



UNIVERSIDAD VERACRUZANA

**FACULTAD DE INGENIERÍA DE LA
CONSTRUCCIÓN Y EL HÁBITAT**

**Balanceo dinámico de un rotor acoplado a un
motor de inducción mediante armónicas de 30 y 90 Hz
de la corriente residual.**

TRABAJO RECEPCIONAL EN LA MODALIDAD DE:
TESIS

COMO REQUISITO PARCIAL PARA OBTENER EL TÍTULO DE:
MAESTRO EN INGENIERÍA APLICADA

PRESENTA:
ING. RUBÍ ARELY MORENO PÉREZ

ASESOR:
DR. ALFONSO CUAUHEMOC GARCÍA REYNOSO

MAYO 2021
BOCA DEL RIO, VER.

Agradecimientos

Esta tesis está dedicada especialmente a mi familia, ya que sin su inspiración y motivación habría sido imposible llevar a cabo la culminación de mi preparación a nivel posgrado. A mis padres Rufina y Alejandro, que confiaron en mis habilidades para lograr una meta más en mi carrera profesional, siendo un ejemplo de lucha, compromiso y honestidad. De igual forma a mi hermano Ruhle, ya que, sin él no habría sobrevivido a la dura batalla contra el COVID-19, por su arduo trabajo y dedicación médica.

Para mi asesor Dr. Alfonso Cuauhtemoc García Reynoso, por su colaboración en el desarrollo del proyecto mediante la facilitación de las instalaciones, aparatos, además de su apoyo constante y adaptabilidad debido a las limitantes generadas por la contingencia sanitaria.

Para mi tutor Dr. Agustín Leobardo Herrera May, quien me alentó y motivando, a dar lo mejor de mí durante el transcurso de la maestría. A mis maestros, quienes nunca desistieron al enseñarme, sembrando una curiosidad para la adquisición de conocimiento continuo y profesional, a ellos que continuaron depositando su esperanza en mí. Muchas gracias

Finalmente, debo agradecer al consejo Nacional de ciencia y Tecnología (CONACYT) por el apoyo económico brindado mediante la beca con CVU 947673, el cual se implementó para absorber los gastos generados durante mi estancia en la Facultad de Ingeniería de la Construcción y el Hábitat, en la sección de posgrados de la Universidad Veracruzana.

Índice

Agradecimientos	2
Índice de figuras.....	6
Índice de tablas.....	10
Índice de ecuaciones	11
Nomenclatura	16
Resumen	17
CAPITULO 1. Introducción.....	18
1.1 Planteamiento del problema.....	20
1.2 Hipótesis.....	21
1.3 Objetivo general	22
1.4 Objetivo específico	22
1.5 Justificación.....	22
CAPITULO 2. Estado del arte	25
2.1 Introducción.....	25
2.2 Máquinas de inducción	25
2.3 Motor de inducción	25
2.3.1 Funcionamiento del motor de inducción	26
2.4 Clasificación del motor de inducción.....	27
2.4.1 Motor monofásico de inducción con espiras en cortocircuito	27
2.4.2 Motor monofásico de inducción tipo fase partida.....	27
2.4.3 Motor de jaula de ardilla	27
2.5 Máquinas rotativas	28
2.6 Desbalance	28
2.7 Vibraciones mecánicas.....	29
2.8 Detección de señales	31
2.9 Metodología para detección de fallas	32
2.10 Tipos de análisis de vibración.....	32
2.10.1 Análisis espectral de vibraciones	32
2.10.2 Análisis espectral de corriente.....	32
2.10.3 Análisis espectral del flujo axial.....	33
2.10.4 Impedancia de secuencia inversa	33
2.10.5 Par electromagnético	33
2.11 Equipos implementados para el balanceo analógico	34
2.12 Balanceo	34

2.12.1 Balanceo estático (un plano)	36
2.12.2 Balanceo dinámico (2 planos)	38
2.13 Balanceo implementando la corriente residual.....	39
2.13.1 Análisis del Balanceo en un plano mediante la corriente residual	39
2.13.2 Análisis de balanceo en dos planos a través de la corriente residual	43
2.14 Determinación del offset de las armónicas de la corriente eléctrica	46
2.15 Condición de intersección de los círculos en un punto común.	47
2.16 Equipo e instrumentación implementado para el balanceo mediante la corriente residual	49
2.16.1 Adquisición de datos	49
2.16.2 Software controlador	50
2.16.3 Software de aplicación	50
2.17 Implementación de instrumentación virtual	50
CAPITULO 3. Metodología.....	52
3.1 Calibración del motor.....	52
3.1.1 Equipo y/o herramienta para el balanceo analógico	52
3.1.2 Evaluación de vibración	52
3.2 Balanceo dinámico implementando armónicas de 30 y 90 Hz de la corriente residual.....	55
3.2.1 Determinación del peso de balanceo aplicando dos pesos de prueba	56
3.3 Selección de peso y ángulos de prueba para el balanceo en un plano.....	59
3.3.1 Peso de prueba 6.7g.....	60
3.3.2 Peso de prueba 8.2g.....	61
3.4 Selección de pesos de prueba para el balanceo en dos planos.....	62
3.4.1 Ángulos iguales.....	62
3.4.2 Ángulos opuestos.....	64
3.5 Implementación de LabVIEW	65
3.6 Algoritmo 12A y 12AA.....	70
3.6.1 Implementación de fasores con parte real e imaginaria.....	72
3.6.2 Implementación de fases	73
3.7 Algoritmo 8H6.....	74
3.8 Algoritmo 8H16.....	76
Capítulo 4. Discusiones y resultados	79
4.1 Introducción.....	79
4.2 Balanceo dinámico implementando armónicas de 30 Hz.....	79
4.2.1 Balanceo en un plano con peso de prueba de 6.7g.....	79

4.2.2	Balanceo en un plano con peso de prueba de 8.2g.....	82
4.2.3	Balanceo en dos planos ángulos iguales con peso de prueba de 6.7g.....	85
4.2.4	Balanceo en dos planos ángulos distintos con peso de prueba de 6.7g.....	91
4.2.5	Balanceo en dos planos ángulos iguales con peso de prueba de 10g.....	94
4.3	Balanceo dinámico implementando armónicas de 90 Hz.....	100
4.3.1	Balanceo en un plano con peso de prueba de 6.7g.....	100
4.3.2	Balanceo en un plano con peso de prueba de 8.2g.....	103
4.3.3	Balanceo en dos planos ángulos iguales con peso de prueba de 6.7g.....	106
4.3.4	Balanceo en dos planos ángulos distintos con peso de prueba de 6.7g.....	112
4.3.5	Balanceo en dos planos ángulos iguales con peso de prueba de 10g.....	115
4.4	Discusión de resultados.....	121
Capítulo 5. Conclusiones y trabajos a futuro		123
5.1	Conclusiones.....	123
5.2	Recomendaciones y trabajos a futuro.....	124
Referencias.....		125
Anexos.....		127
A1.	Algoritmo 12A (armónicas de 30Hz).....	127
A12.	Algoritmo 12AA (armónicas de 90 Hz).....	131
A3.	Algoritmo 8H6	135
A4.	Algoritmo 8H16	154

Índice de figuras

Figura 1 Balanceo en un plano, analizador de vibraciones	36
Figura 2 Marcas de fase	37
Figura 3 Balanceo en dos planos.....	38
Figura 4 Rotor rígido con dos presos de desbalance	43
Figura 5 Descomposición de los pesos de balanceo	45
Figura 6 Partes de un sistema DQA	49
Figura 7 Marcas de perforaciones en los platos del rotor.....	55
Figura 8 Platos y flecha del rotor	55
Figura 9 Armónicas representadas por vectores.....	58
Figura 10 Medición del peso de prueba, 6.7g	60
Figura 11 Medición de peso de prueba, 8.2g.....	61
Figura 12 Medición de pesos fijos.....	62
Figura 13 Magnitud y ángulo de balanceo en dos planos	63
Figura 14 Conexión de fases y pinzas Fluke.....	65
Figura 15 Pinzas Fluke en posición 1, 2 y 3, correspondientes a cada línea de alimentación.....	66
Figura 16 Tarjeta National Instruments	66
Figura 17 Selección de canales físico.....	67
Figura 18 Señales de entrada.....	68
Figura 19 Ajuste de filtro	68
Figura 20 Señales filtradas	69
Figura 21 Guardar datos.....	69
Figura 22 Apertura de algoritmo 12A en Scilab.....	71
Figura 23 Archivos corrfas generados por el algoritmo 12A.....	71
Figura 24 Datos del archivo corrf4	72

Figura 25 Resultados de la 1ra combinación del peso de prueba 6.7g	75
Figura 26 Resultados de la 2da combinación del peso de prueba 6.7g	75
Figura 27 Rango de evaluación (offset)	75
Figura 28 Amplitudes de corriente en miliamperios a 30Hz con valor mínimo de 10.13, máximo de 44.20 y promedio de 26.28	76
Figura 29 Amplitudes de corriente en miliamperios a 90 Hz, con valores mínimo de 178, máximo de 211.12 y promedio de 194.15	77
Figura 30 Resultados de la 1ra combinación de 6.7g en un plano	77
Figura 31 Resultados de la 2da combinación de 6.7g en un plano	78
Figura 32 Comportamientos de las fases respecto a W_{c1} , 6.7g a 30 Hz	80
Figura 33 Comportamiento de W_{c1} para 6.7g a 30 Hz.....	81
Figura 34 Comportamiento de las fases respecto a W_{c1} para 8.2g a 30 Hz	83
Figura 35 Comportamiento de W_{c1} promedio para 8.2g a 30 Hz.....	84
Figura 36 Comportamiento de las fases para W_{c1} , ángulos iguales con peso de prueba 6.7g a 30Hz	86
Figura 37 Comportamiento de las fases para W_{c2} , ángulos iguales con peso de prueba 6.7g a 30 Hz	87
Figura 38 Comportamiento de W_{c1} promedio, ángulos iguales con peso de prueba 6.7g a 30 Hz	89
Figura 39 Comportamiento de W_{c2} promedio, ángulos iguales con peso de prueba 6.7g a 30 Hz	90
Figura 40 Comportamiento de las fases para W_{c1} , ángulos opuestos con peso de prueba 6.7g a 30 Hz	92
Figura 41 Comportamiento de W_{c1} promedio, ángulos distintos con peso de prueba 6.7g a 30 Hz	93

Figura 42 Comportamiento de las fases para W_{c1} , ángulos iguales con peso de prueba 10g a 30 Hz	95
Figura 43 Comportamiento de las fases para W_{c2} , ángulos iguales con peso de prueba 10g a 30 Hz	96
Figura 44 Comportamiento del offset para W_{c1} promedio, ángulos iguales con peso de prueba 10g a 30 Hz	98
Figura 45 Comportamiento del offset para W_{c2} , ángulos iguales con peso de prueba 10g a 30 Hz	99
Figura 46 Comportamientos de las fases respecto a W_{c1} , 6.7g a 90 Hz	101
Figura 47 Comportamiento de W_{c1} promedio para 6.7g a 90 Hz	102
Figura 48 Comportamiento de las fases respecto a W_c , para 8.2g a 90 Hz	104
Figura 49 Comportamiento de W_{c1} promedio para 8.2g a 90 Hz	105
Figura 50 Comportamiento de las fases para W_{c1} , ángulos iguales con peso de prueba 6.7g a 90 Hz	107
Figura 51 Comportamiento de las fases para W_{c2} , ángulos iguales con peso de prueba 6.7g a 90 Hz	108
Figura 52 Comportamiento de W_{c1} promedio, ángulos iguales con peso de prueba 6.7g a 90 Hz	110
Figura 53 Comportamiento de W_{c2} promedio, ángulos iguales con peso de prueba 6.7g a 90 Hz	111
Figura 54 Comportamiento de las fases para W_{c1} , ángulos distintos con peso de prueba 6.7g a 90 Hz	113
Figura 55 Comportamiento de W_{c1} promedio, ángulos distintos con peso de prueba 6.7g a 90 Hz	114
Figura 56 Comportamiento de las fases para W_{c1} , ángulos iguales con peso de prueba 10g a 90 Hz	116

Figura 57 Comportamiento de las fases para W_{c2} , ángulos iguales con peso de prueba 10g a 90 Hz	117
Figura 58 Comportamiento de W_{c1} promedio, ángulos iguales con peso de prueba 10g a 90 Hz	119
Figura 59 Comportamiento de W_{c2} promedio, ángulos iguales con peso de prueba 10g a 90 Hz	120

Índice de tablas

Tabla 1 Valor inicial de la medición	53
Tabla 2 Datos para realizar el primer ajuste	53
Tabla 3 Resultados del primer ajuste	54
Tabla 4 Datos para realizar el segundo ajuste	54
Tabla 5 Valor inicial en un plano	54
Tabla 6 Resultado final	54
Tabla 7 Mediciones de amplitud de armónica a 30hz o a 90hz	56
Tabla 8 Magnitud y ángulo de balanceo en ambos planos	63
Tabla 9 Valores de la medición con 6.7g a 0° con filtro de 90 Hz	73
Tabla 10 Datos de peso de prueba 6.7g en 0° a 30 Hz	74
Tabla 11 Valores de W_c con peso de prueba 6.7g a 30 Hz	81
Tabla 12 Valores de W_c con peso de prueba 8.2g a 30 Hz	82
Tabla 13 Valores de W_c ángulos iguales a 30 Hz	88
Tabla 14 Valores de W_c para ángulos distintos a 30Hz	91
Tabla 15 Valores de W_c para ángulos iguales con 10 g a 30 Hz	97
Tabla 16 Valores de W_c con peso de prueba de 6.7g a 90Hz	100
Tabla 17 Valores de W_c con peso de prueba de 8.2g a 90 Hz	103
Tabla 18 Valores de W_c con ángulos iguales con peso de prueba 6.7g a 90 Hz	109
Tabla 19 Valores de W_c , ángulos distintos con peso de prueba de 6.7g a 90 Hz	112
Tabla 20 Valores de W_c , ángulos distintos con peso de prueba de 10g a 90 Hz	118
Tabla 21 Resultados del balanceo en un plano a 30 Hz	121
Tabla 22 Resultados del balanceo en un plano a 90 Hz	121
Tabla 23 Resultados del balanceo en dos planos a 30 Hz	122
Tabla 24 Resultados del balanceo en dos planos a 90 Hz	122

Índice de ecuaciones

(1).....	28
(2).....	37
(3).....	38
(4).....	40
(5).....	40
(6).....	40
(7).....	40
(8).....	40
(9).....	40
(10).....	40
(11).....	41
(12).....	41
(13).....	41
(14).....	41
(15).....	41
(16).....	41
(17).....	41
(18).....	41
(19).....	42
(20).....	42
(21).....	42
(22).....	42
(23).....	42
(24).....	42
(25).....	42

(26).....	44
(27).....	44
(28).....	44
(29).....	44
(30).....	44
(31).....	44
(32).....	44
(33).....	44
(34).....	44
(35).....	44
(36).....	45
(37).....	45
(38).....	45
(39).....	45
(40).....	45
(41).....	45
(42).....	45
(43).....	46
(44).....	46
(45).....	47
(46).....	47
(47).....	47
(48).....	47
(49).....	47
(50).....	47
(51).....	47

(52).....	47
(53).....	47
(54).....	48
(55).....	48
(56).....	48
(57).....	48
(58).....	48
(59).....	48
(60).....	48
(61).....	49
(62).....	49
(63).....	56
(64).....	56
(65).....	56
(66).....	57
(67).....	57
(68).....	57
(69).....	57
(70).....	57
(71).....	57
(72).....	57
(73).....	57
(74).....	58
(75).....	58
(76).....	58
(77).....	58

(78).....	58
(79).....	59
(80).....	59
(81).....	59
(82).....	59
(83).....	60
(84).....	60
(85).....	60
(86).....	60
(87).....	60
(88).....	60
(89).....	60
(90).....	60
(91).....	61
(92).....	61
(93).....	61
(94).....	61
(95).....	61
(96).....	61
(97).....	61
(98).....	62
(99).....	62
(100).....	62
(101).....	62
(102).....	63
(103).....	63

(104).....	63
(105).....	63
(106).....	63
(107).....	64
(108).....	64
(109).....	64
(110).....	64
(111).....	64
(112).....	64
(113).....	64
(114).....	64
(115).....	64
(116).....	64
(117).....	64
(118).....	64
(119).....	65
(120).....	65
(121).....	65
(122).....	72
(123).....	72
(124).....	73
(125).....	73
(126).....	73
(127).....	73
(128).....	74

Nomenclatura

DQA	Adquisición de datos.
A	Coefficiente de influencia
W_c	Peso de balanceo
O	Offset (fasor)
N	Fasor de desbalance
r	Relación $\frac{W_2}{W_1}$ (fasor).
W_r	Peso de balanceo
R_0	Lectura de amplitud tal cual y magnitud de N.
R_{21}	Lectura de amplitud con primer peso de prueba y magnitud de N_{21} (fasor de prueba 1).
R_{22}	Lectura de amplitud con primer peso de prueba y magnitud de N_{22} (fasor de prueba 2).
W_1	Peso de prueba 1.
W_2	Peso de prueba 2
θ_0	Posición angular de N.
fem	Es la diferencia de potencial entre dos puntos de un circuito abierto o de producir una corriente eléctrica en un circuito cerrado.

Resumen

Se plantea un procesamiento de mediciones de corriente residual de un motor de inducción que se acopla directamente a un rotor con desbalance mecánico. Las armónicas de 30 y 90hz tienen relación indirecta con el desbalance mecánico. Un algoritmo que ajusta a estos datos para obtener fasores proporcionales al desbalance. El rotor de pruebas se utiliza en tres series de experimentos.

La primera serie de pruebas parte de un rotor balanceado, al cual se le adiciona un peso de desbalance en un plano, que se irá posicionando en diferentes ángulos con la finalidad de obtener mediciones en distintas áreas de desbalance del motor, que permitirán la realización de combinaciones para el cálculo de balanceo.

En la segunda serie de experimentos, seleccionamos dos pesos de prueba iguales que se irán posicionando en ángulos iguales en ambos planos del motor, con el propósito de comprobar si la metodología es apta para el balanceo en dos planos.

La tercera serie de experimentos selecciona dos pesos de pruebas iguales, que serán posicionados en ángulos opuestos en ambos planos del motor, esto implica realizar corridas modificando la posición angular del peso de prueba.

La serie de experimentos tiene como objetivo determinar el peso y posición de balanceo mediante un algoritmo capaz de procesar estos datos, además de la implementación de una innovadora metodología.

CAPITULO 1. Introducción

El desbalance es la condición en la que el centro de gravedad de una pieza no coincide con su eje de rotación, produciendo fuerzas y/o momentos, los cuales generan vibraciones mecánicas que a su vez producen armónicas en la corriente residual a diferentes frecuencias simultáneamente, debido a que las armónicas son corrientes o voltajes presentes en un sistema eléctrico.

El motor de inducción es usado para la transformación de energía eléctrica en energía mecánica. Son empleados en conjunto con otros elementos para proporcionar energía a diferentes tipos de equipos o sistemas, como lo son: ventiladores, bombas, bandas transportadoras, etc.

Al acoplarse a dichos sistemas, se modifican sus características mecánicas, esto debido al uso de cargas no lineales, lo cual genera cambios en su respuesta mecánica, provocando una susceptibilidad en su rotor (desbalance).

Un motor eléctrico puede presentar este tipo de alteraciones, debido a que es un sistema electromecánico, el cual además se encuentra expuesto a problemas presentes en las instalaciones eléctricas causados por los efectos de las componentes armónicas de corrientes y voltajes en el sistema eléctrico.

Esta situación puede llegar a causar un funcionamiento incorrecto de muchos equipos (especialmente en los menos robustos) que han sido diseñados para operar bajo condiciones normales (poca distorsión armónica).

Debido a la exposición continua del motor de inducción a distintos tipos de alteraciones, se ha decidido implementar la detección de fallas, principalmente de cinco maneras diferentes: mediciones de temperatura, mediciones químicas, mediciones de vibraciones mecánicas, mediciones eléctricas y detección de PD.

Siendo de mayor interés la técnica de medición eléctrica, ya que se encuentra fácilmente disponible en los convertidores de potencia, además de ser accesible para la implementación y desarrollo del procesamiento de señal.

Dicho método desarrolló la curiosidad de distintos investigadores, uno de ellos fue García et al. (2008), implementó un método de balanceo en un plano mediante el coeficiente de influencia, que permite calcular el peso de balanceo del rotor tanto en magnitud como en posición angular, mediante fórmulas matemáticas para las vibraciones con pulsaciones en donde existen dos o más armónicas con frecuencias similares, las cuales se suman y producen una resultante cuya magnitud varía entre un mínimo y un máximo, con una periodicidad que depende de la diferencia entre las frecuencias de las armónicas. (García Reynoso, y otros, 2008)

En el mismo año Alberto Bellini menciona la incorporación del diagnóstico de fallas del motor por medio de múltiples señales eléctricas, esto mediante el método de monitoreo de "Viena" basado en el voltaje, las señales de corriente y la posición medida del rotor para verificar las desviaciones obtenidas por dos modelos de máquinas diferentes. El procedimiento se propone como un índice de diagnóstico cuantitativo independiente de la inercia y la carga, implementando una amplia gama de parámetros para la realización del cálculo (estator, resistencia, reactancia de magnetización del rotor, posición del rotor y la constante de tiempo del rotor).

Sin embargo, la identificación confiable y el aislamiento de fallas aún están bajo investigación, ya que existen algunos problemas sin concluir, como lo son:

- Definición de un único procedimiento de diagnóstico para la identificación y aislamiento de cualquier tipo de fallas.
- Insensibilidad a las condiciones de operación.
- Detección confiable de fallas para accionamientos controlados por posición, velocidad y torque.

- Detección de fallas confiable para variadores en condiciones de tiempo variable.
- Detección cuantitativa de fallas para establecer un umbral de falla absoluto, independiente de las condiciones de operación. (Bellini, Filippetti, Tassoni, & Capolino, 2008)

1.1 Planteamiento del problema

Este trabajo analiza la posibilidad de determinar los pesos y ángulos de desbalance mediante la implementación de fasores de determinadas armónicas, que son proporcionales a las vibraciones presentes en un rotor acoplado al motor de inducción, el cual es generado por las alteraciones en sus campos magnéticos (devanados del motor).

Por lo cual se plantean los siguientes cuestionamientos:

- ¿Por qué se debe implementar el fasor? Debido a la proporcionalidad de la fuerza de inercia que se presenta en el motor.
- ¿Para qué se implementa la corriente eléctrica del motor? para utilizar la medición de la corriente eléctrica generada en el motor debido a la alteración de sus campos magnéticos presentes en las bobinas al aplicarle algún esfuerzo externo.
- ¿Por qué se usan las tres fases de la corriente? Esto se implementa para tener más información en el comportamiento del motor.
- ¿Cuál es la relación entre las frecuencias de 30 y 90 Hz? Debido a la relación de los fasores de 30 y 90 Hz, ya que se mantienen constantes sin importar las fuerzas de desbalance presentes en el rotor, convirtiéndose en una alternativa viable para la realización del estudio.

Este método es impulsado por distintas investigaciones realizadas por científicos, los cuales se centran en el procesamiento de la señal eléctrica proporcionada por el motor. Prácticamente García et al. (2016) midieron el espectro de la corriente residual, para determinar los fasores de desbalance equivalentes a los datos de vibración, sin embargo, enfrenta

dificultades en la variabilidad de los datos en la muestra generado en un tiempo mayor de lectura, además de la presencia de un offset de los datos obtenidos en cada corrida de prueba, el cual es ajustado por un algoritmo que sustrae el efecto para formar fasores de desbalance que son proporcionales a las fuerzas desbalanceadas. Los casos de prueba realizados sobre un rotor acoplado directamente a un motor eléctrico de inducción dan como resultado masas cuyo error con respecto al valor esperado es menor al 8% en magnitud y menor a 4° en la posición angular. (García Reynoso, y otros, 2016)

Mientras que Marco Tarabini et. all (2018) desarrollaron una metodología que consiste en combinar técnicas de procesamiento de señales para la identificación del desequilibrio del rotor, evaluándolos con diferentes algoritmos para obtener un promedio de los datos. La técnica se basa en la fusión de datos a nivel característicos e implementando la incertidumbre de medición del desequilibrio para el cálculo del peso de manera estática, dinámica e híbrida. Donde el estático representa los pesos calculados en una fase utilizando cuatro algoritmos (transformadas de Fourier, ajustes cuasi- armónicos de la señal sintetizada con las transformaciones de Hilbert-Huang, descomposición de la vibración de Hilbert y descomposición Wavelet) para estimar el desequilibrio conocido. (Tarabini & Scaccabarozzi, 2018)

1.2 Hipótesis

Se propone la ejecución de una metodología para determinar los fasores de desbalance en magnitud y ángulo de fase, a través de la medición de la corriente eléctrica generada en el motor, debido a las alteraciones de sus campos magnéticos, implementando la relación de los fasores de frecuencia 30 y 90 Hz para un motor de inducción de 1800 rpm, que se mantienen constante sin importar las fuerzas de desbalance del rotor. El cual servirá para la obtención de datos que posteriormente se aplicarán para la determinación del balanceo del rotor acoplado a un motor de inducción.

1.3 Objetivo general

Desarrollar una metodología que incorpora un instrumento virtual para realizar el balanceo dinámico de rotores acoplados a motores de inducción, a través del análisis de su espectro de la corriente residual.

1.4 Objetivo específico

Implementar una técnica para la realización del balanceo dinámico de rotores acoplados a motores de inducción, utilizando información obtenida del espectro de frecuencias de las armónicas de 30 y 90 Hz presentes en la corriente eléctrica del motor.

1.5 Justificación

Prabhat Kumar y Rajiv Tiwari (2020) proponen un nuevo enfoque para la estimación de la desalineación como un concepto similar al desbalance de prueba en el balanceo del rotor, mediante el modelado y análisis del comportamiento vibratorio dinámico de un rotor rígido desbalanceado con desalineamiento en los discos descentrados sostenidos por dos cojinetes AMB. Para la ejecución de su metodología desarrollaron un modelo matemático sustentado por los principios de la transformada rápida de Fourier (FFT), el cual proporciona números complejos como salida: magnitud y fase de respuesta, siendo la fase el único parámetro variante durante el proceso de conversión. Esta señal es obtenida mediante sensores sin contactos posicionados cerca del eje del rotor. Asimismo, integraron una técnica de prueba de desalineación adicional para identificar cuantitativamente la falla de desalineación residual. Desarrollaron un algoritmo de identificación que implementa armónicas de respuestas generados por FFT para estimar diversos parámetros del sistema relacionados con los parámetros de falla del equilibrio, junto con los parámetros AMB desalineados. El algoritmo es robusto y eficaz contra señales ruidosas y errores de modelado a múltiples velocidades de giro del rotor. Los autores consideran validar de manera experimental su metodología en investigaciones futuras, implementando el modelado de elemento finito del rotor flexible desbalanceado y desalineado con un mayor número de discos. (Kumar & Tiwari, 2020)

Asimismo, A R Bhende (2020) planteó una técnica de equilibrio en un rotor de dos planos, utilizando un método de sustracción de amplitud (ASM), donde el rotor se equilibra observando amplitudes de variación en los cojinetes de soporte, y no requiere de medición del ángulo de fase. Mediante un acelerómetro posicionado en el eje del rotor se capturan las vibraciones generadas, que son recolectadas por una tarjeta de adquisición de datos. Estas señales de vibración se analizaron aplicando la técnica de transformada de Fourier (FFT), para examinar el dominio de la frecuencia en el software LabVIEW 2015. Realizó la experimentación física, mediante la colocación de una masa de prueba (10% del peso del rotor) en el plano de transmisión del rotor a un ángulo de 0° , seguidamente encendió el rotor y se midieron las amplitudes de vibración en ambos cojinetes de soporte, para continuar con un cambio de posición del mismo peso de prueba a cada una de las posiciones angulares asignadas (120° y 240°). A continuación, se calcularon los pesos de equilibrio correctivos mediante un método gráfico, en donde se dibujan tres círculos en una gráfica polar. Con los datos generados se emplea el algoritmo para el cálculo del ángulo correctivo. Este método muestra una reducción de vibración del 40% y 17.6% respectivamente, en comparación a otras investigaciones, sin embargo, el algoritmo se desarrolló considerando la dirección de rotación a un ángulo menor a 180° , lo que restringe la utilización de ángulos mayores, ya que al exceder este rango es considerado contra la rotación del rotor. (Bhende, 2020)

Por lo cual se plantea desarrollar y ejecutar una técnica para el balanceo dinámico mediante la medición de la corriente residual del motor, para así poder disminuir el error en las lecturas producido por el factor humano (incorrecta colocación del sensor durante la lectura), debido a que estos dispositivos deben colocarse cerca de los puntos de apoyos del rotor, el cual no es de fácil acceso para el operador.

Además de ofrecer una opción de bajo costo, con ventajas en su aplicación industrial, ya que las mediciones se producirán en el origen de los cables de alimentación eléctrica.

Se crea un algoritmo que calcula el ángulo de fase de las lecturas ajustadas de las tres líneas de alimentación para 30 y 90 Hertz, ejecutando un análisis matemático que sustenta al algoritmo. Para poder comprobar este método se efectúan pruebas con distintos pesos y ángulos que desbalancearán al rotor.

CAPITULO 2. Estado del arte

2.1 Introducción

Este capítulo abarcará conceptos utilizados a lo largo de la investigación de manera general. Abordará terminologías de balanceo que se emplearán constantemente, además de explorar nuevos dispositivos y plataformas que impactarán en el desarrollo de una nueva metodología para el mantenimiento preventivo y correctivo en las máquinas rotativas.

2.2 Máquinas de inducción

Son sistemas eléctricos altamente simétricos debido a su campo magnético giratorio, por lo que cualquier tipo de falla modifica sus propiedades simétricas.

La asimetría en los devanados de una máquina eléctrica genera la redistribución de corriente, esto debido al disturbio generado en el campo magnético y por consiguiente al esfuerzo mecánico al que se somete la máquina. Cualquier disturbio en la máquina será detectable en la parte mecánica o eléctrica.

2.3 Motor de inducción

Los motores asíncronos o de inducción son un tipo de motor de corriente alterna, en el que la corriente eléctrica del rotor necesaria para producir torsión es generada por inducción electromagnética del campo de la bobina del estator. Por lo cual, no requiere de una conmutación mecánica aparte de su misma excitación.

El primer prototipo de motor eléctrico capaz de funcionar con corriente alterna fue desarrollado y construido por el ingeniero Nikola Tesla, presentado en el American Institute of Electrical Engineers (Instituto Americano de Ingenieros Eléctricos, actualmente IEEE) en 1888.

El motor asíncrono trifásico está formado por un rotor, que puede ser de dos tipos:

- Jaula de ardilla.
- Bobinado y un estator: cuenta con bobinas inductoras trifásicas y están desfasadas entre sí, 120° en el espacio.

El teorema de Ferraris, Indica que cuando circula un sistema de corrientes trifásicas equilibradas en la bobina, cuyo desfase en el tiempo es de 120° , incita un campo magnético

giratorio que envuelve al rotor. Este campo variable induce una tensión eléctrica en el rotor, según la Ley de inducción de Faraday.

La diferencia entre el motor de inducción y el universal es que, en el motor a inducción el devanado del rotor no está conectado al circuito de excitación del motor, sino que está eléctricamente aislado.

2.3.1 Funcionamiento del motor de inducción

Implementa el principio de inducción mutua de Faraday, ya que al aplicar corriente alterna trifásica a las bobinas inductoras, produce un campo magnético giratorio conocido como “campo rotante”, cuya frecuencia será igual a la corriente alterna de alimentación del motor. El campo realiza un movimiento giratorio alrededor del motor en reposo, induciendo corriente para generar un campo magnético que seguirá el movimiento del campo estático, produciendo un par en el motor que hace que el rotor gire (principio de inducción mutua).

Debido a que la inducción en el rotor se produce al haber una diferencia de velocidades relativas del campo estático y rotórico, la velocidad del rotor nunca alcanzará al campo rotante. Inversamente, si ambas velocidades fueran iguales, no habría inducción y el rotor no produciría par.

Los motores de inducción son denominados asíncronos, ya que la velocidad rotatoria difiere levemente del campo rotante. Este tipo de rotor durante su arranque presenta una velocidad relativa entre campo estático y rotatorio muy elevado (7 veces la intensidad nominal). Estos valores no generan ningún daño al motor debido a que son transitorios, sin embargo, al ser repetitivos y con arranques sin periodos de descanso, pueden elevar progresivamente la temperatura del estator, comprometiendo la vida útil de los devanados hasta originar fallas por derretimiento del aislamiento.

2.4 Clasificación del motor de inducción

2.4.1 Motor monofásico de inducción con espiras en cortocircuito

Su arranque es directamente por sí mismo, debido al efecto producido por las espiras en corto circuito, su sistema divide los polos en dos partes desiguales. Tienen aplicación en electrodomésticos y pequeños aparatos, ya que al no contar con escobillas no requiere de mantenimiento y están casi exentos de averías.

2.4.2 Motor monofásico de inducción tipo fase partida

Para su arranque necesita de una fase de partida, el cual consiste en crear un campo magnético giratorio que mueva al inducido durante el periodo de arranque. El campo magnético se obtiene por capacidad o inductancia.

Para producir el campo giratorio por capacidad, se implementa dos arrollamientos de bobinas en serie (una de ellas debe contar con un condensador), el cual realiza el desfase provocando el arranque del motor.

2.4.3 Motor de jaula de ardilla

Son compactos y cuentan con un núcleo de hierro laminado. Para su arranque necesita de un campo rotatorio en lugar de un campo alterno. Sus conductores son de cobre y se encuentran soldados en las terminales del metal, cada conductor forma una espira con el conductor opuesto conectado por dos piezas circulares externos. Cuando este rotor se encuentra entre dos polos de campos electromagnéticos magnetizados por una corriente alterna, se induce una fem en las espiras de la jaula de ardilla, una corriente grande las recorre, produciendo un fuerte campo que contrarresta al producido por la corriente (ley de Lenz).

Cuando el campo produce un efecto rotatorio, el motor es clasificado como jaula de ardilla. El desplazamiento real máximo de intensidad del campo magnético desde un polo al siguiente atrae al rotor de jaula de ardilla con sus corrientes y campos inducidos, provocando un giro que genera el arranque del motor por sí mismo.

2.5 Máquinas rotativas

Este tipo de máquinas presentan vibraciones producidas por diversas causas. Su intensidad depende principalmente de la interacción entre las fuerzas dinámicas que ocurren dentro del equipo y su flexibilidad mecánica.

Diversos estudios realizados, han determinado que el desbalance es una causa fundamental de las vibraciones con un 40%, mientras que el 30% es producido por el desalineamiento.

2.6 Desbalance

Es la condición donde el eje de inercia del rotor no coincide con su eje de rotación, generando un giro no concéntrico, el cual crea fuerzas y/o movimientos dinámicos que producen vibraciones. La cantidad de desbalance se calcula como:

$$U = m \cdot r_0 \quad (1)$$

U = cantidad de desbalance [g·m]

m = masa desbalanceada [g]

r_0 = distancia de m al eje de rotación [mm]

La norma ISO 1925 clasifica el desbalance según la distribución de masa desbalanceadora en:

- Desbalanceo estático: cuando el eje principal de inercia es paralelo al eje de rotación.
- Desbalanceo de cupla: cuando el eje principal de inercia intercepta el eje de rotación en el centro de gravedad del rotor.
- Desbalance cuasi- estático: cuando el eje principal de inercia intercepta al eje de rotación en un punto distinto al centro de gravedad del rotor.
- Desbalance dinámico: cuando el eje principal de inercia no intercepta el eje de rotación.

El desbalanceamiento puede ser producido por diversas causas, entre las cuales se encuentra:

- Desgaste de partes rotativas de las máquinas.
- Montaje defectuoso de componentes.
- Falta de simetría en las partes rotativas de las máquinas, debidas a la fundición, forjado, maquinado, a carga o a dilaciones no homogéneas.
- Falta de homogeneidad causada por soldaduras.
- Variaciones en la estructura química y cristalina del material, causada por el vacío o tratamiento térmico.
- Variaciones en el tamaño de tornillos, tuercas y otros sujetadores.

En un equipo con rotor desbalanceado puede producir problemas, tales como:

- Desgaste excesivo y/o fallas prematuras.
- Aumento del nivel de ruido durante la operación del equipo.
- Desajuste de los sistemas de fijación.
- Falla por fatiga y aumento de consumo de energía.

Para balancear un rotor se requiere de instrumentación capaz de medir la vibración y de cálculos que permitan encontrar una relación directa entre la vibración, la magnitud y posición del desbalanceamiento del rotor. (Estupiñán P., San Martín, & Canales M., 2006)

2.7 Vibraciones mecánicas

La vibración de un sistema implica la transformación de su energía potencial en energía cinética y de ésta en energía potencial, en forma alterna. Cuando la frecuencia de la fuerza externa coincide con una de las frecuencias naturales del sistema, ocurre una condición conocida como resonancia, la cual genera oscilaciones peligrosamente grandes. Si no se pierde o disipa energía por fricción u otra resistencia durante la oscilación, la vibración se conoce como vibración no amortiguada. Sin embargo, si se pierde energía se llama vibración amortiguada. (S.S., 2012)

Las vibraciones producidas por las máquinas generalmente no tienen una frecuencia determinada, sino que son una mezcla de vibraciones de diferentes frecuencias.

En función a la frecuencia, las vibraciones se clasifican en:

- Muy baja frecuencia <1Hz: el movimiento de balanceo de trenes, barcos, aviones, etc.
- Baja frecuencia 1-20Hz: vibraciones originadas por carretillas elevadoras, tractores, vehículos de transporte urbano, máquinas excavadoras.
- Alta frecuencia 20-100Hz: máquinas neumáticas y rotativas, tales como, martillos picadores neumáticos, pulidoras, motosierras, lijadoras, etc. (Chapaman, 2012)

Todas las máquinas rotativas están sometidas a ciertos niveles de vibración, debido a sus imperfecciones constructivas, lo cual genera la aparición de desequilibrios, desalineaciones residuales, desgastes, fricciones y todo tipo de fuerzas no compensadas.

Debido al exceso de vibraciones en el sistema del rotor, se producen deformaciones y tensiones que originan una sobredimensión, provocando el aflojamiento de conexiones, carga excesiva del cojinete e incluso daños en el sistema. (Zhao, Ren, Deng, Yongfeng, & Fu, 2021)

Estas averías ocasionan la aparición de fuerzas, que pueden ser de origen eléctrico, mecánico, o proceso físico en el que está implicada la máquina. Todas ellas incrementan el nivel global de vibración o modifican las frecuencias de vibración ya existentes.

Las principales causas de vibración en una máquina eléctrica son:

- La respuesta del núcleo del estator a la fuerza atractiva desarrollada entre el rotor y estator.
- La reacción de los devanados terminales del estator a las fuerzas electromagnéticas sobre los conductores.
- El comportamiento dinámico del rotor.
- La respuesta de los rodamientos del eje a la vibración transmitida desde el rotor. (Villa Duque, Parra, & Ocampo, 2007)

La presencia de pulsaciones en la señal de vibración requiere de la implementación de metodologías y herramientas adecuadas para obtener una disminución de la vibración residual del rotor hasta límites permitidos. El análisis de vibraciones ha sido el primer método de

aplicación y el más extendido de todos los implementados en el mantenimiento predictivo. Sin embargo, cuenta con limitaciones para la detección de fallos de origen eléctrico. La mayoría de las anomalías presentes en las máquinas rotativas pueden ser diagnosticadas, al efectuar un correcto análisis de su espectro de vibración.

2.8 Detección de señales

Un elemento clave para la detección de cualquier falla es el acondicionamiento y procesamiento adecuados de la señal.

El procedimiento de diagnóstico más común se puede clasificar en tres clases:

- Basado en modelos: implementa el análisis teórico de la máquina asimétrica cuyo modelo se utiliza para predecir firmas de fallas.
- Basado en señales: busca las firmas de fallas conocidas en cantidades muestreadas de la máquina real, se controlan mediante un procesamiento de señal adecuado. Implementa el procesamiento para mejorar la relación señal/ruido para normalizar los datos, aislar la falla de otros fenómenos y disminuir la sensibilidad de operación.
- Basado en datos: no requiere ningún conocimiento de los parámetros y el modelo de la máquina, solo se basa en el procesamiento de señales y en las técnicas de agrupamiento.

(Bellini, Filippetti, Tassoni, & Capolino, 2008)

Estos procedimientos son base para la ejecución del análisis de la vibración, ya que la mayoría de los sistemas vibratorios prácticos son muy complejos, el análisis solo considera los detalles más importantes para predecir el comportamiento del sistema en condiciones de entrada específica.

Por lo que el análisis de un sistema vibratorio suele implicar el modelado matemático, la derivación de las ecuaciones rectoras, la solución de ecuación y la interpretación de los resultados.

2.9 Metodología para detección de fallas

El mantenimiento predictivo ha sido una filosofía ampliamente difundida en el sector productivo y en especial en equipos como las máquinas eléctricas. Debido a ello se han desarrollado diferentes técnicas de diagnóstico, como lo son:

- Análisis de vibraciones.
- Análisis espectral de corriente.
- Análisis del flujo axial de dispersión.
- Modelos de simulación del comportamiento de las fallas y la aplicación de redes neuronales artificiales para identificar las fallas entre espiras del estator. (Villa Duque, Parra, & Ocampo, 2007)

2.10 Tipos de análisis de vibración

2.10.1 Análisis espectral de vibraciones

Actualmente, se están realizando algunos intentos por definir las frecuencias características de vibración asociadas a algunos defectos de origen eléctrico. La teoría predice que los cambios en la corriente debido al deterioro eléctrico de los devanados en las máquinas rotativas alterarán las fuerzas magnéticas internas, las cuales causarán una modificación en las características vibratorias de la máquina.

La relación matemática que existe entre la corriente eléctrica que fluye a través de los devanados y las vibraciones mecánicas de la máquina, es representada mediante la teoría de la onda permeancia.

2.10.2 Análisis espectral de corriente

El análisis espectral de la corriente es un complemento para el diagnóstico de problemas eléctricos como: excentricidades en el entrehierro, corto circuito entre espiras, barras rotar en los rotores de los motores de inducción causadas por esfuerzos mecánicos, magnéticos o térmicos.

Para la detección y monitoreo de los cortocircuitos originados en los motores de inducción, se emplea el análisis espectral de corrientes, el cual expone los armónicos de fuerza

magnetomotriz producidos por las ranuras del rotor, el estator y la saturación de los materiales que componen dicha máquina.

2.10.3 Análisis espectral del flujo axial

El flujo axial de dispersión es la pérdida de flujo inducido por las corrientes circulantes en la máquina. Se presenta en todas las máquinas eléctricas y se produce debido a que las máquinas no pueden ser construidas con una simetría perfecta en los circuitos eléctrico y magnético debido a las tolerancias en la fabricación, anisotropía de los materiales y otros factores que conllevan al desequilibrio entre las corrientes del rotor y el estator.

El análisis espectral de flujo contempla el incremento de la simetría, como un efecto de corto circuito entre las espiras del bobinado, debido a una modificación del circuito eléctrico que ocasiona un cambio subsecuente en la densidad de flujo del entrehierro de la máquina. Considerando este fenómeno como un efecto general de cambio, se estima que una pequeña variación en la densidad del flujo magnético se verá reflejada inmediatamente en el flujo axial de dispersión, convirtiendo el método de diagnóstico viable para este tipo de fallas.

2.10.4 Impedancia de secuencia inversa

Este método posibilita el diagnóstico del cortocircuito producido en la bobina del estator de un motor de inducción, sin la necesidad de realizar un paro operacional de la máquina, esto mediante la implementación de un parámetro llamado “impedancia efectiva de secuencia inversa o negativa”.

2.10.5 Par electromagnético

Es uno de los nuevos y no convencionales métodos de diagnóstico. Busca evitar el paro y desmontaje de la máquina. Su metodología consiste en calcular el par (generado por una serie de armónicos en las fuerzas magnetomotrices del rotor y el estator) en el entrehierro de los voltaje y corrientes instantáneas, basadas en la teoría de vectores espaciales, particularmente en la aplicación de la transformada de Park. (Villa Duque, Parra, & Ocampo, 2007)

El desbalance se relaciona con algunas de las componentes del espectro de la intensidad de la corriente eléctrica que alimenta al motor de inducción.

2.11 Equipos implementados para el balanceo analógico

Analizador de espectro: es una herramienta capaz de representar las componentes espectrales de una determinada señal a partir de su transformada de Fourier. Esta representación se encuentra en el dominio de la frecuencia permitiendo visualizar parámetros de la señal que difícilmente podrían ser descubiertos trabajando en el dominio del tiempo con ayuda de un osciloscopio.

Estroboscopio: es un instrumento implementado para verificar la velocidad de giro de las máquinas y motores de diversas clases, sin efectuar acoplamientos eléctricos o mecánico, debido a que permite visualizar los objetos en rotación como si se encontraran parados o girando muy lentamente. Existen diversos tipos de lámparas, la diferencia radica en su longitud o diámetro, tipo de electrodo, composición y presión del gas cargado, todo ello con la finalidad de cubrir un amplio espectro de uso y aplicación.

Tacómetro: es un dispositivo que mide las revoluciones por minuto (RPM) del rotor de un motor o una turbina. Existen dos tipos de tacómetros: el tacómetro óptico, no requiere del contacto entre las partes. Sin embargo, el tacómetro de contacto necesita del contacto entre el instrumento y la pieza que rota.

2.12 Balanceo

El balanceo de rotores es un procedimiento que consiste en alterar la distribución de la masa del rotor, a través del cálculo de contrapesos que generan el equilibrio estático y dinámico requerido para la reducción de la vibración, cuyo origen es producido en las chumaceras o puntos de apoyo.

Según el tipo de desbalance que se vaya a corregir, es el método de balanceo que se debe implementar. (La Malfa & Ercoli, 2013)

Métodos clásicos para la ejecución de pruebas de equilibrio:

- El método de equilibrio modal: propuesto en 1959 por Bishop y Gladwell, posteriormente desarrollado por otros investigadores. Este método consiste en la identificación del desequilibrio del rotor, el cual es afectado por las propiedades de apoyo de los rodamientos, especialmente los altos coeficientes de amortiguación.
- El método de equilibrio del coeficiente de influencia: Propuesto en 1964 por Goodman y desarrollado por Lund y Tonnesen, demuestran que las propiedades de apoyo de los rodamientos no tienen un impacto en la identificación del desequilibrio, la posición inadecuada de los puntos de medición puede resultar en una matriz de coeficientes mal acondicionada y conducir al fracaso de este método. (Wang, y otros, 2017)

Asimismo, existen otros tipos de equilibrio sin ejecuciones de prueba, los cuales son un método viable en la práctica.

- Equilibrio sin excitación externa: Los experimentos muestran que el método es apropiado para rotores con rodamientos, pero no se puede aplicar a rotores con cojinetes de aceite. Esto se debe a que los coeficientes de rigidez y amortiguación de los cojinetes de lubricación varían a medida que cambia la velocidad de rotación.
- Equilibrio con excitación externa: este método implica el uso de un excitador de alta potencia, sin embargo, esto puede dañar o afectar a los rotores.

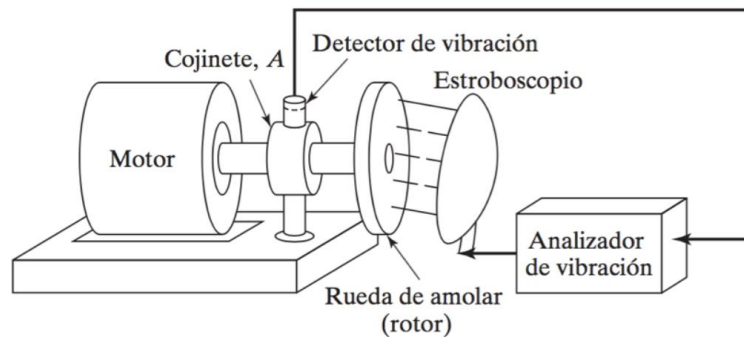
Una vez identificado el método aplicable para equilibrar el rotor, seleccionamos el tipo de balanceo a ejecutar en el equipo.

2.12.1 Balanceo estático (un plano)

Cuando las masas no balanceadas yacen en un plano. Se coloca la flecha del rotor sobre unos rieles, debido a que la masa de desbalance del rotor tiende a quedar en línea vertical hacia abajo, la masa de balanceo se hace por tanteo, hasta que se obtiene el estado de balance. La prueba estática puede detectar la fuerza resultante.

Una mejor opción de balanceo se realiza con un analizador de vibración. Es un proceso en el cual se monta una rueda de amolar (disco) en una flecha rotatoria que tiene un cojinete en A y la mueve un motor eléctrico a una velocidad angular ω .

Figura 1 Balanceo en un plano, analizador de vibraciones



Nota. Adaptado de *vibraciones mecánicas* (p. 62), por de Singiresu S. Rao

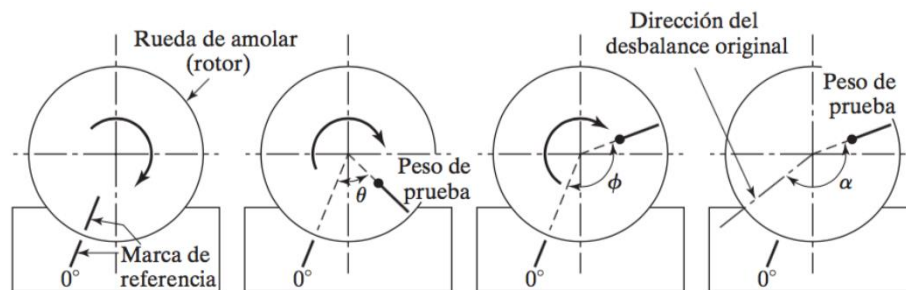
Al iniciar el procedimiento se deberá colocar unas marcas de referencia (marcas de fase), en ambos elementos (rotor y estator). Se posiciona el detector de vibración en el cojinete, mientras el analizador se ajusta a la frecuencia correspondiente a la velocidad angular de la rueda de amolar.

La señal de vibración (la amplitud de desplazamiento) producida por el desbalance se lee en el medidor del analizador de vibración. Posteriormente el analizador de vibración enciende una luz estroboscópica a la frecuencia de la rueda rotatoria.

Cuando el rotor gira a una velocidad ω , la marca de fase en el rotor aparece estacionaria bajo la luz estroboscópica, posicionada en un ángulo θ del estator, debido al retraso de fase en la respuesta.

Se registra el ángulo θ y la amplitud A_u (leídos en el analizador de vibración) provocados por el desbalance original.

Figura 2 Marcas de fase



Nota. Adaptado de *vibraciones mecánicas* (p. 653), por de Singiresu S. Rao

Posteriormente se detiene el rotor, para agregar un peso de prueba conocido W . Cuando el rotor gira a la velocidad ω , se captura la nueva posición angular de la marca de fase ϕ en el rotor y la amplitud de vibración A_{u+w} provocada por el desbalance combinado del rotor y el peso de prueba.

Seguidamente se construye un diagrama vectorial para determinar la magnitud y localización de la masa de corrección para balancear la rueda. El vector de diferencia $A_w = A_u + w - A_u$ representa el vector de desbalance debido al peso de prueba W . La magnitud de A_w se calcula aplicando la ley de cosenos, desarrollada en la ecuación (2)

$$A_w = [A_u^2 + A_{u+w}^2 - 2A_u A_{u+w} \cos(\phi - \theta)]^{1/2} \quad (2)$$

Debido a que la magnitud del peso de prueba W y su dirección con respecto al desbalance original se conocen, entonces el desbalance original puede estar a un ángulo α de

la posición del peso de prueba y éste se obtiene a partir de la ley de los cosenos, desglosado en la siguiente ecuación (3)

$$a = \cos^{-1} \left[\frac{A_u^2 + A_w^2 - A_{u+w}^2}{2A_u A_w} \right] \quad (3)$$

Una vez conocidas la localización y magnitud del desbalance original, se puede agregar un peso de corrección para balancear adecuadamente la rueda. (S.S., 2012)

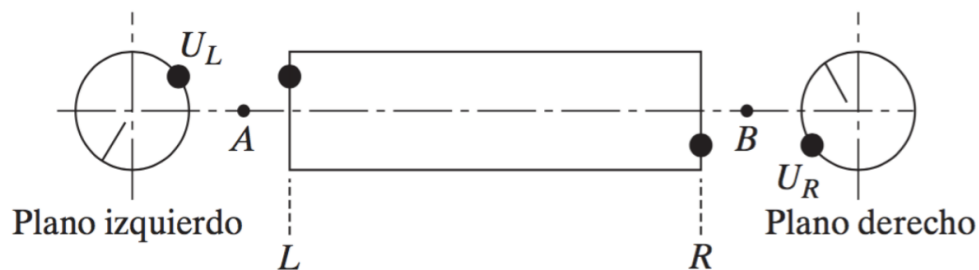
2.12.2 Balanceo dinámico (2 planos)

El desbalance se presenta en más de un plano, la resultante es una fuerza y un momento de balanceo. Debido a ello el procedimiento se torna complicado, ya que existen diversos métodos para su realización.

Durante la ejecución del balanceo dinámico, se implementa el método de coeficientes de influencia, el cual demanda la medición de la amplitud y fase de la vibración presente en los apoyos, mediante varias corridas de prueba, este procedimiento se efectúa mediante un analizador de vibración.

El desbalance total en el rotor es reemplazado por dos pesos desbalanceados U_L y U_R en los planos izquierdo y derecho, respectivamente, a la velocidad de operación ω del rotor.

Figura 3 Balanceo en dos planos



Nota. Adaptado de *vibraciones mecánicas* (p. 656), por de Singiresu S. Rao

La magnitud del vector de vibración AB se considera como la amplitud de vibración, mientras que la dirección del vector se considera como el valor negativo del ángulo observado bajo la luz estroboscópica respecto de la línea de referencia.

Primero agregamos un peso conocido W_L en el plano izquierdo a una posición angular conocida después medimos el desplazamiento y la fase de vibración en los dos cojinetes, mientras el rotor gira a una velocidad ω . Indicamos estas medidas de vibraciones como vectores.

Luego eliminamos W_L y agregamos un peso conocido W_R , en el plano derecho a una posición angular conocida y medimos las vibraciones resultantes mientras el rotor gira a una velocidad ω . (S.S., 2012)

2.13 Balanceo implementando la corriente residual

Cuando el rotor a balancear se encuentra posicionado en una estructura, donde otras máquinas trabajan a velocidades iguales o aproximadas a la del rotor y estas no se pueden detener por diferentes razones, la señal obtenida contiene los efectos combinados de las máquinas, manifestando una variación periódica de la amplitud.

Estas variaciones pueden ser descartadas mediante el procesamiento numérico de las armónicas (30 y 90 Hz). Las referencias bibliográficas señalan algunas formas de desarrollar el algoritmo de balanceo. Una posibilidad es calcular el peso de balanceo a colocar en el rotor acoplado al motor de inducción sin tomar en cuenta la fase entre los vectores de corriente. Otra forma calcula la fase entre las armónicas de la corriente residual para manejar mejor los datos y lograr mejores estimaciones del balanceo.

2.13.1 Análisis del Balanceo en un plano mediante la corriente residual

Para el balanceo en un plano utilizando las amplitudes sin fase de las armónicas de 30 o 90 Hz, se realizan tres corridas; una con el desbalance original (tal cual), otra con peso de prueba W_1 y finalmente con peso de prueba W_2 agregado. Con la finalidad de demostrar el cambio de respuesta en el rotor debido a su vibración.

