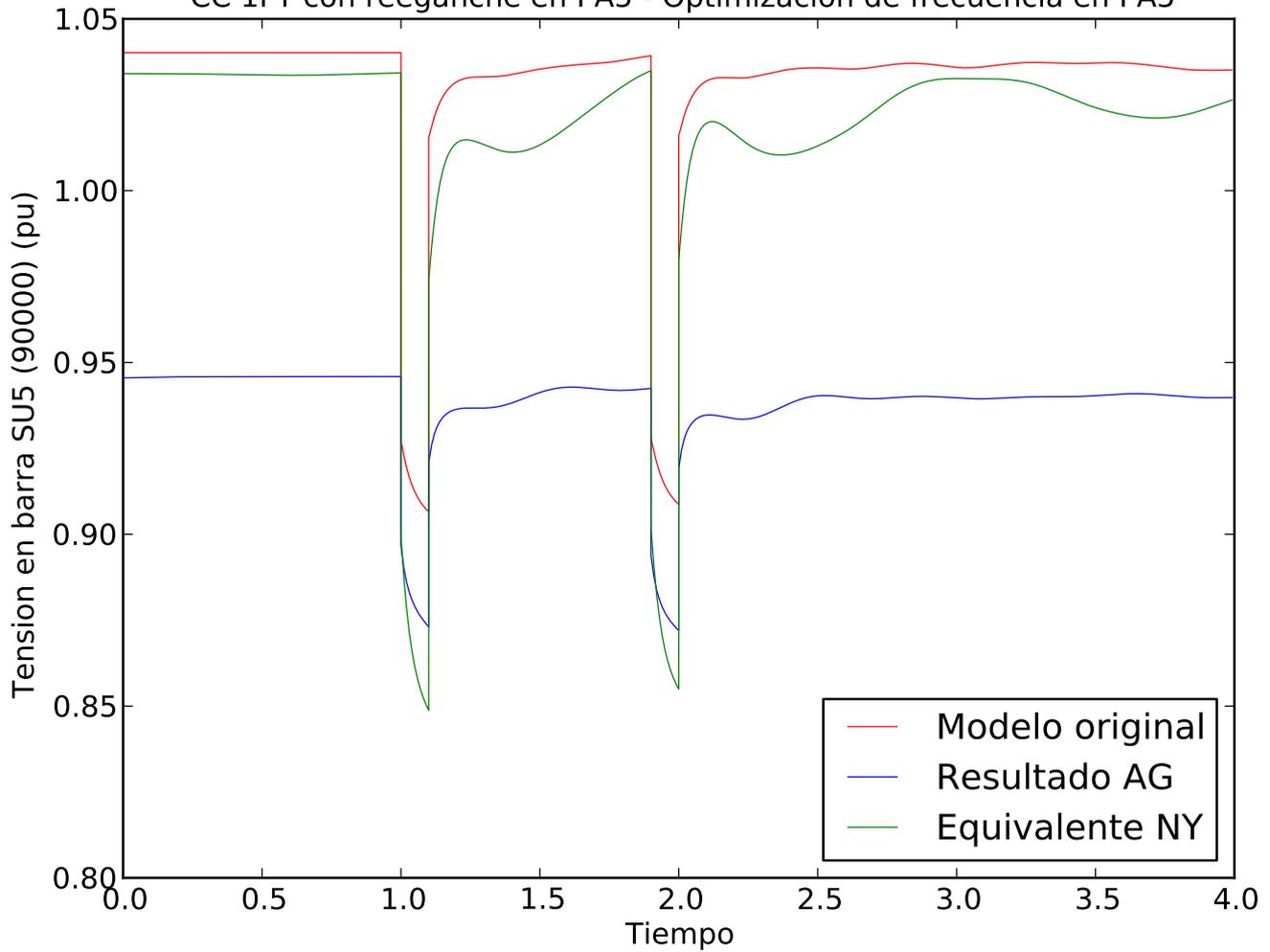
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



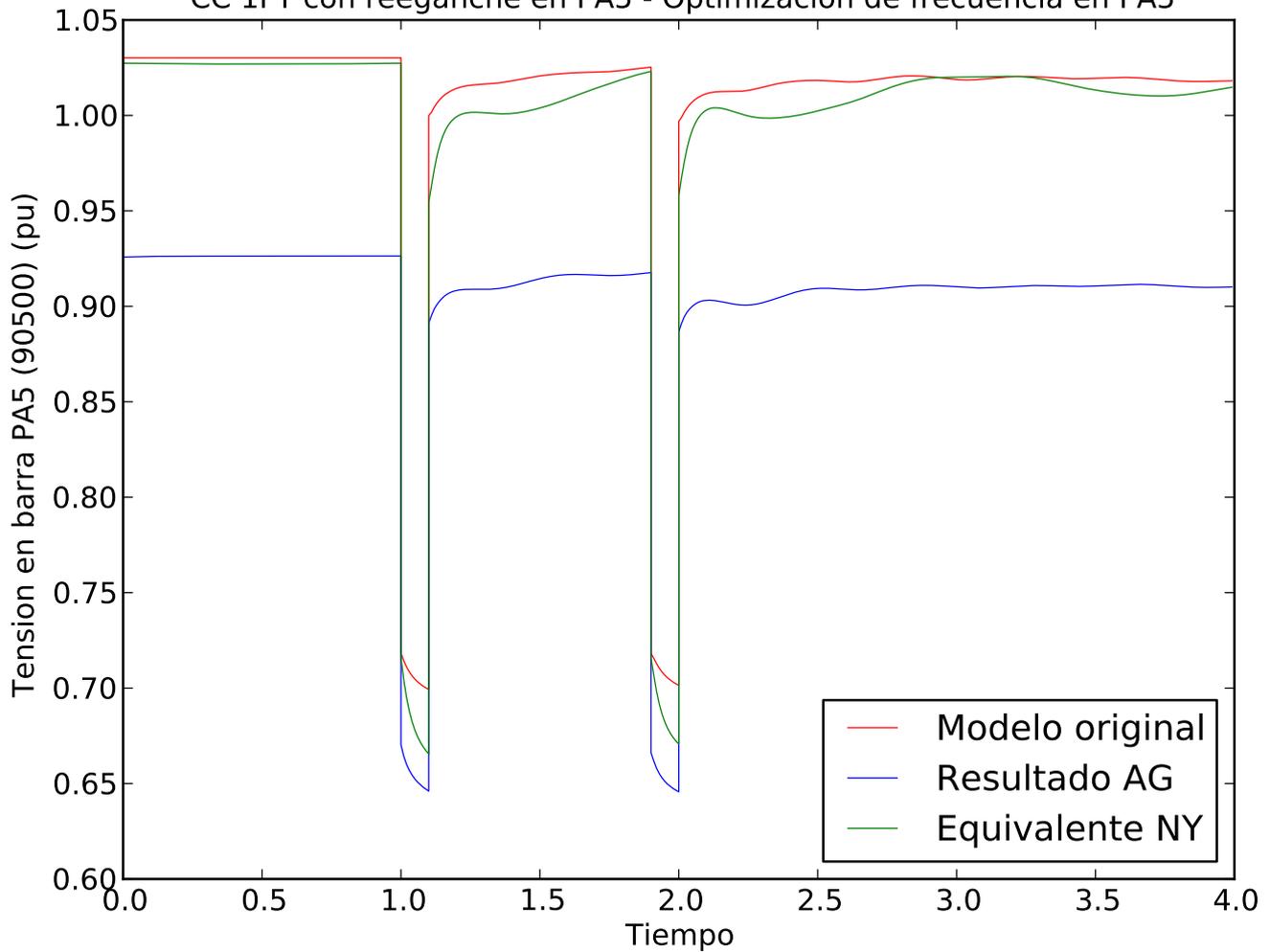
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



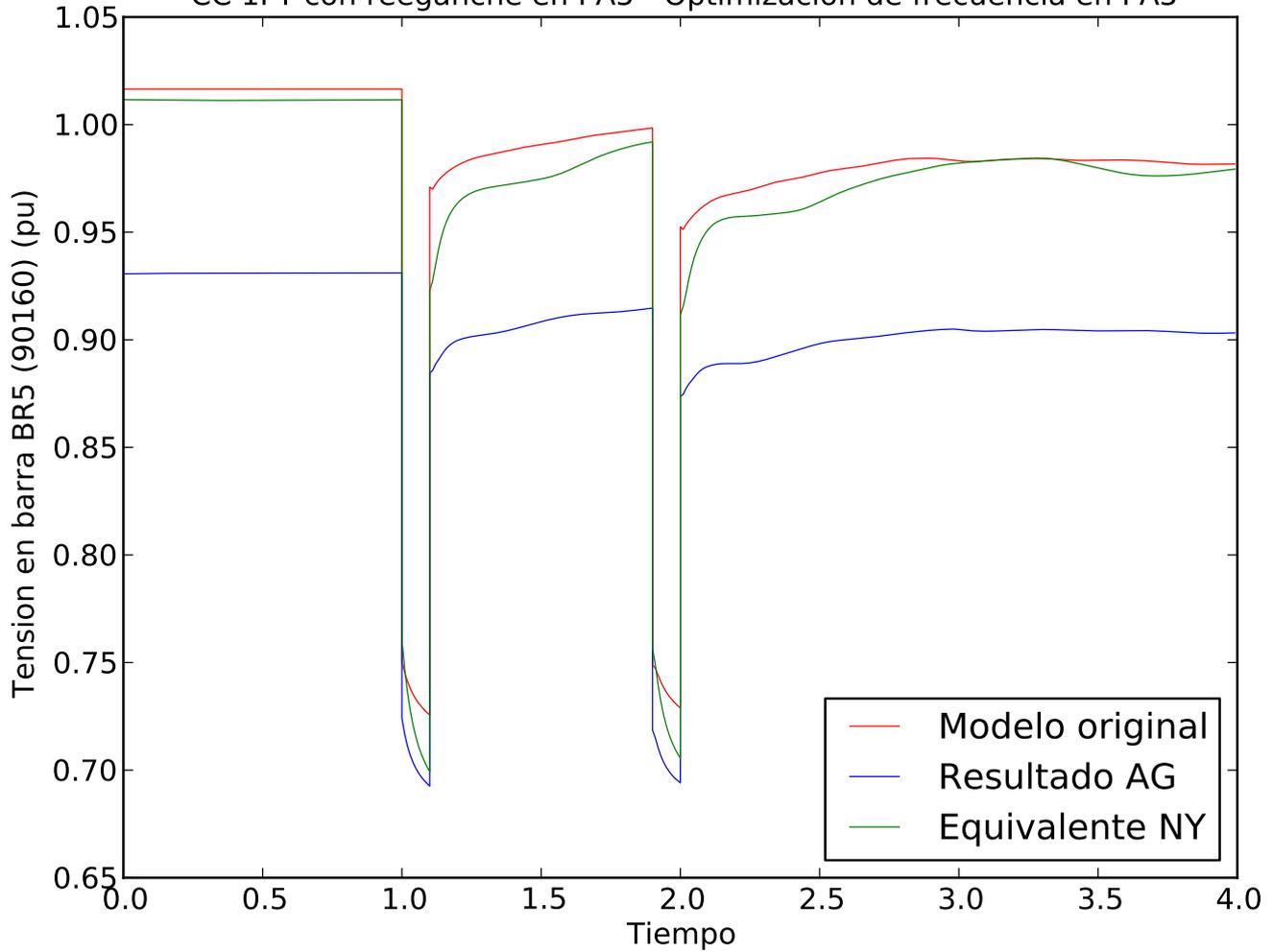
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



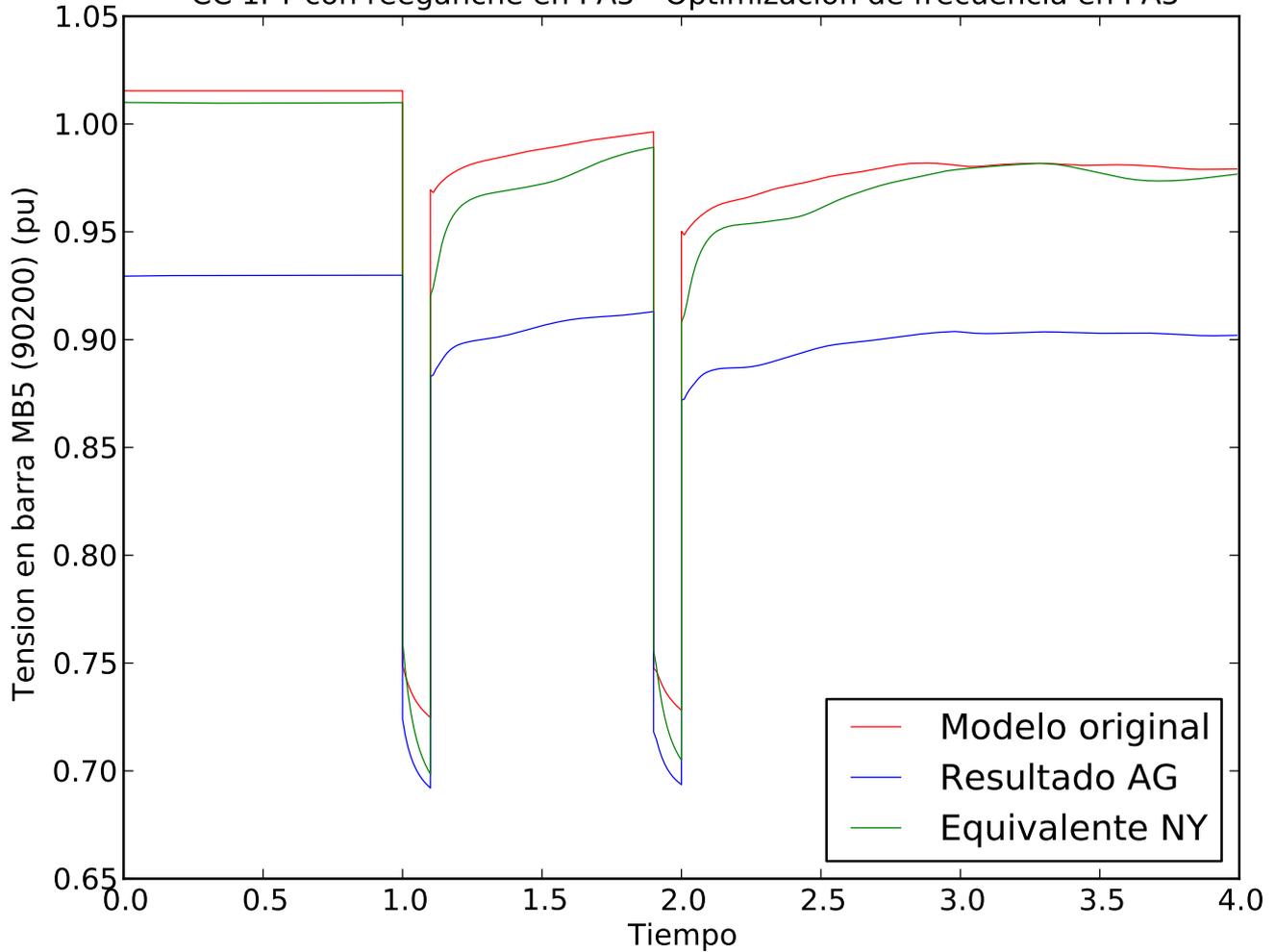
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



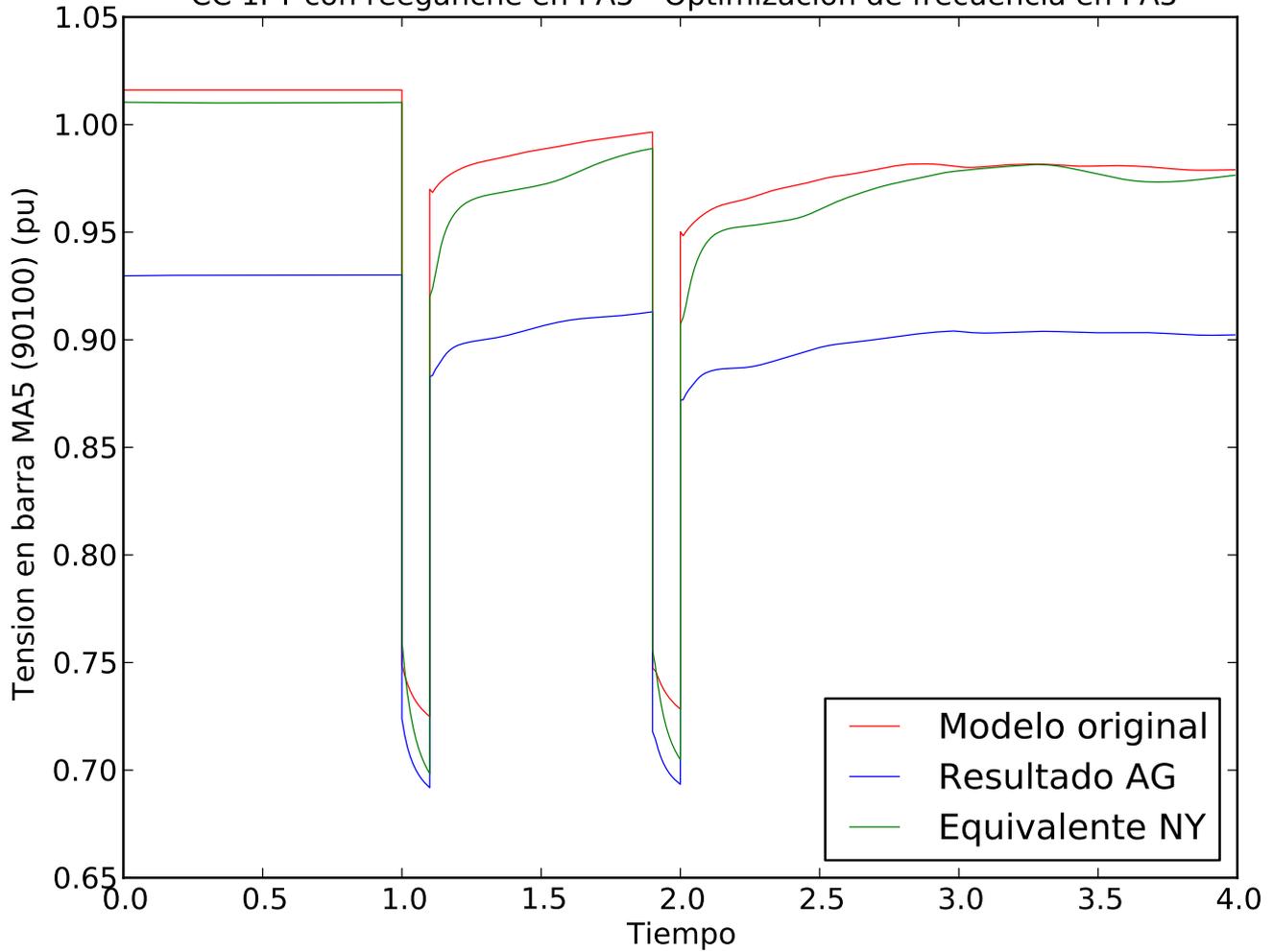
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



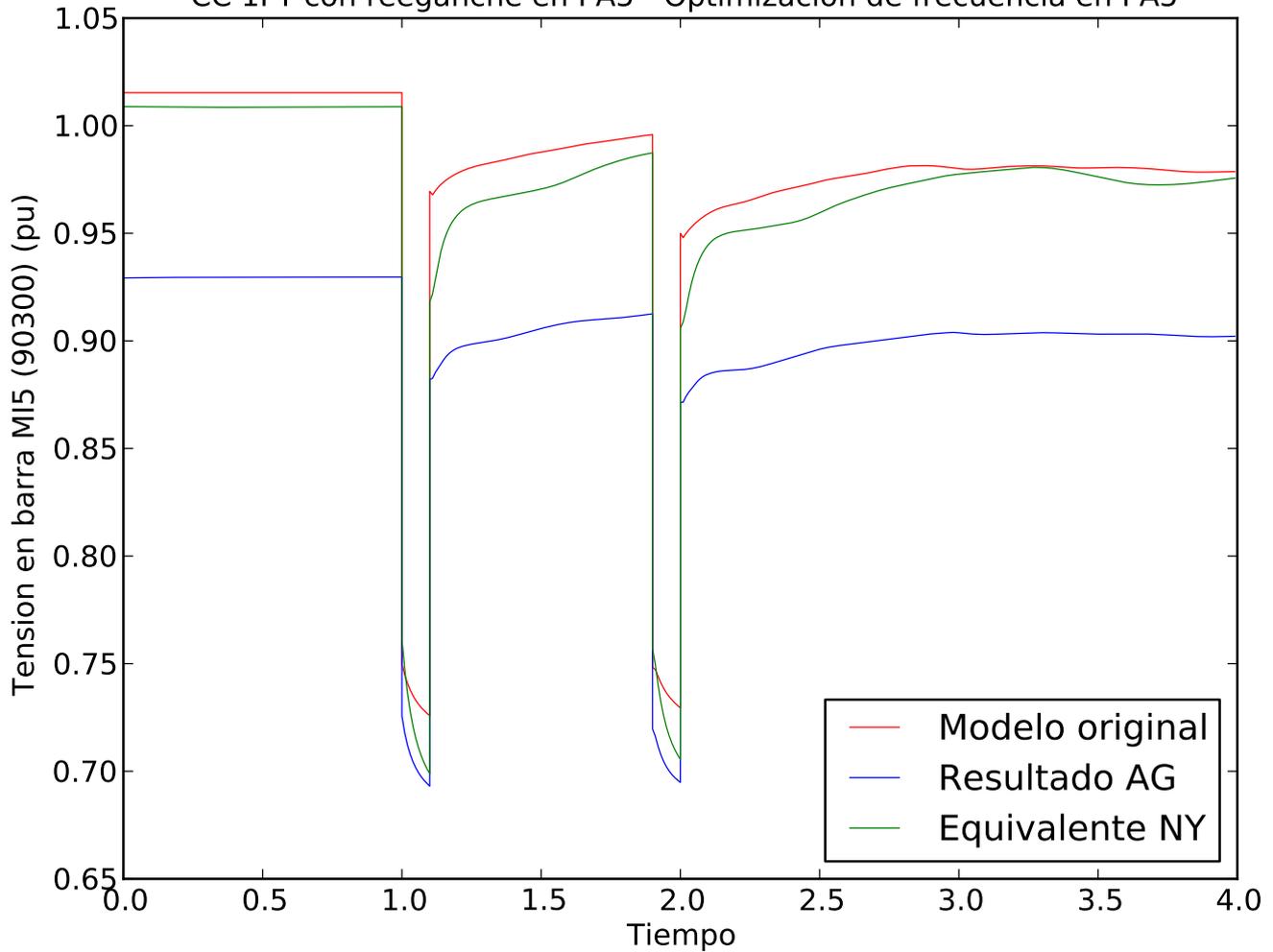
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



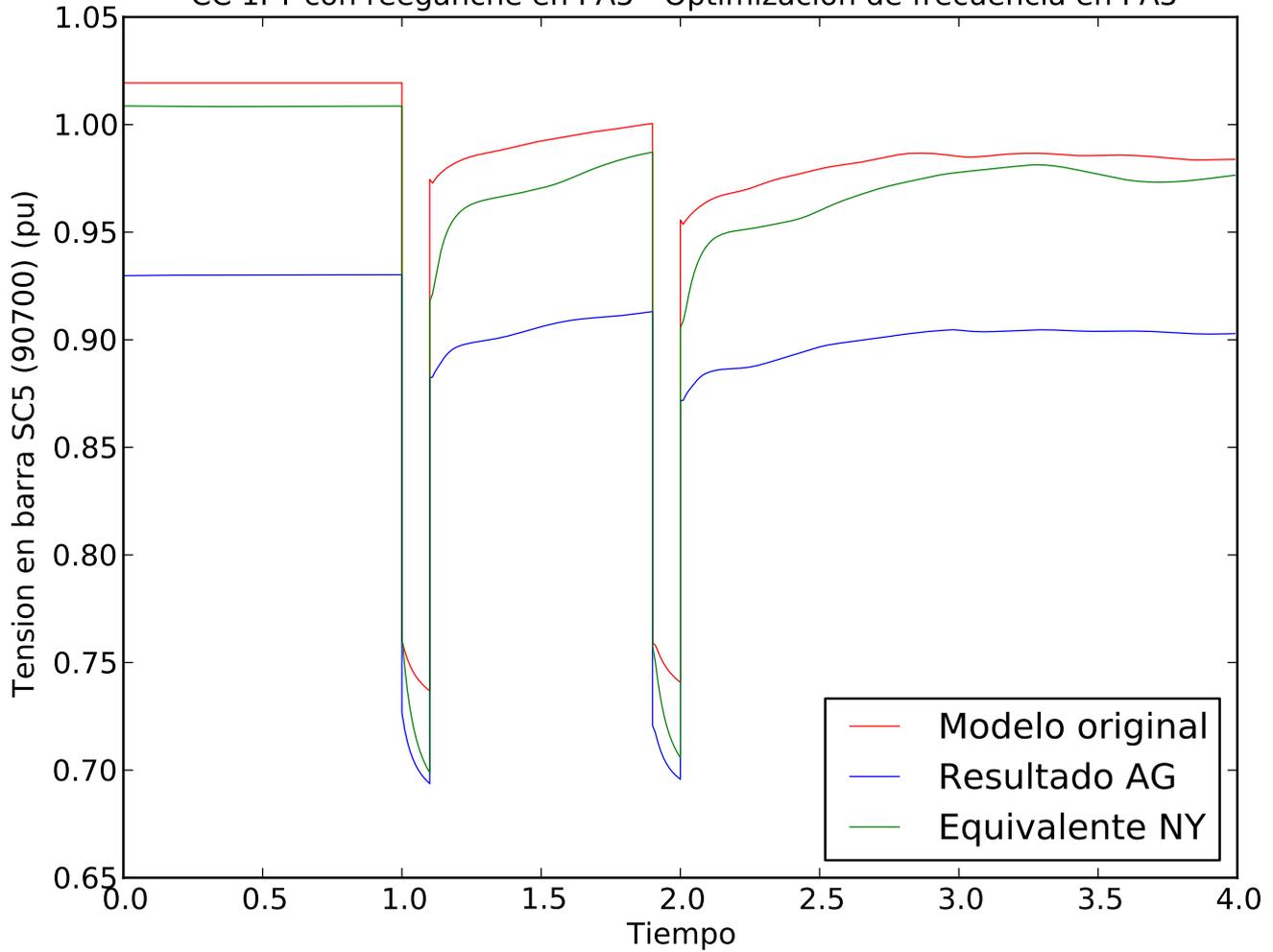
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



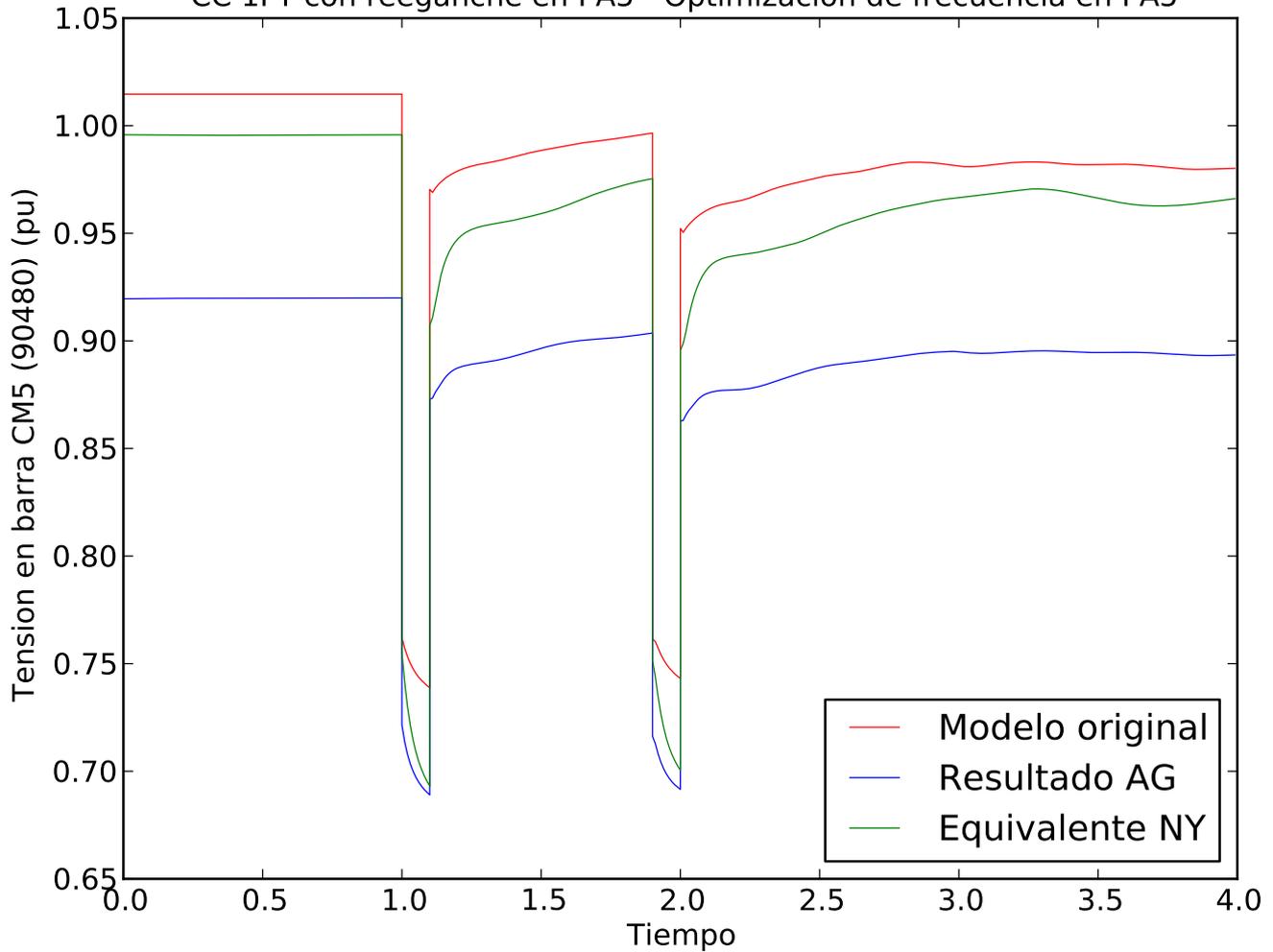
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



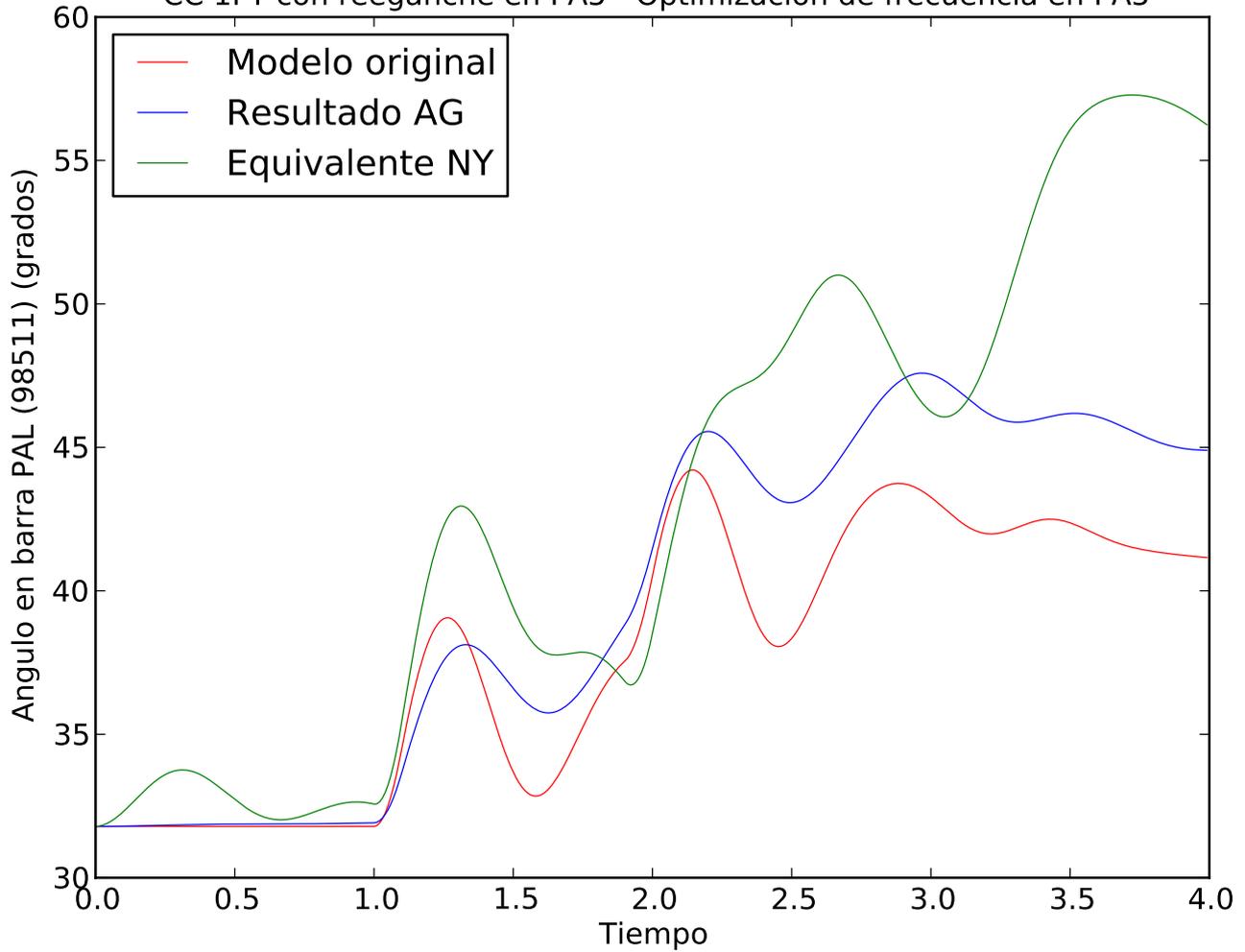
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



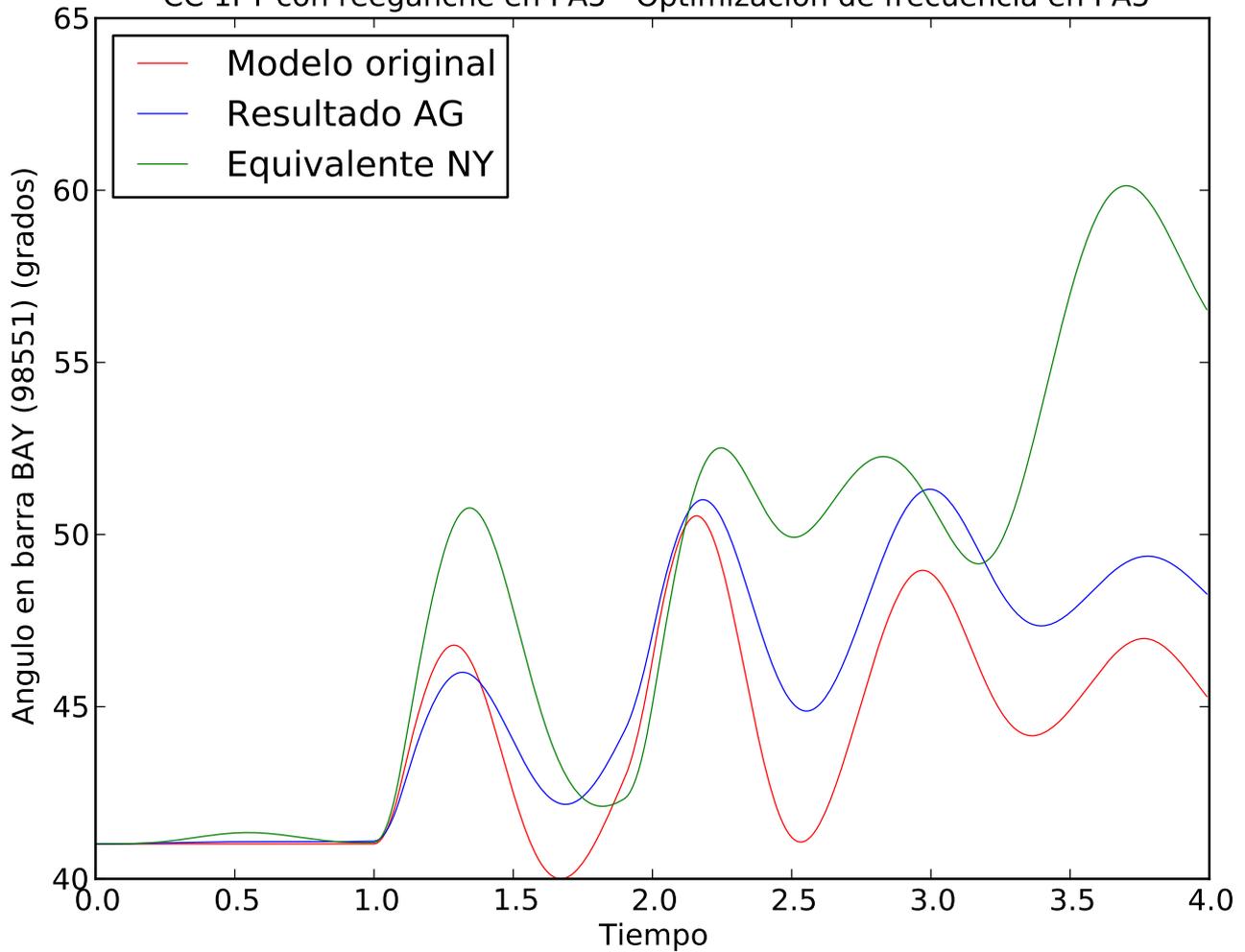
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



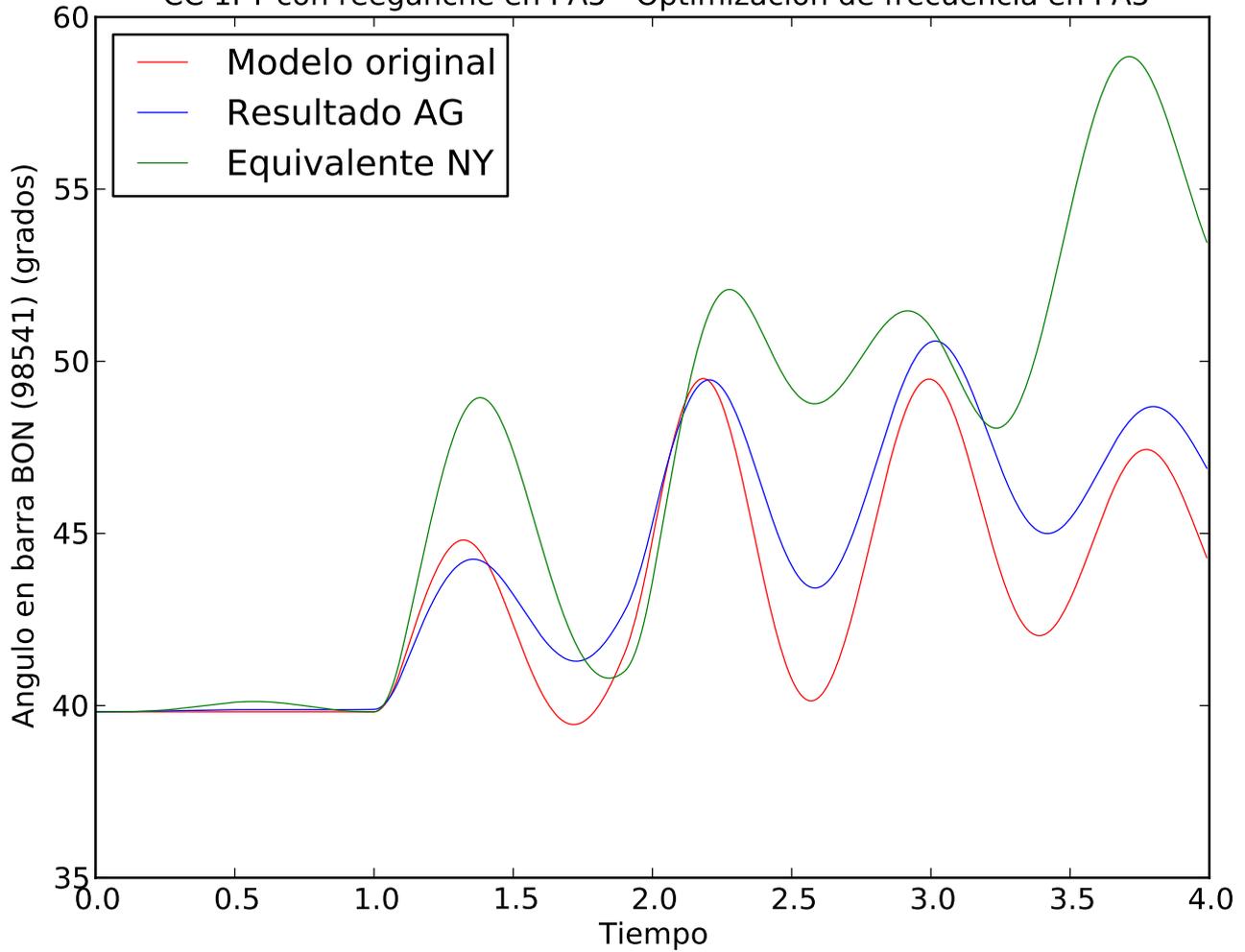
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



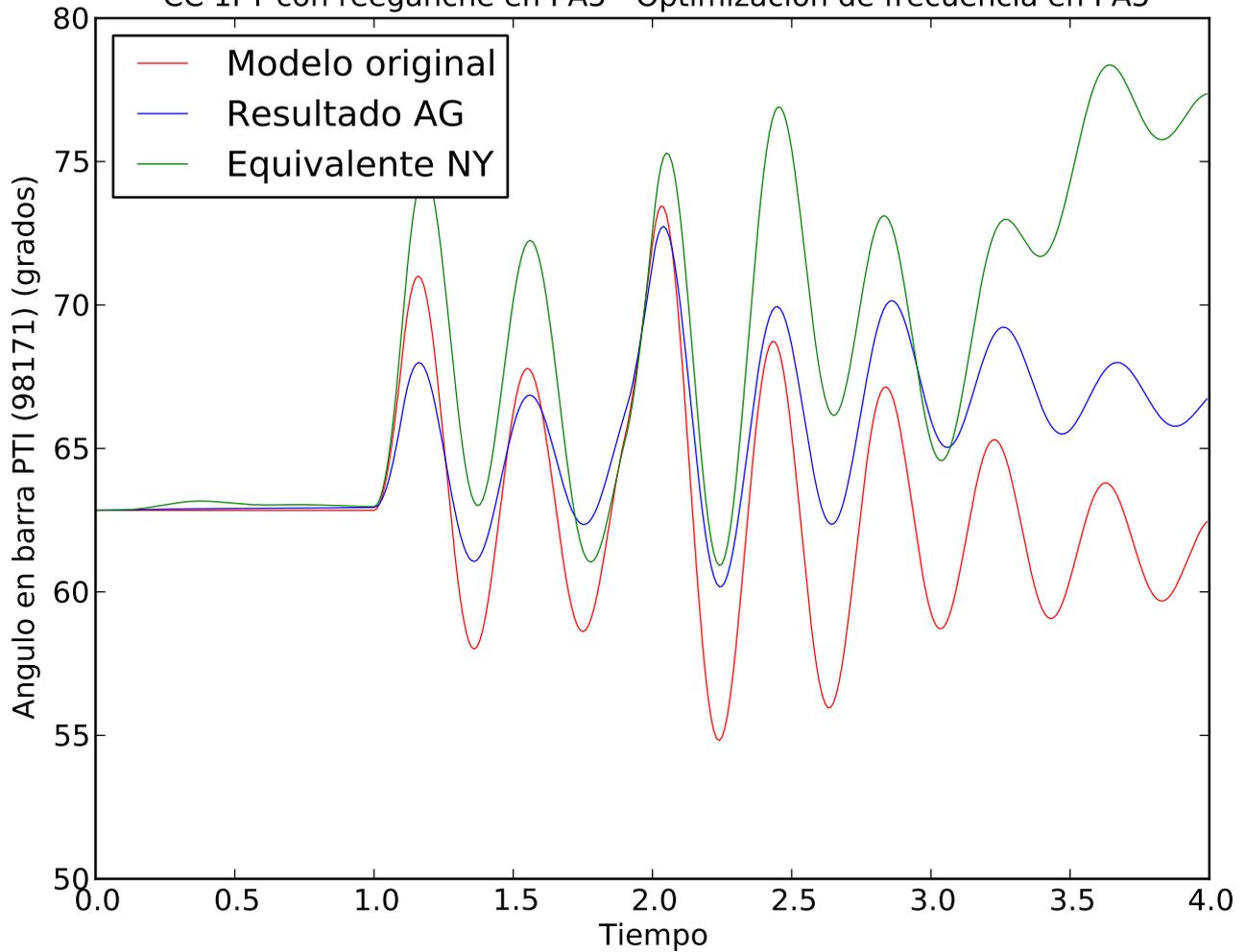
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5

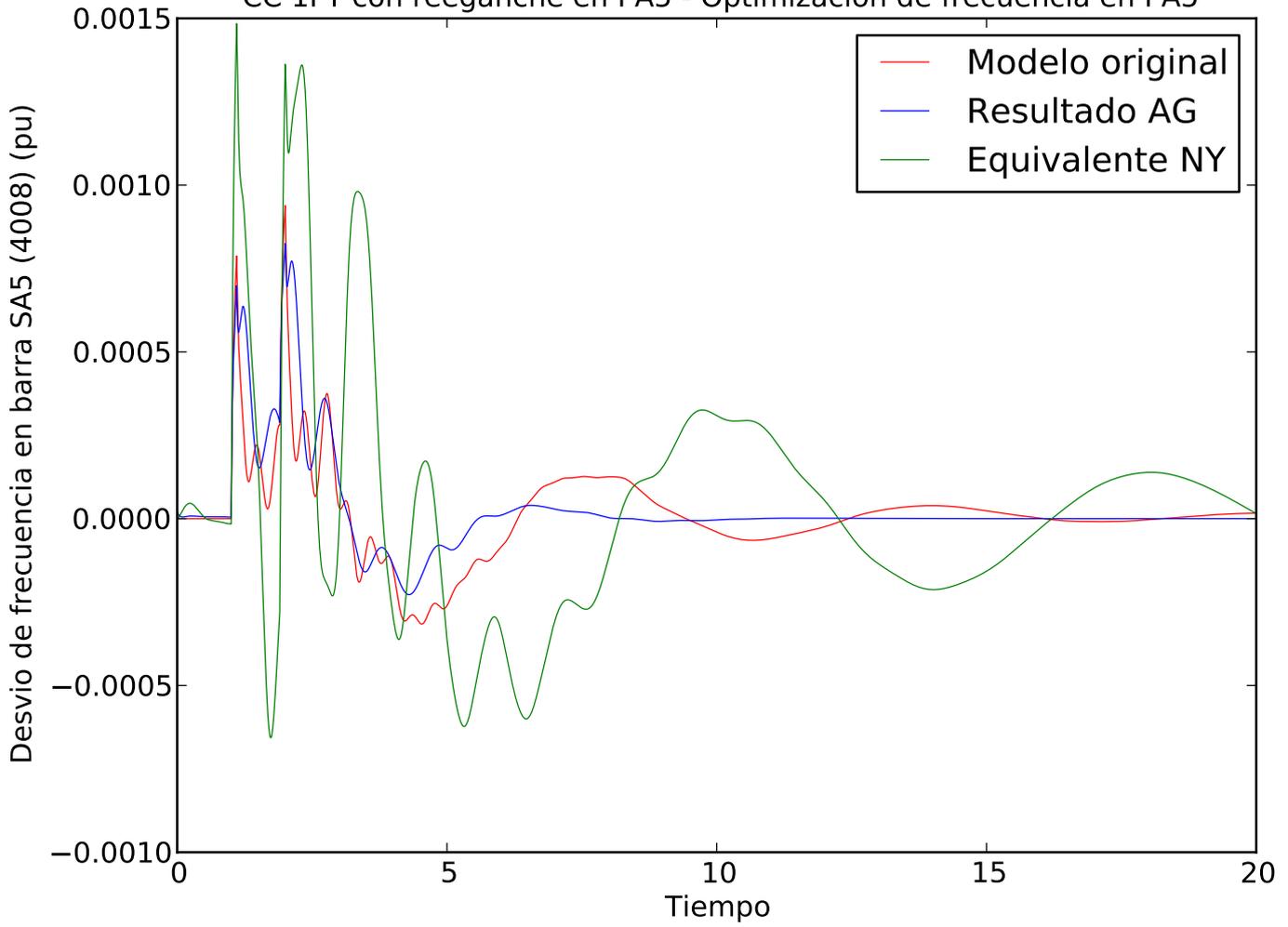


CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5

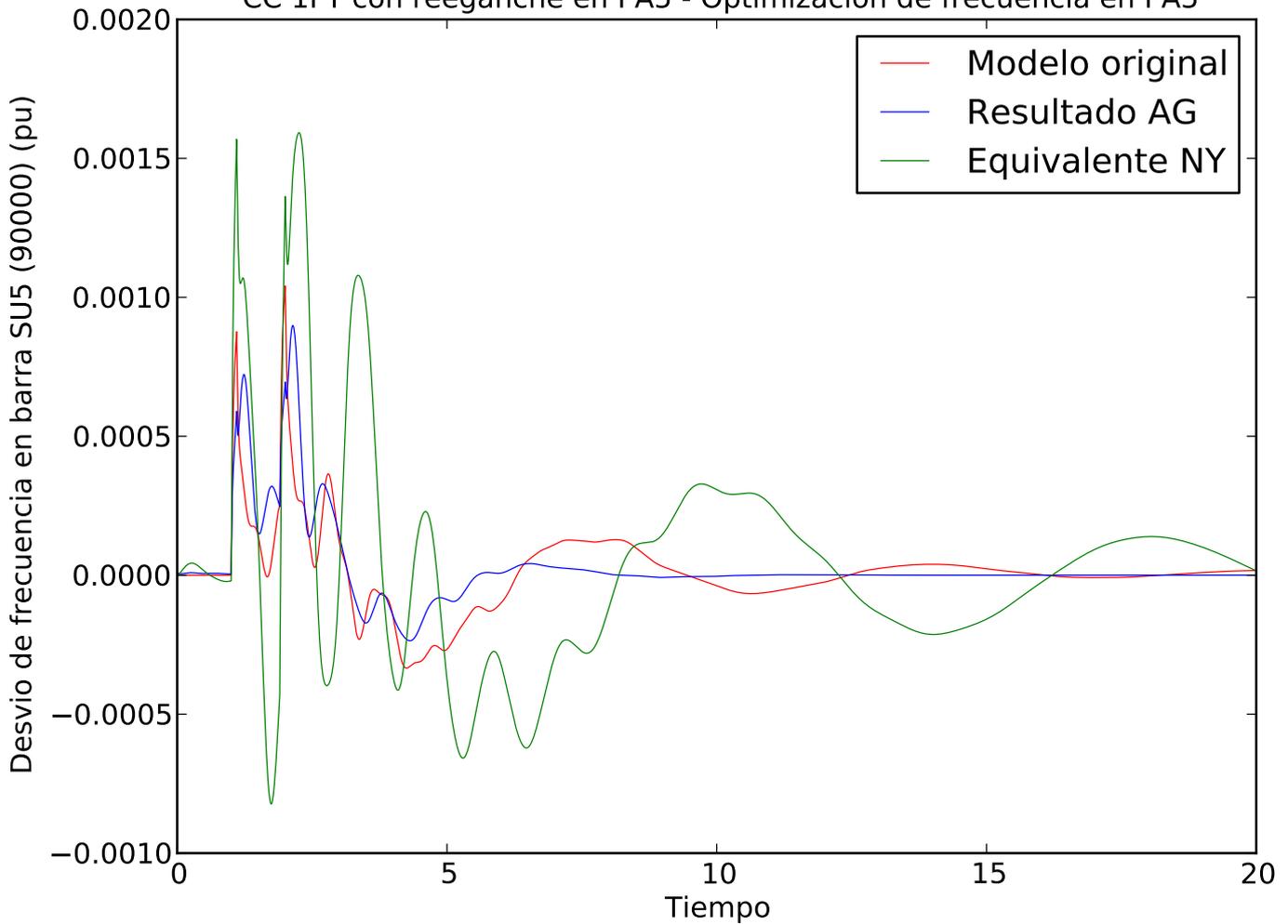


- C.11. **Simulación 7: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando frecuencia en PA5 en 4 segundos. Comportamiento en 20 segundos**

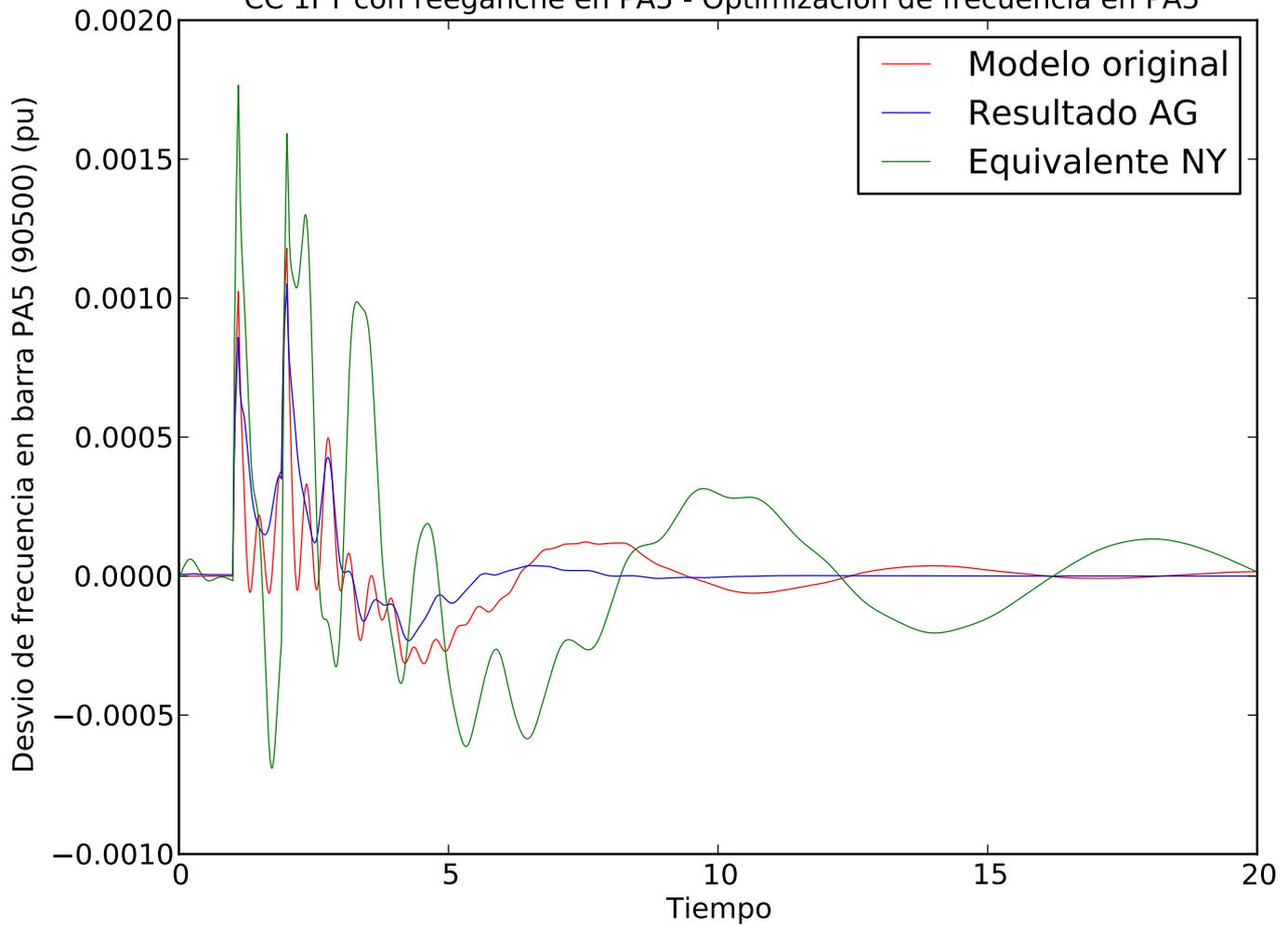
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



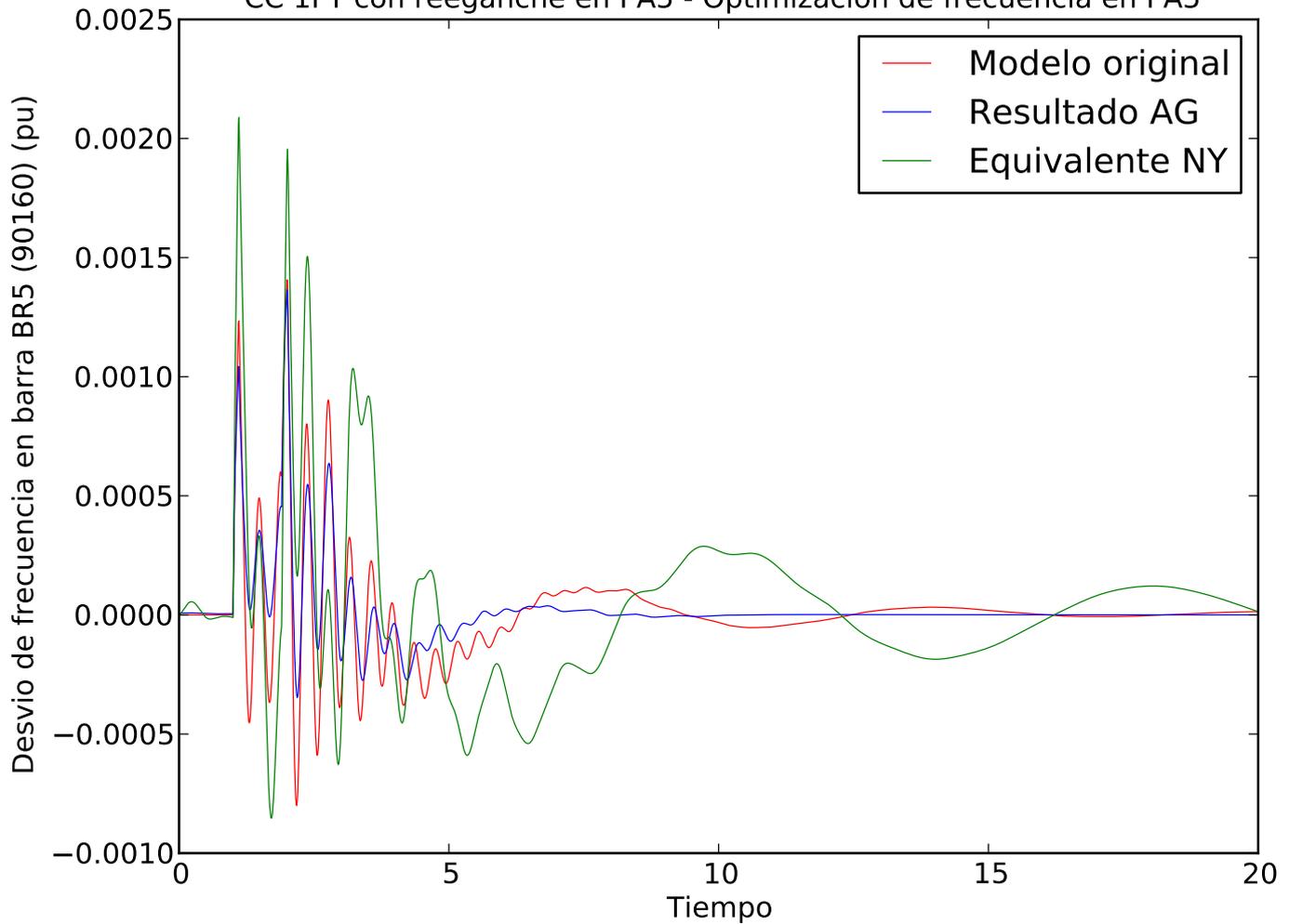
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



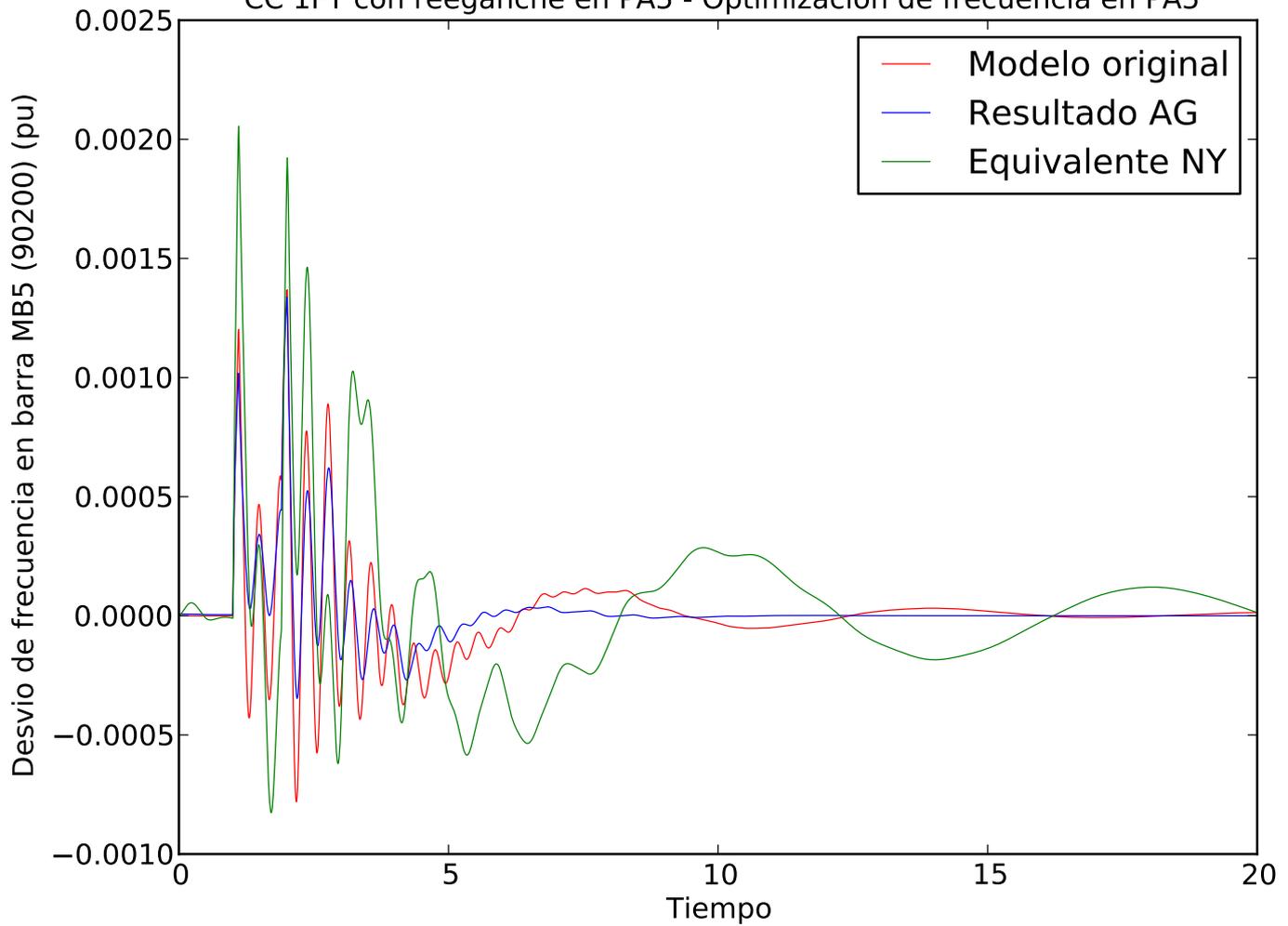
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



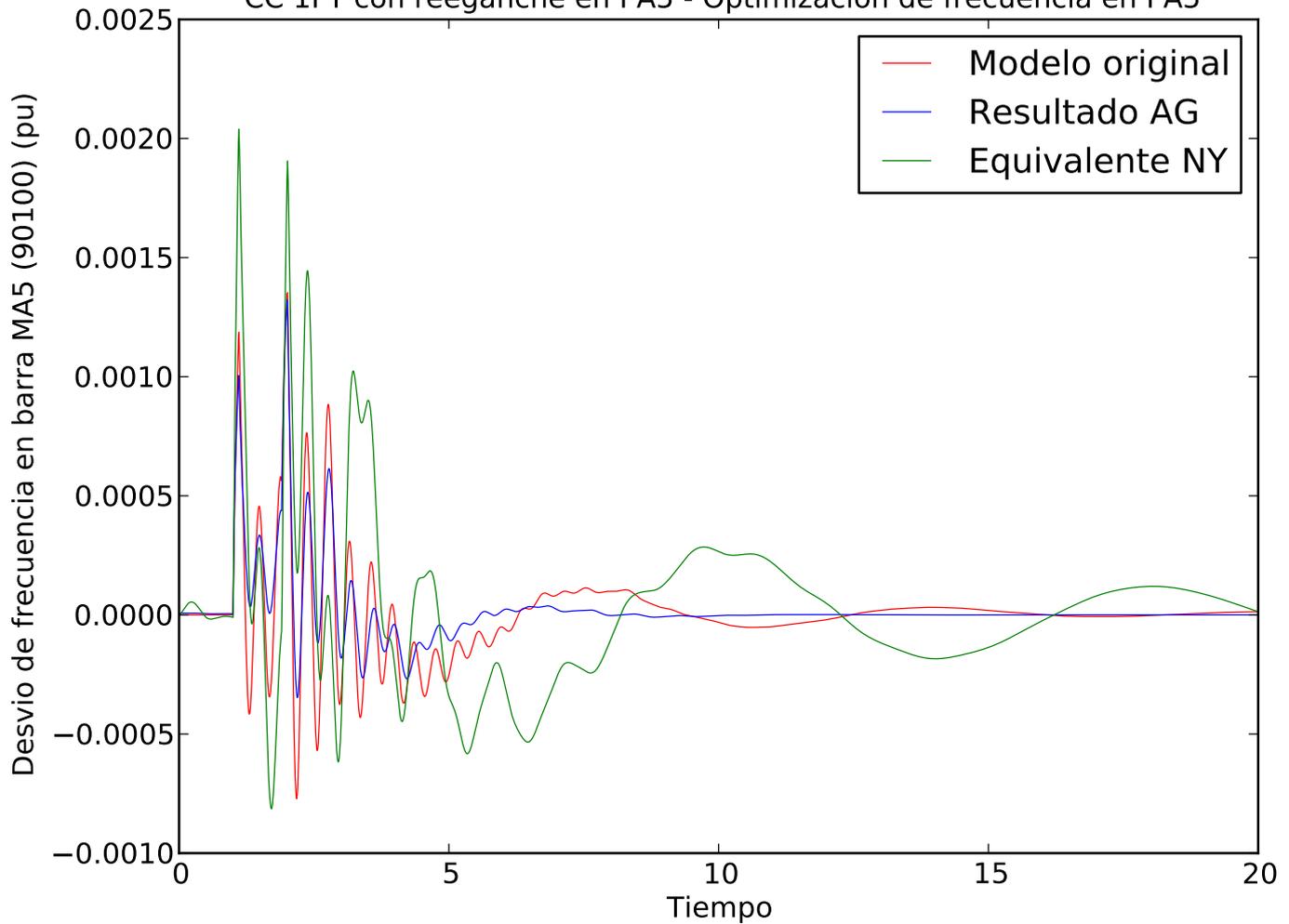
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



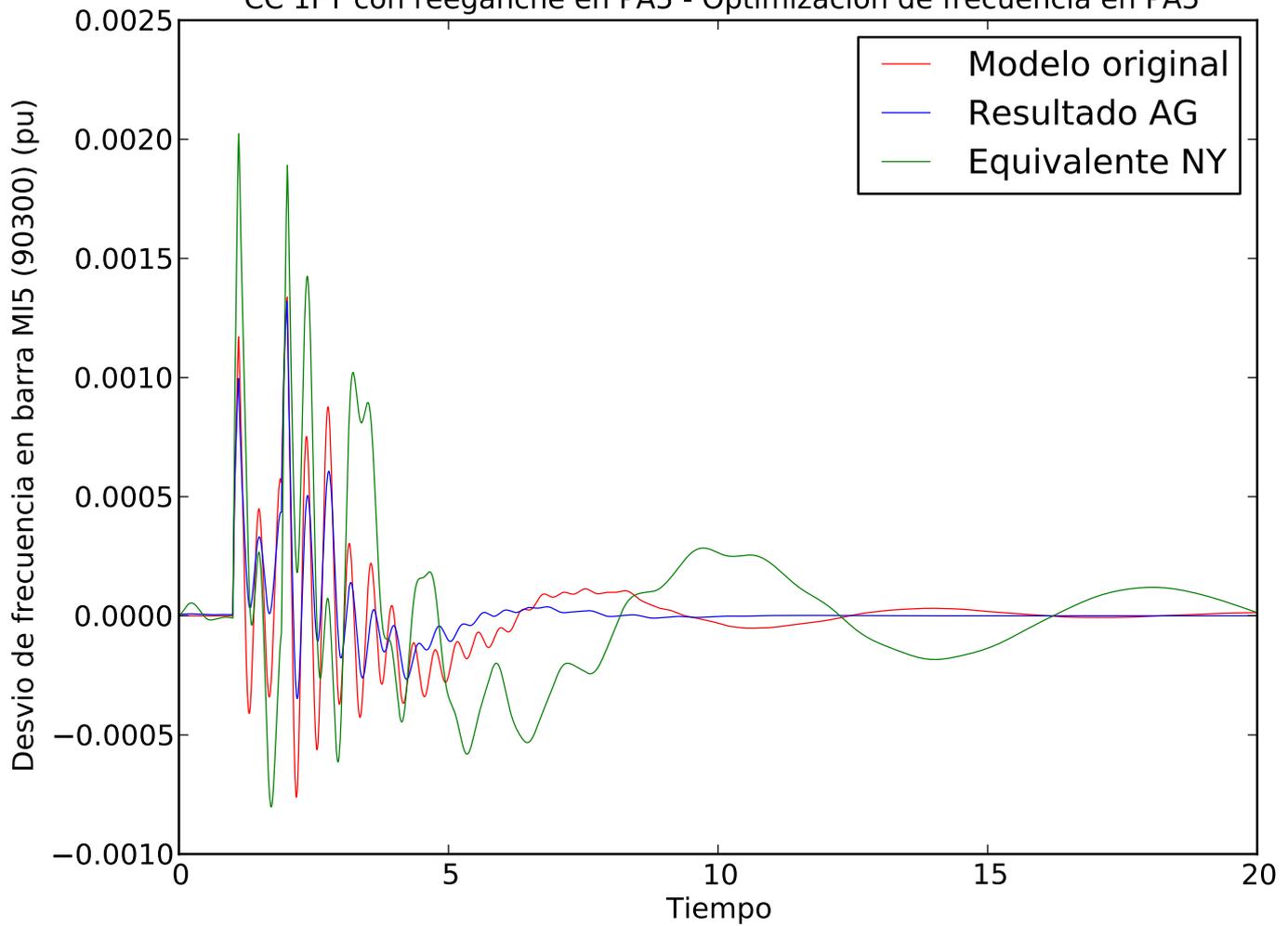
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