Implementando los fasores del peso de balanceo, se obtiene mediante el desarrollo de un cociente, el cual contiene datos del coeficiente de influencia.

$$W_c = -\frac{N}{A} \quad (4)$$

Para obtener el coeficiente de influencia, implementamos la siguiente igualdad algebraica.

$$A = \frac{N_{21} - N}{W_1} = \frac{N_{22} - N}{W_2} \quad (5)$$

Reacomodando e introduciendo la relación de “r”, formamos la siguiente ecuación (6)

$$r = \frac{N_{22} - N}{N_{21} - N} \quad (6)$$

A continuación, se desarrollan los fasores de desbalance, generando las siguientes ecuaciones.

$$\begin{aligned} N &= R_0 e^{i\theta_0} + 0 \\ N_{21} &= R_{21} e^{i\theta_{21}} + 0 \\ N_{22} &= R_{22} e^{i\theta_{22}} + 0 \end{aligned} \quad (7)$$

Donde O es el offset presente en las armónicas. Sustituyendo los valores generados de la ecuación (7) en la (6), obtenemos.

$$r = \frac{R_{22} e^{i\theta_{22}} - R_0 e^{i\theta_0}}{R_{21} e^{i\theta_{21}} - R_0 e^{i\theta_0}} \quad (8)$$

Consecutivamente, reacomodamos la ecuación para producir la siguiente expresión.

$$R_{22} e^{i\theta_{22}} = r R_{21} e^{i\theta_{21}} + (1 - r) R_0 e^{i\theta_0} \quad (9)$$

El primer miembro representa un círculo de radio R_{22} con centro en el origen y el segundo miembro es un círculo de radio $r R_{21}$ y con un centro en.

$$C_{1x} + iC_{1y} = (1 - r) R_0 e^{i\theta_0} \quad (10)$$

Disponiendo de las ecuaciones anteriores, procedemos a elaborar las ecuaciones paramétricas.

$$x^2 + y^2 = R_{22}^2 \quad (11)$$

$$(x - C_{1x})^2 + (y - C_{1y})^2 = (rR_{21})^2 \quad (12)$$

Aplicando la adición miembro a miembro de la ecuación (12) en la ecuación (11), obtenemos una nueva ecuación, la cual representa una recta que va a intersecar al primer círculo.

$$-2C_{1x}x - 2C_{1y}y = (rR_{21})^2 - R_{22}^2 - C_{1x}^2 - C_{1y}^2 = b(2C_{1y}) \quad (13)$$

Posteriormente, despejamos la ecuación para obtener, la siguiente expresión.

$$y = b - \frac{C_{1x}}{C_{1y}}x \quad (14)$$

Sustituimos los valores obtenidos de la expresión “y” en la ecuación (11).

$$x^2 + \left(b - \frac{C_{1x}}{C_{1y}}x\right)^2 = R_{22}^2 \quad (15)$$

Se desarrolla la expresión algebraica, para obtener una cuadrática.

$$x^2 \left[1 + \frac{C_{1x}^2}{C_{1y}^2}\right] + x \left[-2b \frac{C_{1x}}{C_{1y}}\right] + [b^2 - R_{22}^2] = 0 \quad (16)$$

La resolución de la cuadrática (16) genera dos puntos de cruce sobre el primer círculo, (x_{11}, y_{11}) y (x_{12}, y_{12}) .

Para encontrar las intersecciones en el segundo círculo (ecuación (11)) se repite este proceso, despejando el círculo de R_{21} .

$$R_{21}e^{i\theta_{21}} = \frac{1}{r}R_{22}e^{i\theta_{22}} + \left(1 - \frac{1}{r}\right)R_0e^{i\theta_0} \quad (17)$$

El primer miembro presenta un círculo de radio R_{21} con centro en el origen y el segundo miembro es un círculo de radio $\frac{1}{r}R_{22}$ y con centro en.

$$C_{1x} + iC_{1y} = \left(1 - \frac{1}{r}\right)R_0e^{i\theta_0} \quad (18)$$

De esta manera se puede escribir las ecuaciones paramétricas.

$$x^2 + y^2 = R_{21}^2 \quad (19)$$

$$(x - C_{1x})^2 + (y - C_{1y})^2 = \left(\frac{1}{r} R_{22}\right)^2 \quad (20)$$

Sustrayendo miembro a miembro la ecuación (20) de la (19), generamos la siguiente expresión algebraica. Esta expresión constituye la recta que interseca al primer círculo.

$$-2C_{1x}x - 2C_{1y}y = \left(\frac{1}{r} R_{22}\right)^2 - R_{21}^2 - C_{1x}^2 - C_{1y}^2 = b(2C_{1y}) \quad (21)$$

A continuación, despejamos “y” de la ecuación (21) para generar la siguiente expresión.

$$y = b - \frac{C_{1x}}{C_{1y}}x \quad (22)$$

Posteriormente sustituimos los valores obtenidos de “y” en la ecuación (19), para crear una nueva expresión algebraica.

$$x^2 + \left(b - \frac{C_{1x}}{C_{1y}}x\right)^2 = R_{21}^2 \quad (23)$$

Consecutivamente, se desarrolla la ecuación que genera una ecuación cuadrática.

$$x^2 \left[1 + \frac{C_{1x}^2}{C_{1y}^2}\right] + x \left[-2b \frac{C_{1x}}{C_{1y}}\right] + [b^2 - R_{21}^2] = 0 \quad (24)$$

Finalmente se concluye la solución de la cuadrática, produciendo dos puntos de cruce sobre el círculo de R_{21} , digamos (x_1, y_1) y (x_2, y_2) .

Contando con la solución de la ecuación (8), procedemos a calcular los coeficientes de influencia a partir de las siguientes expresiones algebraicas. En donde (x_0, y_0) se obtiene de la expresión $R_0 e^{i\theta_0}$ al iterar valores del ángulo θ_0 .

$$\begin{aligned} (N_{21} - N)_1 &= (x_1 - x_0) + i(y_1 - y_0) \\ (N_{21} - N)_2 &= (x_2 - x_0) + i(y_2 - y_0) \\ (N_{22} - N)_1 &= (x_{11} - x_0) + i(y_{11} - y_0) \\ (N_{22} - N)_2 &= (x_{12} - x_0) + i(y_{12} - y_0) \end{aligned} \quad (25)$$

Para calcular el peso de balanceo se requiere conocer N y ésta depende de igual manera del offset O .

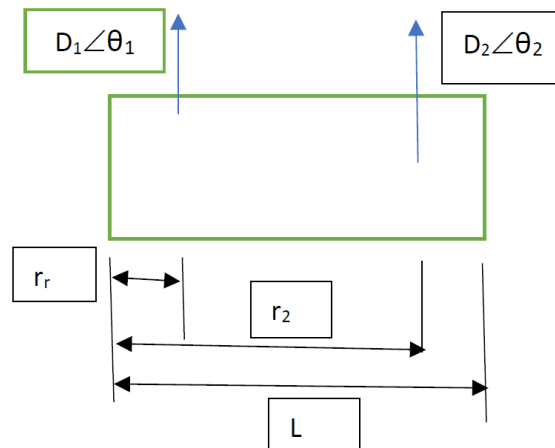
2.13.2 Análisis de balanceo en dos planos a través de la corriente residual

Para el método de balanceo utilizando la corriente residual, no se aplican dos puntos de medición a diferencia del método tradicional.

El innovador procedimiento consiste en aplicar la técnica de balanceo en un plano poniendo dos pesos iguales y orientados en la misma dirección, uno en cada plano de balanceo. A continuación, se realiza el balanceo dinámico que calcula un par de pesos iguales, pero en sentido contrario que producen un par centrífugo. Al final se tienen dos pesos en cada plano de balanceo, cuya resultante coincide con lo que se obtendría por el método tradicional.

Para un rotor rígido con dos pesos de desbalance: $D_1 \angle \theta_1$ en la posición axial r_1 y $D_2 \angle \theta_2$ en la posición axial r_2 como se muestra en la **Figura 4**.

Figura 4 Rotor rígido con dos pesos de desbalance



Suponer que los pesos de balanceo en los extremos del rotor (planos de balanceo) son:

- $W_1 \angle \theta_1, W_{21} \angle \theta_1$ componentes relacionados con $D_1 \angle \theta_1$
- $W_{12} \angle \theta_2, W_{22} \angle \theta_2$ componentes relacionados con $D_2 \angle \theta_2$

Se calcula $W_1 \angle \theta_1, W_{21} \angle \theta_1$ que están en el mismo plano de $D_1 \angle \theta_1$ como si fueran las reacciones de una viga (suponiendo los planos de balanceo como si fueran los apoyos):

$$\sum M_2 = 0 = (L - r_1)D_1\angle\theta_1 + LW_{11}\angle\theta_1 \quad (26)$$

$$W_{11}\angle\theta_1 = -\frac{(L - r_1)}{L}D_1\angle\theta_1 = \left[\left(\frac{r_1}{L}\right) - 1\right]D_1\angle\theta_1 \quad (27)$$

$$\begin{aligned} \sum F = 0 &= D_1\angle\theta_1 + W_{11}\angle\theta_1 + W_{21}\angle\theta_1 - W_{21}\angle\theta_1 \\ &= D_1\angle\theta_1 + \left[\left(\frac{r_1}{L}\right) - 1\right]D_1\angle\theta_1 = \left(\frac{r_1}{L}\right)D_1\angle\theta_1 \end{aligned} \quad (28)$$

$$W_{21}\angle\theta_1 = -\left(\frac{r_1}{L}\right)D_1\angle\theta_1 \quad (29)$$

Repitiendo el proceso para el plano de $D_2\angle\theta_2$ se tiene.

$$\sum M_2 = 0 = (L - r_2)D_2\angle\theta_2 + LW_{12}\angle\theta_2 \quad (30)$$

$$W_{12}\angle\theta_2 = -\frac{(L - r_2)}{L}D_2\angle\theta_2 = \left[\left(\frac{r_2}{L}\right) - 1\right]D_2\angle\theta_2 \quad (31)$$

$$\begin{aligned} \sum F = 0 &= D_2\angle\theta_2 + W_{12}\angle\theta_2 + W_{22}\angle\theta_2 - W_{22}\angle\theta_2 \\ &= D_2\angle\theta_2 + \left[\left(\frac{r_2}{L}\right) - 1\right]D_2\angle\theta_2 = \left(\frac{r_2}{L}\right)D_2\angle\theta_2 \end{aligned} \quad (32)$$

$$W_{22}\angle\theta_2 = -\left(\frac{r_2}{L}\right)D_2\angle\theta_2 \quad (33)$$

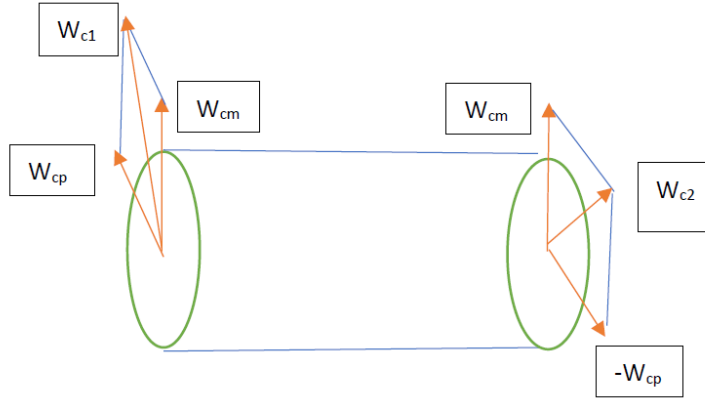
Superponiendo los dos casos se obtienen los pesos de balanceo.

$$W_{c1} = W_{11}\angle\theta_1 + W_{12}\angle\theta_2 = \left[\left(\frac{r_1}{L}\right) - 1\right]D_1\angle\theta_1 + \left[\left(\frac{r_2}{L}\right) - 1\right]D_2\angle\theta_2 \quad (34)$$

$$W_{c2} = W_{21}\angle\theta_1 + W_{22}\angle\theta_2 = \left(\frac{r_1}{L}\right)D_1\angle\theta_1 - \left(\frac{r_2}{L}\right)D_2\angle\theta_2 \quad (35)$$

Consideramos la situación en que se desea descomponer los pesos de balanceo como un par de pesos iguales (misma magnitud y dirección) trabajando en los dos planos de balanceo y un momento producido por un par de pesos (misma magnitud y dirección contraria), como se muestra en la **Figura 5**

Figura 5 Descomposición de los pesos de balanceo



Las condiciones de esta descomposición son.

$$W_{c1} = W_{cm} + W_{cp} \quad (36)$$

$$W_{c2} = W_{cm} - W_{cp} \quad (37)$$

Sumando miembro a miembro las ecuaciones (36) y (37) se obtiene.

$$W_{c1} + W_{c2} = 2W_{cm} \quad (38)$$

Por lo tanto.

$$W_{cm} = \frac{1}{2}(W_{c1} + W_{c2}) \quad (39)$$

Combinando la ecuación (36) y (39), se obtiene.

$$W_p = W_{c1} - W_{cm} = \frac{1}{2}(W_{c1} - W_{c2}) \quad (40)$$

Sustituyendo las ecuaciones (34) y (35) en la (39) se crea.

$$W_{cm} = \frac{1}{2} \left[\left[\left(\frac{r_1}{L} \right) - 1 \right] D_1 \angle \theta_1 + \left[\left(\frac{r_2}{L} \right) - 1 \right] D_2 \angle \theta_2 - \left(\frac{r_1}{L} \right) D_1 \angle \theta_1 - \left(\frac{r_2}{L} \right) D_2 \angle \theta_2 \right] \quad (41)$$

Simplificando.

$$W_{cm} = \frac{1}{2} [-D_1 \angle \theta_1 - D_2 \angle \theta_2] \quad (42)$$

Sustituyendo las ecuaciones (34) y (35) en la (40), genera.

$$W_{cp} = \frac{1}{2} \left[\left[\left(\frac{r_1}{L} \right) - 1 \right] D_1 \angle \theta_1 + \left[\left(\frac{r_2}{L} \right) - 1 \right] D_2 \angle \theta_2 + \left(\frac{r_1}{L} \right) D_1 \angle \theta_1 + \left(\frac{r_2}{L} \right) D_2 \angle \theta_2 \right] \quad (43)$$

Simplificando.

$$W_{cp} = \frac{1}{2} \left[\left[\left(\frac{2r_1}{L} - 1 \right) \right] D_1 \angle \theta_1 + \left[\left(\frac{2r_2}{L} - 1 \right) \right] D_2 \angle \theta_2 \right] \quad (44)$$

El procedimiento de balanceo es el siguiente:

1. Con peso de prueba W_{pm} (par de pesos iguales en magnitud y dirección) en los dos planos de balanceo se hacen dos corridas de prueba, es decir, una en la posición (de los pesos W_{pm}) de 0° y otra con la posición de 90° . Con estos datos se corre el programa de balanceo en un plano y se obtienen los pesos W_{cm} .
2. Colocamos los pesos W_{cm} , se ponen los pesos de prueba W_{par} (par de pesos iguales en magnitud y en dirección opuesta) en los dos planos de balanceo para hacer dos corridas de prueba, por decir una con W_{par1} en 0° (W_{par2} en 180°) y otra corrida con W_{par1} en 90° (W_{par2} en 270°). Con estas corridas, se vuelve a ejecutar nuevamente el algoritmo de balanceo en un plano para obtener los pesos W_{cp} .

2.14 Determinación del offset de las armónicas de la corriente eléctrica

El offset obtenido de las mediciones de la corriente eléctrica representa el valor que tendría la medición si el rotor estuviera balanceado, es decir, es independiente del peso de desbalance del rotor.

Este offset podría medirse, sin embargo, se supone un valor al correrlo, con el propósito de producir círculos base de las configuraciones promedio, mediante una función presente en el algoritmo, encargado de encontrar el factor de escalación del offset, el cual varía directamente proporcional a los radios de estos círculos, determinando un punto de intersección común.

2.15 Condición de intersección de los círculos en un punto común.

Una vez asignado el valor (rango de evaluación) del offset, el algoritmo determinará la condición del punto de intersección de los círculos, mediante la siguiente ecuación

$$C_1 + fR_1e^{i\theta_1} = C_2 + fR_2e^{i\theta_2} = C_3 + fR_3e^{i\theta_3} \quad (45)$$

Donde los centros de los círculos de pesos de balanceo son C_1 , C_2 y C_3 , los radios de los círculos son R_1 , R_2 y R_3 y el factor de escalación del offset es f . También los valores de θ_1 , θ_2 y θ_3 son las orientaciones de los radios.

Desarrollando las exponenciales de exponente imaginario por la relación de Euler

$$e^{i\theta} = \cos\theta + i \operatorname{sen}\theta \quad (46)$$

Se obtienen las siguientes ecuaciones

$$C_{1x} + f R_1 \cos\theta_1 = C_{2x} + f R_2 \cos\theta_2 \quad (47)$$

$$C_{1y} + f R_1 \operatorname{sen}\theta_1 = C_{2y} + f R_2 \operatorname{sen}\theta_2 \quad (48)$$

$$C_{1x} + f R_1 \cos\theta_1 = C_{3x} + f R_3 \cos\theta_3 \quad (49)$$

$$C_{1y} + f R_1 \operatorname{sen}\theta_1 = C_{3y} + f R_3 \operatorname{sen}\theta_3 \quad (50)$$

De las ecuaciones (47) y (48) se obtienen:

$$f R_2 \cos\theta_2 = C_{1x} - C_{2x} + f R_1 \cos\theta_1 \quad (51)$$

$$f R_2 \operatorname{sen}\theta_2 = C_{1y} - C_{2y} + f R_1 \operatorname{sen}\theta_1 \quad (52)$$

Elevando al cuadrado ambas ecuaciones y sumando miembro a miembro resulta.

$$\begin{aligned} & f^2 R_2^2 (\cos^2\theta_2 + \operatorname{sen}^2\theta_2) \\ &= (C_{1x} - C_{2x})^2 + (C_{1y} - C_{2y})^2 \\ &+ f^2 R_1^2 (\cos^2\theta_1 + \operatorname{sen}^2\theta_1) + 2(C_{1x} - C_{2x})f R_1 \cos\theta_1 \\ &+ 2(C_{1y} - C_{2y})f R_1 \operatorname{sen}\theta_1 \end{aligned} \quad (53)$$

Simplificando por la identidad trigonométrica $\cos^2\theta_2 + \operatorname{sen}^2\theta_2 = 1$ se obtiene:

$$f^2 R_2^2 = (C_{1x} - C_{2x})^2 + (C_{1y} - C_{2y})^2 + f^2 R_1^2 + 2(C_{1x} - C_{2x})f R_1 \cos\theta_1 + 2(C_{1y} - C_{2y})f R_1 \sin\theta_1 \quad (54)$$

Similarmente de las ecuaciones (67) y (68) se obtiene.

$$f^2 R_3^2 = (C_{1x} - C_{3x})^2 + (C_{1y} - C_{3y})^2 + f^2 R_1^2 + 2(C_{1x} - C_{3x})f R_1 \cos\theta_1 + 2(C_{1y} - C_{3y})f R_1 \sin\theta_1 \quad (55)$$

De la ecuación (54) se forma la cuadrática en f .

$$f^2 [R_1^2 - R_2^2] + f [2(C_{1x} - C_{2x}) R_1 \cos\theta_1 + 2(C_{1y} - C_{2y}) R_1 \sin\theta_1] + [(C_{1x} - C_{2x})^2 + (C_{1y} - C_{2y})^2] = 0 \quad (56)$$

De la ecuación (55) se forma la cuadrática en f .

$$f^2 [R_1^2 - R_3^2] + f [2(C_{1x} - C_{3x}) R_1 \cos\theta_1 + 2(C_{1y} - C_{3y}) R_1 \sin\theta_1] + [(C_{1x} - C_{3x})^2 + (C_{1y} - C_{3y})^2] = 0 \quad (57)$$

La solución de la cuadrática (56) es de la forma:

$$\frac{-b_1 \pm \sqrt{b_1^2 - 4a_1c_1}}{2a_1} \quad (58)$$

La solución de la cuadrática (57) es de la forma

$$\frac{-b_2 \pm \sqrt{b_2^2 - 4a_2c_2}}{2a_2} \quad (59)$$

Igualando estas dos soluciones generamos.

$$\frac{-b_1 \pm \sqrt{b_1^2 - 4a_1c_1}}{2a_1} = \frac{-b_2 \pm \sqrt{b_2^2 - 4a_2c_2}}{2a_2} \quad (60)$$

La ecuación (60) tiene una sola incógnita que es el ángulo θ_1 .

Definiendo la variable $x = \sin\theta_1$ entonces se tiene.

$$\cos\theta_1 = \pm\sqrt{1 - x^2} \quad (61)$$

Desarrollando el Álgebra se llega a la ecuación bi-cuadrática siguiente:

$$Ax^4 + Bx^2 + C = 0 \quad (62)$$

La cual tiene cuatro raíces, aunque solamente se consideran los valores reales.

2.16 Equipo e instrumentación implementado para el balanceo mediante la corriente residual

2.16.1 Adquisición de datos

La adquisición de datos (DAQ) es el proceso de medir con una PC un fenómeno eléctrico o físico como voltaje, corriente, temperatura, presión o sonido. Un sistema DAQ (**Figura 6**) consiste en sensores, hardware de medidas DAQ y una PC con software programable. Comparados con los sistemas de medidas tradicionales, los sistemas DAQ basados en PC aprovechan la potencia del procesamiento, la productividad, la visualización y las habilidades de conectividad de las PCs estándares en la industria proporcionando una solución de medidas más potente, flexible y rentable. (NI, 2020)

Figura 6 Partes de un sistema DQA



Nota: adaptado de la web <http://www.ni.com/data-acquisition/what-is/esa/>

2.16.2 Software controlador

El software controlador ofrece al software de aplicación la habilidad de interactuar con un dispositivo DAQ. Simplifica la comunicación con el dispositivo DAQ al abstraer comandos de hardware de bajo nivel y programación a nivel de registro. Generalmente, el software controlador DAQ expone una interfaz de programación de aplicaciones (API) que es usada en un entorno de programación para construir software de aplicación.

2.16.3 Software de aplicación

El software de aplicación facilita la interacción entre la PC y el usuario para adquirir, analizar y presentar datos de medidas. Puede ser una aplicación preconstruida con funcionalidad predefinida o un entorno de programación para construir aplicaciones con funcionalidad personalizada. Las aplicaciones personalizadas generalmente son usadas para automatizar múltiples funciones de un dispositivo DAQ, realizar algoritmos de procesamiento de señales y mostrar interfaces de usuario personalizadas.

2.17 Implementación de instrumentación virtual

La implementación de un instrumento virtual para el balanceo de un motor considera las señales producidas de los componentes a través de la corriente residual. Esto mediante un sistema de medición que detecta la corriente fundamental consumida por el motor, así como las armónicas relacionadas con los problemas electromagnéticos y mecánicos.

Debido a que no requiere de ningún conocimiento de los parámetros o el modelo de la máquina, solo se basa en el procesamiento de señales y en las técnicas de agrupamiento.

Por lo que se decidió desarrollar una interfaz gráfica en el software LabVIEW 2018®, el cual cuenta con pantallas amigables para que el usuario realice la lectura y captura de información. Esto mediante las pinzas de gancho Fluke, encargadas de medir las variaciones del voltaje presente en las líneas de alimentación del motor, mientras que la tarjeta de adquisición de datos (NI 9233) gestiona el enlace de los instrumentos de medición a la interfaz.

La interfaz cuenta con las siguientes secciones:

- Calibración del instrumento.
- Vista de señales capturadas.
- Configuración del filtro.
- Vista de señales filtradas.
- Gráficas del espectro de frecuencia antes y después del filtrado.
- Vista del diagrama fasorial de armónicas de corriente seleccionadas.
- Especificación de las condiciones de guardado de información.

CAPITULO 3. Metodología

Con la finalidad de validar la metodología para la ejecución del balanceo dinámico de un rotor acoplado a un motor de inducción mediante armónicas de 30 y 90 Hz de la corriente residual. Se deberá efectuar el balanceo analógico, con el objetivo de proporcionar un equipo calibrado para la ejecución del método a desarrollar.

3.1 Calibración del motor

Este procedimiento se realiza con la finalidad de poseer un estándar de medición, el cual determinará la relación entre el valor mostrado por los instrumentos de medición y el valor verdadero (presente en el equipo). Generalmente, la calibración se realiza dos veces: antes y después de usar el instrumento de medición.

3.1.1 Equipo y/o herramienta para el balanceo analógico

Para la ejecución del balanceo analógico se deberá contar con los siguientes equipos y herramientas.

- Lámpara estroboscópica.
- Espectrómetro.
- Báscula de bolsillo.
- APP balanceo de rotores.
- Plastilina y/o tornillería.

3.1.2 Evaluación de vibración

- I. El motor de 0.75 HP, debe ser conectado de la siguiente manera: el cable de tierra marcado con cinta negra deberá ser acoplado a la tierra física que se encuentra posicionada en la parte trasera del equipo. Seguidamente acoplaremos cada una de las líneas de alimentación marcadas (1,2,3) a un conector saliente de una caja, que funge como interruptor para el encendido (botón verde) y apagado (botón rojo) del motor.

- II. Verificamos que el equipo cuente con la administración de energía necesaria para su funcionamiento continuo y sin interrupciones.
- III. Posteriormente apagamos el motor, para continuar con la colocación de los equipos de medición.
- IV. Se posicionan dos imanes del lado derecho del motor, conectados al espectrómetro, los cuales son encargados de indicar la magnitud de falla que presenta el rotor.
- V. Posteriormente se enciende el motor (cuenta con dos platos: plato interior-plano 1, plato exterior-plano 2).
- VI. Seguidamente se activa la lámpara estroboscópica, posicionándola frontalmente a los planos del motor, la cual genera una marca en cualquier parte del rotor (marca de ángulos) indicando la localización del desbalance.
- VII. Anotamos la magnitud y ángulos obtenidos por los dispositivos implementados.

Tabla 1 Valor inicial de la medición

Planos	Magnitud	Ángulo
Plano 1	0.038 g	225 °
Plano 2	0.093 g	250 °

- VIII. Con el soporte de la APP “balanceo de rotores”, ingresamos los datos (magnitud-espectrómetro y ángulo-lámpara estroboscópica), para generar el peso agregado y los ángulos de posición para ambos planos.
- IX. Con la implementación de la báscula realizamos la medición de los pesos agregados, y los colocamos en los ángulos marcados.

Tabla 2 Datos para realizar el primer ajuste

Planos	Magnitud	Ángulo
Plano 1	6 g	270°
Plano 2	5.5 g	90°

- X. Al realizar la segunda medición de vibración en el motor, generó los siguientes datos.

Tabla 3 Resultados del primer ajuste

Planos	Magnitud	Ángulo
Plano 1	0.026 g	220°
Plano 2	0.13 g	270°

- XI. Debido a que los resultados generados indica una alteración mayor a la inicial, se procede a ejecutar los pasos VIII y IX reiteradamente.

Tabla 4 Datos para realizar el segundo ajuste

Planos	Magnitud	Ángulo
Plano 1	7 g	-249.7°= 110.3°
Plano 2	5.57 g	-38.9°= 321°

- XII. Nuevamente se observa un comportamiento errante (aumento de la magnitud y el ángulo), lo cual implica un desbalance persistente solo en el primer plano, por lo cual se procede a ingresar los datos iniciales (retirando las cargas colocadas para su ajuste), para recalcular la magnitud y el ángulo con la APP, especificando la alteración en un solo plano.

- XIII. Se coloca el peso en el ángulo indicado sobre el primer plano.

Tabla 5 Valor inicial en un plano

Planos	Magnitud	Ángulo
Plano 1	1.5 g	232°
Plano 2	N/A	N/A

- XIV. Finalmente, las corridas y correcciones reflejaron un balanceo recomendable en ambos planos, con una magnitud mínima de cero, el cual no genera vibración. Proporcionando un equipo óptimo para medición y ejecución del experimento.

Tabla 6 Resultado final

Planos	Magnitud	Ángulo
Plano 1	0	N/A
Plano 2	0	N/A

3.2 Balanceo dinámico implementando armónicas de 30 y 90 Hz de la corriente residual

El método propone una nueva forma para la detección de fallas en el desbalance recurrente en los motores de inducción, mediante la aplicación de armónicas de 30 y 90 Hz presentes en la corriente residual del equipo.

Esto mediante la colocación de pesos de pruebas en distintos ángulos, marcados por perforaciones separadas a 15° hasta formar un giro de 360° en los planos posicionados de la siguiente manera:

Figura 7 Marcas de perforaciones en los platos del rotor

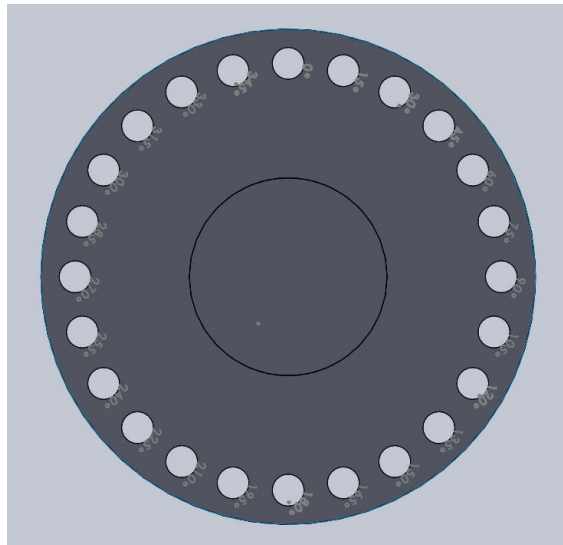
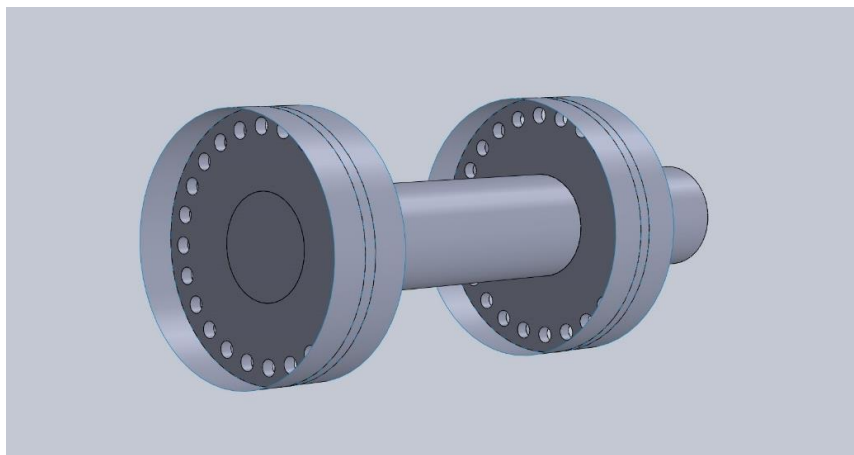


Figura 8 Platos y flecha del rotor



La colocación del peso de prueba alrededor del plano tiene la finalidad de medir las variaciones presentes en cada posición angular, además de proporcionar al operador una amplia gama para la selección de ángulos de pruebas.

3.2.1 Determinación del peso de balanceo aplicando dos pesos de prueba

Este método de balanceo dinámico utiliza las amplitudes de las armónicas de 30 o 90 Hz, de la corriente de alimentación de un motor de inducción acoplado a un rotor desbalanceado. Esta técnica requiere de tres corridas del rotor de prueba.

Tabla 7 Mediciones de amplitud de armónica a 30hz o a 90hz

CORRIDA	AMPLITUD FASE 1	AMPLITUD FASE 2	AMPLITUD FASE 3
Rotor original	R_{01}	R_{02}	R_{03}
Rotor con W_{p1}	R_{11}	R_{12}	R_{13}
Rotor con W_{p2}	R_{21}	R_{22}	R_{23}

En la **Tabla 7**, las mediciones de las amplitudes de las armónicas están dadas en miliamperios y pueden capturarse con el filtro pasa-banda centrado en 30 o 90 Hz.

Cuando se miden vibraciones en los apoyos del rotor se toman amplitudes y ángulos de fase, sin embargo, en las mediciones de corriente la fase presenta numerosas variaciones, mientras que la amplitud muestra una variación menor. Por otro lado, las armónicas relacionadas al desbalance presentan un offset que se supone es el mismo para las tres fases. A continuación, mostraremos las ecuaciones implementadas para las armónicas de fase 1.

$$N = R_{01}e^{i\theta_{01}} + O \quad (63)$$

$$N_{11} = R_{11}e^{i\theta_{11}} + O \quad (64)$$

$$N_{21} = R_{21}e^{i\theta_{21}} + O \quad (65)$$

Donde las N , N_{21} y N_{22} son fasores, lo mismo que el offset O . Los ángulos de fase θ_{01} , θ_{11} y θ_{21} no se miden, sino que se determinan en el proceso numérico.

El coeficiente de influencia para la fase 1 tiene dos expresiones:

$$A = \frac{N_{11} - N}{W_{p1}} = \frac{N_{21} - N}{W_{p2}} \quad (66)$$

Se define el fasor.

$$r = \frac{W_{p2}}{W_{p1}} = r_r + i r_i \quad (67)$$

La ecuación (66) se puede escribir como.

$$N_{21} - N = r(N_{11} - N) \quad (68)$$

Sustituyendo las ecuaciones (63),(64) y (65) en la (68) obtiene.

$$R_{21}e^{i\theta_{21}} - R_{01}e^{i\theta_{01}} = r(R_{11}e^{i\theta_{11}} - R_{01}e^{i\theta_{01}}) \quad (69)$$

Se reescribe la ecuación (69)

$$R_{21}e^{i\theta_{21}} = rR_{11}e^{i\theta_{11}} + (1 - r)R_{01}e^{i\theta_{01}} \quad (70)$$

En el proceso numérico siguiente, se iterar el ángulo θ_{01} de 0° a 359° para generar un círculo de la fase 1. Entonces se supone conocido el dato y se introduce el fasor.

$$z_{01} = (1 - r)R_{01}e^{i\theta_{01}} = z_{01r} + i z_{01i} \quad (71)$$

Descomponiendo la ecuación (70) en su parte real e imaginaria se tiene:

$$R_{21} \cos \theta_{21} = r_r R_{11} \cos \theta_{11} - r_i R_{11} \sin \theta_{11} + z_{01r} \quad (72)$$

$$R_{21} \sin \theta_{21} = r_i R_{11} \cos \theta_{11} + r_r R_{11} \sin \theta_{11} + z_{01i} \quad (73)$$

Elevando al cuadrado las ecuaciones(72) y (73), además de sumar miembro a miembro se obtiene:

$$\begin{aligned}
R_{21}^2(\cos \theta_{21}^2 + \sen \theta_{21}^2) &= (r_r R_{11})^2(\cos \theta_{11}^2 + \sen \theta_{11}^2) \\
&+ (r_i R_{11})^2(\cos \theta_{11}^2 + \sen \theta_{11}^2) + z_{01r}^2 + z_{01i}^2 \\
&+ 2z_{01r}(r_r R_{11} \cos \theta_{11} - r_i R_{11} \sen \theta_{11}) \\
&+ 2z_{01i}(r_i R_{11} \cos \theta_{11} + r_r R_{11} \sen \theta_{11})
\end{aligned} \tag{74}$$

Simplificando la expresión.

$$\begin{aligned}
R_{21}^2 &= (r_r R_{11})^2 + (r_i R_{11})^2 + z_{01r}^2 + z_{01i}^2 \\
&+ 2z_{01r}(r_r R_{11} \cos \theta_{11} - r_i R_{11} \sen \theta_{11}) \\
&+ z_{01i}(r_i R_{11} \cos \theta_{11} + r_r R_{11} \sen \theta_{11})
\end{aligned} \tag{75}$$

Reacomodando términos resulta.

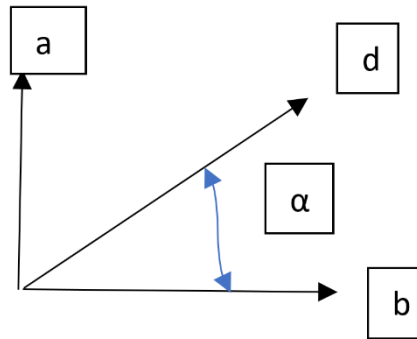
$$\begin{aligned}
R_{21}^2 - R_{11}^2(r_r^2 + r_i^2) - z_{01r}^2 - z_{01i}^2 &= 2R_{11} \cos \theta_{11} (r_r z_{01r} + r_i z_{01i}) \\
&+ 2R_{11} \sen \theta_{11} (r_r z_{01i} - r_i z_{01r})
\end{aligned} \tag{76}$$

Dando origen a la ecuación.

$$a \cos \theta_{11} + b \sen \theta_{11} = c \tag{77}$$

La suma de armónicas se puede representar mediante vectores que giran en la siguiente forma.

Figura 9 Armónicas representadas por vectores



$$d = \sqrt{a^2 + b^2} \tag{78}$$

$$\tan \alpha = \frac{a}{b} \quad (79)$$

Entonces la ecuación (77) se convierte en:

$$d \operatorname{sen}(\theta_{11} + \alpha) = c \quad (80)$$

Por lo tanto.

$$\operatorname{sen}(\theta_{11} + \alpha) = \frac{c}{d} \quad (81)$$

Teniendo θ_{11} de la ecuación (81) se puede obtener θ_{21} de las ecuaciones (72) y (73). Después se calcula el coeficiente de influencia de la ecuación (67) y luego el peso de balanceo de la siguiente ecuación.

$$W_c = -\frac{N}{A} \quad (82)$$

En realidad, el cálculo de N requiere conocer el offset O, por lo que este valor se supone o se itera. Mientras que el ángulo θ_{01} se itera de 0° a 359° con la finalidad de calcular cada valor del ángulo en peso de balanceo, generando un círculo para la fase 1. Al graficar los otros dos círculos de las fases 2 y 3 crea intersecciones comunes para el peso de balanceo deseado.

3.3 Selección de peso y ángulos de prueba para el balanceo en un plano

Los pesos de prueba nos otorgan la seguridad de un correcto balanceo de rotor, ya que, al tener conocimiento del peso implementado para crear el desbalance, podremos comprobar el funcionamiento del algoritmo, el cual, obtiene proximidades en los puntos de diferentes posiciones de peso de desbalance, debido a que los coeficientes de influencia no varían con el peso.

3.3.1 Peso de prueba 6.7g

Figura 10 Medición del peso de prueba, 6.7g



Selección de ángulos de medición.

$$PP1 = 6.7g \angle 0^\circ \quad (83)$$

$$PP2 = 6.7g \angle 90^\circ \quad (84)$$

$$PP3 = 6.7g \angle 180^\circ \quad (85)$$

Se realizan las combinaciones de las ecuaciones (83), (84) y (85).

$$6.7 \angle 90^\circ - 6.7 \angle 0^\circ = 9.4752 \angle 135^\circ \approx 9.5 \angle 135^\circ \quad (86)$$

$$6.7 \angle 180^\circ - 6.7 \angle 0^\circ = 13.4 \angle 180^\circ \quad (87)$$

Cálculo de w_c .

$$w_c = 6.7 \angle 0^\circ + 180^\circ = 6.7 \angle 180^\circ \quad (88)$$

Cálculo de los puntos de intersección.

$$6.7 \cos(180^\circ) = -6.7 \quad (89)$$

$$6.7 \sin(180^\circ) = 0 \quad (90)$$

Al obtener los resultados de las ecuaciones (89) y (90), convertiremos los números complejos.

$$(-6.7, 0i) = 6.7 \angle 180^\circ \quad (91)$$

3.3.2 Peso de prueba 8.2g

Figura 11 Medición de peso de prueba, 8.2g



Selección de ángulos de medición.

$$PP1 = 8.2 \text{ g} \angle 90^\circ \quad (92)$$

$$PP2 = 8.2 \text{ g} \angle 180^\circ \quad (93)$$

$$PP3 = 8.2 \text{ g} \angle 270^\circ \quad (94)$$

Realizar la combinación de las ecuaciones (92), (93) y (94).

$$8.2 \angle 180^\circ - 8.2 \angle 90^\circ = 11.596 \angle -135^\circ \approx 11.6 \angle 225^\circ \quad (95)$$

$$8.2 \angle 270^\circ - 8.2 \angle 90^\circ = 16.4 \angle -90^\circ \approx 16.4 \angle 270^\circ \quad (96)$$

Cálculo de W_c .

$$W_c = 8.2 \text{ g} \angle 90^\circ + 180^\circ = 8.2 \text{ g} \angle 270^\circ \quad (97)$$

Cálculo de los puntos de intersección.

$$8.2 \cos(270^\circ) = 0 \quad (98)$$

$$8.2 \sin(270^\circ) = -8.2 \quad (99)$$

Una vez obtenido los resultados de las ecuaciones (98) y (99), procederemos a la conversión de los números complejos.

$$(0, 8.2i) = 8.2 \angle -90^\circ = 8.2 \angle 270^\circ \quad (100)$$

3.4 Selección de pesos de prueba para el balanceo en dos planos

El balanceo dinámico consiste en agregar peso (magnitud) en ambos planos, para visualizar la influencia generada en ellos. Vectorialmente se puede calcular la solución, la cual proporciona la posición de colocación de los pesos en los planos uno y dos.

A continuación, desglosaremos las magnitudes y ángulos seleccionados para el desarrollo del balanceo en dos planos con ángulos iguales y distintos.

3.4.1 Ángulos iguales

Figura 12 Medición de pesos fijos



Colocaremos pesos fijos de desbalance en ambos planos.

$$\text{Plano 1} = 6.7g \angle 0^\circ \quad (101)$$

$$\text{Plano 2} = 4.5g \angle 90^\circ \quad (102)$$

Agregamos 180° a la posición inicial para generar los W_c .

$$W_{c_{\text{plano1}}} = 6.7g \angle 180^\circ \quad (103)$$

$$W_{c_{\text{plano2}}} = 4.5g \angle 270^\circ \quad (104)$$

Procederemos a calcular el promedio de W_c .

$$\frac{W_{c_1} + W_{c_2}}{2} = \frac{6.7 \angle 180^\circ + 4.5 \angle 270^\circ}{2} = 4.0354 \angle -146.1130^\circ \approx 4.0354 \angle 213.887^\circ \quad (105)$$

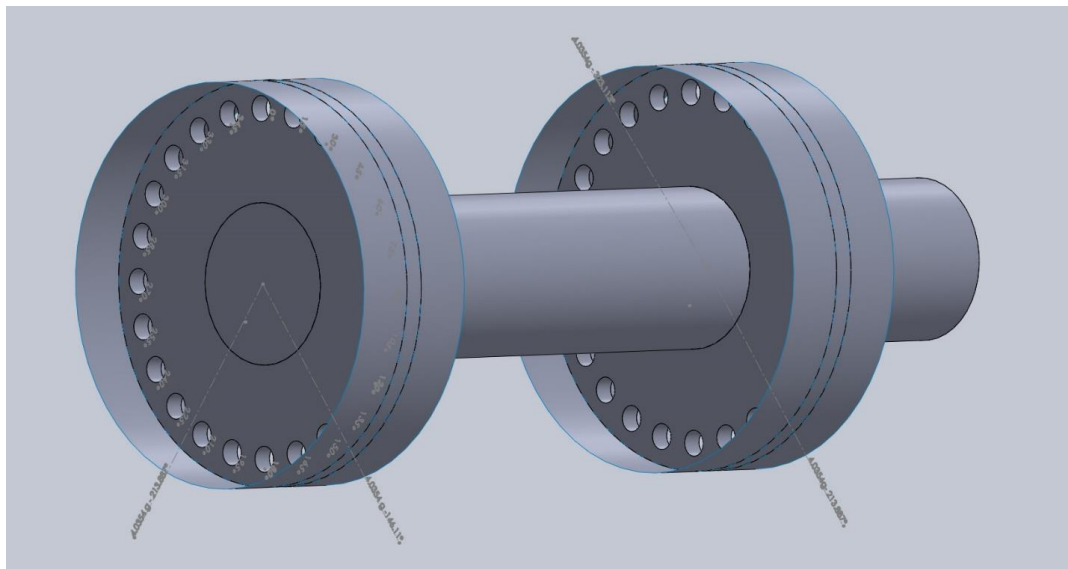
$$\frac{W_{c_1} - W_{c_2}}{2} = \frac{6.7 \angle 180^\circ - 4.5 \angle 270^\circ}{2} = 4.0354 \angle 146.1130^\circ \quad (106)$$

El cálculo de los promedios W_{c_1} y W_{c_2} nos proporcionaran las magnitudes y ángulos de los pesos para el balanceo en ambos planos, mostrados en la **Tabla 8**

Tabla 8 Magnitud y ángulo de balanceo en ambos planos

Plano 1	Plano 2
4.0354 \angle 213.887°	4.0354 \angle 213.887°
4.0354 \angle 146.1130°	4.0354 \angle 323.1130°

Figura 13 Magnitud y ángulo de balanceo en dos planos



Seguidamente procedemos al cálculo de los puntos de intersección de la ecuación (105)

$$4.0354 \cos(213.887) = -3.3499 \quad (107)$$

$$4.0354 \sin(213.887) = -2.2499$$

Inmediatamente deducimos los puntos de intersección de la ecuación (106)

$$4.0354 \cos(146.1130) = -3.3499 \quad (108)$$

$$4.0354 \sin(146.1130) = 2.2499$$

seleccionamos y colocamos el peso de prueba de desbalance número uno con la siguiente estructura.

$$\text{Plano 1} = 6.7g \angle 300^\circ \quad (109)$$

$$\text{Plano 1} = 6.7g \angle 300^\circ \quad (110)$$

Finalmente posicionamos el peso de prueba de desbalance número 2, de la siguiente manera.

$$\text{Plano 2} = 6.7g \angle 210^\circ \quad (111)$$

$$\text{Plano 2} = 6.7g \angle 210^\circ \quad (112)$$

3.4.2 Ángulos opuestos

Colocación de pesos fijos

$$\text{Plano 1} = 6.7g \angle 0^\circ \quad (113)$$

$$\text{Plano 2} = 4.5g \angle 90^\circ \quad (114)$$

Calculamos W_c

$$W_c = \frac{W_{c_1} + W_{c_2}}{2} = \frac{6.7 \angle 180^\circ + 4.5 \angle 270^\circ}{2} = 4.0354 \angle -146.1130^\circ \quad (115)$$

$$\approx 4.0354 \angle 213.887^\circ$$

Colocamos en el rotor el valor de W_c calculado en ángulos iguales (manteniendo el peso fijo), convirtiéndolo en el nuevo tal cual.

$$\text{Plano 1} = 4g \angle 215^\circ \quad (116)$$

$$\text{Plano 2} = 4g \angle 215^\circ \quad (117)$$

Agregar los pesos de prueba No.1 para el desbalance.

$$\text{Plano 1} = 6.7g \angle 30^\circ \quad (118)$$

$$\text{Plano 2} = 6.7\text{g} \angle 210^\circ \quad (119)$$

Medir y retirar el peso de prueba No.1 para colocar el segundo peso de prueba.

$$\text{Plano 1} = 6.7\text{g} \angle 120^\circ \quad (120)$$

$$\text{Plano 2} = 6.7\text{g} \angle 300^\circ \quad (121)$$

3.5 Implementación de LabVIEW

LabVIEW es un software de ingeniería enfocado al desarrollo de programación gráfica, permite mostrar una visualización del sistema implementado para el desarrollo de aplicaciones que requieren de pruebas, medidas y control con acceso rápido a hardware e información de datos.

Debido a ello, se utiliza como medio de comunicación entre la parte analógica y digital, con el objetivo de obtener los datos de variación presentes en el motor. A continuación, desglosaremos el procedimiento adecuado para la ejecución de la interfaz gráfica en el software LabVIEW.

- 1) Acoplamos cada una de las tres líneas de alimentación del motor, señaladas por los números: 1, 2 y 3, al conector saliente unido a la caja interruptora, seguidamente, conectamos el cable de tierra física.

Figura 14 Conexión de fases y pinzas Fluke



- 2) Posicionamos los amperímetros de gancho de acuerdo a los números marcados en cada una de las líneas con los mismos dígitos.

Figura 15 Pinzas Fluke en posición 1, 2 y 3, correspondientes a cada línea de alimentación



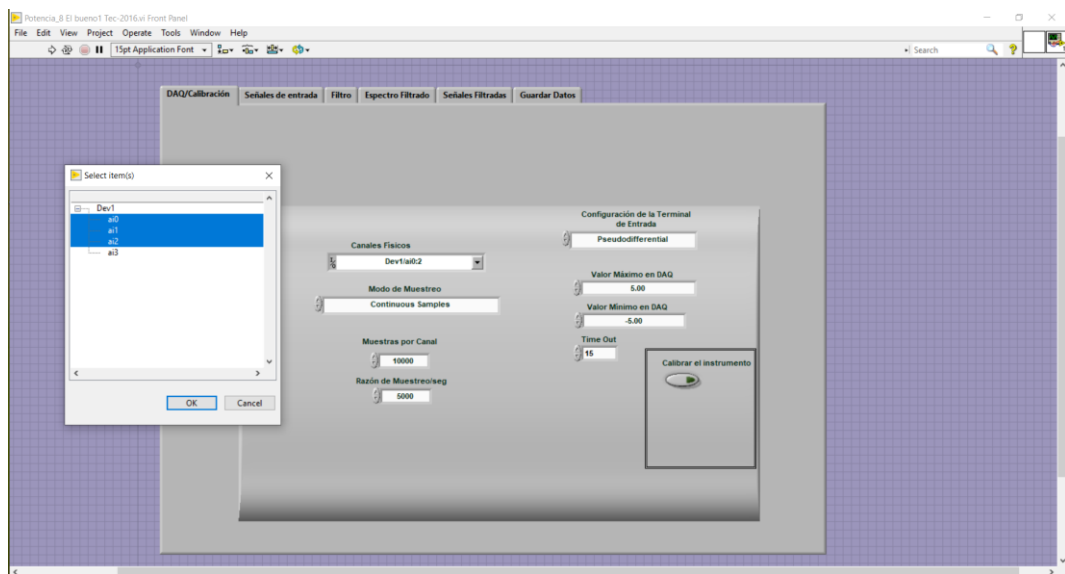
- 3) Después de situar los amperímetros a cada línea, procedemos a conectarlos a la tarjeta de adquisición de datos NATIONAL INSTRUMENTS NI USB-9233, en los canales de entrada AI0-AI2.

Figura 16 Tarjeta National Instruments



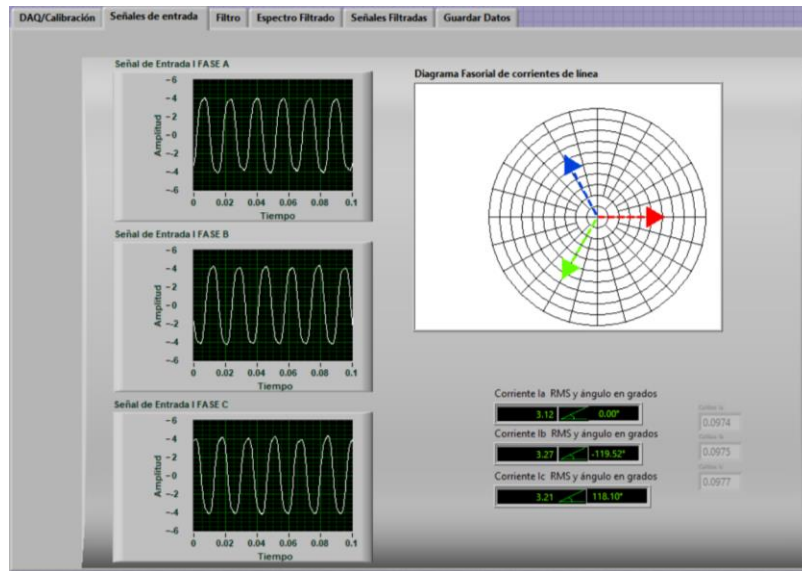
- 4) Inmediatamente conectamos la tarjeta de adquisición de datos al puerto USB de la computadora.
- 5) Se coloca el peso de prueba en los ángulos correspondientes.
- 6) Procedemos al encendido de los amperímetros y del motor.
- 7) Seguidamente ejecutamos la interfaz gráfica titulada “potencia_tec2018”, en el software NI LabVIEW 2018®.
- 8) Posteriormente la interfaz mostrará una ventana inicial, en la cual el operador deberá seleccionar los canales físicos, para cerciorarse de la correcta conexión y detección de los canales en la tarjeta, donde ingresarán los datos de captura.

Figura 17 Selección de canales físico



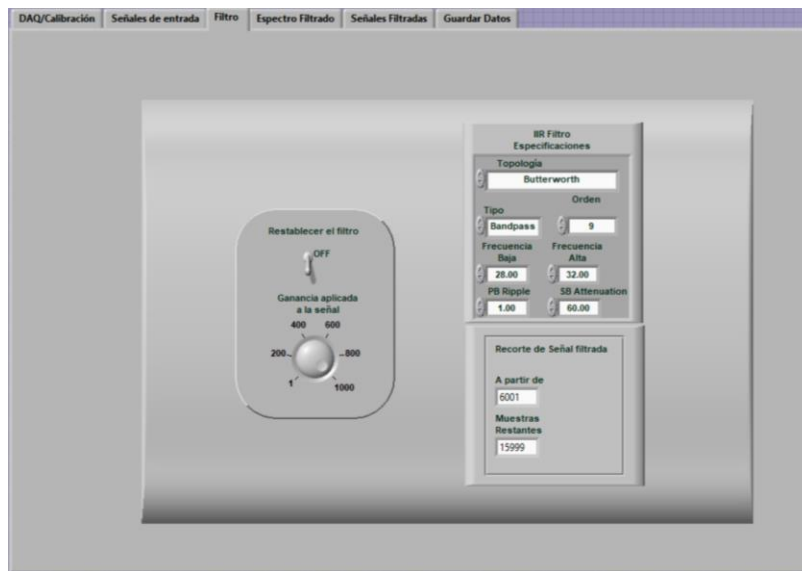
- 9) En la segunda pestaña “Señales de entrada”, muestra el comportamiento de la señal de entrada de cada fase, en relación amplitud-tiempo.

Figura 18 Señales de entrada



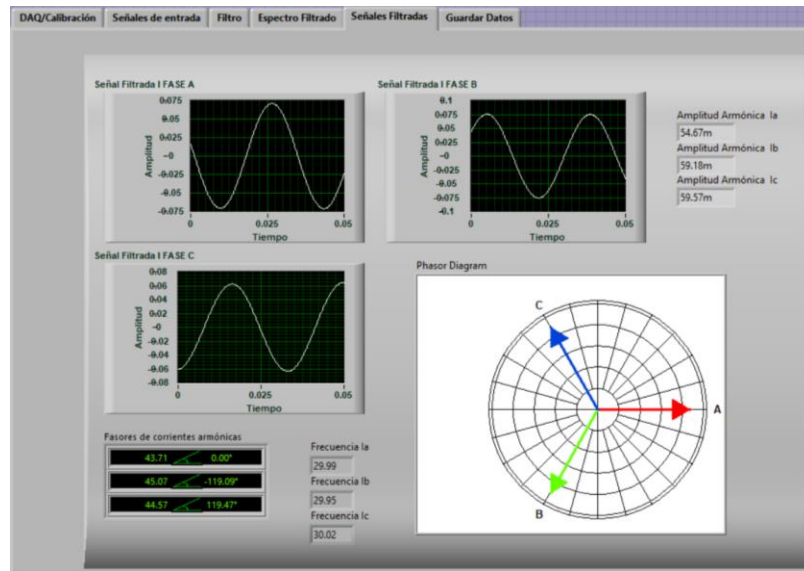
10) Consecutivamente, seleccionamos la tercera pestaña “filtro”, donde el operador puede manipular el rango de frecuencia a medir, en este caso se ajustarán medidas de $28 < 30 < 32$ para 30 Hz y $88 < 90 < 92$ para 90Hz.