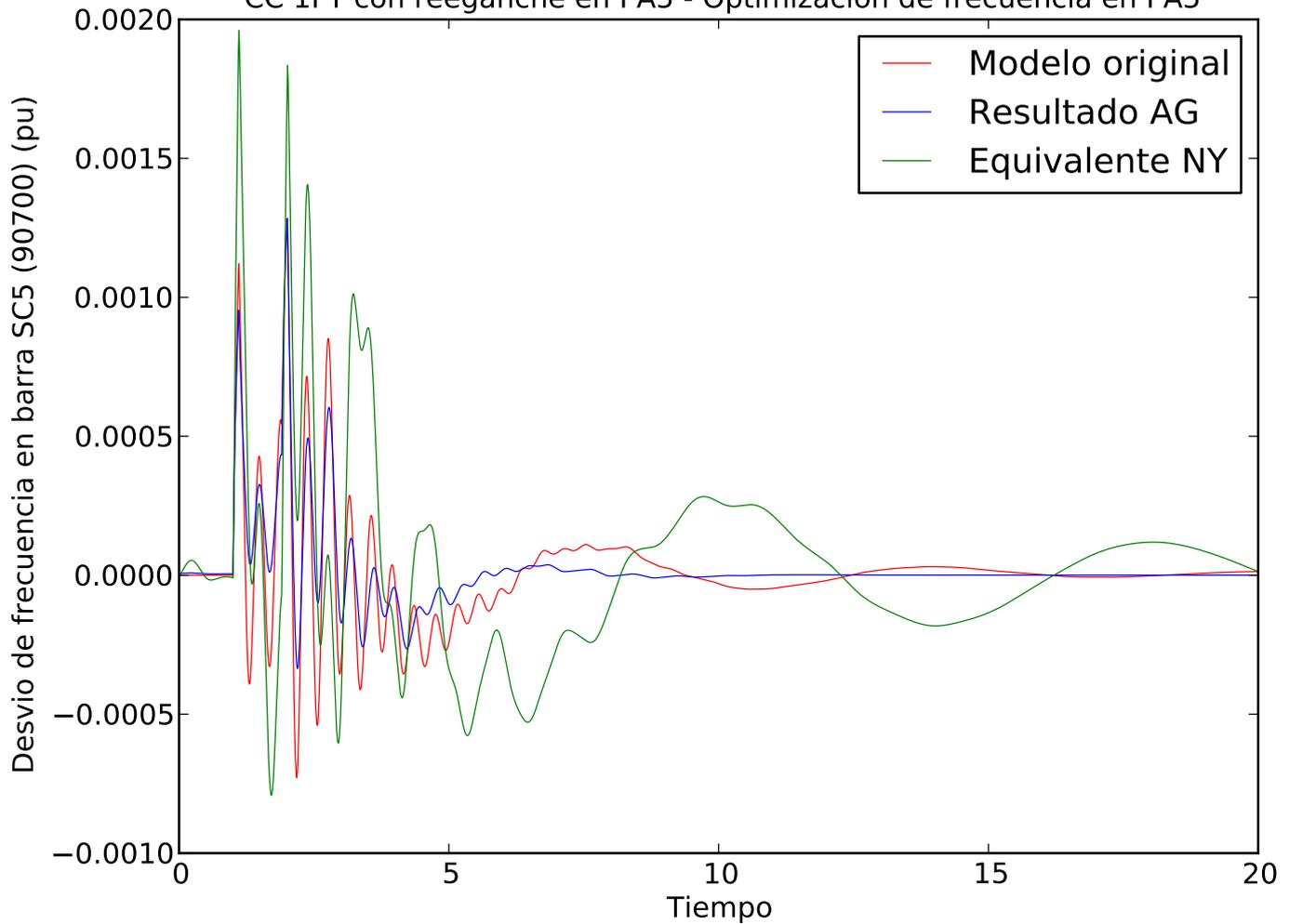
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



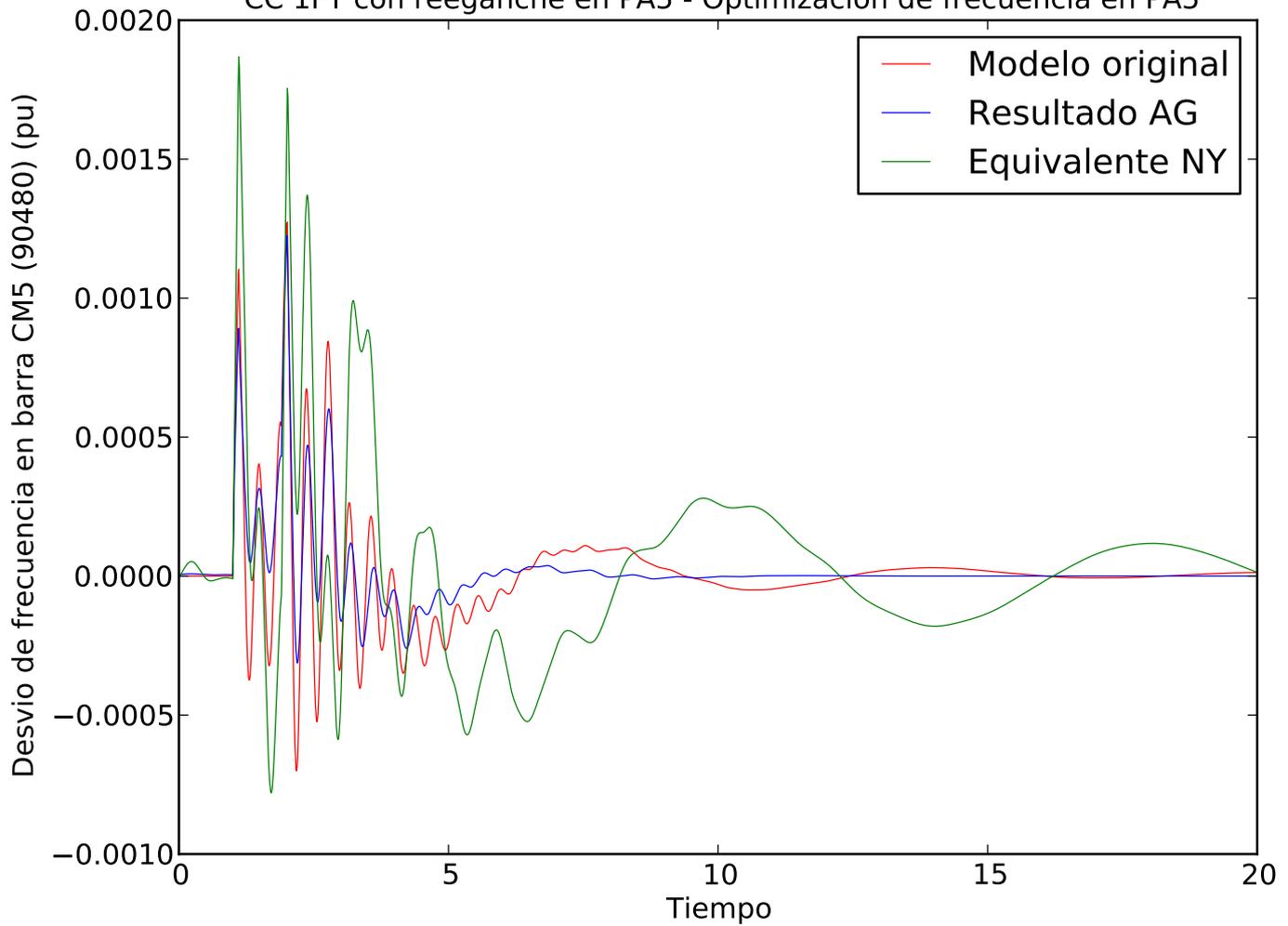
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



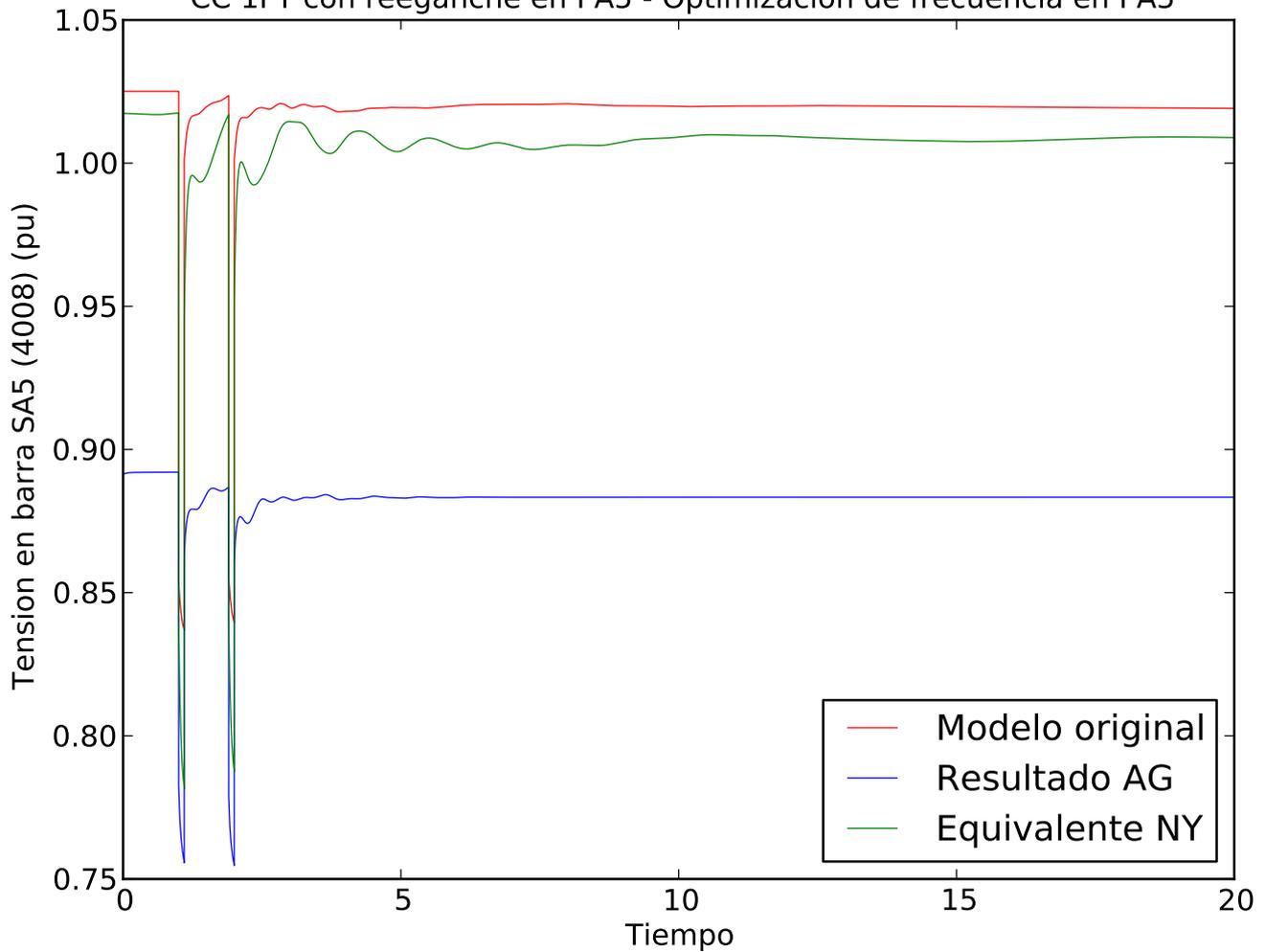
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



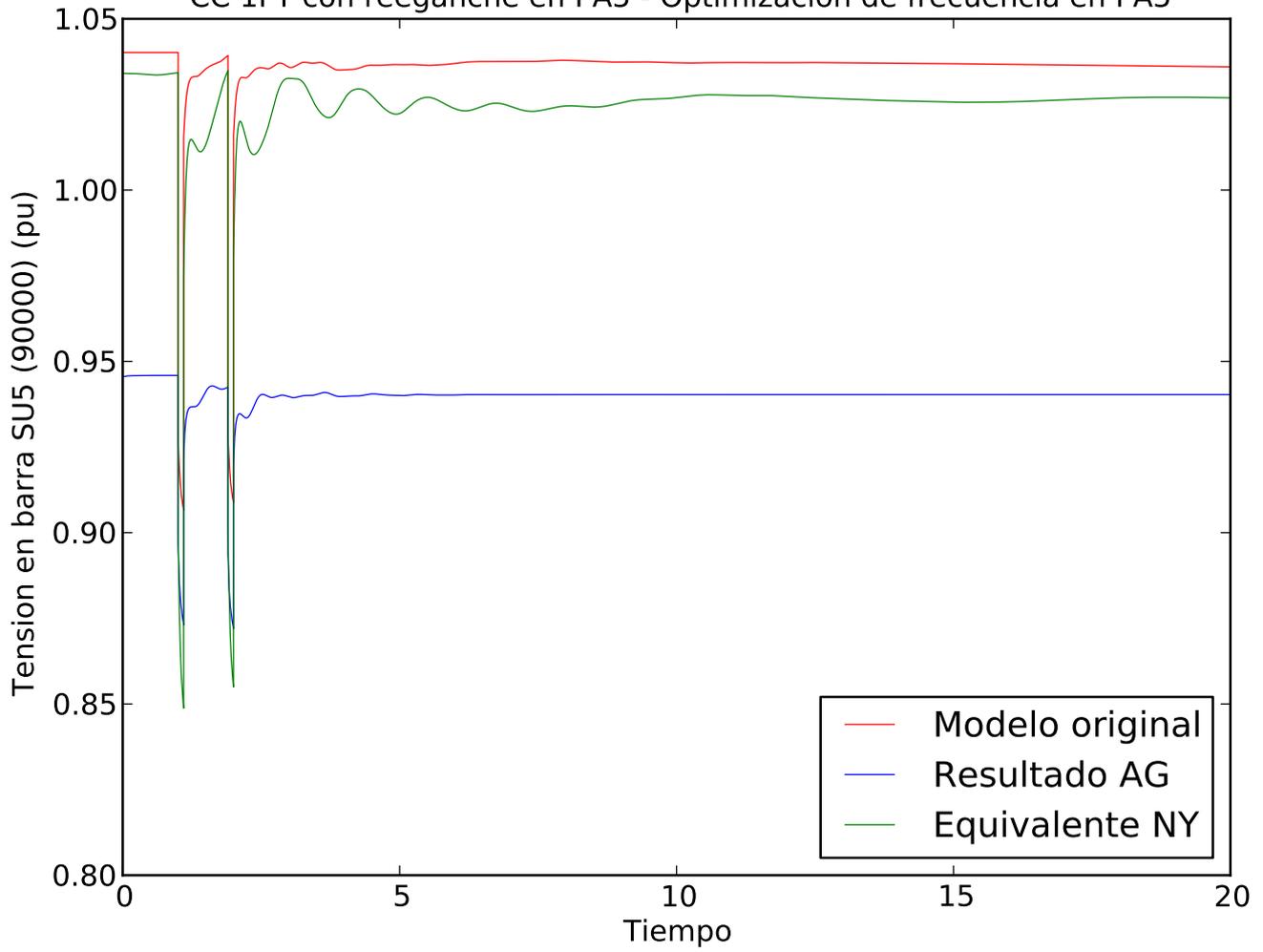
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



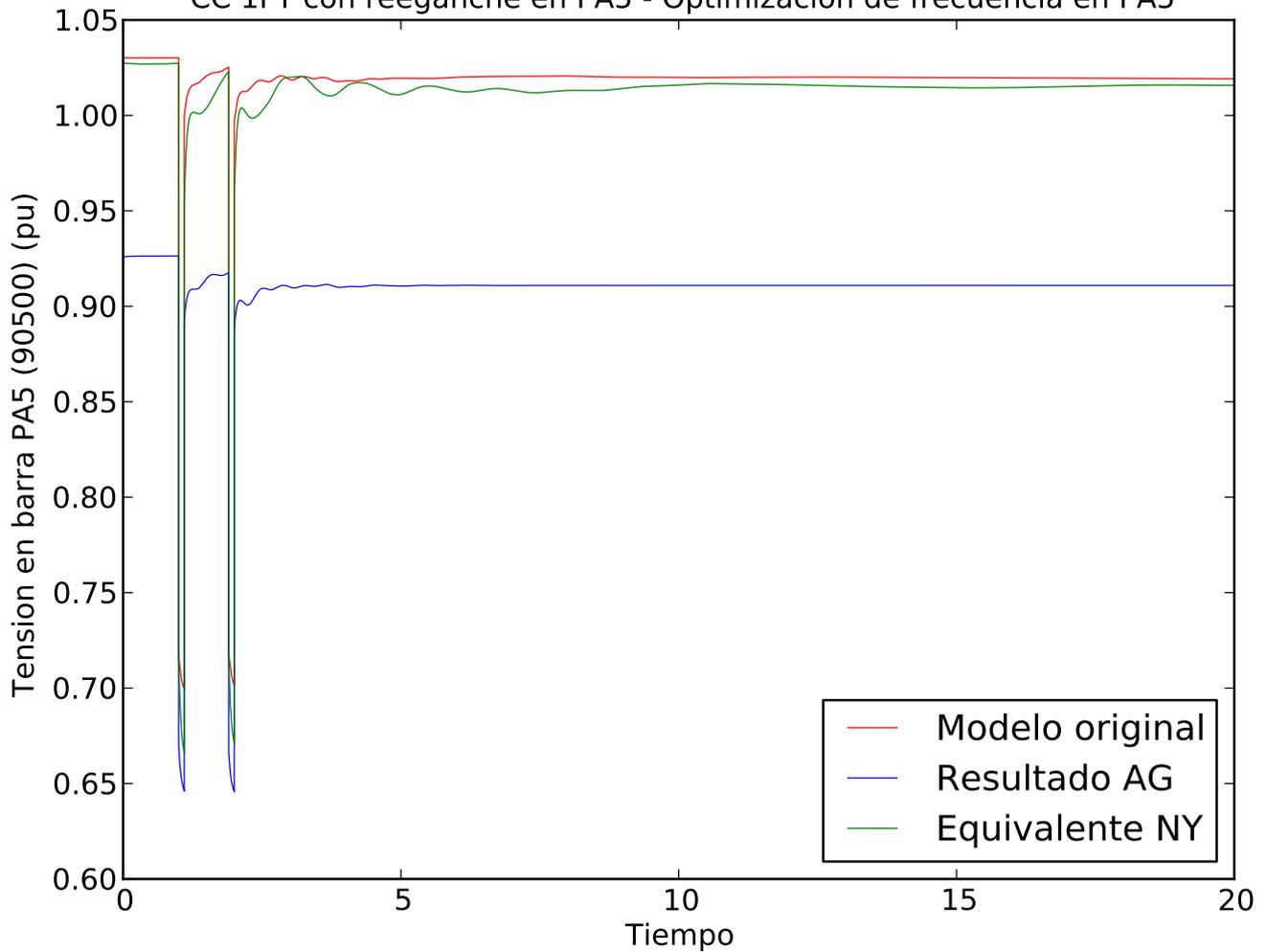
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



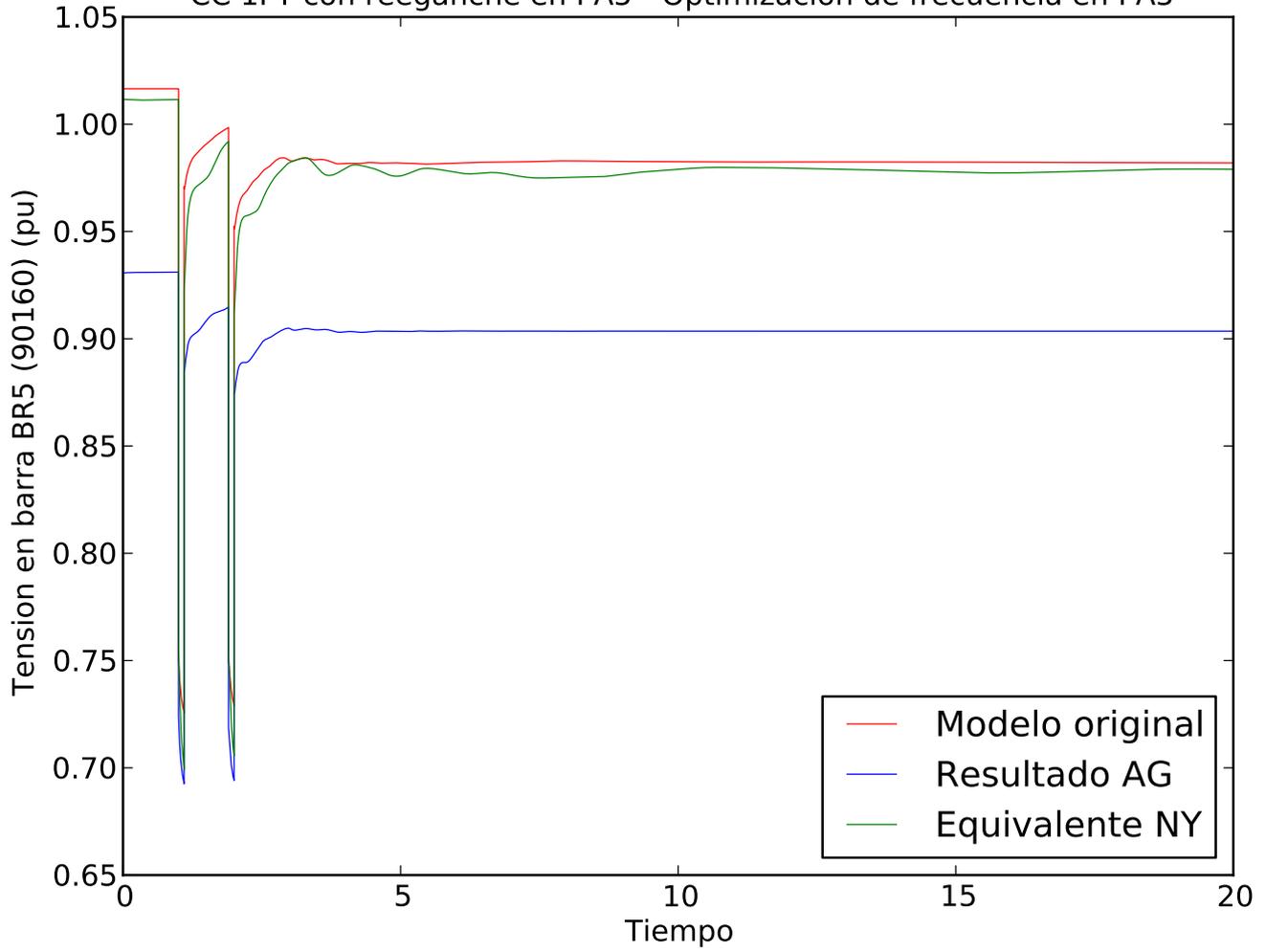
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



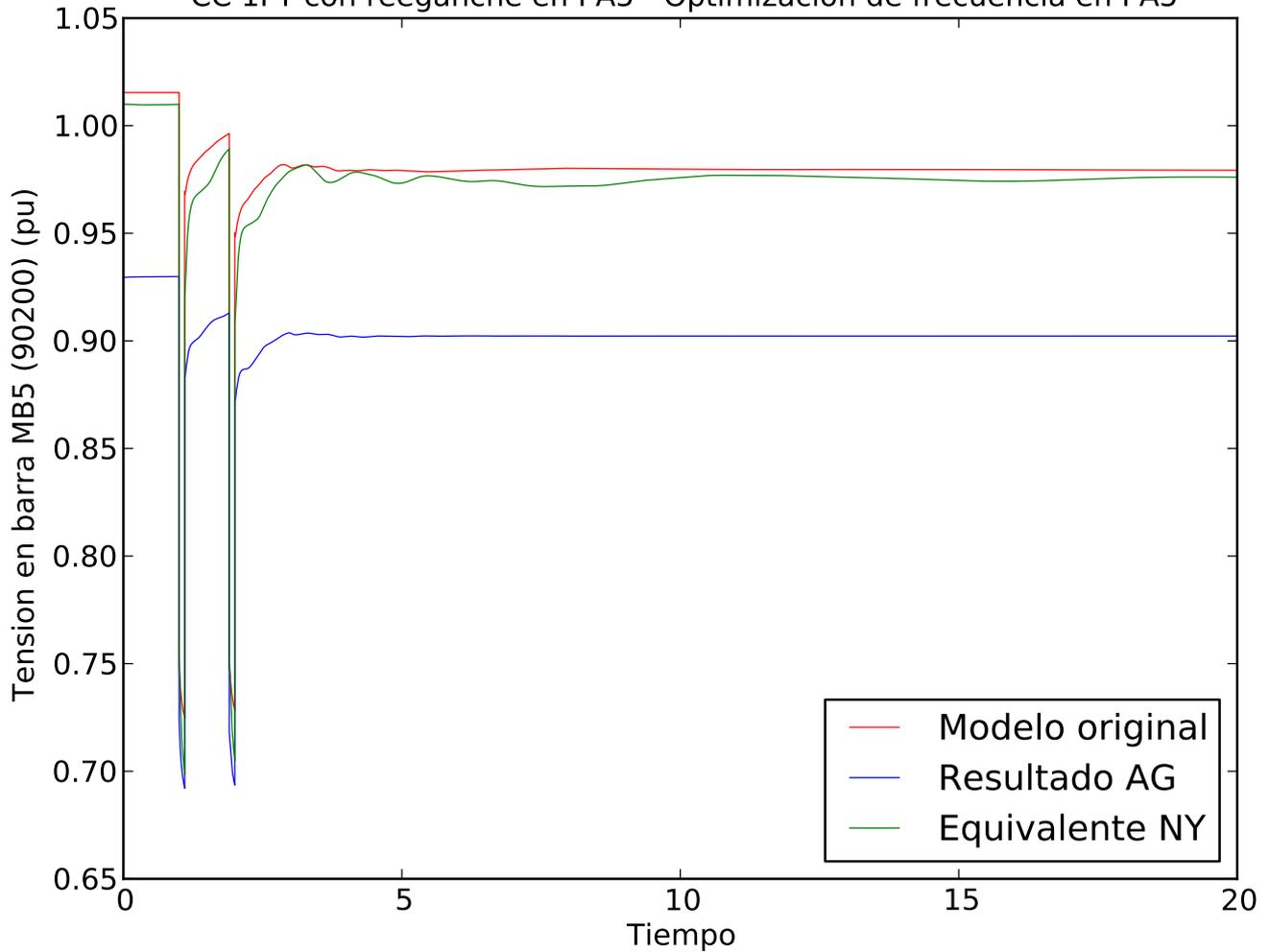
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



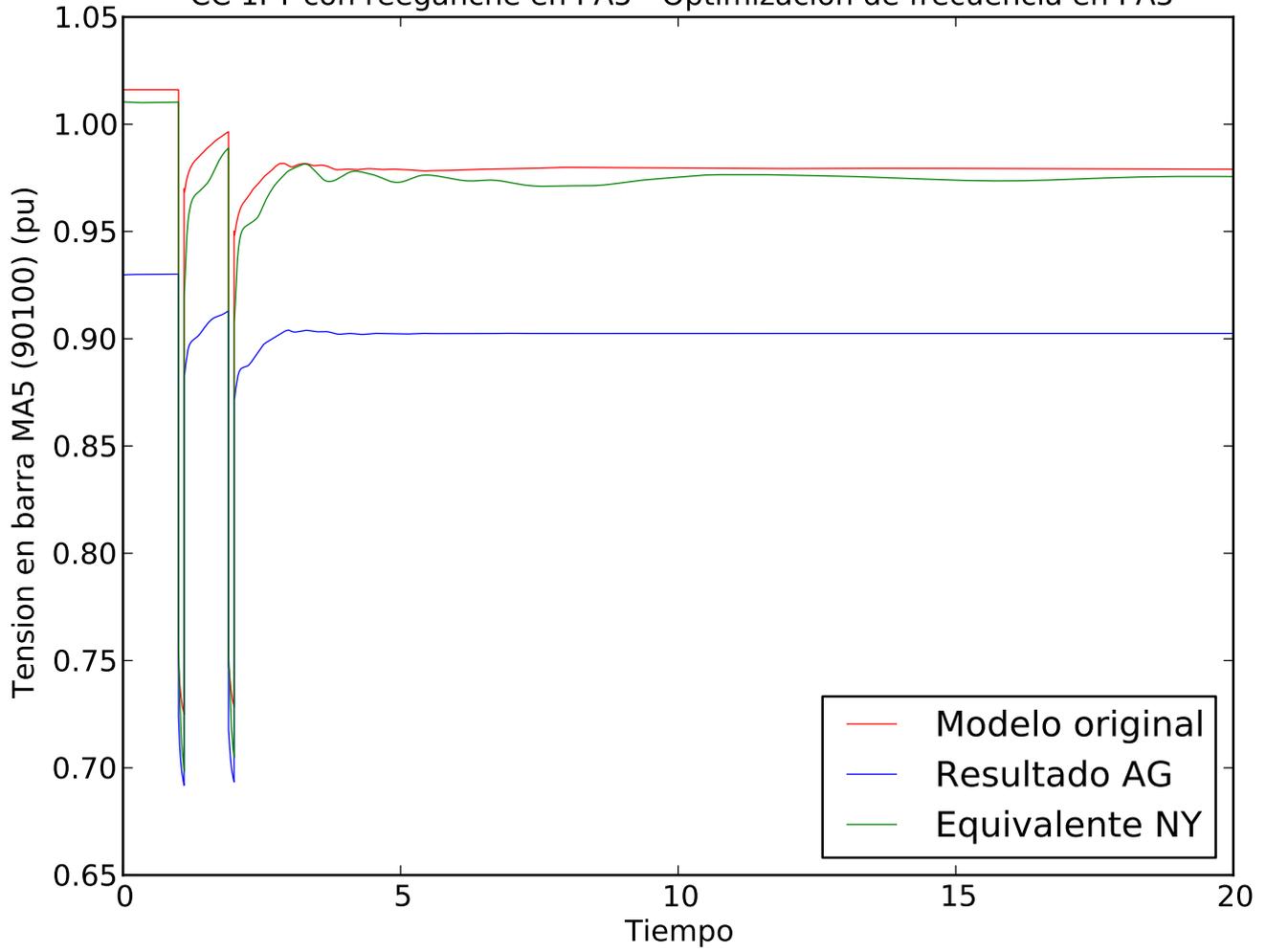
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



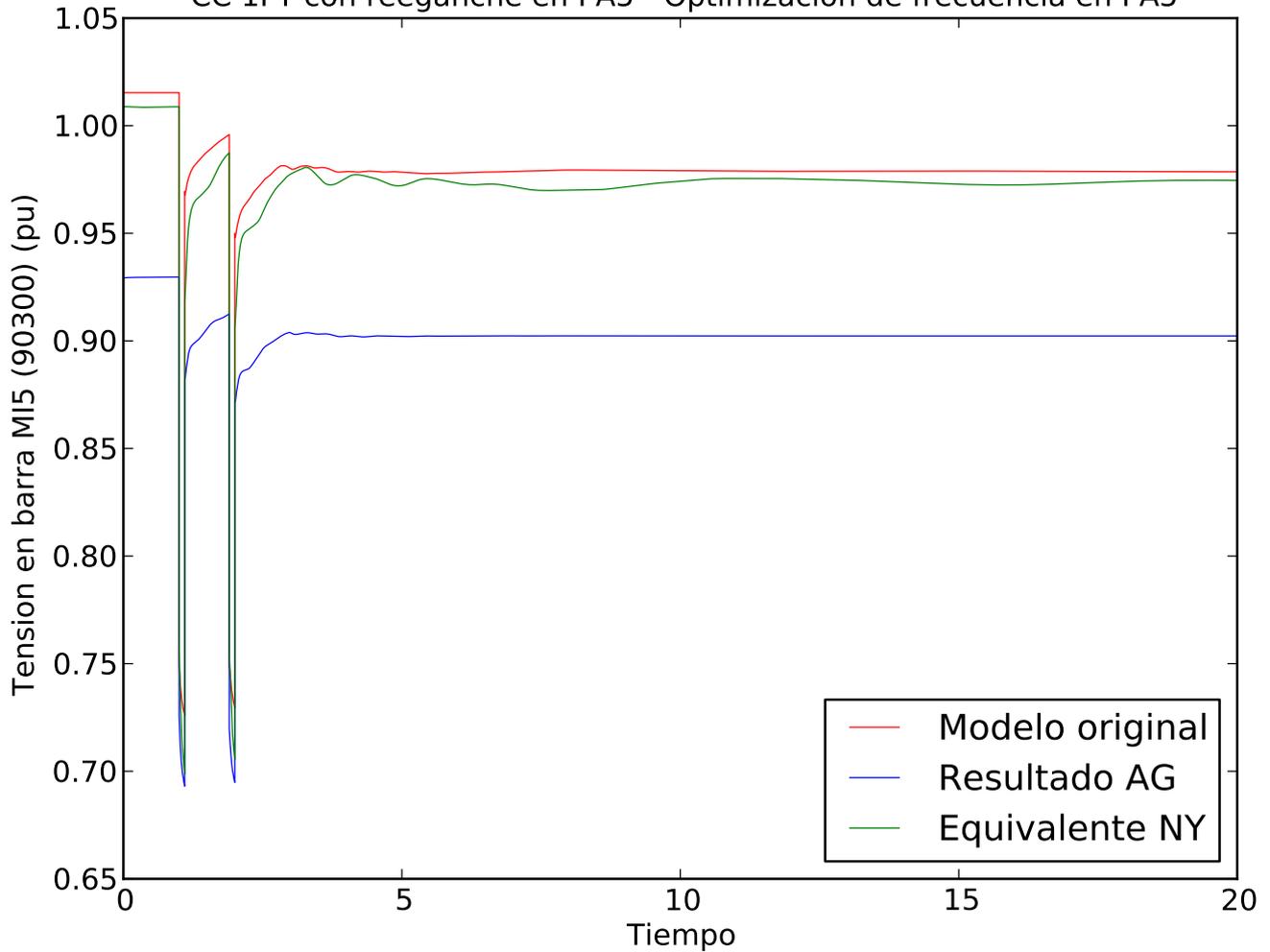
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



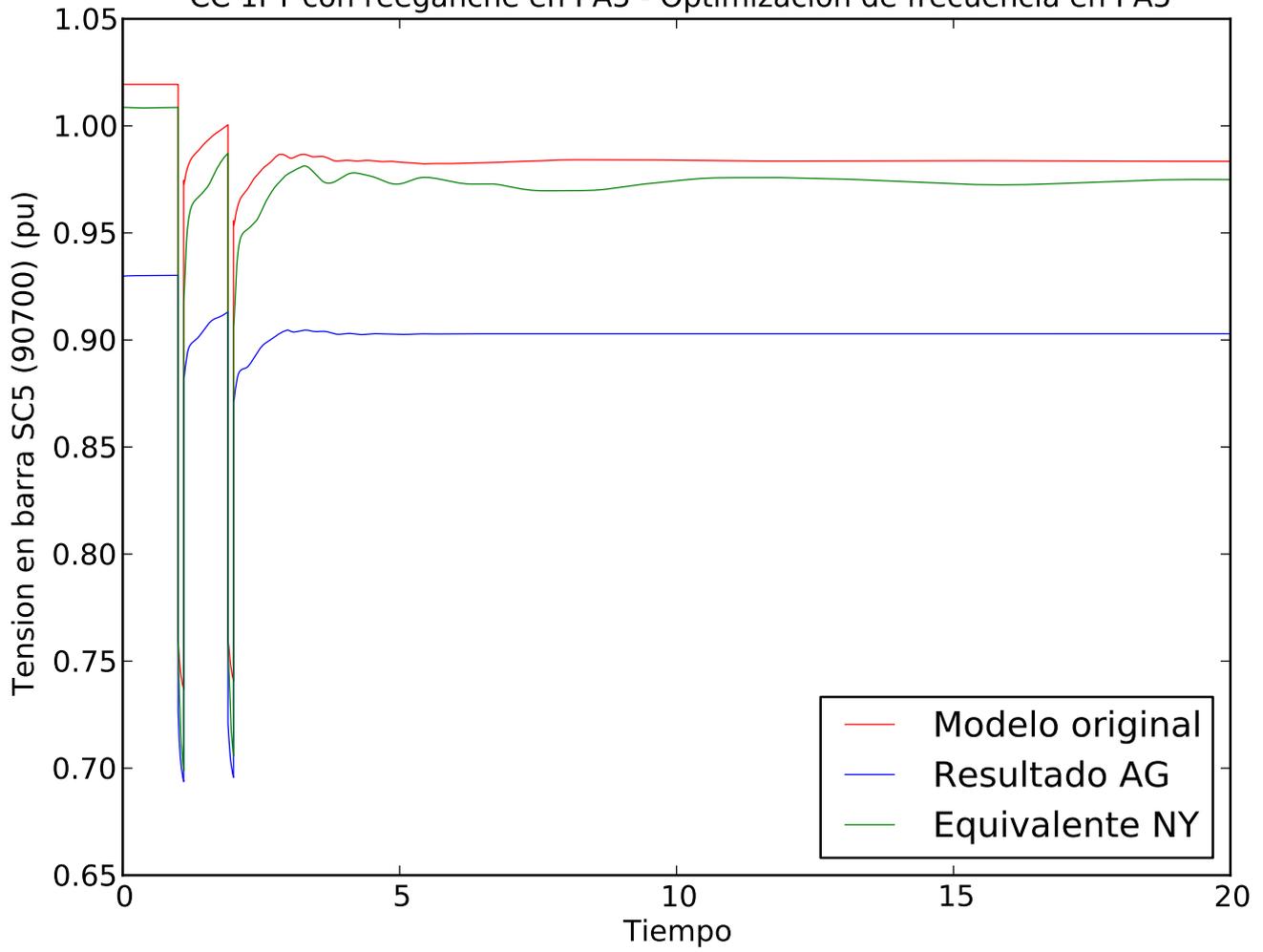
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



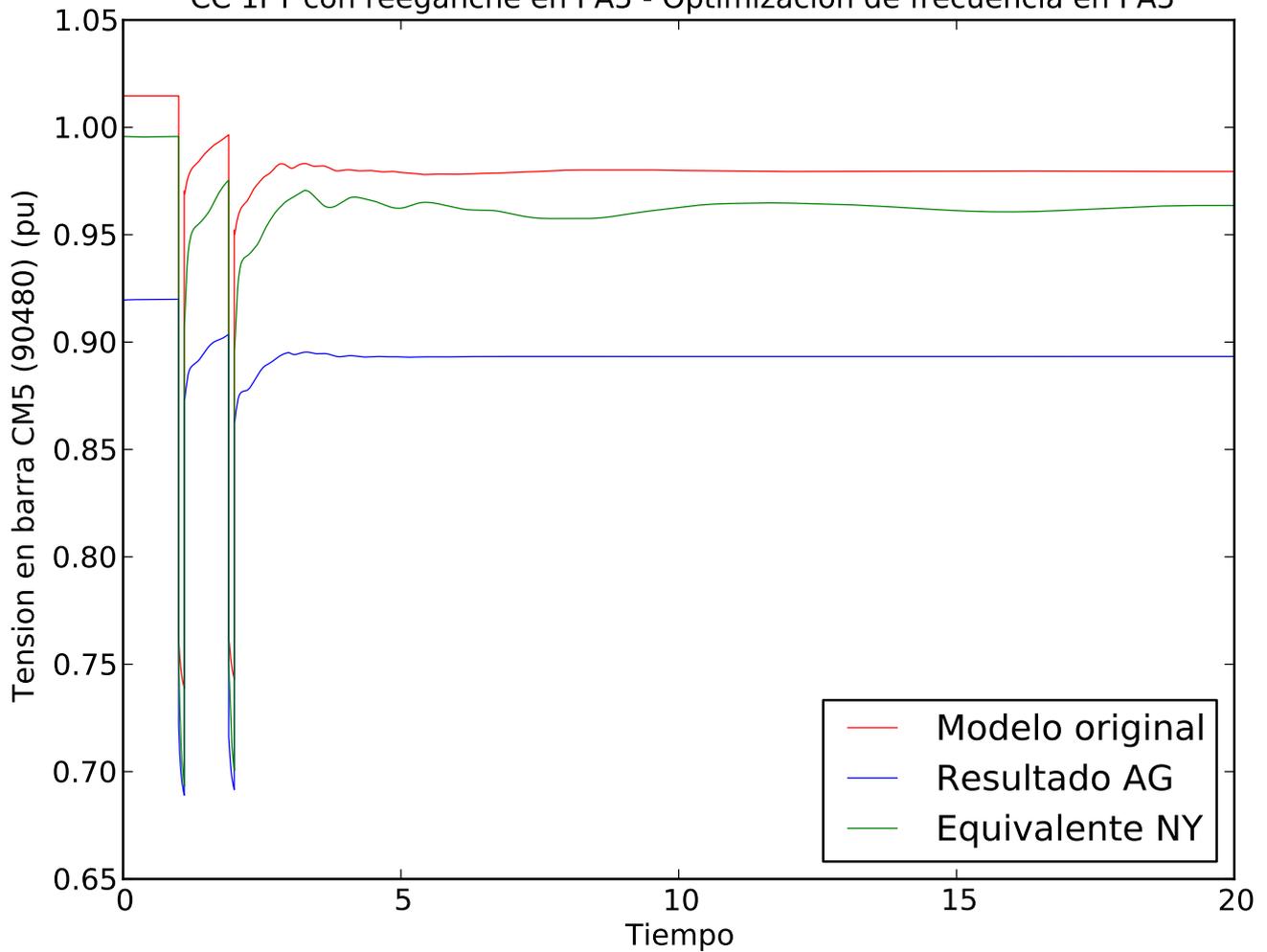
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



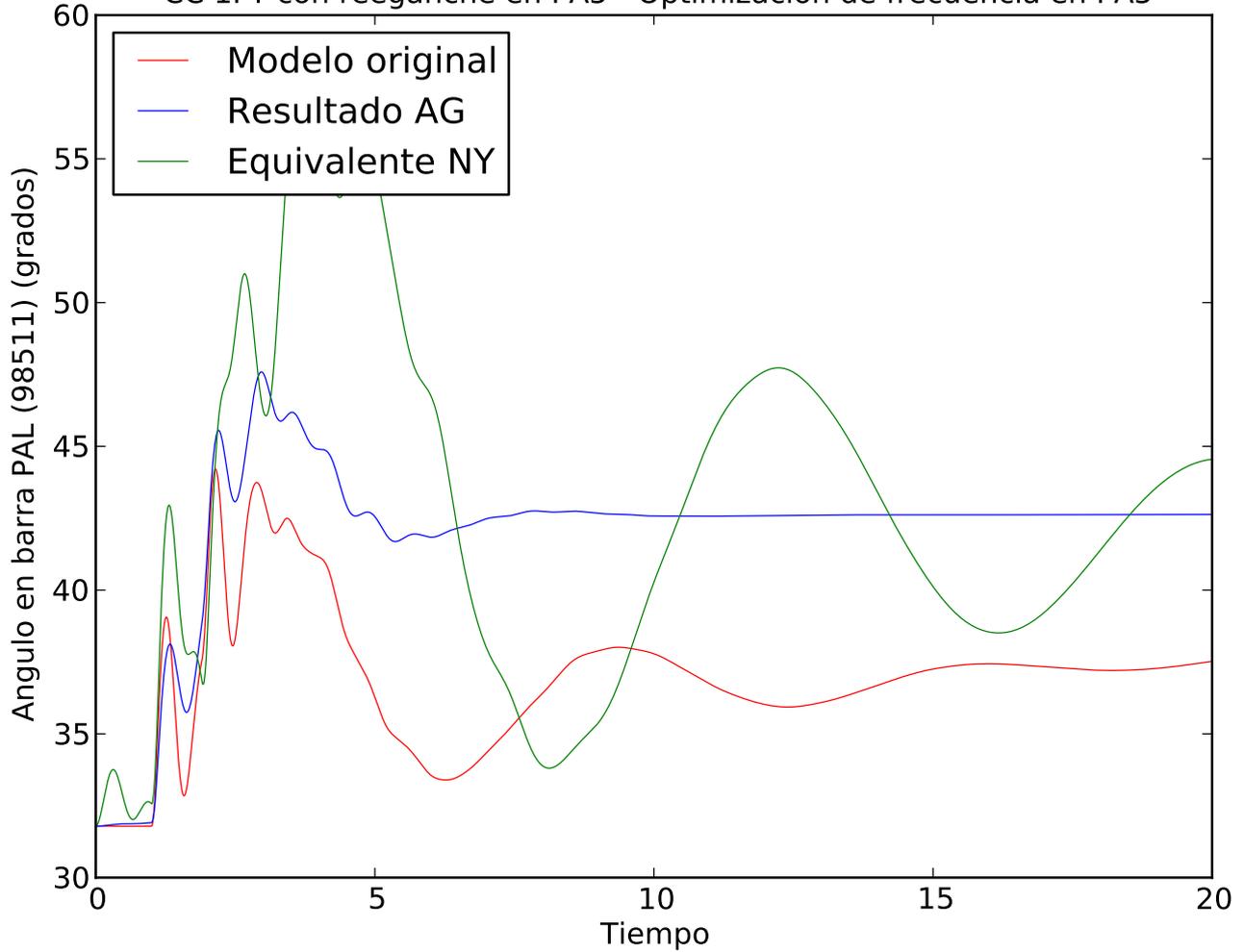
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



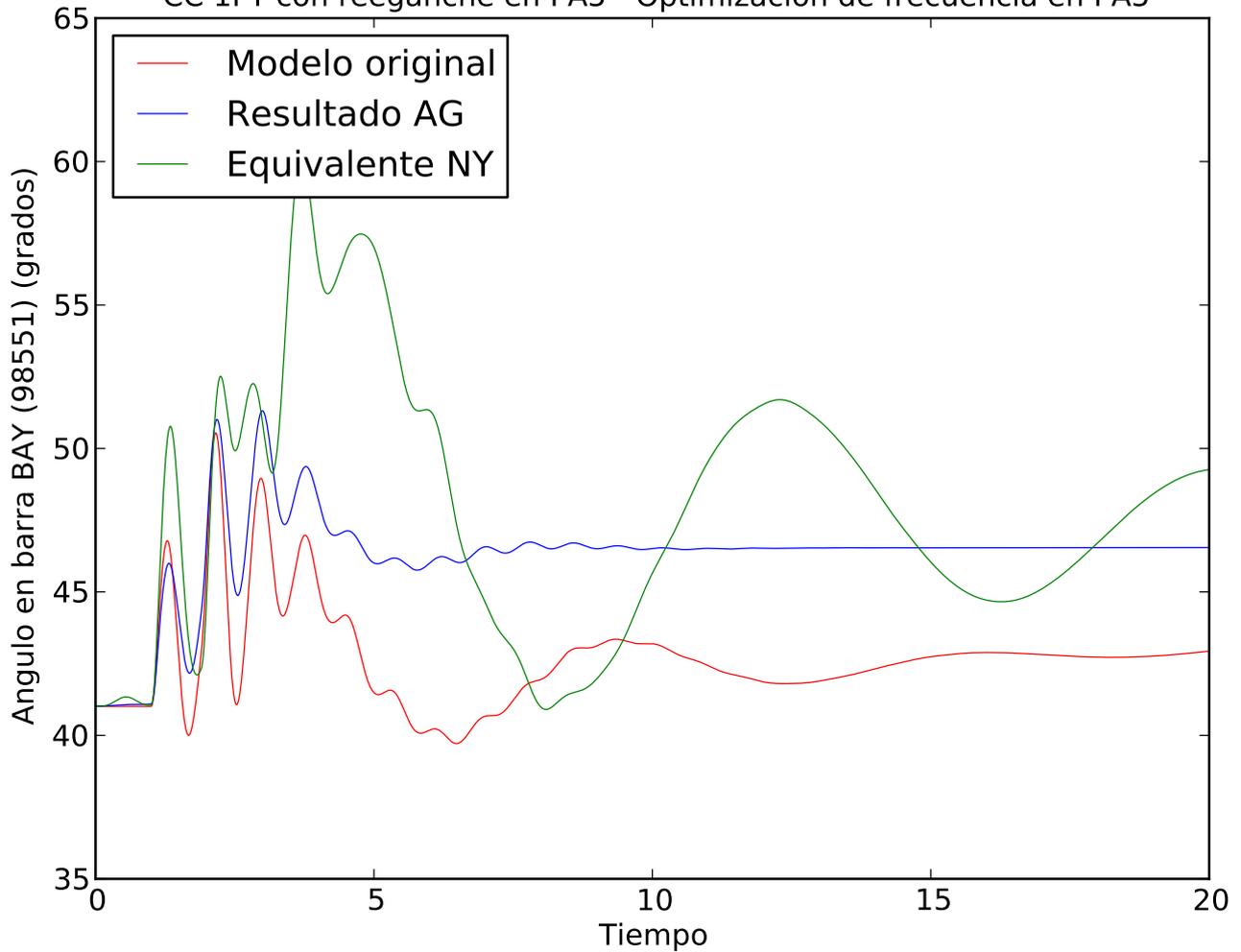
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



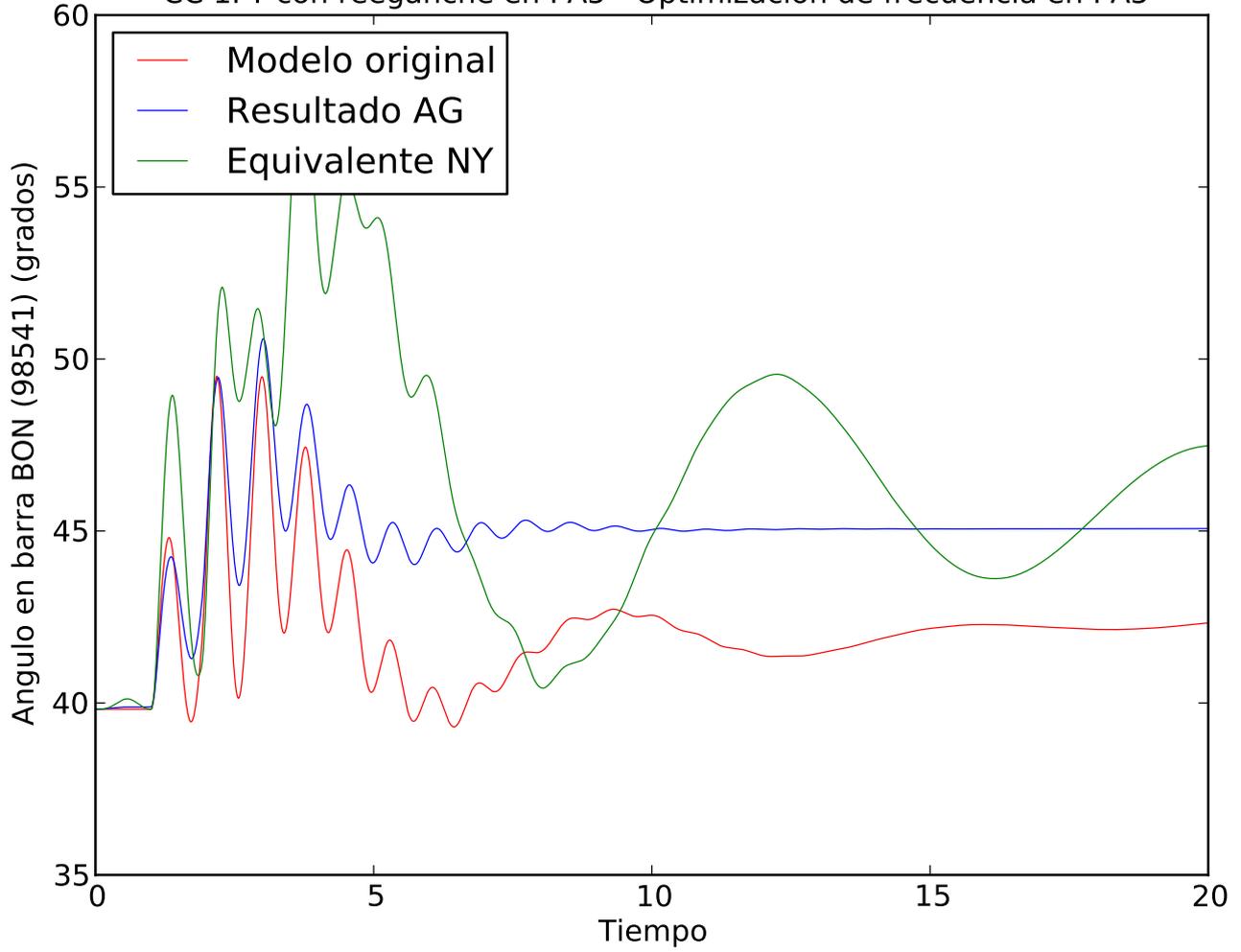
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



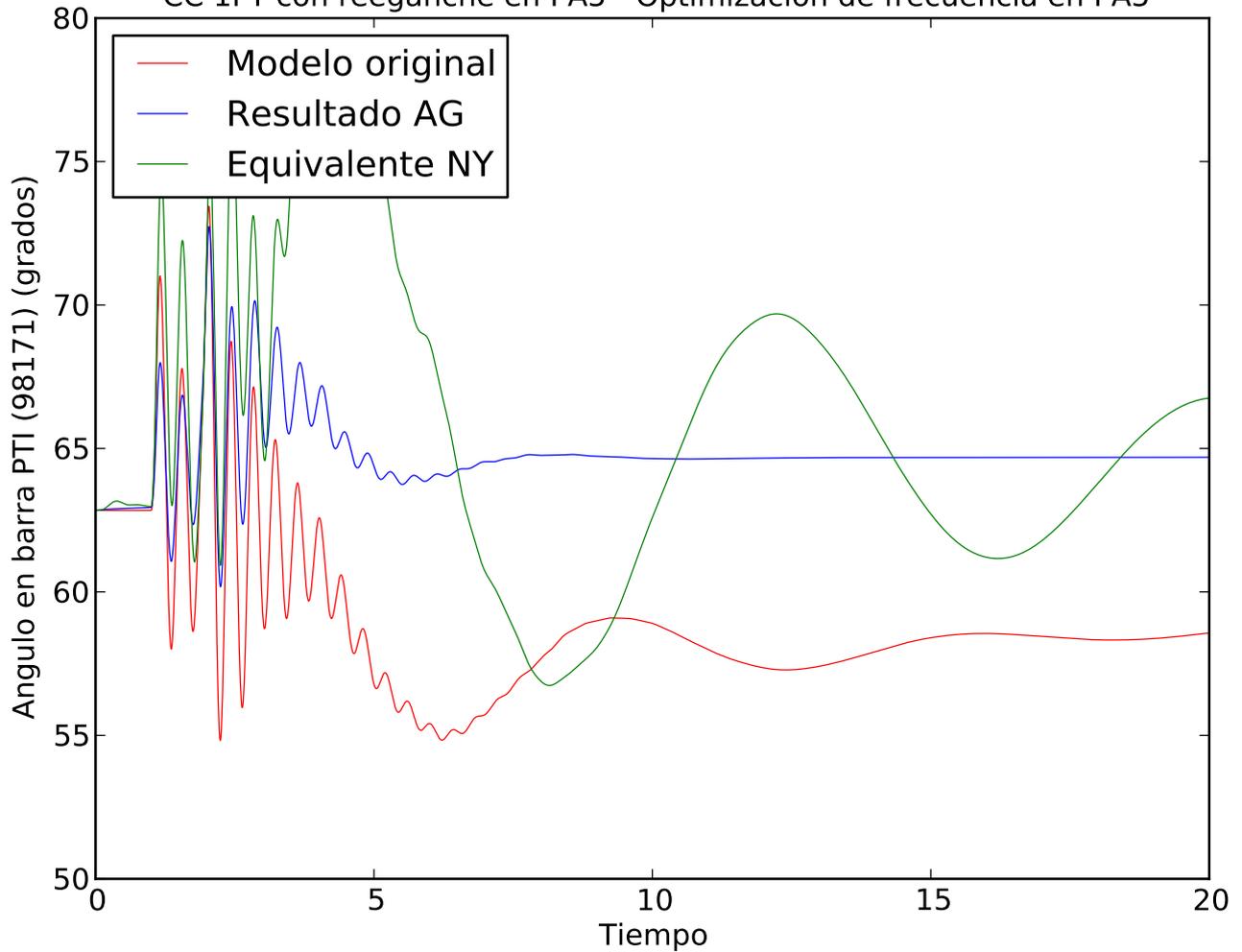
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5



CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5

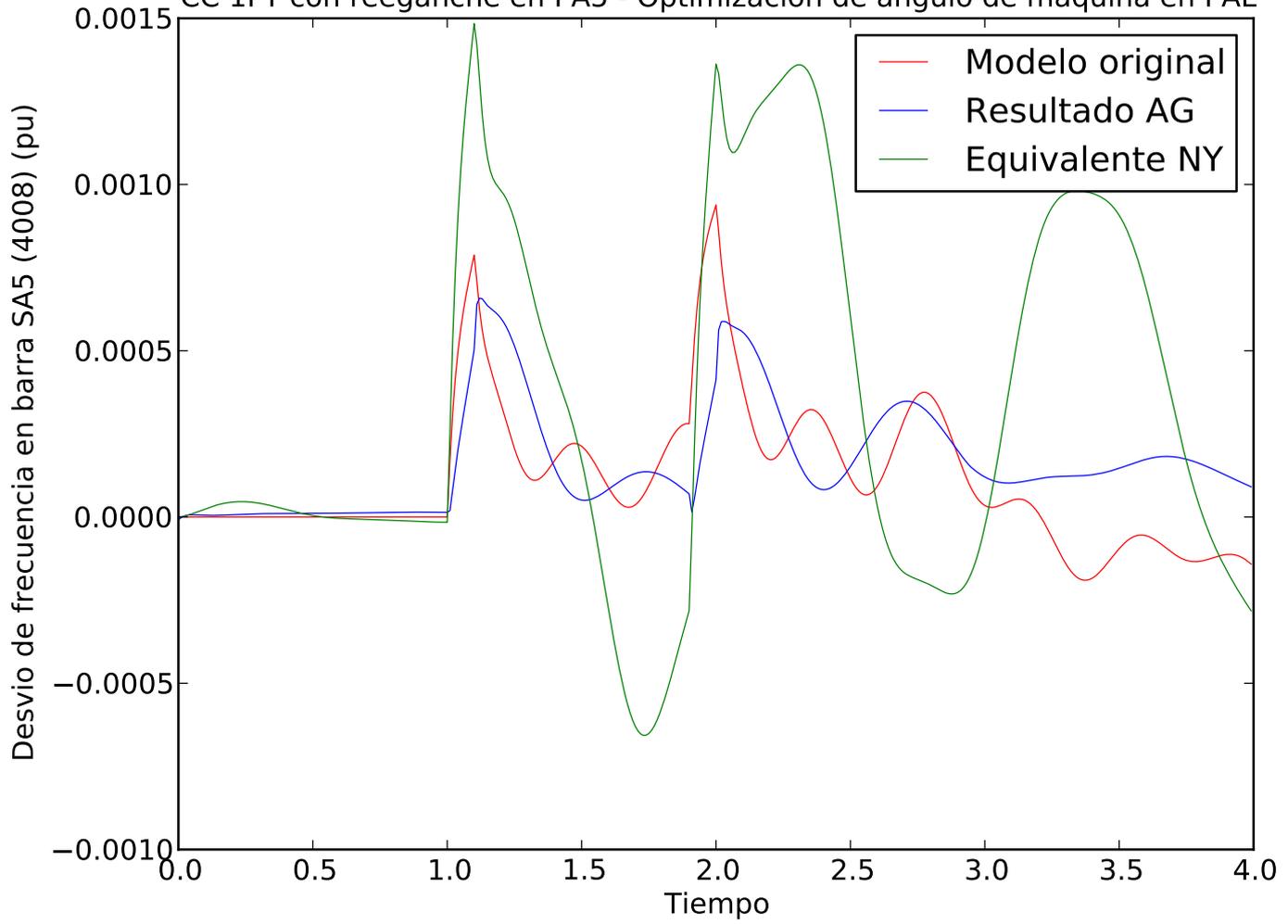


CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en PA5

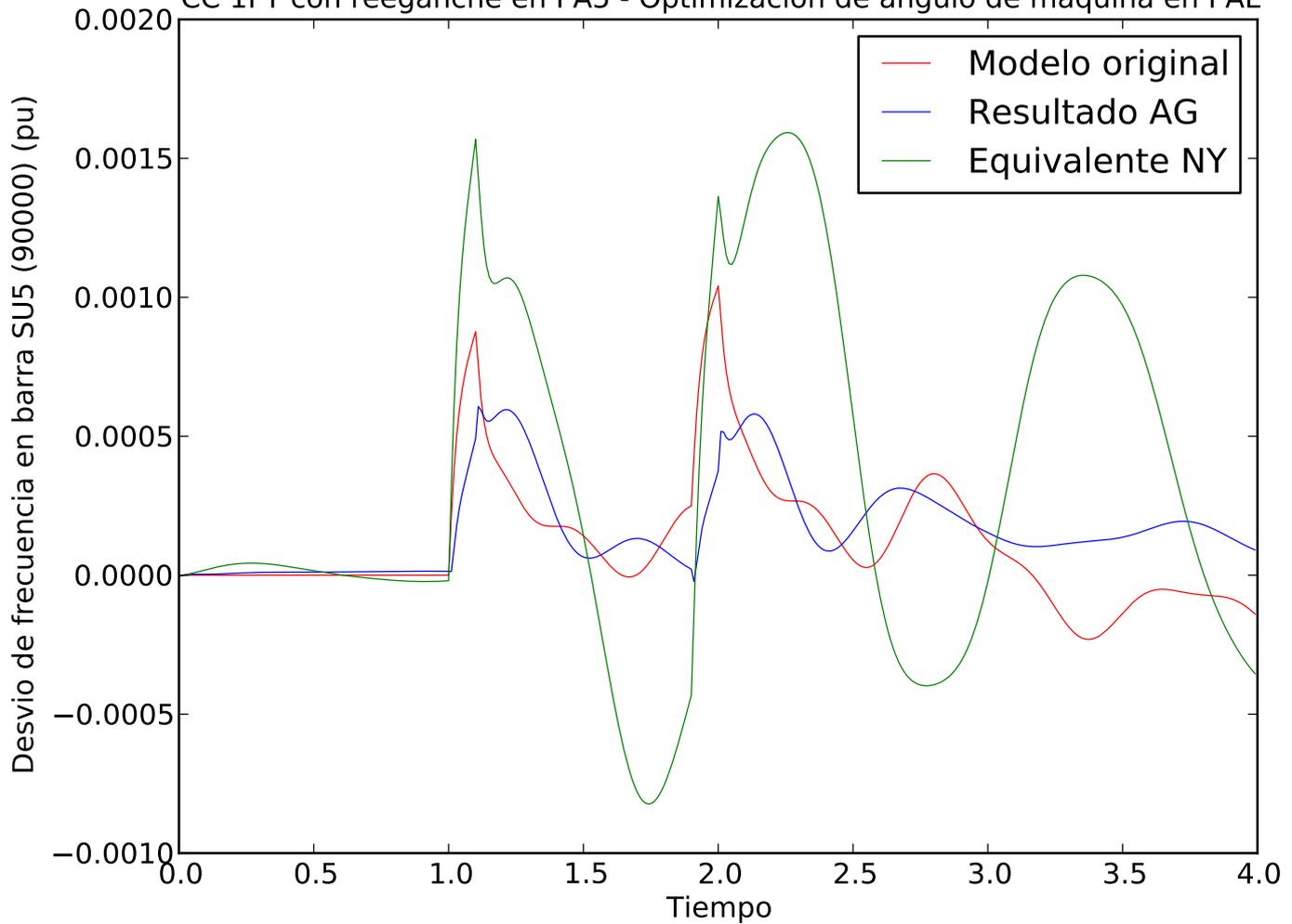


- C.12. Simulación 8: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando ángulo de un máquina de PAL en 4 segundos. Comportamiento en 4 segundos

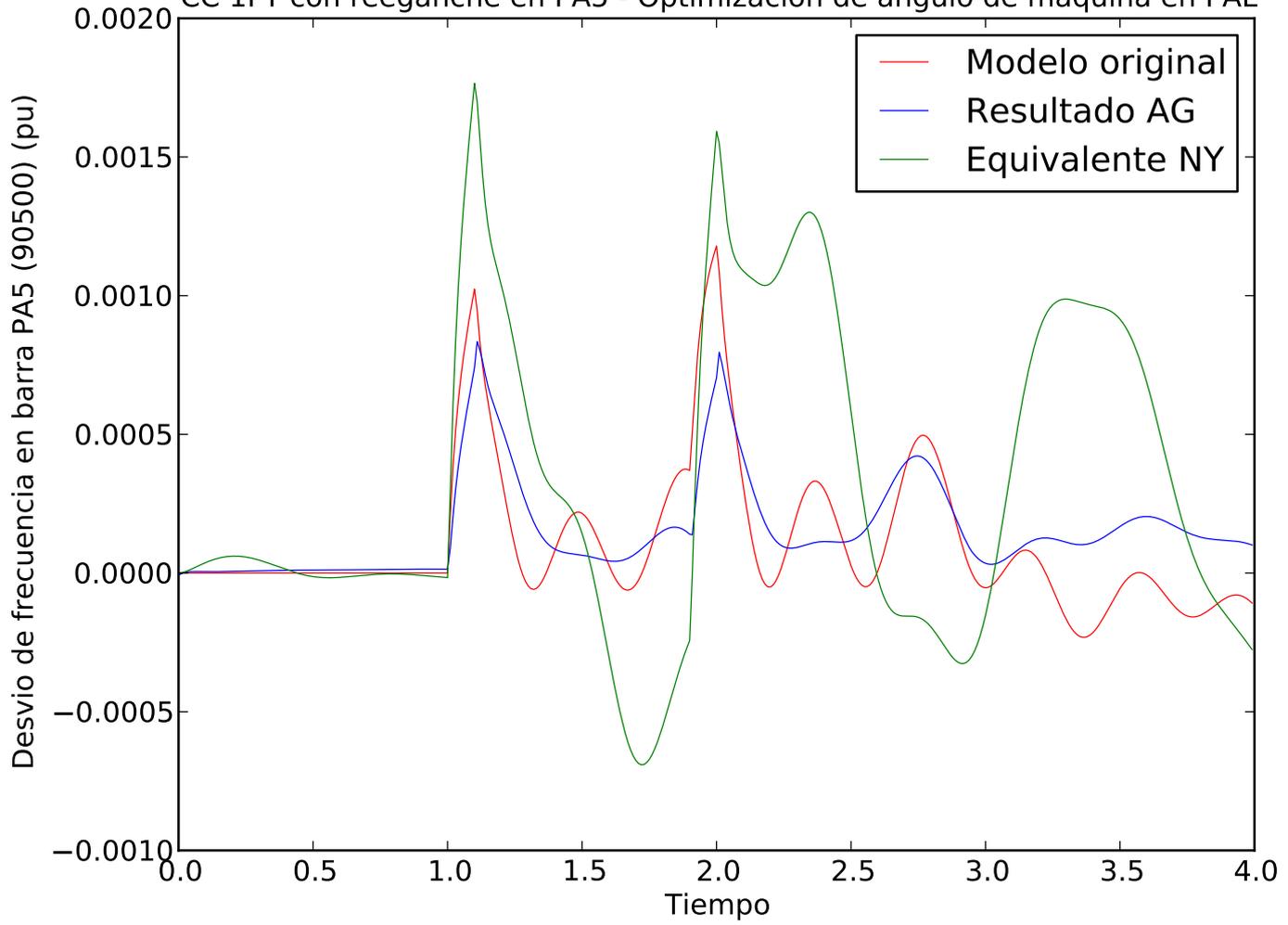
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



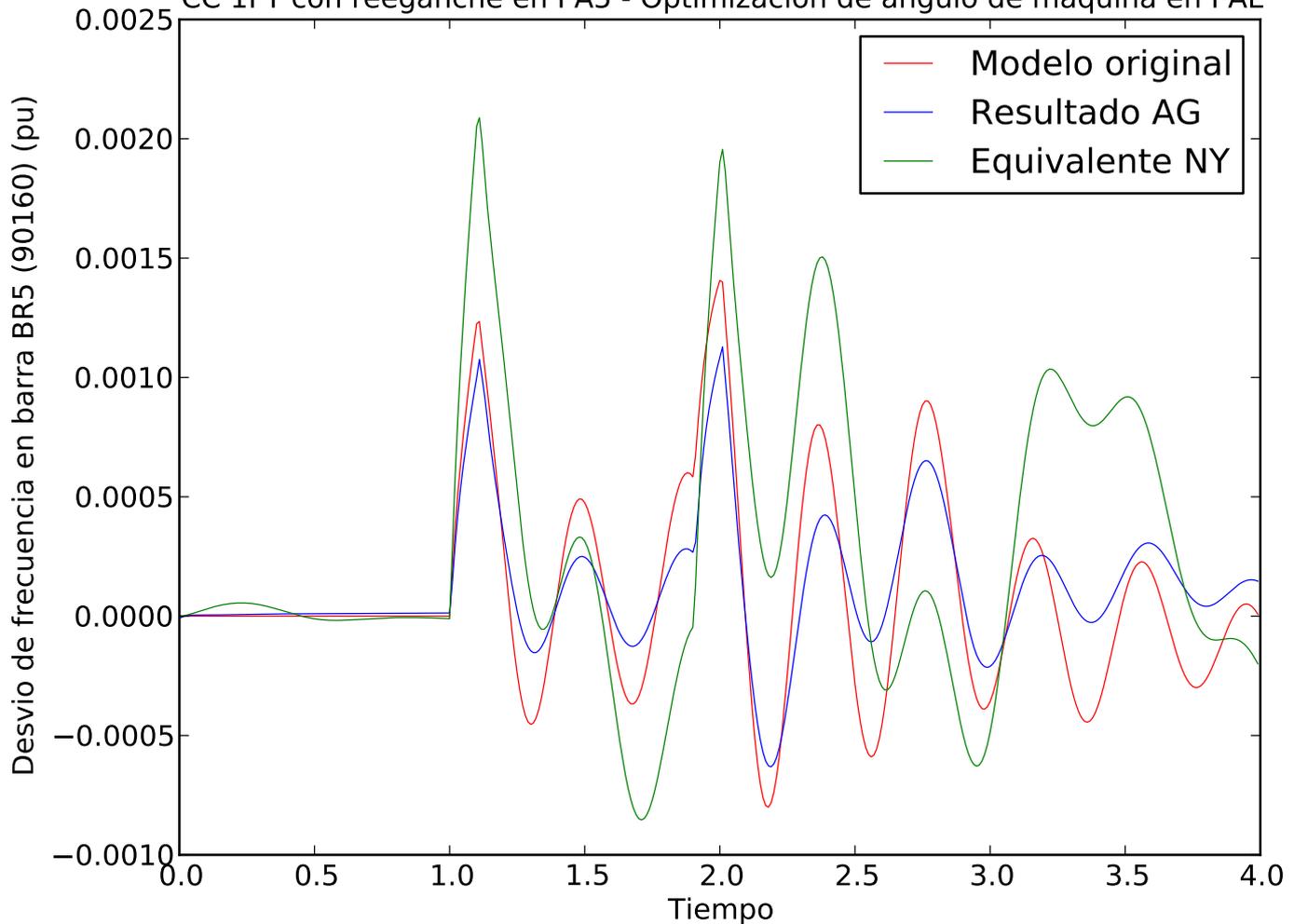
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



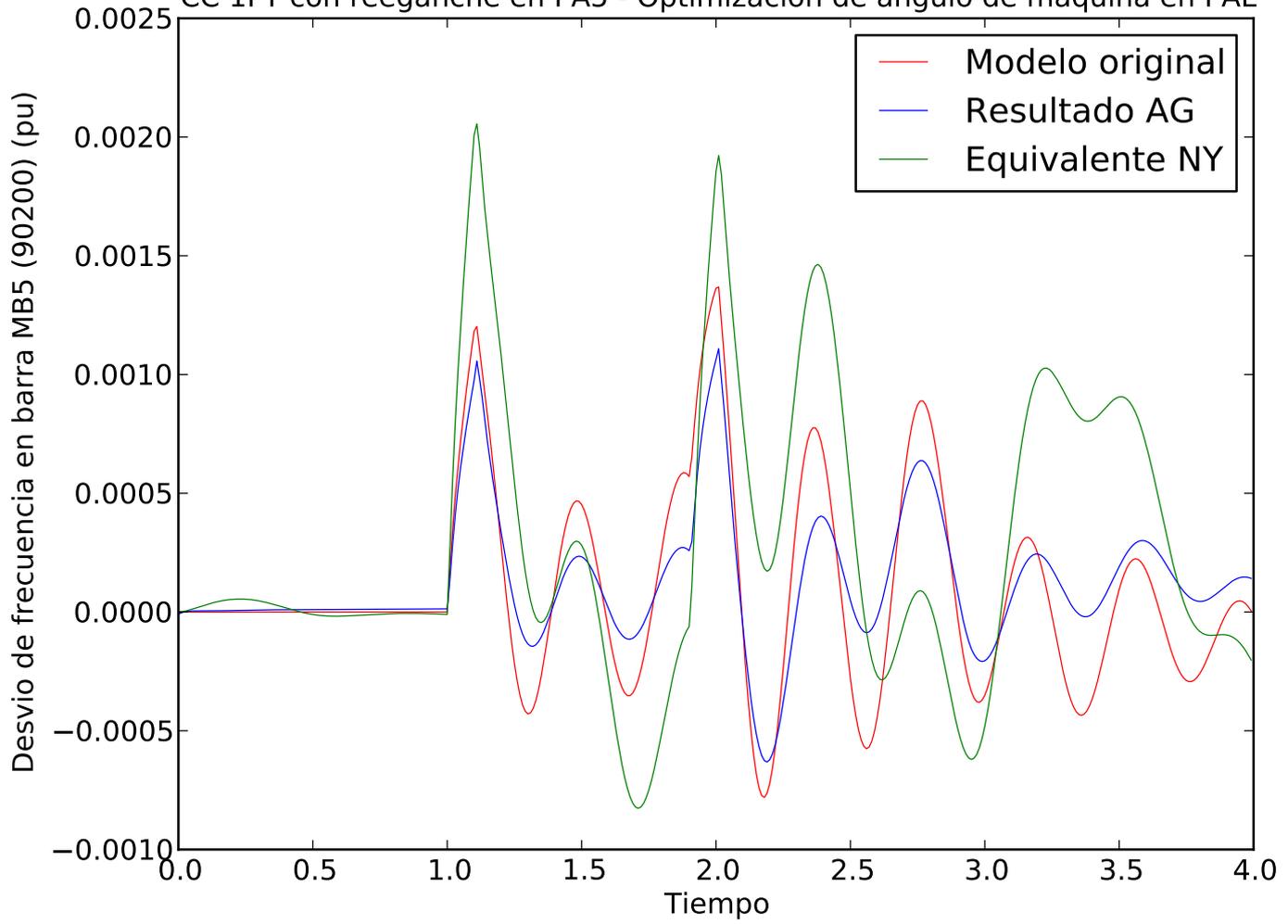
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