Figura 19 Ajuste de filtro



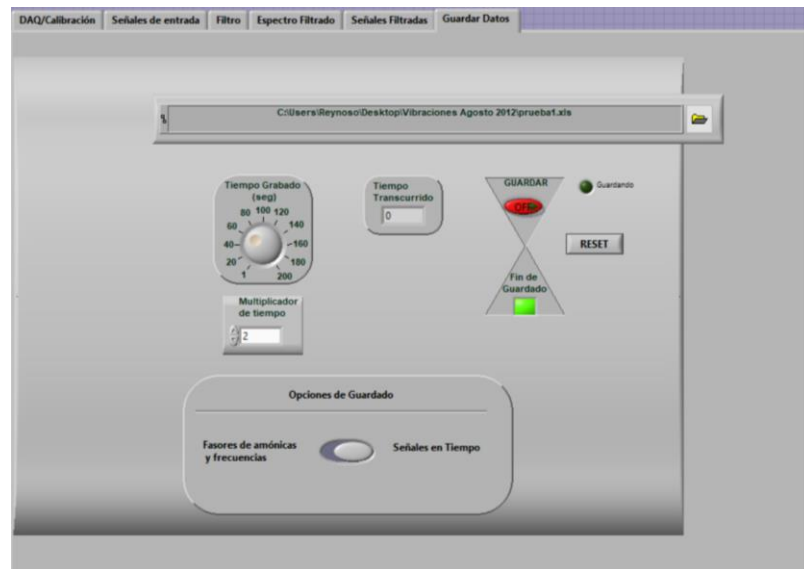
11) La pestaña “espectro filtrado” y “señales filtradas”, mostrarán gráficamente el comportamiento de la señal filtrada.

Figura 20 Señales filtradas



- 12) La pestaña “guardar datos”, cuenta con distintas opciones; como la selección del tiempo de toma de datos, en el cual se especificará un tiempo de 120 segundos.
- 13) Para el almacenamiento de datos, seleccionaremos la carpeta destino del archivo, nombrándolo, sin olvidar ajustar la extensión .xls

Figura 21 Guardar datos



14) Finalmente, pulsamos el botón con el símbolo de flecha localizado en la parte superior izquierda “ejecutar”, seguidamente tecleamos el botón guardar (color rojo) para iniciar con el respaldo de información.

Este procedimiento se realizará para los tres casos de medición correspondientes a los tipos de balanceo.

- Rotor con desbalance original (a ser balanceado)
- Rotor con un peso de prueba Wp1 agregado
- Rotor con un peso de prueba Wp2 agregado (después de retirar Wp1)

3.6 Algoritmo 12A y 12AA

Una vez establecido y calculado los datos a evaluar de la sección: “selección de pesos de prueba para el balanceo en un plano”, procederemos a la adquisición de datos generados por el motor mediante el software LabVIEW, el cual generará archivos de extensión “.dat”.

A continuación, procederemos a la extracción de datos mediante una hoja de cálculo (Microsoft Excel), el cual genera un archivo con un aproximado de 19953 datos por cada línea de alimentación (3 columnas).

Posteriormente se debe resguardar el archivo con extensión .txt (MS-DOS) en la dirección C:\scilab, con el nombre “prueba”, ya que el algoritmo cuenta con un comando de reconocimiento del archivo con el mismo nombre.

Seguidamente abrimos el software Scilab®, donde se ejecutarán los algoritmos correspondientes al tipo de archivo que será procesado. Para la transformación de archivos con filtros de 30 Hz ocuparemos el algoritmo “lector_archivos_12A”, mientras que para filtros de 90Hz implementaremos el algoritmo “lector_archivos_12AA”.

A continuación, pulsamos el botón de ejecutar localizado en la parte superior izquierda.

Figura 22 Apertura de algoritmo 12A en Scilab

```

1 // lector de archivos 12A lee el archivo en el tiempo y primero forma subgrupos de 50 ciclos, después promedia, en cada subgrupo la fase y luego calcula los fasores.
2 //lector_archivos_12A.sce analiza la fase
3
4 function [lim,OC,OD,RE,teta1,teta2,teta3]=intervalo(nint)
5
6 i=0; iflag=0; j=0;
7 // escribe B(i),B(2),B(3)
8 while i<=nint & iflag==0
9     i=i+1;
10    //for i=1:n
11    // load(fu,[i,n2,n3]);
12    [B]=read(u,i,3);
13    if B(i)~=0 then
14        iflag=i;
15    end
16    disp(iflag,"iflag=");
17    disp("B",B(i),B(2),B(3));
18    if i==999 then
19        C(i-999)=B(i); D(i-999)=B(2); E(i-999)=B(3);
20        // escribe B(i),B(2),B(3) a (i/50)
21        j=j+1;
22    end
23 // fin de while
24 disp("C",C(i),C(2),C(3));
25 disp(j,""); disp(iflag,"iflag");
26 if iflag==0 then
27 // pause
28 end
29 i1=0; i2=0; k1=0; k2=0; k3=0;
30 C1m=C(i); D1m=D(i); E1m=E(i); i1m=0; i2m=0; i3m=0;
31 C1m=C(i); D1m=D(i); E1m=E(i); i1m=0; i2m=0; i3m=0;
32
33 while i<=lim
34     i=i+1;
35     disp("i",i);
36     disp([C1m,C1m,C(i1),k1],"C1m,C1m,C(i1),k1");
37     if C1m<C1m & C(i1)<C1m then
38         Cmax(k1)=C1m; i1(k1)=i1m; k1=k1+1;
39     end
40 end

```

El algoritmo generará un total de 8 archivos (corrfa.dat), del cual se utilizará el “corrfa4.dat”, ya que este contiene datos de fase, ángulo y filtro.

Figura 23 Archivos corrfa generados por el algoritmo 12A

Nombre	Fecha de modificación	Tipo	Tamaño
corrfa2.dat	06/02/2021 12:35 p. m.	Archivo DAT	42 KB
corrfa3.dat	06/02/2021 12:34 p. m.	Archivo DAT	0 KB
corrfa4.dat	06/02/2021 12:35 p. m.	Archivo DAT	90 KB
corrfa5.dat	06/02/2021 12:35 p. m.	Archivo DAT	95 KB
corrfa6.dat	06/02/2021 12:34 p. m.	Archivo DAT	1 KB
corrfa7.dat	06/02/2021 12:34 p. m.	Archivo DAT	0 KB
corrfa8.dat	06/02/2021 12:34 p. m.	Archivo DAT	0 KB
corrfa9.dat	06/02/2021 12:34 p. m.	Archivo DAT	0 KB
prueba	11/01/2021 04:51 p. m.	Documento de texto	5,826 KB

Posteriormente abrimos el archivo “corrfa4.dat” en Excel, delimitando las especificaciones: punto, coma, espacio, etc., esto con la finalidad de generar 6 columnas con datos (tres líneas de alimentación del motor).

Figura 24 Datos del archivo corrfa4

2	Cmax(i)	teta1(i)	frec1	DDmax(i)	teta2(i)	frec2	EEmax(i)	teta3(i)	frec3
3	170.416992	0	89.969412	194.866613	14.215862	89.956759	232.058753	181.982484	90.002764
4	171.060184	0	89.963836	193.275286	14.164346	89.95274	230.443806	182.104601	90.011908
5	172.01718	0	89.959929	191.565394	14.119326	89.947607	228.638208	182.288946	90.018993
6	173.255323	0	89.955439	189.776455	14.069306	89.944719	226.696477	182.51604	90.026815
7	174.73508	0	89.954296	187.944681	14.026222	89.943112	224.668473	182.799136	90.033643
8	176.416226	0	89.951879	186.109179	13.981083	89.941247	222.609143	183.111489	90.039644
9	178.256836	0	89.952213	184.302934	13.938581	89.942655	220.56471	183.46308	90.045674
10	180.212048	0	89.952862	182.559324	13.900425	89.942852	218.585079	183.838058	90.049555
11	182.240956	0	89.953736	180.90819	13.860492	89.945818	216.710737	184.226405	90.0541
12	184.298642	0	89.957069	179.374452	13.829315	89.949287	214.982853	184.634462	90.055664
13	186.345602	0	89.958695	177.983052	13.79842	89.952913	213.433423	185.031945	90.058258
14	188.342782	0	89.963154	176.750523	13.775964	89.959126	212.092643	185.439132	90.05739
15	190.253832	0	89.966218	175.693134	13.760311	89.963588	210.982183	185.822163	90.057812
16	192.048102	0	89.97052	174.821419	13.750444	89.970761	210.120163	186.197208	90.054602
17	193.695083	0	89.975487	174.14176	13.752167	89.976465	209.517255	186.543629	90.052928
18	195.173004	0	89.979253	173.658741	13.756657	89.983031	209.179287	186.861031	90.047752
19	196.460378	0	89.985255	173.369822	13.772693	89.989921	209.106433	187.147362	90.044288
20	197.543518	0	89.988901	173.273087	13.791921	89.995138	209.293	187.390973	90.037773
21	198.411996	0	89.995011	173.359406	13.817807	90.001967	209.729991	187.599115	90.032945
22	199.059865	0	89.998983	173.619762	13.846242	90.005822	210.401686	187.759081	90.025968
23	199.487599	0	90.004471	174.040791	13.874441	90.011496	211.291956	187.878444	90.020166
24	199.69671	0	90.009063	174.60723	13.903249	90.014086	212.377072	187.950797	90.013725
25	199.697382	0	90.013493	175.302854	13.924021	90.017886	213.636049	187.978693	90.007293
26	199.498156	0	90.018614	176.10773	13.942383	90.019403	215.039995	187.964589	90.002153
27	199.116164	0	90.021983	177.004112	13.94606	90.020935	216.564074	187.905777	89.995668
28	198.566779	0	90.027069	177.969343	13.942654	90.021664	218.178184	187.811121	89.991875
29	197.871466	0	90.029531	178.985173	13.921419	90.020971	219.854987	187.675462	89.986129
30	197.050895	0	90.034115	180.02861	13.887894	90.020973	221.566041	187.511372	89.983353

3.6.1 Implementación de fasores con parte real e imaginaria

Con la finalidad de obtener datos exactos, se implementan los fasores, el cual está compuesta por una parte real proporcionada por la magnitud, mientras que la parte imaginaria será compensada por el ángulo de la fase, esto mediante la conversión de los datos extraídos de cada línea. Sin embargo, para la línea 1 no aplica el desarrollo de las fórmulas, ya que los datos no sufren ninguna modificación.

Mediante las siguientes ecuaciones, obtendremos los datos deseados.

Generación de números reales:

$$= \text{magnitud} * \text{COS} (\text{RADIANES} (\text{ángulo de fase})) \quad (122)$$

Generación de números imaginarios:

$$= \text{magnitud} * \text{SENO} (\text{RADIANES} (\text{ángulo de fase})) \quad (123)$$

Una vez obtenido los datos reales e imaginarios, procederemos a calcular el promedio de cada línea. Seguidamente se deberá efectuar las conversiones, el cual producirá el valor neto, mediante las fórmulas.

Valor total de número real:

$$= \text{RAIZ}(\text{promedio real}^2 + \text{promedio imaginario}^2) \quad (124)$$

Valor total de número imaginario:

$$= \text{ATAN2}(\text{promedio real}, \text{promedio imaginario}) * 180/\text{PI}() \quad (125)$$

Consecutivamente posicionamos de manera correcta cada valor en una fila que contará con 9 columnas (magnitud, ángulo y frecuencia). Como se muestra en la **Tabla 9**

Tabla 9 Valores de la medición con 6.7g a 0° con filtro de 90 Hz

CCmax(i)	teta1(i)	frec1	DDmax(i)	teta2(i)	frec2	EEmax(i)	teta3(i)	frec3
191.169	0	89.977	182.956	11.9305	89.977	257.3169	-151.417	89.973

Posteriormente se utilizarán los datos generados de los archivos corrfa4, para crear un archivo compuesto por 3 filas y 9 columnas.

Cmax(i)	teta1(i)	frec1	DDmax(i)	teta2(i)	frec2	EEmax(i)	teta3(i)	frec3
191.169	0	89.977	182.956	11.930	89.977	257.316	-151.417	89.973
171.024	0	89.951	217.958	2.455	89.940	213.053	-164.025	89.944
204.882	0	89.997	219.681	-13.067	90.003	195.043	-9.620	89.994

Al culminar la elaboración de la tabla, se deberá guardar el archivo con el nombre "prueba" con extensión .txt (MS-DOS) en la dirección C:\scilab.

3.6.2 Implementación de fases

Este método consiste en evaluar los valores de varias mediciones entre sí, de los datos generados por las fases (magnitudes), con la finalidad de tener una precisión más cercana de los datos extraídos. Esto mediante la implementación de las siguientes ecuaciones.

Calcular el valor mínimo de la fase:

$$= \text{MIN}(\text{valor inicial de la fase: valor final de la fase}) \quad (126)$$

Calcular el promedio de la fase:

$$= \text{PROMEDIO}(\text{valor inicial de la fase: valor final de la fase}) \quad (127)$$

Calcular el valor máximo de la fase:

$$= \text{MAX}(\text{valor inicial de la fase: valor final de la fase}) \quad (128)$$

Al finalizar el procesamiento de datos de cada fase, procederemos a posicionar de derecha a izquierda los valores obtenidos de la siguiente manera: mínimo, promedio y máximo. Hasta crear una fila con 9 columnas.

Tabla 10 Datos de peso de prueba 6.7g en 0° a 30 Hz

FASE 1			FASE 2			FASE 3		
mínimo	promedio	máximo	mínimo	promedio	máximo	mínimo	promedio	máximo
10.138	26.288	44.209	8.293	22.879	43.030	10.309	99.874	175.077

Consecutivamente se emplearán los datos generados de los archivos corrfa4, para crear un archivo compuesto por 3 filas y 9 columnas.

10.138	26.288	44.209	8.293	22.879	43.030	10.309	99.874	175.077
0.025	13.031	37.690	31.792	49.709	67.655	11.295	31.467	55.562
17.413	40.564	57.596	28.531	45.769	67.511	0.662	12.191	33.090

Finalmente, se deberá guardar el archivo con el nombre “prueba15” con extensión .txt

(MS-DOS) en la dirección C:\scilab.

3.7 Algoritmo 8H6

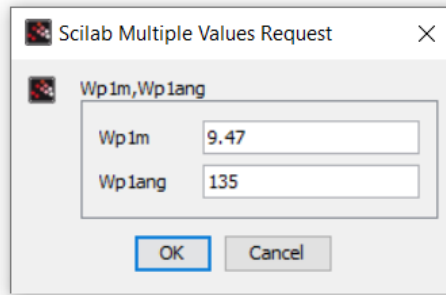
Este algoritmo realiza la iteración del offset, el ángulo de N y la fase de las lecturas.

Mediante el promedio de las tres fases, genera dos soluciones posibles; una es la correcta para el desbalance del rotor, mientras que la otra solución tendría las mismas lecturas con otro desbalance.

Una vez obtenido y generado el archivo con los datos de la sección “implementación de fasores con parte real e imaginaria”, se ejecutará el algoritmo “pesos_diferentes_8H6”, en el software Scilab®.

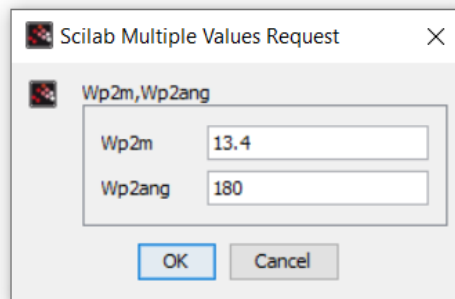
Al efectuarse la inicialización del algoritmo, mostrará una ventana donde se deberán ingresar los datos de la magnitud y ángulo, de la primera combinación derivada de la sección “selección de pesos de prueba y ángulos para el balanceo en uno o dos planos”.

Figura 25 Resultados de la 1ra combinación del peso de prueba 6.7g



Posteriormente se desplegará una segunda ventana, en donde se ingresarán los valores de magnitud y ángulo, derivados de la segunda combinación.

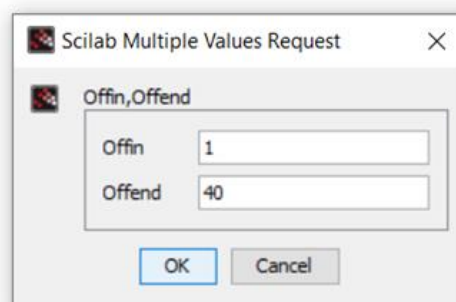
Figura 26 Resultados de la 2da combinación del peso de prueba 6.7g



Después de ingresar las magnitudes y ángulos de ambas combinaciones, se deberá especificar el rango de evaluación de entrada:

- Para 30Hz: Offin=1 y de salida Offout=40
- Para 90Hz: Offin=80 y Offout=100

Figura 27 Rango de evaluación (offset)



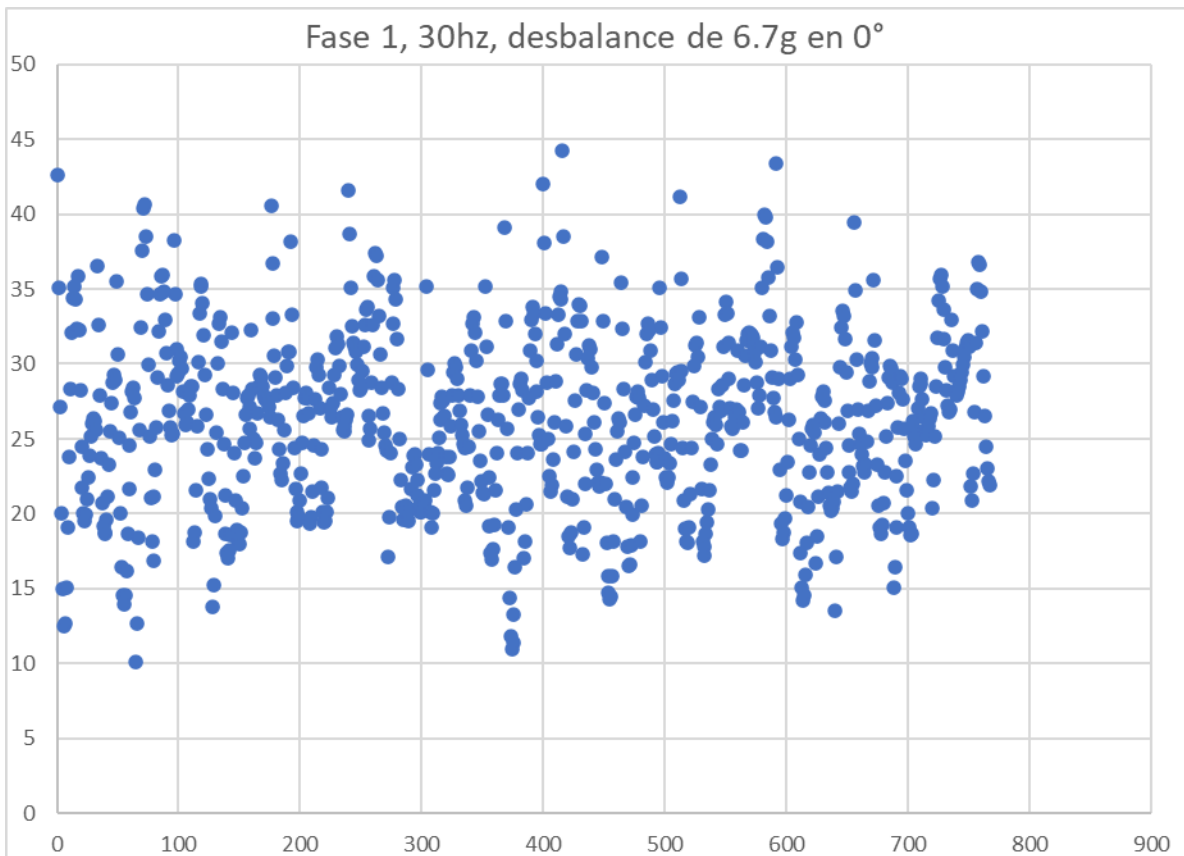
Al completar las iteraciones requeridas por el algoritmo, generará un total de 9 archivos corrfa.dat, de los cuales se utilizarán: los corrfa7.dat, en el cual mostrará los posibles pesos de balanceo en las columnas Wc21r y Wc21i, mientras que el archivo corrfa9.dat proporciona información de los promedios de las intersecciones de los tres círculos de fase por cada posición del offset.

3.8 Algoritmo 8H16

El algoritmo procesa los datos de medición de las armónicas de 30 y 90 Hz, implementando el promedio de las amplitudes. Sin embargo, éstas varían durante la toma de la muestra.

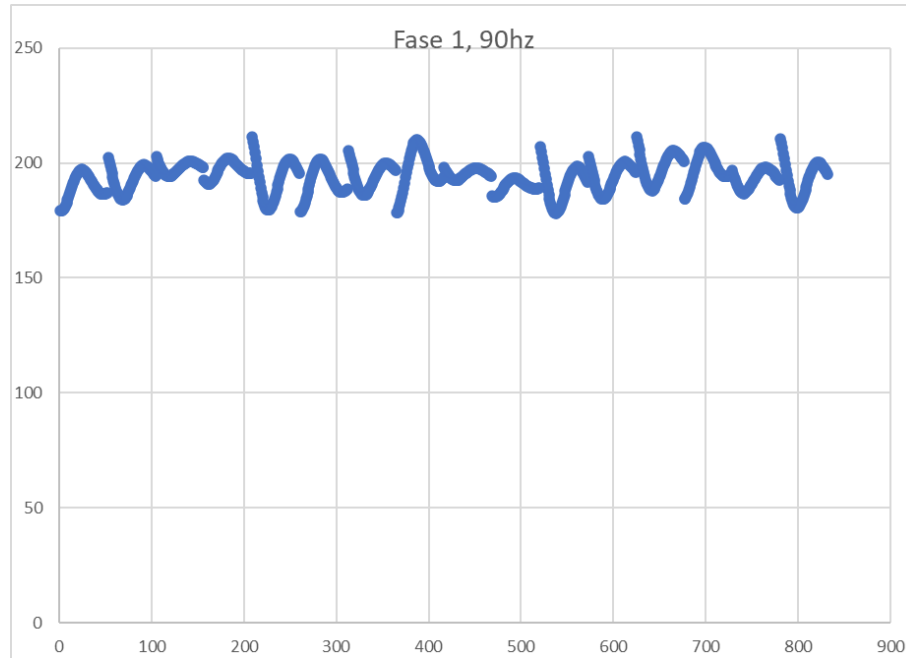
En la **Figura 28**, evidencia una gran dispersión de los datos que ocurre durante las mediciones a 30 Hz.

Figura 28 Amplitudes de corriente en miliamperios a 30Hz con valor mínimo de 10.13, máximo de 44.20 y promedio de 26.28



Mientras que en la **Figura 29** muestra una pequeña dispersión de los datos que ocurre en mediciones a 90hz.

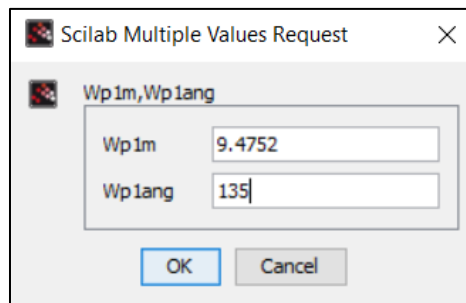
Figura 29 Amplitudes de corriente en miliamperios a 90 Hz, con valores mínimo de 178, máximo de 211.12 y promedio de 194.15



Para la ejecución del programa 8H16, se necesitará proporcionar un archivo base elaborado en la sección “implementación de fases”.

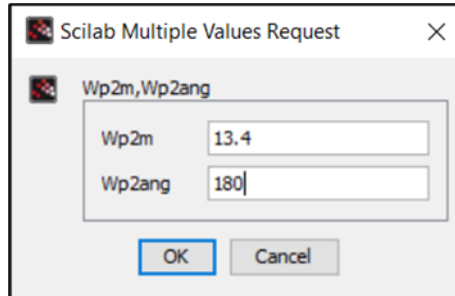
Al correr el algoritmo, mostrará una ventana donde se deberán ingresar los datos de la magnitud y ángulo, de la primera combinación derivada de la sección “selección de pesos de prueba y ángulos para el balanceo en uno o dos planos”.

Figura 30 Resultados de la 1ra combinación de 6.7g en un plano



Posteriormente se desplegará una segunda ventana, en donde se ingresarán los valores de magnitud y ángulo, derivados de la segunda combinación.

Figura 31 Resultados de la 2da combinación de 6.7g en un plano



Consecutivamente se deberá especificar el rango de evaluación de entrada, estableciendo un offset único para las tres fases que generarán un círculo de fase 1, donde presente los posibles pesos de balanceo.

- Para 30Hz: Offin=19
- Para 90Hz: Offin=200

Al completar las iteraciones requeridas por el algoritmo, producirá un total de 9 archivos corrfa.dat, de los cuales se utilizarán: corrfa3, corrfa4, corrfa5, encargados de proporcionar información para la generación de los círculos de las fases 1, 2 y 3. Este círculo tiene un centro (Cx1, Cy1) y un radio R1 que tiene las siguientes propiedades.

- El centro no cambia si se mantiene la configuración, aunque se escale el vector.
- El radio varía inversamente proporcional al factor de escalación del vector, por ejemplo, si el factor es 0.5 entonces el radio es 2R1.

El algoritmo encuentra el rango de factores de escalación de tal manera que no se excedan los mínimos y máximos de las tres fases.

Mientras que el archivo corrfa7 aporta indicadores de confiabilidad de los círculos, siendo 0 un valor exacto, mientras que 1 es un valor aproximado de la fase. Corrfa9 es el encargado de proporcionar información de los círculos originales del offset a evaluar.

Capítulo 4. Discusiones y resultados

4.1 Introducción

Este capítulo tiene como objetivo realizar el análisis de los resultados obtenidos de los algoritmos, así como la validación del método. Se presentan los datos originados para el peso de balanceo, la representación gráfica del comportamiento de las fases con armónicas de 30 y 90 Hz. Como se ha mencionado en los capítulos anteriores, es muy importante la validación del método, debido a ello se realizaron distintas pruebas, concluyendo que el filtro con mayor precisión y exactitud es 90 Hz.

Es importante enfatizar que el método de balanceo en dos planos con dos pesos de prueba proporciona dos soluciones, es decir, dos pesos de balanceo de los cuales uno es el correcto para el rotor.

4.2 Balanceo dinámico implementando armónicas de 30 Hz

4.2.1 Balanceo en un plano con peso de prueba de 6.7g

Los archivos corrfa3, corrfa4 y corrfa5 generados por el algoritmo 8H16, proporcionan el comportamiento de las fases respecto al cálculo de W_c , mostrados en la **Figura 32**

Se analizan distintas selecciones de datos manteniendo la misma proporción de los promedios, sin salirse de los intervalos de variación que se tiene en las mediciones de las armónicas.

Para 30Hz exhibe numerosas dispersiones, en este caso el indicador del programa marca a la fase 3 como la solución aproximada, debido a los ajustes para evitar raíces de números negativos, el resultado es que estos círculos no intersecan a los de las otras fases, en donde se encuentra la solución correcta. Para la fase 1 y 2 existen intersección en el punto correcto (rojo).

Mientras que el archivo corrfa7, nos proporcionara los valores de W_c calculados para la configuración promedio, representados en la **Figura 33**.

Figura 32 Comportamientos de las fases respecto a $Wc1$, 6.7g a 30 Hz

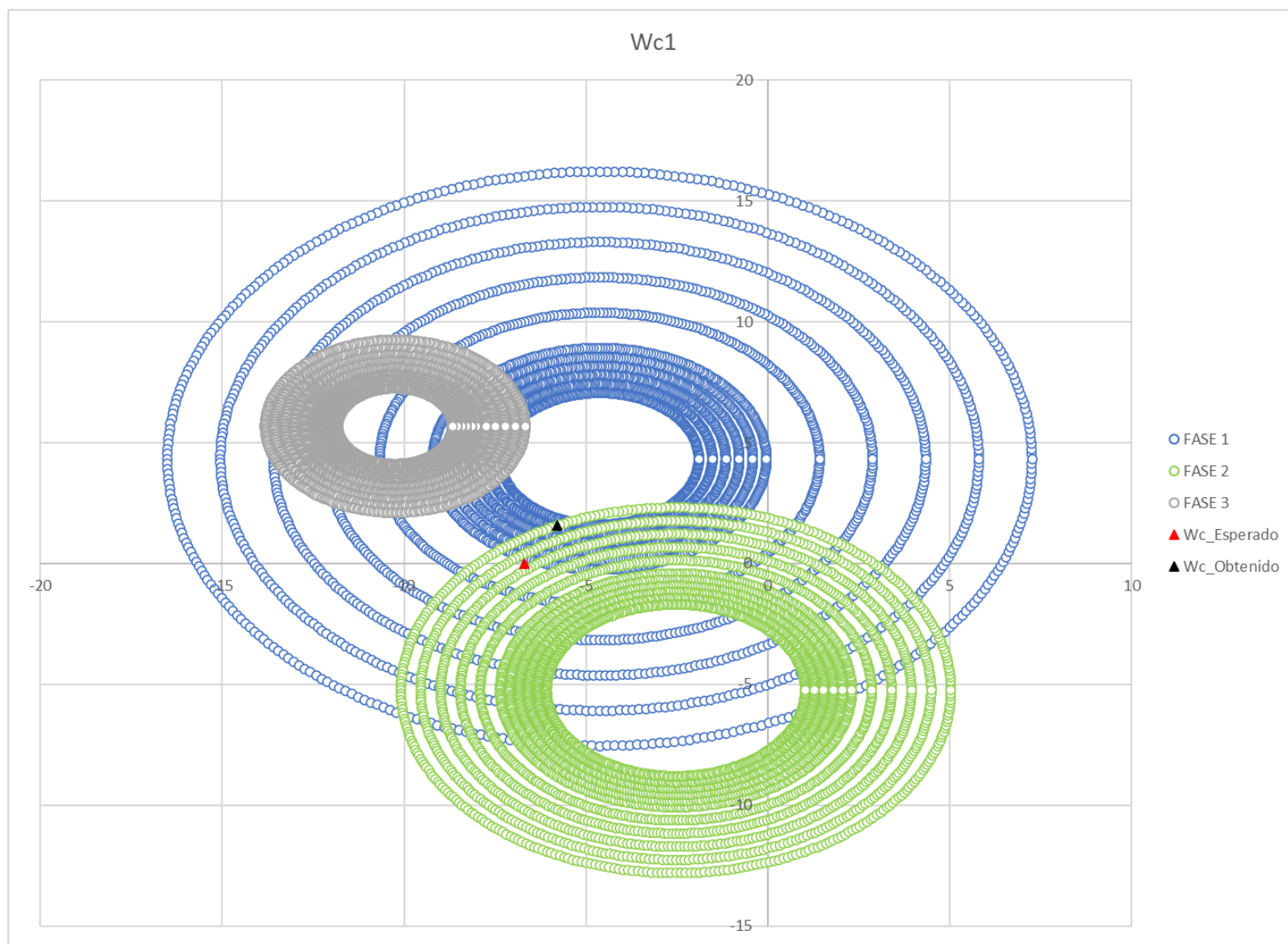


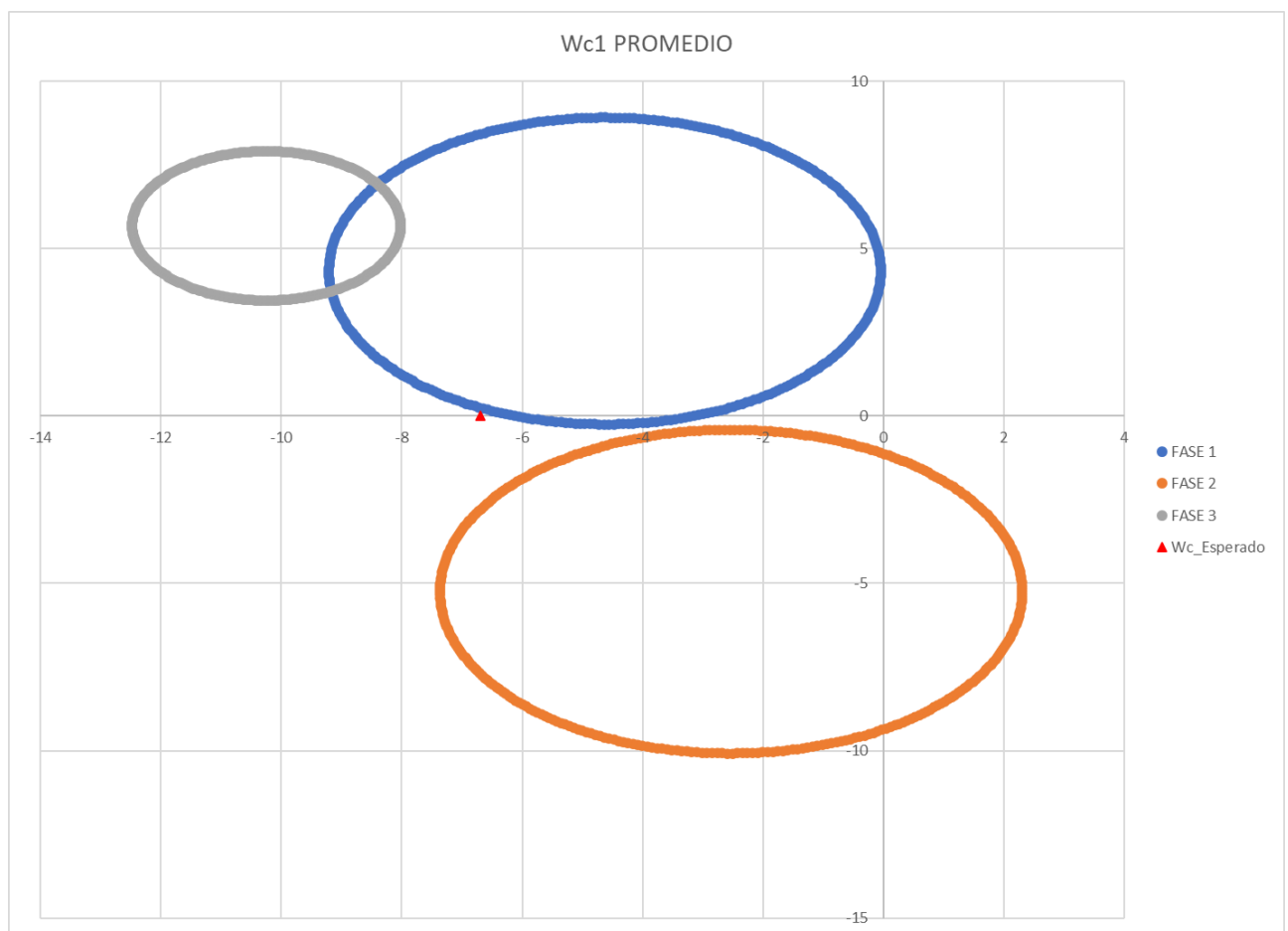
Tabla 11 Valores de W_c con peso de prueba 6.7g a 30 Hz

Calculado		Generado por el algoritmo	
W_{c1r}	W_{c1i}	W_{c1r}	W_{c1i}
-6.7	0	-5.8025	1.59

Los datos obtenidos muestran una exactitud del 86.6044% del valor neto de W_{c1r} , sin embargo, el valor aumenta considerablemente para W_{c1i} , creando un margen de error considerable para su evaluación.

El archivo corffa9 muestra el comportamiento del offset (para filtro de 30 se evalúa un offset de 19) de las fases de alimentación del motor.

Figura 33 Comportamiento de W_{c1} para 6.7g a 30 Hz



4.2.2 Balanceo en un plano con peso de prueba de 8.2g

Este experimento implicó un aumento en el peso de prueba, para comprobar algún tipo de alteración o delimitación que llegara a presentar el algoritmo. Utilizando los archivos corrfa3, corrfa4 y corrfa5 obtendremos el comportamiento de las fases respecto al cálculo de W_c demostrado en la **Figura 34**

Mientras la interpretación de los datos proporcionados por el archivo corrfa7, demuestran un comportamiento diferente de la fase 1, ya que los indicadores jy proveen un valor de 1 para esta, indicando cifras de cálculos aproximados al valor deseado, sin embargo, las fases 2 y 3 poseen un valor de 0, lo cual considera el cálculo realizado por el algoritmo como resultados exactos al valor esperado. Por lo cual se deduce que en la intersección de los círculos 2 y 3 (fases), presentara el valor del W_c esperado, mostrado en la **Figura 34**

Asimismo, el archivo corrfa7 contará con una segunda función, el cual consiste el proporcionar los valores de W_c calculados, tanto para la parte imaginaria y real, como se desglosa en la **Tabla 12**.

Tabla 12 Valores de W_c con peso de prueba 8.2g a 30 Hz

Calculado		Generado por el algoritmo	
W_{c1r}	W_{c1i}	W_{c1r}	W_{c1i}
0	-8.2	-1.9113	-8.7808

Los datos muestran una cifra menor para W_{c1r} , mientras que para W_{c1i} presenta un aumento del 106.09% del valor real. Produciendo un error considerable para su evaluación.

Por otra parte, el archivo corrfa9 muestra el comportamiento del offset (para filtro de 30 se evalúa un offset de 19) de las fases de alimentación del motor.

Figura 34 Comportamiento de las fases respecto a W_{c1} para 8.2g a 30 Hz

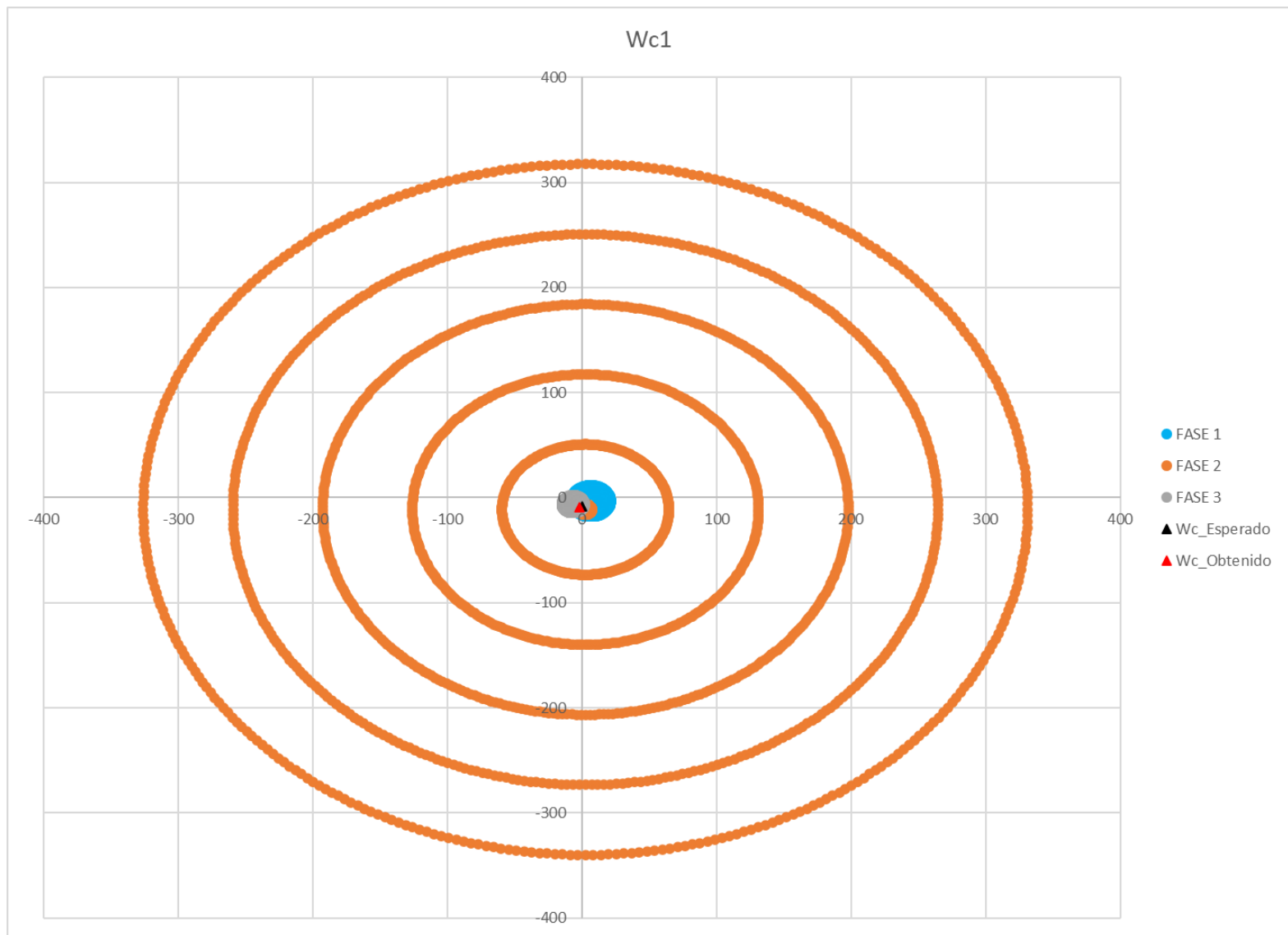
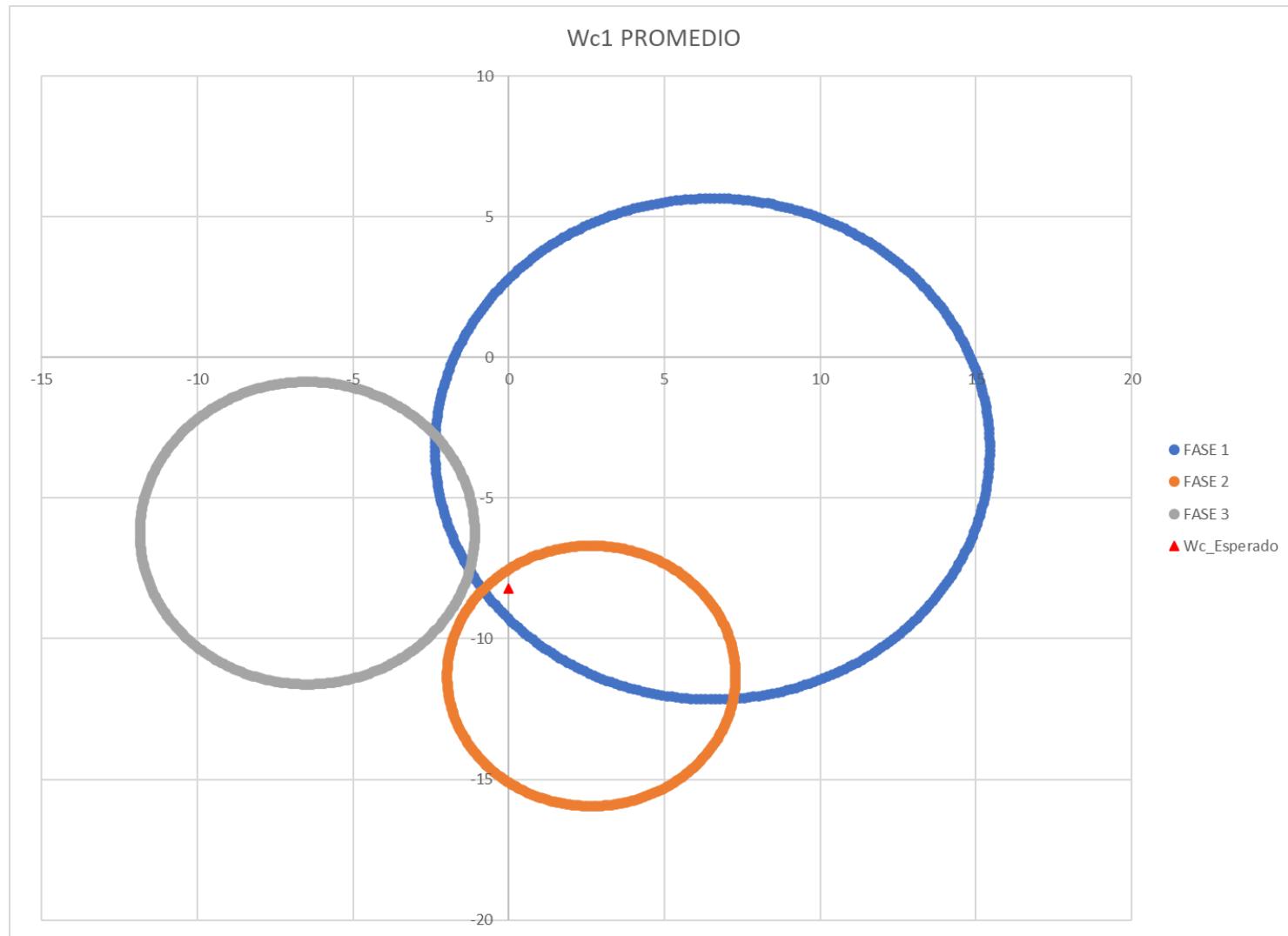


Figura 35 Comportamiento de W_{c1} promedio para 8.2g a 30 Hz



4.2.3 Balanceo en dos planos ángulos iguales con peso de prueba de 6.7g

La prueba para el balanceo en dos planos implica la colocación de dos pesos de pruebas iguales, en este caso 6.7g, ambos posicionados en ángulos iguales en el plano uno y dos, además de implementar el mismo algoritmo para el balanceo en un plano. Lo cual involucra emplear los archivos corrfa3, corrfa4 y corrfa5, encargados de proporcionar información del comportamiento de las fases respecto al cálculo de W_c .

El archivo corrfa7 proporciona datos para la interpretación del cálculo del valor correspondiente a cada fase, para este caso contará con dos planos.

La obtención de las fases en el plano 1, muestran un comportamiento homogéneo debido a que el indicador j_y proporcionan valores de 0 para las tres fases, esto señala que los valores de la fase son exactos, por lo cual la intersección de los círculos 1, 2 y 3, contendrán el valor del W_c en su unión, como se muestra en la **Figura 36**.

Igualmente, para las fases del plano 2, proporciona indicadores j_y con valores de 0 demostrando un cálculo exacto del valor en las tres fases, representadas en la **Figura 37**

Asimismo, el archivo corrfa7 contará con una segunda función, el cual consiste el proporcionar los valores de W_c calculados, tanto para la parte imaginaria y real, como se desglosa en la **Tabla 13**.

Figura 36 Comportamiento de las fases para W_{c1} , ángulos iguales con peso de prueba 6.7g a 30Hz

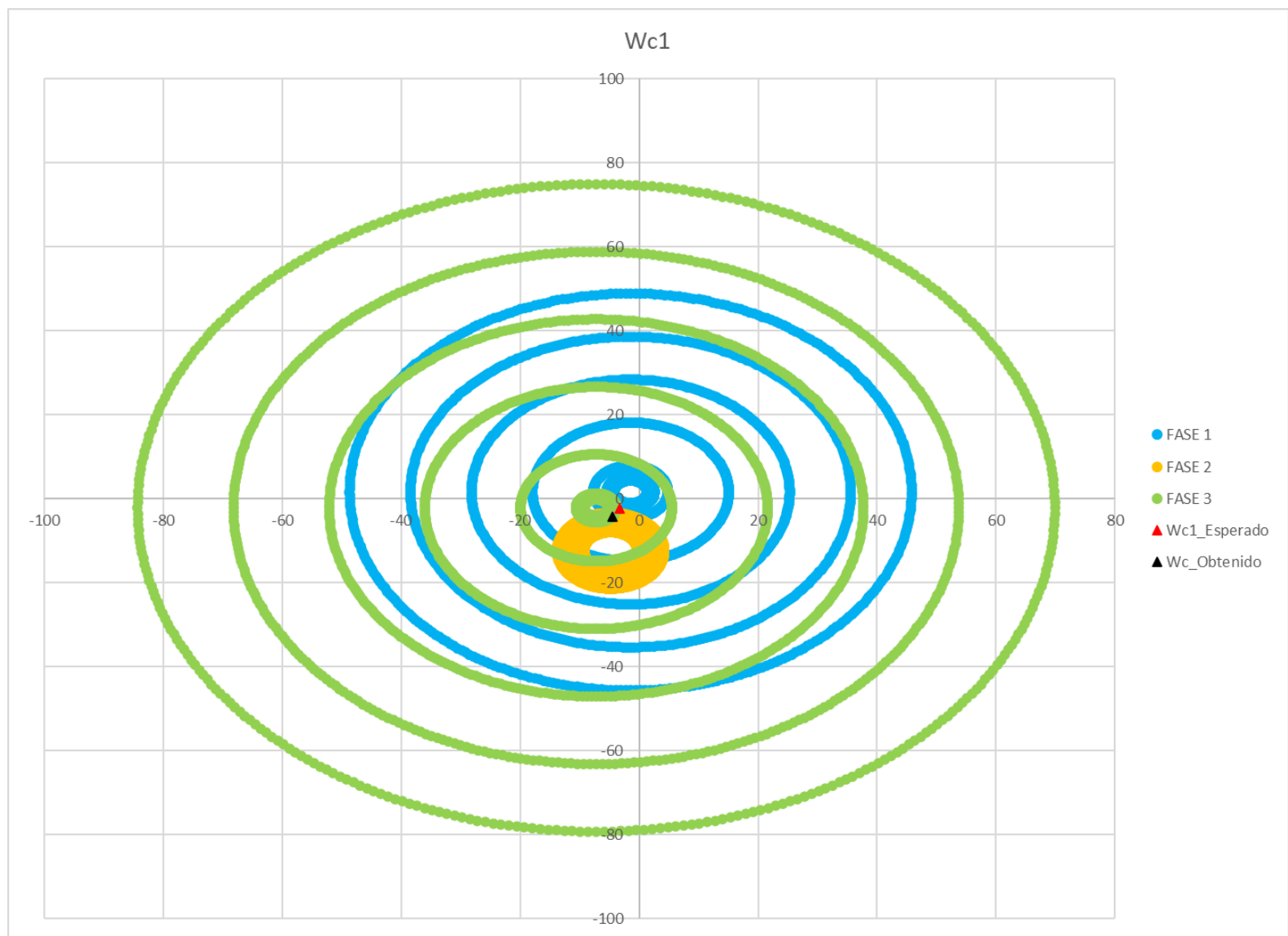


Figura 37 Comportamiento de las fases para $Wc2$, ángulos iguales con peso de prueba 6.7g a 30 Hz

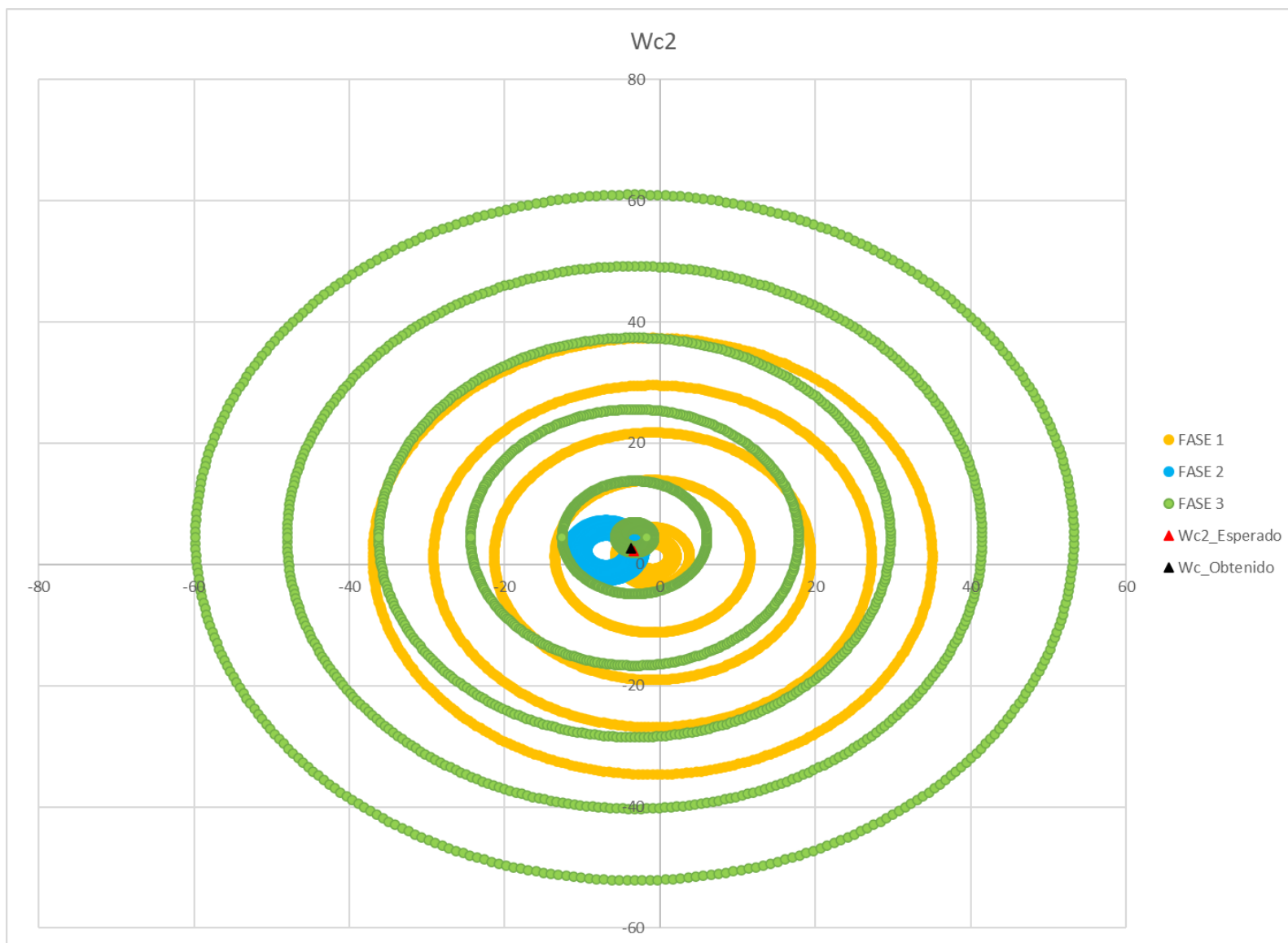


Tabla 13 Valores de W_c ángulos iguales a 30 Hz

Plano 1				Plano 2			
calculado		algoritmo		calculado		algoritmo	
W_{c1r}	W_{c1i}	W_{c1r}	W_{c1i}	W_{c2r}	W_{c2i}	W_{c2r}	W_{c2i}
-3.3499	-2.2499	-4.4756	-4.3059	-3.3449	2.2499	-3.7313	2.7007

Los datos generados por el algoritmo para el primer plano muestran un aumento de 133.60% del valor de W_{c1r} , mientras que para W_{c1i} el incremento es aún mayor con un 191.381%.

Sin embargo, para el segundo plano las cifras disminuyen considerablemente, ya que W_{c1r} presenta una elevación de 111.55%, mientras que W_{c1i} mantiene una cifra de 120.03%.

Por otra parte, el archivo corffa9 muestra el comportamiento de las fases de alimentación del motor respecto al valor de W_{c1} y W_{c2} esperado, el cual es representado por la **Figura 38** y la **Figura 39** respectivamente.

De acuerdo con los datos generados, la **Figura 39** muestra un comportamiento homogéneo de los círculos, creando así una intersección de las tres fases cercana al valor esperado de W_c , siendo W_{c2} la mejor opción para el balanceo en dos planos.

Figura 38 Comportamiento de W_{c1} promedio, ángulos iguales con peso de prueba 6.7g a 30 Hz

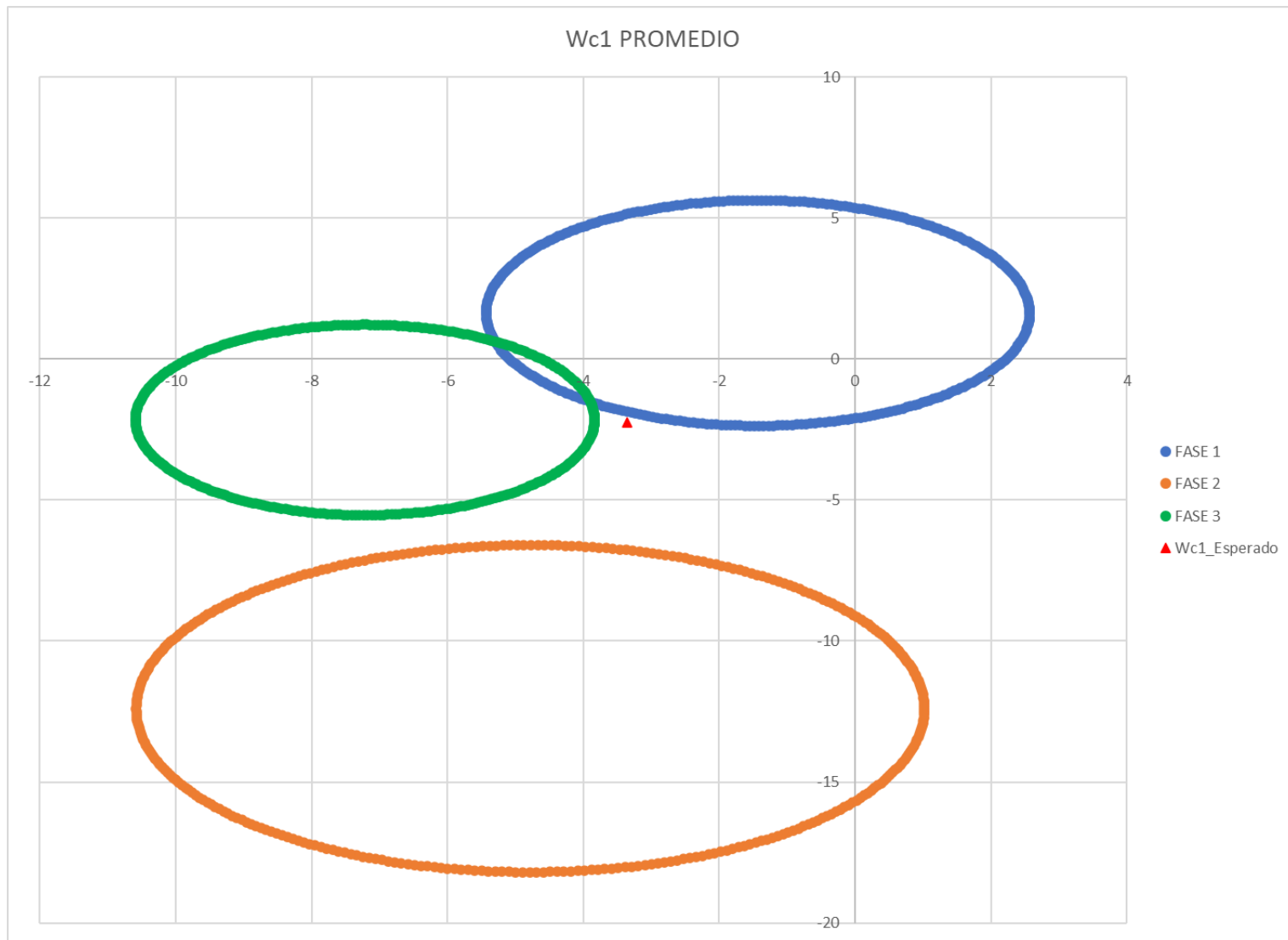
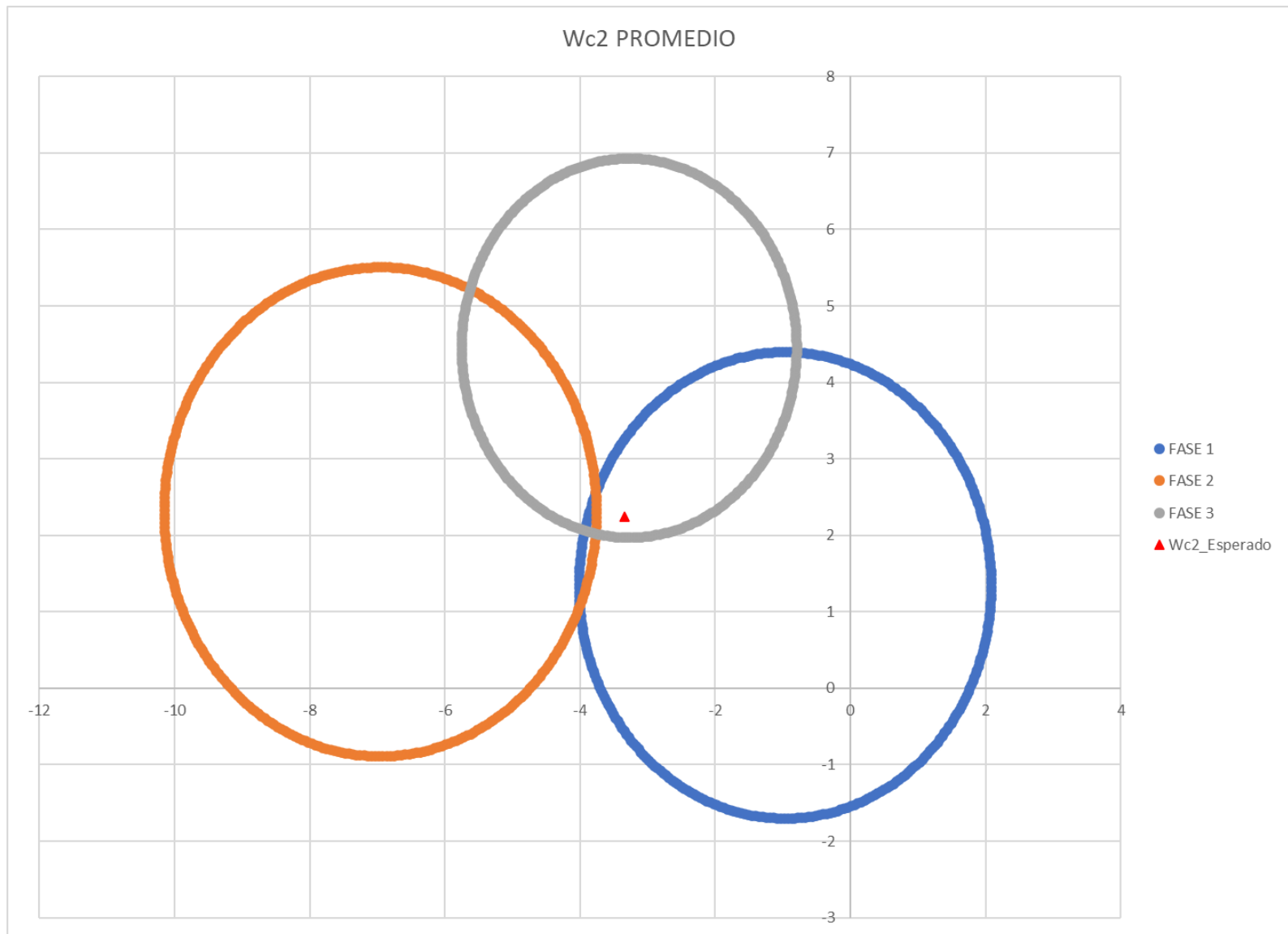


Figura 39 Comportamiento de W_{c2} promedio, ángulos iguales con peso de prueba 6.7g a 30 Hz



4.2.4 Balanceo en dos planos ángulos distintos con peso de prueba de 6.7g

La prueba para el balanceo en dos planos con ángulos distintos implica la colocación de dos pesos de pruebas con la misma magnitud (6.7g), ambos posicionados en ángulos diferentes en el plano uno y dos, además de implementar el mismo algoritmo para el balanceo en un plano. Por lo cual utiliza los archivos corrfa3, 4 y 5, encargados de proporcionar información del comportamiento de las fases respecto al cálculo de W_c .

El archivo corrfa7 proporciona datos para la interpretación del cálculo del valor correspondiente a cada fase, para este caso contará con dos planos.

La obtención de las fases en el plano 1, muestran un comportamiento homogéneo debido a que el indicador j_y proporcionan valores de 0 para las tres fases, esto señala que los valores de la fase son exactos, por lo cual la intersección de los círculos 1, 2 y 3, contendrán el valor del W_c en su unión, como se muestra en la **Figura 40**

Igualmente, para las fases del plano 2, proporciona indicadores j_y con valores de 0 demostrando un cálculo exacto del valor en las tres fases, representadas en la **Figura 41**

Asimismo, el archivo corrfa7 contará con una segunda función, el cual consiste el proporcionar los valores de W_c calculados, tanto para la parte imaginaria y real, como se desglosa en la **Tabla 14**

Tabla 14 Valores de W_c para ángulos distintos a 30Hz

Calculado		Algoritmo	
W_{c1r}	W_{c1i}	W_{c1r}	W_{c1i}
-3.3449	2.2499	1.123	4.5078

Los datos muestran una diferencia de -2.2219 para W_{c1r} , mientras que para W_{c1i} presenta un aumento de 200.35% del valor real. Este caso donde los círculos son casi concéntricos y no existe una intersección, es debido a que las configuraciones de los datos $\begin{Bmatrix} R_{01} \\ R_{02} \\ R_{03} \end{Bmatrix}$

en cada fase son muy similares. Esto se soluciona aumentando la magnitud de los pesos de desbalance.

Mientras que el archivo corrfa9 nos proporcionará los datos del comportamiento de las fases de alimentación del motor respecto al valor de W_{c1} esperado, de acuerdo con la **Figura 41** la formación de los círculos comparte el mismo centro u origen, respaldando lo anteriormente planteado.

Figura 40 Comportamiento de las fases para W_{c1} , ángulos opuestos con peso de prueba 6.7g a 30 Hz

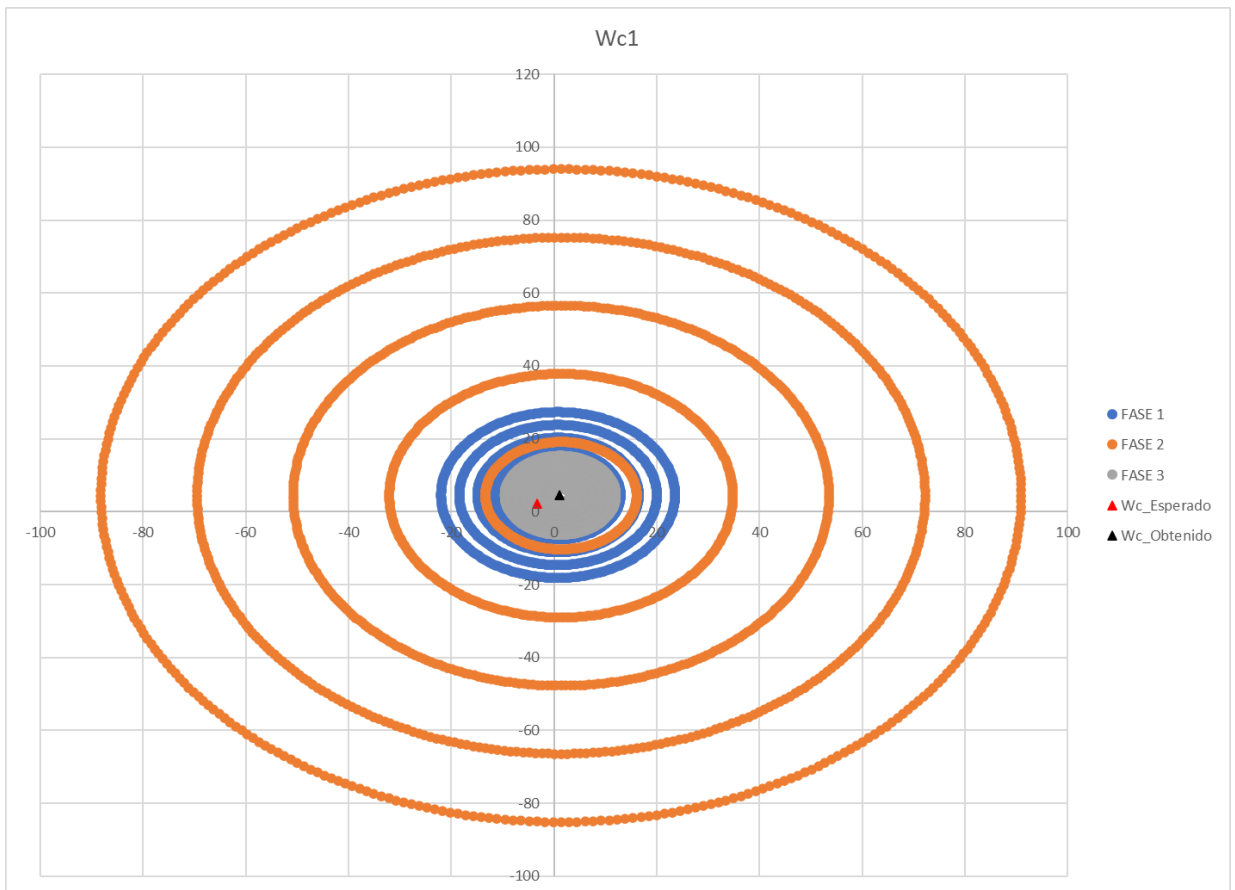
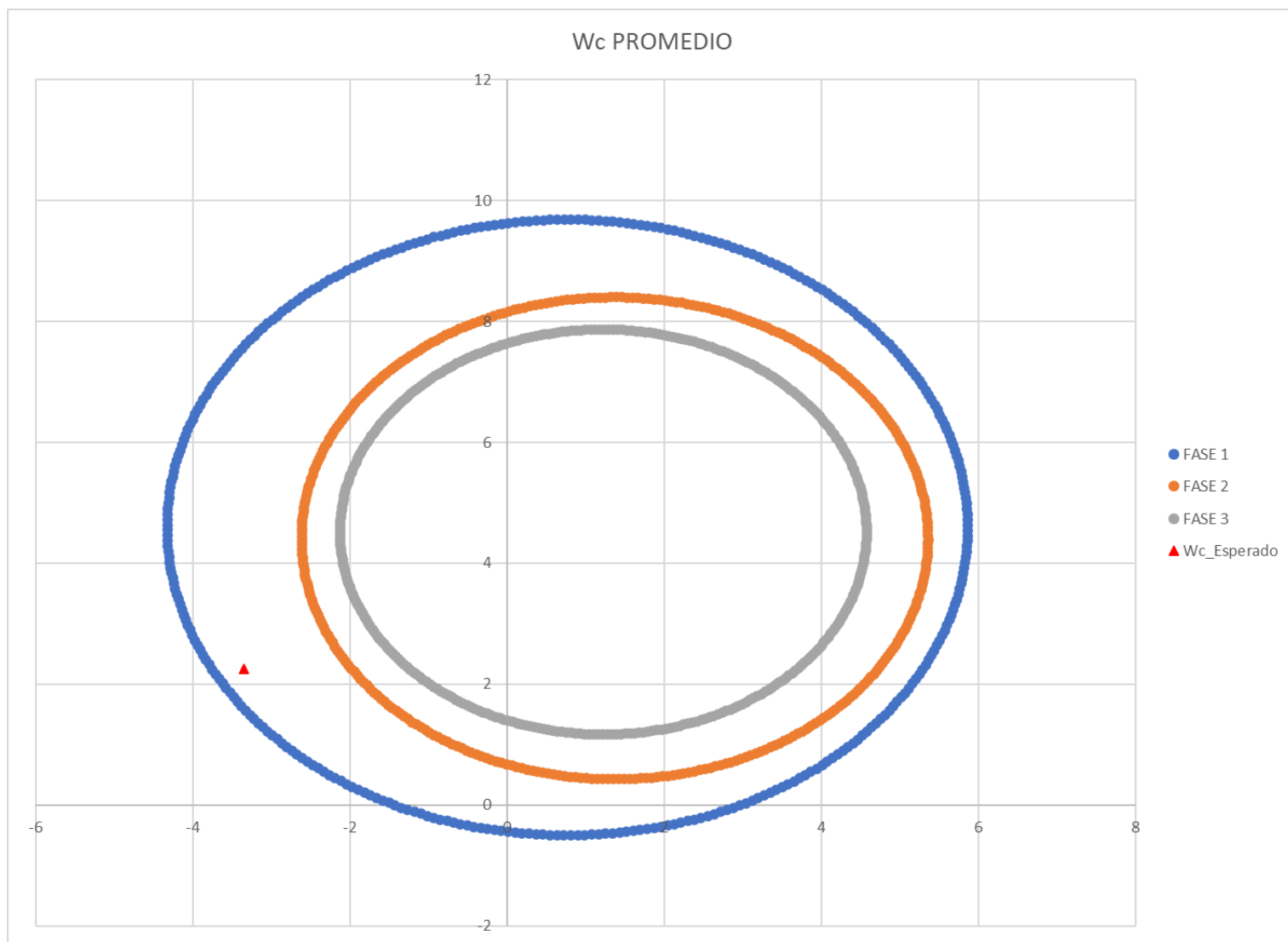


Figura 41 Comportamiento de W_{c1} promedio, ángulos distintos con peso de prueba 6.7g a 30 Hz



4.2.5 Balanceo en dos planos ángulos iguales con peso de prueba de 10g

La última prueba para el balanceo en dos planos a 30 Hz posiciona dos pesos de pruebas iguales, para este caso 10g, ambos posicionados en ángulos iguales en el plano uno y dos. El algoritmo genera los archivos corrfa3, 4 y 5, encargados de brindar información del comportamiento de las fases respecto al cálculo de W_c .

El archivo corrfa7 proporciona datos para la interpretación del cálculo del valor correspondiente a cada fase, para este caso contará con dos planos.

La obtención de las fases en el plano 1, muestran un comportamiento homogéneo para las fases 1 y 2 debido a que el indicador j_y proporcionan valores de 0, mientras que para la fase 3 la dispersión de sus datos es significativo debido a que el indicador j_y proporcionan el valor de 1, esto indica que la intersección de los círculos 1, 2 contendrán el valor del W_c en su unión, como se muestra en la **Figura 42**

Para el segundo plano, muestra un comportamiento homogéneo en las fases 1 y 2 debido a que el indicador j_y proporcionan valores de 0, mientras que para la fase 3 la dispersión de sus datos es evidente ya que el indicador j_y proporcionan el valor de 1, deduciendo que en la intersección de los círculos 1, 2 contendrán el valor del W_c en su unión, como se muestra en la **Figura 43**.

Asimismo, el archivo corrfa7 contará con una segunda función, el cual consiste el proporcionar los valores de W_c , desglosado en la **Tabla 15**

Figura 42 Comportamiento de las fases para $Wc1$, ángulos iguales con peso de prueba 10g a 30 Hz

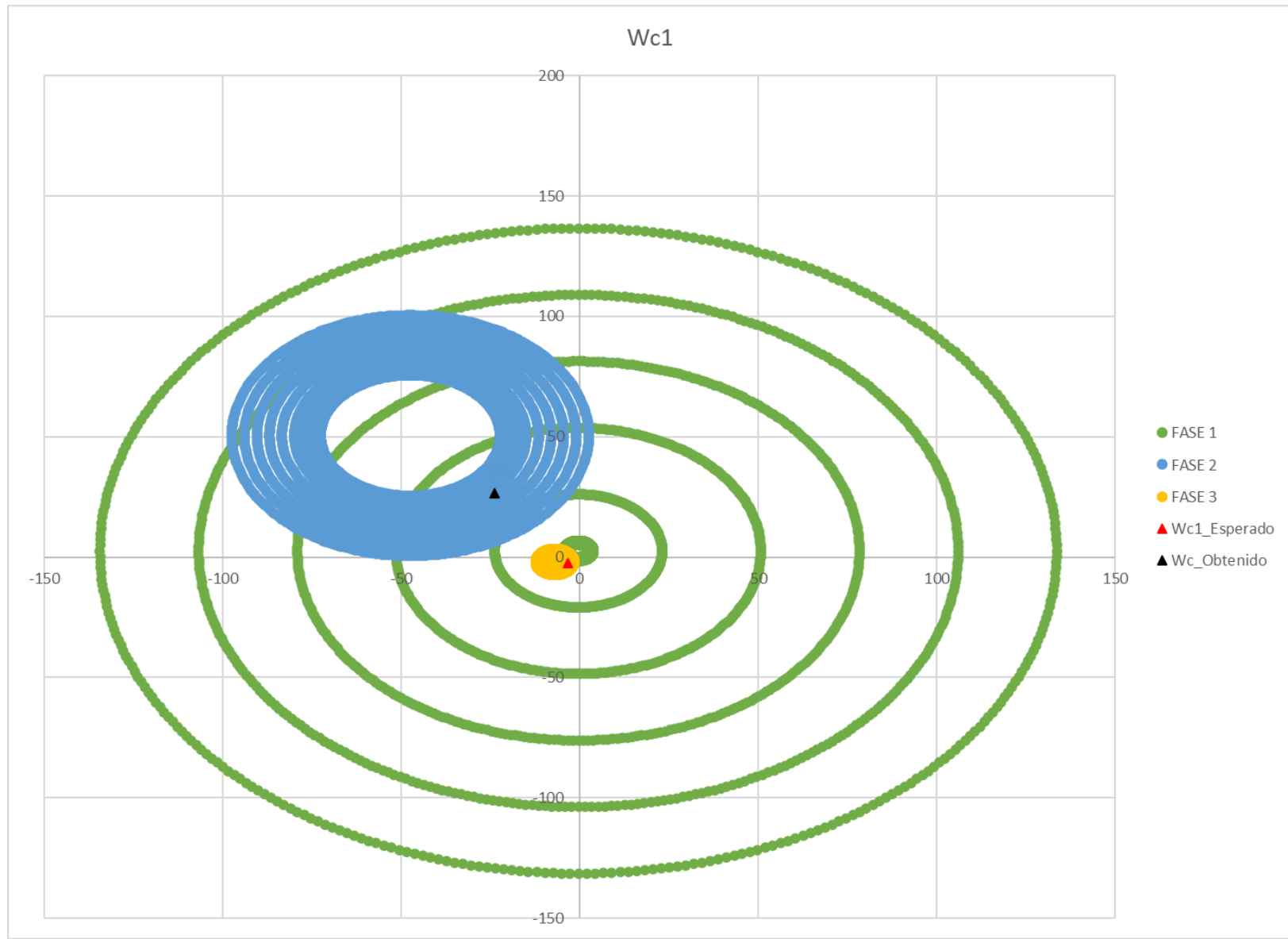


Figura 43 Comportamiento de las fases para Wc2, ángulos iguales con peso de prueba 10g a 30 Hz

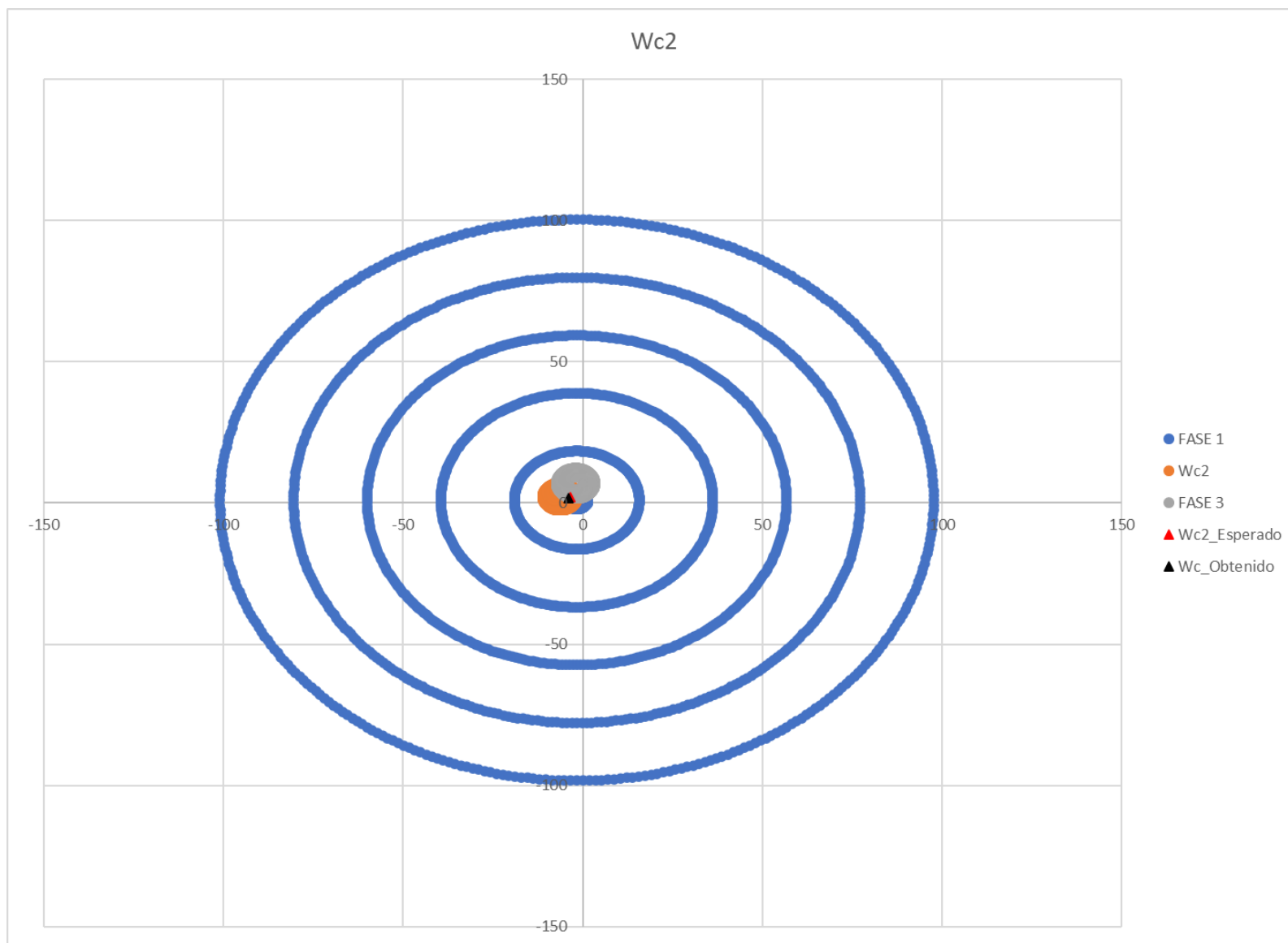


Tabla 15 Valores de W_c para ángulos iguales con 10 g a 30 Hz

Plano 1				Plano 2			
calculado		algoritmo		calculado		algoritmo	
W_{c1r}	W_{c1i}	W_{c1r}	W_{c1i}	W_{c2r}	W_{c2i}	W_{c2r}	W_{c2i}
-3.3499	-2.2499	23.9241	26.5692	-3.3449	2.2499	-3.9737	1.642

Los datos generados por el algoritmo para el primer plano exponen un aumento de 20.57 del valor de W_{c1r} , mientras que para W_{c1i} el incremento es aún mayor con 24.3193.

Sin embargo, para el segundo plano las cifras muestran una mejora en su comportamiento, debido a que W_{c1r} presenta una elevación de 118.79%, mientras que W_{c1i} provee de una exactitud del 72.98% del valor esperado.

El archivo corfa9 muestra el comportamiento de las fases de alimentación del motor respecto a los valores de W_{c1} y W_{c2} representados por la **Figura 44** y la **Figura 45**, respectivamente.

En la **Figura 45** se expone el comportamiento de las tres fases de alimentación, en donde se crea una intersección cercana al valor esperado de W_c , de esta forma se propone a W_{c2} como el valor óptimo para el balanceo en dos planos.

Figura 44 Comportamiento del offset para W_{c1} promedio, ángulos iguales con peso de prueba 10g a 30 Hz

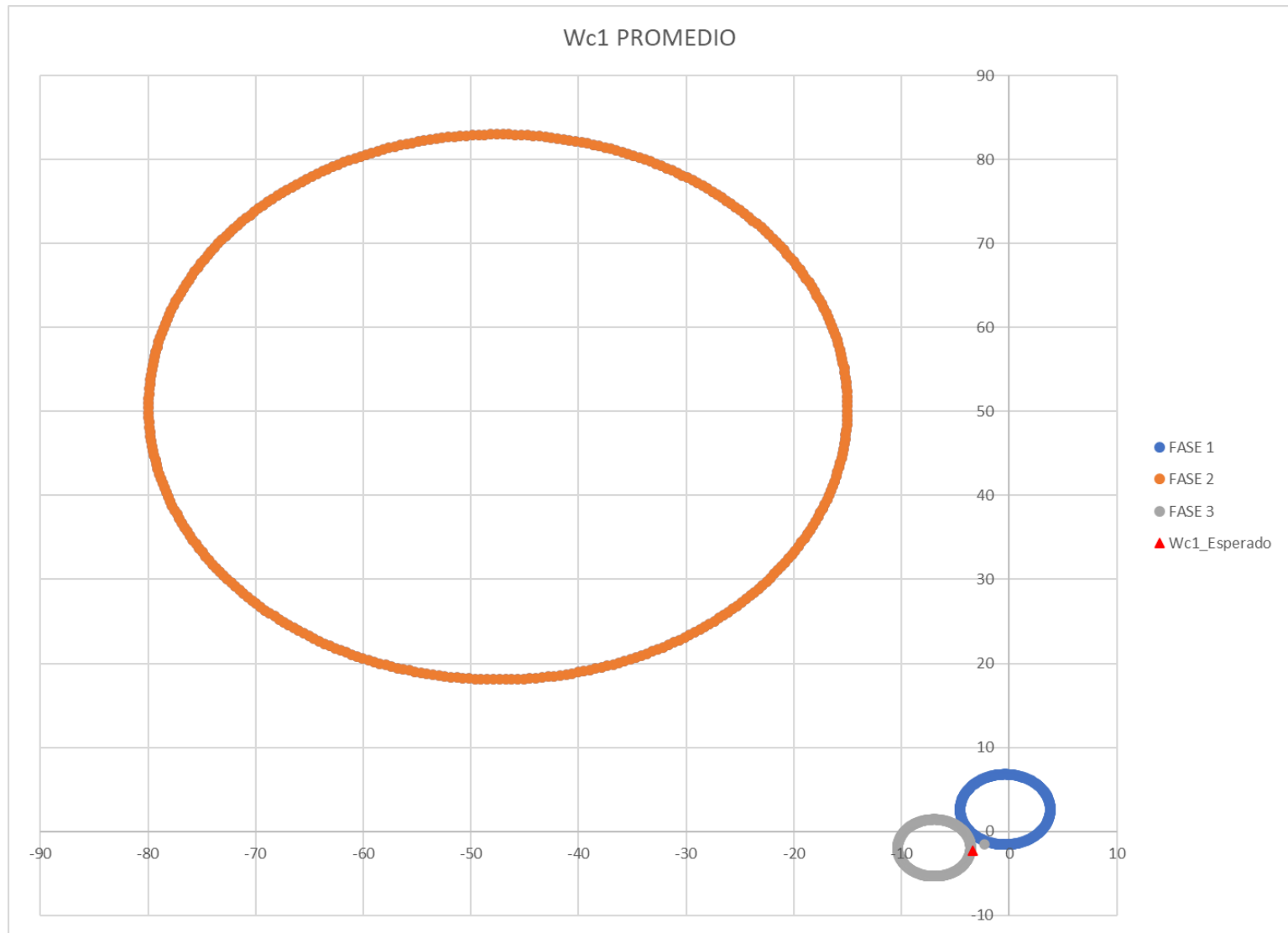
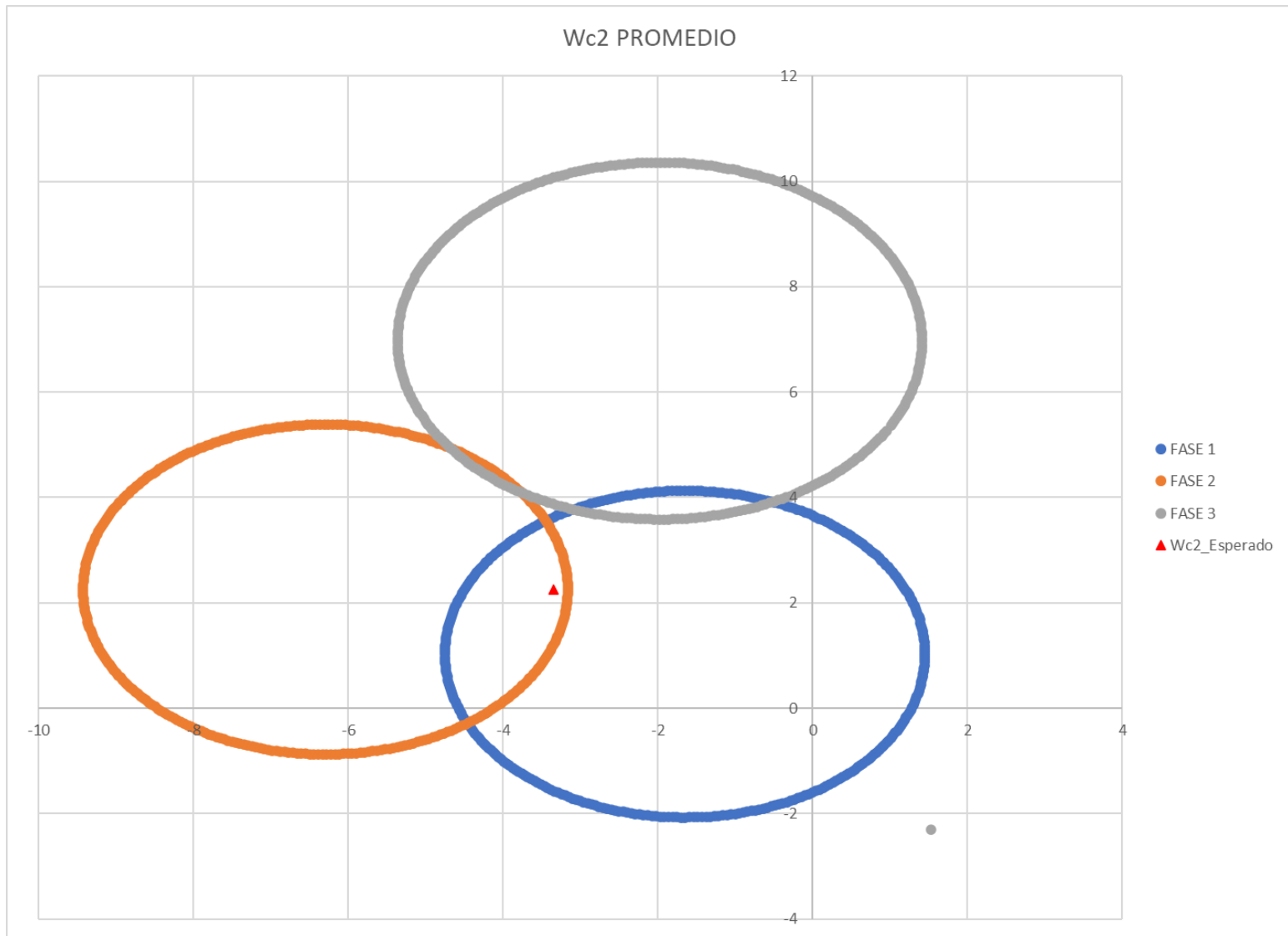


Figura 45 Comportamiento del offset para Wc2, ángulos iguales con peso de prueba 10g a 30 Hz



4.3 Balanceo dinámico implementando armónicas de 90 Hz

La tesis implica la evaluación de las armónicas de 30 y 90 Hz, debido a que sus fasores se mantienen constantes sin importar el desbalance mecánico presente en el equipo.

4.3.1 Balanceo en un plano con peso de prueba de 6.7g

Para el desarrollo de la metodología implementando armónicas de 90 Hz, involucra la modificación del rango de evaluación del “offset” a 200. De igual forma, se utilizará el algoritmo 8H16, encargado de generar los archivos corrfa’s, el cual nos brindará la información necesaria para la evaluación del método.

En los archivos corrfa3, corrfa4 y corrfa5 mostrará el comportamiento de las fases respecto al cálculo de W_c , para 90Hz exhibe un comportamiento homogéneo.

Una manera precisa de comprobar el comportamiento de las fases es mediante los resultados de los indicadores j_y del archivo corrfa7. Para este caso los indicadores j_y en las tres fases proporcionan un valor de 0, lo cual determina la exactitud de los datos en las fases. Asimismo, el archivo corrfa7, proporcionara los valores de W_c calculados, como se muestran en la **Tabla 16**

Tabla 16 Valores de W_c con peso de prueba de 6.7g a 90Hz

Calculado		Generado por el algoritmo	
W_{c1r}	W_{c1i}	W_{c1r}	W_{c1i}
-6.7	0	-6.7734	0.2188

Estos resultados muestran un comportamiento aceptable para el cálculo de balanceo, ya que las diferencias presentes en W_{c1r} es de 0.0734 indicando una exactitud de 101.09% del valor real, mientras que para W_{c1i} es de 0.21. Creando un respaldo sobre el método ya que sus resultados presentan mayor exactitud a su contraparte de 30Hz.

Conjuntamente, el archivo corrfa9 muestra el comportamiento del offset de las fases de alimentación del motor

Figura 46 Comportamientos de las fases respecto a $Wc1$, 6.7g a 90 Hz

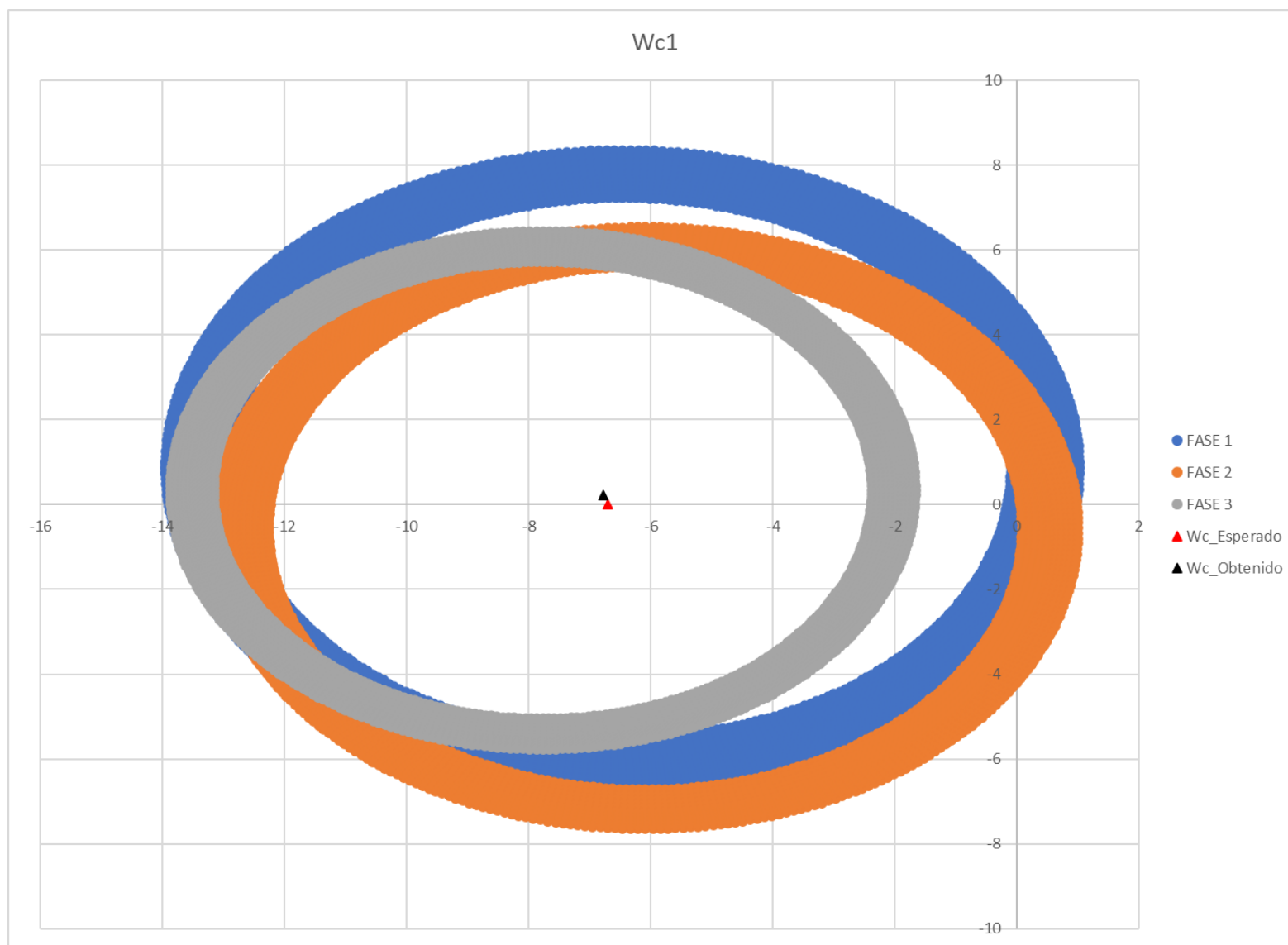
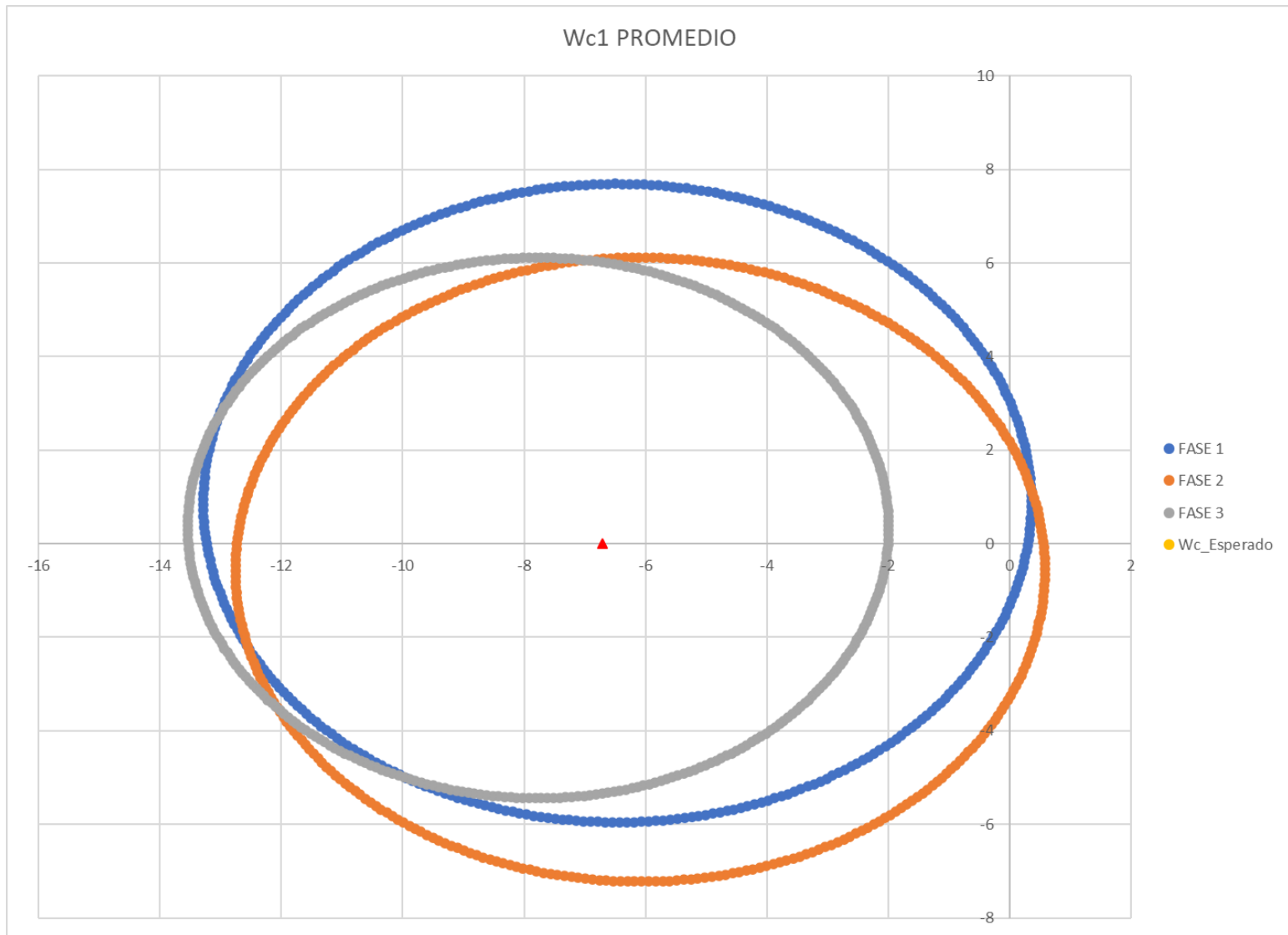


Figura 47 Comportamiento de W_{c1} promedio para 6.7g a 90 Hz



4.3.2 Balanceo en un plano con peso de prueba de 8.2g

Para el peso de desbalance de prueba con 8.2g, se implementaron los archivos corffa3, corffa4 y corffa5, encargados de desmostar el comportamiento de las fases respecto al cálculo de W_c , representados por la **Figura 48**

En los archivos corffa3, corffa4 y corffa5 mostrará el comportamiento de las fases respecto al cálculo de W_c , para 90Hz exhibe un comportamiento homogéneo.

Una manera precisa de comprobar el comportamiento de las fases es mediante los resultados de los indicadores j_y del archivo corffa7. Para este caso los indicadores j_y en las tres fases proporcionan un valor de 0, lo cual determina la exactitud de los datos en las fases. Asimismo, el archivo corffa7, proporcionara los valores de W_c calculados, como se muestran en la **Tabla 17**

Tabla 17 Valores de W_c con peso de prueba de 8.2g a 90 Hz

Calculado		Generado por el algoritmo	
W_{c1r}	W_{c1i}	W_{c1r}	W_{c1i}
0	-8.2	0.0844	-8.1715

Los resultados generados por el algoritmo muestran un comportamiento aceptable para el cálculo de balanceo, ya que las diferencias presentes en W_{c1r} es de -0.0285 indicando una exactitud de 99.65% del valor real, mientras que para W_{c1i} es de 0.08. Los resultados presentan mayor exactitud a su contraparte de 30Hz.

Conjuntamente, el archivo corffa9 muestra un comportamiento casi concéntrico de las fases, esto se debe a que las configuraciones promedio de los datos de las fases son casi iguales. Por lo cual se deben cambiar los peso de desbalance.

Figura 48 Comportamiento de las fases respecto a W_c , para 8.2g a 90 Hz

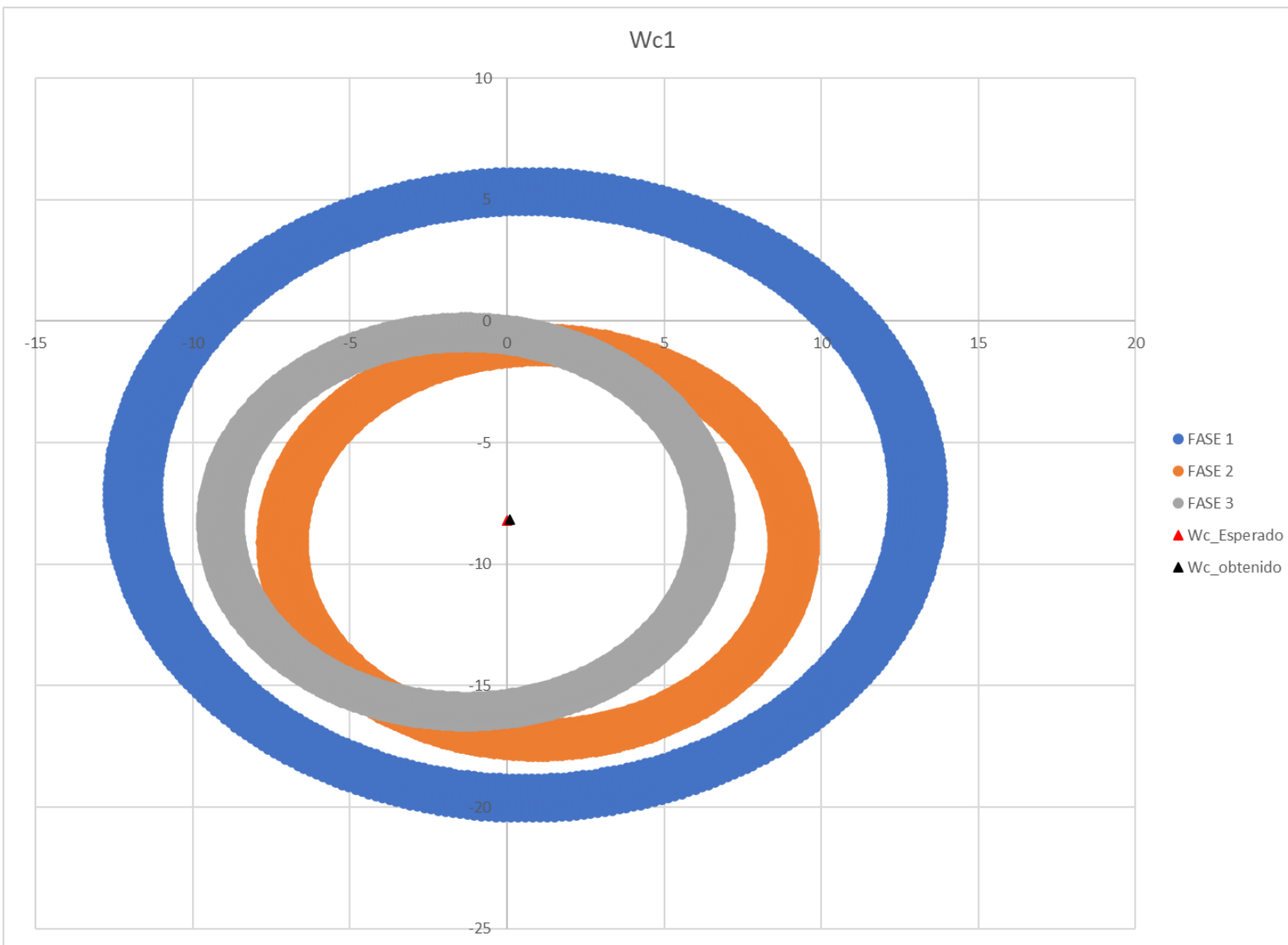
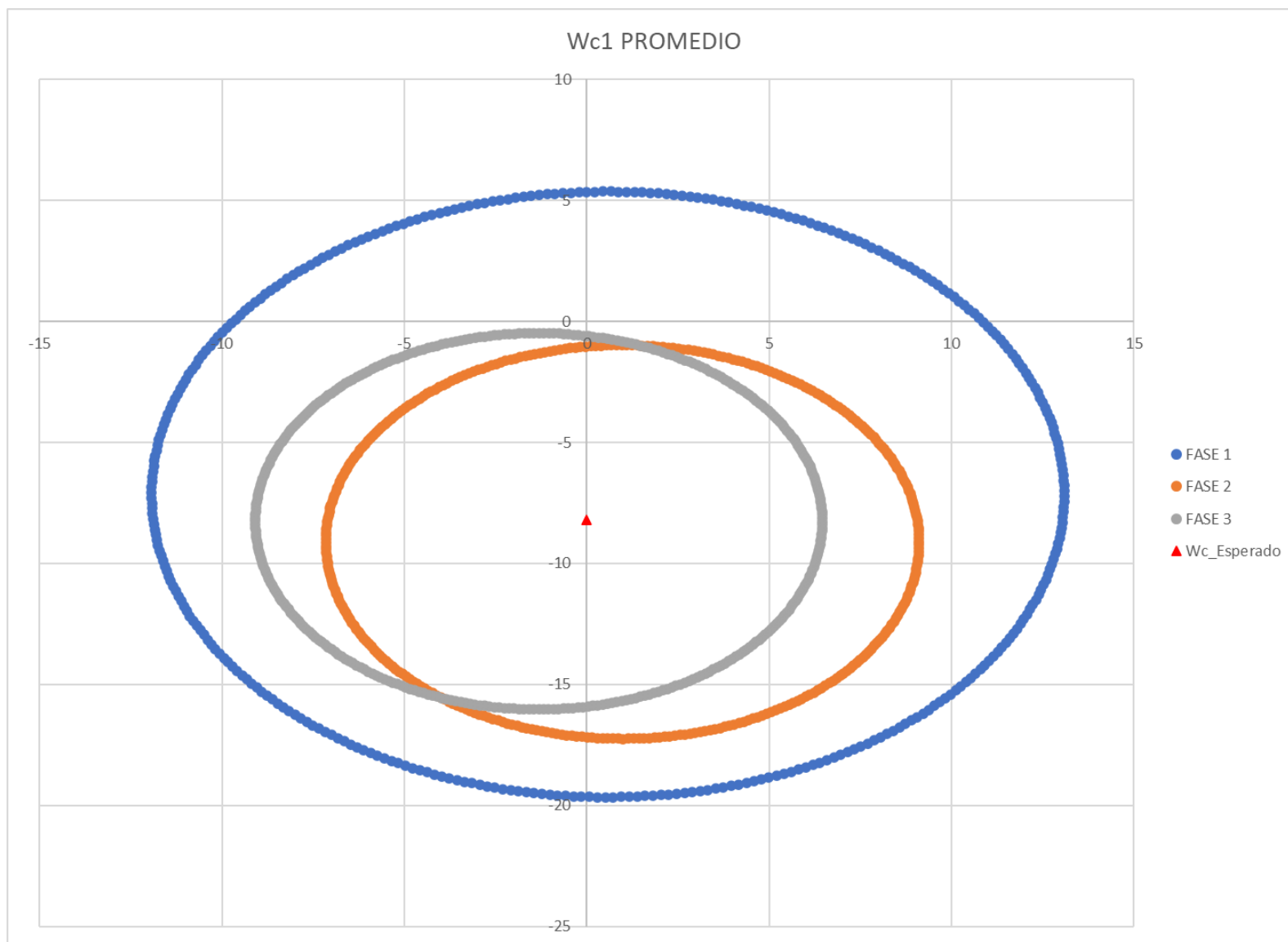


Figura 49 Comportamiento de W_{c1} promedio para 8.2g a 90 Hz



4.3.3 Balanceo en dos planos ángulos iguales con peso de prueba de 6.7g

El balanceo en dos planos implica la colocación de dos pesos de pruebas iguales (6.7g), ambos posicionados en ángulos iguales en el plano uno y dos, además de implementar el mismo algoritmo para el balanceo en un plano. Lo cual involucra emplear los archivos corrfa3, 4 y 5, encargados de proporcionar información del comportamiento de las fases respecto al cálculo de W_c .

El archivo corrfa7 proporciona datos para la interpretación del cálculo del valor correspondiente a cada fase, para este caso contará con dos planos.