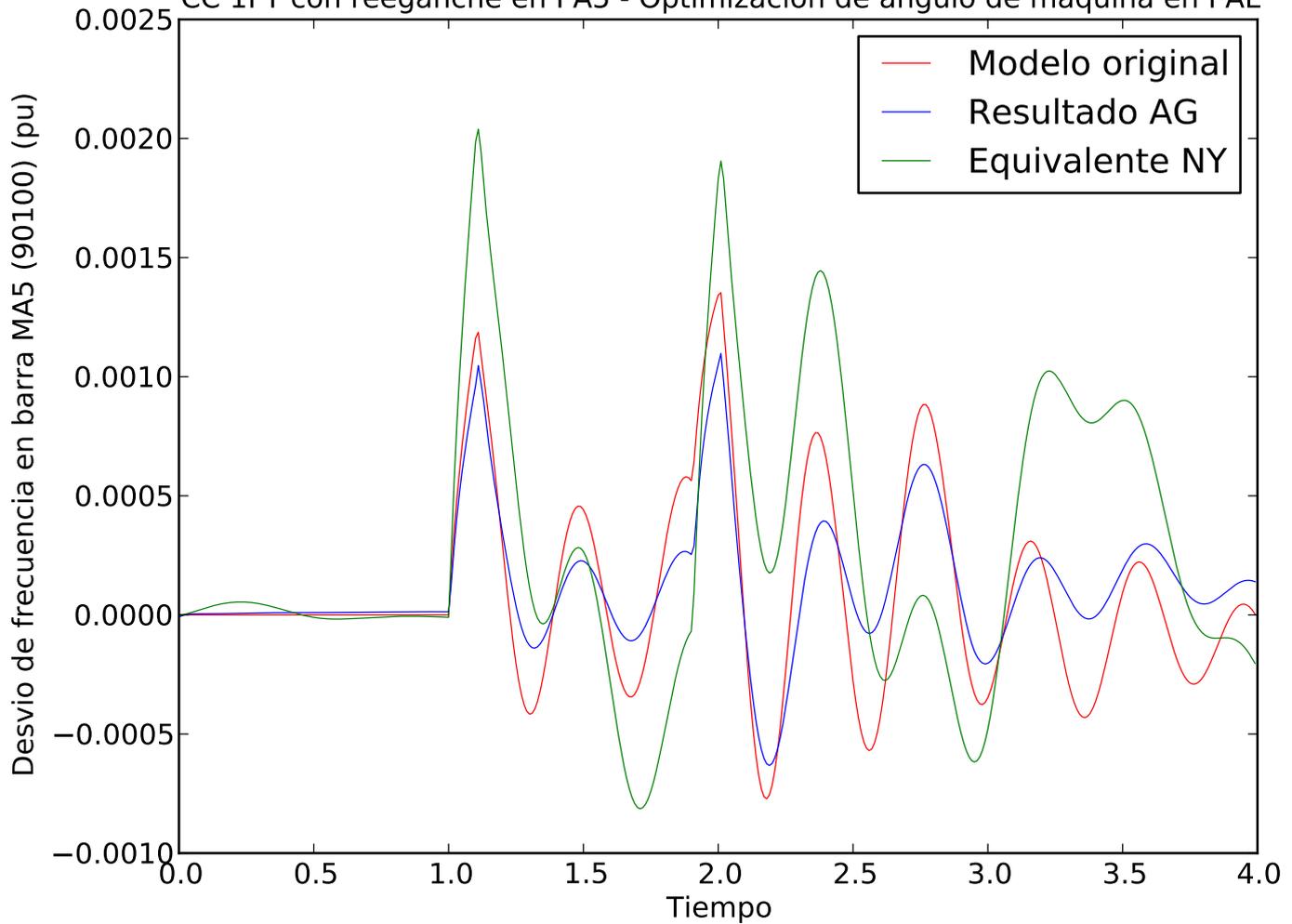
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



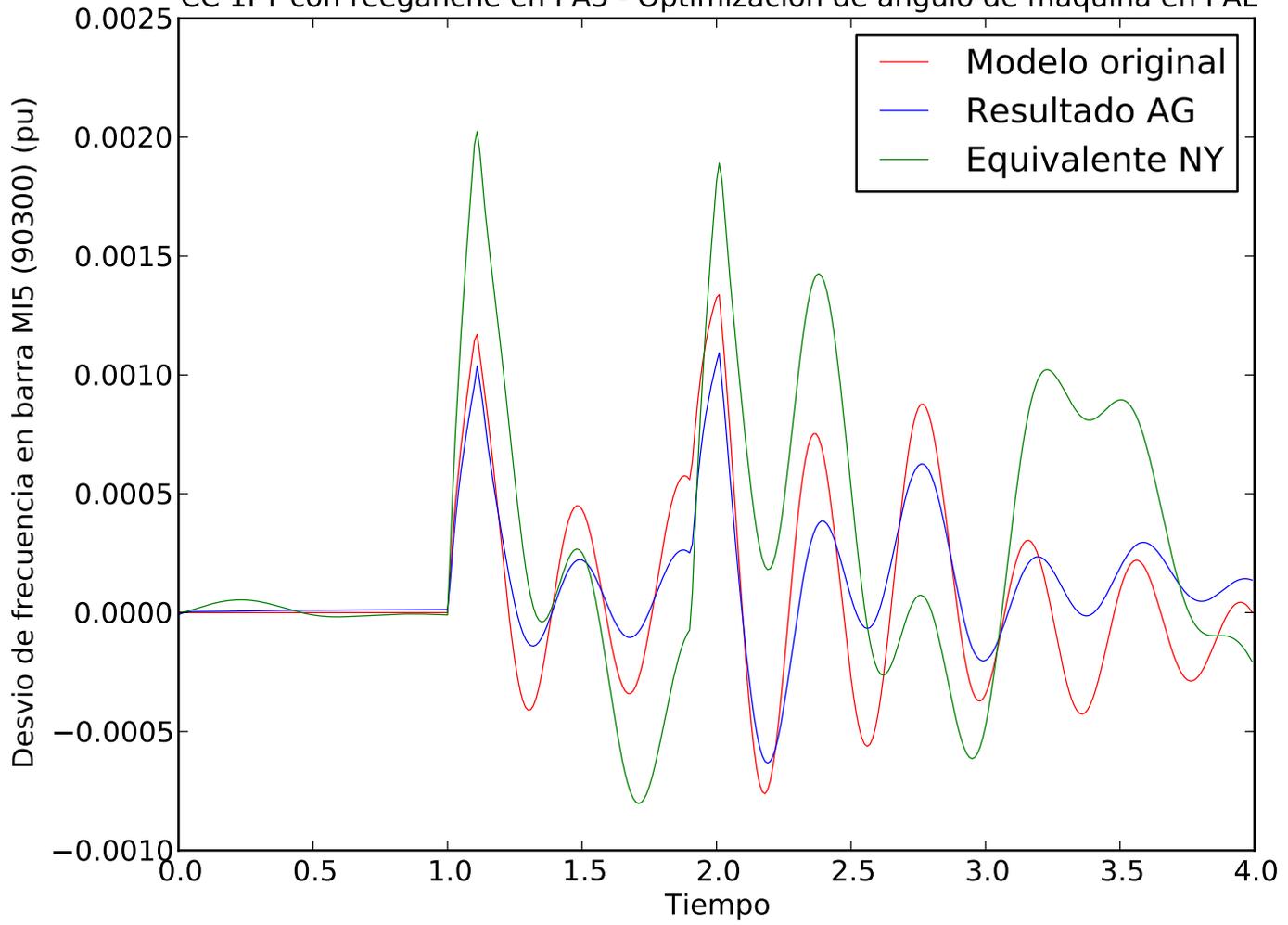
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



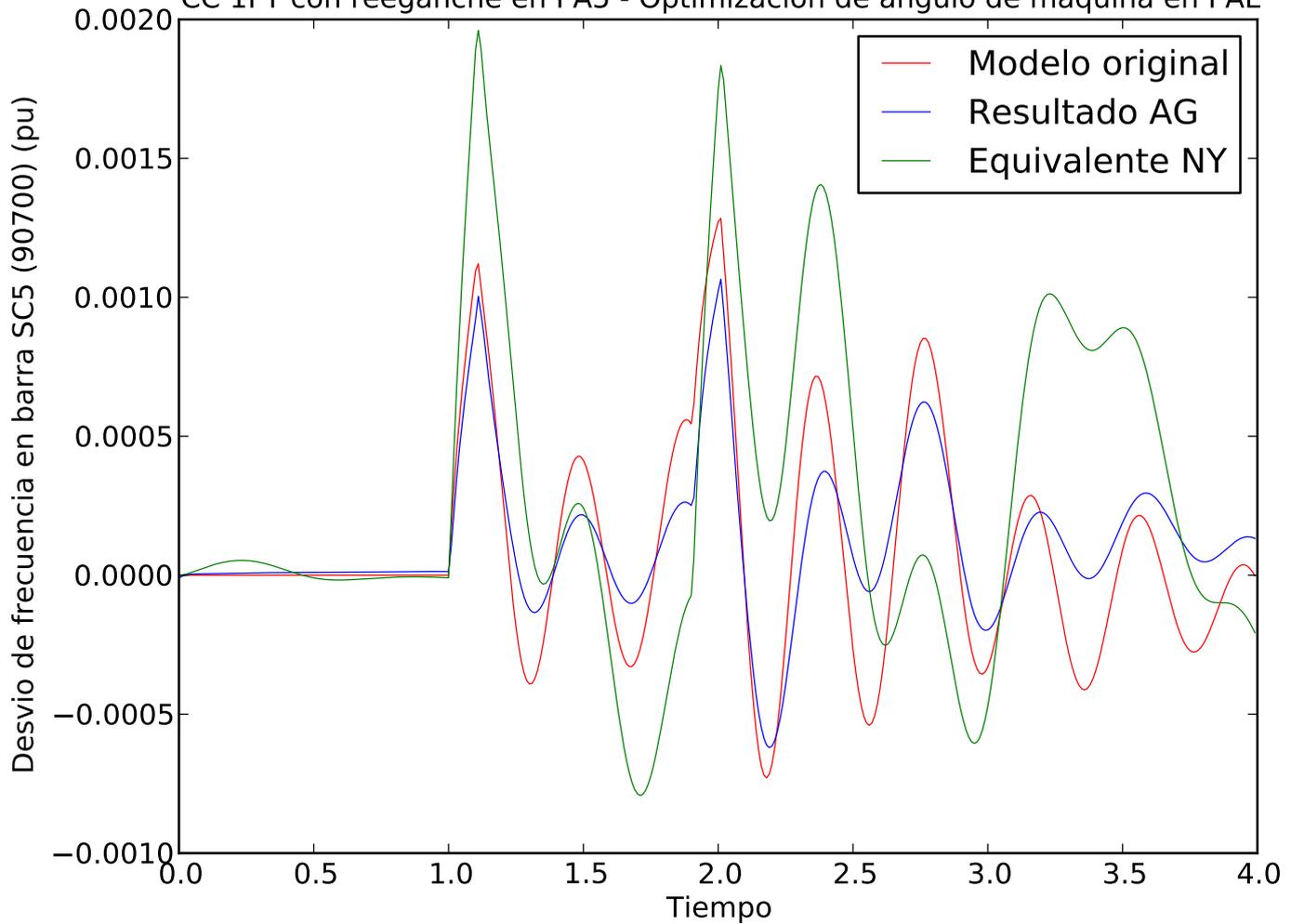
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



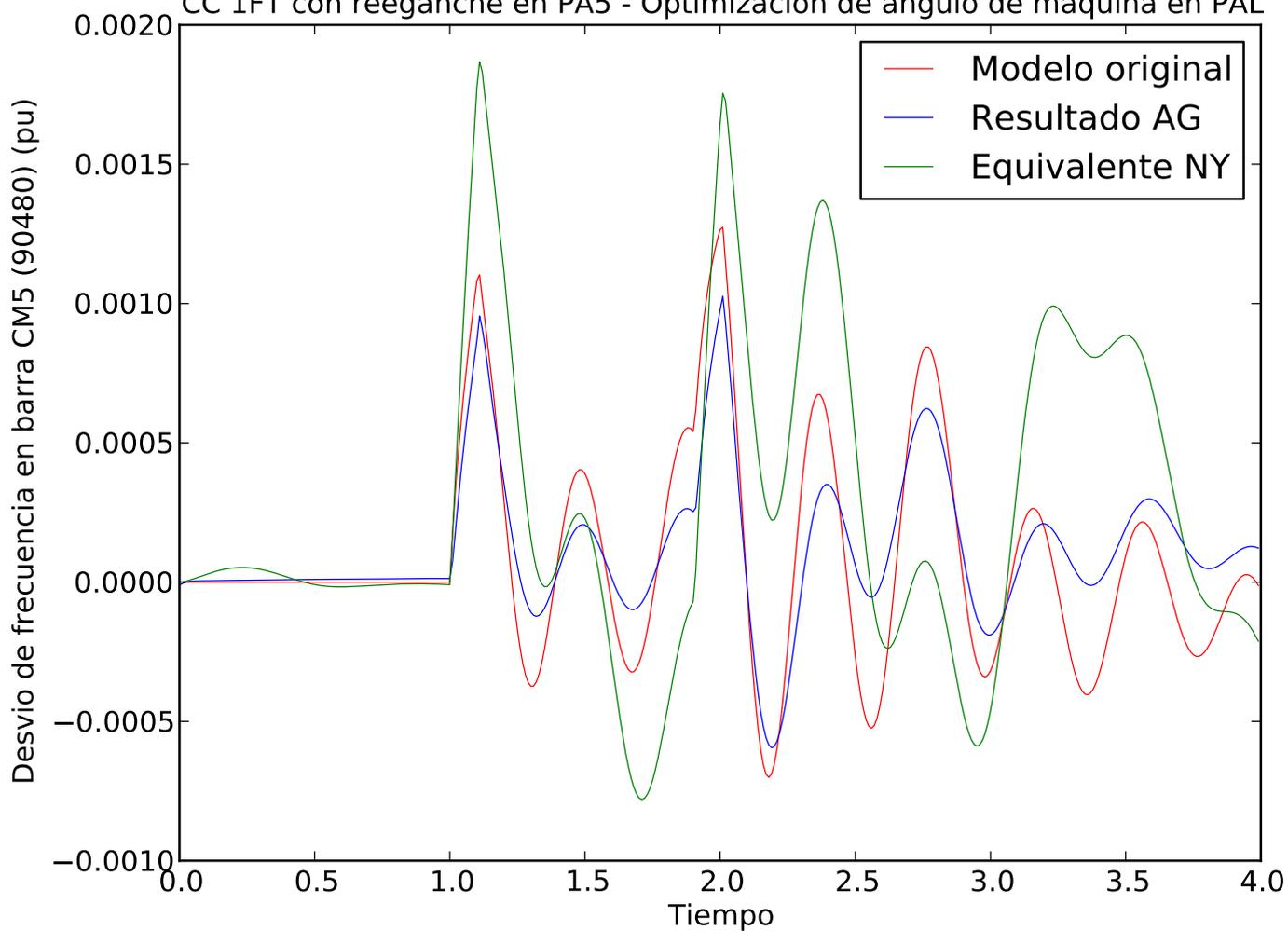
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



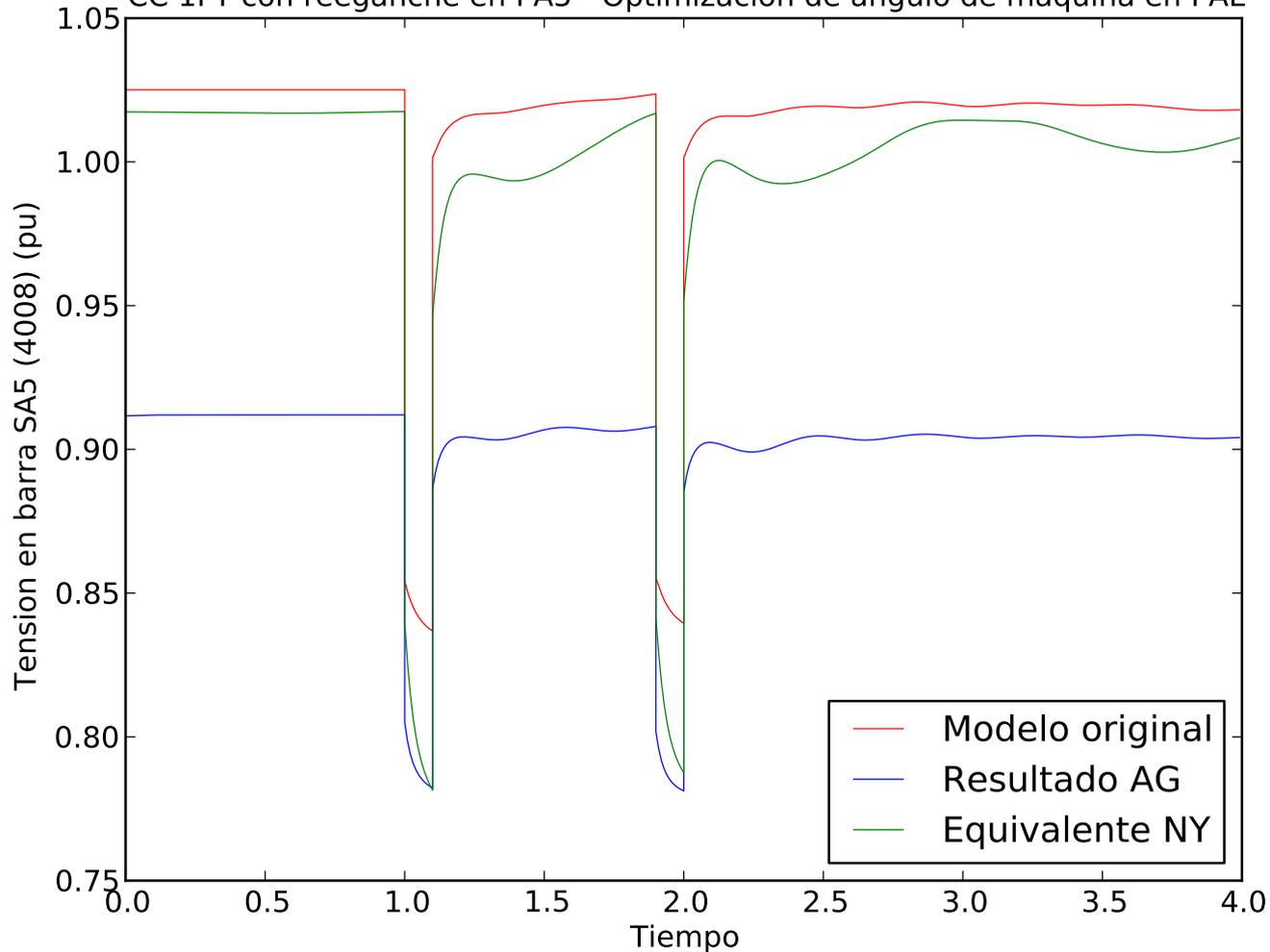
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



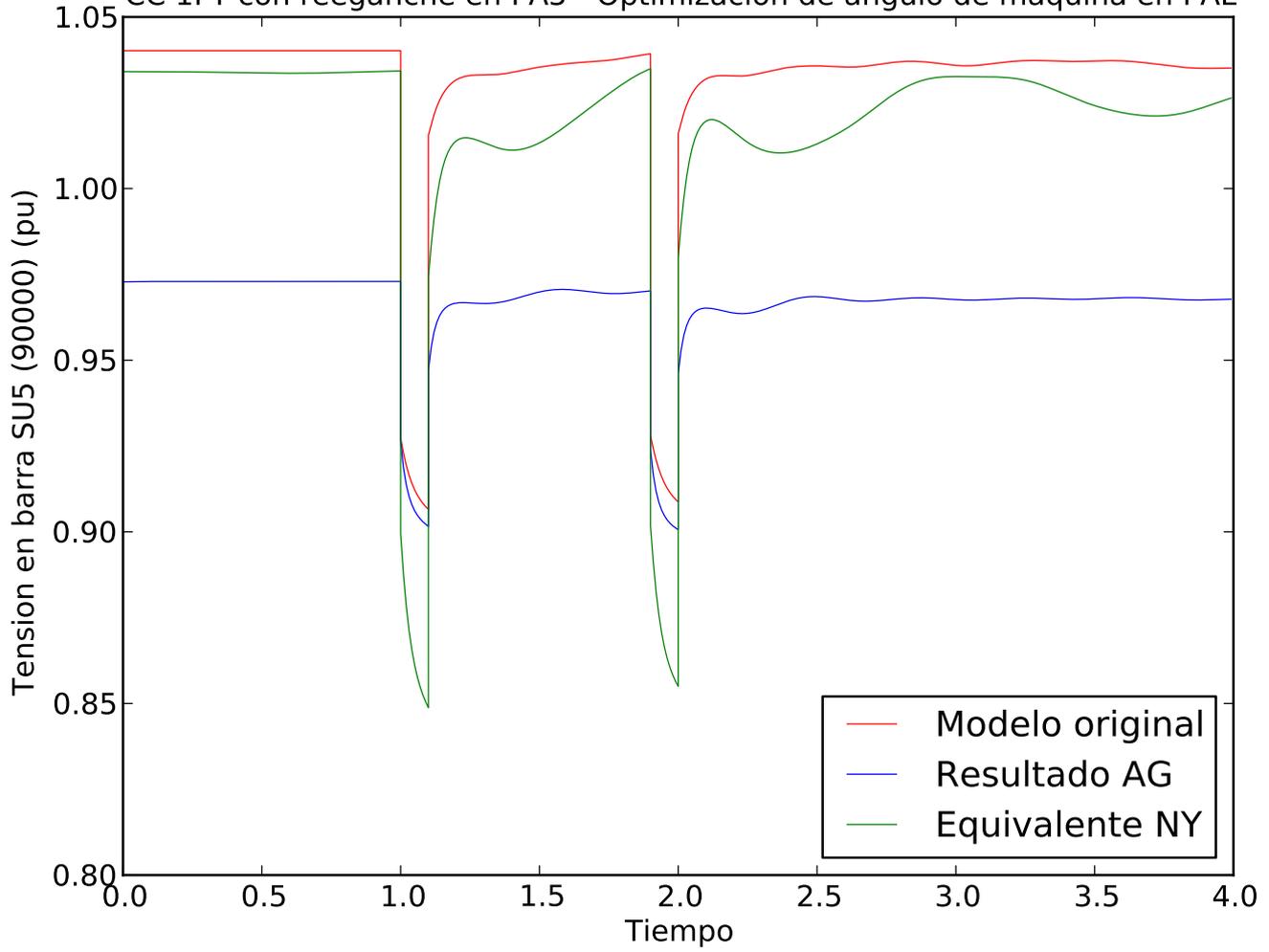
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



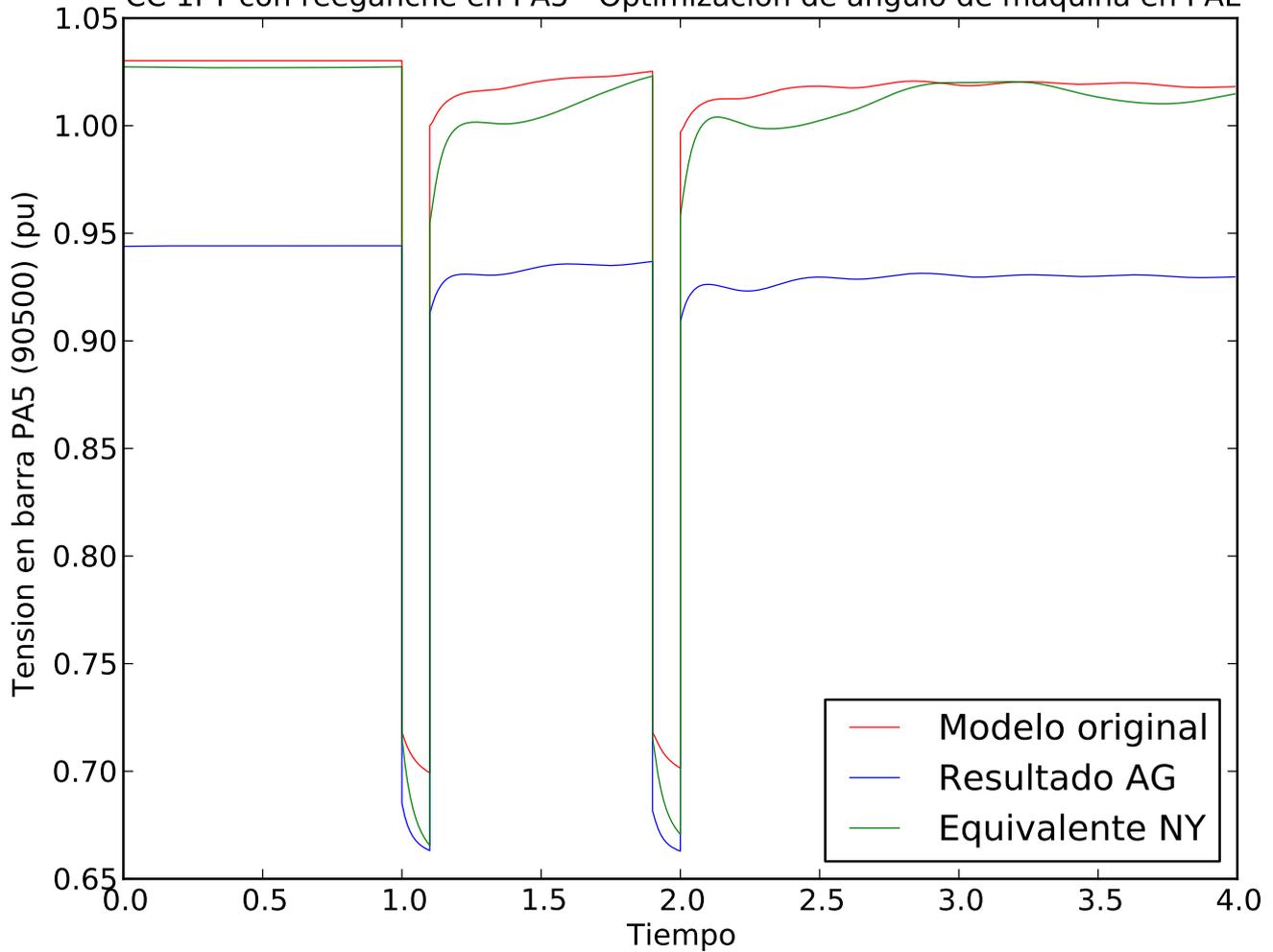
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



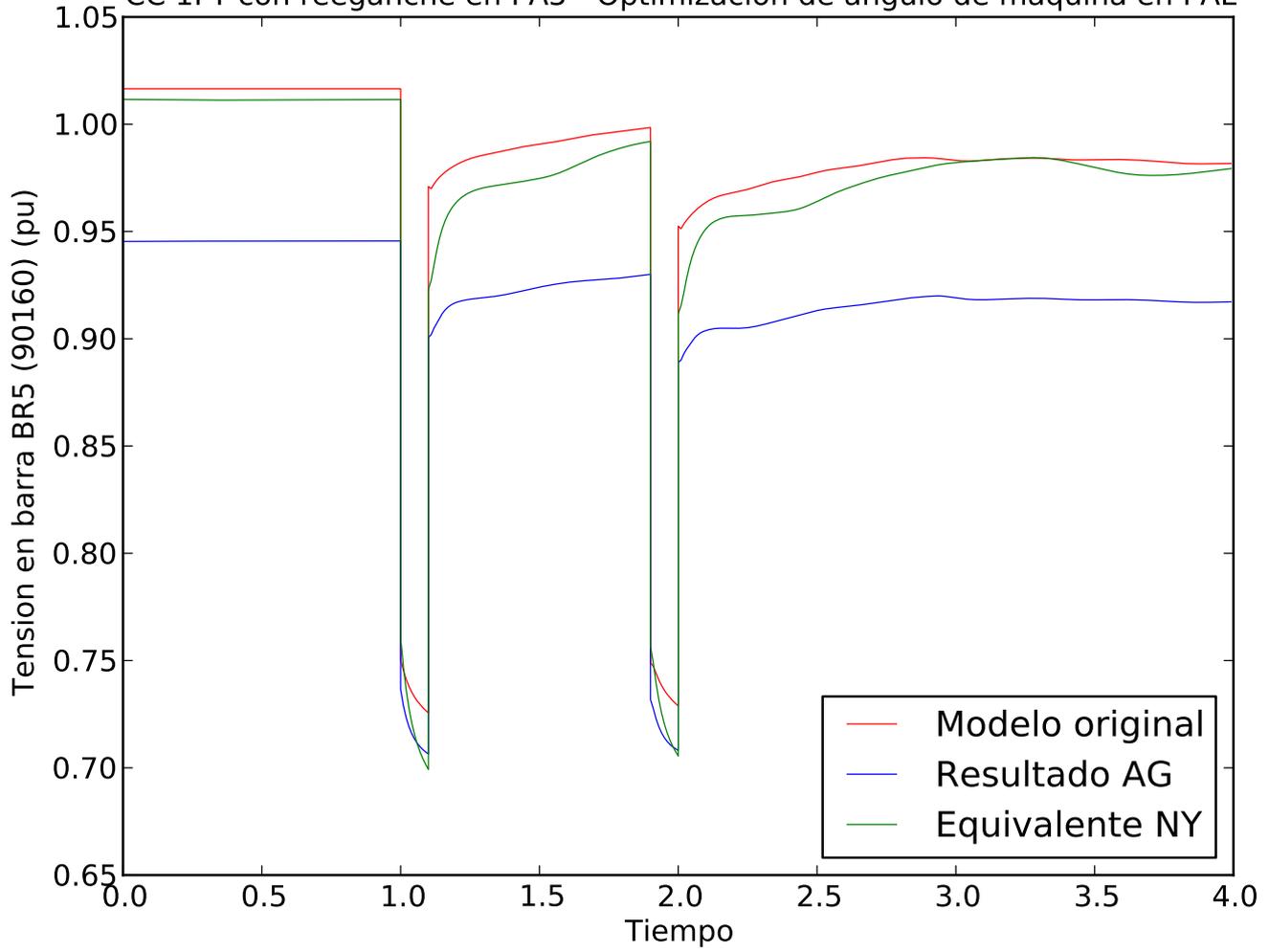
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



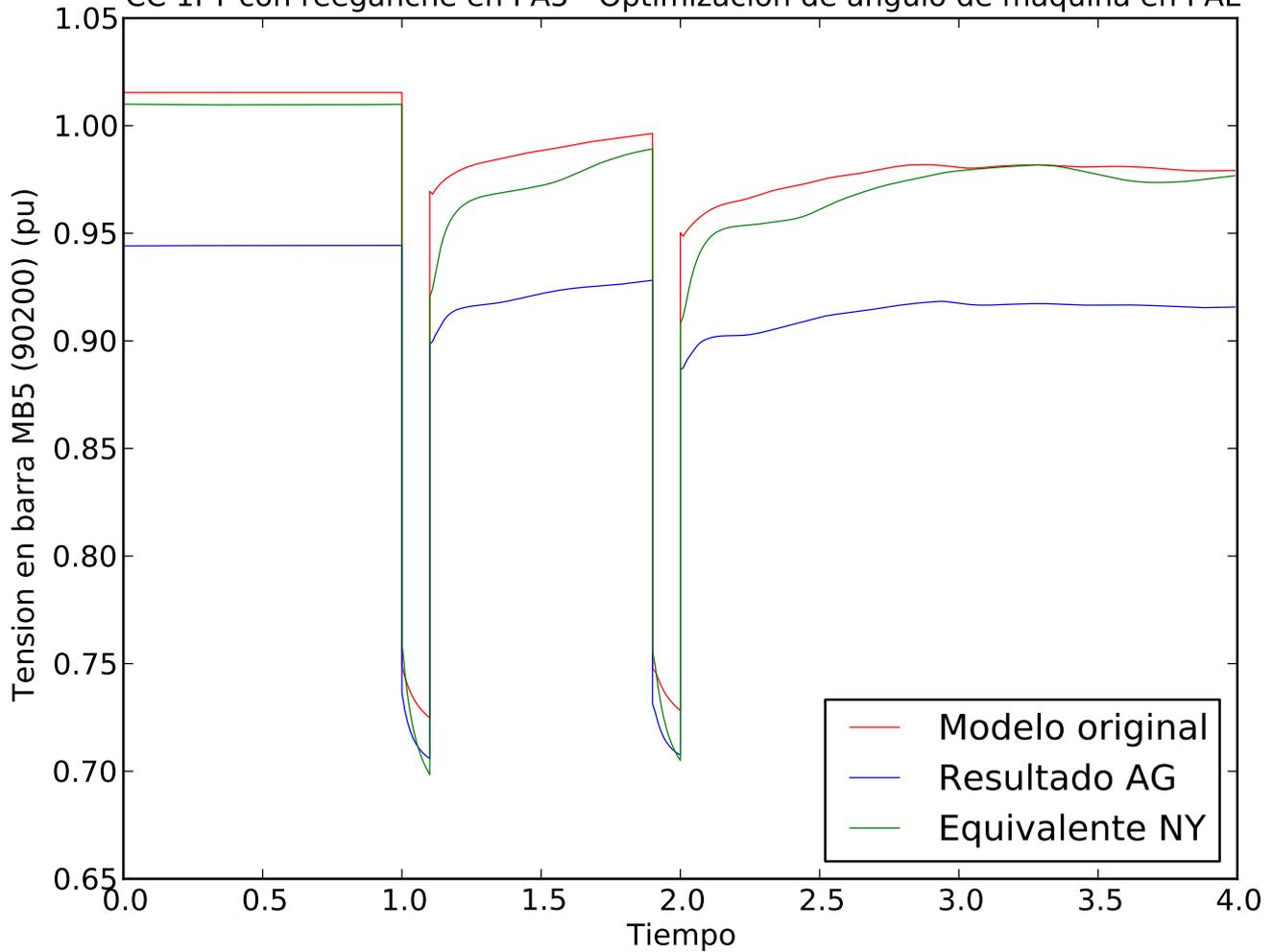
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



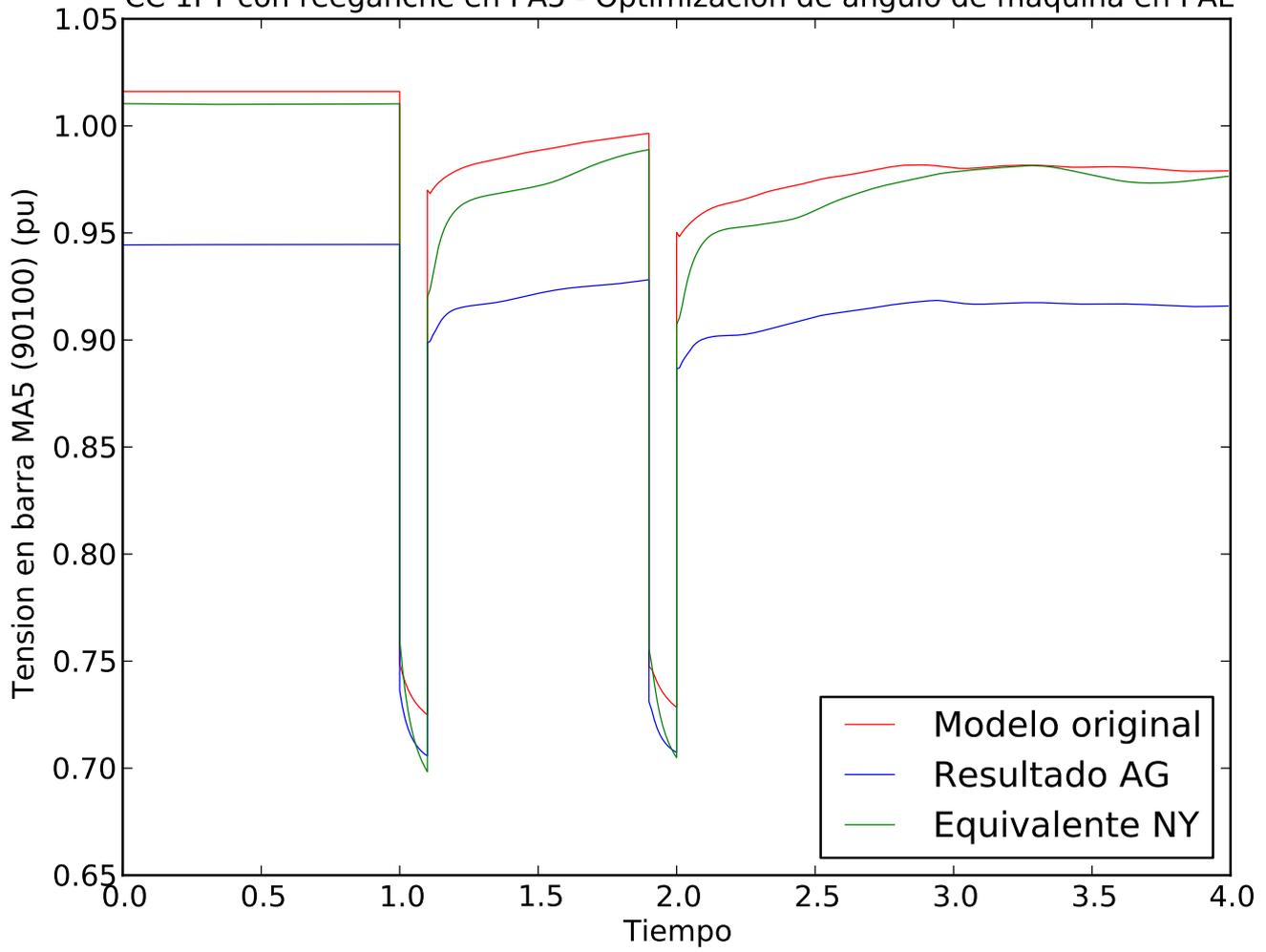
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



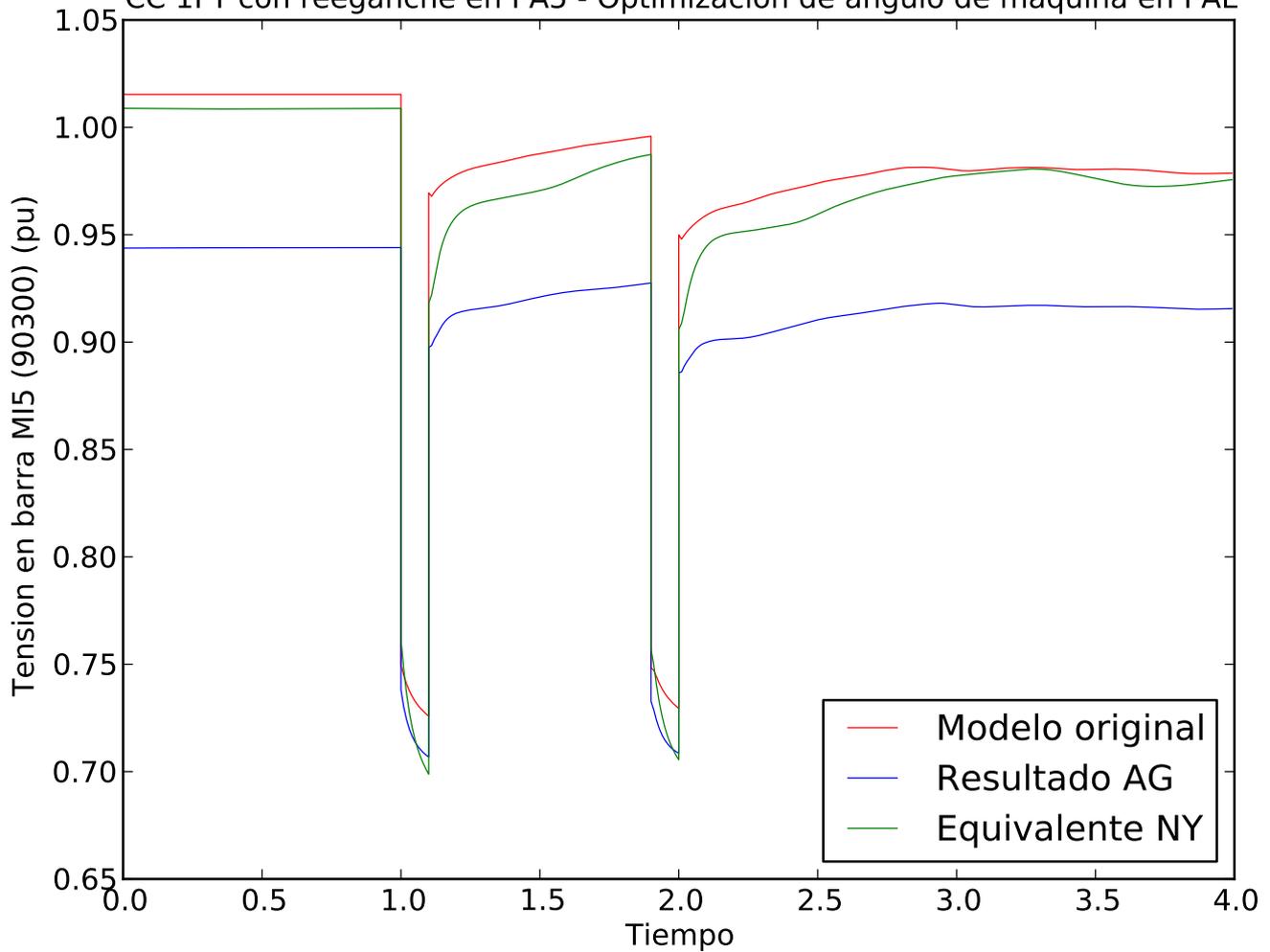
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



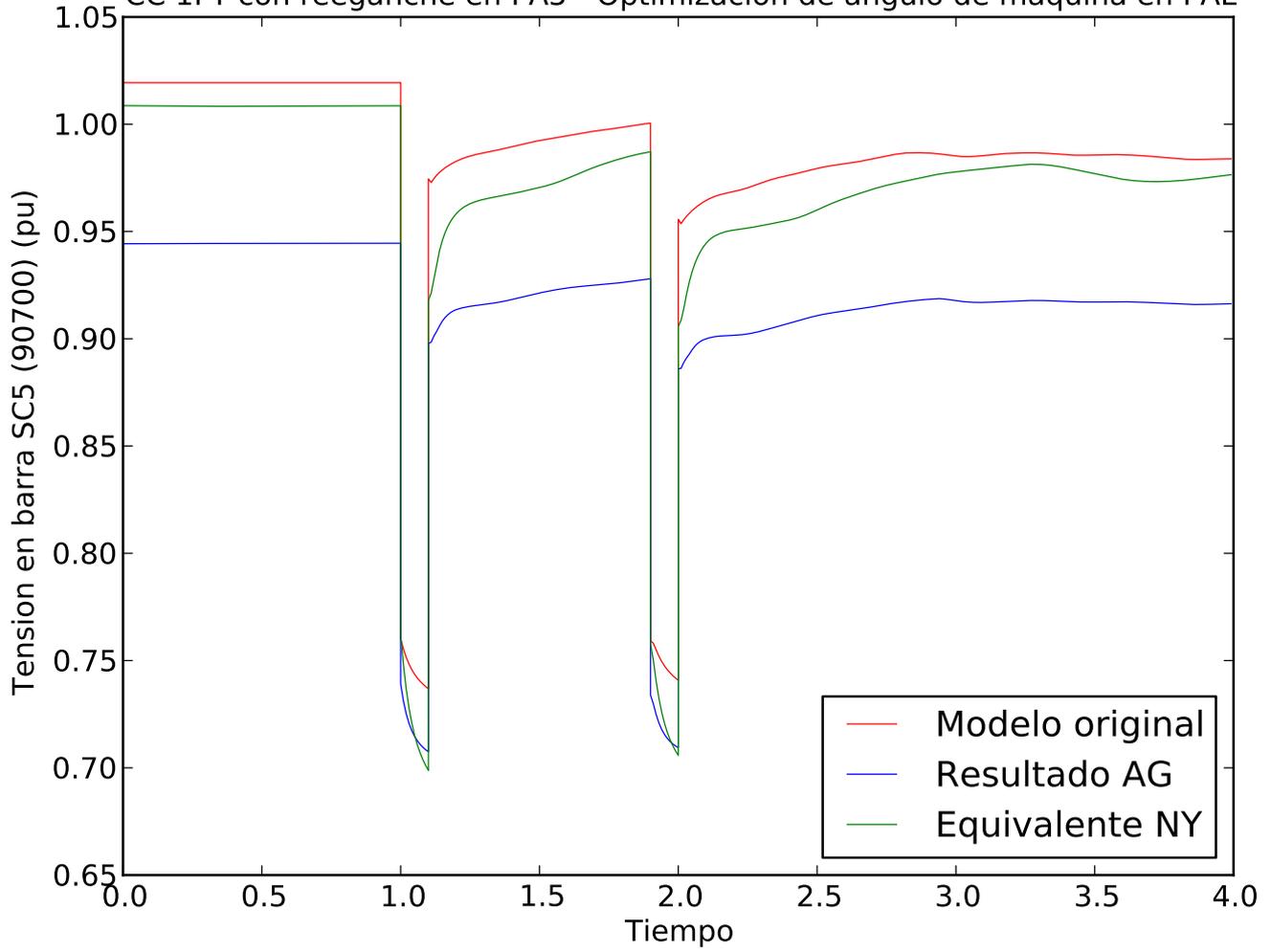
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



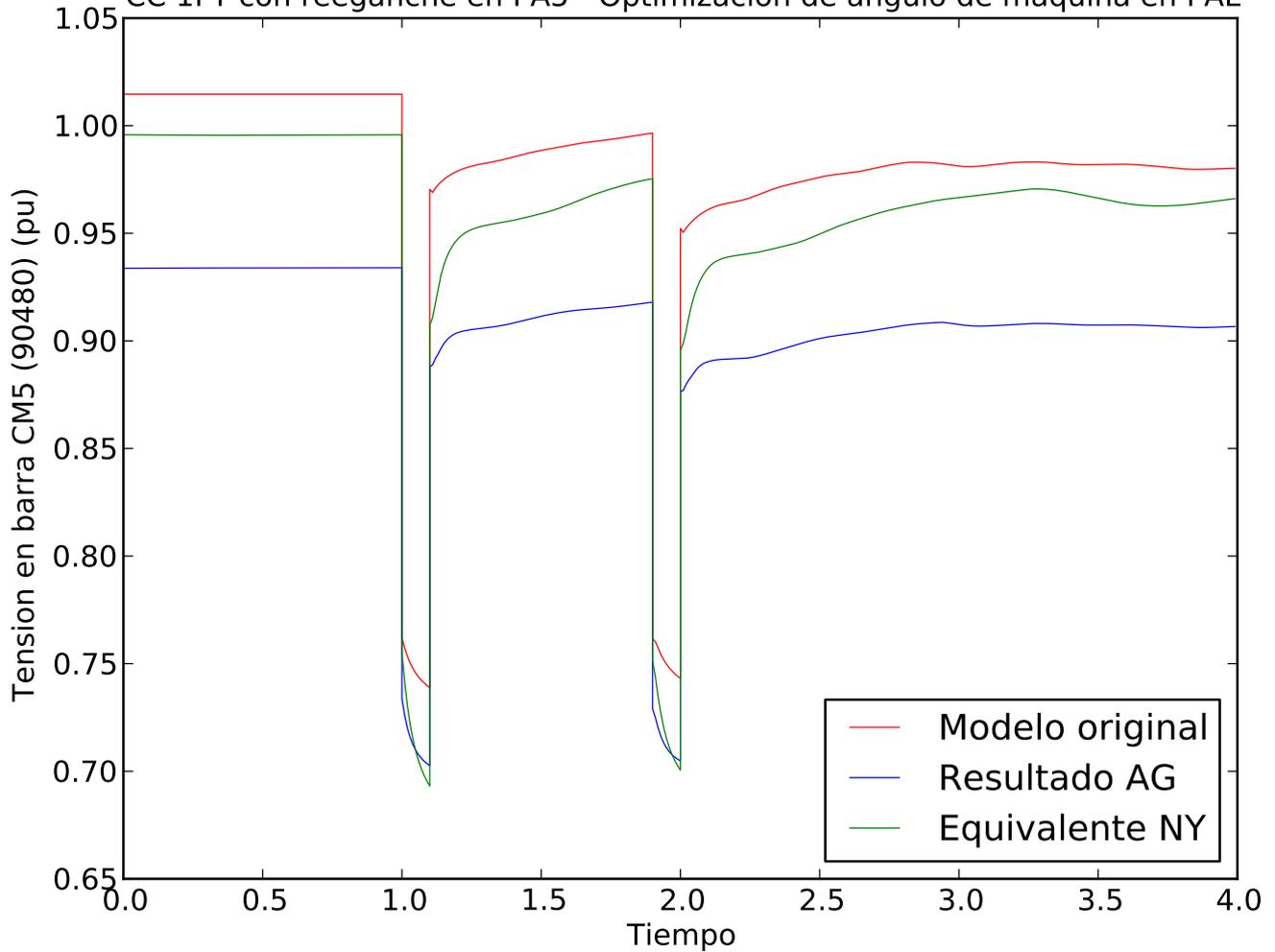
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



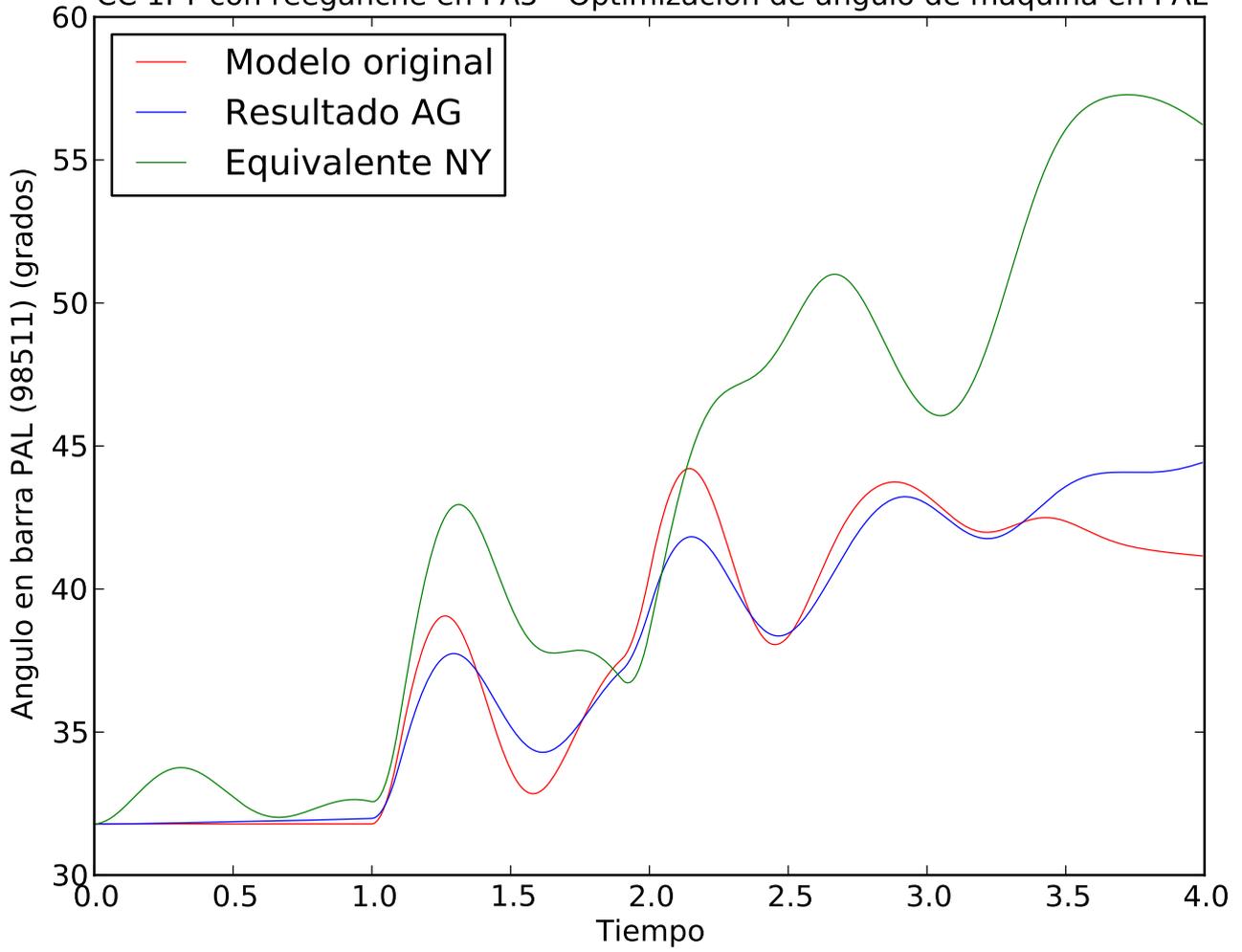
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



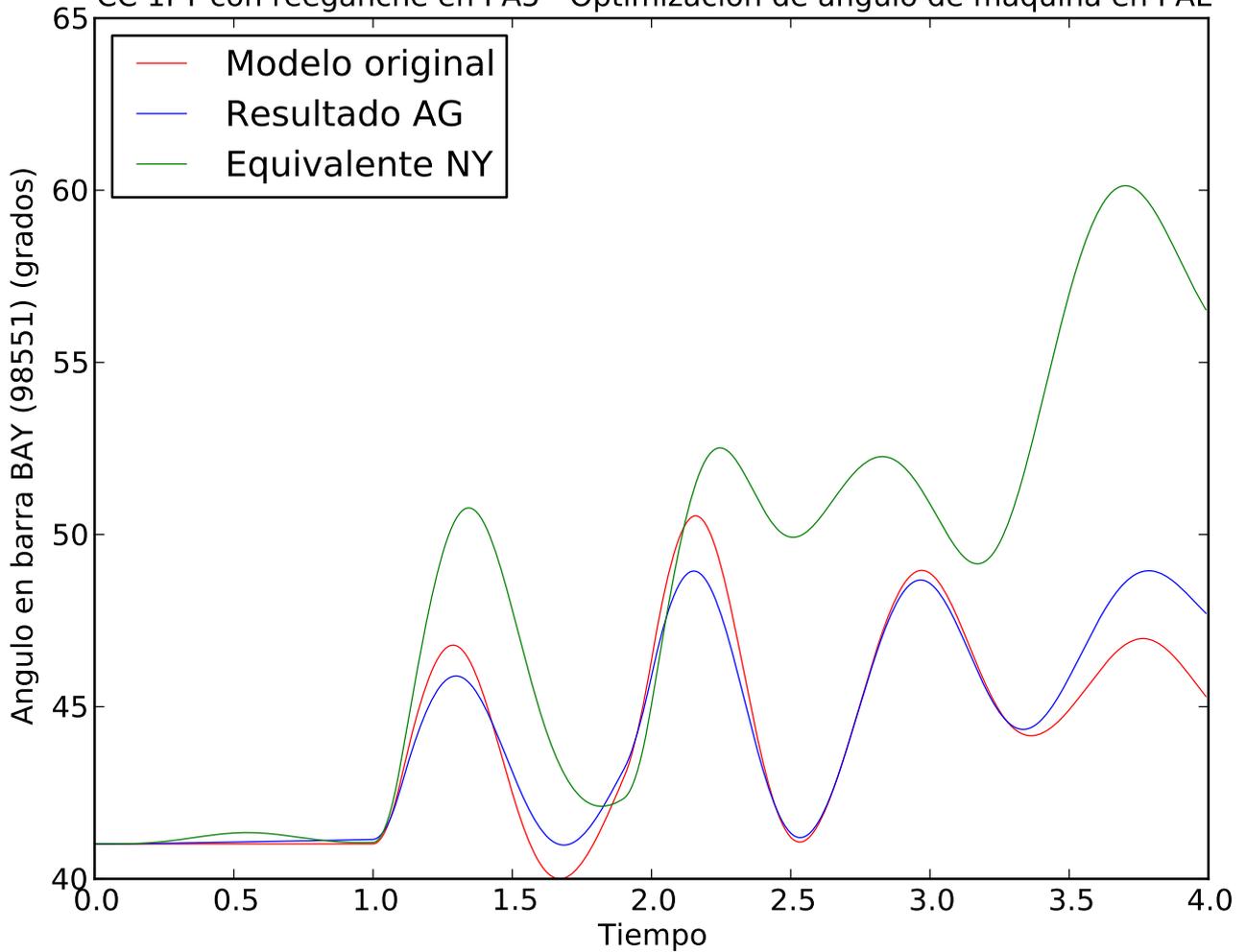
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PAL



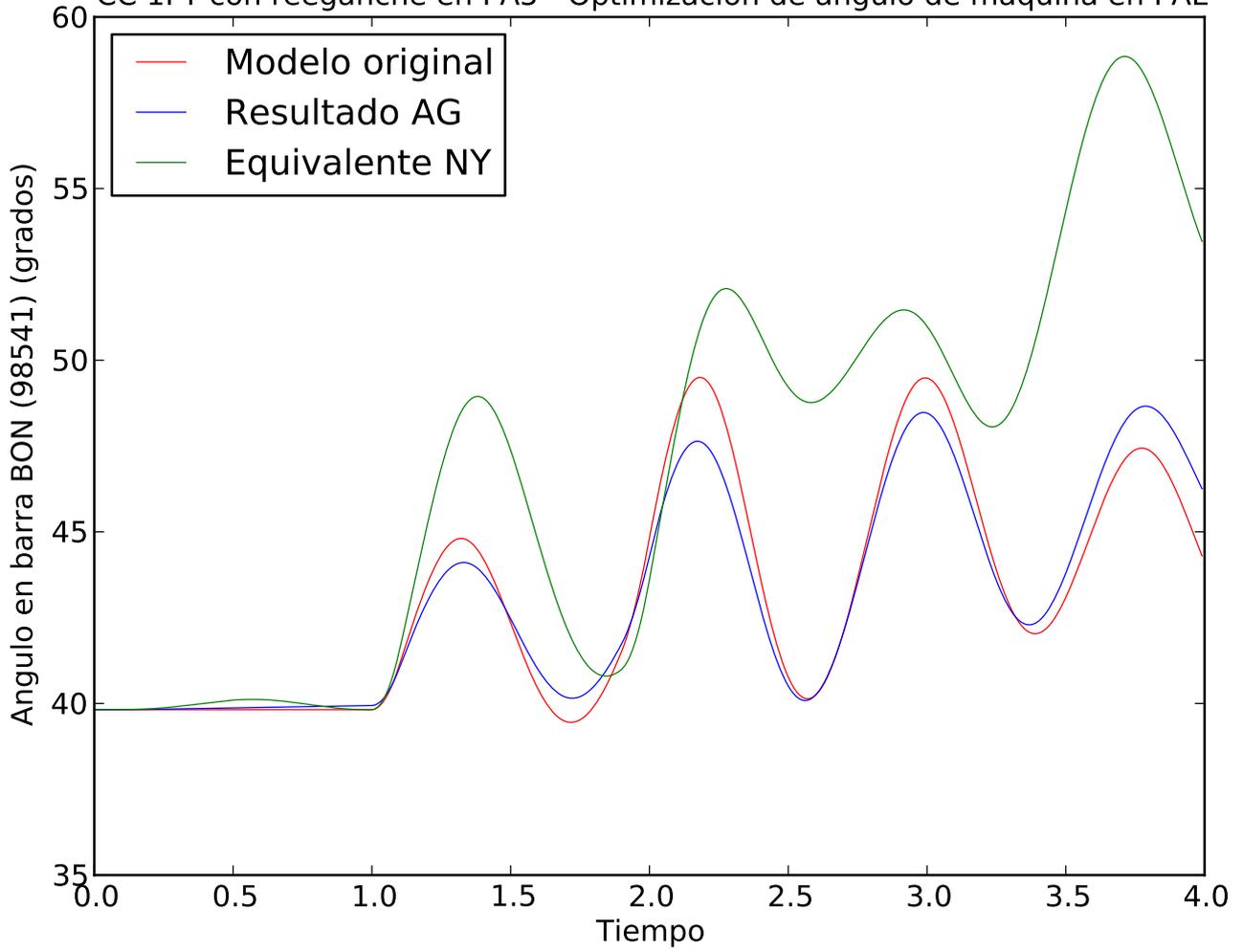
CC 1FT con reeganche en PA5 - Optimizacion de angulo de maquina en PAL



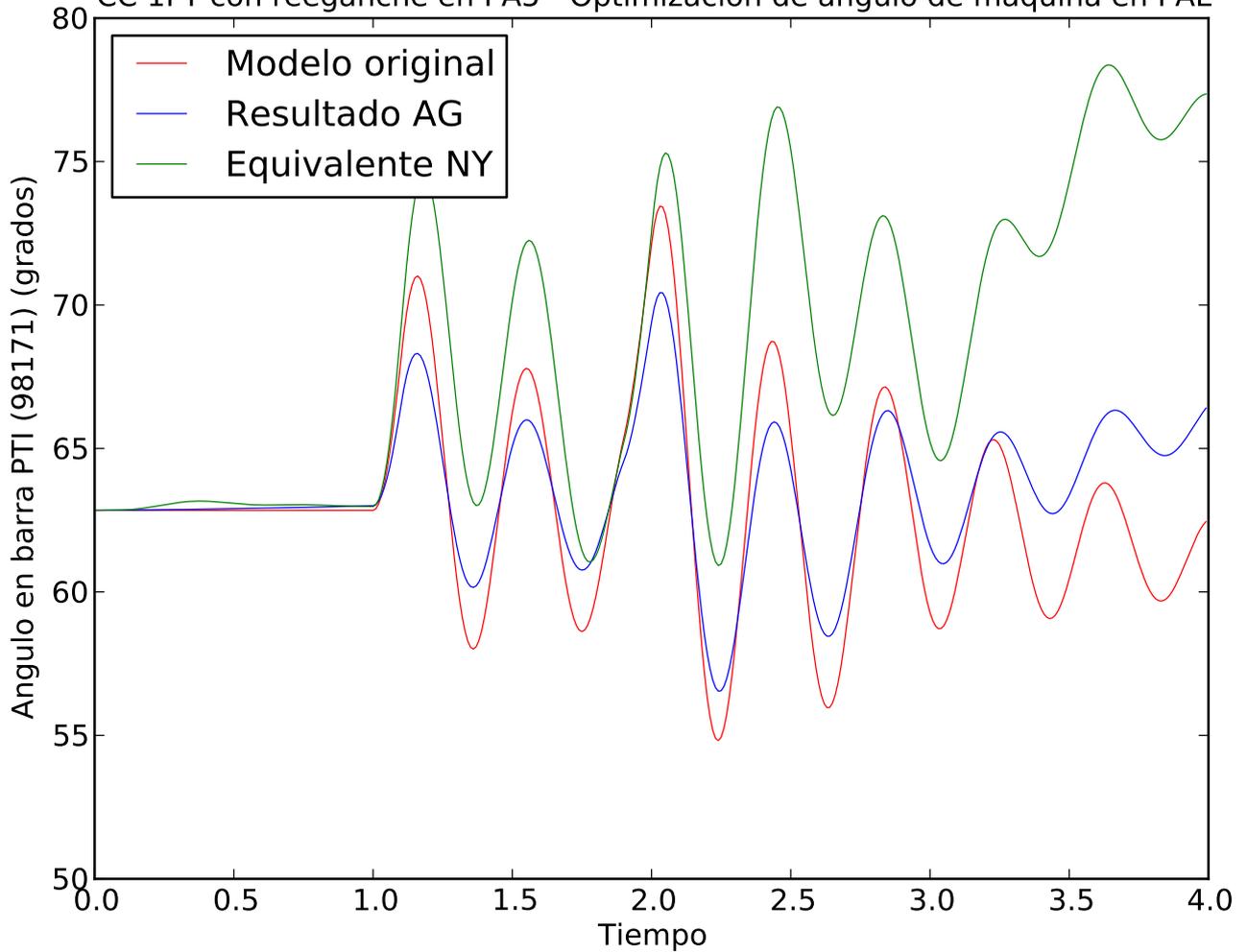
CC 1FT con reeganche en PA5 - Optimizacion de angulo de maquina en PAL



CC 1FT con reeganche en PA5 - Optimizacion de angulo de maquina en PAL

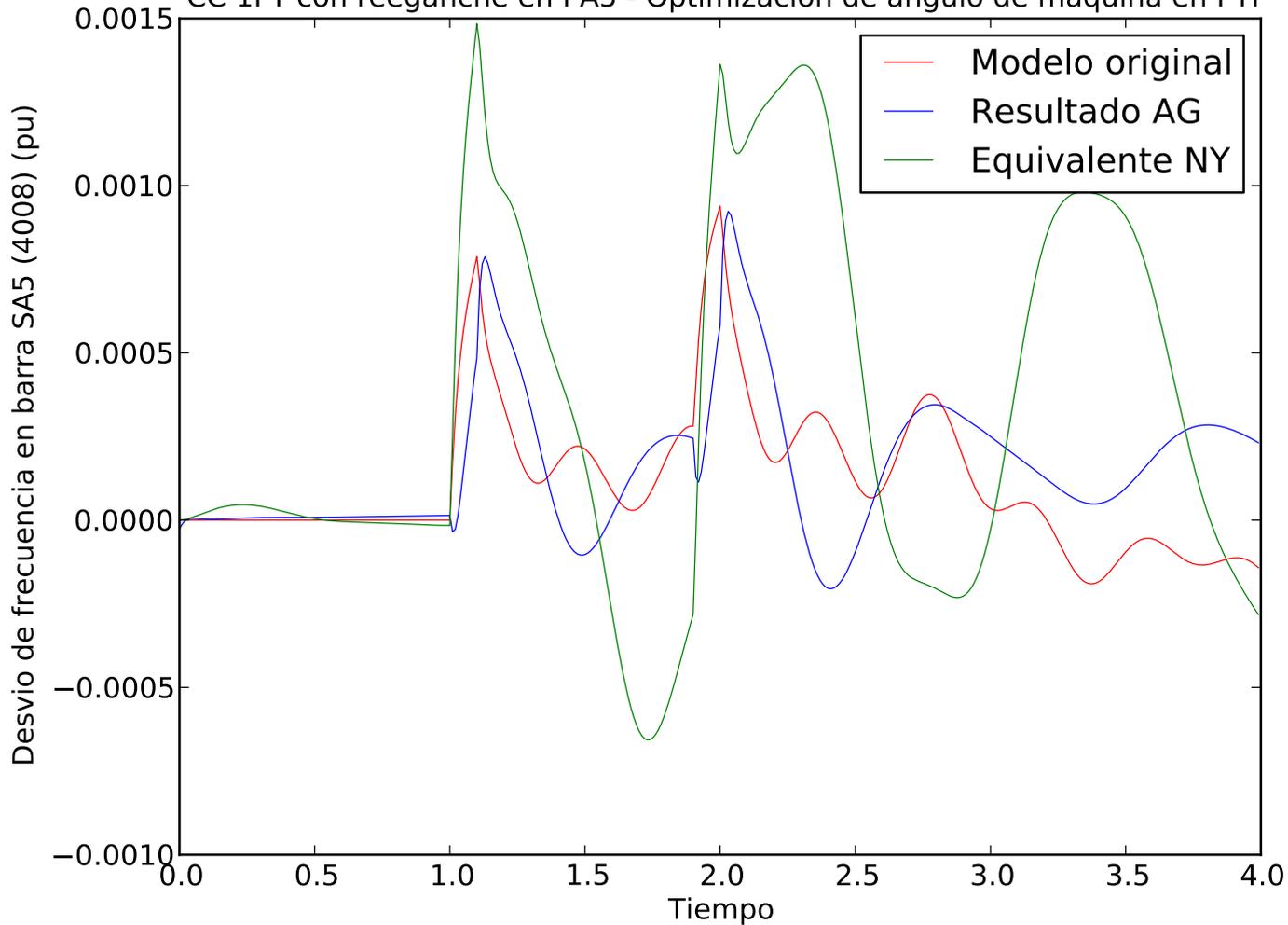


CC 1FT con reeganche en PA5 - Optimizacion de angulo de maquina en PAL

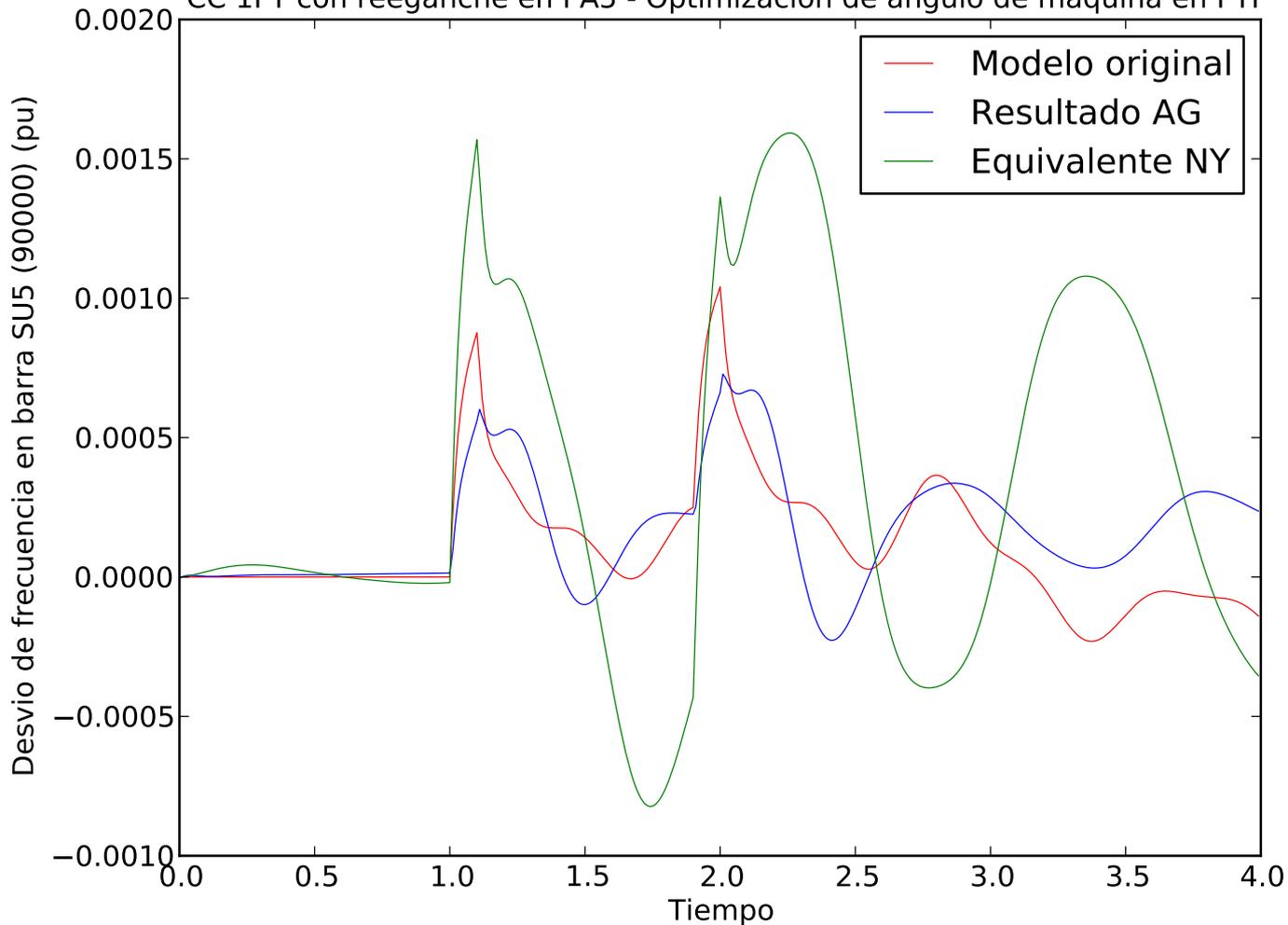


- C.13. Simulación 9.1: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando ángulo de un máquina de PTI en 4 segundos. Comportamiento en 4 segundos**