La obtención de las fases en el plano 1, muestran un comportamiento homogéneo debido a que el indicador j_y proporcionan valores de 0 para las tres fases, esto señala que los valores de la fase son exactos, por lo cual la intersección de los círculos 1, 2 y 3, contendrán el valor del W_c en su unión, como se muestra en la **Figura 50**

Igualmente, para las fases del plano 2, proporciona indicadores j_y con valores de 0 demostrando un cálculo exacto del valor en las tres fases, representadas en la **Figura 51**, donde claramente se puede apreciar que el peso de W_{c1} es el valor buscado.

Asimismo, el archivo corrfa7 contará con una segunda función, el cual consiste el proporcionar los valores de W_c calculados, tanto para la parte imaginaria y real, como se desglosa en la **Tabla 18**

Figura 50 Comportamiento de las fases para W_{c1} , ángulos iguales con peso de prueba 6.7g a 90 Hz

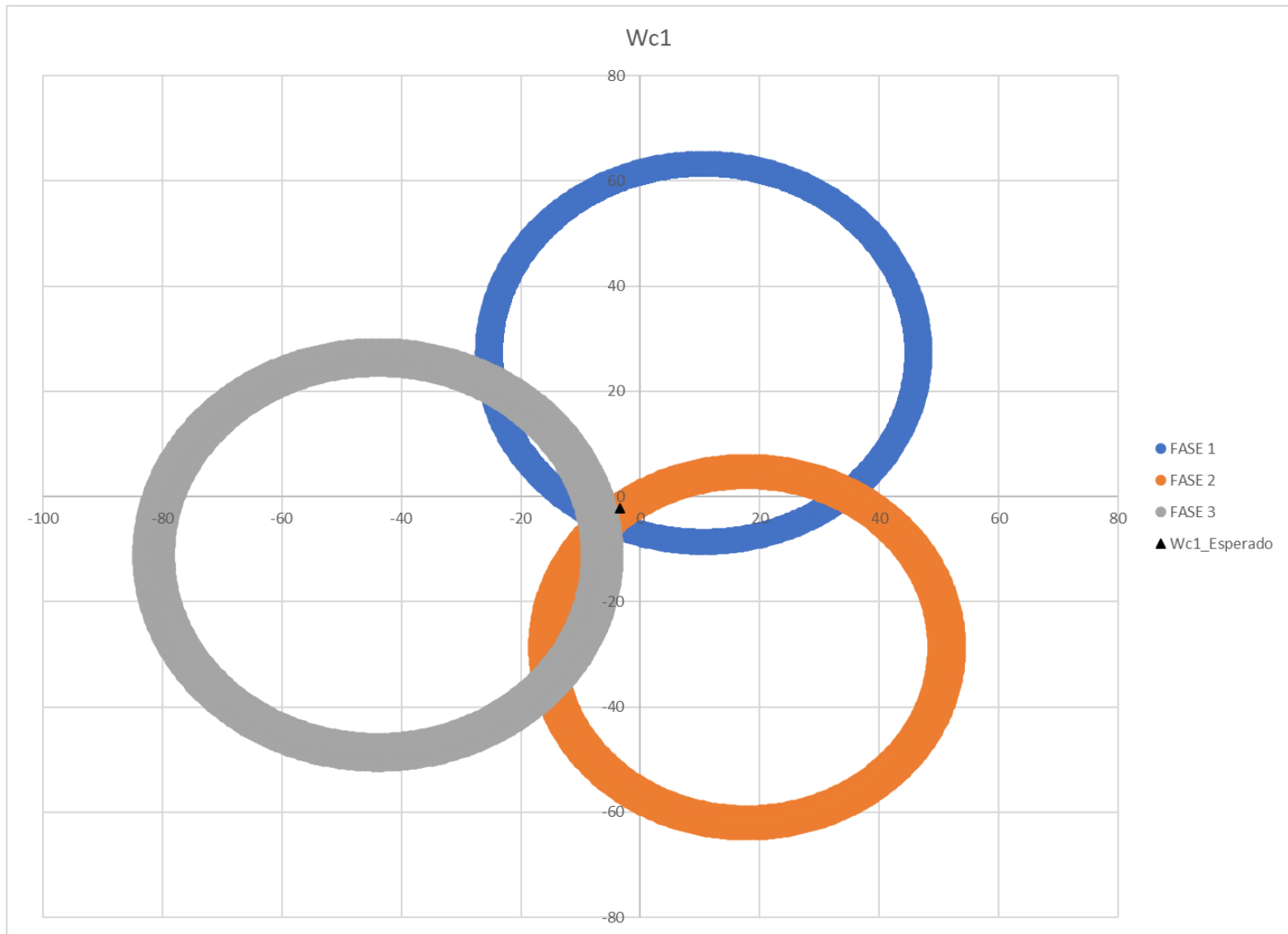
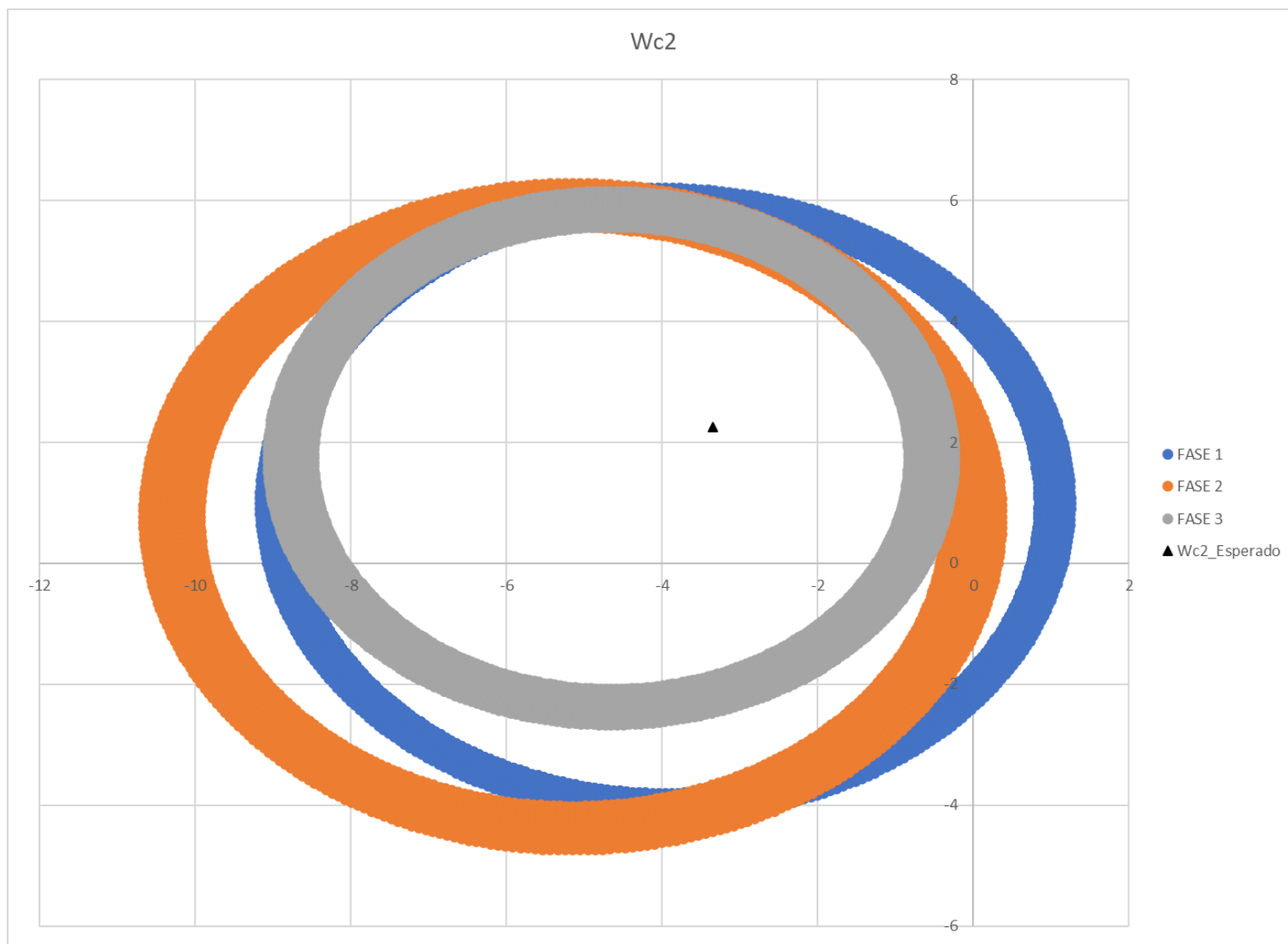


Figura 51 Comportamiento de las fases para $Wc2$, ángulos iguales con peso de prueba 6.7g a 90 Hz



El archivo corffa7 proporcionara los resultados calculados por el algoritmo para W_c .

Tabla 18 Valores de W_c con ángulos iguales con peso de prueba 6.7g a 90 Hz

Plano 1				Plano 2			
calculado		algoritmo		calculado		algoritmo	
W_{c1r}	W_{c1i}	W_{c1r}	W_{c1i}	W_{c2r}	W_{c2i}	W_{c2r}	W_{c2i}
-3.3499	-2.2499	-5.1402	-4.1277	-3.3449	2.2499	-4.5869	1.1694

Los datos proporcionados para el balanceo en dos planos muestran las siguientes características: el primer plano presenta una diferencia de 1.7903 con exactitud de 153.44% del valor de W_{c1r} , mientras que para W_{c1i} el incremento es de 1.8728 proporcionando una exactitud de 183.346%.

Sin embargo, para el segundo plano las cifras disminuyen considerablemente, ya que W_{c1r} presenta una diferencia de 1.242 indicando una exactitud de 136.92%, mientras que W_{c1i} origina una diferencia significativa de -1.0805 demostrando una exactitud 51.97%.

Además, el archivo corffa9 muestra el comportamiento de las fases de alimentación del motor respecto a los valores de W_{c1} y W_{c2} representados por la **Figura 52** y la **Figura 53** respectivamente.

De acuerdo con la **Figura 52**, constituye la formación de los círculos de manera homogénea, creando una intersección en la cual se aprecia claramente el valor de W_c esperado, siendo W_{c1} el valor aproximado óptimo para el balanceo en dos planos con ángulos iguales.

Figura 52 Comportamiento de W_{c1} promedio, ángulos iguales con peso de prueba 6.7g a 90 Hz

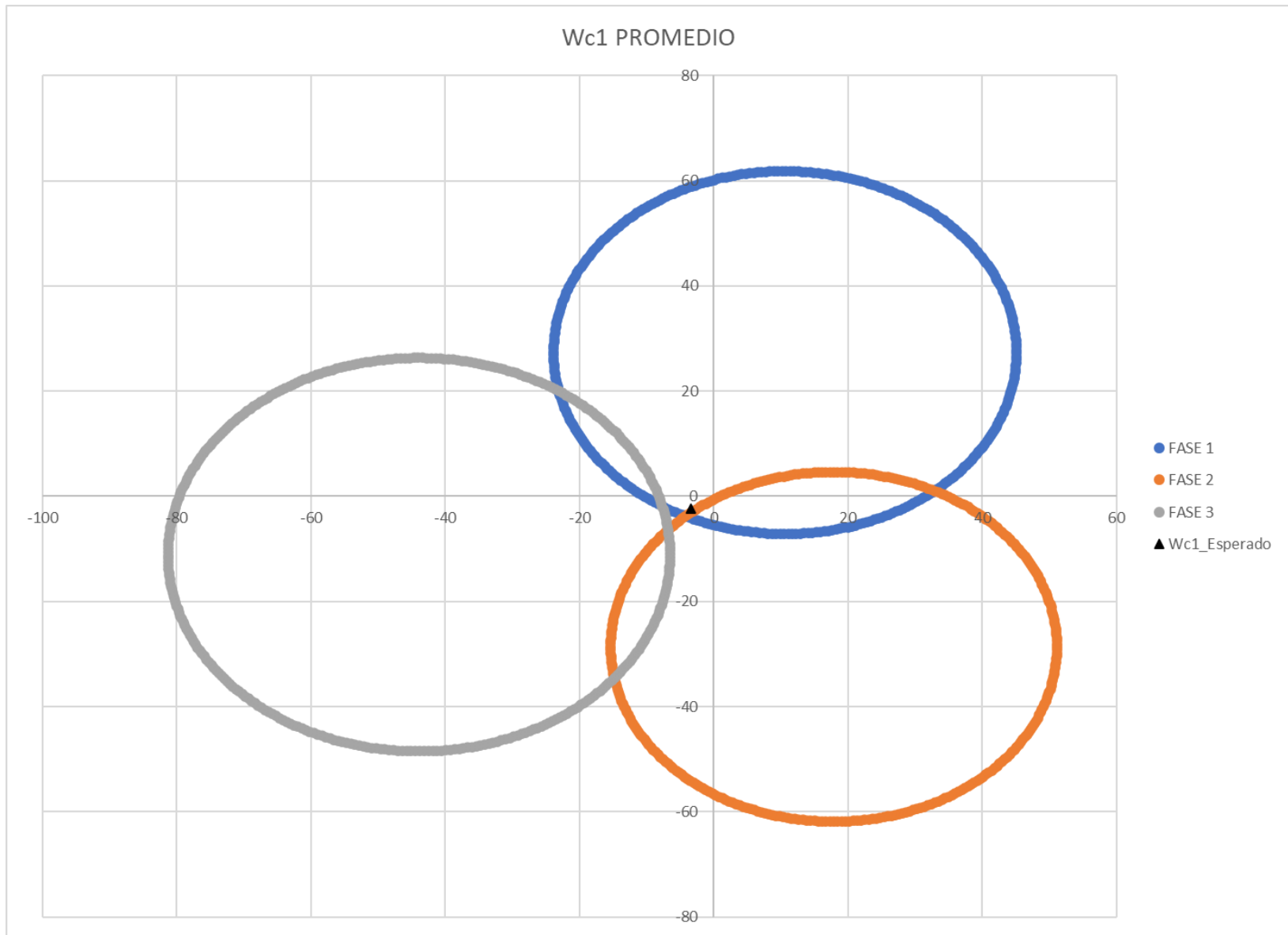
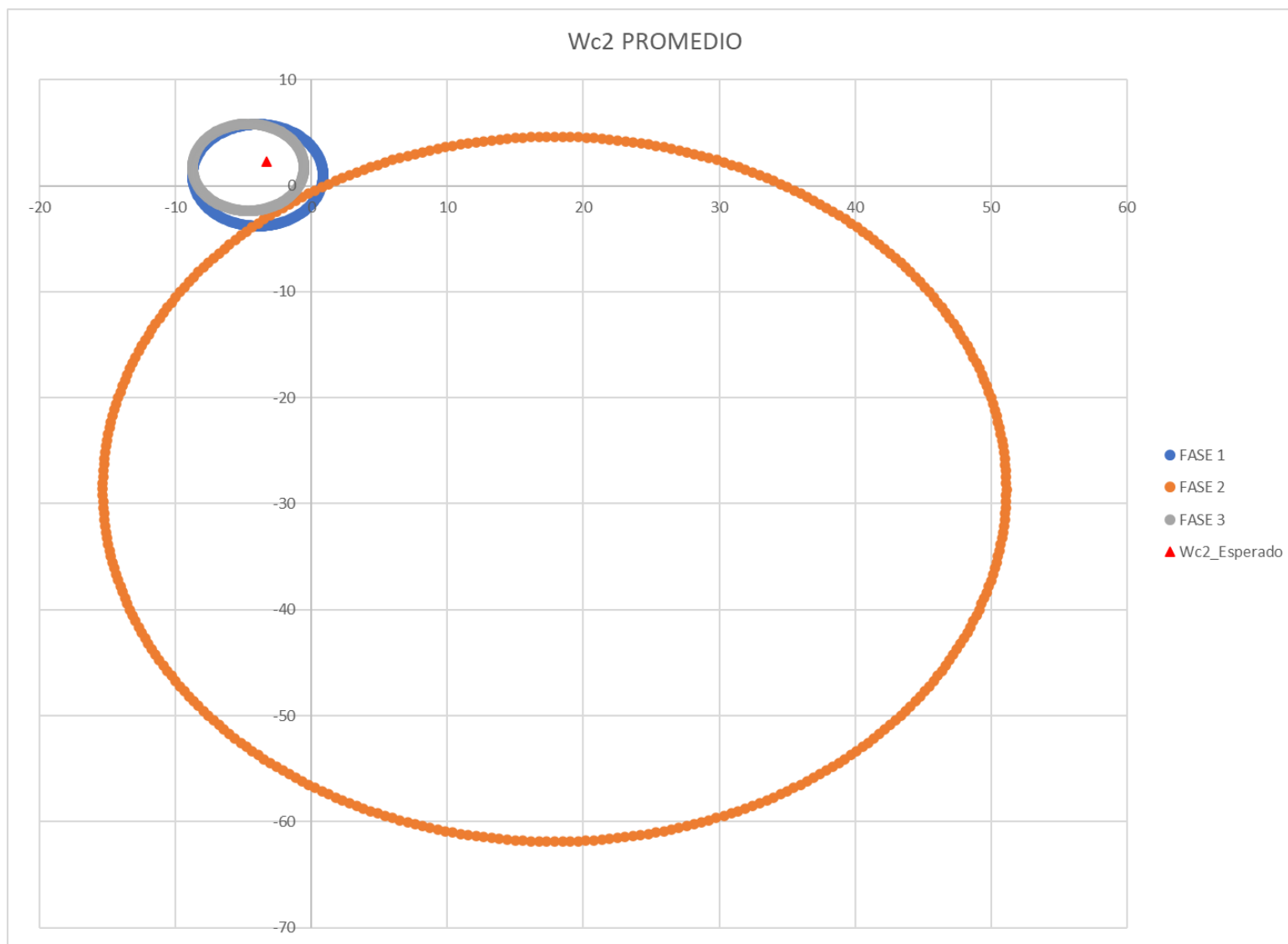


Figura 53 Comportamiento de Wc2 promedio, ángulos iguales con peso de prueba 6.7g a 90 Hz



4.3.4 Balanceo en dos planos ángulos distintos con peso de prueba de 6.7g

El balanceo en dos planos con ángulos distintos implica la colocación de dos pesos de pruebas con la misma magnitud (6.7g), ambos posicionados en ángulos diferentes en el plano uno y dos, además de implementar el mismo algoritmo para el balanceo en un plano. Por lo cual utiliza los archivos corrfa3, 4 y 5, encargados de proporcionar información del comportamiento de las fases respecto al cálculo de W_c .

El archivo corrfa7 proporciona datos para la interpretación del cálculo del valor correspondiente a cada fase, para este caso contará con dos planos.

La obtención de las fases en el plano 1, muestran un comportamiento homogéneo debido a que el indicador j_y proporcionan valores de 0 para las tres fases, esto señala que los valores de la fase son exactos, por lo cual la intersección de los círculos 1, 2 y 3, contendrán el valor del W_c en su unión, como se muestra en la **Figura 54**.

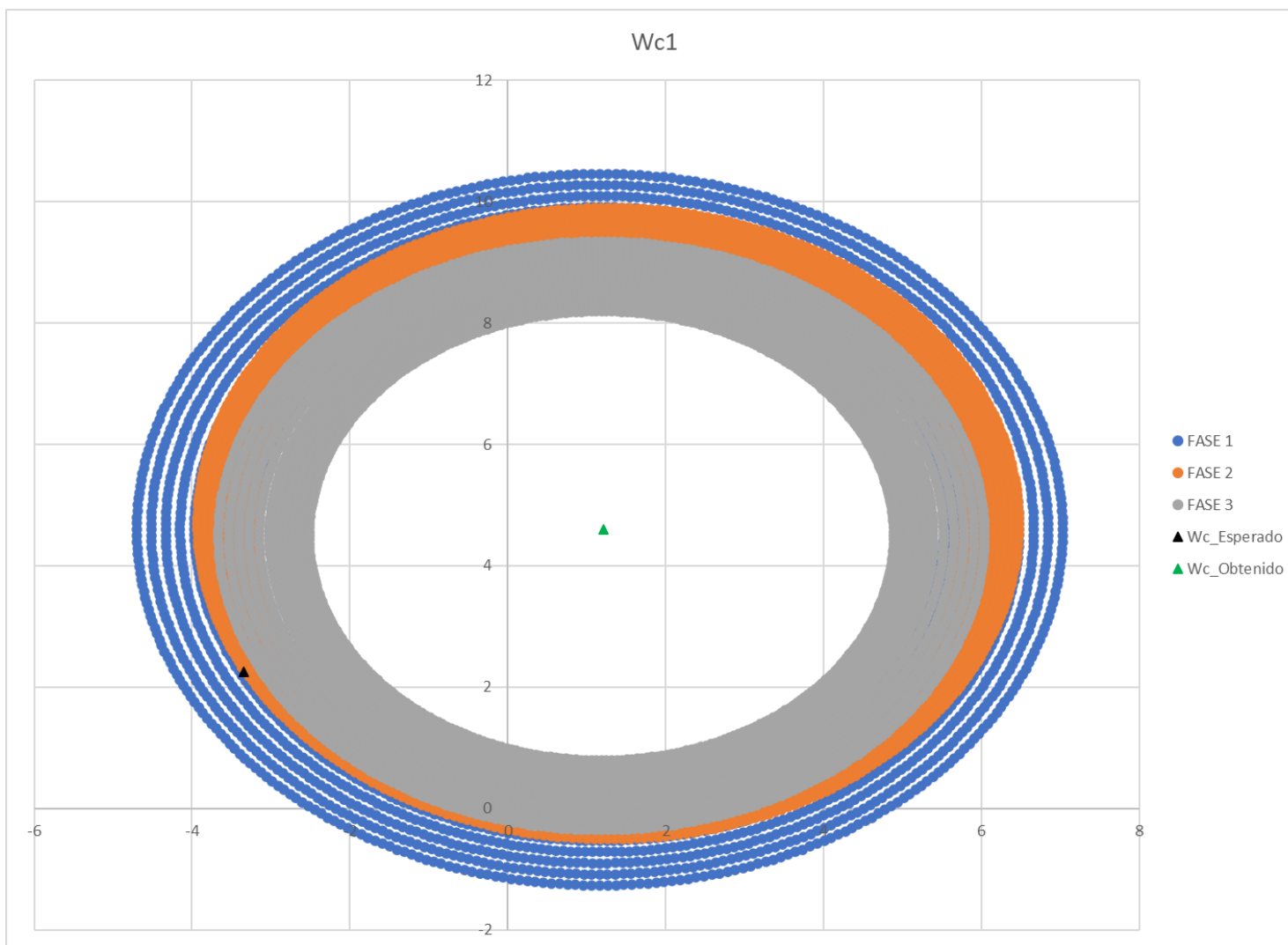
Asimismo, el archivo corrfa7 contará con una segunda función, el cual consiste el proporcionar los valores de W_c calculados, tanto para la parte imaginaria y real, como se desglosa en la **Tabla 19**

Tabla 19 Valores de W_c , ángulos distintos con peso de prueba de 6.7g a 90 Hz

Calculado		Algoritmo	
W_{c1r}	W_{c1i}	W_{c1r}	W_{c1i}
-3.3449	2.2499	1.120	4.595

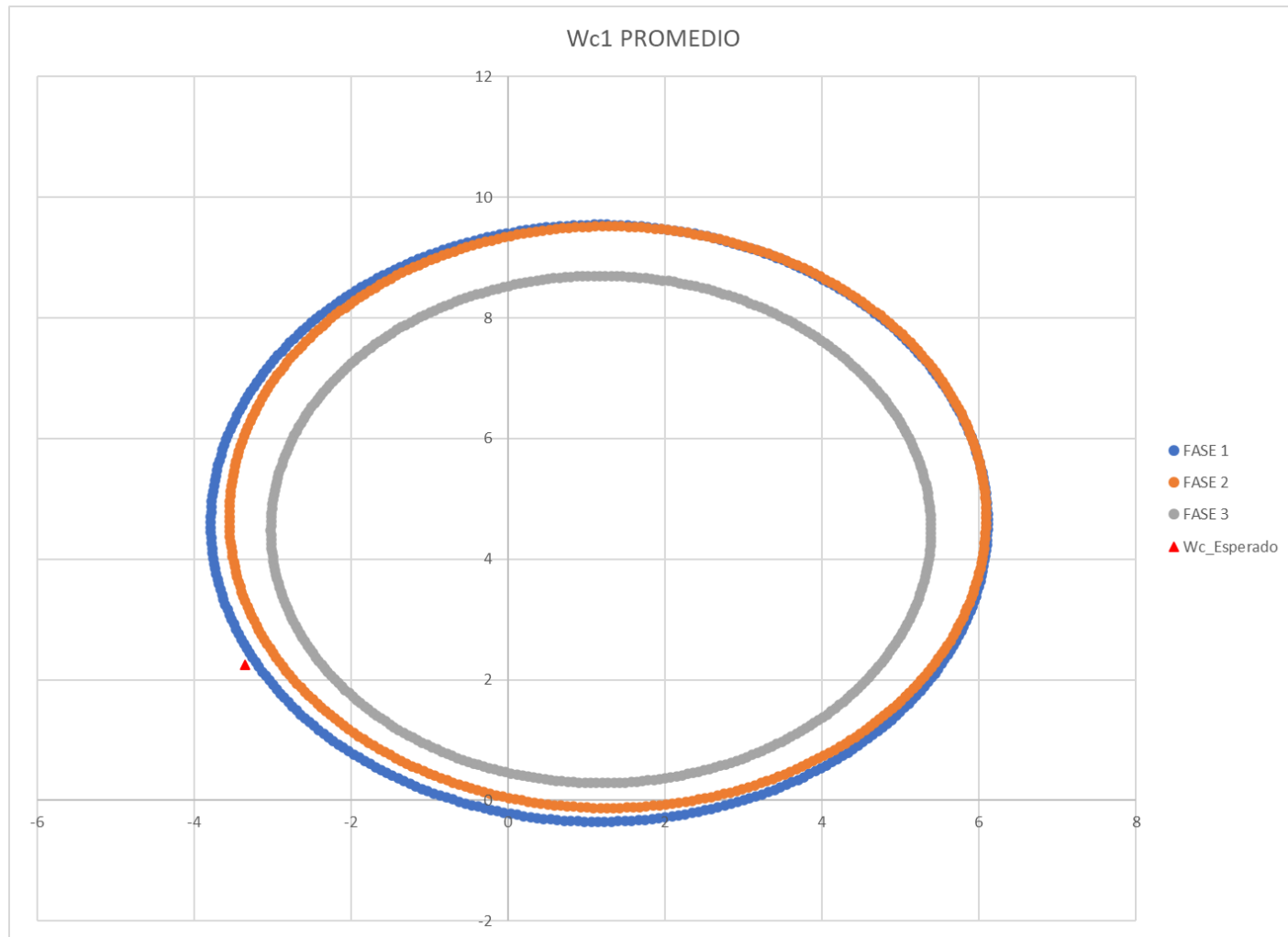
Los datos muestran una discrepancia de -2.2249 para W_{c1r} . Mientras que para W_{c1i} existe una diferencia del doble del valor real 204.23%. Exponiendo un error amplio para su evaluación.

Figura 54 Comportamiento de las fases para $Wc1$, ángulos distintos con peso de prueba 6.7g a 90 Hz



Mientras que el archivo corffa9 muestra la formación de círculos casi concreticos, indicando que las configuraciones promedio de los datos de cada fase son casi iguales. Proponiendo el cambio de magnitud para los pesos de desbalance.

Figura 55 Comportamiento de $Wc1$ promedio, ángulos distintos con peso de prueba 6.7g a 90 Hz



4.3.5 Balanceo en dos planos ángulos iguales con peso de prueba de 10g

Esta prueba consistió en agregar un peso de desbalance mayor, con la finalidad de comprobar la respuesta de los fasores debido a la alteración de la magnitud. El balanceo en dos planos a 90 Hz posiciona dos pesos de pruebas iguales en magnitud y ángulo.

El algoritmo genera los archivos corrfa3, 4 y 5, encargados de brindar información del comportamiento de las fases respecto al cálculo de W_c .

El archivo corrfa7 proporciona datos para la interpretación del cálculo del valor correspondiente a cada fase, para este caso contará con dos planos.

La obtención de las fases en el plano 1, muestran un comportamiento homogéneo para las fases 1, 2 y 3 debido a que el indicador j_y proporcionan valores de 0, esto indica que la intersección de los círculos contendrá el valor del W_c , como se muestra en la **Figura 56**, el cual evidencia a W_{c1} como el valor buscado.

Para las fases en el plano 2, muestra un comportamiento homogéneo para las fases 1, 2 y 3 debido a que el indicador j_y proporcionan valores de 0, mostrado por la **Figura 57**.

Asimismo, el archivo corrfa7 contará con una segunda función, el cual consiste en proporcionar los valores de W_c , desglosado en la **Tabla 20**.

Figura 56 Comportamiento de las fases para W_{c1} , ángulos iguales con peso de prueba 10g a 90 Hz

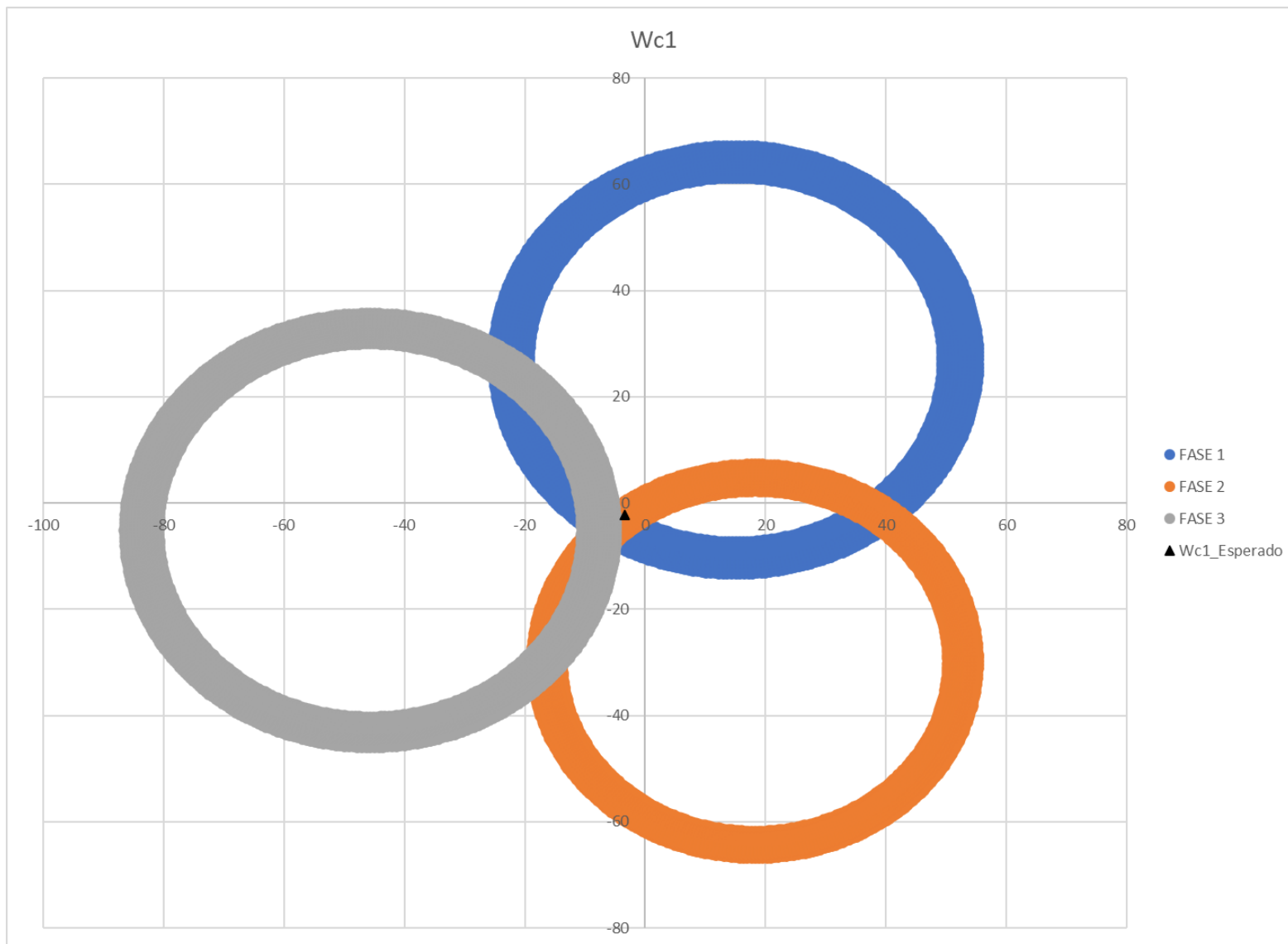


Figura 57 Comportamiento de las fases para $Wc2$, ángulos iguales con peso de prueba 10g a 90 Hz

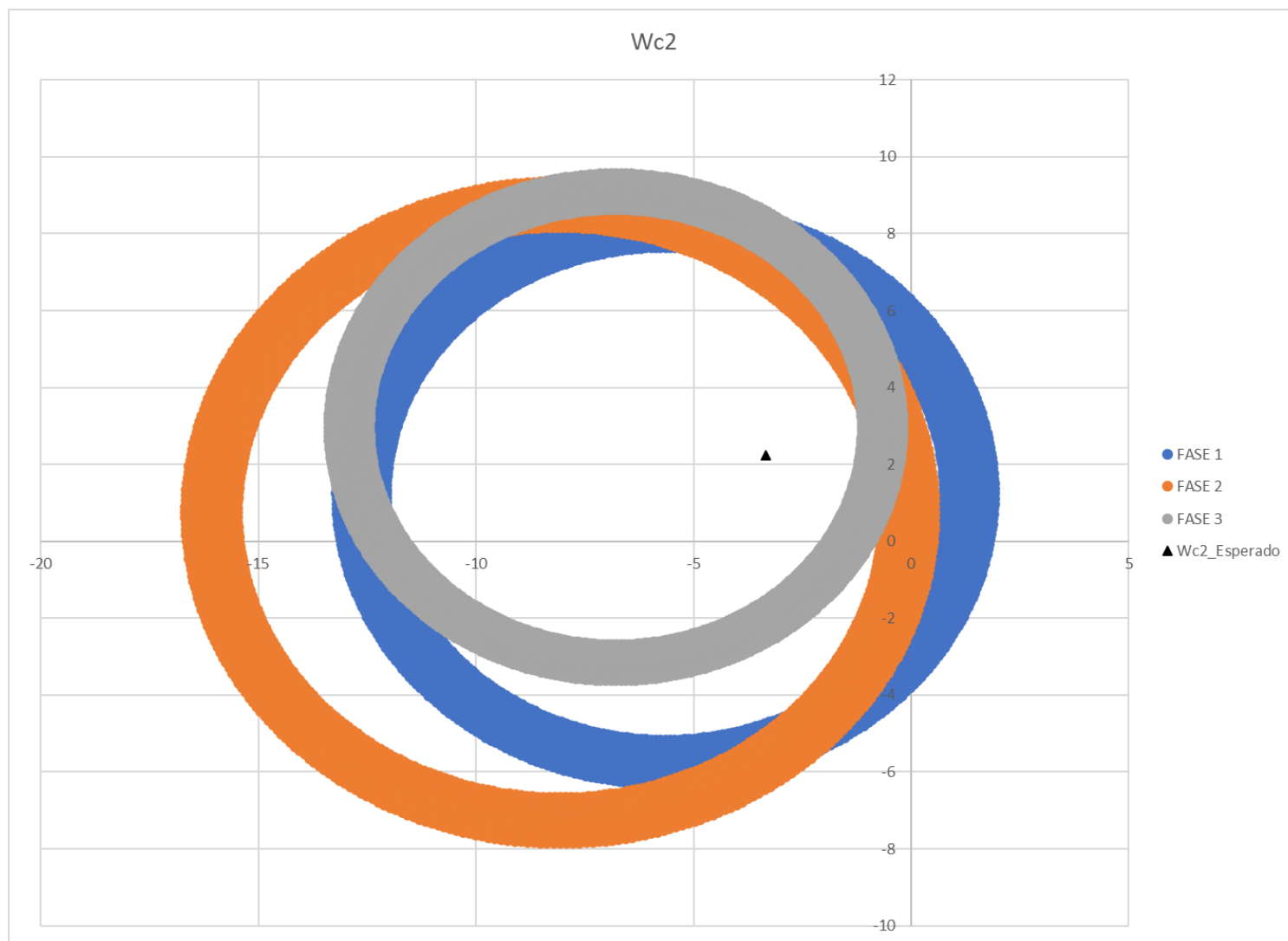


Tabla 20 Valores de W_c , ángulos distintos con peso de prueba de 10g a 90 Hz

Plano 1				Plano 2			
calculado		algoritmo		calculado		algoritmo	
W_{c1r}	W_{c1i}	W_{c1r}	W_{c1i}	W_{c2r}	W_{c2i}	W_{c2r}	W_{c2i}
-3.3499	-2.2499	-4.1058	-2.647	-3.3449	2.2499	-6.8321	1.6526

Los datos obtenidos del algoritmo para el primer plano exponen una divergencia de 0.7559, creando así una exactitud de 122.56% del valor real de W_{c1r} , mientras que W_{c1i} muestra una diferencia menor de 0.3971 con exactitud de 117.64% el valor esperado.

Sin embargo, para el segundo plano las cifras muestran un aumento significativo en W_{c1r} duplicando su valor con 3.4872, propiciando una desigualdad de 203.94%. Mientras que W_{c1i} exhibe disminución de 0.5973 creando una exactitud del 73.45% del valor original.

Conjuntamente, el archivo corrfa9 muestra los comportamientos de W_{c1} y W_{c2} promedio en las fases de alimentación del motor, a través de la **Figura 58** y la **Figura 59**.

Donde la **Figura 58** modela la intersección de los tres círculos de cada fase, mostrando el valor de W_c esperado, proporcionando a W_{c1} como la opción viable para la implementación del balanceo en dos planos ángulos iguales.

Figura 58 Comportamiento de W_{c1} promedio, ángulos iguales con peso de prueba 10g a 90 Hz

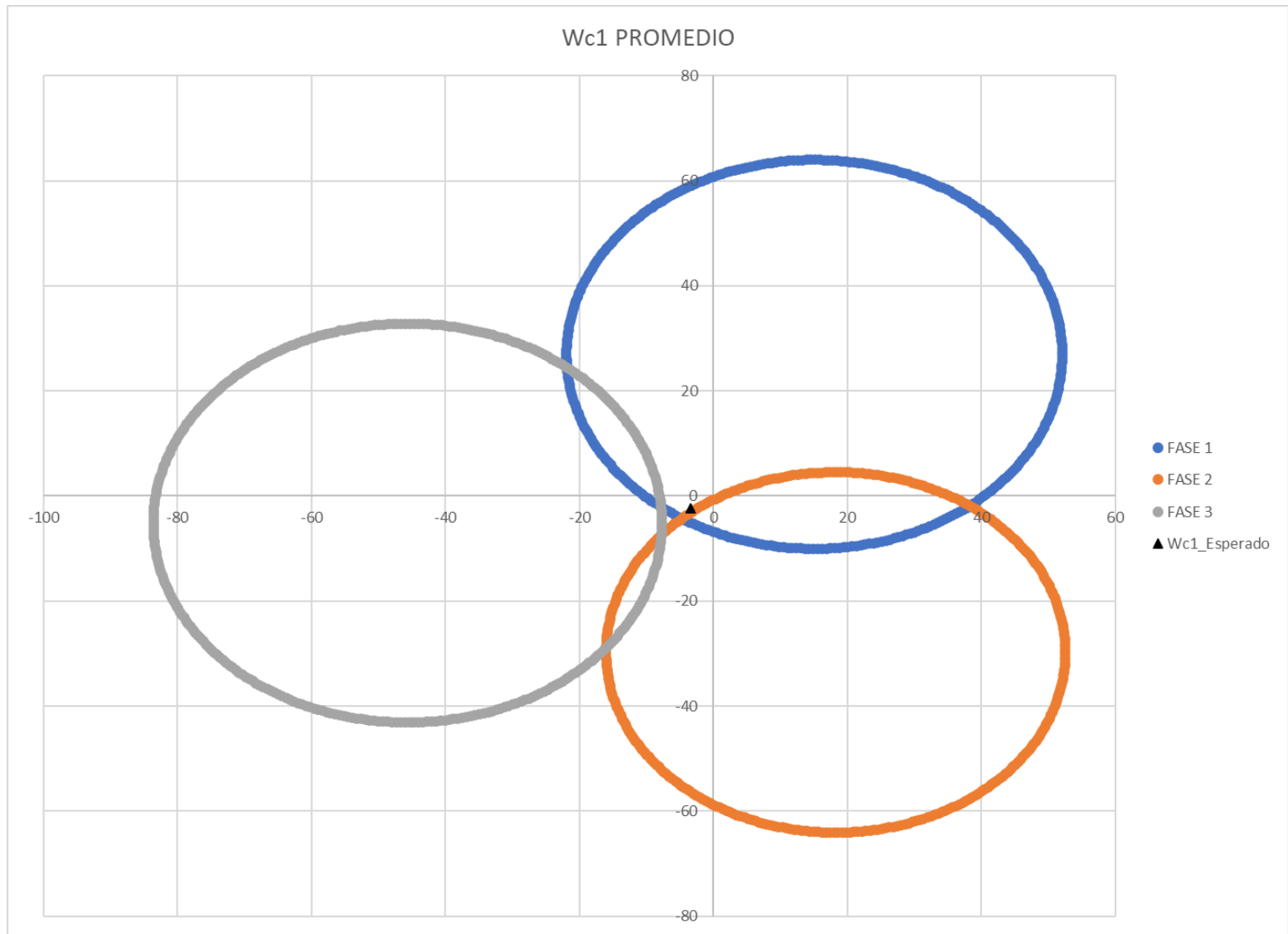
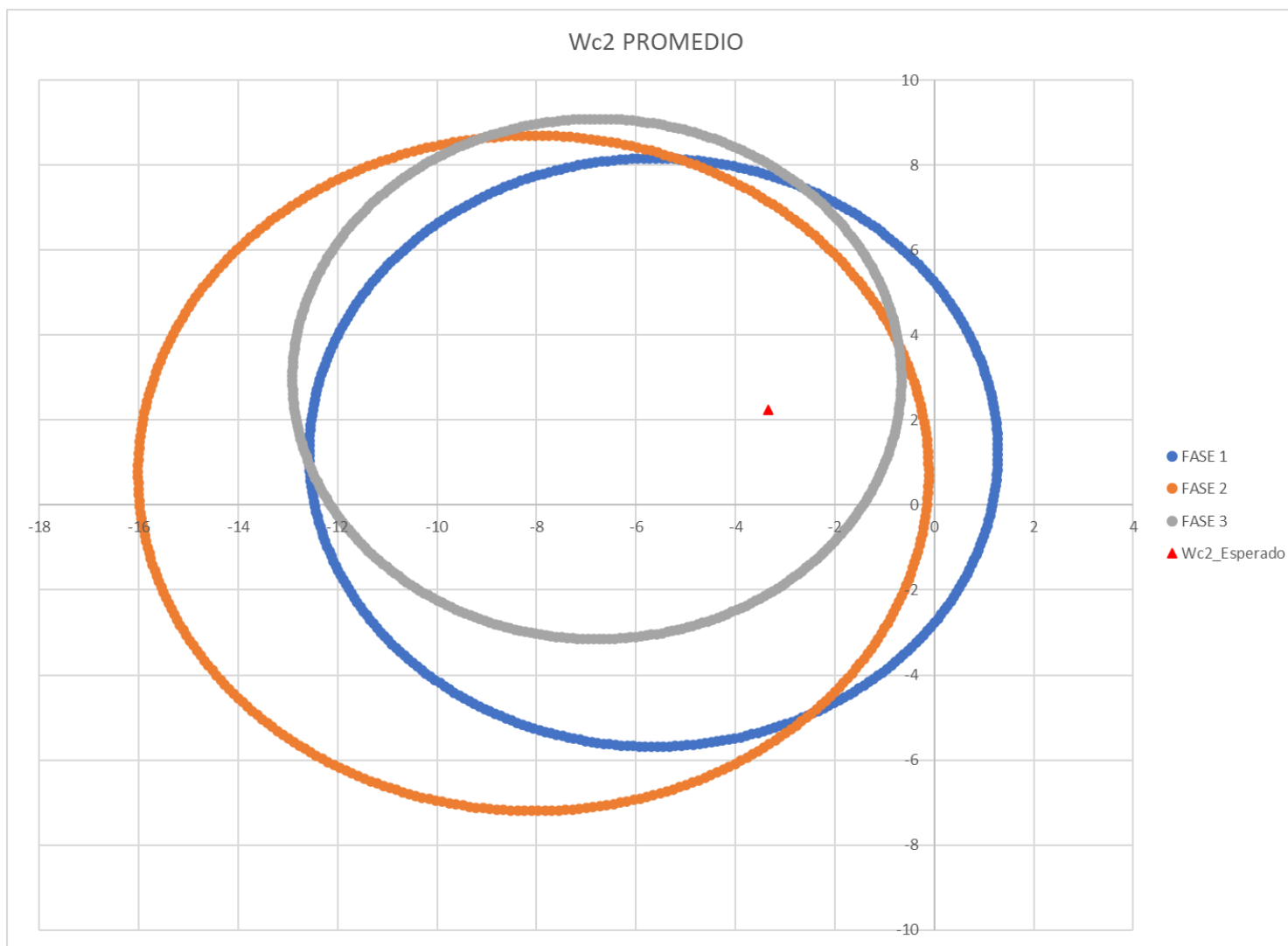


Figura 59 Comportamiento de W_{c2} promedio, ángulos iguales con peso de prueba 10g a 90 Hz



4.4 Discusión de resultados.

Esta sección muestra y compara los resultados obtenidos de cada caso propuesto en la metodología del balanceo dinámico de un rotor acoplado a un motor de inducción mediante armónicas de 30 y 90 Hz de la corriente residual.

Tabla 21 Resultados del balanceo en un plano a 30 Hz

Peso de desbalance	Esperado		Algoritmo		Diferencia	
	W_{c1r}	W_{c1i}	W_{c1r}	W_{c1i}	W_{c1r}	W_{c1i}
6.7 g	-6.7	0	-5.8025	1.5914	0.8975	1.5914
8.2 g	0	-8.2	-1.9113	-8.7808	-1.9113	-0.5808

Tabla 22 Resultados del balanceo en un plano a 90 Hz

Peso de desbalance	Esperado		Algoritmo		Diferencia	
	W_{c1r}	W_{c1i}	W_{c1r}	W_{c1i}	W_{c1r}	W_{c1i}
6.7 g	-6.7	0	-6.7734	0.2188	-0.0734	0.2188
8.2 g	0	-8.2	0.0844	-8.1715	0.0844	0.0285

Para el balanceo en un plano con peso de desbalance de 6.7g, el resultado generado por el algoritmo 8H16 utilizando datos de 90 Hz, presenta un rango menor de error: en donde W_{c1r} muestra un exceso de 0.07 de la magnitud colocada, de igual forma en W_{c1i} el rango de error es menor ya que excede en 0.02 del valor inicial. Los fasores de 90Hz proporcionan una exactitud superior al 100% del valor original.

Igualmente, para los cálculos con pesos de prueba de desbalance de 8.2g a 90 Hz, expone un resultado similar para valores de W_{c1r} y W_{c1i} , en donde exhibe una pequeña elevación de los datos obtenidos de 0.08 y 0.02 respectivamente sobre el valor original.

Tabla 23 Resultados del balanceo en dos planos a 30 Hz

Pesos de desbalance	Plano 1						Plano 2					
	Esperado		Algoritmo		Diferencia		Esperado		Algoritmo		Diferencia	
	Wc1r	Wc1i	Wc1r	Wc1i	Wc1r	Wc1i	Wc2r	Wc2i	Wc2r	Wc2i	Wc2r	Wc2i
6.7 g ángulos iguales	-3.3499	-2.2499	-4.4756	-4.3059	-1.1257	-2.056	-3.3449	2.2499	-3.7313	2.7007	-0.3864	0.4508
6.7 g ángulos distintos	-3.3449	2.2499	1.123	4.5078	4.4679	2.2579	-3.3449	2.2499	148.343	7.3394	151.6879	5.0895
10 g ángulos iguales	-3.3399	-2.2499	-23.9241	26.5692	-20.5842	28.8191	-3.3449	2.2499	-3.9737	1.642	-0.6288	-0.6079

Tabla 24 Resultados del balanceo en dos planos a 90 Hz

Pesos de desbalance	Plano 1						Plano 2					
	Esperado		Algoritmo		Diferencia		Esperado		Algoritmo		Diferencia	
	wc1r	wc1i	wc1r	wc1i	wc1r	wc1i	wc2r	wc2i	wc2r	wc2i	wc2r	wc2i
6.7 g ángulos iguales	-3.349	-2.249	-5.1402	-4.127	-1.790	-1.877	-3.3449	2.2499	-4.5869	1.1694	-1.242	-1.0805
6.7 g ángulos distintos	-3.344	2.249	1.2064	4.595	4.551	2.345	-3.3449	2.2499	-20.2602	-122.74	-16.915	-124.9948
10 g ángulos iguales	-3.339	-2.249	-4.1058	-2.647	-0.765	-0.397	-3.3449	2.2499	-6.8321	1.6526	-3.487	-0.5973

Para el balanceo en dos planos en ángulos iguales con peso de desbalance de 6.7g, el resultado generado por el algoritmo

8H16 utilizando datos de 30 Hz, presenta un rango menor de error: en donde W_{c1r} muestra un exceso de 1.12 de la magnitud colocada, asimismo, W_{c1i} tiene un excedente de 2.056 del valor inicial.

Igualmente, para los cálculos con ángulos distintos con pesos de prueba de desbalance de 6.7g a 30 Hz, expone un resultado similar para valores de W_{c1r} y W_{c1i} , en donde exhibe una pequeña elevación de los datos obtenidos de 4.46 y 2.25 respectivamente sobre el valor original.

Mientras que los resultados del balanceo en dos planos ángulos iguales con peso de prueba de 10g, proyecta un mejor comportamiento con los fasores de 90Hz, donde W_{c1r} tiene un exceso mínimo de 0.76 de la magnitud posicionada, de igual forma para W_{c1i} muestra un aumento minúsculo de 0.39 del valor inicial.

Capítulo 5. Conclusiones y trabajos a futuro

5.1 Conclusiones

Como resultado de una larga investigación, se decidió implementar una nueva metodología para el balanceo dinámico de un motor en uno y dos planos, mediante información extraída de los fasores de 30 y 90 Hz de la corriente de alimentación.

Se realizaron un total de 66 mediciones con tres pesos de pruebas distintos, como se muestran a continuación:

- Balanceo en un plano con peso de prueba de 6.7g en un plano, demandó 24 mediciones (para 30 y 90 Hz).
- Balanceo en un plano con peso de prueba de 8.2g en un plano, solicitó realizar 24 mediciones (para 30 y 90 Hz).
- Balanceo en dos planos ángulos iguales con peso de prueba de 6.7g requirió de 6 mediciones, para armónicas de 30 y 90 Hz.
- Balanceo en dos planos ángulos distintos con peso de prueba de 6.7g demandó un total de 6 mediciones, para armónicas de 30 y 90 Hz.
- finalmente, para el balanceo en dos planos ángulos iguales con peso de desbalance de 10g, solicitó un total de 6 mediciones.

Proporcionando una amplia gama de información para la selección de casos a evaluar, del cual se planteó el estudio de diez casos, en donde se expusieron distintas anomalías.

Para el balanceo en un plano, se demostró una mayor exactitud utilizando armónicas de 90 Hz, ya que sus rangos de errores para 6.7g rondaban entre 0.07g a 0.2g. Sin embargo, al utilizar una magnitud superior el margen de error disminuía considerablemente, ya que para 8.2g el rango de error resultó entre 0.08g a 0.02g.

Mientras que, para el balanceo en dos planos posicionados en ángulos iguales se revelaron datos insólitos, ya que al utilizar un peso de desbalance menor las armónicas de 30 Hz presentaban una mayor exactitud para el cálculo de W_C , sin embargo, al aumentar el peso de la

magnitud las armónicas de 90Hz proporcionan resultados precisos. Esto se debe a que, en ocasiones, las configuraciones de los datos de cada fase son cercanas entre sí.

Además, se demostró que al realizar variaciones de los datos fuera de los promedios, pero conservándose dentro de los rangos de variación de éstos, existe una gama de resultados o de intersecciones en los tres círculos que corresponden a posibles valores de pesos de balanceo. Lo reportado en los resultados anteriores corresponde a la configuración promedio de las mediciones.

5.2 Recomendaciones y trabajos a futuro

Debido a que los datos obtenidos en las mediciones presentan cuantiosas dispersiones que en ocasiones reflejan una falta de precisión en los resultados, se recomienda el uso de un regulador de voltaje trifásico para reducir la variabilidad en los datos

Para el caso del balanceo en dos planos con pesos en ángulos opuestos, presentó dificultades para el cálculo de W_c , ya que, al generar el resultado cuenta con un margen de error muy amplio que va de 16.915 hasta 125 para W_{cr} y W_{ci} respectivamente.

El algoritmo 8H16 se debe perfeccionar para determinar si las configuraciones de los datos van a producir círculos casi concéntricos que no se intercepten antes de producir los resultados.

Referencias

- Bellini, A., Filippetti, F., Tassoni, C., & Capolino, G.-A. (2008). Advances in Diagnostic Techniques for Induction Machines. *EEI TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 55, No. 12*, 4109-4216.
- Bhende, A. R. (2020). Dynamic balancing of a two-plane rotor without phase angles measurement using the amplitude subtraction method. *DYNAMIC BALANCING*, 1-45.
- Chapaman, S. (2012). *Máquinas Eléctricas (5ta ed)*. México: MCGRAW HILL EDDUCATION.
- Estupiñán P., E., San Martín, C., & Canales M., L. (2006). DESARROLLO DE UN INSTRUMENTO VIRTUAL PARA EL BALANCEO DINÁMICO DE ROTORES. *Ingeniare-Revista Chilena de Ingeniería, vol 14 No.12*, 146-152.
- García Reynoso, A. C., Cervantes Moya, M. A., Camargo Méndez, J., Ladrón de Guevara Durán, E., Hernández Marceliz, E., & Peña Pardo, B. C. (2008). Método de Balanceo de Rotores Rígidos con Datos de Vibración. *SOMIM Sociedad Mexicana de Ingeniería Mecánica*, (págs. 021-028). Puebla, México.
- García Reynoso, A. C., Ladrón de Guevara Durán, E., Lorandi Medina, P. A., Hermida Saba, G., Villalobos Zúñiga, V., Moreno Pérez, R., & Morales Sánchez, O. (2016). Determinación de fasores de desbalance mecánico obtenidos de corriente eléctrica residual de motor de inducción acoplado a rotor desbalanceado. *MEMORIAS DEL XXII CONGRESO DELA SOMIN 2016* (págs. 533-540). MÉRICA, YUCATAN: SOMIN.
- Kumar, P., & Tiwari, R. (2020). Dynamic analysis and identification of unbalance and misalignment in a rigid rotor with two offset discs levitated by active magnetic bearings: a novel trail misalignment approach. *Propulsion and Power Research*, 1-25.
- La Malfa, S., & Ercoli, L. (2013). *Teoría y Práctica del Balanceo de Rotores Industriales*. LIBRERIA Y EDITORIAL ALSINA.
- NI. (15 de 02 de 2020). NI. Obtenido de NI: <http://sine.ni.com/np/app/main/p/ap/daq/lang/es/pg/1/sn/n17:daq/fmid/652/>

- S.S., R. (2012). *Vibraciones Mecánicas*. México: Pearson.
- Tarabini, M., & Scaccabarozzi, D. (2018). Uncertainty-based combination of signal processing techniques for the identification of rotor imbalance. *ELSEVIER*, 409-416.
- Villa Duque, F., Parra, D., & Ocampo, G. (2007). Estudio del comportamiento de motores de inducción ante fallas estáticas. *Revista Facultad de Ingeniería No. 40*, 76-94.
- Wang, A., Cheng, X., Meng, G., Xia, Y., Wo, L., & Wang, Z. (2017). Dynamic analysis and numerical experiments for balancing of the continuous single-disc and single-span rotor-bearing system. *ELSEVIER*, 151-176.
- Zhao, S., Ren, X., Deng, W., Yongfeng, Y., & Fu, C. (2021). A transient characteristic-based balancing method of rotor system without trial weights. *ELSEVIER*, 148.

Anexos

A1. Algoritmo 12A (armónicas de 30Hz)

```
function [lim, CC, DD, EE, teta1, teta2, teta3]=intervalo(nint)
i=0; iflag=0; j=0;
// write(u,"B(1),B(2),B(3)");
while i<3999 & iflag==0;
    i=i+1;
//for i=1:4
    // load(u1,[x1,x2,x3]);
    [B]=read(u1,1,3);
    if B(1)==0 & B(2)==0 then
        iflag=1;
    end
    disp(iflag,"iflag=");
    disp("B",[B(1),B(2),B(3)]);
    disp(i,"i=");
    if i>999 then
        C(i-999)=B(1); D(i-999)=B(2); E(i-999)=B(3);
    //    write(u,[B(1),B(2),B(3)],'(3(f12.6))');
        j=j+1; // cuenta los registros a considerar
    end
end // fin de while

disp("C",[C(1),C(2),C(3)]);
disp(j,"j"); disp(iflag,"iflag");
if iflag==1 then
    disp(nint,"nint=");
    // pause
end
ii=2; lim=j; k1=1; k2=1; k3=1;
Cim2=C(1); Dim2=D(1); Eim2=E(1); i1m2=1; i2m2=1; i3m2=1;
Cim1=C(2); Dim1=D(2); Eim1=E(2); i1m1=2; i2m1=2; i3m1=2;

while ii<lim
    ii=ii+1;
    disp("ii",ii);
    disp([Cim1,Cim2,C(ii),k1],"Cim1,Cim2,C(ii),k1");
    if Cim1>Cim2 & C(ii)<Cim1 then
        Cmax(k1)=Cim1; i1(k1)=i1m1; k1=k1+1;
    end
    Cim2=C(ii-1); Cim1=C(ii); i1m2=ii-1; i1m1=ii;
    disp([Cim1,Cim2,C(ii),k1],"Cim1,Cim2,C(ii),k1");

    if Dim1>Dim2 & D(ii)<Dim1 then
        Dmax(k2)=Dim1; i2(k2)=i2m1; k2=k2+1;
    end
    Dim2=D(ii-1); Dim1=D(ii); i2m2=ii-1; i2m1=ii;
    if Eim1>Eim2 & E(ii)<Eim1 then
        Emax(k3)=Eim1; i3(k3)=i3m1; k3=k3+1;
    end
end
```

```

    Eim2=E(ii-1); Eim1=E(ii); i3m2=ii-1; i3m1=ii;
end // fin de while

// averigua cual de k1,k2,k3 es máspequeña
k=k1;
if k2<=k1 & k2<=k3 then
    k=k2;
end
if k3<=k1 & k3<=k2 then
    k=k3;
end
disp([k1,k2,k3,k],"k1,k2,k3,k");
//pause
//write(u2,"i1,Cmax");
for i=1:k1-1
    disp("i1,Cmax",[i1(i),Cmax(i)]);
    // write(u2,[i1(i),Cmax(i)],'(2(f12.6))');
end
disp(k2,"k2");
//write(u2,"i2,Dmax");
for i=1:k2-1
    disp("i2,Dmax",[i2(i),Dmax(i)]);
    // write(u2,[i2(i),Dmax(i)],'(2(f12.6))');
end
//write(u2,"i3,Emax");
disp([k1,k2,k3],"k1,k2,k3");
//pause
for i=1:k3-1
    disp("i3,Emax",[i3(i),Emax(i)]);
    // write(u2,[i3(i),Emax(i)],'(2(f12.6))');
end
//sC=mad(Cmax); sD=mad(Dmax); sE=mad(Emax);
// calcula las fases relativas
// write(u2,"teta1(i),teta2(i),teta3(i)");
write(u2,"i,i1(i),Cmax(i),i2(i),Dmax(i),i3(i),Emax(i),teta1(i),teta2(i),teta3(i)");
for i=1:k-2

    // calcula frecuencias
    // for k1=1:k-2
    k1=i;
    //fase 1
    x1=i1(k1)-1; x2=i1(k1); x3=i1(k1)+1;
    y1=C(i1(k1)-1); y2=C(i1(k1)); y3=C(i1(k1)+1);
    // y1=C(x1); y2=C(x2); y3=C(x3);
    // escalar
    x10=x1;
    x1=1; x2=x2/x10; x3=x3/x10;
    //x1=x1/1000; x2=x2/1000; x3=x3/1000;
    [pos1,valor1]=curva(x1,x2,x3,y1,y2,y3)
    pos1=pos1*x10;
    disp(k1,"k1");

```



```

disp([x1,x2,x3,y1,y2,y3],"x1,x2,x3,y1,y2,y3");
disp([pos1,valor1],"pos1,valor1");
// write(u5,[k1,x1,x2,x3,y1,y2,y3],'(7(f16.4))');
ii1(i)=pos1; CC(i)=valor1;
//fase 2
x1=i2(k1)-1; x2=i2(k1); x3=i2(k1)+1;
y1=D(x1); y2=D(x2); y3=D(x3);

// escalar
x10=x1;
x1=1; x2=x2/x10; x3=x3/x10;
//x1=x1/1000; x2=x2/1000; x3=x3/1000;

[pos2,valor2]=curva(x1,x2,x3,y1,y2,y3)
pos2=pos2*x10;
ii2(i)=pos2; DD(i)=valor2;
//fase 3
x1=i3(k1)-1; x2=i3(k1); x3=i3(k1)+1;
y1=E(x1); y2=E(x2); y3=E(x3);
// escalar
x10=x1;
x1=1; x2=x2/x10; x3=x3/x10;
//x1=x1/1000; x2=x2/1000; x3=x3/1000;
[pos3,valor3]=curva(x1,x2,x3,y1,y2,y3)
pos3=pos3*x10;
ii3(i)=pos3; EE(i)=valor3;
write(u5,[k1, pos1,valor1, pos2,valor2, pos3,valor3],'(7(f16.4))');
// end; // fin de for k1
end; // fin de for i
for i=1:k-3
// calcula ángulos de fase relativos
teta1(i)=0;
teta2(i)=360*(ii2(i)-ii1(i))/(ii1(i+1)-ii1(i));
teta3(i)=360*(ii3(i)-ii1(i))/(ii1(i+1)-ii1(i));
write(u2,[k,i,ii1(i), CC(i)],'(4(f12.6))');
// invierte signos el valor mayor es retraso
teta2(i)=-teta2(i); teta3(i)=-teta3(i);
if teta2(i)<0. then
teta2(i)=teta2(i)+360.;
end
if teta3(i)<0. then
teta3(i)=teta3(i)+360.;
end

frec1=1/((ii1(i+1)-ii1(i))/5000); frec2=1/((ii2(i+1)-ii2(i))/5000); frec3=1/((ii3(i+1)-ii3(i))/5000);
// write(u2,[i,ii1(i), Cmax(i),i2(i), Dmax(i),i3(i), Emax(i), teta1(i), teta2(i), teta3(i)],'(10(f12.6))');
write(u4,[CC(i), teta1(i),frec1, DD(i), teta2(i),frec2, EE(i), teta3(i),frec3],'(9(f12.6))');
// write(u4,[Cmax(i), Dmax(i), Emax(i), teta1(i), teta2(i), teta3(i),frec1,frec2,frec3],'(9(f12.6))');
end; // fin de for i
write(u2,"i,ii1(i),CC(i)");

```

```

endfunction
function [pos, valor]=curva(x1, x2, x3, y1, y2, y3)
A=[x1^2,x1,1;x2^2,x2,1;x3^2,x3,1];
b=[y1;y2;y3];
x=inv(A)*b;
disp(x,"coeficientes");
// posición de máximo
pos=-x(2)/(2*x(1));
valor=x(1)*pos^2+x(2)*pos+x(3);
disp([pos,valor],"pos,valor");

endfunction

// file('close',file())
funcprot(0)
u5=file('open','C:\scilab\corrfa5.dat','unknown','formatted')
u1=file('open','C:\scilab\prueba.txt','unknown')
u2=file('open','C:\scilab\corrfa2.dat','unknown','formatted')
u3=file('open','C:\scilab\corrfa3.dat','unknown','formatted')
u4=file('open','C:\scilab\corrfa4.dat','unknown','formatted')

u6=file('open','C:\scilab\corrfa6.dat','unknown','formatted')
u7=file('open','C:\scilab\corrfa7.dat','unknown','formatted')
u8=file('open','C:\scilab\corrfa8.dat','unknown','formatted')
u9=file('open','C:\scilab\corrfa9.dat','unknown','formatted')
// lint=65536/3999;
lint=3*65000/3999;
//lint=16;
write(u4,"CCmax(i),teta1(i),frec1,DDmax(i),teta2(i),frec2,EEmax(i),teta3(i),frec3");

write(u6,"nint,sum21,sum22,sum23,ncount1,ncount2,ncount3,rms1,rms2,rms3,teta1,teta2,teta3"
);
write(u5,"k1,pos1,valor1,pos2,valor2,pos3,valor3");
for nint=1:lint
[lim,CC,DD,EE,teta1,teta2,teta3]=intervalo(nint);
disp([nint,lim],"nint,lim");
// pause
end // fin de for nint

file('close',file())

```

A12. Algoritmo 12AA (armónicas de 90 Hz)

```
function [lim, CC, DD, EE, teta1, teta2, teta3]=intervalo(nint)
i=0; iflag=0; j=0;
// write(u,"B(1),B(2),B(3)");
while i<3999 & iflag==0;
    i=i+1;
//for i=1:4
    // load(u1,[x1,x2,x3]);
    [B]=read(u1,1,3);
    if B(1)==0 then
        iflag=1;
    end
    disp(iflag,"iflag=");
    disp("B",[B(1),B(2),B(3)]);
    if i>999 then
        C(i-999)=B(1); D(i-999)=B(2); E(i-999)=B(3);
    //    write(u,[B(1),B(2),B(3)],'(3f12.6)');
        j=j+1;
    end
end // fin de while

disp("C",[C(1),C(2),C(3)]);
disp(j,"j"); disp(iflag,"iflag");
if iflag==1 then
    disp(nint,"nint=");
    // pause
end
ii=2; lim=j; k1=1; k2=1; k3=1;
Cim2=C(1); Dim2=D(1); Eim2=E(1); i1m2=1; i2m2=1; i3m2=1;
Cim1=C(2); Dim1=D(2); Eim1=E(2); i1m1=2; i2m1=2; i3m1=2;

while ii<lim
    ii=ii+1;
    disp("ii",ii);
    disp([Cim1,Cim2,C(ii),k1],"Cim1,Cim2,C(ii),k1");
    if Cim1>Cim2 & C(ii)<Cim1 then
        Cmax(k1)=Cim1; i1(k1)=i1m1; k1=k1+1;
    end
    Cim2=C(ii-1); Cim1=C(ii); i1m2=ii-1; i1m1=ii;
    disp([Cim1,Cim2,C(ii),k1],"Cim1,Cim2,C(ii),k1");

    if Dim1>Dim2 & D(ii)<Dim1 then
        Dmax(k2)=Dim1; i2(k2)=i2m1; k2=k2+1;
    end
    Dim2=D(ii-1); Dim1=D(ii); i2m2=ii-1; i2m1=ii;
    if Eim1>Eim2 & E(ii)<Eim1 then
        Emax(k3)=Eim1; i3(k3)=i3m1; k3=k3+1;
    end
    Eim2=E(ii-1); Eim1=E(ii); i3m2=ii-1; i3m1=ii;
end // fin de while
```

```

// averigua cual de k1,k2,k3 es máspequeña
k=k1;
if k2<=k1 & k2<=k3 then
    k=k2;
end
if k3<=k1 & k3<=k2 then
    k=k3;
end
disp([k1,k2,k3,k],"k1,k2,k3,k");
//pause
//write(u2,"i1,Cmax");
for i=1:k1-1
    disp("i1,Cmax",[i1(i),Cmax(i)]);
    // write(u2,[i1(i),Cmax(i)],'(2(f12.6))');
end
disp(k2,"k2");
//write(u2,"i2,Dmax");
for i=1:k2-1
    disp("i2,Dmax",[i2(i),Dmax(i)]);
    // write(u2,[i2(i),Dmax(i)],'(2(f12.6))');
end
//write(u2,"i3,Emax");
disp([k1,k2,k3],"k1,k2,k3");
//pause
for i=1:k3-1
    disp("i3,Emax",[i3(i),Emax(i)]);
    // write(u2,[i3(i),Emax(i)],'(2(f12.6))');
end
//sC=mad(Cmax); sD=mad(Dmax); sE=mad(Emax);
// calcula las fases relativas
// write(u2,"teta1(i),teta2(i),teta3(i)");
write(u2,"i,i1(i),Cmax(i),i2(i),Dmax(i),i3(i),Emax(i),teta1(i),teta2(i),teta3(i)");
for i=1:k-2

    // calcula frecuencias
    // for k1=1:k-2
    k1=i;
    //fase 1
    x1=i1(k1)-1; x2=i1(k1); x3=i1(k1)+1;
    y1=C(i1(k1)-1); y2=C(i1(k1)); y3=C(i1(k1)+1);
    // y1=C(x1); y2=C(x2); y3=C(x3);
    // escalar
    x10=x1;
    x1=1; x2=x2/x10; x3=x3/x10;
    //x1=x1/1000; x2=x2/1000; x3=x3/1000;
    [pos1,valor1]=curva(x1,x2,x3,y1,y2,y3)
    pos1=pos1*x10;
    disp(k1,"k1");
    disp([x1,x2,x3,y1,y2,y3],"x1,x2,x3,y1,y2,y3");
    disp([pos1,valor1],"pos1,valor1");
    // pause

```

```

// write(u5,[k1,x1,x2,x3,y1,y2,y3],'(7(f16.4))');
ii1(i)=pos1; CC(i)=valor1;
    //fase 2
    x1=i2(k1)-1; x2=i2(k1); x3=i2(k1)+1;
    y1=D(x1); y2=D(x2); y3=D(x3);

    // escalar
    x10=x1;
x1=1; x2=x2/x10; x3=x3/x10;
//x1=x1/1000; x2=x2/1000; x3=x3/1000;

[pos2,valor2]=curva(x1,x2,x3,y1,y2,y3)
pos2=pos2*x10;
ii2(i)=pos2; DD(i)=valor2;
    //fase 3
    x1=i3(k1)-1; x2=i3(k1); x3=i3(k1)+1;
    y1=E(x1); y2=E(x2); y3=E(x3);
    // escalar
    x10=x1;
x1=1; x2=x2/x10; x3=x3/x10;
//x1=x1/1000; x2=x2/1000; x3=x3/1000;
[pos3,valor3]=curva(x1,x2,x3,y1,y2,y3)
pos3=pos3*x10;
ii3(i)=pos3; EE(i)=valor3;
write(u5,[k1,pos1,valor1,pos2,valor2,pos3,valor3],'(7(f16.4))');
// end; // fin de for k1
end; // fin de for i
for i=1:k-3
// calcula ángulos de fase relativos
    teta1(i)=0;
    teta2(i)=360*(ii2(i)-ii1(i))/(ii1(i+1)-ii1(i));
    teta3(i)=360*(ii3(i)-ii1(i))/(ii1(i+1)-ii1(i));
    write(u2,[k,i,ii1(i),CC(i)'],'(4(f12.6))');
    // invierte signos el valor mayor es retraso
    teta2(i)=-teta2(i); teta3(i)=-teta3(i);
    if teta2(i)<0. then
        teta2(i)=teta2(i)+360.;
    end
    if teta3(i)<0. then
        teta3(i)=teta3(i)+360.;
    end

    frec1=1/((ii1(i+1)-ii1(i))/5000); frec2=1/((ii2(i+1)-ii2(i))/5000); frec3=1/((ii3(i+1)-ii3(i))/5000);
// write(u2,[i,ii1(i),Cmax(i),i2(i),Dmax(i),i3(i),Emax(i),teta1(i),teta2(i),teta3(i)'],'(10(f12.6))');
write(u4,[CC(i),teta1(i),frec1,DD(i),teta2(i),frec2,EE(i),teta3(i),frec3'],'(9(f12.6))');
// write(u4,[Cmax(i),Dmax(i),Emax(i),teta1(i),teta2(i),teta3(i),frec1,frec2,frec3'],'(9(f12.6))');
end; // fin de for i
write(u2,"i,ii1(i),CC(i)");

endfunction
function [pos, valor]=curva(x1, x2, x3, y1, y2, y3)

```

```

A=[x1^2,x1,1;x2^2,x2,1;x3^2,x3,1];
b=[y1;y2;y3];
x=inv(A)*b;
disp(x,"coeficientes");
// posición de máximo
pos=-x(2)/(2*x(1));
valor=x(1)*pos^2+x(2)*pos+x(3);
disp([pos,valor],"pos,valor");

endfunction

// file('close',file())
funcprot(0)
u5=file('open','C:\scilab\corrfa5.dat','unknown','formatted')
u1=file('open','C:\scilab\prueba.txt','unknown')
u2=file('open','C:\scilab\corrfa2.dat','unknown','formatted')
u3=file('open','C:\scilab\corrfa3.dat','unknown','formatted')
u4=file('open','C:\scilab\corrfa4.dat','unknown','formatted')

u6=file('open','C:\scilab\corrfa6.dat','unknown','formatted')
u7=file('open','C:\scilab\corrfa7.dat','unknown','formatted')
u8=file('open','C:\scilab\corrfa8.dat','unknown','formatted')
u9=file('open','C:\scilab\corrfa9.dat','unknown','formatted')
lint=65536/3999;
lint=16;
write(u4,"CCmax(i),teta1(i),frec1,DDmax(i),teta2(i),frec2,EEmax(i),teta3(i),frec3");

write(u6,"nint,sum21,sum22,sum23,ncount1,ncount2,ncount3,rms1,rms2,rms3,teta1,teta2,teta3"
);
write(u5,"k1,pos1,valor1,pos2,valor2,pos3,valor3");
for nint=1:lint
[lim,CC,DD,EE,teta1,teta2,teta3]=intervalo(nint);
disp([nint,lim],"nint,lim");

end // fin de for nint

file('close',file())

```

A3. Algoritmo 8H6

```

funcprot(0)
function [Wc1, Wc2, Wc11, Wc21, indica, A11r0, A11i0, A21r0, A21i0, A111r0, A111i0,
A211r0, A211i0, Wc1r0, Wc1i0, Wc2r0, Wc2i0, Wc11r0, Wc11i0, Wc21r0, Wc21i0, indoff,
WWc1r0, WWc1i0, WWc11r0, WWc11i0, Fx1, Fy1, Fx2, Fy2, Fx3, Fy3, Gx1, Gy1, Gx2, Gy2,
Gx3, Gy3, jx]=offset(i, k1, j, R21, R22, R10, r, W1, W2, teta0, eO1, indica, Wc1r0, Wc1i0,
Wc2r0, Wc2i0, Wc11r0, Wc11i0, Wc21r0, Wc21i0, indoff, WWc1r0, WWc1i0, WWc11r0,
WWc11i0)
    teta121=0; teta122=0; ind=0; x1=0; y1=0; x2=0; y2=0; zO2=complex(0,0);
// calcula fase 1
disp([R21,R22,r],"R21,R22,r")

for i=1:3
Fx1(i)=0; Fy1(i)=0; Fx2(i)=0; Fy2(i)=0; Fx3(i)=0; Fy3(i)=0;
Gx1(i)=0; Gy1(i)=0; Gx2(i)=0; Gy2(i)=0; Gx3(i)=0; Gy3(i)=0;
end

//calcula punto de cruce en círculo de R2

zR10=complex(R10*cosd(teta0),R10*sind(teta0)); zR21=complex(R21,0);
zR22=complex(R22,0);
z=(1-r)*zR10;
C1x=real(z); C1y=imag(z);
z=r*zR21;
Rp1=abs(z);
disp([C1x,C1y,Rp1],"C1x,C1y,Rp1");
imdi=0;
//if C1y==0 then
    if abs(C1y)<0.0001 then
        x1=(Rp1^2-R22^2-C1x^2)/(-2*C1x);
        disc=R22^2-x1^2;
        if disc>=0 then
            y1=sqrt(R22^2-x1^2);
x2=x1; y2=-y1; else
        [x1,y1,x2,y2]=nuevo(C1x,C1y,R22,Rp1)
        imdi=1;
disp([x1,y1,x2,y2],"x1,y1,x2,y2....C1y<0.0001")
end
//else
end

    if abs(C1x)<0.0001 then
        y1=(Rp1^2-R22^2-C1y^2)/(-2*C1y);
        disc=R22^2-y1^2;
        if disc>=0 then
            x1=sqrt(R22^2-y1^2);
y2=y1; x2=-x1; else
        [x1,y1,x2,y2]=nuevo(C1x,C1y,R22,Rp1)
        imdi=1;

```

```

disp([x1,y1,x2,y2],"x1,y1,x2,y2....C1x<0.0001")//else
end
end

//if C1y<>0 then
    if abs(C1y)>0.0001 & abs(C1x)>0.0001 then
b=(R22^2-Rp1^2+C1x^2+C1y^2)/(2*C1y);
disp([C1x,C1y,Rp1,b],"C1x,C1y,Rp1,b");
// forma cuadrática
a1=1+(C1x/C1y)^2;
disp([R22,Rp1,C1x,C1y,b,a1],"R22,Rp1,C1x,C1y,b,a1")
a2=-2*b*C1x/C1y;
a3=b^2-R22^2;
disc=a2^2-4*a1*a3;
disp([r,zR10,C1x,C1y,Rp1,b],"r,zR10,C1x,C1y,Rp1,b")
disp([a1,a2,a3,disc],"a1,a2,a3,disc");
//pause
if disc>=0 then
    x1=(-a2+sqrt(disc))/(2*a1);
    x2=(-a2-sqrt(disc))/(2*a1);
y1=b-(C1x/C1y)*x1; y2=b-(C1x/C1y)*x2; else;
[x1,y1,x2,y2]=nuevo(C1x,C1y,R22,Rp1)
    imdi=1;

//y1=sqrt(R2^2-x1^2); y2=sqrt(R2^2-x2^2);
end;
// x1=0; y1=0; x2=0; y2=0;

end// fin de if C1y

disp("cruce de círculo R2")
disp([x1,y1,x2,y2],"x1,y1,x2,y2")
if j==3 then
    // pause
end
//
x11=x1; x12=x2; y11=y1; y12=y2;
N2N11r=x11-real(zR10); N2N11i=y11-imag(zR10);
N2N12r=x12-real(zR10); N2N12i=y12-imag(zR10);

disp([x1,y1,x2,y2],"x1,y1,x2,y2")
// checa soluciones
[rhs11,rhs12,rhs21,rhs22,teta221,teta222]=checaR2(x1,y1,x2,y2,R22,Rp1,C1x,C1y)
if teta0==135 then
    disp(rhs11,rhs12,rhs21,rhs22,teta221,teta222)
end
//write(u4,[j,teta0,x1,y1,x2,y2,rhs11,rhs12,rhs21,rhs22,R22,teta221,teta222],'(13(f16.4))');

//else
// ind=1; disp([j,disc],"j,disc");
//end

```



```

// selecciona  $0 < \text{teta1} < 180^\circ$ 
xx2=x1; yy2=y1;
if y2>0 then
xx2=x2; yy2=y2;
end
if ind==0 then
// calcula para  $C1y=0$ ;
//  $x=(Rp1^2-R12^2-C1x^2)/(-2*C1x)$ ;
//  $x=b*(C1y/C1x)$ ;
//disp(x,"si  $C1y=0$ ")

//calcula punto de cruce en círculo de R1
z=(1-1/r)*zR10;
C1x=real(z); C1y=imag(z);
z=zR22/r;
Rp2=abs(z);

// modificación 3 febrero 2021
if abs(C1y)<0.0001 then
//if  $C1y==0$  then
x1=(Rp2^2-R21^2-C1x^2)/(-2*C1x);
disc=R21^2-x1^2;
if disc>=0 then
y1=sqrt(R21^2-x1^2);
x2=x1; y2=-y1; else
[x1,y1,x2,y2]=nuevo(C1x,C1y,R22,Rp1)
end
end

if abs(C1x)<0.0001 then
//if  $C1x==0$  then
y1=(Rp2^2-R21^2-C1y^2)/(-2*C1y);
disc=R21^2-y1^2;
if disc>=0 then
x1=sqrt(R21^2-y1^2);
x2=-x1; y2=y1; else
[x1,y1,x2,y2]=nuevo(C1x,C1y,R22,Rp1) end
end

if abs(C1y)>0.0001 & abs(C1x)>0.0001 then
//if  $C1y \neq 0$  then
b=(R21^2-Rp2^2+C1x^2+C1y^2)/(2*C1y);
disp([C1x,C1y,Rp2,b],"C1x,C1y,Rp2,b");
// forma cuadrática
a1=1+(C1x/C1y)^2;
a2=-2*b*C1x/C1y;
a3=b^2-R21^2;

disc=a2^2-4*a1*a3;
disp([r,zR10,C1x,C1y,Rp2,b],"r,zR10,C1x,C1y,Rp2,b")
disp([a1,a2,a3,disc],"a1,a2,a3,disc");

```

```

//pause
if disc>=0 then
    x1=(-a2+sqrt(disc))/(2*a1);
    x2=(-a2-sqrt(disc))/(2*a1);
y1=b-(C1x/C1y)*x1; y2=b-(C1x/C1y)*x2; else
//y1=sqrt(R1^2-x1^2); y2=sqrt(R1^2-x2^2);
[x1,y1,x2,y2]=nuevo(C1x,C1y,R22,Rp1)
end
end; // fin de if C1y

disp("cruce de círculo R1")
disp([x1,y1,x2,y2],"x1,y1,x2,y2")
xx11=x1; xx12=x2; yy11=y1; yy12=y2;
N2N13r=xx11-real(zR10); N2N13i=yy11-imag(zR10);
N2N14r=xx12-real(zR10); N2N14i=yy12-imag(zR10);
// checa soluciones
[rhs11,rhs12,rhs21,rhs22,teta211,teta212]=checaRp21(x1,y1,x2,y2,R21,Rp2,C1x,C1y)
if teta0==135 then
    disp("checaRp21")
    disp(rhs11,rhs12,rhs21,rhs22,teta211,teta212)
end
// checa soluciones en forma polar
[rhs11,rhs21,rhs12,rhs22]=polarcheck(R21,R22,teta211,teta212,teta221,teta222,r,R10,teta0)
//write(u4,[j,teta0,teta211,teta212,teta221,teta222,rhs11,rhs12,rhs21,rhs22]','(10(f16.4))');
if teta0==135 then
    disp("polarcheck")
    disp(rhs11,rhs21,rhs12,rhs22)
end
// selecciona segun rhs11, rhs12, rhs21, rhs22
[xx1,yy1,xx2,yy2,xxx1,yyy1,xxx2,yyy2]=seleccionpolar(teta211, teta212, teta221,
teta222,rhs11,rhs21,rhs12,rhs22,R21,R22)
if teta0==135 then
    disp("checa seleccionpolar")
    disp(xx1,yy1,xx2,yy2,xxx1,yyy1,xxx2,yyy2)
// pause
end

//write(u4,[j,teta0,x1,y1,x2,y2,rhs11,rhs12,rhs21,rhs22]','(10(f16.4))');

rr=real(r); ri=imag(r);
//write(u3,[j,teta0,N2N11r,N2N11i,N2N12r,N2N12i,N2N13r,N2N13i,N2N14r,N2N14i,rr,ri]','(12(f16
.4))');
// llama función selección para tomar los puntos correctos
if imdi==0 then
[xx1,yy1,xx2,yy2,ind10,xxx1,yyy1,xxx2,yyy2,ind20]=seleccion(r,x11,x12,y11,y12,xx11,xx12,yy1
1,yy12,R21,R22,zR10)
else
    ind1=0; ind2=0; ind10=0; ind20=0;
end
//end
//xx1=xx11; yy1=yy11; xx2=xx12; yy2=yy12;

```

```

// selecciona solucion fase 1-2
//[xx1,yy1,xx2,yy2]=seleccion(r,x11,x12,y11,y12,xx11,xx12,yy11,yy12,R21,R22,teta121,teta122)
disp([xx1,yy1,xx2,yy2],"xx1,yy1,xx2,yy2");
disp([teta0,R11,R12,R21,R22,R10],"teta0,R11,R12,R21,R22,R10")
//pause
if ind10==0 then

disp([xx1,yy1,xx2,yy2],"selección: xx1,yy1,xx2,yy2");

// calcula coeficiente de influencia
zR0L21=complex(xx1,yy1); zR0L22=complex(xx2,yy2);

// calcula los coeficientes de influencia
A11=(zR0L21-zR10)/W1; A21=(zR0L22-zR10)/W2;
A11r=real(A11); A11i=imag(A11); A21r=real(A21); A21i=imag(A21); else
    A11r=0; A11i=0; A21r=0; A21i=0;
    A11r0=A11r; A11i0=A11i; A21r0=A21r; A21i0=A21i;
end; // fin de if ind10

if ind20==0 then

disp([xxx1,yyy1,xxx2,yyy2],"selección: xxx1,yyy1,xxx2,yyy2");

// calcula coeficiente de influencia
zR0L211=complex(xxx1,yyy1); zR0L221=complex(xxx2,yyy2);

// calcula los coeficientes de influencia
A111=(zR0L211-zR10)/W1; A211=(zR0L221-zR10)/W2;
A111r=real(A111); A111i=imag(A111); A211r=real(A211); A211i=imag(A211); else
    A111r=0; A111i=0; A211r=0; A211i=0;
    A111r0=A111r; A111i0=A111i; A211r0=A211r; A211i0=A211i;
end; // fin de if ind20

// uniformiza los arreglos
//if indica==0 then
    if indica==0 & ind10==0 & ind20==0 then
        A11r0=A11r; A11i0=A11i; A21r0=A21r; A21i0=A21i;
A111r0=A111r; A111i0=A111i; A211r0=A211r; A211i0=A211i;
        disp([indica,A11r0],"indica,A11r0")
    end

//if indica==1 then
[A11r,A11i,A21r,A21i,A111r,A111i,A211r,A211i]=uniforme(A11r,A11i,A21r,A21i,A111r,A111i,A211r,A211i,A211r,A211i,A11r0,A11i0,A21r0,A21i0,A111r0,A111i0,A211r0,A211i0)
//end
//indica=1;

//write(u6,[j,teta0,A11r,A11i,A21r,A21i,A111r,A111i,A211r,A211i],'(10(f16.4))');
// calcula N
//for i=1:80
    for i=1:1

```

```

// eeO1=eO1+5*(i-1)*0.05
eeO1=eO1;
O1=complex(eeO1,0);

N=zR10-O1;
// calcula Wc
if ind10==0 then
Wc1=-N/A11; Wc2=-N/A21; else
Wc1=0; Wc2=0;
end
if ind20==0 then
Wc11=-N/A111; Wc21=-N/A211; else
Wc11=0; Wc21=0;
end

// calcula offset2
//zO2=(zR0L22-r*zR0L21)/(1-r);
disp(zO2,"zO2");

O1x=real(O1); O1y=imag(O1); O2x=real(zO2); O2y=imag(zO2);
if ind10==0 then
R0L21r=real(zR0L21); R0L21i=imag(zR0L21);
R0L22r=real(zR0L22); R0L22i=imag(zR0L22);
end
if ind20==0 then
zR00L21=zR0L21-O1; zR00L22=zR0L22-O1;
R00L21r=real(zR00L21); R00L21i=imag(zR00L21);
R00L22r=real(zR00L22); R00L22i=imag(zR00L22);
end
//A11=zR00L21/W1; A12=zR00L22/W2;

Nr=real(N); Ni=imag(N);
Wc1r=real(Wc1); Wc1i=imag(Wc1); Wc2r=real(Wc2); Wc2i=imag(Wc2);
Wc11r=real(Wc11); Wc11i=imag(Wc11); Wc21r=real(Wc21); Wc21i=imag(Wc21);

if indica==0 then
Wc1r0=Wc1r; Wc1i0=Wc1i; Wc2r0=Wc2r; Wc2i0=Wc2i;
Wc11r0=Wc11r; Wc11i0=Wc11i; Wc21r0=Wc21r; Wc21i0=Wc21i;
end
disp([indica,A11r0],"indica,A11r0")
jx=0;
if indica==1 then
[Wc1r,Wc1i,Wc2r,Wc2i,Wc11r,Wc11i,Wc21r,Wc21i,jx]=uniformeW(j,Wc1r,Wc1i,Wc2r,Wc2i,Wc11r,Wc11i,Wc21r,Wc21i,Wc1r0,Wc1i0,Wc2r0,Wc2i0,Wc11r0,Wc11i0,Wc21r0,Wc21i0)
end
indica=1;
A11r0=A11r; A11i0=A11i; A21r0=A21r; A21i0=A21i;
A111r0=A111r; A111i0=A111i; A211r0=A211r; A211i0=A211i;

if indoff==0 then

```

```

//
[Wc1r,Wc1i,Wc2r,Wc2i,Wc11r,Wc11i,Wc21r,Wc21i]=uniformeoff(j,O1x,Wc1r,Wc1i,Wc2r,Wc2i,
Wc11r,Wc11i,Wc21r,Wc21i,WWc1r0,WWc1i0,WWc11r0,WWc11i0)
WWc1r0(j)=Wc1r; WWc1i0(j)=Wc1i; WWc2r0(j)=Wc2r; WWc2i0(j)=Wc2i;
WWc11r0(j)=Wc11r; WWc11i0(j)=Wc11i; WWc21r0(j)=Wc21r; WWc21i0(j)=Wc21i;

indoff=1;
end; // fin de if indoff
Wc1r0=Wc1r; Wc1i0=Wc1i; Wc2r0=Wc2r; Wc2i0=Wc2i;
Wc11r0=Wc11r; Wc11i0=Wc11i; Wc21r0=Wc21r; Wc21i0=Wc21i;
Wc1=complex(Wc1r,Wc1i); Wc2=complex(Wc2r,Wc2i); Wc11=complex(Wc11r,Wc11i);
Wc21=complex(Wc21r,Wc21i);
// selecciona puntos de círculo
if teta0==0 then
Fx1(j)=Wc21r; Fy1(j)=Wc21i; Gx1(j)=Wc2r; Gy1(j)=Wc2i;
end
if teta0==120 then
Fx2(j)=Wc21r; Fy2(j)=Wc21i; Gx2(j)=Wc2r; Gy2(j)=Wc2i;
end
if teta0==240 then
Fx3(j)=Wc21r; Fy3(j)=Wc21i; Gx3(j)=Wc2r; Gy3(j)=Wc2i;
end

write(u9,[j,O1x,ix,teta0,Nr,Ni,A11r,A11i,Wc1r,Wc1i,Wc2r,Wc2i,Wc11r,Wc11i,Wc21r,Wc21i],'(16
(f16.4)'));

end; // fin de for i
//else
// write(u6,[k1,j,i,ind'],'(4(f16.4)'));

end; // fin de if ind

endfunction
function [x1, y1, x2, y2]=nuevo(Cx1, Cy1, R22, Rp1)
d=sqrt(C1x^2+C1y^2);
if d<R22 then
h=(d+Rp1+R22)/2; else
h=(d-Rp1+R22)/2;
end
ang=atand(C1y,C1x);
x1=h*cosd(ang); y1=h*sind(ang);
x2=x1;y2=y1;
endfunction

function [Cx, Cy, R]=param_circulo(x1, y1, x2, y2, x3, y3)
// pruebas círculo
// Datos tres puntos determina los parámetros del círculo
//x1=-11.7163; y1=-0.0126; x2=-6.2341; y2=0.0184; x3=-9.9657; y3=3.1147;
A=[-2*x2+2*x1,-2*y2+2*y1,-2*x3+2*x1,-2*y3+2*y1];
B=[-x2^2+x1^2-y2^2+y1^2;-x3^2+x1^2-y3^2+y1^2];

```

```

de=det(A);
disp([de,x1,y1,x2,y2,x3,y3],"de,x1,y1,x2,y2,x3,y3");
//pause
if abs(de)>0.000001 then
    D=inv(A)*B;
    Cx=D(1); Cy=D(2);
    R=sqrt((x1-Cx)^2+(y1-Cy)^2); else
        Cx=0; Cy=0; R=0;
    end

disp([Cx,Cy,R], "Cx,Cy,R");
endfunction
function [xprom, yprom, xx12, yy12, xx13, yy13, xx23, yy23, dist, disc1, disc2,
disc3]=interseccion(Cx1, Cy1, Cx2, Cy2, Cx3, Cy3, R1, R2, R3)
    // pruebas círculo 2
    // Datos tres círculos, encuentra las dos líneas de interseccion
    //Cx1=-5.541529; Cy1=0.045436; R1=2.649218;
    //Cx2=-6.830059; Cy2=1.60642; R2=3.60435;
    //Cx3=-8.97717; Cy3=0.351564; R3=2.76323;
    dist=1000; xprom=0; yprom=0;
    disp([Cx1,Cy1,Cx2,Cy2,Cx3,Cy3,R1,R2,R3],"Cx1,Cy1,Cx2,Cy2,Cx3,Cy3,R1,R2,R3")
    A=[2*(Cx1-Cx2),2*(Cy1-Cy2);2*(Cx1-Cx3),2*(Cy1-Cy3)];
    B=[R2^2-R1^2+Cx1^2-Cx2^2+Cy1^2-Cy2^2;R3^2-R1^2+Cx1^2-Cx3^2+Cy1^2-Cy3^2];
    disp(A,"A")
    D=inv(A)*B;
    xi=D(1); yi=D(2);

    disp([xi,yi], "xi,yi");
    // calcula puntos en círculos
    a=(R2^2-R1^2+Cx1^2-Cx2^2+Cy1^2-Cy2^2)/(2*(Cy1-Cy2));
    b=-(Cx1-Cx2)/(Cy1-Cy2);
    c=(R3^2-R1^2+Cx1^2-Cx3^2+Cy1^2-Cy3^2)/(2*(Cy1-Cy3));
    d=-(Cx1-Cx3)/(Cy1-Cy3);
    e=(R3^2-R2^2+Cx2^2-Cx3^2+Cy2^2-Cy3^2)/(2*(Cy2-Cy3));
    f=-(Cx2-Cx3)/(Cy2-Cy3);
    // 1a cuadrática
    a1=1+b^2; a2=-2*Cx1+2*b*(a-Cy1); a3=Cx1^2+(a-Cy1)^2-R1^2;
    disc1=a2^2-4*a1*a3;
    if disc1>=0 then
        x1=(-a2+sqrt(disc1))/(2*a1);
        x2=(-a2-sqrt(disc1))/(2*a1);
        y1=a+b*x1; y2=a+b*x2;
        rhs1=(x1-xi)^2+(y1-yi)^2; rhs2=(x2-xi)^2+(y2-yi)^2;
        xx12=x1; yy12=y1
        if rhs1>rhs2 then
            xx12=x2; yy12=y2;
        end
        disp([xx12,yy12], "xx12,yy12")
    else
        disp(disc1,"disc1= no hay intersección entre círculo 1 y 2")
        xx12=0; yy12=0;
    end
endfunction

```

```

end

// 2a cuadrática
a1=1+d^2; a2=-2*Cx1+2*d*(c-Cy1); a3=Cx1^2+(c-Cy1)^2-R1^2;
disc2=a2^2-4*a1*a3;
if disc2>=0 then
    x1=(-a2+sqrt(disc2))/(2*a1);
    x2=(-a2-sqrt(disc2))/(2*a1);
    y1=c+d*x1; y2=c+d*x2;
    rhs1=(x1-xi)^2+(y1-yi)^2; rhs2=(x2-xi)^2+(y2-yi)^2;
    xx13=x1; yy13=y1
    if rhs1>rhs2 then
        xx13=x2; yy13=y2;
    end
    disp([xx13,yy13],"xx13,yy13")
else
    disp(disc2,"disc2= no hay intersección entre círculo 1 y 3")
    xx13=0; yy13=0;
end

// 3a cuadrática
a1=1+f^2; a2=-2*Cx2+2*f*(e-Cy2); a3=Cx2^2+(e-Cy2)^2-R2^2;
disc3=a2^2-4*a1*a3;
xx23=0; yy23=0;
if disc3>=0 then
    x1=(-a2+sqrt(disc3))/(2*a1);
    x2=(-a2-sqrt(disc3))/(2*a1);
    y1=e+f*x1; y2=e+f*x2;
    rhs1=(x1-xi)^2+(y1-yi)^2; rhs2=(x2-xi)^2+(y2-yi)^2;
    xx23=x1; yy23=y1
    if rhs1>rhs2 then
        xx23=x2; yy23=y2;
    end
    disp([xx23,yy23],"xx23,yy23")
    // modificación 4 febrero 2021
else
    disp(disc3,"disc3= no hay intersección entre círculo 2 y 3")
    xx23=0; yy23=0;
end

// determina distancia entre interseccion de rectas y círculos
if disc1>=0 then
    if disc2>=0&disc3>=0 then
        xprom=(xx12+xx13+xx23)/3;
        yprom=(yy12+yy13+yy23)/3;
        dist=sqrt((xprom-xi)^2+(yprom-yi)^2);
    end;

end

if dist<1000 then
    // pause
end

```

```
endfunction
```

```
function [rhs11, rhs12, rhs21, rhs22, teta221, teta222]=checaR2(x1, y1, x2, y2, R22, Rp21, C1x, C1y)
rhs11=x1^2+y1^2-R22^2; rhs12=(x1-C1x)^2+(y1-C1y)^2-Rp21^2;
rhs21=x2^2+y2^2-R22^2; rhs22=(x2-C1x)^2+(y2-C1y)^2-Rp21^2;
//x1=real(x1); y1=real(y1); x2=real(x2); y2=real(y2);
disp([x1,y1,x2,y2], "x1,y1,x2,y2...dentro de checaR2")
//z1=complex(x1,y1); z2=complex(x2,y2);
teta221=atand(y1,x1); teta222=atand(y2,x2);
endfunction
```

```
function [rhs11, rhs12, rhs21, rhs22, teta211, teta212]=checaRp21(x1, y1, x2, y2, R21, Rp22, C1x, C1y)
rhs11=x1^2+y1^2-R21^2; rhs12=(x1-C1x)^2+(y1-C1y)^2-Rp22^2;
rhs21=x2^2+y2^2-R21^2; rhs22=(x2-C1x)^2+(y2-C1y)^2-Rp22^2;
//z1=complex(x1,y1); z2=complex(x2,y2);
//x1=real(x1); y1=real(y1); x2=real(x2); y2=real(y2);
teta211=atand(y1,x1); teta212=atand(y2,x2);
endfunction
```

```
function [rhs11, rhs21, rhs12, rhs22]=polarcheck(R21, R22, teta211, teta212, teta221, teta222, r, R10, teta0)
z2=complex(R22*cosd(teta221),R22*sind(teta221));
z1=complex(R21*cosd(teta211),R21*sind(teta211));
z0=complex(R10*cosd(teta0),R10*sind(teta0));
zrh=z2-r*z1-(1-r)*z0;
rhs11=abs(zrh);
z2=complex(R22*cosd(teta222),R22*sind(teta222));
zrh=z2-r*z1-(1-r)*z0;
rhs21=abs(zrh);
z2=complex(R22*cosd(teta221),R22*sind(teta221));
z1=complex(R21*cosd(teta212),R21*sind(teta212));
zrh=z2-r*z1-(1-r)*z0;
rhs12=abs(zrh);
z2=complex(R22*cosd(teta222),R22*sind(teta222));
zrh=z2-r*z1-(1-r)*z0;
rhs22=abs(zrh);
```

```
endfunction
```

```
function [xx1, yy1, xx2, yy2, xxx1, yyy1, xxx2, yyy2]=seleccionpolar(teta211, teta212, teta221, teta222, rhs11, rhs21, rhs12, rhs22, R21, R22)
xx1=0; xx2=0; xxx1=0; xxx2=0; yy1=0; yy2=0; yyy1=0; yyy2=0;
if rhs11<0.1 then
xx1=R21*cosd(teta211); yy1=R21*sind(teta211); xx2=R22*cosd(teta221);
yy2=R22*sind(teta221);
end
if rhs21<0.1 then
xx1=R21*cosd(teta211); yy1=R21*sind(teta211); xx2=R22*cosd(teta222);
yy2=R22*sind(teta222);
end
```



```

if rhs12<0.1 then
xxx1=R21*cosd(teta212); yyy1=R21*sind(teta212); xxx2=R22*cosd(teta221);
yyy2=R22*sind(teta221);
end
if rhs22<0.1 then
xxx1=R21*cosd(teta212); yyy1=R21*sind(teta212); xxx2=R22*cosd(teta222);
yyy2=R22*sind(teta222);
end
endfunction

function [A11r, A11i, A21r, A21i, A111r, A111i, A211r, A211i]=uniforme(A11r, A11i, A21r,
A21i, A111r, A111i, A211r, A211i, A11r0, A11i0, A21r0, A21i0, A111r0, A111i0, A211r0,
A211i0)
dif1=(A11r-A11r0)^2+(A11i-A11i0)^2+(A21r-A21r0)^2+(A21i-A21i0)^2;
dif2=(A111r-A11r0)^2+(A111i-A11i0)^2+(A211r-A21r0)^2+(A211i-A21i0)^2;
dif3=(A11r-A111r0)^2+(A11i-A111i0)^2+(A21r-A211r0)^2+(A21i-A211i0)^2;
dif4=(A111r-A111r0)^2+(A111i-A111i0)^2+(A211r-A211r0)^2+(A211i-A211i0)^2;
if dif1>dif2 then
T11r=A11r; T11i=A11i; T21r=A21r; T21i=A21i;
A11r=A111r; A11i=A111i; A21r=A211r; A21i=A211i;
A111r=T11r; A111i=T11i; A211r=T21r; A211i=T21i;
end
endfunction

function [Wc1r, Wc1i, Wc2r, Wc2i, Wc11r, Wc11i, Wc21r, Wc21i, jx]=uniformeW(j, Wc1r,
Wc1i, Wc2r, Wc2i, Wc11r, Wc11i, Wc21r, Wc21i, Wc1r0, Wc1i0, Wc2r0, Wc2i0, Wc11r0,
Wc11i0, Wc21r0, Wc21i0)
// dif1=(Wc1r-Wc1r0)^2+(Wc1i-Wc1i0)^2+(Wc2r-Wc2r0)^2+(Wc2i-Wc2i0)^2;
dif2=(Wc1r-Wc21r0)^2+(Wc1i-Wc21i0)^2;
// dif3=(Wc11r-Wc1r0)^2+(Wc11i-Wc1i0)^2+(Wc21r-Wc2r0)^2+(Wc21i-Wc2i0)^2;
dif4=(Wc11r-Wc21r0)^2+(Wc11i-Wc21i0)^2;
dif1=(Wc1r-Wc1r0)^2+(Wc1i-Wc1i0)^2;
dif3=(Wc11r-Wc1r0)^2+(Wc11i-Wc1i0)^2;
//
write(u5,[j,Wc1r,Wc1i,Wc11r,Wc11i,Wc1r0,Wc1i0,Wc11r0,Wc11i0,dif1,dif2,dif3,dif4],'(13(f16.4))'
);
// if dif3>dif1 | dif4>dif2 then
if dif1>dif2&dif4>dif3 then
//if dif1>500 & dif4>500 then
T11r=Wc1r; T11i=Wc1i; T21r=Wc2r; T21i=Wc2i;
Wc1r=Wc11r; Wc1i=Wc11i; Wc2r=Wc21r; Wc2i=Wc21i;
Wc11r=T11r; Wc11i=T11i; Wc21r=T21r; Wc21i=T21i;
end
end

// define jx que indica que hay que intercambiar toda la fase
jx=0;
ddif1=abs(Wc1r-Wc2r)+abs(Wc1i-Wc2i);
ddif2=abs(Wc1r-Wc11r)+abs(Wc1i-Wc11i);
if ddif2<ddif1 then
jx=1;
end
// if dif1>dif2 then

```

```

// end

endfunction
function [Wc1r, Wc1i, Wc2r, Wc2i, Wc11r, Wc11i, Wc21r, Wc21i]=uniformeoff(j, O1x,
Wc1r, Wc1i, Wc2r, Wc2i, Wc11r, Wc11i, Wc21r, Wc21i, WWc1r0, WWc1i0, WWc11r0,
WWc11i0)
    dif1=(Wc1r-WWc1r0(j))^2+(Wc1i-WWc1i0(j))^2;
    dif2=(Wc1r-WWc11r0(j))^2+(Wc1i-WWc11i0(j))^2;
    // dif3=(Wc11r-WWc11r0(j))^2+(Wc11i-WWc11i0(j))^2+(Wc21r-WWc21r0(j))^2+(Wc21i-
    WWc21i0(j))^2;
    // dif4=(Wc11r-WWc11r0(j))^2+(Wc11i-WWc11i0(j))^2+(Wc21r-WWc21r0(j))^2+(Wc21i-
    WWc21i0(j))^2;
    if j==1 then
        //
        write(u5,[j,O1x,Wc1r,Wc1i,Wc11r,Wc11i,WWc1r0(j),WWc1i0(j),WWc11r0(j),WWc11i0(j),dif1,dif2
        ],'(12(f16.4))');
    end

    if dif2<dif1 then
        T11r=Wc1r; T11i=Wc1i; T21r=Wc2r; T21i=Wc2i;
        Wc1r=Wc11r; Wc1i=Wc11i; Wc2r=Wc21r; Wc2i=Wc21i;
        Wc11r=T11r; Wc11i=T11i; Wc21r=T21r; Wc21i=T21i;
    end

endfunction

function [xx1, yy1, xx2, yy2, ind1, xxx1, yyy1, xxx2, yyy2, ind2]=seleccion(r, x11, x12, y11,
y12, xx11, xx12, yy11, yy12, R21, R22, zR10)
// ensaya x11,y11,xx11,yy11
ind1=1; ind2=1;
N2N11r=x11-real(zR10); N2N11i=y11-imag(zR10);
N2N12r=x12-real(zR10); N2N12i=y12-imag(zR10);
N2N13r=xx11-real(zR10); N2N13i=yy11-imag(zR10);
N2N14r=xx12-real(zR10); N2N14i=yy12-imag(zR10);
xx1=0; yy1=0; xx2=0; yy2=0; ind=1;
xxx1=0; yyy1=0; xxx2=0; yyy2=0;

[ind11]=ensayo(r,N2N11r,N2N11i,N2N13r,N2N13i)
disp(ind11,"ind11");
[ind12]=ensayo(r,N2N11r,N2N11i,N2N14r,N2N14i)
disp(ind12,"ind12");
[ind21]=ensayo(r,N2N12r,N2N12i,N2N13r,N2N13i)
disp(ind21,"ind21");
[ind22]=ensayo(r,N2N12r,N2N12i,N2N14r,N2N14i)
disp(ind22,"ind22");
if ind11==0 then
xx1=xx11; yy1=yy11; xx2=x11; yy2=y11; ind1=0; else
    if ind12==0 then
        xx1=xx12; yy1=yy12; xx2=x11; yy2=y11; ind1=0;
    end
end

```

```

end
end
if ind21==0 then
xxx1=xx11; yyy1=yy11; xxx2=x12; yyy2=y12; ind2=0; else
  if ind22==0 then
    xxx1=xx12; yyy1=yy12; xxx2=x12; yyy2=y12; ind2=0;
  end
end
end

endfunction
function [ind11]=ensayo(r, N2N11r, N2N11i, N2N13r, N2N13i)
//N2N11r=real(N2N11r); N2N11i=real(N2N11i); N2N13r=real(N2N13r); N2N13i=real(N2N13i);
z1=complex(N2N11r,N2N11i); z2=complex(N2N13r,N2N13i);
z=z1/z2;
ind1=1; ind2=1; ind11=1;
//if abs(real(z)-real(r))<1 then
  if abs(real(z)-real(r))<0.05 then
    ind1=0;
  end
//if abs(imag(z)-imag(r))<1 then
  if abs(imag(z)-imag(r))<0.05 then
    ind2=0;
  end
if ind1==0 & ind2==0 then
  ind11=0;
end;
endfunction

function [Wc1r, Wc1i, Wc2r, Wc2i, Wc11r, Wc11i, Wc21r, Wc21i]=intercambio(jx1, jx2, jx3,
sum1r, sum1i, sum2r, sum2i, sum3r, sum3i, sum4r, sum4i, sum5r, sum5i, sum6r, sum6i)
  Wc1r=(sum1r+sum3r+sum5r)/(i1+i3+i5); Wc1i=(sum1i+sum3i+sum5i)/(i1+i3+i5);
  Wc2r=(sum2r+sum4r+sum6r)/(i2+i4+i6); Wc2i=(sum2i+sum4i+sum6i)/(i2+i4+i6);
  Wc11r=(sum11r+sum31r+sum51r)/(i1+i3+i5); Wc11i=(sum11i+sum31i+sum51i)/(i1+i3+i5);
  Wc21r=(sum21r+sum41r+sum61r)/(i2+i4+i6); Wc21i=(sum21i+sum41i+sum61i)/(i2+i4+i6);

  if jx1==1 then
    Wc1r=(zum1r+sum3r+sum5r)/(i1+i3+i5); Wc1i=(zum1i+sum3i+sum5i)/(i1+i3+i5);
    Wc2r=(sum2r+sum4r+sum6r)/(i2+i4+i6); Wc2i=(sum2i+sum4i+sum6i)/(i2+i4+i6);
    Wc11r=(sum11r+sum31r+sum51r)/(i1+i3+i5); Wc11i=(sum11i+sum31i+sum51i)/(i1+i3+i5);
    Wc21r=(zum21r+sum41r+sum61r)/(i2+i4+i6); Wc21i=(zum21i+sum41i+sum61i)/(i2+i4+i6);
    dif1=abs(Wc1r-Wc2r)+abs(Wc1i+Wc2i);
    if dif1>0.01 then
      Wc1r=(sum1r+sum3r+sum5r)/(i1+i3+i5); Wc1i=(sum1i+sum3i+sum5i)/(i1+i3+i5);
      Wc2r=(zum2r+sum4r+sum6r)/(i2+i4+i6); Wc2i=(zum2i+sum4i+sum6i)/(i2+i4+i6);
      Wc11r=(zum11r+sum31r+sum51r)/(i1+i3+i5); Wc11i=(zum11i+sum31i+sum51i)/(i1+i3+i5);
      Wc21r=(sum21r+sum41r+sum61r)/(i2+i4+i6); Wc21i=(sum21i+sum41i+sum61i)/(i2+i4+i6);
    end
  end
  if jx2==1 then
    Wc1r=(sum1r+zum3r+sum5r)/(i1+i3+i5); Wc1i=(sum1i+zum3i+sum5i)/(i1+i3+i5);
    Wc2r=(sum2r+sum4r+sum6r)/(i2+i4+i6); Wc2i=(sum2i+sum4i+sum6i)/(i2+i4+i6);

```

```

Wc11r=(sum11r+sum31r+sum51r)/(i1+i3+i5); Wc11i=(sum11i+sum31i+sum51i)/(i1+i3+i5);
Wc21r=(sum21r+zum41r+sum61r)/(i2+i4+i6); Wc21i=(sum21i+zum41i+sum61i)/(i2+i4+i6);
dif1=abs(Wc1r-Wc2r)+abs(Wc1i+Wc2i);
if dif1>0.01 then
Wc1r=(sum1r+sum3r+sum5r)/(i1+i3+i5); Wc1i=(sum1i+sum3i+sum5i)/(i1+i3+i5);
Wc2r=(sum2r+zum4r+sum6r)/(i2+i4+i6); Wc2i=(sum2i+zum4i+sum6i)/(i2+i4+i6);
Wc11r=(sum11r+zum31r+sum51r)/(i1+i3+i5); Wc11i=(sum11i+zum31i+sum51i)/(i1+i3+i5);
Wc21r=(sum21r+sum41r+sum61r)/(i2+i4+i6); Wc21i=(sum21i+sum41i+sum61i)/(i2+i4+i6);
end
end

```

```

if jx3==1 then
Wc1r=(sum1r+sum3r+zum5r)/(i1+i3+i5); Wc1i=(sum1i+sum3i+zum5i)/(i1+i3+i5);
Wc2r=(sum2r+sum4r+sum6r)/(i2+i4+i6); Wc2i=(sum2i+sum4i+sum6i)/(i2+i4+i6);
Wc11r=(sum11r+sum31r+sum51r)/(i1+i3+i5); Wc11i=(sum11i+sum31i+sum51i)/(i1+i3+i5);
Wc21r=(sum21r+sum41r+zum61r)/(i2+i4+i6); Wc21i=(sum21i+sum41i+zum61i)/(i2+i4+i6);
dif1=abs(Wc1r-Wc2r)+abs(Wc1i+Wc2i);
if dif1>0.01 then
Wc1r=(sum1r+sum3r+sum5r)/(i1+i3+i5); Wc1i=(sum1i+sum3i+sum5i)/(i1+i3+i5);
Wc2r=(sum2r+sum4r+zum6r)/(i2+i4+i6); Wc2i=(sum2i+sum4i+zum6i)/(i2+i4+i6);
Wc11r=(sum11r+sum31r+zum51r)/(i1+i3+i5); Wc11i=(sum11i+sum31i+zum51i)/(i1+i3+i5);
Wc21r=(sum21r+sum41r+sum61r)/(i2+i4+i6); Wc21i=(sum21i+sum41i+sum61i)/(i2+i4+i6);
end
end

```

endfunction

```
file('close',file())
```

```

u1=file('open','C:\scilab\prueba.txt','unknown')
u2=file('open','C:\scilab\prueba3.txt','unknown')
u4=file('open','C:\scilab\corrfa4.dat','unknown','formatted')
u6=file('open','C:\scilab\corrfa6.dat','unknown','formatted')
u9=file('open','C:\scilab\corrfa9.dat','unknown','formatted')
u3=file('open','C:\scilab\corrfa3.dat','unknown','formatted')
u$=file('open','C:\scilab\corrfa4.dat','unknown','formatted')
u8=file('open','C:\scilab\corrfa8.dat','unknown','formatted')
u7=file('open','C:\scilab\corrfa7.dat','unknown','formatted')
u5=file('open','C:\scilab\corrfa5.dat','unknown','formatted')

```

```

labels5=["Wp1m";"Wp1ang"];
[ok,Wp1m,Wp1ang]=getvalue("Wp1m,Wp1ang",labels5,...
list("vec",1,"vec",1),["";""])

```

```

labels4=["Wp2m";"Wp2ang"];
[ok,Wp2m,Wp2ang]=getvalue("Wp2m,Wp2ang",labels4,...
list("vec",1,"vec",1),["";""])

```

```

labels3=["Offin";"Offend"];
[ok,Offin,Offend]=getvalue("Offin,Offend",labels3,...