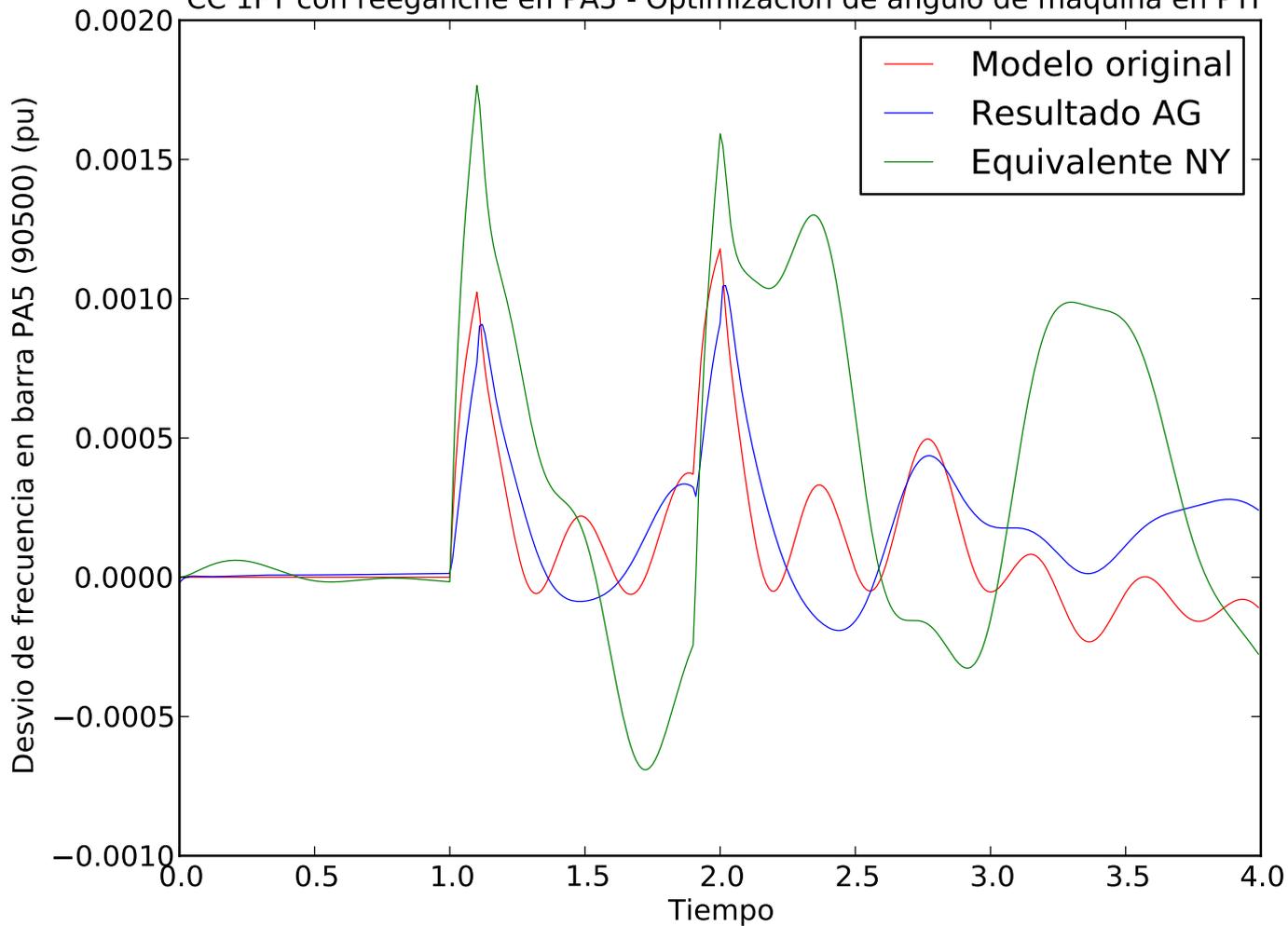
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



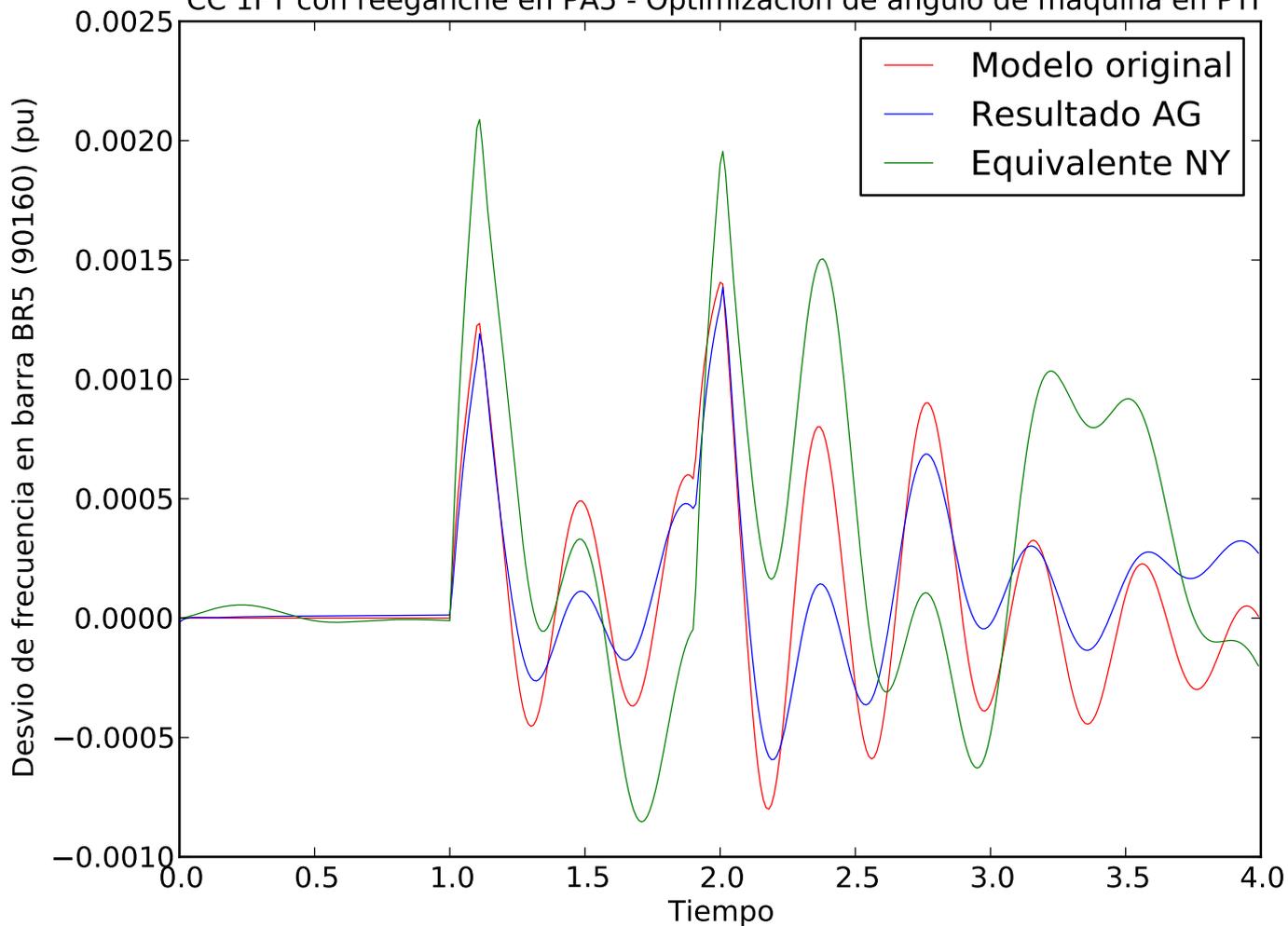
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



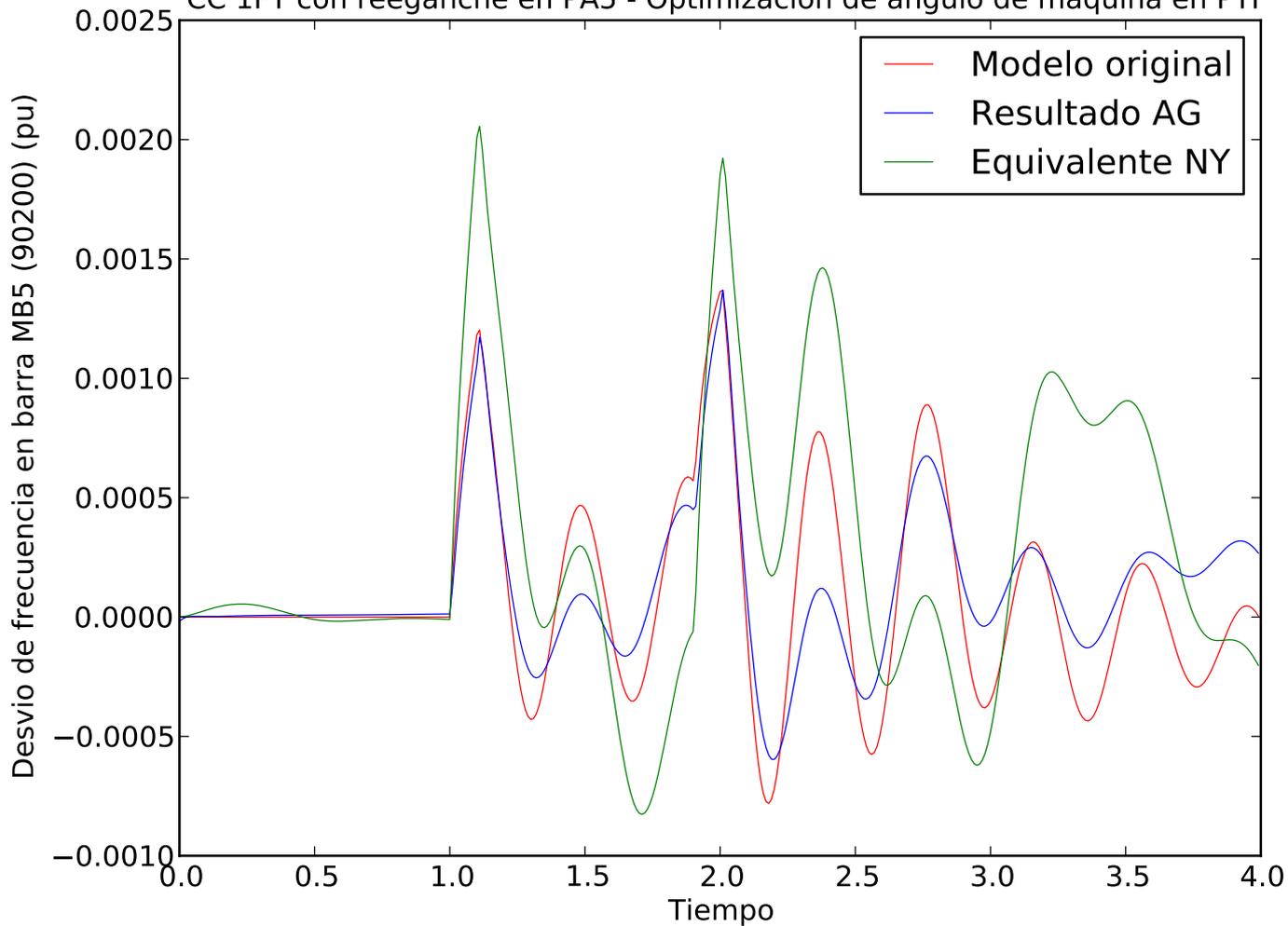
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



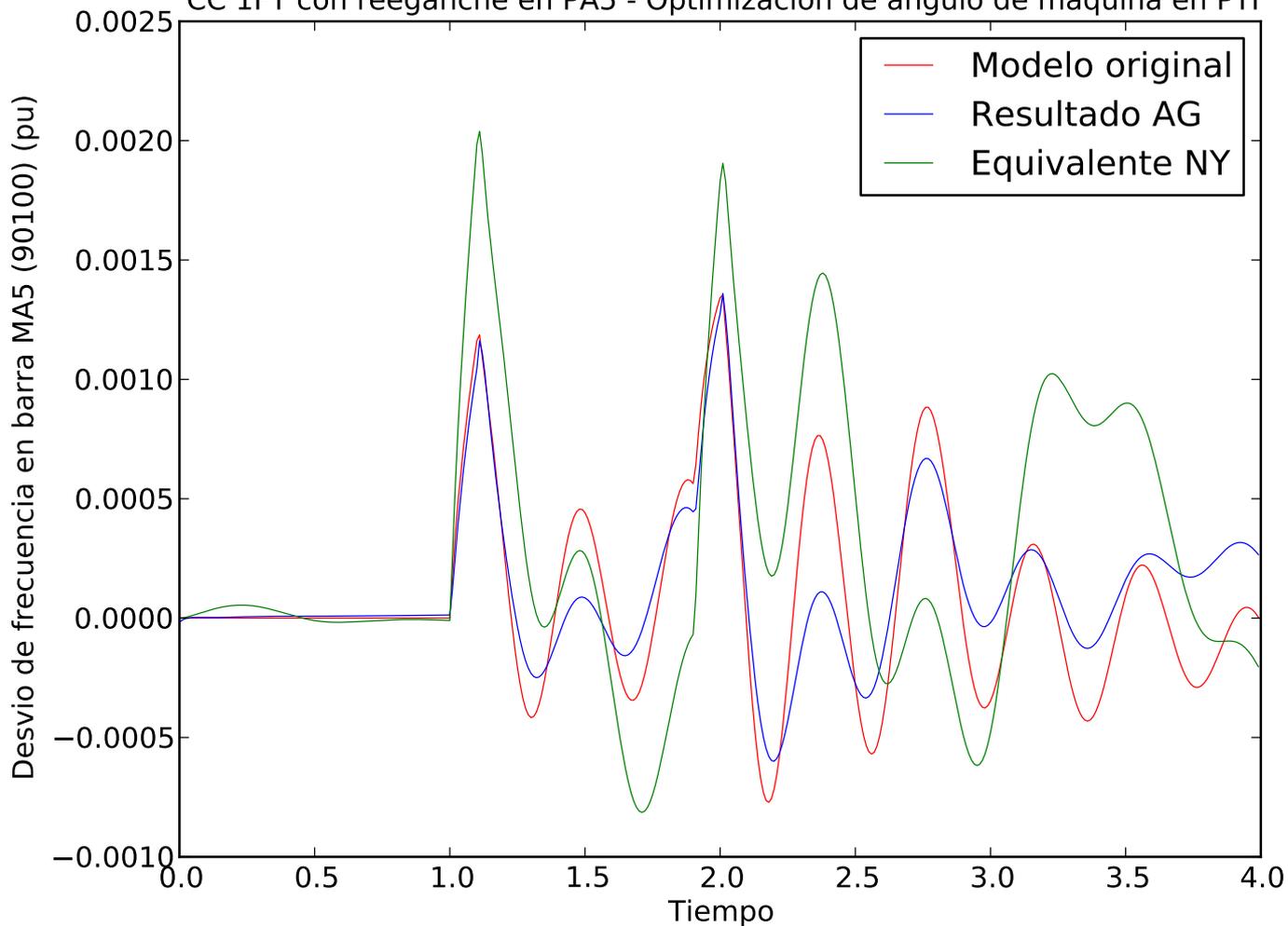
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



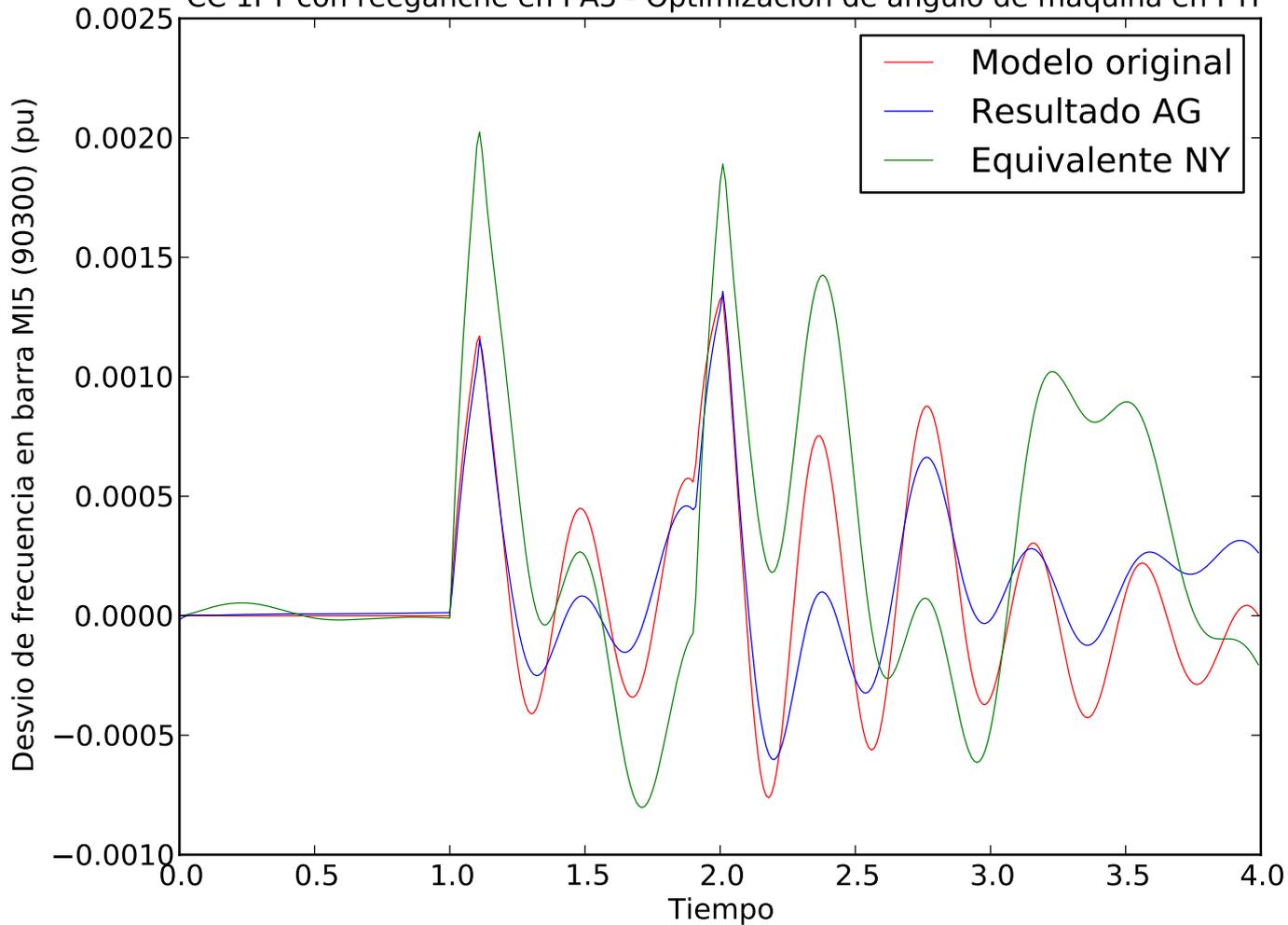
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



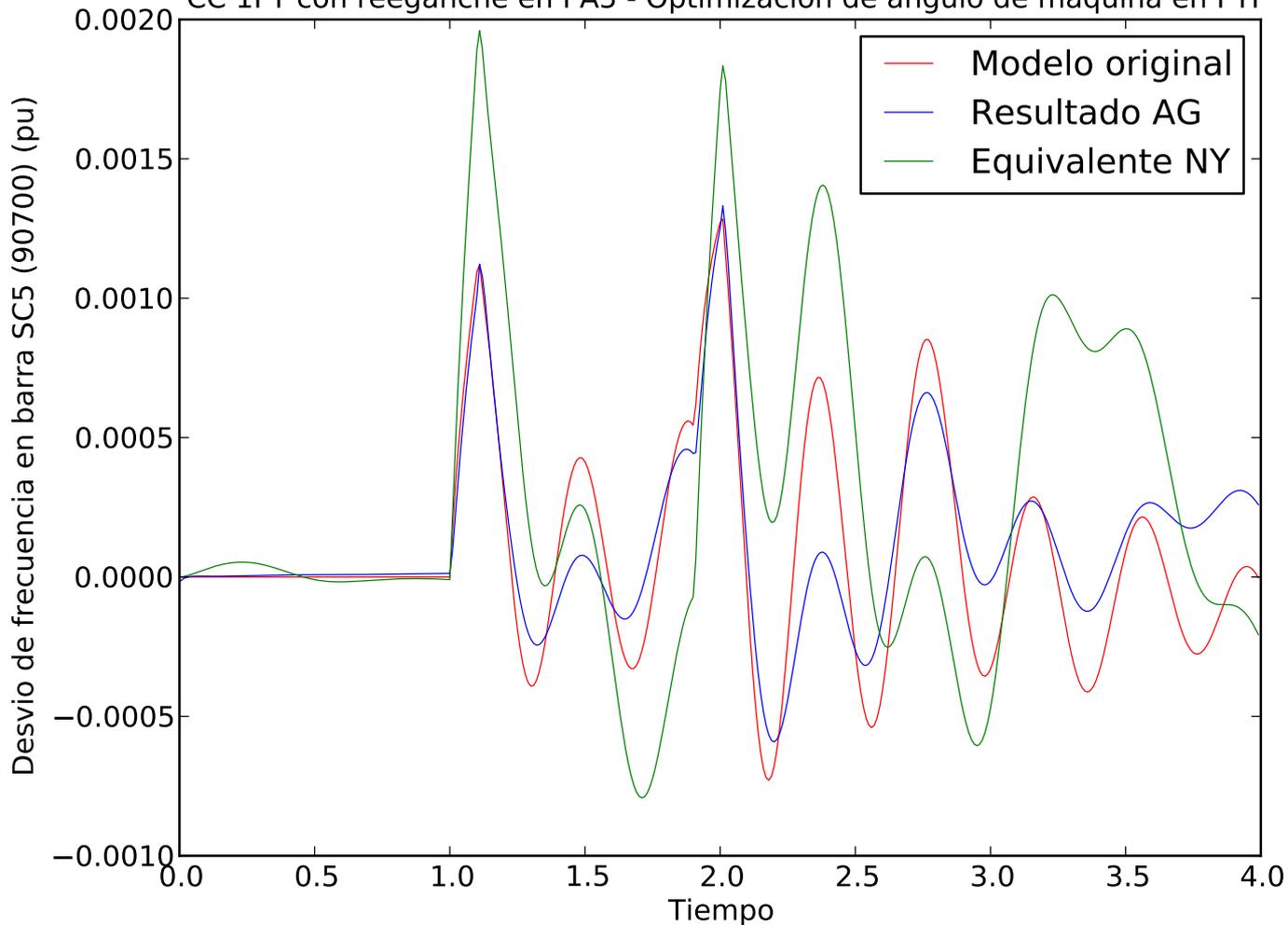
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



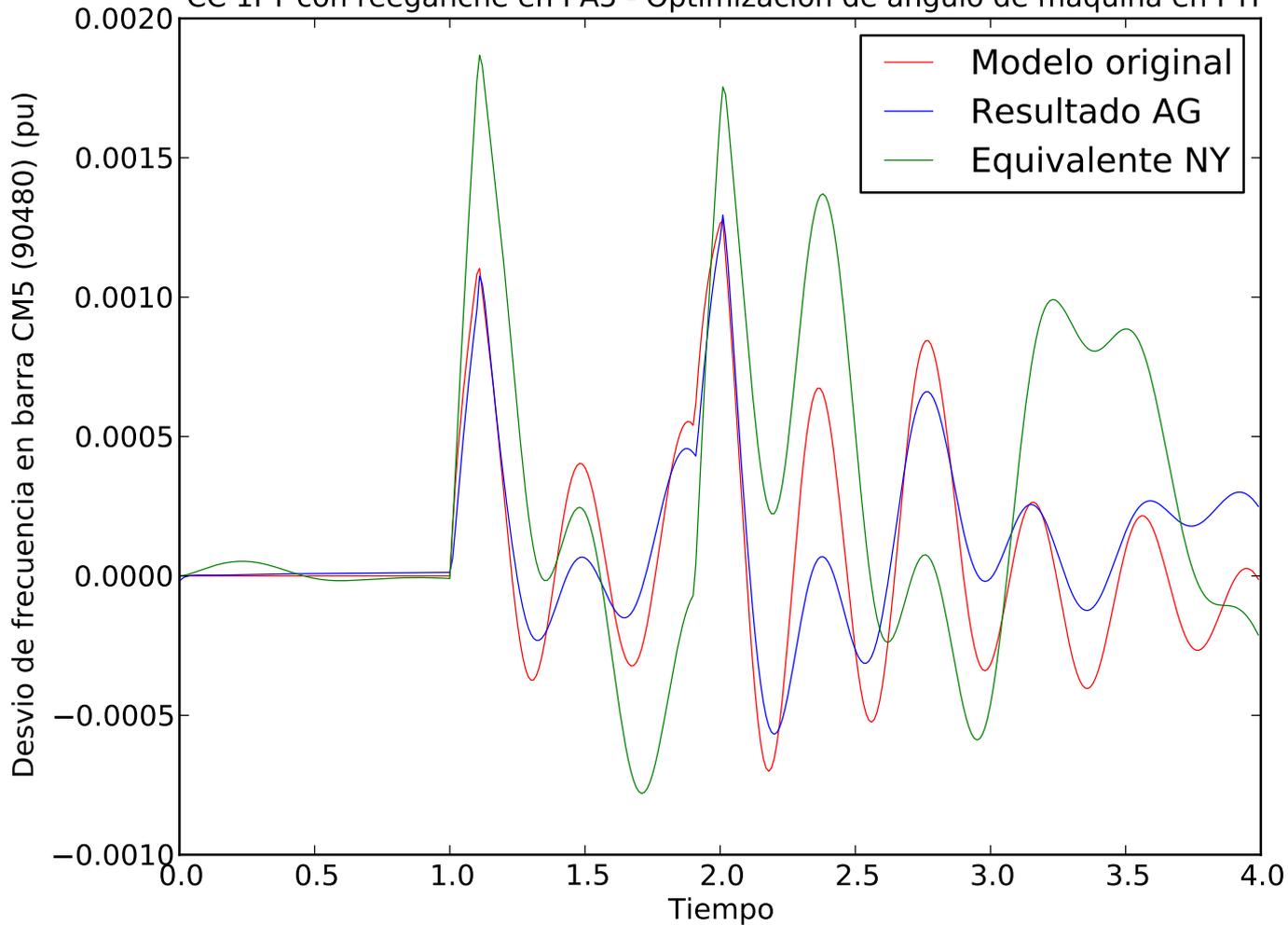
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



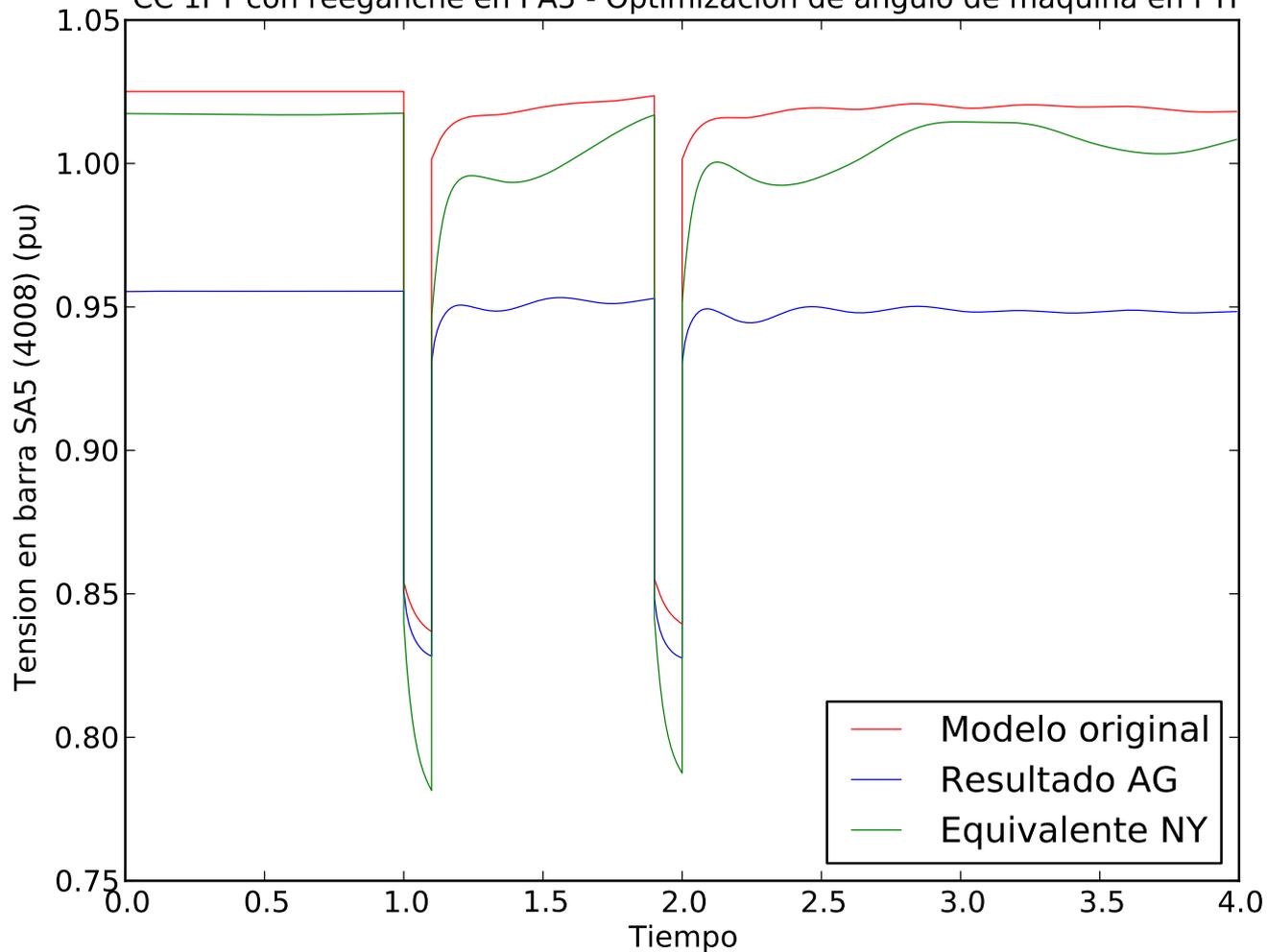
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



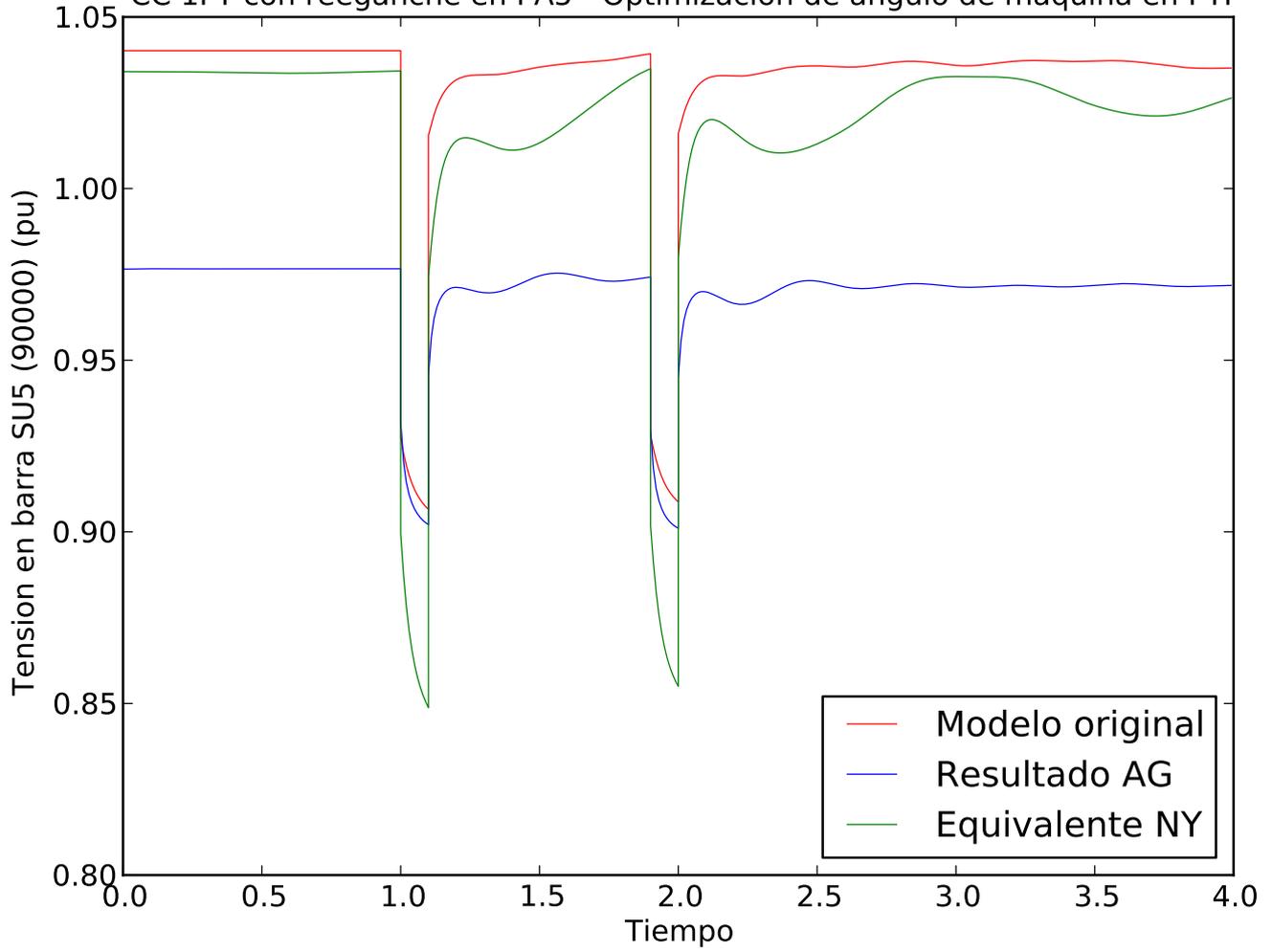
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



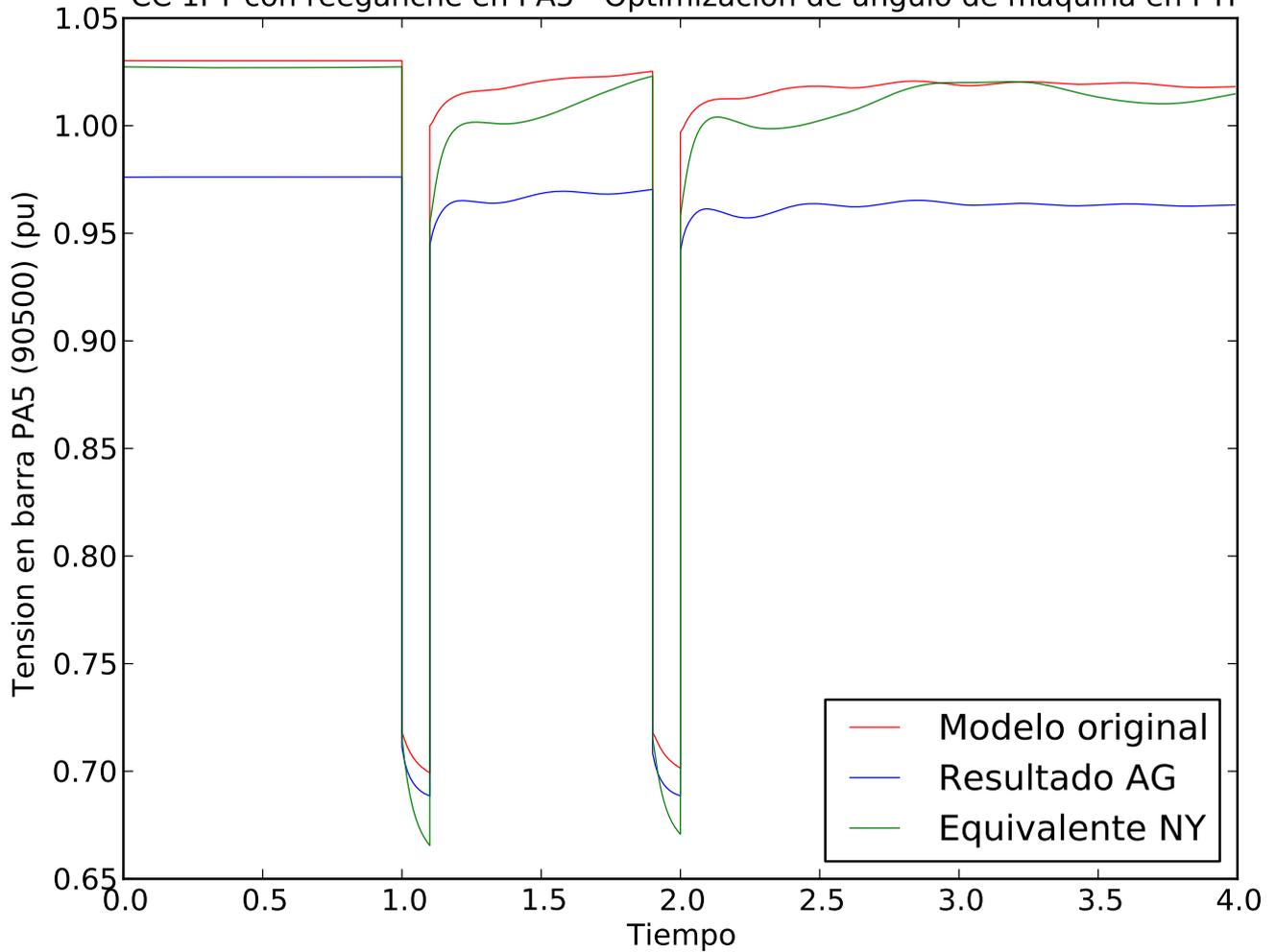
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



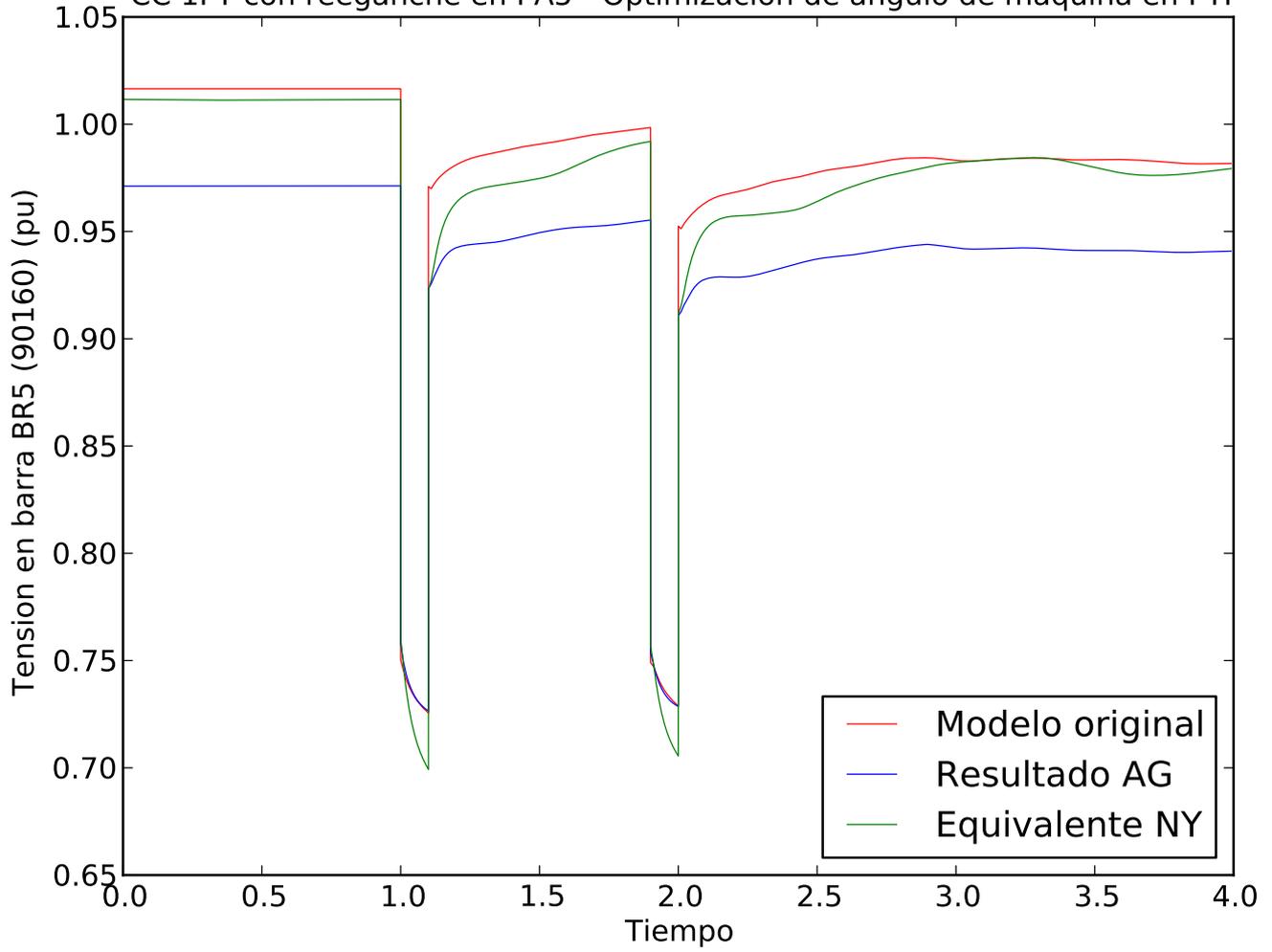
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



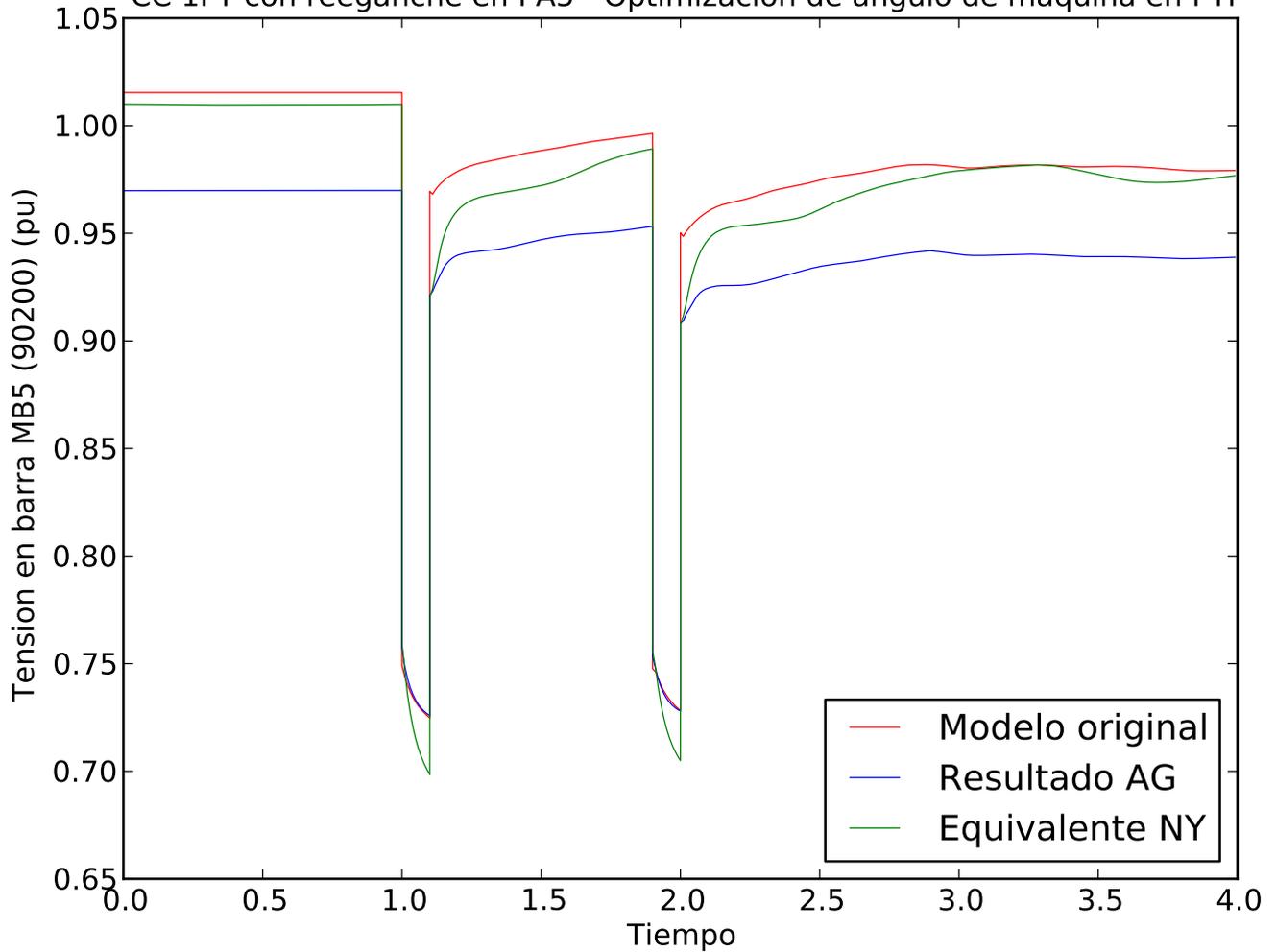
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



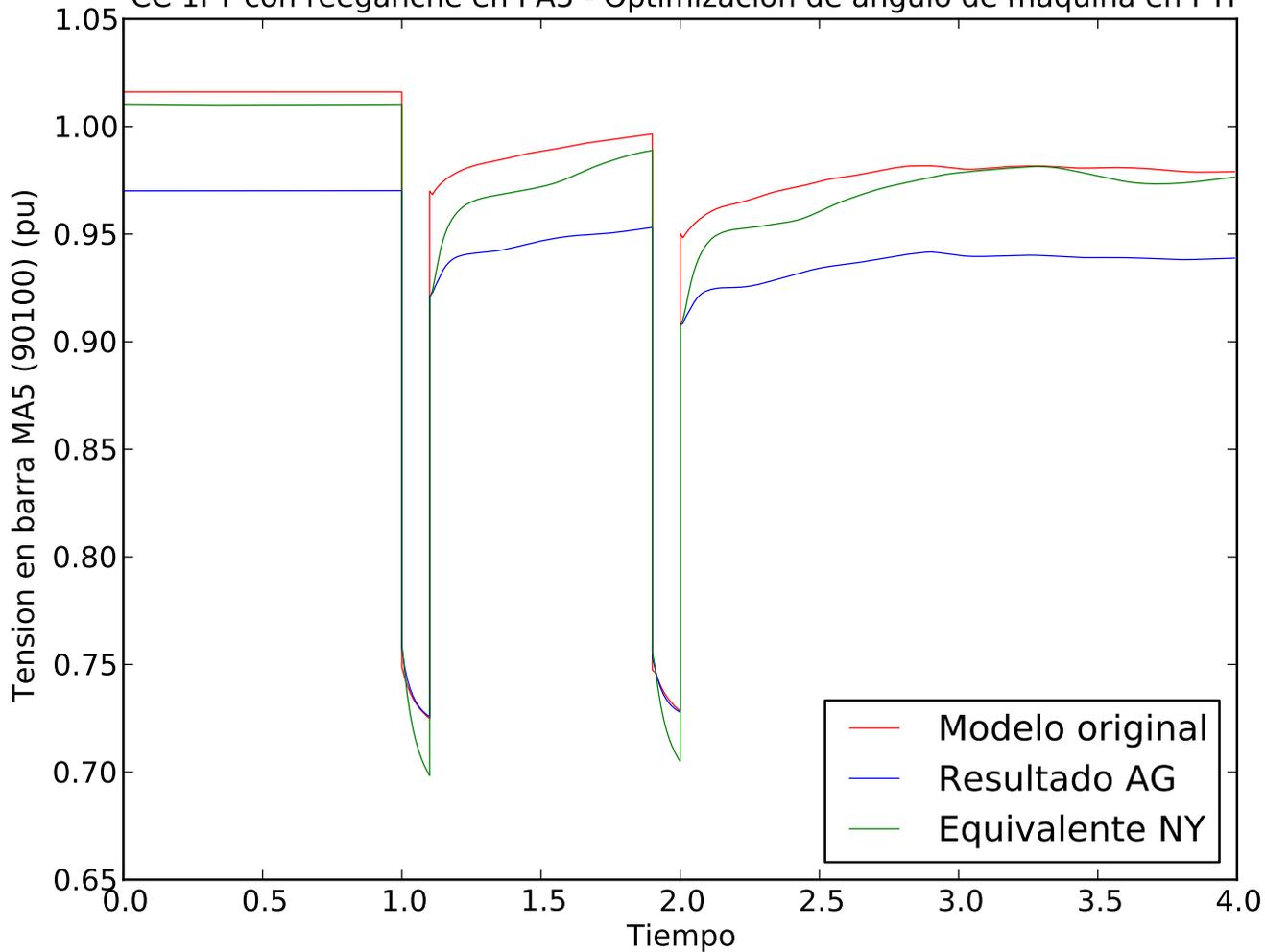
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



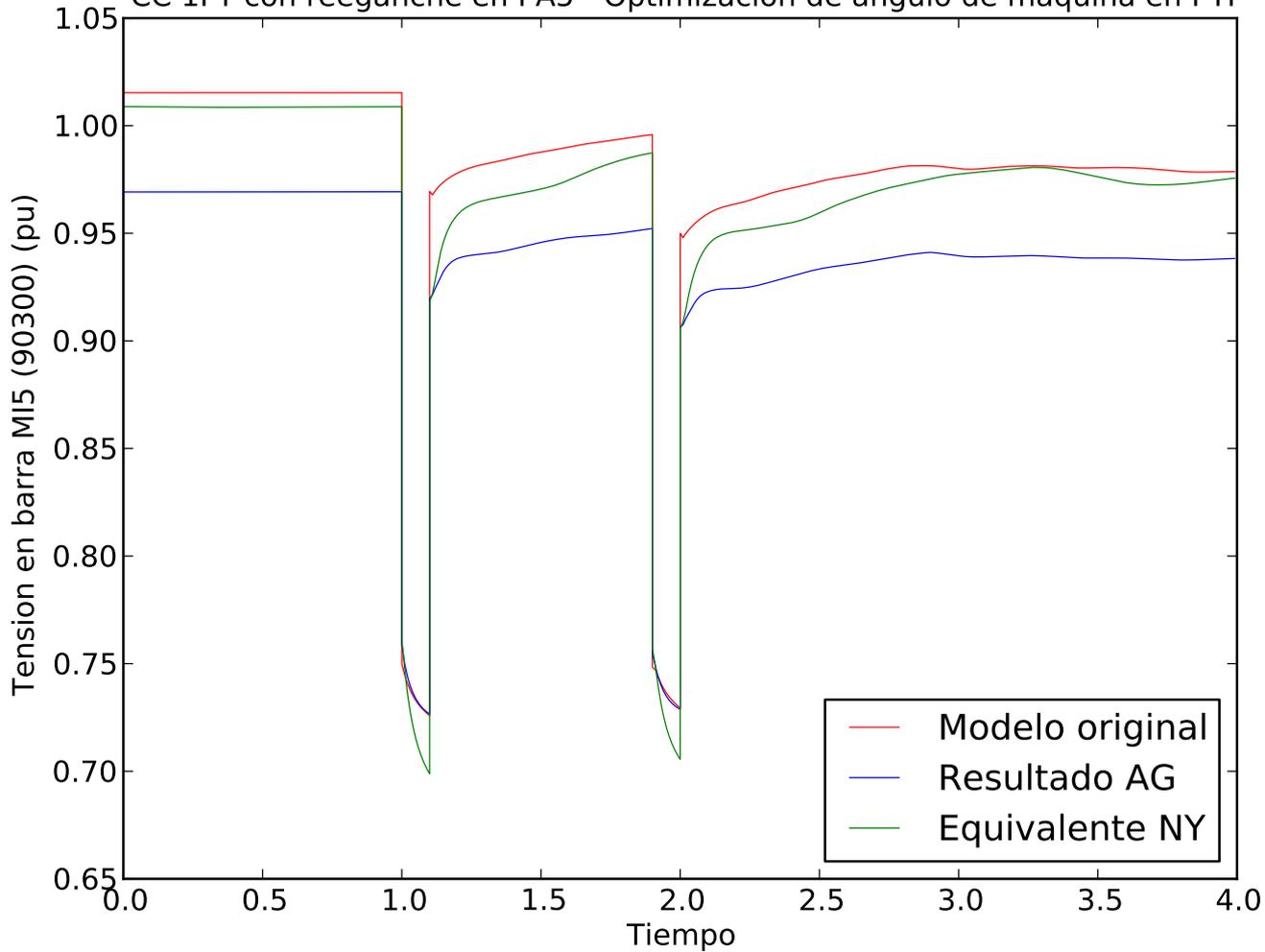
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



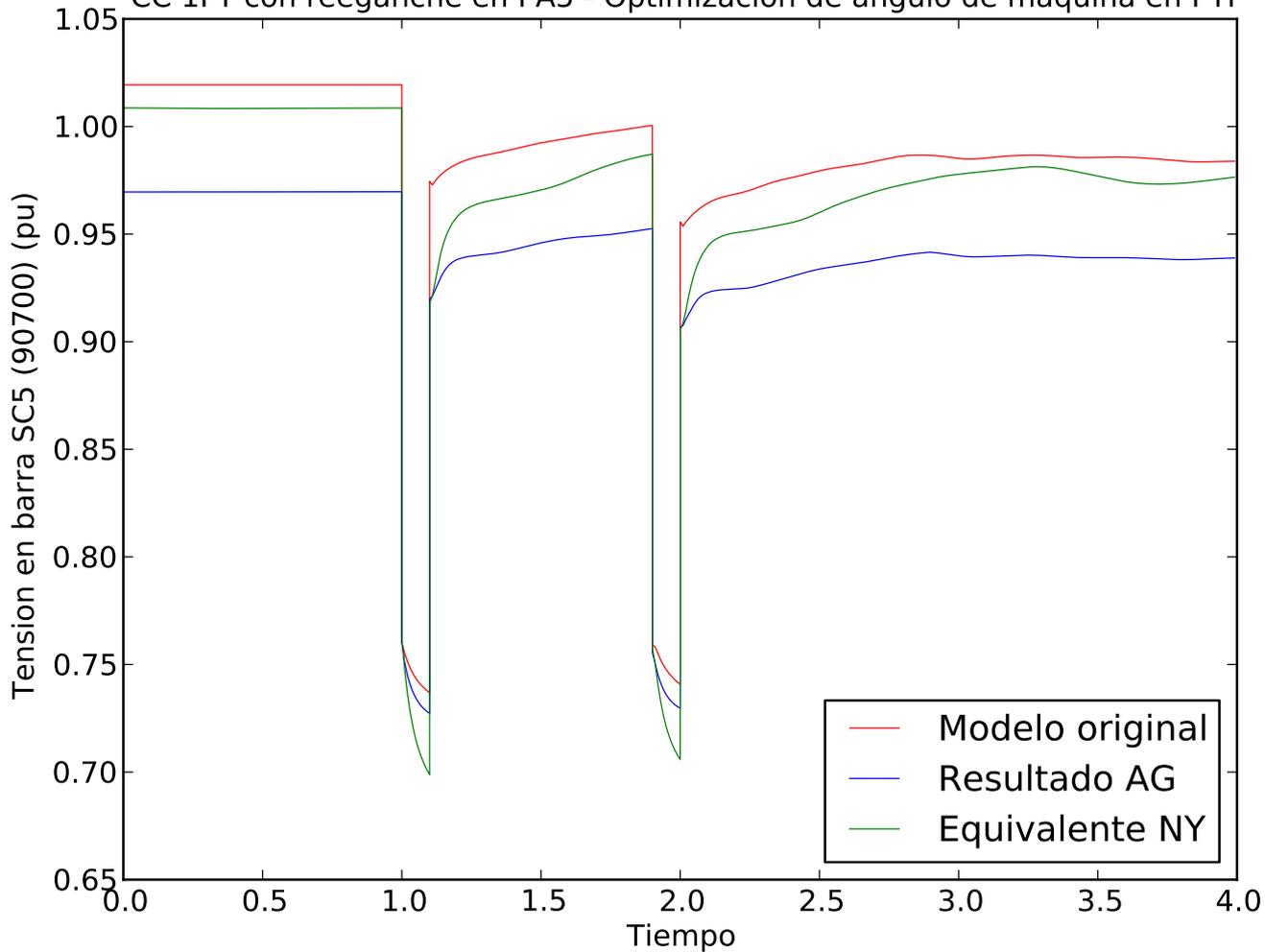
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



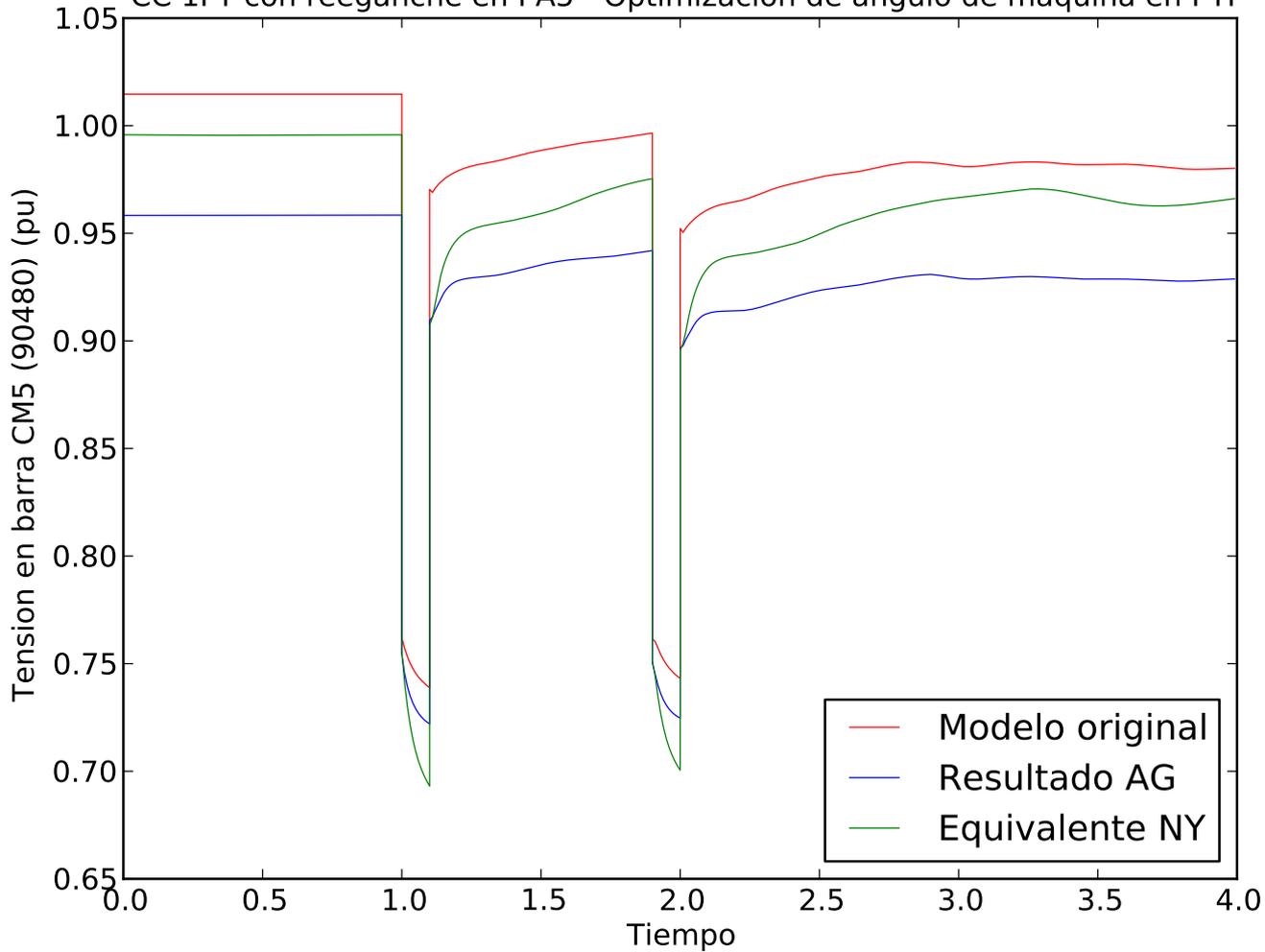
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



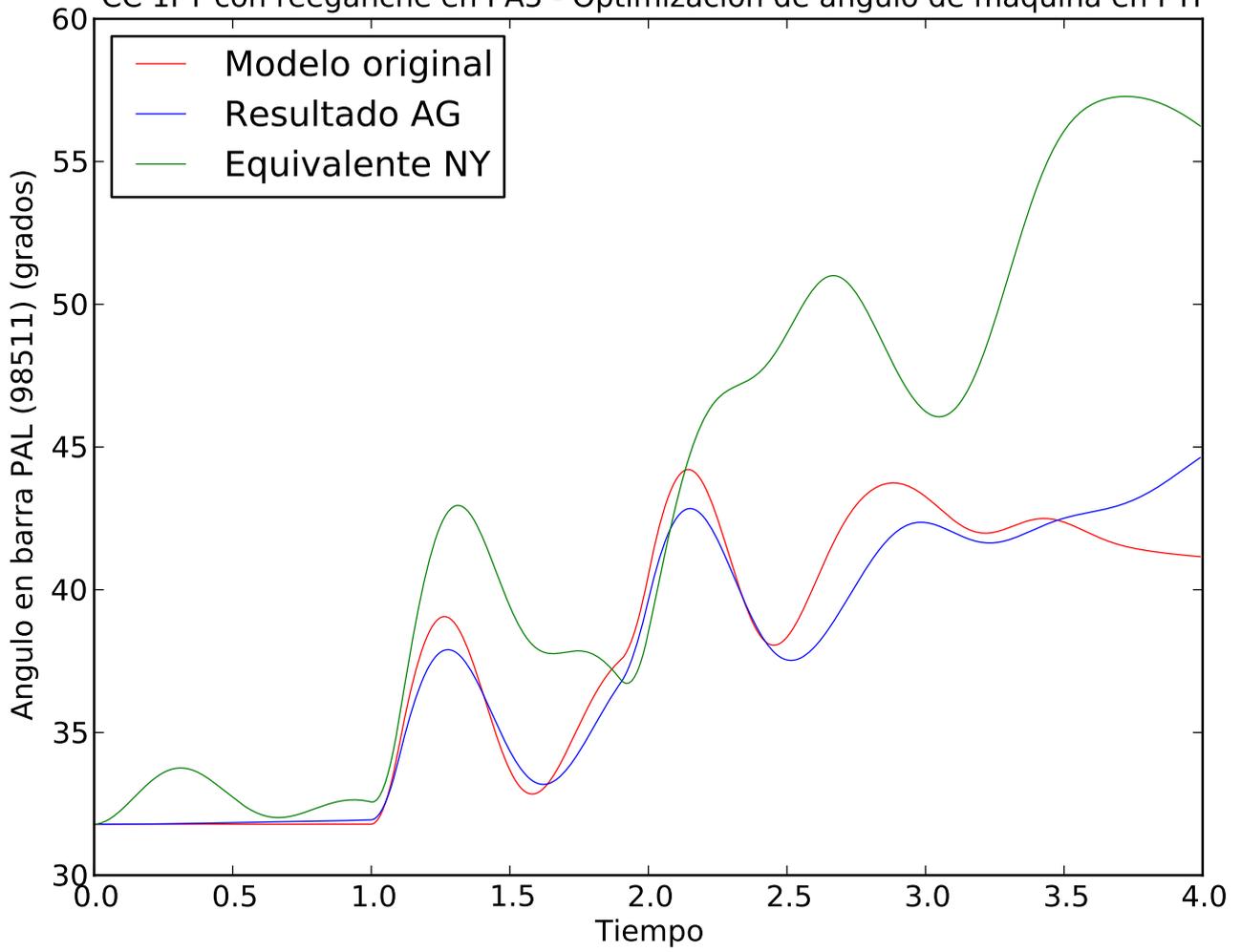
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



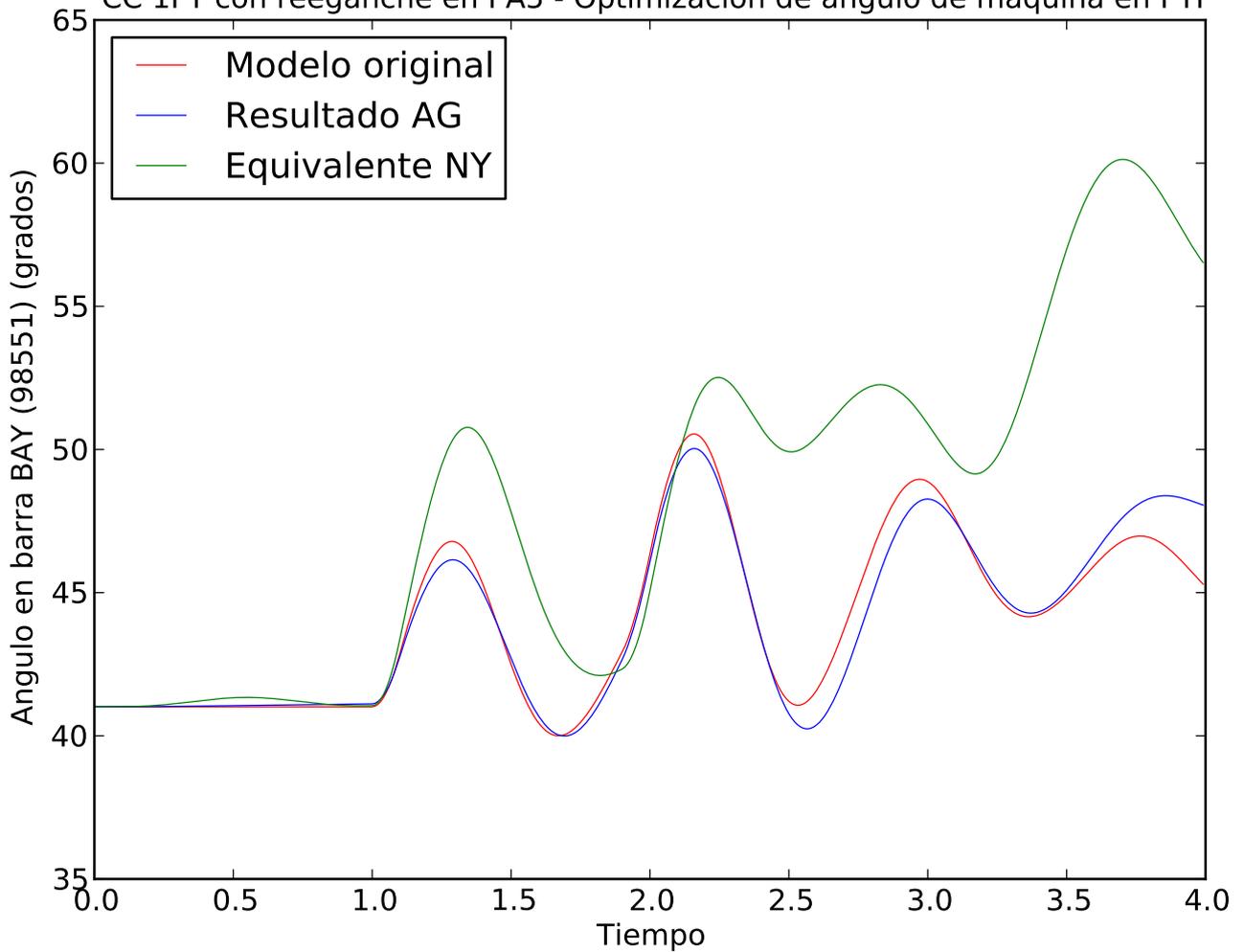
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



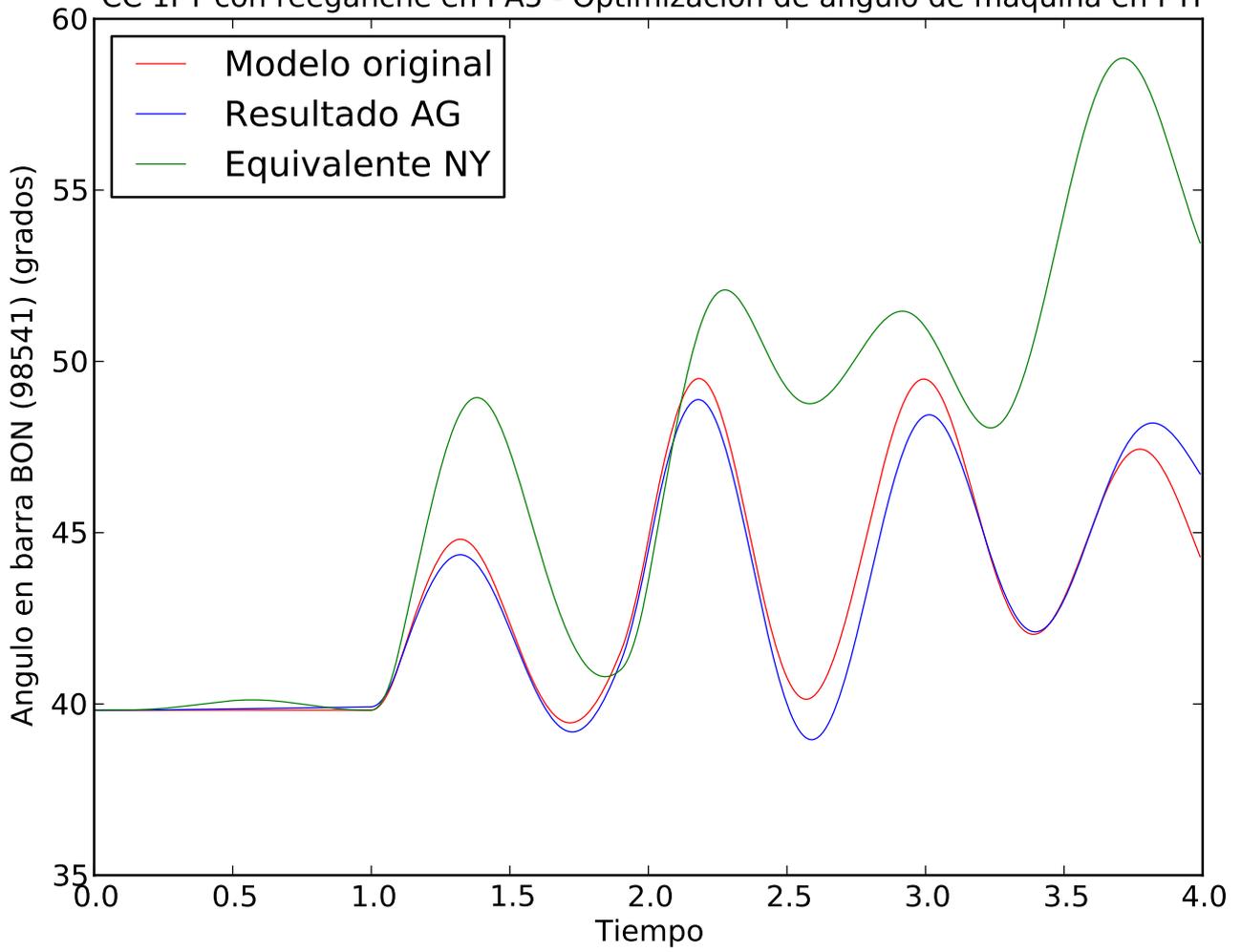
CC 1FT con reeganche en PA5 - Optimizacion de angulo de maquina en PTI



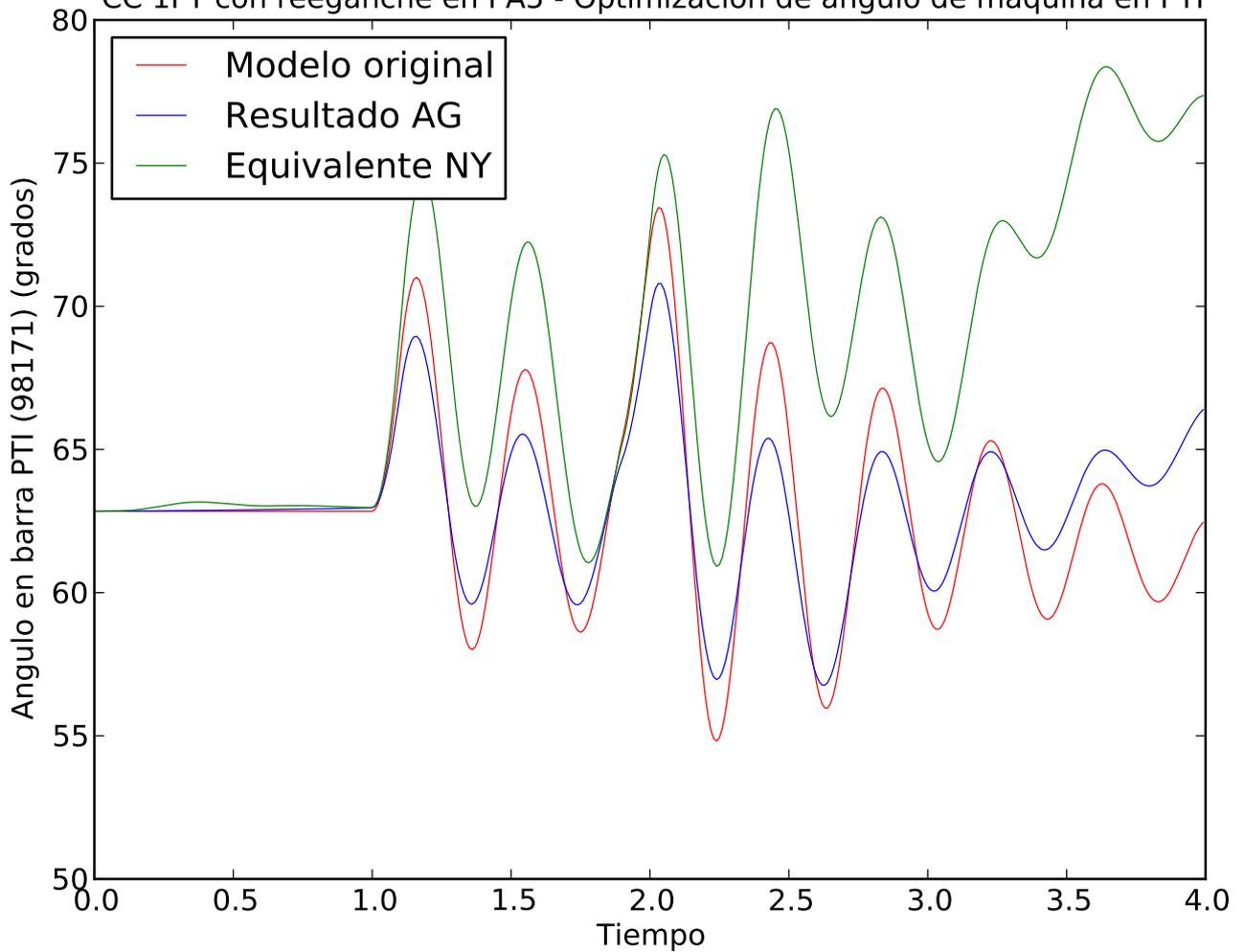
CC 1FT con reeganche en PA5 - Optimizacion de angulo de maquina en PTI



CC 1FT con reeganche en PA5 - Optimizacion de angulo de maquina en PTI

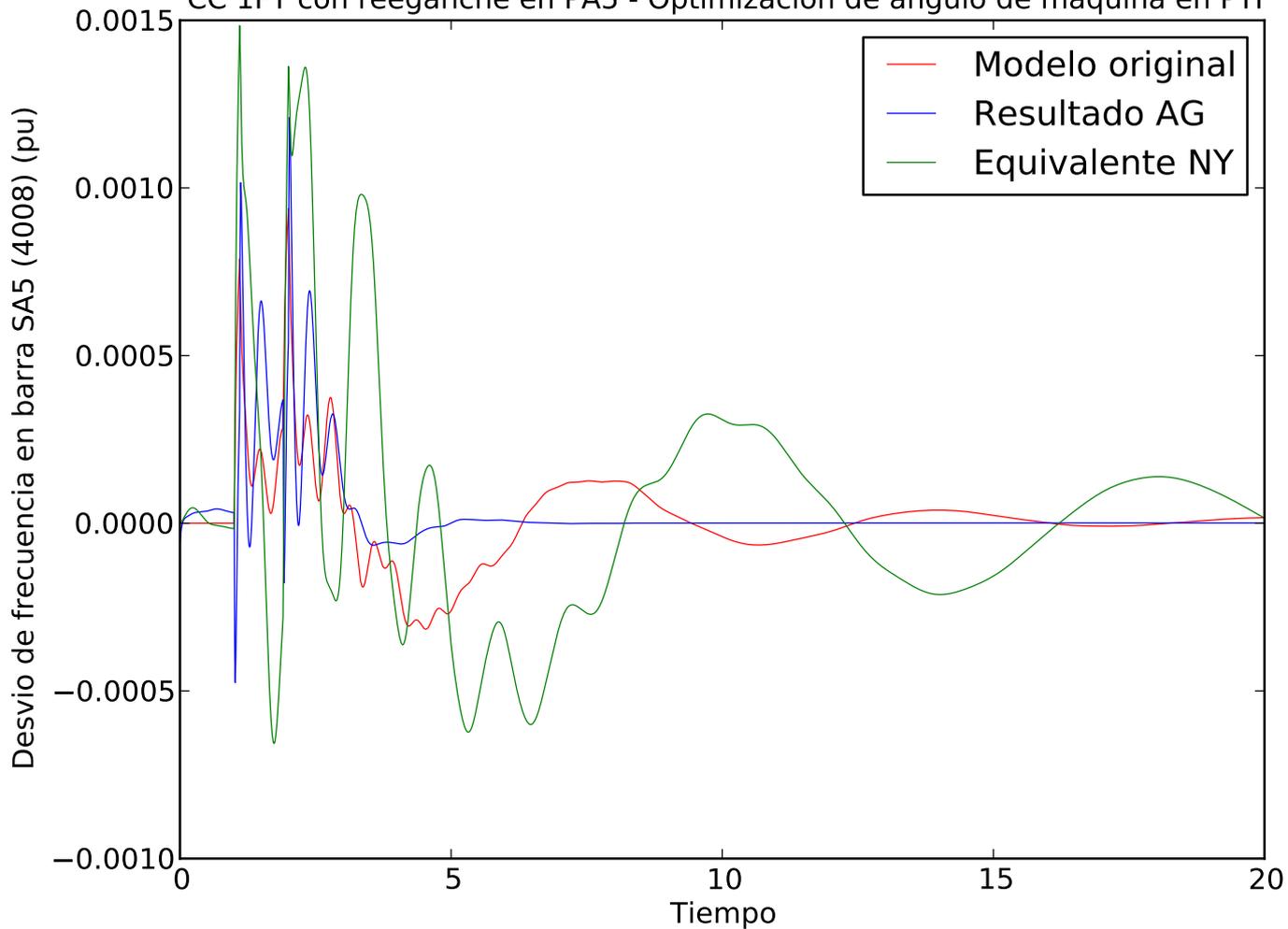


CC 1FT con reeganche en PA5 - Optimizacion de angulo de maquina en PTI

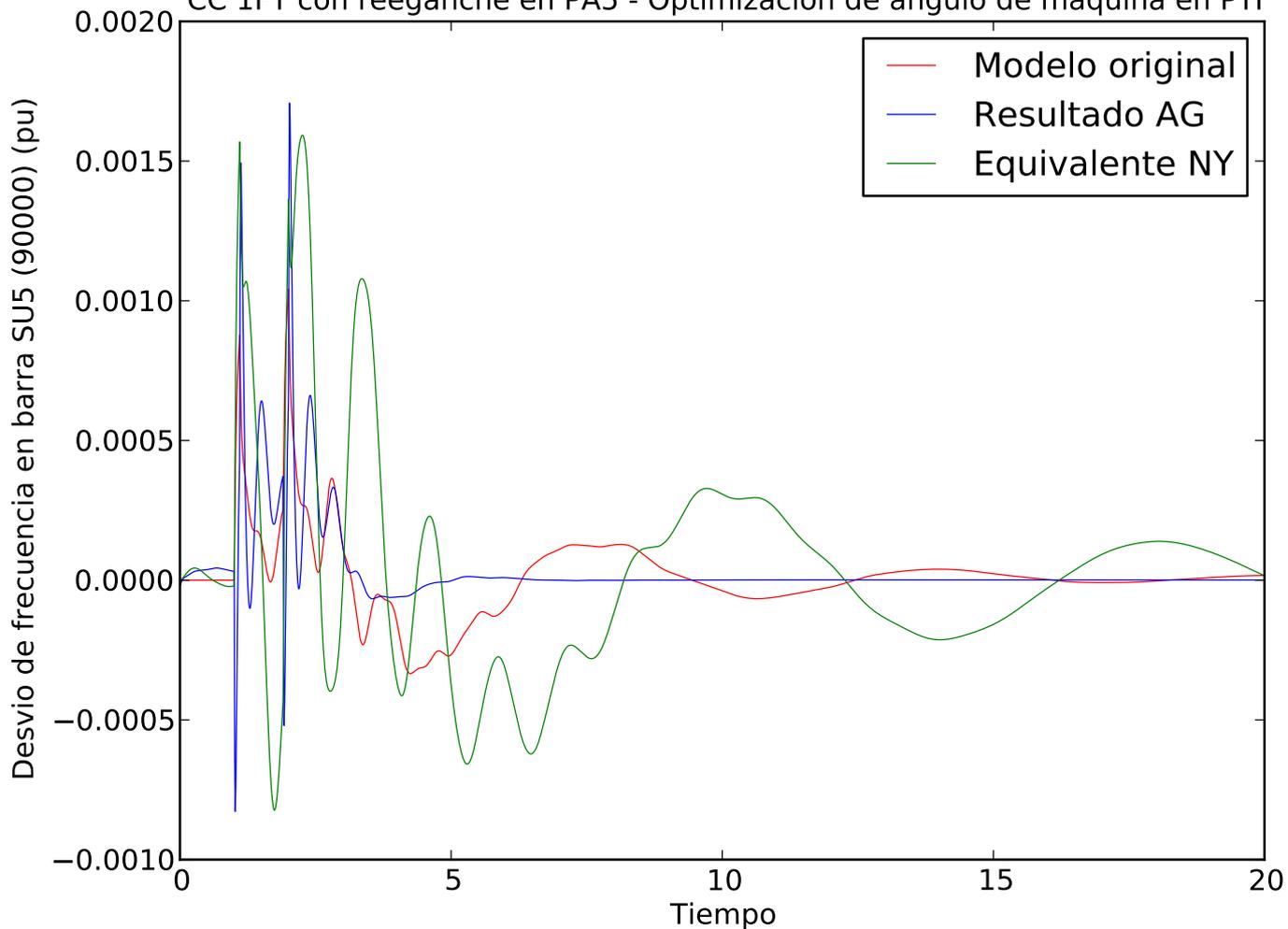


- C.14. Simulación 9.2: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando ángulo de un máquina de PTI en 9 segundos. Comportamiento en 20 segundos**

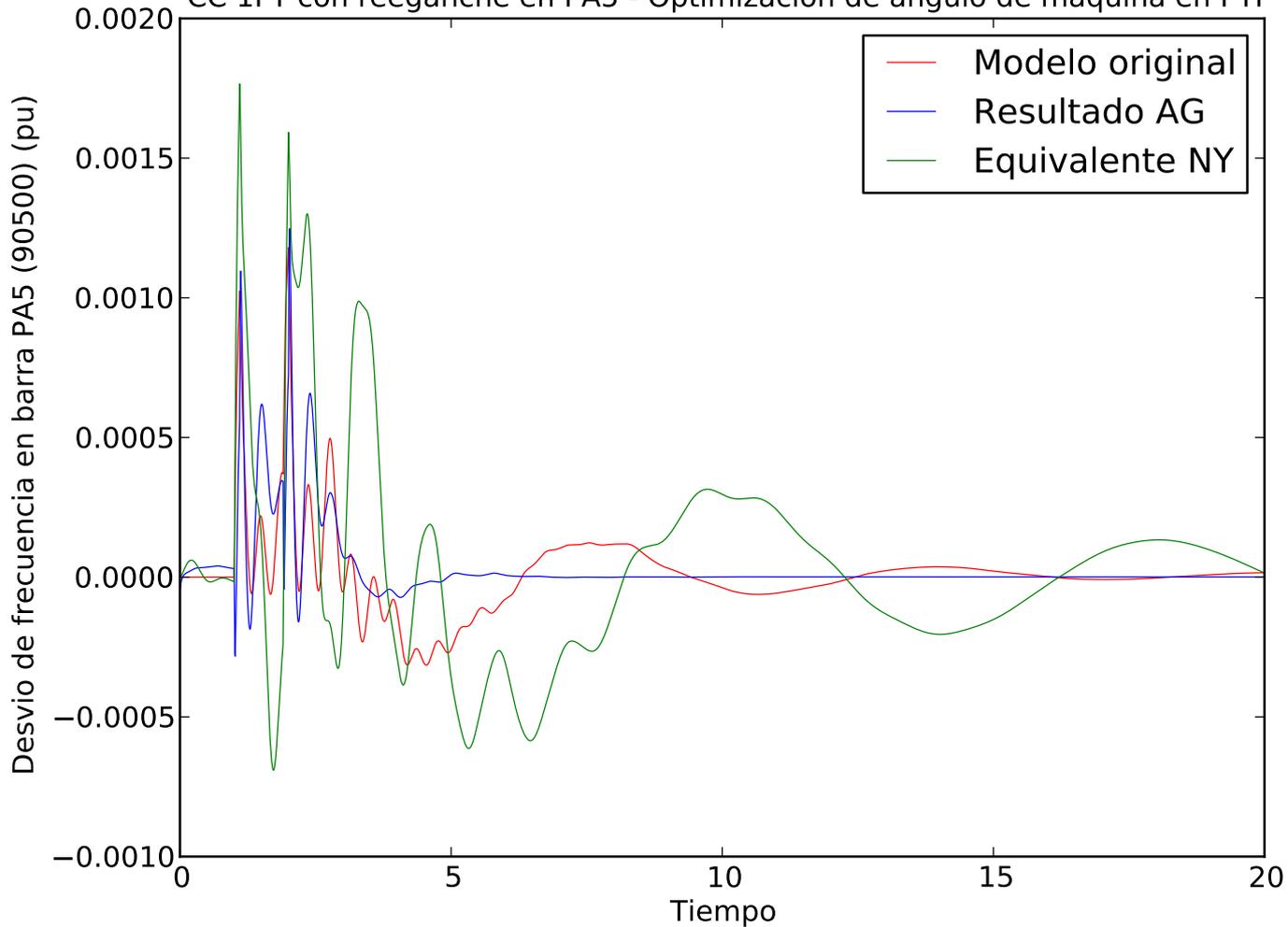
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



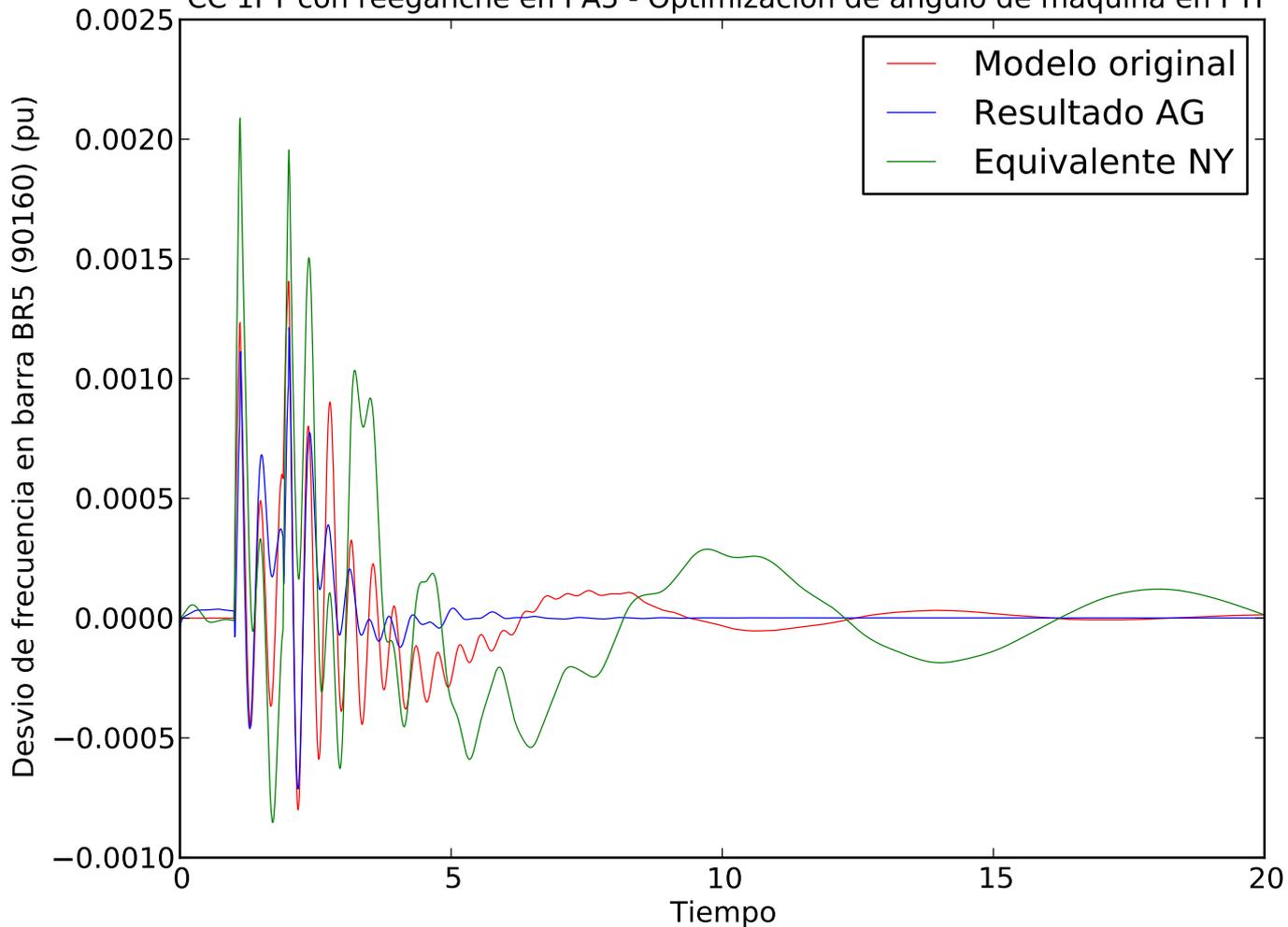
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



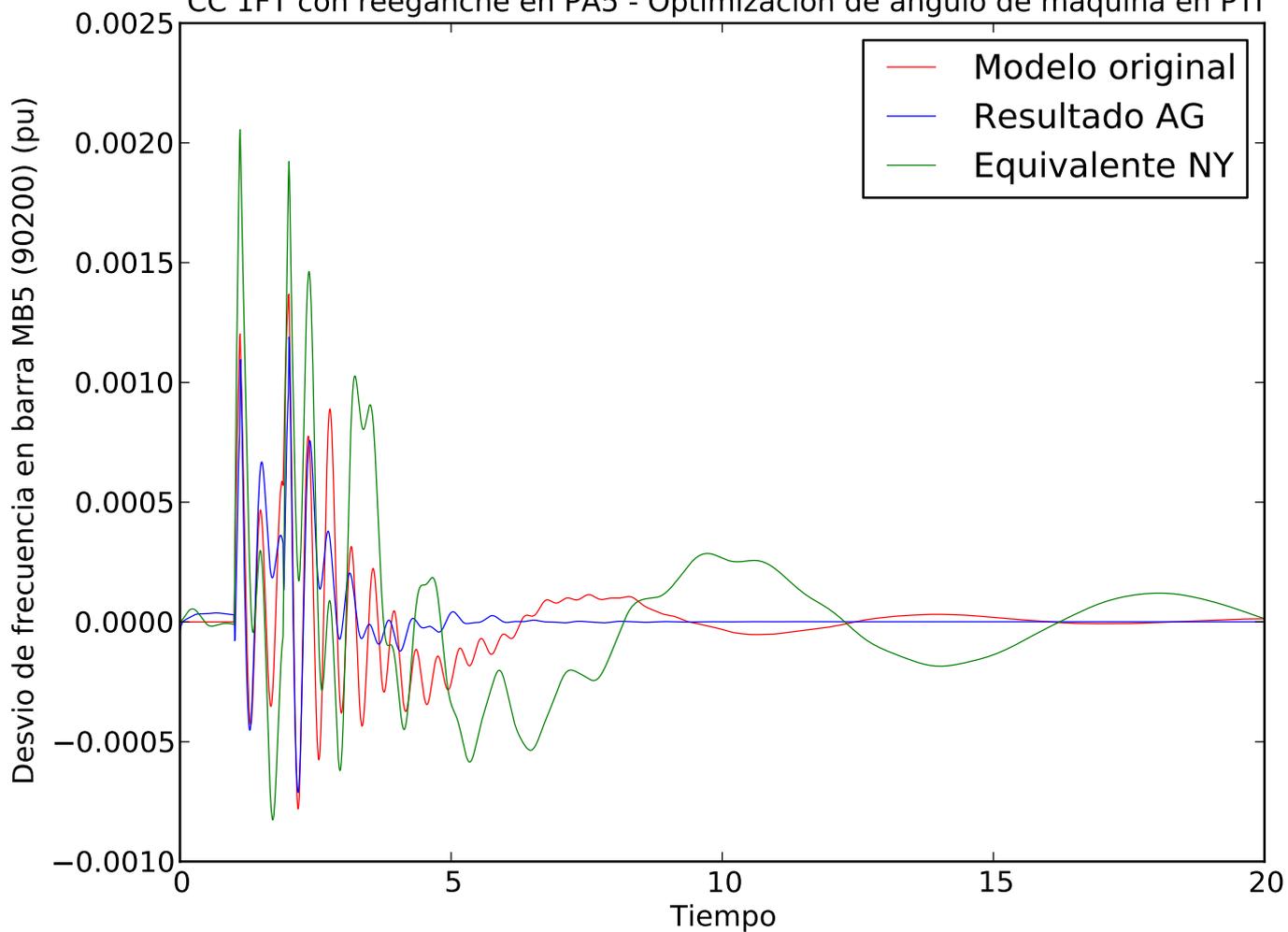
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



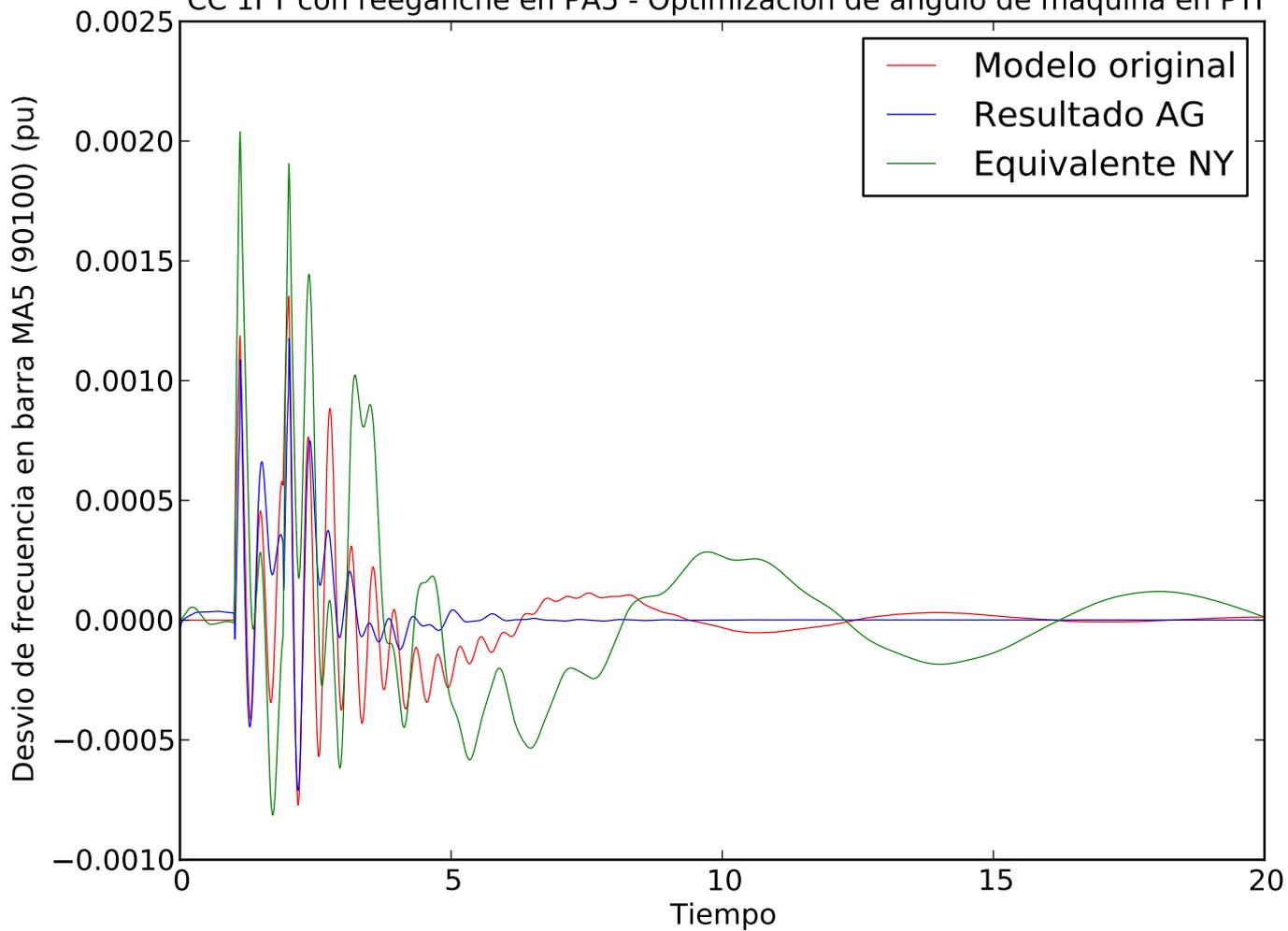
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



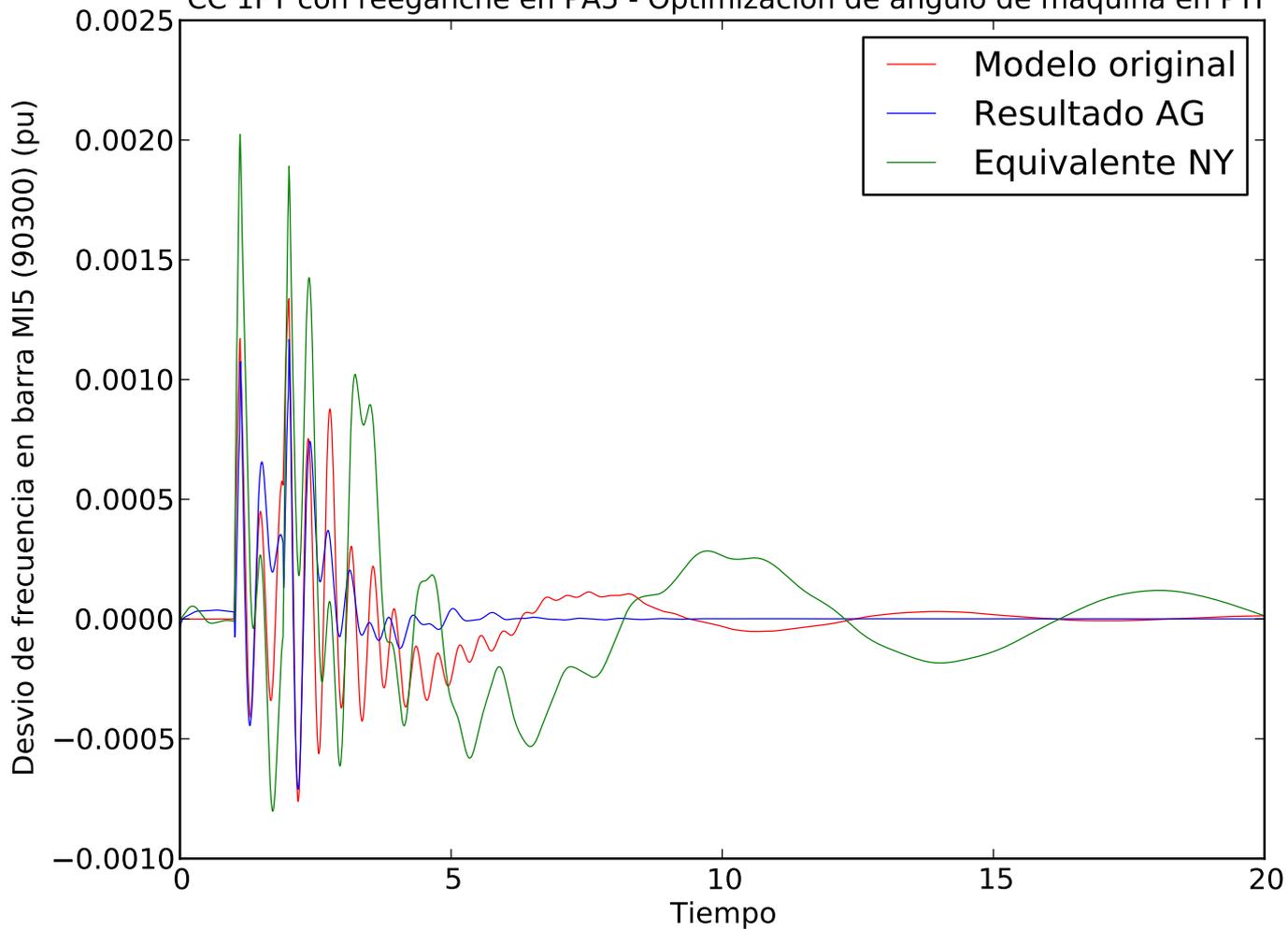
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



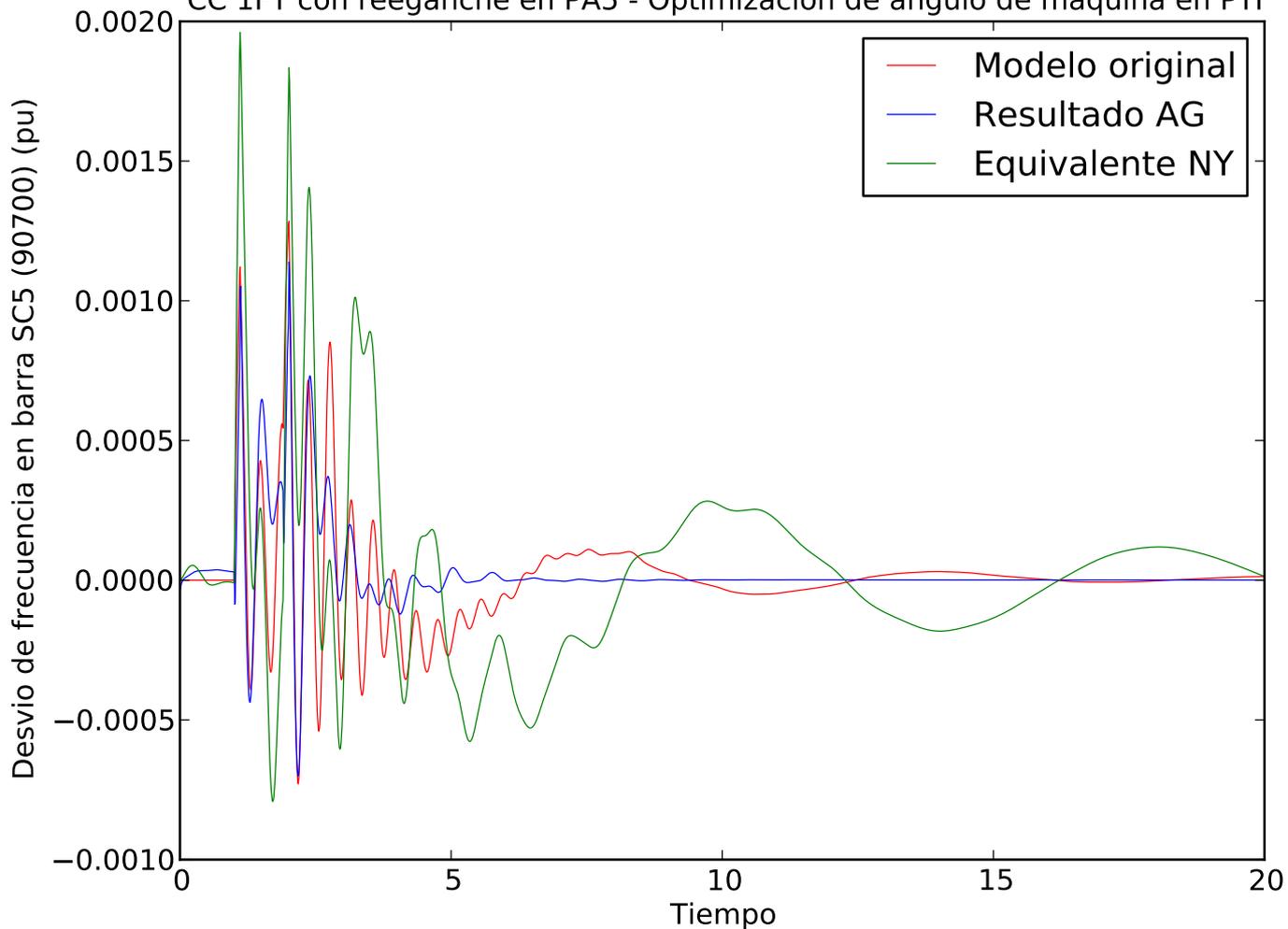
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



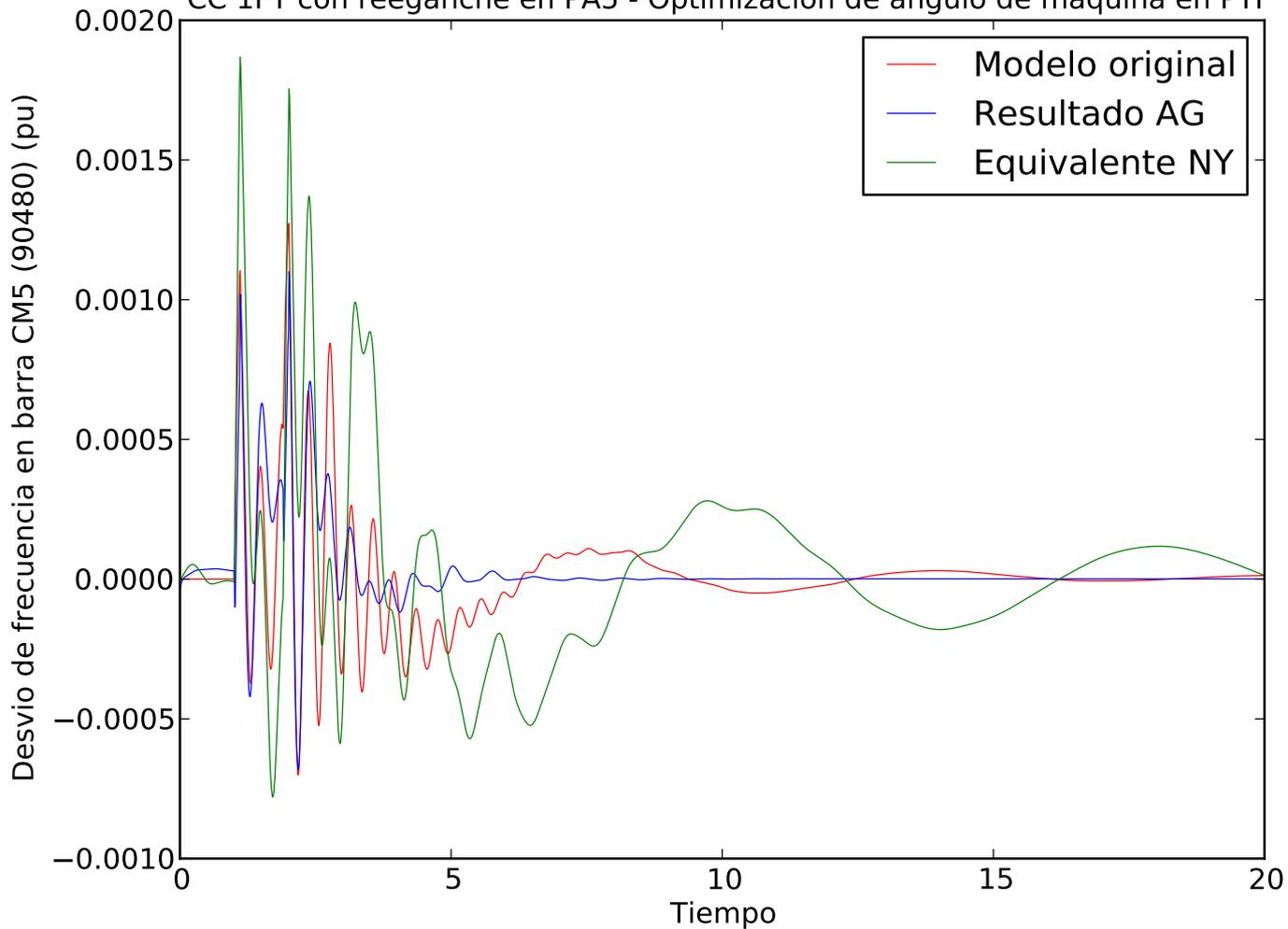
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



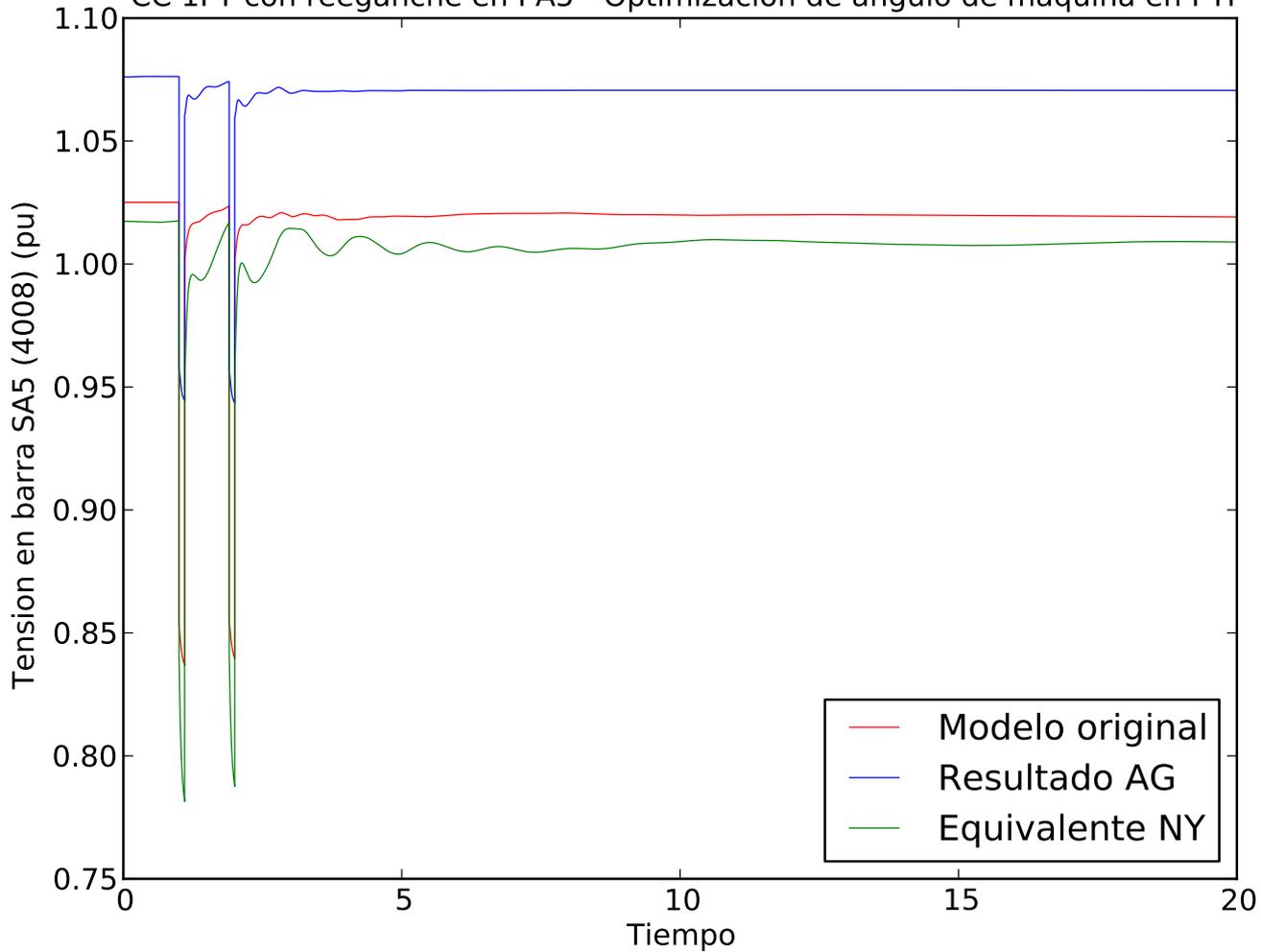
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



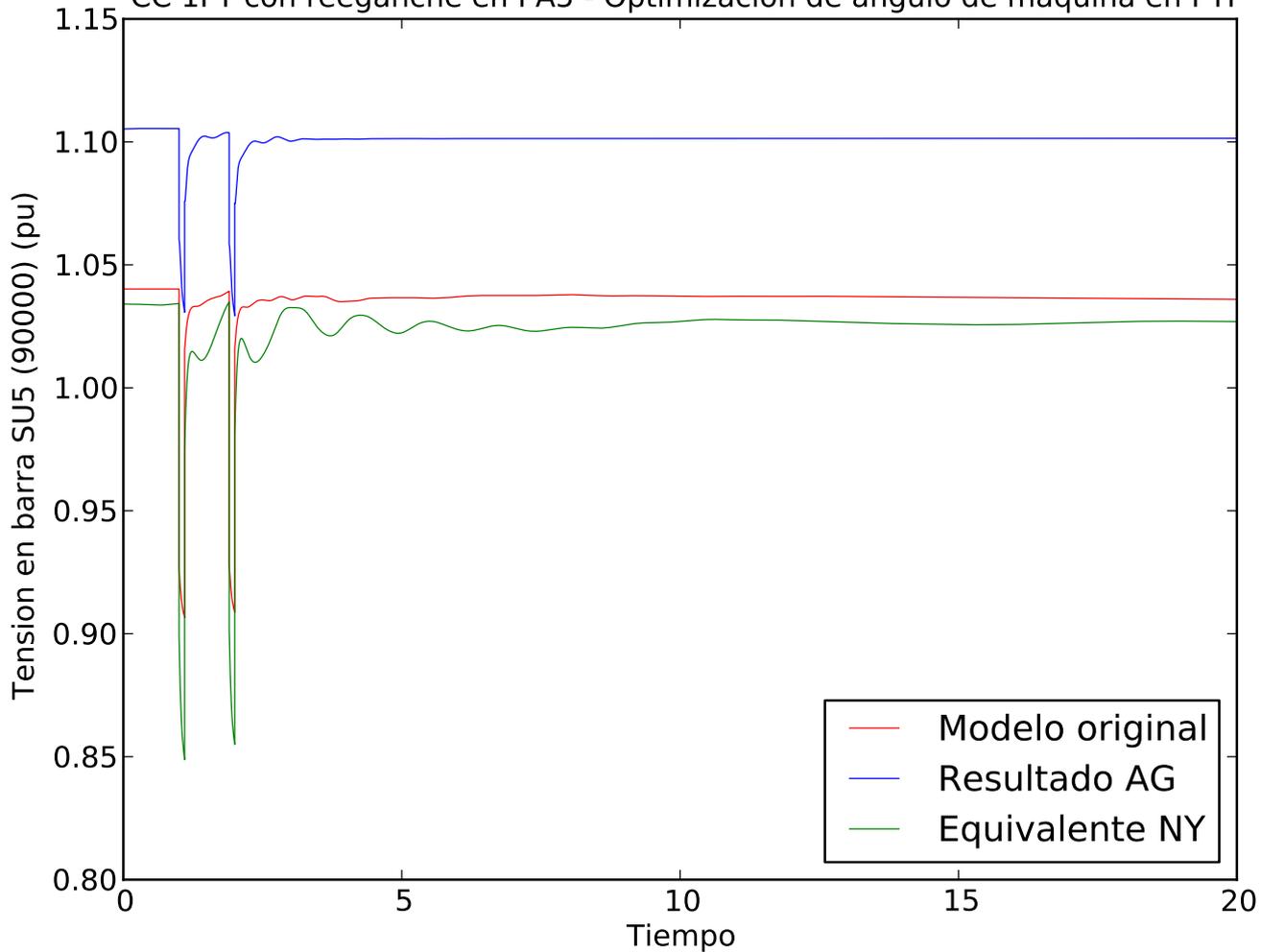
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



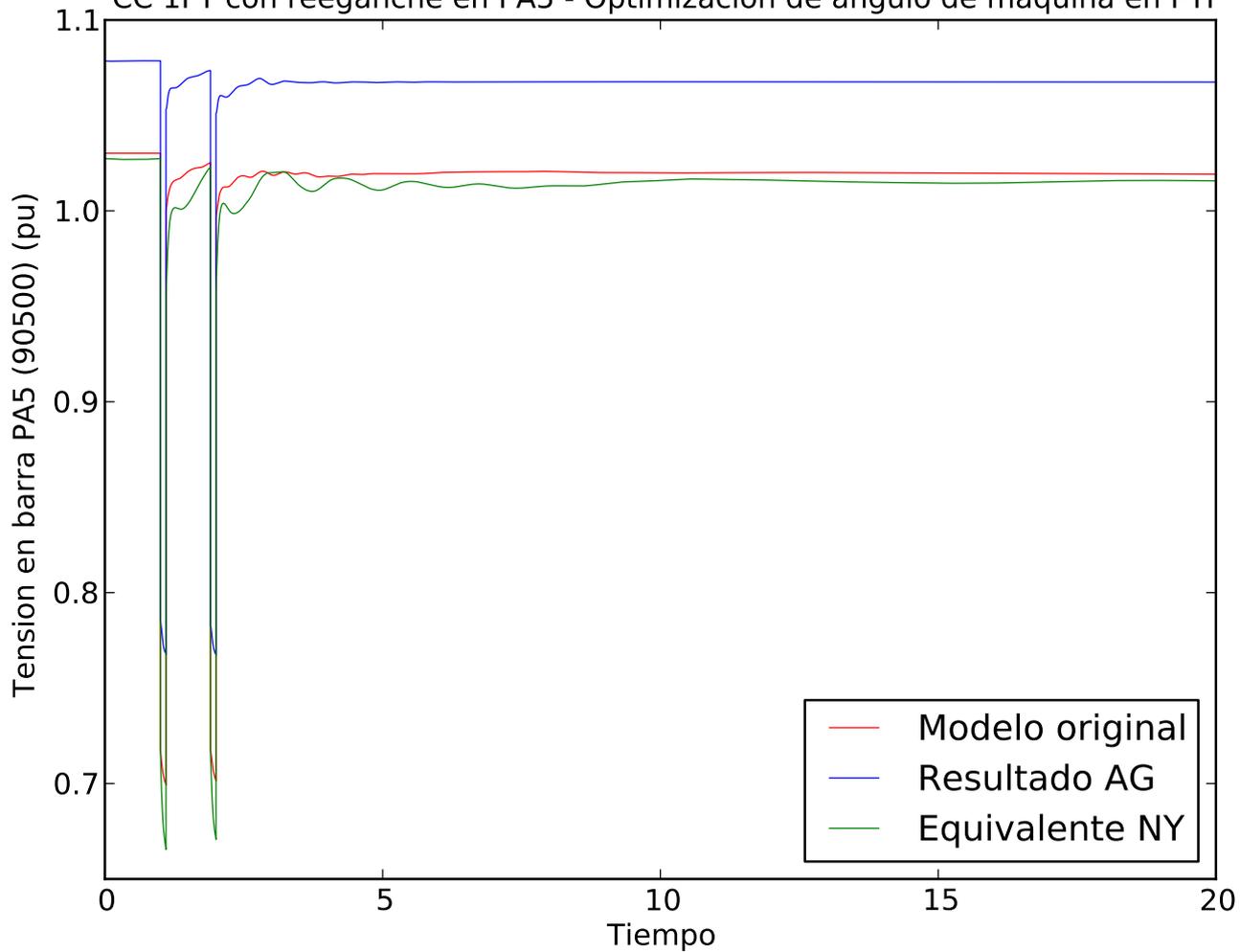
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



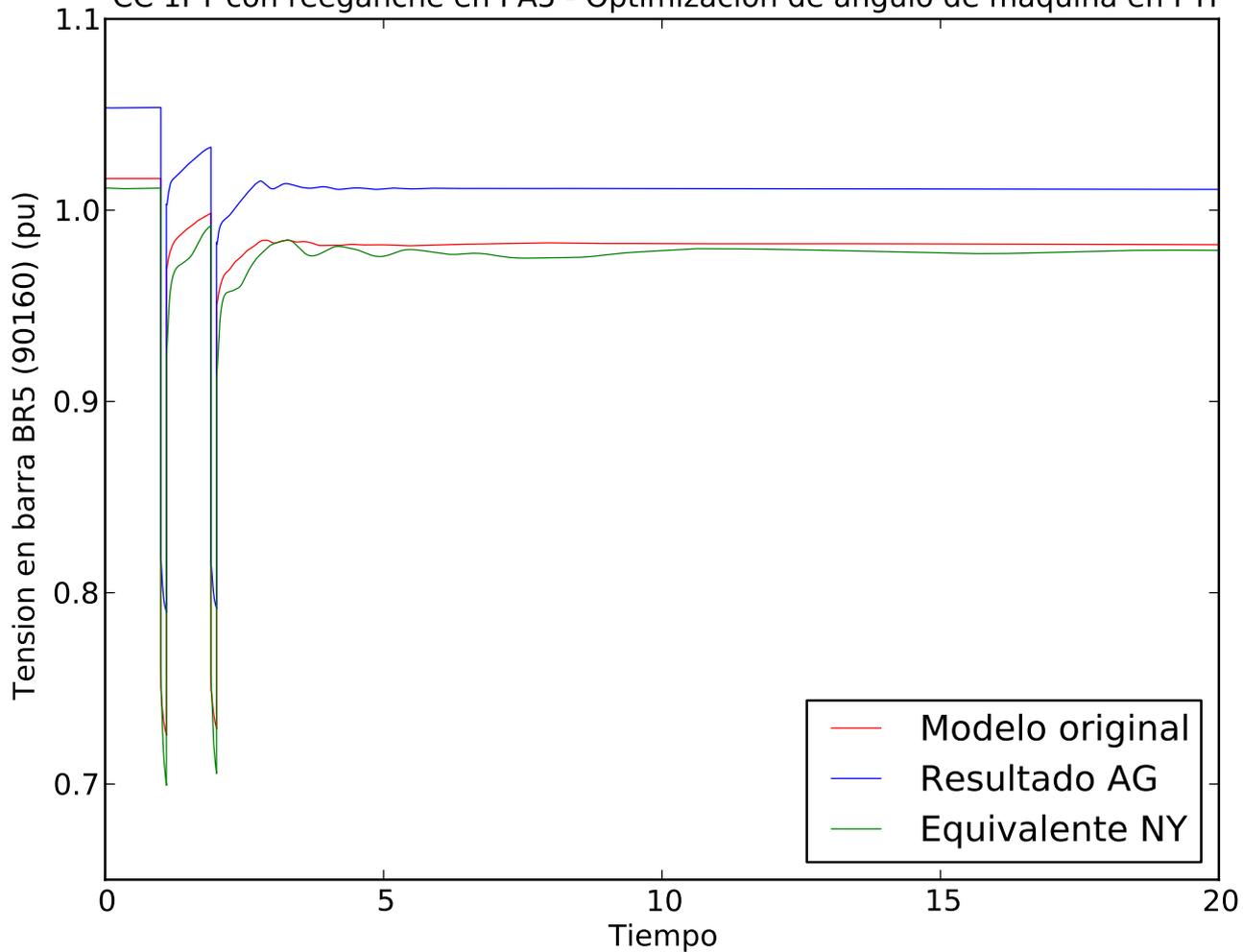
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



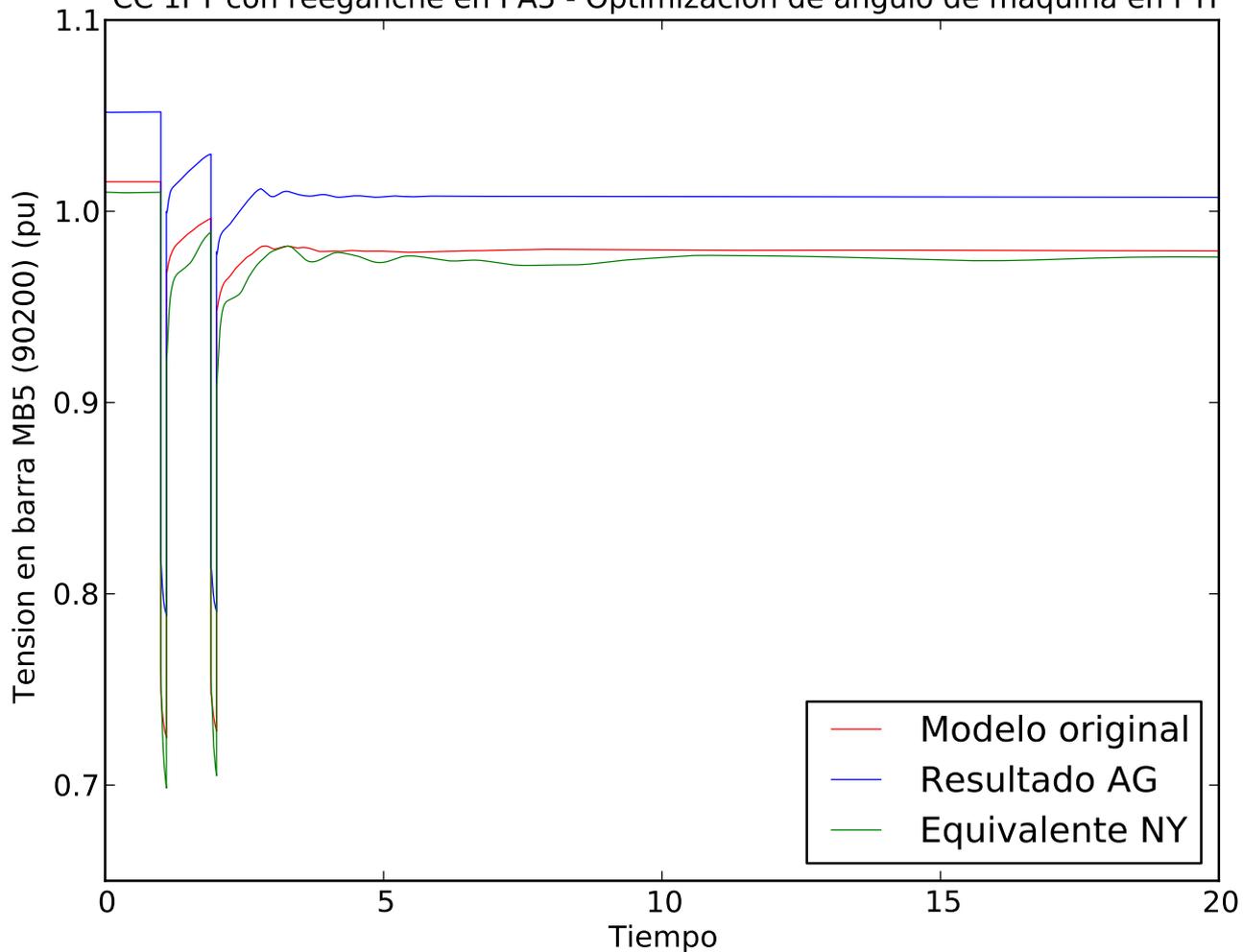
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



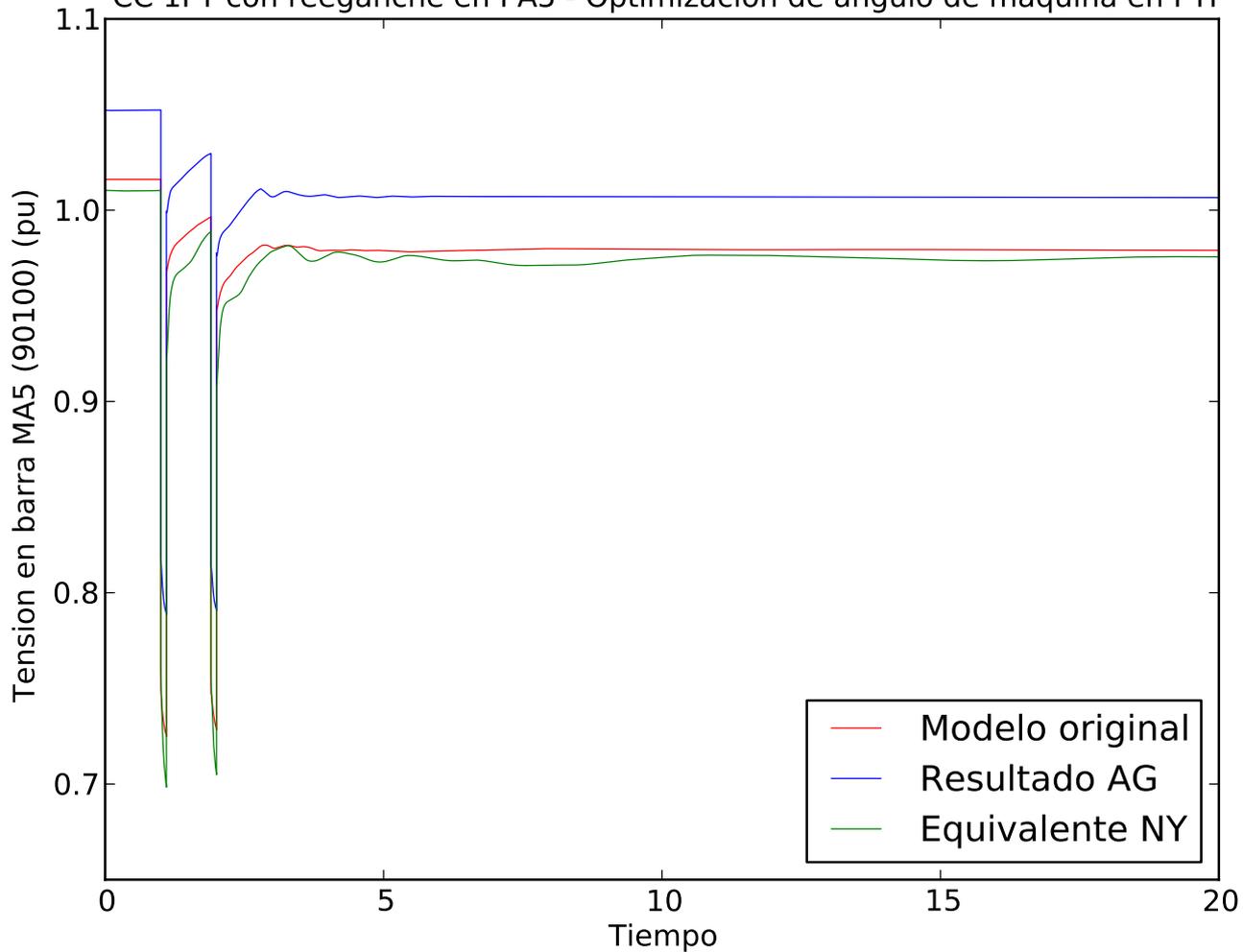
CC 1FT con reeganche en PA5 - Optimizacion de angulo de maquina en PTI



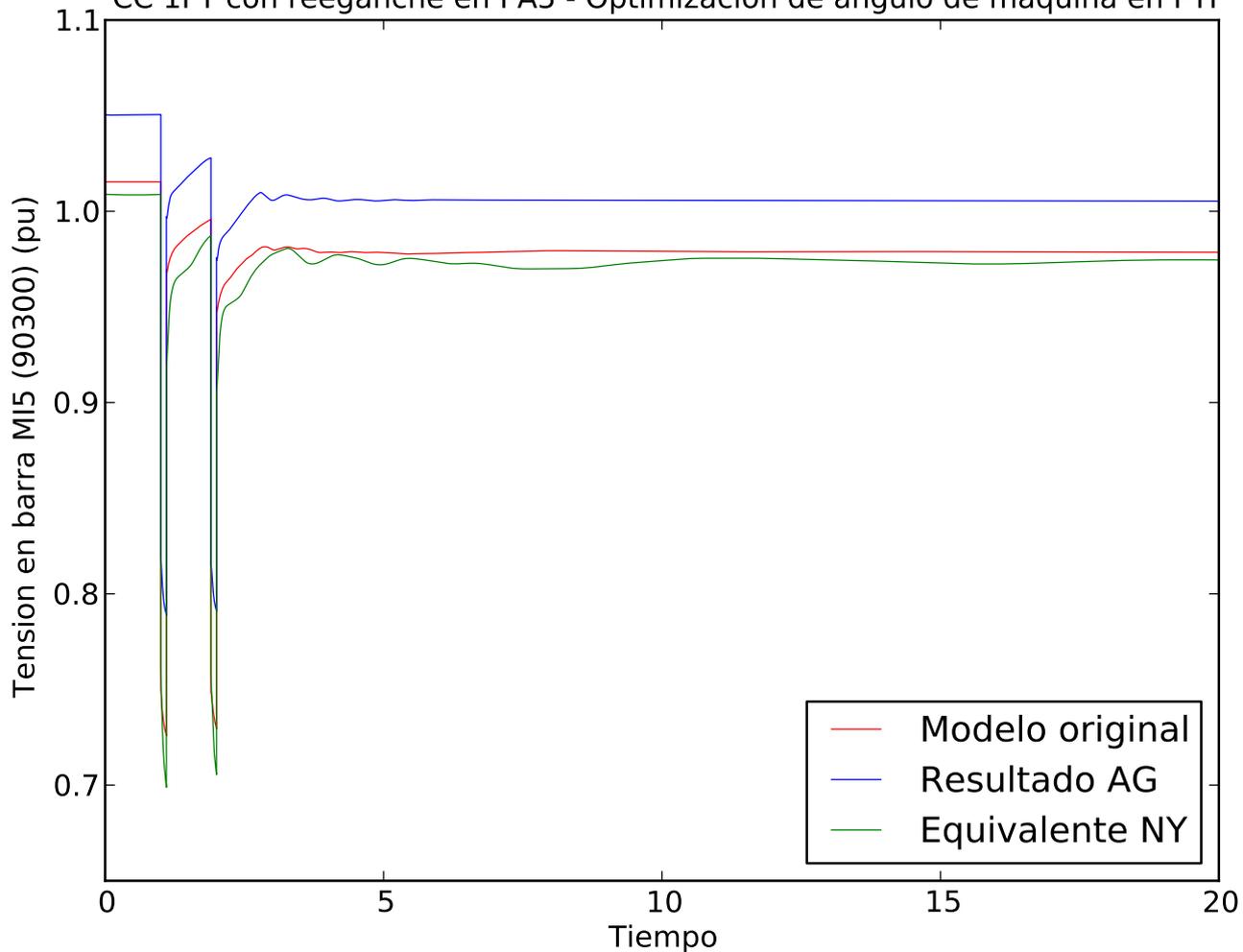
CC 1FT con reeganche en PA5 - Optimizacion de angulo de maquina en PTI



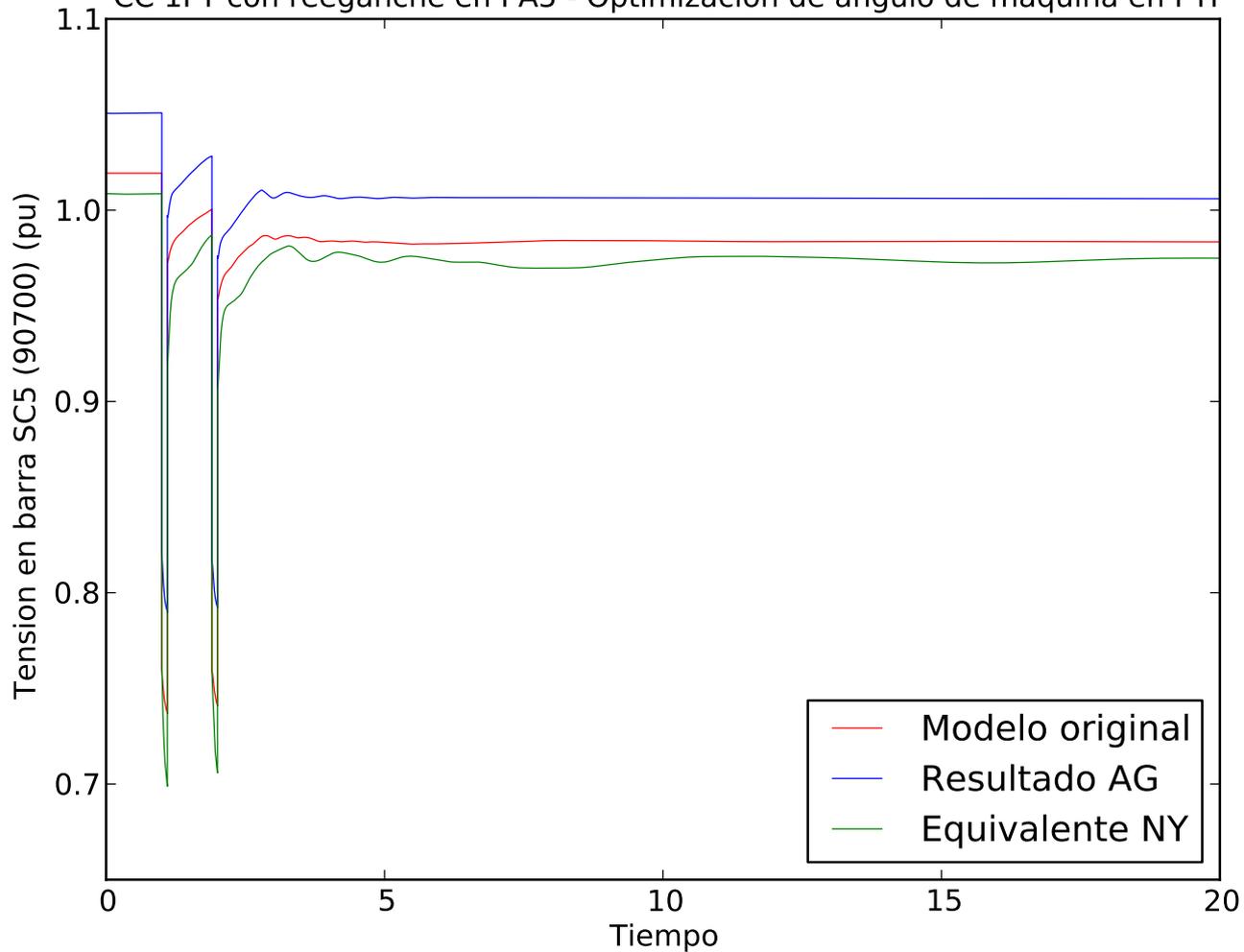
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



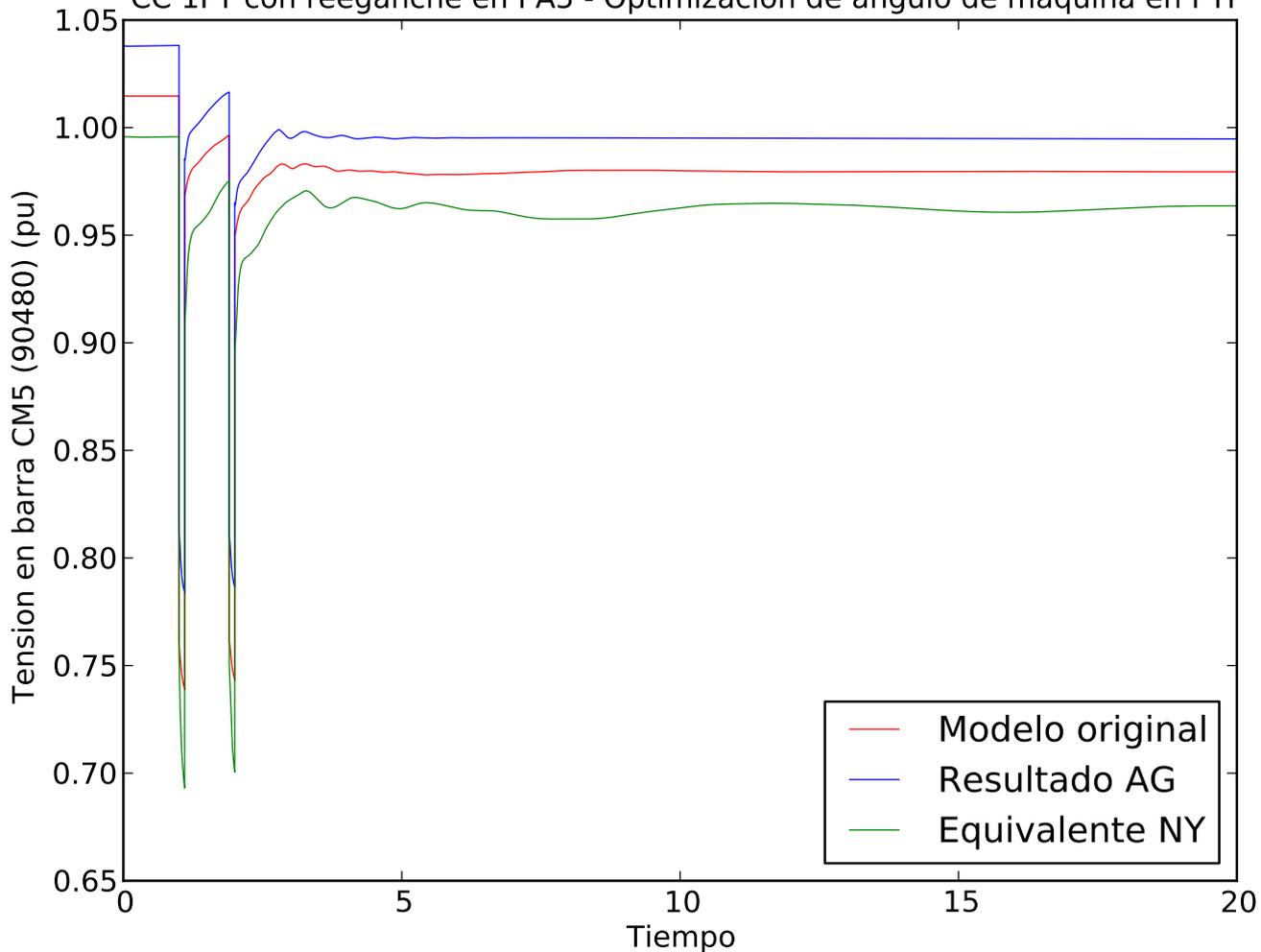
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