```

```

list("vec",1,"vec",1),["";""])

eO1=1; nint=2; nint2=nint^2; iflag=0;
nint=3; nint2=1;
//Wp1m=8; Wp1ang=15; Wp2m=12.1; Wp2ang=30;
W1=complex(Wp1m*cosd(Wp1ang),Wp1m*sind(Wp1ang));
W2=complex(Wp2m*cosd(Wp2ang),Wp2m*sind(Wp2ang));
//teta121=327; teta131=150; teta122=-73.51; teta132=144.74;

r=W2/W1;
// Lee datos
i=0;
while i<nint &iflag==0;
    disp("antes de i=i+1;")
    i=i+1;
//for i=1:4
// load(u1,[x1,x2,x3]);
[B]=read(u1,1,9);
if B(1)==0 then
    iflag=1;
end
disp(iflag,"iflag=");
disp("B",[B(1),B(2),B(3),B(4),B(5),B(6)],B(7),B(8),B(9));
select i
case 1 then
    RR10(1)=B(1); RR20(1)=B(4); RR30(1)=B(7);
case 2 then
    RR11(1)=B(1); RR21(1)=B(4); RR31(1)=B(7);
case 3 then
    RR12(1)=B(1); RR22(1)=B(4); RR32(1)=B(7);
end; // fin de select

end; // fin de while

// Itera 10 valores de O1, O2, O3
//write(u6," eO1,Wcr1,Wci1,dist1,disc11,disc21,disc31,Wcr2,Wci2,dist2,disc12,disc22,disc32");
write(u9," j,O1x,jx,teta0,Nr,Ni,A11r,A11i,Wc1r,Wc1i,Wc2r,Wc2i,Wc11r,Wc11i,Wc21r,Wc21i");
write(u3," eO1,Wcr1,Wci1,dist1,disc11,disc21,disc31,Wcr2,Wci2,dist2,disc12,disc22,disc32");
write(u4," eO1,Wcr1,Wci1,dist1,disc11,disc21,disc31,Wcr2,Wci2,dist2,disc12,disc22,disc32");
write(u8,"
eO1,Wc1r,Wc1i,Wc3r,Wc3i,Wc5r,Wc5i,Wc11r,Wc11i,Wc31r,Wc31i,Wc51r,Wc51i,Wcr,Wci,Wccr
,Wcci");
write(u7," eO1,Wc1r,Wc1i,Wc2r,Wc2i,Wc11r,Wc11i,Wc21r,Wc21i");
write(u5," eO1,Wcr1,Wci1,dist1,disc11,disc21,disc31,Wcr2,Wci2,dist2,disc12,disc22,disc32");
Wc1r0=0; Wc1i0=0; Wc2r0=0; Wc2i0=0; Wc11r0=0; Wc11i0=0; Wc21r0=0; Wc21i0=0;
for i=1:3
WWc1r0(i)=0; WWc1r0(i)=0; WWc1i0(i)=0; WWc11r0(i)=0; WWc11i0(i)=0;
end
k=Offin-1; indica=0;
while k<Offend then

```

```

k=k+1;
eO1=k;
sum1r=0; sum1i=0; sum2r=0; sum2i=0; sum3r=0; sum3i=0; i1=0; i2=0; i3=0;
sum11r=0; sum11i=0; sum21r=0; sum21i=0; sum31r=0; sum31i=0;
sum4r=0; sum4i=0; sum5r=0; sum5i=0; sum6r=0; sum6i=0; i4=0; i5=0; i6=0;
sum41r=0; sum41i=0; sum51r=0; sum51i=0; sum61r=0; sum61i=0;
zum1r=0; zum1i=0; zum2r=0; zum2i=0; zum3r=0; zum3i=0;
zum11r=0; zum11i=0; zum21r=0; zum21i=0; zum31r=0; zum31i=0;
zum4r=0; zum4i=0; zum5r=0; zum5i=0; zum6r=0; zum6i=0;
zum41r=0; zum41i=0; zum51r=0; zum51i=0; zum61r=0; zum61i=0;
//for k=1,nint2
//select k
//case 1 then
    R10=RR10(1); R20=RR20(1); R30=RR30(1);
    R11=RR11(1); R21=RR21(1); R31=RR31(1);
    R12=RR12(1); R22=RR22(1); R32=RR32(1);
//end; // fin de select

for j=1:3
indoff=0;
//for i=1:80
// eeO1=eO1+5*(i-1)*0.05
// O1=complex(eeO1,0);

//write(u4," i,reL,m1,tteta1,teta11");
//write(u6," i,rhs2,reL,d11,d21");
// calcula offset 2
disp(j,"OFFSET j")
disp("");
disp("");
disp("");
disp("");

//write(u6," R11,R12,R21,R22,teta121,teta122,Wp1m,Wp1ang,Wp2m,Wp2ang");
//write(u6,[R11,R12,R21,R22,teta121,teta122,Wp1m,Wp1ang,Wp2m,Wp2ang]','(10(f16.4))');

for k1=1:360
    teta0=(k1-1)*1;

select j
case 1 then
    disp(indica,"indica")

[Wc1,Wc2,Wc11,Wc21,indica,A11r0,A11i0,A21r0,A21i0,A111r0,A111i0,A211r0,A211i0,Wc1r0,
Wc1i0,Wc2r0,Wc2i0,Wc11r0,Wc11i0,Wc21r0,Wc21i0,indoff,WWc1r0,WWc1i0,WWc11r0,WWc1
1i0,Fx1,Fy1,Fx2,Fy2,Fx3,Fy3,Gx1,Gy1,Gx2,Gy2,Gx3,Gy3,jx1]=offset(i,k1,j,R11,R12,R10,r,W1,
W2,teta0,eO1,indica,Wc1r0,Wc1i0,Wc2r0,Wc2i0,Wc11r0,Wc11i0,Wc21r0,Wc21i0,indoff,WWc1r
0,WWc1i0,WWc11r0,WWc11i0)
    sum1r=sum1r+real(Wc1); sum1i=sum1i+imag(Wc1); sum11r=sum11r+real(Wc11);
    sum11i=sum11i+imag(Wc11); i1=i1+1;

```

```

sum2r=sum2r+real(Wc2); sum2i=sum2i+imag(Wc2); sum21r=sum21r+real(Wc21);
sum21i=sum21i+imag(Wc21); i2=i2+1;
zum1r=zum1r+real(Wc2); zum1i=zum1i+imag(Wc2); zum11r=zum11r+real(Wc21);
zum11i=zum11i+imag(Wc21);
zum2r=zum2r+real(Wc1); zum2i=zum2i+imag(Wc1); zum21r=zum21r+real(Wc11);
zum21i=zum21i+imag(Wc11);

```

```

if teta0==0 then
  F1x1=Fx1(j); F1y1=Fy1(j); G1x1=Gx1(j); G1y1=Gy1(j);
end

```

```

end
if teta0==120 then
  F1x2=Fx2(j); F1y2=Fy2(j); G1x2=Gx2(j); G1y2=Gy2(j);
end

```

```

if teta0==240 then
  F1x3=Fx3(j); F1y3=Fy3(j); G1x3=Gx3(j); G1y3=Gy3(j);
end

```

```

end
case 2 then

```

```

[Wc1,Wc2,Wc11,Wc21,indica,A11r0,A11i0,A21r0,A21i0,A111r0,A111i0,A211r0,A211i0,Wc1r0,
Wc1i0,Wc2r0,Wc2i0,Wc11r0,Wc11i0,Wc21r0,Wc21i0,indoff,WWc1r0,WWc1i0,WWc11r0,WWc1
1i0,Fx1,Fy1,Fx2,Fy2,Fx3,Fy3,Gx1,Gy1,Gx2,Gy2,Gx3,Gy3,jx2]=offset(i,k1,j,R21,R22,R20,r,W1,
W2,teta0,eO1,indica,Wc1r0,Wc1i0,Wc2r0,Wc2i0,Wc11r0,Wc11i0,Wc21r0,Wc21i0,indoff,WWc1r
0,WWc1i0,WWc11r0,WWc11i0)

```

```

sum3r=sum3r+real(Wc1); sum3i=sum3i+imag(Wc1); sum31r=sum31r+real(Wc11);
sum31i=sum31i+imag(Wc11); i3=i3+1;
sum4r=sum4r+real(Wc2); sum4i=sum4i+imag(Wc2); sum41r=sum41r+real(Wc21);
sum41i=sum41i+imag(Wc21); i4=i4+1;
zum3r=zum3r+real(Wc2); zum3i=zum3i+imag(Wc2); zum31r=zum31r+real(Wc21);
zum31i=zum31i+imag(Wc21);
zum4r=zum4r+real(Wc1); zum4i=zum4i+imag(Wc1); zum41r=zum41r+real(Wc11);
zum41i=zum41i+imag(Wc11);

```

```

if teta0==0 then
  F2x1=Fx1(j); F2y1=Fy1(j); G2x1=Gx1(j); G2y1=Gy1(j);
end

```

```

end
if teta0==120 then
  F2x2=Fx2(j); F2y2=Fy2(j); G2x2=Gx2(j); G2y2=Gy2(j);
end

```

```

if teta0==240 then
  F2x3=Fx3(j); F2y3=Fy3(j); G2x3=Gx3(j); G2y3=Gy3(j);
end

```

```

end
case 3 then

```

```

[Wc1,Wc2,Wc11,Wc21,indica,A11r0,A11i0,A21r0,A21i0,A111r0,A111i0,A211r0,A211i0,Wc1r0,
Wc1i0,Wc2r0,Wc2i0,Wc11r0,Wc11i0,Wc21r0,Wc21i0,indoff,WWc1r0,WWc1i0,WWc11r0,WWc1
1i0,Fx1,Fy1,Fx2,Fy2,Fx3,Fy3,Gx1,Gy1,Gx2,Gy2,Gx3,Gy3,jx3]=offset(i,k1,j,R31,R32,R30,r,W1,
W2,teta0,eO1,indica,Wc1r0,Wc1i0,Wc2r0,Wc2i0,Wc11r0,Wc11i0,Wc21r0,Wc21i0,indoff,WWc1r
0,WWc1i0,WWc11r0,WWc11i0)

```

```

sum5r=sum5r+real(Wc1); sum5i=sum5i+imag(Wc1); sum51r=sum51r+real(Wc11);
sum51i=sum51i+imag(Wc11); i5=i5+1;

```

```

sum6r=sum6r+real(Wc2); sum6i=sum6i+imag(Wc2); sum61r=sum61r+real(Wc21);
sum61i=sum61i+imag(Wc21); i6=i6+1;
zum5r=zum5r+real(Wc2); zum5i=zum5i+imag(Wc2); zum51r=zum51r+real(Wc21);
zum51i=zum51i+imag(Wc21);
zum6r=zum6r+real(Wc1); zum6i=zum6i+imag(Wc1); zum61r=zum61r+real(Wc11);
zum61i=zum61i+imag(Wc11);

if teta0==0 then
    F3x1=Fx1(j); F3y1=Fy1(j); G3x1=Gx1(j); G3y1=Gy1(j);
end
if teta0==120 then
    F3x2=Fx2(j); F3y2=Fy2(j); G3x2=Gx2(j); G3y2=Gy2(j);
end
if teta0==240 then
    F3x3=Fx3(j); F3y3=Fy3(j); G3x3=Gx3(j); G3y3=Gy3(j);
end
end ; // fin de select
Wc1r=real(Wc1); Wc1i=imag(Wc1); Wc2r=real(Wc2); Wc2i=imag(Wc2); Wc11r=real(Wc11);
Wc11i=imag(Wc11); Wc21r=real(Wc21); Wc21i=imag(Wc21);
// write(u7,j,Wc1r,Wc1i,Wc2r,Wc2i,Wc11r,Wc11i,Wc21r,Wc21i,sum1r,sum1i,'(11(f16.4))');

end; // fin de for k1
//end; // fin de for i
end; // fin de for j

// calcula los parámetros de círculos y la intersección por cada fase
disp([F1x1,F1y1,F1x2,F1y2,F1x3,F1y3],"F1x1,F1y1,F1x2,F1y2,F1x3,F1y3")
disp([F2x1,F2y1,F2x2,F2y2,F2x3,F2y3],"F2x1,F2y1,F2x2,F2y2,F2x3,F2y3")
disp([F3x1,F3y1,F3x2,F3y2,F3x3,F3y3],"F3x1,F3y1,F3x2,F3y2,F3x3,F3y3")
[Cx12,Cy12,R12]=param_circulo(F1x1,F1y1,F1x2,F1y2,F1x3,F1y3)
[Cx22,Cy22,R22]=param_circulo(F2x1,F2y1,F2x2,F2y2,F2x3,F2y3)
[Cx32,Cy32,R32]=param_circulo(F3x1,F3y1,F3x2,F3y2,F3x3,F3y3)
[Wcr2,Wci2,xx12,yy12,xx13,yy13,xx23,yy23,disc2,disc12,disc22,disc32]=interseccion(Cx12,Cy12,Cx22,Cy22,Cx32,Cy32,R12,R22,R32)
[Cx11,Cy11,R11]=param_circulo(G1x1,G1y1,G1x2,G1y2,G1x3,G1y3)
[Cx21,Cy21,R21]=param_circulo(G2x1,G2y1,G2x2,G2y2,G2x3,G2y3)
[Cx31,Cy31,R31]=param_circulo(G3x1,G3y1,G3x2,G3y2,G3x3,G3y3)
[Wcr1,Wci1,xx12,yy12,xx13,yy13,xx23,yy23,disc1,disc11,disc21,disc31]=interseccion(Cx11,Cy11,Cx21,Cy21,Cx31,Cy31,R11,R21,R31)

write(u3,[eO1,Wcr1,Wci1,disc1,disc11,disc21,disc31,Wcr2,Wci2,disc2,disc12,disc22,disc32],'(13(f16.4))');
if disc11|disc12<0 then
    // permuta la fase 1

[Wcr2,Wci2,xx12,yy12,xx13,yy13,xx23,yy23,disc2,disc1,disc2,disc3]=interseccion(Cx11,Cy11,Cx22,Cy22,Cx32,Cy32,R11,R22,R32)

[Wcr1,Wci1,xx12,yy12,xx13,yy13,xx23,yy23,disc1,disc1,disc2,disc3]=interseccion(Cx12,Cy12,Cx21,Cy21,Cx31,Cy31,R12,R21,R31)

```



```

write(u3,[eO1,Wcr1,Wci1,dist1,disc11,disc21,disc31,Wcr2,Wci2,dist2,disc12,disc22,disc32]','(13(
f16.4)'));
end
if disc21|disc22<0 then
    // permuta la fase 3

[Wcr2,Wci2,xx12,yy12,xx13,yy13,xx23,yy23,dist2,disc1,disc2,disc3]=interseccion(Cx12,Cy12,C
x22,Cy22,Cx31,Cy31,R12,R22,R31)

[Wcr1,Wci1,xx12,yy12,xx13,yy13,xx23,yy23,dist1,disc1,disc2,disc3]=interseccion(Cx11,Cy11,C
x21,Cy21,Cx32,Cy32,R11,R21,R32)

write(u3,[eO1,Wcr1,Wci1,dist1,disc11,disc21,disc31,Wcr2,Wci2,dist2,disc12,disc22,disc32]','(13(
f16.4)'));
end
if disc31|disc32<0 then
    // permuta la fase 2

[Wcr2,Wci2,xx12,yy12,xx13,yy13,xx23,yy23,dist2,disc1,disc2,disc3]=interseccion(Cx12,Cy12,C
x21,Cy21,Cx32,Cy32,R12,R21,R32)

[Wcr1,Wci1,xx12,yy12,xx13,yy13,xx23,yy23,dist1,disc1,disc2,disc3]=interseccion(Cx11,Cy11,C
x22,Cy22,Cx31,Cy31,R11,R22,R31)

write(u3,[eO1,Wcr1,Wci1,dist1,disc11,disc21,disc31,Wcr2,Wci2,dist2,disc12,disc22,disc32]','(13(
f16.4)'));
end

Wc1r=sum1r/i1; Wc1i=sum1i/i1; Wc2r=sum2r/i2; Wc2i=sum2i/i2;
Wc11r=sum11r/i1; Wc11i=sum11i/i1; Wc21r=sum21r/i2; Wc21i=sum21i/i2;
Wc3r=sum3r/i3; Wc3i=sum3i/i3; Wc31r=sum31r/i3; Wc31i=sum31i/i3;
Wc4r=sum4r/i4; Wc4i=sum4i/i4; Wc41r=sum41r/i4; Wc41i=sum41i/i4;
Wc5r=sum5r/i5; Wc5i=sum5i/i5; Wc51r=sum51r/i5; Wc51i=sum51i/i5;
Wc6r=sum6r/i6; Wc6i=sum6i/i6; Wc61r=sum61r/i6; Wc61i=sum61i/i6;

Wcr=(Wc1r+Wc3r+Wc5r)/3; Wci=(Wc1i+Wc3i+Wc5i)/3; Wccr=(Wc11r+Wc31r+Wc51r)/3;
Wcci=(Wc11i+Wc31i+Wc51i)/3;
write(u8,[eO1,Wc1r,Wc1i,Wc3r,Wc3i,Wc5r,Wc5i,Wc11r,Wc11i,Wc31r,Wc31i,Wc51r,Wc51i,Wcr,
Wci,Wccr,Wcci]','(17(f16.4)'));
// promedio de las tres fases
// primero checa intercambio
[Wc1r,Wc1i,Wc2r,Wc2i,Wc11r,Wc11i,Wc21r,Wc21i]=intercambio(jx1,jx2,jx3,sum1r,sum1i,sum2r
,sum2i,sum3r,sum3i,sum4r,sum4i,sum5r,sum5i,sum6r,sum6i)

write(u7,[eO1,Wc1r,Wc1i,Wc2r,Wc2i,Wc11r,Wc11i,Wc21r,Wc21i]','(9(f16.4)'));
end; // fi de while k

file('close',file())

```

A4. Algoritmo 8H16

```
funcprot(0)
function [Wc1, Wc2, indica, A11r0, A11i0, A21r0, A21i0, Wc1r0, Wc1i0, Wc2r0, Wc2i0,
indoff, Fx1, Fy1, Fx2, Fy2, Fx3, Fy3, Gx1, Gy1, Gx2, Gy2, Gx3, Gy3, jx, jy, AA11, teta10,
sign1, disc0, s10, s20]=offset(i, k1, j, R21, R22, R10, r, W1, W2, teta0, eO1, indica, Wc1r0,
Wc1i0, Wc2r0, Wc2i0, Wc21r0, Wc21i0, indoff, AA11, teta10, sign1, disc0, s10, s20)
    teta121=0; teta122=0; ind=0; x1=0; y1=0; x2=0; y2=0; zO2=complex(0,0); jy=0;
    // jy es el indicador de que se aplicó una aproximación a la solución (cuando vale 1)
// calcula fase 1
disp([R21,R22,r],"R21,R22,r")

for i=1:3
    Fx1(i)=0; Fy1(i)=0; Fx2(i)=0; Fy2(i)=0; Fx3(i)=0; Fy3(i)=0;
    Gx1(i)=0; Gy1(i)=0; Gx2(i)=0; Gy2(i)=0; Gx3(i)=0; Gy3(i)=0;
end

//calcula punto de cruce en círculo de R2

zR10=complex(R10*cosd(teta0),R10*sind(teta0)); zR21=complex(R21,0);
zR22=complex(R22,0);
z=(1-r)*zR10;
real0=real(z); imag0=imag(z);
rr=real(r); ri=imag(r);
a=2*R21*(rr*real0+ri*imag0);
b=2*R21*(rr*imag0-ri*real0);
c=R22^2-R21^2*(rr^2+ri^2)-real0^2-imag0^2;
A=a^2+b^2;
B=-2*b*c;
C=c^2-a^2;
disc=B^2-4*A*C;
disp([a,b,c,A,B,C,disc],"a,b,c,A,B,C,disc")

[disc,discmas,discmen,jak]=mindisc(R21,R22,zR10,r,teta0)
disp([disc,discmas,discmen,jak],"disc,discmas,discmen,jak")
//if teta0==52 then
// pause
//end

if teta0==0 then
    sign1=1; disc0=disc;
end
//if abs(disc)<10 then
// if abs(disc/disc0)<1e-06 then
    if jak==1 then
        sign1=-sign1;
        disc=0; // prueba aproximar disc
    end
end

if disc>=0 then
    sen1=(-B+sign1*sqrt(disc))/(2*A);
    sen2=(-B-sign1*sqrt(disc))/(2*A);
end
```

```

if abs(sen1)>1 | abs(sen2)>1 then
    sen1=1; sen2=1; jy=1;
end

s1=1; s2=1; ccos1=0; ccos2=0;
rel=abs(a/c);
// if abs(a)>0.0001 then
    if rel>1e-04 then
        ccos1=(c-b*sen1)/a; s1=ccos1/abs(ccos1);
        ccos2=(c-b*sen2)/a; s2=ccos2/abs(ccos2);
s10=s1; s20=s2;
    else
        s1=s10; s2=s20;
    end

    cos1=s1*sqrt(1-sen1^2);
    cos2=s2*sqrt(1-sen2^2);
    disp([ccos1,ccos2,s1,s2,cos1,cos2,sen1,sen2],"ccos1,ccos2,s1,s2,cos1,cos2,sen1,sen2")
    disp([a,c,rel],"a,c,rel")

// checa la continuidad de teta 1
teta1=atan(sen1,cos1); teta2=atan(sen2,cos2);

if teta1<0 then
    teta1=teta1+2*%pi
end
if teta2<0 then
    teta2=teta2+2*%pi;
end

if teta0==0 then
teta10=atan(sen1,cos1);
    if teta10<0 then
        teta10=teta10+2*%pi;
    end;
else
    if abs(teta10-2*%pi)<0.019 then
        teta10=teta10-2*%pi;
    end
    disp([teta1,teta2,teta10],"teta1,teta2,teta10")
    dif1=teta1-teta10; dif2=teta2-teta10;
    if abs(dif1)>2*%pi then
        dif1=dif1-2*%pi;
    end
    if abs(dif2)>2*%pi then
        dif2=dif2-2*%pi;
    end
    if abs(dif1)>abs(dif2) then
// if abs(teta1-teta10)>abs(teta2-teta10) then

```

```

        cccos1=cos1; cccos2=cos2;
        cos1=cccos2; cos2=cccos1;
        tteta1=teta1; tteta2=teta2;
        teta1=tteta2; teta2=tteta1;
    end
    disp([cos1,cos2],"cos1,cos2")
    if teta0==52 & j==2 then
        // pause
    end
    write(u2,[teta0,teta10,teta1,teta2,sen1,sen2,cos1,cos2,sign1,disc],'(10(e16.4))');
    if abs(a)>0.0001 then
        teta10=teta1;
    end

end

z11=complex(R21*cos1,R21*sen1);
z12=complex(R21*cos2,R21*sen2);
z21=r*z11+z; z22=r*z12+z;
xx11=real(z11); yy11=imag(z11); xx12=real(z12); yy12=imag(z12);
x11=real(z21); y11=imag(z21); x12=real(z22); y12=imag(z22);

disp([teta0,x11,y11,x12,y12,xx11,yy11,xx12,yy12],"teta0,x11,y11,x12,y12,xx11,yy11,xx12,yy12"
)
disp([sen1,sen2,cos1,cos2,ccos1,ccos2],"sen1,sen2,cos1,cos2,ccos1,ccos2")
// if teta0==47 then
// pause
// end

else
    // solución aproximada
    jy=1;
    // sen1=(-B)/(2*A);
    // sen2=(-B)/(2*A);
    sen1=b/sqrt(a^2+b^2);
    sen2=-sen1;
    // cos1=(c-b*sen1)/a;
    // cos2=(c-b*sen2)/a;
    cos1=sqrt(1-sen1^2);
    cos2=-sqrt(1-sen2^2);
    // determina la combinación de sen1 y cos1
    sen=sen1;
    cose=cos1;
    f1=a*cose+b*sen-c;
    cose=cos2;
    f2=a*cose+b*sen-c;
    sen=sen2;
    cose=cos1;
    f3=a*cose+b*sen-c;
    cose=cos2;

```

```

f4=a*cose+b*sen-c;
[m, k] = min([f1 f2 f3 f4]);
select k
case 1
    sen=sen1; cose=cos1;
case 2
    sen=sen1; cose=cos2;
case 3
    sen=sen2; cose=cos1;
case 4
    sen=sen2; cose=cos2;
end
disp([sen,cose],"sen,cose");
// equivalentes
sen1=sen; sen2=sen; cos1=cose; cos2=cose;
teta1=atan(sen,cose); teta2=teta1;
// sign1=-sign1;
write(u2,[teta0,teta10,teta1,teta2,sen1,sen2,cos1,cos2,sign1,disc],'(10(e16.4))');
z11=complex(R21*cose,R21*sen);

z21=r*z11+z;
xx11=real(z11); yy11=imag(z11); xx12=xx11; yy12=yy11;
x11=real(z21); y11=imag(z21); x12=x11; y12=y11;
// disp([sen1,cos1,sen2,cos2],"sen1,cos1,sen2,cos2");
// pause
end
disp([x11,y11,x12,y12],"x11,y11,x12,y12")
disp([xx11,yy11,xx12,yy12],"xx11,yy11,xx12,yy12")
//pause

disp([zR10,zR21,zR22],"zR10,zR21,zR22");
if j==3 then
    // pause
end
imdi=0;

N2N11r=x11-real(zR10); N2N11i=y11-imag(zR10);
N2N12r=x12-real(zR10); N2N12i=y12-imag(zR10);

N2N13r=xx11-real(zR10); N2N13i=yy11-imag(zR10);
N2N14r=xx12-real(zR10); N2N14i=yy12-imag(zR10);

xxx11=xx11; yyy11=yy11; xxx12=xx12; yyy12=yy12;
xx1=xx11; yy1=yy11; xx2=xx12; yy2=yy12;
rr=real(r); ri=imag(r);
//write(u3,[j,teta0,N2N11r,N2N11i,N2N12r,N2N12i,N2N13r,N2N13i,N2N14r,N2N14i,rr,ri],'(12(f16
.4))');
// llama función selección para tomar los puntos correctos
//if imdi==0 then
//[xx1,yy1,xx2,yy2,ind10,xxx1,yyy1,xxx2,yyy2,ind20]=seleccion(r,x11,x12,y11,y12,xx11,xx12,yy
11,yy12,R21,R22,zR10)

```

```

//else
    ind1=0; ind2=0; ind10=0; ind20=0;
//end

//disp([ind10,ind20,xx1,yy1,xx2,yy2,xxx1,yyy1,xxx2,yyy2],"ind10,ind20,xx1,yy1,xx2,yy2,xxx1,yyy
1,xxx2,yyy2")
disp([xx1,yy1,xx2,yy2],"xx1,yy1,xx2,yy2")
//disp([xx1,yy1,xx2,yy2],"xx1,yy1,xx2,yy2");
disp([teta0,R11,R12,R21,R22,R10],"teta0,R11,R12,R21,R22,R10")
//pause
if ind10==0 then

//disp([xx1,yy1,xx2,yy2],"selección: xx1,yy1,xx2,yy2");

// calcula coeficiente de influencia
zR0L21=complex(xx1,yy1); //zR0L22=complex(x1,y1);

// calcula los coeficientes de influencia
A11=(zR0L21-zR10)/W1; //A21=(zR0L22-zR10)/W2;
A11r=real(A11); A11i=imag(A11); //A21r=real(A21); A21i=imag(A21); else
    // A11r=0; A11i=0; A21r=0; A21i=0;
    A11r0=A11r; A11i0=A11i; //A21r0=A21r; A21i0=A21i;
end; // fin de if ind10
disp([A11r,A11i],"A11r,A11i");

if ind20==0 then

// calcula coeficiente de influencia
zR0L22=complex(xx2,yy2);

// calcula los coeficientes de influencia
A21=(zR0L22-zR10)/W2;
A21r=real(A21); A21i=imag(A21); else
    // A11r=0; A11i=0; A21r=0; A21i=0;
    A11r0=A11r; A11i0=A11i; A21r0=A21r; A21i0=A21i;
end; // fin de if ind20
disp([A11r,A11i,A21r,A21i],"A11r,A11i,A21r,A21i")
if j==3 then
    // pause
end
// checa si se mantiene abs(A)=constante durante la fase
absA11=abs(A11);
disp([teta0,AA11,absA11],"teta0,AA11,absA11")
if teta0==0 then
AA11=abs(A11); AA21=abs(A21); else
    // if abs(A11)<> AA11 then
    if abs(abs(A11)-AA11)>0.25 then
// if abs(abs(A11)-AA11)>0.01 then
        A0=A11; A11=A21; A21=A0;
        A11r=real(A11); A11i=imag(A11); A21r=real(A21); A21i=imag(A21);
        A11r0=A11r; A11i0=A11i; A21r0=A21r; A21i0=A21i;

```

```

    disp("intercambia A")
    disp([A11r,A11i,A21r,A21i],"A11r,A11i,A21r,A21i")
    // pause
end

end

// uniformiza los arreglos
//if indica==0 then
    if indica==0 & ind10==0 & ind20==0 then
        A11r0=A11r; A11i0=A11i; A21r0=A21r; A21i0=A21i;
        //A111r0=A111r; A111i0=A111i; A211r0=A211r; A211i0=A211i;
        disp([indica,A11r0],"indica,A11r0")
    end
    //if teta0==45 then
    // pause
    //end
    //if indica==1 then
    // [A11r,A11i,A21r,A21i,A111r,A111i,A211r,A211i]=uniforme(A11r,A11i,A21r,A21i,A111r,A111i,A
    // 211r,A211i,A11r0,A11i0,A21r0,A21i0,A111r0,A111i0,A211r0,A211i0)
    //end
    //indica=1;

//write(u6,[j,teta0,A11r,A11i,A21r,A21i,A111r,A111i,A211r,A211i]','(10(f16.4)');
// calcula N
//for i=1:80
    for i=1:1
        // eeO1=eO1+5*(i-1)*0.05
        eeO1=eO1;
        O1=complex(eeO1,0); zO2=O1; zO3=O1;

N=zR10-O1;
// calcula Wc
if ind10==0 then
    Wc1=-N/A11; Wc2=-N/A21; else
        Wc1=0; Wc2=0;
end
if ind20==0 then
//Wc11=-N/A111; Wc21=-N/A211; else
    // Wc11=0; Wc21=0;
end

// calcula offset2
//zO2=(zR0L22-r*zR0L21)/(1-r);
disp(zO2,"zO2");

O1x=real(O1); O1y=imag(O1); O2x=real(zO2); O2y=imag(zO2);
if ind10==0 then
R0L21r=real(zR0L21); R0L21i=imag(zR0L21);
R0L22r=real(zR0L22); R0L22i=imag(zR0L22);
end

```

```

if ind20==0 then
//zR00L21=zR0L21-O1; zR00L22=zR0L22-O1;
//R00L21r=real(zR00L21); R00L21i=imag(zR00L21);
//R00L22r=real(zR00L22); R00L22i=imag(zR00L22);
end
//A11=zR00L21/W1; A12=zR00L22/W2;

Nr=real(N); Ni=imag(N);
Wc1r=real(Wc1); Wc1i=imag(Wc1); Wc2r=real(Wc2); Wc2i=imag(Wc2);
//Wc11r=real(Wc11); Wc11i=imag(Wc11); Wc21r=real(Wc21); Wc21i=imag(Wc21);

if indica==0 then
Wc1r0=Wc1r; Wc1i0=Wc1i; Wc2r0=Wc2r; Wc2i0=Wc2i;
//Wc11r0=Wc11r; Wc11i0=Wc11i; Wc21r0=Wc21r; Wc21i0=Wc21i;
end
disp([indica,A11r0],"indica,A11r0")
jx=0;
if j==3 & teta0==136 then
// pause
end

// prueba de impresión
write(u9,[j,O1x,jx,jy,teta0,Nr,Ni,A11r,A11i,A21r,A21i,Wc1r,Wc1i,Wc2r,Wc2i],'(15(f16.4))');

indica=1;
A11r0=A11r; A11i0=A11i; A21r0=A21r; A21i0=A21i;

if indoff==0 then
//
[Wc1r,Wc1i,Wc2r,Wc2i,Wc11r,Wc11i,Wc21r,Wc21i]=uniformeoff(j,O1x,Wc1r,Wc1i,Wc2r,Wc2i,
Wc11r,Wc11i,Wc21r,Wc21i,WWc1r0,WWc1i0,WWc11r0,WWc11i0)
WWc1r0(j)=Wc1r; WWc1i0(j)=Wc1i; WWc2r0(j)=Wc2r; WWc2i0(j)=Wc2i;
//WWc11r0(j)=Wc11r; WWc11i0(j)=Wc11i; WWc21r0(j)=Wc21r; WWc21i0(j)=Wc21i;

indoff=1;
end; // fin de if indoff
Wc1r0=Wc1r; Wc1i0=Wc1i; Wc2r0=Wc2r; Wc2i0=Wc2i;
//Wc11r0=Wc11r; Wc11i0=Wc11i; Wc21r0=Wc21r; Wc21i0=Wc21i;
Wc1=complex(Wc1r,Wc1i); Wc2=complex(Wc2r,Wc2i); //Wc11=complex(Wc11r,Wc11i);
Wc21=complex(Wc21r,Wc21i);
// selecciona puntos de círculo
if teta0==0 then
Fx1(j)=Wc1r; Fy1(j)=Wc1i; Gx1(j)=Wc2r; Gy1(j)=Wc2i;
disp([Fx1,Fy1],"Fx1,Fy1")
end
if teta0==120 then
Fx2(j)=Wc1r; Fy2(j)=Wc1i; Gx2(j)=Wc2r; Gy2(j)=Wc2i;
end
if teta0==240 then

```



```

    Fx3(j)=Wc1r; Fy3(j)=Wc1i; Gx3(j)=Wc2r; Gy3(j)=Wc2i;
end
//write(u9,[j,O1x,jx,jy,teta0,Nr,Ni,A11r,A11i,A21r,A21i,Wc1r,Wc1i,Wc2r,Wc2i],'(15(f16.4))');

end; // fin de for i
//else
// write(u6,[k1,j,i,ind],'(4(f16.4))');

//end; // fin de if ind

endfunction
function [disc, discmas, discmen, jak]=mindisc(R21, R22, zR10, r, teta0)
    // determina si disc es un mínimo
    jak=0;
    zR10=complex(R10*cosd(teta0),R10*sind(teta0));
    z=(1-r)*zR10;
    real0=real(z); imag0=imag(z);
    rr=real(r); ri=imag(r);
    a=2*R21*(rr*real0+ri*imag0);
    b=2*R21*(rr*imag0-ri*real0);
    c=R22^2-R21^2*(rr^2+ri^2)-real0^2-imag0^2;
    A=a^2+b^2;
    B=-2*b*c;
    C=c^2-a^2;
    disc=B^2-4*A*C;

    teta0=teta0+1;
    zR10=complex(R10*cosd(teta0),R10*sind(teta0));
    z=(1-r)*zR10;
    real0=real(z); imag0=imag(z);
    rr=real(r); ri=imag(r);
    a=2*R21*(rr*real0+ri*imag0);
    b=2*R21*(rr*imag0-ri*real0);
    c=R22^2-R21^2*(rr^2+ri^2)-real0^2-imag0^2;
    A=a^2+b^2;
    B=-2*b*c;
    C=c^2-a^2;
    discmas=B^2-4*A*C;

    teta0=teta0-2;
    zR10=complex(R10*cosd(teta0),R10*sind(teta0));
    z=(1-r)*zR10;
    real0=real(z); imag0=imag(z);
    rr=real(r); ri=imag(r);
    a=2*R21*(rr*real0+ri*imag0);
    b=2*R21*(rr*imag0-ri*real0);
    c=R22^2-R21^2*(rr^2+ri^2)-real0^2-imag0^2;
    A=a^2+b^2;
    B=-2*b*c;
    C=c^2-a^2;

```

```

discmen=B^2-4*A*C;

if discmas>disc & discmen> disc then
    jak=1;
end
endfunction

function [x1, y1, x2, y2]=nuevo2(C1x, C1y, b)
    // if disc<0 then
    // x1=0; y1=0; x2=0; y2=0
    // solución aproximada, cruce de dos rectas
    x1=C1x*C1y*b/(C1x^2+C1y^2);
    y1=b/(1+C1x^2/C1y^2);
    x2=x1; y2=y1;
//end
endfunction
function [x1, y1, x2, y2]=nuevo(Cx1, Cy1, R22, Rp1)
    d=sqrt(C1x^2+C1y^2);
    if d<R22 then
        h=(d+Rp1+R22)/2; else
        h=(d-Rp1+R22)/2;
    end
    ang=atand(C1y,C1x);
    x1=h*cosd(ang); y1=h*sind(ang);
    x2=x1;y2=y1;
endfunction

function [Cx, Cy, R]=param_circulo(x1, y1, x2, y2, x3, y3)
    // pruebas círculo
    // Datos tres puntos determina los parámetros del círculo
    //x1=-11.7163; y1=-0.0126; x2=-6.2341; y2=0.0184; x3=-9,9657; y3=3.1147;
    A=[-2*x2+2*x1,-2*y2+2*y1,-2*x3+2*x1,-2*y3+2*y1];
    B=[-x2^2+x1^2-y2^2+y1^2;-x3^2+x1^2-y3^2+y1^2];
    de=det(A);
    disp([de,x1,y1,x2,y2,x3,y3],"de,x1,y1,x2,y2,x3,y3");
    //pause
    if abs(de)>0.000001 then
        D=inv(A)*B;
        Cx=D(1); Cy=D(2);
        R=sqrt((x1-Cx)^2+(y1-Cy)^2); else
        Cx=0; Cy=0; R=0;
    end

    disp([Cx,Cy,R],"Cx,Cy,R");
    disp("dentro de función param_circulo")
    disp([x1,y1,x2,y2,x3,y3,Cx,Cy,R],"x1,y1,x2,y2,x3,y3,Cx,Cy,R")

endfunction
function [xprom, yprom, xx12, yy12, xx13, yy13, xx23, yy23, dist, disc1, disc2,
disc3]=interseccion(Cx1, Cy1, Cx2, Cy2, Cx3, Cy3, R1, R2, R3)
    // pruebas círculo 2

```

```

// Datos tres círculos, encuentra las dos líneas de interseccion
//Cx1=-5.541529; Cy1=0.045436; R1=2.649218;
//Cx2=-6.830059; Cy2=1.60642; R2=3.60435;
//Cx3=-8.97717; Cy3=0.351564; R3=2.76323;
dist=1000; xprom=0; yprom=0;
disp([Cx1,Cy1,Cx2,Cy2,Cx3,Cy3,R1,R2,R3],"Cx1,Cy1,Cx2,Cy2,Cx3,Cy3,R1,R2,R3")
A=[2*(Cx1-Cx2),2*(Cy1-Cy2);2*(Cx1-Cx3),2*(Cy1-Cy3)];
B=[R2^2-R1^2+Cx1^2-Cx2^2+Cy1^2-Cy2^2;R3^2-R1^2+Cx1^2-Cx3^2+Cy1^2-Cy3^2];
disp(A,"A")
D=inv(A)*B;
xi=D(1); yi=D(2);

disp([xi,yi],"xi,yi");
// calcula puntos en círculos
a=(R2^2-R1^2+Cx1^2-Cx2^2+Cy1^2-Cy2^2)/(2*(Cy1-Cy2));
b=-(Cx1-Cx2)/(Cy1-Cy2);
c=(R3^2-R1^2+Cx1^2-Cx3^2+Cy1^2-Cy3^2)/(2*(Cy1-Cy3));
d=-(Cx1-Cx3)/(Cy1-Cy3);
e=(R3^2-R2^2+Cx2^2-Cx3^2+Cy2^2-Cy3^2)/(2*(Cy2-Cy3));
f=-(Cx2-Cx3)/(Cy2-Cy3);
// 1a cuadrática
a1=1+b^2; a2=-2*Cx1+2*b*(a-Cy1); a3=Cx1^2+(a-Cy1)^2-R1^2;
disc1=a2^2-4*a1*a3;
if disc1>=0 then
    x1=(-a2+sqrt(disc1))/(2*a1);
    x2=(-a2-sqrt(disc1))/(2*a1);
    y1=a+b*x1; y2=a+b*x2;
    rhs1=(x1-xi)^2+(y1-yi)^2; rhs2=(x2-xi)^2+(y2-yi)^2;
    xx12=x1; yy12=y1
    if rhs1>rhs2 then
        xx12=x2; yy12=y2;
    end
    disp([xx12,yy12],"xx12,yy12")
else
    disp(disc1,"disc1= no hay intersección entre círculo 1 y 2")
xx12=0; yy12=0;
end

// 2a cuadrática
a1=1+d^2; a2=-2*Cx1+2*d*(c-Cy1); a3=Cx1^2+(c-Cy1)^2-R1^2;
disc2=a2^2-4*a1*a3;
if disc2>=0 then
    x1=(-a2+sqrt(disc2))/(2*a1);
    x2=(-a2-sqrt(disc2))/(2*a1);
    y1=c+d*x1; y2=c+d*x2;
    rhs1=(x1-xi)^2+(y1-yi)^2; rhs2=(x2-xi)^2+(y2-yi)^2;
    xx13=x1; yy13=y1
    if rhs1>rhs2 then
        xx13=x2; yy13=y2;
    end
    disp([xx13,yy13],"xx13,yy13")

```

```

else
    disp(disc2,"disc2= no hay intersección entre círculo 1 y 3")
    xx13=0; yy13=0;
end

// 3a cuadrática
a1=1+f^2; a2=-2*Cx2+2*f*(e-Cy2); a3=Cx2^2+(e-Cy2)^2-R2^2;
disc3=a2^2-4*a1*a3;
xx23=0; yy23=0;
if disc3>=0 then
    x1=(-a2+sqrt(disc3))/(2*a1);
    x2=(-a2-sqrt(disc3))/(2*a1);
    y1=e+f*x1; y2=e+f*x2;
    rhs1=(x1-xi)^2+(y1-yi)^2; rhs2=(x2-xi)^2+(y2-yi)^2;
    xx23=x1; yy23=y1
    if rhs1>rhs2 then
        xx23=x2; yy23=y2;
    end
    disp([xx23,yy23],"xx23,yy23")
    // modificación 4 febrero 2021
else
    disp(disc3,"disc3= no hay intersección entre círculo 2 y 3")
    xx23=0; yy23=0;
end
// determina distancia entre interseccion de rectas y círculos
if disc1>=0 then
    if disc2>=0&disc3>=0 then
        xprom=(xx12+xx13+xx23)/3;
        yprom=(yy12+yy13+yy23)/3;
        dist=sqrt((xprom-xi)^2+(yprom-yi)^2);
    end;

end

if dist<1000 then
    // pause
end
endfunction

function [rhs11, rhs12, rhs21, rhs22, teta221, teta222]=checaR2(x1, y1, x2, y2, R22, Rp21,
C1x, C1y)
rhs11=x1^2+y1^2-R22^2; rhs12=(x1-C1x)^2+(y1-C1y)^2-Rp21^2;
rhs21=x2^2+y2^2-R22^2; rhs22=(x2-C1x)^2+(y2-C1y)^2-Rp21^2;
//x1=real(x1); y1=real(y1);x2=real(x2); y2=real(y2);
disp([x1,y1,x2,y2],"x1,y1,x2,y2...dentro de checaR2")
//z1=complex(x1,y1); z2=complex(x2,y2);
teta221=atand(y1,x1); teta222=atand(y2,x2);
endfunction

function [rhs11, rhs12, rhs21, rhs22, teta211, teta212]=checaRp21(x1, y1, x2, y2, R21, Rp22,
C1x, C1y)
rhs11=x1^2+y1^2-R21^2; rhs12=(x1-C1x)^2+(y1-C1y)^2-Rp22^2;

```

```

rhs21=x2^2+y2^2-R21^2; rhs22=(x2-C1x)^2+(y2-C1y)^2-Rp22^2;
//z1=complex(x1,y1); z2=complex(x2,y2);
//x1=real(x1); y1=real(y1);x2=real(x2); y2=real(y2);
teta211=atand(y1,x1); teta212=atand(y2,x2);
endfunction
function [rhs11, rhs21, rhs12, rhs22]=polarcheck(R21, R22, teta211, teta212, teta221,
teta222, r, R10, teta0)
z2=complex(R22*cosd(teta221),R22*sind(teta221));
z1=complex(R21*cosd(teta211),R21*sind(teta211));
z0=complex(R10*cosd(teta0),R10*sind(teta0));
zrh=z2-r*z1-(1-r)*z0;
rhs11=abs(zrh);
z2=complex(R22*cosd(teta222),R22*sind(teta222));
zrh=z2-r*z1-(1-r)*z0;
rhs21=abs(zrh);
z2=complex(R22*cosd(teta221),R22*sind(teta221));
z1=complex(R21*cosd(teta212),R21*sind(teta212));
zrh=z2-r*z1-(1-r)*z0;
rhs12=abs(zrh);
z2=complex(R22*cosd(teta222),R22*sind(teta222));
zrh=z2-r*z1-(1-r)*z0;
rhs22=abs(zrh);

endfunction
function [xx1, yy1, xx2, yy2, xxx1, yyy1, xxx2, yyy2]=seleccionpolar(teta211, teta212,
teta221, teta222, rhs11, rhs21, rhs12, rhs22, R21, R22)
xx1=0; xx2=0; xxx1=0; xxx2=0; yy1=0; yy2=0; yyy1=0; yyy2=0;
if rhs11<0.1 then
xx1=R21*cosd(teta211); yy1=R21*sind(teta211); xx2=R22*cosd(teta221);
yy2=R22*sind(teta221);
end
if rhs21<0.1 then
xx1=R21*cosd(teta211); yy1=R21*sind(teta211); xx2=R22*cosd(teta222);
yy2=R22*sind(teta222);

end
if rhs12<0.1 then
xxx1=R21*cosd(teta212); yyy1=R21*sind(teta212); xxx2=R22*cosd(teta221);
yyy2=R22*sind(teta221);
end
if rhs22<0.1 then
xxx1=R21*cosd(teta212); yyy1=R21*sind(teta212); xxx2=R22*cosd(teta222);
yyy2=R22*sind(teta222);
end
endfunction

function [A11r, A11i, A21r, A21i, A111r, A111i, A211r, A211i]=uniforme(A11r, A11i, A21r,
A21i, A111r, A111i, A211r, A211i, A11r0, A11i0, A21r0, A21i0, A111r0, A111i0, A211r0,
A211i0)
dif1=(A11r-A11r0)^2+(A11i-A11i0)^2+(A21r-A21r0)^2+(A21i-A21i0)^2;
dif2=(A111r-A11r0)^2+(A111i-A11i0)^2+(A211r-A21r0)^2+(A211i-A21i0)^2;

```

```

dif3=(A11r-A11r0)^2+(A11i-A11i0)^2+(A21r-A21r0)^2+(A21i-A21i0)^2;
dif4=(A111r-A111r0)^2+(A111i-A111i0)^2+(A211r-A211r0)^2+(A211i-A211i0)^2;
if dif1>dif2 then
    T11r=A11r; T11i=A11i; T21r=A21r; T21i=A21i;
    A11r=A111r; A11i=A111i; A21r=A211r; A21i=A211i;
    A111r=T11r; A111i=T11i; A211r=T21r; A211i=T21i;
end
endfunction
function [Wc1r, Wc1i, Wc2r, Wc2i, Wc11r, Wc11i, Wc21r, Wc21i, jx]=uniformeW(j, Wc1r,
Wc1i, Wc2r, Wc2i, Wc11r, Wc11i, Wc21r, Wc21i, Wc1r0, Wc1i0, Wc2r0, Wc2i0, Wc11r0,
Wc11i0, Wc21r0, Wc21i0)
// dif1=(Wc1r-Wc1r0)^2+(Wc1i-Wc1i0)^2+(Wc2r-Wc2r0)^2+(Wc2i-Wc2i0)^2;
dif2=(Wc1r-Wc21r0)^2+(Wc1i-Wc21i0)^2;
// dif3=(Wc11r-Wc1r0)^2+(Wc11i-Wc1i0)^2+(Wc21r-Wc2r0)^2+(Wc21i-Wc2i0)^2;
dif4=(Wc11r-Wc21r0)^2+(Wc11i-Wc21i0)^2;
dif1=(Wc1r-Wc1r0)^2+(Wc1i-Wc1i0)^2;
dif3=(Wc11r-Wc1r0)^2+(Wc11i-Wc1i0)^2;
//
write(u5,[j, Wc1r, Wc1i, Wc11r, Wc11i, Wc1r0, Wc1i0, Wc11r0, Wc11i0, dif1, dif2, dif3, dif4], '(13(f16.4))'
);
// if dif3>dif1 | dif4>dif2 then
if dif1>dif2&dif4>dif3 then
//if dif1>500 & dif4>500 then
    T11r=Wc1r; T11i=Wc1i; T21r=Wc2r; T21i=Wc2i;
    Wc1r=Wc11r; Wc1i=Wc11i; Wc2r=Wc21r; Wc2i=Wc21i;
    Wc11r=T11r; Wc11i=T11i; Wc21r=T21r; Wc21i=T21i;
end

// define jx que indica que hay que intercambiar toda la fase
jx=0;
ddif1=abs(Wc1r-Wc2r)+abs(Wc1i-Wc2i);
ddif2=abs(Wc1r-Wc11r)+abs(Wc1i-Wc11i);
if ddif2<ddif1 then
    jx=1;
end
// if dif1>dif2 then

// end

endfunction
function [Wc1r, Wc1i, Wc2r, Wc2i, Wc11r, Wc11i, Wc21r, Wc21i]=uniformeoff(j, O1x,
Wc1r, Wc1i, Wc2r, Wc2i, Wc11r, Wc11i, Wc21r, Wc21i, WWc1r0, WWc1i0, WWc11r0,
WWc11i0)
dif1=(Wc1r-WWc1r0(j))^2+(Wc1i-WWc1i0(j))^2;
dif2=(Wc1r-WWc11r0(j))^2+(Wc1i-WWc11i0(j))^2;
// dif3=(Wc11r-WWc1r0(j))^2+(Wc11i-WWc1i0(j))^2+(Wc21r-WWc2r0(j))^2+(Wc21i-
WWc2i0(j))^2;
// dif4=(Wc11r-WWc11r0(j))^2+(Wc11i-WWc11i0(j))^2+(Wc21r-WWc21r0(j))^2+(Wc21i-
WWc21i0(j))^2;
if j==1 then