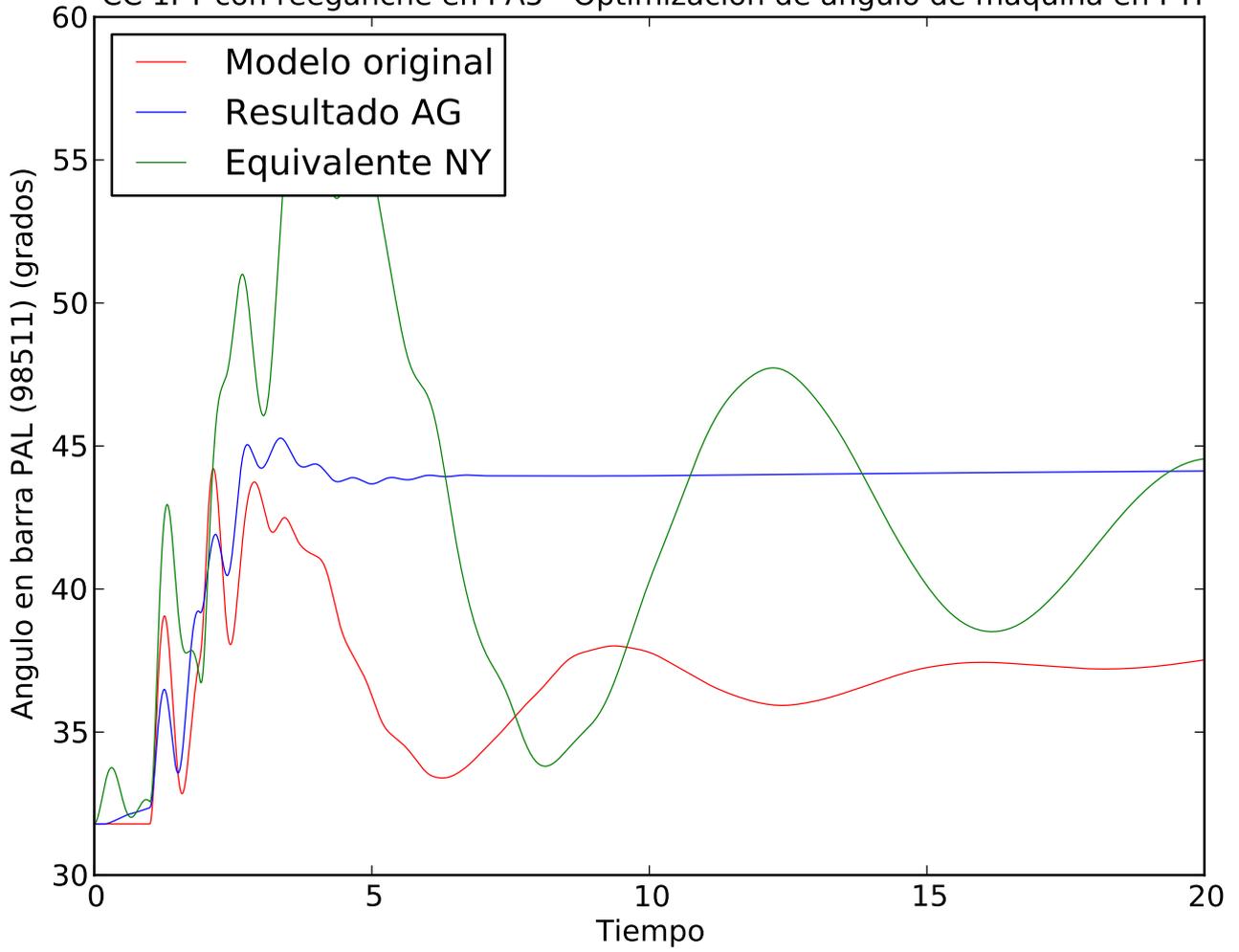
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



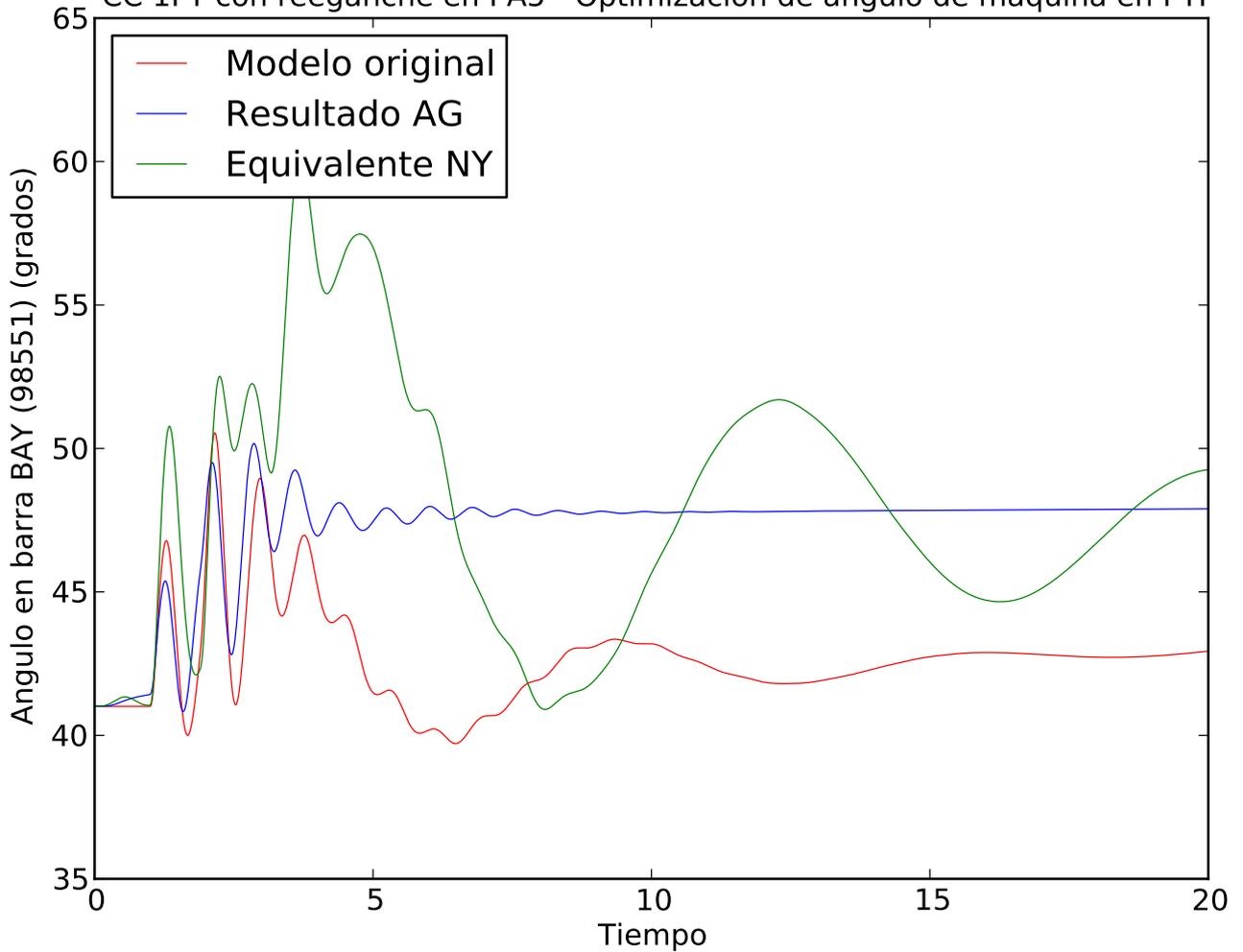
CC 1FT con reenganche en PA5 - Optimizacion de angulo de maquina en PTI



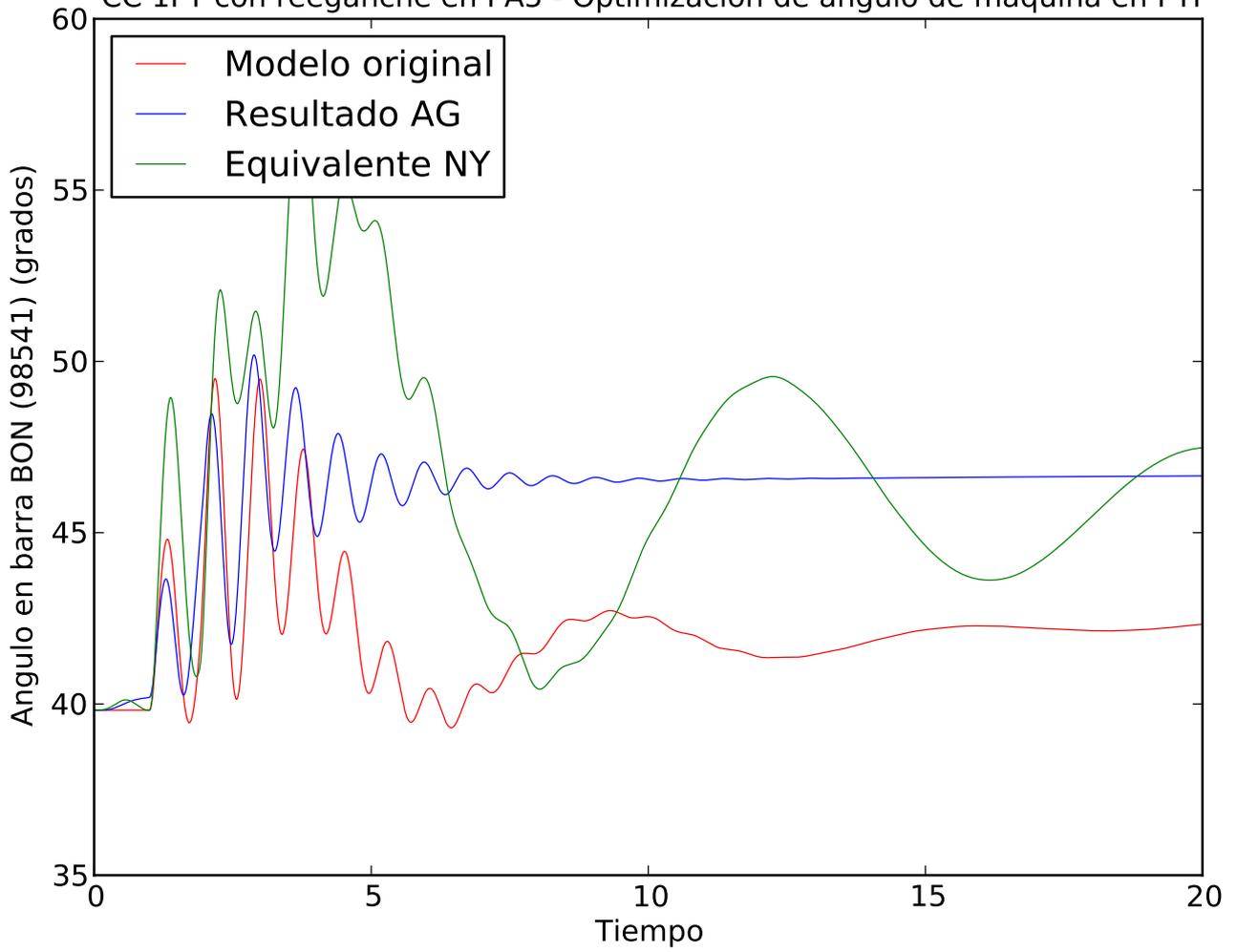
CC 1FT con reeganche en PA5 - Optimizacion de angulo de maquina en PTI



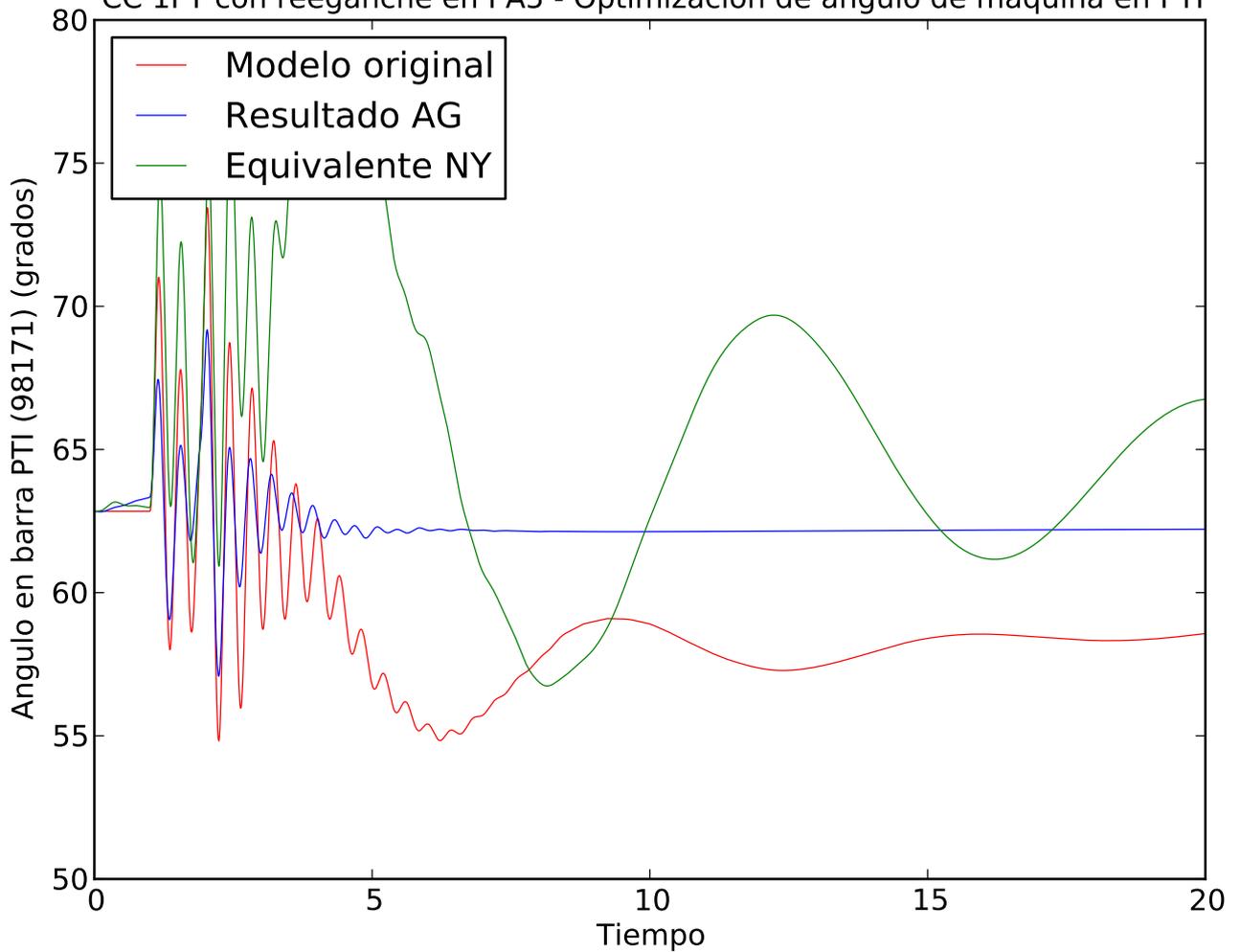
CC 1FT con reeganche en PA5 - Optimizacion de angulo de maquina en PTI



CC 1FT con reeganche en PA5 - Optimizacion de angulo de maquina en PTI

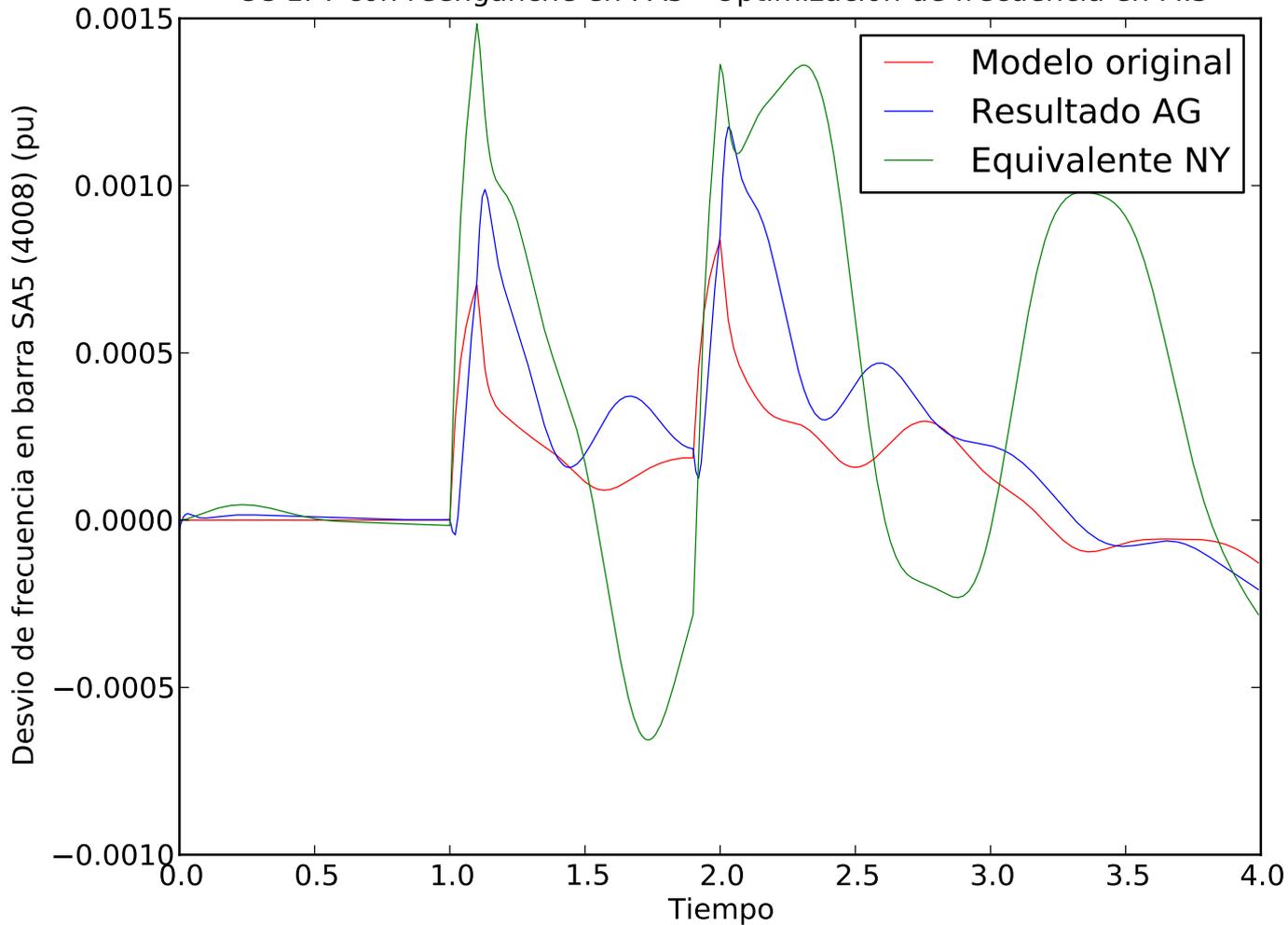


CC 1FT con reeganche en PA5 - Optimizacion de angulo de maquina en PTI

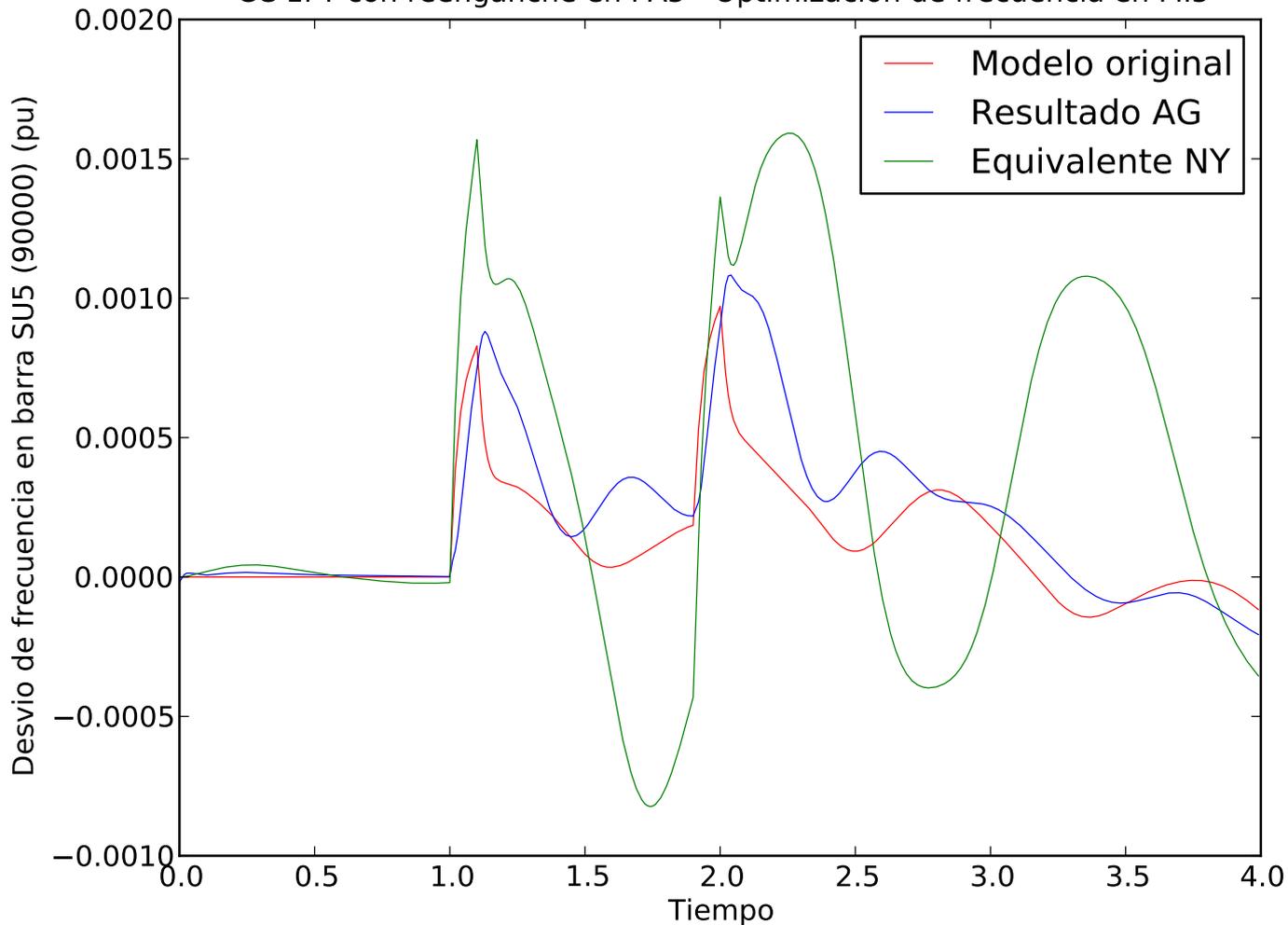


- C.15. Simulación 10: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando frecuencia en MI5 en 4 segundos. Comportamiento en 4 segundos

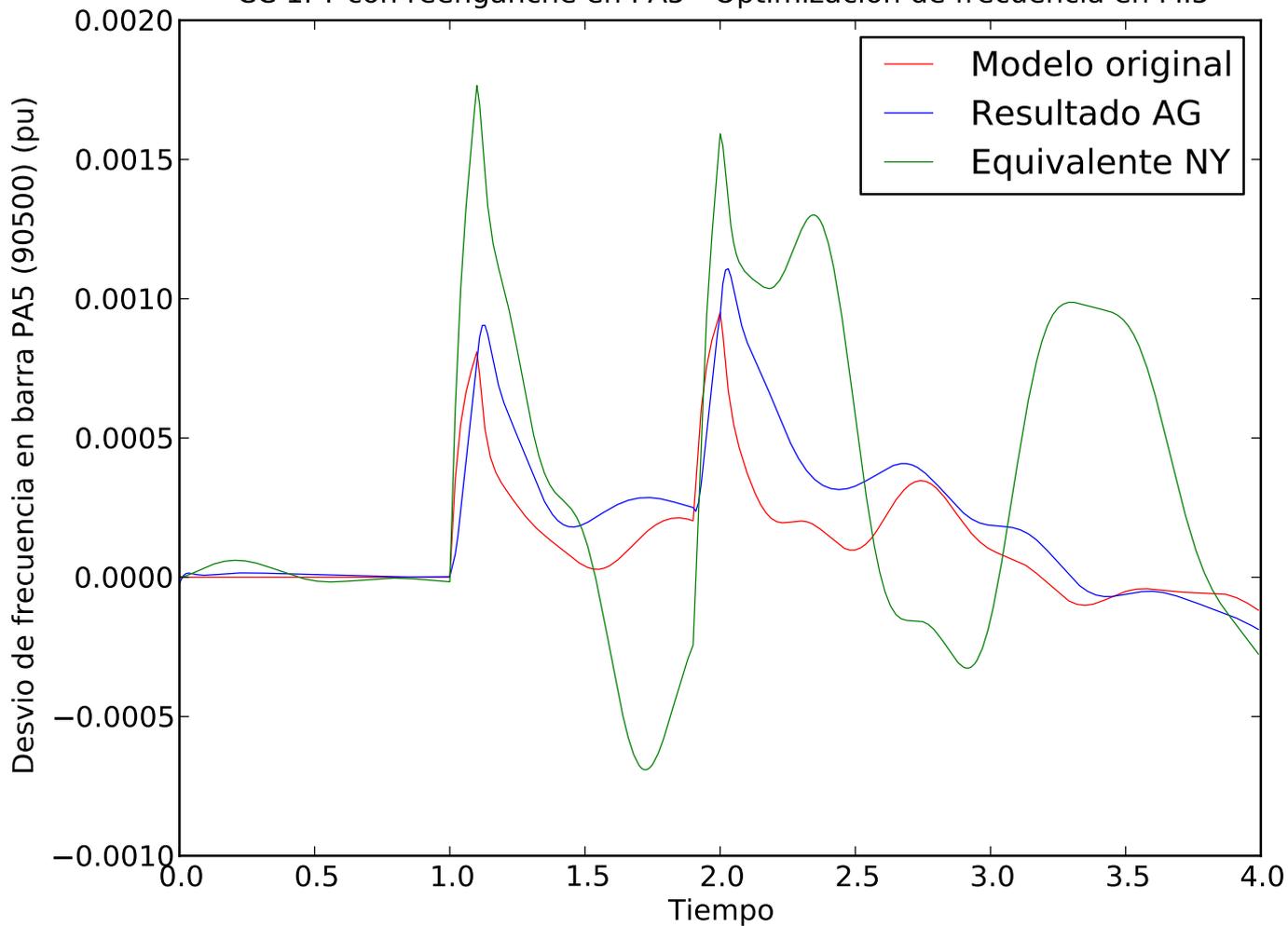
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



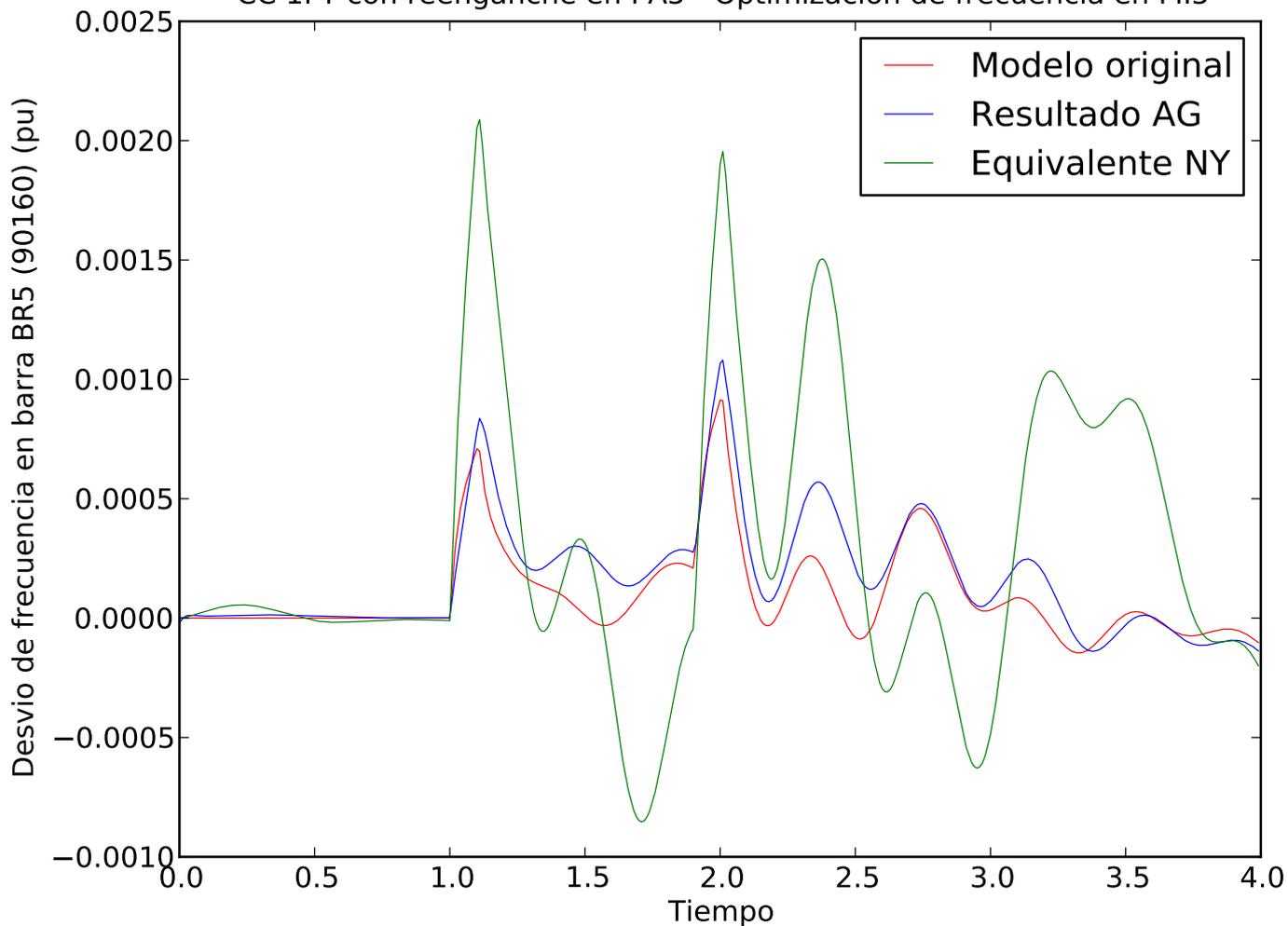
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



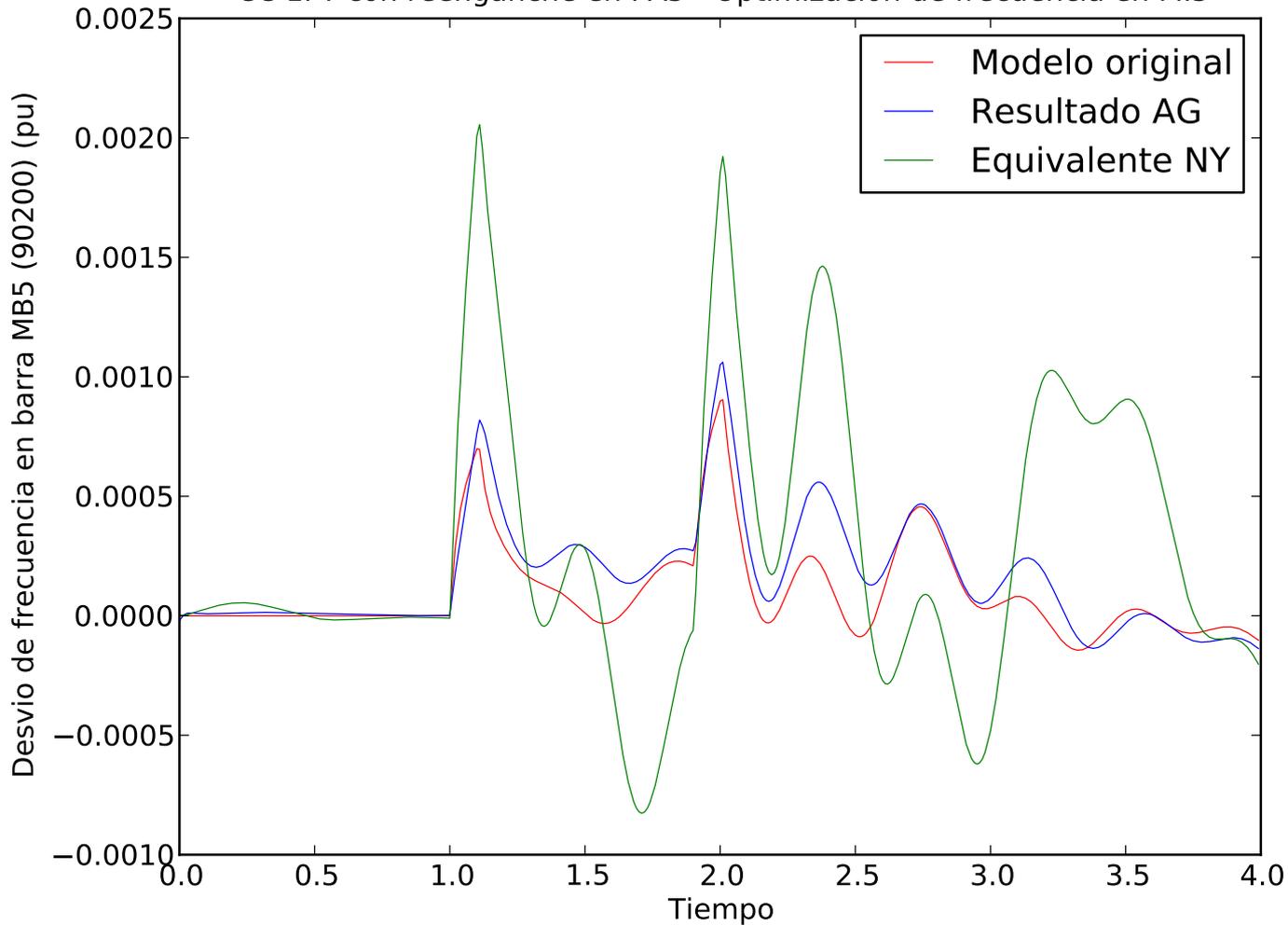
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



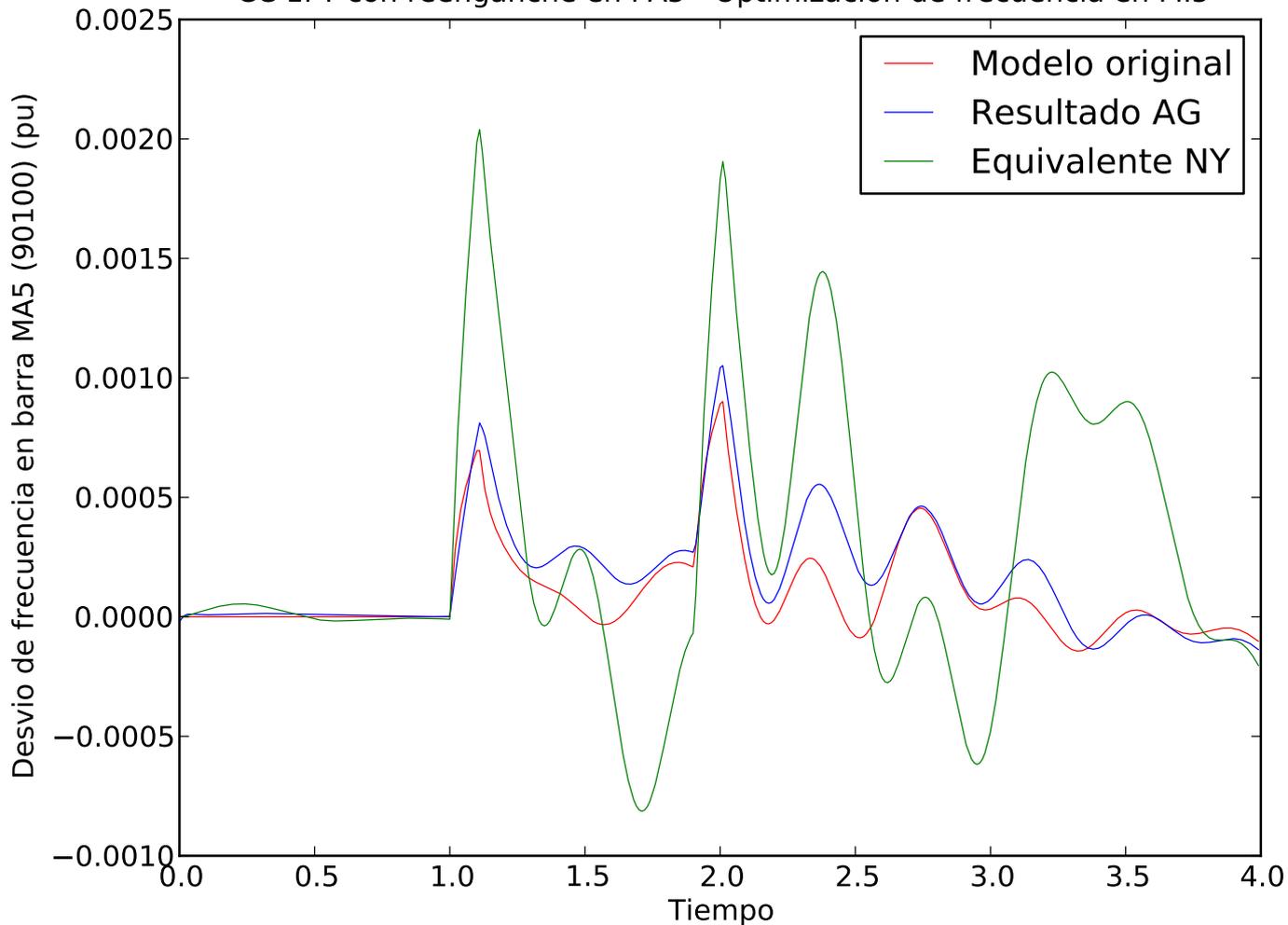
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



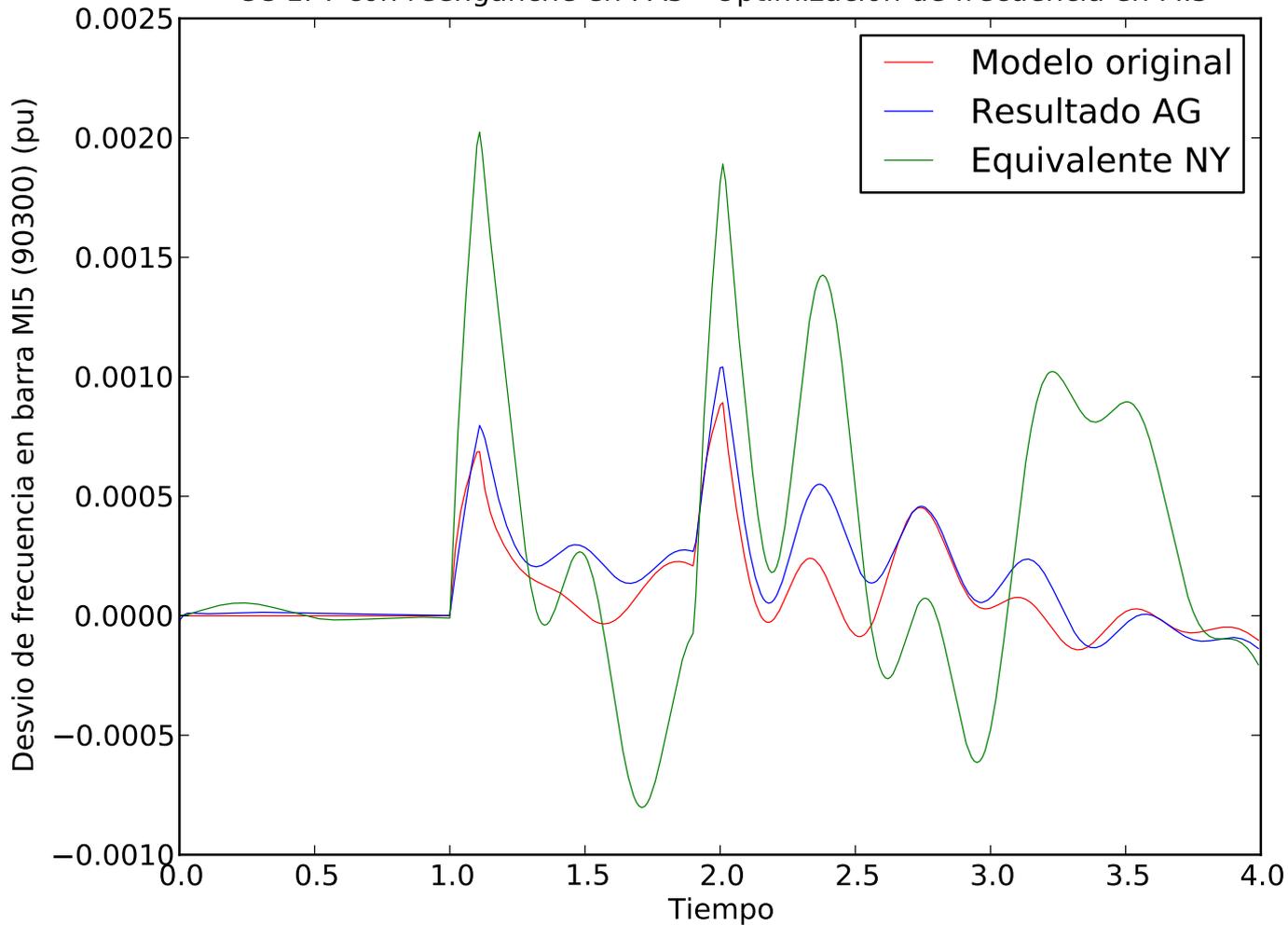
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



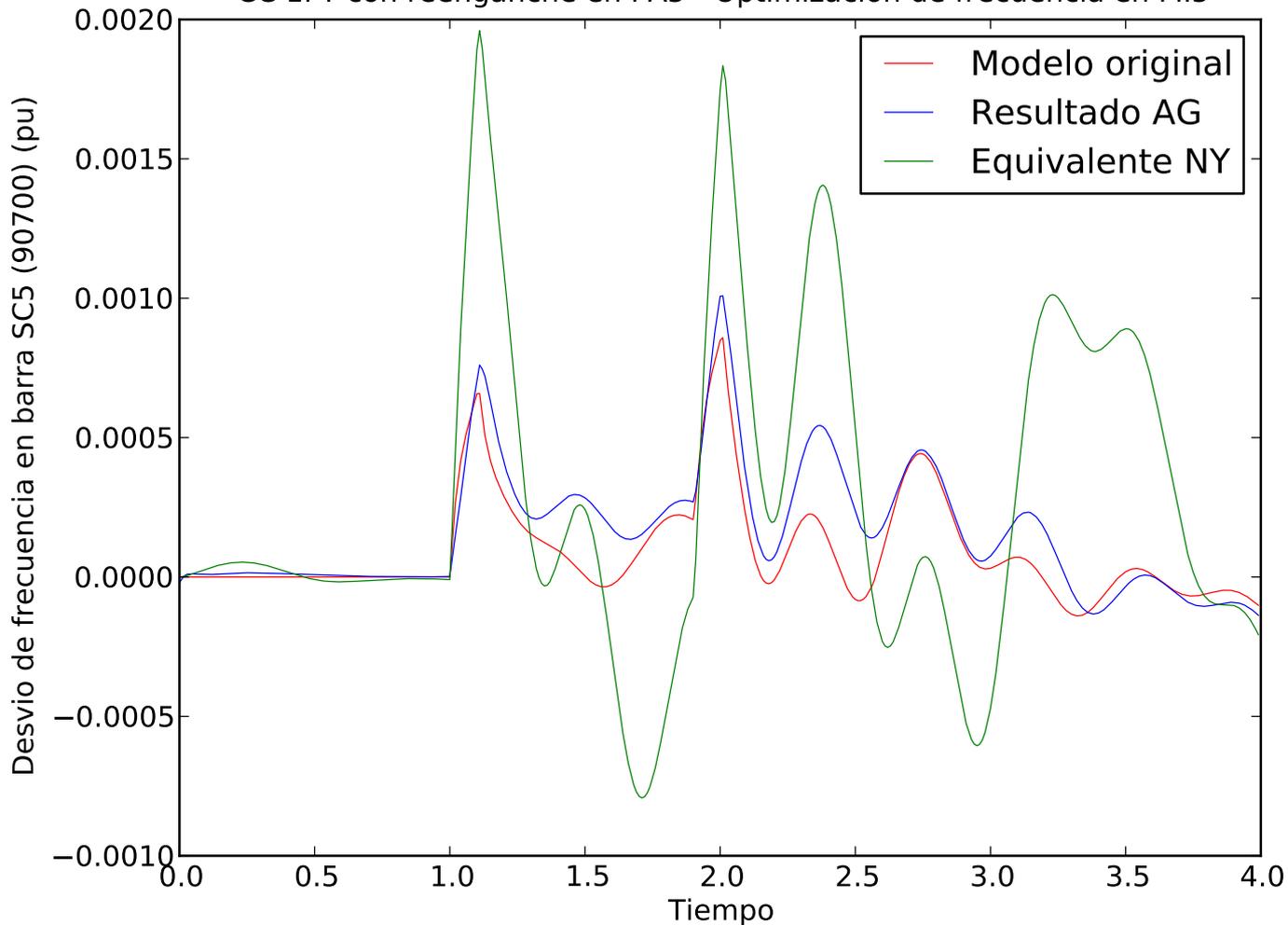
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



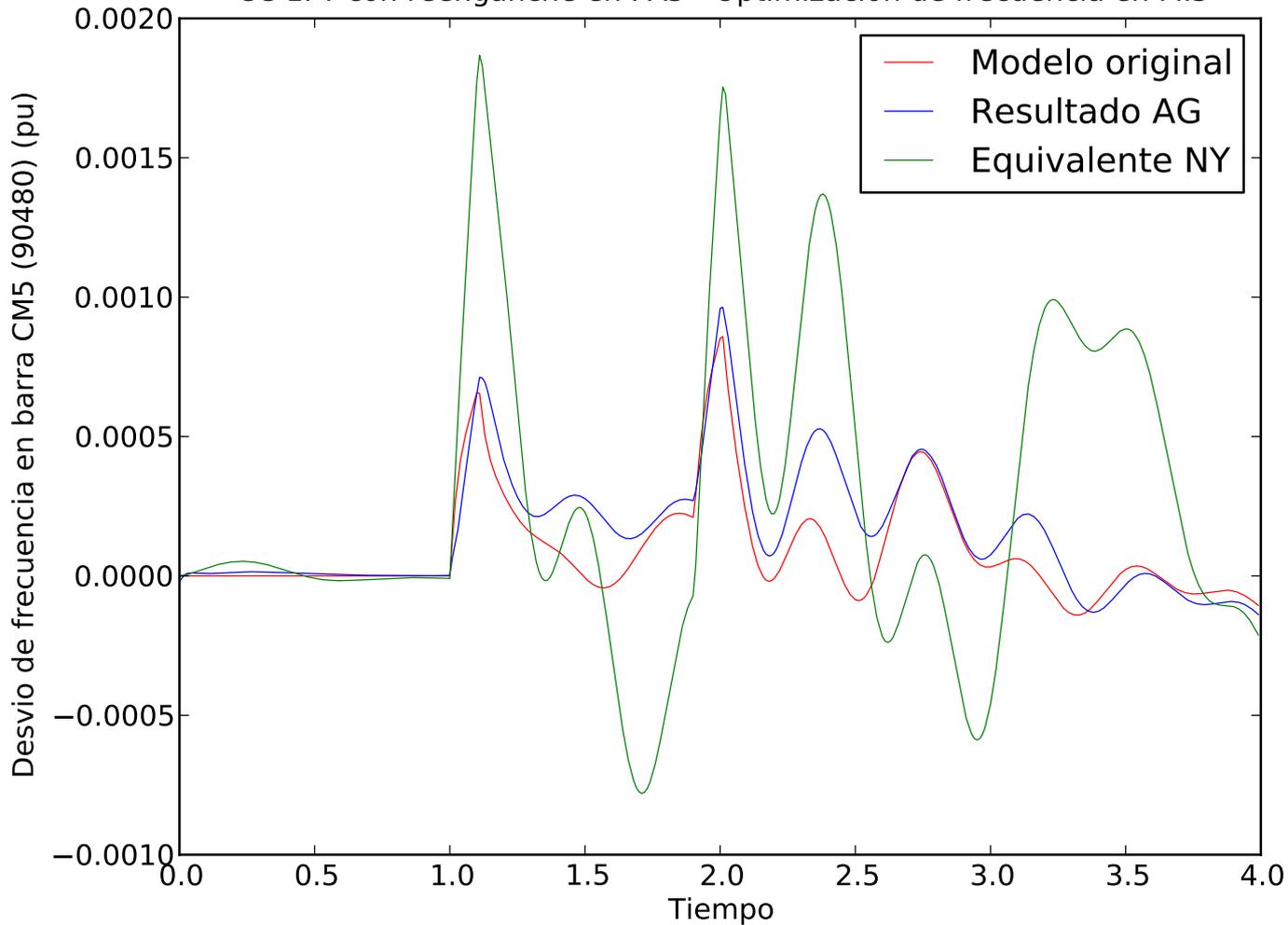
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



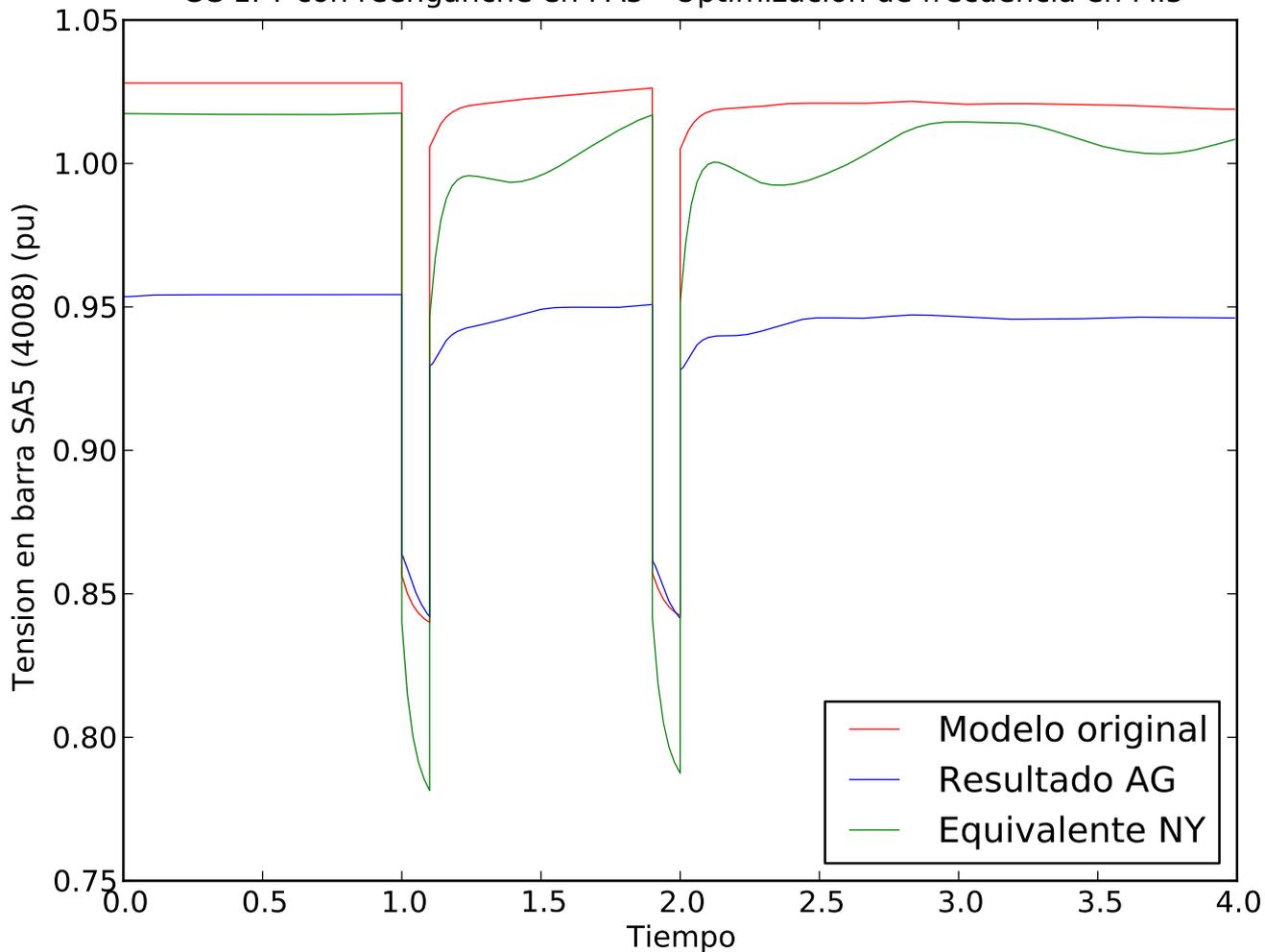
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



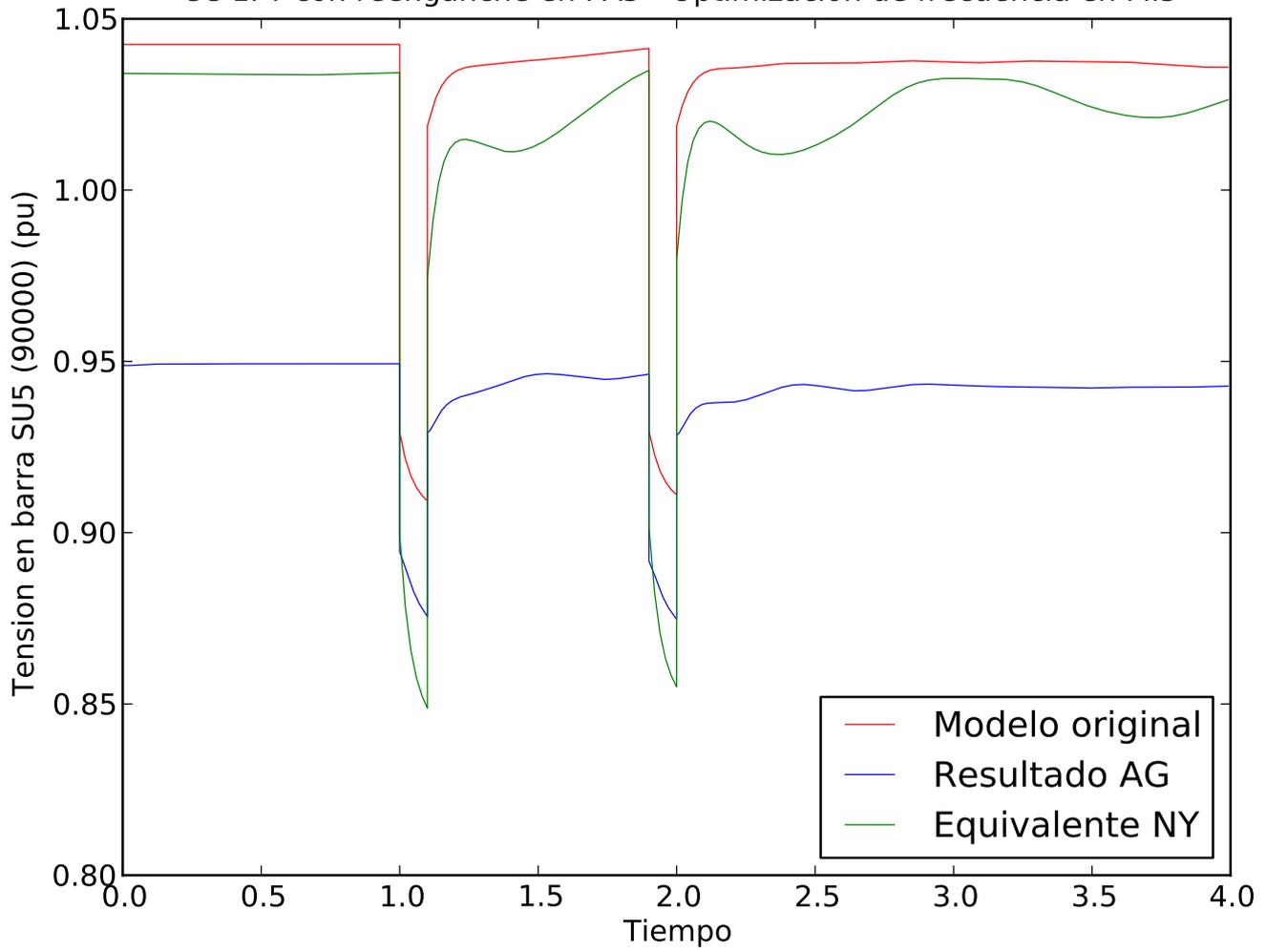
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



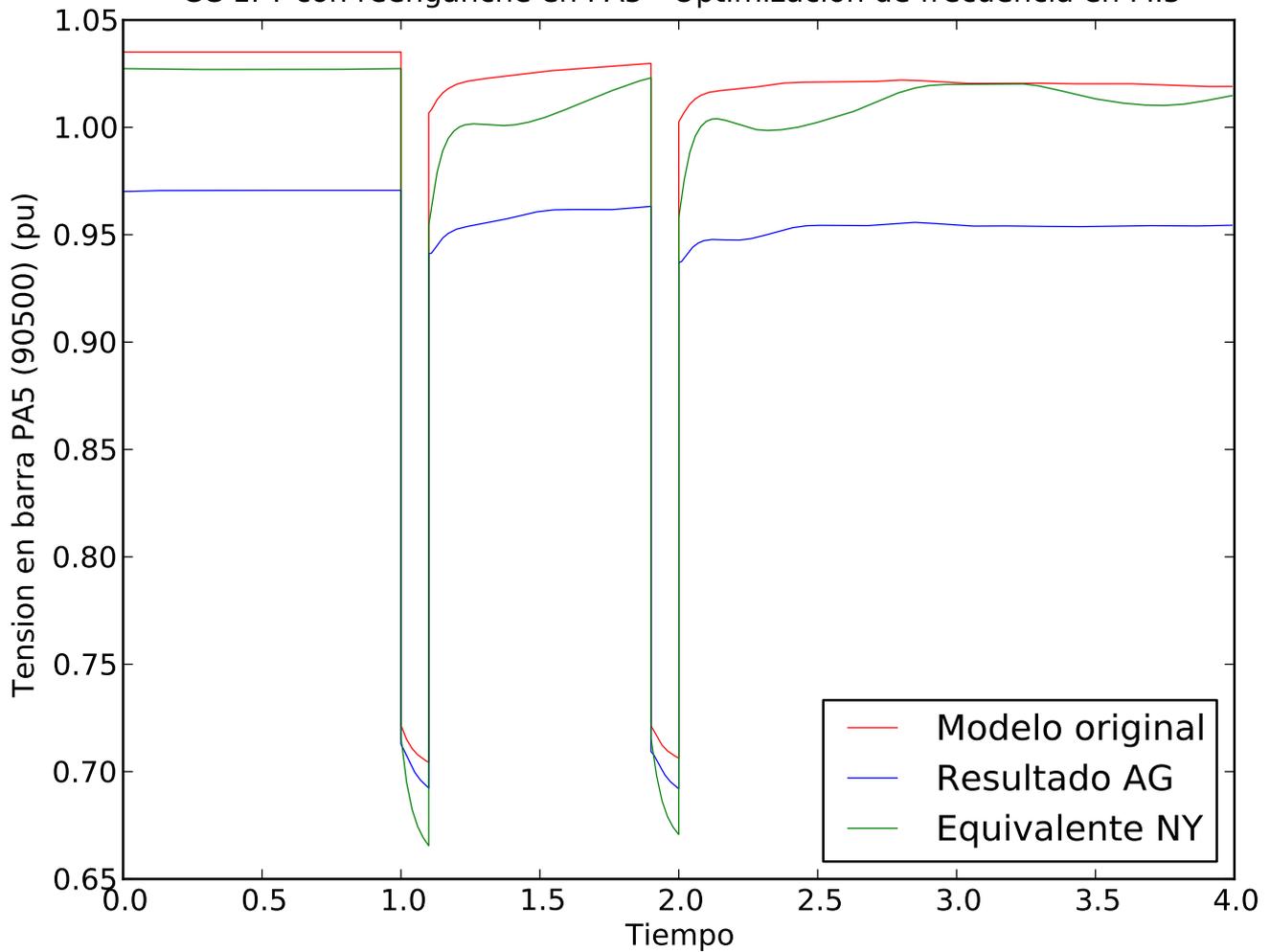
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



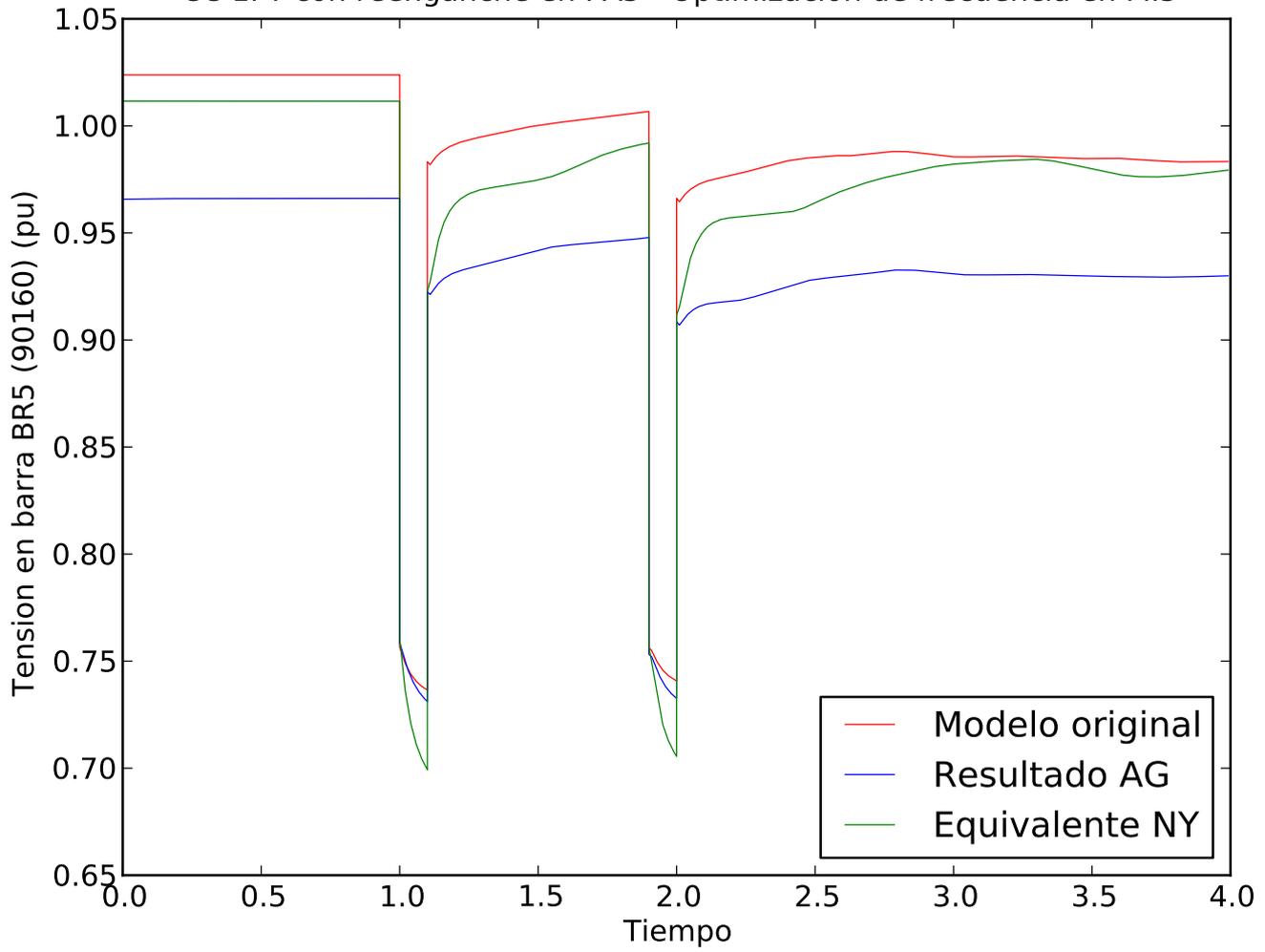
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



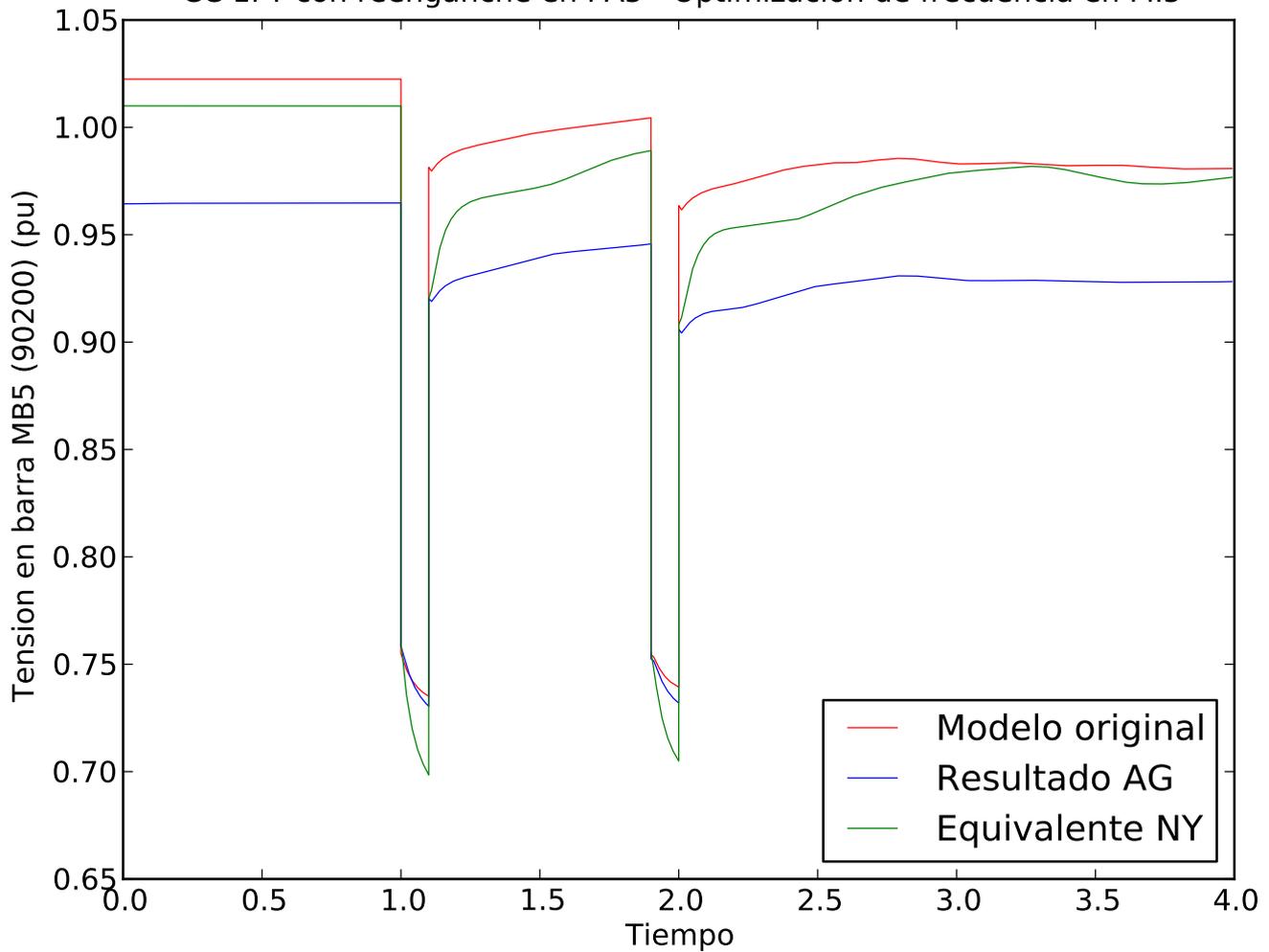
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



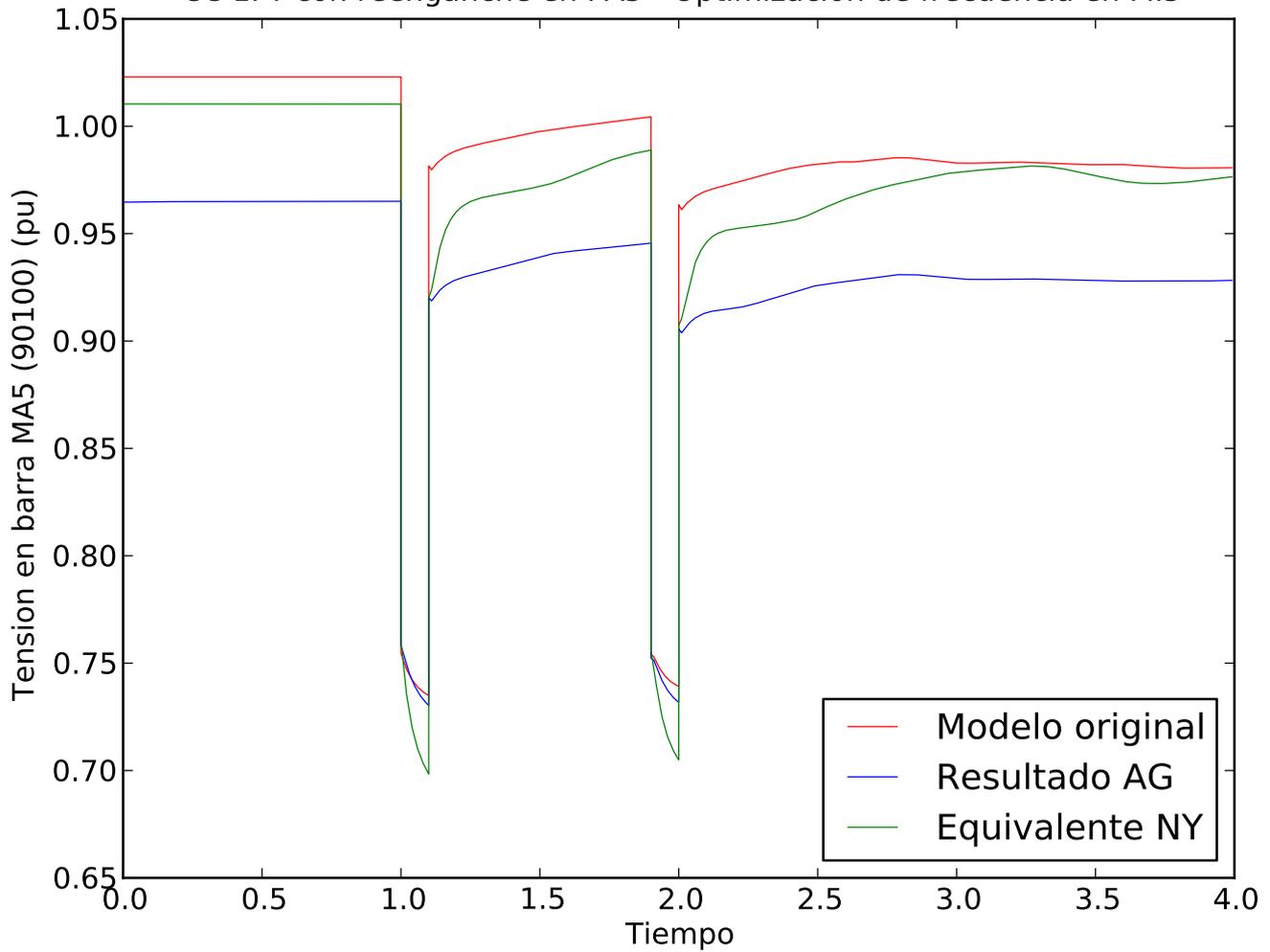
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



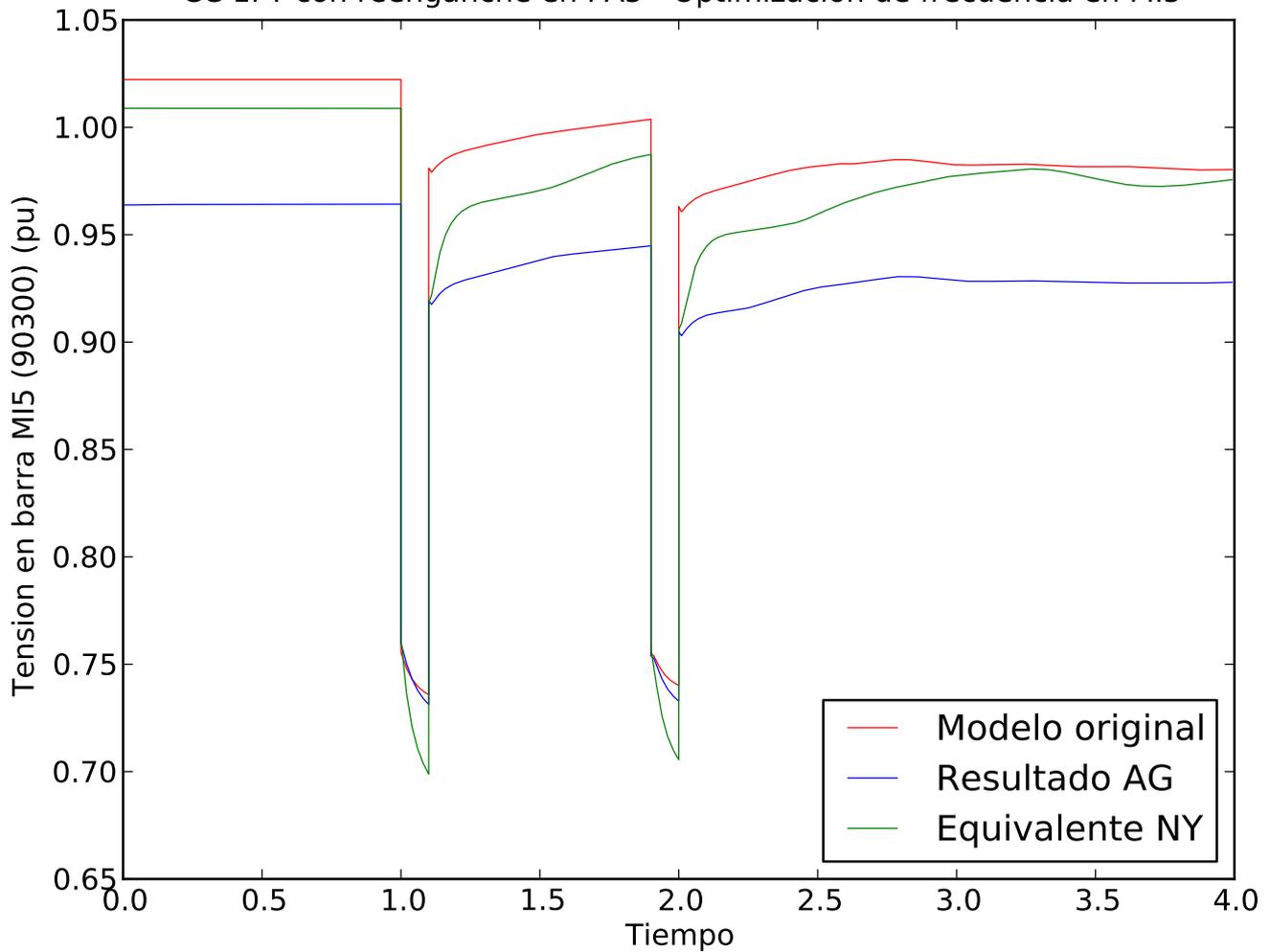
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



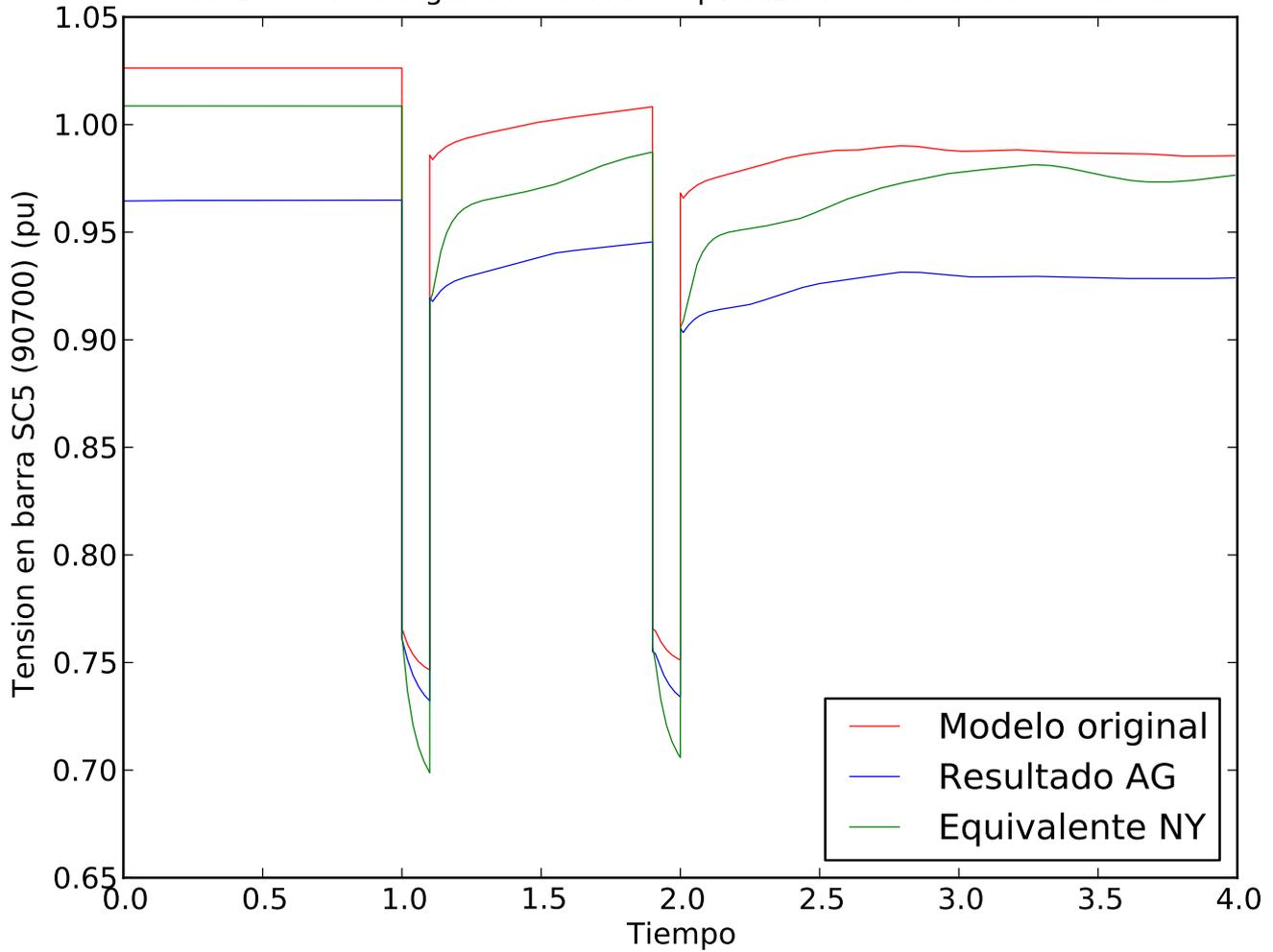
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



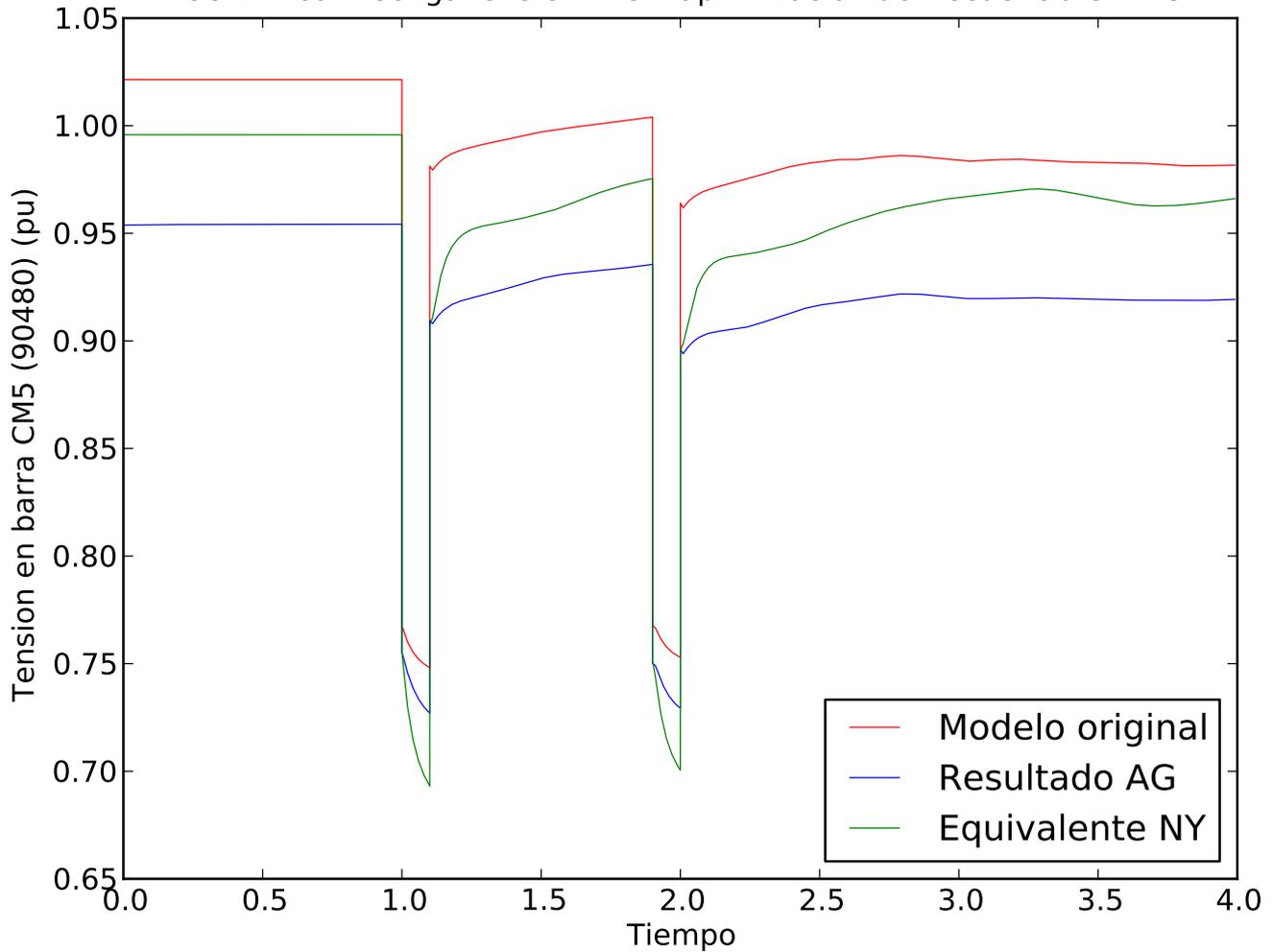
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



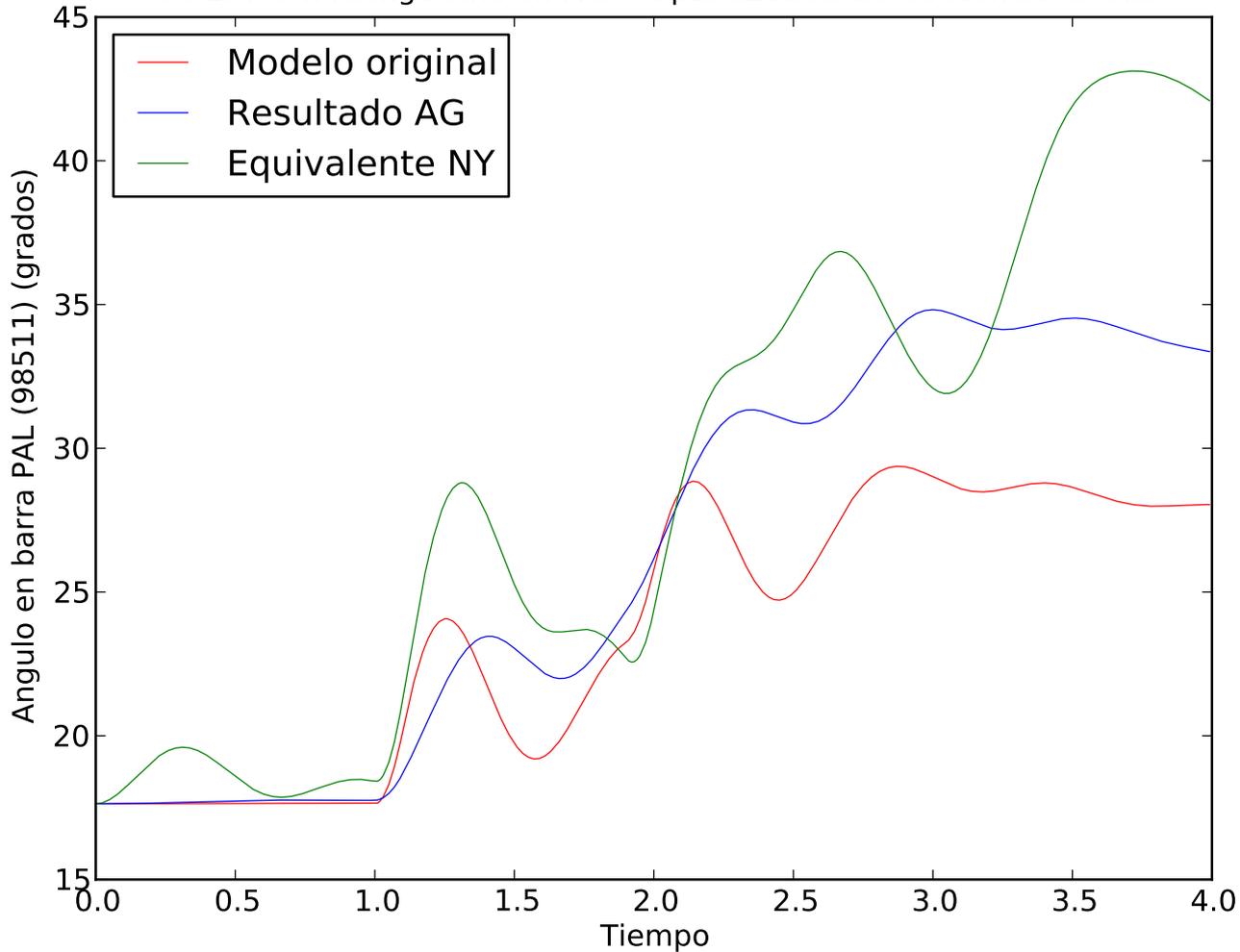
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



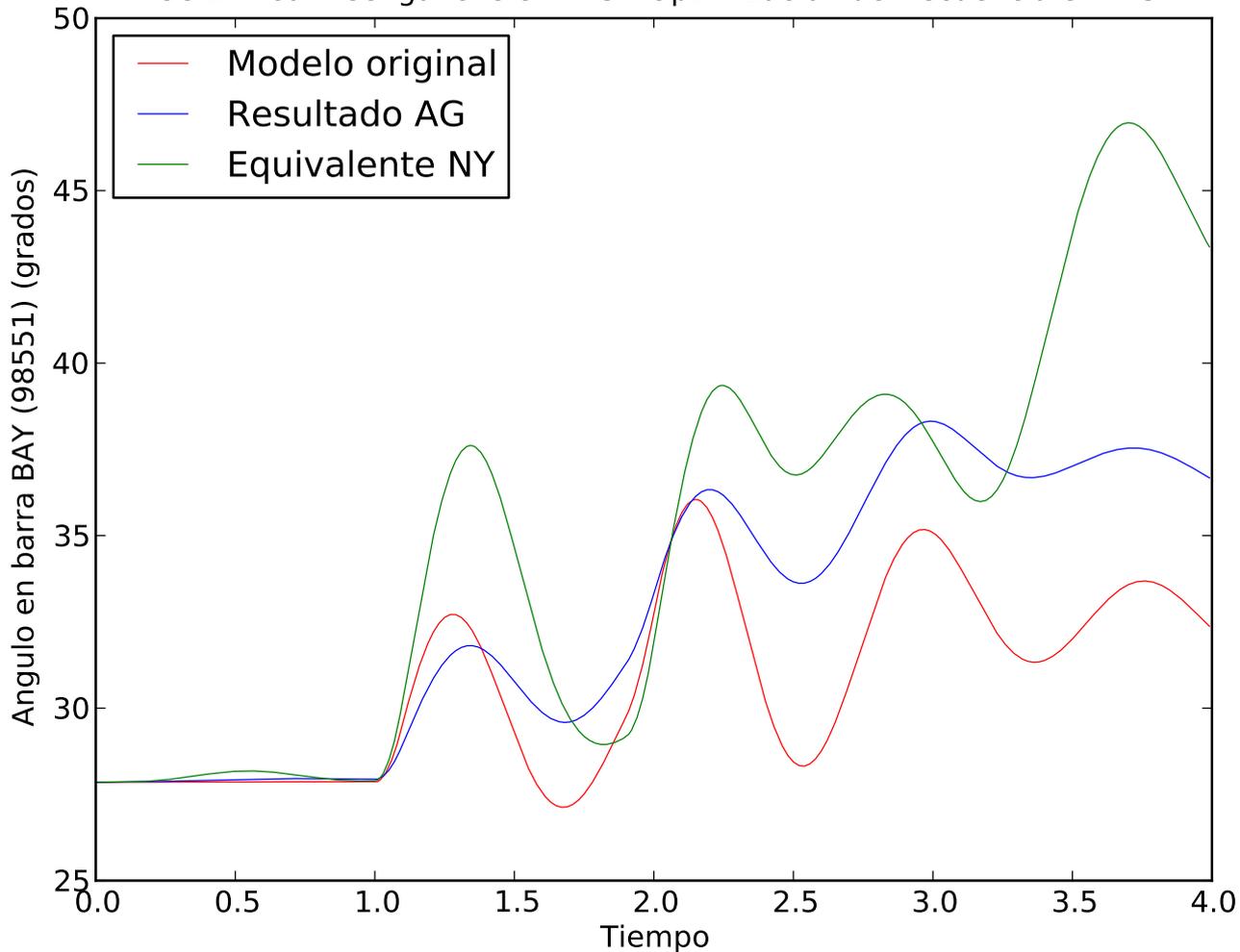
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



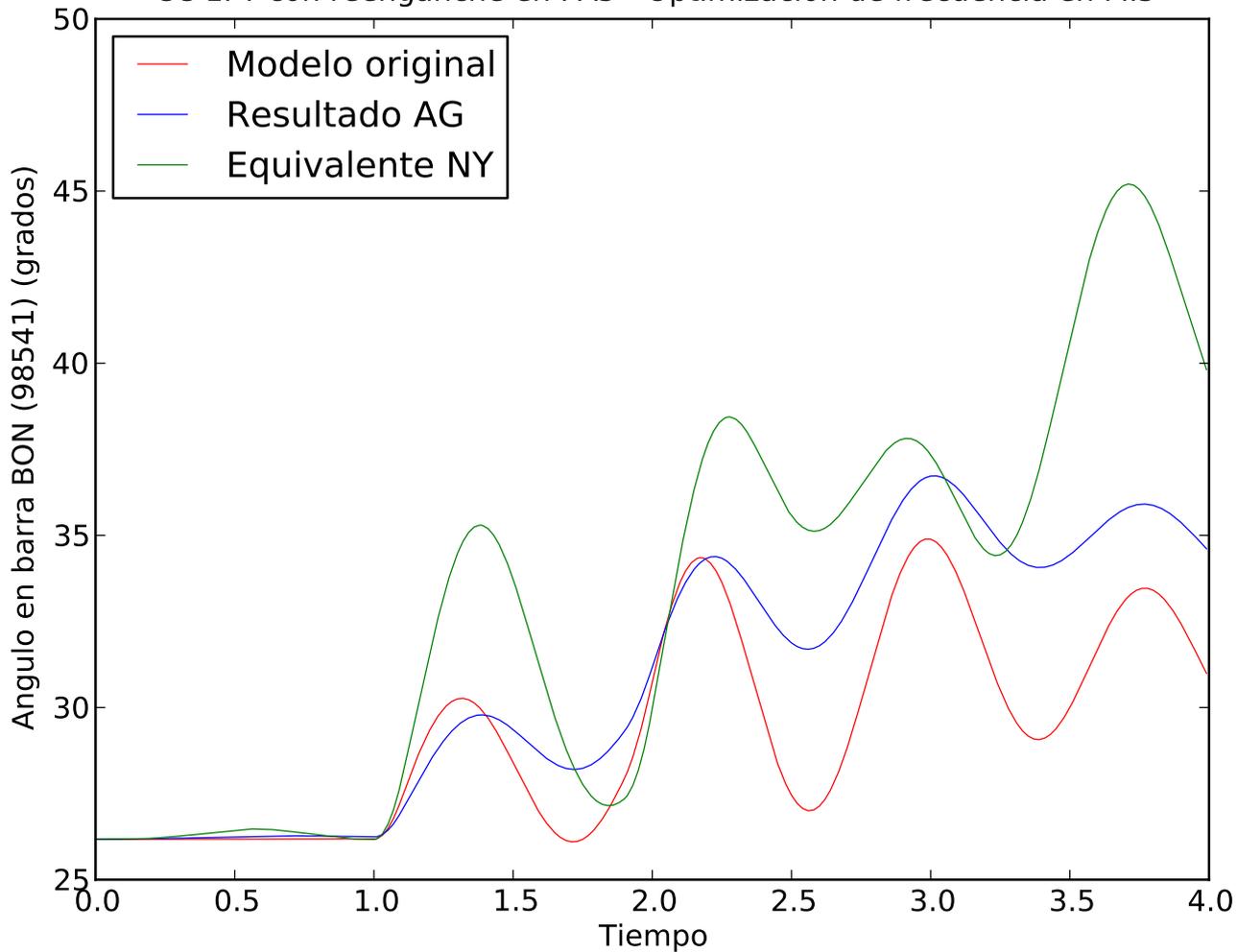
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



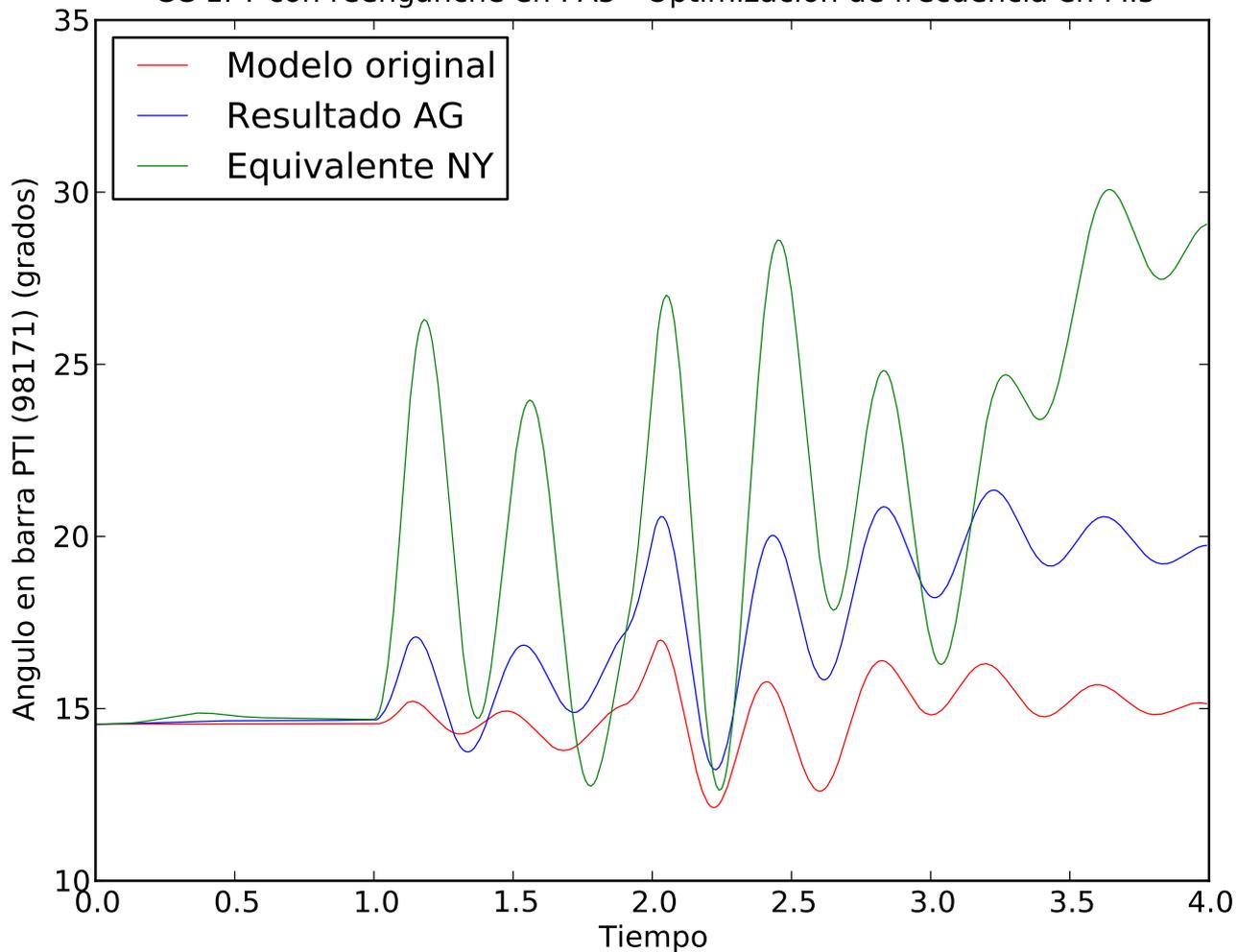
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5

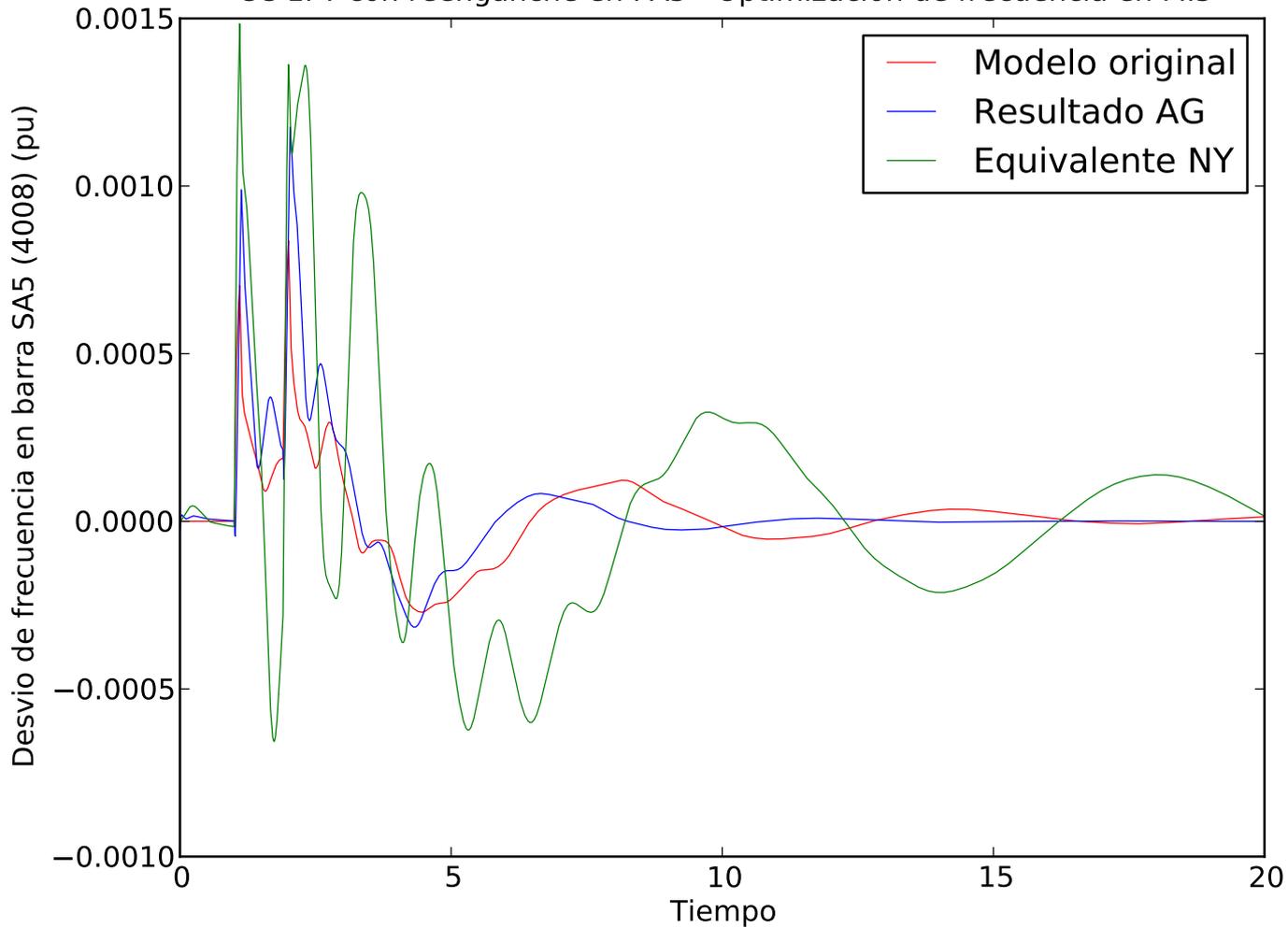


CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5

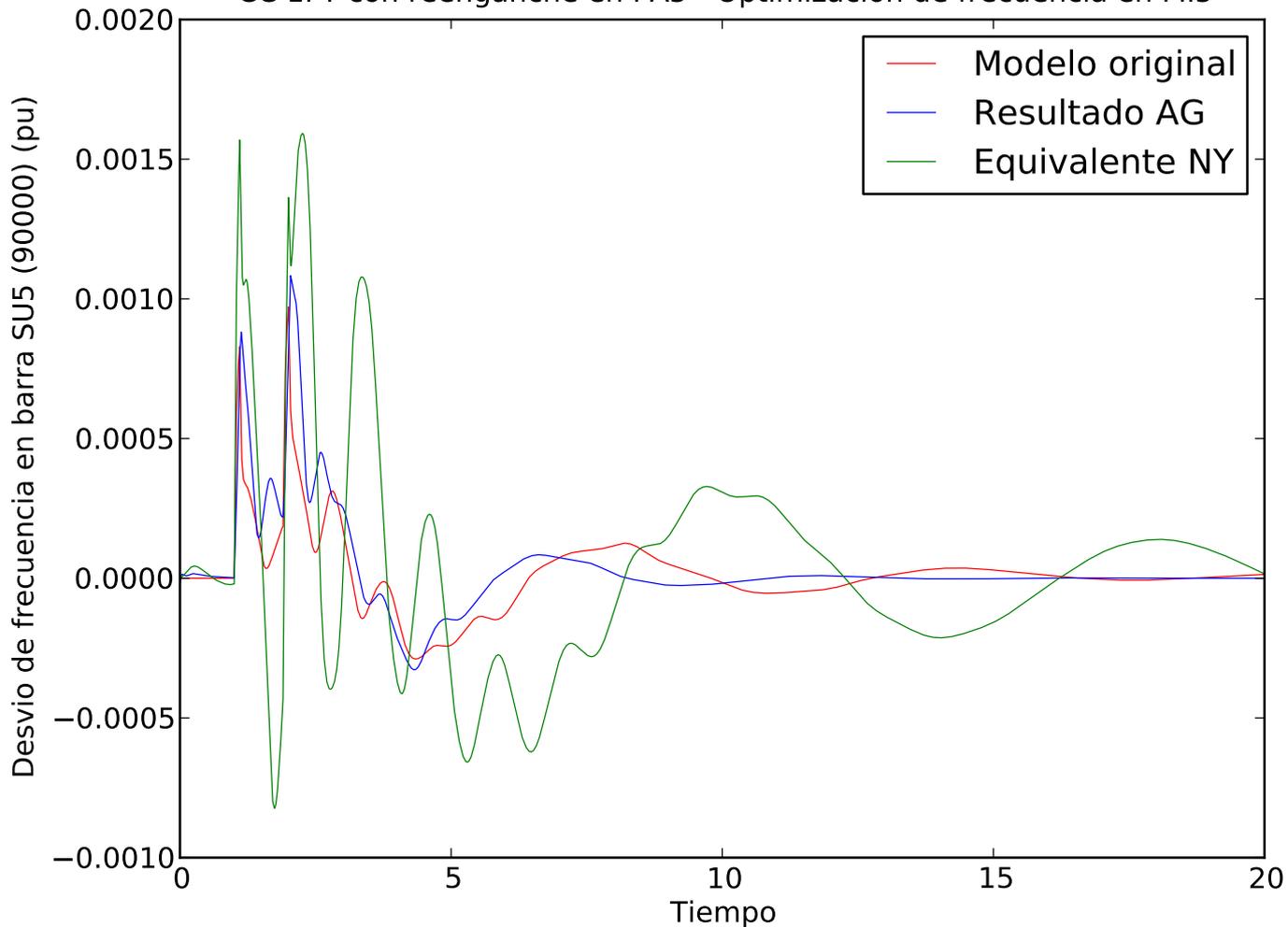


- C.16. Simulación 10: Cortocircuito 3F en PA5 con salida de línea en equivalente NY 3, optimizando frecuencia en MI5 en 4 segundos. Comportamiento en 20 segundos**

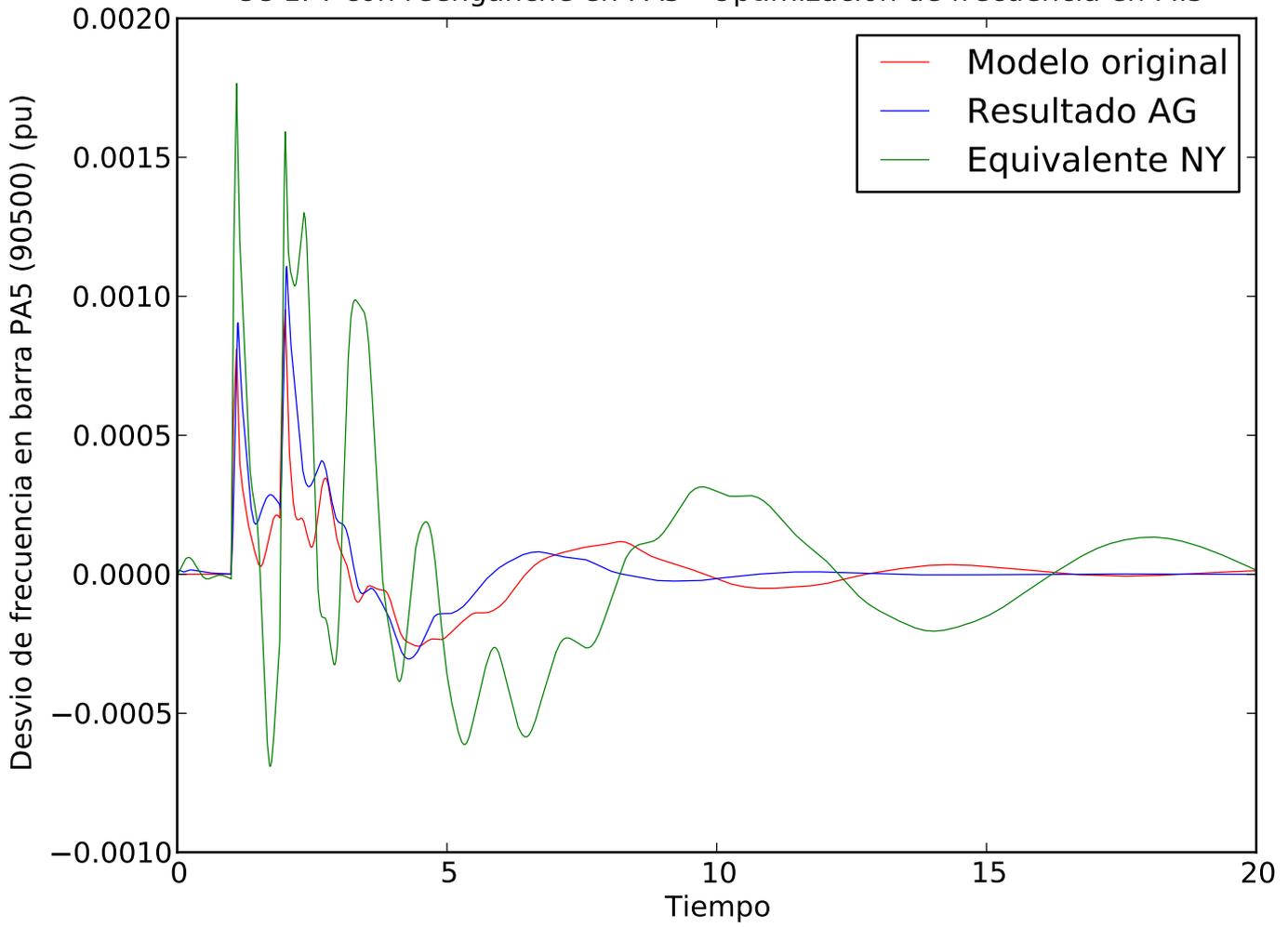
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



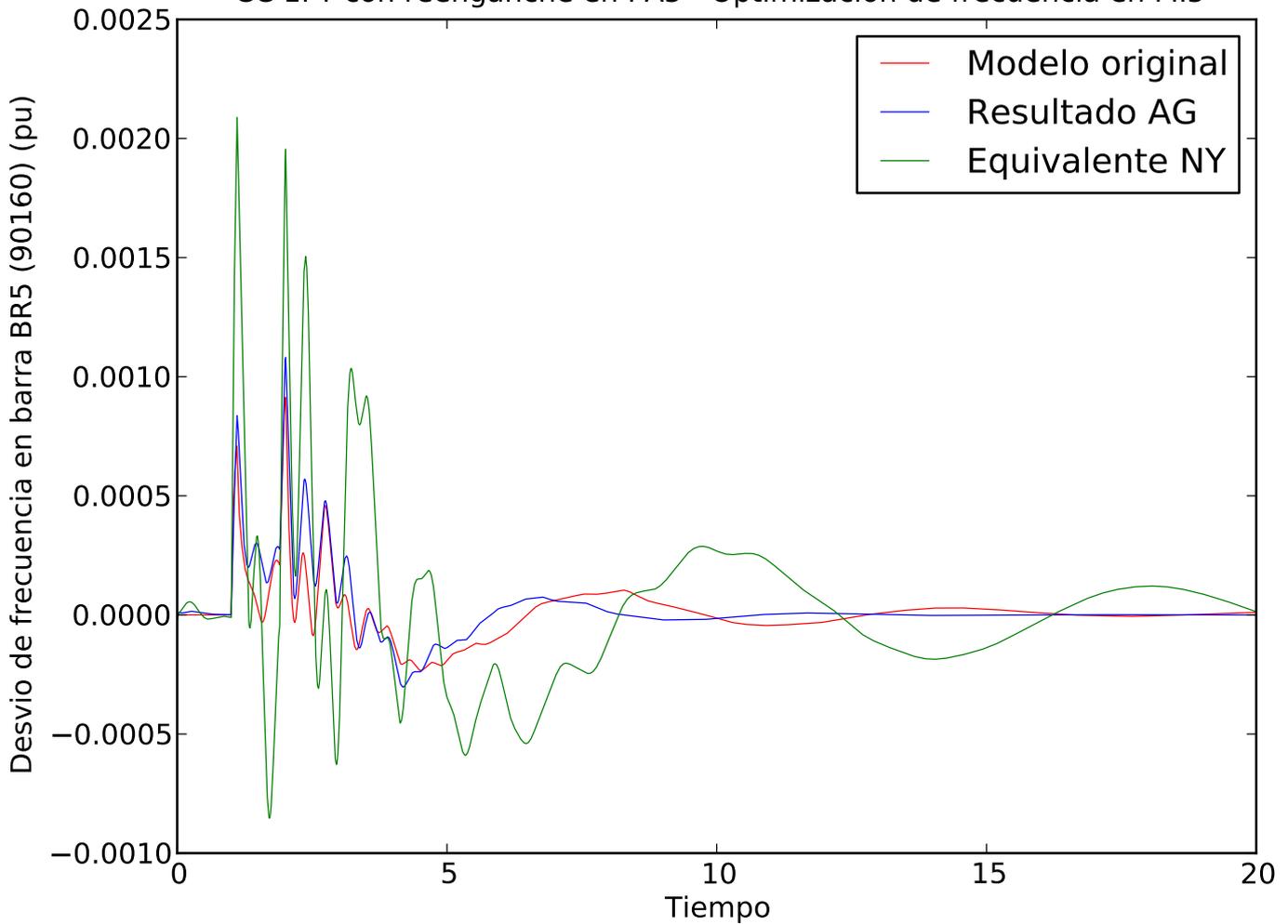
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



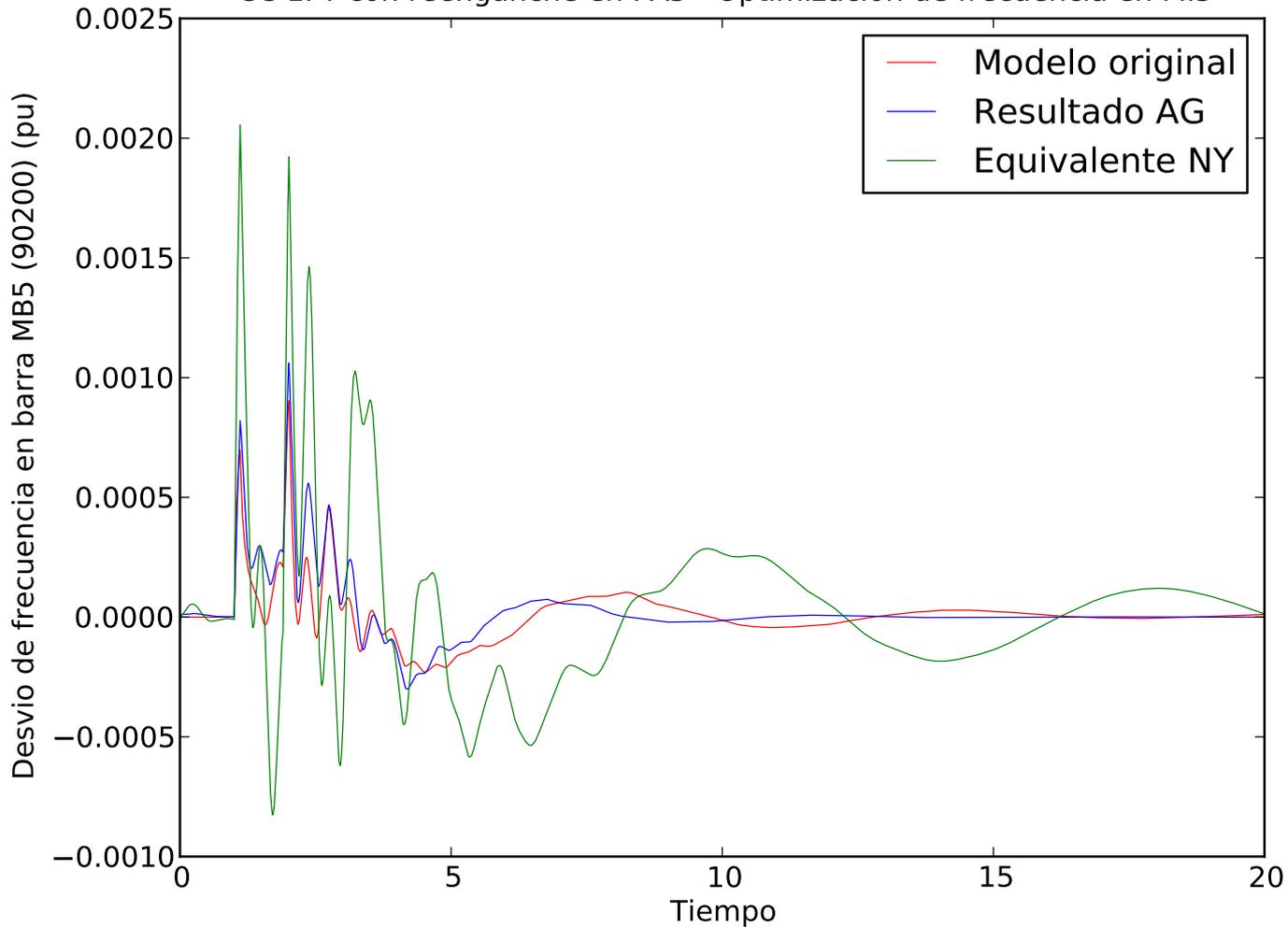
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



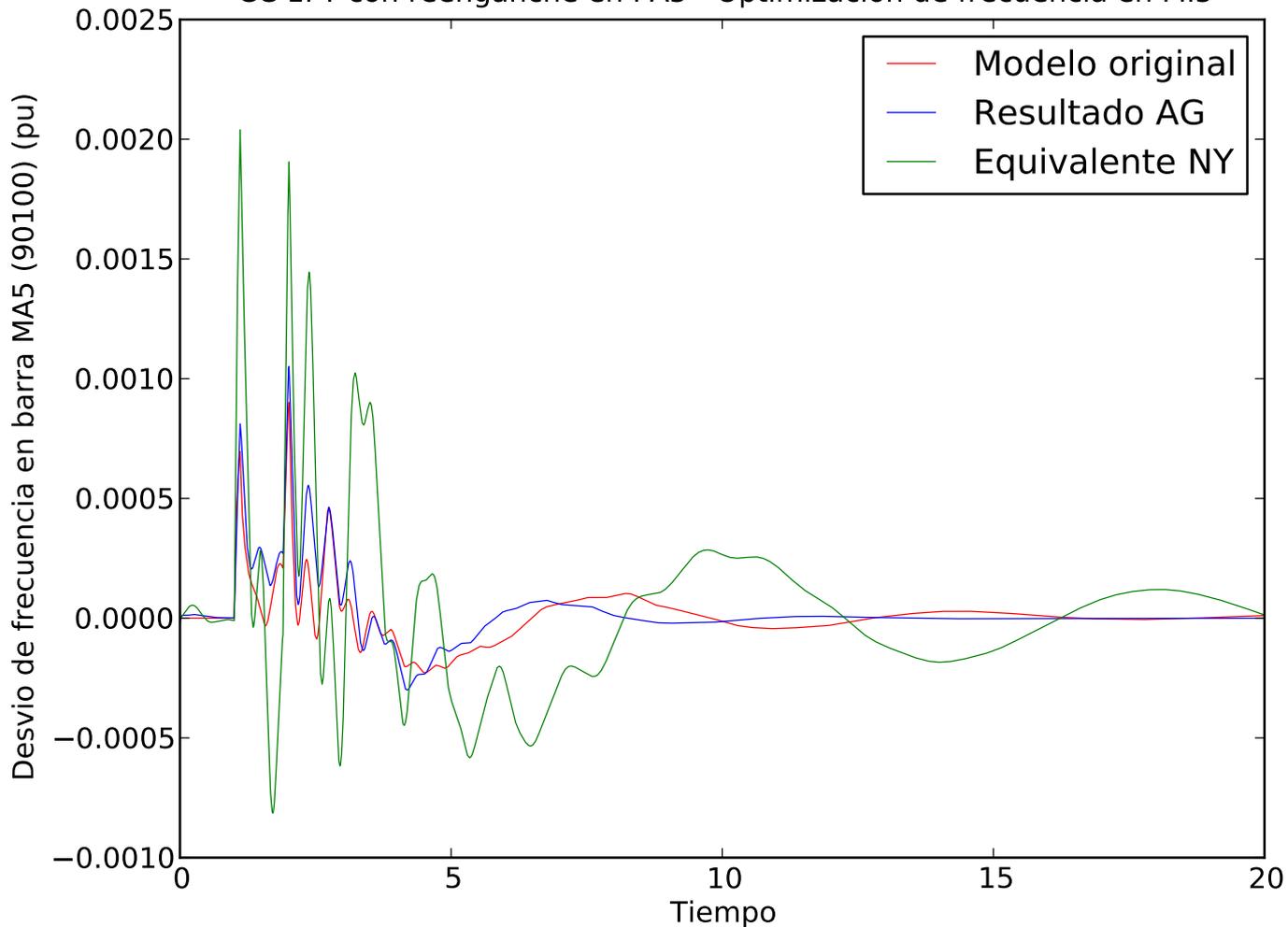
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



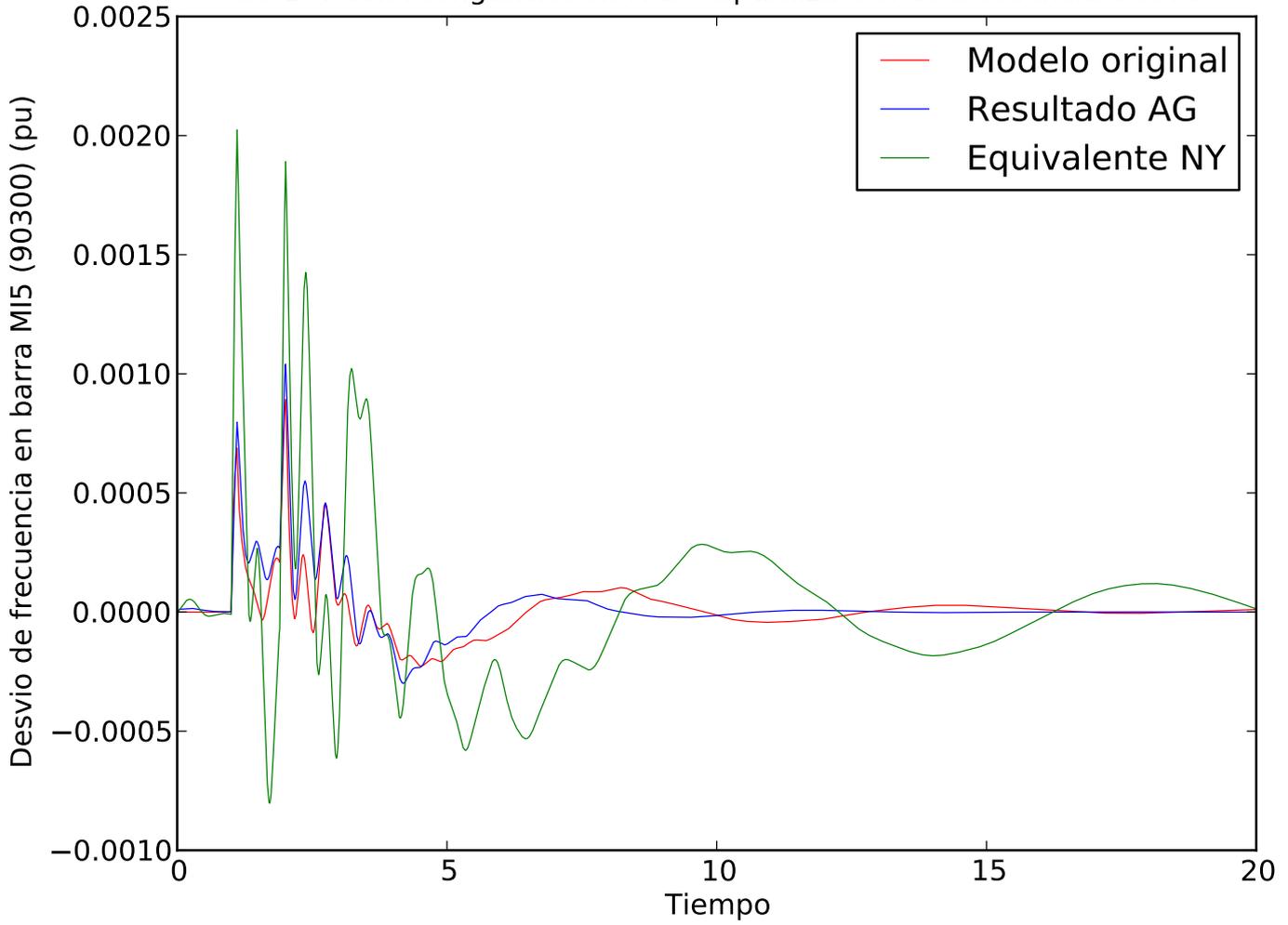
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



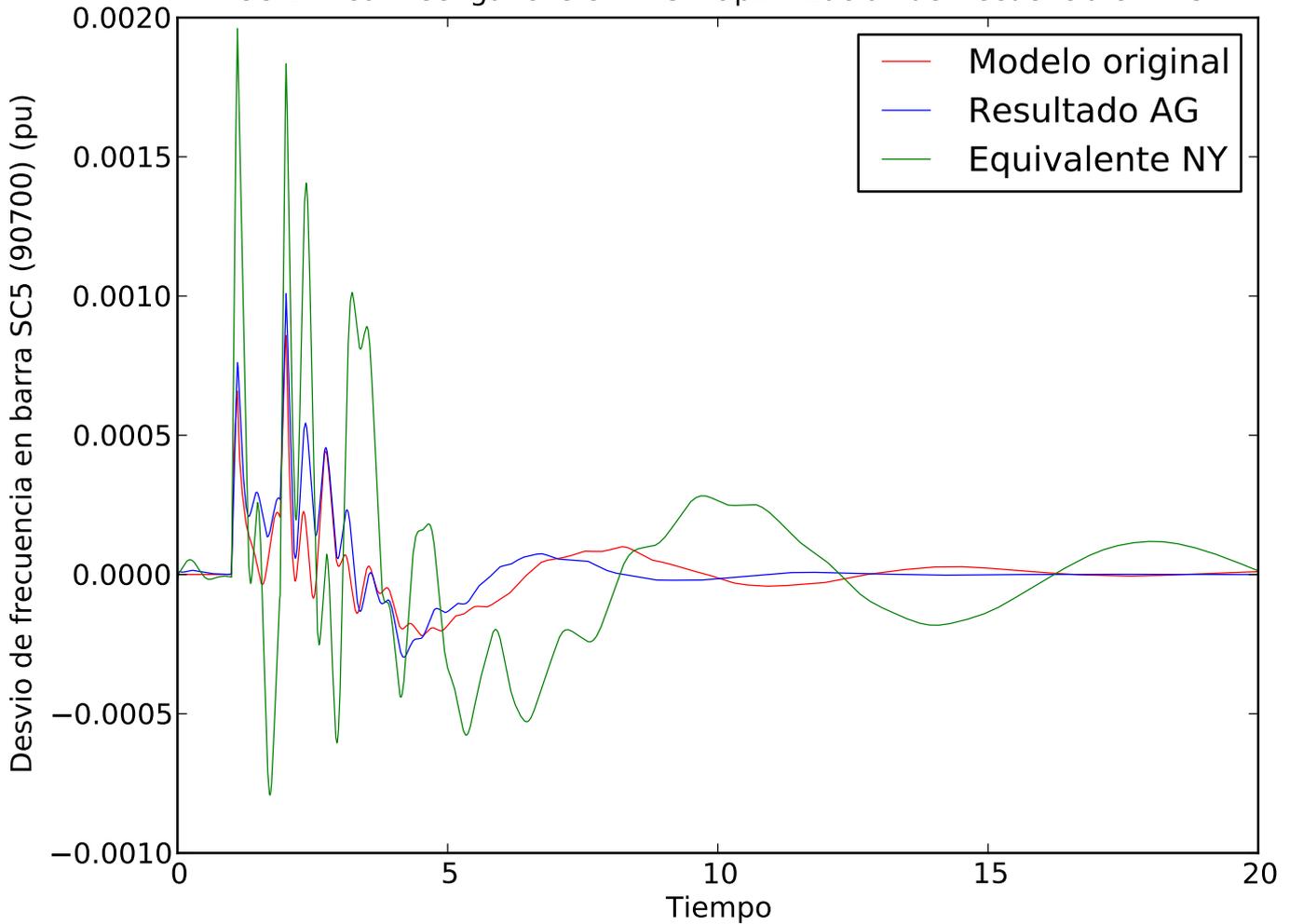
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



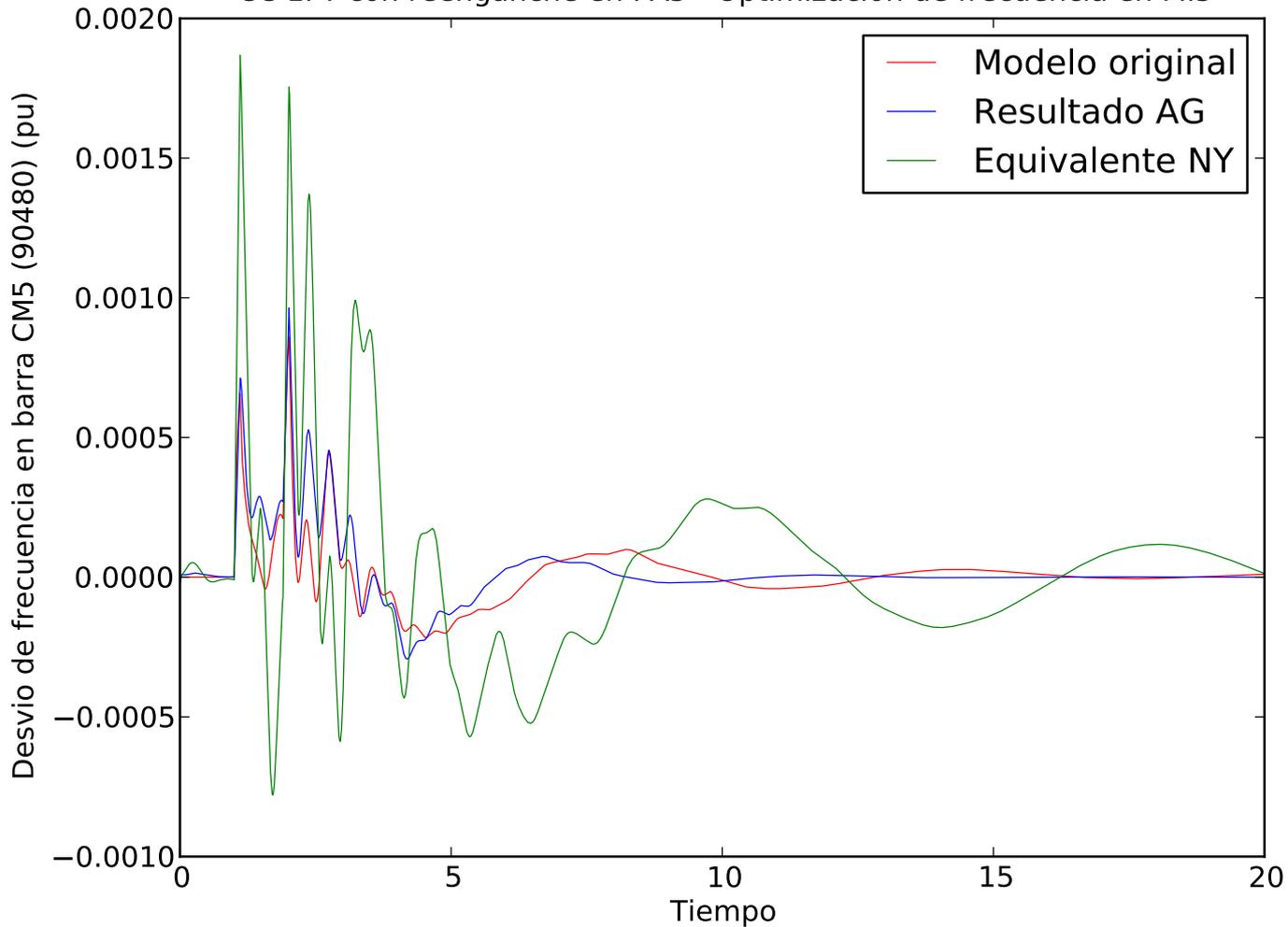
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



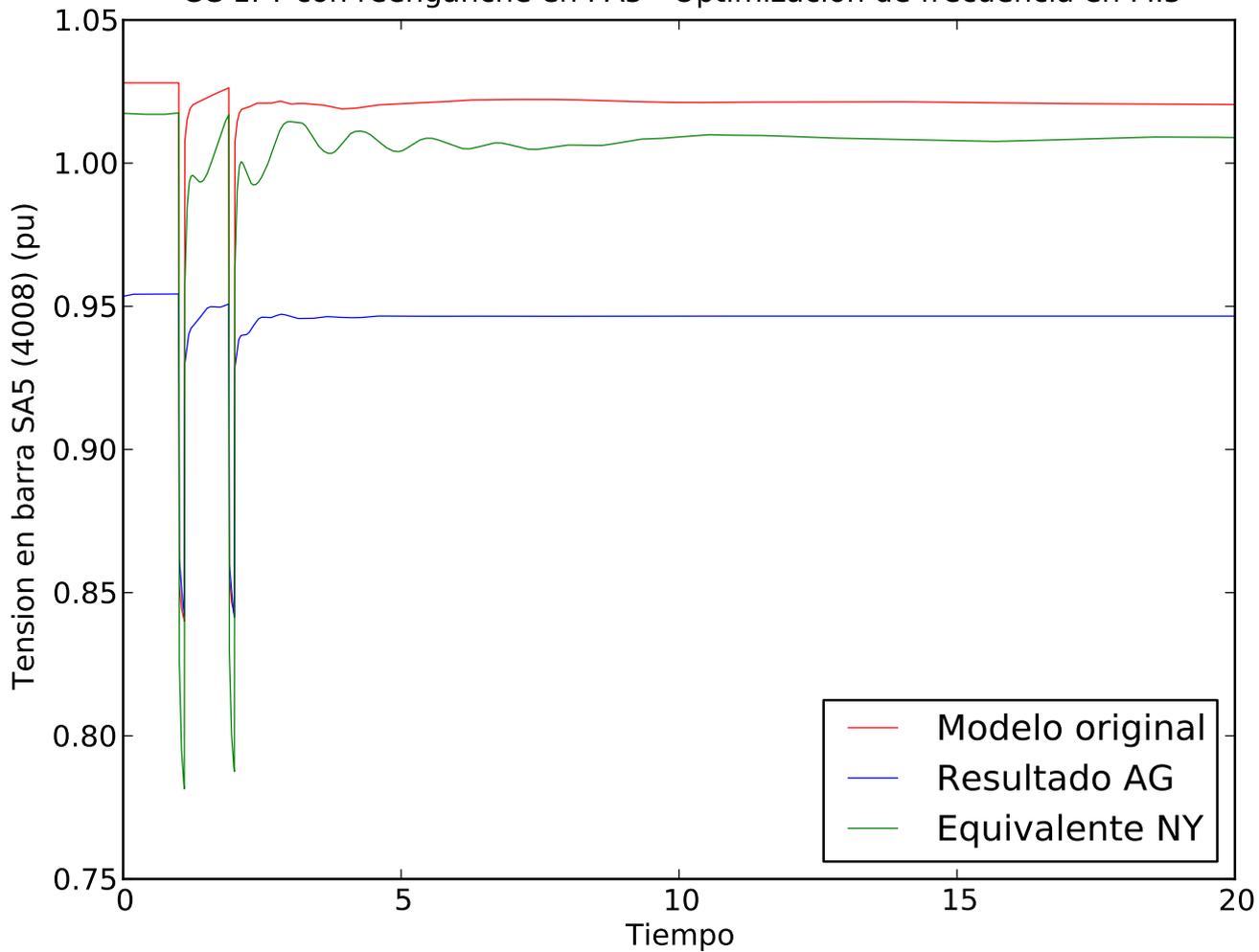
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



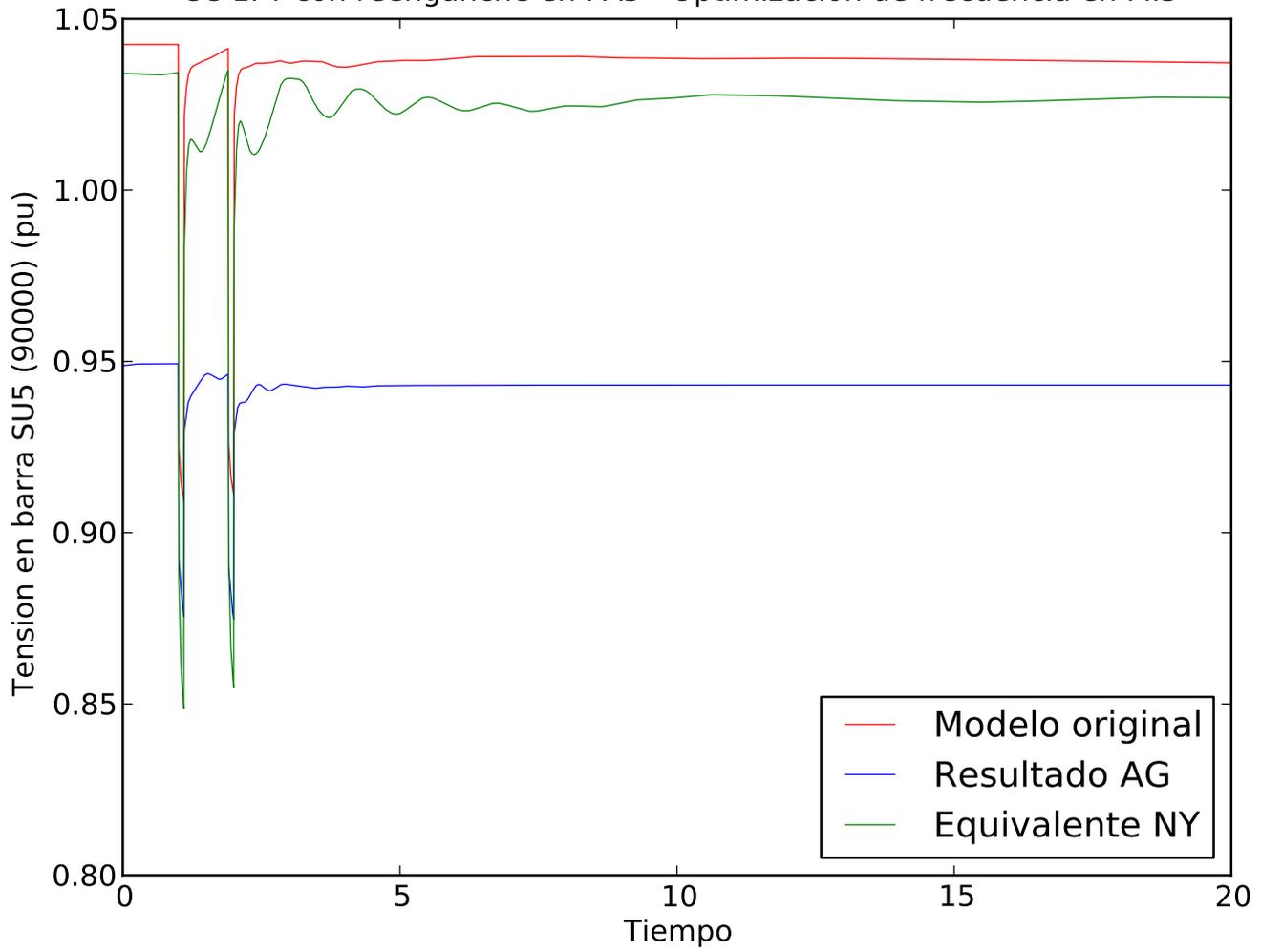
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



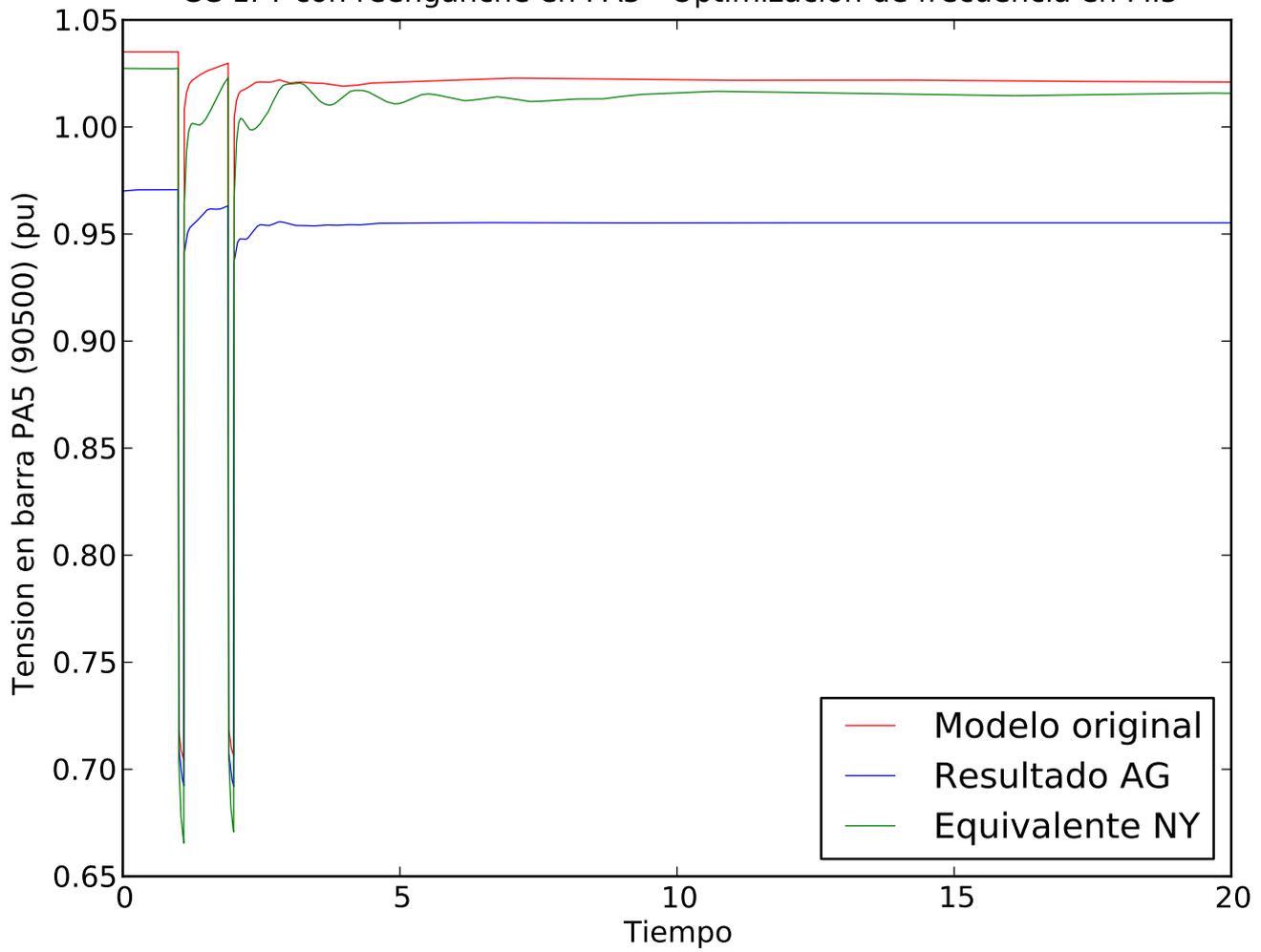
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



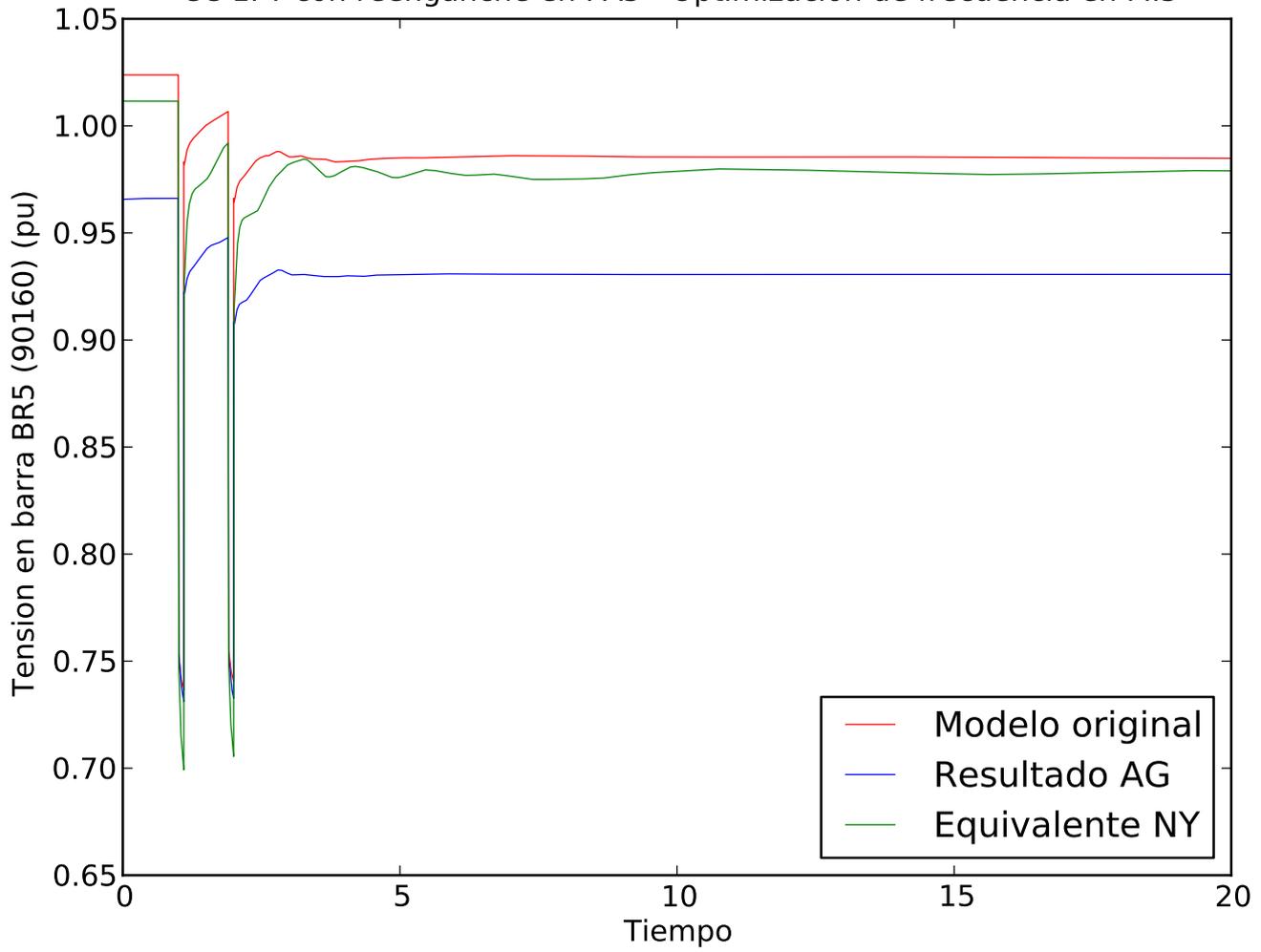
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