```

```

//
write(u5,[j,O1x,Wc1r,Wc1i,Wc11r,Wc11i,WWc1r0(j),WWc1i0(j),WWc11r0(j),WWc11i0(j),dif1,dif2
],'(12(f16.4)'));
end

if dif2<dif1 then
    T11r=Wc1r; T11i=Wc1i; T21r=Wc2r; T21i=Wc2i;
    Wc1r=Wc11r; Wc1i=Wc11i; Wc2r=Wc21r; Wc2i=Wc21i;
    Wc11r=T11r; Wc11i=T11i; Wc21r=T21r; Wc21i=T21i;
end

endfunction

function [xx1, yy1, xx2, yy2, ind1, xxx1, yyy1, xxx2, yyy2, ind2]=seleccion(r, x11, x12, y11,
y12, xx11, xx12, yy11, yy12, R21, R22, zR10)
// ensaya x11,y11,xx11,yy11
ind1=1; ind2=1;
N2N11r=x11-real(zR10); N2N11i=y11-imag(zR10);
N2N12r=x12-real(zR10); N2N12i=y12-imag(zR10);
N2N13r=xx11-real(zR10); N2N13i=yy11-imag(zR10);
N2N14r=xx12-real(zR10); N2N14i=yy12-imag(zR10);
xx1=0; yy1=0; xx2=0; yy2=0; ind=1;
xxx1=0; yyy1=0; xxx2=0; yyy2=0;

[ind11]=ensayo(r,N2N11r,N2N11i,N2N13r,N2N13i)
disp(ind11,"ind11");
[ind12]=ensayo(r,N2N11r,N2N11i,N2N14r,N2N14i)
disp(ind12,"ind12");
[ind21]=ensayo(r,N2N12r,N2N12i,N2N13r,N2N13i)
disp(ind21,"ind21");
[ind22]=ensayo(r,N2N12r,N2N12i,N2N14r,N2N14i)
disp(ind22,"ind22");
if ind11==0 then
    xx1=xx11; yy1=yy11; xx2=x11; yy2=y11; ind1=0; else
    if ind12==0 then
        xx1=xx12; yy1=yy12; xx2=x11; yy2=y11; ind1=0;
    end
end
if ind21==0 then
    xxx1=xx11; yyy1=yy11; xxx2=x12; yyy2=y12; ind2=0; else
    if ind22==0 then
        xxx1=xx12; yyy1=yy12; xxx2=x12; yyy2=y12; ind2=0;
    end
end

endfunction
function [ind11]=ensayo(r, N2N11r, N2N11i, N2N13r, N2N13i)
//N2N11r=real(N2N11r); N2N11i=real(N2N11i); N2N13r=real(N2N13r); N2N13i=real(N2N13i);
z1=complex(N2N11r,N2N11i); z2=complex(N2N13r,N2N13i);
z=z1/z2;

```

```

ind1=1; ind2=1; ind11=1;
//if abs(real(z)-real(r))<1 then
  if abs(real(z)-real(r))<0.05 then
    ind1=0;
  end
//if abs(imag(z)-imag(r))<1 then
  if abs(imag(z)-imag(r))<0.05 then
    ind2=0;
  end
if ind1==0 & ind2==0 then
  ind11=0;
end;
endfunction

```

```

function [Wc1r, Wc1i, Wc2r, Wc2i, Wc11r, Wc11i, Wc21r, Wc21i]=intercambio(jx1, jx2, jx3,
sum1r, sum1i, sum2r, sum2i, sum3r, sum3i, sum4r, sum4i, sum5r, sum5i, sum6r, sum6i)

```

```

  Wc1r=(sum1r+sum3r+sum5r)/(i1+i3+i5); Wc1i=(sum1i+sum3i+sum5i)/(i1+i3+i5);
  Wc2r=(sum2r+sum4r+sum6r)/(i2+i4+i6); Wc2i=(sum2i+sum4i+sum6i)/(i2+i4+i6);
  Wc11r=(sum11r+sum31r+sum51r)/(i1+i3+i5); Wc11i=(sum11i+sum31i+sum51i)/(i1+i3+i5);
  Wc21r=(sum21r+sum41r+sum61r)/(i2+i4+i6); Wc21i=(sum21i+sum41i+sum61i)/(i2+i4+i6);

```

```

if jx1==1 then

```

```

  Wc1r=(zum1r+sum3r+sum5r)/(i1+i3+i5); Wc1i=(zum1i+sum3i+sum5i)/(i1+i3+i5);
  Wc2r=(sum2r+sum4r+sum6r)/(i2+i4+i6); Wc2i=(sum2i+sum4i+sum6i)/(i2+i4+i6);
  Wc11r=(sum11r+sum31r+sum51r)/(i1+i3+i5); Wc11i=(sum11i+sum31i+sum51i)/(i1+i3+i5);
  Wc21r=(zum21r+sum41r+sum61r)/(i2+i4+i6); Wc21i=(zum21i+sum41i+sum61i)/(i2+i4+i6);
  dif1=abs(Wc1r-Wc2r)+abs(Wc1i+Wc2i);

```

```

if dif1>0.01 then

```

```

  Wc1r=(sum1r+sum3r+sum5r)/(i1+i3+i5); Wc1i=(sum1i+sum3i+sum5i)/(i1+i3+i5);
  Wc2r=(zum2r+sum4r+sum6r)/(i2+i4+i6); Wc2i=(zum2i+sum4i+sum6i)/(i2+i4+i6);
  Wc11r=(zum11r+sum31r+sum51r)/(i1+i3+i5); Wc11i=(zum11i+sum31i+sum51i)/(i1+i3+i5);
  Wc21r=(sum21r+sum41r+sum61r)/(i2+i4+i6); Wc21i=(sum21i+sum41i+sum61i)/(i2+i4+i6);
end
end

```

```

if jx2==1 then

```

```

  Wc1r=(sum1r+zum3r+sum5r)/(i1+i3+i5); Wc1i=(sum1i+zum3i+sum5i)/(i1+i3+i5);
  Wc2r=(sum2r+sum4r+sum6r)/(i2+i4+i6); Wc2i=(sum2i+sum4i+sum6i)/(i2+i4+i6);
  Wc11r=(sum11r+sum31r+sum51r)/(i1+i3+i5); Wc11i=(sum11i+sum31i+sum51i)/(i1+i3+i5);
  Wc21r=(sum21r+zum41r+sum61r)/(i2+i4+i6); Wc21i=(sum21i+zum41i+sum61i)/(i2+i4+i6);
  dif1=abs(Wc1r-Wc2r)+abs(Wc1i+Wc2i);

```

```

if dif1>0.01 then

```

```

  Wc1r=(sum1r+sum3r+sum5r)/(i1+i3+i5); Wc1i=(sum1i+sum3i+sum5i)/(i1+i3+i5);
  Wc2r=(sum2r+zum4r+sum6r)/(i2+i4+i6); Wc2i=(sum2i+zum4i+sum6i)/(i2+i4+i6);
  Wc11r=(sum11r+zum31r+sum51r)/(i1+i3+i5); Wc11i=(sum11i+zum31i+sum51i)/(i1+i3+i5);
  Wc21r=(sum21r+sum41r+sum61r)/(i2+i4+i6); Wc21i=(sum21i+sum41i+sum61i)/(i2+i4+i6);
end
end

```

```

if jx3==1 then

```

```

  Wc1r=(sum1r+sum3r+zum5r)/(i1+i3+i5); Wc1i=(sum1i+sum3i+zum5i)/(i1+i3+i5);
  Wc2r=(sum2r+sum4r+sum6r)/(i2+i4+i6); Wc2i=(sum2i+sum4i+sum6i)/(i2+i4+i6);

```



```

Wc11r=(sum11r+sum31r+sum51r)/(i1+i3+i5); Wc11i=(sum11i+sum31i+sum51i)/(i1+i3+i5);
Wc21r=(sum21r+sum41r+zum61r)/(i2+i4+i6); Wc21i=(sum21i+sum41i+zum61i)/(i2+i4+i6);
dif1=abs(Wc1r-Wc2r)+abs(Wc1i+Wc2i);
if dif1>0.01 then
Wc1r=(sum1r+sum3r+sum5r)/(i1+i3+i5); Wc1i=(sum1i+sum3i+sum5i)/(i1+i3+i5);
Wc2r=(sum2r+sum4r+zum6r)/(i2+i4+i6); Wc2i=(sum2i+sum4i+zum6i)/(i2+i4+i6);
Wc11r=(sum11r+sum31r+zum51r)/(i1+i3+i5); Wc11i=(sum11i+sum31i+zum51i)/(i1+i3+i5);
Wc21r=(sum21r+sum41r+sum61r)/(i2+i4+i6); Wc21i=(sum21i+sum41i+sum61i)/(i2+i4+i6);
end
end
endfunction

```

// función de escalación

// recibe los parámetros de los tres círculos de Wc1 (o de Wc2) y aplica procedimiento de escalación que obtiene un punto común de intersección

```

function [R3, disc]=escala(C1x, C1y, C2x, C2y, C3x, C3y, R1, R2, R30)

```

// itera R3 para buscar cuando disc>=0

```

ind=0; R3=R30;

```

```

disp(R30,"R30")

```

```

while ind==0 then

```

```

    a=R1^2-R2^2;

```

```

    b1=2*R1*(C1x-C2x);

```

```

    b2=2*R1*(C1y-C2y);

```

```

    c=(C1x-C2x)^2+(C1y-C2y)^2;

```

```

    aa=R1^2-R3^2;

```

```

    bb1=2*R1*(C1x-C3x);

```

```

    bb2=2*R1*(C1y-C3y);

```

```

    cc=(C1x-C3x)^2+(C1y-C3y)^2;

```

```

    A=(c/a)*aa^2;

```

```

    B=(1/a^2)*aa^3*cc;

```

```

    C=-(c/a^2)*aa^3-(1/a)*aa^2*cc;

```

```

    D=aa^2*cc^2+(c/a)^2*aa^4-2*(c/a)*aa^3*cc;

```

```

    g1=A*(bb2^2-bb1^2)+B*(b2^2-b1^2)+C*(b2*bb2-b1*bb1);

```

```

    g2=D+A*bb1^2+B*b1^2+C*b1*bb1;

```

```

    g3=2*A*bb1*bb2+2*B*b1*b2+C*(b1*bb2+b2*bb1);

```

```

    f1=g1^2+g3^2;

```

```

    f2=2*g1*g2-g3^2;

```

```

    f3=g2^2;

```

```

    disc=f2^2-4*f1*f3;

```

```

    disp([R3,disc],"dentro de función escala, R3,disc")

```

```

    if disc>=0 then

```

```

        ind=1;

```

```

    else

```

```

        if R3<R30/2 then

```

```

            ind=1;

```

```

        disp("no hay solución")
    end
end

R3=R3*0.999;
end; // fin de while

endfunction
function [Wcr1, Wci1, f1, ind1]=escalacion2(C1x, C1y, C2x, C2y, C3x, C3y, R1, R2, R3)
//C1x=6.538195833
//C1y=-3.251009167
//C2x=3.051614722
//C2y=-11.38284417
//C3x=-7.659611667
//C3y=-11.05737083
//R1=8.9071
//R2=4.8277
//R3=9.222413372
//R3=10.0955
disp("dentro de función escalación 2")
disp([C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3],"C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3")

// promedio de tres círculos
Wr=(C1x+C2x+C3x)/3; Wi=(C1y+C2y+C3y)/3;
disp([Wr,Wi],"promedio Wr,Wi")
// escala R3
//[R3,disc]=escala(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3)
//disp([R3,disc],"R3,disc")
f11_1=0; f11_2=0; f12_1=0; f12_2=0;
f21_1=0; f21_2=0; f22_1=0; f22_2=0;
f31_1=0; f31_2=0; f32_1=0; f32_2=0;
f41_1=0; f41_2=0; f42_1=0; f42_2=0;

a=R1^2-R2^2;
b1=2*R1*(C1x-C2x);
b2=2*R1*(C1y-C2y);
c=(C1x-C2x)^2+(C1y-C2y)^2;

aa=R1^2-R3^2;
bb1=2*R1*(C1x-C3x);
bb2=2*R1*(C1y-C3y);
cc=(C1x-C3x)^2+(C1y-C3y)^2;

A=(c/a)*aa^2;
B=(1/a^2)*aa^3*cc;
C=-(c/a^2)*aa^3-(1/a)*aa^2*cc;
D=aa^2*cc^2+(c/a)^2*aa^4-2*(c/a)*aa^3*cc;

g1=A*(bb2^2-bb1^2)+B*(b2^2-b1^2)+C*(b2*bb2-b1*bb1);
g2=D+A*bb1^2+B*b1^2+C*b1*bb1;
g3=2*A*bb1*bb2+2*B*b1*b2+C*(b1*bb2+b2*bb1);

```

```

f1=g1^2+g3^2;
f2=2*g1*g2-g3^2;
f3=g2^2;
disp([f1,f2,f3],"f1,f2,f3")
// calcula seno de teta1 con ecuación bi-cuadrática
x12=0; x22=0;
disc=f2^2-4*f1*f3;
disp(disc,"disc")
if disc>=0 then
    x12=(-f2+sqrt(disc))/(2*f1);
    x22=(-f2-sqrt(disc))/(2*f1);
    disp([x12,x22],"x12,x22")
end
if x12>=0 then
    x1=sqrt(x12); x2=-x1;
    disp([x1,x2],"x1,x2")
end
if x22>=0 then
    x3=sqrt(x22); x4=-x3;
    disp([x3,x4],"x3,x4")
end
// calcula f
// para x1:
cos1=sqrt(1-x1^2); cos2=-cos1;
b=b1*cos1+b2*x1;
disc1=b^2-4*a*c;
if disc1>=0 then
    f11_1=(-b+sqrt(disc1))/(2*a);
    f11_2=(-b-sqrt(disc1))/(2*a);
    disp([f11_1,f11_2],"f11_1,f11_2")
end

b=b1*cos2+b2*x1;
disc1=b^2-4*a*c;
if disc1>=0 then
    f12_1=(-b+sqrt(disc1))/(2*a);
    f12_2=(-b-sqrt(disc1))/(2*a);
    disp([f12_1,f12_2],"f12_1,f12_2")
end

// para x2:
cos1=sqrt(1-x2^2); cos2=-cos1;
disp("para f22_2")
disp([cos1,cos2],"cos1,cos2")
b=b1*cos1+b2*x2;
disc1=b^2-4*a*c;
if disc1>=0 then
    f21_1=(-b+sqrt(disc1))/(2*a);
    f21_2=(-b-sqrt(disc1))/(2*a);
    disp([f21_1,f21_2],"f21_1,f21_2")

```

```

end

b=b1*cos2+b2*x2;
disc1=b^2-4*a*c;
if disc1>=0 then
    f22_1=(-b+sqrt(disc1))/(2*a);
    f22_2=(-b-sqrt(disc1))/(2*a);
    disp([f22_1,f22_2],"f22_1,f22_2")
end

// para x3:
cos1=sqrt(1-x3^2); cos2=-cos1;
b=b1*cos1+b2*x3;
disc1=b^2-4*a*c;
if disc1>=0 then
    f31_1=(-b+sqrt(disc1))/(2*a);
    f31_2=(-b-sqrt(disc1))/(2*a);
    disp([f31_1,f31_2],"f31_1,f31_2")
end

b=b1*cos2+b2*x3;
disc1=b^2-4*a*c;
disp([disc1,"de f32_1, disc=")
if disc1>=0 then
    f32_1=(-b+sqrt(disc1))/(2*a);
    f32_2=(-b-sqrt(disc1))/(2*a);
    disp([f32_1,f32_2],"f32_1,f32_2")
end

// para x4:
cos1=sqrt(1-x4^2); cos2=-cos1;
b=b1*cos1+b2*x4;
disc1=b^2-4*a*c;
if disc1>=0 then
    f41_1=(-b+sqrt(disc1))/(2*a);
    f41_2=(-b-sqrt(disc1))/(2*a);
    disp([f41_1,f41_2],"f41_1,f41_2")
end

b=b1*cos2+b2*x4;
disc1=b^2-4*a*c;
if disc1>=0 then
    f42_1=(-b+sqrt(disc1))/(2*a);
    f42_2=(-b-sqrt(disc1))/(2*a);
    disp([f42_1,f42_2],"f42_1,f42_2")
end

// para f>0 calcula peso de balanceo
//write(u3," eO1,Wcr1,Wci1,f1,ind1,sen1,cos1,sen2,cos2,sen3,cos3");
Wcr1=0; Wci1=0; icon=0; dif1=100; f1=1; ccos1=0; ssen1=0; ind1=0;
disp([x1,x2,x3,x4],"x1,x2,x3,x4")

```

```

if f11_1>0 then
  Wcr=C1x+f11_1*R1*(sqrt(1-x1^2));
  Wci=C1y+f11_1*R1*x1;
  difr=Wr-Wcr; difi=Wi-Wci; icon=icon+1;
  dif=sqrt(difr^2+difi^2);
  disp([Wcr,Wci,difr,difi],"Wcr,Wci,difr,difi")
  cos1=sqrt(1-x1^2); sen1=x1;

[ind1,sen2,cos2,sen3,cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f11_1,cos1,sen1
)
  if ind1==0 then
    Wcr1=Wcr; Wci1=Wci; f1=f11_1; ccos1=sqrt(1-x1^2); ssen1=x1; eOO=f1*eO1;
    write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
    write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
  end

icon=icon+1;
end
disp([dif1,Wcr1,Wci1],"dif1,Wcr1,Wci1")
ind1=0;
if f11_2>0 then
  Wcr=C1x+f11_2*R1*(sqrt(1-x1^2));
  Wci=C1y+f11_2*R1*x1;
  difr=Wr-Wcr; difi=Wi-Wci;
  dif=sqrt(difr^2+difi^2);
  disp([Wcr,Wci,difr,difi],"Wcr,Wci,difr,difi")
  cos1=sqrt(1-x1^2); sen1=x1;

[ind1,sen2,cos2,sen3,cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f11_2,cos1,sen1
)
  if ind1==0 then
    Wcr1=Wcr; Wci1=Wci; f1=f11_2; ccos1=sqrt(1-x1^2); ssen1=x1; eOO=f1*eO1;
    write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
    write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
  end

icon=icon+1;
end
ind1=0;
if f12_1>0 then
  Wcr=C1x+f12_1*R1*(-sqrt(1-x1^2));
  Wci=C1y+f12_1*R1*x1;
  difr=Wr-Wcr; difi=Wi-Wci;
  dif=sqrt(difr^2+difi^2);
  disp([Wcr,Wci,difr,difi],"Wcr,Wci,difr,difi")
  cos1=-sqrt(1-x1^2); sen1=x1;

[ind1,sen2,cos2,sen3,cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f12_1,cos1,sen1
)

```

```

    if ind1==0 then
        Wcr1=Wcr; Wci1=Wci; f1=f12_1; ccos1=-sqrt(1-x1^2); ssen1=x1; eOO=f1*eO1;
        write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
        write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
    end

icon=icon+1;
end
ind1=0;
if f12_2>0 then
    Wcr=C1x+f12_2*R1*(-sqrt(1-x1^2));
    Wci=C1y+f12_2*R1*x1;
    difr=Wr-Wcr; difi=Wi-Wci;
    dif=sqrt(difr^2+difi^2);
    disp([Wcr,Wci,difr,difi],"Wcr,Wci,difr,difi")
    cos1=-sqrt(1-x1^2); sen1=x1;

[ind1, sen2, cos2, sen3, cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f12_2,cos1, sen1
)
    if ind1==0 then
        Wcr1=Wcr; Wci1=Wci; f1=f12_2; ccos1=-sqrt(1-x1^2); ssen1=x1; eOO=f1*eO1;
write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
    end
icon=icon+1;
end
ind1=0;
if f21_1>0 then
    Wcr=C1x+f21_1*R1*(sqrt(1-x2^2));
    Wci=C1y+f21_1*R1*x2;
    difr=Wr-Wcr; difi=Wi-Wci;
    dif=sqrt(difr^2+difi^2);
    disp([Wcr,Wci,difr,difi],"Wcr,Wci,difr,difi")
    cos1=sqrt(1-x2^2); sen1=x2;

[ind1, sen2, cos2, sen3, cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f21_1,cos1, sen1
)
    if ind1==0 then
        Wcr1=Wcr; Wci1=Wci; f1=f21_1; ccos1=sqrt(1-x2^2); ssen1=x2; eOO=f1*eO1;
write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
    end
icon=icon+1;
end
ind1=0;
if f21_2>0 then
    Wcr=C1x+f21_2*R1*(sqrt(1-x2^2));

```

```

Wci=C1y+f21_2*R1*x2;
difr=Wr-Wcr; difi=Wi-Wci;
dif=sqrt(difr^2+difi^2);
disp([Wcr,Wci,difr,difi],"Wcr,Wci,difr,difi")
cos1=sqrt(1-x2^2); sen1=x2;

[ind1, sen2, cos2, sen3, cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f21_2,cos1, sen1
)
    if ind1==0 then
        Wcr1=Wcr; Wci1=Wci; f1=f21_2; ccos1=sqrt(1-x2^2); ssen1=x2; eOO=f1*eO1;
write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
    end
icon=icon+1;
end
ind1=0;
if f22_1>0 then
    Wcr=C1x+f22_1*R1*(-sqrt(1-x2^2));
    Wci=C1y+f22_1*R1*x2;
    difr=Wr-Wcr; difi=Wi-Wci;
    dif=sqrt(difr^2+difi^2);
    disp([Wcr,Wci,difr,difi],"Wcr,Wci,difr,difi")
    cos1=-sqrt(1-x2^2); sen1=x2;

[ind1, sen2, cos2, sen3, cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f22_1,cos1, sen1
)
    if ind1==0 then
        Wcr1=Wcr; Wci1=Wci; f1=f22_1; ccos1=-sqrt(1-x2^2); ssen1=x2; eOO=f1*eO1;
write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
    end
icon=icon+1;
end
ind1=0;
if f22_2>0 then
    Wcr=C1x+f22_2*R1*(-sqrt(1-x2^2));
    Wci=C1y+f22_2*R1*x2;
    difr=Wr-Wcr; difi=Wi-Wci;
    dif=sqrt(difr^2+difi^2);
    disp([Wcr,Wci,difr,difi],"Wcr,Wci,difr,difi")
    cos1=-sqrt(1-x2^2); sen1=x2;

[ind1, sen2, cos2, sen3, cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f22_2,cos1, sen1
)
    if ind1==0 then
        Wcr1=Wcr; Wci1=Wci; f1=f22_2; ccos1=-sqrt(1-x2^2); ssen1=x2; eOO=f1*eO1;
write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)

```

```

    end
    icon=icon+1;
end
ind1=0;
if f31_1>0 then
    Wcr=C1x+f31_1*R1*(sqrt(1-x3^2));
    Wci=C1y+f31_1*R1*x3;
    difr=Wr-Wcr; difi=Wi-Wci;
    dif=sqrt(difr^2+difi^2);
    disp([Wcr,Wci,difr,difi],"Wcr,Wci,difr,difi")
    cos1=sqrt(1-x3^2); sen1=x3;

[ind1,sen2,cos2,sen3,cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f31_1,cos1,sen1
)
    if ind1==0 then
        Wcr1=Wcr; Wci1=Wci; f1=f31_1; ccos1=sqrt(1-x3^2); ssen1=x3; eOO=f1*eO1;
write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
    end
    icon=icon+1;
end
ind1=0;
if f31_2>0 then
    Wcr=C1x+f31_2*R1*(sqrt(1-x3^2));
    Wci=C1y+f31_2*R1*x3;
    difr=Wr-Wcr; difi=Wi-Wci;
    dif=sqrt(difr^2+difi^2);
    disp([Wcr,Wci,difr,difi],"Wcr,Wci,difr,difi")
    cos1=sqrt(1-x3^2); sen1=x3;

[ind1,sen2,cos2,sen3,cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f31_2,cos1,sen1
)
    if ind1==0 then
        Wcr1=Wcr; Wci1=Wci; f1=f31_2; ccos1=sqrt(1-x3^2); ssen1=x3; eOO=f1*eO1;
write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
    end
    icon=icon+1;
end
ind1=0;
if f32_1>0 then
    Wcr=C1x+f32_1*R1*(-sqrt(1-x3^2));
    Wci=C1y+f32_1*R1*x3;
    difr=Wr-Wcr; difi=Wi-Wci;
    dif=sqrt(difr^2+difi^2);
    disp([Wcr,Wci,difr,difi],"Wcr,Wci,difr,difi")
    cos1=-sqrt(1-x3^2); sen1=x3;

```



```

[ind1,sen2,cos2,sen3,cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f32_1,cos1,sen1
)
  if ind1==0 then
    Wcr1=Wcr; Wci1=Wci; f1=f32_1; ccos1=-sqrt(1-x3^2); ssen1=x3; eOO=f1*eO1;
write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
  end
icon=icon+1;
end
ind1=0;
if f32_2>0 then
  Wcr=C1x+f32_2*R1*(-sqrt(1-x3^2));
  Wci=C1y+f32_2*R1*x3;
  difr=Wr-Wcr; difi=Wi-Wci;
  dif=sqrt(difr^2+difi^2);
  disp([Wcr,Wci,difr,difi],"Wcr,Wci,difr,difi")
  cos1=-sqrt(1-x3^2); sen1=x3;

[ind1,sen2,cos2,sen3,cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f32_2,cos1,sen1
)
  if ind1==0 then
    Wcr1=Wcr; Wci1=Wci; f1=f32_2; ccos1=-sqrt(1-x3^2); ssen1=x3; eOO=f1*eO1;
write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
  end
icon=icon+1;
end
ind1=0;
if f41_1>0 then
  Wcr=C1x+f41_1*R1*(sqrt(1-x4^2));
  Wci=C1y+f41_1*R1*x4;
  difr=Wr-Wcr; difi=Wi-Wci;
  dif=sqrt(difr^2+difi^2);
  disp([Wcr,Wci,difr,difi],"Wcr,Wci,difr,difi")
  cos1=sqrt(1-x4^2); sen1=x4;

[ind1,sen2,cos2,sen3,cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f41_1,cos1,sen1
)
  if ind1==0 then
    Wcr1=Wcr; Wci1=Wci; f1=f41_1; ccos1=sqrt(1-x4^2); ssen1=x4; eOO=f1*eO1;
write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
  end
icon=icon+1;
end
ind1=0;
if f41_2>0 then

```

```

Wcr=C1x+f41_2*R1*(sqrt(1-x4^2));
Wci=C1y+f41_2*R1*x4;
difr=Wr-Wcr; difi=Wi-Wci;
dif=sqrt(difr^2+difi^2);
disp([Wcr,Wci,difr,difi,"Wcr,Wci,difr,difi")
cos1=sqrt(1-x4^2); sen1=x4;

```

```

[ind1,sen2,cos2,sen3,cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f41_2,cos1,sen1
)

```

```

    if ind1==0 then
        Wcr1=Wcr; Wci1=Wci; f1=f41_2; ccos1=sqrt(1-x4^2); ssen1=x4; eOO=f1*eO1;
write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
    end
icon=icon+1;
end
ind1=0;

```

```

if f42_1>0 then
    Wcr=C1x+f42_1*R1*(-sqrt(1-x4^2));
    Wci=C1y+f42_1*R1*x4;
    difr=Wr-Wcr; difi=Wi-Wci;
    dif=sqrt(difr^2+difi^2);
    disp([Wcr,Wci,difr,difi,"Wcr,Wci,difr,difi")
    cos1=-sqrt(1-x4^2); sen1=x4;

```

```

[ind1,sen2,cos2,sen3,cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f42_1,cos1,sen1
)

```

```

    if ind1==0 then
        Wcr1=Wcr; Wci1=Wci; f1=f42_1; ccos1=-sqrt(1-x4^2); ssen1=x4; eOO=f1*eO1;
write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
    end
icon=icon+1;
end
ind1=0;

```

```

if f42_2>0 then
    Wcr=C1x+f42_2*R1*(-sqrt(1-x4^2));
    Wci=C1y+f42_2*R1*x4;
    difr=Wr-Wcr; difi=Wi-Wci;
    dif=sqrt(difr^2+difi^2);
    disp([Wcr,Wci,difr,difi,"Wcr,Wci,difr,difi")
    cos1=-sqrt(1-x4^2); sen1=x4;

```

```

[ind1,sen2,cos2,sen3,cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f42_2,cos1,sen1
)

```

```

    if ind1==0 then
        Wcr1=Wcr; Wci1=Wci; f1=f42_2; ccos1=-sqrt(1-x4^2); ssen1=x4; eOO=f1*eO1;
write(u3,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');
write(u7,[eO1,Wcr1,Wci1,f1,eOO],'(5(f16.4))');

```

```

[jfase]=sintetiza(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1)
    end
icon=icon+1;
end

// selección del peso de prueba según el promedio de centros
disp('***** peso de balanceo *****')
disp([Wcr1,Wci1,dif1,f1,ccos1,ssen1],"Wcr1,Wci1,dif1,f1,ccos1,ssen1")

// checa las tres fases
cos2=(C1x-C2x+f1*R1*ccos1)/(f1*R2);
sen2=(C1y-C2y+f1*R1*ssen1)/(f1*R2);
cos3=(C1x-C3x+f1*R1*ccos1)/(f1*R3);
sen3=(C1y-C3y+f1*R1*ssen1)/(f1*R3);
[ind1,sen2,cos2,sen3,cos3]=checatrigo(C1x,C1y,C2x,C2y,C3x,C3y,R1,R2,R3,f1,ccos1,ssen1)
if ind1==1 then
    disp('***** no es solución *****')
end
teta1=atan(ssen1,ccos1)*180/%pi;
teta2=atan(sen2,cos2)*180/%pi;
teta3=atan(sen3,cos3)*180/%pi;

disp([teta1,teta2,teta3],"teta1,teta2,teta3")
Wcr1=C1x+f1*R1*ccos1; Wci1=C1y+f1*R1*ssen1;
Wcr2=C2x+f1*R2*cos2; Wci2=C2y+f1*R2*sen2;
Wcr3=C3x+f1*R3*cos3; Wci3=C3y+f1*R3*sen3;
disp([C3x,C3y,f1,R3,cos3,sen3],"C3x,C3y,f1,R3,cos3,sen3")
disp([Wcr1,Wci1],"Wcr1,Wci1")
disp([Wcr2,Wci2],"Wcr2,Wci2")
disp([Wcr3,Wci3],"Wcr3,Wci3")
disp([Wcr1,Wci1,Wcr2,Wci2,Wcr3,Wci3],"Wcr1,Wci1,Wcr2,Wci2,Wcr3,Wci3")
disp([sen2,cos2,sen3,cos3],"sen2,cos2,sen3,cos3")
disp([R1,R3,C1x,C1y,C3x,C3y,f1,ssen1,ccos1],"R1,R3,C1x,C1y,C3x,C3y,f1,ssen1,ccos1")
endfunction

function [ind1, sen2, cos2, sen3, cos3]=checatrigo(C1x, C1y, C2x, C2y, C3x, C3y, R1, R2,
R3, f1, ccos1, ssen1)
ind1=0;
cos2=(C1x-C2x+f1*R1*ccos1)/(f1*R2);
sen2=(C1y-C2y+f1*R1*ssen1)/(f1*R2);
cos3=(C1x-C3x+f1*R1*ccos1)/(f1*R3);
sen3=(C1y-C3y+f1*R1*ssen1)/(f1*R3);
disp([ssen1,ccos1],"ssen1,ccos1")
disp([sen2,cos2,sen3,cos3],"sen2,cos2,sen3,cos3")
if abs(sen2)>1 then
    ind1=1;
end
if abs(sen3)>1 then
    ind1=1;
end
if abs(cos2)>1 then

```

```

    ind1=1;
end
if abs(cos3)>1 then
    ind1=1;
end
if abs(sen2^2+cos2^2-1)>0.01 then
    ind1=1
end
if abs(sen3^2+cos3^2-1)>0.01 then
    ind1=1
end
disp([ind1],"ind1")
endfunction
function [jfase]=sintetiza(Cx11, Cy11, Cx21, Cy21, Cx31, Cy31, R11, R21, R31, f1)
    write(u3," círculos sintetizados")
    write(u3," jfase,teta,x,y");

    jfase=1;
    for jj=1:360
        teta=jj-1;
        x1=Cx11+f1*R11*cosd(teta);
        y1=Cy11+f1*R11*sind(teta);
        write(u3,[jfase,teta,x1,y1],'(4(f16.4))');
    end

    jfase=2;
    for jj=1:360
        teta=jj-1;
        x1=Cx21+f1*R21*cosd(teta);
        y1=Cy21+f1*R21*sind(teta);
        write(u3,[jfase,teta,x1,y1],'(4(f16.4))');
    end

    jfase=3;
    for jj=1:360
        teta=jj-1;
        x1=Cx31+f1*R31*cosd(teta);
        y1=Cy31+f1*R31*sind(teta);
        write(u3,[jfase,teta,x1,y1],'(4(f16.4))');
    end
endfunction

function [reL_min, reL_max]=rango(C1x, C1y, R1, RR10_min, RR10, RR10_max, RR20_min,
RR20, RR20_max, RR30_min, RR30, RR30_max, j)
// define rango por cada fase
// fase 1
// promedio RR10, RR11, RR12
reL1_min=RR10/RR10_min; reL1_max=RR10/RR10_max;
reL2_min=RR20/RR20_min; reL2_max=RR20/RR20_max;
reL3_min=RR30/RR30_min; reL3_max=RR30/RR30_max;

```

```

disp([reL1_min,reL1_max],"reL1_min,reL1_max")
disp([reL2_min,reL2_max],"reL2_min,reL2_max")
disp([reL3_min,reL3_max],"reL3_min,reL3_max")

[m, k] = min([reL1_min reL2_min reL3_min])
reL_min=m;
[m, k] = max([reL1_max reL2_max reL3_max])
reL_max=m;
disp([reL_min,reL_max],"reL_min,reL_max")

select j
  case 1 then
write(u3," R,k,x,y");
  case 2 then
  write(u4," R,k,x,y");
  case 3 then
  write(u5," R,k,x,y");
end

for i=1:6
  R=R1+(i-1)*R1*(reL_min-1)/5;
  for k=1:360
    x=C1x+R*cosd(k); y=C1y+R*sind(k);
    select j
      case 1 then
write(u3,[R,k,x,y],'(4(f16.4))');
      case 2 then
  write(u4,[R,k,x,y],'(4(f16.4))');
      case 3 then
  write(u5,[R,k,x,y],'(4(f16.4))');
    end
  end
end

  end
end
for i=1:5
  R=R1+i*R1*(reL_max-1)/5;
  for k=1:360
    x=C1x+R*cosd(k); y=C1y+R*sind(k);
    select j
      case 1 then
write(u3,[R,k,x,y],'(4(f16.4))');
      case 2 then
  write(u4,[R,k,x,y],'(4(f16.4))');
      case 3 then
  write(u5,[R,k,x,y],'(4(f16.4))');
    end
  end
end
end
endfunction

file('close',file())

```

```

u1=file('open','C:\scilab\prueba15.txt','unknown')
u2=file('open','C:\scilab\corrfa2.dat','unknown','formatted')
u4=file('open','C:\scilab\corrfa4.dat','unknown','formatted')
u6=file('open','C:\scilab\corrfa6.dat','unknown','formatted')
u9=file('open','C:\scilab\corrfa9.dat','unknown','formatted')
u3=file('open','C:\scilab\corrfa3.dat','unknown','formatted')
u$=file('open','C:\scilab\corrfa4.dat','unknown','formatted')

u8=file('open','C:\scilab\corrfa8.dat','unknown','formatted')
u7=file('open','C:\scilab\corrfa7.dat','unknown','formatted')
u5=file('open','C:\scilab\corrfa5.dat','unknown','formatted')

        labels5=["Wp1m";"Wp1ang"];
[ok,Wp1m,Wp1ang]=getvalue("Wp1m,Wp1ang",labels5,...
        list("vec",1,"vec",1),["";""])

        labels4=["Wp2m";"Wp2ang"];
[ok,Wp2m,Wp2ang]=getvalue("Wp2m,Wp2ang",labels4,...
        list("vec",1,"vec",1),["";""])
        labels3=["Offin"];
[ok,Offin]=getvalue("Offin",labels3,...
        list("vec",1),[""])

eO1=1; nint=2; nint2=nint^2; iflag=0;
nint=3; nint2=1;
//Wp1m=8; Wp1ang=15; Wp2m=12.1; Wp2ang=30;
W1=complex(Wp1m*cosd(Wp1ang),Wp1m*sind(Wp1ang));
W2=complex(Wp2m*cosd(Wp2ang),Wp2m*sind(Wp2ang));
//teta121=327; teta131=150; teta122=-73.51; teta132=144.74;

r=W2/W1;
// Lee datos

i=0; nint=3;
while i<nint &iflag==0;
    disp("antes de i=i+1;")
    i=i+1;
//for i=1:4
// load(u1,[x1,x2,x3]);
[B]=read(u1,1,9);
if B(1)==0 then
    iflag=1;
end
disp(iflag,"iflag=");
disp("B",[B(1),B(2),B(3),B(4),B(5),B(6)],B(7),B(8),B(9));
select i
case 1 then
    RR10=B(2); RR20=B(5); RR30=B(8); // valores promedio
    RR10_min=B(1); RR20_min=B(4); RR30_min=B(7); // valores mínimos
    RR10_max=B(3); RR20_max=B(6); RR30_max=B(9); // valores máximos

```

```

case 2 then
  RR11=B(2); RR21=B(5); RR31=B(8); // valores promedio
  RR11_min=B(1); RR21_min=B(4); RR31_min=B(7); // valores mínimos
  RR11_max=B(3); RR21_max=B(6); RR31_max=B(9); // valores máximos
case 3 then
  RR12=B(2); RR22=B(5); RR32=B(8); // valores promedio
  RR12_min=B(1); RR22_min=B(4); RR32_min=B(7); // valores mínimos
  RR12_max=B(3); RR22_max=B(6); RR32_max=B(9); // valores máximos
end; // fin de select

end; // fin de while
disp([RR10_min,RR10,RR10_max,RR20_min,RR20,RR20_max,RR30_min,RR30,RR30_max],"
RR10_min,RR10,RR10_max,RR20_min,RR20,RR20_max,RR30_min,RR30,RR30_max")
disp([RR11_min,RR11,RR11_max,RR21_min,RR21,RR21_max,RR31_min,RR31,RR31_max],"
RR11_min,RR11,RR11_max,RR21_min,RR21,RR21_max,RR31_min,RR31,RR31_max")
disp([RR12_min,RR12,RR12_max,RR22_min,RR22,RR22_max,RR32_min,RR32,RR32_max],"
RR12_min,RR12,RR12_max,RR22_min,RR22,RR22_max,RR32_min,RR32,RR32_max")

// Itera 10 valores de O1, O2, O3
//write(u6," eO1,Wcr1,Wci1,dist1,disc11,disc21,disc31,Wcr2,Wci2,dist2,disc12,disc22,disc32");
write(u9," j,O1x,jx,jy,teta0,Nr,Ni,A11r,A11i,A21r,A21i,Wc1r,Wc1i,Wc2r,Wc2i");

//write(u4," eO1,Wcr1,Wci1,dist1,,disc11,disc21,disc31,Wcr2,Wci2,dist2,disc12,disc22,disc32");
write(u8,"
eO1,Wc1r,Wc1i,Wc3r,Wc3i,Wc5r,Wc5i,Wc11r,Wc11i,Wc31r,Wc31i,Wc51r,Wc51i,Wcr,Wci");

//write(u5," eO1,Wc1r,Wc1i,Wc3r,Wc3i,Wc5r,Wc5i,Wcr,Wci,Wccr,Wcci");
write(u2," teta0,teta10,teta1,teta2,sen1,sen2,cos1,cos2,sign1");

Wc1r0=0; Wc1i0=0; Wc2r0=0; Wc2i0=0; Wc11r0=0; Wc11i0=0; Wc21r0=0; Wc21i0=0; AA11=0;
teta10=0; sign1=1; disc0=1; s10=1; s20=1;
for i=1:3
  WWc1r0(i)=0; WWc1r0(i)=0; WWc1i0(i)=0; WWc11r0(i)=0; WWc11i0(i)=0;
end
k=Offin-1; indica=0; Offend=Offin;
while k<Offend then
k=k+1;
eO1=k;
sum1r=0; sum1i=0; sum2r=0; sum2i=0; sum3r=0; sum3i=0; i1=0; i2=0; i3=0;
sum11r=0; sum11i=0; sum21r=0; sum21i=0; sum31r=0; sum31i=0;
sum4r=0; sum4i=0; sum5r=0; sum5i=0; sum6r=0; sum6i=0; i4=0; i5=0; i6=0;
sum41r=0; sum41i=0; sum51r=0; sum51i=0; sum61r=0; sum61i=0;
zum1r=0; zum1i=0; zum2r=0; zum2i=0; zum3r=0; zum3i=0;
zum11r=0; zum11i=0; zum21r=0; zum21i=0; zum31r=0; zum31i=0;
zum4r=0; zum4i=0; zum5r=0; zum5i=0; zum6r=0; zum6i=0;
zum41r=0; zum41i=0; zum51r=0; zum51i=0; zum61r=0; zum61i=0;
//for k=1,nint2
//select k
//case 1 then
  R10=RR10; R20=RR20; R30=RR30;
  R11=RR11; R21=RR21; R31=RR31;

```

```
R12=RR12; R22=RR22; R32=RR32;  
//end; // fin de select
```

```
for j=1:3  
indoff=0;
```

```
for k1=1:360  
teta0=(k1-1)*1;
```

```
select j  
case 1 then  
disp(indica,"indica")
```

```
[Wc1,Wc2,indica,A11r0,A11i0,A21r0,A21i0,Wc1r0,Wc1i0,Wc2r0,Wc2i0,indoff,Fx1,Fy1,Fx2,Fy2,  
Fx3,Fy3,Gx1,Gy1,Gx2,Gy2,Gx3,Gy3,jx1,jy1,AA11,teta10,sign1,disc0,s10,s20]=offset(i,k1,j,R11,  
R12,R10,r,W1,W2,teta0,eO1,indica,Wc1r0,Wc1i0,Wc2r0,Wc2i0,Wc21r0,Wc21i0,indoff,AA11,tet  
a10,sign1,disc0,s10,s20)
```

```
sum1r=sum1r+real(Wc1); sum1i=sum1i+imag(Wc1); i1=i1+1; //sum11r=sum11r+real(Wc11);  
sum11i=sum11i+imag(Wc11);  
sum2r=sum2r+real(Wc2); sum2i=sum2i+imag(Wc2); i2=i2+1; //sum21r=sum21r+real(Wc21);  
sum21i=sum21i+imag(Wc21);  
zum1r=zum1r+real(Wc2); zum1i=zum1i+imag(Wc2); //zum11r=zum11r+real(Wc21);  
zum11i=zum11i+imag(Wc21);  
zum2r=zum2r+real(Wc1); zum2i=zum2i+imag(Wc1); //zum21r=zum21r+real(Wc11);  
zum21i=zum21i+imag(Wc11);
```

```
if teta0==0 then  
F1x1=Fx1(j); F1y1=Fy1(j); G1x1=Gx1(j); G1y1=Gy1(j);  
end  
if teta0==120 then  
F1x2=Fx2(j); F1y2=Fy2(j); G1x2=Gx2(j); G1y2=Gy2(j);  
end  
if teta0==240 then  
F1x3=Fx3(j); F1y3=Fy3(j); G1x3=Gx3(j); G1y3=Gy3(j);  
end  
case 2 then
```

```
[Wc1,Wc2,indica,A11r0,A11i0,A21r0,A21i0,Wc1r0,Wc1i0,Wc2r0,Wc2i0,indoff,Fx1,Fy1,Fx2,Fy2,  
Fx3,Fy3,Gx1,Gy1,Gx2,Gy2,Gx3,Gy3,jx2,jy2,AA11,teta10,sign1,disc0,s10,s20]=offset(i,k1,j,R21,  
R22,R20,r,W1,W2,teta0,eO1,indica,Wc1r0,Wc1i0,Wc2r0,Wc2i0,Wc21r0,Wc21i0,indoff,AA11,tet  
a10,sign1,disc0,s10,s20)
```

```
sum3r=sum3r+real(Wc1); sum3i=sum3i+imag(Wc1); i3=i3+1; //sum31r=sum31r+real(Wc11);  
sum31i=sum31i+imag(Wc11);  
sum4r=sum4r+real(Wc2); sum4i=sum4i+imag(Wc2); i4=i4+1; //sum41r=sum41r+real(Wc21);  
sum41i=sum41i+imag(Wc21);  
zum3r=zum3r+real(Wc2); zum3i=zum3i+imag(Wc2); //zum31r=zum31r+real(Wc21);  
zum31i=zum31i+imag(Wc21);  
zum4r=zum4r+real(Wc1); zum4i=zum4i+imag(Wc1); //zum41r=zum41r+real(Wc11);  
zum41i=zum41i+imag(Wc11);
```



```

if teta0==0 then
    F2x1=Fx1(j); F2y1=Fy1(j); G2x1=Gx1(j); G2y1=Gy1(j);
end
if teta0==120 then
    F2x2=Fx2(j); F2y2=Fy2(j); G2x2=Gx2(j); G2y2=Gy2(j);
end
if teta0==240 then
    F2x3=Fx3(j); F2y3=Fy3(j); G2x3=Gx3(j); G2y3=Gy3(j);
end
case 3 then

[Wc1,Wc2,indica,A11r0,A11i0,A21r0,A21i0,Wc1r0,Wc1i0,Wc2r0,Wc2i0,indoff,Fx1,Fy1,Fx2,Fy2,
Fx3,Fy3,Gx1,Gy1,Gx2,Gy2,Gx3,Gy3,jx3,jy3,AA11,teta10,sign1,disc0,s10,s20]=offset(i,k1,j,R31,
R32,R30,r,W1,W2,teta0,eO1,indica,Wc1r0,Wc1i0,Wc2r0,Wc2i0,Wc21r0,Wc21i0,indoff,AA11,tet
a10,sign1,disc0,s10,s20)

    sum5r=sum5r+real(Wc1); sum5i=sum5i+imag(Wc1); i5=i5+1; //sum51r=sum51r+real(Wc11);
sum51i=sum51i+imag(Wc11);
    sum6r=sum6r+real(Wc2); sum6i=sum6i+imag(Wc2); i6=i6+1; //sum61r=sum61r+real(Wc21);
sum61i=sum61i+imag(Wc21);
zum5r=zum5r+real(Wc2); zum5i=zum5i+imag(Wc2); //zum51r=zum51r+real(Wc21);
zum51i=zum51i+imag(Wc21);
zum6r=zum6r+real(Wc1); zum6i=zum6i+imag(Wc1); //zum61r=zum61r+real(Wc11);
zum61i=zum61i+imag(Wc11);

    if teta0==0 then
        F3x1=Fx1(j); F3y1=Fy1(j); G3x1=Gx1(j); G3y1=Gy1(j);
    end
    if teta0==120 then
        F3x2=Fx2(j); F3y2=Fy2(j); G3x2=Gx2(j); G3y2=Gy2(j);
    end
    if teta0==240 then
        F3x3=Fx3(j); F3y3=Fy3(j); G3x3=Gx3(j); G3y3=Gy3(j);
    end
end ; // fin de select
Wc1r=real(Wc1); Wc1i=imag(Wc1); Wc2r=real(Wc2); Wc2i=imag(Wc2); //Wc11r=real(Wc11);
Wc11i=imag(Wc11); Wc21r=real(Wc21); Wc21i=imag(Wc21);
// write(u7,j,Wc1r,Wc1i,Wc2r,Wc2i,Wc11r,Wc11i,Wc21r,Wc21i,sum1r,sum1i,'(11(f16.4))');
disp([A11r0,A11i0,A21r0,A21i0],"A11r0,A11i0,A21r0,A21i0")
disp([Wc1r,Wc1i,Wc2r,Wc2i],"Wc1r,Wc1i,Wc2r,Wc2i")

end; // fin de for k1
//end; // fin de for i
end; // fin de for j
write(u7,"indicadores jy1,jy2,jy3")
write(u7,[jy1,jy2,jy3],'(3(f16.4))');
write(u7," intersecciones comunes de círculos escalados")
write(u7," eO1,Wc1,Wc1i,f1,eOO");
// calcula los parámetros de círculos y la intersección por cada fase
disp([F1x1,F1y1,F1x2,F1y2,F1x3,F1y3],"F1x1,F1y1,F1x2,F1y2,F1x3,F1y3")

```

```

disp([F2x1,F2y1,F2x2,F2y2,F2x3,F2y3],"F2x1,F2y1,F2x2,F2y2,F2x3,F2y3")
disp([F3x1,F3y1,F3x2,F3y2,F3x3,F3y3],"F3x1,F3y1,F3x2,F3y2,F3x3,F3y3")
[Cx11,Cy11,R11]=param_circulo(F1x1,F1y1,F1x2,F1y2,F1x3,F1y3)
[Cx21,Cy21,R21]=param_circulo(F2x1,F2y1,F2x2,F2y2,F2x3,F2y3)
[Cx31,Cy31,R31]=param_circulo(F3x1,F3y1,F3x2,F3y2,F3x3,F3y3)
// promedio de tres círculos, primera opción
Wr=(Cx11+Cx21+Cx31)/3; Wi=(Cy11+Cy21+Cy31)/3;

//disp([Wr,Wi],"promedio  Wr,Wi")

//write(u3," peso promedio, primera opción")
//write(u3,[Wr,Wi],[2(f16.4)]);
// llama función de escalación2 que da la intersección en un punto común que es el peso de
balanceo
//write(u3," eO1,Wcr1,Wci1,f1,eOO");
//[Wcr1,Wci1,f1,ind1]=escalacion2(Cx11,Cy11,Cx21,Cy21,Cx31,Cy31,R11,R21,R31)
//fase 1
j=1;
[reL_min,reL_max]=rango(Cx11,Cy11,R11,RR10_min,RR10,RR10_max,RR20_min,RR20,RR20_max,RR30_min,RR30,RR30_max,j)
//fase 2
j=2;
[reL_min,reL_max]=rango(Cx21,Cy21,R21,RR11_min,RR11,RR11_max,RR21_min,RR21,RR21_max,RR31_min,RR31,RR31_max,j)
//fase 3
j=3;
[reL_min,reL_max]=rango(Cx31,Cy31,R31,RR12_min,RR12,RR12_max,RR22_min,RR22,RR22_max,RR32_min,RR32,RR32_max,j)

write(u7," ")

//[Wcr2,Wci2,xx12,yy12,xx13,yy13,xx23,yy23,dist2,disc12,disc22,disc32]=interseccion(Cx12,Cy12,Cx22,Cy22,Cx32,Cy32,R12,R22,R32)
[Cx12,Cy12,R12]=param_circulo(G1x1,G1y1,G1x2,G1y2,G1x3,G1y3)
[Cx22,Cy22,R22]=param_circulo(G2x1,G2y1,G2x2,G2y2,G2x3,G2y3)
[Cx32,Cy32,R32]=param_circulo(G3x1,G3y1,G3x2,G3y2,G3x3,G3y3)
disp([Cx12,Cy12,R12,Cx22,Cy22,R22,Cx32,Cy32,R32],"Cx12,Cy12,R12,Cx22,Cy22,R22,Cx32,Cy32,R32")
//fase 1
write(u3,"")
write(u4,"")
write(u5,"")
write(u3," *****")
write(u4," *****")
write(u5," *****")
write(u3,"")
write(u4,"")
write(u5,"")
j=1;
[reL_min,reL_max]=rango(Cx12,Cy12,R12,RR10_min,RR10,RR10_max,RR20_min,RR20,RR20_max,RR30_min,RR30,RR30_max,j)

```

```

//fase 2
j=2;
[reL_min,reL_max]=rango(Cx22,Cy22,R22,RR11_min,RR11,RR11_max,RR21_min,RR21,RR2
1_max,RR31_min,RR31,RR31_max,j)
//fase 3
j=3;
[reL_min,reL_max]=rango(Cx32,Cy32,R32,RR12_min,RR12,RR12_max,RR22_min,RR22,RR2
2_max,RR32_min,RR32,RR32_max,j)

//pause
//promedio de tres círculos, segunda opción
Wr=(Cx12+Cx22+Cx32)/3; Wi=(Cy12+Cy22+Cy32)/3;
write(u3, "*****")
disp([Wr,Wi], "promedio Wr,Wi")
//write(u3, " peso promedio, segunda opción")
//write(u3,[Wr,Wi], '(2(f16.4))');
// llama función de escalación2 que da la intersección en un punto común que es el peso de
balanceo
//write(u3, " eO1,Wcr2,Wci2,f2,eOO");
// [Wcr2,Wci2,f2,ind2]=escalacion2(Cx12,Cy12,Cx22,Cy22,Cx32,Cy32,R12,R22,R32)

//[Wcr1,Wci1,xx12,yy12,xx13,yy13,xx23,yy23,dist1,disc11,disc21,disc31]=interseccion(Cx11,Cy
11,Cx21,Cy21,Cx31,Cy31,R11,R21,R31)
//write(u3, " eO1,Wcr1,Wci1,dif1,f1,Wcr2,Wci2,f2,ind1,ind2");
//write(u3,[eO1,Wcr1,Wci1,f1,Wcr2,Wci2,f2,ind1,ind2], '(11(f16.4))');
// prepara datos para graficar círculos sintetizados
// offset escalados
f1=1; f2=1;
eeO1=eO1*f1; eeO2=eO1*f2;

//[jfase]=sintetiza(Cx11,Cy11,Cx12,Cy12,R11,R12,Cx21,Cy21,Cx22,Cy22,R21,R22,Cx31,Cy31,
Cx32,Cy32,R31,R32)

disp([R31,Cx31,Cy31,f1], "R3,C3x,C3y,f1")

Wc1r=sum1r/i1; Wc1i=sum1i/i1; Wc2r=sum2r/i2; Wc2i=sum2i/i2;
//Wc11r=sum11r/i1; Wc11i=sum11i/i1; Wc21r=sum21r/i2; Wc21i=sum21i/i2;
Wc3r=sum3r/i3; Wc3i=sum3i/i3; // Wc31r=sum31r/i3; Wc31i=sum31i/i3;
Wc4r=sum4r/i4; Wc4i=sum4i/i4; //Wc41r=sum41r/i4; Wc41i=sum41i/i4;
Wc5r=sum5r/i5; Wc5i=sum5i/i5; //Wc51r=sum51r/i5; Wc51i=sum51i/i5;
Wc6r=sum6r/i6; Wc6i=sum6i/i6; // Wc61r=sum61r/i6; Wc61i=sum61i/i6;

Wcr=(Wc1r+Wc3r+Wc5r)/3; Wci=(Wc1i+Wc3i+Wc5i)/3; //Wccr=(Wc11r+Wc31r+Wc51r)/3;
Wcci=(Wc11i+Wc31i+Wc51i)/3;
write(u8,[eO1,Wc1r,Wc1i,Wc3r,Wc3i,Wc5r,Wc5i,Wcr,Wci], '(9(f16.4))');
// promedio de las tres fases
// primero checa intercambio
//[Wc1r,Wc1i,Wc2r,Wc2i,Wc11r,Wc11i,Wc21r,Wc21i]=intercambio(jx1,jx2,jx3,sum1r,sum1i,sum
2r,sum2i,sum3r,sum3i,sum4r,sum4i,sum5r,sum5i,sum6r,sum6i)

WWc1r=(Wc1r+Wc3r+Wc5r)/3; WWc1i=(Wc1i+Wc3i+Wc5i)/3;

```

```

WWc2r=(Wc2r+Wc4r+Wc6r)/3; WWc2i=(Wc2i+Wc4i+Wc6i)/3;
if jy1==1 & jy2==0 & jy3==0 then
  WWc1r=(Wc3r+Wc5r)/2; WWc1i=(Wc3i+Wc5i)/2;
WWc2r=(Wc4r+Wc6r)/2; WWc2i=(Wc4i+Wc6i)/2;
end

if jy1==0 & jy2==1 & jy3==0 then
  WWc1r=(Wc1r+Wc5r)/2; WWc1i=(Wc1i+Wc5i)/2;
WWc2r=(Wc2r+Wc6r)/2; WWc2i=(Wc2i+Wc6i)/2;
end

if jy1==0 & jy2==0 & jy3==1 then
  WWc1r=(Wc1r+Wc3r)/2; WWc1i=(Wc1i+Wc3i)/2;
WWc2r=(Wc2r+Wc4r)/2; WWc2i=(Wc2i+Wc4i)/2;
end
write(u7," pesos promedio")
write(u7," eO1,Wc1r,Wc1i,Wc2r,Wc2i");
write(u7,[eO1,WWc1r,WWc1i,WWc2r,WWc2i],'(5(f16.4))');
end; // fi de while k

file('close',file())

```