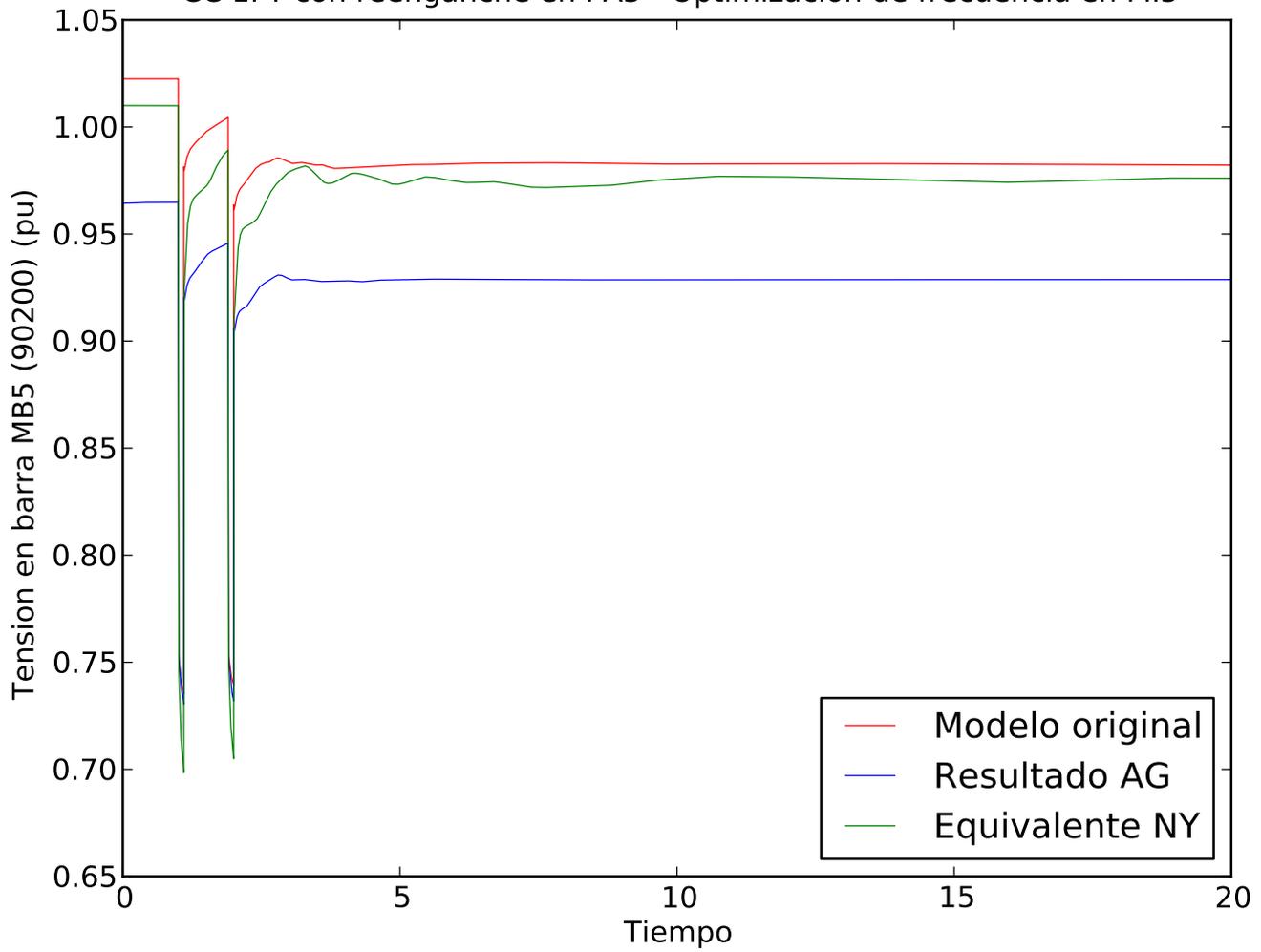
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



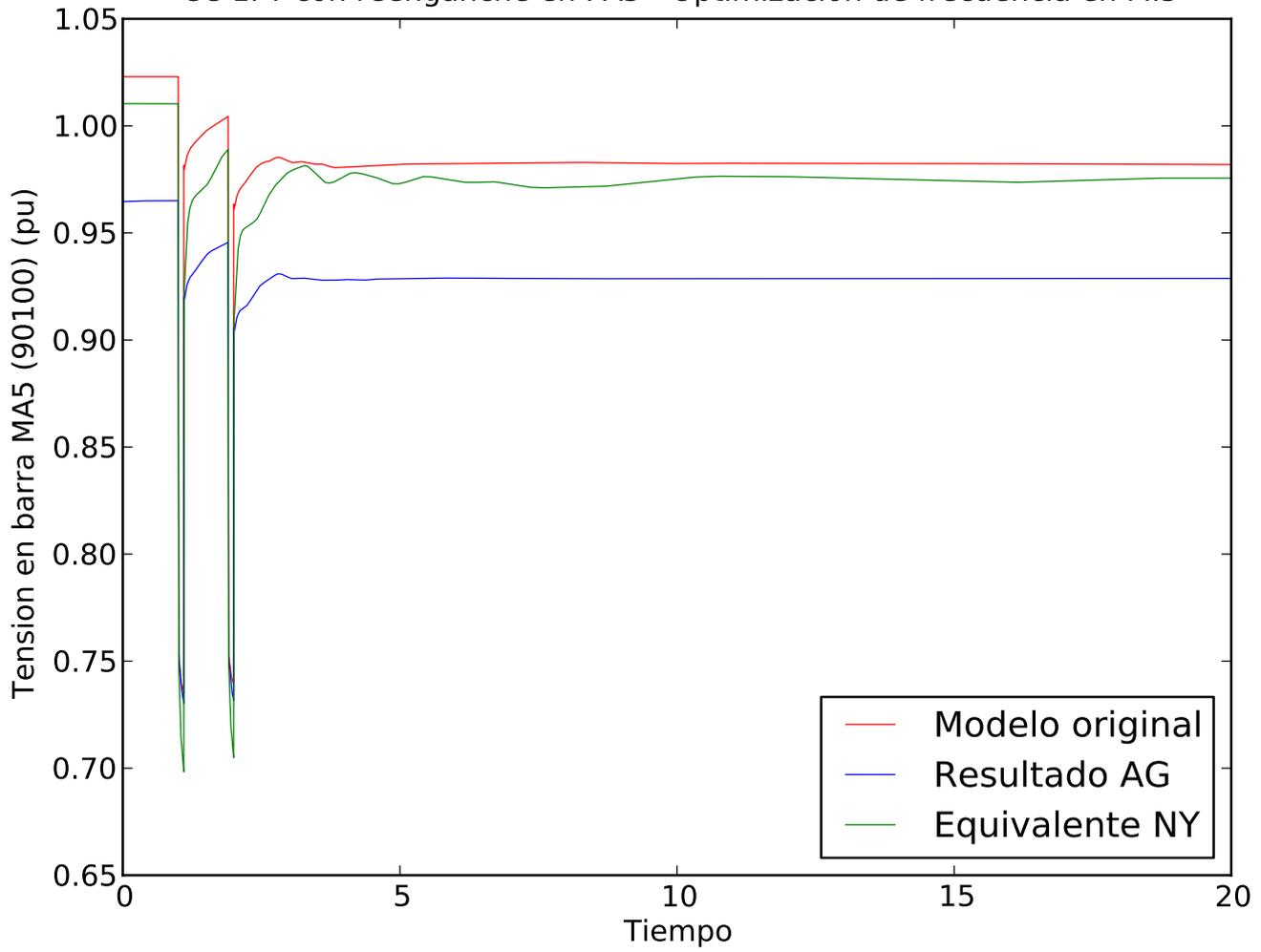
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



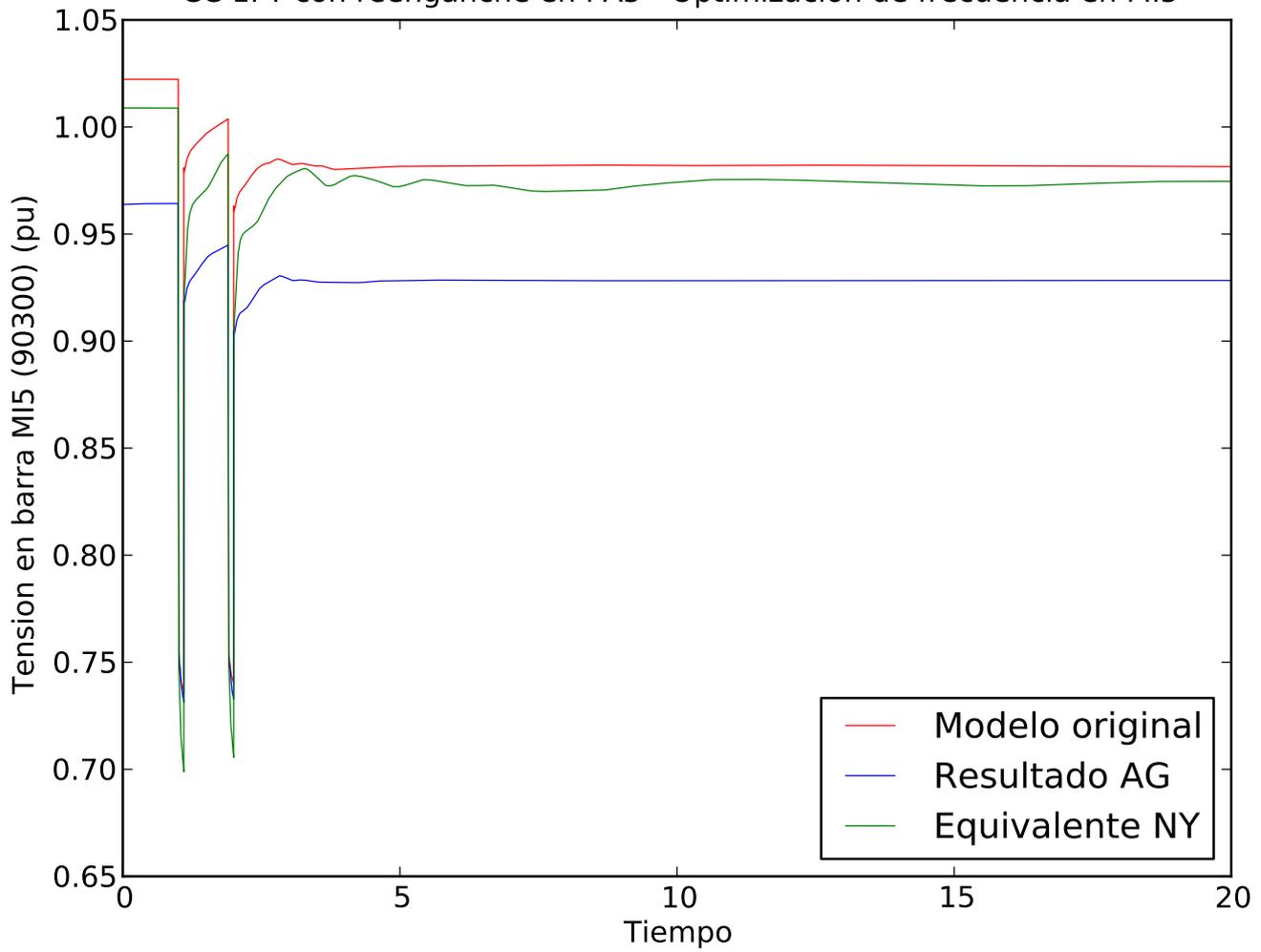
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



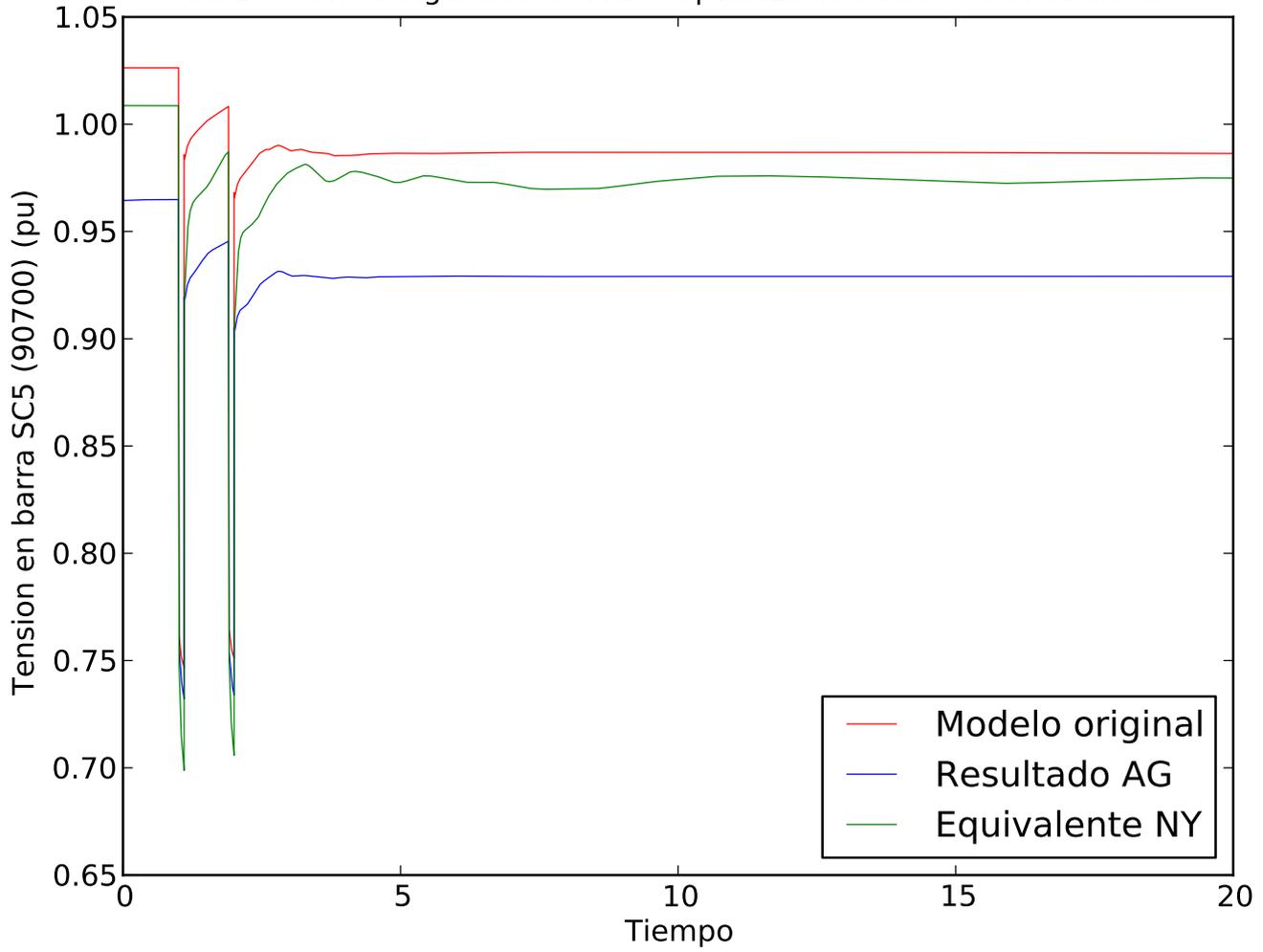
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



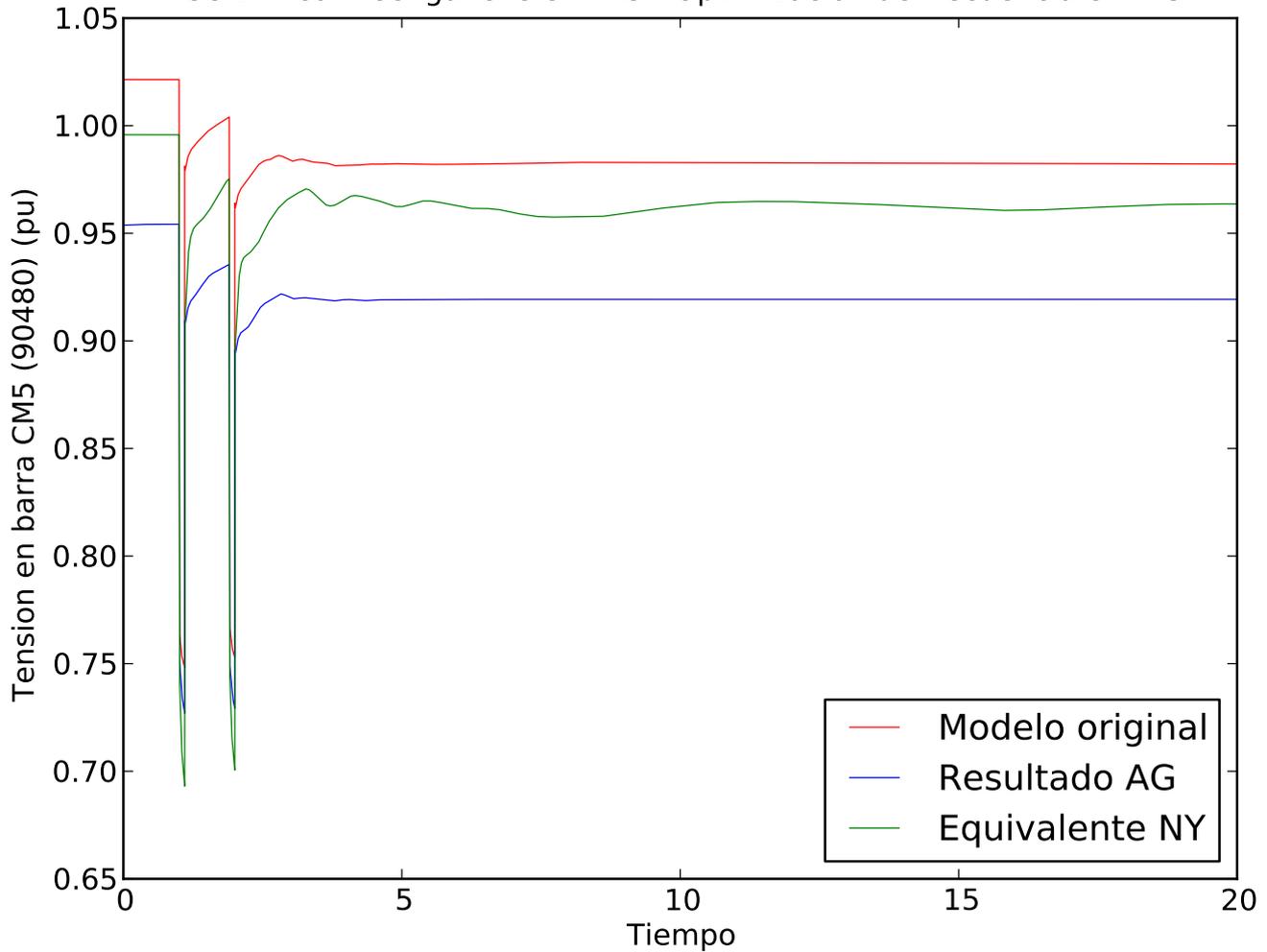
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



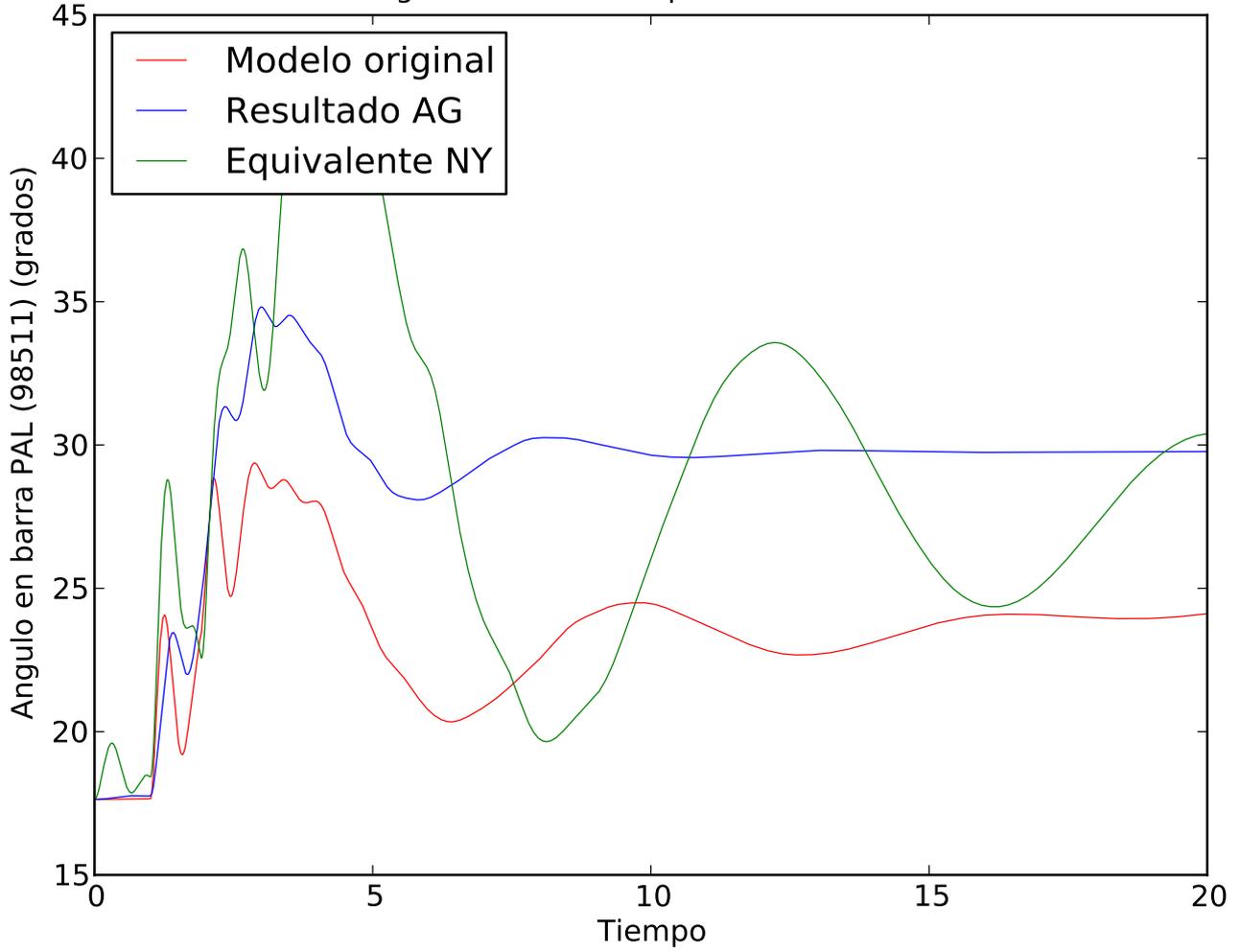
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



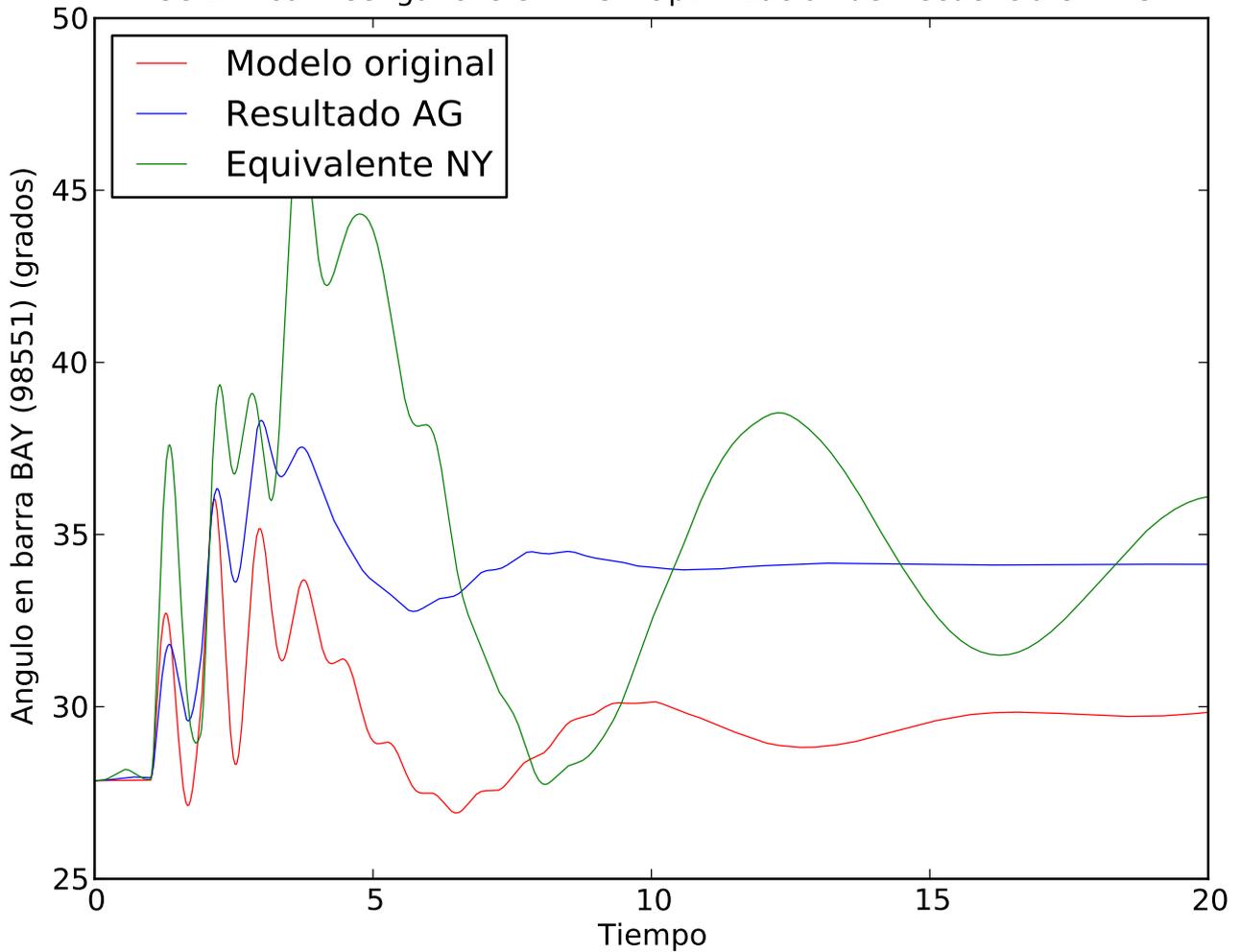
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



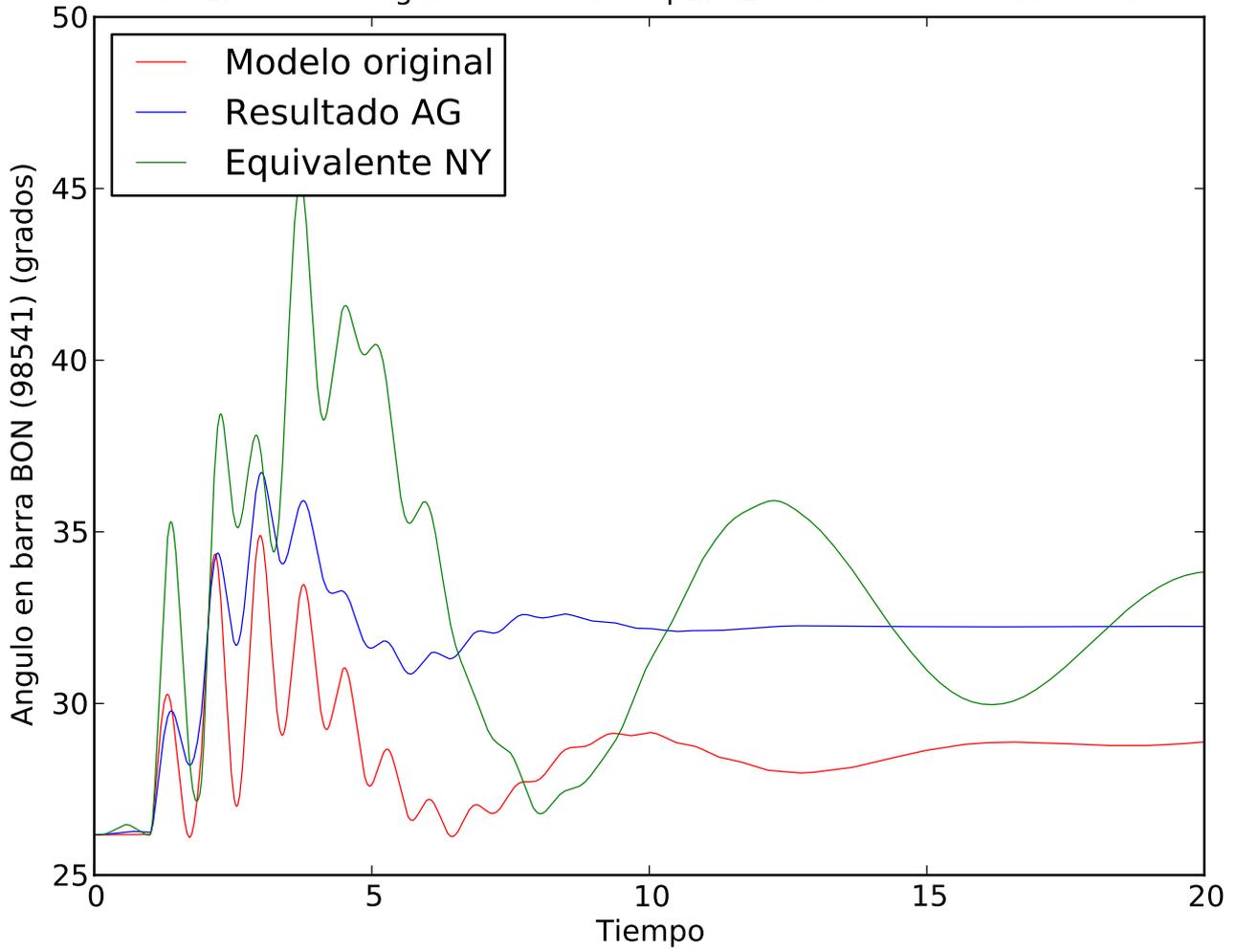
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



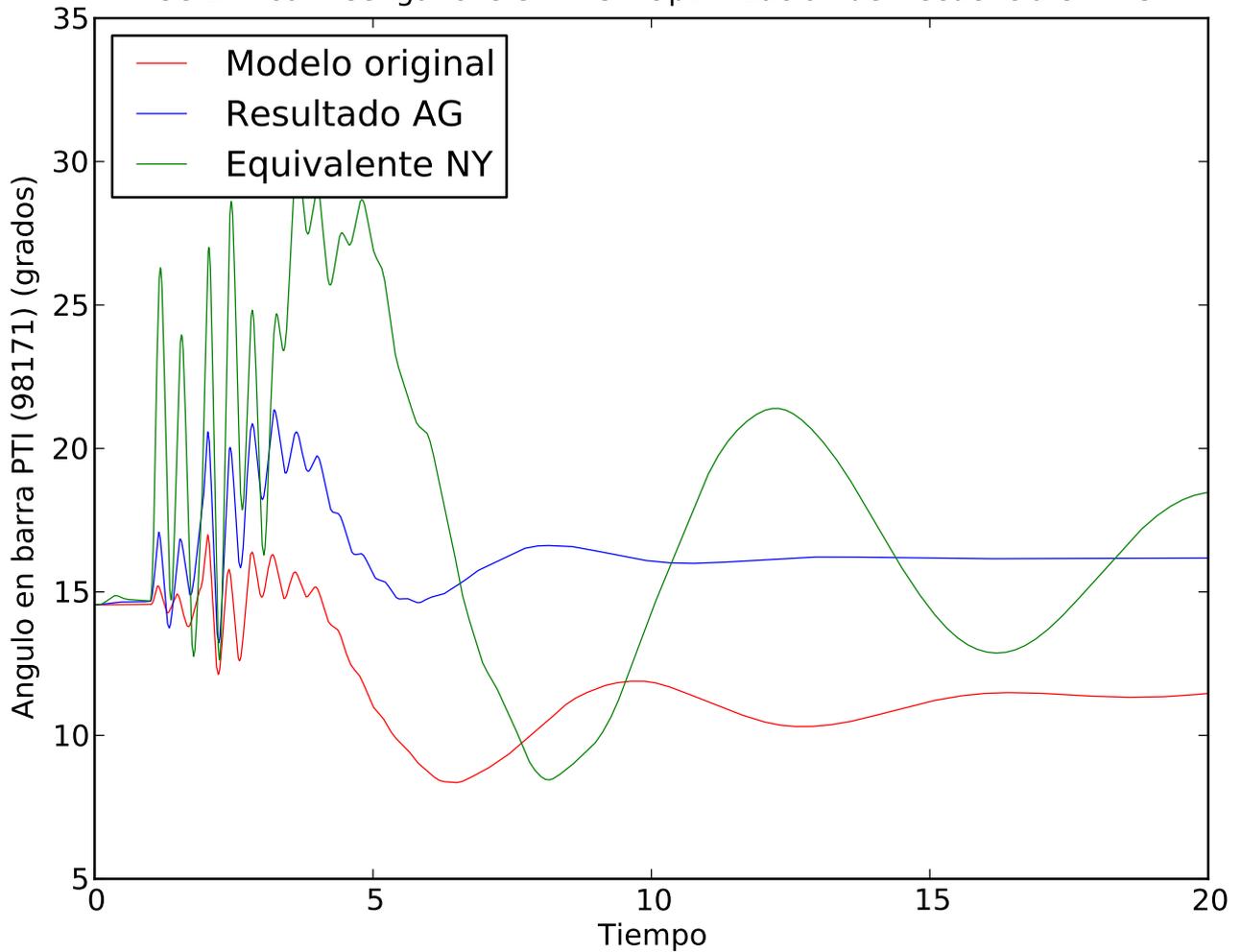
CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



CC 1FT con reenganche en PA5 - Optimizacion de frecuencia en MI5



# EDCMAG: Equivalente Dinámico Calculado Mediante Algoritmos Genéticos

Franco Pesce, Ignacio Cibils, Facundo Motta  
Facultad de Ingeniería de la Universidad de la República  
Montevideo, Uruguay

fgpesce@gmail.com, ignaciocibils@gmail.com, facundogmotta@gmail.com

**Resumen**—El presente estudio propone la optimización del cálculo de equivalentes dinámicos de sistemas eléctricos de gran potencia, previamente definidos en su topología eléctrica, mediante la aplicación de métodos no tradicionales, específicamente la aplicación de Algoritmos Genéticos, método enmarcado dentro de los llamados Algoritmos Evolutivos. El estudio es realizado desde la red eléctrica uruguaya con el objetivo de obtener un equivalente dinámico de la red eléctrica argentina, para el estudio de fenómenos dinámicos desde el SIN, buscando un equilibrio entre la magnitud del equivalente y la precisión del mismo. Finalmente, se realizaron diversas simulaciones de fallas eléctricas para medir la eficiencia del equivalente y compararlo con el sistema eléctrico original.

**Palabras Claves**—equivalente dinámico, algoritmo genético

## I. INTRODUCCIÓN

EL sistema eléctrico de Uruguay se encuentra fuertemente interconectado con el de Argentina a través de líneas de 500kV con una capacidad de interconexión de 2.000MW [1]. Esto determina que las acciones o perturbaciones que se toman u ocurren en un país afectan al otro. El sistema eléctrico argentino es aproximadamente 10 veces mayor que el uruguayo haciendo que la influencia de Argentina sobre Uruguay sea la predominante. Por este motivo, en mayor o menor medida siempre se debe considerar al país vecino para realizar estudios del sistema eléctrico.

Siempre existe la posibilidad de modelar el sistema eléctrico de ambos países de manera completa. De hecho, UTE cuenta con el modelo completo. Sin embargo, una simulación dinámica con tal grado de detalle insume mucho tiempo de ejecución y una base de datos enorme. Para UTE además implica trabajar con un sistema del que solo conoce en detalle el 10%, el otro 90% corresponde a Argentina. En algunas aplicaciones esto puede ser

una desventaja y surge la necesidad de contar con un sistema eléctrico equivalente de Argentina que represente de manera correcta el tipo de fenómenos que se quiere analizar.

## II. EQUIVALENTE DINÁMICO

Para el modelado de un sistema eléctrico se divide la red en dos zonas, la zona de estudio que comprende la red a estudiar donde se aplican las faltas y se trabaja con modelos lo más detallados posibles y la zona externa donde no se aplican perturbaciones y es posible hacer una reducción del modelo dinámico original.

Para el estudio de transitorios en sistemas eléctricos juega un papel importante la dimensión de la red a estudiar ya que realizar simulaciones dinámicas de redes muy extensas insumen tiempos de ejecución muy grandes y bases de datos enormes.

La interconexión de los sistemas eléctricos juega un rol muy importante a la hora de estudiar el comportamiento dinámico de los mismos por lo que no se puede hacer un estudio sin tener en cuenta la interconexión e interacción de los dos sistemas. La solución al problema de las dimensiones de los sistemas es encontrar un método de reducción sistemática del modelado del sistema externo de manera tal que la influencia del mismo sobre el sistema a estudiar se vea inalterada ante eventos que ocurran dentro del área de estudio. Un sistema de orden reducido del sistema externo que posea estas características es lo que se denomina *equivalente dinámico* [2].

## III. ALGORITMOS GENÉTICOS

El algoritmo de optimización propuesto se basa en la teoría de algoritmos genéticos, método de optimización que pertenece a la rama de algoritmos

evolutivos [3]. Los algoritmos genéticos se basan en la evolución de una población de individuos determinados, los cuales se reproducen y mutan, con determinadas reglas preestablecidas por el usuario. Los individuos son un conjunto de soluciones numéricas del problema de optimización. En nuestro caso fueron definidos como vectores formados por valores numéricos de constantes dinámicas de máquinas y líneas de los modelos del PSS®E, con márgenes preestablecidos para cada constante. En el caso de las máquinas se utilizó el modelo GENSAL (generador de polos salientes) y las líneas se modelan como un circuito  $\pi$ .

El modelo GENSAL cuenta con 12 constantes dinámicas [4]:

- $T'_{do}$  [sec]: constante de tiempo transitoria en el eje d.
- $T''_{do}$  [sec]: constante de tiempo subtransitoria en el eje d.
- $T''_{qo}$  [sec]: constante de tiempo subtransitoria en el eje q.
- $H$  [pu]: constante de inercia de la máquina
- $D$  [pu]: speed damping, constante de los efectos de amortiguamiento de carga.
- $X_d$  [pu]: reactancia síncrona directa.
- $X_q$  [pu]: reactancia síncrona en cuadratura.
- $X'_d$  [pu]: reactancia transitoria directa.
- $X''_d = X''_q$  [pu]: reactancias subtransitoria directa y en cuadratura.
- $X_l$  [pu]: reactancia del bobinado inductor.
- S(1,0): factor de saturación 1.0 de la curva.
- S(1,2): factor de saturación 1.2 de la curva.

El modelo  $\pi$  de cada línea de modelo con 3 constantes:

- $R$  [pu]: resistencia de la línea.
- $X$  [pu]: reactancia de la línea.
- $B$  [pu]: susceptancia de la línea, lo que equivale a la parte imaginaria de la admitancia.

### III-A. Operadores

Las reglas determinan la evolución de la población de forma de adaptarse lo mejor posible al problema de aplicación, dando lugar a la supervivencia de los individuos (soluciones) más aptos para el problema. Estas reglas (u operadores) de los AG

son tres: selección, reproducción o cruzamiento, y mutación.

La selección consiste en la elección de dos individuos de la población que se cruzarán, obteniendo otro individuo “hijo” de los anteriores. La selección de cada individuo se logra de manera probabilística, calculada mediante una función de fitness  $f(x)$  definida por el usuario. La función se definió de la siguiente forma:

$$f(x) = e^{-\alpha x}, \quad x \geq 0, \quad (1)$$

donde  $\alpha$  es un factor de corrección que aumenta la diferencia entre los individuos más aptos y los menos aptos. Cuando éste crece la brecha entre los individuos también lo hace.

Llamamos  $x$  a la suma de los errores punto a punto entre la curva de la simulación original  $a$  y la curva de simulación de nuestro equivalente  $b$ . De esta forma, se definen matemáticamente las curvas y la variable  $x$ :

$$\sum_{i=1}^n e_i = \sum_{i=1}^n |a_i - b_i| = x, \quad x \geq 0.$$

La función de fitness le asigna un valor a cada individuo, el cual determinará la probabilidad  $p_j$  de ser seleccionado para su reproducción, donde:

$$p_j = \frac{f(x_j)}{\sum_{i=1}^m f(x_i)}.$$

La reproducción consiste en mezclar el material genético (los valores numéricos de cada vector) de los individuos seleccionados. Esto se realiza de diferentes formas. La más utilizada suele elegir un punto  $i$  del vector y tomar los valores  $1, \dots, i$  de un padre y los valores  $i + 1, \dots, l$  del otro padre, siendo  $l$  el largo del vector. Los individuos “hijos” van reemplazando la generación anterior, ya que el tamaño de la población se mantiene constante. En lo que refiere al reemplazo de la población, se decidió sobrescribir el 90% de la población con fitness más bajo. Este tipo de reemplazo se denomina de tipo elitista, ya que mantiene los individuos genéticamente superiores.

Por último, la mutación de los individuos de la población consiste en la alteración aleatoria de algunos genes de ciertos individuos. Tanto los individuos como los genes a mutar de cada individuo son elegidos de manera randómica. Para cada individuo

seleccionado se sortea un gen (es decir, uno de los valores del vector) y se reemplaza por otro valor randómico, el cual deberá encontrarse entre los límites definidos al principio del programa, teniendo en cuenta la magnitud que se trate.

A continuación se describe el diagrama de flujo para un algoritmo genético general.

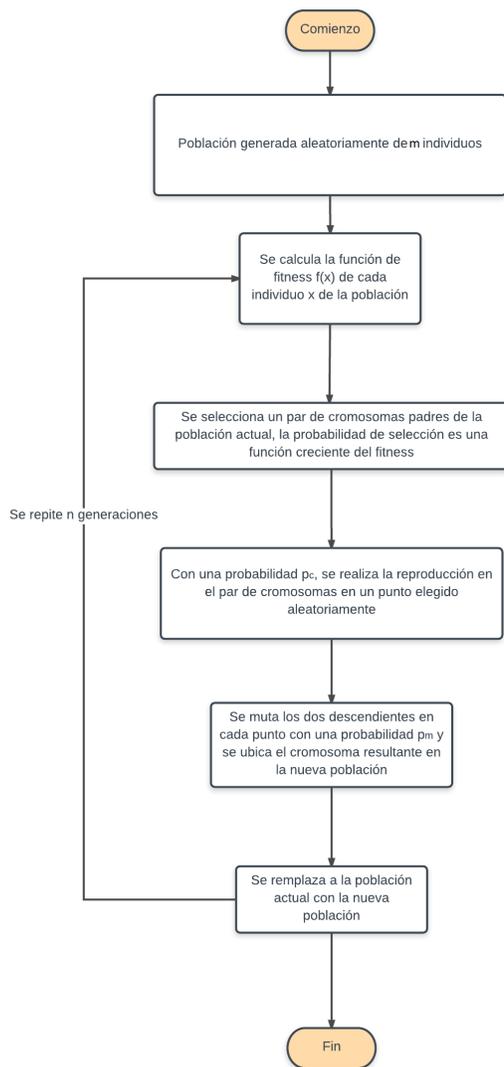


Figura 1. Diagrama de flujo para un AG genérico.

#### IV. DESCRIPCIÓN DE LA SOLUCIÓN

El problema consiste en encontrar un modelo equivalente simplificado a un sistema eléctrico complejo utilizando algoritmos genéticos como método de optimización. En este caso, el sistema eléctrico corresponde al sistema eléctrico argentino y el desafío es simplificar la totalidad de la red a

unos pocos elementos (generadores, barras, cargas y líneas) para facilitar el análisis de su influencia sobre el sistema eléctrico uruguayo cuando ocurren determinadas fallas. Los resultados consisten en simplificaciones a gran escala de la red argentina.

Por lo tanto, los pasos a seguir para cada equivalente son los siguientes:

- En principio se cuenta con la información total en PSS®E de los sistemas eléctricos argentino y uruguayo.
- A partir del sistema anterior se plantea una determinada falla en el sistema uruguayo, localizada en un determinado punto y se simula en el sistema original.
- A partir de la simulación anterior se elige la variable que se quiere optimizar (sea frecuencia, tensión, etc.) y el tiempo de la simulación (se simula los primeros 4 segundos de la falla) y se guarda el vector como resultado.
- Se elimina en su totalidad la red argentina y se realizan diversas configuraciones eléctricas que puedan representar a la red argentina, sustituyendo a la original. Estas configuraciones son realizadas manualmente, buscando equivalentes del totalidad de la red.
- Para cada una de las configuraciones, se sorteos distintos elementos del sistemas y se modifican sus magnitudes mediante el algoritmo genético (constantes de máquinas y de líneas), donde cada vector de magnitudes es un individuo de la población.
- El algoritmo genera una población inicial de individuos de forma aleatoria, donde se sorteos las diferentes magnitudes del equivalente, aunque estableciendo márgenes para cada magnitud.
- Con la población inicial generada, se ejecutan n generaciones del algoritmo genético, guardando en cada una de ellas el mejor individuo. La particularidad del algoritmo es que la función de fitness compara la simulación del sistema original con la simulación para cada individuo de la población. La función de fitness calcula la distancia entre ambas curvas con el fin de asignarles un valor y poder evaluar los diferentes individuos de la población.
- Una vez finalizado el algoritmo se tiene como resultado una matriz con n elementos, en la cual se encuentran los mejores individuos de

cada generación.

- Como resultado se elige el mejor individuo de la matriz, correspondiente al mejor individuo de las  $n$  generaciones.

## V. EQUIVALENTE NY

El equivalente analizado se trata de una optimización de un equivalente existente, creado y utilizado para estudios de la red. El equivalente se muestra en la figura 2.

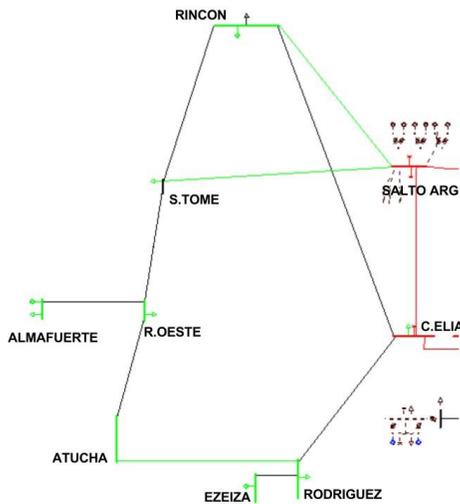


Figura 2. Equivalente NY.

En la figura 2 se muestra el equivalente dinámico que se ajusta con el programa de optimización. El sistema propuesto es significativamente menor comparado con el sistema original. El mismo está compuesto de 3 máquinas GENSAL, 7 barras, 6 cargas y 10 líneas. En primera instancia se ajustaron las 3 máquinas GENSAL pero, debido a problemas de estabilidad, finalmente, se ajustaron únicamente 2 máquinas (Almafuerte y Ezeiza) y las 10 líneas de transmisión. Esto se traduce en individuos de  $2 \times 12 + 10 \times 3 = 54$  variables de largo.

## VI. FALLAS Y SIMULACIONES

Para el estudio dinámico de nuestros equivalentes se eligieron dos tipos de faltas a probar, en diferentes puntos de la red y optimizando diferentes magnitudes también en diferentes puntos, a saber:

- Cortocircuito trifásico con salida de una línea: esta falta simula un cortocircuito trifásico en

una línea. La falta se ejecuta por 3 ciclos (60ms) y luego se despeja la falta y se abren las líneas, simulando la apertura de las protecciones.

- Cortocircuito fase a tierra con reenganche: esta falta describe el cierre y recierre luego de una falta monofásica. Se ejecuta la falta y luego se abre una fase durante 800ms. Luego se reengancha sobre la falta, ya que esta no se extinguió, por lo que el reenganche no fue exitoso. Finalmente se abren las tres fases 100ms después y se saca de servicio toda la línea.

Las magnitudes optimizadas fueron tres: frecuencia y tensión en barras y el ángulo  $\delta$  de los diferentes generadores, es decir, el ángulo que determina la potencia de las máquinas síncronas. Para las máquinas de polos salientes la potencia se define de la siguiente forma [5]:

$$P = \frac{3V_s E}{X_d} \sin \delta + \frac{3V_s^2}{2} \left( \frac{1}{X_q} - \frac{1}{X_d} \right) \sin 2\delta. \quad (2)$$

Este ángulo es el ángulo entre las fuerzas magnetomotrices del estator y del rotor.

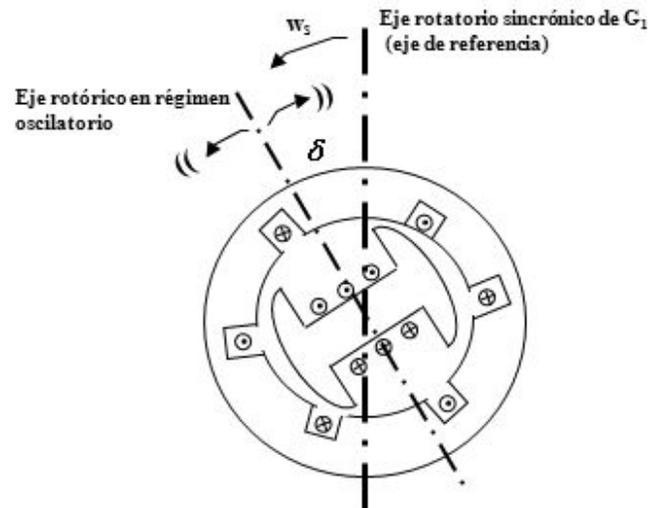


Figura 3. Ángulo  $\delta$  en un corte transversal de una MS de polos salientes.

## VII. IMPLEMENTACIÓN DEL PROGRAMA

El programa principal consiste en la combinación de dos algoritmos: un algoritmo genético general, el cual se adaptó para aplicar la función de fitness

al resultado de la simulación de cada individuo generado en la población, y no al individuo en sí, como suele hacerse generalmente, y un algoritmo de simulación de los diferentes casos que se irán probando (una simulación de una falta en PSS®E y un algoritmo genético ordinario). Luego, se modifica el cálculo de fitness del AG, para poder comparar el resultado de la falta en el sistema original con los resultados obtenidos para el equivalente dinámico. Para ello, se intentó buscar una forma de comparar curvas que minimice la distancia entre ellas y le dé mayores probabilidades de "sobrevivir" a aquellas curvas que guardan una mayor similitud con la curva que arrojó la simulación del sistema original. Es un enfoque a resultados, que quizás pueda facilitar y ahorrar la difícil tarea de cálculo que imponen los métodos tradicionales. A su vez, el programa simula sobre dos sistemas eléctricos previamente definidos en PSS®E, por lo que se debe contar con los respectivos modelos y sus archivos para poder ejecutar la simulación. Fue desarrollado en lenguaje python, en la plataforma Eclipse Luna, y ejecuta los modelos eléctricos en el PSS®E 32. Se utilizó la versión de python 2.5, ya que es la que mejor se adaptó a los requerimientos del PSS®E 32, y la cual está incluida en la instalación del programa.

### VIII. RESULTADOS

Se constataron algunos de los resultados más intuitivos. Los resultados más próximos se dieron para la magnitud y el punto que estábamos optimizando. Asimismo, a medida que nos alejamos del punto de optimización y nos acercamos al equivalente, vamos perdiendo precisión en la magnitud que se optimizó. Otro punto quizás también intuitivo es que se obtuvieron mejores resultados en general para las faltas monofásicas que para las faltas trifásicas. Esto se desprende de que las faltas monofásicas son las menos agresivas para el sistema, ya que la potencia del cortocircuito es menor, lo que varía en menor medida la velocidad de las máquinas, y por lo tanto la frecuencia y los ángulos  $\delta$  de las máquinas.

Los puntos de la red elegidos para las simulaciones fueron los siguientes:

- MA5: Estación de 500kV Montevideo A.
- MA5: Estación de 500kV Montevideo I.
- PA5: Estación de 500kV Palmar.
- PAL: Máquinas de Represa de Palmar.

- PTI: Máquinas de Central de Ciclo Combinado Punta del Tigre.

A continuación se muestran algunos de los resultados obtenidos. En rojo se puede apreciar el sistema original, en verde el equivalente NY con sus valores originales y en azul el equivalente NY luego de la optimización EDCMAG.

#### *Cortocircuito IFT en barra PA5 Optimización de ángulo de máquina en PTI*

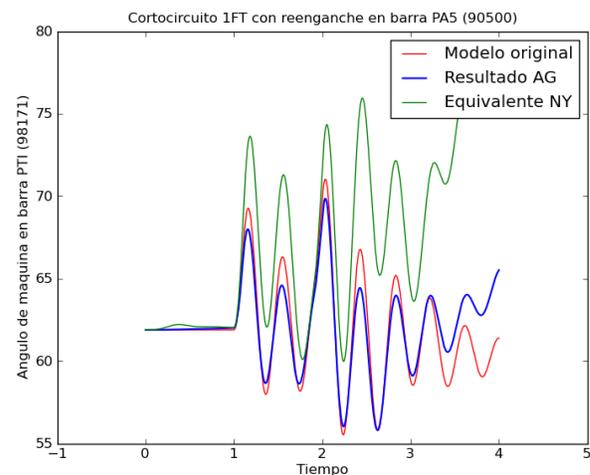


Figura 4. Resultado del mejor individuo de todas las generaciones optimizando el ángulo  $\delta$ .

#### *Cortocircuito IFT en barra PA5 Optimización de frecuencia en MI5*

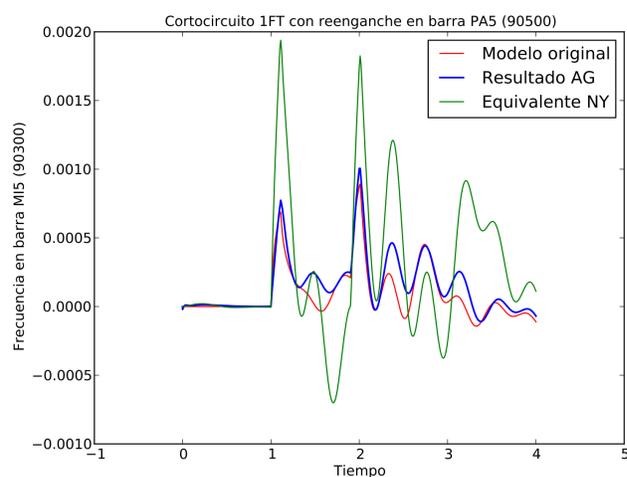


Figura 5. Resultado del mejor individuo de todas las generaciones optimizando frecuencia.

### Cortocircuito 3F en barra MA5 Optimización de tensión en MA5

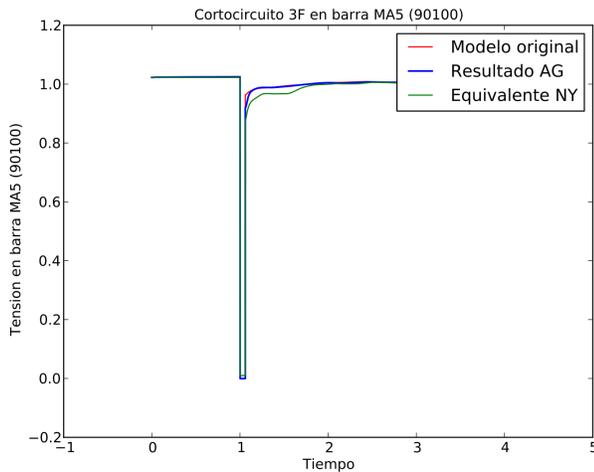


Figura 6. Resultado del mejor individuo de todas las generaciones optimizando tensión.

En general se encontraron tendencias para las simulaciones en que se optimizaron las mismas magnitudes, aunque también se pudo encontrar una tendencia general en los datos. Esto se aprecia mayormente en la inercia y el damping de las máquinas, las cuales varían ambas en un amplio rango. El damping se varía en un rango de 0 a 40 pu, mientras que la inercia se varía en un rango de 1 a 40 pu. Sobre este punto vale la pena recalcar que los resultados de los equivalentes dieron en general valores muy por encima de los valores límite de las máquinas especificados en la documentación del PSS®E: la inercia debe variar entre 1 y 10 pu y el damping debe hacerlo entre 0 y 3 pu. De aquí se desprende que con los límites recomendados los resultados no fueron buenos y se decidió ampliar los márgenes para ambas constantes. Respecto a las demás constantes de las máquinas, los márgenes fueron más restrictivos, sea por su sensibilidad y el funcionamiento del programa, así como que la ampliación de los márgenes tampoco arrojaba una mejora sustancial en los resultados, sino que todo lo contrario, resultaba en la caída de python para algunos valores fuera de los márgenes definidos.

## IX. CONCLUSIONES

En primer lugar, se encontró que la optimización de más de dos máquinas a la vez no es recomendable y difícilmente se obtengan buenos resultados.

Es por esto que se recomienda optimizar de dos en dos las máquinas del sistema equivalente que quiera optimizar.

En el caso de la red argentina, se logra una reducción a 3 máquinas, 10 líneas y 7 barras, y se obtienen resultados con un margen de error aceptable. Esta es una reducción muy significativa ya que la red argentina tiene un número de barras que es aproximadamente 10 veces mayor que el de la red eléctrica uruguaya. El hecho de optimizar un equivalente de este tamaño ya nos impone limitantes para representar todo el sistema argentino.

Otro punto a tener en cuenta es que la precisión del algoritmo depende de la magnitud que se esté optimizando, y en el punto en el que se está optimizando. Vale la pena destacar que se tuvo una mayor precisión en otras magnitudes al optimizar el sistema por el ángulo de las máquinas.

Del análisis de las gráficas para los distintos tiempos se pudieron comprobar varios puntos:

- Se tuvieron muy buenos resultados en la optimización de frecuencia para las distintas faltas. Incluso los resultados para tiempos de simulación de 4 segundos pueden ser utilizados para tiempos mayores ya que ambos valores logran estabilizarse luego de unos pocos segundos a la frecuencia nominal.
- En el caso de los ángulos de las máquinas, los resultados son muy buenos para tiempos cortos como los que se simuló, pero no logran ajustarse para tiempos mayores, ya que los ángulos del sistema original y el equivalente no logran coincidir luego de estabilizarse el sistema.
- Al ajustar el equivalente dinámico por los ángulos de las máquinas, el sistema tiene un buen comportamiento en frecuencia. Esto se debe a que la frecuencia debe estabilizarse siempre al mismo valor de la red, así como que una variación del ángulo de las máquinas causa una variación proporcional en la frecuencia.
- Los resultados en tensión fueron buenos al optimizar justamente en tensión en el lugar de la falta y en lugares cercanos, pero no ocurrió lo mismo cuando se intentó ajustar por otras magnitudes. Esto se debe principalmente a que no se optimizaron reguladores de tensión.

También se ha constatado que las oscilaciones de las máquinas, y por lo tanto el comportamiento dinámico de la red, está regido principalmente por

las constantes de inercia  $H$  y el damping  $D$ , como se desprende de la ecuación de oscilación, y de las constantes de las líneas de transmisión  $R$ ,  $X$  y  $B$ .

#### REFERENCIAS

- [1] REPRESA HIDROELÉCTRICA SALTO GRANDE ARGENTINA-URUGUAY, *Cuadrilátero de Potencia* <https://www.saltogrande.org/transmision.php>
- [2] JOSÉ JUAN MANUEL RAMÍREZ ARREONDO, *Técnicas Modernas para el Análisis de Estabilidad Transitoria en Sistemas Eléctricos Interconectados Mediante la Aplicación de Equivalentes Dinámicos*, Universidad Autónoma de Nuevo León, Facultad de Ingeniería Mecánica y Eléctrica, Tesis de doctorado, Monterrey N.L. Diciembre de 1992.
- [3] MELANIE MITCHELL, *An Introduction to Genetics Algorithms*, A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England, Sixth Printing, 1999.
- [4] *Siemens PSSE 32.0.5 Volume 2 Program Application Guide, Chapter 12.4 Classification of Variables*, Octubre 2010.
- [5] CURSO DE MÁQUINAS ELÉCTRICAS, FACULTAD DE INGENIERÍA, UDELAR, *Capítulo 22: Máquina Síncrona en Régimen Lineal y Capítulo 24: Estabilidad de MS*.

# EDCMAG: Equivalente Dinámico Calculado Mediante Algoritmos Genéticos

Autores: Franco Pesce, Ignacio Cibils, Facundo Motta

Tutor: Nicolás Yedrzejewski

## Problema

El sistema eléctrico de Uruguay se encuentra fuertemente interconectado con el de Argentina a través de líneas de 500kV con una capacidad de interconexión de 2.000MW. Esto determina que las acciones o perturbaciones que se toman u ocurren en un país afectan al otro. El sistema eléctrico argentino es aproximadamente 10 veces mayor que el uruguayo haciendo que la influencia de Argentina sobre Uruguay sea la predominante. Por este motivo, en mayor o menor medida siempre se debe considerar al país vecino para realizar estudios del sistema eléctrico. Siempre existe la posibilidad de modelar el sistema eléctrico de ambos países de manera completa. De hecho, UTE cuenta con el modelo completo. Sin embargo, una simulación dinámica con tal grado de detalle insume mucho tiempo de ejecución y una base de datos enorme. En algunas aplicaciones esto puede ser una desventaja y surge la necesidad de contar con un sistema eléctrico equivalente de Argentina que represente de manera correcta el tipo de fenómenos que se quiere analizar.

## Equivalente dinámico

Para el estudio de transitorios en sistemas eléctricos juega un papel importante la dimensión de la red a estudiar ya que realizar simulaciones dinámicas de redes muy extensas insumen tiempos de ejecución muy grandes y bases de datos enormes.

La interconexión de los sistemas eléctricos juega un rol muy importante a la hora de estudiar el comportamiento dinámico de los mismos por lo que no se puede hacer un estudio sin tener en cuenta la interconexión e interacción de los dos sistemas. La solución al problema de las dimensiones de los sistemas es encontrar un método de reducción sistemática del modelado del sistema externo de manera tal que la influencia del mismo sobre el sistema a estudiar se vea inalterada ante eventos que ocurran dentro del área de estudio. Un sistema de orden reducido del sistema externo que posea estas características es lo que se denomina **equivalente dinámico**.

## Algoritmos genéticos

El algoritmo de optimización propuesto se basa en la teoría de algoritmos genéticos, método de optimización que pertenece a la rama de algoritmos evolutivos. Los algoritmos genéticos se basan en la evolución de una población de individuos determinados, los cuales se reproducen y mutan, con determinadas reglas preestablecidas por el usuario. Los individuos son un conjunto de soluciones numéricas del problema de optimización.

## Referencias

- [1] JOSÉ JUAN MANUEL RAMÍREZ ARREONDO, *Técnicas Modernas para el Análisis de Estabilidad Transitoria en Sistemas Eléctricos Interconectados Mediante la Aplicación de Equivalentes Dinámicos*, Universidad Autónoma de Nuevo León, Facultad de Ingeniería Mecánica y Eléctrica, Tesis de doctorado, Monterrey N.L. Diciembre de 1992.
- [2] MELANIE MITCHELL, *An Introduction to Genetics Algorithms*, A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England, Sixth Printing, 1999.

## Equivalente dinámico de la red eléctrica argentina

En la figura 1 se muestra el equivalente que se utilizó - en verde - que denominamos equivalente NY interconectado con la red uruguaya. El sistema propuesto es significativamente menor comparado con el sistema original. El mismo está compuesto de 3 máquinas GENSA, 7 barras, 6 cargas y 10 líneas. Se ajustaron únicamente 2 máquinas (Almafuerte y Ezeiza) y las 10 líneas de transmisión. Esto se traduce en individuos de  $2 \times 12 + 10 \times 3 = 54$  variables de largo.

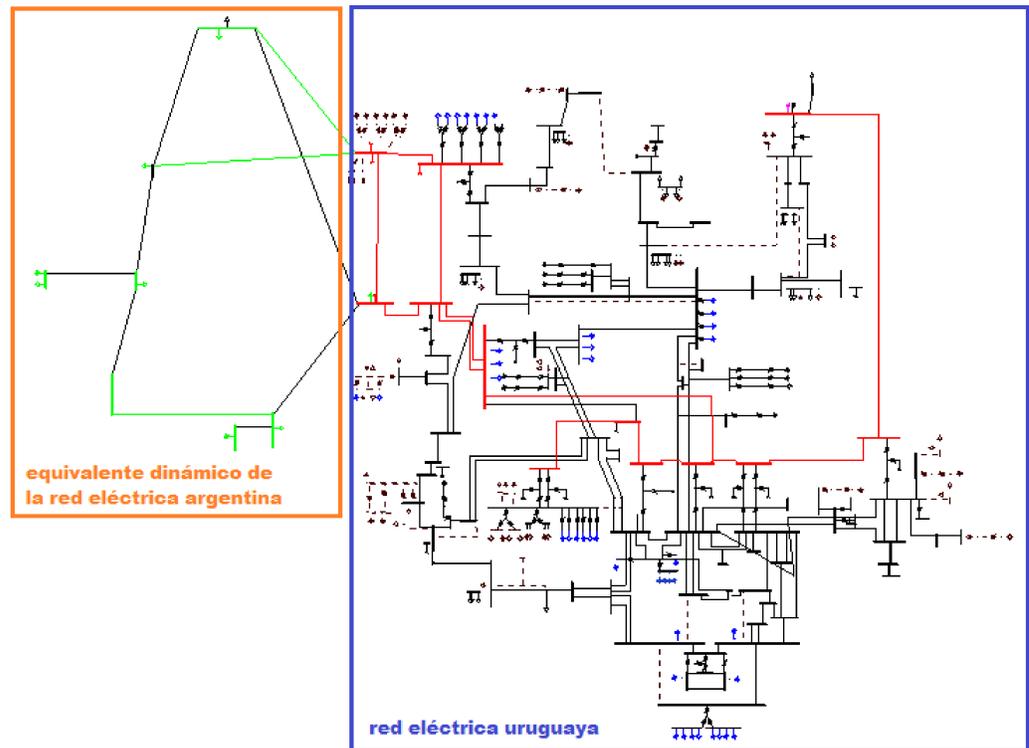


Figura 1: Equivalente NY interconectado con red uruguaya.

## Comportamiento en frecuencia y en ángulo $\delta$

Del análisis de las gráficas para los distintos tiempos se pudieron comprobar varios puntos:

- La optimización en frecuencia fue muy buena para las distintas faltas. Los resultados para tiempos de simulación de 4 segundos pueden ser utilizados para tiempos mayores ya que la frecuencia tiende a estabilizarse rápidamente.

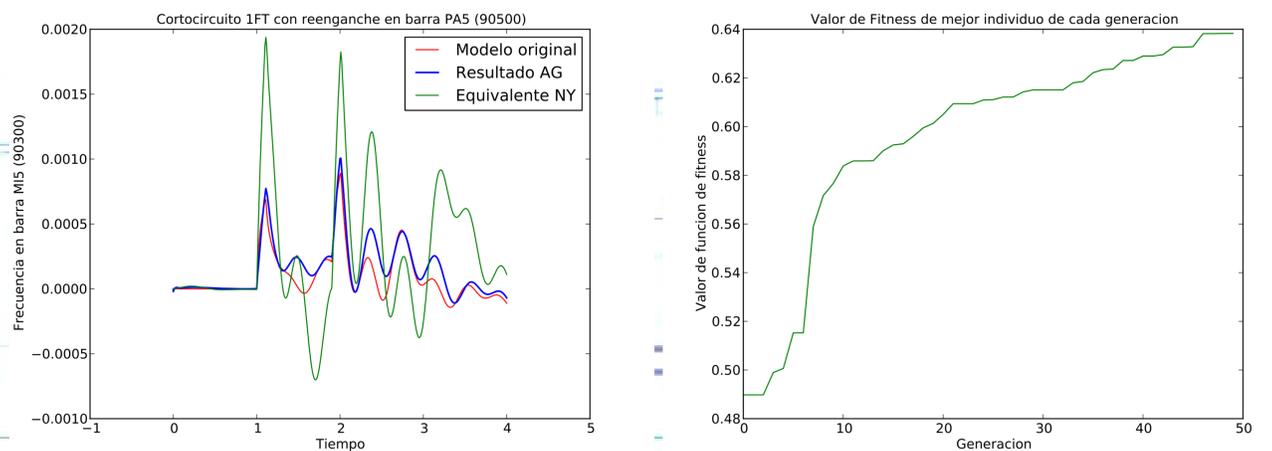


Figura 2: Comportamiento en frecuencia y evolución de la función de ajuste de la solución.

- En el caso de los ángulos de las máquinas, los resultados son buenos para los tiempos simulados, pero no logran ajustarse para tiempos mayores, ya que los ángulos del sistema original y el equivalente no logran coincidir luego de estabilizarse el sistema.
- Al ajustar el equivalente dinámico por los ángulos de las máquinas, el sistema tiene un buen comportamiento en frecuencia. Esto se debe a que la frecuencia debe estabilizarse siempre al mismo valor de la red.

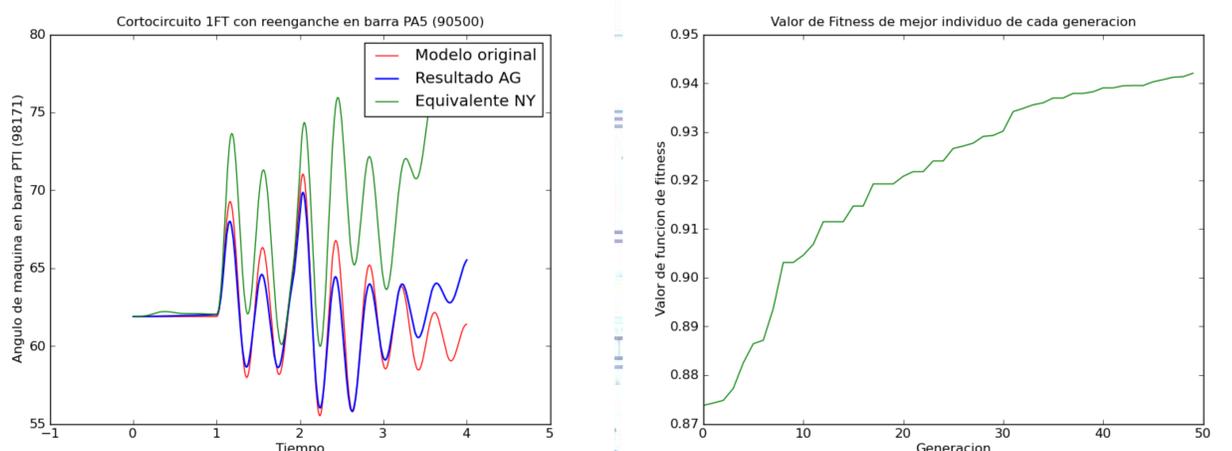


Figura 3: Comportamiento en ángulo y evolución de la función de ajuste de la solución.